国　　际　　电　　信　　联　　盟

ITU-R

国际电联无线电通信部门

ITU-R BT.1907-0 建议书

(01/2012)

在有完整参考信号情况下采用HDTV的广播应用的客观感知视频质量测量技术

BT 系列

广播业务

(电视)

国际电信联盟

前言

无线电通信部门的职责是确保卫星业务等所有无线电通信业务合理、平等、有效、经济地使用无线电频谱，不受频率范围限制地开展研究并在此基础上通过建议书。

无线电通信部门的规则和政策职能由世界或区域无线电通信大会以及无线电通信全会在研究组的支持下履行。

知识产权政策（IPR）

ITU-R的IPR政策述于ITU-R第1号决议中所参引的《ITU-T/ITU-R/ISO/IEC的通用专利政策》。专利持有人用于提交专利声明和许可声明的表格可从http://www.itu.int/ITU-R/go/patents/en获得，在此处也可获取《ITU-T/ITU-R/ISO/IEC的通用专利政策实施指南》和ITU-R专利信息数据库。

ITU-R系列建议书

（也可在线查询http://www.itu.int/publ/R-REC/en）

系列	标题
BO	卫星传送
BR	用于制作、存档和播出的录制；电视电影
BS	广播业务（声音）
BT	**广播业务（电视）**
F	固定业务
M	移动、无线电定位、业余和相关卫星业务
P	无线电波传播
RA	射电天文
RS	遥感系统
S	卫星固定业务
SA	空间应用和气象
SF	卫星固定业务和固定业务系统间的频率共用和协调
SM	频谱管理
SNG	卫星新闻采集
TF	时间信号和频率标准发射
V	词汇和相关问题

说明： 该ITU-R建议书的英文版本根据ITU-R第1号决议详述的程序予以批准。

电子出版
2020年，日内瓦

ITU-R BT.1907-0 建议书*

在有完整参考信号的情况下采用HDTV的广播应用的客观感知视频质量测量技术

（2012年）

范围

本建议书规定了在有完整的参考信号可以利用的情况下，评估采用HDTV的广播应用的感知视频质量的方法。

国际电联无线电通信全会，

考虑到

a) 自动测量广播视频质量的能力长期以来被认为是该行业的一个重要优势；

b) ITU-R BT.1683建议书描述了在有完整参考信号的情况下，测量标准清晰度数字广播电视的感知视频质量的客观方法；

c) ITU-R BT.709建议书描述了适用于制作和国际节目交流的HDTV标准的参数值，ITU-R BT.500建议书描述了高清晰度电视中图像质量的主观评价方法；

d) HDTV正在广播领域得到广泛地应用；

e) ITU-T第9研究组已经根据VQEG发布的HDTV报告的结果，制定了ITU-T J.341建议书，该建议书规定了在有完整参考信号的情况下HDTV的客观感知视频质量测量方法；

f) HDTV感知视频质量的客观测量可以作为主观评价方法的补充，

建议

在附件1、2和3中给出的客观视频质量模型应该用于在有完整参考信号的情况下采用HDTV的广播应用的感知视频质量的客观测量。

* 无线电通信第6研究组于2020年2月根据ITU-R第1号决议对此建议书进行了编辑性修正。

附件1

1 引言

本建议书提供了在能够采用完整参考信号（FR）方法的情况下，适用于高清晰度电视（HDTV）非交互式应用的感知视频质量测量方法，提出了用于评估主观质量得分的模型。

当在测量点可以容易地获得无损的参考视频信号时，例如在实验室或者在类似电视广播站的封闭环境内测量单台设备或者系列设备这种情况，就能采用完整参考信号测量方法。评估方法既包括校准，又包括客观视频质量评估。

验证测试的素材既包括H.264和MPEG-2编码劣化，又包括各种传输差错情况（例如比特错、丢失数据包）。本建议书中的模型可以用于监视已部署网络的质量以确保它们运行准备就绪。劣化的视觉效果可能包括空间以及时间的劣化，本建议书中的模型也能用于视频系统的实验室测试，当该模型用于比较不同的视频系统时，对于这种特殊情形，建议采用一种定量方法（例如ITU-T J.149建议书中描述的方法）来确定模型的准确度。

本建议书被认为适用于传输速率在1 Mbit/s到30 Mbit/s之间的广播业务，验证测试时设想采用下列分辨率和帧频：

- 1080/59.94/I
- 1080/25/P
- 1080/50/I
- 1080/29.97/P

对于每一种分辨率，进行验证测试时要考虑到以下条件：

测试因素
视频分辨率：1920×1080隔行和逐行
视频帧频：每秒29.97帧和每秒25帧
视频比特率：1到30 Mbit/s
最长2秒的短暂定格（带跳动的暂停）
丢失数据的传输错误
从1080到720/P的SRC转换，压缩、传输、解压缩，再转换回到1080。
编码技术
H.264/AVC （MPEG-4 第10部分）
MPEG-2

注意到在验证测试计划中720/P被当作是测试条件（HRC）的一部分。由于目前720/P通常被提升为显示的一部分，因此认为720/P HRC会更适合处理这种格式。

1.1 应用

适合于本建议书描述评估模型的应用包括但不限于：

1) 在信源处可能实时的、在工作中进行的质量监视；

2) 在测量点可以获得信源的情况下进行的远程目的地质量监视；

3) 适用于采用视频压缩和解压缩技术的存储或传输系统监测的质量测量，这些系统可以单向地采用这些技术，或采用这些技术的级联；

4) 视频系统的实验室测试。

1.2 限制

本建议书中描述的视频质量评估模型不能用于取代主观测试，两个精心设计和实施的主观测试（即在两个不同的实验室）之间的相关值通常会在0.95到0.98的范围内，如果利用本建议书进行视频系统的对比（例如比较两个编解码器），建议针对这种特殊情形采用一种定量的方法确定模型的准确度。

当出现帧定格时，测试条件通常是帧定格的持续时间小于2秒。本建议书中的模型不适用于重新缓存条件下（即时延或者无跳动定格总是递增的视频）视频质量的测量，该模型没有在除TV系统所使用那些帧频（即每秒29.97帧和每秒25帧，采用隔行或者逐行模式）以外的帧频上被测试过。

应该注意到如果新的编码和传输技术产生的人为现象没有包含在这次评估中，客观模型可能会产生错误的结果，这时就需要主观评价。

2 参考文献

无。

3 定义

3.1 在其他地方规定的术语

本建议书使用了下列在其他地方定义的术语：

3.1.1 主观评价（图像）：确定在观看期间呈现给评估小组的节目类图像的质量或者损伤。

3.1.2 客观感知测量（图像）：利用节目类图像和客观（采用仪器的）测量方法测量系列节目的性能，从而获得一个与主观评价测试得到的评定值相接近的指示。

3.1.3 支持者：在某一个国际电联建议书中提出适用于验证测试的视频质量模型以及可能包含的组织或者公司。

3.2 本建议书规定的术语

本建议书规定了下列术语：

3.2.1 帧频：每秒不同帧的数量（即全部帧–重复的帧）。

3.2.2 模拟传输错误：在高度受控的环境下发生在数字视频比特流中的错误，错误举例包括模拟包丢失率和模拟比特错误，明确规定了用于控制模拟传输错误的参数。

3.2.3 传输错误：发生在视频传输中的任何错误，错误类型举例包括模拟传输错误和实际网络环境。

4 缩写和首字母缩略语

本建议书采用了下列缩写和首字母缩略语：

ACR	绝对等级评定值（见ITU-R BT.500建议书）
ACR-HR	具有隐藏参考的绝对等级评定值（见ITU-T P.910建议书）
AVI	音频视频交错
DMOS	差分平均评价得分
FR	完整参考信号
FRTV	完整参考电视
HRC	假设基准电路
ILG	VQEG的独立实验组
MOS	平均评价得分
MOSp	预测的平均评价得分
NR	无（或者零）参考
PSNR	峰值信噪比
PVS	处理过的视频序列
RMSE	均方根误差
RR	减弱的参考信号
SFR	信源帧频
SRC	信源参考信道或电路
VQEG	视频质量专家组
YUV	颜色空间和文件格式

