

Unión Internacional de Telecomunicaciones

# UIT-T

SECTOR DE NORMALIZACIÓN  
DE LAS TELECOMUNICACIONES  
DE LA UIT

# H.263

(01/2005)

SERIE H: SISTEMAS AUDIOVISUALES Y  
MULTIMEDIOS

Infraestructura de los servicios audiovisuales –  
Codificación de imágenes vídeo en movimiento

---

**Codificación de vídeo para comunicación  
a baja velocidad binaria**

Recomendación UIT-T H.263

UIT-T



RECOMENDACIONES UIT-T DE LA SERIE H  
SISTEMAS AUDIOVISUALES Y MULTIMEDIOS

CARACTERÍSTICAS DE LOS SISTEMAS VIDEOTELEFÓNICOS	H.100–H.199
INFRAESTRUCTURA DE LOS SERVICIOS AUDIOVISUALES	
Generalidades	H.200–H.219
Multiplexación y sincronización en transmisión	H.220–H.229
Aspectos de los sistemas	H.230–H.239
Procedimientos de comunicación	H.240–H.259
<b>Codificación de imágenes vídeo en movimiento</b>	<b>H.260–H.279</b>
Aspectos relacionados con los sistemas	H.280–H.299
Sistemas y equipos terminales para los servicios audiovisuales	H.300–H.349
Arquitectura de servicios de directorio para servicios audiovisuales y multimedios	H.350–H.359
Arquitectura de la calidad de servicio para servicios audiovisuales y multimedios	H.360–H.369
Servicios suplementarios para multimedios	H.450–H.499
PROCEDIMIENTOS DE MOVILIDAD Y DE COLABORACIÓN	
Visión de conjunto de la movilidad y de la colaboración, definiciones, protocolos y procedimientos	H.500–H.509
Movilidad para los sistemas y servicios multimedios de la serie H	H.510–H.519
Aplicaciones y servicios de colaboración en móviles multimedios	H.520–H.529
Seguridad para los sistemas y servicios móviles multimedios	H.530–H.539
Seguridad para las aplicaciones y los servicios de colaboración en móviles multimedios	H.540–H.549
Procedimientos de interfuncionamiento de la movilidad	H.550–H.559
Procedimientos de interfuncionamiento de colaboración en móviles multimedios	H.560–H.569
SERVICIOS DE BANDA ANCHA Y DE TRÍADA MULTIMEDIOS	
Servicios multimedios de banda ancha sobre VDSL	H.610–H.619

*Para más información, véase la Lista de Recomendaciones del UIT-T.*

## Recomendación UIT-T H.263

### Codificación de vídeo para comunicación a baja velocidad binaria

#### Resumen

Esta Recomendación especifica una representación codificada, que se puede usar para comprimir el componente de imagen en movimiento de los servicios audiovisuales a velocidades binarias bajas. La configuración básica del algoritmo de codificación de fuente vídeo se basa en la Rec. UIT-T H.261, y es un híbrido de la predicción entre imágenes que utiliza la redundancia temporal y la codificación con transformada de la señal restante para reducir la redundancia espacial. El codificador de fuente puede funcionar con cinco formatos de fuente de vídeo normalizados: sub-cuarto de CIF, cuarto de CIF, CIF, 4 veces CIF y 16 veces CIF, y puede funcionar también utilizando una amplia gama de formatos de vídeo personalizados.

El decodificador tiene capacidad de compensación de movimiento, lo que permite la incorporación opcional de esta técnica en el codificador. En la compensación de movimiento se utiliza la precisión de mitad de píxel, mientras que en la Rec. UIT-T H.261 se utiliza la precisión de píxel entero y un filtro de bucle. Se emplea la codificación de longitud variable para la transmisión de los símbolos.

Además del algoritmo de codificación de fuente de vídeo básico, se incluyen dieciocho opciones de codificación negociables para mejorar el funcionamiento de la compresión y el soporte de capacidades adicionales. Se puede incluir también información suplementaria adicional en el tren de bits para mejorar la capacidad de visualización de imágenes y para uso externo.

En esta tercera edición de la Rec. UIT-T H.263 se incorporan los anexos U, V, W y X que fueron aprobados y publicados de manera separada, y se hacen algunas correcciones y aclaraciones:

- Corrección de la figura 8.
- Aclaración en el cuadro 1 relacionada con el cuadro de BPPmaxKb y el relleno de imagen.
- Aclaración en la cláusula 5.3.2 relacionada con el relleno de macrobloques antes de un código de comienzo.
- Aclaración relacionada con la interacción entre el anexo J/H.263 y el error por redondeo de la IDCT.
- Aclaración en el anexo N relacionada con el análisis sintáctico del campo GN/MBA en el BCM.
- Aclaración en el anexo O relacionada con MB de predicción bi-dir directa y la extrapolación de la imagen.
- Aclaración acerca del empleo de imágenes B con imágenes de referencia intra en el anexo O.
- Aclaración acerca de la utilización del anexo P junto con el anexo N.
- Corrección de la figura U.7 en el anexo U.

#### Orígenes

La Recomendación UIT-T H.263 fue aprobada el 13 de enero de 2005 por la Comisión de Estudio 16 (2005-2008) del UIT-T por el procedimiento de la Recomendación UIT-T A.8.

## PREFACIO

La UIT (Unión Internacional de Telecomunicaciones) es el organismo especializado de las Naciones Unidas en el campo de las telecomunicaciones. El UIT-T (Sector de Normalización de las Telecomunicaciones de la UIT) es un órgano permanente de la UIT. Este órgano estudia los aspectos técnicos, de explotación y tarifarios y publica Recomendaciones sobre los mismos, con miras a la normalización de las telecomunicaciones en el plano mundial.

La Asamblea Mundial de Normalización de las Telecomunicaciones (AMNT), que se celebra cada cuatro años, establece los temas que han de estudiar las Comisiones de Estudio del UIT-T, que a su vez producen Recomendaciones sobre dichos temas.

La aprobación de Recomendaciones por los Miembros del UIT-T es el objeto del procedimiento establecido en la Resolución 1 de la AMNT.

En ciertos sectores de la tecnología de la información que corresponden a la esfera de competencia del UIT-T, se preparan las normas necesarias en colaboración con la ISO y la CEI.

## NOTA

En esta Recomendación, la expresión "Administración" se utiliza para designar, en forma abreviada, tanto una administración de telecomunicaciones como una empresa de explotación reconocida de telecomunicaciones.

La observancia de esta Recomendación es voluntaria. Ahora bien, la Recomendación puede contener ciertas disposiciones obligatorias (para asegurar, por ejemplo, la aplicabilidad o la interoperabilidad), por lo que la observancia se consigue con el cumplimiento exacto y puntual de todas las disposiciones obligatorias. La obligatoriedad de un elemento preceptivo o requisito se expresa mediante las frases "tener que, haber de, hay que + infinitivo" o el verbo principal en tiempo futuro simple de mandato, en modo afirmativo o negativo. El hecho de que se utilice esta formulación no entraña que la observancia se imponga a ninguna de las partes.

## PROPIEDAD INTELECTUAL

La UIT señala a la atención la posibilidad de que la utilización o aplicación de la presente Recomendación suponga el empleo de un derecho de propiedad intelectual reivindicado. La UIT no adopta ninguna posición en cuanto a la demostración, validez o aplicabilidad de los derechos de propiedad intelectual reivindicados, ya sea por los miembros de la UIT o por terceros ajenos al proceso de elaboración de Recomendaciones.

En la fecha de aprobación de la presente Recomendación, la UIT ha recibido notificación de propiedad intelectual, protegida por patente, que puede ser necesaria para aplicar esta Recomendación. Sin embargo, debe señalarse a los usuarios que puede que esta información no se encuentre totalmente actualizada al respecto, por lo que se les insta encarecidamente a consultar la base de datos sobre patentes de la TSB.

© UIT 2005

Reservados todos los derechos. Ninguna parte de esta publicación puede reproducirse por ningún procedimiento sin previa autorización escrita por parte de la UIT.

# ÍNDICE

	<b>Página</b>
1 Alcance .....	1
2 Referencias .....	1
2.1 Referencias normativas .....	1
2.2 Referencias informativas .....	1
3 Especificación breve .....	2
3.1 Entrada y salida vídeo .....	2
3.2 Entrada y salida digital .....	2
3.3 Frecuencia de muestreo .....	2
3.4 Algoritmo de codificación de fuente .....	2
3.5 Velocidad binaria .....	6
3.6 Almacenamiento en tampón .....	6
3.7 Simetría de la transmisión .....	7
3.8 Tratamiento de los errores .....	7
3.9 Funcionamiento multipunto .....	7
4 Codificador de fuente .....	7
4.1 Formato de fuente .....	7
4.2 Algoritmo de codificación de fuente de vídeo .....	10
4.3 Control de la codificación .....	14
4.4 Actualización forzada .....	14
4.5 Alineación en byte de los códigos de comienzo .....	14
5 Sintaxis y semántica .....	15
5.1 Capa de imagen .....	23
5.2 Capa de grupo de bloques .....	34
5.3 Capa de macrobloque .....	35
5.4 Capa de bloque .....	43
6 Proceso de decodificación .....	47
6.1 Compensación de movimiento .....	47
6.2 Coeficientes de decodificación .....	49
6.3 Reconstrucción de los bloques .....	51
Anexo A – Especificación de la exactitud de la transformada inversa .....	51
Anexo B – Decodificador ficticio de referencia .....	52
Anexo C – Consideraciones sobre el funcionamiento multipunto .....	54
C.1 Petición de imagen congelada .....	54
C.2 Petición de actualización rápida .....	54
C.3 Liberación de imagen congelada .....	54
C.4 Multipunto de presencia continua (CPM) y múltiplex de vídeo .....	54

	<b>Page</b>
Anexo D – Modo vector de movimiento sin restricción.....	57
D.1    Vectores de movimiento en fronteras de imagen .....	57
D.2    Ampliación de la gama de vectores de movimiento.....	58
Anexo E – Modo codificación aritmética basada en sintaxis .....	60
E.1    Introducción.....	60
E.2    Especificación del codificador SAC.....	61
E.3    Especificación del decodificador SAC.....	62
E.4    Sintaxis .....	62
E.5    PSC_FIFO .....	63
E.6    Símbolos de la capa de encabezamiento .....	63
E.7    Símbolos de capa de macrobloque y bloque .....	64
E.8    Modelos SAC .....	65
Anexo F – Modo predicción avanzada .....	68
F.1    Introducción.....	68
F.2    Cuatro vectores de movimiento por macrobloque.....	68
F.3    Compensación de movimiento para bloques de luminancia superpuestos.....	69
Anexo G – Modo tramas PB.....	72
G.1    Introducción.....	72
G.2    Tramas PB y bloques INTRA.....	73
G.3    Capa de bloque .....	73
G.4    Cálculo de los vectores de imagen B y con trama PB.....	73
G.5    Predicción de un bloque B en una trama PB .....	74
Anexo H – Corrección de errores hacia adelante para señales de vídeo codificadas .....	76
H.1    Introducción.....	76
H.2    Alineación de trama para la corrección de errores .....	76
H.3    Código de corrección de errores.....	76
H.4    Plazo de reenganche para la alineación de trama del corrector de errores.....	76
Anexo I – Modo codificación INTRA avanzada .....	77
I.1    Introducción.....	77
I.2    Sintaxis .....	78
I.3    Proceso de decodificación .....	78
Anexo J – Modo filtro de desbloqueo.....	86
J.1    Introducción.....	86
J.2    Relación con los modos vector de movimiento sin restricción (UMV) y predicción avanzada (AP) (anexos D y F).....	86
J.3    Definición de filtro de borde de desbloqueo .....	87
Anexo K – Modo estructura en rebanada.....	90
K.1    Introducción.....	90
K.2    Estructura de la capa de rebanada .....	91

	<b>Page</b>
Anexo L – Especificación de la información sobre mejoras suplementarias .....	94
L.1    Introducción.....	94
L.2    Formato PSUPP .....	95
L.3    No hacer nada .....	95
L.4    Petición de congelación de toda la imagen.....	95
L.5    Petición de congelación de parte de la imagen.....	96
L.6    Petición de congelación y redimensionamiento de parte de la imagen .....	96
L.7    Petición de liberación de la parte de la imagen congelada .....	97
L.8    Rótulo de la instantánea de toda la imagen .....	97
L.9    Rótulo de la instantánea de parte de la imagen .....	97
L.10   Rótulo de comienzo de segmento de tiempo de vídeo .....	97
L.11   Rótulo de terminación de segmento de tiempo de vídeo.....	97
L.12   Rótulo de comienzo de segmento de refinamiento progresivo .....	98
L.13   Rótulo de terminación de segmento de refinamiento progresivo .....	98
L.14   Información de referencia de cromaticidad.....	98
L.15   Tipo de función ampliada.....	101
Anexo M – Modo tramas PB mejoradas.....	101
M.1    Introducción.....	101
M.2    Modos de predicción del macrobloque $B_{PB}$ .....	101
M.3    Cálculo de vectores para la predicción bidireccional de un macrobloque $B_{..}$ .....	102
M.4    Cuadro MODB .....	102
Anexo N – Modo selección de la imagen de referencia .....	103
N.1    Introducción.....	103
N.2    Algoritmo de codificación de fuente vídeo .....	104
N.3    Canal para mensajes de canal de retorno.....	105
N.4    Sintaxis .....	105
N.5    Proceso del decodificador.....	109
Anexo O – Modo escalabilidad temporal, SNR y espacial.....	109
O.1    Visión general.....	109
O.2    Orden de transmisión de las imágenes .....	113
O.3    Sintaxis de la capa de imagen.....	115
O.4    Sintaxis de la capa de macrobloque.....	116
O.5    Decodificación de vector de movimiento.....	120
O.6    Filtros de interpolación.....	120
Anexo P – Repetición de muestreo de la imagen de referencia.....	123
P.1    Introducción.....	123
P.2    Sintaxis .....	126
P.3    Algoritmo de repetición de muestreo .....	128
P.4    Ejemplo de implementación .....	132

	<b>Page</b>
P.5 Repetición de muestreo con factor de 4 .....	134
Anexo Q – Modo actualización de resolución reducida .....	137
Q.1 Introducción.....	137
Q.2 Procedimiento de decodificación .....	138
Q.3 Ampliación de la imagen referenciada .....	141
Q.4 Reconstrucción de vectores de movimiento .....	142
Q.5 Compensación de movimiento con superposición ampliada para luminancia .....	144
Q.6 Muestreo a velocidad superior del error de predicción reconstruido de resolución reducida.....	146
Q.7 Filtro de frontera de bloque .....	149
Anexo R – Modo decodificación de segmento independiente.....	151
R.1 Introducción.....	151
R.2 Funcionamiento del modo .....	151
R.3 Constricciones a la utilización.....	152
Anexo S – Modo códigos de longitud variable (VLC) INTER alternativos.....	153
S.1 Introducción.....	153
S.2 Códigos de longitud variable (VLC) INTER alternativos para coeficientes..	153
S.3 Códigos de longitud variable (VLC) INTRA alternativos para CBPY .....	154
Anexo T – Modo cuantificación modificada .....	154
T.1 Introducción.....	154
T.2 Autorización de DQUANT modificado .....	154
T.3 Tamaño de paso de cuantificación alterado para coeficientes de crominancia .....	155
T.4 Gama de coeficientes modificados.....	156
T.5 Restricciones de utilización.....	156
Anexo U – Modo de selección mejorada de imagen de referencia.....	157
U.1 Introducción.....	157
U.2 Algoritmo de codificación de fuente de vídeo .....	158
U.3 Sintaxis del canal hacia adelante .....	160
U.4 Proceso de decodificación .....	179
U.5 Mensajes en el canal de retorno.....	185
Anexo V – Modo rebanada particionada de datos .....	189
V.1 Introducción.....	189
V.2 Estructura de la partición de datos.....	189
V.3 Interacción con otros modos opcionales.....	192
Anexo W – Especificación de información adicional de mejora suplementaria .....	196
W.1 Introducción.....	196
W.2 Referencias .....	196
W.3 Valores adicionales de FTYPE.....	196



	<b>Page</b>
W.4 Número máximo recomendado de octetos PSUPP .....	197
W.5 IDCT de punto fijo .....	197
W.6 Mensaje de imagen .....	207
Anexo X – Definición de perfiles y niveles .....	212
X.1 Alcance .....	212
X.2 Perfiles de soporte del modo preferido .....	212
X.3 Formatos de imagen y frecuencias de reloj de imagen .....	217
X.4 Niveles de la capacidad de rendimiento .....	218
X.5 Definiciones de capacidad genérica para uso con la Rec. UIT-T H.245 .....	223
Apéndice I – Seguimiento de errores .....	227
I.1 Introducción .....	227
I.2 Seguimientos de errores .....	227
Apéndice II – Mejora opcional recomendada .....	228



# Recomendación UIT-T H.263

## Codificación de vídeo para comunicación a baja velocidad binaria

### 1 Alcance

Esta Recomendación especifica una representación codificada que se puede usar para comprimir el componente de imagen en movimiento de los servicios audiovisuales a velocidades binarias bajas. La configuración básica del algoritmo de codificación de fuente vídeo se basa en la Rec. UIT-T H.261. Se incluyen dieciocho opciones de codificación negociables para mejorar el funcionamiento y aumentar la funcionalidad. Esta Recomendación contiene la versión 2, que es totalmente compatible con la Rec. UIT-T H.263 original, añadiendo solamente características opcionales al contenido de la versión 1 original.

### 2 Referencias

#### 2.1 Referencias normativas

Las siguientes Recomendaciones del UIT-T y otras referencias contienen disposiciones que, mediante su referencia en este texto, constituyen disposiciones de la presente Recomendación. Al efectuar esta publicación, estaban en vigor las ediciones indicadas. Todas las Recomendaciones y otras referencias son objeto de revisiones, por lo que se preconiza que los usuarios de esta Recomendación investiguen la posibilidad de aplicar las ediciones más recientes de las Recomendaciones y otras referencias citadas a continuación. Se publica periódicamente una lista de las Recomendaciones UIT-T actualmente vigentes. En esta Recomendación, la referencia a un documento, en tanto que autónomo, no le otorga el rango de una Recomendación.

- [1] Recomendación UIT-R BT.601-5 (1995), *Parámetros de codificación de televisión digital para estudios con formatos de imagen normal 4:3 de pantalla ancha 16:9*.

Se remite aquí a la referencia [1] para definir el espacio cromático ( $Y$ ,  $C_B$ ,  $C_R$ ) y su representación a base de enteros de 8 bits de imágenes utilizadas por códecs de vídeo diseñados de conformidad con esta Recomendación. (La referencia [1] no se utiliza para definir ningún otro aspecto de esta Recomendación.)

#### 2.2 Referencias informativas

Las siguientes Recomendaciones UIT-T adicionales se mencionan en el texto a efectos ilustrativos y no constituyen disposiciones de esta Recomendación.

- [2] Recomendación UIT-T H.223 (2001), *Protocolo de multiplexación para comunicación multimedios a baja velocidad binaria*.
- [3] Recomendación UIT-T H.242 (2004), *Sistema para el establecimiento de comunicaciones entre terminales audiovisuales con utilización de canales digitales de hasta 2 Mbit/s*.
- [4] Recomendación UIT-T H.245 (2005), *Protocolo de control para comunicaciones multimedios*.
- [5] Recomendación UIT-T H.261 (1993), *Códec vídeo para servicios audiovisuales a  $p \times 64$  kbit/s*.
- [6] Recomendación UIT-T H.262 (2000) | ISO/CEI 13818-2:2000, *Tecnología de la información – Codificación genérica de imágenes en movimiento e información de audio asociada: Vídeo*.
- [7] Recomendación UIT-T H.324 (2002), *Terminal para comunicación multimedios a baja velocidad binaria*.

### 3 Especificación breve

En la figura 1 aparece un diagrama de bloques resumido del códec.

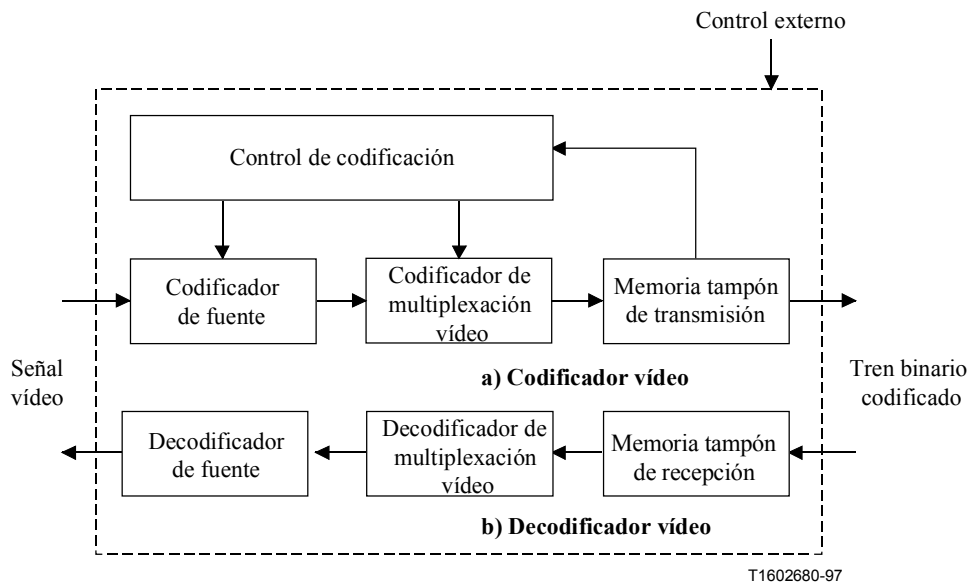


Figura 1/H.263 – Diagrama de bloques resumido del códec vídeo

#### 3.1 Entrada y salida vídeo

Para poder abarcar con una sola Recomendación la utilización dentro y entre regiones que emplean normas de televisión de 625 y 525 líneas, los formatos de fuente normalizados sobre los que el codificador de fuente actúa están basados en un formato intermedio común (CIF, *common intermediate format*). Con el empleo de la negociación externa (por ejemplo, aplicando la Rec. UIT-T H.245) se puede también facilitar la utilización de una amplia gama de formatos de fuente personalizados opcionales. Las normas de las señales de televisión de entrada y salida, que pueden, por ejemplo, ser compuestas o de componentes analógicas o digitales, no son objeto de Recomendaciones, como tampoco lo son los métodos para realizar cualquier conversión necesaria de y hacia el formato de codificación de fuente.

#### 3.2 Entrada y salida digital

El codificador vídeo proporciona un tren binario digital autocontenido que se puede combinar con otras señales multifacilidades (tal como se define en la Rec. UIT-T H.223, por ejemplo). El decodificador vídeo efectúa el proceso inverso.

#### 3.3 Frecuencia de muestreo

Las imágenes se muestrean a un múltiplo entero de la frecuencia de línea vídeo. Este reloj de muestreo y el reloj de red digital son asíncronos.

#### 3.4 Algoritmo de codificación de fuente

Se adopta un híbrido de predicción entre imágenes para utilizar redundancia temporal y codificación de la transformada de la señal restante para reducir la redundancia espacial. El decodificador tiene capacidad de compensación de movimiento, lo que permite la incorporación opcional de esta técnica en el codificador. Se utiliza una precisión de mitad de píxel para la compensación de movimiento, en vez de la precisión de píxel entero y filtro de bucle de la Rec. UIT-T H.261. En la transmisión de los símbolos se emplea la codificación de longitud variable.

Además del algoritmo de codificación básico de la presente Rec. UIT-T H.263 se pueden utilizar dieciocho opciones de codificación, juntas o por separado (sujeto a ciertas restricciones). También se puede incluir en el tren de bits información suplementaria adicional para mejorar la capacidad de visualización de imágenes y para uso externo. Se proporciona un método de corrección de errores hacia adelante para aplicarlo, cuando sea necesario, al tren de bits de vídeo resultante. La utilización de las opciones de codificación negociables, corrección de errores hacia adelante e información suplementaria se describe en las subcláusulas que siguen.

#### **3.4.1 Modo multipunto de presencia continua y modo multiplex de vídeo**

En este modo opcional se pueden enviar hasta cuatro "subtrenes de bits" de vídeo separados dentro del mismo canal vídeo. Esta característica se ha diseñado para que se utilice en aplicaciones multipunto de presencia continua o en otras situaciones en las que no se dispone de canales lógicos separados pero se desea la utilización de múltiples trenes de bits de vídeo (véase también el anexo C).

#### **3.4.2 Modo vector de movimiento sin restricción**

En este modo opcional, los vectores de movimiento están autorizados a apuntar fuera de la imagen. Los píxeles de borde se utilizan como predicción de los píxeles "no existentes". Con este modo se consigue una ganancia significativa cuando hay movimiento a través de los bordes de la imagen, especialmente con los formatos de imagen más pequeños (véase también el anexo D). Además, este modo incluye una extensión de la gama de vectores de movimiento de modo que se puedan utilizar vectores de movimiento mayores, lo que es especialmente útil en el caso de movimiento de la cámara y grandes formatos de imagen.

#### **3.4.3 Modo de codificación aritmética basada en sintaxis**

En este modo facultativo, se utiliza la codificación aritmética en vez de la codificación de longitud variable. La relación señal/ruido (SNR) y las imágenes reconstruidas serán las mismas, pero se producirá un número considerablemente menor de bits (véase también el anexo E).

#### **3.4.4 Modo predicción avanzada**

En este modo opcional, se utiliza la compensación de movimiento bloques superpuestos (OBMC, *overlapped block motion compensation*) para la parte de luminancia de las imágenes P (véase también el anexo F). En algunos macrobloques de la imagen, se utilizan cuatro vectores de  $8 \times 8$  en vez de un vector de  $16 \times 16$ . El codificador decidirá el tipo de vector que utilizará. Cuatro vectores utilizan más bits, pero dan una predicción mejor. La aplicación de este modo permite, en general, un perfeccionamiento considerable. Se consigue una ganancia subjetiva porque OBMC produce menos perturbaciones de bloqueo.

#### **3.4.5 Modo tramas PB**

Una trama PB consiste en dos imágenes que se codifican como una unidad. El nombre "PB" proviene del nombre de los tipos de imagen de la Rec. UIT-T H.262, en la que hay imágenes P e imágenes B. Por consiguiente, una trama PB consiste en una imagen P, que se predice a partir de la anterior imagen P decodificada, y una imagen B, que se predice a partir de la anterior imagen P decodificada y la imagen P que está siendo decodificada en ese momento. Se eligió la denominación imagen B porque partes de imágenes B se pueden predecir bidireccionalmente a partir de imágenes P pasadas y futuras. Con esta opción de codificación se consigue aumentar considerablemente el periodo de transmisión de la imagen sin aumentar sustancialmente la velocidad binaria (véase también el anexo G). No obstante, se proporciona también un modo tramas PB mejorado (véase además el anexo M). El modo tramas PB original se mantiene aquí sólo a efectos de compatibilidad con sistemas diseñados antes de la adopción del modo tramas PB mejorado.

### **3.4.6 Corrección errores hacia adelante**

Se especifica la utilización de un método de corrección de errores hacia adelante para proteger, cuando sea necesario, del tren de bits de vídeo. El método propuesto de corrección de errores hacia adelante es el mismo método de código BCH especificado también en la Rec. UIT-T H.261 (véase también el anexo H).

### **3.4.7 Modo codificación INTRA avanzada**

En este modo opcional, los bloques INTRA se predicen primero a partir de los bloques INTRA contiguos antes de la codificación (véase también el anexo I). Se definen tablas de códigos de longitud variable (VLC, *variable length code*) separadas para los bloques INTRA. La técnica se aplica a los macrobloques INTRA dentro de las imágenes INTRA y a los macrobloques INTRA dentro de las imágenes INTER. Es un modo que mejora considerablemente la característica de compresión en comparación con la codificación INTRA de la sintaxis básica de la presente Rec. UIT-T H.263.

### **3.4.8 Modo filtro de desbloqueo**

En este modo opcional, se aplica un filtro  $8 \times 8$  en los límites de los bordes de imágenes I y P decodificadas para reducir las perturbaciones de bloqueo (véase también el anexo J). El filtro tiene por objeto limitar la ocurrencia de perturbaciones de borde de bloque en la imagen decodificada. El filtro afecta a la imagen que se utiliza para la predicción de las imágenes subsiguientes y por ello permanece dentro del bucle de predicción de movimiento.

### **3.4.9 Modo estructura en rebanada**

En este modo opcional, una capa de "rebanada" sustituye a la capa de GOB de la sintaxis del tren de bits (véase también el anexo K). Este modo tiene por objeto proporcionar una capacidad mejorada de elasticidad a errores, hacer que el tren de bits sea más receptivo al uso con una entrega de transporte de paquetes subyacentes, y reducir al mínimo el retardo de vídeo. Una rebanada es similar a un grupo de bloques (GOB) en el sentido de que es una capa de la sintaxis que se halla entre la capa de imagen y la capa de macrobloque. Sin embargo, la utilización de la capa de rebanada permite una división flexible de la imagen, en contraste con la división fija y el orden de transmisión fijo que requiere la estructura GOB.

### **3.4.10 Información suplementaria sobre mejoras**

En el tren de bits se puede incluir información adicional suplementaria para señalar capacidad de visualización de imágenes mejorada o dar información para utilización externa (véase también el anexo L). Esta información suplementaria se puede emplear para señalar una petición de congelación o liberación de congelación de toda la imagen o de parte de la imagen con redimensionamiento o sin él, rotular imágenes o secuencias de imágenes concretas dentro del tren de vídeo para utilización externa, y transmitir información sobre referencia de cromaticidad para composición de vídeo. La información suplementaria puede estar presente en el tren de bits aun cuando el decodificador no sea capaz de proporcionar la capacidad mejorada de utilizarla o siquiera de interpretarla adecuadamente. Los decodificadores pueden desechar simplemente la información suplementaria salvo que se haya negociado por medios externos el requisito de proporcionar la capacidad pedida.

### **3.4.11 Modo tramas PB mejoradas**

Este modo opcional representa una mejora en comparación con la opción modo tramas PB (véanse también los anexos G y M). La diferencia principal entre ambos modos es que en el modo tramas PB mejoradas, cada bloque B se puede predecir hacia adelante utilizando un vector de movimiento separado o predecir hacia atrás con un vector cero. Esto mejora considerablemente la eficacia de la codificación en situaciones en las que los vectores P escalados en sentido descendente no son

buenos candidatos para predicción B. Esta predicción hacia atrás es particularmente útil cuando hay un corte de escena entre la trama P previa y la trama PB.

#### **3.4.12 Modo selección de imagen de referencia**

Se proporciona un modo opcional que mejora la calidad de funcionamiento de las comunicaciones vídeo en tiempo real por canales con tendencia a errores, permitiendo así la predicción temporal a partes de imágenes distintas de la imagen de referencia enviada más recientemente (véase también el anexo N). Este modo se puede utilizar con mensajes sobre el estado del canal de retorno que se devuelven al codificador para informarle de si su tren de bits se está recibiendo correctamente. En los entornos de canales con tendencia a errores, este modo permite al codificador optimizar su codificación vídeo para las condiciones del canal.

#### **3.4.13 Modo escalabilidad temporal, SNR y espacial**

En este modo opcional se admite la escalabilidad temporal, SNR y espacial (véase también el anexo O). Escalabilidad significa que un tren de bits se compone de una capa básica y de una o más capas de mejora asociadas. La capa básica es un tren de bits decodificable separadamente. Las capas de mejora se pueden decodificar junto con la capa básica para aumentar la calidad percibida incrementando el periodo de transmisión, la calidad o el tamaño de la imagen. La escalabilidad SNR se refiere a la información de mejora para aumentar la calidad de la imagen sin incrementar resolución de la imagen. La escalabilidad espacial se refiere a la información de mejora para aumentar la calidad de la imagen incrementando la resolución de la imagen en la dirección horizontal, en la dirección vertical, o en ambas. La utilización de imágenes B permite también soportar la escalabilidad temporal. Una imagen B es una mejora de escalabilidad que contiene imágenes que se pueden predecir bidireccionalmente a partir de dos imágenes en una capa de referencia, una temporalmente previa a la imagen vigente y otra temporalmente subsiguiente. Las imágenes B permiten utilizar la información de capa de mejora para aumentar la calidad percibida incrementando la velocidad de las imágenes de la secuencia de vídeo mejorado visualizada. Este modo puede ser útil para redes heterogéneas con capacidad de anchura de banda variable y también en combinación con esquemas de corrección de errores.

#### **3.4.14 Modo repetición de muestreo de imagen de referencia**

Se proporciona la sintaxis de soporte de un modo opcional por el que se procesa la imagen de referencia utilizada en la predicción de la imagen de vídeo mediante una operación de repetición del muestreo previa a su utilización en la formación del predictor de la imagen de entrada en curso (véase también el anexo P). Esto permite la selección dinámica y eficaz de una resolución de imagen apropiada para la codificación de vídeo, y también el alabeo de la imagen para utilizarlo como compensador de movimiento global o generador de efectos especiales.

#### **3.4.15 Modo actualización de resolución reducida**

Modo opcional que permite actualizaciones de resolución reducida de una imagen de referencia cuya resolución es más alta (véase también el anexo Q). Se prevé que éste sea el modo utilizado cuando se codifique una escena con mucha actividad, modo que permite a un codificador aumentar la velocidad de imagen a la que pueden ser representadas las partes en movimiento de una escena, mientras se mantiene una representación de resolución superior en partes de la escena más estáticas.

#### **3.4.16 Modo decodificación de segmento independiente**

Modo opcional que permite que una imagen se reconstruya sin dependencias de datos que cruzan segmentos de imagen de vídeo GOB o multi-GOB o límites de rebanadas (véase también el anexo R). Este modo proporciona solidez frente a errores evitando la propagación de datos erróneos a través de los límites de las zonas del segmento de imagen de vídeo.

### 3.4.17 Modo códigos de longitud variable (VLC) INTER alternativos

Se proporciona un modo opcional que mejora la eficacia de la codificación de imagen INTER cuando son evidentes cambios importantes en la imagen (véase también el anexo S). La mejora de la eficacia se consigue permitiendo que un código VLC concebido originalmente para imágenes INTRA se utilice también para algunos coeficientes de imagen INTER.

### 3.4.18 Modo cuantificación modificada

Se proporciona un modo opcional que mejora la capacidad de controlar la velocidad binaria de la codificación, reduce el error de cuantificación de la crominancia, amplía la gama de coeficientes DCT representables e impone ciertas restricciones a los valores de los coeficientes (véase también el anexo T). Este modo modifica la semántica del parámetro tamaño de paso de cuantificación diferencial del tren de bits ampliando la gama de cambios de tamaño de paso que se pueden especificar. Reduce además el tamaño del paso de cuantificación utilizado para datos de crominancia. La gama de niveles de coeficientes DCT se amplía para asegurar que cualquier valor de coeficiente posible se puede codificar dentro del grado de exactitud que permite el tamaño del paso. En este modo, se imponen también algunas restricciones a los coeficientes para aumentar la característica de detección de errores y reducir al mínimo la complejidad del decodificador.

## 3.5 Velocidad binaria

El reloj de transmisión se proporciona por medios externos. La velocidad binaria de vídeo puede ser variable. En esta Recomendación no se imponen restricciones a la velocidad binaria de vídeo; estas restricciones las determinará el terminal o la red.

## 3.6 Almacenamiento en tampón

El codificador controlará su tren binario de salida para cumplir los requisitos del decodificador ficticio de referencia definido en el anexo B. Los datos de vídeo se proporcionarán en cada ciclo de reloj válido. Esto se consigue mediante relleno de tipo de macrobloque y patrón de grupo codificado (MCBCP, *macroblock type and coded block pattern*) (véanse los cuadros 7 y 8) o, cuando se utiliza corrección de errores hacia adelante, también mediante tramas de relleno de corrección de errores hacia adelante (véase el anexo H).

El número de bits creado por la codificación de cualquier imagen no excederá de un valor máximo especificado por el parámetro BPPmaxKb que se mide en unidades de 1024 bits. El valor admisible mínimo del parámetro BPPmaxKb depende del tamaño de la imagen más grande que ha sido negociado para su utilización en el tren de bits (véase el cuadro 1). El tamaño de imagen se mide por las veces que su anchura está contenida en su altura para la componente de luminancia (Y), medida en píxels. Un codificador puede utilizar un valor mayor para BPPmaxKb que el especificado en el cuadro 1 a condición de que este valor mayor sea negociado primero por medios externos, por ejemplo, aplicando la Rec. UIT-T H.245.

Cuando se emplea el modo escalabilidad temporal, SNR y espacial (véase el anexo O), el número de bits enviados para cada imagen en cada capa de mejora no debería exceder del valor máximo especificado por el parámetro BPPmaxKb.



## Cuadro 1/H.263 – Parámetro BPPmaxKb mínimo para diferentes formatos de imagen fuente

Tamaño de imagen Y en píxeles (= anchura por altura tal y como aparece en el encabezamiento de imagen)	BPPmaxKb mínima
hasta 25 344 (o cuarto de CIF)	64
25 360 a 101 376 ( o CIF)	256
101 392 a 405 504 (ó 4 veces CIF)	512
405 520 o más	1024

### 3.7 Simetría de la transmisión

El códec se puede utilizar para la comunicación visual bidireccional o unidireccional.

### 3.8 Tratamiento de los errores

El tratamiento de los errores se suministra por medios externos (por ejemplo, Rec. UIT-T H.223). Si no se proporciona por medios externos (por ejemplo, la Rec. UIT-T H.221), se puede utilizar la alineación de trama y la corrección de errores opcional, como se describe en el anexo H.

El decodificador puede enviar una instrucción para codificar uno o más GOB de la imagen próxima en modo INTRA con los parámetros de codificación adecuados para evitar el desbordamiento (o rebanadas si se emplea el anexo K) de la memoria tampón. El decodificador también puede enviar una instrucción de que se transmita únicamente los encabezamientos GOB no vacíos si no se utiliza el modo estructura de rebanadas (véase el anexo K). Estas señales se transmiten por medios externos (por ejemplo, Rec. UIT-T H.245).

### 3.9 Funcionamiento multipunto

En el anexo C se describen las facilidades necesarias para el funcionamiento multipunto con conmutación.

## 4 Codificador de fuente

### 4.1 Formato de fuente

El codificador de fuente funciona con imágenes no entrelazadas cuyo formato de fuente que se define en términos de:

- 1) el formato de imagen, determinado por el número de píxeles por línea, el número de líneas por imagen, y la relación de aspecto de píxel, y
- 2) la temporización entre imágenes, determinada por la frecuencia de reloj de imagen (PCF, *picture clock frequency*). Por ejemplo, el formato intermedio común (CIF) tiene 352 píxeles por línea, 288 líneas, una relación de aspecto de píxel de 12:11, y una frecuencia de reloj de imagen de 30 000/1001 imágenes por segundo.

El codificador de fuente trabaja con imágenes no entrelazadas que se presentan con una frecuencia de reloj de imagen (PCF) de 30 000/1001 (aproximadamente 29,97) veces por segundo, denominada PCF CIF. Es posible también negociar la utilización de una PCF personalizada opcional por medios externos. Una PCF personalizada viene dada por  $1\,800\,000 / (\text{divisor de reloj} * \text{factor de conversión de reloj})$  donde el divisor de reloj puede tener valores de 1 a 127 y el factor de conversión de reloj puede ser 1000 o bien 1001. La tolerancia de la frecuencia de reloj de imagen es de  $\pm 50$  ppm.

Las imágenes se codifican como una componente de luminancia y dos componentes de diferencia de color ( $Y$ ,  $C_B$  y  $C_R$ ). Estas componentes y los códigos que representan sus valores muestreados son los que define la Rec. UIT-R BT.601-5.

- Negro = 16.
- Blanco = 235.
- Diferencia de color nula = 128.
- Diferencia de color máxima = 16 y 240.

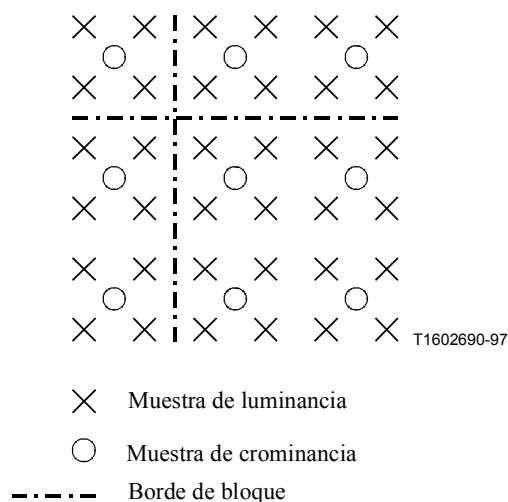
Estos valores son nominales y el algoritmo de codificación funciona con valores de entrada comprendidos entre 1 y 254.

Hay cinco formatos de imagen normalizados: sub-cuarto de CIF, cuarto de CIF, CIF, 4 veces CIF y 16 veces CIF. También es posible negociar un formato de imagen personalizado. Para todos estos formatos de imagen, la estructura de muestreo de luminancia es  $dx$  píxels por línea y  $dy$  líneas por imagen en una disposición ortogonal. El muestreo de cada una de las dos componentes de diferencia de color se efectúa a  $dx/2$  píxels por línea y  $dy/2$  líneas por imagen, ortogonal. Los valores de  $dx$ ,  $dy$ ,  $dx/2$  y  $dy/2$  se ofrecen en el cuadro 2 para cada uno de los formatos de imagen normalizados.

**Cuadro 2/H.263 – Número de píxels por línea y número de líneas de cada formato de imagen H.263 normalizado**

Formato de imagen	Número de píxels de luminancia (dx)	Número de líneas de luminancia (dy)	Número de píxels de crominancia (dx/2)	Número de líneas de crominancia (dy/2)
sub-cuarto de CIF	128	96	64	48
cuarto de CIF	176	144	88	72
CIF	352	288	176	144
4 veces CIF	704	576	352	288
16 veces CIF	1408	1152	704	576

Para todos los formatos de imagen, las muestras de diferencia de color se colocan de forma tal que sus límites de bloque coincidan con los límites de bloque de luminancia, como se muestra en la figura 2. La relación de aspecto de píxel es la misma para cada uno de los formatos de imagen normalizados, y es la misma que se define para cuarto de CIF y CIF en la Rec. UIT-T H.261:  $(288/3):(352/4)$ , que se simplifica a 12:11 en números relativamente primos. El área de la imagen cubierta por todos los formatos de imagen normalizados, salvo el sub-cuarto de CIF, tiene una relación de aspecto 4:3.



**Figura 2/H.263 – Posición de las muestras de luminancia y crominancia**

Los formatos de imagen personalizados pueden tener una relación de aspectos de píxel personalizada como se describe en el cuadro 3, si la utilización de la relación de aspecto de píxel personalizada se negocia primero por medios externos. Los formatos de imagen personalizados pueden tener cualquier número de líneas y cualquier número de píxels por línea, siempre que el número de líneas sea divisible por 4 y se encuentre en la gama [4, ..., 1152], y siempre que el número de píxels por línea sea también divisible por 4 y se encuentre en la gama [4, ..., 2048]. Para formatos de imagen cuya anchura o altura no sea divisible por 16, la imagen se decodifica como si su anchura o altura correspondieran al tamaño mayor siguiente divisible por 16, y a continuación se recorta la imagen a la derecha y en la parte inferior hasta una anchura y altura adecuadas a efectos de visualización solamente.

**Cuadro 3/H.263 – Relaciones de aspecto de píxel personalizadas**

Relación de aspecto de píxel	Anchura de píxel: altura de píxel
Cuadrada	1:1
CIF	12:11
Tipo 525 para imagen 4:3	10:11
CIF para imagen 16:9	16:11
Tipo 525 para imagen 16:9	40:33
Ampliada	m:n, m y n son números relativamente primos

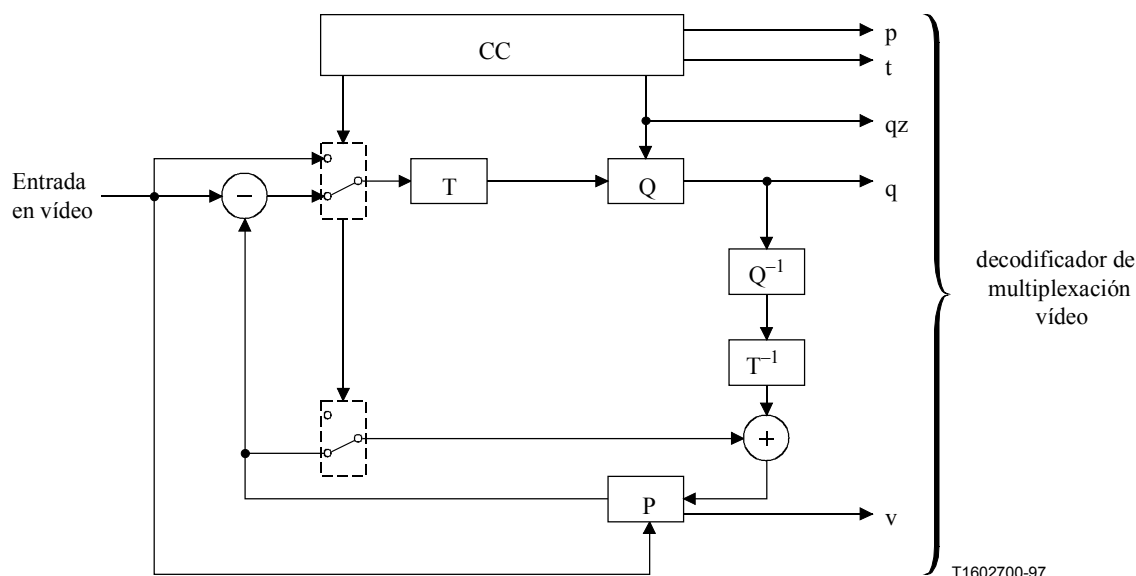
Todos los decodificadores y codificadores han de poder funcionar utilizando la frecuencia de reloj de imagen CIF. Algunos decodificadores y codificadores pueden admitir además frecuencias de reloj de imagen personalizadas. Todos los decodificadores deberán poder funcionar con el formato de imagen sub-cuarto de CIF. Asimismo, todos los decodificadores deben poder funcionar con el formato de imagen cuarto de CIF. Algunos decodificadores también pueden funcionar con CIF, 4 veces CIF o 16 veces CIF, o formatos de imagen personalizados. Los codificadores deben poder funcionar con uno de los formatos de imagen sub-cuarto de CIF y cuarto de CIF. Los codificadores determinarán cuál de esos dos formatos se utilizará, y no están obligados a funcionar con ambos. Algunos codificadores también pueden funcionar con CIF, 4 veces CIF, 16 veces CIF, o con formatos de imagen personalizados. Los medios externos, como la Rec. H.245, indican los formatos opcionales y las frecuencias de reloj de imagen que puede tratar el decodificador. Para un panorama completo de los posibles formatos de imagen y los algoritmos de codificación vídeo, se remite al lector a la descripción del terminal, por ejemplo, la Rec. UIT-T H.324.

NOTA – En el caso de CIF, el número de píxeles por línea es compatible para fines prácticos con el muestreo de las porciones activas de las señales de luminancia y de diferencia de color de fuentes de 525 ó 625 líneas a 6,75 y 3,375 MHz, respectivamente. Estas frecuencias tienen una relación simple con las de la Rec. UIT-R BT.601-5.

Se proporcionarán medios para restringir el periodo máximo de transmisión de imagen de los codificadores colocando un número mínimo de imágenes no transmitidas entre las imágenes transmitidas. La selección de esa cantidad mínima se hará por medios externos (por ejemplo, Rec. UIT-T H.245). En el cálculo de la cantidad mínima de imágenes no transmitidas en el modo trama PB, las imágenes P y B de una unidad de tramas PB se tratan como dos imágenes separadas.

## 4.2 Algoritmo de codificación de fuente de vídeo

El codificador de fuente se muestra de forma generalizada en la figura 3. Los principales elementos son la predicción, la transformación de bloques y la cuantificación.



- T Transformación
- Q Cuantificador
- P Memoria de imagen con retardo variable y compensación de movimiento
- CC Control de codificación
- p Bandera de INTRA/INTER
- t Bandera de transmitido o no transmitido
- qz Indicador de cuantificador
- q Índice de cuantificación para los coeficientes de la transformada
- v Vector de movimiento

**Figura 3/H.263 – Codificador de fuente**

### 4.2.1 Grupos de bloques, rebanadas, macrobloques y bloques

Cada imagen se divide en grupos de bloques (GOB) o bien en rebanadas.

Un grupo de bloques (GOB, *group of blocks*) comprende hasta  $k * 16$  líneas, donde  $k$  depende del número de líneas en el formato de imagen y de si se utiliza el modo actualización de resolución reducida opcional (véase el anexo Q). Las dependencias se muestran en el cuadro 4. Si el número de líneas es menor o igual que 400 y no se utiliza el modo actualización de resolución reducida opcional, el valor de  $k$  es 1. Si el número de líneas es menor o igual que 800 y se utiliza el modo autorización de resolución reducida opcional o el número de líneas es mayor que 400, el valor de  $k$  es 2. Si el número de líneas es mayor que 800, el valor de  $k$  es 4. Cuando se utilizan tamaños de imagen personalizados, el número de líneas en el último GOB (el situado más abajo) puede ser

menor que  $k * 16$  si el número de líneas en la imagen no es divisible por  $k * 16$ . Sin embargo, cada GOB en cada formato de imagen normalizado tiene  $k * 16$  líneas, ya que el número de líneas en cada formato de imagen normalizado es un múltiplo entero de  $k * 16$ . Así, por ejemplo, si no se utiliza el modo opcional resolución reducida el número de GOB por imagen es 6 para sub-cuarto de CIF, 9 para cuarto de CIF, y 18 para CIF, 4 veces CIF y 16 veces CIF. La numeración de GOB se efectúa mediante la exploración vertical de los GOB, comenzando por el GOB superior (número 0) y terminando con el GOB situado más abajo. En la figura 4 se da un ejemplo de la disposición de los GOB en una imagen de formato CIF. Los datos de cada GOB consisten en un encabezamiento GOB (que puede estar vacío), seguido de datos de macrobloques. Los datos de GOB se transmiten por GOB en orden de GOB creciente.

**Cuadro 4/H.263 – Parámetro k para definición del tamaño de los GOB**

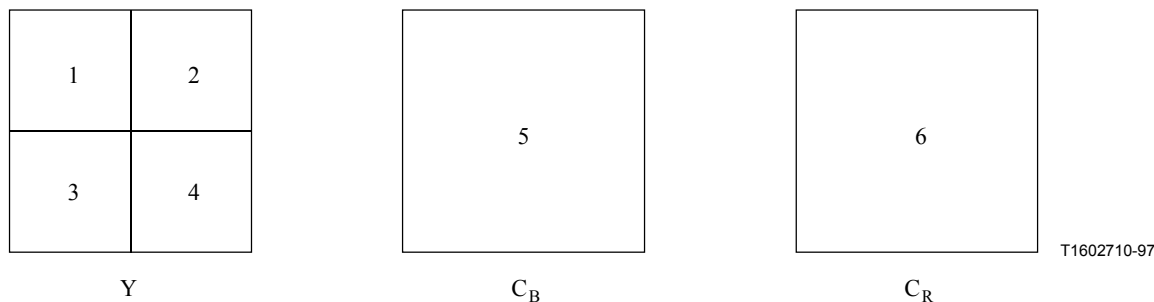
Número de líneas dy	Valor de k cuando no se utiliza el modo RRU	Valor de k cuando se utiliza el modo RRU
4,...,400	1	2
404,...,800	2	2
804,...,1152	4	4

En el anexo K se describe el modo estructura en rebanada. Las rebanadas son similares a los GOB en el sentido de que constituyen una capa de múltiples macrobloques de la sintaxis, pero las rebanadas tienen una forma y un uso más flexibles que los GOB, y, en determinadas condiciones, pueden aparecer en cualquier orden en el tren de datos.

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17

**Figura 4/H.263 – Disposición de grupos de bloques en una imagen CIF**

Cada GOB se divide en macrobloques. La estructura de un macrobloque depende de si se utiliza el modo opcional actualización de resolución reducida (RRU, *reduced-resolution update*) (véase el anexo Q). A menos que se utilice el modo RRU, cada macrobloque se relaciona con 16 píxels por 16 líneas de Y, y con 8 píxels por 8 líneas correspondientes espacialmente de  $C_B$  y  $C_R$ . Además, un macrobloque consiste en 4 bloques de luminancia y los dos bloques de diferencia de color correspondientes espacialmente, como se muestra en la figura 5. Cada bloque de luminancia o de crominancia se relaciona así con 8 píxels por 8 líneas de Y,  $C_B$  o  $C_R$ . Si no se utiliza el modo RRU, el GOB comprende una fila de macrobloques para sub-cuarto de CIF, cuarto de CIF y CIF, dos filas de macrobloques para 4 veces CIF y 4 filas de macrobloques para 16 veces CIF.



**Figura 5/H.263 – Disposición de los bloques en un macrobloque**

Cuando se utiliza el modo RRU, un macrobloque se relaciona con 32 píxels por 32 líneas de Y, y con 16 píxels por 16 líneas de  $C_B$  y  $C_R$  correspondientes espacialmente, y cada bloque de luminancia o crominancia se relaciona con 16 píxels por 16 líneas de Y,  $C_B$  o  $C_R$ . Además, un GOB comprende una fila de macrobloques para CIF y 4 veces CIF, y dos filas de macrobloques para 16 veces CIF.

La numeración de macrobloque se efectúa mediante un barrido horizontal de las filas de macrobloque de izquierda a derecha, comenzando por la línea de macrobloque superior y terminando por la fila de macrobloque más baja. Los datos por los macrobloques se transmiten por macrobloque aumentando el número de macrobloque. Los datos para los bloques se transmiten por bloque al aumentar el número de bloque (véase la figura 5).

Los criterios de elección del modo y la transmisión de un bloque no son objeto de recomendación y pueden variar dinámicamente como parte de la estrategia de control de la codificación. Los bloques transmitidos se transforman y los coeficientes resultantes se cuantifican y se codifican en entropía.

#### 4.2.2 Predicción

La forma primaria de predicción se efectúa entre imágenes y se puede aumentar por compensación de movimiento (véase 4.2.3). El modo de codificación en el que se aplica la predicción temporal se denomina INTER; el modo de codificación se denomina INTRA cuando no se aplica la predicción temporal. El modo de codificación INTRA se puede señalar en el nivel de la imagen (INTRA para imágenes I o INTER para imágenes P) o en el nivel de macrobloque en las imágenes P. En el modo opcional tramas PB, las imágenes B están siempre codificadas en modo INTER. Las imágenes B se predicen parcialmente en forma bidireccional (véase el anexo G).

En total, la Rec. UIT-T H.263 presenta 7 tipos de imágenes básicas (de las cuales sólo las dos primeras son obligatorias) que se definen esencialmente en términos de su estructura de predicción:

- 1) INTRA: Imagen que no tiene imagen(es) de referencia para predicción (también denominada imagen I);
- 2) INTER: Imagen que utiliza una imagen de referencia temporalmente previa (también denominada imagen P);

- 3) PB: Trama que representa dos imágenes y que tiene una imagen de referencia temporalmente previa (véase el anexo G);
- 4) PB mejorada: Trama funcionalmente similar pero normalmente mejor que una trama PB (véase el anexo M);
- 5) B: Imagen que tiene dos imágenes de referencia, una de las cuales precede temporalmente a la imagen B y la otra sucede temporalmente a la imagen B y tiene el mismo tamaño de imagen (véase el anexo O);
- 6) EI: Imagen que tiene una imagen de referencia temporalmente simultánea cuyo tamaño de imagen es el mismo o menor (véase el anexo O); y
- 7) EP: Imagen que tiene dos imágenes de referencia, una que precede temporalmente a la imagen EP y otra que es temporalmente simultánea y tiene el mismo tamaño de imagen o un tamaño menor (véase el anexo O).

Según se utiliza en la presente Recomendación, una imagen de "referencia" o de "anclaje" es una imagen que contiene datos que se pueden utilizar por referencia como base para la decodificación de otra imagen. Este uso por referencia se conoce también como "predicción", aunque a veces puede indicar en realidad la utilización en un sentido de inversión temporal.

### 4.2.3 Compensación de movimiento

El decodificador aceptará un vector por macrobloque o si se utiliza el modo predicción avanzada o el modo filtro de desbloqueo, aceptará uno o cuatro vectores por macrobloque (véanse los anexos F y J). Cuando se utiliza el modo tramas PB, se puede transmitir un vector delta adicional por macrobloque para la adaptación de vectores de movimiento destinados a la predicción de macrobloque B. De manera similar, un macrobloque de tramas PB mejoradas (véase el anexo M) puede incluir un vector de movimiento hacia adelante adicional. Los macrobloques de imagen B (véase el anexo O) se pueden transmitir junto con un vector de movimiento hacia adelante y un vector de movimiento hacia atrás, y las imágenes EP se pueden transmitir con un vector de movimiento hacia adelante.

Los componentes horizontal y vertical de los vectores de movimiento tienen valores de un entero o de mitad de entero. En el modo de predicción por defecto, estos valores están restringidos a la gama  $[-16, 15,5]$  que también es válida para los componentes de vectores de movimiento hacia adelante y hacia atrás de las imágenes B.

Sin embargo, en el modo de vector de movimiento sin restricción, se aumenta la gama máxima para componentes de vector. Cuando PLUSPTYPE está ausente, la gama es  $[-31,5, 31,5]$ , con la restricción de que sólo se pueden alcanzar los valores que están dentro de una gama  $[-16, 15,5]$  alrededor del predictor para cada componente de vector de movimiento si el predictor está en la gama  $[-15,5, 16]$ . Cuando PLUSPTYPE está ausente y el predictor está fuera de la gama  $[-15,5, 16]$ , se pueden alcanzar todos los valores dentro de la gama  $[-31,5, 31,5]$  con el mismo signo que el predictor más el valor cero. Si PLUSPTYPE está presente, los valores del vector de movimiento están menos restringidos (véase también el anexo D).

En el modo actualización de resolución reducida, la gama del vector de movimiento se amplía a un tamaño de aproximadamente el doble, y cada componente de vector está limitado a tener solamente un valor de medio entero o cero. Por tanto, la gama de cada componente de vector de movimiento es  $[-31,5, 30,5]$  en el modo actualización de resolución reducida por defecto (véase el anexo Q) y es mayor si también se utiliza el vector de movimiento sin restricción (véase también el anexo D).

Un valor positivo del componente horizontal o vertical del vector significa que la predicción se forma a partir de los píxeles de la imagen referenciada que están espacialmente a la derecha o por debajo de los píxeles que se predicen.

Los vectores de movimiento están restringidos de modo que todos los píxels referenciados por ellos se encuentren dentro de la zona de imagen codificada, salvo cuando se utiliza el modo vector de movimiento sin restricción, el modo predicción avanzada o el modo filtro de desbloqueo (véanse los anexos D, F y J), o en las imágenes B y EP del modo escalabilidad temporal, SNR y espacial (véase el anexo O).

#### **4.2.4 Cuantificación**

Salvo que se utilicen los modos opcionales codificación INTRA avanzada o cuantificación modificada, el número de cuantificadores es 1 para el primer coeficiente de bloque INTRA y 31 para todos los demás coeficientes. Dentro de un macrobloque, se utiliza el mismo cuantificador para todos los coeficientes, excepto el primero de los bloques INTRA. No se definen los niveles de decisión. El primer coeficiente de los bloques INTRA es, nominalmente, el valor de la transformada en continua (cc) cuantificado uniformemente con un paso de valor 8. Cada uno de los restantes 31 cuantificadores utilizan niveles de reconstrucción de espaciado igual con una zona muerta central alrededor de cero y un paso de valor par comprendido entre 2 y 62. Para las fórmulas exactas, véase 6.2. Para la cuantificación que utiliza el modo codificación INTRA avanzado, véase el anexo I. Para la cuantificación que utiliza el modo de cuantificación modificado, véase el anexo T.

NOTA – En el caso de tamaños más pequeños de pasos de cuantificación, no se puede representar la gama dinámica completa de los coeficientes de la transformada, a menos que se utilice el modo opcional cuantificación modificada.

### **4.3 Control de la codificación**

Se pueden cambiar varios parámetros para controlar la velocidad de generación de datos de vídeo codificados. Se trata del proceso efectuado antes del codificador de fuente, el cuantificador, el criterio de significación de bloque y el submuestreo temporal. Las proporciones de estas medidas en la estrategia de control global no son objeto de recomendación.

Al ser invocado, el submuestreo temporal se ejecuta mediante el descarte de imágenes completas.

Un decodificador puede indicar su preferencia por un compromiso específico entre la resolución espacial y la temporal de la señal vídeo. El codificador señalará su compromiso por defecto al comienzo de la llamada e indicará si es capaz de responder a las solicitudes del decodificador de cambiar ese compromiso. Estas señales se transmiten por medios externos (por ejemplo, Rec. UIT-T H.245).

### **4.4 Actualización forzada**

Esta función se realiza forzando la utilización del modo INTRA del algoritmo de codificación. El esquema de actualización no se define. Para controlar la acumulación de errores por desajuste de la transformada inversa, cada macrobloque se codificará en modo INTRA al menos una vez cada 132 veces cuando se transmiten los coeficientes de este macrobloque en imágenes P. Un requisito similar es aplicable cuando se utilizan imágenes EP opcionales (véase el anexo O), de las que cada macrobloque se codificará en un modo INTRA o superior al menos una vez de cada 132 veces cuando se transmiten los coeficientes al macrobloque.

### **4.5 Alineación en byte de los códigos de comienzo**

La alineación en byte de los códigos de comienzo se consigue insertando una palabra de código de menos de 8 bits cero antes del código del comienzo, de manera que el primer bit del código de comienzo es el primer bit (más significativo) de un byte. Por consiguiente, un código de comienzo está alineado en byte si la posición de su bit más significativo es un múltiplo de 8 bits del primer bit del tren binario H.263. Es obligatorio que los códigos de comienzo de todos los tipos de imagen,



rebanada y EOSBS estén alineados en byte; los códigos de comienzo de GOB y EOS no están sujetos a esa obligación.

NOTA 1 – El número de bits utilizado para una imagen determinada es variable pero siempre un múltiplo de 8 bits.

NOTA 2 – El procedimiento H.324 requiere codificadores de la Rec. UIT-T H.263 para alinear los códigos de comienzo de imagen con el comienzo de las unidades de información lógicas pasadas a la capa de adaptación (las SDU de AL).

## 5 Sintaxis y semántica

La sintaxis de vídeo tiene una estructura jerárquica de cuatro capas primarias. De arriba hacia abajo, esas capas son:

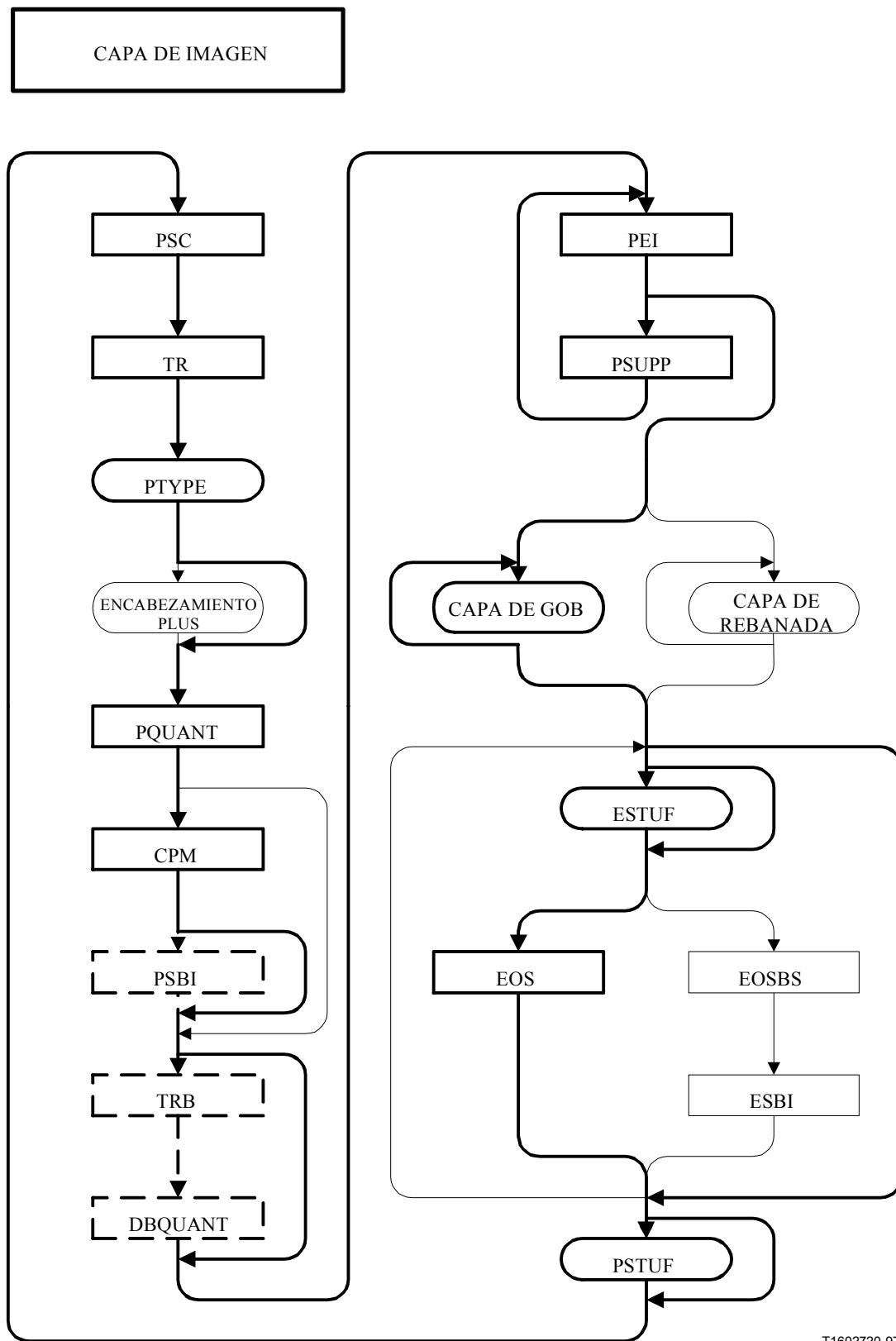
- imagen;
- grupo de bloques, o rebanada, o segmento de imagen de vídeo;
- macrobloque;
- bloque.

El diagrama de sintaxis aparece en la figura 6. La guía para la interpretación del diagrama es la siguiente:

- 1) Los caminos de flechas muestran los flujos de elementos de sintaxis posibles. Cualquier elemento de sintaxis de longitud cero se considera ausente para fines de diagramación del camino de flechas (así, por ejemplo, hay un camino de flechas que elude PSTUF a pesar del carácter obligatorio del campo PSTUF ya que la longitud del campo PSTUF puede ser cero).
- 2) Las abreviaturas y la semántica para cada elemento de sintaxis son tal como se definen en las cláusulas siguientes.
- 3) El conjunto de elementos de sintaxis y caminos de flechas mostrados utilizando líneas continuas gruesas representa el flujo de sintaxis del modo de operación "línea básica" sin la utilización de ninguna mejora opcional. (Esta sintaxis también estaba presente en la versión 1 de esta Recomendación, y permanece sin cambios de cualquier clase.)
- 4) El conjunto de elementos de sintaxis y caminos de flechas mostrados utilizando líneas de puntos gruesas representa los elementos adicionales en el flujo de sintaxis de las mejoras opcionales presentes en las versiones 1 y 2 de esta Recomendación. (Esta sintaxis permanece sin cambios de cualquier clase.)
- 5) El conjunto de elementos de sintaxis y caminos de flechas mostrados utilizando líneas continuas delgadas representa los nuevos elementos adicionales en el flujo de sintaxis de las mejoras opcionales específicas de las características opcionales adicionales incorporadas en la versión 2. (Esta sintaxis no está presente en la versión 1.)
- 6) Los campos de elementos de sintaxis mostrados con límites de bordes cuadrados indican campos de longitud fija, y los de límites con bordes redondeados indican campos de longitud variable. Se muestra un elemento de sintaxis (DQUANT) con ambos tipos de límites, porque puede tener longitud variable o fija.
- 7) Un campo de longitud fija se define como aquel cuya longitud no depende de los datos del contenido del propio campo. La longitud de este campo es siempre la misma, o bien se determina por los datos previos en el flujo de sintaxis.
- 8) El término "capa" se utiliza para referirse a cualquier capa de la sintaxis que pueda ser comprendida y representada en forma de diagrama como una entidad distinta.

A menos que se especifique otra cosa, el bit más significativo se transmite primero. Es el bit 1, situado en el extremo izquierdo de las tablas de bits de esta Recomendación. A menos que se

especifique otra cosa, todos los bits no utilizados o de reserva se ponen a "1". Los bits de reserva no se utilizarán hasta que el UIT-T especifique sus funciones.

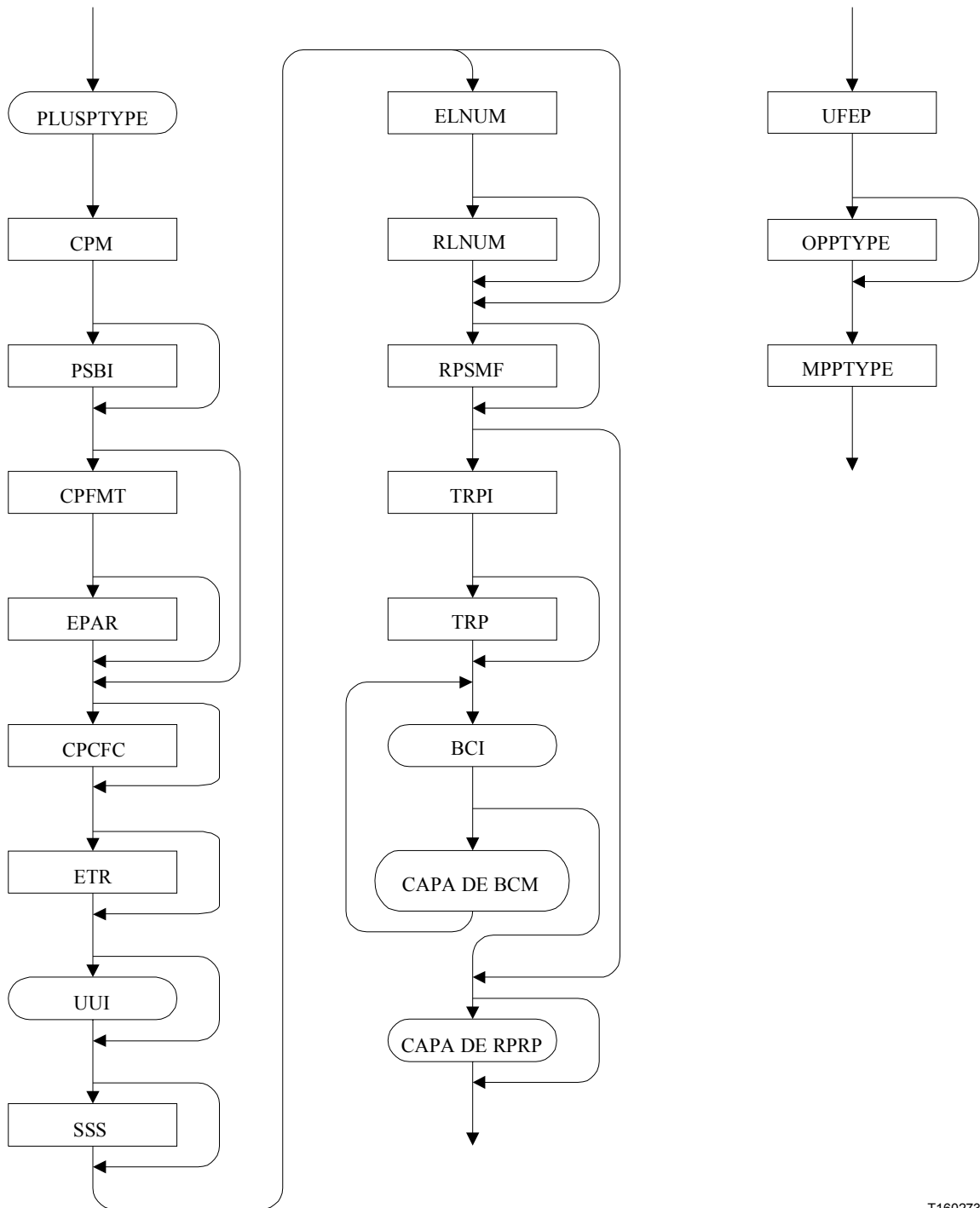


T1602720-97

Figura 6/H.263 – Diagrama de sintaxis para el tren de bits de vídeo (parte 1 de 7)

ENCABEZAMIENTO PLUS

PLUSPTYPE



T1602730-97

Figura 6/H.263 – Diagrama de sintaxis para el tren de bits de vídeo (parte 2 de 7)

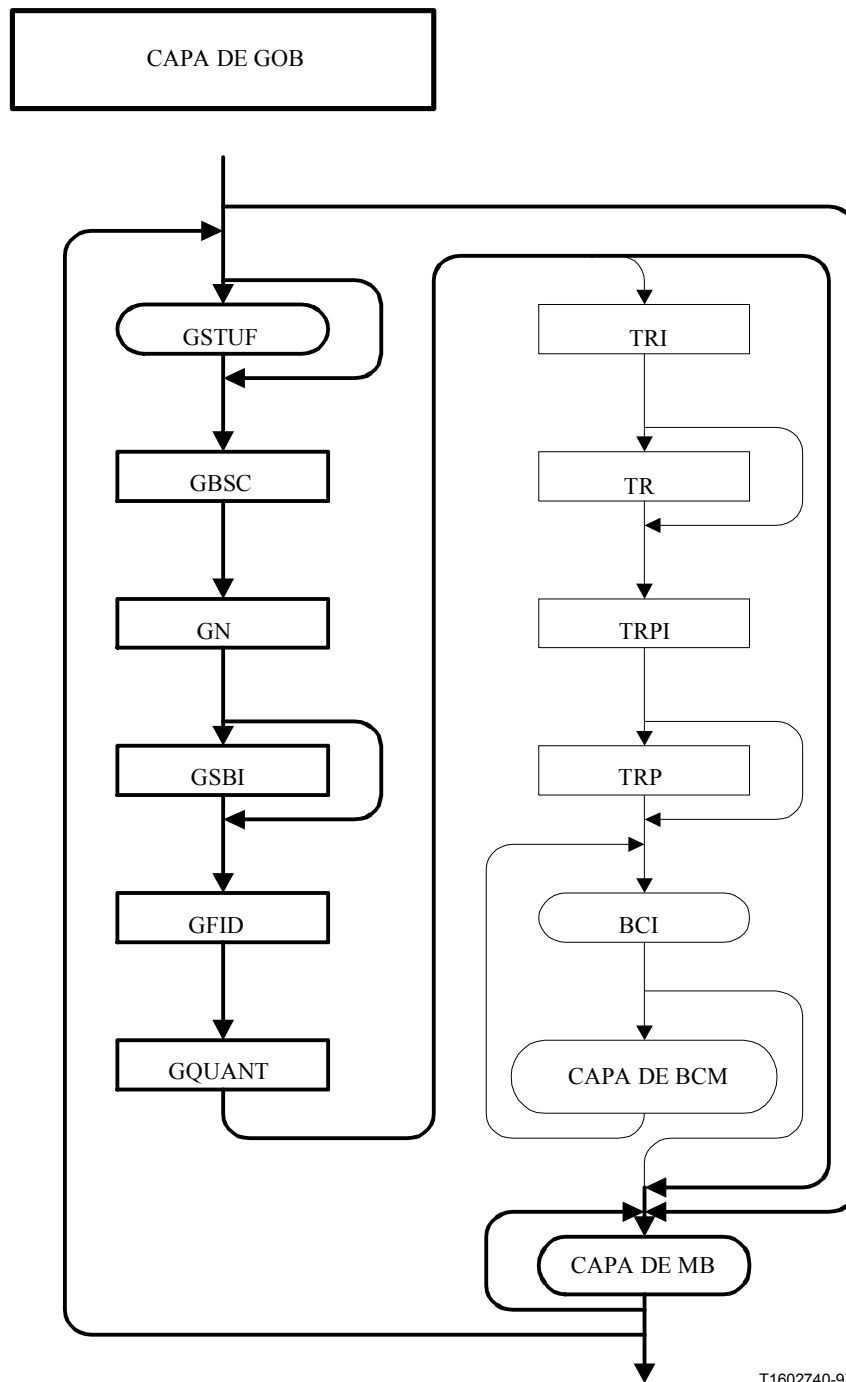


Figura 6/H.263 – Diagrama de sintaxis para el tren de bits de vídeo (parte 3 de 7)

CAPA DE REBANADA

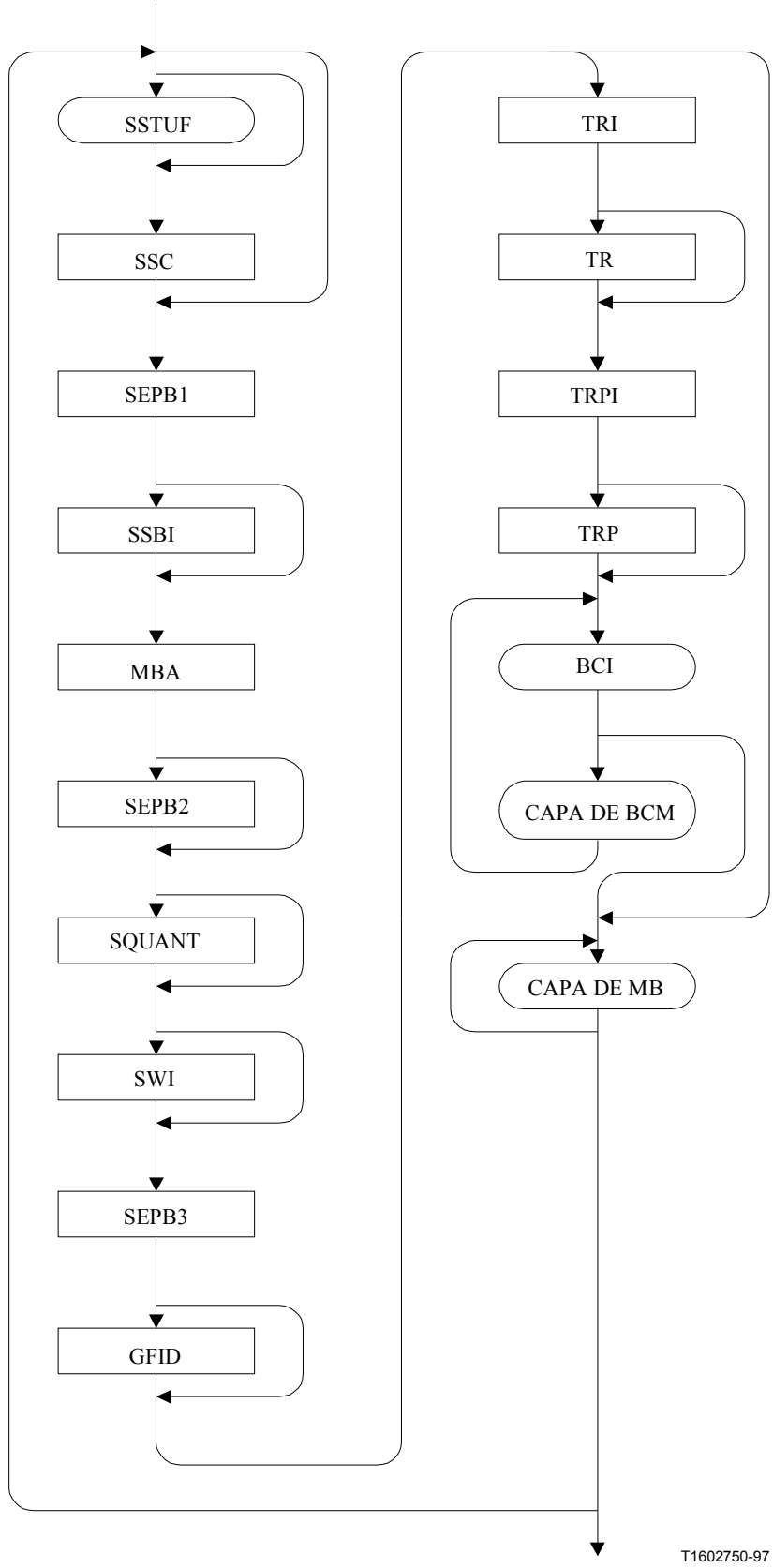


Figura 6/H.263 – Diagrama de sintaxis para el tren de bits de vídeo (parte 4 de 7)

CAPA DE CANAL LÓGICO  
SEPARADO BCM

CAPA DE BCM

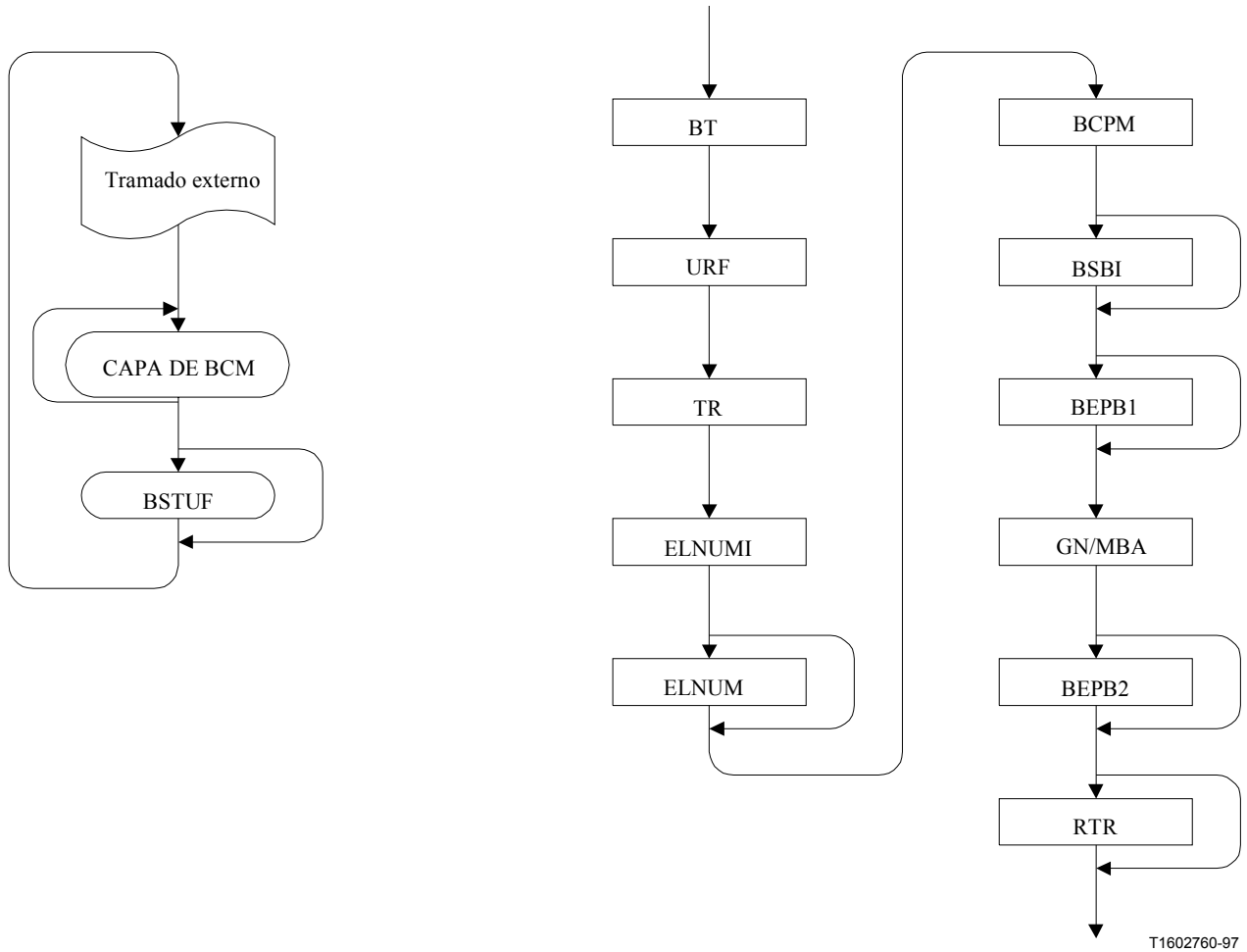


Figura 6/H.263 – Diagrama de sintaxis para el tren de bits de vídeo (parte 5 de 7)

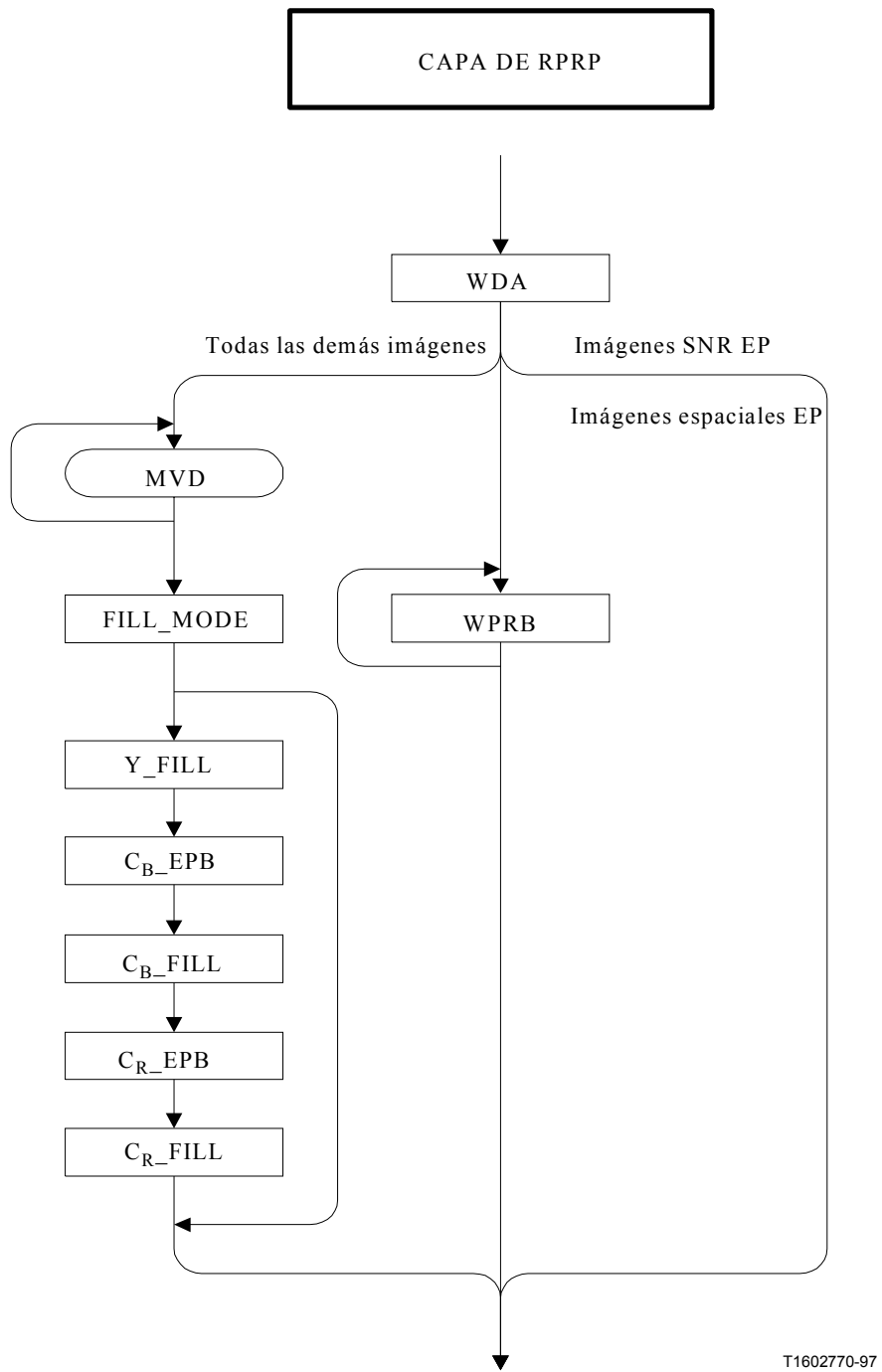
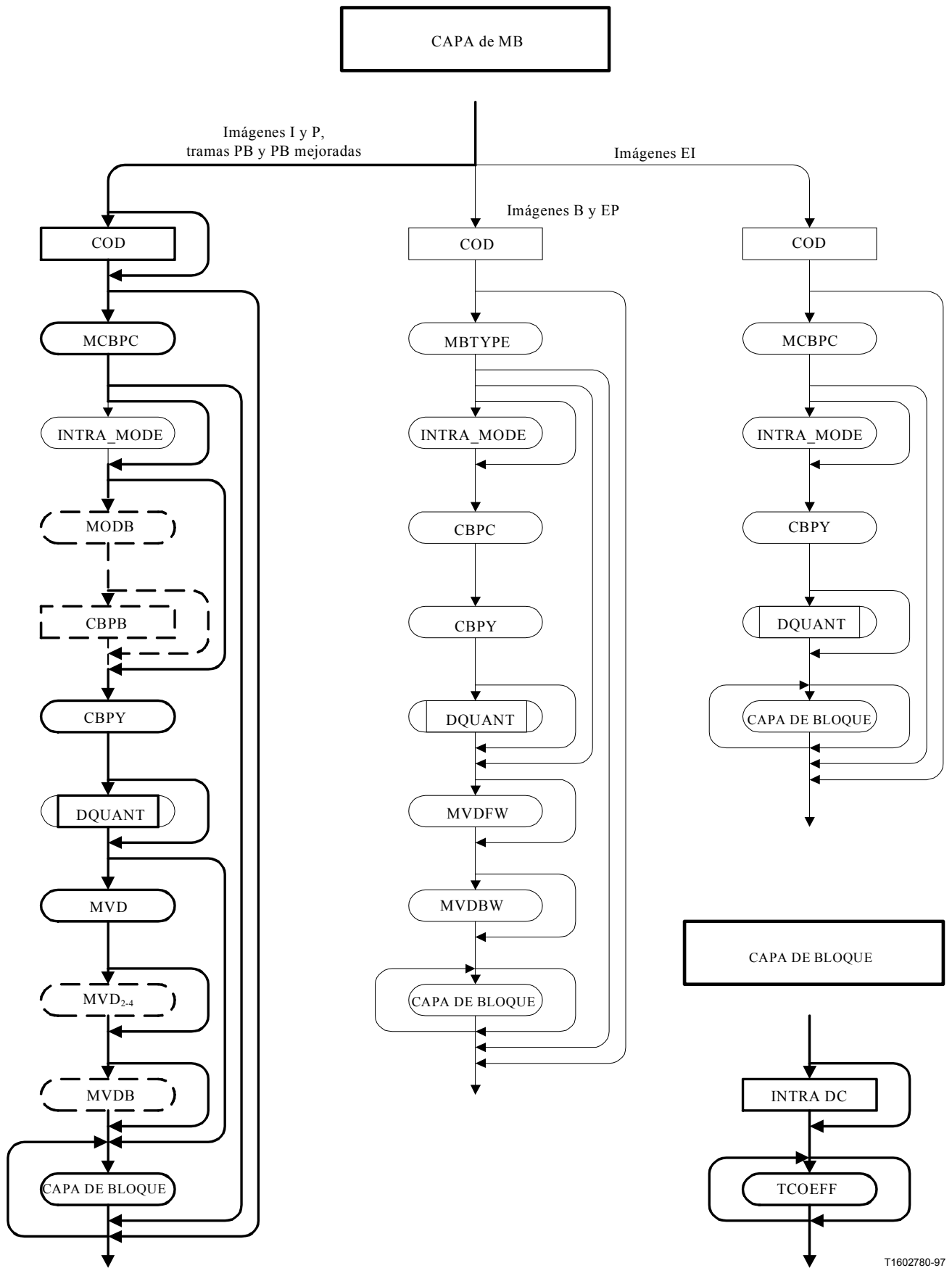


Figura 6/H.263 – Diagrama de sintaxis para el tren de bits de vídeo (parte 6 de 7)



T1602780-97

Figura 6/H.263 – Diagrama de sintaxis para el tren de bits de vídeo (parte 7 de 7)

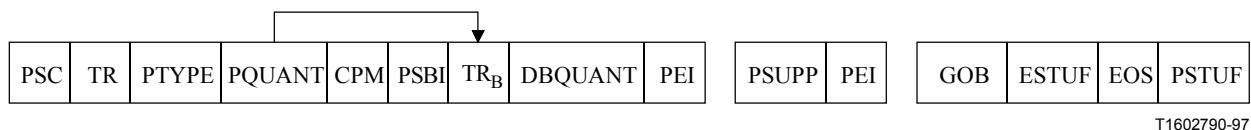


## 5.1 Capa de imagen

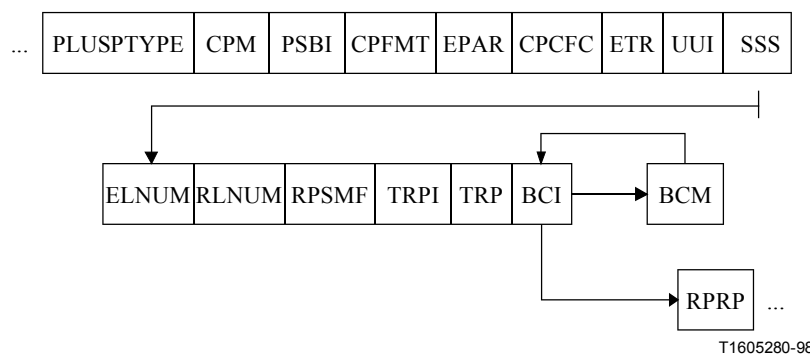
Los datos de cada imagen consisten en un encabezamiento de imagen, seguido de datos para grupo de bloques o rebanadas, seguidos en algunos casos de un código de fin de secuencia opcional y bits de relleno. La estructura aparece en la figura 7 para imágenes que no incluyen el campo de datos PLUSPTYPE opcional. PSBI sólo está presente si así lo indica CPM. TR<sub>B</sub> y DBQUANT sólo están presentes si PTYPE indica que se utiliza el modo "tramas PB" (salvo que esté presente el campo PLUSPTYPE y que en el mismo se indique la utilización de DBQUANT).