5 惯例

无。

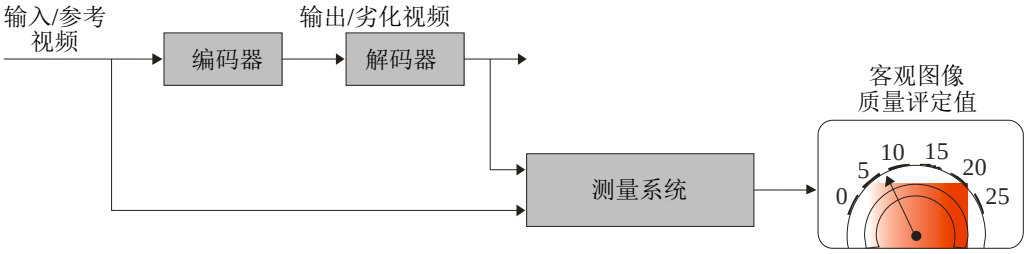
6 完整参考信号方法描述

适用于客观测量感知视频质量的、具有完整参考信号的双端测量方法，通过比较系统输入端的无失真输入或参考信号、视频信号和系统输出端的劣化信号来评估系统的性能（图1）。

图1所示的是在实验室测试编解码器的完整参考信号方法的应用举例。

图 1

在实验室测试编解码器的完整参考感知质量测量方法应用



比较输入信号和输出信号可能需要时间对准或者空间对准过程，后者用于补偿一些水平或垂直方向上的图像位移或者裁剪，可能还需要校正一些偏移量或者亮度和色度通道的增益差异，然后典型地应用一种人类视觉的感知模型来计算客观图像质量评定值。

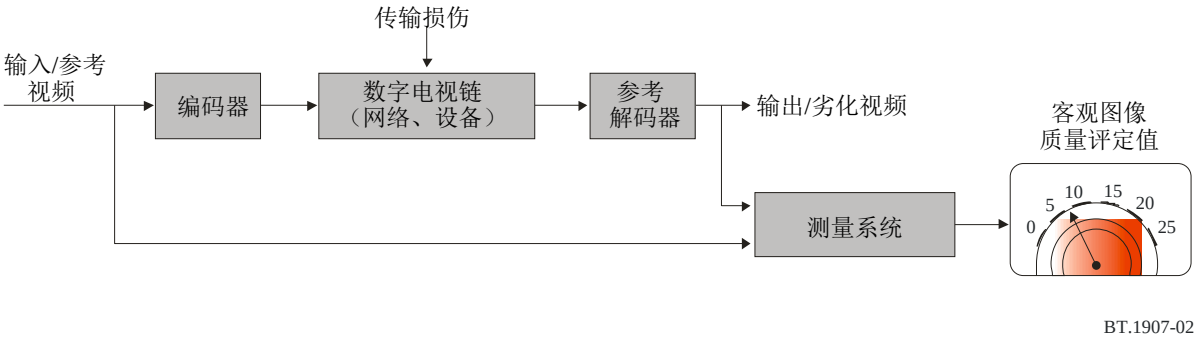
对准和增益调整被称作配准，之所以需要这个过程是因为绝大多数完整参考信号方法实际上是以逐像素的方式比较参考的和处理过的图像，附件2中描述的视频质量度量包含了配准的方法。

由于视频质量度量典型地是以近似人类视觉的响应为基础，而不是以测量特定编码的人为现象为基础，它们基本上对于模拟系统和数字系统同样有效，也基本上对于模拟和数字系统混合链，或者数字压缩系统串联链有效。

图2所示的是完整参考信号方法测试传输链的应用举例。

图 2

完整参考信号感知质量测量方法测试传输链的应用



在这种情况下，可以从传输链上不同的点输入到参考解码器，例如，解码器可以位于网络中的某个点，如图2所示，或者直接在编码器的输出端，如图1所示。如果数字传输链是透明的，信源处的客观图像质量评定值测量等同于在链上任何后续点位上的测量。

通常认为完整参考信号方法为感觉图像质量测量提供了最高的准确度，已经证实该方法可能与按照ITU-T P.910建议书规定的ACE-HR方法进行的主观评价具有很高的相关性。

7 视频质量专家组（VQEG）的成果

一个被称作视频质量专家组（VQEG）的非正式团体在进行感知视频质量测量的研究工作，该团体向ITU-T 第9、12研究组和ITU-R第6研究组报告情况。VQEG最近完成的高清晰度电视阶段I测试评估了提议的完整参考信号感知视频质量测量算法的性能。

下列统计数字来自最终的VQEG HDTV报告，注意到VQEG HDTV报告的主体包含了其他的度量，包括：以独立实验为基础计算得到的皮尔森相关性和RMSE，置信区间，基于独立实验的统计显著性测试，分析包含特定损伤（例如只有H.264编码）在内的数据子集，散布图，以及适合的系数。

主要分析

FR模型的性能总结见表1。按照ITU-T J.340建议书计算PSNR，并将PSNR纳入到用于对比目的的这个分析中，“超集RMSE”表示以汇总超集（即映射到一个范围内的全部6次实验）为基础计算得到的基本度量（RMSE），“最佳表现组的总数”表示其模型是最佳表现的模型或者在统计上相当于最佳表现模型的实验的数量（0到6），“优于PSNR的总数”表示其模型在统计上优于PSNR的实验的数量（0到6），“优于超集PSNR”显示各个模型在统计上是否优于汇总超集上的PSNR，“超集相关性”表示以汇总超集为基础计算得到的皮尔森相关性。

表1

度量	PSNR	SwissQual
超集RMSE	0.71	0.56
最佳表现组的总数	1	5
优于PSNR的总数	–	4
优于超集PSNR	–	是
超集相关性	0.78	0.87

附件2

模型描述

编者注：纳入到本节的源代码构成了本建议书的强制性部分，在 http://ifatemp.itu.int/t/2009/sg9/exchange/q2/可获得该源代码。

模型概述

该模型预测了视频质量，如同试验者在实验中做出的预测一样。预测模型采用心理视觉和感知激励建模来模拟主观预测。

作为一种完整参考信号方法，该模型比较了被测试的输入或者高质量参考视频和相关的劣化视频序列，这个过程如图3所示。

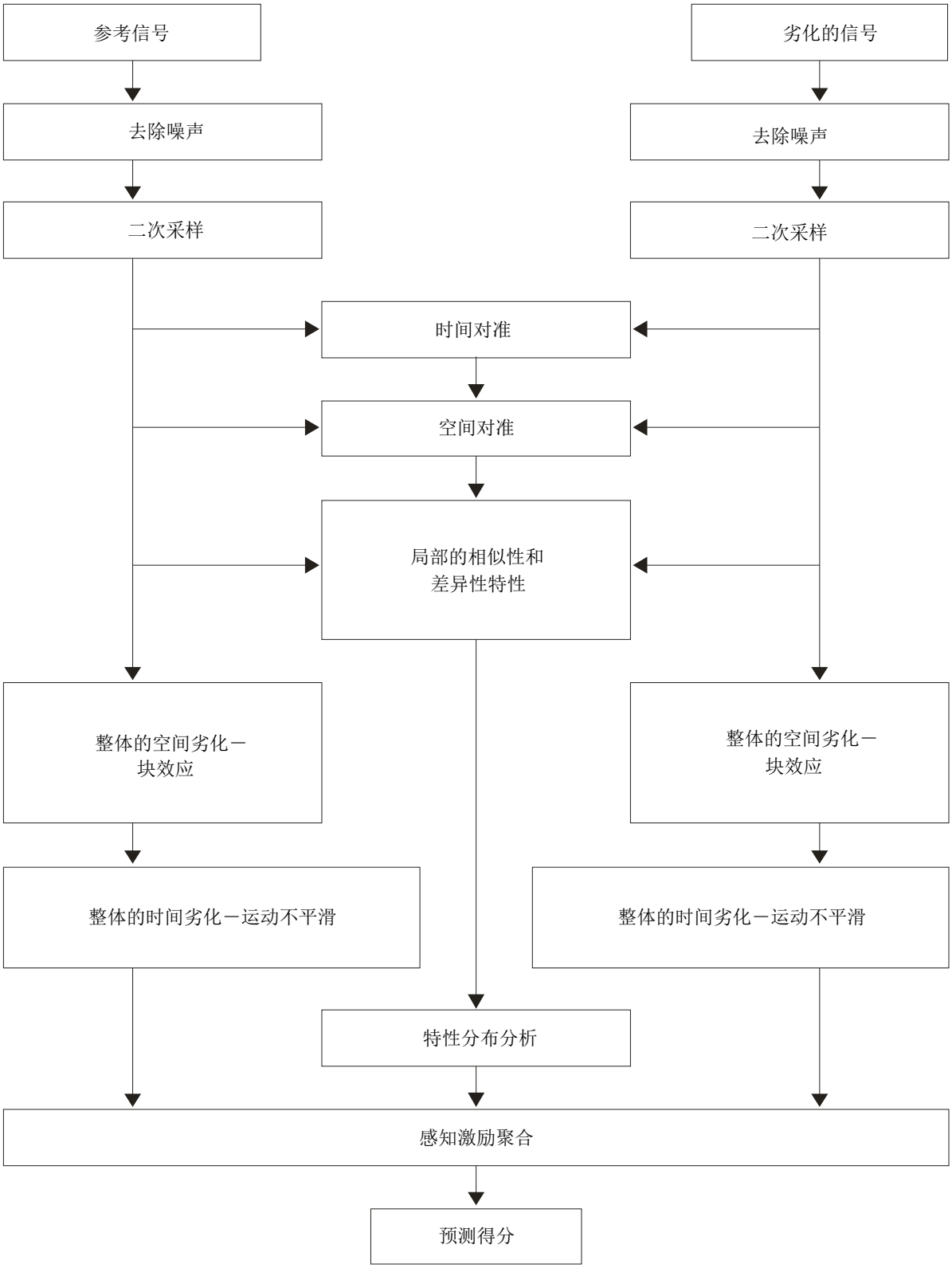
得分评估根据以下步骤：

1) 首先，对视频序列进行预处理。特别地，通过对帧进行滤波和对帧进行二次采样去除噪声。
2) 进行参考的和处理过的视频序列之间时间上的帧对准。
3) 进行处理过的视频帧和相应的参考视频帧之间空间上的帧对准。
4) 计算局部的空间质量特性：视觉激励的局部相似性和局部差异性测量。
5) 进行局部相似性和差异性特性分布分析。
6) 采用*块效应*特性测量整体的空间劣化。
7) 采用*运动不平滑*特性测量整体的时间劣化，通过评估局部和整体运动强度和帧显示时间来计算运动不平滑量。
8) 基于上述特性的非线性聚合来评估质量得分。
9) 在参考的和处理过的视频序列之间存在着相对较大的空间偏移的情况下，为了避免错误预测，要对3个不同的在水平和垂直方向上空间对齐的视频序列进行上述步骤的计算，各空间位置的最大预测得分就是最终的评估质量得分。

第2.1节到第2.9节对各个步骤进行了更为详细的解释，第2.10节包含了一个嵌入式文档以及涵盖了与模型描述相符的实现所必需部分和函数的C++源代码，在第2.1节到第2.9节中提到的C++函数名称可查阅该参考源代码（例如第2.2节提到的`CFrameAnalysisFullRef::ContentTimeAlignment`）。

图 3

模型处理步骤的流程图概述。在上面，输入是参考信号和劣化的（或处理过的）视频序列。不同的处理步骤产生主要的模型输出—底部的预测得分



BT.1907-03

2.1 预处理

对参考的和处理过的视频序列中的每一帧进行空间上的低通滤波，并且二次采样到3种不同的分辨率R1、R2、R3：

	原始帧		R1		R2		R3
高×宽	1080 × 1920	→	540 × 960	→	270 × 480	→	96 × 128

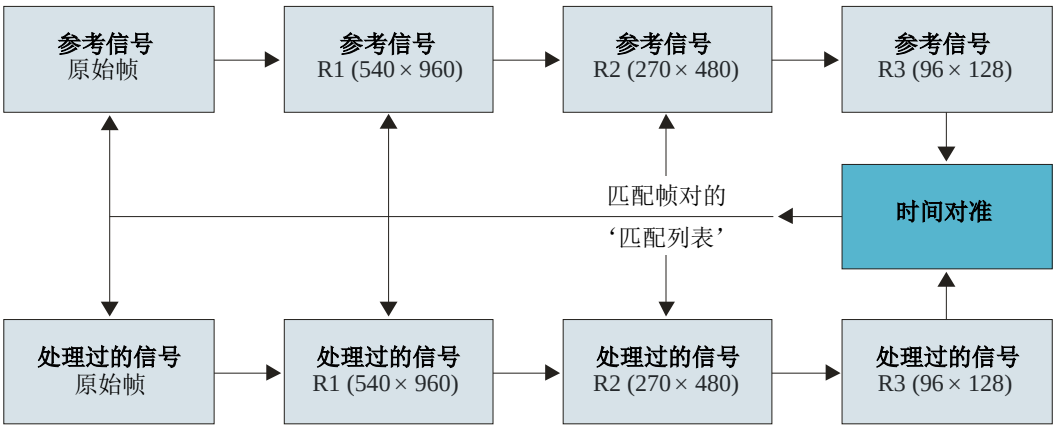
以分辨率R3生成帧见方法`CFrameAnalysisFullRef::ContentTimeAlignment`，以分辨率R1和R2生产帧见方法`CFrameSeq::ReadFrame`。

注意到由于内存的限制实现并不是直接的。

图4所示的是3种二次采样的分辨率。

图 4

参考的和处理过的视频序列中的帧经过低通滤波，然后二次采样到3种不同的分辨率。最低的分辨率R3用于实行帧时间对准，最后得到的已匹配帧列表能用于匹配采用任何分辨率的帧



BT.1907-04

2.2 时间对准

使用低分辨率R3的参考视频序列和处理过的视频序列进行时间对准。

时间对准采取以下面递归的方式进行：

1) 在参考序列中寻找一个‘固定’帧（`Ref_anchor`）。

2) 将这一帧与劣化序列中最匹配的帧进行匹配（`Deg_best_match`）。

从劣化序列中取出这最匹配的帧（Deg_best_match），将它与参考信号‘固定’帧（Ref_anchor）附近的帧进行匹配。根据相似性判据，从Deg_best_match和Ref_anchor周围帧之间找到一个更匹配的帧，将它存为最匹配的帧对。至于处理过的帧x和参考帧y的Y平面之间的相似性判据，采用函数：

```
sim = exp(-mean_square_diff(a*x+b,y))
```
(2.1)