El campo de datos PLUSPTYPE opcional está presente cuando así lo indican los bits 6-8 de PTYPE. Si está presente, se incluye un conjunto adicional de datos en el tren de bits que sigue inmediatamente al PTYPE y precede al PQUANT. Además, en caso de presencia de PLUSPTYPE, los campos CPM y PSBI se desplazan hacia adelante en el encabezamiento de imagen, de modo que aparecen inmediatamente después de PLUSPTYPE en vez de estar situados después de PQUANT. El formato de los datos adicionales que siguen al campo PLUSPTYPE se muestra en la figura 8. Todos los campos de estos datos de encabezamiento de imagen adicionales después de PLUSPTYPE son opcionales, y dependen de que su presencia se indique en el campo PLUSPTYPE. Cuando se utiliza el modo estructura en rebanada (véase el anexo K), las rebanadas se sustituyen por grupos de bloques en el lugar indicado en la figura 7.

Las combinaciones de PSUPP y PEI pueden no estar presentes, y pueden estar repetidas cuando estén presentes. EOS y EOSBS+ESBI pueden no estar presentes, mientras que ESTUF puede estar presente sólo si EOS o EOSBS está presente. EOS no estará repetido salvo que aparezca como mínimo un código de comienzo de imagen entre cada par de códigos EOS. Los encabezamientos de imagen de las imágenes descartadas no se transmiten.



**Figura 7/H.263 – Estructura de la capa de imagen (sin campos opcionales relacionados con PLUSPTYPE)**



**Figura 8/H.263 – Estructura de campos opcionales relacionados con PLUSPTYPE (situados inmediatamente después de PTYPE cuando están presentes)**

### 5.1.1 Código de comienzo de imagen (PSC, *picture start code*) (22 bits)

El código de comienzo de imagen (PSC) es una palabra de 22 bits. Su valor es 0000 0000 0000 0000 1 00000. Todos los códigos de comienzo de imagen estarán alineados en bytes. Esto se consigue insertando los bits PSTUF que sean necesarios antes del código de comienzo, de manera que el primer bit del código de comienzo es el primer bit (más significativo) de un byte.

### 5.1.2 Referencia temporal (TR, *temporal reference*) (8 bits)

El valor de referencia temporal (TR) se forma incrementando en 1 el valor que tenía en el encabezamiento de imagen de referencia temporalmente previo más el número de imágenes saltadas o no referenciadas en la frecuencia de reloj de imagen desde la transmitida anteriormente. La interpretación de la TR depende de la frecuencia de reloj de imagen activa. En la frecuencia de reloj de imagen de formato CIF normalizado, TR es un número de 8 bits que puede tener 256 valores posibles. La aritmética se ejecuta con sólo los 8 bits menos significativos (LBS, *least significant bit*). Si se señala la utilización de una frecuencia de reloj de imagen personalizada, la referencia temporal ampliada (ETR, *extended temporal reference*) conforme a 5.1.8 y la referencia temporal (TR) forman un número de 10 bits de los que TR almacena los 8 bits menos significativos (LBS, *least significant bit*) y ETR almacena los 2 bits más significativos (MSB, *most significant bit*). La aritmética en este caso se ejecuta con los 10 bits menos significativos. En el modo tramas PB opcional o en el modo tramas PB mejoradas, TR se refiere únicamente a las imágenes P; para la referencia temporal de la parte imagen B de las tramas PB mejoradas véase 5.1.22.

### 5.1.3 Tipo de información (PTYPE, *type information*) (Longitud variable)

Información sobre la imagen completa:

- Bit 1: Siempre "1", para evitar emulación del código de comienzo.
- Bit 2: Siempre "0" para distinción con la Rec. UIT-T H.261.
- Bit 3: Indicador de pantalla dividida, "0" desactivado, "1" activado.
- Bit 4: Indicador de cámara de documento, "0" desactivado, "1" activado.
- Bit 5: Liberación de congelación de toda la imagen, "0" desactivado, "1" activado.
- Bits 6-8: Formato de fuente "000", prohibido, "001" sub-cuarto de CIF, "010" cuarto de CIF, "011" CIF, "100" 4 veces CIF, "101" 16 veces CIF, "110" reservado, "111" PTYPE extendido.

Si los bits 6-8 no son iguales a "111", que indica un PTYPE (PLUSPTYPE) extendido, los siguientes cinco bits también están presentes en PTYPE.

- Bit 9: Tipo de codificación de imagen, "0" INTRA, (imagen I), "1" INTER, (imagen P).
- Bit 10: Modo vector de movimiento sin restricción, opcional (véase el anexo D), "0" desactivado, "1" activado.
- Bit 11: Modo codificación aritmética basada en sintaxis, opcional (véase el anexo E), "0" desactivado, "1" activado.
- Bit 12: Modo predicción avanzada, opcional (véase el anexo F), "0" desactivado, "1" activado.
- Bit 13: Modo trama PB, opcional (véase el anexo G), "0" imagen I o P normal, "1" trama PB.

El indicador de pantalla dividida es una señal que indica que las mitades superior e inferior de la imagen decodificada se pueden visualizar una al lado de la otra. Este bit no tiene efectos directos sobre la codificación o decodificación de la imagen.

La señal liberación de congelación de toda la imagen es una señal procedente de un codificador que responde a una solicitud de retransmisión de paquetes (cuando no hay acuse de recibo) o a una petición de actualización rápida (véase también el anexo C) o petición de congelación de imagen (véase también el anexo L), y permite a un decodificador salir del modo imagen congelada y presentar la imagen decodificada de manera normal.

Si los bits 6 a 8 indican un formato fuente diferente del encabezamiento de imagen anterior, la imagen vigente será una imagen I, salvo que en los bits 6 a 8 se indique PTYPE extendido y la

capacidad de utilizar el modo opcional repetición de muestreo de imagen de referencia (véase el anexo P) haya sido negociada externamente (por ejemplo, Rec. UIT-T H.245).

Los bits 10 a 13 se refieren a modos opcionales que se utilizan únicamente después de una negociación entre el codificador y el decodificador (véanse también los anexos D, E, F y G respectivamente). Cuando el bit 9 se fija a "0", el bit 13 también se fija en "0".

Los bits 6 a 8 no tendrán el valor de "111" que indica la presencia de un PTYPE (PLUSPTYPE) ampliado salvo si la capacidad se ha negociado por medios externos (por ejemplo, aplicando la Rec. UIT-T H.245) para permitir la utilización de un formato fuente personalizado o uno o más de los otros modos opcionales disponibles solamente por el uso de un PTYPE ampliado (véanse los anexos I a K y M a T). Cuando los bits 6 a 8 no tengan el valor de "111", se considerará que todos los modos adicionales disponibles sólo por el uso de un PTYPE ampliado han sido fijados en el estado "desactivado" y se deducirá que permanecen en ese estado salvo que se activen posteriormente en el tren de bits.

#### **5.1.4 PTYPE plus (PLUSPTYPE) (Longitud variable)**

Palabra de código de 12 ó 30 bits que sólo está presente si en los bits 6-8 de PTYPE se indica la presencia de PTYPE ampliado.

PLUSPTYPE se compone de hasta tres subcampos, UFEP, OPPTYPE y MPPTYPE. OPPTYPE sólo está presente si UFEP tiene un valor particular.

##### **5.1.4.1 PTYPE extendido total de actualización (UFEP, *update full extended PTYPE*) (3 bits)**

Palabra de código de longitud fija de 3 bits que sólo está presente si en los bits 6-8 de PTYPE se indica "PTYPE ampliado". Cuando se fija a "000", indica que en el encabezamiento de imagen vigente sólo se incluyen los campos de PTYPE extendido que han de ser señalizados en cada encabezamiento de imagen (MPPTYPE). Cuando se fija a "001", indica que en el encabezamiento de imagen vigente se incluyen todos los campos PTYPE ampliados. Si el tipo de imagen es INTRA o EI, este campo se fijará a "001".

Además, si PLUSPTYPE está presente en cada secuencia continua de imágenes, este campo se fijará a "001" al menos tantas veces como esté especificado por un periodo de temporización de cinco segundos o de cinco imágenes, el que permita un intervalo de tiempo mayor. De manera más específica, el periodo de temporización requiere que UFEP = "001" aparezca en el campo PLUSPTYPE (si PLUSPTYPE está presente en cada imagen intermedia) del encabezamiento de la primera imagen con referencia temporal que indique un intervalo de tiempo superior o igual a cinco segundos desde la última ocurrencia de UFEP = "001", o de la quinta imagen después de la última ocurrencia de UFEP = "001" (de los dos requisitos, el que permita un periodo de tiempo superior al medido por referencia temporal).

Los codificadores deben fijar UFEP a "001" más frecuentemente en entornos con tendencia a errores. Los valores de UFEP distintos de "000" y "001" quedan en reserva.

##### **5.1.4.2 Parte opcional de PLUSPTYPE (OPPTYPE, *optional part of PLUSPTYPE*) (18 bits)**

Si UFEP es "001", en PLUSPTYPE están presentes los siguientes bits:

- Bits 1-3 Formato fuente, "000" reservado, "001" sub-cuarto de CIF, "010" cuarto de CIF, "011" CIF, "100" 4 veces CIF, "101" 16 veces CIF, "110" formato fuente personalizado, "111" reservado.
- Bit 4 PCF personalizado opcional, "0" PCF CIF, "1" PCF personalizado.
- Bit 5 Modo vector de movimiento sin restricción (UMV, *unrestricted motion vector*), opcional, (véase el anexo D), "0" desactivado, "1" activado.

- Bit 6 Modo codificación aritmética basada en la sintaxis (SAC, *syntax-based arithmetic coding*), opcional, (véase el anexo E), "0" desactivado, "1" activado.
- Bit 7 Modo predicción avanzada (AP, *advanced prediction*), opcional, (véase el anexo F), "0" desactivado, "1" activado.
- Bit 8 Modo codificación INTRA avanzada (AIC, *advanced INTRA coding*), opcional, (véase el anexo I), "0" desactivado, "1" activado.
- Bit 9 Modo filtro de desbloqueo (DF, *deblocking filter*), opcional, (véase el anexo J), "0" desactivado, "1" activado.
- Bit 10 Modo estructura en rebanada (SS, *slice structured*), opcional, (véase el anexo K), "0" desactivado, "1" activado.
- Bit 11 Modo selección de imagen de referencia (RPS, *reference picture selection*), opcional, (véase el anexo N), "0" desactivado, "1" activado.
- Bit 12 Modo codificación de segmento independiente (ISD, *independent segment decoding*), opcional, (véase el anexo R), "0" desactivado, "1" activado.
- Bit 13 Modo VLC INTER alternativo (AIV, *alternative INTER VLC*), opcional, (véase el anexo S), "0" desactivado, "1" activado.
- Bit 14 Modo cuantificación modificada (MQ, *modified quantization*), opcional, (véase el anexo T), "0" desactivado, "1" activado.
- Bit 15 Igual a "1" para evitar la emulación del código de comienzo.
- Bit 16 Reservado, será igual a "0".
- Bit 17 Reservado, será igual a "0".
- Bit 18 Reservado, será igual a "0".

#### 5.1.4.3 Parte obligatoria de PLUSPTYPE cuando PLUSPTYPE está presente (MPPTYPE, *mandatory part of PLUSPTYPE when PLUSPTYPE present*) (9 bits)

Independientemente del valor de UFEP, también están presentes en PLUSPTYPE los 9 bits siguientes:

- Bits 1-3 Código de tipo de imagen:  
"000" Imagen I (INTRA),  
"001" Imagen P (INTER),  
"010" Trama PB mejorada (véase el anexo M),  
"011" Imagen B (véase el anexo O),  
"100" Imagen EI (véase el anexo O),  
"101" Imagen EP (véase el anexo O),  
"110" Reservado,  
"111" Reservado.
- Bit 4 Modo repetición de muestreo de imagen de referencia (RPR, *reference picture resampling*), opcional, (véase el anexo P), "0" desactivado, "1" activado.
- Bit 5 Modo actualización de resolución reducida (RRU, *reduced-resolution update*), opcional, (véase el anexo Q), "0" desactivado, "1" activado.
- Bit 6 Tipo de redondeo (RTYPE, *rounding type*), (véase 6.1.2).
- Bit 7 Reservado, deberá ser igual a "0".

- Bit 8 Reservado, deberá ser igual a "0".
- Bit 9 Igual a "1" para evitar emulación del código de comienzo.

El codificador debe controlar el tipo de redondeo de modo que las imágenes P, tramas PB mejoradas, e imágenes EP tengan valores del bit 6 diferentes (tipo de redondeo para imágenes P) con respecto a sus imágenes de referencia para compensación de movimiento. El bit 6 puede tener un valor cualquiera si la imagen de referencia es una imagen I o una imagen EI. Sólo se puede fijar a "1" cuando los bits 1-3 indiquen una imagen P, trama PB mejorada, o imagen EP. Para otros tipos de imágenes, este bit se fijará siempre a "0".

#### 5.1.4.4 Semántica de PLUSPTYPE

La parte obligatoria de PLUSPTYPE contiene características que probablemente se cambien de una imagen a otra. Dichas características están representadas principalmente por los bits que indican el tipo de imagen entre imágenes I, P, PB mejorada, B, EI y EP. (Cabe señalar que no puede utilizarse el modo tramas PB del anexo G si PLUSPTYPE está presente; en su lugar debe utilizarse el modo tramas PB mejoradas del anexo M.) También se incluyen, no obstante, indicaciones para la utilización de los modos RPR y RRU, que pueden cambiar asimismo de imagen a imagen.

Las características que probablemente permanezcan en uso en vez de ser cambiadas de una imagen a otra (salvo en aspectos obvios como se indica en 5.1.4.5), han sido ubicadas en la parte opcional de PLUSPTYPE. Cuando UFEP es 000, la información de modo faltante se infiere del tipo de imagen y de la información de modo enviada en un PLUSPTYPE anterior con UFEP igual a 001.

Si PLUSPTYPE está presente pero UFEP es 000, se tiene que:

- 1) Para una imagen P o trama PB mejorada (véase el anexo M), la relación de aspecto de píxel, la anchura de imagen y la altura de imagen permanecen inalteradas con respecto a las de la imagen de referencia.
- 2) Para una imagen B de escalabilidad temporal (véase el anexo O) en una capa de mejora, el número de capa de referencia (RLNUM, *reference layer number*) es el mismo que el número de capa de mejora (ELNUM, *enhancement layer number*) si la última imagen enviada en la capa de mejora era una imagen EI o EP. Si la última imagen enviada en la capa de mejora era una imagen B el número de capa de referencia es el mismo que el número de capa de referencia de la última imagen B. La relación de aspecto de píxel, la anchura de la imagen y la altura de una imagen B, permanecen inalteradas con respecto a las de la imagen de capa de referencia temporalmente subsiguiente.

Se señala que si en la misma capa de mejora existen imágenes EI o EP temporalmente adyacentes como la imagen B, el RLNUM (explícito o implícito) será siempre igual al ELNUM. Se señala además que la relación de aspecto de píxel, anchura de imagen y altura de imagen de una imagen B (explícita o implícita) serán siempre iguales a las de su imagen de capa de referencia temporalmente subsiguiente.

- 3) Para una imagen EP de escalabilidad SNR/espacial (véase el anexo O), la relación de aspecto de píxel, la anchura de la imagen y la altura de la imagen permanecen inalteradas con respecto a las de la imagen de referencia temporalmente previa en la misma capa de mejora.

#### 5.1.4.5 Restricciones de modo para determinados tipos de imagen y reglas de inferencia de modo

Determinados modos no se aplican a ciertos tipos de imágenes. Las restricciones son, en concreto, como sigue:

- 1) No aplicación de los modos siguientes con imágenes I (INTRA): vector de movimiento sin restricción (véase el anexo D), predicción avanzada (véase en anexo F), VLC INTER

alternativo (véase el anexo S), repetición de muestreo de imagen de referencia (véase el anexo P), y actualización de resolución reducida (véase el anexo Q).

- 2) No aplicación de los modos siguientes con imágenes B (véase el anexo O): codificación aritmética basada en la sintaxis (véase el anexo E), filtro de desbloqueo (véase al anexo J), y predicción avanzada (véase el anexo F).
- 3) No aplicación de los modos siguientes con imágenes EI (véase el anexo O): vector de movimiento sin restricción (véase el anexo D), codificación aritmética basada en la sintaxis (véase el anexo E), predicción avanzada (véase el anexo F), repetición de muestreo de imagen de referencia (véase el anexo P), actualización de resolución reducida (véase el anexo Q), y VLC INTER alternativa (véase el anexo S).
- 4) No aplicación de los modos siguientes con imágenes EP (véase el anexo O): codificación aritmética basada en la sintaxis (véase el anexo E) y predicción avanzada (véase el anexo F).

Uno o más de los modos indicados en la lista anterior de cuatro ítems puede tener una bandera de modo con un valor "1" en la parte opcional de PLUSPTYPE dentro de una imagen de un tipo que está prohibida para ese modo (tipos I, B, EI, o EP). Esta condición está permitida y se interpretará sujeta a las reglas de inferencia de modo que se indican en el párrafo siguiente.

Los estados de modo están sujetos a las siguientes reglas de inferencia de modo:

- 1) Una vez que la banda de modo se haya fijado a "1" en la parte opcional de PLUSPTYPE, a la imagen vigente y a cada imagen subsiguiente en el tren de bits se les asignará el estado de "activado" para ese modo.
- 2) Se asignará un estado inferido de "desactivado" a cualquier modo no aplicable dentro de una imagen que tenga el código de tipo de imagen vigente. Sin embargo, cada imagen subsiguiente en el tren de bits tendrá un estado inferido de "activado" para ese modo (a menos que esto también produzca un conflicto obvio, que se resolverá del mismo modo). En el caso de trenes de bits escalables en capas (véase el anexo O), el estado del modo sólo se inferirá desde dentro de la misma capa del tren de bits.
- 3) La inferencia de estado continuará hasta que se envíe una imagen en la misma capa que contenga la parte opcional de PLUSPTYPE o que no contenga PLUSPTYPE en absoluto. Si se envía una nueva imagen que contiene la parte opcional de PLUSPTYPE, el estado enviado en el nuevo mensaje invalidará el estado antiguo. Si se envía una imagen que no contiene PLUSPTYPE (una imagen en la que los bits 6-8 de PTYPE no son "111"), se asignará el estado de "desactivado" a todos los modos no fijados explícitamente a "activado" en el campo PTYPE, y todos los modos continuarán teniendo un estado inferido de "desactivado" hasta que se envíe una nueva imagen que contenga la parte opcional de PLUSPTYPE.
- 4) Dos modos no requieren inferencia de estado de modo, ya que las banderas de modo de estos modos aparecen en la parte obligatoria de PLUSPTYPE. Se trata del modo repetición muestreo de imagen de referencia (véase el anexo P) y el modo actualización de resolución reducida (véase el anexo Q). La bandera de modo de cualquiera de estos modos no se fijará a menos que la imagen vigente permita el uso del modo. Por ejemplo, el bit del modo actualización de resolución reducida no se fijará en una imagen INTRA.

#### **5.1.4.6 Restricciones de interacción de modo**

Hay algunos modos que no se pueden utilizar en combinación con otros determinados modos.

- 1) El modo codificación aritmética basada en la sintaxis (véase el anexo E) no se debe utilizar con el modo VLC INTER alternativa (véase el anexo S) o el modo cuantificación modificada (véase el anexo T).

- 2) Si está presente PLUSPTYPE, el modo vector de movimiento sin restricción (véase el anexo D) no se debe utilizar con el modo codificación aritmética basada en la sintaxis (véase el anexo E).
- 3) El modo decodificación de segmento independiente (véase el anexo R) no se debe utilizar con el modo repetición de muestreo de imagen de referencia (véase el anexo P).
- 4) El modo de codificación de segmento independiente (véase el anexo R) no se utilizará con el modo estructura en rebanada sin la utilización simultánea del submodo rebanada rectangular del modo estructura en rebanada (véase el anexo K).

#### 5.1.4.7 Ubicación en el encabezamiento de imagen de CPM (1 bit) y PSBI (2 bits)

La ubicación de los campos CPM y PSBI en el encabezamiento de imagen depende de si PLUSPTYPE está presente o no (véanse 5.1.20 y 5.1.21). Si PLUSPTYPE está presente, CPM sigue inmediatamente después de PLUSPTYPE en el encabezamiento de imagen. Si PLUSPTYPE no está presente, CPM figura inmediatamente después de PQUANT en el encabezamiento de imagen. PSBI siempre sigue inmediatamente a CPM (si CPM = "1").

#### 5.1.5 Formato de imagen personalizado (CPFMT, *custom picture format*) (23 bits)

Palabra de código de longitud fija de 23 bits que sólo está presente si en el campo PLUSPTYPE se señala la utilización de un formato de imagen personalizado y UFEP es "001". Cuando está presente, CPFMT está constituido por:

- Bits 1-4 Código de relación de aspecto de píxel: un índice de 4 bits para el valor PAR en el cuadro 5. Para PAR ampliada, la relación de aspecto de píxel exacta se especificará en EPAR (véase 5.1.6).
- Bits 5-13 Indicación de anchura de imagen: gama [0, ..., 511]; número de píxels por línea =  $(PWI + 1) * 4$ .
- Bit 14 Igual a "1" para evitar la emulación del código de comienzo.
- Bits 15-23 Indicación de altura de imagen: gama [1, ..., 288]; número de líneas =  $PHI * 4$ .

**Cuadro 5/H.263 – Definición de código PAR**

Código PAR	Relación de aspecto de píxel
0000	Prohibido
0001	1:1 (cuadrado)
0010	12:11 (CIF para imagen 4:3)
0011	10:11 (tipo 525 para imagen 4:3)
0100	16:11 (CIF expandido para imagen 16:9)
0101	40:33 (tipo 525 expandido para imagen 16:9)
0110-1110	Reservado
1111	PAR ampliada

#### 5.1.6 Relación de aspecto de píxel ampliada (EPAR, *extended pixel aspect ratio*) (16 bits)

Palabra de código de longitud fija de 16 bits que sólo está presente si CPFMT también lo está y en el mismo se indica PAR ampliada. Cuando está presente, EPAR consiste en:

- Bits 1-8 Anchura del PAR: "0" está prohibido. Representación binaria natural de la anchura de PAR.
- Bits 9-16 Altura de PAR: "0" está prohibido. Representación binaria natural de la altura de PAR.

La anchura y altura de PAR deben ser números primos entre sí.

### **5.1.7 Código de frecuencia de reloj de imagen personalizada (CPCFC, *custom picture clock frequency code*) (8 bits)**

Palabra de código de longitud fija de 8 bits que sólo está presente si PLUSPTYPE también lo está y UFEP es 001 y en PLUSPTYPE se señala una frecuencia de reloj de imagen personalizada. Cuando está presente, CPCFC está constituido por:

- Bit 1 Código de conversión de reloj: "0" indica un factor de conversión de reloj de 1000 y "1" indica 1001.
- Bits 2-8 Divisor de reloj: "0" está prohibido. Representación binaria natural del valor del divisor de reloj.

La frecuencia de reloj de imagen personalizada viene dada por  $1\ 800\ 000 / (\text{divisor de reloj} * \text{factor de conversión de reloj})$  Hz.

El contador de referencia temporal contará unidades de la inversa de la frecuencia de reloj de imagen, en segundos. Cuando PCF cambia de la especificada para la imagen anterior, la referencia temporal para la imagen vigente se mide en términos de la PCF anterior, de modo tal que la nueva PCF tiene efecto sólo para la interpretación de la referencia temporal de las imágenes futuras.

### **5.1.8 Referencia temporal ampliada (ETR, *extended temporal reference*) (2 bits)**

Palabra de código de longitud fija de 2 bits que sólo está presente si se utiliza una frecuencia de reloj de imagen personalizada (independientemente del valor de UFEP). Son los dos bits menos significativos del número de 10 bits definido en 5.1.2.

### **5.1.9 Indicador de vectores de movimiento sin restricción no limitado (UUI, *unlimited unrestricted motion vectors indicator*) (Longitud variable)**

Palabra de código de longitud variable de 1 ó 2 bits que sólo está presente si en PLUSPTYPE se indica el modo vector de movimiento sin restricción y UFEP es 001. Cuando UUI está presente indica la limitación efectiva de la gama de vectores de movimiento que se utilizan.

- UUI = "1": La gama de vectores de movimiento está limitada conforme a los cuadros D.1 y D.2.
- UUI = "01": La gama de vectores de movimiento no está limitada salvo por el tamaño de la imagen.

### **5.1.10 Bits del submodo estructura en rebanada (SSS, *slice structured submode*) (2 bits)**

Palabra de código de longitud fija de 2 bits que sólo está presente si en PLUSPTYPE se indica el modo opcional estructura en rebanada (véase el anexo K) y UFEP es 001. Si se utiliza el modo estructura en rebanada pero UFEP no es 001, quedarán en efecto los últimos valores enviados para SSS.

- Bit 1 Rebanadas rectangulares, "0" indica rebanadas libres, "1" indica rebanadas rectangulares.
- Bit 2 Ordenamiento de rebanadas arbitrario, "0" indica orden secuencial, "1" indica orden arbitrario.

### **5.1.11 Número de capa de mejora (ELNUM, *enhancement layer number*) (4 bits)**

Palabra de código de longitud fija de 4 bits que sólo está presente si se utiliza el modo escalabilidad temporal, SNR, y espacial opcional (independientemente del valor de UFEP). Una determinada capa de mejora se identifica por un número de capa de mejora, ELNUM. La correspondencia de imágenes entre capas se efectúa a través de la referencia temporal. El tamaño de la imagen se indica en cada capa de mejora utilizando los campos de formato de fuente existentes o bien se infiere por



la relación con la capa de referencia. La primera capa de mejora por encima de la capa básica se denomina capa de mejora número 2, y la capa básica tiene el número 1.

#### **5.1.12 Número de capa de referencia (RLNUM, *reference layer number*) (4 bits)**

Palabra de código de longitud fija de 4 bits que sólo está presente si se utiliza el modo escalabilidad temporal, SNR, y espacial opcional (véase el anexo O) y UFEP es 001. El número de capa para las imágenes utilizadas como anclajes de referencia se identifica mediante un número de capa de referencia (RLNUM). La correspondencia de tiempo entre capas se efectúa por medio de la referencia temporal.

Cabe señalar que para imágenes B en una capa de mejora con imágenes EI o EP temporalmente adyacentes que estén presentes en la misma capa de mejora, RLNUM será igual a ELNUM (véase el anexo O).

#### **5.1.13 Banderas de modo de selección de imagen de referencia (RPSMF, *reference picture selection mode flags*) (3 bits)**

Palabra de código de longitud fija de 3 bits que sólo está presente si se utiliza el modo selección de imagen de referencia y UFEP es 001. Cuando está presente, RPSMF indica qué tipo de mensajes de canal de retorno necesita el codificador. Si se utiliza el modo selección de imagen de referencia pero RPSMF no está presente, quedará en efecto el último valor enviado de RPSMF.

- 100: no son necesarias señales ACK ni señales NACK;
- 101: es necesario devolver señales ACK;
- 110: es necesario devolver señales NACK;
- 111: es necesario devolver señales ACK y señales NACK;
- 000-011: reservada.

#### **5.1.14 Referencia temporal para indicación de predicción (TRPI, *temporal reference for prediction indication*) (1 bit)**

Palabra de código de longitud fija de 1 bit que sólo está presente si se utiliza el modo selección de imagen de referencia opcional (independientemente del valor UFEP). Cuando está presente, TRPI indica la presencia del campo TRP:

- 0: el campo TRP no está presente;
- 1: el campo TRP está presente.

TRPI será 0 cuando el encabezamiento de imagen indique una imagen I o EI.

#### **5.1.15 Referencia temporal para predicción (TRP, *temporal reference for prediction*) (10 bits)**

Cuando está presente (como se indica en TRPI), TRP indica la referencia temporal que se utiliza para la predicción de la codificación, salvo si se trata de imágenes B. Para las imágenes B, la imagen que tiene la referencia temporal TRP se utiliza para la predicción en el sentido hacia adelante. (La predicción en el sentido temporal inverso utiliza siempre la imagen inmediata temporalmente subsiguiente.) TRP es un número de 10 bits. Si no se utilizó una frecuencia de reloj de imagen personalizada para la imagen de referencia, los dos bits más significativos (MSB) de la TRP son 0 y los bits menos significativos (LSB) contienen la TR de 8 bits hallada en el encabezamiento de imagen de la imagen de referencia. Si se utilizó una frecuencia de reloj de imagen personalizada para la imagen de referencia, TRP es un número de 10 bits formado por la concatenación de ETR y TR a partir del encabezamiento de imagen de referencia.

Cuando TRP no esté presente, se utilizará la imagen de anclaje temporalmente previa más reciente, como cuando no se está en el modo selección de imagen de referencia. TRP es válido hasta el próximo PSC, GSC o SSC.

### **5.1.16 Indicación de mensaje de canal de retorno (BCI, *back-channel message indication*) (Longitud variable)**

Campo de longitud variable de uno o dos bits que sólo está presente si se utiliza el modo selección de imagen de referencia opcional. Cuando está fijado a "1", este campo señala la presencia del siguiente campo opcional mensaje en canal de retorno (BCM) de vídeo. El valor "01" indica la ausencia o el final del campo mensaje en canal de retorno de vídeo. Las combinaciones de BCM y BCI pueden no estar presentes, y pueden repetirse cuando están presentes. BCI se fijará a "01" si no se utiliza el submodo múltiplex de vídeo del modo selección de imagen de referencia opcional.

### **5.1.17 Mensaje de canal de retorno (BCM, *back-channel message*) (Longitud variable)**

Mensaje de canal de retorno con la sintaxis especificada en N.4.2, que sólo está presente si el campo BCI precedente también lo está y está fijado a "1".

### **5.1.18 Parámetros de repetición de muestreo de imagen de referencia (RPRP, *reference picture resampling parameters*) (Longitud variable)**

Campo de longitud variable que sólo está presente si se fija en PLUSPTYPE el bit del modo repetición de muestreo de imagen de referencia opcional. Este campo lleva a los parámetros del modo repetición de muestreo de imagen de referencia (véase el anexo P). Se señala que el modo repetición de muestreo de imagen de referencia también puede ser invocado implícitamente por la ocurrencia de un encabezamiento de imagen para una imagen codificada INTER cuyo tamaño de imagen difiera del tamaño de imagen codificada anterior, en cuyo caso el campo RPRP no está presente y no se fija el bit del modo repetición de muestreo de imagen de referencia.

### **5.1.19 Información de cuantificador (PQUANT, *quantizer information*) (5 bits)**

Palabra de código de longitud fija de 5 bits que indica el cuantificador QUANT que se utilizará para la imagen hasta que sea actualizada por cualquier GQUANT o DQUANT subsiguiente. Las palabras de código son la representación binaria natural de los valores de QUANT que, puesto que tienen un paso mitad, están comprendidos entre 1 y 31.

### **5.1.20 Multipunto de presencia continua y múltiplex de vídeo (CPM, *continuous presence multipoint and video multiplex*) (1 bit)**

Palabra de código de 1 bit que señala la utilización del modo multipunto de presencia continua y múltiplex de vídeo (CPM opcional); "0" indica desactivado, "1" indica activado. Para el uso de CPM véase el anexo C. CPM sigue inmediatamente después de PLUSPTYPE si PLUSPTYPE está presente, pero sigue a PQUANT en el encabezamiento de imagen si PLUSPTYPE no está presente.

### **5.1.21 Indicador de subtren de bits de imagen (PSBI, *picture sub-bitstream indicator*) (2 bits)**

Palabra de código de longitud fija de 2 bits que sólo está presente si el modo multipunto de presencia continua y múltiplex de vídeo es indicado por CPM. Las palabras de código son la representación binaria natural del número del subtren de bits para el encabezamiento de la imagen y toda la información siguiente hasta la próxima imagen o código de comienzo GOB (véase también el anexo C). PSBI sigue inmediatamente después de CPM si CPM es "1" (la ubicación de CPM y PSBI en el encabezamiento de la imagen depende de si PLUSPTYPE está presente).

### **5.1.22 Referencia temporal para imágenes B en tramas PB ( $TR_B$ , *temporal reference for B-pictures in PB-frames*) (3/5 bits)**

$TR_B$  está presente si PTYPE o PLUSPTYPE indica "trama PB" o "trama PB mejorada" (véanse también los anexos G y M), e indica el número de imágenes no transmitidas o sin referencia (a 29,97 Hz o la frecuencia de reloj de imagen personalizada en CPCFC) desde la última imagen P o I o la parte P de una trama PB o PB mejorada y antes de la parte de imagen PB de la trama PB o PB mejorada. La palabra de código es la representación binaria natural del número de imágenes no transmitidas más uno. Tiene una longitud de 3 bits para la frecuencia de reloj de imagen CIF

normalizada, longitud que se amplía a 5 bits cuando se utiliza una frecuencia de reloj de imagen personalizada. El número máximo de imágenes no transmitidas es de 6 para la frecuencia de reloj de imagen CIF normalizada y de 30 cuando se utiliza una frecuencia de reloj de imagen personalizada.

### 5.1.23 Información de cuantificación para imágenes B en tramas PB (DBQUANT, *quantization information for B-pictures in PB-frames*) (2 bits)

DBQUANT está presente si PTYPE o PLUSPTYPE indica "trama PB" o "trama PB mejorada" (véanse también los anexos G y M). En el proceso de decodificación, se obtiene un parámetro de cuantificación QUANT para cada macrobloque. Con tramas PB, QUANT se utiliza para el bloque P, mientras que para el bloque B se utiliza un parámetro de cuantificación diferente, BQUANT. QUANT está comprendido entre 1 y 31. DBQUANT indica la relación entre QUANT y BQUANT, como se define en el cuadro 6. En este cuadro, "/" significa división por truncamiento. BQUANT está comprendido entre 1 y 31; si el valor de BQUANT resultante del cuadro 6 es mayor que 31, se recorta a 31.

**Cuadro 6/H.263 – Códigos DBQUANT y relación entre QUANT y BQUANT**

DBQUANT	BQUANT
00	$(5 \times \text{QUANT})/4$
01	$(6 \times \text{QUANT})/4$
10	$(7 \times \text{QUANT})/4$
11	$(8 \times \text{QUANT})/4$

### 5.1.24 Información de inserción suplementaria (PEI, *extra insertion information*) (1 bit)

Un bit que, cuando se ha fijado a "1" señala la presencia del campo de datos opcional siguiente.

### 5.1.25 Información de mejora suplementaria (PSUPP, *supplemental enhancement information*) (0/8/16 ... bits)

Cuando PEI se fija a "1", le siguen 9 bits que consisten en 8 bits de datos (PSUPP) y luego otro bit PEI para indicar si siguen otros 9 bits, y así sucesivamente. Los codificadores utilizarán PSUPP como se especifica en el anexo L. Los decodificadores que no admiten las capacidades ampliadas descritas en el anexo L serán diseñados de modo que descarten PSUPP si PEI se fija a 1. Esto permite la compatibilidad hacia atrás para las capacidades ampliadas del anexo L de tal manera que un tren de bits que utiliza las capacidades ampliadas también puede ser utilizado sin alteración por los decodificadores que no admiten esas capacidades.

### 5.1.26 Relleno (ESTUF, *stuffing*) (Longitud variable)

Palabra de código de longitud variable que consiste en menos de 8 bits cero. Los codificadores pueden insertar esta palabra de código directamente antes de una palabra de código EOS. Los codificadores insertarán esta palabra de código según sea necesario para obtener la alineación en bytes obligatoria directamente antes de una palabra de código EOSBS. Si ESTUF está presente, el último bit de ESTUF será el último bit (el menos significativo) de un byte, de modo que el inicio de la palabra de código EOS o EOSBS esté alineada en bytes. Los decodificadores serán diseñados de modo que pueden descartar ESTUF. Véase el anexo C para la descripción de EOSBS y su utilización.

### 5.1.27 Fin de secuencia (EOS, *end of sequence*) (22 bits)

Palabra de código de 22 bits. Su valor es 0000 0000 0000 0000 1 11111. Este decodificador decidirá si inserta o no esta palabra de código. EOS puede estar alineado en bytes. Esto se puede

lograr insertando ESTUF antes del código EOS, de manera que el primer bit del código EOS sea el primer bit (el más significativo) de un byte. EOS no se repetirá salvo que aparezca al menos un código de comienzo de imagen entre cada par de códigos EOS.

### 5.1.28 Relleno (PSTUF, *stuffing*) (Longitud variable)

Palabra de código de longitud variable que consiste en menos de 8 bits cero. Los codificadores insertarán esta palabra de código para la alineación en byte de la siguiente PSC. El último bit de PSTUF será el último bit (el menos significativo) de un byte, de modo que el tren de bits de vídeo que incluye PSTUF es un múltiplo de 8 bits del primer bit en el tren de bits H.263. Los decodificadores se diseñarán de modo que puedan descartar PSTUF.

Si por algún motivo el codificador detiene la codificación de imágenes durante un determinado periodo de tiempo y reanuda la codificación ulteriormente, PSTUF se retransmitirá antes de que el codificador se detenga, para impedir que los últimos 7 bits de la imagen anterior no se envíen hasta que el codificador reanude la codificación.

## 5.2 Capa de grupo de bloques

Los datos de cada grupo de bloques (GOB) consisten en un encabezamiento GOB, seguido de datos de macrobloque. La estructura se muestra en la figura 9. Cada GOB contiene una o más filas de macrobloques. En el primer GOB (número 0), no se transmitirá el encabezamiento GOB. En todos los demás GOB, el encabezamiento GOB puede estar vacío, según la estrategia elegida por el codificador. Un decodificador puede señalar al codificador distante por medios externos, por ejemplo, Rec. UIT-T H.245, que transmita sólo encabezamientos GOB no vacíos. GSTUF puede estar presente cuando GBSC está presente. GN, GFID y GQUANT están presentes cuando GBSC está presente. GSBI está presente cuando CPM es "1" en el encabezamiento de imagen.



Figura 9/H.263 – Estructura de la capa de GOB

### 5.2.1 Relleno (GSTUF, *stuffing*) (Longitud variable)

Palabra de código de longitud variable que consiste en menos de 8 bits cero. Los codificadores pueden insertar esta palabra de código directamente antes de una palabra de código GBSC. Si GSTUF está presente, el último bit de GSTUF será el último bit (el menos significativo) de un byte, de modo que el comienzo de la palabra de código GBSC esté alineada en byte. Los decodificadores serán diseñados de modo que puedan descartar GSTUF.

### 5.2.2 Código de comienzo de grupo de bloques (GBSC, *group of block start code*) (17 bits)

Palabra de 17 bits. Su valor es 0000 0000 0000 0000 1. Los códigos de comienzo GOB pueden estar alineados en bytes. Esto se consigue insertando GSTUF antes del código de comienzo, de manera que el primer bit del código de comienzo es el primer bit (el más significativo) de un byte.

### 5.2.3 Número de grupo (GN, *group number*) (5 bits)

Palabra de código de longitud fija de 5 bits. Los bits son la representación binaria del número del grupo de bloques. En el caso del GOB numerado 0, el encabezamiento GOB que incluye GSTUF, GBSC, GN, GSBI, GFID y GQUANT está vacío; en el PSC se utiliza 0 como número de grupo. Los números de grupo 1 a 17 se utilizan en encabezamientos GOB de formatos de imagen normalizado. Los números de grupo 1 a 24 se utilizan en encabezamientos de GOB de formatos de imagen personalizados. Los números de grupo 16 a 28 son emulados en el encabezamiento de rebanada (véase el anexo K) cuando CPM = "0", y los números de grupo 25-27 y 29 son emulados en el encabezamiento de rebanada (véase el anexo K) cuando CPM = "1". El número de grupo 31 se utiliza en el código EOS, y el número de grupo 30 se utiliza en el código EOSBS.

#### 5.2.4 Indicador de subtren de bits de GOB (GSBI, *GOB sub-bitstream indicator*) (2 bits)

Palabra de código de longitud fija de 2 bits que sólo está presente si CPM es "1" en el encabezamiento de imagen. Las palabras de código son la representación binaria natural del número de subtren de bits para el encabezamiento GOB y toda la información siguiente hasta el próximo código de comienzo de imagen o de GOB (véase también el anexo C).

#### 5.2.5 Identificador de trama GOB (GFID, *GOB frame ID*) (2 bits)

Palabra de código de longitud fija de 2 bits. GFID tendrá el mismo valor en cada encabezamiento GOB (o rebanada) de una imagen dada. Además, si el PTYPE indicado en un encabezamiento de imagen es el mismo que el de la imagen transmitida previamente, GFID tendrá el mismo valor que el de esa imagen anterior, siempre que PLUSPTYPE no esté presente. Sin embargo, si PTYPE en un determinado encabezamiento de imagen difiere del PTYPE en el encabezamiento de imagen transmitida anteriormente, el valor para GFID en esa imagen diferirá del valor en la imagen previa.

Si PLUSPTYPE está presente, el valor de GFID será el mismo que el de la imagen anterior (en la misma capa) si el campo PTYPE, y PLUSPTYPE, y todos los campos presentes entre CPFMT, EPAR, CPCFC, SSS, ELNUM, RLNUM, UUI, RPSMF y RPRP quedan en efecto como para la imagen anterior; de no ser así, el valor de GFID será distinto del de la imagen previa.

#### 5.2.6 Información de cuantificador (GQUANT, *quantizer information*) (5 bits)

Palabra de código de longitud fija de 5 bits, que indica el cuantificador QUANT que se utilizará para la parte restante de la imagen hasta que sea actualizado por cualquier GQUANT o DQUANT subsiguiente. Las palabras de código son la representación binaria natural de los valores de QUANT que, puesto que tiene un paso mitad, están comprendidos en la gama de 1 a 31.

### 5.3 Capa de macrobloque

Los datos de cada macrobloque consisten en un encabezamiento de macrobloque, seguido de los datos de bloque. La estructura aparece en la figura 10. COD sólo está presente en las imágenes que no son de tipo 'INTRA' para cada macrobloque de esas imágenes. MCBPC está presente cuando lo indica COD o cuando la imagen es de tipo 'INTRA'. MODB está presente para el tipo de macrobloque 0-4 si PTYPE indica "trama PB". CBPY, DQUANT, MVD y MVD<sub>2-4</sub> están presentes cuando lo indica MCBPC. CBPB y MVDB están presentes sólo si lo indica MODB. "Datos de bloque" está presente cuando lo indican MCBPC y CBPY. MVD<sub>2-4</sub> sólo están presentes cuando se utiliza el modo predicción avanzada (véase el anexo F) o el modo filtro de desbloqueo (véase el anexo J). MODB, CBPB y MVDB sólo están presentes en el modo tramas PB (véase el anexo G). Para la codificación de los símbolos del modo codificación aritmética basada en sintaxis, véase el anexo E. Para la codificación de la capa de macrobloque en imágenes B, EI, y EP, véase el anexo O.

COD	MCBPC	MODB	CBPB	CBPY	DQUANT	MVD	MVD <sub>2</sub>	MVD <sub>3</sub>	MVD <sub>4</sub>	MVDB	Datos de bloque
-----	-------	------	------	------	--------	-----	------------------	------------------	------------------	------	-----------------

Figura 10/H.263 – Estructura de la capa de macrobloque

#### 5.3.1 Indicación de macrobloque codificado (COD, *coded macroblock indication*) (1 bit)

Un bit que, cuando está puesto a "0", señala que el bloque está codificado. Cuando está puesto a "1", no se transmite más información para este macrobloque en este caso, el decodificador tratará el macrobloque como un bloque INTER con vector de movimiento para todo el bloque igual a cero y sin datos de coeficiente. COD sólo está presente en las imágenes que no son de tipo "INTRA", en cada macrobloque de esas imágenes.

NOTA – En el modo predicción avanzada (véase el anexo F), se realiza también la compensación de movimiento para bloque con superposición si COD se fija a "1"; y en el modo filtro de desbloqueo (véase el

anexo J), el filtrado de desbloqueo puede afectar también a los valores de algunos pixels de macrobloques cuya COD se ha fijado a "1".

### 5.3.2 Tipo de macrobloque y patrón de bloque codificado para la crominancia (MCBPC, *macroblock type & coded block pattern for chrominance*) (Longitud variable)

MCBPC es una palabra de código de longitud variable que dé información sobre el tipo de macrobloque y el patrón de grupo codificado para la crominancia. Las palabras clave para MCBPC figuran en los cuadros 7 y 8. MCBPC se incluye siempre en macrobloques de código.

**Cuadro 7/H.263 – Tabla de códigos de longitud variable (VLC) para MCBPC (imágenes I)**

Índice	Tipo de macrobloque	CBPC (56)	Número de bits	Código
0	3	00	1	1
1	3	01	3	001
2	3	10	3	010
3	3	11	3	011
4	4	00	4	0001
5	4	01	6	0000 01
6	4	10	6	0000 10
7	4	11	6	0000 11
8	Relleno	–	9	0000 0000 1

**Cuadro 8/H.263 – Tabla de códigos de longitud variable (VLC) para MCBPC (imágenes P)**

Índice	Tipo de macrobloque	CBPC (56)	Número de bits	Código
0	0	00	1	1
1	0	01	4	0011
2	0	10	4	0010
3	0	11	6	0001 01
4	1	00	3	011
5	1	01	7	0000 111
6	1	10	7	0000 110
7	1	11	9	0000 0010 1
8	2	00	3	010
9	2	01	7	0000 101
10	2	10	7	0000 100
11	2	11	8	0000 0101
12	3	00	5	0001 1
13	3	01	8	0000 0100
14	3	10	8	0000 0011
15	3	11	7	0000 011

**Cuadro 8/H.263 – Tabla de códigos de longitud variable (VLC) para MCBPC (imágenes P)**

Índice	Tipo de macrobloque	CBPC (56)	Número de bits	Código
16	4	00	6	0001 00
17	4	01	9	0000 0010 0
18	4	10	9	0000 0001 1
19	4	11	9	0000 0001 0
20	Relleno	–	9	0000 0000 1
21	5	00	11	0000 0000 010
22	5	01	13	0000 0000 0110 0
23	5	10	13	0000 0000 0111 0
24	5	11	13	0000 0000 0111 1

En las tablas de relleno de bits se dispone de una palabra de código adicional. Esta palabra de código debe ser descartada por los decodificadores. Si los bits 1-3 de MPPTYPE indican una trama PB mejorada y los bits 1-3 de OPPTYPE indican formato de fuente personalizado, MBA no deberá indicar relleno antes del primer macrobloque de la imagen (para evitar la emulación del código de comienzo).

NOTA – Se deben diseñar los codificadores de manera que el tipo de macrobloque indique la existencia de relleno de bits inmediatamente antes de la ubicación de un código de comienzo de rebanada, GOB o imagen en el tren de bits. Sin embargo, los codificadores no deben utilizar el relleno de la capa de macrobloques de esta manera (para mantener compatibilidad con decodificadores que fueron diseñados antes de que se determinara que era necesario tener decodificadores que soportaran esta característica).

El tipo de macrobloque da información sobre el macrobloque y los elementos de datos que están presentes. En los cuadros 9 y 10 se indican los tipos de macrobloque y los elementos incluidos. El tipo de macrobloque 5 (índices 21-24 en el cuadro 8) no estará presente a menos que sí lo esté un PTYPE ampliado (PLUSPTYPE) en el encabezamiento de la imagen y se utilice el modo predicción avanzada (véase el anexo F) o bien el modo filtro de desbloqueo (véase el anexo J), y no estará presente para el primer macrobloque de una imagen. Además, los codificadores no permitirán que un código MCBPC de macrobloque de tipo 5 siga inmediatamente a siete ceros consecutivos en el tren de bits (lo que pueden provocar códigos INTRADC particulares seguidos de COD = 0), con objeto de evitar la emulación del código de comienzo. Los códigos de macrobloque de tipo 5 pueden estar precedidos por bits de relleno cuando sea necesario para satisfacer este requisito (para macrobloques distintos del primero de una imagen).

**Cuadro 9/H.263 – Tipos de macrobloque y elementos de datos incluidos para las imágenes normales**

Tipo de imagen	Tipo de MB	Nombre	COD	MCBPC	CBPY	DQUANT	MVD	MVD <sub>2,4</sub>
INTER	No codificado	–	X					
INTER	0	INTER	X	X	X		X	
INTER	1	INTER+Q	X	X	X	X	X	
INTER	2	INTER4V	X	X	X		X	X
INTER	3	INTRA	X	X	X			

**Cuadro 9/H.263 – Tipos de macrobloque y elementos de datos incluidos para las imágenes normales**

Tipo de imagen	Tipo de MB	Nombre	COD	MCBPC	CBPY	DQUANT	MVD	MVD <sub>2-4</sub>
INTER	4	INTRA+Q	X	X	X	X		
INTER	5	INTER4V+Q	X	X	X	X	X	X
INTER	Relleno	–	X	X				
INTRA	3	INTRA		X	X			
INTRA	4	INTRA+Q		X	X	X		
INTRA	Relleno	–		X				

NOTA – "X" significa que el ítem está presente en el macrobloque.

**Cuadro 10/H.263 – Tipos de macrobloque y elementos de datos incluidos para las tramas PB**

Tipo de imagen	Tipo de MB	Nombre	COD	MCBPC	MODB	CBPY	CBPB	DQUANT	MVD	MVDB	MVD <sub>2,4</sub>
INTER	No codificado	–	X								
INTER	0	INTER	X	X	X	X	(X)		X	(X)	
INTER	1	INTER+Q	X	X	X	X	(X)	X	X	(X)	
INTER	2	INTER4V	X	X	X	X	(X)		X	(X)	X
INTER	3	INTRA	X	X	X	X	(X)		X	(X)	
INTER	4	INTRA+Q	X	X	X	X	(X)	X	X	(X)	
INTER	5	INTER4V+Q	X	X	X	X	(X)	X	X	(X)	X
INTER	Relleno	–	X	X							

NOTA 1 – "X" significa que el ítem está presente en el macrobloque.

NOTA 2 – CBPB et MVDB sólo están presentes si lo indica MODB.

NOTA 3 – Los bloques B siempre están codificados en modo INTER, incluso si el tipo de macrobloque del macrobloque PB indica INTRA.

El patrón de bloque codificado de crominancia indica bloques  $C_B$  y/o  $C_R$  cuando se transmite al menos un coeficiente de transformada no INTRADC (INTRADC es el coeficiente dc para bloques INTRA, véase 5.4.1), salvo que se utilice el modo opcional codificación INTRA avanzada.  $CBPC_N = 1$  si cualquier coeficiente no INTRADC está presente para el bloque N, de lo contrario 0, para  $CBPC_5$  y  $CBPC_6$  en el patrón de bloque codificado. Si se utiliza codificación INTRA avanzada, el uso es similar pero se indica el coeficiente INTRADC de la misma manera que los otros coeficientes (véase el anexo I). La numeración de bloques se ofrece en la figura 5. Cuando  $MCBPC =$  relleno, se salta la parte restante de la capa de macrobloque. En este caso, el  $COD = 0$  anterior no está relacionado con ningún macrobloque codificado o no codificado y, por consiguiente, no se incrementa el número de macrobloques. En el caso de las imágenes P, los rellenos múltiples se consiguen mediante conjuntos múltiples de  $COD = 0$  y  $MCBPC =$  relleno. Véanse los cuadros 7 y 8.



### 5.3.3 Modo macrobloque para bloques B (MODB, *macroblock mode for B-blocks*) (Longitud variable)

MODB está presente para MB-tipo 0-4 si PTYPE indica "trama PB" y es una palabra de código de longitud variable que indica si CBPB está presente (indica que se transmiten coeficientes B para este macrobloque) y/o MVDB está presente. En el cuadro 11 se definen las palabras de código para MODB. El campo MODB se codifica de manera diferente para tramas PB mejoradas, como se especifica en el anexo M.

**Cuadro 11/H.263 – Tabla de códigos de longitud variable (VLC) para MODB**

Índice	CBPB	MVDB	Número de bits	Código
0			1	0
1		X	2	10
2	X	X	2	11

NOTA – "X" significa que el ítem está presente en el macrobloque.

### 5.3.4 Patrón de bloque codificado para bloque B (CBPB, *coded block pattern for B-blocks*) (6 bits)

CBPB sólo está presente en el modo tramas PB si lo indica MODB.  $CBPB_N = 1$  si cualquier coeficiente está presente para N bloques B, de lo contrario, 0 para cada bit  $CBPB_N$  del patrón de bloque codificado. La numeración de bloques se ofrece en la figura 5, y el bit del extremo izquierdo de CBPB corresponde al número de bloque 1.

### 5.3.5 Patrón de bloque codificado para la luminancia (CBPY, *coded block pattern for luminance*) (Longitud variable)

Palabra de código de longitud variable, que proporciona un número de patrón que se refiere a aquellos bloques Y del macrobloque para los que se transmite al menos un coeficiente de transformada no INTRADC (INTRADC es el coeficiente dc para bloques INTRA, véase 5.4.1), salvo que se utilice el modo codificación INTRA avanzada. Si se utiliza codificación INTRA avanzada, se indica INTRADC de la misma manera que los otros coeficientes (véase el anexo I).

$CBPY_N = 1$  si cualquier coeficiente no INTRADC está presente para el bloque N, de lo contrario, 0 para cada bit  $CBPY_N$  del patrón de bloque codificado. La numeración de bloques se ofrece en la figura 5, y el bit del extremo izquierdo de CBPY corresponde al número de bloque 1. Con un patrón  $CBPY_N$  determinado se utilizan diferentes palabras de código según si el modo es INTER o INTRA, como se indica en el cuadro 12.

**Cuadro 12/H.263 – Tabla de códigos de longitud variable (VLC) para CBPY**

Índice	CBPY(INTRA) (12, 34)	CBPY(INTER) (12, 34)	Número de bits	Código
0	00 00	11 11	4	0011
1	00 01	11 10	5	0010 1
2	00 10	11 01	5	0010 0
3	00 11	11 00	4	1001
4	01 00	10 11	5	0001 1

**Cuadro 12/H.263 – Tabla de códigos de longitud variable (VLC) para CBPY**

Índice	CBPY(INTRA) (12, 34)	CBPY(INTER) (12, 34)	Número de bits	Código
5	01 01	10 10	4	0111
6	01 10	10 01	6	0000 10
7	01 11	10 00	4	1011
8	10 00	01 11	5	0001 0
9	10 01	01 10	6	0000 11
10	10 10	01 01	4	0101
11	10 11	01 00	4	1010
12	11 00	00 11	4	0100
13	11 01	00 10	4	1000
14	11 10	00 01	4	0110
15	11 11	00 00	2	11

### 5.3.6 Información de cuantificador (DQUANT, *quantizer information*) (2 bits/Longitud variable)

Si no se utiliza el modo cuantificación modificada, DQUANT es un código de dos bits para definir un cambio en QUANT. En el cuadro 13 se ofrecen los valores diferenciales de las diferentes palabras de código. QUANT tiene una gama de 1 a 31; si después de añadir el valor diferencial, el valor de QUANT es inferior a 1 o superior a 31, se lo recorta a 1 y 31, respectivamente. Si se utiliza el modo cuantificación modificada, DQUANT es una palabra de código de longitud variable como se especifica en el anexo T.

**Cuadro 13/H.263 – Códigos DQUANT y valores diferenciales para QUANT**

Índice	Valor diferencial	DQUANT
0	-1	00
1	-2	01
2	1	10
3	2	11

### 5.3.7 Datos de vector de movimiento (MVD, *motion vector data*) (Longitud variable)

MVD se incluye en todos los macrobloques INTER (en el modo trama PB, también en los macrobloques INTRA), y consiste en una palabra de código de longitud variable para la componente horizontal, seguida de una palabra de código de longitud variable para la componente vertical. Los códigos de longitud variable se ofrecen en el cuadro 14. Si se utiliza el modo vector de movimiento sin restricción y PLUSPTYE está presente los vectores de movimiento se codifican empleando el cuadro D.3 en vez del cuadro 14 (véase el anexo D).

**Cuadro 14/H.263 – Tabla de códigos de longitud variable (VLC) para MVD**

Índice	Vector	Diferencias	Número	Códigos
0	-16	16	13	0000 0000 0010 1
1	-15,5	16,5	13	0000 0000 0011 1
2	-15	17	12	0000 0000 0101
3	-14,5	17,5	12	0000 0000 0111
4	-14	18	12	0000 0000 1001
5	-13,5	18,5	12	0000 0000 1011
6	-13	19	12	0000 0000 1101
7	-12,5	19,5	12	0000 0000 1111
8	-12	20	11	0000 0001 001
9	-11,5	20,5	11	0000 0001 011
10	-11	21	11	0000 0001 101
11	-10,5	21,5	11	0000 0001 111
12	-10	22	11	0000 0010 001
13	-9,5	22,5	11	0000 0010 011
14	-9	23	11	0000 0010 101
15	-8,5	23,5	11	0000 0010 111
16	-8	24	11	0000 0011 001
17	-7,5	24,5	11	0000 0011 011
18	-7	25	11	0000 0011 101
19	-6,5	25,5	11	0000 0011 111
20	-6	26	11	0000 0100 001
21	-5,5	26,5	11	0000 0100 011
22	-5	27	10	0000 0100 11
23	-4,5	27,5	10	0000 0101 01
24	-4	28	10	0000 0101 11
25	-3,5	28,5	8	0000 0111
26	-3	29	8	0000 1001
27	-2,5	29,5	8	0000 1011
28	-2	30	7	0000 111
29	-1,5	30,5	5	0001 1
30	-1	31	4	0011
31	-0,5	31,5	3	011
32	0		1	1
33	0,5	-31,5	3	010
34	1	-31	4	0010
35	1,5	-30,5	5	0001 0

**Cuadro 14/H.263 – Tabla de códigos de longitud variable (VLC) para MVD**

Índice	Vector	Diferencias	Número	Códigos
36	2	-30	7	0000 110
37	2,5	-29,5	8	0000 1010
38	3	-29	8	0000 1000
39	3,5	-28,5	8	0000 0110
40	4	-28	10	0000 0101 10
41	4,5	-27,5	10	0000 0101 00
42	5	-27	10	0000 0100 10
43	5,5	-26,5	11	0000 0100 010
44	6	-26	11	0000 0100 000
45	6,5	-25,5	11	0000 0011 110
46	7	-25	11	0000 0011 100
47	7,5	-24,5	11	0000 0011 010
48	8	-24	11	0000 0011 000
49	8,5	-23,5	11	0000 0010 110
50	9	-23	11	0000 0010 100
51	9,5	-22,5	11	0000 0010 010
52	10	-22	11	0000 0010 000
53	10,5	-21,5	11	0000 0001 110
54	11	-21	11	0000 0001 100
55	11,5	-20,5	11	0000 0001 010
56	12	-20	11	0000 0001 000
57	12,5	-19,5	12	0000 0000 1110
58	13	-19	12	0000 0000 1100
59	13,5	-18,5	12	0000 0000 1010
60	14	-18	12	0000 0000 1000
61	14,5	-17,5	12	0000 0000 0110
62	15	-17	12	0000 0000 0100
63	15,5	-16,5	13	0000 0000 0011 0

**5.3.8 Datos de vector de movimiento (MVD<sub>2-4</sub>, *motion vector data*) (Longitud variable)**

Las tres palabras de código MVD<sub>2-4</sub> están incluidas si lo indica PTYPE y MCBPC, y cada una consiste en una palabra de código de longitud variable para el componente horizontal, seguida de una palabra de código de longitud variable para el componente vertical de cada vector. Los códigos de longitud variable se ofrecen en el cuadro 14. MVD<sub>2-4</sub> sólo está presente en el modo predicción avanzada (véase el anexo F) o en el modo filtro de desbloqueo (véase el anexo J).

### 5.3.9 Datos de vector de movimiento para macrobloque B (MVDB, *motion vector data for B-macroblock*) (Longitud variable)

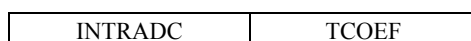
MVDB sólo está presente en el modo tramas PB o en modo tramas PB mejoradas si lo indica MODB, y consiste en una palabra de código de longitud variable para el componente horizontal, seguida de una palabra de código de longitud variable para el componente vertical de cada vector. Los códigos de longitud variable se ofrecen en el cuadro 14. Para el uso de MVDB, véanse los anexos G y M.

### 5.4 Capa de bloque

Si no se utiliza el modo tramas PB, un macrobloque comprende cuatro bloques de luminancia y uno de cada uno de los dos bloques de diferencia de color (véase la figura 5). La estructura de la capa de bloque se muestra en la figura 11. INTRADC está presente para cada bloque del macrobloque si MCBPC indica macrobloque tipo 3 ó 4 (véanse los cuadros 7 y 8). TCOEF está presente si es indicado por MCBPC o CBPY.

En el modo tramas PB, un macrobloque comprende doce bloques. Primero los datos para los seis bloques P se transmiten como en el modo H.263 por defecto, después los datos para los seis bloques B. INTRADC está presente para cada bloque P del macrobloque si MCBPC indica macrobloque tipo 3 ó 4 (véanse los cuadros 7 y 8). INTRADC no está presente para los bloques B. TCOEF está presente para los bloques B si es indicado por MCBPC o CBPY. TCOEF está presente para los bloques B si es indicado por CBPB.

Para la codificación de los símbolos en el modo codificación aritmética basada en sintaxis, véase el anexo E.



**Figura 11/H.263 – Estructura de la capa de bloque**

#### 5.4.1 Coeficiente DC de bloque INTRA (INTRADC, *DC coefficient for INTRA blocks*) (8 bits)

Palabra de código de 8 bits. El código 0000 0000 no se utiliza. El código 1000 0000 no se utiliza, y el nivel de reconstrucción de 1024 se codifica como 1111 1111 (véase el cuadro 15).

**Cuadro 15/H.263 – Niveles de reconstrucción del coeficiente DC para el modo INTRA**

Índice	FLC	Nivel de reconstrucción a transformada inversa
0	0000 0001 (1)	8
1	0000 0010 (2)	16
2	0000 0011 (3)	24
.	.	.
.	.	.
126	0111 1111 (127)	1016
127	1111 1111 (255)	1024
128	1000 0001 (129)	1032
.	.	.
.	.	.

**Cuadro 15/H.263 – Niveles de reconstrucción del coeficiente DC para el modo INTRA**

Índice	FLC	Nivel de reconstrucción a transformada inversa
252	1111 1101 (253)	2024
253	1111 1110 (254)	2032

**5.4.2 Coeficiente de la transformada (TCOEF, *transform coefficient*) (Longitud variable)**

Los EVENTOS (EVENTS) que ocurren con más frecuencia se codifican con los códigos de longitud variable que aparecen en el cuadro 16. El último bit "s" denota el signo de nivel: "0" si es positivo y "1" si es negativo.

Un EVENT es una combinación de la última indicación de coeficiente no cero [ÚLTIMO (LAST); "0": no hay más coeficientes no cero en este bloque, "1": éste es el último coeficiente no cero de este bloque], el número de ceros sucesivos que preceden al coeficiente codificado [PASADA (RUN)], y el último valor no cero del coeficiente codificado [NIVEL (LEVEL)].

Las combinaciones restantes de (LAST, RUN y LEVEL) se codifican con una palabra de 22 bits, que consiste en 7 bits ESCAPE (ESCAPE), 1 bit LAST, 6 bits RUN y 8 bits LEVEL. El uso de esta palabra de 22 bits para codificar las combinaciones enumeradas en el cuadro 16 no está prohibido. En la palabra LEVEL de 8 bits, el código 0000 0000 está prohibido, y el código 1000 0000 también está prohibido salvo que se utilice el modo cuantificación modificada (véase anexo T). Los códigos para RUN y LEVEL figuran en el cuadro 17.

**Cuadro 16/H.263 – Tabla de códigos de longitud variable (VLC) para TCOEF**

Índice	LAST	RUN	LEVEL	Bits	Código VLC
0	0	0	1	3	10s
1	0	0	2	5	1111s
2	0	0	3	7	0101 01s
3	0	0	4	8	0010 111s
4	0	0	5	9	0001 1111s
5	0	0	6	10	0001 0010 1s
6	0	0	7	10	0001 0010 0s
7	0	0	8	11	0000 1000 01s
8	0	0	9	11	0000 1000 00s
9	0	0	10	12	0000 0000 111s
10	0	0	11	12	0000 0000 110s
11	0	0	12	12	0000 0100 000s
12	0	1	1	4	110s
13	0	1	2	7	0101 00s
14	0	1	3	9	0001 1110s
15	0	1	4	11	0000 0011 11s
16	0	1	5	12	0000 0100 001s
17	0	1	6	13	0000 0101 0000s
18	0	2	1	5	1110s

**Cuadro 16/H.263 – Tabla de códigos de longitud variable (VLC) para TCOEF**

Índice	LAST	RUN	LEVEL	Bits	Código VLC
19	0	2	2	9	0001 1101s
20	0	2	3	11	0000 0011 10s
21	0	2	4	13	0000 0101 0001s
22	0	3	1	6	0110 1s
23	0	3	2	10	0001 0001 1s
24	0	3	3	11	0000 0011 01s
25	0	4	1	6	0110 0s
26	0	4	2	10	0001 0001 0s
27	0	4	3	13	0000 0101 0010s
28	0	5	1	6	0101 1s
29	0	5	2	11	0000 0011 00s
30	0	5	3	13	0000 0101 0011s
31	0	6	1	7	0100 11s
32	0	6	2	11	0000 0010 11s
33	0	6	3	13	0000 0101 0100s
34	0	7	1	7	0100 10s
35	0	7	2	11	0000 0010 10s
36	0	8	1	7	0100 01s
37	0	8	2	11	0000 0010 01s
38	0	9	1	7	0100 00s
39	0	9	2	11	0000 0010 00s
40	0	10	1	8	0010 110s
41	0	10	2	13	0000 0101 0101s
42	0	11	1	8	0010 101s
43	0	12	1	8	0010 100s
44	0	13	1	9	0001 1100s
45	0	14	1	9	0001 1011s
46	0	15	1	10	0001 0000 1s
47	0	16	1	10	0001 0000 0s
48	0	17	1	10	0000 1111 1s
49	0	18	1	10	0000 1111 0s
50	0	19	1	10	0000 1110 1s
51	0	20	1	10	0000 1110 0s
52	0	21	1	10	0000 1101 1s
53	0	22	1	10	0000 1101 0s
54	0	23	1	12	0000 0100 010s
55	0	24	1	12	0000 0100 011s
56	0	25	1	13	0000 0101 0110s

**Cuadro 16/H.263 – Tabla de códigos de longitud variable (VLC) para TCOEF**

Índice	LAST	RUN	LEVEL	Bits	Código VLC
57	0	26	1	13	0000 0101 0111s
58	1	0	1	5	0111s
59	1	0	2	10	0000 1100 1s
60	1	0	3	12	0000 0000 101s
61	1	1	1	7	0011 11s
62	1	1	2	12	0000 0000 100s
63	1	2	1	7	0011 10s
64	1	3	1	7	0011 01s
65	1	4	1	7	0011 00s
66	1	5	1	8	0010 011s
67	1	6	1	8	0010 010s
68	1	7	1	8	0010 001s
69	1	8	1	8	0010 000s
70	1	9	1	9	0001 1010s
71	1	10	1	9	0001 1001s
72	1	11	1	9	0001 1000s
73	1	12	1	9	0001 0111s
74	1	13	1	9	0001 0110s
75	1	14	1	9	0001 0101s
76	1	15	1	9	0001 0100s
77	1	16	1	9	0001 0011s
78	1	17	1	10	0000 1100 0s
79	1	18	1	10	0000 1011 1s
80	1	19	1	10	0000 1011 0s
81	1	20	1	10	0000 1010 1s
82	1	21	1	10	0000 1010 0s
83	1	22	1	10	0000 1001 1s
84	1	23	1	10	0000 1001 0s
85	1	24	1	10	0000 1000 1s
86	1	25	1	11	0000 0001 11s
87	1	26	1	11	0000 0001 10s
88	1	27	1	11	0000 0001 01s
89	1	28	1	11	0000 0001 00s
90	1	29	1	12	0000 0100 100s
91	1	30	1	12	0000 0100 101s
92	1	31	1	12	0000 0100 110s
93	1	32	1	12	0000 0100 111s
94	1	33	1	13	0000 0101 1000s



**Cuadro 16/H.263 – Tabla de códigos de longitud variable (VLC) para TCOEF**

Índice	LAST	RUN	LEVEL	Bits	Código VLC
95	1	34	1	13	0000 0101 1001s
96	1	35	1	13	0000 0101 1010s
97	1	36	1	13	0000 0101 1011s
98	1	37	1	13	0000 0101 1100s
99	1	38	1	13	0000 0101 1101s
100	1	39	1	13	0000 0101 1110s
101	1	40	1	13	0000 0101 1111s
102	ESCAPE			7	0000 011

**Cuadro 17/H.263 – Tabla de códigos de longitud fija (FLC) para RUN Y LEVEL**

Índice	Run	Código
0	0	000 000
1	1	000 001
2	2	000 010
.	.	.
.	.	.
63	63	111 111

Índice	Level	Código
–	–128	Véase el texto
0	–127	1000 0001
.	.	.
125	–2	1111 1110
126	–1	1111 1111
–	0	PROHIBIDO
127	1	0000 0001
128	2	0000 0010
.	.	.
253	127	0111 1111

## 6 Proceso de decodificación

### 6.1 Compensación de movimiento

En esta subcláusula se describe la compensación de movimiento del modo de predicción H.263 por defecto. Para una descripción de la compensación de movimiento en el modo vector de movimiento sin restricción, véase el anexo D. Para una descripción de la compensación de movimiento en el modo predicción avanzada, véase el anexo F. Para una descripción de la compensación de movimiento en el modo actualización de resolución reducida, véase el anexo Q.

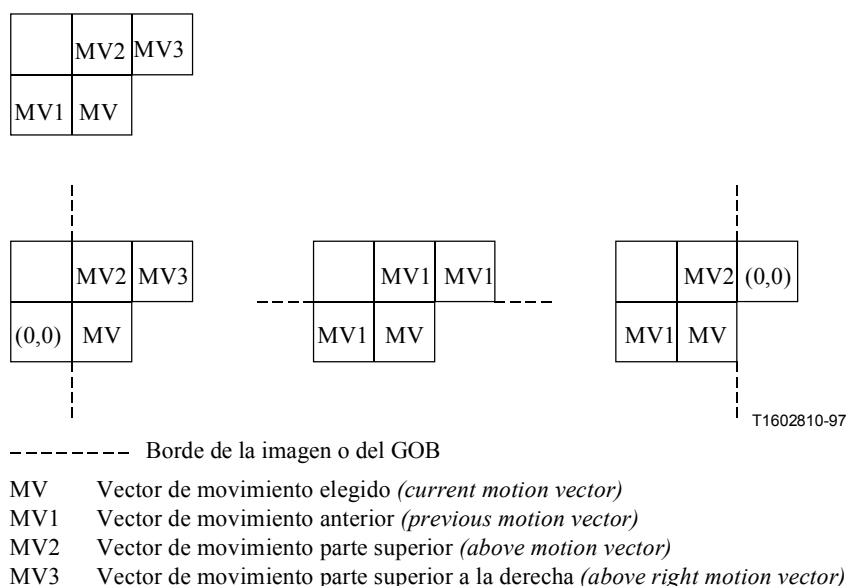
#### 6.1.1 Vectores de movimiento diferenciales

El vector de macrobloque se obtiene añadiendo predictores a las diferencias vectoriales indicadas por MVD (véanse los cuadros 14 y D.3). Para la codificación diferencial de cuatro vectores por macrobloque, véase el anexo F. En el caso de un vector por macrobloque, los predictores candidatos para la codificación diferencial se toman de los tres macrobloques adyacentes, como se indica en la figura 12. Los predictores se calculan por separado para las componentes horizontal y vertical.

En los casos especiales que se dan en los bordes de la imagen, la rebanada, o el GOB objeto de predicción, rigen las reglas de decisión siguientes, en orden creciente:

- 1) Cuando el macrobloque correspondiente se codificó en modo INTRA (si no está en el modo tramas PB con predicción bidireccional) o no se codificó (COD = 1), el predictor candidato se fija a cero.
- 2) El predictor candidato MV1 se fija a cero si el macrobloque correspondiente está fuera de la imagen o de la rebanada (a la izquierda).
- 3) A continuación, los predictores candidatos MV2 y MV3 se fijan a MV1 si los macrobloques correspondientes están fuera de la imagen (parte superior) o fuera de GOB (parte superior) si el encabezamiento GOB del GOB objeto de predicción no está vacío; o fuera de la rebanada cuando está en el modo estructuración en rebanada.
- 4) A continuación, el predictor candidato MV3 se fija a cero, si el macrobloque correspondiente está fuera de la imagen (a la derecha).

Para cada componente, el predictor es la mediana de los tres predictores candidatos para esa componente.



**Figura 12/H.263 – Predicción del vector de movimiento**

Se aprovecha el hecho de que la gama de valores de componentes del vector de movimiento está limitada. Cada palabra VLC representa un par de valores diferenciales. Sólo un elemento del par arrojará un componente de vector de macrobloque que cae dentro de la gama permitida  $[-16, 15,5]$ . Un valor positivo del componente horizontal o vertical del vector de movimiento significa que la predicción se forma a partir de los píxels de la imagen anterior, que están situados a la derecha o por debajo de los píxels objeto de predicción. Si se utiliza el modo vector de movimiento sin restricción (véase el anexo D), la decodificación de los vectores de movimiento se efectuará como se especifica en D.2.

El vector de movimiento se usa para todos los píxels de todos los cuatro bloques de luminancia del macrobloque. Los vectores de movimiento de ambos bloques de crominancia se obtienen dividiendo los valores componentes del vector de macrobloque por dos, debido al formato más bajo de la crominancia. Los valores componentes de los vectores resultantes con resolución de un cuarto de píxel se modifican hacia la posición más cercana de mitad de píxel, como se indica en el cuadro 18.

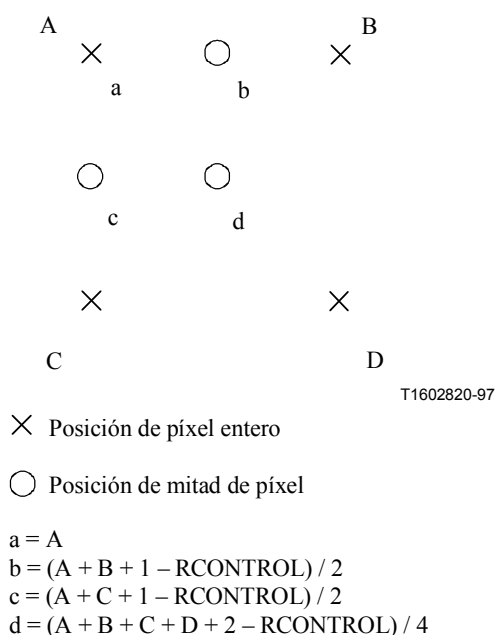
**Cuadro 18/H.263 – Modificación de los componentes de vectores de crominancia con resolución de un cuarto de píxel**

Posición de un cuarto de píxel	0	1/4	1/2	3/4	1
Posición resultante	0	1/2	1/2	1/2	1

### 6.1.2 Interpolación para la predicción de subpíxel

Los valores de mitad de píxel se obtienen mediante interpolación bilineal, como se describe en la figura 13. El signo "/" indica división por truncamiento.

El valor de RCONTROL es igual al valor del bit tipo de redondeo (RTYPE) (bit 6) en MPPTYPE (véase 5.1.4.3), cuando el campo formato de fuente (bits 6-8) en PTYPE indica "PTYPE ampliado". De otro modo, RCONTROL tiene un valor implícito de 0. Independientemente del bit RTYPE, el valor de RCONTROL se fija a 0 para la parte B de las tramas PB mejoradas (véase el anexo M).



**Figura 13/H.263 – Predicción de mitad de píxel por interpolación bilineal**

## 6.2 Coeficientes de decodificación

### 6.2.1 Cuantificación inversa

En esta subcláusula se describe el proceso de cuantificación inversa, excepto para el caso en que se utiliza el modo codificación INTRA avanzada opcional (véase el anexo I). Si LEVEL = "0", el nivel de reconstrucción es REC = "0". El nivel de reconstrucción de INTRADC viene dado por el cuadro 15. Los niveles de reconstrucción de todos los coeficientes no cero diferentes de INTRADC vienen dados por las fórmulas siguientes:

$$|REC| = QUANT \cdot (2 \cdot |LEVEL| + 1) \quad \text{si } QUANT = \text{"impar"}$$

$$|REC| = QUANT \cdot (2 \cdot |LEVEL| + 1) - 1 \quad \text{si } QUANT = \text{"par"}$$

Nótese que este proceso no autoriza ni siquiera los números con valor. Se ha observado que esto previene la acumulación de errores por desajuste de la transformada discreta de coseno inversa (IDCT). Después de calcular |REC|, se añade el signo para obtener REC: REC = sign(LEVEL) · |REC|.

Sign(LEVEL) viene dado por el último bit del código TCOEF (véase el cuadro 16) o por el cuadro 17.

### 6.2.2 Recorte de los niveles de reconstrucción

Después de la cuantificación inversa, se recortan los niveles de reconstrucción de todos los coeficientes diferentes de INTRADC para que caigan en la gama -2048 a 2047.

### 6.2.3 Colocación en zigzag

Los coeficientes de la transformada cuantificados se colocan en un bloque de  $8 \times 8$  conforme a la secuencia que aparece en la figura 14 salvo que se utilice el modo opcional codificación INTRA avanzada (véase el anexo I). El coeficiente 1 es el coeficiente dc.

1	2	6	7	15	16	28	29
3	5	8	14	17	27	30	43
4	9	13	18	26	31	42	44
10	12	19	25	32	41	45	54
11	20	24	33	40	46	53	55
21	23	34	39	47	52	56	61
22	35	38	48	51	57	60	62
36	37	49	50	58	59	63	64

**Figura 14/H.263 – Colocación en zigzag de los coeficientes de la transformada cuantificados**

### 6.2.4 Transformada inversa

Después de la cuantificación inversa y la colocación en zigzag de los coeficientes, los bloques de  $8 \times 8$  resultantes se procesan con una transformada discreta de coseno bidimensional separable de dimensión  $8 \times 8$ . La salida de la transformada inversa varía entre -256 y +255 después del recorte para poder ser representada mediante 9 bits. La función de transferencia de la transformada inversa viene dada por:

$$f(x, y) = \frac{1}{4} \sum_{u=0}^7 \sum_{v=0}^7 C(u)C(v)F(u, v) \cos \left[ \pi(2x+1) \frac{u}{16} \right] \cos \left[ \pi(2y+1) \frac{v}{16} \right]$$

siendo  $u, v, x, y = 0, 1, 2, \dots, 7$

donde:

- $x, y =$  coordenadas espaciales en el dominio de píxel;
- $u, v =$  coordenadas en el dominio de la transformada;
- $C(u) =$   $1 / \sqrt{2}$  para  $u = 0, 1$  en los demás casos;
- $C(v) =$   $1 / \sqrt{2}$  para  $v = 0, 1$  en los demás casos.

NOTA – En los bloques sometidos a transformación,  $x = 0$  e  $y = 0$  se refieren a los píxels más próximos a los bordes izquierdo y superior de la imagen, respectivamente.

Los procedimientos aritméticos para calcular la transformada inversa no se definen, pero ésta debe mantenerse dentro de las tolerancias de error especificadas en el anexo A.

## 6.3 Reconstrucción de los bloques

### 6.3.1 Resumen

Después de la compensación de movimiento y la decodificación de los coeficientes (incluida la transformada inversa), se hace una reconstrucción de cada bloque de luminancia y crominancia. En el caso de los bloques INTRA, la reconstrucción es igual al resultado de la transformación inversa. En el caso de los bloques INTER, la reconstrucción se efectúa sumando la predicción y el resultado de la transformación inversa. La suma se efectúa píxel por píxel. Para el resumen en el modo actualización de resolución reducida, véase el anexo Q.

### 6.3.2 Recorte

Para evitar que la distorsión de cuantificación de las amplitudes de los coeficientes de transformada cause desbordamiento aritmético en los bucles del codificador y decodificador, se insertan funciones de recorte. El recortador actúa después de la suma de la predicción, y si el error de predicción reconstruido de los valores de píxel resultantes es inferior a 0 o superior a 255, los cambia a 0 y 255, respectivamente.

## Anexo A

### Especificación de la exactitud de la transformada inversa

**A.1** Se generan valores de datos de píxel enteros aleatorios en la gama de  $-L$  a  $+H$  de acuerdo con el generador de números aleatorios que se ofrece a continuación (versión "C"). Se disponen en bloques de  $8 \times 8$ . Deben generarse conjuntos de datos de 10 000 bloques cada uno para  $(L = 256, H = 255)$ ,  $(L = H = 5)$  y  $(L = H = 300)$ .

**A.2** Para cada bloque de  $8 \times 8$  se calcula la transformada discreta directa de coseno separable, ortonormal y de multiplicación matricial con una exactitud de coma flotante de 64 kbit/s por lo menos:

$$F(u, v) = \frac{1}{4} C(u) C(v) \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos \left[ \pi(2x+1) \frac{u}{16} \right] \cos \left[ \pi(2y+1) \frac{v}{16} \right]$$

con  $u, v, x, y = 0, 1, 2, \dots, 7$

donde:

$x, y$  = coordenadas espaciales en el dominio de píxel;

$u, v$  = coordenadas en el dominio de la transformada;

$C(u)$  =  $1/\sqrt{2}$  para  $u = 0, 1$  en los demás casos;

$C(v)$  =  $1/\sqrt{2}$  para  $v = 0, 1$  en los demás casos.

**A.3** Para cada bloque se redondean a los valores enteros más cercanos los 64 coeficientes resultantes de la transformada. Se recortan a la gama de  $-2048$  a  $+2047$ , lo que constituye los datos de entrada de 12 bits para la transformada inversa.

**A.4** Para cada bloque de  $8 \times 8$  de datos de 12 bits producido según A.3, se calcula una transformada discreta de coseno inversa (IDCT, *inverse discrete cosine transform*) separable, ortonormal y de multiplicación matricial con una exactitud de coma flotante de 64 bits por lo menos. Se redondean los píxels resultantes al valor entero más cercano y se recortan para que caigan en la gama de  $-256$  a  $+255$ . Estos bloques de  $8 \times 8$  píxels constituyen los datos de salida de la IDCT de referencia.

**A.5** Para cada bloque de  $8 \times 8$  de datos producido según A.3, se aplica la IDCT objeto de prueba y se recorta la salida para que caiga en la gama de  $-256$  a  $+255$ . Estos bloques de  $8 \times 8$  píxels constituyen los datos de salida de la IDCT de prueba.

**A.6** Para cada uno de los 64 píxels de salida de la IDCT y cada uno de los conjuntos de datos de 10 000 bloques generados anteriormente, se miden el error máximo, el error medio y el error cuadrático medio entre los datos de referencia y los de prueba.

- A.7**
- Para cualquier píxel, el valor del error máximo no rebasará la unidad.
  - Para cualquier píxel, el valor del error cuadrático medio no rebasará 0,06.
  - En total, el valor del error cuadrático medio no rebasará 0,02.
  - Para cualquier píxel, el valor del error medio no rebasará 0,015.
  - En total, el valor del error medio no rebasará 0,0015.

**A.8** Una entrada todos ceros debe producir una salida todos ceros.

**A.9** Se vuelven a realizar las medidas utilizando exactamente los mismos valores de datos de A.1, pero cambiando el signo de cada píxel.

```
"C" program for random number generation
/* L and H shall be long, that is 32 bits */
long rand (L,H)
long      L,H;
{
    static long randx = 1;                /* long is 32 bits          */
    static double z = (double) 0x7fffffff;
    long i,j;
    double  x;                            /* double is 64 bits      */
    randx = (randx * 1103515245) + 12345;
    i = randx & 0x7fffffff;                /* keep 30 bits           */
    x = ( (double)i ) / z;                 /* range 0 to 0.99999 ... */
    x *= (L+H+1);                          /* range 0 to < L+H+1    */
    j = x;                                  /* truncate to integer    */
    return(j - L);                          /* range -L to H         */
}
```

## Anexo B

### Decodificador ficticio de referencia

El decodificador ficticio de referencia (HRD, *hypothetical reference decoder*) se define como sigue.

**B.1** El HRD y el codificador tienen la misma frecuencia de reloj y la misma frecuencia de reloj de imagen, y funcionan síncronamente.

**B.2** El tamaño de la memoria tampón receptora de HRD es  $(B + \text{BPPmaxKb} * 1024 \text{ bits})$  donde  $(\text{BPPmaxKb} * 1024)$  es el número máximo de bits por imagen que ha sido negociado para uso en el tren de bits (véase 3.6). El valor de B se define como sigue:

$$B = 4 \cdot R_{\text{máx}} / \text{PCF}$$

donde PCF es la frecuencia de reloj de imagen efectiva y  $R_{\text{máx}}$  es la velocidad binaria de vídeo máxima durante la conexión en bits por segundo. La frecuencia de reloj de imagen efectiva es la frecuencia de reloj de imagen CIF normalizada a menos que en el campo CPCFC del encabezamiento de imagen se especifique una PCF personalizada. Este valor para B es un mínimo. Un codificador puede utilizar un valor mayor para B a condición de que el número más grande sea negociado primero por medios externos, por ejemplo, aplicando la Rec. UIT-T H.245.

El valor para  $R_{\text{máx}}$  depende de la configuración del sistema (por ejemplo red telefónica general conmutada o RDSI, monoenlace o multienlace) y puede ser igual a la velocidad binaria máxima admitida por el enlace físico. La negociación de  $R_{\text{máx}}$  se efectúa por medios externos, por ejemplo, aplicando la Rec. UIT-T H.245.

**B.3** El HRD está vacío inicialmente.

**B.4** La memoria tampón del HRD se examina a intervalos de reloj de imagen (1000/PCF ms). Si por lo menos una imagen codificada completa está en la memoria tampón, todos los datos para la imagen anterior se suprimen instantáneamente (por ejemplo, a  $t_{n+1}$  en la figura B.1). Inmediatamente después de suprimir los datos anteriores, la ocupación de la memoria tampón debe ser menor que  $B$ . Esto es un requisito del tren de bits de salida del codificador que incluye datos de imagen codificados y MCBPC y relleno STUF pero no bits de alineación de trama para corrección de errores, indicador de relleno (Fi) bits de relleno o información de paridad para corrección de errores, descritos en el anexo H.

A los efectos de esta definición, a menos que se utilice el modo opcional escalabilidad temporal, SNR y espacial, una imagen codificada completa es una imagen I o P normal o una trama PB, o PB mejorada.

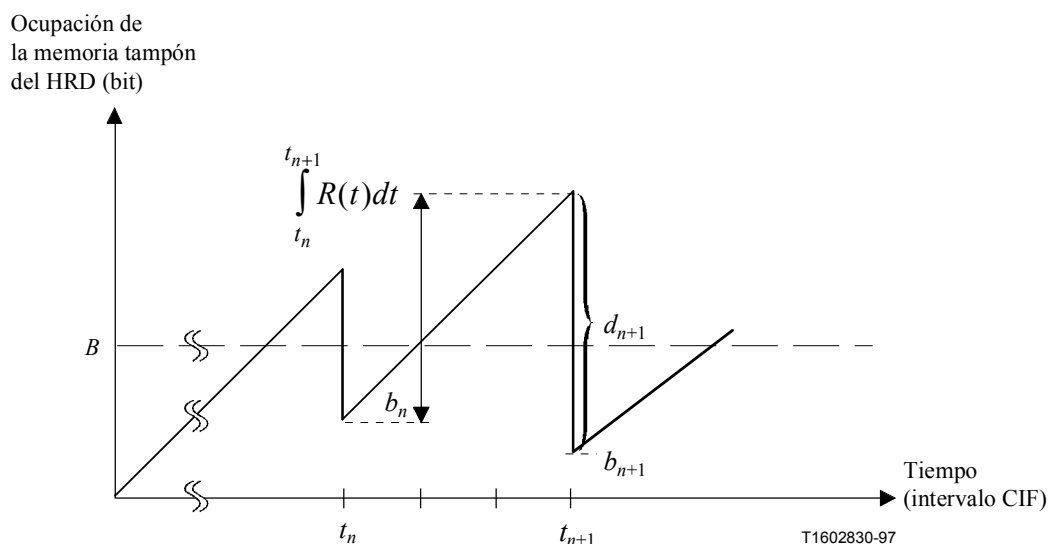
Cuando se utiliza el modo escalabilidad temporal, SNR y espacial (véase el anexo O), a cada capa de mejora se le da un HRD adicional para el que una imagen completa es una imagen EI, EP o B. La memoria tampón de la capa básica retendrá los bits del encabezamiento de imagen a medida que vayan llegando, hasta que haya llegado una porción del encabezamiento de imagen suficiente como para determinar si la imagen es una imagen de capa básica o de capa de mejora, y el número de la capa de mejora. Cuando pueda determinarse que la imagen que llega pertenece a una capa de mejora, todos los bits de esa imagen se transferirán instantáneamente al HRD de capa de mejora apropiado, y cualesquiera bits que lleguen posteriormente se seguirán colocando en el HRD de capa de mejora hasta que haya llegado una porción de un nuevo encabezamiento de imagen suficiente como para determinar que el tren de bits se ha de reencaminar a otra memoria tampón del HRD. El proceso de identificación de la capa de mejora es instantáneo y asíncrono, independiente de los tiempos de comprobación de los intervalos del reloj de imagen.

Para cumplir este requisito, el número de bits para la  $(n+1)$ ésima imagen codificada de  $d_{n+1}$ , debe satisfacer:

$$d_{n+1} \geq b_n + \int_{t_n}^{t_{n+1}} R(t) dt - B$$

donde:

- $b_n$  es la ocupación de la memoria tampón justamente después del instante  $t_n$ ;
- $t_n$  es el instante en el que la  $n$ ésima imagen codificada es suprimida de la memoria tampón del HRD;
- $R(t)$  es la velocidad binaria de vídeo en el instante  $t$ .



NOTA – El instante  $(t_{n+1} - t_n)$  es un número entero de periodos de imagen CIF (1/29. 97, 2/29. 97, 3/29. 97,...).

**Figura B.1/H.263 – Ocupación de la memoria tampón del HRD**

## Anexo C

### Consideraciones sobre el funcionamiento multipunto

Las facilidades que se indican a continuación se proporcionan para soportar el funcionamiento multipunto con conmutación.

#### C.1 Petición de imagen congelada

Hace que el decodificador congele la imagen visualizada hasta que se reciba una señal de liberación de imagen congelada o hasta que expire un periodo de temporización de al menos seis segundos. Esta señal se transmite por medios externos (por ejemplo, de la Rec. UIT-T H.245). Se señala que una instrucción similar puede ser enviada también utilizando información de mejora suplementaria en el encabezamiento de imagen del tren de bits de vídeo (véase L.4).

#### C.2 Petición de actualización rápida

Hace que el codificador codifique la imagen siguiente en modo INTRA con los parámetros de codificación adecuados para evitar el desbordamiento de la memoria tampón. Esta señal se transmite por medios externos (por ejemplo, de la Rec. UIT-T H.245).

#### C.3 Liberación de imagen congelada

Señal procedente de un codificador que ha respondido a una petición de actualización rápida y que permite al decodificador salir del modo de imagen congelada y presentar las imágenes decodificadas de manera normal. Esta señal es transmitida por PTYPE (véase 5.1.3) en el encabezamiento de imagen de la primera imagen codificada en respuesta a la petición de actualización rápida.

#### C.4 Multipunto de presencia continua (CPM) y múltiplex de vídeo

NOTA 1 – No se utiliza en la Rec. UIT-T H.324.



En la presente Recomendación se proporciona un modo multipunto de presencia continua y múltiplex de vídeo en el cual se puede multiplexar hasta cuatro trenes de bits cuarto de CIF H.263 independientes, como "subtrenes" independientes en un nuevo tren de vídeo con los campos PSBI, GSBI, SSBI y ESBI. El intercambio de capacidades para este modo se efectúa por medios externos (por ejemplo, la Rec. H.242). Se admiten datos de canal de retorno de selección de imagen de referencia para responder a subtrenes de bits utilizando campos BCPM y BSBI.

Cuando se funciona en el modo CPM el campo CPM se pondrá a "1" en cada uno de los subtrenes de bits H.263 independientes. Los subtrenes de bits se identifican mediante un número identificador de tren utilizando los indicadores de subtrenes de bits (SBI, *sub-bitstream indicator*) en la imagen y encabezamientos GOB o rebanada y encabezamientos EOSBS de cada tren de bits H.263. El SBI indica el número del tren de bits H.263 al cual pertenece ese encabezamiento y toda la información que sigue hasta la próxima imagen o encabezamiento GOB o encabezamiento de rebanada en el tren de bits de vídeo compuesto.

Cada subtren de bits se considera como un tren de bits H.263 normal y por consiguiente satisfará las capacidades intercambiadas por medios externos. La información para los diferentes trenes de bits H.263 no se transmite en ningún orden predefinido especial, un SBI puede tener cualquier valor independientemente de los SBI precedentes y las velocidades de imagen para los diferentes trenes de bits H.263 pueden ser diferentes. La información en cada tren de bits es también completamente independiente con respecto a la información en los otros trenes de bits. Por ejemplo, las palabras de código GFID en un subtren de bits no están influidas por las palabras de código GFID o PTYPE en otros subtrenes de bits. De manera similar, el modo inferencia de estado rige cuando se utiliza un tipo de imagen ampliada (PLUSPTYPE) en el encabezamiento de imagen y todos los demás aspectos del funcionamiento del tren de bits actuarán independientemente y separadamente para cada subtren de bits.