以及选择的参数a、b，使得处理过的帧x和参考帧y的Y平面数值之间的均方误差最小，见参考实现中的方法`FrameSimilarity::similarity`。

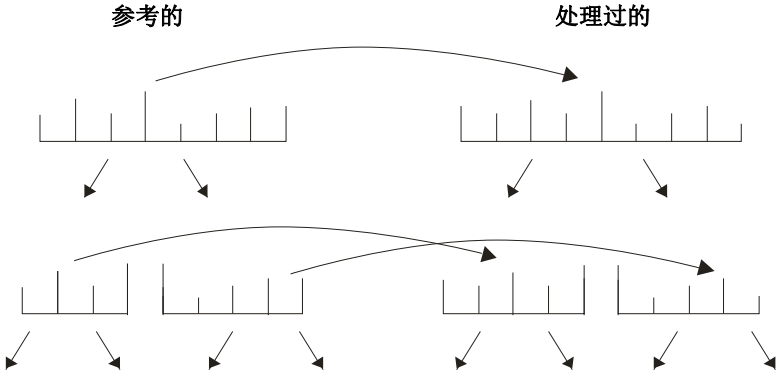
1) 如果已匹配的帧对是完全匹配（相似性判据已经超过了接受门限），在匹配帧之前和匹配帧之后，将匹配帧对中的参考视频序列和处理过的视频序列，各自分成两个视频序列，参考序列对和劣化序列对均从1）开始匹配。

2) 如果匹配帧对不是‘完全匹配’，将使用参考视频中一个不同的‘固定’帧重新从1）开始匹配。由于关于“完全”匹配帧的预测值没有*先验*知识，匹配门限会反复地下降。根据大量的训练数据样本确定以下数值：根据相似性判据公式（2.1），起始门限为0.98。当与10个固定帧匹配失败以后，门限会以因子0.98降低，重新从1）开始匹配。这样，最多会进一步尝试10个固定帧，如果它们都失败了，门限会再一次降低，直至降到最小值0.1，全部实现细节见

`SQ_TimeAlignement::findAncorAndDescend`。

图 5

时间对准递归方法图解，将参考信号的一个固定帧与处理过的序列中的某一帧进行匹配，然后将这两个序列拆分，在每个序列中选择一个固定帧并进行匹配



BT.1907-05

时间对准的结果是一个序列（本质上是一个‘匹配列表’），将处理过的视频序列中的每一帧分配给参考视频序列中的某一帧，或者一个表示没有发现匹配得足够好的帧的指示符。这样，对于随后的处理阶段，处理过的视频序列中每一个已匹配帧都有一个相应的参考视频帧。处理过的视频序列中的那些具有“不匹配”指示符的帧，将要与处理过的视频序列中之前的和之后的‘已匹配’帧的两个相匹配的参考帧进行比较，注意到选择的‘匹配界限’会非常低，这样只有非常严重劣化的帧才会具有 一个“不匹配的”指示符。

实现的全部细节见方法：

`CFrameAnalysisFullRef::sqVTA_ContentFrameTimeAlignement_M`。

2.3 空间上的帧对准

对处理过的视频序列的所有帧反复进行空间对准并且：

1) 如果这个帧是不匹配的，则使用以前的空间对准。如果该帧是匹配的，在处理过的和相应的–根据时间对准的匹配列表–参考帧之间进行空间对准：

a) 对于第一个帧，将空间位移预置为0（在水平和垂直两个方向上）。对于后续的帧，优先使用以前已匹配帧的空间对准。

b) 采用下面2）的位置界限，反复尝试所有可能的空间位移（水平和垂直方向上的）。如果不同的空间位移会导致处理过的和相应的参考帧之间有效的（相对于代价函数）细小差异，就调整空间位移。至于代价函数，采用下面的函数

```
rmse(Y(dv,dh),Y_ref) + abs(dv)+abs(dh),
```

其中Y表示分辨率为R1的处理过帧的Y平面，Y_ref表示分辨率为R1的参考帧，Y(dv,dh)表示位移为dv和dh的、经过移位的帧Y，dv、dh分别是水平和垂直方向上的位移，在代价函数中包含第二项和第三项是为了支持细小的空间位移。注意到计算rmse时，为了避免更加复杂的边界处理，要忽略帧的细小边界。

c) 这样，时变的空间位移就能得到补偿，某一帧的失配可以通过对准后续的帧来进行校正。

2) 自动空间位移对准的第一步限于±4个像素，更大的空间位移见第2.9节。

3) 空间对准之后，根据来自时间对准和明确定义的空间位移校正的匹配列表，处理过的视频序列中每一帧都有一个相对应的参考帧（或者在帧失配的情况下有两个相对应的参考帧），因此，处理过的视频序列帧能够将精确地与参考帧进行比对，这是下面特性提取的基础。

实现的全部细节见方法`CFrameAnalysisFullRef::DetermineSpatialAlignment`。为了适应更大的空间位移，可以增大步骤2）中的固定门限（±4个像素）。

2.4 局部相似性和局部差异性特性计算

针对每一个已对准的帧对，计算一组空间质量特性：

首先，对分辨率为R2的处理过的帧和参考帧中邻近的、均匀分布的大小为13×13的正方形区域重复计算局部*相似性*和*差异性*度量。由于分辨率R2不能被13整除，因此要忽略小的边界。

局部区域称作`s_prc`和`s_ref`，通过下式计算相似性`S`和差异性`D`：

```
S = (cor( s_prc,s_ref) + 25) / (var(s_ref) + 25)        (4.1)

D = sqrt(avg(( S*( s_prc-mean( s_prc)) -
        (s_ref-mean(s_ref)))^2))                        (4.2)
```

其中，`cor`是相关性，`var`是相应正方形区域中像素值的方差，函数`avg`计算正方形区域中所有像素的平均值，`sqrt`表示平方根，数值`D`和`S`对于空间质量数值起主要作用。

在这一点上，相似性和差异性特性S、D构成了每一帧的数值矩阵，每一个数值对应于一个正方形局部区域。对于预测质量十分重要的不仅是S、D的平均值，还包括S、D各自的分布形式。

2.5 局部特性分布分析

本节以介绍一些符号开始：

假设 $q_{quantile}(X,c)$表示矢量或矩阵X的数值（项）分布的c-*quantile*，更确切地说，对于矢量X以及大于等于0、小于等于1的常数c，分位数是数值q

$$q = quantile(X,c)$$

这样，在X的所有数值中有一小部分小于或者等于q。

函数*trimmed_mean*定义如下，这个符号将在以后使用，对于矩阵x，*trimmed mean*是x在c分位数和1-c分位数之间所有项的平均值。

$$trimmedMean(X,c)$$

例如，$trimmedMean(X,0.1)$是忽略了x中10％的最小的数和10％最大的数之后所有数值的平均值。

符号$X(X>c)$表示x中大于c的数值的集合，例如：

$$trimmedMean(X,c) = mean(X(X>quantile(X,c) \text{ and } X<quantile(X,1-c)))$$

使用这些符号，根据来自公式（4.1）的S和来自公式（4.2）的D计算以下特性数值：

$$s\_m = trimmedMean(S,c) \tag{5.1}$$

$$d\_m = trimmedMean(D,c) \tag{5.2}$$

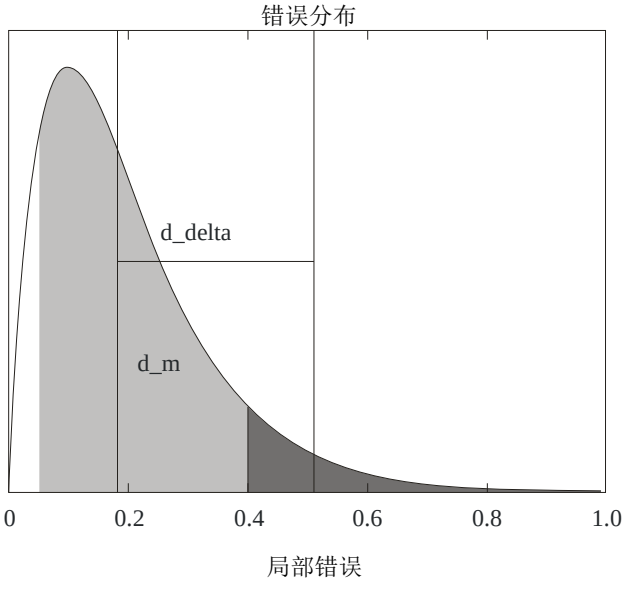
$$s\_delta = s\_m - mean(S(S<quantile(S,c))) \tag{5.3}$$

$$d\_delta = mean(D(D>quantile(D,1-c))) - d\_m \tag{5.4}$$

使用$c=0.2$。图6是对 d_m和d_delta的直观显示。

图 6

局部特性D的分布显示，与浅灰色区域（黑色垂直线）平均值相对应的经过裁剪的平均值d_m，与深灰和浅灰色区域（水平线）中平均值差相对应的数值d_delta



BT.1907-06

关于S和D的计算见方法CFrameAnalysisFullRef::ComputeSimilarity。

2.6 块效应特性计算

块效应计算使用的是分辨率为R1的帧，这个特性度量的是由编码和/或者传输错误引入的块边界的可见度，由于计算是在分辨率R1上进行的，因此设置了对边缘感知可见度的自动聚焦，从概述开始，块效应特性计算：

1) 水平和垂直边缘的方向导数（边缘图像）。对于视频序列中的每一帧，这会产生两个矩阵，一个对应于水平边缘，一个对应于垂直边缘，在下面伪码中被称作verGrad_n和horGrad_n。

2) 水平和垂直边缘的对数按行求和以及按列求和，产生两个矢量，一个对应于水平边缘的和，一个对应于垂直边缘的和，在下面被称作sumW和sumH。

3) 对sumW和sumH以步长n和偏移量m，进行二次采样的平均值，采用下面的函数vq_AvgSubsample计算。

想法是块大小为n的牢固块结构将显示为vq_AvgSubsample中步长为n针对不同偏移量计算得到的delta_edge的明显差异。

例如：原始帧中大小为4的块结构在分辨率R1下具有大小为2的块结构，因此，如果存在牢固的块结构，计算vq_AvgSubsample(x,2,0) 和vq_AvgSubsample(x,2,1)应显示出明显的差异。 为了避免内容依存性，采用大量视频序列样本的实验显示如何把度量处理过的视频序列的计算差值与参考序列的数值关联起来。

采用下面的伪码可以更好地解释计算的更多细节，其中，horGrad和verGrad是帧水平和垂直的空间导数，由相邻像素之差给出，

verGrad_n(i,j) = Y_n(i+1,j) – Y_n(i,j)和

horGrad_n(i,j) = Y_n(i,j+1) – Y_n(i,j),

其中Y_n(i,j)表示帧n的Y平面在(i,j)位置上的像素值。函数

vq_AvgSubsample(x, step, offset)

计算矢量x以步长为step、起始偏移量为offset的所有采样值的平均值。

```
// loop over all frames and compute:
for( UINT i=0; i<horGrad.Height; i++ ){
       for( UINT j=0; j<horGrad.Width; j++ ){
           w = (double)verGrad(i,j);
           h = (double)horGrad(i,j);
           // sum edges (-2: small differences can be the result of integer
           // values used to store frames)
           sumW(i) += log(1.0+max(0.0,fabs(w)-2.0));
           sumH(j) += log(1.0+max(0.0,fabs(h)-2.0));
       }
}

double dH0 = vq_AvgSubsample( sumH, 2, 0 );
double dH1 = vq_AvgSubsample( sumH, 2, 1 );
double dW0 = vq_AvgSubsample( sumW, 2, 0 );
double dW1 = vq_AvgSubsample( sumW, 2, 1 );
```

```
edge_max = 0.5 * (vq_Max(dW0,dW1) + vq_Max(dH0,dH1) );
edge_min = 0.5 * (vq_Min(dW0,dW1) + vq_Min(dH0,dH1) );


// now: denote by edge_max(i) the value of edge_max above, corresponding to
// frame i of the processed video sequence, and by edge_max_ref(i) the values
// edge_max above corresponding to frame i of the reference video sequence,
// and analogously for edge_min(i), edge_min_ref(i). Then compute:

for( UINT i=0; i<nbOfFramesInProcessedVideo; i++ ){
     // get frame nb of ref frame (according to match-list)
       UINT i_ref = (UINT)floor(ref_frameNb_all(i)+0.5f);

       float delta_edge = edge_max(i) - edge_min(i);
       float delta_edge_ref = edge_max_ref(i_ref) - edge_min_ref(i_ref);

       x(i) = vq_Max(0.0f,delta_edge - delta_edge_ref) / (1.0f+edge_max(i));
}
// blockiness(i) is then a non-linear monotone transform of x(i) ...
```

注意到由于可能会过采样到720帧，计算会略为复杂一些。

运动不平滑特性计算实现的全部细节见vquad_hd::vq_BlockinessPhaseDiff and CQualityModelFullRef::Blockiness。

2.7 运动不平滑特性计算（时间质量）

通过对相对显示时间、显示时间的非线性变换和运动强度的非线性变换的结果取平均来计算运动不平滑特性，运动强度主要来自帧的各个区域上的帧间差异，显示时间是以毫秒计的时间，帧显示在屏幕上。注意到由于处理过的视频序列中的帧可能是以前帧的重复，为了确定每一帧的显示时间，要进行局部运动强度分析。

运动不平滑特性考虑到了在回放处理过的视频序列期间丢失信息的数量，在平滑播放的序列中丢失信息的数量会很小，然而在出现暂停或者降低帧频的情况下，丢失信息的数量会增加，另一方面，对于固定的时间损伤，运动不平滑度量会针对较大运动强度的视频序列取较大的数值。

下面的伪码显示了一些细节，注意到输入是运动强度矢量motionInt、帧重复概率矢量repFrame、帧显示时间矢量displayTime，输出是处理过的视频序列中每一帧的运动不平滑矢量jerkiness。 更详细地，矢量motionInt表示在分辨率为R2的Y平面上测得的帧间差异的均方根，矢量repFrame表示帧重复的概率，即取决于运动强度，每一帧都有可能是以前帧的重复：在完全重复以前帧的情况下，实际帧被重复的概率为1，在大运动强度的情况下，实际帧重复以前帧的概率为0。当运动强度非常小但不为0时，概率会出现中间数值，确切地，

$$repFrame(i) = \begin{cases} 0 & \text{if } m(i) < p/2 \\ (m(i)-p/2)/p & \text{if } p(2) <= m(i) < 3/2*p \\ 1 & \text{if } 3/2*p <= m(i) \end{cases}$$