Se considera, por convenio, que el subtren de bits con el número identificador del subtren de bits más bajo (enviado en el SBI) tiene la máxima prioridad en situaciones en las que un conflicto entre requisitos de recursos haga necesaria la selección con prioridad (a menos que se establezca por medios externos un convenio de prioridades diferente).

Para marcar el final de cada subtren de bits del modo CPM, se proporciona una sintaxis como la que se muestra en la figura C.1, suponiendo que la capacidad de enviar esta sintaxis añadida se negocia primero por medios externos, (mientras que en la versión 1 de esta Recomendación se definió el funcionamiento CPM, el final de la sintaxis de subtren de bits se ha añadido en la versión 2, y por ello no se considera parte del funcionamiento CPM de la versión 1). El final de la sintaxis de subtren de bits (ESTUF + EOSBS + ESBI) marca el final de cada subtren de bits, en vez del final del tren en su totalidad como hace un código EOS.

NOTA 2 – La no capacidad de negociar funcionamiento de un subtren de bits CPM que se define aquí para la presente Rec. UIT-T H.263 se incluyó en algunas Recomendaciones UIT-T para terminales (por ejemplo, la Rec. UIT-T H.324) antes de que se creara la versión 2 de esta Recomendación. Así pues, cualquier negociación externa del funcionamiento CPM incluida en Recomendaciones futuras de la serie H implicará el soporte del final de la sintaxis de subtren de bits, a menos que en esa Recomendación se especifique otra cosa.

Al final de la sintaxis de subtren de bits hay tres partes. Tras la alineación en bytes obligatoria en la que se utiliza ESTUF, se envía una palabra de código EOSBS de 23 bits (lo que corresponde a un encabezamiento GOB con  $GN = 30$ , que de otro modo no se utiliza en la sintaxis, seguido de un bit único de valor cero que se reserva para uso futuro). La palabra de código EOSBS va seguida a continuación de una palabra de código ESBI de dos bits que indica qué subtren de bits se ve afectado. Este par de palabras de código significa que el envío de datos para el subtren de bits asociado ha terminado y que cualesquiera datos subsiguientes enviados para el mismo subtren de bits serán totalmente independientes de los que llegaron antes que la EOSBS. En particular, la imagen siguiente del subtren de bits después del código EOSBS no será una imagen INTER o

cualquier otro tipo de imagen que pueda utilizar la predicción temporal hacia adelante (se permite una imagen I o EI pero no en cambio una imagen P, trama PB, trama PB mejorada, imagen B o imagen EP).

La sintaxis de EOSBS y ESBI se describe en las subcláusulas que siguen. ESTUF se define en 5.1.26.



**Figura C.1/H.263 – Diagrama de sintaxis para indicadores de final de subtren de bits**

#### **C.4.1 Código de final de subtren de bits (EOSBS, *end of sub-bitstream code*) (23 bits)**

El código EOSBS es una palabra de código de 23 bits. Su valor es 0000 0000 0000 0000 1 11110 0. Corresponde al codificador decidir si se inserta o no esta palabra de código. No deberá enviarse EOSBS a menos que se haya enviado previamente como mínimo un encabezamiento de imagen para el mismo subtren de bits indicado en el campo ESBI que sigue, y no se enviará a menos que la capacidad de enviar EOSBS haya sido negociada por medios externos. Los códigos EOSBS deberán estar alineados en bytes. Esto se consigue insertando ESTUF antes del código de comienzo EOSBS de tal manera que el primer bit del código de comienzo EOSBS sea el primer bit (el más significativo) de un byte (véase 5.1.26).

El código EOSBS indica que se ha parado el envío de datos para el subtren de bits indicado y que se ha declarado terminado el tren de bits, hasta que comience de nuevo mediante la emisión de otro código de comienzo de imagen para ese subtren de bits. Las imágenes subsiguientes con el mismo número identificador de subtren de bits (ESBI) deberán ser completamente independientes de las imágenes enviadas antes del código EOSBS.

La información de control y de otro tipo asociada con el tren de bits de vídeo en general, sin especificación de a qué subtrenes de bits se pretende que se apliquen estos códigos (por ejemplo, una petición de imagen congelada o una petición de actualización rápida transmitida por medios de la Rec. UIT-T H.242), deberá suponerse que sólo se aplica a los subtrenes de bits *activos*. Se considera que un subtren de bits está activo si se ha recibido por lo menos un código de comienzo de imagen para dicho subtren de bits y los últimos datos enviados aplicables a ese subtren de bits no fueron un EOS o un EOSBS + ESBI.

#### **C.4.2 Indicador de subtren de bits de finalización (ESBI, *ending sub-bitstream indicator*) (2 bits)**

ESBI es una palabra de código de longitud fija de 2 bits que sigue inmediatamente al EOSBS. Indica el número de subtren de bits del subtren de bits de finalización. Su valor es la representación binaria natural del número de subtren de bits.

## Anexo D

### Modo vector de movimiento sin restricción

Este anexo describe el modo vector de movimiento sin restricción facultativo de la presente Recomendación. La capacidad de este modo se señala por medios externos (por ejemplo, aplicando la Rec. UIT-T H.245). La utilización de este modo se indica en PTYPE o PLUSPTYPE.

La gama de vectores de movimiento y la tabla de códigos de longitud variable (VLC) utilizadas para codificar las diferencias de vectores de movimiento para el modo vector de movimiento sin restricción dependen de si está presente el campo PLUSPTYPE del encabezamiento de imagen. Cuando PLUSPTYPE está presente, la gama de vectores de movimiento depende también del tamaño de la imagen y del valor del campo UUI del encabezamiento de imagen.

#### D.1 Vectores de movimiento en fronteras de imagen

En el modo predicción por defecto de la presente Recomendación, los vectores de movimiento están limitados, de manera que todos los píxels referenciados por ellos se encuentran dentro del área de la imagen codificada (véase 4.2.3). Ahora bien, en el modo vector de movimiento sin restricción se suprime esta restricción y, por tanto, los vectores de movimiento *están* autorizados a apuntar fuera de la imagen. Cuando un píxel referenciado mediante un vector de movimiento está fuera del área de la imagen codificada, se utiliza en su lugar un píxel de borde. Este píxel de borde se encuentra limitando el vector de movimiento a la última posición de píxel entero dentro del área de la imagen codificada. La limitación del vector de movimiento se realiza de píxel en píxel y por separado para cada componente del vector de movimiento.

Por ejemplo, si se utiliza el modo vector de movimiento sin restricción para una imagen cuarto de CIF, el valor de píxel referenciado para la componente de luminancia viene dado por la siguiente fórmula:

$$\text{Rumv}(x, y) = R(x', y')$$

donde:

$x, y, x', y'$  = coordenadas espaciales en el dominio de píxel;

$\text{Rumv}(x, y)$  = valor de píxel de la imagen referenciada en  $(x, y)$  en el modo vector de movimiento sin restricción;

$R(x', y')$  = valor de píxel de la imagen referenciada en  $(x', y')$  en el modo vector de movimiento sin restricción;

$$x' \begin{cases} = 0 & \text{si } x < 0; \\ = 175 & \text{si } x > 175; \\ = x & \text{en los demás casos;} \end{cases}$$

$$y' \begin{cases} = 0 & \text{si } y < 0; \\ = 143 & \text{si } y > 143; \\ = y & \text{en los demás casos;} \end{cases}$$

y la zona de imagen codificada de  $R(x', y')$  es  $0 \leq x' \leq 175$ ,  $0 \leq y' \leq 143$ . Las fronteras dadas son posiciones de píxel de enteros; sin embargo,  $(x', y')$  pueden ser también una mitad de posición de píxel dentro de estas fronteras.

### D.1.1 Restricciones de los valores de los vectores de movimiento

Si PLUSPTYPE está presente en el encabezamiento de imagen, los valores de los vectores de movimiento están restringidos de tal manera que ningún elemento de la región  $16 \times 16$  u  $(8 \times 8)$  seleccionada tendrá una distancia horizontal o vertical de más de 15 píxels fuera del área de imagen codificada. Se señala que esta es una gama de extrapolación más pequeña que cuando PLUSPTYPE no está presente.

NOTA 1 – Si PLUSPTYPE está ausente, la gama de extrapolación es como máximo de 31,5 píxels fuera del área de imagen codificada cuando se utiliza el modo vector de movimiento sin restricción, y de 16 píxels fuera del área de imagen codificada cuando se utiliza el modo de predicción avanzada (véase el anexo F) sin el modo vector de movimiento sin restricción.

NOTA 2 – Cuando se utiliza el modo predicción avanzada (véase el anexo F), el vector de movimiento de cada región de  $16 \times 16$  u  $(8 \times 8)$  afecta a una zona mayor, debido a la compensación de movimiento de bloques superpuestos. Esto puede hacer que la gama de extrapolación se incremente para los vectores de movimiento "distantes" del modo predicción avanzada, ya que el grado de superposición (4 píxels, u 8 píxels si se utiliza también el modo actualización de resolución reducida) se añade al grado de extrapolación requerido (incluso aunque la gama de valores permitidos para cada vector de movimiento sigue siendo la misma que cuando no se utiliza el modo predicción avanzada).

### D.2 Ampliación de la gama de vectores de movimiento

En el modo predicción por defecto, los valores para componentes horizontales y verticales de los vectores de movimiento están restringidos a la gama  $[-16, 15,5]$  (esto es válido también para los componentes de vectores de movimiento hacia adelante y hacia atrás para imágenes B). Sin embargo, en el modo vector de movimiento sin restricción se amplía la gama máxima para componentes de vectores.

Si el campo PLUSPTYPE no está presente en el encabezamiento de imagen la gama de vectores de movimiento se amplía a  $[-31,5, 31,5]$  con la restricción de que sólo se pueden alcanzar los valores que están dentro de una gama de  $[-16, 15,5]$  alrededor del predictor para cada componente de vector de movimiento si el predictor está en la gama  $[-15,5, 16]$ . Si el predictor está fuera de la gama  $[-15,5, 16]$  se pueden alcanzar todos los valores dentro de la gama  $[-31,5, 31,5]$  con el mismo signo que el predictor más el valor cero. De este modo, si  $MV_c$  es el componente de vector de movimiento y  $P_c$  es el predictor para el mismo,

$$\begin{array}{ll} -31,5 \leq MV_c \leq 0 & \text{si } -31,5 \leq P_c \leq -16 \\ -16 + P_c \leq MV_c \leq 15,5 + P_c & \text{si } -15,5 \leq P_c \leq 16 \\ 0 \leq MV_c \leq 31,5 & \text{si } 16,5 \leq P_c \leq 31,5 \end{array}$$

En el modo de vector de movimiento sin restricción, la interpretación del cuadro 14 para MVD,  $MVD_{2-4}$  y MVDB es como sigue:

- Si el predictor para el componente de vector de movimiento está en la gama  $[-15,5, 16]$  solamente se aplica la primera columna de las diferencias de vector.
- Si el predictor para el componente de vector de movimiento está fuera de la gama  $[-15,5, 16]$ , se utilizará la diferencia de vector del cuadro 14, lo que resulta en un componente de vector dentro de la gama  $[-31,5, 31,5]$  con el mismo signo que el predictor (incluido cero).

El predictor para MVD y  $MVD_{2-4}$  se define como el valor mediano de los componentes de vector  $MV_1$ ,  $MV_2$  y  $MV_3$  definidos en 6.1.1 y en F.2. Para MVDB, el predictor  $P_c = (TR_B \times MV) / TR_D$ , donde  $MV$  representa un componente de vector para un bloque de luminancia  $8 * 8$  en una imagen P (véase también G.4).

Si PLUSPTYPE está presente, la gama de vectores de movimiento no depende del valor de predicción del vector de movimiento. Si el campo UUI se fija a "1", la gama de vectores de

movimiento depende del formato de la imagen. Para formatos de imagen normalizados hasta CIF, la gama es  $[-32, 31,5]$ , para formatos hasta el de 4 veces CIF la gama es  $[-64, 63,5]$ , para formatos hasta el de 16 veces CIF la gama es  $[-128, 127,5]$ , y para formatos de imagen personalizados incluso mayores la gama es  $[-256, 255,5]$ . Las gamas de vectores de movimiento horizontales y verticales pueden ser diferentes para formatos de imagen personalizados. En los cuadros D.1 y D.2 se especifican las gamas horizontales y verticales.

**Cuadro D.1/H.263 – Gama de vectores de movimiento horizontal cuando PLUSPTYPE está presente y UUI = 1**

Anchura de imagen	Gama de vectores de movimiento horizontal
4, ..., 352	$[-32, 31,5]$
356, ..., 704	$[-64, 63,5]$
708, ..., 1408	$[-128, 127,5]$
1412, ..., 2048	$[-256, 255,5]$

**Cuadro D.2/H.263 – Gama de vectores de movimiento vertical cuando PLUSPTYPE está presente y UUI = 1**

Altura de imagen	Gama de vectores de movimiento vertical
4, ..., 288	$[-32, 31,5]$
292, ..., 576	$[-64, 63,5]$
580, ..., 1152	$[-128, 127,5]$

En el modo actualización de resolución reducida, la gama especificada se aplica a pseudovectores de movimiento. Esto significa que la gama resultante de vectores de movimiento efectivos se amplía hasta aproximadamente un tamaño doble (véase también el anexo Q).

Si UUI se fija a "01", los vectores de movimiento sólo están limitados por su distancia al límite de la zona codificada como se explica en D.1.1. La misma limitación se aplica a los vectores de movimiento efectivos (no sólo los pseudovectores de movimiento) en el modo actualización de resolución reducida.

Para codificar los vectores de movimiento cuando está presente PLUSPTYPE, se utiliza el cuadro D.3 y se codifica la diferencia entre el vector de movimiento y la predicción del vector de movimiento. Cada entrada del cuadro D.3 tiene un solo valor (a diferencia del cuadro 14). La gama de vectores de movimiento y la utilización del cuadro D.3 para codificar datos de vectores de movimiento se aplica a todos los tipos de imagen cuando está presente PLUSPTYPE.

Las diferencias de vectores de movimiento se codifican siempre como un par de componentes, uno horizontal y otro vertical. Si un par es igual a (0,5, 0,5) se producen seis ceros consecutivos. Para evitar la emulación de código de comienzo, esta ocurrencia deberá ir seguida de un bit fijado a "1", lo que corresponde al envío de un componente de vector de movimiento cero adicional.

**Cuadro D.3/H.263 – Tabla de vectores de movimiento utilizada cuando PLUSPTYPE está presente**

Valor absoluto de la diferencia de vectores en unidades de mitad de píxel	Número de bits	Códigos
0	1	1
1	3	0s0
"x <sub>0</sub> " + 2 (2:3)	5	0x <sub>0</sub> 1s0
"x <sub>1</sub> x <sub>0</sub> " + 4 (4:7)	7	0x <sub>1</sub> 1x <sub>0</sub> 1s0
"x <sub>2</sub> x <sub>1</sub> x <sub>0</sub> " + 8 (8:15)	9	0x <sub>2</sub> 1x <sub>1</sub> 1x <sub>0</sub> 1s0
"x <sub>3</sub> x <sub>2</sub> x <sub>1</sub> x <sub>0</sub> " + 16 (16:31)	11	0x <sub>3</sub> 1x <sub>2</sub> 1x <sub>1</sub> 1x <sub>0</sub> 1s0
"x <sub>4</sub> x <sub>3</sub> x <sub>2</sub> x <sub>1</sub> x <sub>0</sub> " + 32 (32:63)	13	0x <sub>4</sub> 1x <sub>3</sub> 1x <sub>2</sub> 1x <sub>1</sub> 1x <sub>0</sub> 1s0
"x <sub>5</sub> x <sub>4</sub> x <sub>3</sub> x <sub>2</sub> x <sub>1</sub> x <sub>0</sub> " + 64 (64:127)	15	0x <sub>5</sub> 1x <sub>4</sub> 1x <sub>3</sub> 1x <sub>2</sub> 1x <sub>1</sub> 1x <sub>0</sub> 1s0
"x <sub>6</sub> x <sub>5</sub> x <sub>4</sub> x <sub>3</sub> x <sub>2</sub> x <sub>1</sub> x <sub>0</sub> " + 128 (128:255)	17	0x <sub>6</sub> 1x <sub>5</sub> 1x <sub>4</sub> 1x <sub>3</sub> 1x <sub>2</sub> 1x <sub>1</sub> 1x <sub>0</sub> 1s0
"x <sub>7</sub> x <sub>6</sub> x <sub>5</sub> x <sub>4</sub> x <sub>3</sub> x <sub>2</sub> x <sub>1</sub> x <sub>0</sub> " + 256 (256:511)	19	0x <sub>7</sub> 1x <sub>6</sub> 1x <sub>5</sub> 1x <sub>4</sub> 1x <sub>3</sub> 1x <sub>2</sub> 1x <sub>1</sub> 1x <sub>0</sub> 1s0
"x <sub>8</sub> x <sub>7</sub> x <sub>6</sub> x <sub>5</sub> x <sub>4</sub> x <sub>3</sub> x <sub>2</sub> x <sub>1</sub> x <sub>0</sub> " + 512 (512:1023)	21	0x <sub>8</sub> 1x <sub>7</sub> 1x <sub>6</sub> 1x <sub>5</sub> 1x <sub>4</sub> 1x <sub>3</sub> 1x <sub>2</sub> 1x <sub>1</sub> 1x <sub>0</sub> 1s0
"x <sub>9</sub> x <sub>8</sub> x <sub>7</sub> x <sub>6</sub> x <sub>5</sub> x <sub>4</sub> x <sub>3</sub> x <sub>2</sub> x <sub>1</sub> x <sub>0</sub> " + 1024 (1024:2047)	23	0x <sub>9</sub> 1x <sub>8</sub> 1x <sub>7</sub> 1x <sub>6</sub> 1x <sub>5</sub> 1x <sub>4</sub> 1x <sub>3</sub> 1x <sub>2</sub> 1x <sub>1</sub> 1x <sub>0</sub> 1s0
"x <sub>10</sub> x <sub>9</sub> x <sub>8</sub> x <sub>7</sub> x <sub>6</sub> x <sub>5</sub> x <sub>4</sub> x <sub>3</sub> x <sub>2</sub> x <sub>1</sub> x <sub>0</sub> " + 2048 (2048:4095)	25	0x <sub>10</sub> 1x <sub>9</sub> 1x <sub>8</sub> 1x <sub>7</sub> 1x <sub>6</sub> 1x <sub>5</sub> 1x <sub>4</sub> 1x <sub>3</sub> 1x <sub>2</sub> 1x <sub>1</sub> 1x <sub>0</sub> 1s0

El cuadro D.3 es un cuadro reversible construido periódicamente. Cada fila representa un intervalo de diferencia de vectores de movimiento en unidades de mitad de píxel. Los bits "...x<sub>1</sub>x<sub>0</sub>" representan todos los bits que siguen al "1" delantero en la representación binaria del valor absoluto de la diferencia de vectores de movimiento. El bit "s" representa el signo de la diferencia de vectores de movimiento, "0" para positivo y "1" para negativo. La representación binaria de la diferencia de vectores de movimiento está entrelazada con bits que indican si el código continúa o finaliza. Por ejemplo, una diferencia de vectores de movimiento de -13 tiene como signo s = 1 y su representación binaria es 1x<sub>2</sub>x<sub>1</sub>x<sub>0</sub> = 1101. Se codifica como 0 x<sub>2</sub>1 x<sub>1</sub>1 x<sub>0</sub>1 s0 = 0 11 01 11 10. El 0 de la segunda posición del último grupo de los bits indica el final del código.

## Anexo E

### Modo codificación aritmética basada en sintaxis

#### E.1 Introducción

En la codificación/decodificación de longitud variable (VLC/VLD, *variable length coding/decoding*), como se describe en la cláusula 5, un símbolo se codifica con VLC mediante un cuadro específico basado en la sintaxis del codificador. Por lo común, ese cuadro contiene las longitudes y los valores de las palabras de código VLC. El símbolo se hace corresponder con una entrada del cuadro en una operación de consulta de cuadro, y luego la palabra de código binario especificada por la entrada se envía normalmente a una memoria tampón para la transmisión al receptor. En la decodificación VLD, el tren binario recibido se empareja, entrada por entrada, con un cuadro específico basado en la sintaxis del codificador. Este cuadro tiene que ser el mismo que el usado en el codificador para codificar el símbolo objeto de decodificación. Después la entrada emparejada en el cuadro se hace corresponder inversamente con el símbolo correspondiente, lo que constituye el resultado final del decodificador VLD y se utiliza más adelante para recuperar las

imágenes de vídeo. Este proceso VLC/VLD implica que cada símbolo tiene que ser codificado con un número total fijo de bits. La supresión de esta restricción de número total fijo de bits para los símbolos permite reducir la velocidad binaria resultante, lo que se consigue mediante codificación aritmética.

En este anexo se describe el modo codificación aritmética basada en sintaxis (SAC) opcional de la presente Recomendación. En este modo, todas las operaciones correspondientes de la presente Recomendación de codificación/decodificación de longitud variable se sustituyen por operaciones de codificación/decodificación aritmética. La capacidad de este modo H.263 se señala con medios externos (por ejemplo, aplicando la Rec. UIT-T H.245). La utilización de este modo se indica en PTYPE.

## E.2 Especificación del codificador SAC

En el modo SAC, un símbolo se codifica utilizando un conjunto específico de números enteros (o un modelo) basado en la sintaxis del codificador y llamando al procedimiento que se ofrece a continuación, especificado en el lenguaje C.

```
#define q1 16384
#define q2 32768
#define q3 49152
#define top 65535

static long low, high, opposite_bits, length;
void encode_a_symbol(int index, int cumul_freq[ ])
{
    length = high - low + 1;
    high = low - 1 + (length * cumul_freq[index]) / cumul_freq[0];
    low += (length * cumul_freq[index+1]) / cumul_freq[0];
    for ( ; ; ) {
        if (high < q2) {
            send out a bit "0" to PSC_FIFO;
            while (opposite_bits > 0) {
                send out a bit "1" to PSC_FIFO;
                opposite_bits--;
            }
        }
        else if (low >= q2) {
            send out a bit "1" to PSC_FIFO;
            while (opposite_bits > 0) {
                send out a bit "0" to PSC_FIFO;
                opposite_bits--;
            }
            low -= q2;
            high -= q2;
        }
        else if (low >= q1 && high < q3) {
            opposite_bits += 1;
            low -= q1;
            high -= q1;
        }
        else break;

        low *= 2;
        high = 2 * high+1;
    }
}
```

Los valores low, high y opposite\_bits se inicializan con 0, top y 0, respectivamente. PSC\_FIFO es una memoria FIFO [primero en entrar, primero en salir (*first in first out*)] que

almacena los bits de salida del codificador aritmético. El modelo se especifica mediante `cumul_freq[ ]`, y el símbolo se especifica con el índice del modelo.

### E.3 Especificación del decodificador SAC

En el decodificador SAC, un símbolo se decodifica utilizando un modelo específico basado en la sintaxis y llamando al procedimiento que se ofrece a continuación, especificado en el lenguaje C.

```
static long          low, high, code_value, bit, length, index, cum;
int decode_a_symbol(int cumul_freq[ ])
{
    length = high - low + 1;
    cum = (-1 + (code_value - low + 1) * cumul_freq[0]) / length;
    for (index = 1; cumul_freq[index] > cum; index++);
    high = low - 1 + (length * cumul_freq[index-1]) / cumul_freq[0];
    low += (length * cumul_freq[index]) / cumul_freq[0];
    for ( ; ; ) {
        if (high < q2);
        else if (low >= q2) {
            code_value -= q2;
            low -= q2;
            high -= q2;
        }
        else if (low >= q1 && high < q3) {
            code_value -= q1;
            low -= q1;
            high -= q1;
        }
        else break;

        low *= 2;
        high = 2 * high + 1;
        get bit from PSC_FIFO;
        code_value = 2 * code_value + bit;
    }
    return (index-1);
}
```

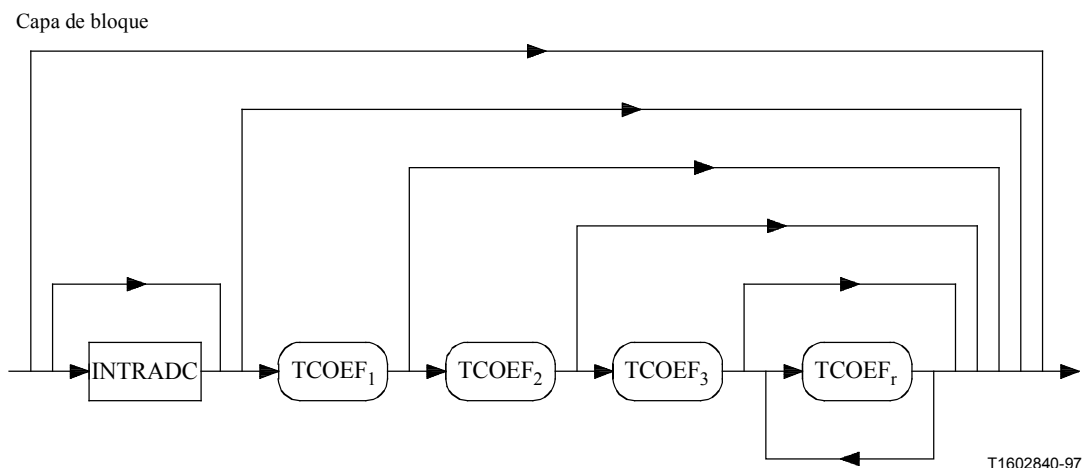
Una vez más, el modelo se especifica mediante `cumul_freq[ ]`. El símbolo decodificado se devuelve a través de su índice en el modelo. `PSC_FIFO` es una memoria FIFO que almacena el tren de datos entrante. El decodificador se inicializa para empezar la decodificación de un tren binario de codificación aritmética llamando al procedimiento que se ofrece a continuación.

```
void decoder_reset( )
{
    code_value = 0;
    low = 0;
    high = top;
    for (int i = 1; i <= 16; i++) {
        get bit from PSC_FIFO;
        code_value = 2 * code_value + bit;
    }
}
```

### E.4 Sintaxis

Como en el modo tabla VLC de esta Recomendación, la sintaxis de los símbolos se distribuye en cuatro capas: imagen, grupo de bloques, macrobloque y bloque. La sintaxis de las tres capas superiores se mantiene exactamente igual. La sintaxis de la capa de bloque también es bastante similar, pero se muestra en la figura E.1.





**Figura E.1/H.263 – Estructura de la capa de bloque SAC**

En la figura E.1, TCOEF1, TCOEF2, TCOEF3 y TCOEFr son símbolos de LAST-RUN-LEVEL, como se define en 5.4.2, y son, respectivamente, los posibles 1º, 2º, 3º y resto de los símbolos. TCOEF1, TCOEF2, TCOEF3 y TCOEFr sólo están presentes si uno, dos, tres o más coeficientes están presentes en la capa de bloque, respectivamente.

### E.5 PSC\_FIFO

PSC\_FIFO en el codificador o en el decodificador es una FIFO de tamaño superior a 17 bits. En la PSC\_FIFO del codificador, se localizan las emulaciones ilegales de PSC y GBSC y se evitan con un relleno de "1" después de cada aparición sucesiva de 14 "0" (que no forman parte de PSC o de GBSC). En la PSC\_FIFO del decodificador, se suprime el primer "1" después de cada serie de 14 "0"; cuando una serie de 14 "0" está seguida por un "0", esto indica que se ha detectado un PSC o GBSC legal. La localización exacta de PSC o GBSC viene determinada por el próximo "1" que sigue a la cadena de ceros.

### E.6 Símbolos de la capa de encabezamiento

Se considera que las capas de encabezamiento de la sintaxis son elementos sintácticos situados encima de las capas de bloque y macrobloque (véase la figura 6 y la especificación de la sintaxis en el texto). Los niveles de encabezamiento de la sintaxis de la versión 1 básica pueden formar tres cadenas posibles, (PSTUF)--PSC--TR--PTYPE--PQUANT--CPM--(PSBI)--(TRB-DBQUANT) --PEI--(PSUPP--PEI--...), (GSTUF)--GBSC--GN--(GSBI)--GFID--GQUANT y (ESTUF)--(EOS)--(PSTUF). En la sintaxis revisada de la versión 2, los niveles de encabezamiento de la sintaxis pueden tener otras estructuras (véase la figura 6 y la especificación de la sintaxis en el texto). Las cadenas de sintaxis de nivel de encabezamiento se envían directamente a PSC\_FIFO como en el modo tabla VLC normal de la presente Recomendación en el lado del codificador, y se envían directamente desde la PSC\_FIFO del decodificador tras la detección de PSC, GBSC, SSC, EOS o EOSBS legales.

Si un encabezamiento no es el primero de una sesión de vídeo, hay que reinicializar el codificador aritmético antes de enviar el encabezamiento llamando al procedimiento que se ofrece a continuación. Este procedimiento también se llamará al final de una sesión de vídeo si no se envía (ESTUF)--EOS [o (ESTUF)--EOSBS para el subtren de bits para el que se envió el último encabezamiento].

```
void encoder_flush( )
{
    opposite_bits++;
    if (low < q1) {
```

```

    send out a bit "0" to PSC_FIFO;
    while (opposite_bits > 0) {
        send out a bit "1" to PSC_FIFO;
        opposite_bits- -;
    }
}
else {
    send out a bit "1" to PSC_FIFO;
    while (opposite_bits > 0) {
        send out a bit "0" to PSC_FIFO;
        opposite_bits- -;
    }
}
low = 0;
high = top;
}

```

En el decodificador, después de cada cadena de símbolos de longitud fija, se llama al procedimiento `decoder_reset`.

## E.7 Símbolos de capa de macrobloque y bloque

En E.8 se incluyen modelos para los símbolos de capa de macrobloque y bloque. Los índices que se indican en las tablas VLC de la cláusula 5 se utilizan para indizar los números enteros de los modelos.

El modelo para COD en imágenes P se designa por `cumf_COD`. El índice de COD es "0" cuando está en 0, y "1" cuando está en 1. El modelo para MCBPC en imágenes P se designa por `cumf_MCBPC_no4MVQ` a menos que esté presente PLUSPTYPE en el encabezamiento de imagen y que se utilice el modo predicción avanzada (anexo F) o el modo filtro de desbloqueo (anexo J), en cuyo caso el modelo se designa por `cumf_MCBPC_4MVQ`. Los índices de MCBPC se definen en el cuadro 7 para imágenes I y en el cuadro 8 para imágenes P. El modelo de MCBPC de imágenes I se designa por `cumf_MCBPC_intra`.

El modelo de MODB es `cumf_MODB_G` si se utiliza el anexo G o `cumf_MODB_M` si se utiliza el anexo M. Los índices de MODB se definen en el cuadro 11 o en el cuadro M.1 respectivamente. El modelo de CBPB<sub>n</sub>, con  $n = 1, 2, \dots, 4$ , es `cumf_YCBPB`, y el modelo de CBPB<sub>n</sub>, con  $n = 5, 6$ , es `cumf_UVCBPB`, con índice 0 para CBPB<sub>n</sub> = 0 e índice 1 para CBPB<sub>n</sub> = 1.

El modelo de CBPY es `cumf_CBPY` para macrobloques INTER y `cumf_CBPY_intra` para macrobloques INTRA. El modelo de DQUANT es `cumf_DQUANT`. La indización para CBPY y DQUANT se define en los cuadros 12 y 13, respectivamente.

El modelo de MVD, MVD<sub>2-4</sub> y MVDB es `cumf_MVD` y el modelo de INTRADC es `cumf_INTRADC`. La indización se define en los cuadros 14 y 15, respectivamente.

Un TCOEF sin escape consiste en un símbolo para TCOEF1/2/3/r, seguido de un símbolo, SIGN, para el signo del TCOEF. Los modelos de TCOEF1, TCOEF2, TCOEF3 y TCOEFr para bloque INTER son `cumf_TCOEF1`, `cumf_TCOEF2`, `cumf_TCOEF3`, `cumf_TCOEFr`. Los modelos de bloque INTRA son `cumf_TCOEF1_intra`, `cumf_TCOEF2_intra`, `cumf_TCOEF3_intra`, `cumf_TCOEFr_intra`. Para todos los TCOEF, la indización se define en el cuadro 16. El modelo de SIGN es `cumf_SIGN`. La indización de SIGN es 0 para signo positivo y 1 para signo negativo.

Los modelos de LAST, RUN, LEVEL después de ESCAPE son `cumf_LAST` (`cumf_LAST_intra`), `cumf_RUN` (`cumf_RUN_intra`), `cumf_LEVEL` (`cumf_LEVEL_intra`) para bloque INTER (INTRA). La indización de LAST es 0 para LAST = 0 y 1 para LAST = 1, mientras que la indización de RUN y LEVEL se define en el cuadro 17.

El modelo de MODO\_INTRA es `cumf_INTRA_AC DC`. La indización se define en el cuadro I.1.

## E.8 Modelos SAC

```
int cumf_COD[3]={16383, 6849, 0};
int cumf_MCBPC_no4MVQ[22]={16383, 4105, 3088, 2367, 1988, 1621, 1612, 1609, 1608, 496,
353, 195, 77, 22, 17, 12, 5, 4, 3, 2, 1, 0};
int cumf_MCBPC_4MVQ[26]={16383, 6880, 6092, 5178, 4916, 3965, 3880, 3795, 3768, 1491,
1190, 889, 655, 442, 416, 390, 360, 337, 334, 331, 327, 326, 88, 57, 26, 0};
int cumf_MCBPC_intra[10]={16383, 7410, 6549, 5188, 442, 182, 181, 141, 1, 0};
int cumf_MODB_G[4]={16383, 6062, 2130, 0};
int cumf_MODB_M[7] = {16383, 6717, 4568, 2784, 1370, 655, 0};
int cumf_YCBPB[3]={16383, 6062, 0};
int cumf_UVCBPB[3]={16383, 491, 0};
int cumf_CBPY[17]={16383, 14481, 13869, 13196, 12568, 11931, 11185, 10814, 9796, 9150,
8781, 7933, 6860, 6116, 4873, 3538, 0};
int cumf_CBPY_intra[17]={16383, 13619, 13211, 12933, 12562, 12395, 11913, 11783, 11004,
10782, 10689, 9928, 9353, 8945, 8407, 7795, 0};
int cumf_DQUANT[5]={16383, 12287, 8192, 4095, 0};
int cumf_MVD[65]={16383, 16380, 16369, 16365, 16361, 16357, 16350, 16343, 16339, 16333,
16326, 16318, 16311, 16306, 16298, 16291, 16283, 16272, 16261, 16249, 16235, 16222, 16207,
16175, 16141, 16094, 16044, 15936, 15764, 15463, 14956, 13924, 11491, 4621, 2264, 1315, 854,
583, 420, 326, 273, 229, 196, 166, 148, 137, 123, 114, 101, 91, 82, 76, 66, 59, 53, 46, 36, 30, 26,
24, 18, 14, 10, 5, 0};
int cumf_INTRADC[255]={16383, 16380, 16379, 16378, 16377, 16376, 16370, 16361, 16360,
16359, 16358, 16357, 16356, 16355, 16343, 16238, 16237, 16236, 16230, 16221, 16220, 16205,
16190, 16169, 16151, 16130, 16109, 16094, 16070, 16037, 16007, 15962, 15938, 15899, 15854,
15815, 15788, 15743, 15689, 15656, 15617, 15560, 15473, 15404, 15296, 15178, 15106, 14992,
14868, 14738, 14593, 14438, 14283, 14169, 14064, 14004, 13914, 13824, 13752, 13671, 13590,
13515, 13458, 13380, 13305, 13230, 13143, 13025, 12935, 12878, 12794, 12743, 12656, 12596,
12521, 12443, 12359, 12278, 12200, 12131, 12047, 12002, 11948, 11891, 11828, 11744, 11663,
11588, 11495, 11402, 11288, 11204, 11126, 11039, 10961, 10883, 10787, 10679, 10583, 10481,
10360, 10227, 10113, 9961, 9828, 9717, 9584, 9485, 9324, 9112, 9019, 8908, 8766, 8584, 8426,
8211, 7920, 7663, 7406, 7152, 6904, 6677, 6453, 6265, 6101, 5904, 5716, 5489, 5307, 5056, 4850,
4569, 4284, 3966, 3712, 3518, 3342, 3206, 3048, 2909, 2773, 2668, 2596, 2512, 2370, 2295, 2232,
2166, 2103, 2022, 1956, 1887, 1830, 1803, 1770, 1728, 1674, 1635, 1599, 1557, 1500, 1482, 1434,
1389, 1356, 1317, 1284, 1245, 1200, 1179, 1140, 1110, 1092, 1062, 1044, 1035, 1014, 1008, 993,
981, 954, 936, 912, 894, 876, 864, 849, 828, 816, 801, 792, 777, 756, 732, 690, 660, 642, 615, 597,
576, 555, 522, 489, 459, 435, 411, 405, 396, 387, 375, 360, 354, 345, 344, 329, 314, 293, 278, 251,
236, 230, 224, 215, 214, 208, 199, 193, 184, 178, 169, 154, 127, 100, 94, 73, 37, 36, 35, 34, 33, 32,
31, 30, 29, 28, 27, 26, 20, 19, 18, 17, 16, 15, 9, 0};
int cumf_TCOEF1[104]={16383, 13455, 12458, 12079, 11885, 11800, 11738, 11700, 11681,
11661, 11651, 11645, 11641, 10572, 10403, 10361, 10346, 10339, 10335, 9554, 9445, 9427, 9419,
9006, 8968, 8964, 8643, 8627, 8624, 8369, 8354, 8352, 8200, 8192, 8191, 8039, 8036, 7920, 7917,
7800, 7793, 7730, 7727, 7674, 7613, 7564, 7513, 7484, 7466, 7439, 7411, 7389, 7373, 7369, 7359,
7348, 7321, 7302, 7294, 5013, 4819, 4789, 4096, 4073, 3373, 3064, 2674, 2357, 2177, 1975, 1798,
1618, 1517, 1421, 1303, 1194, 1087, 1027, 960, 890, 819, 758, 707, 680, 656, 613, 566, 534, 505,
475, 465, 449, 430, 395, 358, 335, 324, 303, 295, 286, 272, 233, 215, 0};
```

int cumf\_TCOEF2[104]={16383, 13582, 12709, 12402, 12262, 12188, 12150, 12131, 12125, 12117, 12113, 12108, 12104, 10567, 10180, 10070, 10019, 9998, 9987, 9158, 9037, 9010, 9005, 8404, 8323, 8312, 7813, 7743, 7726, 7394, 7366, 7364, 7076, 7062, 7060, 6810, 6797, 6614, 6602, 6459, 6454, 6304, 6303, 6200, 6121, 6059, 6012, 5973, 5928, 5893, 5871, 5847, 5823, 5809, 5796, 5781, 5771, 5763, 5752, 4754, 4654, 4631, 3934, 3873, 3477, 3095, 2758, 2502, 2257, 2054, 1869, 1715, 1599, 1431, 1305, 1174, 1059, 983, 901, 839, 777, 733, 683, 658, 606, 565, 526, 488, 456, 434, 408, 380, 361, 327, 310, 296, 267, 259, 249, 239, 230, 221, 214, 0};

int cumf\_TCOEF3[104]={16383, 13532, 12677, 12342, 12195, 12112, 12059, 12034, 12020, 12008, 12003, 12002, 12001, 10586, 10297, 10224, 10202, 10195, 10191, 9223, 9046, 8999, 8987, 8275, 8148, 8113, 7552, 7483, 7468, 7066, 7003, 6989, 6671, 6642, 6631, 6359, 6327, 6114, 6103, 5929, 5918, 5792, 5785, 5672, 5580, 5507, 5461, 5414, 5382, 5354, 5330, 5312, 5288, 5273, 5261, 5247, 5235, 5227, 5219, 4357, 4277, 4272, 3847, 3819, 3455, 3119, 2829, 2550, 2313, 2104, 1881, 1711, 1565, 1366, 1219, 1068, 932, 866, 799, 750, 701, 662, 605, 559, 513, 471, 432, 403, 365, 336, 312, 290, 276, 266, 254, 240, 228, 223, 216, 206, 199, 192, 189, 0};

int cumf\_TCOEFr[104]={16383, 13216, 12233, 11931, 11822, 11776, 11758, 11748, 11743, 11742, 11741, 11740, 11739, 10203, 9822, 9725, 9691, 9677, 9674, 8759, 8609, 8576, 8566, 7901, 7787, 7770, 7257, 7185, 7168, 6716, 6653, 6639, 6276, 6229, 6220, 5888, 5845, 5600, 5567, 5348, 5327, 5160, 5142, 5004, 4900, 4798, 4743, 4708, 4685, 4658, 4641, 4622, 4610, 4598, 4589, 4582, 4578, 4570, 4566, 3824, 3757, 3748, 3360, 3338, 3068, 2835, 2592, 2359, 2179, 1984, 1804, 1614, 1445, 1234, 1068, 870, 739, 668, 616, 566, 532, 489, 453, 426, 385, 357, 335, 316, 297, 283, 274, 266, 259, 251, 241, 233, 226, 222, 217, 214, 211, 209, 208, 0};

int cumf\_TCOEF1\_intra[104]={16383, 13383, 11498, 10201, 9207, 8528, 8099, 7768, 7546, 7368, 7167, 6994, 6869, 6005, 5474, 5220, 5084, 4964, 4862, 4672, 4591, 4570, 4543, 4397, 4337, 4326, 4272, 4240, 4239, 4212, 4196, 4185, 4158, 4157, 4156, 4140, 4139, 4138, 4137, 4136, 4125, 4124, 4123, 4112, 4111, 4110, 4109, 4108, 4107, 4106, 4105, 4104, 4103, 4102, 4101, 4100, 4099, 4098, 4097, 3043, 2897, 2843, 1974, 1790, 1677, 1552, 1416, 1379, 1331, 1288, 1251, 1250, 1249, 1248, 1247, 1236, 1225, 1224, 1223, 1212, 1201, 1200, 1199, 1198, 1197, 1196, 1195, 1194, 1193, 1192, 1191, 1190, 1189, 1188, 1187, 1186, 1185, 1184, 1183, 1182, 1181, 1180, 1179, 0};

int cumf\_TCOEF2\_intra[104]={16383, 13242, 11417, 10134, 9254, 8507, 8012, 7556, 7273, 7062, 6924, 6839, 6741, 6108, 5851, 5785, 5719, 5687, 5655, 5028, 4917, 4864, 4845, 4416, 4159, 4074, 3903, 3871, 3870, 3765, 3752, 3751, 3659, 3606, 3580, 3541, 3540, 3514, 3495, 3494, 3493, 3474, 3473, 3441, 3440, 3439, 3438, 3425, 3424, 3423, 3422, 3421, 3420, 3401, 3400, 3399, 3398, 3397, 3396, 2530, 2419, 2360, 2241, 2228, 2017, 1687, 1576, 1478, 1320, 1281, 1242, 1229, 1197, 1178, 1152, 1133, 1114, 1101, 1088, 1087, 1086, 1085, 1072, 1071, 1070, 1069, 1068, 1067, 1066, 1065, 1064, 1063, 1062, 1061, 1060, 1059, 1058, 1057, 1056, 1055, 1054, 1053, 1052, 0};

int cumf\_TCOEF3\_intra[104]={16383, 12741, 10950, 10071, 9493, 9008, 8685, 8516, 8385, 8239, 8209, 8179, 8141, 6628, 5980, 5634, 5503, 5396, 5327, 4857, 4642, 4550, 4481, 4235, 4166, 4151, 3967, 3922, 3907, 3676, 3500, 3324, 3247, 3246, 3245, 3183, 3168, 3084, 3069, 3031, 3030, 3029, 3014, 3013, 2990, 2975, 2974, 2973, 2958, 2943, 2928, 2927, 2926, 2925, 2924, 2923, 2922, 2921, 2920, 2397, 2298, 2283, 1891, 1799, 1591, 1445, 1338, 1145, 1068, 1006, 791, 768, 661, 631, 630, 615, 592, 577, 576, 561, 546, 523, 508, 493, 492, 491, 476, 475, 474, 473, 472, 471, 470, 469, 468, 453, 452, 451, 450, 449, 448, 447, 446, 0};

int cumf\_TCOEFr\_intra[104]={16383, 12514, 10776, 9969, 9579, 9306, 9168, 9082, 9032, 9000, 8981, 8962, 8952, 7630, 7212, 7053, 6992, 6961, 6940, 6195, 5988, 5948, 5923, 5370, 5244, 5210, 4854, 4762, 4740, 4384, 4300, 4288, 4020, 3968, 3964, 3752, 3668, 3511, 3483, 3354, 3322, 3205, 3183, 3108, 3046, 2999, 2981, 2974, 2968, 2961, 2955, 2949, 2943, 2942, 2939, 2935, 2934, 2933, 2929, 2270, 2178, 2162, 1959, 1946, 1780, 1651, 1524, 1400, 1289, 1133, 1037, 942, 849, 763, 711, 591, 521, 503, 496, 474, 461, 449, 442, 436, 426, 417, 407, 394, 387, 377, 373, 370, 367, 366, 365, 364, 363, 362, 358, 355, 352, 351, 350, 0};

```

int cumf_SIGN[3]={16383, 8416, 0};
int cumf_LAST[3]={16383, 9469, 0};
int cumf_LAST_intra[3]={16383, 2820, 0};
int cumf_RUN[65]={16383, 15310, 14702, 13022, 11883, 11234, 10612, 10192, 9516, 9016, 8623,
8366, 7595, 7068, 6730, 6487, 6379, 6285, 6177, 6150, 6083, 5989, 5949, 5922, 5895, 5828, 5774,
5773, 5394, 5164, 5016, 4569, 4366, 4136, 4015, 3867, 3773, 3692, 3611, 3476, 3341, 3301, 2787,
2503, 2219, 1989, 1515, 1095, 934, 799, 691, 583, 435, 300, 246, 206, 125, 124, 97, 57, 30, 3, 2, 1,
0};
int cumf_RUN_intra[65]={16383, 10884, 8242, 7124, 5173, 4745, 4246, 3984, 3034, 2749, 2607,
2298, 966, 681, 396, 349, 302, 255, 254, 253, 206, 159, 158, 157, 156, 155, 154, 153, 106, 35, 34,
33, 32, 31, 30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6,
5, 4, 3, 2, 1, 0};
int cumf_LEVEL[255]={16383, 16382, 16381, 16380, 16379, 16378, 16377, 16376, 16375, 16374,
16373, 16372, 16371, 16370, 16369, 16368, 16367, 16366, 16365, 16364, 16363, 16362, 16361,
16360, 16359, 16358, 16357, 16356, 16355, 16354, 16353, 16352, 16351, 16350, 16349, 16348,
16347, 16346, 16345, 16344, 16343, 16342, 16341, 16340, 16339, 16338, 16337, 16336, 16335,
16334, 16333, 16332, 16331, 16330, 16329, 16328, 16327, 16326, 16325, 16324, 16323, 16322,
16321, 16320, 16319, 16318, 16317, 16316, 16315, 16314, 16313, 16312, 16311, 16310, 16309,
16308, 16307, 16306, 16305, 16304, 16303, 16302, 16301, 16300, 16299, 16298, 16297, 16296,
16295, 16294, 16293, 16292, 16291, 16290, 16289, 16288, 16287, 16286, 16285, 16284, 16283,
16282, 16281, 16280, 16279, 16278, 16277, 16250, 16223, 16222, 16195, 16154, 16153, 16071,
15989, 15880, 15879, 15878, 15824, 15756, 15674, 15606, 15538, 15184, 14572, 13960, 10718,
7994, 5379, 2123, 1537, 992, 693, 611, 516, 448, 421, 380, 353, 352, 284, 257, 230, 203, 162, 161,
160, 133, 132, 105, 104, 103, 102, 101, 100, 99, 98, 97, 96, 95, 94, 93, 92, 91, 90, 89, 88, 87, 86,
85, 84, 83, 82, 81, 80, 79, 78, 77, 76, 75, 74, 73, 72, 71, 70, 69, 68, 67, 66, 65, 64, 63, 62, 61, 60,
59, 58, 57, 56, 55, 54, 53, 52, 51, 50, 49, 48, 47, 46, 45, 44, 43, 42, 41, 40, 39, 38, 37, 36, 35, 34,
33, 32, 31, 30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6,
5, 4, 3, 2, 1, 0};
int cumf_LEVEL_intra[255]={16383, 16379, 16378, 16377, 16376, 16375, 16374, 16373, 16372,
16371, 16370, 16369, 16368, 16367, 16366, 16365, 16364, 16363, 16362, 16361, 16360, 16359,
16358, 16357, 16356, 16355, 16354, 16353, 16352, 16351, 16350, 16349, 16348, 16347, 16346,
16345, 16344, 16343, 16342, 16341, 16340, 16339, 16338, 16337, 16336, 16335, 16334, 16333,
16332, 16331, 16330, 16329, 16328, 16327, 16326, 16325, 16324, 16323, 16322, 16321, 16320,
16319, 16318, 16317, 16316, 16315, 16314, 16313, 16312, 16311, 16268, 16267, 16224, 16223,
16180, 16179, 16136, 16135, 16134, 16133, 16132, 16131, 16130, 16129, 16128, 16127, 16126,
16061, 16018, 16017, 16016, 16015, 16014, 15971, 15970, 15969, 15968, 15925, 15837, 15794,
15751, 15750, 15749, 15661, 15618, 15508, 15376, 15288, 15045, 14913, 14781, 14384, 13965,
13502, 13083, 12509, 12289, 12135, 11892, 11738, 11429, 11010, 10812, 10371, 9664, 9113,
8117, 8116, 8028, 6855, 5883, 4710, 4401, 4203, 3740, 3453, 3343, 3189, 2946, 2881, 2661, 2352,
2132, 1867, 1558, 1382, 1250, 1162, 1097, 1032, 967, 835, 681, 549, 439, 351, 350, 307, 306, 305,
304, 303, 302, 301, 300, 299, 298, 255, 212, 211, 210, 167, 166, 165, 164, 163, 162, 161, 160, 159,
158, 115, 114, 113, 112, 111, 68, 67, 66, 65, 64, 63, 62, 61, 60, 59, 58, 57, 56, 55, 54, 53, 52, 51,
50, 49, 48, 47, 46, 45, 44, 43, 42, 41, 40, 39, 38, 37, 36, 35, 34, 33, 32, 31, 30, 29, 28, 27, 26, 25,
24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0};
int cumf_INTRA_AC_DC[4]={16383, 9229, 5461, 0};

```

## Anexo F

### Modo predicción avanzada

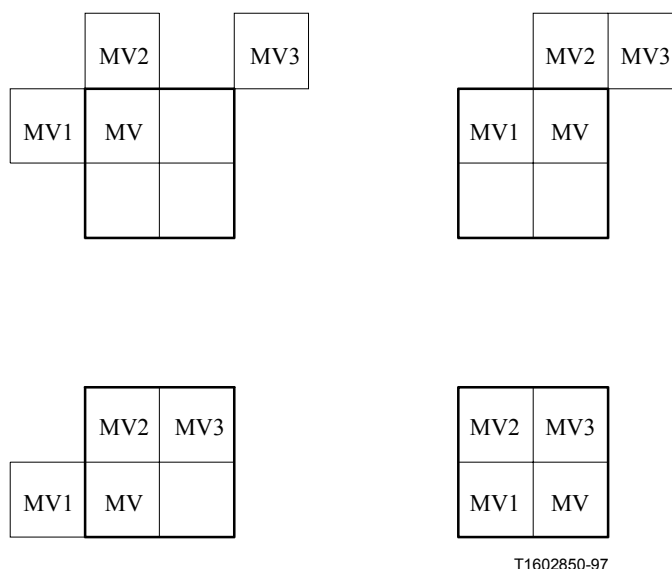
#### F.1 Introducción

Este anexo describe el modo con predicción avanzada facultativo de la presente Recomendación, que incluye la compensación de movimiento de bloques superpuestos y la posibilidad de utilizar cuatro vectores de movimiento por macrobloque. La capacidad de este modo se señala por medios externos (por ejemplo, aplicando la Rec. UIT-T H.245). El uso de este modo se indica en PTYPE. En el modo con predicción avanzada, los vectores de movimiento pueden atravesar fronteras de imagen como en el caso del modo de vector de movimiento sin restricción (para la descripción de esta técnica, véase D.1). La característica de la gama ampliada de vectores de movimiento del modo vector de movimiento sin restricción no se incluye automáticamente en el modo con predicción avanzada, y sólo está activa si se selecciona el modo de vector de movimiento sin restricción. Si se utiliza el modo con predicción avanzada en combinación con el modo tramas PB, la compensación del movimiento con superposición sólo se utiliza para la predicción de las imágenes P, no para las imágenes B.

#### F.2 Cuatro vectores de movimiento por macrobloque

En esta Recomendación, se utiliza un vector de movimiento por macrobloque, salvo en el modo predicción avanzada o en el modo filtro de desbloqueo. En este modo, la elección entre uno o cuatro vectores viene indicada por la palabra de código MCBPC de cada macrobloque. Si sólo se transmite un vector de movimiento para un macrobloque determinado éste se define como cuatro vectores con el mismo valor. Cuando MCBPC indica que se transmiten cuatro vectores de movimiento para el macrobloque en cuestión, la información del primer vector de movimiento se transmite con la palabra de código MVD, y la información de los tres vectores de movimiento adicionales se transmite con las palabras de código  $MVD_{2-4}$  (véanse también 5.3.7 y 5.3.8).

Los vectores se obtienen sumando predictores a las diferencias de vectores indicadas por MVD y  $MVD_{2-4}$  de manera similar a cuando sólo está presente un vector de movimiento por macrobloque, según las reglas de decisión dadas en 6.1.1. Los predictores se calculan por separado para los componentes horizontal y vertical. Pero los predictores candidatos MV1, MV2 y MV3 se vuelven a definir, como se indica en la figura F.1. Si sólo está presente un vector por macrobloque, MV1, MV2 y MV3 se definen como para el bloque 8 \* 8 numerado 1 en la figura 5 (esta definición se muestra en la parte superior izquierda de las cuatro subfiguras de la figura F.1).



**Figura F.1/H.263 – Redefinición de los predictores candidatos MV1, MV2 y MV3 para cada uno de los bloques de luminancia de un macrobloque**

Cuando se utilizan cuatro vectores, cada uno de los vectores de movimiento se utiliza para todos los píxeles de uno de los cuatro bloques de luminancia del macrobloque. La numeración de los vectores de movimiento es equivalente a la numeración de los cuatro bloques de luminancia, como se indica en la figura 5. El vector de movimiento  $MVD_{CHR}$  para los dos bloques de crominancia se obtiene calculando la suma de los cuatro vectores de luminancia y dividiéndola por 8; los valores de componentes de los vectores resultantes con resolución de dieciseisavo de píxel se modifican hacia la posición de mitad de píxel más cercana, como se indica en el cuadro F.1.

**Cuadro F.1/H.263 – Modificación de los componentes de vectores de crominancia con resolución de dieciseisavo de píxel**

<b>Posición de dieciseisavo de píxel</b>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	/16
<b>Posición resultante</b>	0	0	0	1	1	1	1	1	1	1	1	1	1	1	2	2	/2

Los valores de mitad de píxel se calculan por interpolación bilineal como se describe en 6.1.2. En el modo con predicción avanzada, la predicción para la luminancia se obtiene por compensación de movimiento de bloques superpuestos como se describe en F.3. La predicción para la crominancia se obtiene aplicando el vector de movimiento  $MVD_{CHR}$  a todos los píxeles en los dos bloques de crominancia (como se hace en el modo con predicción por defecto).

**F.3 Compensación de movimiento para bloques de luminancia superpuestos**

Cada píxel de un bloque de predicción de luminancia de  $8 * 8$  es una suma ponderada de tres valores de predicción, dividida por 8 (con redondeo). Para obtener los tres valores de predicción se utilizan tres vectores de movimiento: el vector de movimiento del bloque de luminancia elegido y dos de los cuatro vectores "distantes":

- el vector de movimiento del bloque situado a la izquierda o a la derecha del bloque de luminancia elegido;
- el vector de movimiento del bloque situado por encima o por debajo del bloque de luminancia elegido.

Para cada píxel, se utilizan los vectores de movimiento distantes de los bloques situados en el borde de los dos bloques más cercanos. Esto significa que, para la mitad superior del bloque, se usa el vector de movimiento correspondiente al bloque situado por encima del bloque elegido, mientras que para la mitad inferior del bloque, se utiliza el vector de movimiento correspondiente al bloque situado por debajo del bloque elegido (véase la figura F.3). Análogamente, para la mitad izquierda del bloque, se utiliza el vector de movimiento correspondiente al bloque situado a la izquierda del bloque elegido, mientras que para la mitad derecha del bloque, se utiliza el vector de movimiento correspondiente al bloque situado a la derecha del bloque elegido (véase la figura F.4).

Sea  $(x,y)$  una posición en una imagen medida en unidades de píxels enteros.

Sea  $(m,n)$  un índice de bloque entero en una imagen, dado por:

$$m = x / 8 \quad \text{y} \quad n = y / 8$$

donde "/" representa división con truncamiento.

Sea  $(i,j)$  una ubicación de píxel en un bloque de  $8 \times 8$ , dada por:

$$i = x - m \cdot 8 \quad \text{y} \quad j = y - n \cdot 8$$

Todo lo anterior da como resultado:

$$(x,y) = (m \cdot 8 + i, n \cdot 8 + j)$$

Sea  $(MV^k_x, MV^k_y)$  un vector de movimiento, que puede contener desplazamientos de píxels entero o píxels mitad con  $k = 0, 1, \text{ ó } 2$ . Por ejemplo  $(MV^k_x, MV^k_y)$  puede ser igual a  $(-7,0, 13,5)$ .

En este caso  $(MV^0_x, MV^0_y)$  denota el vector de movimiento para el bloque vigente  $(m,n)$ ,  $(MV^1_x, MV^1_y)$  denota el vector de movimiento del bloque por encima o por debajo y  $(MV^2_x, MV^2_y)$  denota el vector de movimiento a la izquierda o a la derecha del bloque vigente  $(m,n)$  definido anteriormente.

La creación entonces de cada píxel,  $P(x,y)$ , en un bloque de predicción de luminancia  $8 \times 8$  con un índice de bloque  $(m,n)$  está regida por la siguiente ecuación:

$$P(x,y) = (q(x,y) \cdot H_0(i, j) + r(x,y) \cdot H_1(i, j) + s(x,y) \cdot H_2(i, j) + 4)/8,$$

donde  $q(x,y)$ ,  $r(x,y)$  y  $s(x,y)$  son los valores de predicción tomados de la imagen referenciada definida por:

$$q(x,y) = p(x + MV^0_x, y + MV^0_y),$$

$$r(x,y) = p(x + MV^1_x, y + MV^1_y),$$

$$s(x,y) = p(x + MV^2_x, y + MV^2_y),$$

donde  $p(x + MV^k_x, y + MV^k_y)$  es el valor de predicción en la posición  $(x + MV^k_x, y + MV^k_y)$  en la imagen referenciada. Se señala que  $(x + MV^k_x, y + MV^k_y)$  puede estar fuera de la imagen y puede ser una posición de píxels enteros o píxels mitad. En los casos en que se utilizan vectores de movimiento de píxels mitad,  $p(x + MV^k_x, y + MV^k_y)$  se refiere el valor obtenido tras la aplicación del proceso de interpolación descrito en 6.1.2.

Las matrices  $H_0(i, j)$ ,  $H_1(i, j)$ , y  $H_2(i, j)$ , se definen en las figuras F.2, F.3 y F.4, donde  $(i, j)$  indican respectivamente la columna y la fila de la matriz.



Cuando no se utiliza el modo estructura en rebanada (véase el anexo K) ni el modo decodificación de segmento independiente (véase el anexo R), se utilizan vectores de movimiento distantes de otros segmentos de imagen de vídeo de la misma manera que vectores de movimiento distantes dentro del GOB vigente. Si se utiliza el modo estructura en rebanada o el modo decodificación de segmento independiente, los vectores de movimiento distantes correspondientes a bloques de otros segmentos de imagen de vídeo se fijan al vector de movimiento del bloque vigente, con independencia de las demás condiciones descritas en el próximo párrafo. (Véase la definición de segmento de imagen de vídeo en el anexo R.)

Si no se codificó uno de los macrobloques adyacentes, el vector de movimiento distante correspondiente se pone a 0. Si uno de los bloques adyacentes se codificó INTRA, el vector de movimiento distante correspondiente se sustituye por el vector de movimiento para el bloque vigente salvo cuando se utiliza el modo tramas PB. En este caso (bloque INTRA en modo tramas PB), se utiliza el vector de movimiento del bloque INTRA (véase también el anexo G). Cuando el bloque vigente está en el borde de la imagen y, por lo tanto, falta un bloque colindante, el vector de movimiento distante correspondiente se sustituye por el vector de movimiento elegido. En todos los casos, si el bloque se encuentra en la parte más baja del macrobloque (para los bloques números 3 ó 4, véase la figura 5), el vector de movimiento distante que corresponde al bloque de luminancia 8 \* 8 del macrobloque situado por debajo del macrobloque elegido se sustituye por el vector de movimiento del bloque elegido.

En las figuras F.2, F.3 y F.4 se ofrecen los valores de ponderación de la predicción.

4	5	5	5	5	5	5	4
5	5	5	5	5	5	5	5
5	5	6	6	6	6	5	5
5	5	6	6	6	6	5	5
5	5	6	6	6	6	5	5
5	5	6	6	6	6	5	5
5	5	5	5	5	5	5	5
4	5	5	5	5	5	5	4

**Figura F.2/H.263 – Valores de ponderación,  $H_0$ , para la predicción con el vector de movimiento del bloque de luminancia elegido**

2	2	2	2	2	2	2	2
1	1	2	2	2	2	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	2	2	2	2	1	1
2	2	2	2	2	2	2	2

**Figura F.3/H.263 – Valores de ponderación,  $H_1$ , para la predicción con los vectores de movimiento de los bloques de luminancia situado por encima y por debajo del bloque de luminancia elegido**

2	1	1	1	1	1	1	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	1	1	1	1	1	1	2

**Figura F.4/H.263 – Valores de ponderación,  $H_2$ , para la predicción con los vectores de movimiento de los bloques de luminancia situados a la izquierda y a la derecha del bloque de luminancia elegido**

## Anexo G

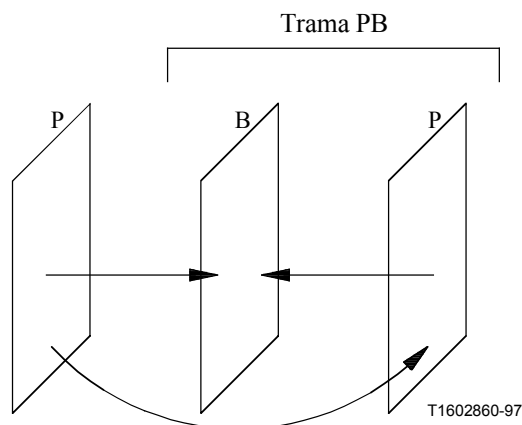
### Modo tramas PB

#### G.1 Introducción

Este anexo describe el modo tramas PB facultativo de la presente Recomendación. La capacidad de este modo se señala por medios externos (por ejemplo, aplicando la Rec. UIT-T H.245). El uso de este modo se indica en PTYPE.

Una trama PB consiste en dos imágenes que se codifican como una unidad. El nombre PB proviene del nombre de los tipos de imagen de la Rec. UIT-T H.262, en la que hay imágenes P e imágenes B. Por consiguiente, una trama PB consiste en una imagen P, que se predice a partir de la anterior imagen P decodificada, y una imagen B, que se predice a partir de la última imagen P decodificada y la imagen P que está siendo decodificada en ese momento. Se eligió la denominación imagen B porque partes de imágenes B se pueden predecir bidireccionalmente a partir de imágenes P pasadas y futuras. El proceso de predicción se ilustra en la figura G.1.

En el anexo M se describe una versión mejorada del modo tramas PB, al que se denomina "modo tramas PB mejoradas". El modo tramas PB descrito en este anexo se mantiene sólo a efectos de compatibilidad con los sistemas concebidos antes de la adopción del modo tramas PB mejoradas. Por este motivo, el modo tramas PB aquí descrito no se puede utilizar con las características adicionales de la sintaxis que requieren la utilización de PLUSPTYPE.



**Figura G.1/H.263 – Predicción con el modo tramas PB**

## G.2 Tramas PB y bloques INTRA

Cuando se utilizan tramas PB, el modo de codificación INTRA tiene el significado siguiente (véase también 5.3.2):

- los bloques P tienen codificación INTRA;
- los bloques B tienen codificación INTER con la predicción correspondiente de bloque INTER.

Si se utilizan tramas PB, los datos de vectores de movimiento (MVD) se incluyen también para macrobloques INTRA en imágenes para las cuales PTYPE indica "INTER". En este caso, el vector se utiliza sólo para bloques B. Las palabras de código  $MVD_{2,4}$  no se utilizan nunca para INTRA (véase también el cuadro 10). En el modo predicción avanzada y en el modo tramas PB y cuando uno de los bloques adyacentes se codificó en el modo INTRA, el correspondiente vector de movimiento distante no es sustituido por el vector de movimiento para el bloque urgente, sino que se usa el vector de movimiento "INTRA".

## G.3 Capa de bloque

En el modo tramas PB, un macrobloque consiste en doce bloques. Primero, se transmiten los datos de los seis bloques P, como en el modo H.263 por defecto; después, los datos de los seis bloques B (véase también 5.4). La figura 11 muestra la estructura de la capa de bloque. INTRADC está presente en todos los bloques P del macrobloque si MCBPC indica tipo de macrobloque 3 ó 4 (véanse los cuadros 7 y 8). INTRADC no está presente en los bloques B. TCOEF está presente en los bloques P si así lo indican MCBPC o CBPY; TCOEF está presente en los bloques B si así lo indica CBPB.

## G.4 Cálculo de los vectores de imagen B y con trama PB

Los vectores de imagen B se calculan como sigue (véase también 6.1.1). Supóngase un componente vectorial MV (vector de movimiento) de unidades de mitad de píxel para utilizar en la imagen P (MV representa un componente vectorial para un bloque de luminancia de  $8 * 8$ ; si sólo se transmite un vector por macrobloque, MV tiene el mismo valor para cada uno de los cuatro bloques de luminancia de  $8 * 8$ ). Para la predicción de una imagen B se necesitan los componentes vectoriales hacia adelante y hacia atrás,  $MV_F$  y  $MV_B$ . Estos componentes de vectores hacia adelante y hacia atrás se derivan de MV y a la larga son mejorados por un vector delta dado por MVDB.

- $TR_D$ : Incremento de la referencia temporal TR (o la combinación de la referencia temporal ampliada ETR y la referencia temporal TR en una trama PB mejorada cuando se utiliza una frecuencia de reloj de imagen personalizada) del encabezamiento de la última imagen (véase 5.1.2). Si  $TR_D$  es negativo,  $TR_D = TR_D + d$ , donde  $d = 256$  para frecuencia de imagen CIF y 1024 para cualquier frecuencia de reloj de imagen personalizada.
- $TR_B$ : Véase 5.1.2.

Supóngase que  $MV_D$  es el componente vectorial delta dado por MVDB y que corresponde con el componente vectorial MV. Si MVDB no está presente,  $MV_D$  se pone a cero. Si MVDB está presente, se utiliza el mismo  $MV_D$  dado por  $MV_{DB}$  para cada uno de los cuatro bloques de luminancia B dentro del macrobloque.

Ahora  $MV_F$  y  $MV_B$  vienen dados en unidades de mitad de píxel por las siguientes fórmulas:

$$\begin{aligned}MV_F &= (TR_B \times MV) / TR_D + MV_D \\MV_B &= ((TR_B - TR_D) \times MV) / TR_D \quad \text{si } MV_D \text{ es igual a } 0 \\MV_B &= MV_F - MV \quad \text{si } MV_D \text{ es distinto de } 0\end{aligned}$$

donde "/" significa división por truncamiento. Se supone que el ajuste por escalón refleja la posición real en el tiempo de las imágenes P y B. Se aprovecha el hecho de que la gama de valores de  $MV_F$

está limitada. Cada palabra VLC para MVDB representa un par de valores diferencia de los que sólo uno tendrá un valor para  $MV_F$  que cae dentro de la gama permitida  $[-16; 15,5]$  en el modo vector de movimiento sin restricción  $[-31,5; 31,5]$ . También se utilizan las fórmulas de  $MV_F$  y  $MV_B$  en el caso de bloques INTRA, en los que se utilizan los datos de vector sólo para la predicción de bloques B.

Para los bloques de crominancia,  $MV_F$  se obtiene calculando la suma de los cuatro vectores de luminancia  $MV_F$  correspondientes y dividiéndola por 8; los componentes de vectores resultantes con resolución de dieciseisavo de píxel se modifican hacia la posición de mitad de píxel más cercana, como se indica en el cuadro F.1.  $MV_B$  para crominancia, se obtiene calculando la suma de los cuatro vectores correspondientes de luminancia  $MV_B$  y dividiendo esta suma por 8; los componentes de vectores resultantes con resolución de un cuarto de píxel se modifican hacia la posición más cercana de mitad de píxel, como se indica en el cuadro F.1.

Un valor positivo del componente horizontal o vertical del vector de movimiento significa que la predicción se forma a partir de píxels de la imagen P referenciada, situados a la derecha o debajo de los píxels objeto de predicción.

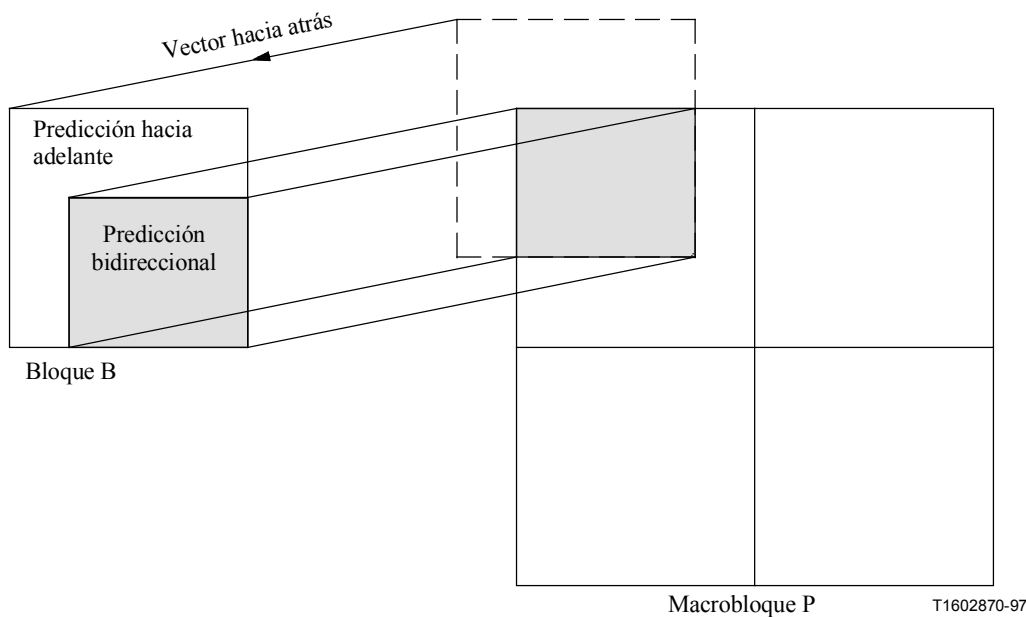
### **G.5 Predicción de un bloque B en una trama PB**

En esta subcláusula, un bloque significa un bloque de  $8 \times 8$ . El procedimiento que se ofrece a continuación es aplicable a bloques de luminancia y de crominancia. Primero, se calculan los vectores hacia adelante y hacia atrás. Se supone que el macrobloque P (luminancia y crominancia) se decodifica, se reconstruye y se recorta (véase 6.3.2). Este macrobloque se denomina  $P_{REC}$ . Sobre la base de  $P_{REC}$  y la predicción de  $P_{REC}$ , se calcula la predicción del bloque B.

La predicción del bloque B tiene dos modos, que se usan en partes diferentes del bloque:

- Para los píxels en los que el vector hacia atrás –  $MV_B$  – apunta hacia adentro de  $P_{REC}$ , se utiliza la predicción bidireccional. Esto se obtiene como el promedio entre la predicción hacia adelante con  $MV_F$  relativo a la imagen P decodificada previamente y la predicción hacia atrás con  $MV_B$  relativo a  $P_{REC}$ . El promedio se calcula dividiendo la suma de las dos predicciones por dos (división por truncamiento).
- Para todos los demás píxels, se emplea la predicción hacia adelante con  $MV_F$  relativo a la imagen P decodificada previamente.

La figura G.2 indica cuáles son las partes de un bloque que se predicen bidireccionalmente (partes sombreadas del bloque B) o sólo mediante predicción hacia adelante (resto del bloque B).



**Figura G.2/H.263 – Predicción hacia adelante y bidireccional para un bloque B**

La predicción bidireccional se utiliza para los píxels en los que el vector hacia atrás –  $MV_B$  – apunta hacia adentro de  $P_{REC}$ . Estos píxels se definen mediante los procedimientos siguientes, especificados en C.

Definiciones:

- nh Posición horizontal del bloque en un macrobloque (0 ó 1).
- nv Posición vertical del bloque en un macrobloque (0 ó 1).
- mh(nh,nv) Componente vectorial horizontal del bloque (nh,nv) en unidades de mitad de píxel.
- mv(nh,nv) Componente vectorial vertical del bloque (nh,nv) en unidades de mitad de píxel.
- mhc Componente horizontal del vector de crominancia.
- mvc Componente vertical del vector de crominancia.

### Procedimiento para la luminancia

```

for (nh = 0; nh <= 1; nh++) {
  for (nv = 0; nv <= 1; nv++) {
    for (i = nh * 8 + max(0,(-mh(nh,nv)+1)/2 - nh * 8);
         i <= nh * 8 + min(7,15-(mh(nh,nv)+1)/2 - nh * 8); i++) {
      for (j = nv * 8 + max(0,(-mv(nh,nv)+1)/2 - nv * 8);
           j <= nv * 8 + min(7,15-(mv(nh,nv)+1)/2 - nv * 8); j++) {
        predict pixel (i,j) bidirectionally
      }
    }
  }
}

```

## Procedimiento para la crominancia

```
for (i = max(0, (-mhc+1)/2); i <= min(7, 7-(mhc+1)/2); i++) {  
    for (j = max(0, (-mvc+1)/2); j <= min(7, 7-(mvc+1)/2); j++) {  
        predict pixel (i,j) bidirectionally;  
    }  
}
```

Los píxeles que no se predicen bidireccionalmente se predicen sólo con predicción hacia adelante.

## Anexo H

### Corrección de errores hacia adelante para señales de vídeo codificadas

#### H.1 Introducción

En este anexo se describe un método de corrección de errores hacia adelante facultativo (código y alineación de trama) para la transmisión de datos de vídeo codificados de la presente Recomendación. Esta corrección de errores hacia adelante se puede emplear en situaciones en las que no se proporciona corrección de errores hacia adelante por medios externos, por ejemplo a nivel de multiplex o de sistema. No se utiliza para la Rec. UIT-T H.324. La alineación de trama y el código de corrección de errores hacia adelante son iguales a los de la Rec. UIT-T H.261.

#### H.2 Alineación de trama para la corrección de errores

Para que el decodificador pueda identificar la información de paridad de los datos de vídeo y de corrección de errores, se incluye un esquema de alineación de trama para la corrección de errores. Este esquema consiste en multitramas de 8 tramas, cada una de las cuales consta de 1 bit de alineación de trama, 1 bit indicador de relleno ( $F_i$ ), 492 bits de datos codificados (o de relleno, todos 1) y 18 bits de paridad (véase la figura H.1). Para cada multitrama, el esquema de alineación de trama constituido por los bits de alineación de trama de las 8 tramas individuales es:

$$(S_1S_2S_3S_4S_5S_6S_7S_8) = (00011011)$$

El indicador de relleno ( $F_i$ ) puede ser puesto a 0 por un codificador. En este caso, se utilizan 492 bits de relleno consecutivos (todos 1) en lugar de 492 bits de datos codificados. Esto se puede usar para datos de relleno (véase 3.6).

#### H.3 Código de corrección de errores

El código de corrección de errores es un código de corrección de errores hacia adelante BCH (511, 493); su uso por el decodificador es opcional. La paridad se calcula con un código de 493 bits, que consta de un indicador de relleno ( $F_i$ ) de 1 bit y 492 bits de datos de vídeo codificados.

El polinomio generador es:

$$g(x) = (x^9 + x^4 + 1)(x^9 + x^6 + x^4 + x^3 + 1)$$

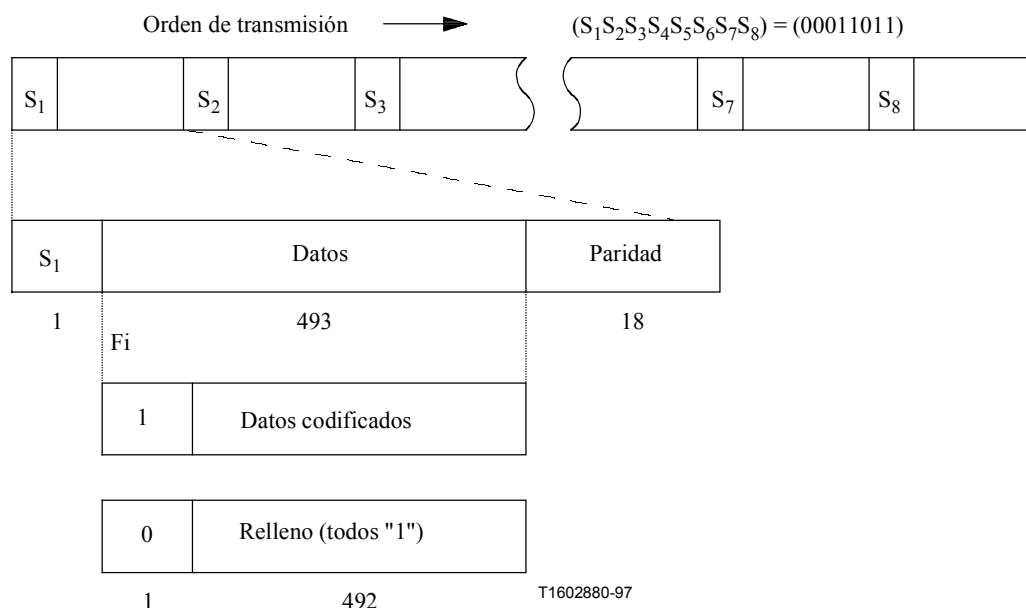
Ejemplo: Para los datos de entrada "01111 ... 11" (493 bits), los bits resultantes de paridad de corrección son "011011010100011011" (18 bits).

#### H.4 Plazo de reenganche para la alineación de trama del corrector de errores

No se puede considerar alcanzado el reenganche de trama antes de recibir tres esquemas consecutivos de alineación de trama para la corrección de errores. El decodificador debe estar

diseñado de manera tal que restablezca el enganche de trama dentro de los 34 000 bits que siguen a un cambio de fase de la alineación de trama del corrector de errores.

NOTA – Esto supone que los datos de vídeo no contienen tres emulaciones correctamente dispuestas en fase de la secuencia de alineación de trama para la corrección de errores durante el periodo de reconstitución.



**Figura H.1/H.263 – Trama de corrección de errores**

## Anexo I

### Modo codificación INTRA avanzada

Este anexo describe el modo codificación INTRA avanzada opcional de la presente Recomendación. La capacidad de este modo se señala por medios externos (por ejemplo, aplicando la Rec. UIT-T H.245). La utilización de este modo se indica en el campo PLUSPTYPE del encabezamiento de imagen.

#### I.1 Introducción

Este modo opcional altera la decodificación de macrobloques de tipo "INTRA" (los macrobloques de otros tipos no se ven afectados). La eficacia de la codificación de macrobloques INTRA se mejora mediante el empleo de:

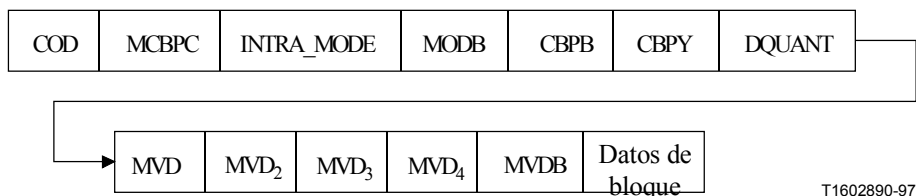
- 1) la predicción de bloques INTRA utilizando bloques INTRA contiguos para el mismo componente (Y, C<sub>B</sub> o C<sub>R</sub>);
- 2) la cuantificación inversa modificada para coeficientes INTRA; y
- 3) un VLC aparte para coeficientes INTRA.

Un bloque codificado INTRA determinado puede ser predicho a partir del bloque situado encima del que en esos momentos se está decodificando, a partir del bloque situado a la izquierda del bloque que se está decodificando o a partir de ambos. Existen casos especiales para situaciones en las que los bloques contiguos no están codificados INTRA o no están en el mismo segmento de imagen de vídeo. En la predicción de bloques se utilizan siempre datos del mismo componente de luminancia o de diferencia de color (Y, C<sub>B</sub> o C<sub>R</sub>) que el bloque que se está decodificando. En la

predicción, los coeficientes DC se predicen siempre de alguna manera. La primera fila de coeficientes AC se puede predecir a partir de los del bloque situado encima, o la primera columna de coeficientes AC se puede predecir a partir de los del bloque situado a la izquierda, o se puede predecir sólo el coeficiente DC como una media del bloque situado encima y el bloque situado a la izquierda, señalado macrobloque por macrobloque. Los coeficientes AC restantes nunca se predicen. La cuantificación inversa del coeficiente INTRADC se modifica para permitir un tamaño de paso de cuantificación variable, al contrario que en el cuerpo principal de la presente Recomendación en donde se utiliza un tamaño de paso fijo de 8 para coeficientes INTRADC. La cuantificación inversa de todos los coeficientes INTRA se efectúa sin una "zona muerta" en el espaciado de reconstrucción del cuantificador.

## I.2 Sintaxis

Cuando se utiliza el modo codificación INTRA avanzada, la sintaxis de la capa de macrobloque se altera como se especifica en la figura I.1. La sintaxis mostrada en dicha figura es la misma que se define en 5.3 salvo por la inserción de un campo INTRA\_MODE para macrobloques INTRA. El INTRA\_MODE sólo está presente cuando el MCBPC indica un macrobloque de tipo INTRA (macrobloque de tipo 3 ó 4). El modo de predicción se codifica utilizando los códigos de longitud variable que se muestran en el cuadro I.1. Se transmite un modo de predicción por cada macrobloque INTRA.



**Figura I.1/H.263 – Estructura de la capa de macrobloque**

**Cuadro I.1/H.263 – Códigos de longitud variable (VLC) para INTRA MODE**

Índice	Modo de predicción	VLC
0	0 (DC solamente)	0
1	1 (DC y AC vertical)	10
2	2 (DC y AC horizontal)	11

## I.3 Proceso de decodificación

Se efectúan dos exploraciones además de la exploración en zigzag. Las dos exploraciones adicionales se muestran en las partes a) y b) de la figura I.2 y la exploración en zigzag, en la figura 14.



1	2	3	4	11	12	13	14
5	6	9	10	18	17	16	15
7	8	20	19	27	28	29	30
21	22	25	26	31	32	33	34
23	24	35	36	43	44	45	46
37	38	41	42	47	48	49	50
39	40	51	52	57	58	59	60
53	54	55	56	61	62	63	64

a) Exploración horizontal alterna

1	5	7	21	23	37	39	53
2	6	8	22	24	38	40	54
3	9	20	25	35	41	51	55
4	10	19	26	36	42	52	56
11	18	27	31	43	47	57	61
12	17	28	32	44	48	58	62
13	16	29	33	45	49	59	63
14	15	30	34	46	50	60	64

b) Exploración vertical alterna  
(como en la Rec. UIT-T H.262)

**Figura I.2/H.263 – Esquemas de exploración DCT alternos para codificación INTRA avanzada**

En el caso de los bloques codificados INTRA, si modo de predicción = 0, se selecciona la exploración en zigzag de la figura 14 para todos los bloques del macrobloque; de no ser así, se utiliza la dirección de predicción para seleccionar la exploración para el macrobloque.

Modo de predicción = 1 utiliza el bloque adyacente en la dirección vertical para efectuar una predicción. Este modo de predicción está concebido para bloques INTRA dominados por un contenido de frecuencias horizontales más fuertes, por lo que el bloque adyacente en la dirección vertical se utiliza para predecir el contenido de frecuencias horizontales del bloque vigente, con una predicción de cero para todos los coeficientes que representan contenido AC vertical. A continuación se elige el esquema de exploración para explorar las frecuencias horizontales más fuertes antes que las verticales, empleando la exploración horizontal alterna.

Modo de predicción = 2 utiliza el bloque adyacente en la dirección horizontal para efectuar una predicción. Este modo de predicción está concebido para bloques INTRA dominados por un contenido de frecuencias verticales más fuertes, por lo que el bloque adyacente en la dirección horizontal se utiliza para predecir el contenido de frecuencias verticales del bloque vigente, con una predicción de cero para todos los coeficientes que representan contenido AC horizontal. A continuación se elige el esquema de exploración para explorar las frecuencias verticales más fuertes antes que las horizontales, empleando la exploración vertical alterna.

En el caso de bloques no INTRA, los bloques  $8 \times 8$  de coeficientes de transformación se exploran mediante la exploración en "zigzag" que se muestra en la figura 14.

Se utiliza una tabla de VLC aparte para todos los coeficientes INTRADC y AC INTRA. Esta tabla se especifica en el cuadro I.2. Se señala que las entradas de código de palabra VLC utilizadas en el cuadro I.2 son las mismas que las utilizadas en la tabla TCOEF normal (cuadro 16) empleada cuando no se utiliza la codificación INTRA avanzada, pero con una interpretación diferente de LEVEL y RUN (sin alterar LAST).

**Cuadro I.2/H.263 – Tabla de códigos de longitud variable (VLC) para coeficientes de transformada (TCOEF) INTRA**

Índice	LAST	RUN	LEVEL	Bits	Código VLC
0	0	0	1	3	10s
1	0	1	1	5	1111s
2	0	3	1	7	0101 01s
3	0	5	1	8	0010 111s

**Cuadro I.2/H.263 – Tabla de códigos de longitud variable (VLC) para coeficientes de transformada (TCOEF) INTRA**

Índice	LAST	RUN	LEVEL	Bits	Código VLC
4	0	7	1	9	0001 1111s
5	0	8	1	10	0001 0010 1s
6	0	9	1	10	0001 0010 0s
7	0	10	1	11	0000 1000 01s
8	0	11	1	11	0000 1000 00s
9	0	4	3	12	0000 0000 111s
10	0	9	2	12	0000 0000 110s
11	0	13	1	12	0000 0100 000s
12	0	0	2	4	110s
13	0	1	2	7	0101 00s
14	0	1	4	9	0001 1110s
15	0	1	5	11	0000 0011 11s
16	0	1	6	12	0000 0100 001s
17	0	1	7	13	0000 0101 0000s
18	0	0	3	5	1110s
19	0	3	2	9	0001 1101s
20	0	2	3	11	0000 0011 10s
21	0	3	4	13	0000 0101 0001s
22	0	0	5	6	0110 1s
23	0	4	2	10	0001 0001 1s
24	0	3	3	11	0000 0011 01s
25	0	0	4	6	0110 0s
26	0	5	2	10	0001 0001 0s
27	0	5	3	13	0000 0101 0010s
28	0	2	1	6	0101 1s
29	0	6	2	11	0000 0011 00s
30	0	0	25	13	0000 0101 0011s
31	0	4	1	7	0100 11s
32	0	7	2	11	0000 0010 11s
33	0	0	24	13	0000 0101 0100s
34	0	0	8	7	0100 10s
35	0	8	2	11	0000 0010 10s
36	0	0	7	7	0100 01s
37	0	2	4	11	0000 0010 01s
38	0	0	6	7	0100 00s
39	0	12	1	11	0000 0010 00s
40	0	0	9	8	0010 110s

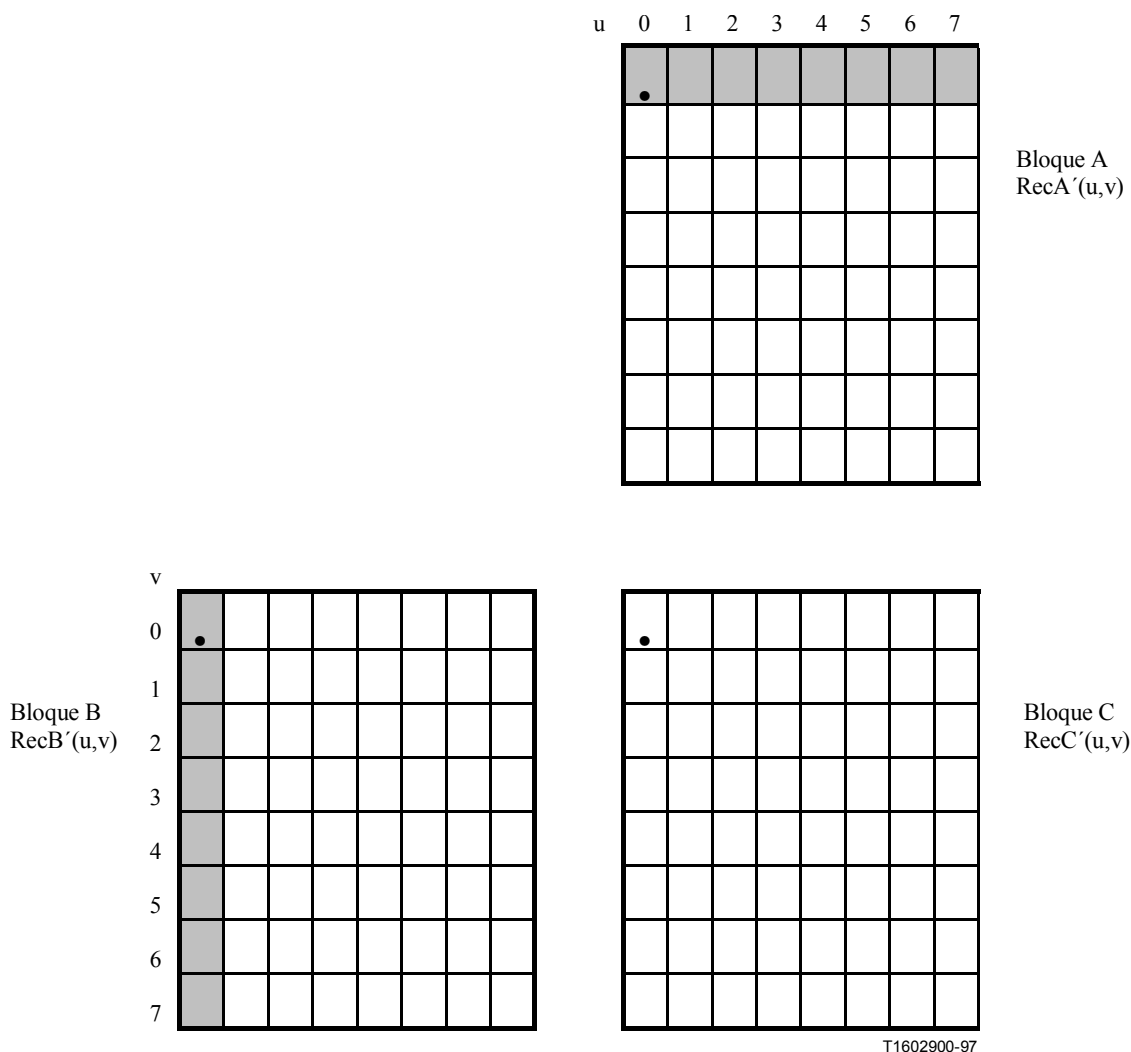
**Cuadro I.2/H.263 – Tabla de códigos de longitud variable (VLC) para coeficientes de transformada (TCOEF) INTRA**

Índice	LAST	RUN	LEVEL	Bits	Código VLC
41	0	0	23	13	0000 0101 0101s
42	0	2	2	8	0010 101s
43	0	1	3	8	0010 100s
44	0	6	1	9	0001 1100s
45	0	0	10	9	0001 1011s
46	0	0	12	10	0001 0000 1s
47	0	0	11	10	0001 0000 0s
48	0	0	18	10	0000 1111 1s
49	0	0	17	10	0000 1111 0s
50	0	0	16	10	0000 1110 1s
51	0	0	15	10	0000 1110 0s
52	0	0	14	10	0000 1101 1s
53	0	0	13	10	0000 1101 0s
54	0	0	20	12	0000 0100 010s
55	0	0	19	12	0000 0100 011s
56	0	0	22	13	0000 0101 0110s
57	0	0	21	13	0000 0101 0111s
58	1	0	1	5	0111s
59	1	14	1	10	0000 1100 1s
60	1	20	1	12	0000 0000 101s
61	1	1	1	7	0011 11s
62	1	19	1	12	0000 0000 100s
63	1	2	1	7	0011 10s
64	1	3	1	7	0011 01s
65	1	0	2	7	0011 00s
66	1	5	1	8	0010 011s
67	1	6	1	8	0010 010s
68	1	4	1	8	0010 001s
69	1	0	3	8	0010 000s
70	1	9	1	9	0001 1010s
71	1	10	1	9	0001 1001s
72	1	11	1	9	0001 1000s
73	1	12	1	9	0001 0111s
74	1	13	1	9	0001 0110s
75	1	8	1	9	0001 0101s
76	1	7	1	9	0001 0100s
77	1	0	4	9	0001 0011s

**Cuadro I.2/H.263 – Tabla de códigos de longitud variable (VLC) para coeficientes de transformada (TCOEF) INTRA**

Índice	LAST	RUN	LEVEL	Bits	Código VLC
78	1	17	1	10	0000 1100 0s
79	1	18	1	10	0000 1011 1s
80	1	16	1	10	0000 1011 0s
81	1	15	1	10	0000 1010 1s
82	1	2	2	10	0000 1010 0s
83	1	1	2	10	0000 1001 1s
84	1	0	6	10	0000 1001 0s
85	1	0	5	10	0000 1000 1s
86	1	4	2	11	0000 0001 11s
87	1	3	2	11	0000 0001 10s
88	1	1	3	11	0000 0001 01s
89	1	0	7	11	0000 0001 00s
90	1	2	3	12	0000 0100 100s
91	1	1	4	12	0000 0100 101s
92	1	0	9	12	0000 0100 110s
93	1	0	8	12	0000 0100 111s
94	1	21	1	13	0000 0101 1000s
95	1	22	1	13	0000 0101 1001s
96	1	23	1	13	0000 0101 1010s
97	1	7	2	13	0000 0101 1011s
98	1	6	2	13	0000 0101 1100s
99	1	5	2	13	0000 0101 1101s
100	1	3	3	13	0000 0101 1110s
101	1	0	10	13	0000 0101 1111s
102	ESCAPE			7	0000 011

Dependiendo del valor de INTRA\_MODE, uno u ocho coeficientes son residuos de la predicción que se deben añadir a un predictor como se describe más adelante. La figura I.3 muestra tres bloques  $8 \times 8$  de niveles de la transformada discreta de coseno (DCT, *discrete cosine transform*) reconstruidos finales para el mismo componente ( $Y$ ,  $C_B$ , o  $C_R$ ), etiquetados  $RecA'(u,v)$ ,  $RecB'(u,v)$  y  $RecC'(u,v)$ , donde  $u$  y  $v$  son índices de columna (horizontal) y fila (vertical), respectivamente. El proceso de reconstrucción difiere del procesamiento descrito en 6.2.1. Los residuos de INTRADC se reconstruyen de manera distinta empleando un tamaño de paso variable, en vez de utilizar el cuadro 15, y añadiendo a continuación un predictor a los valores residuales para obtener el valor de reconstrucción del coeficiente final. Los coeficientes INTRA distintos de los INTRADC se reconstruyen también de manera diferente a la de 6.2.1, utilizando un espaciamiento de reconstrucción sin "zona muerta" y añadiendo a continuación, en algunos casos, un predictor para obtener el valor de reconstrucción del coeficiente final. El bloque puede contener residuos de predicción tanto de DC como de AC.