其中$m(i)$表示帧i的运动强度，根据经验，选择参数 p = 0.01。

```
int vq_CalcJerkiness( const CVector<float> & motionInt,
                              const CVector<float> & repFrame,
                              const CVector<float> & displayTime,
                              CVector<float> & jerkiness ){

      // ------------------------------------------------------------
      // the 4 parameters of the jerkiness measure: determined using a
   // large sample of video sequences containing temporal degradations
   // only.
      float a = 0.9f;
      float b = 5.0f;

      float aT = 40.0f;
      float bT = 5.0f;
      // ------------------------------------------------------------

      // get probability of new frame = 1 - prob of repeated frame:
      CVector<float> newFrame = repFrame*(-1.0f) + 1.0f;

      // count number of non-repeated frames
      float fNbRepeated = repFrame.Sum();
      float fNbNonRepeated = repFrame.Length() - fNbRepeated;

      // calculate jerkiness
      float fR = 0.0f;
      // look for frame repetition intervals (=~ display time)
      // of length i
      for( UINT i=1; i<= iNbFrames; i++ )
      {
         // look for frame repetitions starting at position j
         for( UINT j=0; j<iNbFrames-i+1; j++ )
         {
            float fP = newFrame(j);   // prob. : start of repetition block

            for( UINT k=1; k<i; k++ )
            {
               fP *= repFrame(j+k);  // prob. : all repeated frames
            }
            if( i+j < iNbFrames )
            {
               fP *= newFrame(j+i);  // prob. : end of repetition block
            }

            // calculate the display time (in s) of frame j,
      // if displayed from
            // time t_j until t_(j+i), which occurs with probability fP
            float fDispTime = displayTime.SumPart(j,j+i)/1000.0f;

            // -> measure jerking and add to result
            float fIFDiff = motionInt( j+i-1 );

            // normalisation values: such that at 0 jerkiness value is 0,
      // and saturates at 1
            float c = 1.0f / (1.0f+exp(b));
            float cT = 1.0f / (1.0f+exp(bT));
```

```
                float fJ = 1.0f / (1.0f + exp( -( a * fIFDiff  - b) ));
                float fJT = 1.0f / (1.0f + exp( -( aT * fDispTime - bT)));
                fJ = (fJ - c )/(1.0f - c);
                fJT = (fJT - cT)/(1.0f - cT);

                // jerkiness value: propability * interframeDiffFactor *
        //          displayTimeFactor

                fR = fP * fJ * fJT * fDispTime;

                // add to jerkiness vector at position j+i (corresponding to the
                // end of display time)
                jerkiness( vq_Min(j+i,iNbFrames-1) ) += fR;
            }
        }
        return 0;
}
```

关于实现的细节见方法 `vquad_hd::vq_ProbOfRepeatedFrame` 和 `vquad_hd::vq_CalcJerkiness`。

2.8 聚合为MOS

上述描述的特性，由`s_m`和`s_delta`给出的相似性，由`d_m`、`d_delta`和`jerkiness`给出的差异性，连同帧显示时间矢量`displayTime`，是得分评估的基础，这些矢量的长度等于处理过的视频序列的帧数。

为了将这些特性映射到一个可感知的范围内，使用一个参数化的S形函数：

$$S: \mathbb{R} \rightarrow \mathbb{R},\quad y = S(x)$$

采用三元组 (p_x,p_y,q) 对函数S进行参数化，该参数具有以下的解释：`(p_x,p_y)`是在$\mathbb{R} \times \mathbb{R}$中的位置，`q`是拐点的斜率，具体地：

$$S(x) = a * x^b \quad \text{if } x<=p\_x$$
$$d/(1+ \exp(-c*(x - p\_x))+1-d) \quad \text{其它} \tag{8.1}$$

其中：

```
a =   p_y/p_x^(q*p_x/p_y)
b =   q*p_x/p_y
c =   4*q/d
d =   2*(1-p_y)
```

不同参数的S函数曲线见图7，S形函数从原点开始，按多项式规律上升到拐点，然后按指数规律向1接近。

图 7

采用拐点的位置和斜率的参数化的S形函数，不同参数的两个样本函数显示

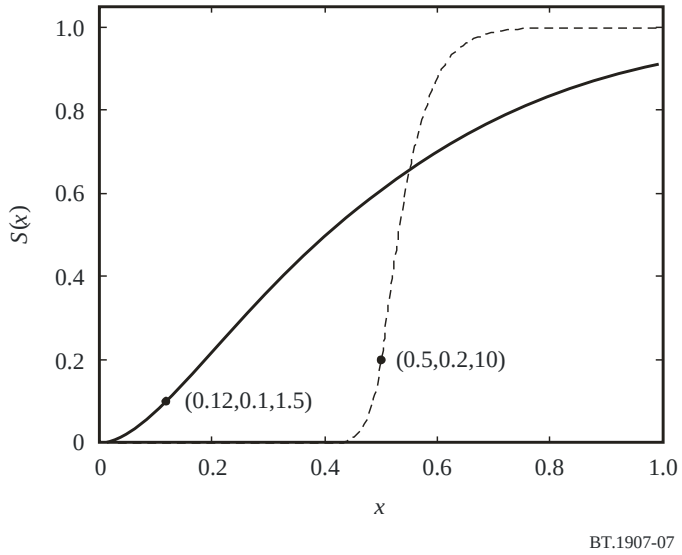
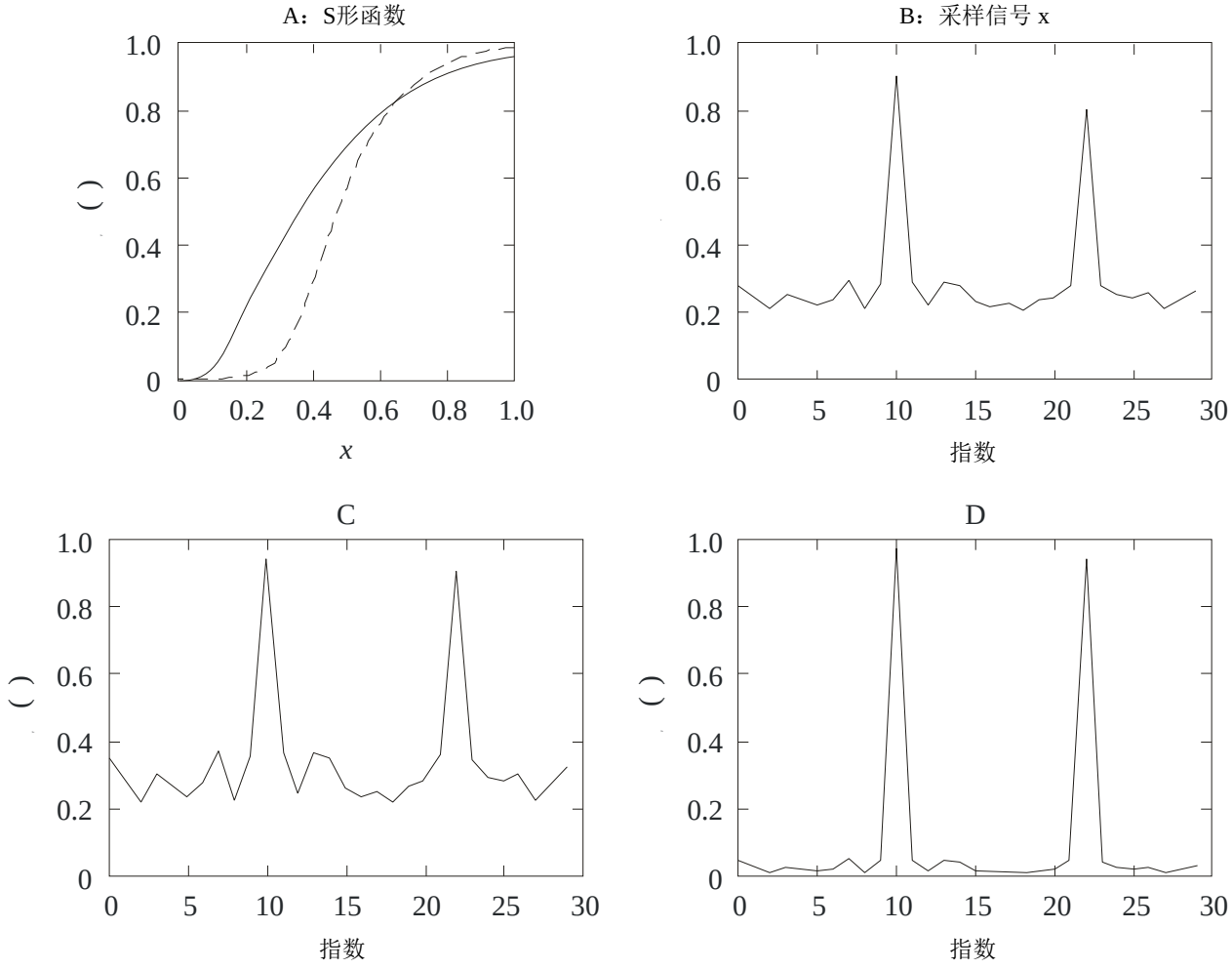