**Figura I.3/H.263 – Tres bloques contiguos en el dominio DCT**

Las definiciones de MCBPC y CBPY se cambian cuando se utiliza codificación INTRA avanzada. Si se utiliza esta codificación, los coeficientes de transformación INTRADC ya no se tratan como un caso aparte sino del mismo modo que los coeficientes AC con respecto a MCBPC y CBPY. Esto significa que un INTRADC cero ya no se codificará como un LEVEL, sino que se incrementará simplemente la pasada de los coeficientes AC siguientes.

El proceso de cuantificación inversa de la parte B de una trama PB mejorada (véase el anexo M) no se altera por la utilización del modo codificación INTRA avanzada.

Se definen  $RecC(u,v)$  como los *residuos* de los coeficientes reconstruidos del bloque vigente. Para todos los coeficientes INTRA, el valor residual reconstruido se obtiene mediante la ecuación:

$$RecC(u,v) = 2 * QUANT * LEVEL(u,v) \quad u = 0, \dots, 7, v = 0, \dots, 7.$$

NOTA – En la ecuación anterior,  $LEVEL(u,v)$  representa una cantidad que tiene valor y signo.

Se definen  $RecC'(u,v)$  como los valores de los coeficientes reconstruidos  *finales*  del bloque vigente (después de los ajustes para la predicción, la "imparización" que se describe más adelante, y el recorte). Los valores de los coeficientes reconstruidos finales  $RecC'(u,v)$  se recuperan añadiendo  $RecC(u,v)$  a la predicción apropiada señalada en el campo INTRA\_MODE, alterando el bit menos significativo si es necesario para hacer que el coeficiente DC sea impar ("imparización") y recortando.

$RecA'(u,v)$  representa los valores de los coeficientes reconstruidos  *finales*  del bloque situado inmediatamente encima del bloque vigente.  $RecB'(u,v)$  representa los valores de los coeficientes reconstruidos  *finales*  del bloque situado inmediatamente a la izquierda del bloque vigente.

La posibilidad de utilizar valores de coeficientes reconstruidos para los bloques A y B en la predicción de los valores de los coeficientes del bloque C depende de si los bloques A y B se hallan en el mismo segmento de imagen de vídeo que el bloque C. Se dice que un bloque está "en el mismo segmento de imagen de vídeo" que otro bloque solamente si se cumplen las siguientes condiciones:

- 1) el bloque pertinente se encuentra dentro de las fronteras de la imagen; y
- 2) si no se utiliza el modo estructura en rebanada (véase el anexo K), el bloque pertinente se encuentra dentro del mismo GOB o no está presente ningún encabezamiento GOB para el GOB vigente; y
- 3) si se utiliza el modo estructura en rebanada, el bloque pertinente se encuentra dentro de la misma rebanada.

El bloque C que se ha de codificar sólo se predice a partir de bloques INTRA del mismo segmento de imagen de vídeo que el bloque C, como se muestra más abajo.

Si se utiliza modo de predicción = 0 (predicción de DC solamente) y los bloques A y B son bloques INTRA del mismo segmento de imagen de vídeo que el bloque C, el coeficiente DC del bloque C se predice a partir de la media (con truncamiento) de los coeficientes DC de los bloques A y B. Si sólo uno de los dos bloques, el A o el B, es un bloque INTRA del mismo segmento de imagen de vídeo que el bloque C, sólo se utiliza el coeficiente DC de ese bloque como predictor para el modo de predicción = 0. Si ninguno de los dos bloques, ni el A ni el B, es un bloque INTRA del mismo segmento de imagen de vídeo que el bloque C, la predicción utiliza el valor de 1024 como predictor del coeficiente DC.

Si modo de predicción = 1 ó 2 (predicción de DC y AC vertical o de DC y AC horizontal) y el bloque referenciado (bloque A o bloque B) no es un bloque INTRA del mismo segmento de imagen de vídeo que el bloque C, la predicción utiliza el valor de 1024 como predictor del coeficiente DC y el valor 0 como predictor de los coeficientes AC del bloque C.

Se aplica un proceso de "imparización" del coeficiente DC para reducir al mínimo la repercusión de los errores por desajuste de la transformada discreta de coseno inversa (IDCT). Determinados valores de los coeficientes pueden provocar desajustes por errores de redondeo entre diferentes implementaciones de la IDCT, sobre todo algunos valores de los coeficientes (0,0), (0,4), (4,0) y (4,4). Por ejemplo, un coeficiente DC de  $8k + 4$ , siendo k un entero, da lugar a un bloque de transformada inversa que tiene un valor constante de  $k + 0,5$ , para el que pequeños errores pueden dar lugar a redondeos de sentido diferente en las diferentes implementaciones.

Se define la función clipAC() para indicar el recorte en la gama de -2048 a 2047. Se define la función clipDC() para indicar el recorte en la gama de 0 a 2047. La función oddifyclipDC(x) se define como sigue:

```
If (x is even) {
    result = clipDC(x+1)
} else {
    result = clipDC(x)
}
```

A continuación se especifica como sigue la reconstrucción para cada modo de predicción INTRA, en donde el operador "/" se define como división por truncamiento:

Modo 0: predicción de DC solamente.

$$RecC'(u,v) = clipAC( RecC(u,v) ) \quad (u,v) \neq (0,0), \quad u = 0, \dots, 7, \quad v = 0, \dots, 7.$$

```

If (block A and block B are both INTRA coded and are both in the same
    video picture segment as block C) {
    tempDC = RecC(0,0) + ( RecA'(0,0) + RecB'(0,0) ) / 2
} else {
    If (block A is INTRA coded and is in the same video picture segment
        as block C) {
        tempDC = RecC(0,0) + RecA'(0,0)
    } else {
        If (block B is INTRA coded and is in the same video picture segment
            as block C) {
            tempDC = RecC(0,0) + RecB'(0,0)
        } else {
            tempDC = RecC(0,0) + 1024
        }
    }
}
RecC'(0,0) = oddifyclipDC( tempDC )

```

Modo 1: predicción de DC y AC a partir del bloque situado encima.

```

If (block A is INTRA coded and is in the same video picture segment as
    block C) {
    tempDC = RecC(0,0) + RecA'(0,0)
    RecC'(u,0) = clipAC( RecC(u,0) + RecA'(u,0) )          u = 1, ..., 7,
    RecC'(u,v) = clipAC( RecC(u,v) )                      u = 0, ..., 7, v = 1, ..., 7.
} else {
    tempDC = RecC(0,0) + 1024
    RecC'(u,v) = clipAC( RecC(u,v) )                      (u,v) ≠ (0,0), u = 0, ..., 7, v =
0, ..., 7
}
RecC'(0,0) = oddifyclipDC( tempDC )

```

Modo 2: predicción de DC y AC a partir del bloque situado a la izquierda.

```

If (block B is INTRA coded and is in the same video picture segment as
    block C) {
    tempDC = RecC(0,0) + RecB'(0,0)
    RecC'(0,v) = clipAC( RecC(0,v) + RecB'(0,v) )          v = 1, ..., 7,
    RecC'(u,v) = clipAC( RecC(u,v) )                      u = 1, ..., 7, v = 0, ..., 7.
} else {
    tempDC = RecC(0,0) + 1024
    RecC'(u,v) = clipAC( RecC(u,v) )                      (u,v) ≠ (0,0), u = 0, ..., 7,
                                                            v = 0, ..., 7
}
RecC'(0,0) = oddifyclipDC( tempDC )

```

## Anexo J

### Modo filtro de desbloqueo

#### J.1 Introducción

Este anexo describe la utilización de un filtro de borde de bloque opcional dentro del bucle de codificación. El objetivo principal del filtro de borde de bloque es reducir las perturbaciones que bloquean. El filtrado se efectúa en los bordes de bloques  $8 \times 8$ . Los vectores de movimiento pueden tener una resolución de  $8 \times 8$  ó  $16 \times 16$  (véase J.2). El procesamiento descrito en este anexo se aplica solamente para las imágenes P, I, EP o EI o a la parte de la imagen P de una trama PB mejorada. (El posible filtrado de imágenes B o de la parte de imagen B de una trama PB mejorada no es objeto de normalización, pero se recomienda la aplicación de alguna forma de filtrado para mejorar la calidad de la imagen.) La capacidad de este modo se señala por medios externos (por ejemplo, aplicando la Rec. UIT-T H.245). La utilización de este modo es la indicada en el campo PLUSPTYPE del encabezamiento de imagen.

NOTA – Puede ser mayor el problema de desajuste de la IDCT si se emplea el anexo J con un paso de cuantización pequeño y no se utiliza la IDCT del anexo W tanto en el codificador como en el decodificador. En algunos casos el proceso de filtrado del anexo J podría amplificar el desajuste por error de redondeo. Se debe evitar este problema en los codificadores, por ejemplo, no utilizando el anexo J si se tiene un paso de cuantización pequeño.

#### J.2 Relación con los modos vector de movimiento sin restricción (UMV) y predicción avanzada (AP) (anexos D y F)

La utilización del modo filtro de desbloqueo tiene unos efectos sobre la calidad de la imagen similares a los de la compensación de movimientos para bloques superpuestos (OBMC), definida en el anexo F, cuando se utiliza sola. Si se utilizan juntas ambas técnicas, puede mejorarse aún más la calidad de la imagen. El modo predicción avanzada (véase también el anexo F) consta de tres elementos:

- 1) Cuatro vectores de movimiento por macrobloque según lo definido en F.2.
- 2) Compensación de movimiento para bloque de luminancia superpuestos según lo definido en F.3.
- 3) Vectores de movimiento en fronteras de imagen según lo definido en D.1.

El modo filtro de desbloqueo incluye la posibilidad de utilizar cuatro vectores de movimiento por macrobloque y vectores de movimiento en fronteras de imagen, para proporcionar una calidad de funcionamiento máxima, cuando consideraciones relativas a su complejidad quizás impiden la utilización de la parte OBMC del modo predicción avanzada.

En resumen, las tres opciones definidas en los anexos D, F y J contienen los cinco elementos de codificación siguientes:

- 1) Vectores de movimiento en fronteras de imagen (D.1).
- 2) Ampliación de la gama de vectores de movimiento (D.2).
- 3) Cuatro vectores de movimiento por macrobloque (F.2).
- 4) Compensación de movimiento para bloques de luminancia superpuestos (F.3).
- 5) Filtro de borde de desbloqueo (J.3).

El cuadro J.1 indica cuáles de los cinco elementos se activan lo que depende de cuáles de las tres opciones definidas en los anexos D, F y J se activan.



**Cuadro J.1/H.263 – Elementos activables de los modos UMV, AP y DF**

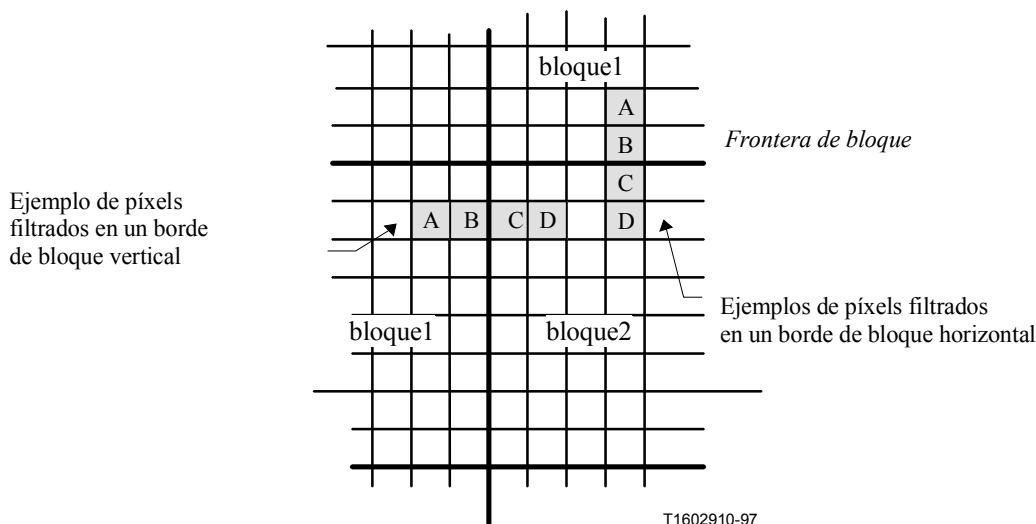
Modo vector de movimiento sin restricción	Modo predicción avanzada	Modo filtro de desbloqueo	Vectores de movimiento en fronteras de imagen	Ampliación de la gama de vectores de movimiento	Cuatro vectores de movimiento por macrobloque	Compensación de movimiento para bloques de luminancia superpuestos	Filtro de borde de desbloqueo
NOACTIVADO	NO ACTIVADO	NO ACTIVADO	NO ACTIVADO	NO ACTIVADO	NO ACTIVADO	NO ACTIVADO	NO ACTIVADO
NO ACTIVADO	NO ACTIVADO	ACTIVADO	ACTIVADO	NO ACTIVADO	ACTIVADO	NO ACTIVADO	ACTIVADO
NO ACTIVADO	ACTIVADO	NO ACTIVADO	ACTIVADO	NO ACTIVADO	ACTIVADO	ACTIVADO	NO ACTIVADO
NO ACTIVADO	ACTIVADO	ACTIVADO	ACTIVADO	NO ACTIVADO	ACTIVADO	ACTIVADO	ACTIVADO
ACTIVADO	NO ACTIVADO	NO ACTIVADO	ACTIVADO	ACTIVADO	NO ACTIVADO	NO ACTIVADO	NO ACTIVADO
ACTIVADO	NO ACTIVADO	ACTIVADO	ACTIVADO	ACTIVADO	ACTIVADO	NO ACTIVADO	ACTIVADO
ACTIVADO	ACTIVADO	NO ACTIVADO	ACTIVADO	ACTIVADO	ACTIVADO	ACTIVADO	NO ACTIVADO
ACTIVADO	ACTIVADO	ACTIVADO	ACTIVADO	ACTIVADO	ACTIVADO	ACTIVADO	ACTIVADO

### J.3 Definición de filtro de borde de desbloqueo

Las operaciones del filtro se efectúan en los bordes de bloques  $8 \times 8$  del lado codificador así como del lado decodificador. Los datos de la imagen reconstruida (la suma del error de predicción y de predicción reconstruida) se recortan para que caigan en la gama de 0 a 255 como se describe en 6.3.2. A continuación se aplica el filtrado, que altera la imagen que se ha de almacenar en el almacenamiento de imágenes para predicción futura. Las operaciones de filtrado incluyen un recorte adicional para asegurar que los valores de píxel resultantes permanecen en la gama de 0 a 255. No se efectúa ningún filtrado en la frontera de una imagen y, si se utiliza el modo codificación de segmento independiente, no se efectúa filtrado en los bordes de una rebanada cuando se utiliza el modo estructura en rebanada (véanse los anexos K y R) ni en el límite superior de los GOB que tienen encabezamiento GOB cuando no se utiliza el modo estructura en rebanada (véase el anexo R). Los datos de crominancia así como los de luminancia si son filtrados.

Cuando se utiliza el modo descrito en este anexo junto con el modo tramas PB mejoradas del anexo M, la predicción hacia atrás del macrobloque B se basa en el macrobloque P reconstruido (al que se denomina  $P_{REC}$  en G.5) después de la operación de recorte pero antes de las operaciones de filtro de borde de desbloqueo. La predicción hacia adelante del macrobloque B se basa en la versión filtrada de la imagen decodificada anterior (los mismos datos de imagen que se utilizan para la predicción del macrobloque P).

El filtro de desbloqueo funciona utilizando un conjunto de cuatro valores de píxel (recortados) en una línea horizontal o vertical de la imagen reconstruida, a los que se designa por A, B, C y D, de los cuales A y B pertenecen a un bloque llamado bloque 1 y C y D pertenecen a un bloque contiguo llamado bloque2, que está a la derecha o debajo del bloque1. La figura J.1 muestra ejemplos de posiciones de estos píxels.



**Figura J.1/H.263 – Ejemplos de posiciones de píxels filtrados**

Una o ambas de las siguientes condiciones se deben cumplir para aplicar el filtro en un determinado borde:

- Condición 1: el bloque1 pertenece a un macrobloque codificado (COD==0 || tipo MB == INTRA), o
- Condición 2: el bloque2 pertenece a un macrobloque codificado (COD==0 || tipo MB == INTRA).

Si se ha de aplicar filtrado en el borde, A, B, C y D deben ser sustituidos por A1, B1, C1 y D1, donde:

$$B1 = \text{clip}(B + d1) \quad [\text{recorte de } (B + d1)]$$

$$C1 = \text{clip}(C - d1) \quad [\text{recorte de } (C - d1)]$$

$$A1 = A - d2$$

$$D1 = D + d2$$

$$d = (A - 4B + 4C - D) / 8$$

$$d1 = \text{UpDownRamp}(d, \text{STRENGTH}) \quad (d1 = \text{función ascendente/descendente de "d"})$$

$$d2 = \text{clipd1}((A - D) / 4, d1/2)$$

$$\text{UpDownRamp}(x, \text{STRENGTH}) = \text{SIGN}(x) * (\text{MAX}(0, \text{abs}(x) - \text{MAX}(0, 2 * (\text{abs}(x) - \text{STRENGTH}))))$$

STRENGTH depende de QUANT y determina el grado de filtrado. La relación entre STRENGTH y QUANT se indica en el cuadro J.2.

QUANT = parámetro de cuantificación utilizado para el bloque2 si el bloque2 pertenece a un macrobloque codificado, o

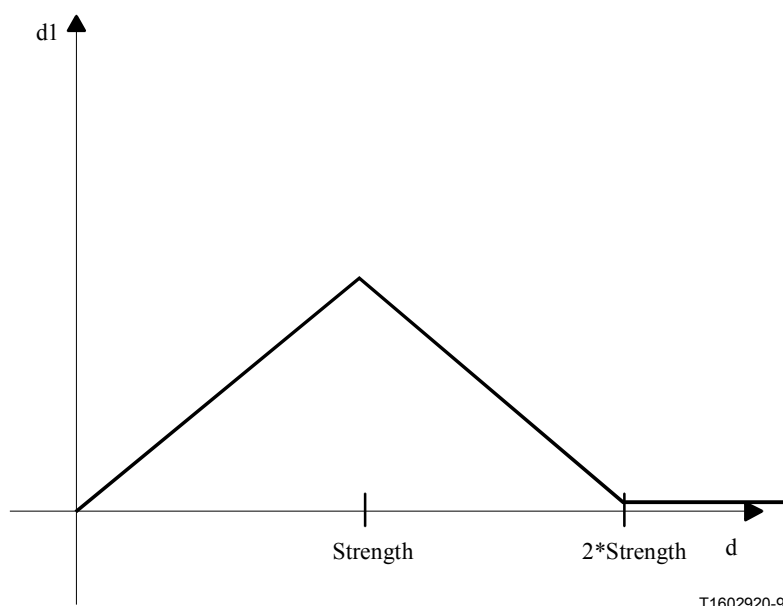
QUANT = parámetro de cuantificación utilizado para el bloque1 si el bloque2 no pertenece a un macrobloque codificado (pero el bloque1 sí).

**Cuadro J.2/H.263 – Relación entre QUANT y STRENGTH de filtro**

QUANT	STRENGTH	QUANT	STRENGTH
1	1	17	8
2	1	18	8
3	2	19	8
4	2	20	9
5	3	21	9
6	3	22	9
7	4	23	10
8	4	24	10
9	4	25	10
10	5	26	11
11	5	27	11
12	6	28	11
13	6	29	12
14	7	30	12
15	7	31	12
16	7		

La función  $\text{clip}(x)$  se define de acuerdo con 6.3.2 y la función  $\text{clipd1}(x, \text{lim})$  recorta  $x$  para que caiga en la gama  $\pm \text{abs}(\text{lim})$ . El símbolo "/" significa división por truncamiento hacia cero.

La figura J.2 muestra cómo varía el valor de  $d1$  en función de  $d$ . De esa dependencia resulta que el filtro sólo tiene efecto si  $d$  es más pequeño que  $2 * \text{STRENGTH}$  (y distinto de cero). Con ello se trata de evitar el filtrado de bordes verdaderos fuertes en el contenido de la imagen. No obstante, si se utiliza el modo actualización de resolución reducida, se fija  $\text{STRENGTH}$  en infinito y, en consecuencia, el valor de  $d1$  es siempre igual al valor de  $d$  (véase Q.7.2).



**Figura J.2/H.263 – Parámetro  $d1$  en función del parámetro  $d$  para modo filtro de desbloqueo**

Con la definición de  $d1$  se pretende garantizar que los pequeños desajustes entre el codificador y el decodificador seguirán siendo de escasa identidad y no se acumularán en múltiples imágenes de una secuencia de vídeo. Si tal cosa ocurriera crearía dificultades ya que, por ejemplo, con una condición que activara o desactivara el filtro, un desajuste de tan sólo  $\pm 1$  para  $d$  podría provocar a continuación la activación del filtro en el lado codificador y la desactivación en el lado de decodificador, o viceversa.

Debido a los efectos del redondeo, se ha de especificar el orden de los bordes en donde se efectúa el filtrado.

Filtrado en bordes horizontales:

Se supone básicamente que este proceso es el que ocurre en primer lugar. De manera más precisa,

los píxels  $\begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix}$  que se utilizan en el filtrado en un borde horizontal no deberán estar influenciados

por un filtrado previo en un borde vertical.

Filtrado en bordes verticales:

Antes de proceder al filtrado en un borde vertical utilizando los píxels (A, B, C, D), deberán producirse todas las modificaciones de píxels (A, B, C, D) resultantes del filtrado en un borde horizontal.

Se señala que si uno o más de los píxels (A, B, C, D) que interviene en un proceso de filtrado queda fuera de una imagen, el filtrado no tiene lugar. Además, si se utiliza el modo codificación de segmento independiente (véase el anexo R) y uno o más de los píxels (A, B, C, D) que intervienen en un proceso de filtrado se encuentran en segmentos de imagen de vídeo diferentes (véase I.3 a propósito del caso en que se considera que un bloque se encuentra en el mismo segmento de imagen de vídeo que otro bloque), no se efectúa el filtrado.

## Anexo K

### Modo estructura en rebanada

#### K.1 Introducción

Este anexo describe el modo estructura en rebanada opcional de la presente Recomendación. La capacidad de este modo se señala por medios externos (por ejemplo, aplicando la Rec. UIT-T H.245). La utilización de este modo se indica en el campo PLUSPTYPE del encabezamiento de imagen. Para facilitar su empleo óptimo en varios ambientes, este modo contiene dos submodos que también pueden ser señalizados por medios externos (por ejemplo, aplicando la Rec. UIT-T H.245). Los submodos se utilizan para indicar si se utilizarán o no rebanadas rectangulares y/o si las rebanadas se transmitirán en orden secuencial o se enviarán en un orden cualquiera.

Una rebanada consiste en un encabezamiento de rebanada seguido de macrobloques consecutivos en orden de exploración. La rebanada que sigue inmediatamente al código de comienzo de imagen en el tren de bits de una imagen (que no es necesariamente la rebanada que comienza con el macrobloque 0) es una excepción. En este caso, sólo se transmite parte del encabezamiento de la imagen, como se describe en K.2. La capa de rebanada define un segmento de imagen de vídeo y se utiliza en vez de la capa de GOB en este modo opcional. Un segmento de imagen de vídeo de

rebanada comienza en el límite de un macrobloque de la imagen y contiene varios macrobloques. Rebanadas diferentes dentro de una misma imagen no deberán superponerse entre sí, y cada macrobloque deberá pertenecer a una, y sólo una, rebanada.

Este modo contiene dos submodos que se señalan en el campo SSS del encabezamiento de la imagen:

- 1) El submodo rebanada rectangular (RS, *rectangular slice*): Cuando se utilice RS, la rebanada deberá ocupar una región rectangular de anchura especificada por el parámetro SWI del encabezamiento de la rebanada en unidades de macrobloques, y contiene varios macrobloques en orden de exploración dentro de la región rectangular. Cuando no se utiliza el submodo rebanada rectangular, el campo SWI no está presente en el encabezamiento de la rebanada y una rebanada contiene varios macrobloques en orden de exploración dentro de la imagen en su conjunto.
- 2) El submodo orden arbitrario de las rebanadas (ASO, *arbitrary slice ordering*): Cuando se utiliza ASO, las rebanadas pueden aparecer en cualquier orden dentro del tren de bits. Cuando no se utilizan, las rebanadas deben ser enviadas en el orden (único) según el cual crece de manera estricta el campo MBA del encabezamiento de rebanada de una rebanada a la siguiente en la imagen.

Los límites de las rebanadas se tratan de manera diferente a la de los límites de simples macrobloques, para permitir que las ubicaciones de los encabezamientos de las rebanadas dentro del tren de bits actúen como puntos de resincronización a efectos de restablecimiento del funcionamiento normal tras producirse errores en los bits y pérdida de paquetes, de modo que sea posible decodificar rebanadas fuera de orden dentro de una imagen. Así pues, ninguna dependencia de datos puede atravesar los límites de las rebanadas dentro de la imagen vigente, excepto en el caso del modo filtro de bloqueo con el que, si se utiliza sin el modo codificación de segmento independiente, se filtra en los límites de los bloques de la imagen. No obstante, los vectores de movimiento dentro de una rebanada pueden hacer que dependencias de datos atraviesen los límites de la rebanada en la imagen de referencia utilizada a efectos de predicción, a menos que se utilice el modo opcional codificación de segmento independiente.

Las reglas que siguen se han adoptado para asegurar que las ubicaciones de los límites de las rebanadas pueden actuar como puntos de resincronización y que las rebanadas pueden ser enviadas en un orden cualquiera sin provocar retardos de decodificación adicionales:

- 1) La predicción de los valores del vector de movimiento es la misma que si estuviera presente un encabezamiento GOB (véase 6.1.1), impidiendo la utilización de vectores de movimiento de bloques fuera de la rebanada de que se trate para la predicción de los valores de los vectores de movimiento dentro de la rebanada.
- 2) El modo codificación INTRA avanzada, (véase el anexo I) trata el límite de la rebanada como si fuera una frontera de imagen con respecto a la predicción de valores de coeficientes DCT de bloque INTRA.
- 3) La asignación de vectores de movimiento distantes para utilizarlos en la compensación de movimiento para bloques superpuestos en el modo predicción avanzada impide también la utilización de vectores de movimiento de bloques fuera de la rebanada de que se trate para utilizarlos como vectores de movimiento distantes (véase F.3).

## **K.2 Estructura de la capa de rebanada**

En la figura K.1 se muestra la estructura de la capa de rebanada de la sintaxis para todas las rebanadas excepto la que sigue inmediatamente al código de comienzo de imagen del tren de bits de una imagen. En el caso de la imagen que sigue al código de comienzo de imagen, sólo se incluyen los bits de prevención de emulación (SEPB1, SEPB3 y, de manera condicional, SEPB2 como se

especifica a continuación), el campo MBA y, cuando se utiliza el submodo RS, también el campo SWI.

SSTUF	SSC	SEPBI	SSBI	MBA	SEPBI	SQUANT	SWI	SEPBI	GFID	Datos de macrobloque
-------	-----	-------	------	-----	-------	--------	-----	-------	------	----------------------

**Figura K.1/H.263 – Estructura de la capa de rebanada**

Para la descripción de GFID y de la capa de macrobloque véanse 5.2.5 y 5.3, respectivamente.

### **K.2.1 Relleno (SSTUF, *stuffing*) (Longitud variable)**

Palabra de código de longitud variable que consta de menos de 8 bits. Los codificadores insertarán esta palabra de código directamente antes de una palabra de código SSC cuando se necesite para asegurar que el SSC está alineado en bytes. Si SSTUF está presente, su último bit será el último (el menos significativo) de un byte, de tal manera que el comienzo de la palabra de código SSC esté alineado en bytes. Los decodificadores se diseñarán de modo que descarten SSTUF. Se señala que para el relleno de SSTUF se utiliza 0.

### **K.2.2 Código de comienzo de rebanada (SSC, *slice start code*) (17 bits)**

Es una palabra de 17 bits. Su valor es 0000 0000 0000 0000 1. Los códigos de comienzo de rebanada estarán alineados en bytes. Esto se puede conseguir insertando SSTUF antes del código de comienzo de tal manera que el primer bit del primer código de comienzo sea el primer bit (el más significativo) de un byte. El código de comienzo de rebanada no está presente en la rebanada que sigue al código de comienzo de imagen.

### **K.2.3 Bit 1 de prevención de emulación de rebanada (SEPBI, *slice emulation prevention bit 1*) (1 bit)**

Un solo bit cuyo valor es siempre "1", que se incluye para evitar la emulación del código de comienzo.

### **K.2.4 Indicador de subtren de bits de rebanada (SSBI, *slice sub-bitstream indicator*) (4 bits)**

Palabra de código de cuatro bits de longitud que sólo está presente cuando CPM = "1" en el encabezamiento de la imagen. El SSBI indica el número del subtren de bits de la rebanada para funcionamiento multipunto de presencia continua y múltiplex de vídeo, como se describe en el anexo C. La correspondencia entre el valor del SSBI y el número del subtren de bits se muestra en el cuadro K.1. El SSBI no está presente para la rebanada que sigue al código de comienzo de imagen.

**Cuadro K.1/H.263 – Valores de SSBI y números de subtren de bits asociados**

Número de subtren de bits	Valor del campo SSBI	Valor de GN emulado
0	1001	25
1	1010	26
2	1011	27
3	1101	29

### **K.2.5 Dirección de macrobloque (MBA, *macroblock address*) (5/6/7/9/11/12/13/14 bits)**

Palabra de código cuya longitud depende del tamaño de la imagen en curso y de si está en vigor el modo actualización de resolución reducida (véase el anexo Q). Los bits son la representación

binaria del número de macrobloque del primer macrobloque de la rebanada de que se trate contando a partir del comienzo de la imagen en orden de exploración y empezando con el macrobloque número 0 en la esquina superior izquierda. La MBA identifica de manera exclusiva el macrobloque de la imagen con el que comienza la rebanada vigente. En el cuadro K.2 se indican las longitudes de palabra de código para esta palabra de código. Para tamaños de imagen personalizados, la anchura de campo viene dada por la primera entrada del cuadro que tiene un número igual o mayor de macrobloques, y el valor máximo es el número de macrobloques de la imagen vigente menos uno. Cuando se utiliza el modo actualización de resolución reducida, el tamaño de imagen pertinente es el tamaño de imagen de actualización de resolución inferior indicado en el encabezamiento de imagen (véase el anexo Q).

**Cuadro K.2/H.263 – Especificación de parámetro MBA**

Formato de imagen	Por defecto		Modo RRU	
	Valor máximo	Anchura de campo	Valor máximo	Anchura de campo
sub-cuarto de CIF	47	6	11	5
cuarto de CIF	98	7	29	6
CIF	395	9	98	7
4 veces CIF	1583	11	395	9
16 veces CIF	6335	13	1583	11
2048 × 1152	9215	14	2303	12

**K.2.6 Bit 2 de prevención de emulación de rebanada (SEPB2, *slice emulation prevention bit 2*) (1 bit)**

Un solo bit cuyo valor es siempre "1", que se incluye bajo determinadas condiciones para evitar la emulación del código de comienzo. Para rebanadas distintas de la que sigue al código de comienzo de imagen, el SEPB2 se incluye sólo si la anchura de campo MBA es superior a 11 bits y CPM = "0" en el encabezamiento de imagen, o si la anchura de campo MBA es superior a 9 bits y CPM = "1" en el encabezamiento de imagen. Para la rebanada que sigue al código de comienzo de imagen, el SEPB2 sólo se incluye si se utiliza el submodo rebanada rectangular.

**K.2.7 Información de cuantificador (SQUANT, *quantizer information*) (5 bits)**

Palabra de código de longitud fija de cinco bits que indica el cuantificador QUANT que se ha de utilizar para esa rebanada hasta que sea actualizada por cualquier DQUANT subsiguiente. Las palabras de código son las representaciones binarias naturales de los valores de QUANT que, teniendo la mitad de tamaño que los pasos, varían de 1 a 31. SQUANT no está presente para la rebanada que sigue al código de comienzo de imagen.

**K.2.8 Indicación de anchura de rebanada (SWI, *slice width indication*) en macrobloques (3/4/5/6/7 bits)**

Palabra de código que sólo está presente si está activo el submodo rebanada rectangular y tiene una longitud que depende del tamaño de imagen vigente y de si está activo el modo actualización de resolución reducida, como se especifica en el cuadro K.3. Para tamaños de imagen personalizados, la anchura de campo viene dada por el tamaño de formato normalizado siguiente cuya anchura es igual o mayor (cuarto de CIF, CIF, ...), y el valor máximo es el número de macrobloques en toda la imagen menos uno. La última fila del cuadro indica la anchura de campo para tamaños de imagen mayores que los de 16 veces CIF. La SWI se refiere a la anchura de la rebanada rectangular vigente

cuyo primer macrobloque (superior izquierdo) lo especifica la MBA. El cálculo de la anchura efectiva de la rebanada es como sigue:

$$\text{Anchura efectiva de la rebanada} = \text{SWI} + 1$$

Cuando se utiliza el modo actualización de resolución reducida, el tamaño de imagen pertinente es el tamaño de imagen de resolución inferior para la información de actualización, en vez del tamaño de imagen indicado en el encabezamiento de imagen.

**Cuadro K.3/H.263 – Especificación de parámetro SWI**

Formato de imagen	Por defecto		Modo RRU	
	Valor máximo	Anchura de campo	Valor máximo	Anchura de campo
sub-cuarto de CIF	7	4	3	3
cuarto de CIF	10	4	5	3
CIF	21	5	10	4
4 veces CIF	43	6	21	5
16 veces CIF	87	7	43	6
1412...2048 píxels de anchura	127	7	63	6

### **K.2.9 Bit 3 de prevención de emulación de rebanada (SEPB3, *slice emulation prevention bit 3*) (1 bit)**

Un solo bit cuyo valor es siempre "1" para evitar la emulación del código de comienzo.

## **Anexo L**

### **Especificación de la información sobre mejoras suplementarias**

#### **L.1 Introducción**

Este anexo describe el formato de la información sobre mejoras suplementarias enviada al campo PSUPP de la capa de imagen de la presente Recomendación. La capacidad de un decodificador de proporcionar algunas o todas las capacidades mejoradas descritas en este anexo puede señalizarse por medios externos (por ejemplo, aplicando la Rec. UIT-T H.245). Los decodificadores que no proporcionan las capacidades mejoradas pueden simplemente descartar cualquier bit de información PSUPP que aparece en el tren de bits. La presencia de esta información sobre mejoras suplementarias se indica en PEI, y se inserta un bit PEI adicional entre todos los octetos de datos PSUPP, como se indica en 5.1.24 y 5.1.25.

En este anexo se hace una distinción entre la "imagen decodificada" y la "imagen visualizada". A los efectos de este anexo, la "imagen visualizada" es una imagen que tiene el mismo formato de imagen especificado para la imagen vigente por la capa de imagen de la sintaxis de tren de bits de vídeo. La "imagen visualizada" se construye tal como se describe en este anexo a partir de la imagen decodificada, la imagen visualizada anterior, la información sobre mejoras suplementarias que aquí se describen y, parcialmente, en algunos casos, a partir de una imagen de segundo plano externamente controlada.



## L.2 Formato PSUPP

Los datos PSUPP constan de una indicación del tipo de función de cuatro bits FTYPE, seguida de la especificación del tamaño de los datos de los parámetros de cuatro bits DSIZE, seguida de octetos DSIZE de los datos de los parámetros de la función, seguida opcionalmente por otra indicación del tipo de función, y así sucesivamente. Un valor de la indicación del tipo de función se define como un código de escape en previsión de una futura extensibilidad que permita definir más de quince funciones distintas. Un decodificador que recibe una indicación del tipo de función que no admite puede descartar los datos de los parámetros de función para esa función y comprobar a continuación una indicación subsiguiente de tipo de función que puede ser admitida. En el cuadro L.1 se indican los valores FTYPE definidos.

**Cuadro L.1/H.263 – Valores del tipo de función FTYPE**

0	Reservado
1	No hacer nada
2	Petición de congelación de toda la imagen
3	Petición de congelación de parte de la imagen
4	Petición de congelación y redimensionamiento de parte de la imagen
5	Petición de liberación de la parte de la imagen congelada
6	Rótulo de instantánea de toda la imagen
7	Rótulo de instantánea de parte de la imagen
8	Rótulo de comienzo de segmento de tiempo de vídeo
9	Rótulo de terminación de segmento de tiempo de vídeo
10	Rótulo de comienzo del segmento de refinamiento progresivo
11	Rótulo de terminación de segmento de refinamiento progresivo
12	Información de referencia de cromaticidad
13	Reservado
14	Reservado
15	Tipo de función ampliada

### L.3 No hacer nada

La no realización de ninguna acción se pide mediante la función No hacer nada. Esta función se utiliza para evitar la emulación del código de comienzo. Siempre que los últimos cinco o más bits del octeto final del octeto PSUPP anterior sean todos ceros y no se tenga que enviar ninguna petición de función PSUPP adicional, la función No hacer nada se insertará en PSUPP para evitar la posibilidad de emulación del código de comienzo. La función No hacer nada puede enviarse también cuando no es requerida por la regla expresada en la oración anterior. DSIZE tendrá un valor cero para la función No hacer nada.

### L.4 Petición de congelación de toda la imagen

La función petición de congelación de toda la imagen indica que los contenidos de toda la imagen de vídeo visualizada con anterioridad deben mantenerse inalterados, sin actualizar la imagen visualizada utilizando los contenidos de la imagen decodificada vigente. La imagen visualizada se mantendrá entonces sin cambios hasta que el bit de liberación de la imagen congelada del PTYPE vigente o de un PTYPE posterior se fije a 1, o hasta que tenga lugar la temporización, lo que ocurra primero. La petición se demorará debido a la temporización tras cinco segundos o cinco imágenes, lo que represente una duración mayor. La temporización puede evitarse emitiendo otra petición de

congelación de toda la imagen antes de la expiración del periodo de temporización o al expirar el mismo (por ejemplo, se repite la petición en el encabezamiento de la primera imagen con una referencia temporal que indique un intervalo de tiempo mayor o igual a cinco segundos a partir de la emisión, o en el encabezamiento de la quinta imagen después de la emisión). DSIZE tendrá el valor cero para la función petición de congelación de toda la imagen.

### **L.5 Petición de congelación de parte de la imagen**

La función petición de congelación de parte de la imagen indica que los contenidos de una zona rectangular especificada de la imagen de vídeo visualizada con anterioridad deben mantenerse inalterados, sin actualizar la zona especificada de la imagen visualizada utilizando los contenidos de la imagen decodificada vigente. La zona especificada de la imagen visualizada se mantendrá entonces sin cambios hasta que el bit de liberación de la imagen congelada del PTYPE vigente o de un PTYPE posterior se fije a 1, hasta que se reciba una petición de liberación de la congelación de parte de la imagen que afecte a la zona especificada, hasta que el formato de fuente especificado en un encabezamiento de imagen sea distinto del formato de los encabezamientos de imágenes anteriores, o hasta que tenga lugar la temporización, lo que ocurra primero. Todo cambio en el formato de la fuente de la imagen actuará como una liberalización de congelación para todas las peticiones activas de congelación de parte de la imagen. La petición se demorará debido a la temporización tras cinco segundos o cinco imágenes, lo que represente una duración mayor. La temporización puede evitarse emitiendo una petición idéntica de congelación de parte de la imagen antes de la expiración del periodo de temporización o al expirar el mismo (por ejemplo, se repite la petición en el encabezamiento de la primera imagen con una referencia temporal que indique un intervalo de tiempo mayor o igual a cinco segundos a partir de la emisión, o en el encabezamiento de la quinta imagen después de la emisión). DSIZE será igual a 4 para la petición de congelación de parte de la imagen. Los cuatro octetos de PSUPP que siguen contienen la localización horizontal y vertical de la esquina superior izquierda del rectángulo de imagen congelado, y la anchura y altura del rectángulo, respectivamente, utilizando ocho bits cada una expresadas en unidades de ocho píxels. Por ejemplo, una zona con una anchura de 24 píxels y una altura de 16 píxels en la esquina superior izquierda de la visualización vídeo se especifica con los cuatro parámetros (0, 0, 3, 2).

### **L.6 Petición de congelación y redimensionamiento de parte de la imagen**

La función petición de congelación y redimensionamiento de parte de la imagen indica que los contenidos de una zona rectangular especificada de la imagen de vídeo visualizada con anterioridad deberían redimensionarse para ajustarlos a una parte más pequeña de la imagen de vídeo visualizada, que debería mantenerse a continuación inalterada, sin actualizar la zona especificada de la imagen visualizada utilizando los contenidos de la imagen decodificada vigente. La zona especificada de la imagen visualizada se mantendrá entonces sin cambios hasta que el bit de liberación de la imagen congelada del PTYPE vigente o de un PTYPE posterior se fije a 1, hasta que se reciba una petición de liberación de la congelación de parte de la imagen que afecte a la zona especificada, hasta que el formato de fuente especificado en un encabezamiento de imagen sea distinto del formato de los encabezamientos de imágenes anteriores, o hasta que tenga lugar la temporización, lo que ocurra primero. Todo cambio en el formato de la fuente de la imagen actuará como una liberalización de congelación para todas las peticiones activas de congelación y redimensionamiento de parte de la imagen. La petición se demorará debido a la temporización tras cinco segundos o cinco imágenes, lo que represente una duración mayor. La temporización puede evitarse emitiendo una petición de congelación de parte de la imagen para la zona afectada de la imagen visualizada antes de la expiración del periodo de temporización o al expirar el mismo (por ejemplo, se emite la petición congelación de parte de la imagen en el encabezamiento de la primera imagen con una referencia temporal que indique un intervalo de tiempo mayor o igual a cinco segundos a partir de la emisión, o en el encabezamiento de la quinta imagen después de la emisión). DSIZE será igual a 8 para la petición de congelación y redimensionamiento de parte de la imagen. Los ocho octetos de datos PSUPP que siguen contienen 32 bits utilizados para especificar la región

rectangular de la zona afectada de la imagen visualizada, y a continuación 32 bits utilizados para especificar la región rectangular correspondiente de la zona afectada de la imagen decodificada. La anchura y la altura de la región rectangular en la imagen decodificada serán iguales a  $2^i$  veces la anchura y la altura especificadas para la región rectangular en la imagen visualizada, donde  $i$  es un entero de la gama de 1 a 8. La localización y el tamaño de cada una de estas dos regiones rectangulares se especifican utilizando el mismo formato en que se especifica esa región en la función petición de congelación de parte de la imagen.

#### **L.7 Petición de liberación de la parte de la imagen congelada**

La función petición de liberación de la parte de la imagen congelada indica que los contenidos de una zona rectangular especificada de la imagen de vídeo visualizada sean actualizados mediante las imágenes decodificadas vigentes y posteriores. DSIZE será igual a 4 para la petición de liberación de la parte de la imagen congelada. Los cuatro octetos de datos PSUPP que siguen especifican una región rectangular de la imagen visualizada en el mismo formato en que se especifica esa región en la función petición de congelación de parte de la imagen.

#### **L.8 Rótulo de la instantánea de toda la imagen**

La función rótulo de la instantánea de toda la imagen indica que la imagen vigente está etiquetada para uso externo como una instantánea de imagen fija del contenido de vídeo. DSIZE será igual a 4 para la función rótulo de la instantánea de toda la imagen. Los cuatro octetos de datos PSUPP que siguen especifican un número de identificación de instantánea para uso externo.

#### **L.9 Rótulo de la instantánea de parte de la imagen**

La función rótulo de la instantánea de parte de la imagen indica que una zona rectangular especificada de la imagen vigente está etiquetada para uso externo como una instantánea de imagen fija del contenido de vídeo. DSIZE será igual a 8 para la función rótulo de la instantánea de parte de la imagen. Los primeros cuatro octetos de datos PSUPP que siguen especifican un número de identificación de instantánea para uso externo, y los cuatro octetos restantes de datos PSUPP que siguen especifican una región rectangular de la imagen decodificada en el mismo formato en que esa región se especifica en la función petición de congelación de parte de la imagen.

#### **L.10 Rótulo de comienzo de segmento de tiempo de vídeo**

La función rótulo de comienzo de segmento de tiempo de vídeo indica que el comienzo de una subsecuencia específica de datos de vídeo está etiquetada como una sección útil del contenido de vídeo para uso externo, que se inicia con la imagen vigente. La subsecuencia rotulada de datos de vídeo continuará hasta que la detenga la recepción de una función rótulo de terminación de segmento de tiempo de vídeo correlativa o hasta la temporización, lo que ocurra primero. La subsecuencia rotulada terminará debido a la temporización tras cinco segundos o cinco imágenes, lo que representa una duración mayor. La temporización puede evitarse emitiendo de una función rótulo del comienzo del segmento de tiempo de vídeo idéntica antes de la expiración del periodo de temporización o al expirar el mismo (por ejemplo, se repite la función rótulo de comienzo del segmento de tiempo de vídeo en el encabezamiento de la primera imagen con una referencia temporal que indique un intervalo de tiempo mayor o igual a cinco segundos a partir de la emisión, o en el encabezamiento de la quinta imagen después de la emisión). DSIZE será igual a 4 para la función rótulo de comienzo del segmento de tiempo de vídeo. Los cuatro octetos de datos PSUPP que siguen especifican un número de identificación del segmento de tiempo de vídeo para uso externo.

#### **L.11 Rótulo de terminación de segmento de tiempo de vídeo**

La función rótulo de terminación de segmento de tiempo de vídeo indica que la terminación de una subsecuencia específica de datos de vídeo está etiquetada como una sección útil del contenido de

vídeo para uso externo, que termina con la imagen anterior. DSIZE será igual a 4 para la función rótulo de comienzo de segmento de tiempo de vídeo. Los cuatro octetos de datos PSUPP que siguen especifican un número de identificación del segmento de tiempo de vídeo para uso externo.

#### **L.12 Rótulo de comienzo de segmento de refinamiento progresivo**

La función rótulo de comienzo de segmento de refinamiento progresivo indica el comienzo de una subsecuencia especificada de datos de vídeo que está etiquetada como la imagen vigente seguida de una secuencia de cero o más imágenes de refinamiento de la calidad de la imagen vigente, y no como una representación de una escena en continuo movimiento. La subsecuencia rotulada de datos de vídeo continuará hasta que la detenga la recepción de una función rótulo de terminación de segmento de refinamiento progresivo correlativa o hasta la temporización, lo que ocurra primero. La subsecuencia rotulada terminará debido a la temporización tras cinco segundos o cinco imágenes, lo que represente una duración mayor. La temporización puede evitarse emitiendo de una función rótulo del comienzo del segmento de refinamiento progresivo idéntica antes de la expiración del periodo de temporización o al expirar el mismo (por ejemplo, se repite la función rótulo de comienzo del segmento de refinamiento progresivo en el encabezamiento de la primera imagen con una referencia temporal que indique un intervalo de tiempo mayor o igual a cinco segundos a partir de la emisión, o en el encabezamiento de la quinta imagen después de la emisión). DSIZE será igual a 4 para la función rótulo del comienzo del segmento de tiempo de vídeo. Los cuatro octetos de datos PSUPP que siguen especifican un número de identificación del segmento de refinamiento progresivo para uso externo.

#### **L.13 Rótulo de terminación de segmento de refinamiento progresivo**

La función rótulo de terminación de segmento de refinamiento progresivo indica la terminación de una subsecuencia especificada de datos de vídeo que está etiquetada como una imagen inicial seguida de una secuencia de cero o más imágenes de refinamiento de la calidad de la imagen inicial, y que termina con la imagen anterior. DSIZE será igual a 4 para la función rótulo de comienzo de segmento de tiempo de vídeo. Los cuatro octetos de datos PSUPP que siguen especifican un número de identificación de refinamiento progresivo para uso externo.

#### **L.14 Información de referencia de cromaticidad**

La función información de referencia de cromaticidad (CKIF, *chroma keying information function*) indica que la técnica "de referencia de cromaticidad" se utiliza para representar píxeles "transparentes" y "semitransparentes" en las imágenes de vídeo decodificadas. Cuando se presentan en la visualización, los píxeles "transparentes" no se visualizan. En vez de ello, se revela una imagen de segundo plano que es una imagen de referencia anterior o bien una imagen controlada externamente. Los píxeles semitransparentes se visualizan combinando el valor del píxel en la imagen vigente con el valor correspondiente en la imagen de segundo plano. Se utiliza un octeto para indicar el valor del color de referencia para cada componente ( $Y$ ,  $C_B$ , o  $C_R$ ) que se utiliza para la referencia de cromaticidad. Para representar píxeles que van a ser "semitransparentes", se utilizan dos valores de umbral, indicados como  $T_1$  y  $T_2$ . Sea  $\alpha$  la transparencia de un píxel;  $\alpha = 255$  indica que el píxel es opaco, y  $\alpha = 0$ , que el píxel es transparente. Con otros valores de  $\alpha$ , el valor resultante para un píxel debe ser una combinación ponderada del valor del píxel en la imagen vigente y el valor del píxel de la imagen de segundo plano (que se especifica externamente). Pueden utilizarse los valores de  $\alpha$  para formar una imagen que se denomina "alpha map". Así pues, el valor resultante para cada componente puede ser:

$$[\alpha \cdot X + (255 - \alpha) \cdot Z] / 255$$

donde  $X$  es el valor del componente píxel decodificado (para  $Y$ ,  $C_B$ , o  $C_R$ ), y  $Z$  es el valor del componente píxel correspondiente de la imagen de segundo plano.

El valor  $\alpha$  puede calcularse como sigue. En primer lugar, se calcula la distancia entre el color del píxel y el valor del color de referencia:

$$d = A_Y(X_Y - K_Y)^2 + A_B(X_B - K_B)^2 + A_R(X_R - K_R)^2$$

en donde  $X_Y$ ,  $X_B$ , y  $X_R$  son los valores  $Y$ ,  $C_B$ , y  $C_R$  del color del píxel decodificado,  $K_Y$ ,  $K_B$ , y  $K_R$  son los parámetros del color de referencia correspondientes, y  $A_Y$ ,  $A_B$  y  $A_R$  son los bits de las banderas de referencia que indican cuáles son los componentes de color utilizados como referencia. Una vez calculada la distancia  $d$ , el valor  $\alpha$  puede estimarse tal como se especifica en el pseudocódigo siguiente:

Para cada píxel

```
if (d < T1) then  $\alpha = 0$ ;  
else if (d > T2) then  $\alpha = 255$ ;  
else  $\alpha = \lfloor 255 \cdot (d - T_1) \rfloor / (T_2 - T_1)$ 
```

No obstante, el método preciso de realización de la operación de referencia de crominancia en el decodificador no se especifica en el presente anexo, porque no es necesaria la especificación normativa del método para el interfuncionamiento. El proceso aquí indicado se describe a efectos ilustrativos, para transmitir la interpretación pretendida de los parámetros de datos.

El valor  $\alpha$  obtenido es simplemente una función de  $X_Y$ ,  $X_B$  y  $X_R$ , por lo que puede elaborarse una tabla de consulta (LUT, *look-up table*) para realizar la operación anterior. Una LUT como esa tiene  $2^{8 \times N}$  entradas correspondientes a todos los valores de píxeles, siendo  $N$  el número de componentes de color utilizados como referencia. Cada entrada de la LUT contendría entonces el valor  $\alpha$  correspondiente.

DSIZE estará en la gama de 1 a 9 (inclusive) para la información de referencia de cromaticidad, según el volumen de datos enviados con la CKIF. No se enviará más de una CKIF con una imagen.

El primer octeto que sigue al octeto DSIZE contendrá el orden de representación de la imagen vigente – los trenes que tengan un orden de representación más bajo que se supone que forman la imagen de segundo plano de los trenes con un orden de representación más alto.

Si DSIZE es mayor que uno, el octeto siguiente después del octeto de orden de representación se utilizará para enviar seis bits de bandera definidos como sigue:

- bit 1:  $A_Y$ : Un bit bandera que indica la presencia de un parámetro de referencia  $K_Y$  para valores  $Y$  de luminancia
- bit 2:  $A_B$ : Un bit bandera que indica la presencia de un parámetro de referencia  $K_B$  para valores  $C_B$  de cromaticidad
- bit 3:  $A_R$ : Un bit bandera que indica la presencia de un parámetro de referencia  $K_R$  para valores  $C_R$  de cromaticidad
- bit 4:  $A_1$ : Un bit bandera que indica la presencia de un parámetro de umbral  $T_1$  para transparencia
- bit 5:  $A_2$ : Un bit bandera que indica la presencia de un parámetro de umbral  $T_2$  para opacidad
- bit 6: RPB: Un bit de bandera que indica la utilización de la imagen de referencia como imagen de segundo plano
- bit 7: reservado
- bit 8: reservado

DSIZE será igual a 1 o igual a 2 más el número de bits de bandera entre  $A_Y$ ,  $A_B$  y  $A_R$  que se fijan a 1, más 2 veces el número de bits bandera entre  $A_1$  y  $A_2$  que se fijan a 1. Si DSIZE es mayor que 1, se enviará un octeto adicional para especificar el valor de cada componente de color para cada uno de

los bits bandera  $A_Y$ ,  $A_B$  y  $A_R$  que se fijan a 1 y se enviarán dos octetos adicionales para especificar cada uno de los valores umbral indicados con bandera entre  $T_1$  y  $T_2$ . Estos octetos seguirán el mismo orden que los bits bandera.

Si DSIZE es igual a 1, o si los tres bits bandera del color de referencia  $A_Y$ ,  $A_B$  y  $A_R$  son iguales a cero, los bits bandera de referencia  $A_Y$ ,  $A_B$  y  $A_R$  y los colores de referencia  $K_Y$ ,  $K_B$  y  $K_R$  que fueron utilizados para la imagen referenciada anterior deben utilizarse también para la imagen vigente. Si no se han enviado valores anteriores para la secuencia de vídeo, los bits bandera de referencia por defecto  $A_Y = 1$ ,  $A_B = 1$  y  $A_R = 1$  y los colores de referencia por defecto  $K_Y = 50$ ,  $K_B = 220$  y  $K_R = 100$  deben utilizarse como los valores anteriores.

Si DSIZE es igual a 1, o si ambos bits bandera del umbral de referencia  $A_1$  y  $A_2$  son iguales a cero, los valores umbral de referencia  $T_1$  y  $T_2$  que fueron utilizados para la imagen referenciada anterior deben utilizarse también para la imagen vigente. Si no se han enviado valores anteriores para la secuencia de vídeo, los valores umbral por defecto  $T_1 = 48$  y  $T_2 = 75$  deben utilizarse como los valores anteriores.

En la porción en la que los píxels son "semitransparentes" (es decir, cuando  $T_1 < d < T_2$ ), los píxels decodificados contienen generalmente el color de referencia de cromaticidad en componentes en los que se han utilizado referencias de cromaticidad. Esto puede dar como resultado ciertas distorsiones del color. Para hacer frente a este problema, se pueden ajustar estos valores de píxels antes de combinarlos con el color de segundo plano. Ese proceso de corrección puede aplicarse a los componentes de color que se utilizan en la operación de referencia de cromaticidad indicada por los bits bandera. El proceso es el siguiente:

$$X' = K + (T_2 / d)(X - K)$$

donde  $X$  es el valor del componente de píxel decodificado original, y  $X'$  es el valor corregido.

Dado que los valores de píxels ajustados  $X'_Y$ ,  $X'_B$  y  $X'_R$  son funciones de  $X_Y$ ,  $X_B$  y  $X_R$ , la corrección del color puede efectuarse utilizando una LUT. La LUT tendría  $2^{8N}$  entradas correspondientes a todos los valores de píxels, siendo  $N$  el número de componentes de color utilizados como referencias. Cada entrada contendría entonces los valores corregidos correspondientes.

Si el bit bandera del segundo plano de la imagen de referencia (RPB, *reference picture background*) se fija a "1", ello indica que la imagen de referencia temporalmente previa (anterior a cualquier repetición de muestreo efectuada para la imagen vigente, según el anexo P) debe mantenerse como el segundo plano (opaco) de la imagen vigente y de todas las imágenes referenciadas en cromaticidad posteriores, hasta que sea sustituida por la llegada de otra imagen con una bandera RPB fijada a "1". Si la imagen vigente no tiene ninguna imagen de referencia temporalmente previa (es decir, si la imagen vigente es una imagen del tipo INTRA o EI), la imagen a la que hace referencia el bit bandera RPB es la imagen que habría sido normalmente la imagen de referencia si la imagen vigente fuera del tipo INTER o EP, según proceda. Si el bit bandera RPB se fija a "0", ello indica que el segundo plano debe seguir controlado como lo estaba anteriormente (bajo control externo o bien utilizando una imagen de referencia almacenada previamente al recibir una imagen anterior con un RPB fijado a "1").

La utilización de la referencia de cromaticidad que se invoca emitiendo la función información de referencia de cromaticidad comenzará con la imagen vigente y continuará hasta que aparezca una imagen posterior de tipo INTRA o EI o hasta que expire un periodo de temporización, lo que ocurra primero. La utilización de la referencia de cromaticidad terminará debido a la temporización tras cinco segundos o cinco imágenes, lo que represente una duración mayor. La temporización puede evitarse emitiendo una función información de referencia de cromaticidad idéntica antes de la expiración del periodo de temporización o al expirar el mismo (por ejemplo, se repite la función información de referencia de cromaticidad en el encabezamiento de la primera imagen con una

referencia temporal que indique un intervalo de tiempo mayor o igual a cinco segundos a partir de la emisión, o en el encabezamiento de la quinta imagen después de la emisión). El codificador enviará suficiente información con la función información de referencia de cromaticidad para que se produzca la resincronización completa con cada imagen de tipo INTRA o EI y dentro de cada intervalo de temporización (no dependerá de la utilización de valores almacenados o por defecto de los colores o umbrales de referencia).

### **L.15 Tipo de función ampliada**

La indicación de tipo de función ampliada se utiliza para señalar que el siguiente octeto PSUPP contiene una función ampliada. Las funciones ampliadas se reservan para que la UIT disponga de la capacidad ulterior de definir un gran número de funciones de datos PSUPP compatibles hacia atrás. DSIZE será igual a cero para la indicación de tipo de función ampliada. Para que sea posible la compatibilidad hacia atrás del futuro uso de la indicación de tipo de función ampliada, los decodificadores tratarán la segunda serie de cuatro bits del octeto que sigue a la indicación de tipo de función ampliada como un valor DSIZE que indique el número de octetos subsiguientes de PSUPP de los que se ha de prescindir en los datos de parámetro de la función ampliada, que puede ir seguida de indicaciones de FTYPE adicionales.

## **Anexo M**

### **Modo tramas PB mejoradas**

#### **M.1 Introducción**

Este anexo describe un modo tramas PB mejoradas opcional de la presente Recomendación. Se considera más conveniente utilizar este modo tramas PB mejoradas que el modo tramas PB definido en el anexo G. La capacidad de este modo se señala por medios externos (por ejemplo, aplicando la Rec. UIT-T H.245). El uso de este modo se indica en el campo PLUSPTYPE del encabezamiento de imagen.

Esta opción es, en su mayor parte, similar a la opción modo tramas PB definida en el anexo G.

Para evitar la confusión con las imágenes B definidas en el anexo O, en este anexo no se utilizarán los términos imágenes B, macrobloque B y bloque B. En su lugar, se utilizará la notación  $B_{PB}$  para referirse a la "parte B" de una trama PB mejorada. Cuando se haga referencia al anexo G, la imagen B y el bloque B serán la imagen  $B_{PB}$  y el bloque  $B_{PB}$ .

La principal diferencia entre el modo tramas PB y el modo tramas PB mejoradas es que en este último, el macrobloque  $B_{PB}$  dispone de un modo de predicción hacia adelante y hacia atrás además del modo de predicción bidireccional. En este anexo, MVDB (cuando está presente) designa un vector de movimiento hacia adelante. (Se señala que en el anexo G, se utilizó MVDB para designar una mejora de los vectores hacia adelante y hacia atrás escalados en sentido descendente para la predicción bidireccional en vez de un vector de movimiento hacia adelante específico.)

En este anexo se identifican todas las diferencias con respecto al anexo G. Cuando no hay ninguna indicación, ello significa que se utiliza el mismo procedimiento que se describe en el anexo G.

#### **M.2 Modos de predicción del macrobloque $B_{PB}$**

Hay tres maneras diferentes de codificar un macrobloque  $B_{PB}$ . Los distintos modos de codificación son señalizados por el parámetro MODB. Los modos de codificación del macrobloque  $B_{PB}$  son los siguientes:

### **M.2.1 Predicción bidireccional**

En el modo de predicción bidireccional, la predicción utiliza las imágenes de referencia antes y después de la imagen  $B_{PB}$  (en el caso de una secuencia de tramas PB mejoradas, esto significa la parte de la imagen P de la trama PB mejorada temporalmente previa y la parte de la imagen P de la trama PB mejorada vigente). Esta predicción es equivalente a la predicción definida en el anexo G cuando  $MV_D = 0$ . Cabe indicar que en este modo (y sólo en este modo) los datos del vector de movimiento (MVD) del macrobloque PB deben incluirse incluso si el macrobloque P está codificado INTRA. (Obsérvese la diferencia entre los datos del vector de movimiento MVD y el vector delta  $MV_D$  definido en el anexo G.)

### **M.2.2 Predicción hacia adelante**

En el modo predicción hacia adelante los datos de vectores contenidos en MVDB se utilizan para la predicción hacia adelante a partir de la imagen de referencia anterior (una imagen INTRA o INTER, o la parte de la imagen P de una trama PB o PB mejorada). Esto significa que siempre hay únicamente un vector  $16 \times 16$  para el macrobloque  $B_{PB}$  en este modo de predicción.

Para codificar el vector de movimiento hacia adelante se utiliza un predictor simple. La regla de este predictor consiste en que si el macrobloque vigente no está en el borde situado más a la izquierda de la imagen o rebanada y el macrobloque de la izquierda tiene un vector de movimiento hacia adelante, el predictor del vector de movimiento hacia adelante para el macrobloque vigente se fija en el valor del vector de movimiento hacia adelante del bloque de la izquierda; de lo contrario, el predictor se fija en cero. La diferencia entre el predictor y el vector de movimiento deseado se codifica entonces con la tabla VLC de la misma manera que los datos del vector que se han de utilizar para la imagen P (MVD).

Con respecto a los vectores de movimiento en fronteras de imagen definidos en D.1, la técnica descrita se aplica también para el vector  $B_{PB}$  hacia adelante si se utiliza esta característica (esto es aplicable tanto al modo de predicción hacia adelante como al modo de predicción bidireccional).

### **M.2.3 Predicción hacia atrás**

En el modo predicción hacia atrás, la predicción del macrobloque  $B_{PB}$  es idéntica a la de  $P_{REC}$  (definida en G.5). Para este modo de predicción no se utiliza ningún dato de vector de movimiento.

## **M.3 Cálculo de vectores para la predicción bidireccional de un macrobloque B**

Cuando se utiliza la predicción bidireccional, los vectores hacia adelante y hacia atrás escalados se calculan tal como se describe en el anexo G para el caso en que  $MV_D = 0$ .

## **M.4 Cuadro MODB**

En el cuadro M.1 se muestra una nueva definición para MODB (que sustituye al cuadro 11). En él se indican los posibles modos de codificación para un bloque  $B_{PB}$ .



**Cuadro M.1/H.263 – Tabla de MODB para el modo tramas PB mejoradas**

Índice	CBPB	MVDB	Número de bits	Código	Modo de codificación
0			1	0	Predicción bidireccional
1	x		2	10	Predicción bidireccional
2		x	3	110	Predicción hacia adelante
3	x	x	4	1110	Predicción hacia adelante
4			5	11110	Predicción hacia atrás
5	x		5	11111	Predicción hacia atrás

NOTA – El símbolo "x" indica que está presente el elemento de sintaxis asociado.

## Anexo N

### Modo selección de la imagen de referencia

#### N.1 Introducción

Este anexo describe el modo selección de imágenes de referencia opcional de la presente Recomendación, que funciona utilizando un método de predicción intertrama modificado llamado "NEWPRED". La capacidad de este modo es señalado por medios externos (por ejemplo, aplicando la Rec. UIT-T H.245). El volumen de la memoria de imágenes adicional instalada en el decodificador puede señalizarse también por medios externos para ayudar a la gestión de la memoria en el codificador. Este modo puede utilizar mensajes de canal de retorno enviados desde un decodificador a un codificador para informar a éste qué parte de qué imágenes se han decodificado correctamente en el decodificador. La utilización de este modo se indica en el campo PLUSPTYPE del encabezamiento de imagen. Este modo tiene dos conmutadores de modo de canal de retorno que definen si se utiliza un canal de retorno y qué clase de mensajes son devueltos por ese canal desde el decodificador, y tiene otro submodo definido en función del canal para los mensajes de canal de retorno.

Los dos conmutadores del modo canal de retorno de este modo determinan el tipo de mensajes enviados por el canal de retorno, y especifican si se envían mensajes de acuse de recibo (ACK, *acknowledgment messages*) o mensajes de acuse de recibo negativo (NACK, *non-acknowledgment messages*). En conjunto, ambos conmutadores definen cuatro métodos básicos de funcionamiento:

- 1) NEITHER (Ninguno de los dos): en el que el decodificador no devuelve ningún dato de canal de retorno al codificador;
- 2) ACK: en el que el decodificador devuelve únicamente mensajes de acuse de recibo;
- 3) NACK: en el que el decodificador devuelve únicamente mensajes de no acuse de recibo; y
- 4) ACK+NACK: en el que el decodificador devuelve mensajes de acuse de recibo y de acuse de recibo negativo.

El tipo específico de mensajes que se envían tal como arriba se indica se señala en el encabezamiento de la imagen.

Hay además dos métodos de funcionamiento definidos en términos del canal para los mensajes de canal de retorno:

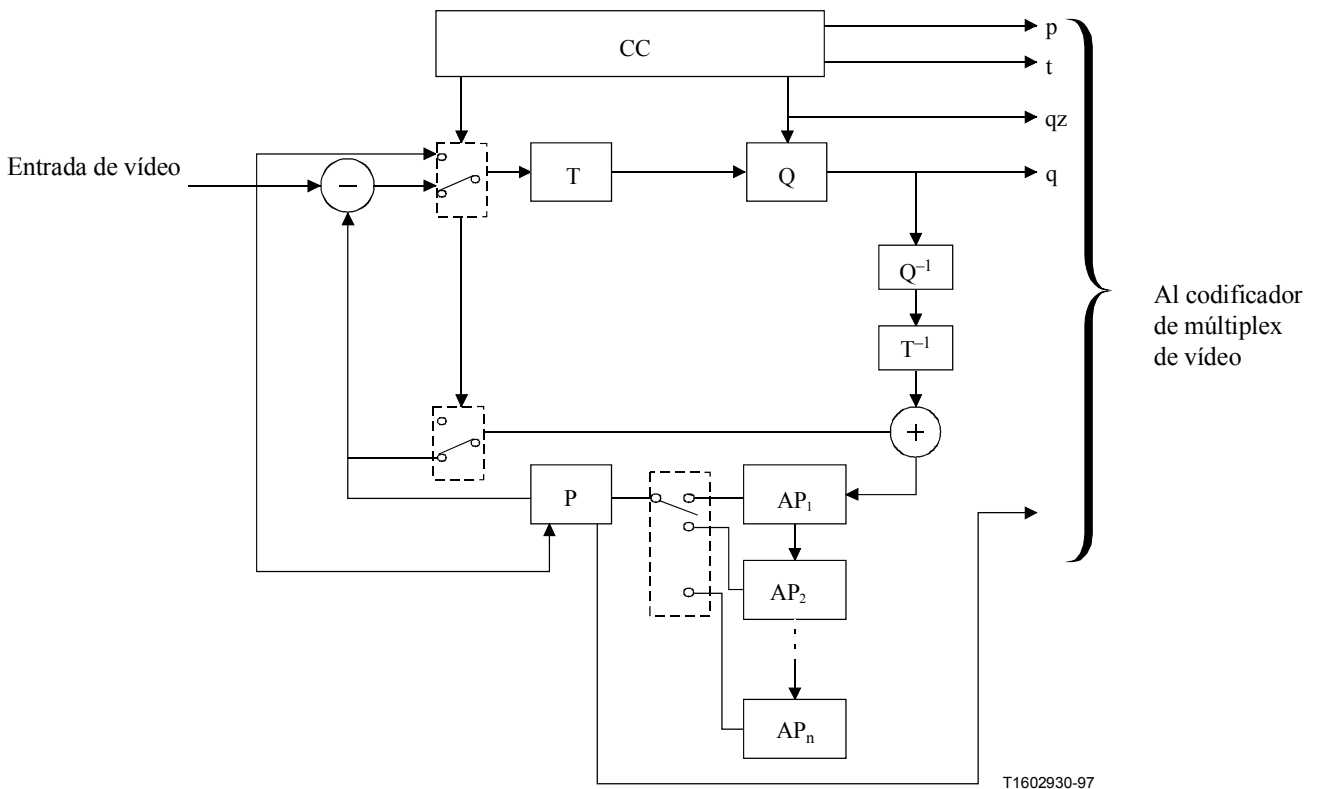
- 1) modo canal lógico separado: este método de funcionamiento entrega datos del canal de retorno a través de un canal lógico separado de la capa múltiplex del sistema; y

- 2) modo múltiplex de vídeo: este método de funcionamiento entrega datos de canal de retorno para vídeo recibido en el marco de los datos de vídeo hacia adelante de un tren de vídeo de datos codificados.

En este anexo se especifica una sintaxis para los mensajes y datos de canal de retorno.

## N.2 Algoritmo de codificación de fuente vídeo

El codificador de fuente de este modo se muestra de forma generalizada en la figura N.1. En esta figura se observa una estructura con un cierto número de memorias de imagen. El codificador de fuente puede seleccionar una de las memorias de imagen para eliminar la propagación temporal de errores debidos a la codificación intertrama. El modo decodificación de segmento independiente (véase el anexo R), que trata los límites de los grupos de bloques (GOB) con encabezamientos o rebanadas no vacíos como fronteras de imagen, puede utilizarse para evitar la propagación de errores debidos a la compensación de movimiento por los límites de los GOB o las rebanadas cuando este modo se aplica a una unidad más pequeña que una imagen, tales como un GOB o una rebanada. La información para señalar cuál es la imagen seleccionada para la predicción se incluye en el tren de bits codificado. La estrategia utilizada por el codificador para seleccionar la imagen que debe utilizarse para la predicción queda fuera del alcance de esta Recomendación.



T	Transformada
Q	Cuantificador
P	Memoria de imagen con retardo variable y compensación de movimiento
AP	Memoria de imagen adicional
CC	Control de codificación
p	Bandera de INTRA/INTER
t	Bandera de transmitido o no transmitido
qz	Indicación de cuantificador
q	Índice de cuantificación para los coeficientes de la transformada
v	Vector de movimiento

**Figura N.1/H.263 – Codificador de fuente para NEWPRED**

### N.3 Canal para mensajes de canal de retorno

Este modo tiene dos métodos de funcionamiento desde el punto de vista del tipo de canal para mensajes de canal de retorno. Uno de ellos es el modo canal lógico separado y el otro, el modo múltiplex de vídeo. El modo canal lógico separado es un modo preferencial y entrega el mensaje de canal de retorno definido en N.4.2 a través del canal lógico especializado. El modo múltiplex de vídeo está previsto para aquel sistema que no pueda establecer un canal suplementario para los mensajes de canal de retorno debido a la restricción del número de combinaciones de canales. El modo múltiplex de vídeo entrega los mensajes de canal de retorno a través del mismo canal lógico con los datos vídeos hacia adelante en el sentido opuesto.

#### N.3.1 Modo canal lógico separado

El modo canal lógico separado entrega mensajes de canal de retorno a través de un canal lógico especializado abierto únicamente a efectos de transmisión de mensajes de canal de retorno. El mecanismo de asociación con el canal hacia adelante que entrega datos de vídeo es proporcionado por medios externos (por ejemplo, aplicando la Rec. UIT-T H.245). El funcionamiento del canal lógico separado necesita un mecanismo de puesta en trama externo para la sincronización de los mensajes en el canal de retorno, dado que la sintaxis del canal de retorno que aquí se define no contiene palabras bandera de sincronización.

#### N.3.2 Modo múltiplex de vídeo

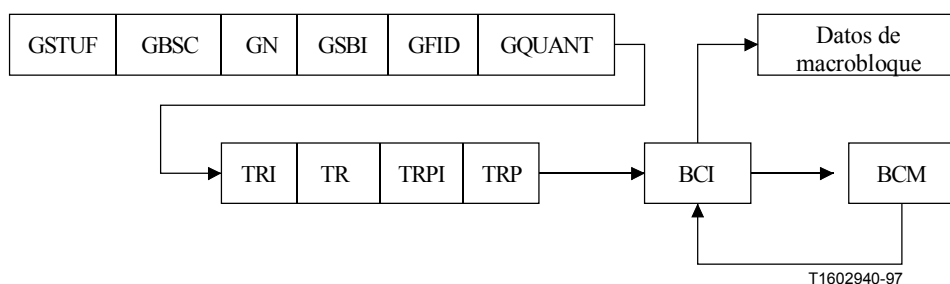
El modo múltiplex de vídeo entrega los mensajes de canal de retorno a través del mismo canal lógico con los datos de vídeo hacia adelante en el sentido opuesto. En N.4.1 se describe la sintaxis del tren de bits multiplexado. Los mensajes de canal de retorno pueden ser insertados utilizando la indicación de mensaje de canal de retorno (BCI) en el GOB o el encabezamiento de rebanada.

### N.4 Sintaxis

#### N.4.1 Canal hacia adelante

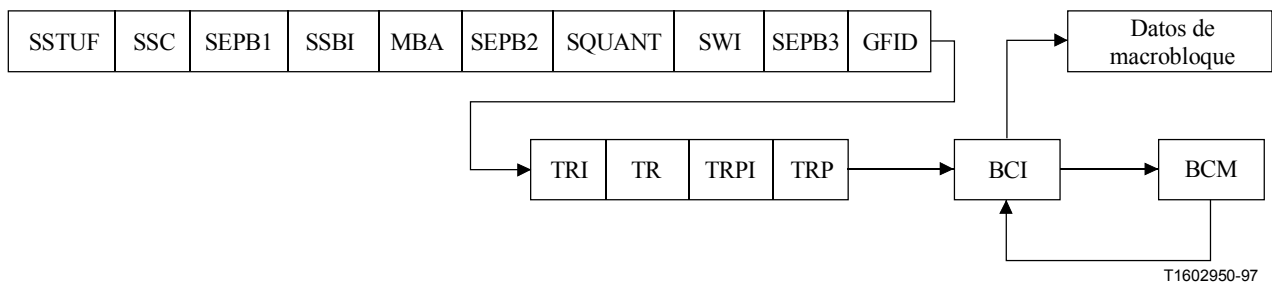
La sintaxis para los datos de canal hacia adelante que llevan la señal de vídeo comprimida sólo se altera en el grupo de bloques o en la capa de rebanada.

La figura N.2 muestra la sintaxis de la capa de GOB. Los campos de TRI, TR, TRPI, TRP, BCI y BCM se añaden a la figura 9.



**Figura N.2/H.263 – Estructura de la capa de GOB para NEWPRED**

Cuando se utiliza el modo estructura en rebanada opcional (véase el anexo K), la sintaxis de la capa de rebanada se modifica en la misma manera que la capa de GOB. En la figura N.3 se muestra la sintaxis.



**Figura N.3/H.263 – Estructura de la capa de rebanada para NEWPRED**

#### **N.4.1.1 Indicador de referencia temporal (TRI, *temporal reference indicator*) (1 bit)**

TRI indica si está presente o no el campo TR.

0: El campo TR no está presente.

1: El campo TR está presente.

#### **N.4.1.2 Referencia temporal (TR) (8/10 bits)**

Cuando está presente, TR es un número de 8 bits a menos que se utilice una frecuencia de reloj de imagen personalizada, en cuyo caso es un número de 10 bits formado por la concatenación de ETR y TR del encabezamiento de imagen.

#### **N.4.1.3 Referencia temporal para el indicador de predicción (TRPI) (1 bit)**

TRPI indica si está presente o no el campo TRP.

0: El campo TRP no está presente.

1: El campo TRP está presente.

TRPI será igual a cero cuando la imagen sea una imagen I o EI.

#### **N.4.1.4 Referencia temporal para predicción (TRP) (10 bits)**

Cuando está presente (tal como se indica en TRPI), TRP indica la referencia temporal que se utiliza para la predicción de la codificación, excepto en el caso de imágenes B y de la parte de imagen B de una trama PB mejorada. Para las imágenes B o la parte de imagen B de una trama PB mejorada, la figura que tiene la referencia temporal TRP se utiliza para la predicción en el sentido hacia adelante. (La predicción en el sentido temporal inverso utiliza siempre la imagen inmediata temporalmente subsiguiente.) TRP es un número de 10 bits. Si no se utilizó una frecuencia de reloj de imagen personalizada, para la imagen de referencia, los dos bits más significativos (MSB) de la TR son 0 y los bits menos significativos (LSB) contienen la TRP de 8 bits hallada en el encabezamiento de imagen de la imagen de referencia. Si se utilizó una frecuencia de reloj de imagen personalizada para la imagen de referencia, TRP es un número de 10 bits formado por la concatenación de ETR y TR a partir del encabezamiento de la imagen de referencia.

Cuando TRP no esté presente, se utilizará la imagen de anclaje temporalmente previa más reciente, como cuando no se está en el modo selección de imagen de referencia. TRP es válido hasta el próximo PSC, GSC o SSC.

#### **N.4.1.5 Indicación de mensaje de canal de retorno (BCI) (Longitud variable)**

Este campo contiene uno o dos bits. Cuando está fijado a "1", señala la presencia del campo mensaje de canal de retorno (BCM, *back-channel message*) de vídeo siguiente. De lo contrario, el valor del campo es "01", lo que indica la ausencia o el final del campo mensaje de canal de retorno de vídeo. Las combinaciones de BCM y BCI pueden no estar presentes, y pueden repetirse cuando están presentes. BCI se fijará siempre a "01" si no se utiliza el modo múltiple de vídeo.

#### N.4.1.6 Mensaje de canal de retorno (BCM) (Longitud variable)

El mensaje de canal de retorno con la sintaxis definida en N.4.2 sólo está presente si el campo BCI precedente está fijado a "1".

#### N.4.2 Sintaxis del mensaje de canal de retorno (BCM)

En la figura N.4 se ilustra la sintaxis para el canal de retorno que lleva los mensajes de acuse de recibo/acuse de recibo negativo. Este mensaje es devuelto de un decodificador a un codificador para indicar si los datos de un canal hacia adelante fueron decodificados correctamente o no.

BT	URF	TR	ELNUMI	ELNUM	BCPM	BSBI	BEPB1	GN/MBA	BEPB2	RTR	BSTUF
----	-----	----	--------	-------	------	------	-------	--------	-------	-----	-------

**Figura N.4/H.263 – Estructura de la sintaxis del mensaje de canal de retorno (BCM) para NEWPRED**

##### N.4.2.1 Tipo de mensaje de canal de retorno (BT, *back-channel message type*) (2 bits)

El tipo de mensaje de canal de retorno indica si la parte correspondiente del mensaje codificado está correctamente decodificada o no. En el encabezamiento de la imagen del canal hacia adelante se indica qué tipo de mensaje es necesario para el codificador.

00: Reservado para uso futuro.

01: Reservado para uso futuro.

10: NACK. Indica la decodificación errónea de la parte correspondiente de los datos del canal hacia adelante.

11: ACK. Indica la decodificación correcta de la parte correspondiente de los datos del canal hacia adelante.

##### N.4.2.2 Bandera de no fiable (URF, *unreliable flag*) (1bit)

La bandera de no fiable se fija a 1 cuando en el decodificador no se dispone de un valor fiable para TR o GN/MBA. (Cuando BT es NACK, puede no estar disponible en el decodificador una TR fiable.)

0: Fiable.

1: No fiable.

##### N.4.2.3 Referencia temporal (TR) (10 bits)

La referencia temporal contiene la información TR del segmento de imagen de vídeo para el que está indicado ACK/NACK en el mensaje de canal de retorno.

NOTA – El significado del término "segmento de imagen de vídeo" tal como aquí se utiliza se define en el anexo R. Si no se utilizó una frecuencia de reloj de imagen personalizada para la imagen de referencia, los dos MSB de la TR son cero y los LSB contienen la TR de 8 bits hallada en el encabezamiento de imagen de la imagen de referencia. Si se utilizó una frecuencia de reloj de imagen personalizada para la imagen de referencia, TR es un número de 10 bits formado por la concatenación de ETR y TR a partir del encabezamiento de la imagen de referencia.

##### N.4.2.4 Indicación de número de capa de mejora (ELNUMI, *enhancement layer number indication*) (1 bit)

La indicación de número de capa de mejora es "0" a menos que se utilice el modo escalabilidad temporal, SNR y espacial (anexo O) optativo en los datos del canal hacia adelante, se combinen algunas capas de mejora del canal hacia adelante por un canal lógico y el mensaje de canal de retorno designe una capa de mejora (y no la capa básica), en cuyo caso la indicación de número de capa de mejora será "1".

#### **N.4.2.5 Número de capa de mejora (ELNUM) (4 bits)**

El número de capa de mejora sólo está presente si ELNUMI es "1", en cuyo caso contiene el número de capa de la capa de mejora a la que se hace referencia en el mensaje de canal de retorno.

#### **N.4.2.6 BCPM (1 bit)**

BCPM es "0" a menos que se utilice el modo CPM en los datos del canal hacia adelante, en cuyo caso es "1". Si BCPM es "1", indica que el BSBI está presente.

#### **N.4.2.7 Indicador de subtren de bits de canal de retorno (BSBI, *back-channel sub-bitstream indicator*) (2 bits)**

Palabra de código de 2 bits de longitud fija que sólo está presente si BCPM es "1". El BSBI es la representación binaria natural del número de subtren de bits adecuado en los datos del canal hacia adelante para el que se indica el mensaje ACK/NACK en el mensaje de canal de retorno, tal como se describe en 5.2.4 y en el anexo C.

#### **N.4.2.8 Bit 1 de prevención de emulación de canal de retorno (BEPB1, *back-channel emulation prevention bit 1*) (1 bit)**

Campo que sólo está presente si se utiliza el modo múltiplex de vídeo. Este campo se fija siempre a "1" para evitar la emulación del código de comienzo.

#### **N.4.2.9 Número de GOB/dirección de macrobloque (GN/MBA, *GOB number/macroblock address*) (5/6/7/9/11/12/13/14 bits)**

En este campo está presente un número de GOB o dirección de macrobloque. Si no se utiliza el modo estructura en rebanada opcional (véase el anexo K), este campo contiene el número de GOB de comienzo del segmento de la imagen de vídeo para el que se indica el mensaje NACK/ACK en el mensaje de canal de retorno. Si se utiliza el modo estructura en rebanada opcional, este campo contiene la dirección del macrobloque de comienzo de rebanada para el que se indica el mensaje NACK/ACK en el mensaje de canal de retorno. La longitud de este campo es la longitud que se especifica en toda esta Recomendación para GN o MBA.

NOTA – Si se recibe este campo en el modo múltiplex de vídeo, el empleo del modo facultativo estructuración en rebanada, indica el empleo de este modo en el tren de bits de vídeo al que se aplica el BCM y no en el tren de bits de vídeo que transporta los datos del BCM.

#### **N.4.2.10 Bit 2 de prevención de emulación de canal de retorno (BEPB2, *back-channel emulation prevention bit 2*) (1 bit)**

Campo que sólo está presente si se utiliza el modo múltiplex de vídeo. Este campo se fija siempre a "1" para evitar la emulación del código de comienzo.

#### **N.4.2.11 Referencia temporal solicitada (RTR, *requested temporal reference*) (10 bits)**

La referencia temporal solicitada sólo está presente si BT es NACK. RTR indica la referencia temporal solicitada del GOB o rebanada asociada con el NACK. Generalmente es la TR del último segmento de imagen de vídeo decodificado correctamente de la posición correspondiente en el decodificador. Si no se utilizó una frecuencia de reloj de imagen personalizada para la imagen de referencia solicitada, los dos MSB de la RTR son cero y los LSB contienen la TR de 8 bits hallada en el encabezamiento de imagen de la imagen de referencia solicitada. Si se utilizó una frecuencia de reloj de imagen personalizada para la imagen de referencia solicitada, RTR es un número de 10 bits formado por la concatenación de ETR y TR a partir del encabezamiento de la imagen de referencia solicitada.

#### **N.4.2.12 Relleno (BSTUF, *stuffing*) (Longitud variable)**

Este campo sólo está presente si se utiliza el modo canal lógico separado y el mensaje de canal de retorno es el último en una trama externa. BSTUF consiste en una palabra de código de longitud

variable formada por cero o más bits de valor "0". Este campo sólo está presente al final de una trama externa.

## **N.5 Proceso del decodificador**

El decodificador de este modo puede necesitar un número adicional de memorias de imagen para almacenar las señales de vídeo decodificadas correctamente y su información de referencia temporal (TR). El decodificador utiliza la imagen almacenada para la que la TR es la TRP como la imagen de referencia para la decodificación intertrama en lugar de la última imagen decodificada, si el campo TRP existe en los datos del canal hacia adelante. Cuando la imagen para la que la TR es la TRP no está disponible en el decodificador, éste puede enviar una señal de actualización INTRA forzada al codificador por medios externos (por ejemplo, aplicando la Rec. UIT-T H.245). A menos que se negocie una política de almacenamiento de trama distinta por medios externos, los segmentos de imágenes de vídeo decodificados correctamente se almacenarán en la memoria para ser utilizados como últimas imágenes de referencia sobre la base "primera en entrar, primera en salir", como se indica en la figura N.1 (excepto en el caso de las imágenes B, que no se utilizan como imágenes de referencia), y los segmentos de imágenes de vídeo de los que se detecte que han sido decodificados incorrectamente no deben sustituir a los decodificados correctamente en esta zona de memoria.

Un mensaje de acuse de recibo (ACK, *acknowledgment message*) y un mensaje de acuse de recibo negativo (NACK, *non-acknowledgment message*) se definen como mensajes de canal de retorno. Un ACK puede ser devuelto cuando el decodificador decodifica un segmento de imagen de vídeo de manera satisfactoria. Los NACK pueden ser devueltos cuando el decodificador no logra decodificar un segmento de imagen de vídeo, y pueden seguir siendo devueltos hasta que el decodificador obtiene los datos del canal hacia adelante previstos que incluyen la TRP solicitada o una actualización INTRA. En el campo RPSMF del encabezamiento de imagen de los datos del canal hacia adelante se indica cuáles son los tipos de mensaje que se enviarán.

En un escenario de utilización conocido como "codificación de redundancia de vídeo", el modo selección de la imagen de referencia puede ser utilizado por algunos codificadores de forma que se envíe más de una representación para la escena de la imagen en el mismo instante temporal (utilizando generalmente distintas imágenes de referencia). En un caso como éste en el que se utiliza el modo selección de la imagen de referencia y en el que las imágenes adyacentes en el tren de bits tienen la misma referencia temporal, el decodificador considerará tal ocurrencia como una indicación de que se han enviado copias redundantes de aproximadamente el mismo contenido de la escena imaginizada, y decodificará y utilizará la primera de esas imágenes recibidas al mismo tiempo que descarta la imagen o imágenes redundantes subsiguientes.

## **Anexo O**

### **Modo escalabilidad temporal, SNR y espacial**

Este anexo describe el modo opcional de la presente Recomendación en apoyo de la escalabilidad temporal, SNR y espacial. Este modo puede utilizarse también junto con esquemas de control de error. La capacidad de este modo y el grado en que se admiten sus características se señalizan por medios externos (por ejemplo, aplicando la Rec. UIT-T H.245). La utilización de este modo se indica en PLUSPTYPE.

#### **O.1 Visión general**

La escalabilidad permite la decodificación de una secuencia con más de un nivel de calidad. Esto se efectúa utilizando una jerarquía de imágenes e imágenes de mejora divididas en una o más capas.

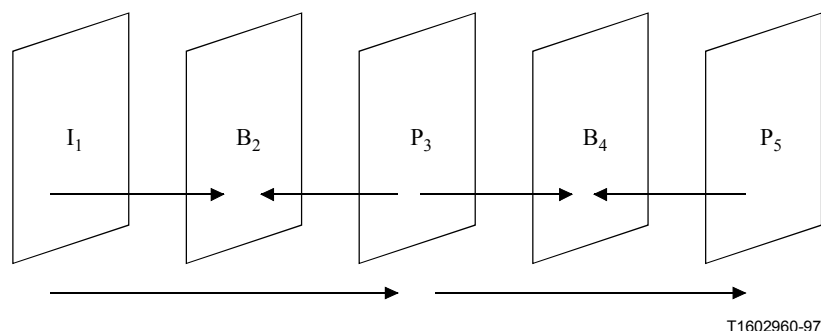
En la escalabilidad se utilizan tres tipos de imágenes utilizadas: imágenes B, imágenes EI, e imágenes EP, como se explica más adelante. Cada imagen tiene un número de capa de mejora ELNUM que indica a qué capa pertenece, y un número de capa de referencia RLNUM que indica qué capa se utiliza para su predicción. La capa más baja se llama capa básica y tiene el número de capa 1.

La escalabilidad se logra mediante tres métodos básicos: temporal, SNR y de mejora espacial.

### O.1.1 Escalabilidad temporal

La escalabilidad temporal se consigue utilizando imágenes predichas bidireccionalmente, o imágenes B. Las imágenes B permiten la predicción a partir de una imagen reconstruida anterior o posterior o a partir de ambas en la capa de referencia. Esta propiedad suele dar como resultado una mejor eficacia de la compresión en comparación con la de las imágenes P. Las imágenes B difieren de la parte de imagen B de una trama PB (o trama PB mejorada) (véanse los anexos G y M) en el sentido de que son entidades separadas en el tren de bits: no se combinan entre sí sintácticamente con una imagen P (o EP) posterior.

Las imágenes B (y la parte B de una trama PB o trama PB mejorada) no se utilizan como imágenes de referencia para la predicción de ninguna otra imagen. Esta propiedad permite que las imágenes B sean descartadas si es necesario sin afectar de manera adversa a ninguna imagen subsiguiente, proporcionando así la escalabilidad temporal. La figura O.1 ilustra la estructura predictiva de las imágenes P y B.



**Figura O.1/H.263 – Ilustración de las dependencias de la predicción de las imágenes B**

La localización de las imágenes B en el tren de bits sigue un orden en dependencia de los datos en vez de un orden temporal estricto. (Esta regla es coherente con el ordenamiento de otras imágenes en el tren de bits, pero con todos los demás tipos de imágenes distintas de las imágenes B no se plantea esa contraposición entre el orden en dependencia de los datos y el orden temporal.) Por ejemplo, si las imágenes de una secuencia de vídeo se numerasen 1, 2, 3, ..., el orden del tren de bits de las imágenes codificadas sería I<sub>1</sub>, P<sub>3</sub>, B<sub>2</sub>, P<sub>5</sub>, B<sub>4</sub>, ..., en los subíndices se refieren al número de imagen original (como se ilustra en la figura O.1).

No hay un límite al número de imágenes B que pueden insertarse entre pares de imágenes de referencia en la capa de referencia (salvo el que se necesita para evitar la ambigüedad temporal a causa de los desbordamientos del campo de referencia temporal en el encabezamiento de la imagen). No obstante, un número máximo de esas imágenes puede ser señalado por medios externos (por ejemplo, aplicando la Rec. UIT-T H.245).

La altura, anchura y relación de aspecto de píxel de una imagen B serán siempre iguales a las de su imagen de capa de referencia temporalmente subsiguiente.