图 8

图（A）显示的是对于2种不同参数集的S形函数，图（B）显示的是采样信号矢量。
使用图（A）中的2个S形函数对x的所有数值进行变换，结果如图（C）和（D）所示。
采用适合于图（C）中所示数据样本的参数，S形变换用于将特性x映射到可感知的范围内。
使用一种S形函数（图（A）中的虚线）以及数据相关的参数（第一个参数设为x信号平均值的倍数），可构造一个瞬时劣化检测器，图（D）



使用固定参数的S形函数，能将特性值映射到可感知的范围。例如，使用与数据相关的参数，S形函数能将所有低于平均值的数值压缩成一个较小的数值，并且扩展所有高于平均值的数值。这样，它能用于计算对于瞬时劣化敏感的特性，如图8所示。

继续描述模型，首先，根据上面的相似性特性定义一种劣化，但是会给予最严重劣化更多的权重：

```
d_s = 1 - s_m + 1.5 s_delta
```

下一步的想法是使用两个S形函数，一个S形函数使用固定参数集`cod_par`，将`d_s`变换为与基本劣化相关的感知范围内的`d_cod`，反映由视频编码产生的错误。

第二个S形函数具有与数据相关的参数，将`d_s`变换为与瞬时劣化相关的感知范围内的`d_trans`，反映传输错误。

具体地，对于下面的伪码，采用矢量`d_s`和帧显示时间矢量`disp_time`作为输入，输出为`d_cod`和`d_trans`，调用函数`SplitCodTrans`，

```
SplitCodTrans(d_s, disp_time, d_cod, d_trans).
```

下面伪码显示了函数的细节，注意到

```
stat.STransform(x,px,py,q)
```

表示S形函数，其输入为将要变换的实际数值x，以及公式（8.1）中(px,py,q)表示的三个参数。

```
SplitCodTrans( const CVec& v, const CVec& dispTime,
        CVec& v_cod, CVec& v_trans ){

   // these parameters are determined empirically
       float qPosSmall = 0.55f;
       float qPosLarge = 0.65f;

   // q is the mean of the values in v between the qPosSmall and qPosLarge
   // quantiles.
       float q = r.TrimmedMean( qPosSmall, qPosLarge, v, dispTime );

       for( UINT i=0; i<v.Length(); i++ ){

     // the parameters used here are the result of fitting to sample
     // data
          v_cod(i) = stat.STransform(v(i), 0.07f, 0.1f, 2.0f);

     // Note that the STransform is not directly applied to v, but
     // to v(i)-q . This is prefered here for numerical reasons of the
     // resulting STransform.

          // v_trans is part of v above quantile value q
          v_trans(i) = vq_Max(0.0f, v(i)-q);

          float px = 0.5f * (q+0.2f);
     // the parameters used here are the result of fitting to sample
     // data
          v_trans(i) = stat.STransform(v_trans(i),px, 0.1f,16.0f);
       }
}
```

实现的全部细节见CQualityModelFullRef::SplitCodTrans。

类似地，d_diff_cod、d_diff_trans来自于from d_m、d_delta，通过设置

$$d\_diff = d\_m + 1.5\ d\_delta,$$

并调用

$$SplitCodTrans(d\_diff,\ disp\_time,\ d\_diff\_cod,\ d\_diff\_trans),$$

使用适用于两种S变换的参数三元线

$$cod\_par = (4.0f,0.05f,0.2f)$$

$$trans\_par = (0.5*(q+4.0f),0.1f,0.4f),$$

其中，与在上面伪码中一样，q表示分位数之间的中间值，函数的输出为d_diff_cod、d_diff_trans。

接下来，通过运动不平滑的S形变换来计算与暂时严重运动不平滑数值相关的特性值：

$$d\_t\_trans = S(max(0,jerkiness-q))$$

采用由(max(0.048,q), 0.2,40.0)给出的S形变换的参数，q表示运动不平滑矢量0.55分位数和0.65分位数之间的分位数中间值，这些参数由大型采样数据集确定。

实现的全部细节见CQualityModelFullRef:: SplitTempTrans。

下一步，基本质量q_cod定义为：

$$q\_cod = (1 - d\_cod) * (1 - d\_diff\_cod) * (1-blockiness) \tag{8.2}$$

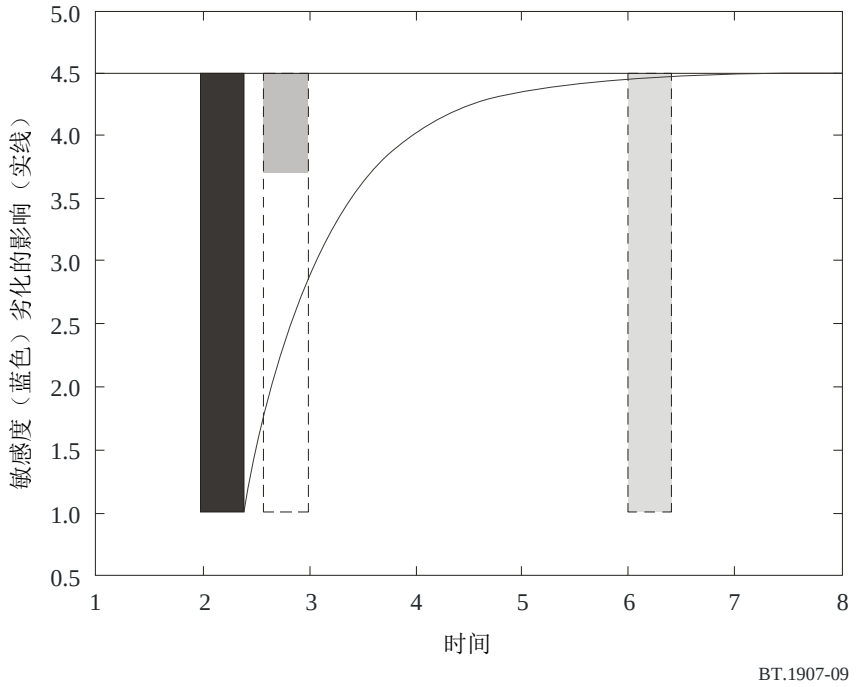
瞬时质量定义为：

$$q\_trans = (1 - d\_trans) * (1 - d\_diff\_trans) * (1 - d\_t\_trans).$$

如果另外一次劣化在第一次劣化之后不久出现，则会减弱这一次劣化的影响。为了说明这种效果，可将q_trans变换为q_fq，这种想法如图9所示。

图9

黑实线表示在假设的视频序列中出现的一次劣化，黑实线表示对后续劣化的敏感度。如果由虚线给定强度的另外一次劣化在第一次劣化之后不久出现，则可以减少这一次劣化的影响（第一个深灰色条）。如果随后的劣化以较大的时间间隔在中间出现（右边的浅灰色条），则不能改变它的影响。