Los vectores de movimiento pueden extenderse más allá de las fronteras de las imágenes B.



### O.1.2 Escalabilidad SNR

El otro método básico para lograr la escalabilidad es a través de la mejora espacial/SNR. La escalabilidad espacial y la escalabilidad SNR son equivalentes excepto por lo que se refiere a la utilización de la interpolación, que se describe brevemente. Dado que la compresión introduce perturbaciones y distorsiones, la diferencia entre una imagen reconstruida y su original en el codificador es (casi siempre) una imagen con un valor distinto de cero, que contiene lo que puede llamarse el error de codificación. Normalmente, el error de codificación se pierde en el codificador y nunca se recupera. Con la escalabilidad SNR, estas imágenes con error de codificación se pueden también codificar y enviar al decodificador, mejorando la imagen decodificada. Los datos adicionales sirven para incrementar la relación señal/ruido de la imagen de vídeo y, por lo tanto, el término escalabilidad SNR. En la figura O.2 se muestra el flujo de datos para la escalabilidad SNR. Las flechas verticales de la capa más baja ilustran la predicción de la imagen en la capa de mejora a partir de una aproximación reconstruida de esa imagen en la capa (más baja) de referencia.

Si la predicción se forma únicamente a partir de la capa más baja, la imagen de la capa de mejora se designa como imagen EI. Es posible, sin embargo, crear una imagen predicha modificada bidireccionalmente utilizando una imagen de capa de mejora anterior y una imagen de referencia de capa más baja temporalmente simultánea. Este tipo de imagen se designa como imagen EP o imagen P "de mejora". En la figura O.2 se muestra el flujo de predicción de imágenes EI y EP. (Aunque no se muestra específicamente en la figura O.2, una imagen EI en una capa de mejora puede tener una imagen P como imagen de referencia de capa más baja, y una imagen EP puede tener una imagen I como imagen de mejora de capa más baja.)

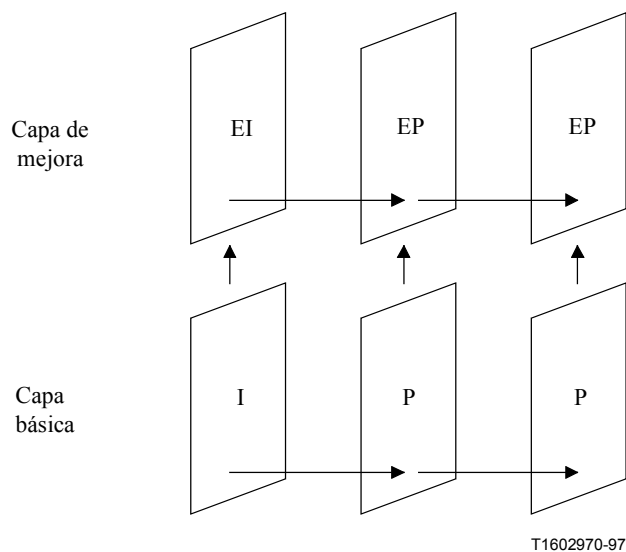


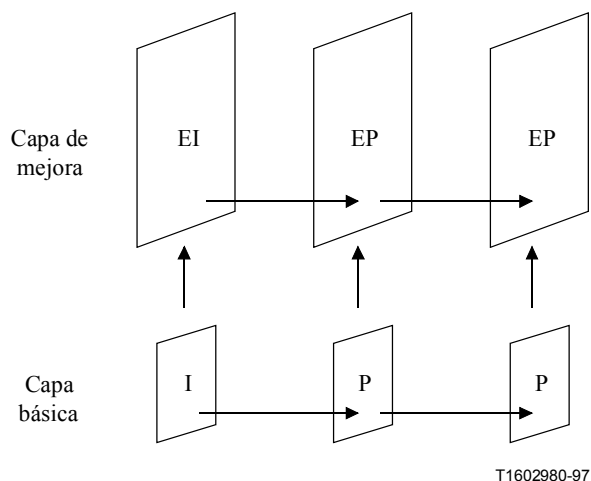
Figura O.2/H.263 – Ilustración de la escalabilidad SNR

Tanto para las imágenes EI como para las imágenes EP, la predicción a partir de la capa de referencia no utiliza vectores de movimiento. No obstante, al igual que con las imágenes P normales, las imágenes EP utilizan vectores de movimiento cuando se efectúa la predicción a partir de sus imágenes de referencia temporalmente anteriores en la misma capa.

### O.1.3 Escalabilidad espacial

El tercer y último método de escalabilidad del modo escalabilidad temporal, SNR y espacial es la escalabilidad espacial, que está estrechamente relacionada con la escalabilidad SNR. La única diferencia es que antes de utilizar la imagen de la capa de referencia para efectuar la predicción de la imagen de la capa de mejora espacial, es interpolada por un factor de dos ya sea horizontal o verticalmente (escalabilidad espacial 1-D), o tanto horizontal como verticalmente (escalabilidad espacial 2-D). En O.6 se definen los filtros de interpolación para esta operación. Para que un

decodificador pueda efectuar algunas formas de escalabilidad espacial, quizá sea necesario que acepte también formatos de imagen personalizados. Por ejemplo, si la capa básica es sub-cuarto de CIF ( $128 \times 96$ ), la imagen de la capa de mejora espacial 2-D podría ser  $256 \times 192$ , lo que no corresponde a un formato de imagen normalizado. Otro ejemplo sería el de una capa básica cuarto de CIF ( $176 \times 144$ ), con la relación de aspecto de píxel normalizado de 12:11. Una capa de mejora espacial horizontal 1-D correspondería entonces a un formato de imagen de  $352 \times 144$  con una relación de aspecto de píxel de 6:11. En estos casos tendría que utilizarse, por tanto, un formato de imagen personalizado para la capa de mejora. Un caso en el que no se necesitaría formato de imagen personalizado sería aquel en que se utilizara una capa básica cuarto de CIF con una capa de mejora espacial 2-D CIF. En la figura O.3 se ilustra la escalabilidad espacial.



**Figura O.3/H.263 – Ilustración de la escalabilidad espacial**

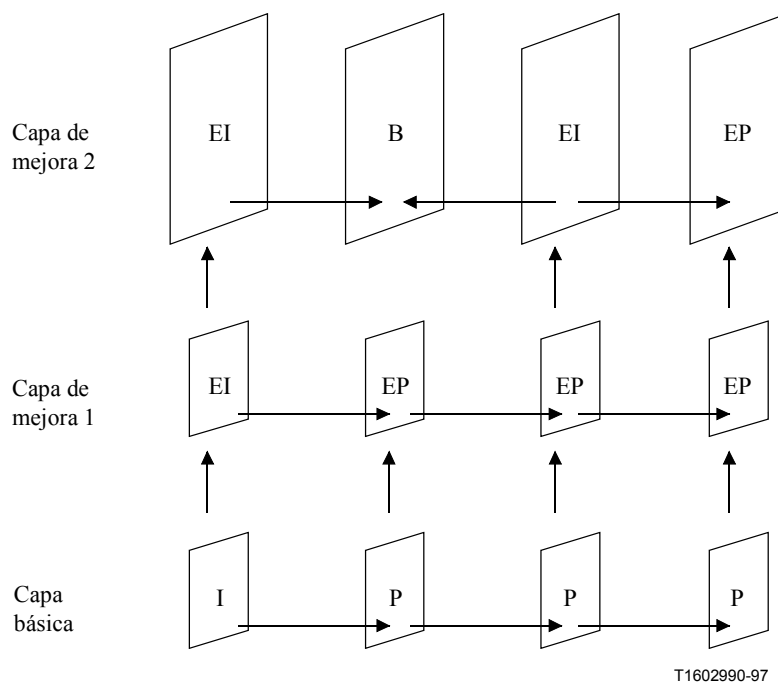
Aparte de requerir un proceso de muestreo a velocidad superior para incrementar el tamaño de la imagen de la capa de referencia antes de utilizarla como referencia en el proceso de codificación, el procedimiento y la sintaxis de una imagen de escalabilidad espacial son funcionalmente idénticos al de una imagen de escalabilidad SNR.

Puesto que hay muy pocas distinciones sintácticas entre imágenes con escalabilidad SNR e imágenes con escalabilidad espacial, las imágenes utilizadas con uno u otro fin se denominan imágenes EI e imágenes EP.

La imagen de la capa básica que se utiliza para la predicción hacia arriba en una imagen EI o EP puede ser una imagen I, una imagen P, o la parte P de una trama PB o PB mejorada (pero no una imagen B ni la parte B de una trama PB o PB mejorada).

#### **O.1.4 Escalabilidad multicapa**

Es posible insertar imágenes B temporalmente no sólo entre imágenes de tipo I, P, PB y PB mejorada, sino también entre imágenes de tipo EI y EP, tanto si éstas consisten en imágenes SNR como de mejora espacial. Es posible también tener más de una capa de SNR o de mejora espacial junto con una capa básica. Por ello, un tren de bits escalable multicapa puede ser una combinación de capas SNR, capas espaciales e imágenes B. No obstante, el tamaño de una imagen no puede disminuir al aumentar el número de capa. Sólo puede mantenerse igual o aumentar con un factor de dos en una o ambas dimensiones. En la figura O.4 se ilustra un tren de bits escalable multicapa.



**Figura O.4/H.263 – Ilustración de la escalabilidad multicapa**

En el caso de la escalabilidad multicapa, la imagen de una capa de referencia que se utiliza para la predicción hacia arriba en una imagen EI o EP puede ser una imagen I, P, EI o EP, o puede ser la parte P de una trama PB o PB mejorada de la capa básica (pero no una imagen B ni la parte B de una trama PB o PB mejorada).

Como en el caso de dos capas, las imágenes B pueden aparecer en cualquier capa. Sin embargo, cualquier imagen de una capa de mejora que es temporalmente simultánea con una imagen B de su capa de referencia debe ser una imagen B o la parte de imagen B de una trama PB o PB mejorada. Se trata así de preservar el carácter disponible de las imágenes B. Cabe indicar, no obstante, que las imágenes B pueden aparecer en capas que no tienen ninguna imagen correspondiente en capas más bajas. Esto permite al codificador enviar imágenes vídeo mejoradas con una velocidad de imagen más alta que las capas más bajas.

El número de la capa de mejora y el número de la capa de referencia de cada imagen de mejora (B, EI o EP) se indican en los campos ELNUM y RLNUM, respectivamente, del encabezamiento de la imagen (cuando están presentes). Véanse las reglas de inferencia descritas en 5.1.4.4 para el caso en que estos campos no están presentes. Si una imagen B aparece en una capa de mejora en la que también aparecen imágenes con escalabilidad SNR o espacial temporalmente adyacentes, el número de la capa de referencia (RLNUM) de la imagen B será el mismo que el número de la capa de mejora (ELNUM).

La altura, anchura y relación de aspecto de píxel de una imagen B serán siempre iguales a las de su imagen de capa de referencia temporalmente subsiguiente.

## **O.2 Orden de transmisión de las imágenes**

Las imágenes que dependen de otras imágenes se colocarán en el tren de bits después de las imágenes de las cuales dependen.

El orden de sintaxis del tren de bits se especifica de manera que para las imágenes de referencia (es decir, imágenes de tipo I, P, EI o EP, o la parte P de la trama PB o PB mejorada), se cumplan las dos reglas siguientes:

- 1) Todas las imágenes de referencia con la misma referencia temporal aparecerán en el tren de bits en un orden de capa de mejora creciente (dado que se necesita cada imagen de

referencia de capa más baja para decodificar la imagen de referencia de capa más alta siguiente).

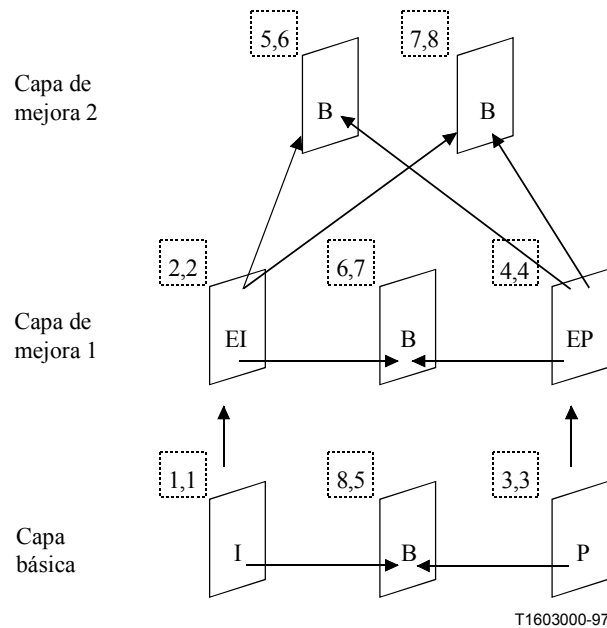
- 2) Todas las imágenes de referencia temporalmente simultáneas a las que se refiere el anterior punto 1) aparecerán en el tren de bits antes que cualquier imagen B de la que cualquiera de esas imágenes de referencia sea la primera imagen de referencia temporalmente subsiguiente en la capa de referencia de la imagen B (a fin de reducir el retardo de decodificación de todas las imágenes de referencia que pueden ser necesarias como referencias para las imágenes B).

A continuación seguirán las imágenes B con referencias temporales anteriores (ordenadas temporalmente dentro de cada capa de mejora).

La ubicación de cada imagen B en el tren de bits cumplirá las siguientes reglas:

- 1) Estará situada después de su primera imagen de referencia temporalmente subsiguiente en la capa de referencia (dado que la decodificación de la imagen B depende generalmente de la decodificación anterior de esa imagen de referencia).
- 2) Estará situada después de la de todas las imágenes de referencia temporalmente simultáneas con la primera imagen de referencia temporalmente subsiguiente en la capa de referencia (para reducir el retardo de decodificación de todas las imágenes de referencia que pueden ser necesarias como referencias para las imágenes B).
- 3) Precederá la ubicación de cualquier imagen adicional temporalmente subsiguiente distinta de las imágenes B en su capa de referencia (ya que de otro modo aumentarían los requisitos de memoria de almacenamiento de imágenes para las imágenes de la capa de referencia).
- 4) Estará situada después de todas las imágenes EI y EP temporalmente simultáneas con la primera imagen de referencia temporalmente subsiguiente.
- 5) Precederá la ubicación de todas las imágenes temporalmente subsiguientes dentro de su misma capa de mejora (ya que de otro modo se introduciría un retardo innecesario y aumentarían los requisitos de memoria de almacenamiento de imágenes para la capa de mejora).

En la figura O.5 se ilustran dos posibles órdenes de transmisión de imágenes en consonancia con las reglas anteriores para la estructuración en capas que en ella se muestra (los números de las casillas de puntos indican el orden del tren de bits, separados por comas para las dos alternativas).



**Figura O.5/H.263 – Ejemplo de orden de transmisión de imágenes**

### O.3 Sintaxis de la capa de imagen

El número de la capa de mejora (ELNUM) (véase 5.1.11) está presente siempre en cualquier imagen (B, EI o EP) de mejora y no debe estar presente en imágenes I o P, ni en tramas PB o PB mejoradas. El número de la capa de referencia (RLNUM) (véase 5.1.12) está presente para algunas imágenes de mejora y se infiere para otras, como se describe en 5.1.12.

Hay exactamente una capa básica y tiene números ELNUM y RLNUM iguales a 1. El RLNUM del número de capa de mejora de las imágenes de referencia hacia adelante y hacia atrás de imágenes B, y de la imagen de referencia hacia arriba de imágenes EI y EP. Las imágenes de referencia de la capa básica pueden estar formadas por imágenes I, PB, PB mejorada y P, ninguna de las cuales incluye los números ELNUM o RLNUM en el encabezamiento de la imagen (sus valores implícitos son 1).

Para las imágenes B, el RLNUM debe ser menor o igual al ELNUM, mientras que para las imágenes EI y EP, el RLNUM debe ser inferior al ELNUM.

El ELNUM puede ser diferente del número de capa utilizado a nivel de sistema. Puesto que las imágenes B no tienen otras imágenes que dependan de ellas, pueden incluso ser colocadas en una capa de mejora separada por componentes del sistema externos a la presente Recomendación (por ejemplo, aplicando las Recs UIT-T H.245 y H.223). Por otra parte, corresponde al implementador decidir si las imágenes de mejora se envían por canales de vídeo separados o permanecen multiplexadas junto con las imágenes de la capa básica.

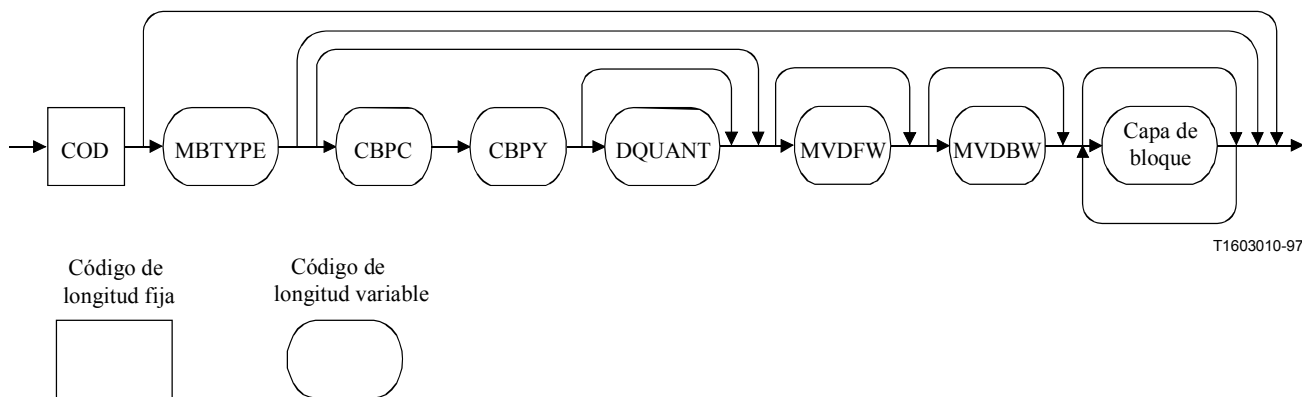
Como se indicó en 5.1.4.5, el modo filtro de desbloqueo (véase el anexo J) no se aplica en las imágenes B. Esto se debe a que las imágenes B no se utilizan para la predicción de ninguna otra imagen y, por lo tanto, la aplicación de un filtro de desbloqueo a esas imágenes es totalmente una técnica de post-tratamiento que queda fuera del alcance de esta Recomendación. No obstante, se recomienda la utilización de algún tipo de filtro de desbloqueo para las imágenes B, y un filtro como el descrito en el anexo J puede ser, de hecho un buen diseño a utilizar a tales efectos.

La referencia temporal (TR) (véase 5.1.2) se define exactamente como para las imágenes I y P.

En el encabezamiento de imagen de las imágenes B, EI o EP, no habrá campos  $TR_B$  (véase 5.1.22) ni DBQUANT (véase 5.1.23).

## O.4 Sintaxis de la capa de macrobloque

La sintaxis de la capa de macrobloque para imágenes B y EP es la misma, ya que cada una de ellas utiliza dos imágenes de referencia de un modo similar. No obstante, la interpretación varía ligeramente según el tipo de imagen. En la figura O.6 se indica la sintaxis B y EP. El campo MBTYPE indica si existe una predicción en modo directo, una predicción hacia adelante, una predicción hacia atrás/hacia arriba o una predicción bidireccional. El MBTYPE se define de manera diferente para las imágenes B y EP, como se describe a continuación.



**Figura O.6/H.263 – Sintaxis de macrobloque para imágenes EP y B**

El modo predicción directa sólo está disponible para las imágenes B. Es un modo de predicción bidireccional similar al modo bidireccional del modo tramas PB mejoradas (anexo M). La única diferencia es que no existe ninguna restricción con respecto a qué píxels se pueden predecir hacia atrás ya que la imagen completa de predicción hacia atrás es conocida en el decodificador. El modo bidireccional utiliza vectores de movimiento separados para la predicción hacia adelante y hacia atrás. En los modos directo y bidireccional, los valores de los píxels de predicción se calculan haciendo un promedio de los píxels de predicción hacia adelante y hacia atrás. El promedio se calcula dividiendo la suma de las dos predicciones por dos (división por truncamiento). En el modo directo, cuando hay cuatro vectores de movimiento en el macrobloque de referencia, se utilizan los cuatro vectores de movimiento como en el modo tramas PB mejoradas (anexo M).

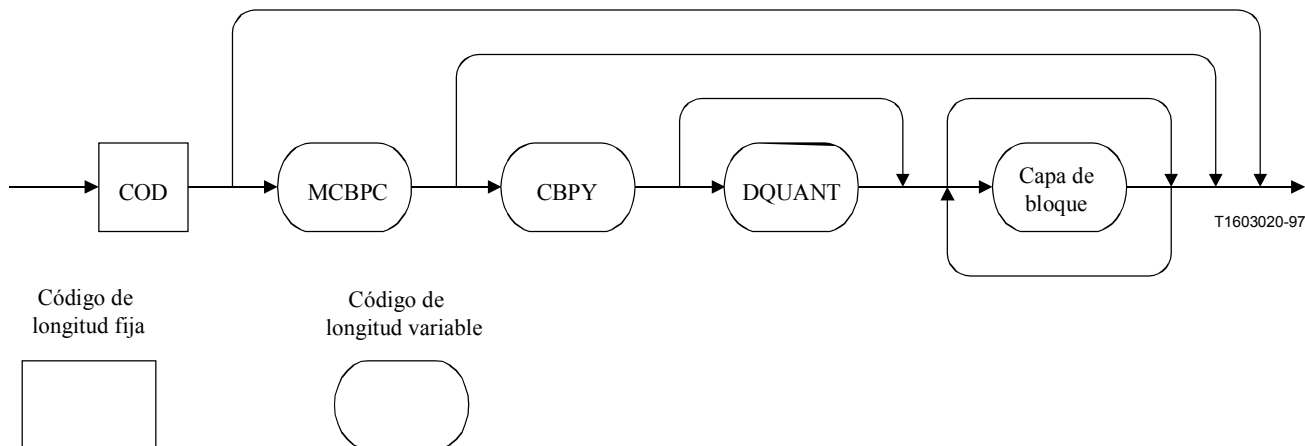
Para las imágenes B, predicción hacia adelante significa predicción a partir de una imagen de referencia anterior en la capa de referencia. Predicción hacia atrás significa predicción a partir de una imagen de referencia temporalmente subsiguiente en la capa de referencia.

Para las imágenes EP, predicción hacia adelante significa predicción a partir de una imagen EI o EP anterior en la misma capa, mientras que predicción hacia arriba significa predicción a partir de la imagen de referencia (posiblemente interpolada) temporalmente simultánea en la capa de referencia. No se utiliza ningún vector de movimiento para la predicción hacia arriba (la cual, sintácticamente, está en el mismo lugar que la predicción hacia atrás para las imágenes B), aunque puede utilizarse uno de ellos para la predicción hacia adelante.

La sintaxis de macrobloque para las imágenes EI es ligeramente distinta. Como se muestra en la figura O.7, MBTYPE y CBPC se combinan en un campo MCBPC. No se utiliza ninguna predicción hacia adelante, únicamente la predicción hacia arriba a partir de la imagen de referencia temporalmente simultánea en la capa de referencia. No se utiliza ningún vector de movimiento.

En las imágenes B y EP, pueden utilizarse vectores de movimiento en las fronteras de la imagen, tal como se describe en D.1 (aunque la extensión de la gama de vectores de movimiento, descrita en D.2, sólo está activa si se utiliza también el modo vector de movimiento sin restricción).

Se debe garantizar que los codificadores estén conformes con D.1.1 para todos los macrobloques, incluidos los que se predicen con el modo de predicción directa; es decir, el codificador no ha de seleccionar el modo de predicción directa a no ser que los valores del vector de movimiento que se infieren en el proceso de predicción de modo directo no correspondan con ningún elemento de la región de predicción de  $16 \times 16$  (u  $8 \times 8$ ) con una distancia horizontal o vertical de más de 15 píxeles fuera del área de la imagen codificada.



**Figura O.7/H.263 – Sintaxis de macrobloque para imágenes EI**

#### **O.4.1 Indicación de macrobloque codificado (COD) (1 bit)**

Se trata de un bit que, cuando se fija a "0", señala que el bloque está codificado. Cuando se fija a "1", no se transmite más información para este macrobloque, que se trata como un macrobloque "saltado", como se describe a continuación.

#### **O.4.2 MBTYPE/MCBPC [Código de longitud variable (VCL)]**

Hay distintos cuadros MBTYPE para imágenes B e imágenes EP. Para imágenes EI, en cambio, hay un cuadro MCBPC. El cuadro O.1 es el cuadro MBTYPE para imágenes B. El cuadro O.2 es el cuadro MBTYPE para imágenes EP. El cuadro O.3 es el cuadro MCBPC para imágenes EI.

Para las imágenes B, el tipo de predicción "directa" (con salto)" indica que en el macrobloque no se transmite ni MBTYPE ni ningún dato y que el decodificador deriva los vectores de movimiento hacia adelante y hacia atrás y la predicción bidireccional correspondiente. Esto es señalizado por el bit COD. Los tipos de predicción "hacia adelante (sin textura)", "hacia atrás (sin textura)" y "Bi-dir (sin textura)" para una imagen B indican predicción hacia adelante, hacia atrás y bidireccional sin coeficientes, y transmisión de un vector de movimiento para la predicción hacia adelante y hacia atrás, y de dos vectores de movimiento para la predicción bidireccional.

Para las imágenes EP, el tipo de predicción "hacia adelante (con salto)" indica que no se envía ningún dato adicional para el macrobloque, por lo que el decodificador debe utilizar la predicción hacia adelante con un vector de movimiento cero y sin coeficientes. Los tipos de predicción "hacia arriba (sin textura)" y "Bi-dir (sin textura)" para una imagen EP indican predicción hacia arriba y bidireccional sin coeficientes, y con vectores de movimiento cero.

Para las imágenes EI, el tipo de predicción "hacia arriba (con salto)" indica que no se envía ningún dato adicional para el macrobloque, por lo que el decodificador debe utilizar la predicción hacia arriba con un vector de movimiento cero y sin coeficientes.

**Cuadro O.1/H.263 – Códigos de longitud variable (VLC) MBTYPE para imágenes B**

Índice	Tipo de predicción	MVDFV	MVDBV	CBPC + CBPY	DQUANT	MBTYPE	Bits
–	Directa (con salto)					(COD = 1)	0
0	Directa			X		11	2
1	Directa + Q			X	X	0001	4
2	Hacia adelante (sin textura)	X				100	3
3	Hacia adelante	X		X		101	3
4	Hacia adelante + Q	X		X	X	0011 0	5
5	Hacia atrás (sin textura)		X			010	3
6	Hacia atrás		X	X		011	3
7	Hacia atrás + Q		X	X	X	0011 1	5
8	Bi-dir (sin textura)	X	X			0010 0	5
9	Bi-dir	X	X	X		0010 1	5
10	Bi-dir + Q	X	X	X	X	0000 1	5
11	INTRA			X		0000 01	6
12	INTRA + Q			X	X	0000 001	7
13	Relleno					0000 0000 1	9

**Cuadro O.2/H.263 – Códigos de longitud variable (VLC) MBTYPE para imágenes EP**

Índice	Tipo de predicción	MVDFV	MVDBV	CBPC + CBPY	DQUANT	MBTYPE	Bits
–	Hacia adelante (con salto)					(COD = 1)	0
0	Hacia adelante	X		X		1	1
1	Hacia adelante + Q	X		X	X	001	3
2	Hacia arriba (sin textura)					010	3
3	Hacia arriba			X		011	3
4	Hacia arriba + Q			X	X	0000 1	5
5	Bi-dir (sin textura)					0001 0	5
6	Bi-dir	X		X		0001 1	5
7	Bi-dir + Q	X		X	X	0000 01	6
8	INTRA			X		0000 001	7
9	INTRA + Q			X	X	0000 0001	8
10	Relleno					0000 0000 1	9



**Cuadro O.3/H.263 – Códigos de longitud variable (VLC) MCBPC para imágenes EI**

Índice	Tipo de predicción	Patrón de bloque de codificación (56)	CBPY	DQUANT	MCBPC	Bits
–	Hacia arriba (con salto)				(COD = 1)	0
0	Hacia arriba	00	X		1	1
1	Hacia arriba	01	X		001	3
2	Hacia arriba	10	X		010	3
3	Hacia arriba	11	X		011	3
4	Hacia arriba + Q	00	X	X	0001	4
5	Hacia arriba + Q	01	X	X	0000 001	7
6	Hacia arriba + Q	10	X	X	0000 010	7
7	Hacia arriba + Q	11	X	X	0000 011	7
8	INTRA	00	X		0000 0001	8
9	INTRA	01	X		0000 1001	8
10	INTRA	10	X		0000 1010	8
11	INTRA	11	X		0000 1011	8
12	INTRA+Q	00	X	X	0000 1100	8
13	INTRA+Q	01	X	X	0000 1101	8
14	INTRA+Q	10	X	X	0000 1110	8
15	INTRA+Q	11	X	X	0000 1111	8
16	Relleno				0000 0000 1	9

**O.4.3 Patrón de bloque codificado para crominancia (CBPC, *coded block pattern for chrominance*) (Longitud variable)**

Si está presente, CBPC indica el patrón de bloque codificado para bloques de crominancia, tal como se describe en el cuadro O.4. CBPC sólo está presente para imágenes EP y B, cuando su presencia es indicada por MBTYPE (véanse los cuadros O.1 y O.2).

**Cuadro O.4/H.263 – Códigos CBPC VLC**

Índice	Patrón de bloque de codificación (56)	CBPC	Bits
0	00	0	1
1	01	10	2
2	10	111	3
3	11	110	3

**O.4.4 Patrón de bloque codificado para luminancia (CBPY) (Longitud variable)**

Si está presente, CBPY indica qué bloques figuran en la porción luminancia del macrobloque. CBPY sólo está presente cuando su presencia es indicada por MBTYPE (véanse los cuadros O.1, O.2 y O.3). CBPY se codifica tal como se describe en 5.3.5 y en el cuadro 12. Los macrobloques con predicción hacia arriba en las imágenes EI y EP, los macrobloques con predicción bidireccional

de las imágenes EP y los macrobloques INTRA de las imágenes EI, EP y B utilizan la definición CBPY para macrobloques INTRA, y otros tipos de macrobloques de las imágenes EI, EP y B utilizan la definición CBPY para macrobloques INTER.

#### **O.4.5 Información de cuantificador (DQUANT) (2 bits/Longitud variable)**

DQUANT se utiliza como en otros tipos de macrobloques de imágenes. Véanse 5.3.6 y el anexo T.

#### **O.4.6 Datos de vector de movimiento (MVDFW, MVDBW) (Longitud variable)**

MVDFW son los datos de vector de movimiento del vector hacia adelante, si está presente. MVDBW son los datos de vector de movimiento del vector hacia atrás, si está presente (autorizado únicamente en las imágenes B). Las palabras de código de longitud variable se indican en el cuadro 14, o en el cuadro D.3 si se utiliza el modo vector de movimiento sin restricción (véase el anexo D).

### **O.5 Decodificación de vector de movimiento**

#### **O.5.1 Vectores de movimiento diferenciales**

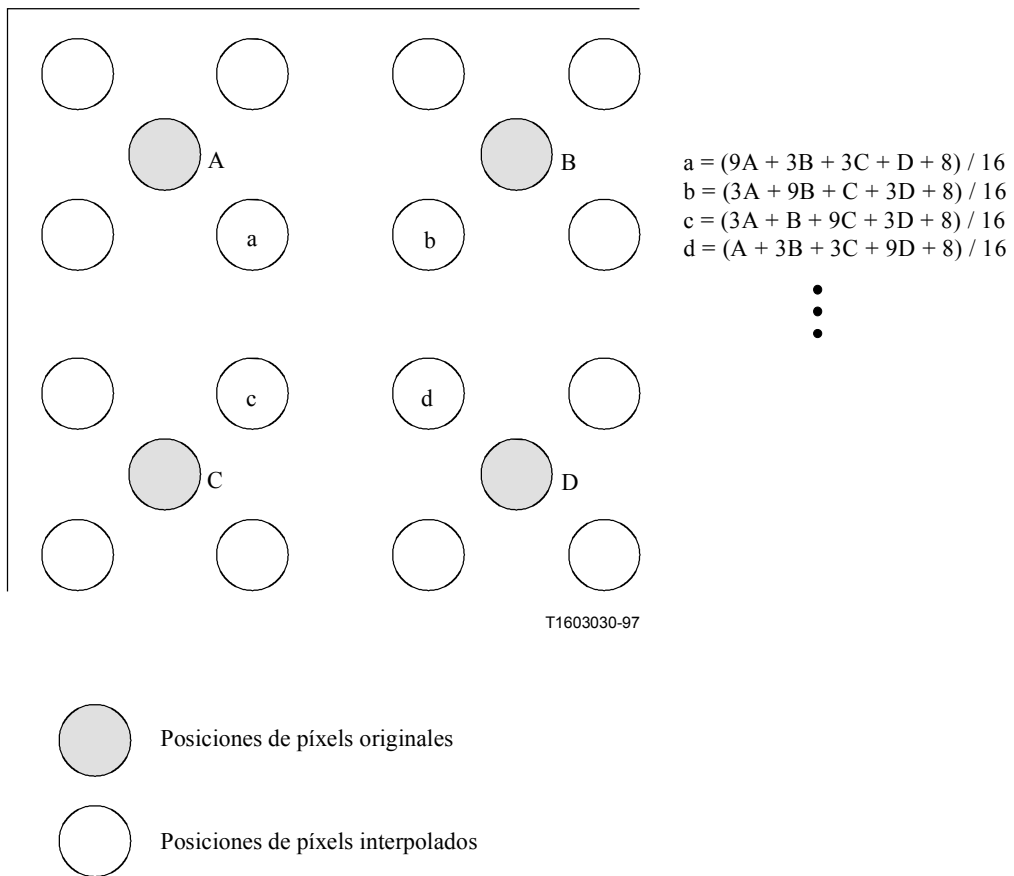
Los vectores de movimiento de bloques con predicción hacia adelante, hacia atrás o bidireccional tienen una codificación diferencial. Para recuperar los vectores de movimiento de un macrobloque, se añade una predicción a las diferencias de los vectores de movimiento. Las predicciones se forman de manera similar a la descrita en 6.1.1, con la salvedad de que la predicción hacia adelante de los vectores de movimiento se efectúa únicamente a partir de vectores de movimiento hacia adelante de macrobloques adyacentes, y la predicción de los vectores de movimiento hacia atrás se efectúa únicamente a partir de vectores de movimiento hacia atrás de macrobloques adyacentes. Se aplican las mismas reglas de decisión para los casos especiales en las fronteras de imagen o límite de GOB o rebanada, como se describe en 6.1.1. Si un macrobloque contiguo no tiene un vector de movimiento del mismo tipo (hacia adelante o hacia atrás), el predictor más probable de ese bloque es cero para ese tipo de vector de movimiento.

#### **O.5.2 Vectores de movimiento en modo directo**

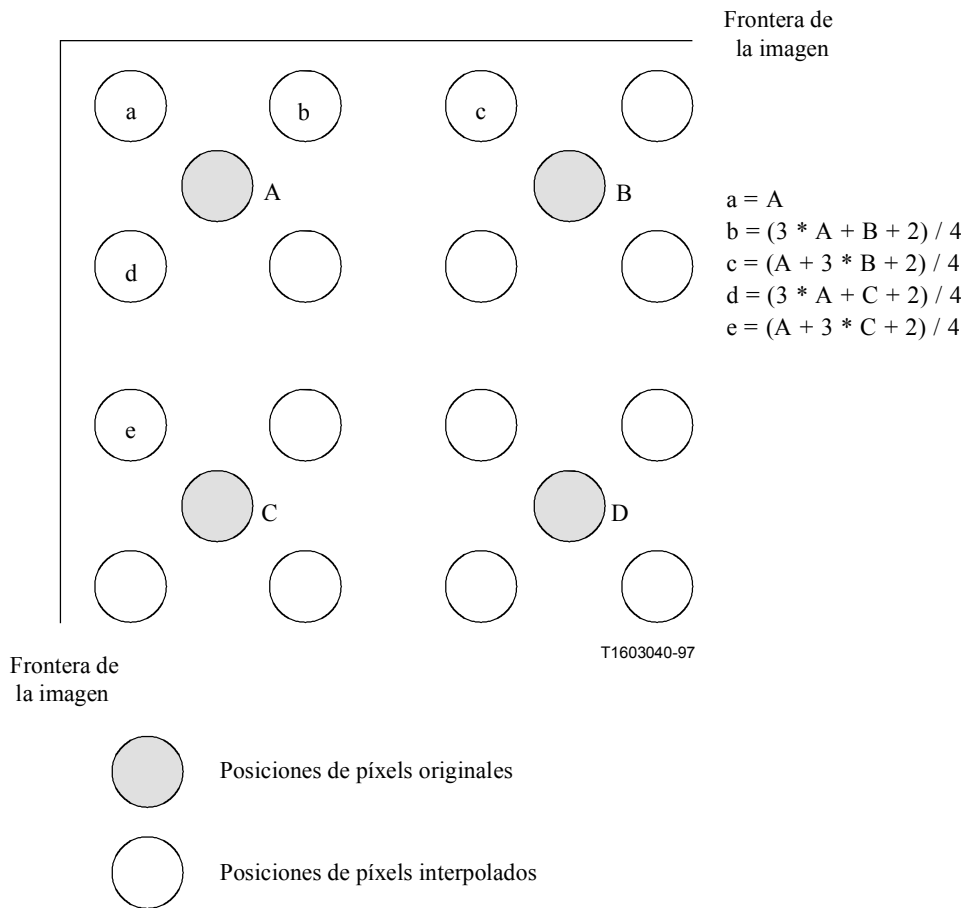
Para macrobloques codificados en modo directo, no se transmiten diferencias vectoriales. En cambio, los vectores de movimiento hacia adelante y hacia atrás se calculan directamente a partir del vector P temporalmente consecutivo, como se describe en G.4, con la restricción de que  $MV_D$  es siempre cero. Estos vectores derivados no se utilizan para la predicción de otros vectores de movimiento. Si el área correspondiente de la imagen de referencia temporalmente subsiguiente está codificada con el modo INTRA, los vectores de movimiento hacia adelante y hacia atrás asignados a esa área para el proceso de predicción en el modo directo, han de tener un valor de cero.

### **O.6 Filtros de interpolación**

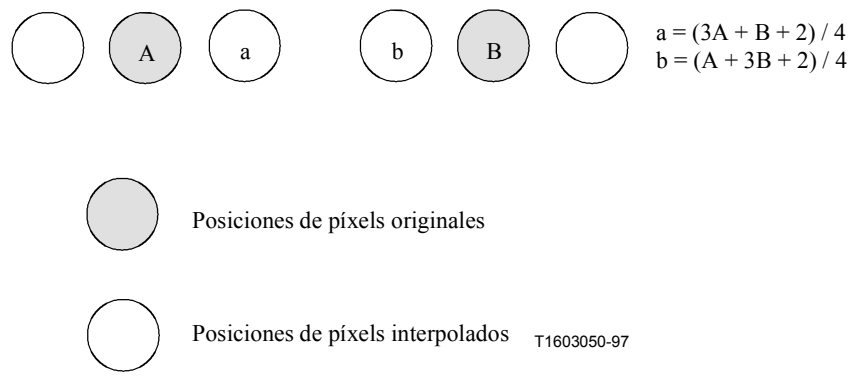
En las figuras O.8 y O.9 se muestra el método mediante el cual se interpola una imagen para escalabilidad espacial 2-D. La primera figura muestra la interpolación de píxeles interiores, en tanto que la segunda muestra la interpolación próxima a las fronteras de la imagen. Es la misma técnica utilizada en el anexo Q y, en algunos casos, en el anexo P. En las figuras O.10 y O.11 se muestra el método mediante el cual se interpola una imagen para escalabilidad espacial 1-D. La figura O.10 muestra la interpolación de píxeles interiores en sentido horizontal. La interpolación en sentido vertical es análoga. La figura O.11 muestra la interpolación de píxeles en las fronteras de la imagen. También en este caso la interpolación en sentido vertical es análoga. Una vez más, se trata de la misma técnica utilizada en algunos casos en el anexo P.



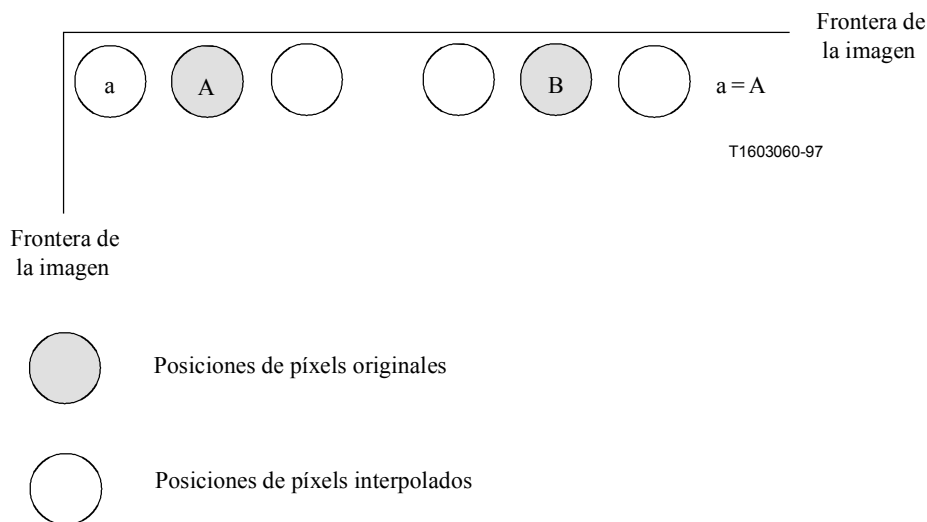
**Figura O.8/H.263 – Método de interpolación de píxels para escalabilidad 2-D**



**Figura O.9/H.263 – Método de interpolación 2-D en las fronteras de la imagen**



**Figura O.10/H.263 – Método de interpolación de píxeles para escalabilidad 1-D**



**Figura O.11/H.263 – Método de interpolación 1-D en las fronteras de la imagen**

## Anexo P

### Repetición de muestreo de la imagen de referencia

#### P.1 Introducción

Este anexo se describe el uso y la sintaxis de un proceso de repetición de muestreo que puede aplicarse a la imagen de referencia decodificada anteriormente a fin de generar una imagen "alabeada" para utilizarla en la predicción de la imagen en curso. La sintaxis de esta repetición de muestreo puede especificar la relación entre la imagen vigente y una imagen anterior con un formato de fuente distinto, y puede especificar también una alteración por alabeo del "movimiento total" de la forma, el tamaño y la localización de la imagen anterior con respecto a la imagen vigente. En particular, el modo repetición de muestreo de la imagen de referencia puede utilizarse para alterar de manera adaptable la resolución de imágenes durante la codificación. Se utiliza un algoritmo rápido para generar coeficientes de interpolación bilineal. La capacidad de utilizar este modo y el grado en que se admiten sus características se negocia externamente (por ejemplo, aplicando la Rec. UIT-T H.245). Este modo puede utilizarse en escenarios restringidos que pueden definirse durante la negociación de la capacidad (por ejemplo, para admitir únicamente el redimensionamiento de la imagen con factor de cuatro, para admitir sólo la distorsión de la imagen con resolución de medio píxel o para sustentar el redimensionamiento y los desplazamientos arbitrarios de la imagen).

NOTA – La transformación de la imagen por defecto entre imágenes de resoluciones diferentes se define para mantener el alineamiento espacial de los bordes de la zona de la imagen y de las ubicaciones relativas de las muestras de luminancia y crominancia. Esto puede tener consecuencias en el diseño de cualquier operación de repetición de muestreo efectuada para generar imágenes de diversas resoluciones en el codificador y para la visualización de imágenes de diversas resoluciones después de la decodificación (en particular, con respecto a los desplazamientos de la ubicación espacial causados por desplazamientos de fase inducidos en la repetición del muestreo). Además, como este modo puede utilizarse para los cambios dinámicos de resolución de imagen adaptativa, el funcionamiento con este modo puede aprovechar la negociación externa para visualizar la imagen decodificada con una resolución más alta que el tamaño de su imagen codificada, a fin de permitir la conmutación entre tamaños de imágenes codificadas sin redimensionar la visualización de la imagen.

Si no se fija el bit de repetición de muestreo de la imagen de referencia en el campo PLUSPTYPE, pero PLUSPTYPE está presente y la imagen es una imagen INTER, B o EP, o una trama PB mejorada, y el tamaño de la imagen es distinto del de la imagen codificada temporalmente previa, esta condición invoca la repetición del muestreo de la imagen de referencia con parámetros de alabeo (véase P.2.2) fijados a cero, el modo relleno (véase P.2.3) fijado a *recorte*, y la exactitud del desplazamiento (véase P.2.1) fijada a una exactitud de  $\frac{1}{16}$  píxel. Esto hace que el proceso de repetición de muestreo actúe fácilmente como un cambio de la resolución de la imagen codificado predictivamente. En los casos sencillos de cambios con resolución de factor de cuatro, tales como las transiciones entre CIF y 4 veces CIF, el proceso de repetición de muestreo se reduce al mismo simple filtrado aplicado para la escalabilidad espacial (anexo O) o para la actualización de resolución reducida (anexo Q), salvo por la aplicación del control de redondeo.

Si la imagen es una imagen EP y en el campo PLUSPTYPE del encabezamiento de imagen de la capa de referencia se fija el bit de repetición de muestreo de la imagen de referencia, se fija también dicho bit en el encabezamiento de imagen de la imagen EP en la capa de mejora.

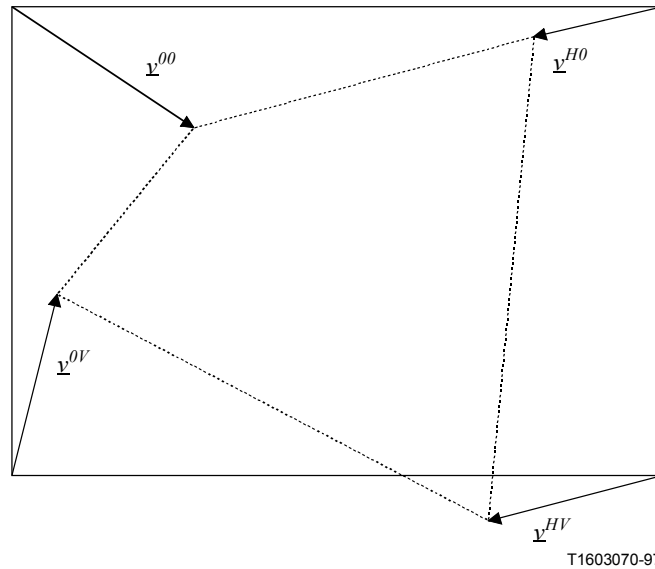
Si una imagen B utiliza el modo repetición de muestreo de la imagen de referencia, el proceso de repetición de muestreo se aplicará a la imagen de anclaje temporalmente previa y no a la imagen temporalmente subsiguiente. La imagen de anclaje temporalmente previa a la que se aplica el proceso de repetición de muestreo será la imagen decodificada (es decir, antes de cualquier repetición de muestreo aplicada por el modo repetición de muestreo de la imagen de referencia si este modo también es invocado para la imagen de referencia subsiguiente). La imagen de anclaje temporalmente subsiguiente tendrá el mismo tamaño que la imagen B.

Si el modo repetición de muestreo de la imagen de referencia es invocado para una trama PB mejorada, se envía una serie de parámetros de alabeo y se utiliza la imagen de referencia remuestreada como referencia tanto como para la parte B como para la parte P de la trama PB mejorada.

No se ha de invocar el modo de repetición de muestreo de la imagen de referencia si se está utilizando el modo selección de la imagen de referencia (véase el anexo N), a no ser que los valores de TRPI y TRP en todos los encabezamientos de rebanada, GOB e imagen de la imagen actual especifiquen la utilización de la misma imagen de referencia, en cuyo caso la imagen de referencia indicada es la imagen que determina si se invoca implícitamente el proceso de repetición de muestreo y es también la imagen a la que se le ha de aplicar el proceso de repetición de muestreo.

La repetición de muestreo de la imagen de referencia se define en función del desplazamiento de las cuatro esquinas de la zona de la imagen en curso. Para el campo de luminancia de la imagen vigente con dimensiones horizontal y vertical  $H$  y  $V$ , respectivamente se definen cuatro vectores de movimiento conceptual  $\underline{v}^{00}$ ,  $\underline{v}^{H0}$ ,  $\underline{v}^{0V}$ , y  $\underline{v}^{HV}$ , para las esquinas superior izquierda, superior derecha, inferior izquierda e inferior derecha, respectivamente. Estos vectores describen cómo desplazar las esquinas de la imagen vigente para establecer una correspondencia con las esquinas correspondientes de la imagen decodificada anterior, tal como se muestra en la figura P.1. Las unidades de estos vectores son las mismas que las de la cuadrícula de la imagen de referencia. Para generar un vector  $\underline{v}(x, y)$  en alguna ubicación con valor real  $(x, y)$  en el interior de la imagen vigente, se utiliza una aproximación a la interpolación bilineal, es decir, como en:

$$\underline{v}(x, y) = \left(1 - \frac{y}{V}\right) \left[ \left(1 - \frac{x}{H}\right) \underline{v}^{00} + \left(\frac{x}{H}\right) \underline{v}^{H0} \right] + \left(\frac{y}{V}\right) \left[ \left(1 - \frac{x}{H}\right) \underline{v}^{0V} + \left(\frac{x}{H}\right) \underline{v}^{HV} \right]$$



**Figura P.1/H.263 – Ejemplo de vectores de movimiento conceptual utilizados para el alabeo**

La dimensión horizontal  $H$  y la dimensión vertical  $V$  de la imagen vigente, y la dimensión horizontal  $H_R$  y la dimensión vertical  $V_R$  de la imagen de referencia son los indicados por el encabezamiento de la imagen, con independencia de si estos valores sean divisibles o no por 16. Si la anchura o altura de la imagen no es divisible por 16, la zona adicional se generará añadiendo píxels a la imagen remuestreada utilizando el mismo modo de relleno empleado en el proceso de repetición del muestreo.

Para hacer más sencilla la descripción, los vectores de repetición de muestreo  $\underline{r}^0$ ,  $\underline{r}^x$ ,  $\underline{r}^y$  y  $\underline{r}^{xy}$  se definen como:

$$\begin{aligned} \underline{r}^0 &= \underline{v}^{00} \\ \underline{r}^x &= \underline{v}^{H0} - \underline{v}^{00} \\ \underline{r}^y &= \underline{v}^{0V} - \underline{v}^{00} \\ \underline{r}^{xy} &= \underline{v}^{00} - \underline{v}^{H0} - \underline{v}^{0V} + \underline{v}^{HV} \end{aligned}$$

Utilizando esta definición, la ecuación para la interpolación bilineal se reescribe de la siguiente manera:

$$\underline{v}(x, y) = \underline{r}^0 + \left(\frac{x}{H}\right)\underline{r}^x + \left(\frac{y}{V}\right)\underline{r}^y + \left(\frac{x}{H}\right)\left(\frac{y}{V}\right)\underline{r}^{xy}$$

A efectos del alabeo se supone que las coordenadas de la esquina superior izquierda de la zona de la imagen son  $(x, y) = (0,0)$  y que cada píxel tiene una altura y una anchura unidad, de manera que los centros de los píxels están en los puntos  $(x, y) = \left(i_L + \frac{1}{2}, j_L + \frac{1}{2}\right)$  para  $i_L = 0, \dots, H-1$  y  $j_L = 0, \dots, V-1$ , donde el subíndice  $L$  indica que  $i_L$  y  $j_L$  pertenecen al campo de luminancia. (Puesto que la relación de aspecto de píxel es constante por lo general o la conversión de la relación de aspecto se efectúa mediante esta repetición de muestreo, no es necesario a estos efectos considerar la relación de aspecto de píxel real.) Utilizando este convenio, los desplazamientos  $x$  e  $y$  en las ubicaciones de interés del campo de luminancia de la imagen de referencia son:

$$v_x(i_L, j_L) = \frac{1}{HV} \left[ HVr_x^0 + \left(i_L + \frac{1}{2}\right)Vr_x^x + \left(j_L + \frac{1}{2}\right)Hr_x^y + \left(i_L + \frac{1}{2}\right)\left(j_L + \frac{1}{2}\right)r_x^{xy} \right]$$

$$v_y(i_L, j_L) = \frac{1}{HV} \left[ HVr_y^0 + \left(i_L + \frac{1}{2}\right) Vr_y^x + \left(j_L + \frac{1}{2}\right) Hr_y^y + \left(i_L + \frac{1}{2}\right) \left(j_L + \frac{1}{2}\right) r_y^{xy} \right]$$

Dado que todas las posiciones y fases deben calcularse en relación con el centro del píxel de la esquina superior izquierda, cuyas coordenadas  $(x, y) = \left(\frac{1}{2}, \frac{1}{2}\right)$ , las cantidades de interés principal son:

$$\begin{aligned} x_R(i_L, j_L) - \frac{1}{2} &= \left(i_L + \frac{1}{2}\right) + v_x(i_L, j_L) - \frac{1}{2} \\ &= \frac{1}{HV} \left[ HVr_x^0 + \left(i_L + \frac{1}{2}\right) (HV + Vr_x^x) + \left(j_L + \frac{1}{2}\right) Hr_x^y + \left(i_L + \frac{1}{2}\right) \left(j_L + \frac{1}{2}\right) r_x^{xy} \right] - \frac{1}{2} \\ y_R(i_L, j_L) - \frac{1}{2} &= \left(i_L + \frac{1}{2}\right) + v_y(i_L, j_L) - \frac{1}{2} \\ &= \frac{1}{HV} \left[ HVr_y^0 + \left(j_L + \frac{1}{2}\right) Vr_y^x + \left(j_L + \frac{1}{2}\right) (HV + Hr_y^y) + \left(i_L + \frac{1}{2}\right) \left(j_L + \frac{1}{2}\right) r_y^{xy} \right] - \frac{1}{2} \end{aligned}$$

Una vez determinada la ubicación en la imagen de referencia decodificada anterior utilizando una aproximación de estas situaciones, se utilizará la interpolación bilineal, tal como se especifica más adelante en este anexo, para generar un valor del píxel remuestreado.

Cada vector de repetición de muestreo puede descomponerse en dos componentes; el primero describe el alabeo geométrico y el segundo representa cualquier diferencia de tamaño entre la imagen predicha (dimensión horizontal  $H$  y dimensión vertical  $V$ ) y la imagen de referencia (dimensión horizontal  $H_R$  y dimensión vertical  $V_R$ ). Esta descomposición se efectúa como sigue:

$$\begin{aligned} \underline{v}^{00} &= \underline{v}_{warp}^{00} + \underline{v}_{size}^{00} = \underline{v}_{warp}^{00} + (0,0) \\ \underline{v}^{H0} &= \underline{v}_{warp}^{H0} + \underline{v}_{size}^{H0} = \underline{v}_{warp}^{H0} + (H_R - H, 0) \\ \underline{v}^{0V} &= \underline{v}_{warp}^{0V} + \underline{v}_{size}^{0V} = \underline{v}_{warp}^{0V} + (0, V_R - V) \\ \underline{v}^{HV} &= \underline{v}_{warp}^{HV} + \underline{v}_{size}^{HV} = \underline{v}_{warp}^{HV} + (H_R - H, V_R - V) \end{aligned}$$

## P.2 Sintaxis

Cuando en el campo PLUSPTYPE del encabezamiento de la imagen se fija el bit de la repetición de muestreo de la imagen de referencia, el campo RPRP de dicho encabezamiento incluye los parámetros que controlan el proceso de repetición de muestreo de la imagen de referencia. Se incluye aquí un campo de exactitud del desplazamiento del alabeo (WDA, *warping displacement accuracy*) de dos bits, pueden incluirse ocho parámetros de alabeo o el refinamiento de parámetros de alabeo de 1 bit, y se incluye un modo de relleno, tal como se describe en esta subcláusula.

### P.2.1 Exactitud del desplazamiento del alabeo (WDA, *warping displacement accuracy*) (2 bits)

En el campo RPRP del tren de bits aparece primero un campo de exactitud del desplazamiento del alabeo (WDA) de 2 bits, e indica la exactitud de los desplazamientos de cada píxel. Un valor de "10" indica que los desplazamientos  $x$  e  $y$  de cada píxel se cuantifican hasta una exactitud de medio píxel. Un valor de "11" indica que los desplazamientos se cuantifican hasta una exactitud de  $\frac{1}{16}$  píxel. La utilización de otros valores queda en reserva.



### P.2.2 Parámetros de alabeo (Longitud variable)

Cuando se envían los parámetros de repetición de muestreo de la imagen de referencia para una imagen INTER o B, o una trama PB mejorada, se incluyen 8 parámetros de alabeo en el encabezamiento de la imagen utilizando el código de longitud variable (VLC) que figura en el cuadro D.3. Para una imagen EP con escalabilidad SNR, se utilizan los parámetros de alabeo de la capa más baja y no se transmite ningún parámetro de alabeo. Si se fija el bit de repetición de muestreo de la imagen de referencia en el campo PLUSPTYPE del encabezamiento de la imagen para una imagen EP con escalabilidad espacial, se efectúa el refinamiento de los parámetros de alabeo de la capa más baja hasta la exactitud necesaria para la capa vigente multiplicando por dos el parámetro de alabeo para cada dimensión muestreada a velocidad superior (parámetros de alabeo con subíndice  $x$  y/o  $y$ ) de la capa más baja y añadiendo el valor de un bit adicional que se envía en lugar del parámetro de alabeo asociado para definir el bit menos significativo del parámetro de alabeo.

Los ocho parámetros de distorsión enteros (o sus refinamientos de un bit) se envían en el orden siguiente:

$$w_x^0, w_y^0, w_x^x, w_y^x, w_x^y, w_y^y, w_x^{xy}, y w_y^{xy}$$

Si no hay refinamiento de un bit, estos parámetros se envían en forma similar a las diferencias de los vectores de movimiento en el modo vector de movimiento sin restricción cuando PLUSPTYPE está presente, utilizando el cuadro D.3 sin ninguna restricción en la gama de los parámetros (es decir, una gama de  $-4095$  a  $+4095$ ). Al igual que cuando se codifican pares de diferencias de vectores de movimiento, se añadirá un bit de prevención de la emulación, según se necesite, después del envío de cada par de parámetros de alabeo, de tal manera que si se utiliza la palabra de código de todo ceros (el valor  $+1$  en unidades de medio píxel) del cuadro D.3 para ambos parámetros de alabeo del par  $(w_x^0$  y  $w_y^0, w_x^x$  y  $w_y^y, o w_x^{xy}$  y  $w_y^{xy})$ , el par de palabras de código irá seguido de un único bit igual a 1 para evitar la emulación del código de comienzo.

Estos ocho parámetros de alabeo se interpretan como desplazamientos de la esquina de la imagen con respecto a los desplazamientos que se inducirían al eliminar el componente de redimensionamiento de los vectores de repetición de muestreo. Los parámetros de alabeo se escalonan para representar desplazamientos de medio píxel en el campo de luminancia de la imagen vigente, y la gama de valores de estos parámetros es  $-4095$  a  $+4095$ . Los parámetros de alabeo se definen utilizando los vectores de repetición de muestreo mediante las relaciones:

$$\begin{array}{ll} w_x^0 = 2r_x^0 & w_y^0 = 2r_y^0 \\ w_x^x = 2(r_x^x - (H_R - H)) & w_y^x = 2r_y^x \\ w_x^y = 2r_x^y & w_y^y = 2(r_y^y - (V_R - V)) \\ w_x^y = 2r_x^y & w_y^{xy} = 2r_y^{xy} \end{array}$$

### P.2.3 Modo relleno (FILL\_MODE) (2 bits)

En el caso de imagen INTER o B o trama PB mejorada, inmediatamente después de los parámetros de alabeo codificados VLC en el encabezamiento de la imagen hay dos bits que definen la acción del modo relleno que debe aplicarse para los valores de píxels cuya ubicación calculada en la imagen de referencia queda fuera de la zona de la imagen de referencia. En el cuadro P.1 se muestra el significado de estos dos bits y su ubicación se indica en la figura P.2. En el caso de imagen EP, la

acción del modo relleno es la misma que para la capa de referencia, y no se envían los dos bits del modo relleno.

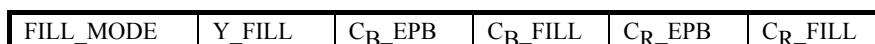
**Cuadro P.1/H.263 – Bits/acción del modo relleno**

Bits del modo relleno	Acción de relleno
0 0	<i>color</i>
0 1	<i>negro</i>
1 0	<i>gris</i>
1 1	<i>recorte</i>

Si el modo relleno es *recorte*, las coordenadas de ubicación en la imagen de referencia previa se limitan de manera independiente, como en el modo vector de movimiento sin restricción, de tal forma que los valores de los píxeles ubicados fuera de la zona de la imagen de referencia previa se calculan mediante la extrapolación de los valores de píxeles en la frontera de la imagen. Si el modo relleno es *negro*, se asigna un valor de  $Y = 16$  a las muestras de luminancia que se hallan fuera de la zona de la imagen de referencia previa y un valor de  $C_B = C_R = 128$  a las muestras de crominancia. Si el modo relleno es *gris*, se asigna un valor de  $Y = C_B = C_R = 128$  a los valores de luminancia y crominancia. Si el modo relleno es *color*, se envían campos adicionales para especificar un color de relleno, tal como se describe en la próxima subcláusula.

**P.2.4 Especificación del color de relleno (Y\_FILL, C<sub>B</sub>\_EPB, C<sub>B</sub>\_FILL, C<sub>R</sub>\_EPB, C<sub>R</sub>\_FILL) (26 bits)**

Si el modo relleno es *color* y la imagen no es una imagen EP, los bits del modo relleno van seguidos en el tren de bits por tres enteros de ocho bits,  $Y\_fill$ ,  $C_B\_fill$ , y  $C_R\_fill$ , que especifican con precisión un color de relleno. Entre estos tres enteros de ocho bits hay dos bits de prevención de la emulación (C<sub>B</sub>\_EPB y C<sub>R</sub>\_EPB), cada uno de los cuales es igual a 1. El formato de esta especificación de color, que sólo está presente cuando el modo relleno es *color*, se muestra en la figura P.2. Cada campo entero de ocho bits se envía utilizando su representación natural. Para una imagen EP, la acción del modo relleno (y color de relleno) es la misma que para la capa de referencia, y no se envía la especificación del color de relleno.



**Figura P.2/H.263 – Formato de los datos del modo relleno y de la especificación del color de relleno**

**P.3 Algoritmo de repetición de muestreo**

El método descrito en esta subcláusula deberá tener un resultado matemáticamente igual al del método utilizado para generar las muestras de la imagen de referencia remuestreada. Utilizando los parámetros de alabeo enteros  $w_x^0, w_y^0, w_x^x, w_y^x, w_x^y, w_y^y, w_x^{xy}, y w_y^{xy}$ , se definen los parámetros enteros  $u_x^{00}, u_y^{00}, u_x^{H0}, u_y^{H0}, u_x^{0V}, u_y^{0V}, u_x^{HV}, y u_y^{HV}$ , que representan los desplazamientos  $x$  e  $y$  en las esquinas del campo de luminancia con una exactitud de  $\frac{1}{32}$  de píxel (los desplazamientos reales se obtienen dividiendo estos valores por 32) como sigue:

$$\begin{aligned}
u_x^{00} &= 16w_x^0 & u_y^{00} &= 16w_y^0 \\
u_x^{H0} &= 16(w_x^0 + w_x^x + 2(H_R - H)) & u_y^{H0} &= 16(w_y^0 + w_y^x) \\
u_x^{0V} &= 16(w_x^0 + w_x^y) & u_y^{0V} &= 16(w_y^0 + w_y^y + 2(V_R - V)) \\
u_x^{HV} &= 16(w_x^0 + w_x^x + w_x^y + w_x^{xy} + 2(H_R - H)) & u_y^{HV} &= 16(w_y^0 + w_y^x + w_y^y + w_y^{xy} + 2(V_R - V))
\end{aligned}$$

A continuación se definen  $H'$  y  $V'$ , que representan la dimensión horizontal y la dimensión vertical de la *trama virtual*, como los enteros más pequeños que satisfacen la condición siguiente:

$$H' \geq H, V' \geq V, H' = 2^m, V' = 2^n, m \text{ y } n \text{ son enteros positivos}$$

Aplicando la extrapolación bilineal a los vectores de esquina del campo de luminancia, se definen los parámetros enteros  $u_x^{LT}, u_y^{LT}, u_x^{RT}, u_y^{RT}, u_x^{LB}, u_y^{LB}, u_x^{RB}, u_y^{RB}$ , que representan los desplazamientos  $x$  e  $y$  del campo de luminancia en los *puntos virtuales*  $(x, y) = (0, 0), (H', 0), (0, V'),$  y  $(H', V')$  con una exactitud de  $\frac{1}{32}$  de píxel (los desplazamientos reales se obtienen dividiendo estos valores por 32) como sigue:

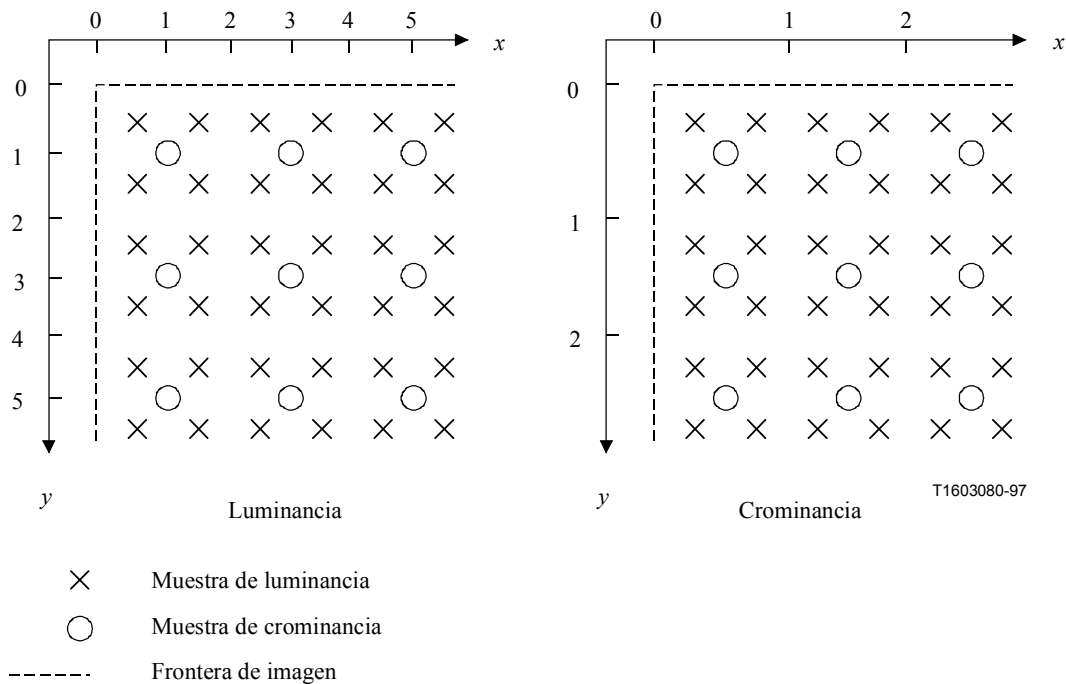
$$\begin{aligned}
u_x^{LT} &= u_x^{00} & u_y^{LT} &= u_y^{00} \\
u_x^{RT} &= ((H - H')u_x^{00} + H'u_x^{H0}) // H & u_y^{RT} &= ((H - H')u_y^{00} + H'u_y^{H0}) // H \\
u_x^{LB} &= ((V - V')u_x^{00} + V'u_x^{0V}) // V & u_y^{LB} &= ((V - V')u_y^{00} + V'u_y^{0V}) // V \\
u_x^{RB} &= ((V - V')((H - H')u_x^{00} + H'u_x^{H0}) + V'((H - H')u_x^{0V} + H'u_x^{HV})) // (HV) \\
u_y^{RB} &= ((V - V')((H - H')u_y^{00} + H'u_y^{H0}) + V'((H - H')u_y^{0V} + H'u_y^{HV})) // (HV)
\end{aligned}$$

donde "//" representa división con resultado entero con redondeo del cociente al entero más próximo y con redondeo de los valores de semienteros a partir de 0.

En lo que queda de este anexo se supone que los centros de los píxels están en los puntos  $(x, y) = (i + \frac{1}{2}, j + \frac{1}{2})$  para los campos de luminancia y crominancia. Los parámetros enteros  $i, j$  se definen como sigue:

- $i = 0, \dots, H - 1$  y  $j = 0, \dots, V - 1$  para luminancia, y
- $i = 0, \dots, \frac{H}{2} - 1$  y  $j = 0, \dots, \frac{V}{2} - 1$  para crominancia.

Lo anterior significa que se utilizan distintos sistemas de coordenadas para luminancia y crominancia, como se muestra en la figura P.3. Si se utiliza el sistema de coordenadas para crominancia, los parámetros enteros  $u_x^{LT}$ ,  $u_y^{LT}$ ,  $u_x^{RT}$ ,  $u_y^{RT}$ ,  $u_x^{LB}$ ,  $u_y^{LB}$ ,  $u_x^{RB}$ , y  $u_y^{RB}$  definidos anteriormente pueden considerarse como los desplazamientos  $x$  e  $y$  del campo de crominancia en los puntos virtuales  $(x, y) = (0, 0)$ ,  $(H'/2, 0)$ ,  $(0, V'/2)$ , y  $(H'/2, V'/2)$  con una exactitud de  $\frac{1}{64}$  de píxel (los desplazamientos reales se obtienen dividiendo estos valores por 64). Con estos parámetros y un parámetro adicional  $S$ , que es 2 para la luminancia y 1 para la crominancia, se define el algoritmo de repetición de muestreo de los campos de luminancia y crominancia utilizando ecuaciones comunes.



**Figura P.3/H.263 – Sistemas de coordenadas para campos de luminancia y crominancia**

Los parámetros enteros  $u_x^L(j)$ ,  $u_y^L(j)$ ,  $u_x^R(j)$ , y  $u_y^R(j)$  que representan los desplazamientos  $x$  e  $y$  del campo de la imagen en  $(x, y) = (0, j + \frac{1}{2})$  y  $(SH'/2, j + \frac{1}{2})$  con una exactitud de  $\frac{S}{64}$  de píxel (los desplazamientos reales se obtienen dividiendo estos valores por  $64/S$ ) se definen utilizando una interpolación lineal dimensional como sigue:

$$u_x^L(j) = ((SV' - 2j - 1)u_x^{LT} + (2j + 1)u_x^{LB}) // (SV') \quad u_y^L(j) = ((SV' - 2j - 1)u_y^{LT} + (2j + 1)u_y^{LB}) // (SV')$$

$$u_x^R(j) = ((SV' - 2j - 1)u_x^{RT} + (2j + 1)u_x^{RB}) // (SV') \quad u_y^R(j) = ((SV' - 2j - 1)u_y^{RT} + (2j + 1)u_y^{RB}) // (SV')$$

donde "//" representa división con resultado entero con redondeo del cociente al entero más próximo y con redondeo de los valores de semienteros a partir de 0.

Finalmente, los parámetros que especifican la posición transformada en la imagen de referencia son:

$$\begin{aligned}
 I_R(i, j) &= Pi + \left( (SH' - 2i - 1)u_x^L(j) + (2i + 1)u_x^R(j) + 32H'/P \right) \text{ /// } (64H'/P) \\
 J_R(i, j) &= Pj + \left( (SH' - 2i - 1)u_y^L(j) + (2i + 1)u_y^R(j) + 32H'/P \right) \text{ /// } (64H'/P) \\
 i_R(i, j) &= I_R(i, j) \text{ /// } P & j_R(i, j) &= J_R(i, j) \text{ /// } P \\
 \emptyset_x &= I_R(i, j) - \left( I_R(i, j) \text{ /// } P \right) P & \emptyset_y &= J_R(i, j) - \left( J_R(i, j) \text{ /// } P \right) P
 \end{aligned}$$

donde:

"///" división con resultado entero con truncamiento hacia el infinito negativo,

"/" división con resultado entero (en este caso no hay pérdida de exactitud),

$P$  exactitud de los desplazamientos  $x$  e  $y$  ( $P=2$  cuando WDA = "10" y  $P=16$  cuando WDA = "11" o está ausente, véase P.2.1 para la definición de WDA),

$\left( \frac{I_R(i, j)}{P} + \frac{1}{2}, \frac{J_R(i, j)}{P} + \frac{1}{2} \right)$  ( $x, y$ ) ubicación de la posición transformada  $I_R(i, j)$  y  $J_R(i, j)$  son enteros),

$\left( i_R(i, j) + \frac{1}{2}, j_R(i, j) + \frac{1}{2} \right)$  ( $x, y$ ) ubicación del punto de la muestra próximo a la posición transformada, ( $i_R(i, j)$  y  $j_R(i, j)$  son enteros),

$(\emptyset_x, \emptyset_y)$  coeficientes de interpolación bilineal de la posición transformada ( $\emptyset_x$  y  $\emptyset_y$  son enteros).

El cálculo de esta ecuación puede simplificarse sustituyendo las divisiones por operaciones de desplazamiento, dado que  $64H'/P = 2^{m+2}$  cuando  $P = 16$  y  $64H'/P = 2^{m+5}$  cuando  $P = 2$ .

Con estos parámetros, se obtiene el valor de muestra,  $E_P(i, j)$ , del píxel ubicado en  $(x, y) = \left( i + \frac{1}{2}, j + \frac{1}{2} \right)$  en la imagen remuestreada aplicando la interpolación bilineal como sigue:

$$\begin{aligned}
 E_P(i, j) &= \left( (P - \emptyset_y) \left( (P - \emptyset_x) E_R(i_R, j_R) + \emptyset_x E_R(i_R + 1, j_R) \right) \right. \\
 &\quad \left. + \emptyset_y \left( (P - \emptyset_x) E_R(i_R, j_R + 1) + \emptyset_x E_R(i_R + 1, j_R + 1) \right) + P^2 / 2 - 1 + RCRPR \right) / P^2
 \end{aligned}$$

donde "/" representa división por truncamiento.  $i_r$  y  $j_r$  son notaciones simplificadas de  $i_R(i, j)$  y  $j_R(i, j)$ , y  $E_R(i_R, j_R)$  indica el valor de muestra del píxel ubicado en  $(x, y) = \left( i_R + \frac{1}{2}, j_R + \frac{1}{2} \right)$  en la imagen de referencia tras la extrapolación utilizando el modo de relleno adecuado, si es necesario. El valor del parámetro RCRPR se define como sigue:

- Para una imagen B (o la parte B de una trama PB mejorada) que tiene una imagen P como imagen de anclaje temporalmente subsiguiente, RCRPR es igual al bit del tipo de redondeo (RTYPE) en MPPTYPE (véase 5.1.4.3) de esta imagen P temporalmente subsiguiente. Esto significa que, en el caso de tramas PB mejoradas, RCRPR tiene el mismo valor para la parte P y la parte B.
- Para otros tipos de imágenes, RCRPR es igual al bit RTYPE de la imagen vigente.

## P.4 Ejemplo de implementación

En esta subcláusula se da un ejemplo de implementación del algoritmo descrito en la subcláusula precedente como pseudocódigo.

### P.4.1 Desplazamientos de los puntos virtuales

Cuando se codifica una imagen grande, es posible que la implementación directa de la ecuación para la obtención de los parámetros  $u_x^{RB}$  y  $u_y^{RB}$  indicados en P.3 exija la utilización de variables que requieran más de 32 bits para su representación binaria. A continuación se muestra un ejemplo de algoritmo que no requiere variables con más de 32 bits en el cálculo de  $u_x^{RB}$  y  $u_y^{RB}$ , para sistemas que no pueden utilizar fácilmente enteros de 64 bits o registros de coma flotante.

Puesto que  $H$ ,  $V$ ,  $H'$  y  $V'$  son divisibles por 4, las definiciones de  $u_x^{RB}$  pueden reformularse como sigue:

$$u_x^{RB} = \left( (V_Q - V_Q') \left( (H_Q - H_Q') u_x^{00} + H_Q' u_x^{H0} \right) + V_Q' \left( (H_Q - H_Q') u_x^{0V} + H_Q' u_x^{HV} \right) \right) // A$$

donde  $H_Q = H/4$ ,  $V_Q = V/4$ ,  $H_Q' = H'/4$ ,  $V_Q' = V'/4$ ,  $A = H_Q V_Q$ , y  $///$  indica división con resultado entero con redondeo del cociente al entero más próximo y con redondeo de los valores de semienteros a partir de 0. A continuación, se definen los parámetros  $T_T$  y  $T_B$  como sigue:

$$T_T = (H_Q - H_Q') u_x^{00} + H_Q' u_x^{H0}$$

$$T_B = (H_Q - H_Q') u_x^{0V} + H_Q' u_x^{HV}$$

para hacer más sencilla la descripción. Utilizando el operador  $///$  que indica división con resultado entero con truncamiento hacia el infinito negativo, y el operador  $%$  que se define como un  $a \% b = a - (a /// b) b$ , puede obtenerse el valor de  $u_x^{RB}$  mediante el siguiente pseudocódigo:

```

q = (V_Q - V_Q') * (T_T /// A) + V_Q' * (T_B /// A) + ((V_Q - V_Q') * (T_T % A) + V_Q' * (T_B % A)) /// A;
r = ((V_Q - V_Q') * (T_T % A) + V_Q' * (T_B % A)) % A;
if (q < 0)
    u_x^{RB} = q + (r + (A-1) / 2) / A;
else
    u_x^{RB} = q + (r + A / 2) / A;

```

El valor de  $u_y^{RB}$  puede también calcularse utilizando este algoritmo.

### P.4.2 Algoritmo de repetición de muestreo

Para simplificar la descripción del algoritmo, se define una función `prior_sample` (muestra anterior). Su finalidad es generar un valor de píxel para cualquier ubicación de entero  $(i_p, j_p)$  relativa a la cuadrícula de muestreo de la imagen de referencia anterior:

```

clip(x_min, x, x_max) {
    if (x < x_min) {
        return x_min;
    } else if (x > x_max) {
        return x_max;
    } else {
        return x;
    }
}

```

```

prior_sample (ip, jp) {
    if (FILL_MODE = clip) {
        ic = clip (0, ip, S*HR /2-1);
        jc = clip (0, jp, S*VR /2-1);
        return prior_ref[ic, jc];
    } else {
        if ((ip < 0) OR (ip > S*HR /2-1) or (jp < 0) OR (jp > S*VR /2-1) {
            return fill_value;
        } else {
            return prior_ref[ip, jp];
        }
    }
}

```

En el pseudocódigo, `prior_ref [i, j]` indica la muestra de la columna  $i$  y fila  $j$  en la imagen de referencia temporalmente previa.

A continuación, se define una función de filtro que implementa la interpolación bilineal descrita en P.3. Se supone que todos los argumentos de la siguiente función son enteros y que los coeficientes de la interpolación bilineal  $\emptyset_x$  y  $\emptyset_y$  se cuantifican en la gama  $0, \dots, P-1$  (inclusive).

```

filter(x0, y0,  $\emptyset_x$ ,  $\emptyset_y$ ) {
    return [(P- $\emptyset_y$ )*((P- $\emptyset_x$ )*prior_sample(x0, y0)+ $\emptyset_x$ *prior_sample(x0+1, y0))+
             $\emptyset_y$ *((P- $\emptyset_x$ )*prior_sample(x0, y0+1)+ $\emptyset_x$ *prior_sample(x0+1, y0+1))+
            P2/2-1+RCRPR]/P2;
}

```

Por último, se puede especificar el método de alabeo de la imagen de referencia para generar una predicción de la imagen vigente en términos de estas funciones. Los píxeles de la imagen de predicción puede generarse en un orden de barrido de trama. Se supone que los valores  $u_x^L(j)$ ,  $u_y^L(j)$ ,  $u_x^R(j)$ , y  $u_y^R(j)$  ya están calculados y cargados en variables  $u_x^L$ ,  $u_y^L$ ,  $u_x^R$ , y  $u_y^R$ . Definiendo el parámetro  $D$  como  $D = 64H'/P$  y teniendo en cuenta que  $H' = 2^m$ , se obtienen los valores de muestra de los píxeles de la línea  $j$  del campo remuestreado (la línea del extremo superior se define como la línea 0) mediante el pseudocódigo siguiente:

```

axi = D * P + 2 * (uxR - uxL);
ayi = 2 * (uyR - uyL);
ax = uxL * S * 2m + (uxR - uxL); + D / 2;
ay = j * D * P + uyL * S * 2m + (uyR - uyL); + D / 2;
for (i = 0; i < S * H / 2; i++) {
    IR = ax /// D;
    JR = ay /// D;
    iR = IR /// P;
    jR = JR /// P;
     $\emptyset_x$  = IR - (iR * P);
     $\emptyset_y$  = JR - (jR * P);
    new_ref[i, j] = filter(iR, jR,  $\emptyset_x$ ,  $\emptyset_y$ );
    ax += axi;
    ay += ayi;
}

```

donde todas las variables utilizadas son variables enteras y `new_ref [i, j]` indica la muestra generada para la columna  $i$  y la fila  $j$  de la imagen de referencia remuestreada. De acuerdo con la definición de los parámetros, todas las divisiones de este código pueden sustituirse por operaciones de desplazamiento binarias. Por ejemplo, cuando  $P=16$ :

```

IR = ax /// D;
JR = ay /// D;
iR = IR /// P;
jR = JR /// P;
Øx = IR - (iR * P);
Øy = JR - (jR * P);

```

Pueden reformularse, suponiendo que  $a_x$ ,  $a_y$ ,  $I_R$  y  $J_R$  son variables enteras con codificación binaria en representación de complemento a 2 como sigue:

```

IR = ax >> (m+2);
JR = ay >> (m+2);
iR = IR >> 4;
jR = JR >> 4;
Øx = IR & 15;
Øy = JR & 15;

```

donde " $>> N_{shift}$ " indica un desplazamiento binario aritmético a la derecha por  $N_{shift}$  de bits ( $N_{shift}$  es un entero positivo), y "&" indica una operación lógica AND.

## P.5 Repetición de muestreo con factor de 4

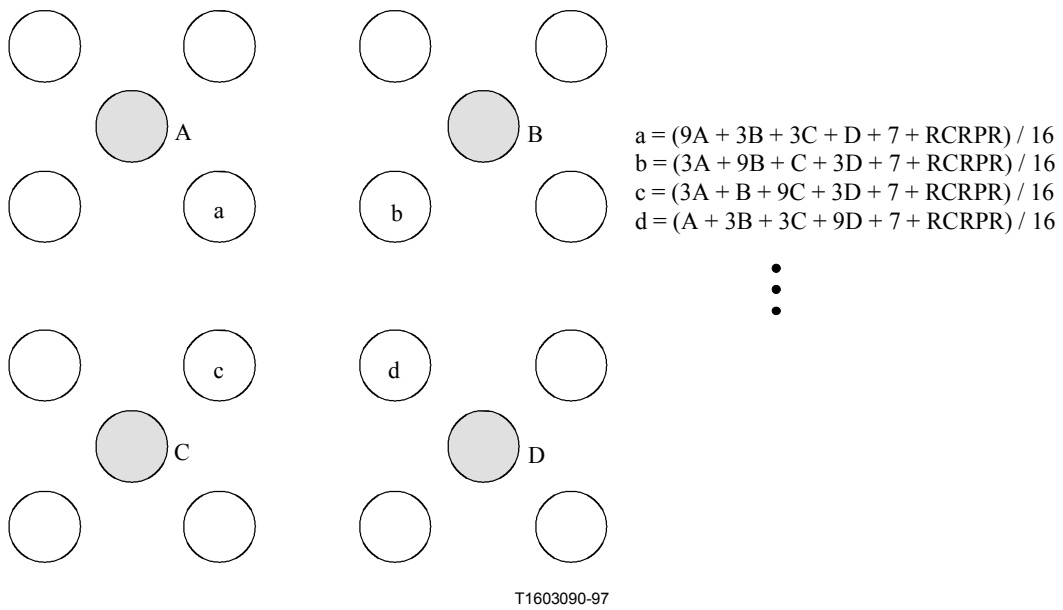
La repetición del muestreo con factor de 4, que convierte las dimensiones horizontal y vertical de la imagen con un factor de 2 o de  $\frac{1}{2}$ , es un caso especial del algoritmo de repetición de muestreo descrito en P.3. En esta subcláusula se da la descripción simplificada del algoritmo de repetición de muestreo para ese caso especial.



El valor del parámetro RCRPR utilizado en las figuras P.4 a P.6 viene determinado por el bit del tipo de redondeo (RTYPE) de MPPTYPE (véase 5.1.4.3) como se describe en P.3. Además, en las figuras siguientes, "/" indica división por truncamiento.

### P.5.1 Muestreo a velocidad superior con factor de 4

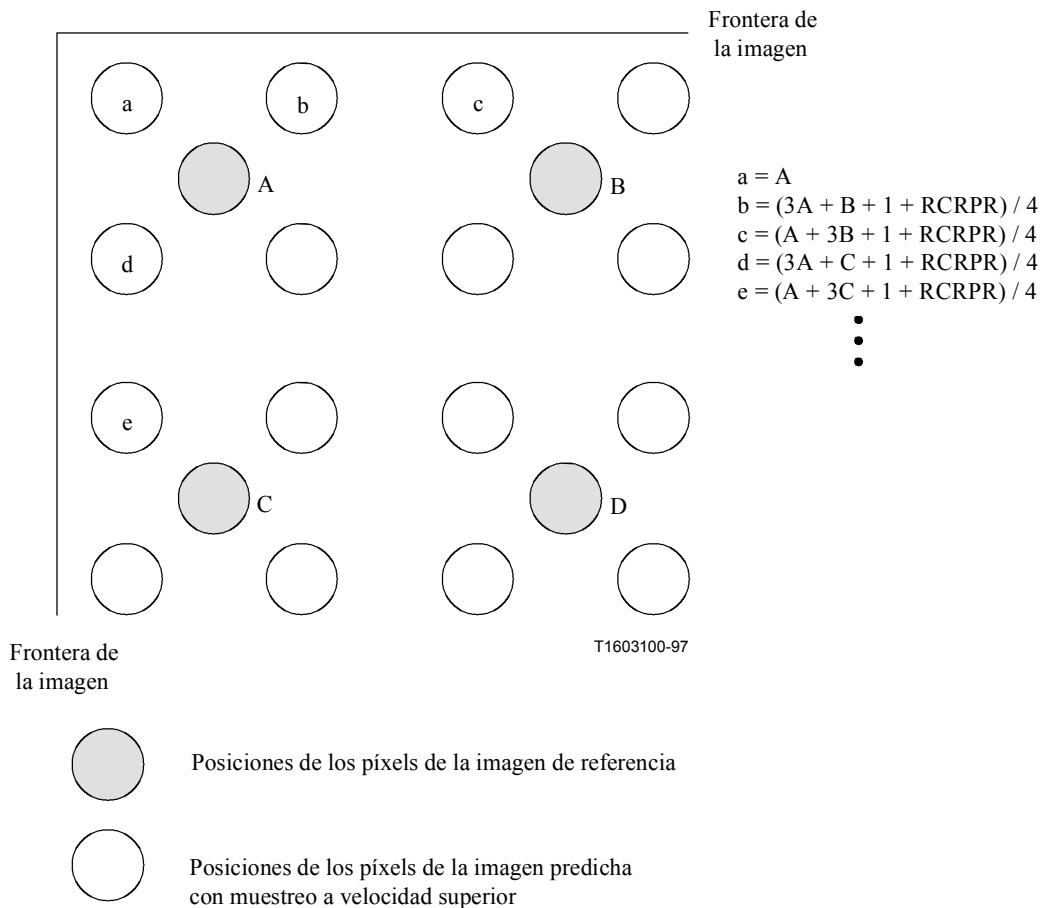
En la figura P.4 se muestra el método de interpolación de valores de píxeles utilizado en el muestreo a velocidad superior con factor de 4 para píxeles internos. Suponiendo que existen píxeles fuera de la imagen de acuerdo con el modo de relleno seleccionado (véanse P.2.3 y P.2.4), se aplica el mismo método de interpolación para los píxeles de la frontera de la imagen. En la figura P.5 se muestra el método de interpolación para los píxeles de la frontera de la imagen cuando se selecciona *clip* (recorte) como modo de relleno. Puesto que un muestreo preciso a velocidad superior con factor de 4 requiere desplazamientos  $x$  e  $y$  con una exactitud de por lo menos  $\frac{1}{4}$  de píxel, el campo de exactitud del desplazamiento del alabeo (WDA) de P.2.1 debe fijarse a "11" o debe invocarse implícitamente la repetición del muestreo para utilizar este método de muestreo a velocidad superior.





-  Posiciones de los píxels de la imagen de referencia
-  Posiciones de los píxels de la imagen de predicha con muestreo a velocidad superior

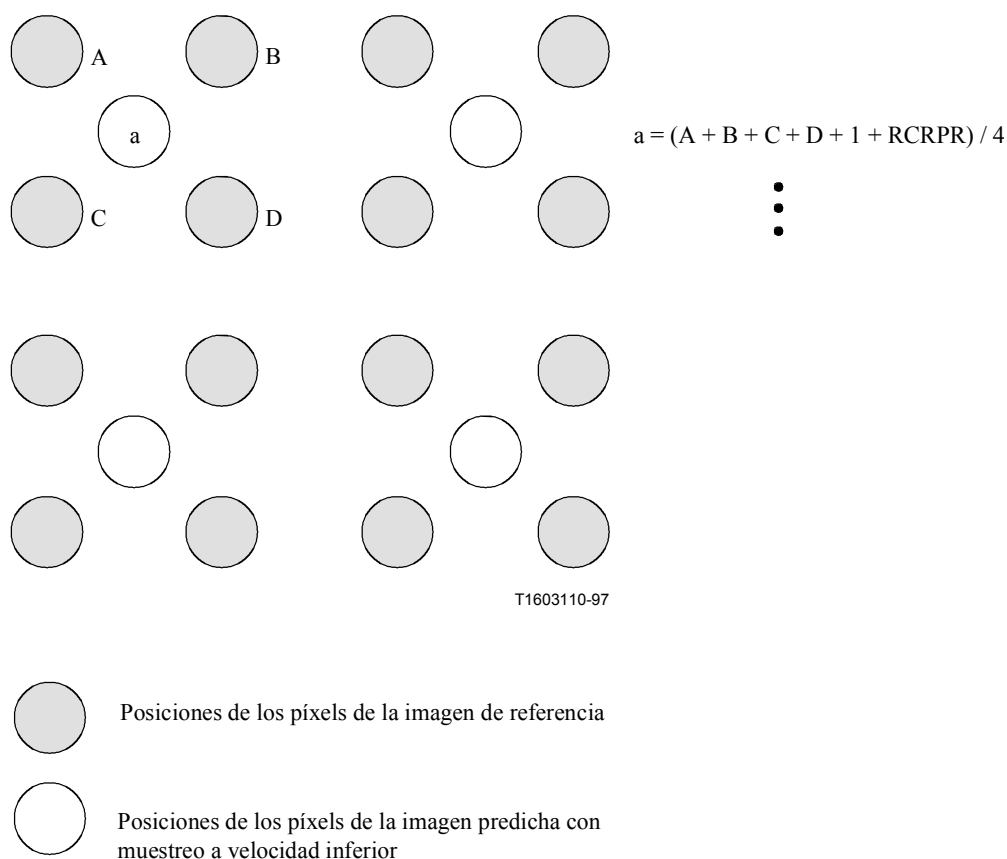
**Figura P.4/H.263 – Muestreo a velocidad superior con factor de 4 para píxels situados dentro de la imagen**



**Figura P.5/H.263 – Muestreo a velocidad superior con factor de 4 para píxels en la frontera de la imagen (modo de recorte = *recorte*)**

**P.5.2 Muestreo a velocidad inferior con factor de 4**

En la figura P.6 se muestra el método de interpolación de valores de píxels en el muestreo a velocidad inferior con factor de 4. Puesto que los desplazamientos x e y con una exactitud de  $\frac{1}{2}$  píxel son suficientes para un muestreo preciso a velocidad inferior con factor de 4, se aceptan "10" y "11" como valores del campo de exactitud del desplazamiento (WDA) (si está presente) especificado en P.2.1.



**Figura P.6/H.263 – Muestreo a velocidad inferior con factor de 4**

## Anexo Q

### Modo actualización de resolución reducida

#### Q.1 Introducción

Este anexo describe un modo actualización de resolución reducida optativo de la presente Recomendación. La capacidad de este modo se señala por medios externos (por ejemplo, aplicando la Rec. UIT-T H.245). La utilización de este modo se indica en el campo PLUSPTYPE del encabezamiento de imagen.

Se prevé que el modo actualización de resolución reducida se utilice al codificar una escena extremadamente activa; este modo permite aumentar la velocidad de codificación de la imagen al mismo tiempo que mantiene una calidad subjetiva suficiente. Con este modo el codificador puede enviar información actualizada para una imagen que está codificada con resolución reducida, conservando al mismo tiempo los detalles en una imagen de referencia con resolución más alta para crear una imagen final con mayor resolución.

La sintaxis del tren de bits en este modo es idéntica a la sintaxis para la codificación sin este modo, pero la semántica, o interpretación del tren de bits, es en cierto modo distinta. En este modo, la porción de la imagen abarcada por un macrobloque es dos veces más ancha y dos veces más alta. Por ello, hay aproximadamente un cuarto de los macrobloques que habría sin este modo. Los datos del vector de movimiento corresponden también a bloques con una altura y una anchura dobles de las normales, o sea de  $32 \times 32$  y  $16 \times 16$ , en vez de las  $16 \times 16$  y  $8 \times 8$  normales. Por otra parte, debe considerarse que la DCT o los datos sobre textura describen bloques de  $8 \times 8$  en una versión

de la imagen con resolución reducida. Para producir la imagen final, los datos sobre textura se decodifican con una resolución reducida y luego se efectúa un muestro a velocidad superior con resolución total de la imagen. Una vez efectuado el muestreo a velocidad superior, la imagen de textura con resolución total se añade a la imagen con movimiento compensado (ya con resolución total) para crear la imagen de visualización y futura referencia.