计算的细节可通过下面的伪码部分得到最好的描述，使用1-q_trans作为输入矢量v，矢量disp_time是帧显示时间矢量，利用采样数据确定衰减时间常数dT=1000，然后设

$$q\_fq = 1 - w, \tag{8.3}$$

其中w是函数的输出矢量：

```
DegFreq( const CVec& v, const CVec& disp_time, CVec& w, float dT ){

       // time constant for temporal integration of degradations
       float t_const = 80.0f;
       w(0) = v(0)*vq_Min(t_const,disp_time(0))/t_const;

       for( UINT i=1; i<v.Length(); i++ ){
          // integrate degradation over the last t_const milliseconds:
          // use an indicator function, which is 1 over the interval
          // [t_i-t_const, t_i] and 0 otherwise
          float dT_sum = 0.0f;
          UINT j=0;
          float v_sum = 0.0f;
          while( dT_sum < t_const && (int)i-(int)j>=0 ){
             // b is the overlap in [0,1] of the display time interval
             // of frame i-j with respect to the integration interval
             //       t_i-t_const   t_i
             //       _______________
             //       |         |       integration window
             // ------------------------------------------------> time
             //  |    |     |     |     |     |  frames
             //
             //      ->| b*dT |<-
             //
             float b = vq_Min(t_const-dT_sum,disp_time(i-j)) / t_const;

             v_sum += v(i-j)*b;
             dT_sum += disp_time(i-j);
             j++;
          }
          // compute the fall-off factor a:
          float a = exp(-disp_time(i-1)/dT);
          // the degradation is now:
          // 1) a linear combination of the fall-off of a previous degradation
          //  and the current (integrated) degradation, or
          // 2) the integrated current degradation (if it is stronger than 1)
).
          w(i) = vq_Max(v_sum, a * w(i-1) + (1.0f-a) * v_sum);


       }
```

取平均过程

假设`disp_time(i)`表示帧`i`的显示时间，`jerkiness(i)`表示相对于帧`i`的值，从公式（8.2）调用`q_cod`，从公式（8.3）调用`q_fq`，设

```
Q_t  = 1 - 1/T sum_i jerkiness(i)
Q_fq = 1/T sum_i q_fq(i)* disp_time (i)
Q_cod = 1/T sum_i q_cod(i)* disp_time (i)
```

其中`T=sum_i disp_time(i)`和`sum_i`表示`i=0,..,number_of_frames`所有指数之和。

最终预测得分是结果，并且调整到[1,5] 的MOS范围：

```
s = 4 * Q_t * Q_cod * Q_fq + 1
```

关于得分预测的细节见方法`CQualityModelFullRef::PredictScoreCodTrans`。

2.9 严重空间未对准视频序列的处理

在参考的和处理过的视频序列之间空间位移相对较大的情况下，为了避免错误预测，要对3个不同的水平和垂直空间对准步幅的视频序列计算上述步骤，各个空间位置中的最大预测得分就是最终的评估质量得分。

各个方向上采用的步长为4个像素，这样，就实现了±8个像素的空间搜索范围，这很容易覆盖测试集中采用的最大空间位移（±5个像素）。由于采用模型的高级函数实现这种扩展，因此校准范围能够很容易地适应更大的位移，或者为了节省计算资源而缩少校准的范围（例如±4个像素）。

见`vquad_hd::vq_vquad08`。

2.10 参考源代码实现

本节包含了一个嵌入式文件以及涵盖与模型描述相符的实现必需的部分和函数的C++源代码。本节上面部分中到实际实现的所有关联可查阅该参考源代码。

附件3

一致性测试

本附件由包含下列信息和文件的数字附件构成：

1) 16个短HD视频序列（用于8个测试用例的参考的和劣化的序列）。这些序列将涵盖不同的失真和内容，用于测试相对于模型的参考实现，本建议书的用户完成的模型实现的一致性。为了减少存储容量，视频序列只包含几个帧，它们不是用于任何视觉测试，只是用于HD模型实现的一致性测试。

2) HD序列的预测MOS得分在1）下面被称为‘HD_ConformanceReferenceResults.xls’，可以通过HD模型的参考实现获得这些得分。

3) 通过VQEG可获得5个公共HD数据库的预测MOS得分，这些数据库能够用于模型实现的扩展一致性测试。

一致性测试判据：

i) 必须通过实现模型精确地再生2）中给出的8个参考得分。精确代表着再生的得分是精确到3位小数的数字。

ii) 必须以非常有限的偏差再生5个公共VQEG数据库的预测MOS得分，微小的变化是允许的，原因是经验表明在速度和存储使用方面的不同优化能够导致最后得分的微小的和可以忽略的偏差。

表2

所有一致性测试数据允许的误差分布

绝对误差	允许发生的概率
> 0.0001	5.00%
> 0.001	1.00%
> 0.01	0.50%
> 0.1	0.05%
> 0.3	0.00%

对于与本附件3规定的数据库不同的数据库，不允许超出同样的误差分布。对于未知的数据，为了那些统计数据必须要使用由至少500个文件对–最好来自完全主观的实验–组成的测试集。

参考资料

VQEG Final Report of HDTV Phase I Validation Test (2010), "*Video Quality Experts Group: report on the validation of video quality models for high definition video content*", Video Quality Experts Group (VQEG), http://www.its.bldrdoc.gov/vqeg/projects/hdtv

ITU-R BT.601-7建议书 (2011)，《标准4:3和宽银幕16:9宽高比的数字电视的工作室编码参数》。

ITU-T J.244建议书(2008)，《用于视频传输系统的完全基准和简化基准校验法，该系统空间域和临时域有常数增益和偏移但常数未校准》。

ITU-R BT.500-12建议书(2009)，《主观评估电视画面质量的方法》。

ITU-T J.149建议书(2004)，《确定视频质量衡量精确度和交叉校准的方法》。

ITU-T J.247建议书(2008)，《存在完全基准的客体感知多媒体视频的质量衡量》。

ITU-T J.144建议书(2004)，《存在完全基准的数字有线电视中用于客体感知视频质量衡量的技术》。

ITU-T P.931建议书(1998)，《多媒体通信延迟、同步和帧速率测量》。

ITU-T J.148建议书(2003)，《客体感知多媒体质量模型的要求》。

ITU-T H.264建议书(2012)，《一般视听业务的高级视频编码》。

ITU-T J.340建议书(2010)，《补偿常数空间偏移、常数临时偏移和常数亮度增益和偏移的已处理视频序列的计算峰值信噪比的基准算法》。

ITU-T P.910建议书(2008)，《多媒体应用的主观视频质量评估方法》。

ITU-T P.911 (1998)建议书，《多媒体应用的主观视听质量评估方法》。

ITU-T J.143 建议书(2000)，《数字有线电视中用于客体感知视频质量衡量的用户要求》。