En este modo, una imagen que tiene una dimensión horizontal  $H$  y una dimensión vertical  $V$  indicados en el encabezamiento de imagen, se crea como imagen de visualización final.

En este modo, una imagen referenciada utilizada para predicción y creada para futura decodificación tiene una dimensión horizontal  $H_R$  y una dimensión vertical  $V_R$ , que son las mismas que en el modo por defecto definido en 4.1. Es decir,  $H_R$  y  $V_R$  son:

$$H_R = ((H + 15) / 16) * 16$$

$$V_R = ((V + 15) / 16) * 16$$

donde  $H$  y  $V$  son las dimensiones horizontal y vertical, respectivamente, indicadas en el encabezamiento de la imagen y "/" significa división por truncamiento.

En este anexo, la textura se codifica entonces a una resolución reducida con altura y anchura de  $H_C$  y  $V_C$  donde:

$$H_C = ((H_R + 31) / 32) * 32$$

$$V_C = ((V_R + 31) / 32) * 32$$

y "/" significa división por truncamiento.

Si  $H_C$  y  $H_R$ , o  $V_C$  y  $V_R$  no son idénticas entre sí, como ocurre en el formato cuarto de CIF, se efectúa una ampliación de la imagen referenciada, y la imagen se decodifica en la misma manera que si la anchura y la altura fueran  $H_C$  y  $V_C$ . La imagen resultante, enlosada en macrobloques de  $32 * 32$ , se recorta a la derecha y en la parte inferior hasta una anchura de  $H_R$  y una altura de  $V_R$ , y a continuación se almacena como imagen de referencia de una futura decodificación.

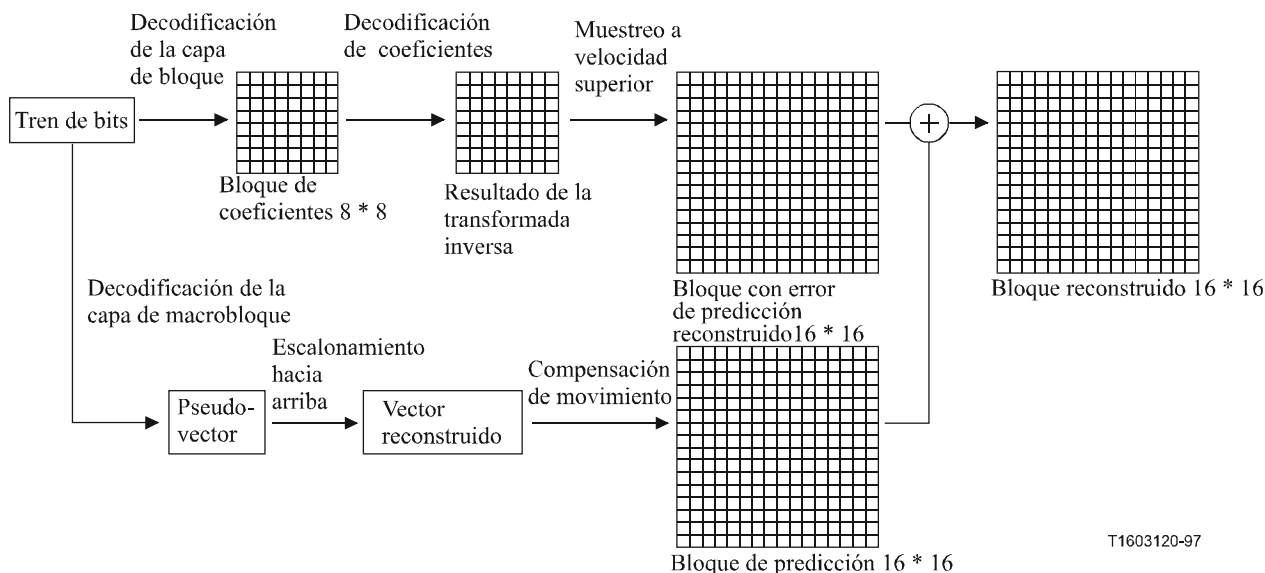
Si  $H$  y  $V$  son las mismas que las de una imagen resultante cuya anchura es  $H_C$  y cuya altura es  $V_C$ , esta imagen resultante se utiliza para visualización. De lo contrario, la imagen resultante se recorta de nuevo hasta un tamaño de  $H * V$  y en este caso se utiliza únicamente a efectos de visualización.

Si el modo escalabilidad temporal, SNR y espacial (anexo O) o el modo repetición de muestreo de la imagen de referencia (anexo P) se utiliza también con esta opción, el formato de fuente de la imagen vigente podría ser distinto del de la imagen de referencia. En tal caso, la repetición del muestreo de la imagen de referencia se efectuará antes de la decodificación.

NOTA – Este modo puede ser utilizado con el modo selección de la imagen de referencia (véase el anexo N) sin modificación, ya que la imagen de referencia (después de la posible repetición del muestreo efectuada por el modo repetición de muestreo de la imagen de referencia) tiene el mismo tamaño que se indica en el encabezamiento de la imagen vigente cuando se utiliza esta opción.

## **Q.2 Procedimiento de decodificación**

La figura Q.1 muestra el diagrama de bloques de la decodificación de bloques del modo actualización de resolución reducida.



**Figura Q.1/H.263 – Diagrama de bloques de la decodificación de bloques del modo actualización de resolución reducida**

El procedimiento de decodificación se describe en las subcláusulas siguientes:

### Q.2.1 Preparación de la imagen de referencia

En algunos casos, la imagen de referencia disponible tiene dimensiones distintas de  $H_c$  y  $V_c$ . La imagen de referencia se convertirá entonces antes del procedimiento de decodificación, según lo indicado en Q.2.1.1 o Q.2.1.2.

#### Q.2.1.1 Repetición del muestreo de la imagen de referencia

Si el modo escalabilidad temporal, SNR y espacial (anexo O) o el modo repetición de muestreo de la imagen de referencia (anexo P) se utiliza también con esta opción, el formato de fuente de la imagen vigente podría ser distinto del de la imagen de referencia. En este caso, se efectuará primero la repetición de muestreo de la imagen de referencia según cada anexo.

#### Q.2.1.2 Ampliación de la imagen de referencia

Si  $H_r$  o  $V_r$  no son divisibles por 32, como ocurre con el formato cuarto de CIF, se amplía la imagen de referencia. En Q.3 se expone en detalle el procedimiento de esta ampliación.

### Q.2.2 Decodificación de la capa de macrobloque

Puede considerarse que la decodificación funciona en bloques "ampliados" de un tamaño de  $32 \times 32$  en luminancia y de  $16 \times 16$  en crominancia. Los datos sobre textura y movimiento de cada bloque ampliado se decodifican para crear un bloque de movimiento de  $32 \times 32$  y un bloque de textura de  $32 \times 32$ , como se indica en Q.2.2.1 y Q.2.2.2, respectivamente. Los bloques de movimiento y textura se añaden a continuación, como se describe en Q.2.2.3.

#### Q.2.2.1 Compensación del movimiento

Primero se forma, cada componente del vector de movimiento por macrobloque (o cuatro vectores de movimiento por macrobloque) a partir de MVD (y posiblemente  $MVD_{2-4}$ ). Si se utiliza el modo tramas PB mejoradas, se forman también  $MV_F$  y  $MV_B$  para la imagen B a partir de  $MV_{DB}$ . El procedimiento detallado de esta formación del vector de movimiento se describe en Q.4. Si el modo de la imagen vigente es una imagen B o imagen EP, el vector de movimiento hacia adelante y hacia atrás se obtiene también según lo indicado en Q.4.

El vector de movimiento para ambos bloques de crominancia del macrobloque se obtiene a partir del vector de movimiento por macrobloque, según lo indicado en 6.1.1. Si se utiliza el modo predicción anticipada o el modo filtro de desbloqueo y se han definido, por tanto, cuatro vectores de movimiento para el macrobloque, el vector de movimiento para ambos bloques de crominancia se obtiene a partir de los cuatro vectores de movimiento, según lo indicado en F.2. Si se utiliza el modo tramas PB mejoradas, la creación del vector de crominancia se especifica en el anexo M. Si se utiliza una imagen B o una imagen EP, la creación del vector de crominancia se especifica en el anexo O.

A continuación se forma una predicción a partir del vector de movimiento para un macrobloque INTER. Del vector de movimiento por macrobloque se obtienen cuatro bloques de predicción de luminancia de  $16 \times 16$ , y del vector de movimiento de crominancia se obtienen dos bloques de predicción de crominancia de  $16 \times 16$ . Por lo que se refiere a la interpolación para la predicción de subpíxels, véase 6.1.2. Si se utiliza también el modo predicción avanzada, se efectúa una compensación del movimiento con superposición ampliada para obtener cuatro bloques de predicción de luminancia de  $16 \times 16$  utilizando matrices de ponderación ampliadas, para lo que se define un procedimiento detallado en Q.5. Si el modo de la imagen vigente es trama PB mejorada, imagen B, o imagen EP, la predicción se obtiene según lo indicado en los demás anexos pertinentes, con la salvedad de que el tamaño de los bloques predichos es de  $16 \times 16$  y no de  $8 \times 8$ .

#### **Q.2.2.2 Decodificación de textura**

Primero se decodifica el tren de bits de la capa de bloque según lo indicado en 5.4. Luego se decodifican los coeficientes y se obtienen bloques con error de predicción reconstruidos con resolución reducida de  $8 \times 8$  como resultado de la transformada inversa, según lo indicado en 6.2.

A continuación se obtienen los bloques con error de predicción reconstruidos de  $16 \times 16$  efectuando el muestreo a velocidad superior de los bloques con error de predicción reconstruidos con resolución reducida de  $8 \times 8$ . Para la creación de los píxels de borde en cada bloque con error de predicción reconstruido de  $16 \times 16$ , se utilizan únicamente los píxels que pertenecen al bloque correspondiente. En Q.6 se describe el procedimiento detallado.

#### **Q.2.2.3 Reconstrucción de bloque**

Para cada bloque de luminancia y crominancia, se efectúa la suma de la predicción y el error de predicción. El procedimiento es idéntico al indicado en 6.3.1, con la salvedad de que el tamaño de los bloques es de  $16 \times 16$  y no de  $8 \times 8$ . Luego se efectúa el recorte, según lo indicado en 6.3.2.

A continuación se aplica un filtro de límite de bloque a los píxels de límites de los bloques reconstruidos de  $16 \times 16$ . En Q.7 se describe el procedimiento detallado.

#### **Q.2.3 Almacenamiento de la imagen**

Si  $H_r$  y  $V_r$  son divisibles por 32, como ocurre con el formato CIF, la imagen resultante reconstruida tal como se describe en Q.2.2, se almacena como imagen de referencia tal cual es para una futura decodificación. Si no son divisibles por 32, como ocurre con el formato cuarto de CIF, la imagen reconstruida, que sólo está cubierta con macrobloques de  $32 \times 32$ , se recoge a la derecha y al fondo con anchura de  $H_r$  y altura de  $V_r$ , y se almacena como imagen de referencia de una futura decodificación.

#### **Q.2.4 Visualización**

Si  $H$  y  $V$  son iguales a  $H_c$  y  $V_c$ , se utiliza la imagen resultante de Q.2.2 tal cual es a efectos de visualización. De lo contrario, esta imagen resultante se recoge de nuevo con un tamaño de  $H \times V$  y se utiliza únicamente a efectos de visualización.

### Q.3 Ampliación de la imagen referenciada

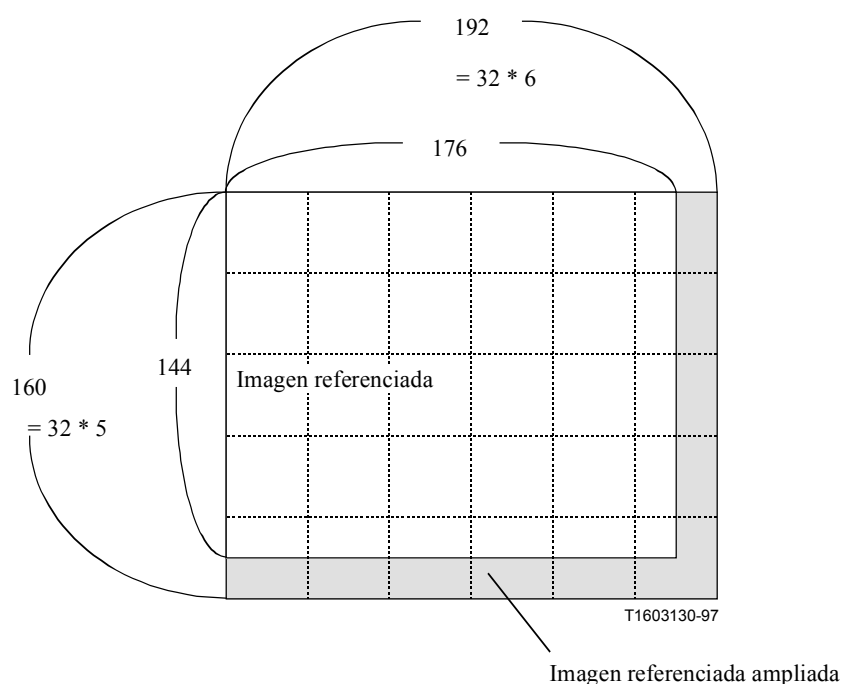
Si  $H_R$  o  $V_R$  no son divisibles por 32, como ocurre con el formato cuarto de CIF, se efectúa la ampliación de la imagen referenciada antes de la decodificación de la capa de macrobloque/bloque. La anchura y la altura de la imagen de referencia ampliada para luminancia tienen las dimensiones siguientes mayores que sean divisibles por 32, y las correspondientes a la crominancia, las dimensiones siguientes mayores que sean divisibles por 16.

NOTA – La anchura y la altura de la imagen referenciada en el modo por defecto se amplían siempre para que sean divisibles por 16 incluso si los formatos de la imagen tienen una anchura o una altura que no son divisibles por 16, porque la imagen deberá decodificarse como si la anchura o altura tuvieran la dimensión siguiente mayor que fuese divisible por 16. Véase 4.1.

Si no se utiliza el modo vector de movimiento sin restricción, ni el de predicción avanzada ni el de filtro de desbloqueo con esta opción, los píxeles ampliados pueden ser unos valores cualesquiera, ya que no se utilizarán nunca como píxeles de referencia de la imagen decodificada que ha de ser reconstruida y visualizada.

Si se utiliza también el modo vector de movimiento sin restricción, o el de predicción avanzada o el de filtro de desbloqueo con esta opción, la ampliación de la imagen referenciada se efectúa duplicando el píxel de borde de la imagen referenciada, para asegurar la decodificación cuando los vectores de movimiento apuntan fuera del borde derecho e inferior de la imagen.

Por ejemplo, si se utiliza el modo actualización de resolución reducida para un cuarto de CIF, la anchura de la imagen referenciada es de 176 y la altura es de 144, que no son divisibles por 32. Para cubrir una imagen cuarto de CIF con macrobloques de tamaño  $32 \times 32$ , el número de filas de macrobloques debe ser 6 y el número de columnas de macrobloques debe ser 5. Por consiguiente, la anchura de la imagen referenciada ampliada es de 192 y su altura, 160.



**Figura Q.2/H.263 – Ampliación de la imagen referenciada para el tamaño de imagen cuarto de CIF**

La figura Q.2 ilustra la ampliación de la imagen referenciada en cuarto de CIF. La imagen referenciada ampliada para luminancia viene dada por la siguiente fórmula:

$$R_{RRU}(x, y) = R(x', y')$$

donde:

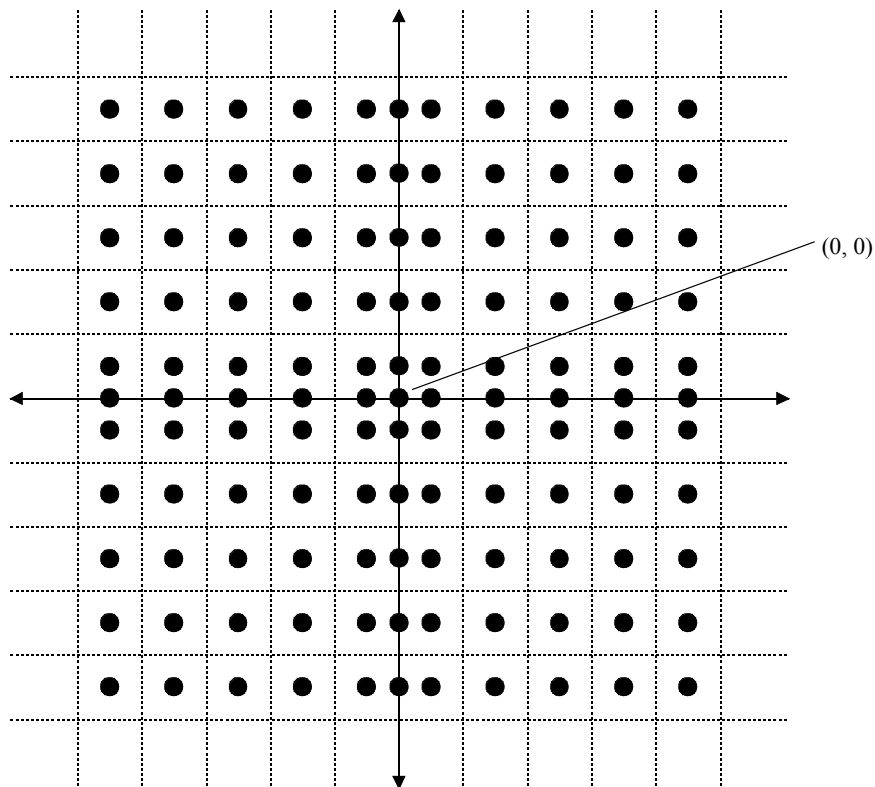
$$\begin{aligned}x, y &= \text{coordenadas espaciales de la imagen referenciada ampliada en el dominio de píxels,} \\x', y &= \text{coordenadas espaciales de la imagen referenciada en el dominio de píxels,} \\R_{RRU}(x, y) &= \text{valor de píxel de la imagen referenciada ampliada en } (x, y), \\R(x', y') &= \text{valor de píxel de la imagen referenciada en } (x', y'), \\x' &= \begin{cases} 175 & \text{si } x > 175 \text{ y } x < 192 \\ x & \text{en los demás casos,} \end{cases} \\y' &= \begin{cases} 143 & \text{si } y > 143 \text{ e } y < 160 \\ y & \text{en los demás casos.} \end{cases}\end{aligned}$$

Las imágenes referenciadas para crominancia se amplían de la misma manera.

#### **Q.4 Reconstrucción de vectores de movimiento**

En el modo actualización de resolución reducida, la gama del vector de movimiento se amplía hasta aproximadamente el doble en la dirección horizontal y vertical. Para realizar una gama ampliada utilizando la tabla de VLC para MVD definida en el cuadro 14, se limita a cada componente de vector a tener sólo un valor de medio elemento de imagen o cero. Por consiguiente, la gama de cada componente del vector de movimiento es  $[-31,5, 30,5]$  en el modo actualización de resolución reducida por defecto. Si se utiliza el modo vector de movimiento sin restricción, la gama del vector  $[-\text{límite}, \text{límite}]$  definida en D.2 se aplica a los vectores de pseudomovimiento y se convierte en  $[-(2 * \text{límite}-0,5), 2 * \text{límite}-1,5]$  para los vectores de movimiento. Para CIF esto significa que la gama del vector de pseudomovimiento es  $[-32, 31,5]$  y la gama del vector de movimiento es  $[-63,5, 62,5]$ . Si el campo UUI se fija a "01" los vectores de movimiento no están limitados. No obstante, los vectores de movimiento (no sólo los vectores de pseudomovimiento) están siempre limitados a apuntar a no más de 15 píxels fuera de la zona codificada, como se describe en D.1.1. La figura Q.3 ilustra las posibles posiciones del vector de movimiento por macrobloque o las cuatro predicciones del vector de movimiento en torno al valor del vector de (0, 0). Las líneas de puntos indican las coordenadas enteras.





T1603140-97

**Figura Q.3/H.263 – Reconstrucción de vector de movimiento**

Para el macrobloque que utiliza vectores de movimiento diferenciales en una imagen B, el vector de movimiento de la predicción hacia adelante y hacia atrás se obtiene independientemente. En el modo actualización de resolución reducida, el componente del vector de movimiento  $MV_C$  para un bloque de luminancia se reconstruye a partir de  $MVD$  y  $MVD_{2-4}$ , como sigue:

- 1) El componente del vector de pseudopredicción **pseudo- $P_C$**  se crea a partir del componente del vector de predicción  $P_C$ .

$$\text{pseudo-}P_C = 0 \quad \text{si } P_C = 0$$

$$\text{pseudo-}P_C = \text{sign}(P_C) * (|P_C| + 0,5) / 2,0 \quad \text{si } P_C \neq 0$$

"/" significa división con coma flotante (sin pérdida de exactitud). El componente del vector de predicción  $P_C$  se define como el valor mediano de los componentes de vector  $MV_1$ ,  $MV_2$  y  $MV_3$ , definidos en 6.1.1 y F.2.

- 2) El componente del vector de pseudomacrobloque **pseudo- $MV_C$**  se obtiene añadiendo las diferencias del vector de movimiento  $MVD$  (y  $MVD_{2-4}$ ) del cuadro 14 al pseudo- $P_C$ .

En el modo actualización de resolución reducida por defecto, el valor del **pseudo- $MV_C$**  está limitado a la gama  $[-16, 15,5]$ . Sólo un valor del par producirá un pseudo- $MV_C$  que quede dentro de la gama permitida. El procedimiento se efectúa de manera similar a la definida en 6.1.1.

Si el modo vector de movimiento sin restricción se utiliza también con el modo actualización de resolución reducida, **pseudo- $MV_C$**  se obtiene añadiendo las diferencias del vector de movimiento  $MVD$  (y  $MVD_{2-4}$ ) del cuadro D.3.

Si están presentes cuatro vectores de movimiento, el procedimiento se efectúa de manera similar a la definida en F.2.

- 3) El componente del vector de movimiento  $MV_C$  se obtiene a partir del pseudo- $MV_C$  en la fórmula siguiente:

$$MV_C = 0 \quad \text{si pseudo-}MV_C = 0$$

$$MV_C = \text{sign}(\text{pseudo-}MV_C) * (2,0 * |\text{pseudo-}MV_C| - 0,5) \quad \text{si pseudo-}MV_C \neq 0$$

Como resultado, se limita a cada componente del vector a tener un valor de medio entero o cero, y la gama de cada componente del vector de movimiento se amplía hasta aproximadamente el doble de la gama del vector de pseudomovimiento.

- 4) Si el modo de la imagen vigente es trama PB mejorada, o cuando MBTYPE indica el modo directo en una imagen B, se crean los componentes del vector de movimiento  $MV_F$  y/o  $MV_B$  para la predicción hacia adelante y hacia atrás.

En primer lugar, se calculan los pseudocomponentes del vector de movimiento **pseudo- $MV_F$**  y/o **pseudo- $MV_B$**  en base a las reglas para los modos de predicción definidas en los anexos O y M.

En el caso de predicción bidireccional en el modo tramas PB mejoradas (véase M.2.1) o cuando MBTYPE indica el modo directo en una imagen B (véase O.5.2), se calculan los componentes pseudo- $MV_F$  y pseudo- $MV_B$  a partir de **pseudo- $MV_D$**  y **pseudo- $MV_C$**  suponiendo que pseudo- $MV_D$  es cero y pseudo- $MV_C$  es  $MV$ , tal como se define en los anexos G y M.

En el caso de predicción hacia adelante en el modo tramas PB mejoradas (véase M.2.2), el **pseudo- $MV_{DB}$**  se obtiene decodificando el código de longitud variable  $MV_{DB}$  según a lo indicado en el cuadro 12. A continuación se obtiene pseudo- $MV_F$  añadiendo pseudo- $MV_{DB}$  al **pseudo-Predictor**. Para formar el pseudo-Predictor, se convierte el predictor obtenido según el procedimiento definido en M.2.2, al vector pseudo-Predictor, de conformidad con la fórmula definida en el punto 1) de esta subcláusula.

En el caso de la predicción hacia atrás en el modo tramas PB mejoradas (véase M.2.3), el pseudo- $MV_B$  se fija a cero.

Los vectores de movimiento  $MV_F$  y/o  $MV_B$  para la predicción hacia adelante y hacia atrás se obtienen a partir del pseudo- $MV_F$  y/o el pseudo- $MV_B$ , según la fórmula definida en el punto 3) de esta subcláusula.

## **Q.5 Compensación de movimiento con superposición ampliada para luminancia**

Si el modo predicción avanzada se utiliza también con el modo actualización de resolución reducida, se emplean matrices ampliadas de valores de ponderación para realizar la compensación de movimiento con superposición. El procedimiento de creación de cada bloque de predicción es idéntico al que se describe en F.3, con la salvedad de que el tamaño de cada bloque y matriz de ponderación es de  $16 \times 16$ .

En las figuras Q.4, Q.5 y Q.6 se muestran las matrices ampliadas de valores de ponderación para la predicción de luminancia de  $16 \times 16$ .

4	4	5	5	5	5	5	5	5	5	5	5	5	5	4	4
4	4	5	5	5	5	5	5	5	5	5	5	5	5	4	4
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
5	5	5	5	6	6	6	6	6	6	6	6	5	5	5	5
5	5	5	5	6	6	6	6	6	6	6	6	5	5	5	5
5	5	5	5	6	6	6	6	6	6	6	6	5	5	5	5
5	5	5	5	6	6	6	6	6	6	6	6	5	5	5	5
5	5	5	5	6	6	6	6	6	6	6	6	5	5	5	5
5	5	5	5	6	6	6	6	6	6	6	6	5	5	5	5
5	5	5	5	6	6	6	6	6	6	6	6	5	5	5	5
5	5	5	5	6	6	6	6	6	6	6	6	5	5	5	5
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
4	4	5	5	5	5	5	5	5	5	5	5	5	5	4	4
4	4	5	5	5	5	5	5	5	5	5	5	5	5	4	4

**Figura Q.4/H.263 – Valores de ponderación,  $H_0$ , para la predicción con vector de movimiento de bloques de luminancia de  $16 \times 16$  vigentes**

2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
1	1	1	1	2	2	2	2	2	2	2	2	1	1	1	1
1	1	1	1	2	2	2	2	2	2	2	2	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	2	2	2	2	2	2	2	2	1	1	1	1
1	1	1	1	2	2	2	2	2	2	2	2	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

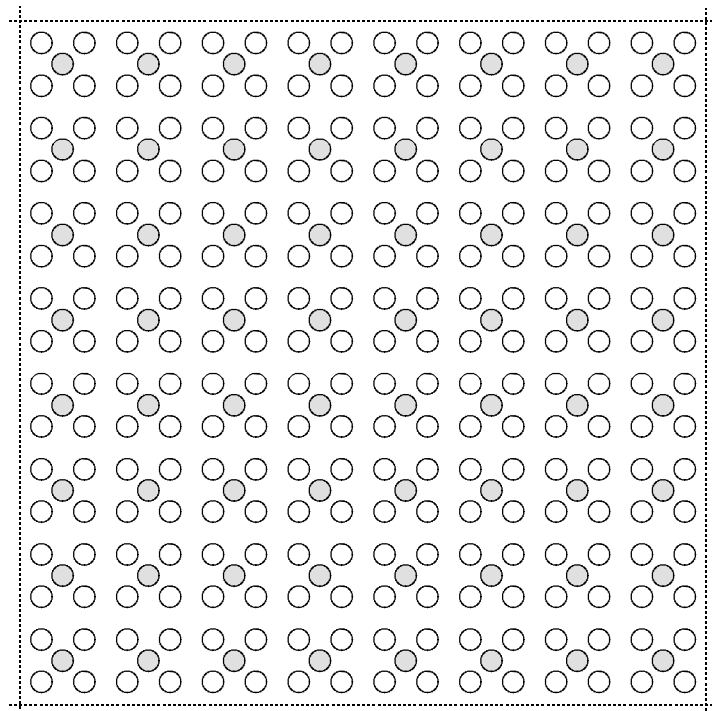
**Figura Q.5/H.263 – Valores de ponderación,  $H_1$ , para la predicción con vector de movimiento de bloques de luminancia de  $16 \times 16$  vigentes en la parte superior o inferior del bloque de luminancia de  $16 \times 16$  vigente**

2	2	1	1	1	1	1	1	1	1	1	1	1	1	2	2
2	2	1	1	1	1	1	1	1	1	1	1	1	1	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	1	1	1	1	1	1	1	1	1	1	1	1	2	2
2	2	1	1	1	1	1	1	1	1	1	1	1	1	2	2

**Figura Q.6/H.263 – Valores de ponderación,  $H_2$ , para la predicción con vector de movimiento de bloques de luminancia de  $16 \times 16$  a la izquierda o derecha del bloque de luminancia de  $16 \times 16$  vigente**

#### **Q.6 Muestreo a velocidad superior del error de predicción reconstruido de resolución reducida**

El bloque con error de predicción reconstruido de  $16 \times 16$  se obtiene mediante el muestreo a velocidad superior del bloque con error de predicción reconstruido de resolución reducida de  $8 \times 8$ . Para realizar una implementación simple, el proceso de filtrado se cierra dentro de un bloque que permite efectuar cada muestreo a velocidad superior a nivel de bloque. La figura Q.7 muestra la posición de las muestras. El procedimiento de muestreo a velocidad superior para los píxeles de luminancia y crominancia que están dentro de los bloques con error de predicción reconstruidos de  $16 \times 16$  se define en Q.6.1. Para la creación de los píxeles de luminancia y crominancia que están en los límites del bloque con error de predicción reconstruido de  $16 \times 16$ , el procedimiento se define en Q.6.2. El muestreo a velocidad superior se efectúa tanto para los bloques de crominancia como para los bloques de luminancia. En las figuras Q.8 y Q.9 el símbolo "/" significa división por truncamiento.



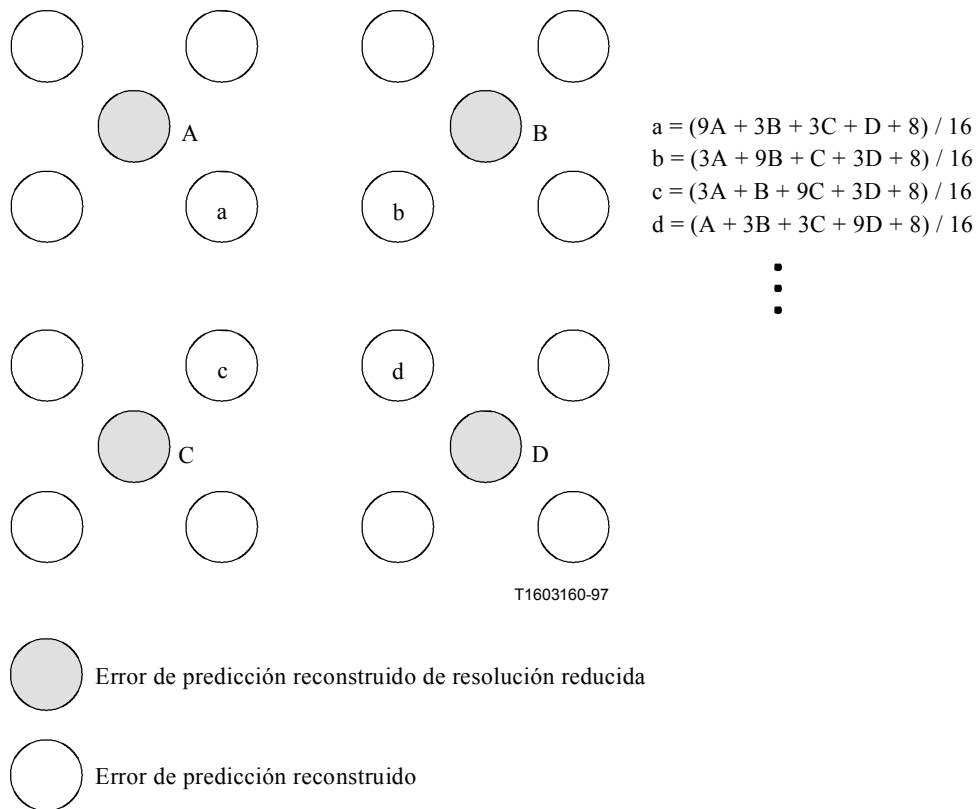
T1603150-97

- Posición de las muestras en un bloque con error de predicción reconstruido de resolución reducida de  $8 \times 8$
- Posición de las muestras en un bloque con error de predicción reconstruido de  $16 \times 16$
- Borde de bloque

**Figura Q.7/H.263 – Posición de las muestras en un bloque con error de predicción reconstruido de resolución reducida de  $8 \times 8$  y en un bloque con error de predicción reconstruido de  $16 \times 16$**

### **Q.6.1 Procedimiento de muestreo a velocidad superior para los píxels situados dentro de un bloque con error de predicción reconstruido de $16 \times 16$**

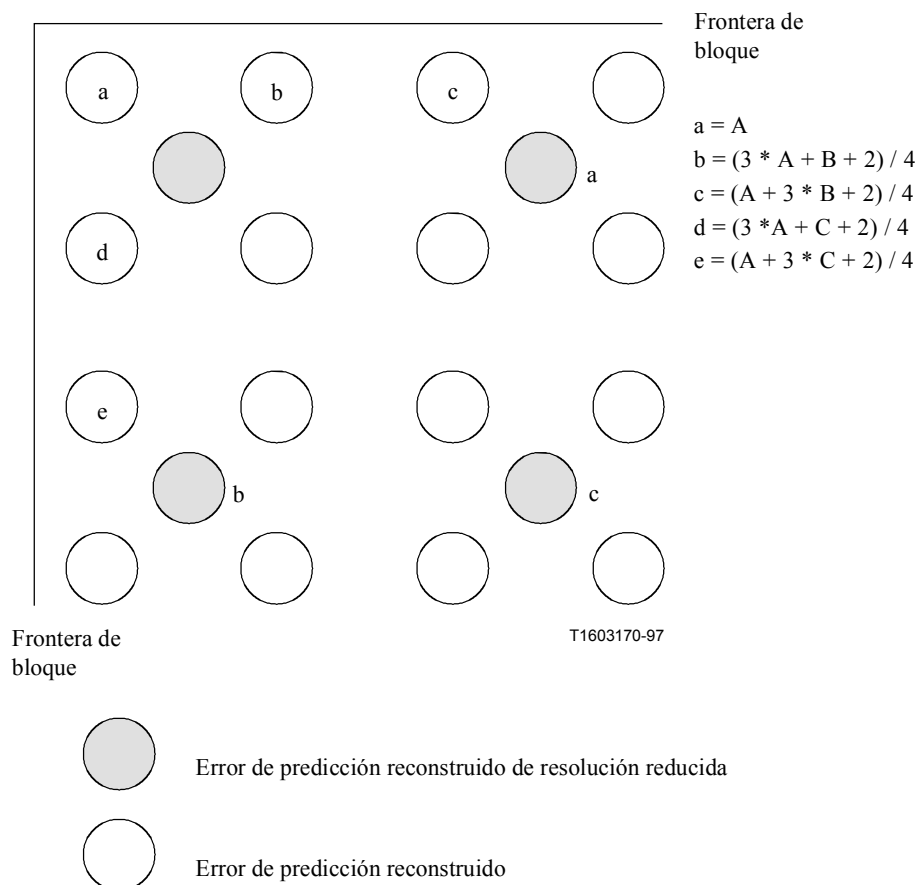
En la figura Q.8 se muestra la creación del error de predicción reconstruido para píxels situados dentro de un bloque. "/" significa división por truncamiento.



**Figura Q.8/H.263 – Creación de error de predicción reconstruido para píxels situados dentro de un bloque**

**Q.6.2 Procedimiento de muestreo a velocidad superior para los píxels situados en la frontera de un bloque con error de predicción reconstruido de  $16 \times 16$**

En la figura Q.9 se describe la creación del error de predicción reconstruido para píxels de un bloque de  $16 \times 16$ .



**Figura Q.9/H.263 – Creación de error de predicción reconstruido para los píxeles situados en la frontera de un bloque**

## Q.7 Filtro de frontera de bloque

Las operaciones de filtrado se efectúan en los bordes de los bloques reconstruidos de  $16 \times 16$  en el lado del codificador y en el decodificador. Existen dos alternativas de filtrado, dependiendo de que se utilice o no el modo filtro de desbloqueo.

El filtrado por defecto en el modo actualización de resolución reducida se efectúa conforme a lo indicado en Q.7.1.

Si el modo filtro de desbloqueo se utiliza también con el modo actualización de resolución reducida, el filtrado se efectúa conforme a lo indicado en Q.7.2.

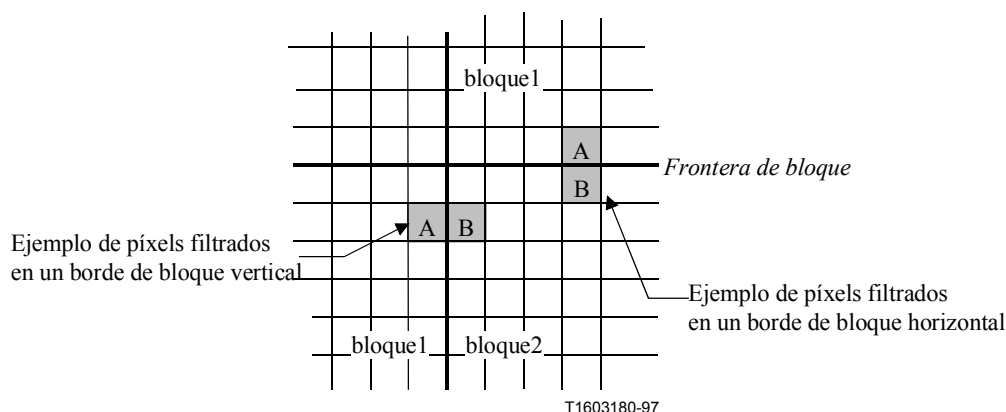
En ambos casos, el filtrado se efectúa en los datos de la imagen reconstruida completa antes de almacenarlos en la memoria de la imagen para una futura predicción. No se realiza ningún filtrado en los bordes de la imagen, en los bordes de rebanada en el modo estructura en rebanada (véase el anexo K), ni en los límites de los GOB que tienen encabezamientos GOB presentes en el modo decodificación de segmento independiente (véase el anexo R). Los datos de crominancia y luminancia son filtrados.

### Q.7.1 Definición del filtro de frontera de bloque por defecto

En el modo actualización de resolución reducida, el filtrado por defecto se efectúa conforme a lo indicado en esta subcláusula.

A y B son dos valores de píxeles en una línea – horizontal o vertical – de la imagen reconstruida, y A pertenece a un bloque de  $16 \times 16$  llamado bloque1 mientras que B pertenece a un bloque de  $16 \times 16$

contiguo llamado bloque2, que está a la derecha o debajo del bloque1. La figura Q.10 muestra ejemplos de la posición de estos píxels.



**Figura Q.10/H.263 – Filtro de frontera de bloque por defecto**

Para activar el filtro en un borde determinado debe cumplirse una de las siguientes condiciones:

- el bloque1 pertenece a un macrobloque codificado (COD==0 || tipo MB == INTRA); o
- el bloque2 pertenece a un macrobloque codificado (COD==0 || tipo MB == INTRA).

A será sustituido por A1 y B por B1. "/" significa división por truncamiento.

$$A1 = (3 * A + B + 2) / 4$$

$$B1 = (A + 3 * B + 2) / 4$$

El orden de los bordes cuando se efectúa el filtrado es idéntico al que se describe en J.3.

### Q.7.2 Definición del filtro de frontera de bloque cuando se utiliza el modo filtro de desbloqueo

Si se utiliza el modo filtro de desbloqueo (véase el anexo J) con el modo actualización de resolución reducida, se efectúa el filtrado que se define en el anexo J con una modificación en los píxels de frontera de los bloques de luminancia y crominancia de  $16 \times 16$ , en lugar del filtrado descrito en Q.7.1. La modificación del filtrado que figura en el anexo J consiste en que al parámetro STRENGTH se le asigna el valor de infinito positivo, lo que tiene por consecuencia el que la función UpDownRamp(x, STRENGTH) definida en J.3 pase a ser una función lineal de x.

Como resultado de todo ello, el procedimiento del filtro de desbloqueo descrito en J.3 se redefine de la siguiente manera:

$$B1 = \text{clip}(B + d1)$$

$$C1 = \text{clip}(C - d1)$$

$$A1 = A - d2$$

$$D1 = D + d2$$

$$d1 = (A - 4B + 4C - D) / 8$$

$$d2 = \text{clipd1}((A - D) / 4, d1/2)$$



## Anexo R

### Modo decodificación de segmento independiente

#### R.1 Introducción

Este anexo describe el modo decodificación de segmento independiente opcional de la presente Recomendación, que permite que una imagen sea decodificada sin la presencia de ninguna dependencia de datos en los límites de rebanadas o de grupos de bloques (GOB) que tienen encabezamientos GOB no vacíos. La utilización de este modo se indica en el campo PLUSPTYPE del encabezamiento de imagen. La capacidad de utilizar este modo opcional se negocia por medios externos (por ejemplo, aplicando la Rec. UIT-T H.245).

Cuando se indica la utilización de este modo, los límites del segmento de la imagen de vídeo (definidos por los límites de las rebanadas o los límites superiores de los GOB para los que se envían encabezamientos GOB o las fronteras de la imagen, lo que limite una región de manera más reducida) se tratan como fronteras de imagen cuando se decodifican, incluido el tratamiento de los vectores de movimiento que atraviesan esas fronteras (que dan por resultado la extrapolación de la frontera cuando se utilizan los modos vector de movimiento sin restricción, predicción avanzada, filtro de desbloqueo, o escalabilidad temporal, SNR y espacial, y que están prohibidos cuando no se utiliza ninguno de estos modos opcionales).

#### R.2 Funcionamiento del modo

Un segmento de imagen de vídeo se define como sigue:

Si no se utiliza el modo estructura en rebanada (véase el anexo K), un GOB o un número múltiple de GOB consecutivos forman un segmento de imagen de vídeo. La ubicación de la parte superior de cada segmento de imagen de vídeo viene indicada por la presencia de un encabezamiento GOB no vacío cuyo límite de segmento de imagen de vídeo se encuentra justo encima de los macrobloques en el GOB para el que está presente un encabezamiento, o por la parte superior de la imagen, lo que esté más bajo. La ubicación de la parte inferior de cada segmento de imagen de vídeo la define la parte superior del segmento de imagen de vídeo siguiente, o la parte inferior de la imagen, lo que esté más alto.

Si se utiliza el modo estructura en rebanada (véase el anexo K), cada rebanada constituye un segmento de imagen de vídeo.

En el modo decodificación de segmento independiente, cada segmento de imagen de vídeo se decodifica con completa independencia de todos los demás segmentos de imagen de vídeo, y también es independiente de todos los datos situados fuera de la misma ubicación del segmento de imagen de vídeo en la imagen o imágenes de referencia. Esto implica:

- 1) La no utilización de vectores de movimiento fuera del segmento de imagen de vídeo vigente para la predicción del vector de movimiento (conforme a 6.1.1).
- 2) La no utilización de vectores de movimiento fuera del segmento de imagen de vídeo vigente como vectores de movimiento distantes para compensación de movimiento de bloques superpuestos cuando se utiliza el modo predicción avanzada (véase F.3).
- 3) El no funcionamiento del filtro de desbloqueo en los límites del segmento de la imagen de vídeo (véase J.3).
- 4) La no utilización de vectores de movimiento que hagan referencia a datos de fuera del segmento de imagen de vídeo vigente salvo que se utilice el modo vector de movimiento sin restricción (véase el anexo D), el modo predicción avanzada (véase el anexo F), el modo filtro de desbloqueo (véase el anexo J) o el modo escalabilidad temporal, SNR y espacial (véase el anexo O); en cuyo caso los límites del segmento de imagen de vídeo vigente en la

imagen anterior se extrapolan como se describe en el anexo D para formar las predicciones de los píxeles que hacen referencia a la región que queda fuera de los límites.

- 5) La no interpolación bilineal en los límites de la región de tamaño 1/4 o de tamaño 1/2 que corresponde al segmento de imagen de vídeo vigente para predicción hacia arriba en imágenes EI y EP de escalabilidad espacial (como se define en el anexo O).
- 6) El no funcionamiento del filtro de límite de bloque en los límites del segmento de imagen de vídeo cuando se utiliza el modo actualización de resolución reducida (véase el anexo Q).
- 7) La no utilización del modo repetición de muestreo de la imagen de referencia con el modo decodificación de segmento independiente.

### **R.3 Constricciones a la utilización**

Cuando se utiliza el modo de decodificación de segmento independiente se imponen ciertas constricciones a la utilización de otros aspectos de la sintaxis de codificación de vídeo. Estas constricciones tienen por objeto evitar dos casos patológicos que, de otro modo, haría muy dificultoso el funcionamiento del modo decodificación de segmento independiente.

#### **R.3.1 Constricciones a la forma de los segmentos**

Cuando se utiliza el modo estructura en rebanada (anexo K) sin utilizar el submodo rebanada rectangular (véase K.1) pueden presentarse casos en los que la forma de un segmento de imagen de vídeo quizás no sea convexa (que tenga "esquinas interiores", o incluso comprenda dos regiones de la imagen separadas y distintas).

Por ello, el modo decodificación de segmento independiente no debe utilizarse con el modo estructura en rebanada sin la utilización simultánea del submodo rebanada rectangular del modo estructura en rebanada (véase el anexo K). Esta constricción se impone para eludir la necesidad de aplicar un difícil tratamiento de caso especial con el que determinar cómo y cuándo efectuar la extrapolación de cada segmento de imagen de vídeo.

#### **R.3.2 Constricciones a los cambios de forma de los segmentos**

Si se permitiera variar de cualquier modo la forma de los segmentos de imagen de vídeo de una imagen a otra en el tren de bits, podrían presentarse casos en los que sería difícil decodificar el tren de bits. Esto se debe a que en tales casos el contenido del tren de bits no es suficiente para determinar la forma de cada segmento de imagen de vídeo antes de la posible aparición de vectores de movimiento en el tren de bits que exigen que se conozca la forma del segmento de imagen de vídeo para una interpretación correcta.

Por ello, cuando se utilice el modo codificación de segmento independiente, la segmentación de imagen de vídeo de todas las imágenes y tramas que permiten predicción temporal (es decir, todas las imágenes P, B, y EP y todas las tramas PB mejoradas) serán las mismas que las utilizadas en su imagen de referencia temporal. Cuando se utilice asimismo, el modo codificación de segmento independiente, la segmentación de imagen de vídeo para todas las imágenes EI será la misma o diferirá solamente por la subdivisión de la segmentación de imagen de vídeo utilizada en su imagen de referencia. Además, el modo codificación de segmento independiente no se utilizará en ninguna imagen o trama que utilice imágenes de referencia (todos los tipos de imagen salvo INTRA) a menos que se utilice también el modo de codificación de segmento independiente en todas las imágenes de referencia para la imagen vigente. Como resultado de esta constricción, la forma de los segmentos de imagen de vídeo en el modo codificación de segmento independiente nunca cambiará de una imagen a otra excepto cuando cambie en imágenes I y EI (y la propia manera según la cual las imágenes EI pueden cambiar la segmentación está también algo restringida).

## Anexo S

### Modo códigos de longitud variable (VLC) INTER alternativos

#### S.1 Introducción

Este anexo describe el modo códigos de longitud variable (VLC) INTER alternativos opcional de la presente Recomendación, que mejora la eficacia de la codificación entre imágenes cuando son evidentes cambios significativos en la imagen. Esta mejora de la eficacia se obtiene permitiendo que algunos VLC concebidos originalmente para imágenes INTRA se utilicen también para algunos coeficientes de imagen INTER y datos CBPY. La utilización de este modo se indica en el campo PLUSPTYPE del encabezamiento de imagen. La capacidad de utilizar este modo opcional se negocia por medios externos (por ejemplo, aplicando la Rec. UIT-T H.245). El modo contiene dos alteraciones de sintaxis, una para la codificación de los coeficientes INTER y otra para la codificación de los valores CBPY INTER.

#### S.2 Códigos de longitud variable (VLC) INTER alternativos para coeficientes

La idea que subyace en el diseño de la tabla VLC INTRA del anexo I es utilizar las mismas palabras de código que en la VLC INTER original pero con una interpretación diferente de LEVEL y RUN. La tabla VLC INTRA conviene más en los casos en los que hay muchos coeficientes y/o sus valores son grandes.

Los VLC INTRA se construyen de modo tal que las palabras de código tengan el mismo valor para LAST (0 ó 1) en las tablas INTER e INTRA. La tabla INTRA se produce, por tanto, "reordenando" el significado de las palabras de código con el mismo valor de LAST. Además, para eventos con |LEVEL| grande la tabla INTRA utiliza una palabra de código que en la tabla INTER tiene un valor RUN grande. En bloques INTER que tiene un considerable número de coeficientes de gran magnitud, puede resultar a veces más eficaz utilizar la tabla INTRA en lugar de la tabla INTER, y en alguno de esos casos la elección de la tabla de VLC puede parecerle obvia al decodificador, ya que la decodificación que utilizará la tabla INTER produciría valores RUN muy grandes como para indicar la presencia de más de 64 coeficientes en un bloque. En estas circunstancias, se puede utilizar la tabla INTRA para mejorar la eficacia de la codificación INTER.

##### S.2.1 Acción del codificador

El codificador puede utilizar la tabla VLC INTRA para codificar un bloque INTER cuando el decodificador pueda detectar su uso, en otras palabras, siempre que la decodificación que utilice la tabla VLC INTER haga que se traten coeficientes distintos de los 64 coeficientes de un bloque.

Normalmente, el codificador sólo optará por utilizar la tabla VLC INTRA para codificar un bloque INTER cuando se satisfaga la condición anterior y además cuando el uso de VLC INTRA genere menos bits que VLC INTER para los mismos valores de coeficiente. Tal será el caso con frecuencia cuando haya muchos coeficientes grandes, debido al modo como se produjo la tabla VLC INTRA (ya que la tabla VLC INTER ordinaria contiene largos tramos de pasada para las mismas palabras de código en las que la tabla VLC INTRA contiene grandes amplitudes de coeficientes).

##### S.2.2 Acción del decodificador

El proceso de decodificación es como sigue:

- 1) El decodificador recibe primero todos los códigos de los coeficientes de un bloque.
- 2) Las palabras de código son interpretadas a continuación suponiendo que se utiliza VLC INTER. Si el tratamiento de los coeficientes permanece dentro de los 64 coeficientes de un bloque, la decodificación VLC concluye.

- 3) Si se tratan coeficientes de fuera del bloque, las palabras de código se interpretarán conforme a VLC INTRA.

### **S.3 Códigos de longitud variable (VLC) INTRA alternativos para CBPY**

Las palabras de código CBPY INTER (cuadro 12) están concebidas en el supuesto de que hay más bloques Y con todos los coeficientes cero, que bloques con al menos un coeficiente distinto de cero. Si los bloques  $C_B$  y  $C_R$  tienen al menos un coeficiente distinto de cero, es decir,  $CBPC_5 = CBPC_6 = 1$ , esta superposición ya no se mantiene. Por tal motivo, cuando se utilice el modo VLC INTER alternativos, las palabras de código CBPY que se definen en el cuadro 12 para macrobloques INTRA se utilizarán también para macrobloques INTER siempre que  $CBPC_5 = CBPC_6 = 1$ .

## **Anexo T**

### **Modo cuantificación modificada**

#### **T.1 Introducción**

Este anexo describe el modo opcional cuantificación modificada de la presente Recomendación, que modifica el funcionamiento del cuantificador. La utilización de este modo se indica en el campo PLUSPTYPE del encabezamiento de imagen. La capacidad de utilizar este modo opcional se negocia por medios externos (por ejemplo, aplicando la Rec. UIT-T H.245).

Este modo tiene cuatro características fundamentales:

- 1) Se mejora la capacidad de controlar la velocidad binaria de la codificación alterando la sintaxis del campo DQUANT.
- 2) Se mejora la fidelidad de la crominancia mediante la especificación de un tamaño de paso menor para los datos de crominancia que para los datos de luminancia.
- 3) Se amplía la gama de valores de coeficientes representables para permitir la representación de cualquier posible valor de coeficiente verdadero dentro de la exactitud permitida por el tamaño de paso de cuantificación.
- 4) Se restringe la gama de niveles de coeficientes cuantificados a los que pueden razonablemente ocurrir, para mejorar la capacidad de detección de errores y reducir al mínimo la complejidad de la decodificación.

#### **T.2 Autorización de DQUANT modificado**

Este modo modifica la semántica del campo DQUANT. Con este modo, es posible utilizar DQUANT para modificar QUANT en más o en menos un pequeño valor, o bien señalar cualquier nuevo valor específico de QUANT. El tamaño del pequeño valor de modificación depende del valor vigente de QUANT. Con la utilización de este modo, se pueden especificar controles más flexibles del tamaño de paso del cuantificador en el campo DQUANT.

La palabra de código para DQUANT en este modo ya no es un campo de longitud fija de 2 bits, sino un campo de longitud variable que puede tener una longitud de dos o seis bits, dependiendo del primer bit del código. Por ello, la descripción que sigue se divide en dos secciones, que dependen del primer bit.

##### **T.2.1 Alteración de QUANT en pequeños pasos**

Cuando el primer bit del campo DQUANT es 1, sólo se envía un bit adicional en DQUANT. El único bit adicional se utiliza para modificar QUANT por un valor diferencial. El cambio del valor

de QUANT depende del segundo bit de DQUANT y del valor previo de QUANT, como se muestra en el cuadro T.1.

**Ejemplo:** Si el valor previo de QUANT es 29 y DQUANT viene señalado con la palabra de código "11", el valor diferencial es +2, y, por tanto, el nuevo valor de QUANT resultante es 31.

**Cuadro T.1/H.263 – Semántica de la alteración de QUANT en pequeños pasos**

QUANT anterior	Cambio de QUANT	
	DQUANT = 10	DQUANT = 11
1	+2	+1
2-10	-1	+1
11-20	-2	+2
21-28	-3	+3
29	-3	+2
30	-3	+1
31	-3	-5

### T.2.2 Selección de QUANT arbitraria

Cuando el primer bit del campo DQUANT es 0, se envían cinco bits adicionales en DQUANT. Los cinco bits siguientes representan un nuevo QUANT como se define en 5.1.19.

**Ejemplo:** Independientemente del valor vigente de QUANT, si DQUANT se señala con la palabra de código "001111", el nuevo valor de QUANT es 15.

### T.3 Tamaño de paso de cuantificación alterado para coeficientes de crominancia

Cuando se utiliza el modo cuantificación modificada, el parámetro de cuantificación de los coeficientes de crominancia es diferente del parámetro de cuantificación de la luminancia. El parámetro de cuantificación de luminancia se señala en el tren de bits y se denomina QUANT. Cuando se utiliza este modo, se emplea un parámetro de cuantificación diferente denominado QUANT\_C para la cuantificación inversa de los coeficientes de crominancia. La relación entre QUANT y QUANT\_C se indica en el cuadro T.2. Si se utiliza el modo filtro de desbloqueo (véase el anexo J), deberá utilizarse también QUANT\_C para la aplicación del filtro de desbloqueo a los datos de crominancia. Cada vez que QUANT se trate aquí en cualquier otro contexto, significará tamaño de paso de cuantificación de la luminancia.

**Cuadro T.2/H.263 – Relación entre QUANT y QUANT\_C**

Gama de QUANT	Valor de QUANT_C
1-6	QUANT_C = QUANT
7-9	QUANT_C = QUANT - 1
10-11	9
12-13	10
14-15	11
16-18	12
19-21	13
22-26	14
27-31	15

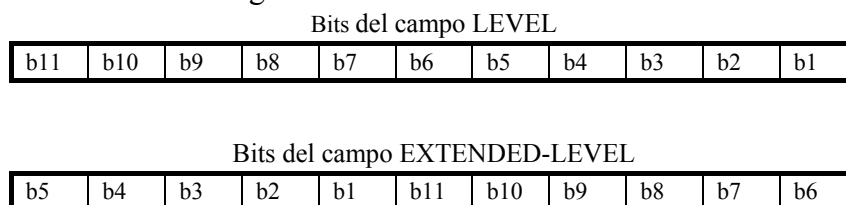
#### T.4 Gama de coeficientes modificados

Cuando se utiliza el modo cuantificación modificada, se pueden representar en el tren de bits los coeficientes DCT cuantificados que tienen unos valores absolutos del nivel de cuantificación superiores a 127. Esto tiene dos ventajas:

- 1) Se mejora la calidad de funcionamiento del codificador al permitir que se represente la gama total verdadera de valores de posibles coeficientes.
- 2) Se reduce la complejidad del codificador al eliminar la necesidad de incrementar el tamaño del paso de cuantificación cuando se encuentran ciertos valores de coeficientes grandes que de otro modo no se podrían representar.

Es posible que el valor correcto de un coeficiente DCT antes de la cuantificación en el codificador tenga un valor absoluto de hasta 2040. Así, una gama de  $-127$  a  $+127$  para LEVEL es insuficiente para abarcar toda la gama de valores de coeficientes posibles cuando el parámetro de cuantificación QUANT o QUANT\_C es menor que 8. La gama de coeficientes ampliada ensancha la gama de LEVEL para permitir que cualquier valor de coeficiente verdadero sea codificado más correctamente.

Cuando se utiliza el modo cuantificación modificada, se altera el significado del campo LEVEL que sigue a un código ESCAPE (0000 011, conforme a 5.4.2). En este modo, en vez de estar prohibida la secuencia de bits 1000 0000, se utiliza para representar un código ESCAPE AMPLIADO (EXTENDED-ESCAPE). Un coeficiente AC de valor absoluto superior a 127 se representa enviando un código EXTENDED-ESCAPE, seguido inmediatamente de un campo NIVEL AMPLIADO (EXTENDED-LEVEL) con una longitud fija de 11 bits. Un valor de coeficiente ampliado del campo EXTENDED-LEVEL se codifica tomando los 11 bits menos significativos de la representación binaria del complemento a dos de LEVEL y rotándolos cíclicamente a la derecha por posiciones de 5 bits. La rotación es necesaria para evitar la emulación del código de comienzo. La rotación cíclica se ilustra en la figura T.1.



**Figura T.1/H.263 – Rotación cíclica de la representación de coeficientes**

#### T.5 Restricciones de utilización

Cuando se utiliza el modo cuantificación modificada, se aplican ciertas restricciones a los valores de coeficientes codificados. Esto tiene varias ventajas:

- 1) Se mejora la capacidad de detectar errores en los bits prohibiendo determinados valores irreales de los coeficientes, con lo que se hace que esos valores sean reconocidos como errores en los bits por el decodificador.
- 2) Se reduce la complejidad del decodificador disminuyendo la longitud de palabra necesaria para la cuantificación inversa previa al recorte.
- 3) Se evita la emulación del código de comienzo para coeficientes codificados utilizando el mecanismo EXTENDED-ESCAPE descrito en T.4.

Esas restricciones son las siguientes. Cuando se utiliza el modo cuantificación modificada:

- 1) Para cualquier coeficiente, el valor absoluto del nivel de reconstrucción |REC| producida por el proceso de cuantificación inverso descrito en 6.2.1, que utiliza el valor vigente de QUANT o QUANT\_C según proceda, y el valor codificado de LEVEL, serán menores que

4096. Esta restricción adicional se aplica a todos los coeficientes, sin tener en cuenta si el coeficiente se envía utilizando o no el mecanismo EXTENDED-ESCAPE.
- 2) El tren de bits no utilizará el código ESCAPE normal o el código EXTENDED-ESCAPE para codificar una combinación de LAST, RUN y LEVEL para la que existe una entrada de palabra de código en la tabla de VLC aplicable, que es la tabla del cuadro 16 (véase 5.4.2) o del cuadro I.2 (véase I.3).
  - 3) El código EXTENDED-ESCAPE sólo se utilizará cuando el parámetro de cuantificación para el coeficiente (QUANT o QUANT\_C) sea menor que ocho (8).
  - 4) El código EXTENDED-ESCAPE sólo se utilizará cuando vaya seguido de un campo EXTENDED-LEVEL que represente un valor de LEVEL que esté fuera de la gama -127 a +127.

## Anexo U

### Modo de selección mejorada de imagen de referencia

#### U.1 Introducción

El presente anexo describe el modo opcional selección mejorada de imagen de referencia (ERPS, *enhanced reference picture selection*) de esta Recomendación. La capacidad para utilizar este modo opcional se negocia por medios externos (por ejemplo, utilizando Rec. UIT-T H.245). La cantidad de memoria de imagen disponible en el decodificador para el funcionamiento ERPS deberá también ser indicada por medios externos. La utilización de este modo se deberá indicar poniendo a "1" el bit 16 previamente reservado de la parte opcional de PLUSPTYPE (OPPTYPE). Este modo proporciona ventajas tanto para la elasticidad a errores como para la eficacia de la codificación por medio de la utilización de una memoria tampón para imágenes de referencia.

Se especifica un submodo del modo ERPS para la eliminación de subimágenes. La finalidad de la eliminación de subimágenes es la reducción de la cantidad de memoria necesaria para el almacenamiento de múltiples imágenes de referencia. La reducción de memoria se lleva a cabo especificando la partición de cada imagen de referencia en unidades rectangulares más pequeñas llamadas subimágenes. El codificador puede entonces indicar al decodificador que determinadas áreas de subimagen de las imágenes de referencia específicas no se utilizarán como referencia para la predicción de imágenes subsiguientes, permitiendo por tanto que la memoria asignada en el decodificador para el almacenamiento de dichas áreas sea utilizada para el almacenamiento de los datos de otras imágenes de referencia. El soporte para este submodo y la fragmentación permitida de la memoria de imagen en unidades mínimas de imagen (MPU, *minimum picture unit*) para la eliminación de subimágenes de la forma aquí definida, se negocia también por medios externos (por ejemplo, utilizando Rec. UIT-T H.245).

Para facilitar en las imágenes B la predicción hacia atrás de dos imágenes, se especifica un submodo del modo ERPS. Este submodo puede mejorar el comportamiento proveyendo codificadores para las imágenes B, no sólo con la capacidad de utilizar múltiples referencias para la predicción hacia adelante, sino también la utilización de más de una imagen de referencia en la predicción hacia atrás. El soporte de este submodo se negocia por medios externos (por ejemplo, utilizando Rec. UIT-T H.245).

Para la elasticidad a errores, el modo ERPS puede utilizar mensajes por el canal de retorno, que se señalizan por medios externos (por ejemplo, mediante Rec. UIT-T H.245), enviados desde un decodificador a un codificador para informarle qué imágenes, o partes de las imágenes, han sido incorrectamente decodificadas. El modo ERPS proporciona un comportamiento mejorado en comparación con el modo selección de imagen de referencia (RPS, *reference picture selection*)

definido en el anexo N. No se deberá utilizar simultáneamente con el modo RPS. (Puede ser utilizado de forma tal que proporcione esencialmente la misma funcionalidad que el modo RPS.)

Para la eficacia de la codificación, se puede ampliar la compensación de movimiento para la predicción con varias imágenes. La ampliación de la compensación de movimiento a la predicción multiimagen se consigue ampliando cada vector de movimiento con un parámetro de referencia de imagen que se utiliza para direccionar a un macrobloque, o región de predicción de bloque, para la compensación de movimiento en cualquiera de las múltiples imágenes de referencia. El parámetro de referencia de imagen es un código de longitud variable que especifica un índice relativo de memoria tampón. Las imágenes de referencia se ensamblan en un esquema de almacenamiento en memoria tampón controlado por el codificador.

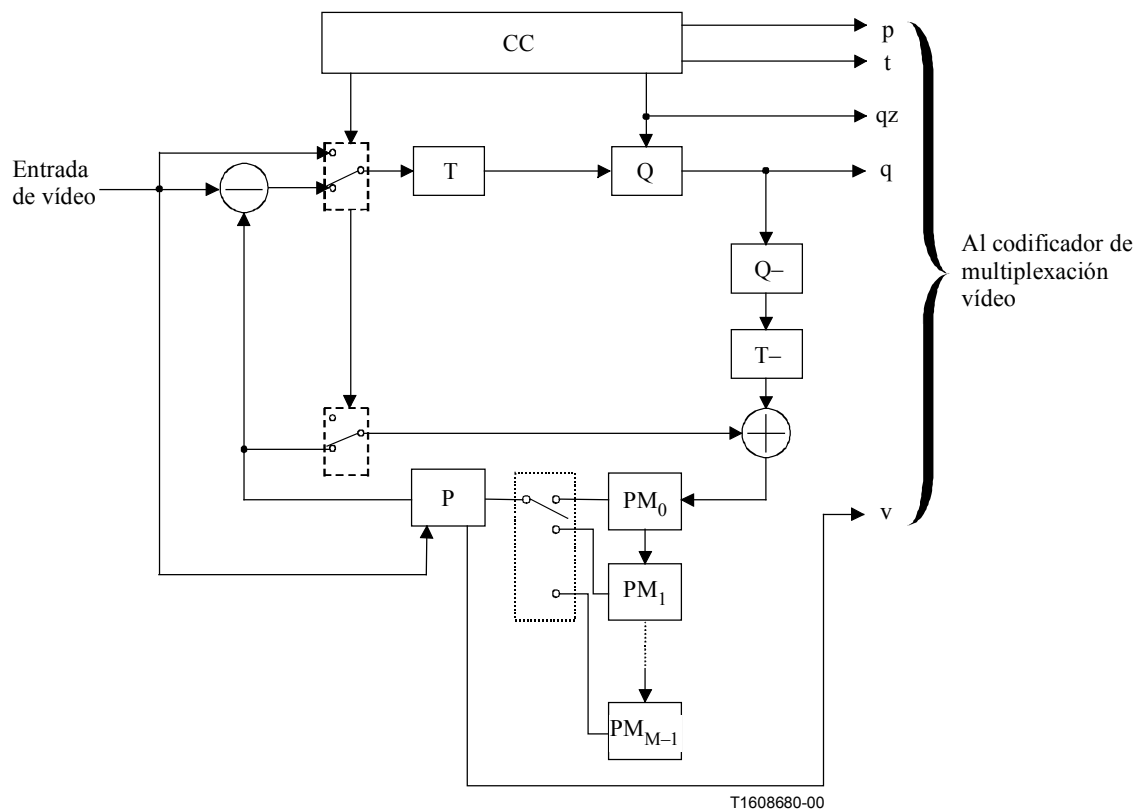
El modo ERPS no se deberá utilizar con el modo codificación aritmética basada en sintaxis (véase el anexo E) o con el modo rebanada partida de datos (véase el anexo V).

Una vez activado, el modo ERPS no se desactivará en las subsiguientes imágenes del tren de bits, a no ser que la desactivación inicial se produzca en una imagen I o EI, y que cualquier reactivación se produzca también en una imagen I o EI, y venga acompañada por una reiniciación de la memoria tampón (RESET igual a "1"). Si se desactiva, todo el contenido del tampón multiimagen ERPS se deberá poner en el estado "sin usar".

## **U.2 Algoritmo de codificación de fuente de vídeo**

En la figura U.1 se presenta un esquema general del codificador de fuente de este modo. Esta figura muestra una estructura que utiliza varias memorias de imagen.





- |    |   |
|----|---|
| T  | Transformación  |
| Q  | Cuantificador   |
| P  | Memoria de imagen con retardo variable y compensación de movimiento |
| PM | Memoria de imagen   |
| CC | Control de codificación   |
| p  | Bandera de INTRA/INTER  |
| t  | Bandera de transmitido o no transmitido                             |
| qz | Indicador de cuantificador  |
| q  | Índice de cuantificación para los coeficientes de la transformada   |
| v  | Vector de movimiento  |

**Figura U.1/H.263 – Codificador de fuente para el modo selección mejorada de imagen de referencia**

El algoritmo de codificación de fuente de vídeo puede ser ampliado a la compensación de movimiento para multiimagen. La eficacia de la codificación mejorada se puede conseguir permitiendo la selección de imagen de referencia a nivel de macrobloque. Se utiliza un esquema de almacenamiento en memoria tampón con indexado relativo para un direccionamiento efectivo de imágenes en el tampón multiimagen.

En el primero de estos tipos de funcionamiento, la unidad de control de almacenamiento puede acomodar una "ventana deslizante" en el tiempo. En un esquema de almacenamiento de este tipo utilizando  $M$  memorias de imagen,  $PM_0 \dots PM_{M-1}$ , se almacenan en ellas las imágenes precedentes más recientes (hasta  $M$ ), decodificadas y reconstruidas, de forma que se puedan utilizar como referencias en la decodificación. Si el número máximo de imágenes acomodado en el tampón multiimagen es  $M$ , la estimación de movimiento al codificar una imagen  $m$ , si  $0 \leq m \leq M - 1$ , puede hacer uso de  $m$  imágenes. Al codificar una imagen  $m \geq M$ , se puede utilizar el máximo número de imágenes  $M$ . Alternativamente, se puede utilizar un segundo tipo de funcionamiento, el "control adaptativo de memoria" para un control más flexible y específico de las memorias de imagen que con el simple esquema de "ventana deslizante".

El funcionamiento del modo ERPS da como resultado la asignación del estado "sin usar" a algunas áreas de imágenes o de subimágenes de aquellas imágenes que han sido enviadas al decodificador. Una vez que se ha asignado el estado "sin usar" a algún área de imagen o a una imagen, el tren de bits no contendrá ningún dato que origine una referencia a ningún área "sin usar" con la finalidad de la predicción de subsiguientes imágenes. Gestionando la asignación del estado "sin usar" de las imágenes previas, el codificador asegurará de que se dispone de memoria suficiente en el decodificador para almacenar todos los datos necesarios para la representación de las imágenes subsiguientes. Se transfiere al decodificador el tamaño total de la memoria tampón y su estructura dentro del tren de bits, y el codificador deberá controlar la memoria tampón de manera que no se supere la capacidad total especificada con datos de imágenes almacenadas que no tengan asignado el estado "sin usar".

El codificador de fuente puede seleccionar una o varias de las memorias de imagen para eliminar la propagación temporal de errores causados por codificaciones entre imágenes. El modo decodificación de segmento independiente (véase el anexo R), que trata a los límites de los grupos de bloques (GOB) con encabezamientos no vacíos, o a las rebanadas como fronteras de imágenes, se puede utilizar para evitar la propagación espacial de errores debida a la compensación del movimiento a través de los límites de GOB o de rebanadas cuando se aplica este modo a una unidad menor que una imagen, tal como a un GOB o a una rebanada. En el tren de bits codificados se incluye la información de qué imagen se ha seleccionado para la predicción.

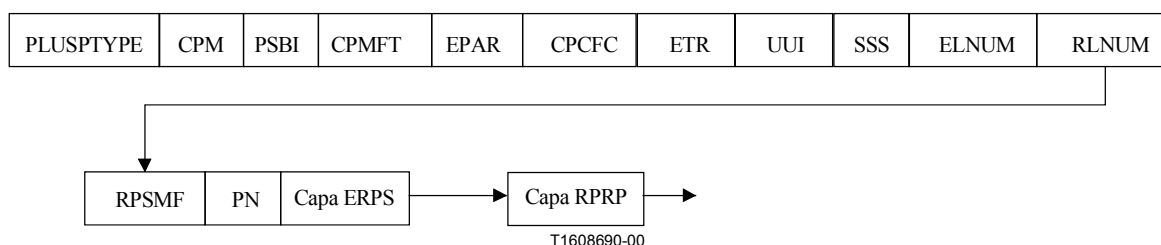
La estrategia utilizada por el codificador en la selección de la imagen, o las imágenes, utilizada en la predicción está fuera del alcance de esta Recomendación.

### U.3 Sintaxis del canal hacia adelante

La sintaxis se altera en las capas de imagen, grupo de bloques (GOB, *group of blocks*) y rebanada. Cuando viene indicado por un parámetro de múltiples imágenes de referencia activas (MRPA, *multiple reference pictures active*) con valor igual a "1", también se altera la sintaxis en la capa de macrobloque. En las capas de imagen, de GOB y de rebanada, se inserta una capa de selección mejorada de imagen de referencia (ERPS, *enhanced reference picture selection layer*). En la capa de macrobloque se insertan, en ciertas condiciones, parámetros de referencia de imagen para hacer posible la compensación de movimiento multiimagen.

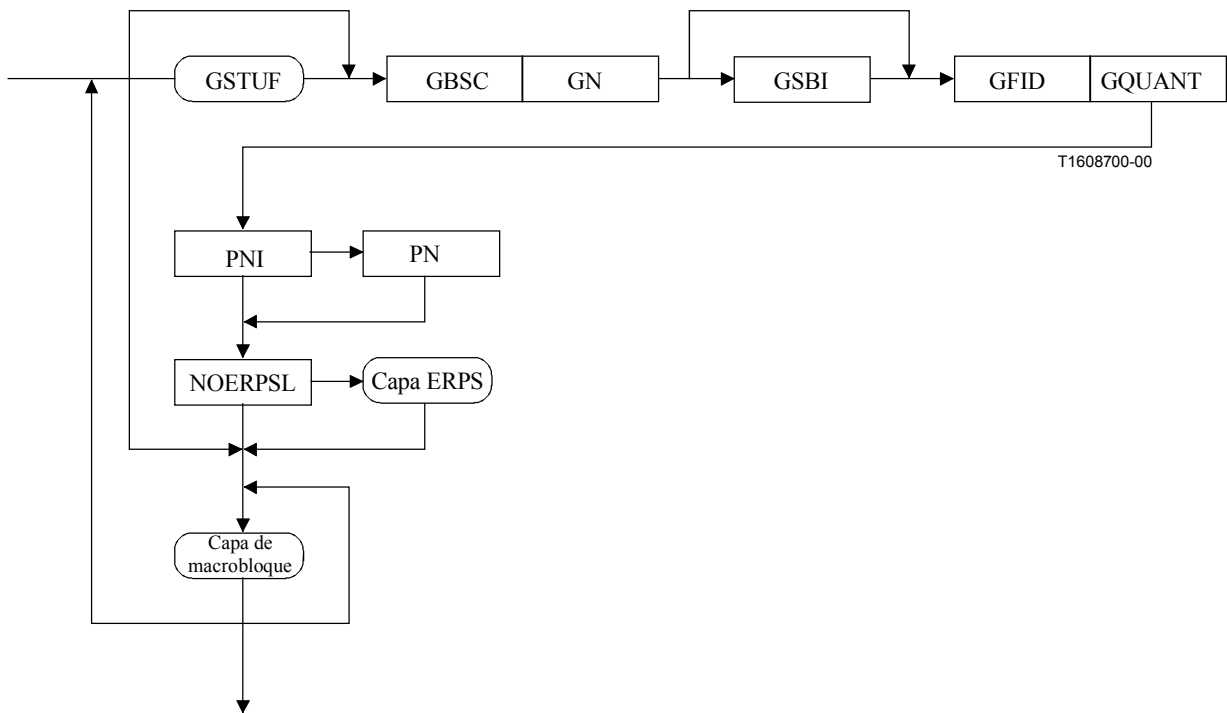
#### U.3.1 Sintaxis de las capas de imagen, de GOB, y de rebanada

La sintaxis de selección mejorada de imagen de referencia para el encabezamiento PLUS (en otro caso como se muestra en la figura 8) se muestra en la figura U.2. Los campos de RPSMF, PN, y la capa ERPS se insertan en el encabezamiento PLUS. No están presentes los campos de TRPI, TRP, BCI ni BCM (ya que sólo son necesarios para el modo RPS del anexo N, que no está permitido en caso de que el modo ERPS esté activo).



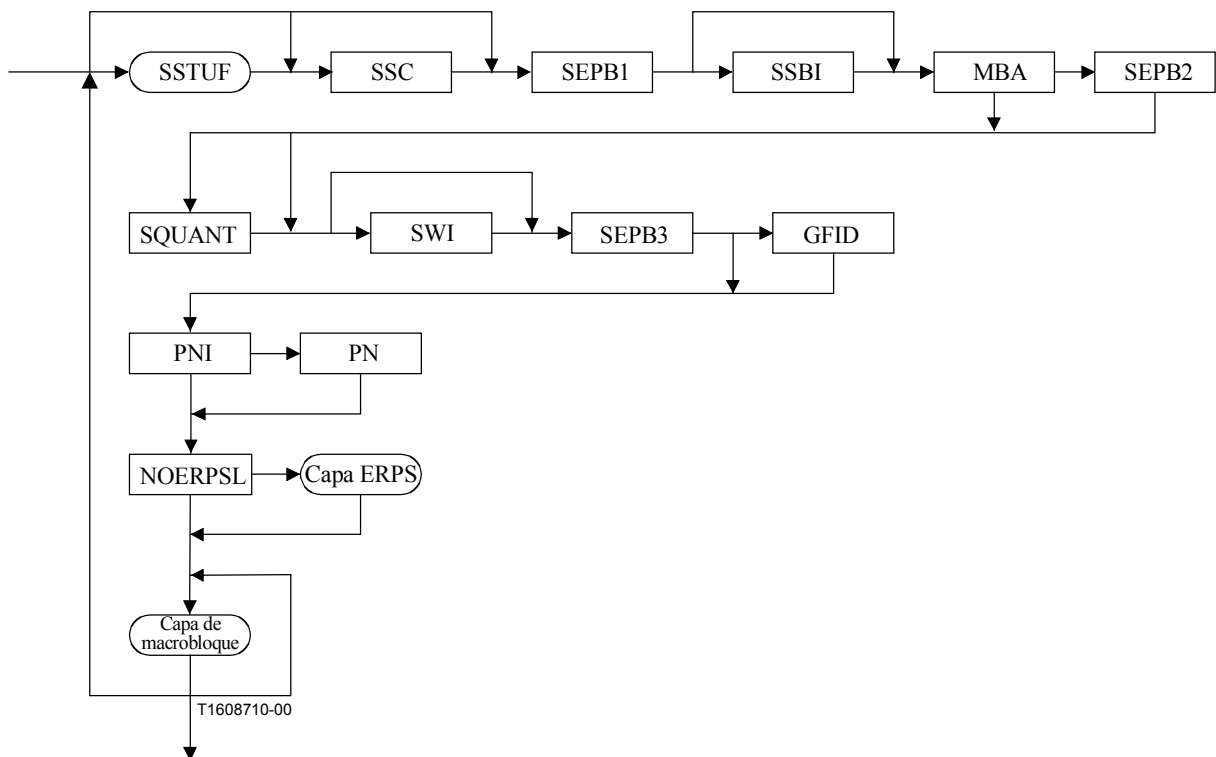
**Figura U.2/H.263 – Estructura del encabezamiento PLUS para el modo ERPS**

La sintaxis para la capa GOB se presenta en la figura U.3. Los campos de PNI, PN, NOERPSL, y de la capa ERPS se añaden a la sintaxis (en otro caso están definidos en la figura 9).



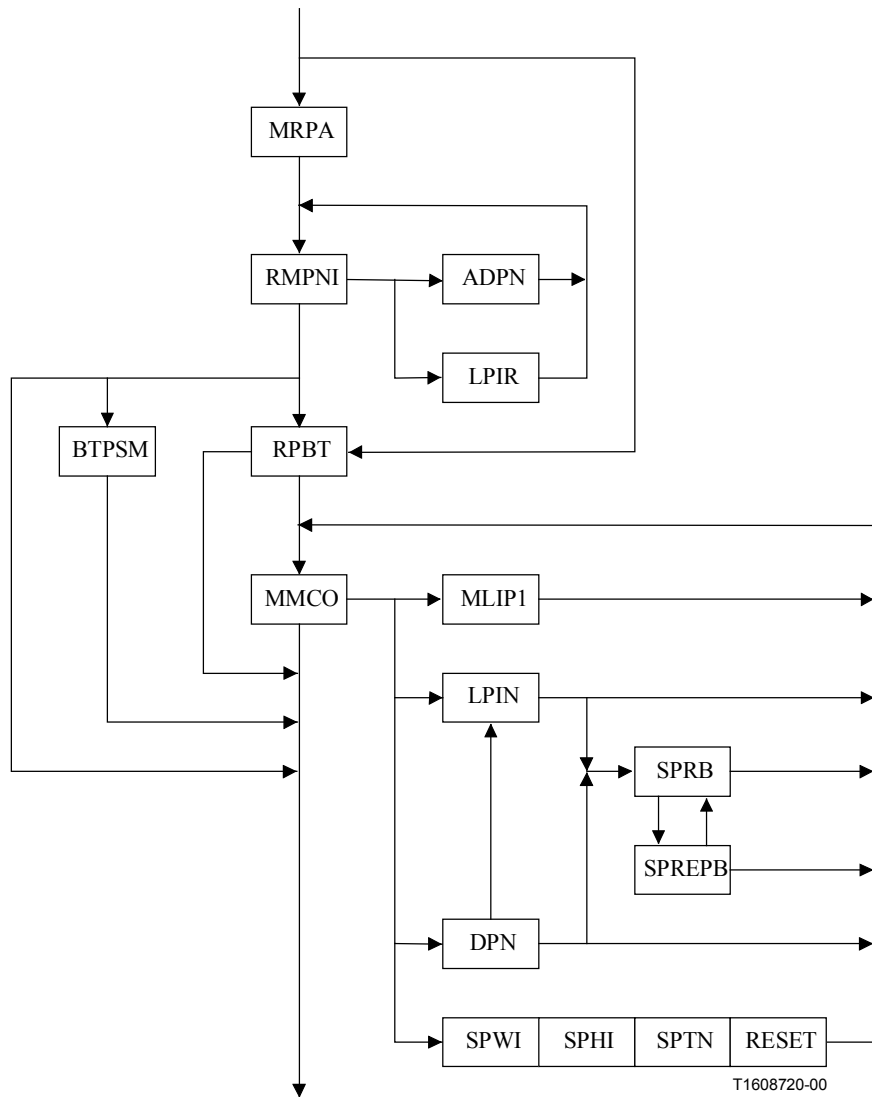
**Figura U.3/H.263 – Estructura de la capa de GOB para el modo ERPS**

Cuando se utiliza el modo opcional estructurada en rebanada (véase el anexo K), se modifica la sintaxis de la capa de rebanada de la misma forma que la de la capa de GOB. En la figura U.4 se presenta la sintaxis para la capa de rebanada. La rebanada que sigue inmediatamente al código de comienzo de imagen en el tren de bits incluye también todos los campos añadidos PNI, PN, NOERPSL, y la capa ERPS.



**Figura U.4/H.263 – Estructura de la capa de rebanada para el modo ERPS**

La capa ERPS se presenta en la figura U.5.



**Figura U.5/H.263 – Estructura de la capa ERPS**

En el cuadro U.1 se dan los códigos de longitud variable para los campos ADPN, LPIR, MLIP1, DPN, LPIN, SPTN, PR, PR<sub>0</sub>, PR<sub>2</sub>, PR<sub>3</sub>, PR<sub>4</sub>, PRB, y PRFW.

**Cuadro U.1/H.263 – Códigos de longitud variable para ADPN, LPIR, MLIP1, DPN, LPIN, SPTN, PR, PR<sub>0</sub>, PR<sub>2</sub>, PR<sub>3</sub>, PR<sub>4</sub>, PRB y PRFW**

Posición absoluta	Número de bits	Códigos
0	1	1
"x <sub>0</sub> " +1 (1:2)	3	0x <sub>0</sub> 0
"x <sub>1</sub> x <sub>0</sub> " +3 (3:6)	5	0x <sub>1</sub> 1x <sub>0</sub> 0
"x <sub>2</sub> x <sub>1</sub> x <sub>0</sub> " +7 (7:14)	7	0x <sub>2</sub> 1x <sub>1</sub> 1x <sub>0</sub> 0
"x <sub>3</sub> x <sub>2</sub> x <sub>1</sub> x <sub>0</sub> " +15 (15:30)	9	0x <sub>3</sub> 1x <sub>2</sub> 1x <sub>1</sub> 1x <sub>0</sub> 0
"x <sub>4</sub> x <sub>3</sub> x <sub>2</sub> x <sub>1</sub> x <sub>0</sub> " +31 (31:62)	11	0x <sub>4</sub> 1x <sub>3</sub> 1x <sub>2</sub> 1x <sub>1</sub> 1x <sub>0</sub> 0
"x <sub>5</sub> x <sub>4</sub> x <sub>3</sub> x <sub>2</sub> x <sub>1</sub> x <sub>0</sub> " +63 (63:126)	13	0x <sub>5</sub> 1x <sub>4</sub> 1x <sub>3</sub> 1x <sub>2</sub> 1x <sub>1</sub> 1x <sub>0</sub> 0
"x <sub>6</sub> x <sub>5</sub> x <sub>4</sub> x <sub>3</sub> x <sub>2</sub> x <sub>1</sub> x <sub>0</sub> " +127 (127:254)	15	0x <sub>6</sub> 1x <sub>5</sub> 1x <sub>4</sub> 1x <sub>3</sub> 1x <sub>2</sub> 1x <sub>1</sub> 1x <sub>0</sub> 0
"x <sub>7</sub> x <sub>6</sub> x <sub>5</sub> x <sub>4</sub> x <sub>3</sub> x <sub>2</sub> x <sub>1</sub> x <sub>0</sub> " +255 (255:510)	17	0x <sub>7</sub> 1x <sub>6</sub> 1x <sub>5</sub> 1x <sub>4</sub> 1x <sub>3</sub> 1x <sub>2</sub> 1x <sub>1</sub> 1x <sub>0</sub> 0
"x <sub>8</sub> x <sub>7</sub> x <sub>6</sub> x <sub>5</sub> x <sub>4</sub> x <sub>3</sub> x <sub>2</sub> x <sub>1</sub> x <sub>0</sub> " +511 (511:1022)	19	0x <sub>8</sub> 1x <sub>7</sub> 1x <sub>6</sub> 1x <sub>5</sub> 1x <sub>4</sub> 1x <sub>3</sub> 1x <sub>2</sub> 1x <sub>1</sub> 1x <sub>0</sub> 0
"x <sub>9</sub> x <sub>8</sub> x <sub>7</sub> x <sub>6</sub> x <sub>5</sub> x <sub>4</sub> x <sub>3</sub> x <sub>2</sub> x <sub>1</sub> x <sub>0</sub> " +1023 (1023:2046)	21	0x <sub>9</sub> 1x <sub>8</sub> 1x <sub>7</sub> 1x <sub>6</sub> 1x <sub>5</sub> 1x <sub>4</sub> 1x <sub>3</sub> 1x <sub>2</sub> 1x <sub>1</sub> 1x <sub>0</sub> 0
"x <sub>10</sub> x <sub>9</sub> x <sub>8</sub> x <sub>7</sub> x <sub>6</sub> x <sub>5</sub> x <sub>4</sub> x <sub>3</sub> x <sub>2</sub> x <sub>1</sub> x <sub>0</sub> " +2047 (2047:4094)	23	0x <sub>10</sub> 1x <sub>9</sub> 1x <sub>8</sub> 1x <sub>7</sub> 1x <sub>6</sub> 1x <sub>5</sub> 1x <sub>4</sub> 1x <sub>3</sub> 1x <sub>2</sub> 1x <sub>1</sub> 1x <sub>0</sub> 0

**U.3.1.1 Banderas del modo de selección de imagen de referencia (RPSMF, *reference picture selection mode flags*) (3 bits)**

RPSMF es una palabra de código de 3 bits de longitud fija, que está presente en el encabezamiento PLUS siempre que se utiliza el modo ERPS (independientemente del valor de UFEP). RPSMF no estará presente en la capa de GOB o en la de rebanada. Cuando esté presente, RPSMF indica qué tipo de mensajes de canal de retorno necesita el codificador. Los valores de RPSMF serán los definidos en 5.1.13.

**U.3.1.2 Indicador de número de imagen (PNI, *picture number indicator*) (1 bit)**

PNI es una palabra de código de un único bit de longitud fija, que está siempre presente en la capa de GOB o en la de rebanada cuando se utiliza el modo ERPS, y que no aparece en el encabezamiento PLUS. Cuando está presente, PNI indica si el siguiente campo PN está también presente o no.

"0": El campo PN no está presente.

"1": El campo PN está presente.

**U.3.1.3 Número de imagen (PN, *picture number*) (10 bits)**

PN es una palabra de código de 10 bits de longitud fija, que está siempre presente en el encabezamiento PLUS cuando se utiliza el modo ERPS, y que está presente en la capa de GOB o en la de rebanada sólo cuando así lo indique PNI.

Cada vez que se haya codificado y transmitido una imagen se incrementará PN en 1, en una operación en módulo 10 bits, y en base al PN de la imagen previamente almacenada. El término "imagen almacenada" se define en U.3.1.5.7. En el caso de imágenes EI y EP, el PN se incrementará partiendo del valor de la última imagen EI o EP almacenada dentro de la misma capa de mejora de escalabilidad. En el caso de imágenes B, se incrementará PN partiendo del valor de la imagen no B más recientemente almacenada en la capa de referencia de imagen B, que precede a la imagen B en el orden del tren de bits (una imagen que es temporalmente subsiguiente a la imagen B). Las imágenes B no se almacenan en la memoria tampón multiimagen, ya que no se utilizan

como referencias para las imágenes subsiguientes. Por tanto, una imagen que siga inmediatamente a una imagen B en la capa de referencia de imagen B, u otra imagen B que siga inmediatamente a una imagen B, tendrá el mismo PN que la imagen B. Del mismo modo, si se presenta una imagen no B en el tren de bits que no está almacenada, la imagen que sigue a esta imagen no B (en la misma capa de mejora, en el caso de funcionamiento según el anexo O) tendrá el mismo PN que la imagen no B que no haya sido almacenada.

En un escenario de uso conocido como "codificación de vídeo con redundancia", algunos codificadores pueden utilizar el modo ERPS, de forma tal que se envía más de una representación de la escena de la imagen en el mismo instante temporal (generalmente utilizando diferentes imágenes de referencia). En este caso, en el que está en uso el modo ERPS y en el que imágenes adyacentes en el tren de bits tienen la misma referencia temporal y el mismo número de imagen, el decodificador deberá considerar esta circunstancia como una indicación de que se han enviado copias redundantes con aproximadamente el mismo contenido de la escena representada, y decodificarán y utilizarán la primera de las imágenes recibidas, al tiempo que descarten la o las subsiguientes imágenes redundantes.

El PN sirve como identificador único para cada una de las imágenes almacenadas en el tampón de imágenes dentro de las 1024 imágenes codificadas y almacenadas. Por tanto, una imagen no puede mantenerse en la memoria tampón después de que se hayan codificado y almacenado 1023 imágenes subsiguientes (en la misma capa de mejora en el caso de funcionamiento según el anexo O) a no ser que se la haya asignado un índice de imagen de larga duración como se especifica más adelante. El codificador deberá asegurar que un tren de bits no especificará el mantenimiento de una imagen de corta duración después de que haya 1023 imágenes subsiguientes almacenadas. Un decodificador que encuentre un número de imagen en una imagen actualmente almacenada que tenga un valor igual al número de imagen de alguna otra imagen de corta duración dentro del tampón multiimagen, deberá tratar esta situación como un error.

#### **U.3.1.4 Capa de selección no mejorada de imagen de referencia (NOERPSL, *no enhanced reference picture selection layer*) (1 bit)**

NOERPSL es una palabra de código de un único bit de longitud fija, que está presente a nivel de GOB o de rebanada siempre que se utiliza el modo ERPS. No aparece en el encabezamiento PLUS. Los valores de NOERPSL serán los siguientes:

"0": Se envía la capa ERPS.

"1": No se envía la capa ERPS.

Si NOERPSL es "1", se aplicarán también al segmento de imagen de vídeo que sigue al GOB o a los datos de capa de la rebanada relevante todos los parámetros ERPS o reorganizaciones en curso que afecten a la imagen. De la información de la capa ERPS enviada a nivel de GOB o de rebanada rige el proceso de decodificación del segmento de imagen de vídeo precedido por los datos a nivel de GOB o rebanada, y no afecta al proceso de decodificación de ningún otro segmento de imagen de vídeo. (Véase en el anexo R la definición de segmento de imagen de vídeo.)

#### **U.3.1.5 Capa de selección mejorada de imagen de referencia (ERPS) (longitud variable)**

La capa ERPS está siempre presente a nivel de imagen cuando se utiliza el modo ERPS, y está presente a nivel de GOB o de rebanada si NOERPSL es "0". Especifica el tipo de indexación del tampón utilizado para la decodificación de la imagen o el segmento de imagen de vídeo actual, y gestiona los contenidos del tampón de imagen.

##### **U.3.1.5.1 Múltiples imágenes de referencia activas (MRPA, *multiple reference pictures active*) (1 bit)**

MRPA es una palabra de código de un único bit de longitud fija, que está presente sólo si el tipo de codificación de imagen indica que se trata de una imagen P, una imagen EP, una trama PB

mejorada o de una imagen B. MRPA especifica si el número de imágenes de referencia activas para la decodificación con predicción hacia adelante o para la predicción hacia atrás de la imagen o segmento de imagen de vídeo actual puede ser mayor que uno. El valor de MRPA será como sigue:

"1": Se puede utilizar más de una imagen de referencia para la compensación de movimiento hacia adelante o hacia atrás.

"0": Sólo se utiliza una imagen de referencia para la compensación de movimiento hacia adelante o hacia atrás. En este caso no son aplicables las extensiones de la sintaxis de la capa de macrobloque de U.3.2.

MRPA puede cambiar de segmento de imagen de vídeo a segmento de imagen de vídeo, de forma que diferentes segmentos de imagen de vídeo puedan direccionar a un número distinto de imágenes de referencia.

MRPA será "0" en cualquier imagen que invoque el modo repetición de muestreo de la imagen de referencia (véase el anexo P), e indicará a la misma imagen como imagen de referencia a utilizar hacia adelante, tanto a nivel de imagen, de GOB o de rebanada, para cualquiera de tales imágenes en curso. Si la imagen actual es una imagen B, la imagen de referencia hacia atrás deberá tener el mismo tamaño que la imagen actual y cualquier proceso de repetición de muestreo de imagen de referencia se aplicará únicamente a la imagen de referencia hacia adelante. La repetición de muestreo de imagen de referencia sólo se invocará si la memoria tampón multiimagen tiene una capacidad suficiente "sin usar" para el almacenamiento de la imagen de referencia hacia adelante que se vuelva a muestrear, aunque después de la utilización de la imagen de referencia remuestreada en la decodificación de la imagen actual, no se almacenará en la memoria tampón multiimagen la imagen de referencia hacia adelante remuestreada.

#### U.3.1.5.2 Repetición de muestreo del indicador de número de imagen (RMPNI, *re-mapping of picture numbers indicator*) (longitud variable)

RMPNI es una palabra de código de longitud variable, que está presente en la capa ERPS si la imagen es una imagen del tipo P, EP, PB mejorada o imagen B. RMPNI indica si es necesaria la reordenación de algún índice de imagen por defecto para la compensación de movimiento de la imagen actual o del segmento de imagen de vídeo, así de cómo se debe especificar la reordenación de los correspondientes índices dentro de la memoria tampón multiimagen, en caso de indicarlo. Se transmite el RMPNI con valores según el cuadro U.2. Si RMPNI indica la presencia de un campo ADPN o LPIR, un campo RMPNI adicional sigue inmediatamente al campo ADPN o LPIR.

**Cuadro U.2/H.263 – Operaciones RMPNI para la reorganización de imágenes de referencia**

Valor	Reorganización especificada
"1"	Está presente el campo ADPN y corresponde a una diferencia negativa a añadir al valor de predicción del número de imagen
"010"	Está presente el campo ADPN y corresponde a una diferencia positiva a añadir al valor de predicción del número de imagen
"011"	Está presente el campo LPIR y especifica el índice de larga duración para una imagen de referencia
"001"	Fin de bucle para la reorganización del orden por defecto del indexado relativo de imágenes

Un parámetro de referencia de imagen es un índice relativo dentro del conjunto ordenado de imágenes. Los campos RMPNI, ADPN y LPIR permiten que el orden de esta indexación relativa dentro de tampón multiimagen sea alterado temporalmente desde el orden de índices por defecto, con el fin de decodificar una determinada imagen o un determinado segmento de imagen de vídeo. El orden del índice por defecto es para que las imágenes de corta duración (es decir, imágenes a las

que no se las ha signado un índice de larga duración) precedan a las imágenes de larga duración en el orden de los índices de referencia. Dentro del conjunto de imágenes de corta duración, el orden por defecto es para que se ordenen las imágenes comenzando con la imagen de referencia almacenada en la memoria tampón más recientemente, continuando hasta la imagen de referencia más antigua (es decir, en orden decreciente del número de imagen, si no hay solapamiento en el campo de 10 bits del número de imagen). Dentro del conjunto de imágenes de larga duración, el orden por defecto es para que las imágenes sean ordenadas comenzando por la imagen con el índice de larga duración más pequeño y continuando hasta la imagen con índice de larga duración igual al valor más reciente de MLIP1-1.

Por ejemplo, si la memoria tampón contiene tres imágenes de corta duración con números 300, 302 y 303 (que fueron transmitidas en un orden creciente de número de imagen) y dos imágenes de larga duración con índices de imagen de larga duración 0 y 3, el orden de índices por defecto es:

- el índice relativo por defecto 0, se refiere a la imagen de corta duración con número de imagen 303;
- el índice relativo por defecto 1, se refiere a la imagen de corta duración con número de imagen 302;
- el índice relativo por defecto 2, se refiere a la imagen de corta duración con número de imagen 300;
- el índice relativo por defecto 3, se refiere a la imagen de larga duración con índice de imagen de larga duración 0, y
- el índice relativo por defecto 4, se refiere a la imagen de larga duración con índice de imagen de larga duración 3.

El primer campo ADPN o LPIR que se recibe (si se recibe alguno) mueve una determinada imagen fuera del orden por defecto, al índice relativo cero. El segundo campo de este tipo, mueve una imagen determinada al índice relativo uno, etc. El conjunto de las imágenes restantes que no han sido movidas al principio del orden de indexación relativa de esta manera, mantendrán su orden por defecto dentro de ellas mismas, y seguirán a las imágenes que han sido movidas al principio del tampón en orden de indexación relativa.

Si MRPA es "0", no más de un campo ADPN o LPIR estará presente en la misma capa ERPS, de no ser que la imagen actual sea una imagen B. Si la imagen actual es una imagen B y MRPA es "0", no deberán estar presentes más de dos campos ADPN o LPIR dentro de la misma capa ERPS.

Cualquier reorganización de números de imagen especificada para alguna de las imágenes, no deberá afectar al proceso de decodificación de ninguna otra imagen. Cualquier reorganización de números de imagen especificada para algún segmento de imagen de vídeo no deberá afectar el proceso de decodificación de ningún otro segmento de imagen de vídeo. Una reorganización de números de imagen especificada por una imagen, sólo deberá afectar al proceso de decodificación para cualquier segmento de imagen de vídeo dentro de esa imagen de dos formas:

- Si NOERPSL es "1" a nivel de GOB o de rebanada, entonces la reordenación especificada a nivel de imagen se utiliza también a nivel de segmento de imagen de vídeo asociado.
- Si la imagen es una imagen B, la reorganización especificada a nivel de imagen deberá especificar el cálculo del valor de  $TR_B$  y  $TR_D$  para la predicción bidireccional directa.

Una indicación "fin de bucle" RMPNI es el último elemento de la capa ERPS para una imagen B si MRPA es "0". En una imagen B con MRPA igual a "1", una indicación "fin de bucle" RMPNI es seguida de BTPSM. En una imagen P o EP o en una trama PB mejorada, una indicación "fin de bucle" RMPNI es seguida de RPBT.

Dentro de una capa ERPS, RMPNI no especificará la colocación de ninguna imagen de referencia en particular, en más de una posición reorganizada en orden de indexado relativo.



### U.3.1.5.3 Diferencia absoluta de números de imágenes (ADPN, *absolute difference of picture numbers*) (longitud variable)

ADPN es una palabra de código de longitud variable, que está presente sólo si lo indica RMPNI. ADPN sigue a RMPNI, cuando está presente. ADPN se transmite utilizando los valores del cuadro U.1, en el que el índice corresponde a  $ADPN - 1$ . ADPN representa la diferencia absoluta entre el número de imagen de la imagen actualmente reorganizada y el valor de predicción para ese número de imagen. Si no se han enviado campos ADPN previos dentro de la capa ERPS actual, el valor de predicción será el número de imagen de la imagen actual. Si ha sido enviado previamente algún campo ADPN, el valor de predicción será el número de la última imagen que fue reorganizada utilizando ADPN.

Si se llama PNP a la predicción del número de imagen, y se llama PNQ al número de la imagen en cuestión, el decodificador deberá determinar PNQ a partir de PNP y ADPN de una forma matemáticamente equivalente a la siguiente:

```
si (RMPNI == "1") { // diferencia negativa
  si (PNP - ADPN < 0)
    PNQ = PNP - ADPN + 1024;
  en otro caso
    PNQ = PNP - ADPN;
}en otro caso{ // diferencia positiva
  if (PNP + ADPN > 1023)
    PNQ = PNP + ADPN - 1024;
  en otro caso
    PNQ = PNP + ADPN;
}
```

El codificador deberá controlar los valores de RMPNI y ADPN de forma que el valor decodificado de ADPN no deberá ser mayor o igual a 1024.

Como un ejemplo de implementación, el codificador puede utilizar el siguiente proceso para determinar los valores de ADPN y RMPNI para determinar el número de imagen reorganizada de que se trata, PNQ:

```
DELTA = PNQ - PNP;
si (DELTA < 0) {
  si (DELTA < -511)
    MDELTA = DELTA + 1024;
  en otro caso
    MDELTA = DELTA;
}en otro caso{
  si (DELTA > 512)
    MDELTA = DELTA - 1024;
  en otro caso
    MDELTA = DELTA;
}
```

```
ADPN = abs(MDELTA);
```

donde `abs()` indica una operación de cálculo del valor absoluto. Hay que tener en cuenta que el índice en el cuadro U.1 corresponde al valor de  $ADPN - 1$ , en lugar del propio valor de ADPN.

Se determinará entonces el valor de RMPNI según el signo de MDELTA.

#### **U.3.1.5.4 Índice de imagen de larga duración para la reorganización (LPIR, *long-term picture index for re-mapping*) (longitud variable)**

LPIR es una palabra de código de longitud variable, que está presente sólo si lo indica RMPNI. LPIR sigue a RMPNI, cuando está presente. LPIR se transmite utilizando los valores del cuadro U.1. Representa el índice de imagen de larga duración que hay que reorganizar. El valor de predicción utilizado para cualquier reorganización subsiguiente de ADPN no se ve afectado por LPIR.

#### **U.3.1.5.5 Submodo predicción de dos imágenes de imagen B (BTPSM, *B-picture two-picture prediction sub-mode*) (1 bit)**

BTPSM es una palabra de código de un único bit de longitud fija, que sólo está presente en una imagen B (véase el anexo O) y sólo cuando MRPA es "1". Sigue a una indicación "fin de bucle" RMPNI y es el último elemento de la capa de ERPS de la imagen B cuando está presente. Indica si está en uso el submodo de predicción hacia atrás de dos imágenes para la imagen de la siguiente forma:

"0": Predicción hacia atrás con una imagen.

"1": Predicción hacia atrás con dos imágenes.

Si no está presente, BTPSM tiene un valor implícito de "0" (cuando MRPA es "0").

El conjunto de imágenes disponible para ser utilizadas como referencias en la predicción hacia adelante, es el conjunto de imágenes en la memoria tampón multiimagen distinto al conjunto de imágenes de referencia hacia atrás. El conjunto de imágenes de referencia hacia atrás viene determinado por el valor de BTPSM. Si BTPSM especifica predicción hacia atrás con una única imagen, la primera imagen en el orden de índice relativo (posiblemente reorganizada) es la única imagen de referencia hacia atrás. Si BTPSM especifica predicción hacia atrás con dos imágenes, las dos primeras imágenes en el orden de índice relativo (posiblemente reorganizadas) son las dos imágenes de referencia hacia atrás. El índice relativo para la predicción hacia adelante se convierte entonces en un índice relativo dentro del conjunto de imágenes de referencia hacia adelante.

Los contenidos de la memoria tampón multiimagen no son influidos por la presencia de una imagen B. La imagen B no se almacena en la memoria tampón multiimagen y no se utiliza como referencia para la codificación de subsiguientes imágenes.

#### **U.3.1.5.6 Tipo almacenamiento en memoria tampón de imagen de referencia (RPBT, *reference picture buffering type*) (1 bit)**

RPBT es una palabra de código de un único bit de longitud fija, que especifica el tipo de almacenamiento en memoria tampón utilizado con la imagen actual decodificada. Sigue a una indicación "fin de bucle" RMPNI cuando la imagen no es una imagen I, EI o B. Es el primer elemento de la capa de ERPS si la imagen no es una imagen I o EI. No está presente si la imagen es una imagen B. Los valores para RPBT se definen de la forma siguiente:

"0": Ventana deslizante.

"1": Control adaptativo de memoria.

En el almacenamiento en la memoria tampón del tipo "ventana deslizante", la imagen actual decodificada se añadirá a la memoria tampón con un índice 0, y la marcación de imágenes como "sin usar" en la memoria tampón, es llevada a cabo automáticamente en el conjunto de imágenes de corta duración según un esquema primera introducida – primera sacada. En este caso, si la memoria tampón tiene capacidad "sin usar" suficiente para almacenar la imagen actual, no se marcarán en la memoria tampón como "sin usar" más imágenes. Si la memoria tampón no tiene capacidad "sin usar" suficiente para almacenar la imagen actual, la imagen (o las imágenes, según se necesite, para liberar la cantidad necesaria de memoria en el caso de eliminación de subimágenes) con el índice por defecto mayor (o índices, según se necesite, en el caso de eliminación de subimágenes) dentro

de las imágenes de corta duración en la memoria tampón, se marcarán como "sin usar". En el caso de almacenamiento en el tampón del tipo de "ventana deslizante", no se transmite información adicional para el control del contenido de la memoria tampón.

En el almacenamiento en la memoria tampón del tipo de control adaptativo de memoria, el codificador especifica de forma explícita cualquier añadido a la memoria tampón o marcación de datos como "sin usar" en la memoria tampón, y puede asignar también índices de larga duración a imágenes de corta duración. El codificador puede especificar el marcado explícito como "sin usar" en la memoria tampón, a la imagen actual o a cualquier otra imagen. Este tipo de almacenamiento en la memoria tampón necesita de más información que es controlada por los parámetros de la operación de control de gestión de memoria (MMCO).

Si RPBT está presente en las capas de GOB o de rebanada, deberá ser el mismo que el de la capa de imagen. Cualquier instrucción MMCO presente en las capa de GOB o de rebanada, deberá indicar la misma operación que alguna instrucción MMCO de la capa de imagen.

Si la imagen es una imagen B, RPBT no estará presente y la imagen decodificada no se almacenará en la memoria tampón multiimagen. Esto asegura que ninguna imagen B afecte a los contenidos de la memoria tampón multiimagen.

Análogamente, la parte de imagen B de una trama PB mejorada, no se almacenará en la memoria tampón. Todos los campos de control asociados con el control de almacenamiento de una trama PB mejorada se considerarán como relacionados con el control del almacenamiento de sólo la parte de la imagen P de la trama PB mejorada.

#### **U.3.1.5.7 Operación de control de gestión de memoria (MMCO, *memory management control operation*) (longitud variable)**

MMCO es una palabra de código de longitud variable, que está presente sólo cuando RPBT indica "control adaptativo de memoria" y puede presentarse varias veces en caso de estar presente. Especifica una operación de control a aplicar en la gestión de la memoria tampón multiimagen. El parámetro MMCO viene seguido de los datos necesarios para la operación especificada por el propio valor de MMCO, y entonces sigue un parámetro MMCO adicional, hasta que el valor de MMCO indica el final de la lista de dichas operaciones. Las instrucciones MMCO no afectan a los contenidos de la memoria tampón ni al proceso de decodificación utilizado en la decodificación de la imagen actual; más bien, especifican las características necesarias de la memoria tampón para la decodificación de las subsiguientes imágenes en el tren de bits. En el cuadro U.3 se definen los valores y operaciones de control asociados con MMCO.

**Cuadro U.3/H.263 – Valores de operación de control de gestión de memoria (MMCO)**

<b>Valor</b>	<b>Operación de control de gestión de memoria</b>	<b>Campos siguientes de datos asociados</b>
"1"	Fin de bucle MMCO	Ninguno (fin de la capa ERPS )
"011"	Marcar una imagen de corta duración como "sin usar"	DPN
"0100"	Marcar una imagen de larga duración como "sin usar"	LPIN
"0101"	Asignar a una imagen un índice de larga duración	DPN y LPIN
"00100"	Marcar áreas de subimagen de corta duración como "sin usar"	DPN y SPRB
"00101"	Marcar áreas de subimagen de larga duración como "sin usar"	LPIN y SPRB
"00110"	Especificar el índice máximo de imagen de larga duración	MLIP1
"00111"	Especificar el tamaño y estructura del tampón	SPWI, SPHI, SPTN, y RESET

Todas las operaciones de control de gestión de memoria especificadas utilizando MMCO, se deberán especificar en la capa de imagen. Algunas o todas las mismas operaciones especificadas en la capa de imagen, pueden también estar especificadas en la capa de GOB o de rebanada, de la forma especificada en dicha capa de imagen (con los mismos datos asociados). MMCO no deberá especificar operaciones de memoria en la capa de GOB o de rebanada que no estén también especificadas con los mismos datos asociados en la capa de imagen.

Una instrucción MMCO, con la especificación del tamaño de la memoria tampón y de su estructura, deberá ser la primera instrucción MMCO, en caso de estar presente. En una determinada capa ERPS no deberá haber más que una instrucción MMCO con la especificación de la memoria tampón y de su estructura. En la primera imagen en la que se active el modo ERPS, dentro de una serie de imágenes de modo ERPS en el tren de bits, deberá estar presente una instrucción MMCO con la especificación del tamaño y la estructura de la memoria tampón y con RESET igual a "1". Una instrucción MMCO con la especificación del tamaño de la memoria tampón y su estructura con "RESET" igual a "1", deberá preceder cualquier uso de MMCO que indique el marcado de "sin usar" de cualquier imagen de corta duración o de larga duración. La anchura y la altura de subimagen especificadas en una instrucción MMCO de tamaño y estructura de la memoria tampón, no serán diferentes de los valores de estos parámetros en una instrucción MMCO previa de especificación de tamaño y estructura de la memoria tampón, a no ser que la imagen actual sea una imagen I o EI con RESET igual a "1". La altura y anchura de la imagen no deberán cambiar dentro del tren de bits, excepto dentro de una imagen que contenga una instrucción MMCO con la especificación del tamaño y la estructura del tampón con RESET igual a "1" (o dentro de una imagen en la que no se utilice el modo ERPS).

Si en el tren de bits está presente una imagen B que utiliza predicción hacia atrás con una única imagen, ésta deberá estar precedida en el orden del tren de bits, por exactamente una imagen no B temporalmente subsiguiente en la capa de referencia de la imagen B, como se especifica en O.2. No se presentarán operaciones de control de gestión de memoria dentro de ninguna capa ERPS de esta imagen no B inmediatamente subsiguiente en el tiempo dentro de la capa de referencia de la imagen B, que marque cómo "sin usar" ninguna parte de esa imagen no B sucesiva inmediatamente en el tiempo, ya que esa imagen de la capa de referencia es necesaria para la visualización hasta después de que se haya decodificado la imagen B.

Las reglas para el orden de la transmisión especificadas en O.2 se ajustan, según sea necesario, para imágenes B que hagan uso de la predicción hacia atrás utilizando dos imágenes. Si en el tren de bits está presente una imagen B que utilice la predicción hacia atrás usando dos imágenes, en el orden del tren de bits, deberán preceder a la imagen B exactamente dos imágenes no B subsiguientes en el tiempo dentro de la capa de referencia de la imagen B. Serán aplicables las demás restricciones en el orden de transmisión de la imagen B dentro del tren de bits especificadas en O.2, aunque adaptadas al uso de dos imágenes subsiguientes en el tiempo de la capa de referencia. No se presentarán operaciones de control de gestión de memoria dentro de ninguna capa ERPS de estas dos imágenes no B inmediatamente subsiguientes en el tiempo dentro de la capa de referencia de la imagen B que marque cómo "sin usar" ninguna parte de estas dos imágenes no B subsiguientes inmediatamente en el tiempo, ya que estas imágenes de la capa de referencia es necesaria para la visualización hasta después de que se haya decodificado la imagen B.

Se define como "imagen almacenada" a una imagen no B que no contiene una instrucción MMCO en su capa ERPS que marque a esa imagen como "sin usar". Si la imagen actual no es una imagen almacenada, su capa ERPS no contendrá ninguno de los siguientes tipos de instrucciones MMCO:

- una instrucción MMCO que especifique el tamaño y estructura de la memoria tampón con RESET igual a "1";
- toda instrucción MMCO que marque cualquier otra imagen como "sin usar" que no haya sido también marcada como "sin usar" en la capa ERPS de una imagen almacenada previamente;

- toda instrucción MMCO que asigne un índice de larga duración a una imagen a la que no se la haya asignado también el mismo índice de larga duración en la capa ERPS de una imagen almacenada previamente; o
- toda instrucción MMCO que marque áreas de subimagen de cualquier imagen como "sin usar" que no haya sido también marcada como "sin usar" en la capa ERPS de una imagen almacenada previamente.

#### **U.3.1.5.8 Diferencia de números de imagen (DPN, *difference of picture numbers*) (longitud variable)**

DPN está presente cuando lo indica MMCO. DPN sigue a MMCO, si está presente. DPN se transmite utilizando las palabras de código del cuadro U.1 y sirve para calcular el PN de una imagen para una operación de control de memoria. Se utiliza para asignar un índice de larga duración a una imagen, marcar una imagen de corta duración como "sin usar" o marcar áreas de subimágenes de imágenes de corta duración como "sin usar". Si el número de la imagen decodificada actual es PNC y el valor decodificado partiendo del cuadro U.1 es DPN, se utilizará una operación matemáticamente equivalente a las siguientes ecuaciones para el cálculo de PNQ, el número de la imagen especificada a calcular será:

```
si (PNC - DPN < 0)
    PNQ = PNC - DPN + 1024;
en otro caso
    PNQ = PNC - DPN;
```

Análogamente, el codificador puede calcular el valor de DPN a codificar utilizando la siguiente relación:

```
si (PNC - PNQ < 0)
    DPN = PNC - PNQ + 1024;
en otro caso
    DPN = PNC - PNQ;
```

Por ejemplo, si el valor decodificado de DPN es cero y MMCO indica que se debe marcar una imagen de corta duración como "sin usar", la imagen decodificada actual se marcará como "sin usar".

#### **U.3.1.5.9 Índice de imagen de larga duración (LPIN, *long-term picture index*) (longitud variable)**

LPIN está presente cuando lo indica MMCO. LPIN se transmite utilizando las palabras de código del cuadro U.1, y especifica el índice de una imagen de larga duración. Sigue a DPN si la operación es la de asignar un índice de larga duración a una imagen. Sigue a MMCO si la operación es la de marcado de una imagen de larga duración como "sin usar" o la de marcado de áreas de subimágenes de una imagen de larga duración como "sin usar".

#### **U.3.1.5.10 Mapa de bits de eliminación de subimágenes (SPRB, *sub-picture removal bit-map*) (longitud fija)**

SPRB es una palabra de código de longitud fija que contiene un bit para cada subimagen y está presente cuando lo indica MMCO. El número de bits de los datos SPRB viene determinado por los valores más recientes de SPWI y SPHI. Se utiliza SPRB para indicar qué áreas de subimagen de una imagen almacenada en la memoria tampón deben ser marcadas como "sin usar". SPRB sigue a DPN si la operación es el marcado de áreas de subimagen de una imagen de corta duración como "sin usar", y sigue a LPIN si la operación es el marcado de áreas de subimagen de una imagen de larga duración como "sin usar".

Las subimágenes se numeran según el orden de barrido, comenzando por la esquina superior izquierda de la imagen. Por ejemplo, se considera un caso en el que una imagen de referencia, especificada por DPN, está partida en seis subimágenes. Sea que "s<sub>1</sub> s<sub>2</sub> s<sub>3</sub> s<sub>4</sub> s<sub>5</sub> s<sub>6</sub>" representen seis bits de datos SPRB. Si el bit s<sub>i</sub> es "1", entonces el decodificador debería marcar la subimagen de orden i de la imagen de referencia indicada como "sin usar". Por ejemplo, si el SPRB es "000110", entonces las áreas de las subimágenes cuarta y quinta se marcan como "sin usar".

Para prevenir el comienzo de emulación de código, todos los bits necesarios de prevención de emulación de SPREPB deberán estar insertados dentro, o siguiendo los datos de SPRB, como se especifica en U.3.1.5.11.

Si está presente SPRB y la imagen especificada ha sido afectada previamente por un mapa de bits SPRB anterior, el mapa de bits especificado por SPRB deberá contener un "1" para cualquier área de subimagen que contuviese un "1" en el mapa de bits del SPBR previo. Cada mapa de SPRB deberá contener al menos un bit que tenga el valor "0" y al menos un bit que tenga el valor "1".

#### **U.3.1.5.11 Bit de prevención de emulación de eliminación de subimagen (SPREPB, *sub-picture removal emulation prevention Bit*) (un bit)**

SPREPB es una palabra de código de un único bit de longitud fija, que tiene el valor "1" y que se deberá insertar inmediatamente después de cualquier cadena con 8 bits cero consecutivos en los datos SPRB.

#### **U.3.1.5.12 Máximo índice más 1 de imagen de larga duración (MLIP1, *maximum long-term picture index plus 1*) (longitud variable)**

MLIP1 es una palabra de código de longitud variable que está presente si lo indica MMCO. MLIP1 sigue a MMCO, si está presente. MLIP1 se transmite utilizando palabras de código del cuadro U.1. Si está presente, se utiliza MLIP1 para determinar el máximo índice permitido para las imágenes de referencia de larga duración (hasta que se reciba otro valor de MLIP1). El decodificador deberá asumir inicialmente que MLIP1 es "0", hasta que se reciba otro valor. Al recibir un parámetro MLIP1, el decodificador deberá considerar a todas las imágenes de larga duración que tengan índices mayores que el valor decodificado de MLIP1 – 1 como "sin usar", como referencia para el proceso de decodificación de las imágenes subsiguientes. Para todas las demás imágenes en la memoria tampón multiimagen, MLIP1 no indicará ningún cambio en su estado.

#### **U.3.1.5.13 Indicación de anchura de subimagen (SPWI, *sub-picture width indication*) (7 bits)**

SPWI es una palabra de código de 7 bits que está presente si lo indica MMCO. SPWI sigue a MMCO, cuando se indique. SPWI especifica la anchura de una subimagen en unidades de 16 muestras de luminancia, de manera que la anchura indicada de la subimagen es  $16 \cdot (\text{SPWI} + 1)$  muestras de luminancia. La imagen actual tiene una anchura, expresada en unidades de subimagen igual a  $\text{ceil}(\text{ceil}(\text{pw}/16)/(\text{SPWI} + 1))$  subimágenes, siendo pw la anchura de la imagen y "/" indica una división en coma flotante. Para números positivos la función  $\text{ceil}(x)$  es igual a x si x es un entero y en otro caso  $\text{ceil}(x)$  es igual a uno más la parte entera de x. Si se ha negociado por medios externos (por ejemplo, utilizando Rec. UIT-T H.245) un tamaño de unidad de imagen mínima (MPU), definiendo la anchura y altura mínimas de una subimagen, la anchura de la subimagen especificada por SPWI será un entero múltiplo de la anchura de la MPU; de otro modo la anchura de la subimagen especificada por SPWI será tal que SPWI sea igual a  $\text{ceil}(\text{pw}/16) - 1$ .

#### **U.3.1.5.14 Indicación de altura de subimagen (SPHI, *sub-picture height indication*) (7 bits)**

SPHI es una palabra de código de 7 bits que está presente si también lo está SPWI (como lo indica MMCO). SPHI sigue a SPWI, si está presente. SPHI especifica la altura de una subimagen en unidades de 16 muestras de luminancia, de manera que la altura de la subimagen es  $16 \cdot \text{SPHI}$ . La gama de valores de SPHI permitida varía de 1 a 72. La imagen actual tiene una altura de  $\text{ceil}(\text{ceil}(\text{ph}/16)/\text{SPHI})$  subimágenes, siendo ph la altura de la imagen y "/" indica una división en

coma flotante. Si se ha negociado por medios externos (por ejemplo, utilizando Rec. UIT-T H.245) un tamaño de unidad de imagen mínima (MPU) definiendo la anchura y la altura mínimas de una subimagen, la altura de la subimagen especificada por SPHI deberá ser un entero múltiplo de la altura de la MPU; en otro caso la altura de la subimagen especificada por SPHI será tal que SPHI sea igual  $\text{ceil}(ph/16)$ .

#### **U.3.1.5.15 Número total de subimágenes (SPTN, *sub-picture total number*) (longitud variable)**

SPTN es una palabra de código de longitud variable que está presente si lo están SPWI y SPHI (como lo indica MMCO). SPTN sigue a SPHI, si está presente. SPTN se codifica utilizando el cuadro U.1, donde el índice del cuadro U.1 corresponde al valor decodificado de SPTN – 1. El valor decodificado de SPTN es el tamaño de la capacidad total operativa de la memoria tampón multiimagen expresado en unidades de subimágenes, como especifican SPWI y SPHI. La capacidad de memoria necesaria para la decodificación de las imágenes actuales no está incluida en SPTN (sólo la capacidad de memoria necesaria para el almacenamiento de las imágenes de referencia usadas para la predicción de otras imágenes). Cuando no se utiliza la eliminación de subimágenes (es decir, cuando SPWI y SPHI tienen las dimensiones de la imagen completa), el número máximo de imágenes de referencia de corta duración activas (por ejemplo, en el funcionamiento con ventana deslizante) viene dado, por tanto, por SPTN menos el número de imágenes que han sido asignadas a índices de larga duración y que no han sido posteriormente marcadas como "sin usar".

#### **U.3.1.5.16 Indicador de reiniciación de memoria tampón (RESET, *buffer reset indicator*) (1 bit)**

RESET es una palabra de código de un único bit de longitud que está presente si lo están SPWI, SPHI, y SPTN (como lo indica MMCO). RESET sigue a SPTN si está presente. Los valores de RESET serán los siguientes:

"0": Los contenidos de la memoria tampón no se reinician.

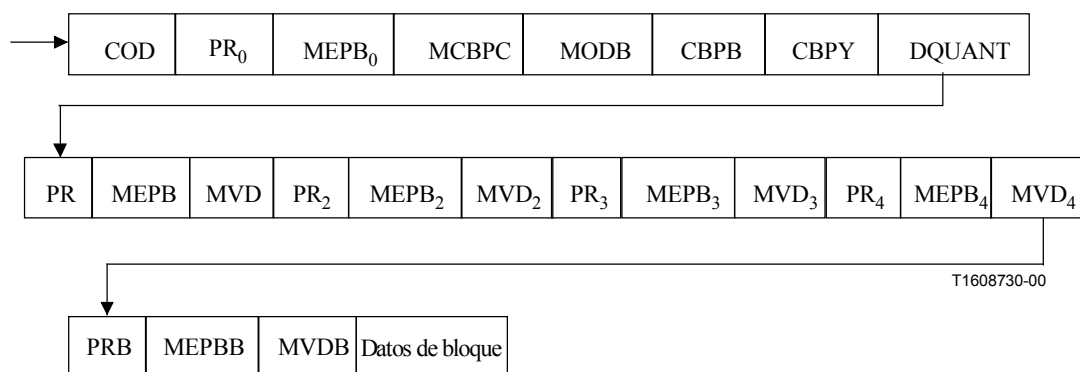
"1": Los contenidos de la memoria tampón se reinician.

Si RESET es "1", se deberán marcar como "sin usar" todas las imágenes de la memoria tampón multiimagen, incluyendo tanto las imágenes de corta como las de larga duración (aunque no la imagen actual, a no ser que se especifique por separado).

### **U.3.2 Sintaxis de capa macrobloque**

#### **U.3.2.1 Sintaxis de macrobloque de imágenes P y tramas PB mejoradas**

La sintaxis de la capa de macrobloque se modifica si está presente la capa ERPS para las imágenes P y para las tramas PB mejoradas cuando el número de imágenes de referencia hacia adelante seleccionadas pueda ser mayor que uno, como indica MRPA. El campo MRPA está indicado en la capa ERPS. En la figura U.6 se presenta la sintaxis de la capa de macrobloque cuando MRPA es "1". En otro caso, el formato de la sintaxis de macrobloque en una imagen P, o en una trama PB mejorada, no difiere de la presentada en la figura 10.



**Figura U.6/H.263 – Estructura de la capa de macrobloque de imagen P y trama PB mejorada para el modo ERPS**

### U.3.2.1.1 Interpretación de COD

Si el bit COD es "1", no se transmite más información para el macrobloque. En este caso, el decodificador tratará al macrobloque como un macrobloque INTER, con el vector de movimiento para todo el macrobloque igual a cero, el parámetro de referencia de imagen igual a cero y sin datos de coeficiente. Si el bit COD es "0", indicando que el macrobloque está codificado, la sintaxis de la capa de macrobloque se muestra en la figura U.6, estando los campos  $PR_0$ , PR,  $PR_2$ ,  $PR_3$ ,  $PR_4$ , y PRB incluidos en la sintaxis.  $PR_0$ , PR,  $PR_2$ ,  $PR_3$ ,  $PR_4$ , y PRB cada uno consta de una palabra de código de longitud variable como se indica en el cuadro U.1.

### U.3.2.1.2 Parámetro 0 de referencia de imagen ( $PR_0$ , *picture reference parameter 0*) (longitud variable)

$PR_0$  es una palabra de código de longitud variable como se especifica en el cuadro U.1. Está presente siempre que COD es "0". Si  $PR_0$  tiene un valor decodificado igual a cero (palabra de código "1"), indica que seguirá más información para el macrobloque. Si la decodificación no resulta ser cero, indica la codificación del macrobloque utilizando sólo un parámetro de referencia de imagen.

Si el campo  $PR_0$  no tiene un valor decodificado igual a cero (palabra de código "1"), no se transmite más información para este macrobloque. En este caso el decodificador tratará al macrobloque como a un macrobloque INTER con el vector de movimiento para el bloque completo igual a cero, el parámetro de referencia de imagen igual a  $PR_0$ , y sin datos de coeficiente.

Si el campo  $PR_0$  tiene un valor decodificado igual a cero (palabra de código "1"), el macrobloque está codificado. El significado y uso de los campos MCBPC, CBPB, CBPY, y DQUANT permanece inalterado. Se incluye el campo PR junto con el campo MVD en todos los macrobloques INTER (y en el modo tramas PB mejoradas también para los macrobloques INTRA). En U.3.2.1.4 se describe el uso de MODB en tramas PB mejoradas.

### U.3.2.1.3 Bit 0 de prevención de emulación de macrobloque ( $MEPB_0$ , *macroblock emulation prevention bit 0*) (1 bit)

$MEPB_0$  es una palabra de código de un único bit de longitud fija que tiene el valor "1", que sigue a  $PR_0$  si, y sólo si,  $PR_0$  está presente y tiene un valor decodificado igual a "1" (palabra de código "000"), y se cumple alguna de las dos condiciones siguientes:

- 1) está en uso el modo estructurado de rebanada (véase el anexo K); o



- 2) la COD para el macrobloque actual sigue inmediatamente después de otro macrobloque que también tiene COD = "0" y PR<sub>0</sub> = "1" (palabra de código "000"), y el PR<sub>0</sub> del macrobloque previo no está seguido por un bit MEPB<sub>0</sub>.

La finalidad de MEPB<sub>0</sub> es la de prevenir la emulación del código de comienzo y, en el modo estructurado de rebanada, ayudar a la determinación del número de macrobloques en una rebanada.

#### **U.3.2.1.4 Parámetros de referencia de imagen de macrobloque (PR, PR<sub>2-4</sub>, y PRB, *macroblock picture reference parameters*) (longitud variable)**

PR es el parámetro primario de referencia de imagen. PR está presente cuando lo está MVD. Las tres palabras de código PR<sub>2-4</sub> están incluidas junto con MVD<sub>2-4</sub> si así lo indica PTYPE, y si MCBPC especifica un macrobloque INTER4V o INTER4V+Q (un macrobloque del tipo 2 ó 5 en los cuadros 8 y 9). PR<sub>2-4</sub> y MVD<sub>2-4</sub> sólo están presentes cuando se opera en el modo predicción avanzada (véase el anexo F) o en el modo filtro de desbloqueo (véase el anexo J). PRB sólo está presente en una trama PB mejorada cuando MODB indica que MVDB está presente. Cada uno de los parámetros PR, PR<sub>2-4</sub>, y PRB especifican un índice relativo de referencia de imagen dentro de la memoria tampón multiimagen.

Si el macrobloque no es del tipo INTER4V o INTER4V+Q, PR se utiliza como el parámetro de referencia de imagen para compensación de movimiento en todo el macrobloque. Si el macrobloque es del tipo INTER4V o INTER4V+Q, PR se utiliza en la predicción compensada de movimiento del primero de los cuatro bloques de luminancia 8 × 8 en el macrobloque y para los dos bloques de crominancia del macrobloque (por otra parte, con el proceso de compensación de movimiento como se especifica en 6.1). PR<sub>2-4</sub> se utilizan en la compensación de movimiento de los restantes tres bloques 8 × 8 de datos de luminancia en el macrobloque. Si MODB indica que está presente MVDB, PRB es el parámetro de referencia de imagen para la predicción hacia adelante de la parte B de la trama PB mejorada.

En tramas PB mejoradas, cuando MODB indica predicción bidireccional B<sub>PB</sub>, los valores de TR<sub>D</sub> y TR<sub>B</sub> se calcularán como los incrementos de referencia temporal basados en los datos de referencia temporal de la imagen actual y en los de la imagen de referencia previa más reciente, con independencia de que la imagen de referencia previa más reciente haya sido, o no, reorganizada con un orden de índice relativo diferente, marcada como "sin usar", o asignada a un índice de larga duración. La imagen utilizada como imagen de referencia hacia adelante para la predicción bidireccional B<sub>PB</sub> en tramas PB mejoradas PB será la imagen especificada por PR.

#### **U.3.2.1.5 Bits de prevención de emulación de macrobloque (MEPB, MEPB<sub>2-4</sub>, y MEPBB, *macroblock emulation prevention bits*) (1 bit cada uno)**

MEPB, MEPB<sub>2-4</sub>, y MEPBB tienen cada uno un único bit con el valor "1" si están presentes. Cada uno de ellos estará presente si, y sólo si, no se utiliza el modo vector de movimiento sin restricción (véase el anexo D) y el campo asociado PR, PR<sub>2-4</sub>, o PRB está presente y tiene el valor decodificado "1" (palabra de código "000"). Su finalidad es la de impedir la emulación del código de comienzo.

#### **U.3.2.2 Sintaxis de macrobloque de imagen B e imagen EP**

La sintaxis de la capa de macrobloque para imágenes B y PB (véase el anexo O) se modifica en una forma similar a la utilizada en las imágenes P. Si el bit COD es igual a "1", indica un macrobloque saltado, como se define en el anexo O, utilizando un parámetro de referencia de imagen igual a cero para la predicción hacia adelante (con salto) en una imagen EP, y por la parte hacia adelante de la predicción bidireccional directa (con salto) en una imagen B y utilizando la primera imagen de predicción hacia atrás para la parte hacia atrás de la predicción bidireccional directa (con salto) en una imagen B (en el caso de predicción hacia atrás de dos imágenes, como es cuando está presente

BSBBW y es igual a "0"). Si COD es "0", se inserta un parámetro  $PR_0$  en la sintaxis, y se utiliza de forma similar a la descrita en U.3.2.1.2. Si  $PR_0$  está presente y no tiene un valor decodificado igual a cero (palabra de código "1"), indica que se debe hacer la predicción del macrobloque con predicción INTER hacia adelante utilizando un vector de movimiento de valor cero y un parámetro de referencia de imagen igual a  $PR_0$ . Si  $PR_0$  tiene un valor decodificado de cero, sigue MBTYPE, y especifica el tipo de macrobloque. El formato de los campos CBPC, CBPY, y DQUANT permanece sin cambios. Los campos MVDFW y MVDBW están codificados de la misma forma que cuando no se utiliza el modo ERPS, pero cada uno de ellos se utiliza conjuntamente con una referencia de imagen, y, posiblemente con un bit de prevención de emulación.

En el caso de una imagen B, las imágenes de referencia hacia atrás en la memoria tampón multiimagen se definen de la siguiente forma:

- en el caso de predicción hacia atrás con una sola imagen, sólo hay una imagen de referencia hacia atrás, que es la primera imagen en orden del índice relativo (posiblemente reorganizada); y
- en el caso de predicción hacia atrás de dos imágenes, hay dos imágenes de referencia hacia atrás, que son las dos primeras imágenes en orden del índice relativo (posiblemente reorganizadas).

Se definen las imágenes de referencia hacia adelante en la memoria tampón multiimagen, como las imágenes de la memoria tampón multiimagen distintas de las imágenes de referencia hacia atrás. La indexación relativa para la predicción hacia adelante es un índice relativo dentro del conjunto de imágenes de referencia hacia adelante, y el índice relativo para la predicción hacia atrás es un índice relativo dentro del conjunto de imágenes de referencia hacia atrás.

Por ejemplo, si la memoria tampón contiene tres imágenes de corta duración con números de imagen de corta duración 300, 302 y 303 (que fueron transmitidas con orden de número de imagen creciente) y dos imágenes de larga duración con índices de imagen de larga duración 0 y 3, el orden del índice por defecto en el caso de predicción hacia atrás de dos imágenes es:

- el índice relativo hacia atrás por defecto 0, se refiere a la imagen de corta duración con número de imagen 303;
- el índice relativo hacia atrás por defecto 1, se refiere a la imagen de corta duración con número de imagen 302;
- el índice relativo hacia adelante por defecto 0, se refiere a la imagen de corta duración con número de imagen 300;
- el índice relativo hacia adelante por defecto 1, se refiere a la imagen de larga duración con índice de imagen de larga duración 0; y
- el índice relativo hacia adelante por defecto 2, se refiere a la imagen de larga duración con índice de imagen de larga duración 3;

y en el caso de predicción hacia atrás de una única imagen:

- la imagen de referencia hacia atrás única es la imagen de corta duración con número de imagen 303;
- el índice relativo hacia adelante por defecto 0, se refiere a la imagen de corta duración con número de imagen 302;
- el índice relativo hacia adelante por defecto 1, se refiere a la imagen de corta duración con número de imagen 300;
- el índice relativo hacia adelante por defecto 2, se refiere a la imagen de larga duración con índice de imagen de larga duración 0, y
- el índice relativo hacia adelante por defecto 3, se refiere a la imagen de larga duración con índice de imagen de larga duración 3;

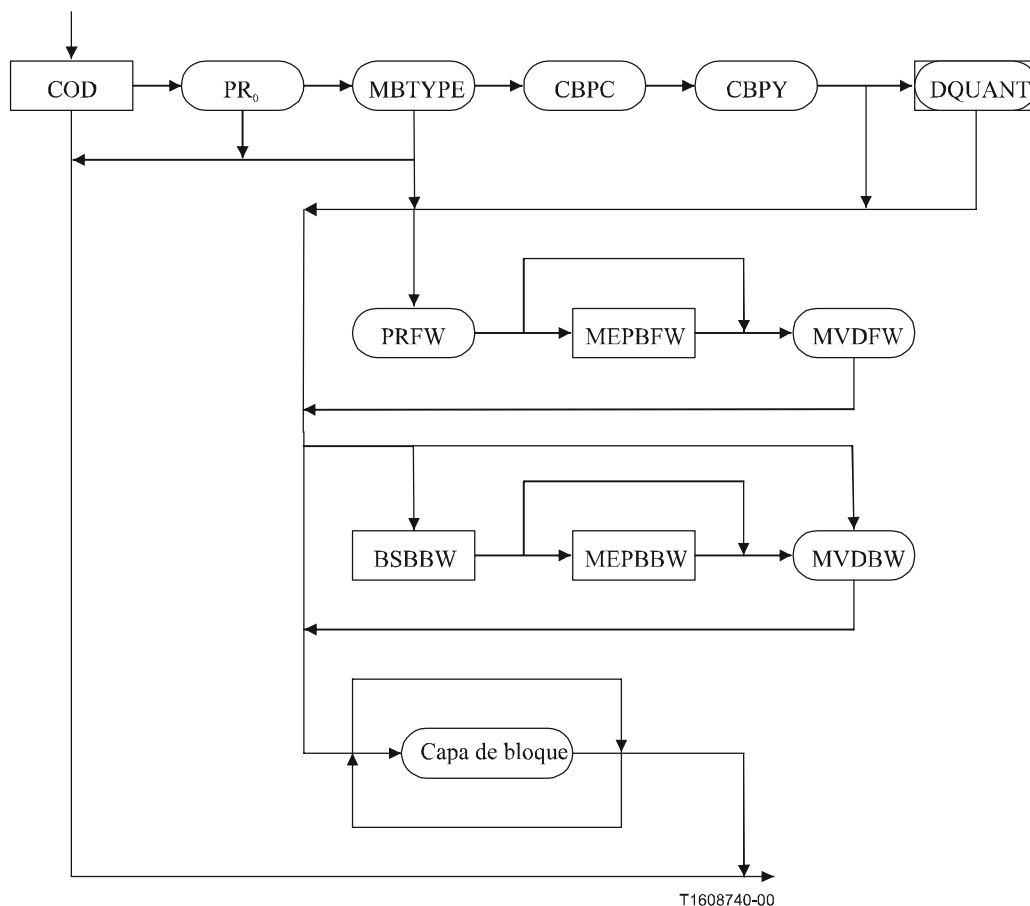
y si se han reorganizado estas imágenes a un nuevo orden de índices de imagen de corta duración 302, seguida por la imagen de corta duración 303, seguida por la imagen de larga duración 0, seguida por la imagen de corta duración 300, seguida por la imagen de larga duración 3, el nuevo orden relativo de los índices en el caso de predicción hacia atrás de dos imágenes es:

- el índice relativo hacia atrás reorganizado 0, se refiere a la imagen de corta duración con número de imagen 302;
- el índice relativo hacia atrás reorganizado 1, se refiere a la imagen de corta duración con número de imagen 303;
- el índice relativo hacia adelante reorganizado 0, se refiere a la imagen de larga duración con índice de imagen de larga duración 0;
- el índice relativo hacia adelante reorganizado 1, se refiere a la imagen de corta duración con número de imagen 300; y
- el índice relativo hacia adelante reorganizado 2, se refiere a la imagen de larga duración con índice de imagen de larga duración 3;

y en el caso de predicción hacia atrás de una única imagen

- la imagen de referencia hacia atrás única reorganizada, es la imagen de corta duración con número de imagen 302;
- el índice relativo hacia adelante reorganizado 0, se refiere a la imagen de corta duración con número de imagen 303;
- el índice relativo hacia adelante reorganizado 1, se refiere a la imagen de larga duración con índice de imagen de larga duración 0;
- el índice relativo hacia adelante reorganizado 2, se refiere a la imagen de corta duración con número de imagen de corta duración 300; y
- el índice relativo hacia adelante reorganizado 3, se refiere a la imagen de larga duración con índice de imagen de larga duración 3.

El  $TR_D$  utilizado para la predicción bidireccional directa en una imagen B se calculará como el incremento de referencia temporal entre la primera imagen de referencia hacia adelante en el orden del índice relativo (posiblemente reorganizado) y la primera imagen de referencia hacia atrás en el orden del índice relativo (posiblemente reorganizado) (es decir, si se utiliza predicción hacia atrás de dos imágenes, ésta sería la imagen referenciada cuando BSBBW es "0", como se describe en U.3.2.2.3). El  $TR_B$  utilizado en la predicción bidireccional directa en una imagen B se calculará como el incremento de referencia temporal entre la imagen B y la primera imagen de referencia hacia adelante en el orden del índice relativo (posiblemente reorganizado). El orden del índice relativo utilizado en el cálculo de  $TR_D$  y  $TR_B$  será el especificado en la capa ERPS a nivel de imagen de la sintaxis de la imagen B (es decir, reorganizaciones a nivel de GOB o de rebanada no afectarán a los valores de  $TR_D$  y  $TR_B$ ). (Véase la figura U.7.)



**Figura U.7/H.263 – Estructura de la capa de macrobloque de imagen EP e imagen B para el modo ERPS**

#### **U.3.2.2.1 Referencia de imagen en la predicción hacia adelante (PRFW, *picture reference for forward prediction*) (longitud variable)**

PRFW es un parámetro de referencia de imagen, de longitud variable, que está presente siempre que estén presentes datos del vector de movimiento hacia adelante, y se codifica utilizando el cuadro U.1. PRFW es un índice relativo dentro del conjunto de imágenes de referencia hacia adelante.

#### **U.3.2.2.2 Bit de prevención de emulación para la predicción hacia adelante (MEPBFW, *emulation prevention bit for forward prediction*) (1 bit)**

MEPBFW es una palabra de código de un único bit de longitud fija, que tiene el valor "1" y que estará insertada después de PRFW si (y sólo si) PRFW está presente y tiene un valor decodificado igual a "1" (palabra de código "000") y no se está utilizando el modo vector de movimiento sin restricción (véase el anexo D).

#### **U.3.2.2.3 Bit de selección de imagen B para la predicción hacia atrás (BSBBW, *B-picture selection bit for backward prediction*) (1 bit)**

BSBBW es una palabra de código de un único bit de longitud fija, que está presente sólo para las imágenes B cuando MVDBW está presente, y sólo cuando se especifica predicción hacia atrás de dos imágenes en el tratamiento de la imagen B. El significado de este bit es:

"0": Predicción desde la primera imagen de referencia hacia atrás, en el orden de índice relativo (en el orden por defecto, esto sería utilizando la imagen de referencia de corta duración más reciente, si a esa imagen no se la ha asignado un índice de larga duración o ha sido marcada como "sin usar").

"1": Predicción desde la segunda imagen de referencia hacia atrás, en orden de índice relativo (en el orden por defecto, esto sería utilizando la segunda imagen de referencia de corta duración más reciente, si a ninguna de las dos últimas imágenes de referencia se las ha asignado índices de larga duración o han sido marcadas como "sin usar").

#### **U.3.2.2.4 Bit de prevención de emulación para la predicción hacia atrás (MEPBBW, *emulation prevention bit for backward prediction*) (1 bit)**

MEPBBW es una palabra de código de un único bit de longitud fija, que tiene el valor "1" y que está presente sólo bajo las siguientes condiciones:

- BSBBW está presente y es igual a "0"; y
- no está en uso el modo vector de movimiento sin restricción (véase el anexo D); y
- BSBBW está precedido de cinco bits con el valor "00000".

#### **U.4 Proceso de decodificación**

El decodificador para el modo ERPS almacena las imágenes de referencia para la decodificación entre imágenes en una memoria tampón multiimagen. El decodificador puede necesitar capacidad de memoria adicional para el almacenamiento de las múltiples imágenes decodificadas (en comparación con la capacidad de memoria necesaria si no se soporta el modo ERPS). El decodificador replica la memoria tampón multiimagen del codificador en función del tipo de almacenamiento en memoria tampón de la imagen de referencia utilizado y de cualesquiera operaciones de control de gestión de memoria especificadas en el tren de bits. El esquema de almacenamiento en memoria tampón también puede ser manejado cuando se decodifican imágenes parcialmente erróneas.

Se asigna un número de imagen (PN) a cada imagen transmitida y almacenada, que se almacena con la imagen en la memoria tampón multiimagen. PN representa un identificador de cuenta secuencial de imágenes para las imágenes almacenadas. PN está limitado, haciendo uso de operaciones aritméticas de módulo 1024. Para la primera imagen transmitida PN deberá ser "0". Para cada una de las demás imágenes transmitidas y almacenadas, se incrementará PN en 1 (dentro de una determinada capa de escalabilidad en caso de que se haga uso del anexo O). Si la diferencia (módulo 1024) de los PN de dos imágenes recibidas consecutivamente y almacenadas no es 1, el decodificador deberá suponer que se ha producido una pérdida de imágenes o una corrupción de los datos. En este caso, se puede enviar al codificador, por un canal de retorno, un mensaje que indique la pérdida de imágenes.

Además del PN, cada imagen almacenada en la memoria tampón multiimagen, tiene asociado un índice, llamado el índice por defecto. Cuando una imagen es añadida por primera vez a la memoria tampón multiimagen se le da un índice por defecto 0 (a no ser que se la asigne a un índice de larga duración). Los índices de las imágenes en la memoria tampón multiimagen se modifican al añadir o retirar imágenes de él.

Las imágenes almacenadas en las memorias tampón multiimagen se pueden dividir también en dos categorías: imágenes de larga duración e imágenes de corta duración. Una imagen de larga duración puede permanecer en la memoria tampón multiimagen durante un largo periodo de tiempo (más de 1023 intervalos de imagen, codificadas y almacenadas). Inicialmente se considera a la imagen actual como una imagen de corta duración. Se puede modificar cualquier imagen de corta duración a imagen de larga duración asignándole un índice de larga duración de acuerdo con la información recibida en el tren de bits. El PN es el único identificador para todas las imágenes de corta duración en la memoria tampón multiimagen. Cuando se cambia una imagen de corta duración a imagen de larga duración, se le asigna también un índice de imagen de larga duración (LPIN, *long-term picture index*). Se asigna a una imagen un índice de imagen de larga duración asociando su PN a un LPIN. Una vez que se ha asignado a una imagen un índice de imagen de larga duración, el único uso posterior posible del PN de la imagen de larga duración dentro del tren de bits, será por medio

de una repetición de la asignación de índices de larga duración. Los PN de las imágenes de larga duración son únicos dentro de las 1024 imágenes transmitidas y almacenadas. Por tanto, el PN de una imagen de larga duración no puede ser utilizado para ser asignado a un índice de larga duración después de 1023 imágenes transmitidas y almacenadas subsiguientemente. LPIN se convierte en el único identificador durante la vida de una imagen de larga duración.

PN (en imágenes de corta duración) o LPIN (en imágenes de larga duración) se pueden utilizar para la reorganización de imágenes a índices reorganizados para el direccionamiento eficaz de imágenes de referencia.

#### **U.4.1 Procesos del decodificador para la gestión de imágenes de corta/larga duración**

El decodificador puede tener tanto imágenes de corta duración como imágenes de larga duración en su memoria tampón multiimagen. El campo MLIP1 se utiliza para indicar el índice máximo permitido en la memoria tampón para imágenes de larga duración. Si no se ha enviado ningún valor previo de MLIP1, no habrá en uso imágenes de larga duración, es decir, MLIP1 tendrá inicialmente un valor implícito de "0" al ser invocado el modo ERPS. A la recepción de un parámetro MLIP1, un nuevo MLIP1 estará en vigor hasta que se reciba otro valor de MLIP1. Cuando se reciba un nuevo parámetro MLIP1 en el tren de bits, se deberán considerar marcadas como "sin usar" todas las imágenes de larga duración que estén asociadas con índices de larga duración mayores o iguales a MLIP1. Está fuera del objetivo de esta Recomendación la frecuencia a la que se transmite MLIP1. Sin embargo, el codificador deberá enviar un parámetro MLIP1 al recibir un mensaje de error, tal como un mensaje de petición INTRA.

Una imagen de corta duración puede ser cambiada a imagen de larga duración utilizando una instrucción MMCO con un DPN y LPIN asociados. El número de imagen de corta duración se obtiene a partir de DPN, y el índice de imagen de larga duración es LPIN. Al recibir una instrucción MMCO de este tipo, el decodificador cambiará la imagen de corta duración con el PN indicado por DPN a una imagen de larga duración, a la que asignará al índice de larga duración indicado por LPIN. Si ya existe en la memoria tampón una imagen de larga duración con el mismo índice de larga duración, se marcará como "sin usar" a la imagen de larga duración preexistente. Un codificador no asignará a ninguna imagen un índice de larga duración mayor que  $MLIP1 - 1$ . El decodificador deberá tratar como error la condición de que LPIN sea mayor que  $MLIP1 - 1$ . Para mejorar la elasticidad a los errores, el codificador puede enviar repetidamente la operación de asignación de índice de larga duración o el mensaje de especificación de MLIP1. Si la imagen especificada en una operación de asignación de larga duración ya está asociada con el LPIN indicado, el decodificador no realizará ninguna acción. Un codificador no asignará la misma imagen a más de un valor de índice de larga duración. Deberá tratarse como un error el caso de que la imagen especificada en una operación de asignación de índice de larga duración esté ya asociada a un índice de larga duración diferente. Un codificador sólo cambiará una imagen de corta duración a imagen de larga duración dentro de la transmisión de 1024 imágenes consecutivas almacenadas. En otras palabras, una imagen de corta duración no permanecerá en el tampón de corta duración más que lo que dure la transmisión de 1023 imágenes subsiguientes almacenadas. Un codificador no asignará un índice de larga duración a una imagen de corta duración que haya sido marcada como "sin usar" por el proceso de decodificación antes del primer mensaje de asignación en el tren de bits. Un codificador no asignará un índice de larga duración a un número de imagen que no se haya enviado.

#### **U.4.2 Proceso del decodificador para la reorganización de la memoria tampón de imagen de referencia**

El decodificador emplea índices al hacer referencia a una imagen para la compensación en la capa de macrobloque utilizando los campos PR<sub>0</sub>, PR, PR<sub>2</sub>, PR<sub>3</sub>, PR<sub>4</sub>, PRB, PRFW, y BSBBW. En las imágenes que no sean imágenes B, estos índices son los índices relativos por defecto en la memoria tampón multiimagen en el caso de que los campos ADPN y LPIR no estén presentes en la capa de la imagen actual, de GOB, o de rebanada, según el caso, y son índices reorganizados cuando estos campos están presentes. En las imágenes B, la primera o las dos primeras de las imágenes (en función de BTSPM) en orden de índice relativo, se utilizan en la predicción hacia atrás, y los parámetros de referencia de imagen hacia adelante especifican un índice relativo dentro de las restantes imágenes a utilizar en la predicción hacia adelante.

Se pueden reorganizar los índices de las imágenes en la memoria tampón multiimagen a índices nuevamente especificados por medio de la transmisión de los campos RMPNI, ADPN, y LPIR. RMPNI indica si está presente ADPN o LPIR. Si está presente ADPN, RMPNI especifica el signo de la diferencia que debe añadirse al valor de predicción de un número de imagen. El valor ADPN corresponde a la diferencia absoluta entre el PN de la imagen a reorganizar, y a una predicción de este valor de PN. Se calcula el primer ADPN transmitido como la diferencia absoluta entre el PN de la imagen actual y el PN de la imagen a reorganizar. El siguiente campo ADPN transmitido representa la diferencia entre el PN de la imagen previa, que fue reorganizada utilizando ADPN, y el de la otra imagen a reorganizar. El proceso continúa hasta que se completa toda la reorganización necesaria. La existencia de reorganizaciones especificadas usando LPIR no afecta al valor de predicción para las subsiguientes reorganizaciones que utilizan ADPN. Si RMPNI indica la presencia de un campo LPIR, la imagen reorganizada corresponde a una imagen de larga duración con un índice LPIR de larga duración. Si no se reorganiza alguna imagen a un orden especificado por RMPNI, estas imágenes restantes deberán situarse después de cualquier imagen que tenga un orden reorganizado en el esquema de indexado, siguiendo el orden que tenían dentro de las imágenes que no hayan sido reorganizadas.

Si el decodificador detecta la falta de una imagen, puede invocar a algún proceso de ocultación, y puede insertar un error de imagen ocultada en la memoria tampón multiimagen. Se puede identificar a las imágenes que faltan si no están presentes uno o varios números de imagen, o si una imagen no almacenada en la memoria tampón multiimagen viene indicada en un ADPN o LPIR transmitido. La ocultación se puede llevar a cabo copiando a la posición de la imagen que falte la imagen precedente temporalmente más próxima de la que se disponga en la memoria tampón multiimagen. El orden temporal de las imágenes de corta duración en la memoria tampón multiimagen se puede inferir a partir de los campos del orden del índice por defecto y del PN. Además, o en su lugar, el decodificador puede enviar una señal de actualización INTRA forzada (por ejemplo, siguiendo Rec. UIT-T H.245), o el decodificador puede hacer uso de medios externos o de mensajes en el canal de retorno (por ejemplo, Rec. UIT-T H.245) para indicar al codificador la pérdida de imágenes. Se puede insertar una imagen ocultada dentro de la memoria tampón multiimagen cuando se utilice el almacenamiento del tipo "ventana deslizante". Si se detecta la falta de una imagen al decodificar una capa de GOB o de rebanada, se puede aplicar la ocultación de la imagen como si se hubiese detectado la imagen que falta en la capa de imagen.

#### **U.4.3 Proceso del decodificador en la eliminación de subimagen**

Se puede utilizar la eliminación de subimagen para reducir la memoria necesaria para el almacenamiento de múltiples imágenes de referencia. En la eliminación de subimagen, se parte cada imagen de referencia en subimágenes iguales del mismo tamaño. Se consigue la reducción de la memoria utilizada marcando como "sin usar" a las imágenes no necesarias. La estrategia utilizada por el codificador para decidir cuál de las subimágenes se marca como "sin usar" está fuera del objetivo de la presente Recomendación. El codificador, utilizando instrucciones MMCO en la capa de selección mejorada de imagen de referencia (ERPS), indica al decodificador el tamaño de las

subimágenes y cuáles de ellas hay que marcar como "sin usar". El codificador no enviará información en el tren de bits que haga que alguna de las muestras en las imágenes de referencia o subimágenes que él haya indicado que se marquen como "sin usar", se señalen para ser utilizadas en la predicción de subsecuentes imágenes.

La capacidad de eliminación de imagen se negocia por procedimientos externos (por ejemplo, utilizando Rec. UIT-T H.245). Además, el decodificador señala, también por medios externos, la unidad de partición mínima (MPU) que se describe en función de una anchura y altura mínimas de una subimagen (en unidades de 16 muestras de luminancia), y la cantidad total de memoria que está disponible en su memoria tampón multiimagen. La gestión de la memoria queda facilitada por las reglas de partición que se describen seguidamente.

Se parte cada imagen de referencia en subimágenes rectangulares del mismo tamaño. El codificador especifica el tamaño de subimagen, que deberá ser un múltiplo entero de la MPU. La anchura y la altura de la subimagen serán múltiplos enteros de la anchura y altura mínimas negociadas externamente como la MPU. La esquina superior izquierda de la primera subimagen coincide con la esquina superior izquierda de la imagen de referencia. Por tanto, se puede describir la partición completa especificando la anchura y la altura de una subimagen. Si el tamaño de la imagen no es un múltiplo entero del tamaño de la subimagen, algunas subimágenes pueden extenderse más allá de los límites derecho e inferior de la imagen de referencia. Cuando se almacena una subimagen que se extiende más allá de los límites de la imagen de referencia, una estrategia útil para la gestión de memoria es la de apartar suficiente memoria adicional para el almacenamiento de toda la subimagen, en lugar de la memoria justamente necesaria para el almacenamiento de la porción de la imagen de referencia que está dentro de esta subimagen. Este es un convenio que se deberá seguir en el cálculo de la capacidad de reserva de la memoria tampón, con el fin de determinar el llenado completo de la memoria tampón (por ejemplo, para determinar si se marcan automáticamente imágenes almacenadas en la memoria tampón como "sin usar" en el funcionamiento de "ventana deslizante"). Un decodificador diseñado de forma que cada subimagen ocupe la misma cantidad de memoria evitará la posibilidad de fragmentación de memoria.

Se describe brevemente a continuación un método ejemplo diseñado para el acceso a muestras referenciadas de imagen cuando está en uso la eliminación de subimagen. Un elemento importante en cualquier técnica de acceso a imagen de referencia es un mecanismo que identifique dónde están almacenadas en la memoria las muestras de cada subimagen. Si hay  $R$  imágenes de referencia y cada imagen está partida en  $S$  subimágenes, entonces hay un total de  $K = R \cdot S$  subimágenes. Por ejemplo, a la subimagen de la esquina superior izquierda del primer número de imagen de referencia, se la puede considerar como subimagen número 0, y a la subimagen a su derecha se la puede considerar como subimagen número 1, y así sucesivamente en un orden de barrido que progrese desde la imagen de referencia 1 a la  $R$ , hasta que se hayan etiquetado todas las  $K$  subimágenes. La capacidad total de la memoria tampón es  $SPTN$  memorias tampón de subimagen, y  $SPTN$  es típicamente menor que  $K$ . Se puede definir una matriz de  $K$  elementos  $subPicMem[K]$ , tal que  $t = subPicMem[k]$  corresponda al área de memoria de subimagen que contiene las muestras en la  $k$ -ésima subimagen. Por ejemplo, se puede considerar un caso en el que haya  $R = 5$  imágenes de referencia y que cada una tenga  $S = 12$  subimágenes. Entonces las muestras para la sexta subimagen en la imagen de referencia 3 se encontrará en el área de memoria de subimagen  $t = subPicMem[k]$ , siendo  $k = 3 \cdot S + 6 = 42$ .

Por ejemplo, cuando se hace referencia a muestras para la predicción compensada de movimiento de un bloque de datos de luminancia o de crominancia y no se utilizan los modos opcionales de predicción avanzada y de actualización de resolución reducida, es necesaria la adquisición de  $n \times m$  muestras, pudiendo tomar  $n$  y  $m$  valores de 8 ó 9 para acomodar la compensación de movimiento de medio entero. Debido a que las muestras en un bloque pueden estar situadas hasta en cuatro subimágenes diferentes, se deben considerar cuatro casos distintos. En todos ellos, el primer paso es encontrar el sitio en la memoria que contiene a la muestra superior izquierda (U) del bloque a



referenciar. Se puede identificar la subimagen que contiene a  $U$  dividiendo la situación horizontal o vertical de  $U$  por la anchura o la altura de la subimagen. Si  $U$  está situada en la subimagen  $k$ , entonces la muestra estará situada en el área de memoria de la subimagen  $\text{subPicMem}[k]$ . A continuación, si tanto la muestra  $m - 1$  está muestreada a la derecha de  $U$  (es decir, la esquina superior derecha del bloque) y la muestra  $n - 1$  está muestreada por debajo de  $U$  (es decir, la esquina inferior izquierda del bloque) pertenecen a la subimagen  $k$ , se puede considerar a este caso como el caso uno. Si la muestra  $n - 1$  está muestreada por debajo de  $U$ , pertenece a  $k$ , pero la muestra  $m - 1$  está muestreada a la derecha de  $U$ , no pertenece a  $k$ , lo cual puede considerarse como caso dos. Si la muestra  $m - 1$  está muestreada a la derecha de  $U$ , pertenece a  $k$ , pero la muestra  $n - 1$  está muestreada por debajo, el caso puede considerarse como caso tres. En otro caso la muestra  $m - 1$  está muestreada a la derecha de  $U$  y la  $n - 1$  está muestreada por debajo, están fuera de la subimagen  $k$ , y puede considerarse este caso como el cuatro.

En el caso número uno, todas las muestras del bloque de referencia están contenidas dentro de la  $k$ -ésima subimagen. En este caso, todas las  $n \times m$  muestras relevantes se pueden encontrar en el área de memoria de subimagen  $\text{subPicMem}[k]$  y es sencillo acceder a ellas. En el caso dos, las muestras que están situadas en la subimagen de orden  $k$ , se pueden obtener del área de memoria de la subimagen  $\text{subPicMem}[k]$ , y las restantes muestras se pueden obtener de  $\text{subPicMem}[k_r]$ , donde  $k_r$  es la subimagen a la derecha de  $k$ . En el caso tres, las muestras que se encuentran en la subimagen de orden  $k$  se pueden obtener del área de memoria  $\text{subPicMem}[k]$ , y las restantes muestras se pueden obtener de  $\text{subPicMem}[k_d]$ , donde  $k_d$  es la subimagen debajo de  $k$ . En el caso cuatro, las muestras que se encuentran en la subimagen de orden  $k$  se pueden obtener del área de memoria de subimagen  $\text{subPicMem}[k]$ , y las restantes muestras se pueden obtener de las áreas de memoria  $\text{subPicMem}[k_r]$ ,  $\text{subPicMem}[k_d]$  y  $\text{subPicMem}[k_{rd}]$  donde  $k_r$  y  $k_d$  están definidas anteriormente y  $k_{rd}$  es la subimagen a la derecha y por debajo de  $k$ .

#### **U.4.4 Proceso decodificador para compensación de movimiento multiimagen**

La compensación de movimiento multiimagen se aplica si el campo MRPA indica la utilización de más de una imagen de referencia. Para la compensación de movimiento multiimagen, el decodificador elige una imagen de referencia como se indica haciendo uso de los campos  $\text{PR}_0$ ,  $\text{PR}$ ,  $\text{PR}_2$ ,  $\text{PR}_3$ ,  $\text{PR}_4$ ,  $\text{PRB}$ ,  $\text{PRFW}$  y  $\text{BSBBW}$  de la capa de macrobloque. Una vez que se ha especificado la imagen, el proceso de decodificación prosigue como se describe en 6.1.

En el caso de que se utilicen cuatro vectores de movimiento por macrobloque y el campo MRPA indique el uso de más de una imagen de referencia, el índice de referencia de imagen para ambos bloques de crominancia es el asociado con el primero de los cuatro vectores de movimiento (siendo el proceso de compensación de movimiento como se especifica en la cláusula 6.1).

#### **U.4.5 Proceso decodificador para el almacenamiento en la memoria tampón de imagen de referencia**

El almacenamiento en la memoria tampón de la imagen actualmente decodificada se puede especificar haciendo uso del tipo almacenamiento en la memoria tampón de imagen de referencia (RPBT) para imágenes no B. El almacenamiento en la memoria tampón puede seguir un modo primero en entrar, primero en salir ("ventana deslizante"). Alternativamente, el almacenamiento en la memoria tampón puede seguir un método de almacenamiento en la memoria tampón adaptativa personalizada ("control adaptativo de memoria") que sea especificado por el codificador en el canal de transmisión hacia adelante. Las imágenes B no afectan los contenidos de la memoria tampón.

El tipo de almacenamiento en la memoria tampón de "ventana deslizante" funciona de la siguiente manera. Primero, el decodificador determina si la imagen puede ser almacenada en la capacidad de la memoria tampón "sin usar". Si la capacidad de la memoria tampón "sin usar" no es suficiente, se marcará como "sin usar" la imagen de corta duración con mayor índice por defecto (es decir, la imagen de corta duración más antigua en la memoria tampón). Este proceso se repite en caso

necesario (en el caso de eliminación de subimagen) hasta que se libere la suficiente capacidad de memoria para albergar la imagen actual decodificada. Se almacena la imagen actual en la memoria tampón y se le asigna un índice de la memoria tampón relativo por defecto igual a cero. El índice relativo por defecto de todas las demás imágenes de corta duración se va incrementando en uno. El índice relativo por defecto de todas las imágenes de larga duración se incrementa en uno menos el número de imágenes de corta duración eliminadas.

En el tipo de almacenamiento en la memoria tampón de "control adaptativo de memoria", se pueden eliminar explícitamente de la memoria tampón multiimagen áreas específicas de imágenes o de subimágenes. La imagen decodificada actual, que inicialmente se considera como una imagen de corta duración, puede ser insertada en la memoria tampón por el codificador con un índice relativo por defecto igual a 0, puede asignarla a un índice de larga duración, o puede marcarla como "sin usar". También se pueden asignar índices de larga duración a otras imágenes de corta duración. El proceso de almacenamiento en la memoria tampón funcionará de una manera funcionalmente equivalente a la siguiente: Primero, la imagen actual se añade a la memoria tampón multiimagen con un índice relativo por defecto 0, y los índices relativos por defecto de todas las demás imágenes se incrementan en uno. Después se procesan las instrucciones MMCO:

- Si MMCO indica una reiniciación del contenido de la memoria tampón utilizando RESET igual a "1", se marcan todas las imágenes de la memoria tampón como "sin usar", excepto la imagen actual (que será la imagen con índice relativo por defecto 0, ya que una reiniciación de la memoria tampón debe ser la primera instrucción MMCO como se señala en U.3.1.5.7).
- Si MMCO indica un índice máximo de larga duración utilizando MLIP1, se marcan como "sin usar" todas las imágenes de larga duración que tengan índices de larga duración mayores o iguales a MLIP1, no modificándose el orden del índice relativo de las demás imágenes.
- Si MMCO indica que una imagen debe ser marcada como "sin usar" en la memoria tampón multiimagen, y si esa imagen no ha sido marcada todavía como "sin usar", se marca la imagen especificada como "sin usar" en la memoria tampón multiimagen y se decrementa en uno el índice relativo por defecto de las imágenes subsiguientes en el orden por defecto.
- Si MMCO indica que áreas de subimagen de alguna imagen deben ser marcadas como "sin usar" en la memoria tampón multiimagen, se marcan como "sin usar" en la memoria tampón multiimagen las áreas de subimagen especificadas y no se modifica el orden del índice relativo por defecto de las imágenes. Como se señala en U.3.1.5.10, no todas las áreas de subimagen de una imagen dada se marcarán como "sin usar" por una instrucción MMCO de eliminación de subimagen (en su lugar el codificador deberá enviar una instrucción MMCO que marque como "sin usar" la imagen como un todo).
- Si MMCO indica la asignación de un índice de larga duración a una determinada imagen de corta duración y, si el índice de larga duración especificado todavía no ha sido asignado a la imagen de corta duración especificada, se marca en la memoria tampón la imagen de corta duración especificada como imagen de larga duración con el índice especificado de larga duración. Si en la memoria tampón está ya presente otra imagen con el mismo índice de larga duración al índice especificado de larga duración, se marca a dicha imagen como "sin usar". Todas las imágenes de corta duración que fuesen subsiguientes a la imagen de corta duración especificada en el orden de índice relativo por defecto, así como todas las imágenes de larga duración que tengan un índice de larga duración menor que el índice de larga duración especificado, verán sus índices relativos por defecto asociados decrementados en uno. Se asigna la imagen especificada a un índice relativo por defecto de uno más el mayor de los índices relativos por defecto decrementados, o a cero si no existen tales índices decrementados.

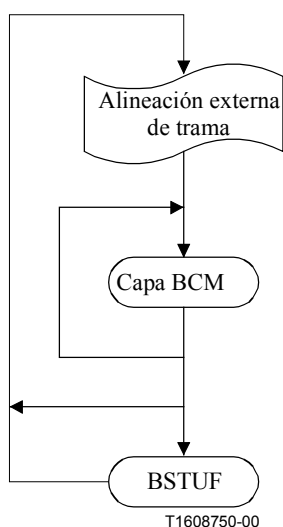
La cantidad resultante de imágenes almacenadas en la memoria tampón, o de regiones de subimágenes no marcadas como "sin usar" no deberá exceder la capacidad de la memoria tampón indicada por el valor más reciente de SPTN. Si el codificador detecta esta condición, deberá tratarla como un error.

## U.5 Mensajes en el canal de retorno

Se puede utilizar un canal fuera de banda, que no es necesario que sea fiable, para el transporte de mensajes de canal de retorno. La sintaxis de este canal fuera de banda (que podría ser un canal lógico separado, por ejemplo, según Rec. UIT-T H.223 o Rec. UIT-T H.225.0) será la aquí definida. El funcionamiento "múltiplex de vídeo" de los mensajes del canal de retorno definidos en el anexo N no está soportado en el modo ERPS.

### U.5.1 Capa de canal lógico separado de BCM

La capa de BCM, como se especifica en U.5.2, deberá ser transportada por una capa de canal lógico separado de BCM, como se muestra en la figura U.8.



**Figura U.8/H.263 – Estructura de la capa de canal lógico separado de BCM para el modo ERPS**

#### U.5.1.1 Alineación externa de trama

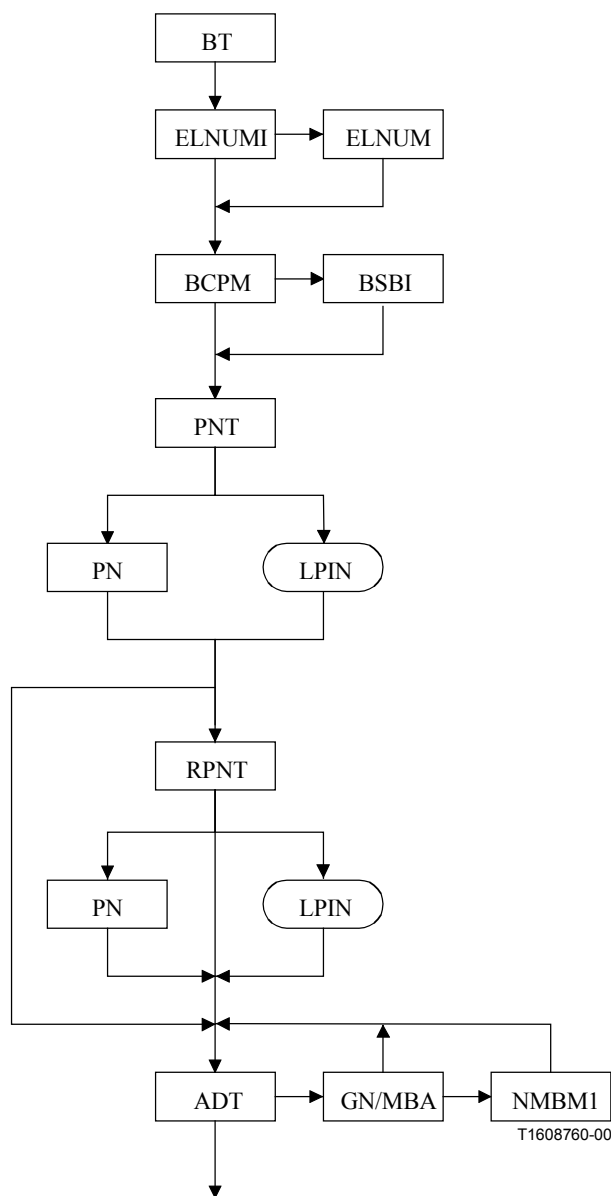
Se deberá proporcionar la alineación externa de trama de los mensajes del canal de retorno como se muestra en la figura U.8. Se utiliza la alineación externa de trama para determinar el momento de comienzo de los mensajes del canal de retorno, así como el tamaño de los datos que siguen en el mensaje del canal de retorno.

#### U.5.1.2 Relleno del canal de retorno (BSTUF, *back-channel stipping*) (longitud variable)

BSTUF es una palabra de código de longitud variable que puede estar presente sólo después del último mensaje del canal de retorno en una trama externa. BSTUF consiste en una palabra de código de longitud variable que consta de uno o más bits de valor "0".

## U.5.2 Sintaxis de la capa de mensaje del canal de retorno

La sintaxis para la capa de mensaje del canal de retorno (BCM, *back-channel message*) aquí definida será como se presenta en la figura U.9.



**Figura U.9/H.263 – Estructura de la capa de mensaje de canal de retorno (BCM) para el modo ERPS**

#### U.5.2.1 Tipo de mensaje del canal de retorno (BT, *back-channel message type*) (2 bits)

BT es una palabra de código de longitud fija de dos bits que indica el tipo de mensaje del canal de retorno. BT es la primera palabra de código presente en cada mensaje del canal de retorno. El tipo o tipos de mensajes que han sido solicitados por el codificador viene indicado en el campo RPSMF de la sintaxis del canal hacia adelante. Los valores de BT se definen como:

- "00": Reservado para uso futuro.
- "01": Reservado para uso futuro.
- "10": NACK. Indica la pérdida, o la decodificación errónea, de la parte correspondiente de los datos del canal hacia adelante.
- "11": ACK. Indica la correcta decodificación de la parte correspondiente de los datos del canal hacia adelante.

### **U.5.2.2 Indicación de número de capa de mejora (ELNUMI, *enhancement layer number indication*) (1 bit)**

ELNUMI es una palabra de código de un único bit de longitud fija, que sigue a BT en el mensaje del canal de retorno. ELNUMI será "0" a no ser que se utilice en el canal hacia adelante el modo escalabilidad temporal, SNR y espacial (véase el anexo O), y que estén combinadas algunas capas de mejora del canal hacia adelante en un canal lógico, y que el mensaje del canal de retorno se refiera a una capa de mejora (en lugar de la capa de base), en cuyo caso ELNUMI será "1".

### **U.5.2.3 Número de capa de mejora (ELNUM, *enhancement layer number*) (4 bits)**

ELNUM es una palabra de código de 4 bits de longitud fija, que está presente sólo si ELNUMI es "1". Si está presente sigue a ELNUMI. Cuando está presente, ELNUM contiene el número de la capa de mejora al que se hace referencia en el mensaje del canal de retorno.

### **U.5.2.4 Indicador CPM de canal de retorno (BCPM, *back-channel CPM indicator*) (1 bit)**

BCPM es una palabra de código de un único bit de longitud fija, que sigue a ELNUMI o a ELNUM en el mensaje del canal de retorno. BCPM será "0" a no ser que se utilice el modo CPM (véanse 5.2.4 y el anexo C) en los datos del canal hacia adelante, en cuyo caso BCPM será "1". Si BCPM es "1", indica que BSBI está presente.

### **U.5.2.5 Indicador de subtren de bits de canal de retorno (BSBI, *back-channel sub-bitstream indicator*) (2 bits)**

BSBI es una palabra de código de 2 bits de longitud fija, que sigue a BCPM cuando está presente. BSBI está presente sólo si BCPM es "1". BSBI es la representación binaria natural del número del subtren de bits en el canal hacia adelante a los que el mensaje del canal de retorno hace referencia (véanse 5.2.4 y el anexo C).

### **U.5.2.6 Tipo de número de imagen (PNT, *picture number type*) (1 bit)**

PNT es una palabra de código de un solo bit de longitud fija, que siempre está presente y sigue a BCPM o a BSBI en el mensaje del canal de retorno. Los valores de PNT serán:

- "0": El mensaje se refiere a una imagen especificada por un número de imagen de corta duración (PN).
- "1": El mensaje se refiere a una imagen especificada por un índice de imagen de larga duración (LPIN).

PNT es seguido por PN o por LPIN, dependiendo del valor de PNT. PN y LPIN se representarán como se especifica en U.3.1.3 y U.3.1.5.9, respectivamente, para su utilización en los datos del canal hacia adelante.

### **U.5.2.7 Tipo de número de imagen solicitada (RPNT, *requested picture number type*) (2 bits)**

RPNT es una palabra de código de 2 bits de longitud fija, que está presente sólo si BT indica un mensaje NACK. Sigue a PN o a LPIN cuando está presente. Determina la forma de identificar a una imagen en la memoria tampón multiimagen que se puede utilizar como una referencia para la codificación de las subsiguientes imágenes. Los valores de RPNT serán:

- "00": Imágenes no válidas en la memoria tampón. Se deberá reiniciar la memoria tampón con una imagen I o EI con RESET igual a "1".
- "01": No se identifica ninguna imagen en particular para ser utilizada como una referencia.
- "10": Se identifica, por medio de un número de imagen de corta duración (PN), una imagen que se puede utilizar como una referencia.
- "11": Se identifica, por medio de un índice de imagen de larga duración (LPIN), una imagen que se puede utilizar como una referencia.

Si RPNT es "10" o "11", RPNT es seguido por PN o LPIN, dependiendo del valor de RPNT. PN y LPIN estarán representados de la forma especificada en U.3.1.3 y U.3.1.5.9, respectivamente, para su uso en los datos del canal hacia adelante. Típicamente el PN, o el LPIN, especificado utilizando RPNT, identifica el último área de imagen correspondiente espacialmente, correctamente decodificado para la imagen, o región, identificada en el mensaje del canal de retorno.

#### **U.5.2.8 Tipo de datos adicionales (ADT, *additional data type*) (2 bits)**

ADT es una palabra de código de 2 bits de longitud fija, que está presente después de PN, LPIN, o RPNT, según determine PNT (en un mensaje ACK) o RPNT (en un mensaje NACK). Puede presentarse varias veces si está presente. Especifica el tipo de datos adicionales utilizados en la identificación de una región de la imagen en cuestión, a la que se refiere el mensaje del canal de retorno. Se definen los valores de ADT como:

"00": Fin de datos adicionales.

"01": Se identifica una región por medio de sólo un campo GN/MBA.

"10": Se identifica una región como a un área de barrido dentro de una imagen por medio de GN/MBA y NMBM1.

"11": Se identifica una región como a un área de barrido dentro de una rebanada rectangular por medio de GN/MBA y NMBM1.

Si ADT es "00", no siguen más datos en el mensaje del canal de retorno. Si ADT es "01", ADT es seguido por GN/MBA y después por otro ADT. Si ADT es "10" o "11", ADT es seguido por GN/MBA y NMBM1, y después por otro ADT.

Si ADT es "10", se identifica a la región como una región que comienza en una localización espacial en concreto, especificada por GN/MBA y que contiene un número especificado de macrobloques en orden de barrido dentro de la imagen. Si ADT es "11", se identifica a la región como una región que comienza en una localización espacial especificada por GN/MBA y que contiene un número especificado de macrobloques en orden de barrido dentro de una rebanada rectangular. Si ADT está presente sólo una vez y es "00", la región identificada es la imagen como un todo. Si ADT está presente más de una vez, el valor "00" se utiliza únicamente para terminar el bucle y no para identificar a una región.

#### **U.5.2.9 Número de GOB/dirección de macrobloque (GN/MBA, *GOB number/macroblock address*) (5/6/7/9/11/12/13/14 bits)**

GN/MBA es una palabra de código de longitud fija que especifica un número de GOB o una dirección de macrobloque. GN/MBA sigue a ADT cuando está presente. GN/MBA está presente cuando lo indica ADT. Si no se utiliza el modo opcional estructura en rebanada (véase el anexo K), GN/MBA contiene el número de GOB del comienzo de un área al que hace referencia el mensaje del canal de retorno. Si se utiliza el modo opcional estructura en rebanada, GN/MBA contiene la dirección del macrobloque del comienzo del área al que hace referencia el mensaje del canal de retorno. La longitud de este campo será la especificada en otra parte de esta Recomendación para GN o MBA.

#### **U.5.2.10 Número de macrobloques menos 1 (NMBM1, *number of macroblocks minus 1*) (5/6/7/9/11/12/13/14 bits)**

NMBM1 es una palabra de código de longitud fija que especifica un número de macrobloques. NMBM1 está presente cuando lo indique ADT. Cuando está presente sigue a GN/MBA. Contiene la representación natural del número especificado de macrobloques menos 1. La longitud de este campo será la longitud definida para la dirección de un macrobloque en K.2.5 y el cuadro K.2.

## Anexo V

### Modo rebanada particionada de datos

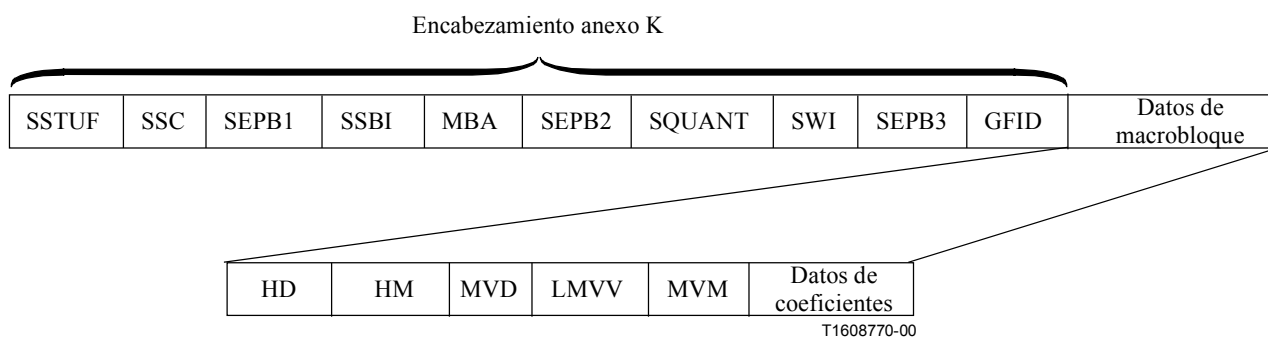
#### V.1 Introducción

Este anexo describe el modo opcional rebanada particionada de datos (DPS, *data-partitioned slice*) de la Rec. UIT-T H.263. La utilización de este modo se indica por medios externos (por ejemplo, mediante Rec. UIT-T H.245). La utilización de este modo se indicará poniendo a "1" el bit 17, anteriormente reservado, de la parte opcional de PLUSPTYPE (OPPTYPE). Este modo utiliza la estructura de encabezamiento definida en el anexo K.

La partición de datos proporciona solidez en ambientes propicios a errores. Esto se consigue utilizando una redistribución de la sintaxis de la presente Recomendación con el fin de permitir una detección precoz y una recuperación frente a los errores que se hayan introducido durante la transmisión.

#### V.2 Estructura de la partición de datos

Cuando se utiliza la partición de datos, se disponen los datos como los de un segmento de datos de vídeo, de la forma definida en R.2. Se disponen los macrobloques en el segmento, de forma que la cabecera de información para todos los macrobloques dentro del segmento se transmite junta, seguida por los vectores de movimiento para todos los macrobloques del segmento, y después por los coeficientes de la transformada discreta del coseno (*DCT, discrete cosine transform*) para todos los macrobloques del segmento. El encabezamiento del segmento utiliza la misma sintaxis que la descrita en K.2. Se separan con marcadores el encabezamiento, los vectores de movimiento y las particiones DCT, de forma que sea posible la resincronización al final de la partición en la que se ha producido un error. Cada segmento contendrá los datos de un número entero de macrobloques. Cuando se utiliza este modo, se usará la sintaxis presentada en la figura V.1.



**Figura V.1/H.263 – Sintaxis de la partición de datos**

Téngase en cuenta que cuando no se hace uso de este anexo, los datos del vector de movimiento y de la DCT se transmiten de forma entrelazada para todos los macrobloques de un segmento de imagen de vídeo, en cuyo caso normalmente se produce la pérdida de toda la información de los restantes macrobloques del paquete.

##### V.2.1 Datos de encabezamiento (HD, *header data*) (longitud variable)

El campo de los datos del encabezamiento contiene la información de COD y MCBPC de todos los macrobloques del paquete, además de los datos MODB en el caso de tramas PB, o de tramas PB mejoradas. Se utiliza un código de longitud variable reversible (*RVLC, reversible variable length code*) para combinar los COD y MCBPC para todos los macrobloques en el paquete. Se muestra este código en los cuadros V.1 a V.5. Si se utiliza el anexo O, la COD sólo se combina con el TYPE

del macrobloque, para formar el RVLC para las imágenes B y EP que hacen uso de los cuadros V.3 y V.4, y que la CBPC se codifica con palabras de código del cuadro O.4. Si  $COD = 0$ , y se utilizan los anexos G o M, la palabra de código para  $COD+MCBPC$  estará seguida inmediatamente por los correspondientes datos codificados reversibles de longitud variable que correspondan al campo MODB del macrobloque. Se utilizará el cuadro V.6 para las tramas PB y el cuadro V.7 para las tramas PB mejoradas.

## V.2.2 Marcador de encabezamiento (HM, *header marker*) (9 bits)

Es una palabra de código de 9 bits. Su valor es 1010 0010 1. El HM finaliza la parte del encabezamiento. Cuando el decodificador utiliza la decodificación invertida, busca este marcador. Naturalmente, este marcador no se puede presentar en el campo HD.

## V.2.3 Capa de datos del vector de movimiento (longitud variable)

### V.2.3.1 Codificación de la diferencia de vectores de movimiento

Para los vectores de movimiento se utilizan las palabras de código RVLC presentadas en el cuadro D.3 para la codificación de la diferencia entre el vector de movimiento y la predicción del vector de movimiento. Téngase en cuenta que este anexo sólo utiliza la codificación en entropía del anexo D, pero no el resto de sus consideraciones, a no ser que esté en uso dicho anexo D.

### V.2.3.2 Predicción de los valores del vector de movimiento

El primer vector de movimiento en el paquete se codifica usando un valor de predictor igual a 0, para ambos componentes, el horizontal y el vertical, y los vectores de movimiento para los subsiguientes macrobloques se codifican de forma predictiva utilizando la diferencia de vectores de movimiento ( $MVD$ , *MV difference*). Esta situación difiere del método utilizado en la codificación de los vectores de movimiento, en la que los vectores de movimiento que siguen a un macrobloque saltado o INTRA se codifican utilizando un valor de predictor igual a 0 para ambos componentes, el horizontal y el vertical.

Dirección hacia adelante:  $MV_i = MV_{i-1} + MVD_i = MV_{i-1} + (MV_i - MV_{i-1})$

Dirección hacia atrás:  $MV_{i-1} = MV_i - MVD_i = MV_i - (MV_i - MV_{i-1})$ .

( $MV_i$  y  $MVD_i$  son los vectores de movimiento de orden  $i$  y la diferencia de los vectores de movimiento, respectivamente, dentro del paquete.)

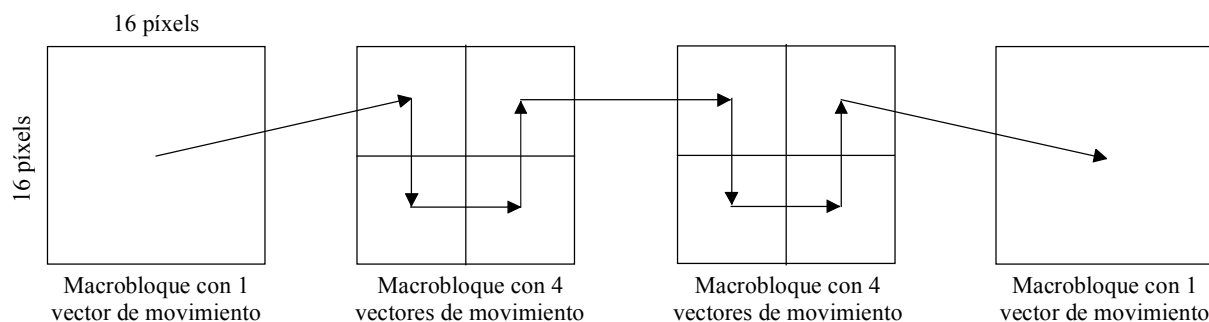
La información del vector de movimiento para el último vector de movimiento en el paquete se codifica de esta forma y se vuelve a codificar otra vez en el campo LMVV, como se describe más adelante en V.2.4. Esto permite al decodificador decodificar independientemente la secuencia de los vectores de movimiento utilizando dos caminos diferentes de predicción:

- 1) en el sentido hacia adelante, comenzando desde el principio de los datos de movimiento del paquete, y
- 2) en el sentido hacia atrás, desde el final de los datos de movimiento en un paquete.

Esto proporciona robustez para una mejor detección y ocultamiento de errores.

NOTA 1 – Cuando no se utiliza el modo DPS, se codifican predictivamente los vectores de movimiento, siendo la predicción del actual vector de movimiento la media de 3 vectores de movimiento en localizaciones próximas como se describe en 6.1.1. Al estar formados los paquetes según este anexo, de forma tal que el número de macrobloques codificados en cada paquete es variable, utilizando el método de codificación predictiva media (que involucra a vectores de movimiento de diferentes filas de la trama) impediría la decodificación reversible de los vectores de movimiento en una rebanada. Cuando se utiliza el modo DPS, se forma un único camino para los vectores de movimiento dentro del paquete completo. Esta situación se muestra en la figura V.2.





T1608780-00

**Figura V.2/H.263 – Predicción del vector de movimiento por camino único**

En el caso de imágenes B o de imágenes EP (véase el anexo O), MVDFW y MVDBW pueden estar presentes, como se indica en los cuadros V.3 y V.4 para la palabra de código MBTYPE. MVDFW se codifica predictivamente utilizando el mismo camino de predicción único descrito anteriormente, y MVDBW (cuando está presente en imágenes B) se codificará como se especifica en O.4.6. MVDFW y MVDBW se codificarán con las palabras de código del cuadro D.3.

En el caso de tramas PB (anexo G), y de tramas PB mejoradas (anexo M), los datos de MVDB se codificarán como se especifica en los correspondientes anexos utilizando las palabras de código del cuadro D.3.

NOTA 2 – Si el modo decodificación hacia atrás se utiliza en una trama B (véase el anexo O) o en tramas PB mejoradas (anexo M), el decodificador debería descartar MVDB y MVDBW ya que los datos del vector de movimiento para la predicción hacia atrás puede que no se recuperen adecuadamente a través de los límites de los paquetes.

### V.2.3.3 Prevención de emulación de código de comienzo en la codificación de diferencia de vectores de movimiento

Se cambia el método de prevención de emulación de comienzo de código de MVD, del método descrito en D.2, con el fin de facilitar el análisis independiente hacia atrás. La partición del MV será sometida a un barrido de izquierda a derecha y se insertará un MVD = 0 (palabra de código "1") después de dos MVD cualesquiera que sean ambos iguales a 1 (palabra de código "000"). Si una tercera palabra de código MVD = 1 sigue a esas dos palabras de código MVD = 1 en el tren de bits original (antes de la inserción), se considerará que se ha detectado la primera palabra de código MVD = 1 en las palabras de código restantes de la partición del MV. No se tendrá en cuenta una segunda palabra de código MVD = 1 ni se insertará una palabra de código MVD = 0 tras ella. Este método difiere del descrito en el anexo D, en el que únicamente se inserta el bit cuando dos MVD = 1 consecutivos (palabra de código "000") forman un par (es decir, cuando el primer MVD es el componente horizontal, y el segundo es el componente vertical). Si están en uso tanto el anexo D como el anexo V, se utilizará el método de prevención de emulación de código de comienzo del anexo V, en lugar del método descrito en D.2.

### V.2.4 Valor del vector de movimiento último (*LMVV*, *last motion vector value*) (longitud variable)

El campo LMVV contiene el último vector de movimiento del paquete. Se codifica utilizando un valor de predictor igual a 0 para ambos componentes, el horizontal y el vertical. En el caso de que no haya vectores de movimiento, o sólo uno, en el paquete, no estará presente LMVV. (Esta utilización de un predictor fijo de valor cero permite el uso de la decodificación reversible.)

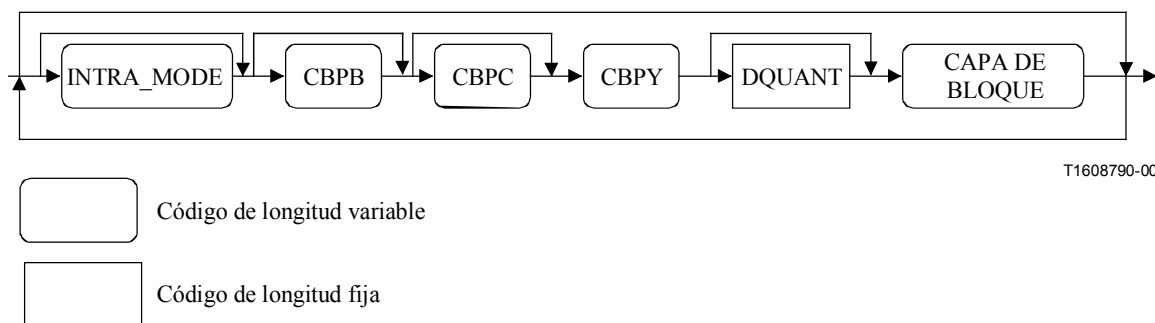
### V.2.5 Marcador del vector de movimiento (*MVM*, *motion vector marker*) (10 bits)

Es una palabra de código que tiene el valor "0000 0000 01". MVM termina la partición del vector de movimiento. Cuando el decodificador utiliza la decodificación inversa, el decodificador busca

este marcador. El marcador del vector de movimiento no estará incluido en el paquete en caso de que el paquete no contenga datos del vector de movimiento (si todos los macrobloques en el paquete están codificados como intra o con COD iguales a 1).

### V.2.6 Capa de datos de coeficiente (longitud variable)

La capa de datos DCT contiene coeficientes INTRA\_MODE (si está presente), CBPB (si está presente), CBPC (si está presente), CBPY, DQUANT (si está presente), y DCT, codificados como se especifica en I.2, 5.3.4, O.4.3, 5.3.5, 5.3.6, y 5.4.2, respectivamente. En la figura V.3 se ilustra el diagrama de sintaxis de los datos DCT. La presencia de CBPC está indicada en los cuadros V.3 y V.4.



**Figura V.3/H.263 – Sintaxis de los datos de coeficiente**

### V.3 Interacción con otros modos opcionales

El modo DPS actúa eficazmente como un submodo del modo estructurada en rebanada del anexo K, y utiliza sus estructuras de imagen externa y de encabezamiento de rebanada. Por tanto, se deberá indicar el modo SS como en uso siempre que lo esté el modo DPS. Los otros dos submodos del modo estructurada en rebanada (los submodos ordenamiento de rebanadas arbitrario y rebanada rectangular) pueden estar en uso conjuntamente con el modo DPS.

No se utilizará el modo codificación aritmética basada en sintaxis del anexo E si se utiliza el presente anexo, ya que no permite la decodificación reversible.

No se utilizará la corrección de errores hacia adelante del anexo H si se utiliza el presente anexo, ya que puede causar la interrupción del tren de bits en sitios no deseados. Sin embargo, no está prohibida la utilización de lo indicado en el anexo H con el modo DPS, debido a que la corrección de errores hacia adelante es necesaria en algunos sistemas diseñados con las normas existentes.

Se puede utilizar el modo escalabilidad temporal, SNR y espacial (TSSS, *temporal*, *SNR*, and *spatial scalability*) del anexo O conjuntamente con el modo DPS. Cuando se utilizan conjuntamente los modos TSSS y DPS, se utilizarán las palabras de código de los cuadros V.3, V.4 y V.5, en lugar de las definidas en el anexo O.

No se utilizará el anexo U con el presente anexo.

**Cuadro V.1/H.263 – Cuadro para RVLC de COD y MCBPC  
para macrobloques INTRA**

<b>Tipo de macrobloque</b>	<b>CBPC (56)</b>	<b>Palabra de código (para COD+MCBPC combinados)</b>	<b>Número de bits</b>
3 (INTRA)	00	1	1
3	01	010	3
3	10	0110	4
3	11	01110	5
4 (INTRA + Q)	00	00100	5
4	01	011110	6
4	10	001100	6
4	11	0111110	7
relleno		0011100	7

**Cuadro V.2/H.263 – Cuadro para RVLC de COD y MCBPC  
para macrobloques INTER**

<b>Tipo de macrobloque</b>	<b>CBPC (56)</b>	<b>Palabra de código (para COD+MCBPC combinados)</b>	<b>Número de bits</b>
con salto		1	1
0 (INTER)	00	010	3
0	10	00100	5
0	01	011110	6
0	11	0011100	7
1 (INTER + Q)	00	01110	5
1	10	00011000	8
1	01	011111110	9
1	11	01111111110	11
2 (INTER4V)	00	0110	4
2	10	01111110	8
2	01	00111100	8
2	11	000010000	9
3 (INTRA)	00	001100	6
3	11	0001000	7
3	10	001111100	9
3	01	000111000	9
4 (INTRA + Q)	00	0111110	7
4	11	0011111100	10
4	10	0001111000	10
4	01	0000110000	10

**Cuadro V.2/H.263 – Cuadro para RVLC de COD y MCBPC para macrobloques INTER**

Tipo de macrobloque	CBPC (56)	Palabra de código (para COD+MCBPC combinados)	Número de bits
5 (INTER4V + Q)	00	00111111100	11
5	01	00011111000	11
5	10	00001110000	11
5	11	00000100000	11
relleno		0111111110	10

**Cuadro V.3/H.263 – RVLC del MBTYPE para macrobloques B**

Índice	Tipo de predicción	MVDFW	MVDBW	CBPC + CBPY	DQUANT	MBTYPE	Bits
–	Directa con salto					1 (COD=1)	1
0	Directa			X		010	3
1	Directa Q			X	X	001100	6
2	Hacia adelante (sin textura)	X				00100	5
3	Hacia adelante	X		X		011110	6
4	Hacia adelante +Q	X		X	X	01111110	8
5	Hacia atrás (sin textura)		X			0110	4
6	Hacia atrás		X	X		01110	5
7	Hacia atrás + Q		X	X	X	00111100	8
8	Bi-Dir (sin textura)	X	X			0011100	7
9	Bi-Dir	X	X	X		0001000	7
10	Bi-Dir + Q	X	X	X	X	0111110	7
11	INTRA			X		00011000	8
12	INTRA + Q			X	X	01111110	9
13	Relleno					001111100	9

**Cuadro V.4/H.263 – Cuadro de RVLC del MBTYPE para macrobloques EP**

Índice	Tipo de predicción	MVDFW	MVDBW	CBPC + CBPY	DQUANT	MBTYPE	Bits
–	Hacia adelante (con salto)					1 (COD=1)	1
0	Hacia adelante	X		X		010	3
1	Hacia adelante + Q	X		X	X	0110	4
2	Hacia arriba (sin textura)					01110	5
3	Hacia arriba			X		00100	5

**Cuadro V.4/H.263 – Cuadro de RVLC del MBTYPE para macrobloques EP**

Índice	Tipo de predicción	MVDFW	MVDBW	CBPC + CBPY	DQUANT	MBTYPE	Bits
4	Hacia arriba + Q			X	X	011110	6
5	Bi-Dir (sin textura)					001100	6
6	Bi-Dir	X		X		0111110	7
7	Bi-Dir + Q	X		X	X	0011100	7
8	INTRA			X		0001000	7
9	INTRA + Q			X	X	01111110	8
10	Relleno					00111100	8

**Cuadro V.5/H.263 – Cuadro de RVLC de COD + MCBPC para macrobloques EI**

Tipo de predicción	QCBP (56)	Palabra de código (para COD+MCBPC combinados)	Número de bits
Hacia arriba (con salto)		1	1
0 (Hacia arriba)	00	010	3
0	01	0110	4
0	10	01110	5
0	11	00100	5
1 (Hacia arriba + Q)	00	011110	6
1	01	001100	6
1	10	0111110	7
1	11	0011100	7
2 (INTRA)	00	0001000	7
2	01	01111110	8
2	10	00111100	8
2	11	00011000	8
3 (INTRA + Q)	00	011111110	9
3	01	001111100	9
3	10	000111000	9
3	11	000010000	9
Relleno		0111111110	10

**Cuadro V.6/H.263 – Cuadro de RVLC para MODB**

Índice	CBPB	MVDB	Número de bits	Código
0			3	010
1		X	4	0110
2	X	X	5	01110

NOTA – "X" indica que el ítem está presente en el macrobloque.

**Cuadro V.7/H.263 – Cuadro de RVLC para MODB  
en el modo tramas PB mejoradas**

Índice	CBPB	MVDB	Número de bits	Código	Modo de codificación
0			3	010	Bidirectional prediction
1	X		4	0110	Bidirectional prediction
2		X	5	01110	Forward prediction
3	X	X	5	00100	Forward prediction
4			6	011110	Backward prediction
5	X		6	001100	Backward prediction

NOTA – El símbolo "X" en el cuadro anterior indica que está presente el elemento de sintaxis asociado.

## Anexo W

### Especificación de información adicional de mejora suplementaria

#### W.1 Introducción

Este anexo describe el formato de la información adicional de mejora suplementaria enviada en el campo PSUPP de la capa de imagen especificada en la Rec. UIT-T H.263, que amplía la funcionalidad definida en el anexo L. Se puede señalar por medios externos (por ejemplo, utilizando Rec. UIT-T H.245) la capacidad de un decodificador para proporcionar alguna, o todas, las capacidades descritas en el presente anexo. Los decodificadores que no soporten las capacidades adicionales pueden simplemente descartar cualesquiera de los bits de información PSUPP de nueva definición que aparecen en el tren de bits. La presencia de esta información de mejora suplementaria viene indicada por la presencia de tanto el bit PEI como la del octeto PSUPP que le sigue, y cuyo campo FTYPE tenga uno de los dos valores de nueva definición. La interpretación básica de PEI, PSUPP, FTYPE, y DSIZE es idéntica a la del anexo L y a la de las cláusulas 5.1.24 y 5.1.25.

#### W.2 Referencias

Las siguientes Recomendaciones del UIT-T y otras referencias contienen disposiciones que, mediante su referencia en este texto, constituyen disposiciones de la presente Recomendación. Al efectuar esta publicación, estaban en vigor las ediciones indicadas. Todas las Recomendaciones y otras referencias son objeto de revisiones por lo que se preconiza que los usuarios de esta Recomendación investiguen la posibilidad de aplicar las ediciones más recientes de las Recomendaciones y otras referencias citadas a continuación. Se publica periódicamente una lista de las Recomendaciones UIT-T actualmente vigentes. En esta Recomendación, la referencia a un documento, en tanto que autónomo, no le otorga el rango de una Recomendación.

- ISO/CEI 10646:2003, *Information technology – Universal Multiple-Octet Coded Character Set (UCS)*.
- IETF RFC 2396 (1998), *Uniform Resource Identifiers (URI): Generic Syntax*.

#### W.3 Valores adicionales de FTYPE

Los dos valores que aparecían como reservados en el cuadro L.1 del anexo L, se definen en el cuadro W.1.

### Cuadro W.1/H.263 – Valores del tipo de función FTYPE

13	IDCT de punto fijo
14	Mensaje de imagen

#### W.4 Número máximo recomendado de octetos PSUPP

Cuando se utilice alguna de las funciones FTYPE antes citadas, y que se definen en este anexo, se deberá mantener razonablemente pequeño, en relación con el tamaño de la imagen codificada, el número total de octetos PSUPP por imagen, que no deberá ser mayor de 256 octetos, con independencia del tamaño de la imagen codificada.

NOTA – Algunos de los protocolos de transmisión utilizados en el transporte del tren de bits de vídeo pueden dar soporte a la repetición externa de los contenidos de los encabezamientos de las imágenes con el fin de mejorar la elasticidad a errores, y pueden limitar la cantidad de dichos datos que se pueden repetir de un encabezamiento de imagen (por ejemplo, 504 bits en el formato de paquetes de IETF RFC 2429). La inclusión de un número mayor de octetos PSUPP puede conllevar la falta de un protocolo externo de este tipo que proporcione la repetición completa de los contenidos de los encabezamientos de imagen.

#### W.5 IDCT de punto fijo

La función IDCT de punto fijo indica el uso de una aproximación IDCT particular en la construcción del tren de bits. DSIZE será igual a 1 para la función IDCT de punto fijo. El octeto de datos PSUPP que sigue, especifica la implementación IDCT particular. Un valor igual a 0 indica la referencia IDCT 0, como se describe en W.5.3; los valores 1 a 255 están reservados.

##### W.5.1 Funcionamiento del decodificador

La capacidad de un decodificador para llevar a cabo una IDCT de punto fijo determinada, puede ser señalizada al codificador utilizando medios externos (por ejemplo, Rec. UIT-T H.245). Al recibir un tren de bits codificados con la indicación de IDCT de punto fijo, un decodificador deberá utilizar la IDCT de punto fijo en cuestión, si es capaz de hacerlo.

##### W.5.2 Eliminación de la actualización forzada

El anexo A especifica los requisitos de exactitud para la transformada discreta de coseno inversa (IDCT), permitiendo numerosas implementaciones que los cumplen. Para controlar la acumulación de errores debidos al desajuste de las IDCT entre el codificador y el decodificador, la cláusula 4.4 que trata de la actualización forzada, exige que los macrobloques se codifiquen en modo INTRA al menos una vez cada 132 veces cuando se transmiten los coeficientes.

Si el tren de bits indica el tipo de función IDCT de punto fijo, entonces se elimina el requisito de actualización forzada, y se deja sin regular la frecuencia a la que debe producirse la codificación INTRA. Sin embargo, un codificador deberá seguir utilizando la actualización forzada, a no ser que se haya asegurado por medios externos de que el decodificador es capaz de operar con la IDCT de punto fijo particular aquí especificada; en otro caso, se producirá un desajuste.

##### W.5.3 Referencia IDCT 0

La referencia IDCT 0 es cualquier implementación que, para cada bloque de entrada, genere valores de salida idénticos a los del programa C con el listado de código fuente que sigue.

NOTA – Esta IDCT de punto fijo cumple con el anexo A, pero no cumple con el requisito de rango ampliado de valores del anexo A de UIT-T Rec. H.262 | ISO/CEI 13818-2.

```

/*****
 *
 *                      FIXED-POINT IDCT
 *
 * Fixed-point fast, separable idct
 * Storage precision: 16 bits signed
 * Internal calculation precision: 32 bits signed
 * Input range: 12 bits signed, stored in 16 bits
 * Output range: [-256, +255]
 * All operations are signed
 *
 *****/

/*
 * Includes
 */

#include <stdlib.h>
#include <stdio.h>

/*
 * Typedefs
 */

typedef short int REGISTER; /* 16 bits signed */
typedef long int LONG; /* 32 bits signed */

/*
 * Global constants
 */

const REGISTER cpo8   = 0x539f; /* 32768*cos(pi/8)*1/sqrt(2) */
const REGISTER spo8   = 0x4546; /* 32768*sin(pi/8)*sqrt(2) */
const REGISTER cpo16  = 0x7d8a; /* 32768*cos(pi/16) */
const REGISTER spo16  = 0x18f9; /* 32768*sin(pi/16) */
const REGISTER c3po16 = 0x6a6e; /* 32768*cos(3*pi/16) */
const REGISTER s3po16 = 0x471d; /* 32768*sin(3*pi/16) */
const REGISTER OoR2   = 0x5a82; /* 32768*1/sqrt(2) */

/*
 * Function declarations
 */

void Transpose(REGISTER block[64]);
void HalfSwap(REGISTER block[64]);
void Swap(REGISTER block[64]);
void Scale(REGISTER block[64], signed char sh);
void Round(REGISTER block[64], signed char sh,
           const REGISTER min, const REGISTER max);
REGISTER Multiply(const REGISTER a, REGISTER x, signed char sh);
void Rotate(REGISTER *x, REGISTER *y,
            signed char sha, signed char shb,
            const REGISTER a, const REGISTER b,
            int inv);
void Butterfly(REGISTER column[8], char pass);
void IDCT(REGISTER block[64]);

```



```

/*
 * Transpose():
 *   Transpose a block
 * Input:
 *   REGISTER block[64]
 * Output:
 *   block
 * Return value:
 *   none
 */
void Transpose(REGISTER block[64])
{
    int i, j;
    REGISTER temp;

    for (i=0; i<8; i++) {
        for (j=0; j<i; j++) {
            temp = block[8*i+j];
            block[8*i+j] = block[8*j+i];
            block[8*j+i] = temp;
        }
    }
    return;
}

/*
 * HalfSwap():
 *   One-dimensional swap
 * Input:
 *   REGISTER block[64]
 * Output:
 *   block
 * Return value:
 *   none
 */
void HalfSwap(REGISTER block[64])
{
    int i;
    REGISTER temp;

    for (i=0; i<8; i++) {
        temp = block[8+i];
        block[8+i] = block[32+i];
        block[32+i] = temp;
        temp = block[24+i];
        block[24+i] = block[48+i];
        block[48+i] = temp;
        temp = block[40+i];
        block[40+i] = block[56+i];
        block[56+i] = temp;
    }
    return;
}

/*
 * Swap():
 *   Swap and transpose a block
 * Input:
 *   REGISTER block[64]
 * Output:
 *   block
 * Return value:
 *   none
 */

```

```

void Swap(REGISTER block[64])
{
    HalfSwap(block);
    Transpose(block);
    HalfSwap(block);
}

/*
 * Scale():
 *     Scale a block
 * Input:
 *     REGISTER block[64]
 *     signed char sh
 * Output:
 *     block
 * Return value:
 *     none
 */
void Scale(REGISTER block[64], signed char sh)
{
    int i;

    if (sh>0) {
        for (i=0; i<64; i++)
            block[i] >>= sh;
    }
    else {
        for (i=0; i<64; i++)
            block[i] <<= -sh;
    }
}

/*
 * Round():
 *     Performs the final rounding of an 8x8 block
 * Input:
 *     REGISTER block[64]
 *     signed char sh
 *     const REGISTER min
 *     const REGISTER max
 * Output:
 *     block
 * Return value:
 *     none
 */
void Round(REGISTER block[64], signed char sh,
           const REGISTER min, const REGISTER max)
{
    int i;

    for (i=0; i<64; i++) {
        if (block[i] < 0x00007FFF - (1<<(sh-1)))
            block[i] += (1<<(sh-1));
        else
            block[i] = 0x00007FFF;
        block[i] >>= sh;
        block[i] = (block[i]<min) ? min : ((block[i]>max) ? max : block[i]);
    }
    return;
}

/*
 * Multiply():
 *     Multiply by a constant with shift

```

```

* Input:
*     const REGISTER a
*     REGISTER x
*     signed char sh
* Output:
*     none
* Return value:
*     REGISTER, the result of the multiply
*/
REGISTER Multiply(const REGISTER a, REGISTER x, signed char sh)
{
    LONG tmp;
    REGISTER reg_out;

    /* multiply */
    tmp = (LONG)a * (LONG)x;

    /* shift */
    if (sh > 0)
        tmp >>= sh;
    else
        tmp <<= -sh;

    /* rounding and saturating */
    if (tmp < 0x7FFFFFFF - 0x00007FFF)
        tmp = tmp + 0x00007FFF;
    else
        tmp = 0x7FFFFFFF;

    reg_out = (REGISTER)(tmp >>16);

    return(reg_out);
}

/*
* Rotate():
*     Perform rotate operation on two registers
* Input:
*     REGISTER *x           pointer to the 1st register
*     REGISTER *y           pointer to the 2nd register
*     signed char sha       shift associated with factor a
*     signed char shb       shift associated with factor b
*     const REGISTER a     factor a
*     const REGISTER b     factor b
*     int inv               1 for inverse dct, 0 for forward dct
* Output:
*     *x, *y
* Return value:
*     none
*/
void Rotate(REGISTER *x, REGISTER *y,
            signed char sha, signed char shb,
            const REGISTER a, const REGISTER b,
            int inv)
{
    LONG tmp1xa, tmp1ya, tmp1xb, tmp1yb;
    LONG tmp11, tmp12;

    /*
     * intermediate calculation
     */

```

```

tmplxa = (LONG)(*x) * (LONG)a;
if (sha > 0)
    tmplxa >>= sha;
else
    tmplxa <<= -sha;

tmplya = (LONG)(*y) * (LONG)a;
if (sha > 0)
    tmplya >>= sha;
else
    tmplya <<= -sha;

tmplxb = (LONG)(*x) * (LONG)b;
if (shb > 0)
    tmplxb >>= shb;
else
    tmplxb <<= -shb;

tmplyb = (LONG)(*y) * (LONG)b;
if (shb > 0)
    tmplyb >>= shb;
else
    tmplyb <<= -shb;

/*
 * rounding and rotation
 */

if (inv) {
    tmplxa += 0x00007FFF;
    tmplxb += 0x00007FFF;

    tmp11 = tmplxb - tmplya;
    tmp12 = tmplxa + tmplyb;
}
else {
    tmplya += 0x00007FFF;
    tmplyb += 0x00007FFF;

    tmp11 = tmplxb + tmplya;
    tmp12 = -tmplxa + tmplyb;
}

/*
 * final rounding
 */

*x = (REGISTER) (tmp11 >>16);
*y = (REGISTER) (tmp12 >>16);

return;
}

/*
 * Butterfly():
 *   Perform 1D IDCT on a column
 * Input:
 *   REGISTER column[8]
 *   char pass
 * Output:
 *   column
 * Return value:
 *   none
 */

```

```

void Butterfly(REGISTER column[8], char pass)
{
    int i;
    REGISTER shadow_column[8];

    /*
     * For readability, we use a shadow column
     * that contains the state of column at the
     * preceding stage of the butterfly.
     */

    /*
     * Initialization
     */

    for (i=0; i<8; i++)
        shadow_column[i] = column[i];

    /*
     * First Phase
     */

    Rotate(column+2, column+6, pass-2, pass-1, cpo8, spo8, 1);
    Rotate(column+1, column+7, pass-1, pass-1, cpo16, spo16, 1);
    Rotate(column+3, column+5, pass-1, pass-1, c3po16, s3po16, 1);

    if (pass) {
        int a, tmp=column[4], b=column[0];
        a = b+tmp;
        b = b-tmp;
        column[0] = (a - ((tmp<0) ? 1 : 0)) >> 1;
        column[4] = (b - ((tmp<0) ? 1 : 0)) >> 1;
    }
    else {
        column[0] = shadow_column[0] + shadow_column[4];
        column[4] = shadow_column[0] - shadow_column[4];
    }

    for (i=0; i<8; i++)
        shadow_column[i] = column[i];

    /*
     * Second Phase
     */

    column[1] = shadow_column[1] - shadow_column[3];
    column[3] = shadow_column[1] + shadow_column[3];

    column[7] = shadow_column[7] - shadow_column[5];
    column[5] = shadow_column[7] + shadow_column[5];

    column[0] = shadow_column[0] + shadow_column[6];
    column[6] = shadow_column[0] - shadow_column[6];

    column[4] = shadow_column[4] + shadow_column[2];
    column[2] = shadow_column[4] - shadow_column[2];

    for (i=0; i<8; i++)
        shadow_column[i] = column[i];

    /*
     * Third Phase
     */
}

```

```

column[7] = shadow_column[7] - shadow_column[3];
column[3] = shadow_column[7] + shadow_column[3];

column[1] = Multiply(OoR2, shadow_column[1], -2);
column[5] = Multiply(OoR2, shadow_column[5], -2);

for (i=0; i<8; i++)
    shadow_column[i] = column[i];

/*
 * Fourth Phase
 */

column[4] = shadow_column[4] + shadow_column[3];
column[3] = shadow_column[4] - shadow_column[3];

column[2] = shadow_column[2] + shadow_column[7];
column[7] = shadow_column[2] - shadow_column[7];

column[0] = shadow_column[0] + shadow_column[5];
column[5] = shadow_column[0] - shadow_column[5];

column[6] = shadow_column[6] + shadow_column[1];
column[1] = shadow_column[6] - shadow_column[1];

return;
}

/*
 * IDCT():
 *   Perform 2D IDCT on a block
 * Input:
 *   REGISTER block[64]
 * Output:
 *   block
 * Return value:
 *   none
 */
void IDCT(REGISTER block[64])
{
    int i;

    Scale(block, -4);

    for (i=0; i<8; i++)
        Butterfly(block+8*i, 0);

    Transpose(block);

    for (i=0; i<8; i++)
        Butterfly(block+8*i, 1);

    Round(block, 6, -256, 255);

    Swap(block);
}

```

Con fines informativos se presenta a continuación la implementación de una transformada discreta del coseno directa (FDCT, *forward discrete cosine transform*), que tiene relación con los temas aquí tratados, aunque no forma parte integral de la presente Recomendación.

```

/*****
 *
 *                      FIXED-POINT FDCT
 *
 * Fixed-point fast, separable fdct
 * Storage precision: 16 bits signed
 * Internal calculation precision: 32 bits signed
 * Input range: 9 bits signed, stored in 16 bits
 * Output range: [-2048, +2047]
 * All operations are signed
 *
 *****/

/*
 * Function declarations
 */

void FButterfly(REGISTER column[8]);
void FDCT(REGISTER block[64]);

/*
 * FButterfly():
 *     Perform 1D FDCT on a column
 * Input:
 *     REGISTER column[8]
 * Output:
 *     column
 * Return value:
 *     none
 */
void FButterfly(REGISTER column[8])
{
    int i;
    REGISTER shadow_column[8];

    /*
     * For readability, we use a shadow column
     * that contains the state of column at the
     * preceding stage of the butterfly.
     */

    /*
     * Initialization
     */

    for (i=0; i<8; i++)
        shadow_column[i] = column[i];

    /*
     * First Phase
     */

    for (i=0; i<4; i++) {
        column[i] = shadow_column[i] + shadow_column[7-i];
        column[7-i] = shadow_column[i] - shadow_column[7-i];
    }

    for (i=0; i<8; i++)
        shadow_column[i] = column[i];
}

```

```

/*
 * Second Phase
 */

column[0] = shadow_column[0] + shadow_column[3];
column[3] = shadow_column[0] - shadow_column[3];

column[1] = shadow_column[1] + shadow_column[2];
column[2] = shadow_column[1] - shadow_column[2];

column[4] = Multiply(OoR2, shadow_column[4], -2);
column[7] = Multiply(OoR2, shadow_column[7], -2);

column[6] = shadow_column[6] - shadow_column[5];
column[5] = shadow_column[6] + shadow_column[5];

for (i=0; i<8; i++)
    shadow_column[i] = column[i];

/*
 * Third Phase
 */

column[0] = shadow_column[0] + shadow_column[1];
column[1] = shadow_column[0] - shadow_column[1];

column[6] = shadow_column[6] - shadow_column[4];
column[4] = shadow_column[6] + shadow_column[4];

column[7] = shadow_column[7] - shadow_column[5];
column[5] = shadow_column[7] + shadow_column[5];

for (i=0; i<8; i++)
    shadow_column[i] = column[i];

/*
 * Fourth Phase
 */

Rotate(column+2, column+3, -2, -1, cpo8, spo8, 0);
Rotate(column+4, column+5, -1, -1, cpo16, spo16, 0);
Rotate(column+6, column+7, -1, -1, c3po16, s3po16, 0);

return;
}

/*
 * FDCT():
 *     Perform 2D FDCT on a block
 * Input:
 *     REGISTER block[64]
 * Output:
 *     block
 * Return value:
 *     none
 */
void FDCT(REGISTER block[64])
{
    int i;

    for (i=0; i<8; i++)
        FButterfly(block+8*i);
}

```



```

Transpose(block);

for (i=0; i<8; i++)
    FButterfly(block+8*i);

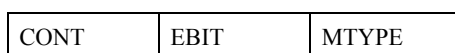
Round(block, 3, -2048, 2047);

Swap(block);
}

```

## W.6 Mensaje de imagen

La función mensaje de imagen indica la presencia de uno o más octetos que representan datos del mensaje. El primer octeto de los datos del mensaje es un encabezamiento de mensaje con la siguiente estructura, como se muestra en la figura W.1.



**Figura W.1/H.263 – Estructura del primer octeto del mensaje**

DSIZE será igual al número de octetos en los datos del mensaje que corresponden a una función mensaje de imagen, y que incluye al primer octeto que se muestra en la figura W.1.

Los decodificadores deberán analizar los datos del mensaje de imagen como requiere la sintaxis PSUPP básica, aunque, por otra parte está sin definir la respuesta del decodificador a los mensajes de imagen.

### W.6.1 Continuación (CONT) (1 bit)

Si es igual a "1", CONT indica que los datos del mensaje asociados con esta función del mensaje de imagen son parte del mismo mensaje lógico que los datos del mensaje asociados con la siguiente función del mensaje de imagen. Si es igual a "0", CONT indica que los datos del mensaje asociados con esta función del mensaje de imagen termina el actual mensaje lógico. Se puede utilizar CONT, por ejemplo, para incluir mensajes lógicos que se extienden por más de 14 octetos.

### W.6.2 Posición del bit final (EBIT, *end bit position*) o número de pista (3 bits)

Para mensajes de imágenes que no son textos, EBIT especifica el número de los bits menos significativos que serán ignorados en el octeto del último mensaje. En mensajes de imágenes que no son textos, si CONT es "1", o si sólo hay un octeto de mensaje, (es decir el octeto de la figura W.1), EBIT será igual "0". El número de bits del mensaje válidos para una función de mensaje de imagen que no es texto, sin tener en cuenta los bits CONT/EBIT/MTYPE, es igual a  $(DSIZE - 1) \times 8 - EBITS$ . El número de bits válidos del mensaje para un mensaje lógico puede ser mayor debido a la prolongación del mensaje.

Para tipos de mensajes de imagen que contienen información de texto, EBIT contendrá un número de pista de texto. El significado preciso del número de pista de texto no se especifica aquí, pero deberá indicar un tipo particular (por ejemplo, idioma) del texto. Se considerará un número de pista cero como pista por defecto.

### W.6.3 Tipo de mensaje (MTYPE, *message type*) (4 bits)

MTYPE indica el tipo de mensaje. Los tipos definidos se presentan en el cuadro W.2.

**Cuadro W.2/H.263 – Valores para el tipo de mensaje, MTYPE**

0	Datos binarios arbitrarios
1	Texto arbitrario
2	Texto de derechos de autor
3	Texto de subtítulo
4	Texto de descripción de vídeo
5	Texto de identificador uniforme de recurso
6	Repetición del encabezamiento de la imagen presente
7	Repetición del encabezamiento de la imagen previa
8	Repetición del encabezamiento de la imagen siguiente, TR fiable
9	Repetición del encabezamiento de la imagen siguiente, TR no fiable
10	Indicación de campo entrelazado superior
11	Indicación de campo entrelazado inferior
12	Número de imagen
13	Imágenes de referencia de reserva
14..15	Reservado

#### **W.6.3.1 Datos binarios arbitrarios**

El tipo datos binarios arbitrarios se utiliza para transportar cualquier mensaje binario codificado que no sea del tipo ISO/CEI 10646 UTF-8. La interpretación de los contenidos de los datos binarios arbitrarios está fuera del alcance de la presente Recomendación, aunque deberán empezar con algún patrón identificativo (por ejemplo, un código identificador de cuatro octetos) que ayude a distinguir a uno entre estos tipos de datos.

#### **W.6.3.2 Texto arbitrario**

Se utiliza el texto arbitrario para transportar un mensaje genérico de texto codificado según ISO/CEI 10646 UTF-8. Mensajes de texto más específicos, tal como los de información de derechos de autor, se representarán por otros tipos de mensajes a propósito (por ejemplo, texto de derechos de autor).

#### **W.6.3.3 Texto de derechos de autor**

El texto de derechos de autor se utilizará sólo para el transporte de información de propiedad intelectual en relación con la autoría de la representación codificada en el tren de bits. El mensaje derechos de autor se codificará según ISO/CEI 10646 UTF-8.

#### **W.6.3.4 Texto de subtítulo**

El texto de subtítulo se utilizará únicamente para transportar la información de subtítulo asociada con la imagen actual, y con las subsiguientes del tren de bits. El mensaje de subtítulo se codificará de acuerdo con ISO/CEI 10646 UTF-8. El texto de subtítulo estará insertado en el tren de bits como si se fuese a presentar en un área de texto separada en la que se añade texto nuevo al final del texto previo, y el texto preexistente se desplaza a partir del punto de inserción. El código de control de Avance de Página (hexadecimal "0x000C") se utilizará para indicar el borrado del área de texto visible. El código de control Fin del Medio Físico (hexadecimal "0x0019") se utilizará para indicar la situación del "fin de texto de subtítulo". Sin embargo, esta Recomendación no pone restricciones sobre la forma en la que en realidad se presenta y almacena este texto de subtítulo.

### **W.6.3.5 Texto de descripción de vídeo**

Se utilizará el texto de descripción de vídeo sólo para transportar información descriptiva asociada con el contenido de la información del tren de bits actual. La descripción de vídeo se codificará de acuerdo con ISO/CEI 10646 UTF-8. El texto de la descripción de vídeo se insertará en el tren de bits como si fuese a ser presentado en un área de texto separada en la que el nuevo texto se añade al final del texto previo, y el texto preexistente se desplaza a partir del punto de inserción. El código de control de Avance de Página (hexadecimal "0x000C") se utilizará para indicar el borrado del área de texto visible. El código de control Fin del Medio Físico (hexadecimal "0x0019") se utilizará para indicar la situación del "fin de texto de descripción". Sin embargo, esta Recomendación no pone restricciones sobre la forma en la que en realidad se presenta y almacena este texto de descripción de vídeo.

### **W.6.3.6 Identificador uniforme de recurso (*URI, uniform resource identifier*)**

El mensaje consiste en un identificador uniforme de recurso (URI) del tipo definido en IETF RFC 2396. Se codificará el URI de acuerdo con ISO/CEI 10646 UTF-8.

### **W.6.3.7 Repetición del encabezamiento de la imagen presente**

En este mensaje se repite el encabezamiento de imagen de la imagen presente. Los bits repetidos no incluyen ninguna información de mejora suplementaria (PEI/PSUPP). Deberán incluirse todos los demás bits hasta los de las capas de GOB o de ranura, sujetos a las limitaciones de W.4.

### **W.6.3.8 Repetición del encabezamiento de la imagen previa**

En este mensaje se repite el encabezamiento de imagen de la imagen previamente transmitida. Los bits repetidos no incluyen a los dos primeros bytes del código de comienzo de imagen (PSC, *picture start code*), ni a ninguna información de mejora suplementaria (PEI/PSUPP). Deberán incluirse todos los demás bits hasta los de las capas de GOB o de ranura, sujetos a las limitaciones de W.4.

### **W.6.3.9 Repetición del encabezamiento de la imagen siguiente, referencia temporal fiable**

En este mensaje se repite el encabezamiento de imagen de la siguiente imagen a transmitir. Los bits repetidos no incluyen a los dos primeros bytes del código de comienzo de imagen (PSC), ni a ninguna información de mejora suplementaria (PEI/PSUPP). Deberán incluirse todos los demás bits hasta los de las capas de GOB o de ranura, sujetos a las limitaciones de W.4.

### **W.6.3.10 Repetición del encabezamiento de la imagen siguiente, referencia temporal no fiable**

En este mensaje se repite el encabezamiento de imagen de la siguiente imagen a transmitir. Los bits repetidos no incluyen a los tres primeros bytes del encabezamiento de imagen, ni a ninguna información de mejora suplementaria (PEI/PSUPP). Deberán incluirse todos los demás bits hasta los de las capas de GOB o de ranura, sujetos a las limitaciones de W.4. Cualquiera de los bits de TR o ETR del encabezamiento de imagen repetido, no tienen porque ser necesariamente los mismos que los bits correspondientes del encabezamiento de la siguiente imagen.

### **W.6.3.11 Indicaciones de campo entrelazado**

En el caso de indicaciones de campo entrelazado, el mensaje consiste en una indicación de la codificación del campo entrelazado. Esta indicación no afecta al proceso de decodificación. Sin embargo, indica que la imagen actual no ha sido barrida como una imagen de barrido progresivo. En otras palabras, indica que la imagen codificada actual contiene sólo la mitad de las líneas que la imagen fuente con resolución completa. En las indicaciones de campo entrelazado DSIZE será 1, CONT será 0, y EBIT será 0. En el caso de codificación de campo entrelazado, cada incremento de la referencia temporal denota el tiempo entre el muestreo de los campos alternados de media imagen, en lugar del tiempo entre dos imágenes completas. En el caso de una indicación de campo entrelazado superior, la imagen actual contiene a las líneas primera (la superior), tercera, quinta, etc. de la imagen completa. En el caso de una indicación de campo entrelazado inferior, la imagen

actual contiene a las líneas segunda, cuarta, sexta, etc. de la imagen completa. Cuando se envían indicaciones de campo entrelazado, el codificador cumplirá las siguientes normas:

- 1) El codificador utilizará una frecuencia de reloj de imagen (frecuencia de reloj de imagen personalizada, en caso necesario) tal que, cada nuevo campo del vídeo fuente original se corresponda con un incremento 1 de la referencia temporal.
- 2) El codificador utilizará un tamaño de imagen (tamaño de imagen personalizado, en caso necesario) tal que, las dimensiones de la imagen se correspondan con las de un único campo.
- 3) El codificador utilizará una relación de aspecto de píxel (relación de aspecto de píxel personalizada, en caso necesario) tal que, la relación de aspecto de la altura total de la imagen se corresponda con la relación de aspecto de imagen, derivada de la relación de aspecto de píxel del campo único representado por la imagen actual codificada.

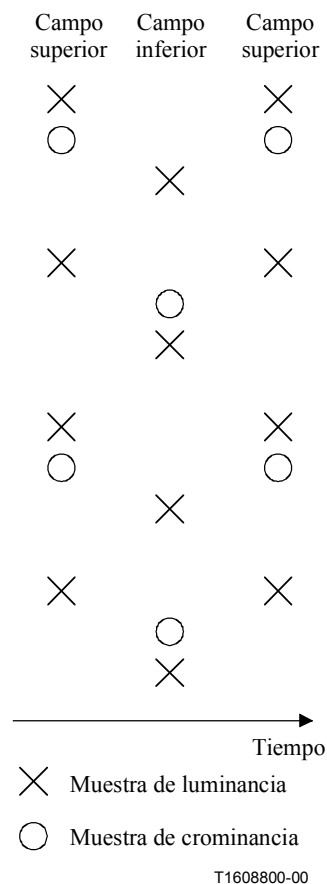
Originalmente se introdujo el barrido entrelazado de campos como una técnica de compresión de vídeo analógico. Aunque el barrido progresivo de imagen se considera generalmente superior para la compresión y presentación digitales, el uso del barrido entrelazado de campos ha persistido en muchos diseños de cámaras y visualizadores. La codificación de campo entrelazado (que puede implementarse con un retardo inferior que, tanto la imagen completa entrelazada, o que la codificación de la imagen con barrido progresivo a velocidad mitad que la del campo entrelazado) está por tanto soportada por las indicaciones dadas aquí.

Un codificador no enviará indicaciones de campo entrelazado a no ser que se haya establecido previamente, por medios externos (por ejemplo, Rec. UIT-T H.245) la capacidad del decodificador para recibir y procesar adecuadamente este tipo de imágenes basadas en el campo. El fallo en el establecimiento de esta capacidad del decodificador puede producir un molesto temblor vertical de pequeña amplitud en la imagen decodificada, recibida y presentada por un decodificador.

Por ejemplo, un codificador puede utilizar la codificación de campo entrelazado aplicando el modo selección de la imagen de referencia (especificado en el anexo N) o el modo selección mejorada de imagen de referencia (especificado en el anexo U) para permitir el direccionamiento de más de un campo previo. Para una codificación de campo entrelazado "525/60" de una imagen con relación de aspecto 4:3, con 704 muestras de luminancia codificadas por línea y 240 líneas de luminancia codificadas por campo, el codificador utilizará un tamaño de imagen personalizado que tenga una anchura de imagen de 704 y una altura de imagen de 240, una relación de aspecto de píxel personalizada de 5:11, y una frecuencia de reloj de imagen personalizada especificada con un código de conversión de reloj "1" y un divisor de reloj de 30. Para una codificación de campo entrelazado "625/50" de una imagen con relación de aspecto 4:3, con 704 muestras de luminancia codificadas por línea y 288 líneas de luminancia codificadas por campo, el codificador utilizará un tamaño de imagen personalizado que tenga una anchura de imagen de 704 y una altura de imagen de 288, una relación de aspecto de píxel personalizada de 6:11, y una frecuencia de reloj de imagen personalizada especificada con un código de conversión de reloj "0" y un divisor de reloj de 36.

Las posiciones verticales de muestreo de las muestras de crominancia en la codificación de campo entrelazado de una imagen de campo superior, se especifican como si estuviesen desplazadas hacia arriba en 1/4 de la altura de las muestras de luminancia con referencia a la rejilla de muestreo del campo, de forma que dichas muestras queden alineadas verticalmente con la posición normal relativa de la rejilla de muestreo de la imagen completa. Las posiciones verticales de muestreo de las muestras de crominancia en la codificación de campo entrelazado de una imagen de campo inferior, se especifican como si estuviesen desplazadas hacia abajo en 1/4 de la altura de las muestras de luminancia con referencia a la rejilla de muestreo del campo, de forma que dichas muestras queden alineadas verticalmente con la posición normal en relación con la rejilla de muestreo de la imagen completa. Las posiciones de muestreo horizontal de las muestras de crominancia se especifican como si no estuviesen afectadas por la aplicación de la codificación de

campo entrelazado. En la figura W.2 se muestran las posiciones verticales de muestreo con sus correspondientes posiciones de muestreo en función del tiempo.



**Figura W.2/H.263 – Alineación vertical y temporal de las muestras de crominancia en la codificación de campo entrelazado**

### W.6.3.12 Número de imagen

No se hará uso de este mensaje si está en uso el anexo U. El mensaje contiene dos bytes de datos que albergan a un número de imagen de 10 bits . Por tanto, DSIZE será 3, CONT será 0, y EBIT será 6. El número de imagen se incrementará en 1, en una operación de módulo 10, cada vez que se codifique y transmita una imagen I o P, o una trama PB o PB mejorada. Para imágenes EI y EP, el número de imagen se incrementará para cada imagen EI o EP que esté dentro de la misma capa de mejora de escalabilidad. Para las imágenes B, se incrementará el número de imagen en relación con el valor que tiene en la imagen no B más reciente en la capa de referencia de la imagen B que precede a la imagen B en el orden del tren de bits (una imagen que es temporalmente subsiguiente a la imagen B). Si imágenes adyacentes en la misma capa de mejora tienen la misma referencia temporal, y si está en uso el modo selección de imagen de referencia (véase el anexo N), el decodificador deberá considerar esta situación como una indicación de que se han enviado copias redundantes con aproximadamente el mismo contenido de escena de la imagen, y que todas estas imágenes compartirán el mismo número de imagen. Si la diferencia (módulo 1024) de los números de imagen de dos imágenes no B, recibidas consecutivamente en la misma capa de mejora, no es 1, y si las imágenes no representan aproximadamente el mismo contenido de escena de la imagen, como se ha descrito antes, el decodificador inferirá una pérdida de imágenes o una corrupción de los datos.

### **W.6.3.13 Imágenes de referencia de reserva**

Los codificadores pueden utilizar este mensaje para instruir a los decodificadores acerca de qué imágenes se parecen lo suficiente a la imagen actual de referencia para la compensación de movimiento, de manera que una de ellas se pueda utilizar como imagen de referencia si se perdiese en la transmisión la imagen de referencia actual. Si un decodificador no dispone de una imagen de referencia actual, pero puede acceder a una imagen de referencia de reserva, no pedirá una actualización de imagen INTRA. Depende de los codificadores la elección de las imágenes de referencia de reserva, en caso de elegir alguna. Los bytes de los datos del mensaje contienen los números de imagen de las imágenes de referencia de reserva, en orden de preferencia (apareciendo como primera la de mayor preferencia). Los números de imagen se refieren a los valores que se transmiten de acuerdo con el anexo U o W.6.3.12. Se puede utilizar este mensaje para los tipos de imágenes P, B, PB, PB mejorada y EP. Sin embargo, si se está utilizando el anexo N, o el anexo U, y si la imagen está asociada con varias imágenes de referencia, no se utilizará este mensaje. En el caso de imágenes EP, sólo se utilizará el mensaje en la predicción hacia adelante, mientras que la predicción hacia arriba se hace siempre a partir de la imagen de la capa de referencia que corresponde temporalmente. Para los tipos de imágenes B, PB, y PB mejorada, especifica una imagen para ser utilizada como referencia de predicción hacia adelante. Este mensaje no se utilizará si la imagen es del tipo I o EI.

## **Anexo X**

### **Definición de perfiles y niveles**

#### **X.1 Alcance**

Con la variedad de modos facultativos disponibles en la presente Recomendación, es de vital importancia que se definan varias combinaciones de modos preferidos para la operación, de manera que los terminales mejorados con opciones tengan grandes probabilidades de conectarse entre sí utilizando alguna sintaxis mejor que la del perfil de "línea de base". Este anexo contiene una lista de combinaciones preferidas de características, estructuradas en "perfiles" de soporte. También define algunas agrupaciones de parámetros de máxima calidad de funcionamiento como "niveles" de soporte para estos perfiles. Los objetivos primarios de este anexo son:

- 1) proporcionar un medio sencillo para describir o negociar las capacidades de un decodificador (mediante la especificación de parámetros de perfil y de nivel),
- 2) promover el soporte, en decodificadores, de características de mejoras comunes para conseguir una interoperabilidad máxima, y
- 3) describir conjuntos de características seleccionados como particularmente apropiados para el tratamiento de ciertas aplicaciones esenciales.

Los perfiles y los niveles se definen en las siguientes cláusulas y los cuadros X.1 y X.2. El intervalo mínimo de imagen especificado en el cuadro X.2 es la diferencia mínima de tiempo entre decodificaciones de imágenes consecutivas en el tren de bits. El sistema que soporta un nivel diferente del nivel 45 debe soportar todos los niveles inferiores. El sistema que soporta el nivel 45 debe soportar también el nivel 10.

#### **X.2 Perfiles de soporte del modo preferido**

Los perfiles de soporte se definen por el conjunto de características soportadas en el decodificador para cada perfil. El soporte en el decodificador de un perfil determinado implica el soporte de todas las combinaciones de subconjuntos válidas para los modos constitutivos de ese perfil. Se establece este requisito de manera que se minimicen las limitaciones impuestas a la selección de las

combinaciones de modo del codificador. Esto concuerda con el objetivo primario de este anexo, que es el de indicar los modos facultativos que estar soportados en el decodificador para el tratamiento de las aplicaciones esenciales, en lugar de imponer al codificador un pequeño conjunto particular de combinaciones de modos.

### **X.2.1 El perfil de la línea de base (perfil 0)**

Se define aquí el perfil de línea de base, designado como perfil 0, con objeto de proporcionar una designación de perfil para la capacidad de "línea de base" mínima de esta Recomendación. "Línea de base" se refiere a la sintaxis de esta Recomendación sin modos de operación facultativos. Este perfil de soporte está compuesto solamente del diseño de la línea de base.

### **X.2.2 Perfil de compatibilidad hacia atrás, versión 2, eficiencia de codificación H.320 (perfil 1)**

Se define aquí el perfil de compatibilidad hacia atrás, versión 2, eficiencia de codificación H.320 designado como perfil 1, para proporcionar compatibilidad con un conjunto de características adoptado dentro del mecanismo de intercambio de capacidades H.242 para uso por los sistemas terminales con conmutación de circuitos H.320. Este perfil proporciona eficiencia de codificación mejorada básica y funcionalidad mejorada simple dentro del conjunto de características disponible en la segunda versión de esta Recomendación (la cual no incluye los anexos U, V y W). Este perfil de soporte está compuesto del diseño de línea de base más los siguientes modos:

- 1) **Codificación INTRA avanzada (anexo I)** – La utilización de este modo mejora la eficacia de la codificación de macrobloques INTRA (ya sea en imágenes INTRA o en imágenes con codificación predictiva). Las exigencias computacionales de este modo son mínimas tanto en el codificador como en el decodificador (a lo sumo sólo se requieren 8 adiciones/substracciones por bloque de  $8 \times 8$  en el proceso de decodificación y el empleo de una tabla de VLC distinta, aunque muy similar, para obtener una mejora significativa en la eficacia de la codificación). Por ello, se incluye la codificación INTRA avanzada en este lote básico de soporte.
- 2) **Filtro de desbloqueo (anexo J)** – Debido a la importante mejora en la calidad subjetiva que se puede conseguir con un filtro de desbloqueo, estos filtros ya se utilizan ampliamente como un método de posprocesamiento en terminales de comunicación vídeo. El modo del anexo J es el modo preferido de funcionamiento para un filtro de desbloqueo, ya que coloca el filtro dentro del bucle de codificación. Esta colocación facilita la implementación del filtro (reduciendo la necesidad de memoria) y mejora algo la calidad de la codificación con respecto a una implementación posprocesamiento. Al igual que el modo predicción avanzada, este modo también incluye la característica de cuatro vectores de movimiento por macrobloque y extrapolación de demarcación de imagen para la compensación de movimiento, que pueden mejorar más aún la eficacia de la codificación. Las exigencias computacionales del filtro de desbloqueo son de varios cientos de operaciones por macrobloque codificado, pero los accesos a la memoria y las dependencias computacionales no presentan complicaciones. Esto último es lo que hace preferible el filtro de desbloqueo a la predicción avanzada, para algunas implementaciones. Además, las ventajas de la predicción avanzada no son tan notorias cuando se utiliza también el filtro de desbloqueo. Por ello, se incluye el filtro de desbloqueo en este lote básico de soporte.
- 3) **Información de mejora suplementaria de la congelación de toda la imagen (anexo L, cláusula L.4)** – La congelación de toda la imagen es muy sencilla de aplicar, ya que sólo requiere que el decodificador pueda detener la transferencia de nuevas imágenes de su memoria tampón de salida a la visualización de vídeo. Esta capacidad sirve para evitar la visualización de imágenes de baja fidelidad mientras el codificador está formando una imagen de mayor fidelidad.

- 4) **Cuantificación modificada (anexo T)** – Este modo incluye una gama de coeficientes DCT ampliada, sintaxis DQUANT modificada, y tamaño de escalón modificado para crominancia. Las dos primeras características permiten una mayor flexibilidad en el codificador y pueden hacer que disminuya realmente la carga computacional del codificador (eliminando la necesidad de recodificar los macrobloques necesarios cuando se llega a la saturación del nivel del coeficiente). La tercera característica mejora notablemente la fidelidad de la crominancia, normalmente con poco costo añadido en materia de velocidad binaria y prácticamente sin que aumente la carga computacional. En el decodificador, la única carga computacional significativa es la del análisis sintáctico de algunos símbolos nuevos del tren de bits.

### X.2.3 Perfil de compatibilidad hacia atrás versión 1 (perfil 2)

Se define aquí el perfil de compatibilidad hacia atrás versión 1, designado como perfil 2, para proporcionar una mejor calidad de funcionamiento en cuanto a la eficiencia de codificación dentro del conjunto de características disponible en la primera versión de la Rec. UIT-T H.263 (que no incluye información de mejora suplementaria ni ninguna de las características facultativas que utilizan PLUSPTYPE). Este perfil de soporte está compuesto del diseño de línea de base más el siguiente modo único:

- 1) **Predicción avanzada (anexo F)** – Desde el punto de vista de la eficacia de la codificación, este modo es el más importante de los disponibles en la primera versión (versión 1) de esta Recomendación. Incluye la compensación de movimiento de bloques superpuestos y la posibilidad de utilizar cuatro vectores de movimiento por macrobloque, y permite a los vectores de movimiento apuntar hacia afuera de demarcaciones de imagen. La utilización de la predicción avanzada proporciona considerables mejoras en la calidad de funcionamiento subjetiva y objetiva. Sin embargo, requiere un aumento apreciable del cálculo e introduce dependencias de datos complicadas en el orden de procesamiento en el decodificador. Sin embargo, puesto que algunas implementaciones de la presente Recomendación diseñadas antes de la adopción de los otros modos que figuran en esta lista, podrían haber incorporado la predicción avanzada por sí mismas, se recomienda la operación sólo predicción avanzada para una máxima calidad con la compatibilidad hacia atrás para los decodificadores de la versión 1.

### X.2.4 Perfil inalámbrico interactivo y de flujo continuo versión 2 (perfil 3)

Se define aquí el perfil inalámbrico interactivo y de flujo continuo versión 2, designado como perfil 3, para proporcionar una calidad de funcionamiento mejorada en cuanto a la eficiencia de codificación y una resiliencia a los errores mejorada la entrega a dispositivos inalámbricos dentro del conjunto de características disponible en la segunda versión de esta Recomendación (que no incluye los anexos U, V y W). Este perfil de soporte está compuesto del diseño de la línea de base más los siguientes modos:

- 1) **Codificación INTRA avanzada (anexo I)** – Véase el apartado 1 de X.2.2.
- 2) **Filtro de desbloqueo (anexo J)** – Véase el apartado 2 de X.2.2.
- 3) **Modo estructurado en rebanada (anexo K)** – Se incluye aquí el modo estructurado en rebanada debido a su aptitud mejorada para proporcionar puntos de resincronización dentro del tren de bits de vídeo para la recuperación tras situaciones de datos erróneos o perdidos. No se incluye en este perfil el soporte para los submodos ordenación arbitraria de rebanadas (ASO, *arbitrary slice ordering*) y rebanada rectangular (RS, *rectangular slice*) del modo estructurado en rebanadas, con objeto de limitar las exigencias de complejidad del decodificador. La carga computacional adicional impuesta por el modo estructurado en rebanada es mínima, y está limitada principalmente a la generación del tren de bits y al análisis sintáctico.
- 4) **Cuantificación modificada (anexo T)** – Véase el apartado 4 de X.2.2.



### X.2.5 Perfil inalámbrico interactivo y de flujo continuo versión 3 (perfil 4)

Se define aquí el perfil inalámbrico interactivo y de flujo continuo versión 3, designado como perfil 4, para proporcionar una calidad de funcionamiento mejorada en cuanto a la eficiencia de codificación y una resiliencia a los errores mejorada para la entrega a dispositivos inalámbricos, aprovechando las ventajas de las características mejoradas de la tercera versión de esta Recomendación. Este perfil de soporte está compuesto del diseño de la línea de base, más las siguientes características adicionales:

- 1) **Perfil 3** – Este conjunto de características proporciona varias mejoras útiles para el soporte de la transmisión de vídeo inalámbrica.
- 2) **Modo rebanadas particionadas para datos (anexo V)** – Esta característica mejora la resiliencia a los errores separando los datos del vector de movimiento de los datos del coeficiente DCT dentro de las rebanadas y protege la información del vector de movimiento (la parte más importante de los datos del macrobloque detallado) mediante la utilización de codificación de longitud variable reversible. No se incluye en este perfil el soporte de los submodos ordenación arbitraria de las rebanadas (ASO) y rebanada rectangular (RS), a fin de limitar las exigencias de complejidad del decodificador.
- 3) **Información de mejora suplementaria de repetición del encabezamiento de la imagen previa (anexo W, cláusula W.6.3.8)** – Esta característica permite al decodificador recibir y recuperar la información de encabezamiento a partir de una imagen previa en caso de pérdida o corrupción de datos.

### X.2.6 Perfil de alta compresión conversacional (perfil 5)

Se define aquí el perfil de alta compresión conversacional, designado como perfil 5, para proporcionar una calidad de funcionamiento mejorada en cuanto a la eficiencia de codificación sin añadir el retardo asociado con la utilización de imágenes B y sin añadir características de resiliencia a los errores. Este perfil de soporte está compuesto del diseño de línea de base, más las siguientes características adicionales:

- 1) **Perfil 1** – Este conjunto de características proporciona varias mejoras útiles para la eficiencia de codificación mejorada.
- 2) **Perfil 2** – Este perfil añade el modo predicción avanzado (anexo F), que proporciona una mejora adicional de la calidad de funcionamiento en cuanto a la eficiencia de codificación y compatibilidad hacia atrás con implementaciones de la primera versión de esta Recomendación.
- 3) **Vectores de movimiento sin restricciones con UUI = "1" (anexo D)** – El anexo D tiene dos características principales:
  - a) extrapolación de demarcación de imagen; y
  - b) soporte de vectores de movimiento más largos.

La primera de estas características ya está soportada con la inclusión del anexo J en el perfil 1. El soporte de vectores de movimiento más largos puede proporcionar una mejora significativa de la eficacia de la codificación, especialmente en casos de grandes tamaños de imagen, movimiento rápido, movimiento de la cámara, y bajas velocidades de imagen. Cuando se utiliza este modo estando presente PLUSTYPE, permite también diferencias de vectores de movimiento más largos, lo que puede simplificar considerablemente el funcionamiento del codificador. Los vectores de movimientos más largos representan, en efecto, un problema potencial para el decodificador en términos de acceso a la memoria, pero los límites impuestos al tamaño máximo del vector de movimiento en función del tamaño de la imagen impiden que este problema se convierta en un obstáculo apreciable en la implementación.

- 4) **Selección de imagen de referencia mejorada (anexo U)** – Este modo mejora significativamente la calidad de funcionamiento en cuanto a la eficiencia de la compresión ya que se pueden utilizar múltiples imágenes previas como datos de referencia para la predicción del nivel de macrobloque de las imágenes subsiguientes. En el perfil 5 no se incluye el submodo supresión de subimagen (anexo U, cláusula U.4.3) del modo selección de imagen de referencia mejorada.

### **X.2.7 Perfil Internet conversacional (perfil 6)**

Se define aquí el perfil Internet conversacional, designado como perfil 6, para proporcionar una calidad de funcionamiento mejorada en cuanto a la eficiencia de codificación sin añadir el retardo asociado con la utilización de imágenes B, pero añadiendo alguna resiliencia a los errores apropiada para uso en redes de protocolo Internet (IP, *Internet protocol*) (las cuales utilizan protocolos de datos basados en paquetes, con paquetes relativamente grandes, y en las que se producen pérdidas de datos en lugar de corrupción de datos). Este perfil de soporte está compuesto del diseño de línea de base, más las siguientes características adicionales:

- 1) **Perfil 5** – Este conjunto de características proporciona varias mejoras útiles para la eficiencia de codificación mejorada.
- 2) **Modo estructurado en rebanadas (anexo K) con submodo ordenación arbitraria de las rebanadas (ASO)** – Se incluye aquí el modo estructurado en rebanadas debido a su aptitud mejorada para proporcionar puntos de resincronización dentro del tren de bits de vídeo para la recuperación tras situaciones de pérdida de paquetes de datos. También se incluye el submodo ordenación arbitraria de rebanadas (ASO) del modo estructurado en rebanadas con objeto de permitir la paquetización entrelazada para la ocultación de error compensado por movimiento y para la recepción de datos fuera de secuencia. No se incluye en este perfil el soporte del submodo rebanada rectangular (RS) del modo estructurado en rebanadas, con objeto de limitar las exigencias de complejidad del decodificador. La carga computacional adicional impuesta por el modo estructurado en rebanadas es mínima y está limitada principalmente a la generación del tren de bits y al análisis sintáctico.

### **X.2.8 Perfil entrelazado conversacional (perfil 7)**

Se define aquí el perfil entrelazado conversacional, designado como perfil 7, para proporcionar una calidad de funcionamiento mejorada en cuanto a la eficiencia de codificación para aplicaciones de bajo retardo, más el soporte de fuentes de vídeo entrelazadas. Este perfil de soporte está compuesto del diseño de línea de base, más las siguientes características adicionales:

- 1) **Perfil 5** – Este conjunto de características proporciona varios medios útiles para mejorar la eficiencia de codificación sin añadir retardo.
- 2) **Indicaciones de campo entrelazado para imágenes de 240 líneas y de 288 líneas (anexo W, cláusula W.6.3.11)** – Esta característica permite enviar vídeo en un formato de imagen fuente entrelazada para que sea compatible con los diseños de cámara existentes.

### **X.2.9 Perfil de alta latencia (perfil 8)**

Se define aquí el perfil de alta latencia, designado como perfil 8, para proporcionar una calidad de funcionamiento mejorada en cuanto a la eficiencia de codificación para aplicaciones sin constricciones de retardo críticas. Este perfil de soporte está compuesto del diseño de línea de base, más las siguientes características adicionales:

- 1) **Perfil 6** – Este conjunto de características proporciona varios medios útiles para mejorar la eficiencia de codificación y la resiliencia a las pérdidas de datos.
- 2) **Remuestreo de la imagen de referencia (sólo el modo factor de 4 implícito) (anexo P, cláusula P.5)** – El modo factor de 4 implícito de remuestreo de la imagen de referencia permite el remuestreo (o repetición de muestreo) automático de la imagen de referencia solamente cuando cambia el tamaño de la nueva trama, como se indica en el

encabezamiento de la imagen. No se requiere tasa de tren de bits para este modo de funcionamiento. Los cambios de resolución dinámica predictiva permiten a un codificador efectuar compromisos inteligentes entre resolución temporal y espacial. Además, este sencillísimo modo de funcionamiento del anexo P (únicamente muestreo a velocidad superior o inferior con factor de 4) sólo añade modesta cantidad de complejidad computacional al codificador o decodificador, ya que en el caso del factor de 4 se utiliza un filtro fijo simple de respuesta a impulso finito (FIR, *finite impulse response*) (que requiere a lo sumo cuatro operaciones por píxel).

- 3) **Imágenes B (escalabilidad temporal, anexo O, cláusula O.1.1)** – Esta característica consiste en imágenes B, que son imágenes que permiten la predicción temporal bidireccional. La adición de imágenes B mejora la calidad de funcionamiento en cuanto a la eficiencia de codificación, pero con algún costo potencia de procesamiento y retardo de codificación y decodificación añadidos. En el perfil 8 no está soportado el submodo predicción hacia atrás de dos imágenes, para imágenes B en el modo selección de imagen de referencia mejorada (anexo U, cláusula U.3.1.5.5).

### X.3 Formatos de imagen y frecuencias de reloj de imagen

Para asegurar un alto nivel de calidad de interoperabilidad, los codificadores y los decodificadores que soportan un formato de imagen normalizado grande (QCIF, CIF, 4CIF, 16CIF) deben soportar todos los formatos de imagen normalizada más pequeños. Éste es un requisito para todos los decodificadores que se ajustan a los perfiles y niveles definidos en este anexo. (Como se especifica en otra parte de esta Recomendación, los decodificadores soportarán sub-QCIF y QCIF, y los codificadores deben soportar sub-QCIF o QCIF.) Por ejemplo, un decodificador que se ajusta a un perfil y nivel definidos en este anexo y que es capaz de decodificar imágenes 4CIF soportará también la decodificación de imágenes CIF.

Los decodificadores deben poder funcionar con un formato de imagen más pequeño a velocidades de imagen máximas no inferiores a la máxima velocidad de imagen a la que pueden funcionar con un formato de imagen normalizado más grande. Éste es un requisito para todos los decodificadores que se ajustan a los perfiles y niveles definidos en este anexo. Por ejemplo, un decodificador que se ajusta a un perfil y nivel definidos en este anexo y que puede decodificar imágenes 4CIF a 25 imágenes por segundo podrá también decodificar imágenes CIF, QCIF y SQCIF al menos a 25 imágenes por segundo.

Se recomienda que los codificadores y decodificadores que soportan formato de imagen personalizados y/o frecuencias de reloj de imagen personalizadas sigan las reglas definidas en este párrafo. Estas reglas deben ser respetadas por todos los decodificadores que se ajustan a los perfiles y niveles definidos en este anexo:

- 1) Un decodificador para cualquier perfil y nivel aquí definidos que soporta un formato de imagen máximo soportará todos los formatos de imagen normalizados cuyas dimensiones de alto y ancho sean menores o iguales que las del formato de imagen máximo soportado. Por ejemplo, un decodificador que soporta un formato de imagen personalizado de  $720 \times 288$  soportará la decodificación de imágenes CIF, QCIF y sub-QCIF.
- 2) Un decodificador para cualquier perfil y nivel aquí definidos que soporta formatos de imagen personalizados soportará todos los formatos de imagen normalizados o personalizados cuyas dimensiones de alto y ancho sean menores o iguales que las del formato de imagen máximo soportado.
- 3) Un decodificador para cualquier perfil y nivel aquí definidos que soporta un intervalo mínimo de imagen con la frecuencia de reloj de imagen normalizada de  $(30\ 000)/1001$  unidades por segundo, soportará el intervalo mínimo de imagen igual o más pequeño para todos los formatos de imagen soportados cuyas dimensiones de alto y ancho sean menores

o iguales que las del formato máximo de imagen al cual está especificado el intervalo mínimo de imagen.

- 4) Un decodificador para cualquier perfil y nivel aquí definidos que soporta un intervalo mínimo de imagen y frecuencias de reloj de imagen personalizadas, soportará la utilización de cualquier frecuencia de reloj de imagen con un intervalo de imagen igual o mayor para todos los formatos de imagen soportados cuyas dimensiones de alto y ancho sean menores o iguales que las del formato máximo de imagen al cual se especificó el intervalo mínimo de imagen.

#### X.4 Niveles de la capacidad de rendimiento

Se definen ocho niveles de calidad de funcionamiento para la implementación del decodificador. El decodificador de referencia hipotético tiene el tamaño mínimo especificado en el cuadro X.1 para todos los niveles de los perfiles 0 a 4. En los perfiles 5 a 8, el decodificador de referencia hipotético (HRD, *hypothetical reference decoder*) tiene un tamaño mayor y la selección de imagen de referencia mejorada está soportada con múltiples imágenes de referencia. El cuadro X.2 define los distintos parámetros de calidad de funcionamiento de cada uno de estos niveles:

- 1) **Nivel 10** – Soporte de la decodificación de resolución QCIF y sub-QCIF capaz de funcionar con una velocidad binaria de hasta 64 000 bits por segundo con una velocidad de decodificación de imagen de hasta (15 000)/1001 imágenes por segundo.
- 2) **Nivel 20** – Soporte de decodificación de resolución CIF, QCIF, y sub-QCIF, capaz de funcionar con una velocidad binaria de hasta  $2 \cdot (64\ 000) = 128\ 000$  bits por segundo con una velocidad de decodificación de imagen de hasta (15 000)/1001 imágenes por segundo para imágenes CIF y de (30 000)/1001 imágenes por segundo para imágenes QCIF y sub-QCIF.
- 3) **Nivel 30** – Soporte de decodificación con resolución CIF, QCIF, y sub-QCIF capaz de funcionar con una velocidad binaria de hasta  $6 \cdot (64\ 000) = 384\ 000$  bits por segundo con una velocidad de decodificación de imagen de hasta (30 000)/1001 imágenes por segundo.
- 4) **Nivel 40** – Soporte de decodificación con resolución CIF, QCIF, y sub-QCIF, capaz de funcionar con una velocidad binaria de hasta  $32 \cdot (64\ 000) = 2\ 048\ 000$  bits por segundo con una velocidad de decodificación de imagen de hasta (30 000)/1001 imágenes por segundo.
- 4.5) **Nivel 45** – Soporte de decodificación con resolución QCIF y sub-QCIF, capaz de funcionar con una velocidad binaria de hasta  $2 \cdot (64\ 000) = 128\ 000$  bits por segundo con una velocidad de decodificación de imagen de hasta (15 000)/1001 imágenes por segundo. Además, en perfiles diferentes de los perfiles 0 y 2, soporte de formatos de imagen personalizados del tamaño QCIF y más pequeños.
- 5) **Nivel 50** – Soporte de formatos de imagen personalizados y normalizados de tamaño CIF y más pequeños, capaces de funcionar con una velocidad binaria de hasta  $64 \cdot (64\ 000) = 4\ 096\ 000$  bits por segundo con una velocidad de decodificación de imagen de hasta 50 imágenes por segundo para CIF o formato de imagen más pequeños y de hasta (60 000)/1001 imágenes por segundo para formatos de imagen de  $352 \times 240$  y más pequeños.
- 6) **Nivel 60** – Soporte de formatos de imagen personalizados y normalizados de tamaño  $720 \times 288$  y más pequeños, capaces de funcionar con una velocidad binaria de hasta  $128 \cdot (64\ 000) = 8\ 192\ 000$  bits por segundo con una velocidad de decodificación de imagen de hasta 50 imágenes por segundo para formatos de imagen de  $720 \times 288$  o más pequeños y hasta (60 000)/1001 para formatos de imagen de  $720 \times 240$  y más pequeños.
- 7) **Nivel 70** – Soporte de formatos de imagen personalizados y normalizados de tamaño  $720 \times 576$  y más pequeños, capaces de funcionar con una velocidad binaria de hasta  $256 \cdot (64\ 000) = 16\ 384\ 000$  bits por segundo con una velocidad de decodificación de

imagen de hasta 50 imágenes por segundo para formatos de imagen de 720 × 576 o más pequeños y de hasta (60 000)/1001 imágenes por segundo para formatos de imagen de 720 × 480 y más pequeños.

La velocidad binaria a la cual se utilizan un perfil y nivel particulares en un sistema nunca excederá la especificada en este anexo. Sin embargo, determinados sistemas pueden incluir otros medios para señalar límites adicionales sobre la velocidad binaria. Otros aspectos de las capacidades de perfil y nivel pueden estar sujetos también a restricciones de capacidad adicionales cuando se utilizan en determinados sistemas, pero las capacidades requeridas para decodificar cualquier tren de bits para un perfil y nivel dados aquí definidos, nunca excederán los especificados en este anexo.

**Cuadro X.1/H.263 – Resumen de perfiles**

<b>Anexo/cláusula indicada más abajo para perfil indicado a la derecha</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>
<b>5.1.5:</b> Formato de imagen personalizado (CPFMT, custom picture format)	L	L	L	L	L	L	L	L	L
<b>5.1.7:</b> Código de frecuencia de reloj de imagen personalizada (CPCFC, custom picture clock frequency code)	L	L	L	L	L	L	L	L	L
<b>C:</b> Multipunto de presencia continua y multiplexor de vídeo									
<b>D.1:</b> Vectores de movimiento sobre demarcaciones de imagen		X	X	X	X	X	X	X	X
<b>D.2 con UUI = '1' o UUI no presente:</b> Extensión de la gama de vectores de movimiento						X	X	X	X
<b>D.2 con UUI = '01':</b> Extensión ilimitada de la gama de vectores de movimiento									
<b>E:</b> Codificación aritmética basada en sintaxis									
<b>F.2:</b> Cuatro vectores de movimiento por macrobloque		X	X	X	X	X	X	X	X
<b>F.3:</b> Compensación de movimiento para bloques superpuestos			X			X	X	X	X
<b>G:</b> Tramas PB									
<b>H:</b> Corrección de errores hacia adelante (la utilización puede ser impuesta a nivel de sistema como en la Rec. UIT T H.320)									
<b>I:</b> Codificación intra avanzada		X		X	X	X	X	X	X
<b>J:</b> Filtro de desbloqueo		X		X	X	X	X	X	X
<b>K sin submodos:</b> Codificación estructurada en rebanadas – Sin submodos				X	X		X		X
<b>K con ASO:</b> Codificación estructurada en rebanadas – Con submodo de orden arbitrario de las rebanadas							X		X
<b>K con RS:</b> Codificación estructurada en rebanadas – Con submodo rebanada rectangular									
<b>L.4:</b> Mejora suplementaria – Congelación de toda la imagen		X				X	X	X	X
<b>L:</b> Mejora suplementaria – Otras características SEI									
<b>M:</b> Tramas PB mejoradas									
<b>N:</b> Selección de la imagen de referencia (y submodos)									

**Cuadro X.1/H.263 – Resumen de perfiles**

Anexo/cláusula indicada más abajo para perfil indicado a la derecha	0	1	2	3	4	5	6	7	8
<b>O.1.1 Temporal (imágenes B):</b> Escalabilidad temporal, SNR, y espacial – Imágenes B para escalabilidad temporal									X
<b>O SNR y Espacial:</b> Escalabilidad temporal, SNR, y espacial – Imágenes EI y EP para escalabilidad SNR y espacial									
<b>P.5:</b> Remuestreo de imagen de referencia – Factor implícito de cuatro									X
<b>P:</b> Remuestreo de imagen de referencia – Remuestreo más general									
<b>Q:</b> Actualización de resolución reducida									
<b>R:</b> Decodificación de segmento independiente									
<b>S:</b> VLC inter alternativo									
<b>T:</b> Cuantificación modificada		X		X	X	X	X	X	X
<b>U sin submodos:</b> Selección de imagen de referencia mejorada – Sin submodos						X	X	X	X
<b>U con SPR:</b> Selección de imagen de referencia mejorada – Con submodo de supresión de subimagen									
<b>U con BTPSM:</b> Selección de imagen de referencia mejorada – Con submodo dos imágenes de imagen B									
<b>V:</b> Rebanadas particionadas para datos					X				
<b>W.6.3.8:</b> Especificación SEI adicional – Repetición del encabezamiento de la imagen previa					X				
<b>W.6.3.11:</b> Especificación SEI adicional – Indicaciones de campo entrelazadas								X	
<b>W:</b> Especificación SEI adicional – Otras características SEI									
"X" indica que el soporte de una característica es parte de un perfil.									
"L" indica que la inclusión de una característica depende del nivel dentro del perfil.									

**Cuadro X.2/H.263 – Niveles de operación**

<b>Parámetro indicado más abajo para el nivel indicado a la derecha</b>	<b>10</b>	<b>20</b>	<b>30</b>	<b>40</b>	<b>45</b>	<b>50</b>	<b>60</b>	<b>70</b>
<b>Formato máximo de imagen</b>	QCIF (176 × 144)	CIF (352 × 288)	CIF (352 × 288)	CIF (352 × 288)	QCIF (176 × 144) Soporte de CPFMT en perfiles distintos de 0 y 2	CIF (352 × 288) Soporte de CPFMT	CPFMT: 720 × 288 Soporte de CPFMT	CPFMT: 720 × 576 Soporte de CPFMT
<b>Intervalo mínimo de imagen</b>	2002/(30 000) s	2002/(30 000) s para CIF  1001/(30 000) s para QCIF y sub-QCIF	1001/(30 00) s	1001/(30 000) s	2002/(30 000) s  Soporte de CPCFC en perfiles distintos de 0 y 2	1/50 s en CIF o menor  1001/(60 000) s en 352 × 240 o menor  Soporte de CPCFC	1/50 s en 720 × 288 o menor  1001/(60 000) s en 720 × 240 o menor  Soporte de CPCFC	1/50 s en 720 × 576 o menor  1001/(60 000) s en 720 × 480 o menor  Soporte de CPCFC
<b>Velocidad binaria máxima en unidades de 64 000 bits/s</b>	1	2	6	32	2	64	128	256
<b>B de HRD máximo en unidades de 16 384 bits</b>	1: Perf. 5-8	2: Perf. 5-8	6: Perf. 5-8	32: Perf. 5-8	2: Perf. 5-8	64: Perf. 5-8	64: Perf. 5-8	256: Perf. 5-8

**Cuadro X.2/H.263 – Niveles de operación**

<b>Parámetro indicado más abajo para el nivel indicado a la derecha</b>	<b>10</b>	<b>20</b>	<b>30</b>	<b>40</b>	<b>45</b>	<b>50</b>	<b>60</b>	<b>70</b>
<b>BPPmaxKb máximo en unidades de 1024 bits</b>	128: Perf. 5-8	512: Perf. 5-8	512: Perf. 5-8	512: Perf. 5-8	128: Perf. 5-8	512: Perf. 5-8	1024: Perf. 5-8	1024: Perf. 5-8
<b>Imágenes de referencia ERPS máximas (anexo U)</b>	5: Perf. 5-7 10: Perf. 8	5: Perf. 5-7 10: Perf. 8 multiplicado por 2 para QCIF o sub-QCIF en Perf. 5-8	5: Perf. 5-7 10: Perf. 8 multiplicado por 2 para QCIF o sub-QCIF en Perf. 5-8	5: Perf. 5-7 10: Perf. 8 multiplicado por 2 para QCIF o sub-QCIF en Perf. 5-8	5: Perf. 5-7 10: Perf. 8	5: Perf. 5-7 10: Perf. 8 multiplicado por 2 para QCIF o menor en Perf. 5-8	5: Perf. 5-7 10: Perf. 8 multiplicado por 2 para CIF o menor, y por 4 para QCIF o menor en Perf. 5-8	5: Perf. 5-7 10: Perf. 8 multiplicado por 2 para CIF o menor, y por 4 para QCIF o menor en Perf. 5-8
<p>NOTA 1 – En los perfiles en los cuales no se especifica en el cuadro X.2 un número máximo de memorias tampón de imagen de referencia, no se requiere el soporte de múltiples memorias tampón de imagen de referencia.</p> <p>NOTA 2 – En los perfiles para los cuales no se especifican en el cuadro X.2 un BPPmaxKb y un HRD B máximos, se aplica el valor mínimo especificado en el cuadro X.1 para la velocidad binaria y la resolución máximas especificadas.</p>								



## X.5 Definiciones de capacidad genérica para uso con la Rec. UIT-T H.245

El cuadro X.3 define un identificador de capacidad para establecer las capacidades H.263 y para uso en sistemas que utilizan la Rec. UIT-T H.245 para la determinación de capacidad. Estos parámetros solamente serán incluidos como **genericVideoCapability (capacidad vídeo genérico)** dentro de la estructura **VideoCapability (capacidad vídeo)** y como **genericVideoMode (modo vídeo genérico)** dentro de la estructura **VideoMode (modo vídeo)** de la Rec. UIT-T H.245. Los cuadros X.4 a X.14 definen los parámetros de capacidad asociados.

Cuando se incluyen los parámetros en la señalización de canal lógico o en la petición de modo, estará presente exactamente un parámetro con valor de identificador de parámetro en la gama de cero a ocho, esto es, solamente se especificará un perfil.

**Cuadro X.3/H.263 – Identificador de capacidad para capacidad H.263**

Nombre de la capacidad:	H.263
Clase de capacidad:	Códec de vídeo
Tipo de identificador de capacidad:	Normalizado
Valor de identificador de capacidad:	itu-t (0) recommendation (0) h (8) 263 generic-capabilities (1) 0
MaxBitRate:	El campo maxBitRate se incluirá siempre.
nonCollapsingRaw:	No se incluirá este campo.
Transporte:	No se incluirá este campo.

**Cuadro X.4/H.263 – Capacidad del perfil línea de base (perfil 0)**

Nombre del parámetro:	baselineProfile
Descripción del parámetro:	Éste es un GenericParameter desplomable. baselineProfile indica el nivel máximo de soporte para el perfil línea de base cuando está presente en el intercambio de capacidad, el nivel máximo de transmisión cuando está presente en la señalización de canal lógico, y el nivel deseado cuando está presente en la petición de modo.
Valor del identificador de parámetro:	0
Status del parámetro:	Obligatorio
Tipo de parámetro:	unsignedMin
Reemplaza:	–

**Cuadro X.5/H.263 – Capacidad Perfil de compatibilidad hacia atrás Versión 2  
Eficiencia de codificación H.320 (perfil 1)**

Nombre del parámetro:	h320Profile
Descripción del parámetro:	Éste es un GenericParameter desplomable. h320Profile indica el nivel máximo de soporte para el perfil de compatibilidad hacia atrás Versión 2 eficiencia de codificación H.320 cuando está presente en el intercambio de capacidad, el nivel máximo de transmisión cuando está presente en la señalización de canal lógico, y el nivel deseado cuando está presente en la petición de modo.
Valor del identificador de parámetro:	1
Status del parámetro:	Facultativo
Tipo de parámetro:	unsignedMin
Reemplaza:	–

**Cuadro X.6/H.263 – Capacidad Perfil de compatibilidad hacia atrás Versión 1 (perfil 2)**

Nombre del parámetro:	backwardCompatibleProfile
Descripción del parámetro:	Éste es un GenericParameter desplomable. backwardCompatibleProfile indica el nivel máximo de soporte para el perfil de compatibilidad hacia atrás Versión 1 cuando está presente en el intercambio de capacidad, el nivel máximo de transmisión cuando está presente en la señalización de canal lógico, y el nivel deseado cuando está presente en la petición de modo.
Valor del identificador de parámetro:	2
Status del parámetro:	Facultativo
Tipo de parámetro:	unsignedMin
Reemplaza:	–

**Cuadro X.7/H.263 – Capacidad Perfil inalámbrico interactivo y de  
flujo continuo Versión 2 (perfil 3)**

Nombre del parámetro:	v2WirelessProfile
Descripción del parámetro:	Éste es un GenericParameter desplomable. v2WirelessProfile indica el nivel máximo de soporte para el perfil inalámbrico interactivo y de flujo continuo Versión 2 cuando está presente en el intercambio de capacidad, el nivel máximo de transmisión cuando está presente en la señalización de canal lógico, y el nivel deseado cuando está presente en la petición de modo.
Valor del identificador de parámetro:	3
Status del parámetro:	Facultativo
Tipo de parámetro:	unsignedMin
Reemplaza:	–

**Cuadro X.8/H.263 – Capacidad Perfil inalámbrico interactivo y de flujo continuo versión 3 (perfil 4)**

Nombre del parámetro:	v3WirelessProfile
Descripción del parámetro:	Éste es un GenericParameter desplomable. v3WirelessProfile indica el nivel máximo de soporte para el perfil inalámbrico interactivo y de flujo continuo Versión 3 cuando está presente en el intercambio de capacidad, el nivel máximo de transmisión cuando está presente en la señalización de canal lógico, y el nivel deseado cuando está presente en la petición de modo.
Valor del identificador de parámetro:	4
Status de parámetro:	Facultativo
Tipo de parámetro:	unsignedMin
Reemplaza:	–

**Cuadro X.9/H.263 – Capacidad Perfil de alta compresión conversacional (perfil 5)**

Nombre del parámetro:	conversationalProfile
Descripción del parámetro:	Éste es un GenericParameter desplomable. conversationalProfile indica el nivel máximo de soporte para el perfil de alta compresión conversacional cuando está presente en el intercambio de capacidad, el nivel máximo de transmisión cuando está presente en la señalización de canal lógico, y el nivel deseado cuando está presente en la petición de modo.
Valor del identificador de parámetro:	5
Status de parámetro:	Facultativo
Tipo de parámetro:	unsignedMin
Reemplaza:	–

**Cuadro X.10/H.263 – Capacidad Perfil Internet conversacional (perfil 6)**

Nombre del parámetro:	conversationalInternetProfile
Descripción del parámetro:	Éste es un GenericParameter desplomable. conversationalInternetProfile indica el nivel máximo de soporte para el perfil Internet conversacional cuando está presente en el intercambio de capacidades, el nivel máximo de transmisión cuando está presente en la señalización de canal lógico, y el nivel deseado cuando está presente en la petición de modo.
Valor del identificador de parámetro:	6
Status de parámetro:	Facultativo
Tipo de parámetro:	unsignedMin
Reemplaza:	–

**Cuadro X.11/H.263 – Capacidad Perfil entrelazado plus conversacional (perfil 7)**

Nombre del parámetro:	conversationalInterlaceProfile
Descripción del parámetro:	Éste es un GenericParameter desplomable. conversationalInterlaceProfile indica el nivel máximo de soporte para el perfil entrelazado plus conversacional cuando está presente en el intercambio de capacidades, el nivel máximo de transmisión cuando está presente en la señalización de canal lógico y el nivel deseado cuando está presente en la petición de modo.
Valor del identificador de parámetro:	7
Status de parámetro:	Facultativo
Tipo de parámetro:	unsignedMin
Reemplaza:	–

**Cuadro X.12/H.263 – Capacidad Perfil de alta latencia (perfil 8)**

Nombre del parámetro:	highLatencyProfile
Descripción del parámetro:	Éste es un GenericParameter desplomable. highLatencyProfile indica el nivel máximo de soporte para el perfil de alta latencia cuando está presente en el intercambio de capacidades, el nivel máximo de transmisión cuando está presente en la señalización de canal lógico, y el nivel deseado cuando está presente en la petición de modo.
Valor del identificador de parámetro:	8
Status de parámetro:	Facultativo
Tipo de parámetro:	unsignedMin
Reemplaza:	–

**Cuadro X.13/H.263 – Capacidad Compromiso temporal espacial**

Nombre del parámetro:	temporalSpatialTradeOffCapability
Descripción de parámetro:	Éste es un GenericParameter desplomable. La presencia de este parámetro indica que el codificador tiene capacidad para variar su compromiso entre resolución temporal y espacial según lo ordene el terminal distante. No tiene significado cuando es parte de una capacidad de recepción.
Valor del identificador de parámetro:	9
Status de parámetro:	Facultativo
Tipo de parámetro:	lógico
Reemplaza:	–

**Cuadro X.14/H.263 – Capacidad Macrobloques malos de vídeo**

Nombre del parámetro:	videoBadMBsCap
Descripción del parámetro:	Éste es un GenericParameter desplomable. La presencia de este parámetro indica la capacidad de un codificador para recibir o de un decodificador para transmitir la instrucción videoBadMBs. Cuando es parte de una capacidad de transmisión, indica la aptitud del codificador para procesar instrucciones videoBadMBs y para ejecutar acciones correctivas apropiadas para la recuperación de la calidad de vídeo. Cuando es parte de una capacidad de recepción, indica la aptitud del decodificador para enviar indicaciones videoBadMBs apropiadas.
Valor del identificador de parámetro:	10
Status de parámetro:	Facultativo
Tipo de parámetro:	lógico
Reemplaza:	–

## Apéndice I

### Seguimiento de errores

#### I.1 Introducción

Este apéndice describe un método de recuperación eficaz tras los errores de transmisión si se informa de la existencia de macrobloques erróneos que pasan a través de un canal de realimentación al codificador. La capacidad de enviar y procesar información de realimentación se señala por medios externos (por ejemplo, mediante la Rec. UIT-T H.245). Además, se definen externamente el formato y el contenido del mensaje de realimentación (por ejemplo, mediante la Rec. UIT-T H.245).

#### I.2 Seguimientos de errores

Como la codificación INTRA detiene la propagación temporal de errores, debe utilizarse para los macrobloques que resultan seriamente afectados por los errores de transmisión. Ello requiere que la localización y la extensión de los efectos secundarios de imagen puedan pasarse al codificador. El algoritmo indicado a continuación ofrece una distribución del error estimada que se basa en la información de realimentación recibida por el codificador. Considera la propagación espacial del error causada por la predicción compensada del movimiento, así como el retardo, hasta la recepción del mensaje de realimentación. El algoritmo ilustra un enfoque posible para evaluar los mensajes de realimentación destinados al seguimiento de errores espacial-temporal. También son posibles otros algoritmos.

Se supone que hay  $N$  macrobloques en cada trama denominados  $mb = 1..N$  desde la esquina superior izquierda a la inferior derecha.  $\{n_{err}, mb_{first}, mb_{last}\}$  es el mensaje de realimentación al codificador, en donde  $mb_{first} \leq mb \leq mb_{last}$  indica un conjunto de macrobloques erróneos en la trama  $n_{err}$ .

Para evaluar el mensaje de realimentación, el codificador debe registrar continuamente información durante la codificación de cada trama. En primer lugar, ha de almacenarse el error inicial,  $E_0(mb, n)$  que se producirá por la pérdida del macrobloque  $mb$  de la trama  $n$ . Suponiendo una ocultación de errores simple en la que los macrobloques erróneos no se tratan como si estuvieran codificados,

$E_0(mb, n)$  se calcula como la diferencia aditiva absoluta (SAD, *summed absolute difference*) del macrobloque  $mb$  en la trama  $n$  y  $n - 1$ . En segundo lugar, se almacena el número de píxels transformados del macrobloque  $mb_{source}$  de la trama  $n - 1$  al macrobloque  $mb_{dest}$  en la trama  $n$  en dependencias  $d(mb_{source}, mb_{dest}, n)$ . Estas dependencias se derivan de los vectores de movimiento.

Se supone que un mensaje de realimentación llega antes de que se codifique la trama  $n_{next}$ , de forma que  $n_{next} > n_{err}$ . El error estimado  $E(mb, n_{err})$  en el macrobloque  $n_{err}$  se inicializa de la siguiente manera:

$$E(mb, n_{err}) = \begin{cases} E_0 (mb, n_{err}) & \text{para } mb_{first} \leq mb \leq mb_{last} \\ 0 & \text{en caso contrario} \end{cases}$$

Para las tramas  $n$  posteriores, siendo  $n_{err} < n < n_{next}$ , el error puede estimarse de la siguiente manera:

$$E(mb, n) = \sum_{i=1}^N E(i, n-1) \frac{d(i, mb, n)}{256}$$

en donde se supone que los errores están distribuidos uniformemente en cada macrobloque después de cada repetición.

El error estimado  $E(mb, n_{next} - 1)$  se incorpora en la decisión del modo de la trama siguiente. Por ejemplo, el macrobloque  $mb$  se codifica en modo INTRA, si  $E(mb, n_{next} - 1)$  excede de un cierto umbral.

En la práctica, sólo se almacenará la información de seguimiento de errores correspondiente a las últimas  $M$  tramas. Siendo así, si  $n_{err} < n_{next} - M$ , no se dispone de información de seguimiento de errores y el codificador puede adoptar medidas especiales. Por ejemplo, la trama siguiente puede codificarse en modo INTRA. No obstante, es posible aplicar otros procedimientos que pueden ser más eficaces.

## Apéndice II

### Mejora opcional recomendada

El contenido del apéndice II a la Rec. UIT-T H.263 aprobado en febrero de 1998 se ha vuelto obsoleto después de la aprobación del anexo X a dicha Recomendación. Teniendo en cuenta que este apéndice se utiliza como referencia por muchos usuarios de textos del UIT-T, el objetivo principal de su inclusión es servir de referencia para los usuarios que no tienen conocimiento del contenido del anexo X a la Rec. UIT-T H.263.



## SERIES DE RECOMENDACIONES DEL UIT-T

Serie A	Organización del trabajo del UIT-T
Serie D	Principios generales de tarificación
Serie E	Explotación general de la red, servicio telefónico, explotación del servicio y factores humanos
Serie F	Servicios de telecomunicación no telefónicos
Serie G	Sistemas y medios de transmisión, sistemas y redes digitales
<b>Serie H</b>	<b>Sistemas audiovisuales y multimedios</b>
Serie I	Red digital de servicios integrados
Serie J	Redes de cable y transmisión de programas radiofónicos y televisivos, y de otras señales multimedios
Serie K	Protección contra las interferencias
Serie L	Construcción, instalación y protección de los cables y otros elementos de planta exterior
Serie M	Gestión de las telecomunicaciones, incluida la RGT y el mantenimiento de redes
Serie N	Mantenimiento: circuitos internacionales para transmisiones radiofónicas y de televisión
Serie O	Especificaciones de los aparatos de medida
Serie P	Calidad de transmisión telefónica, instalaciones telefónicas y redes locales
Serie Q	Conmutación y señalización
Serie R	Transmisión telegráfica
Serie S	Equipos terminales para servicios de telegrafía
Serie T	Terminales para servicios de telemática
Serie U	Conmutación telegráfica
Serie V	Comunicación de datos por la red telefónica
Serie X	Redes de datos, comunicaciones de sistemas abiertos y seguridad
Serie Y	Infraestructura mundial de la información, aspectos del protocolo Internet y Redes de la próxima generación
Serie Z	Lenguajes y aspectos generales de soporte lógico para sistemas de telecomunicación