

国 际 电 信 联 盟

ITU-R

国际电联无线电通信部门

ITU-R BT.1885 建议书

(03/2011)

**在减少带宽参考的情况下
标清数字广播电视的
客观感知视频质量测量**

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号决议的附件1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
2011年，日内瓦

© ITU 2011

版权所有。未经国际电联书面许可，不得以任何手段复制本出版物的任何部分。

ITU-R BT.1885建议书

在减少带宽参考的情况下标清数字广播电视的
客观感知视频质量测量

(2011年)

范围

本建议书描述了当从参考视频信号中提取的特征在测量点可用时，可衡量移动和静止接收条件下客观视频质量的标清数字广播电视的客观感知视频质量。

国际电联无线电通信全会，

考虑到

- a) 需要自动测量广播视频信号衰减的能力；
- b) 根据接收条件的不同，移动接收的感知视频质量可能会大幅变化；
- c) 对感知视频质量的客观测量有可能对主观评价方法进行有益的补充；
- d) 已经向国际电联无线电通信部门建议在减少带宽参考的情况下标清数字广播电视的三种视频质量测量方法且已发现这三种方法均可提供等效且一致的结果；
- e) ITU-R BT.1683建议书中规定了完整参考情况下标清数字广播电视的客观感知视频质量测量方法，

做出建议

1 附件1中给出的任何客观视频质量模型均应用于在减少带宽参考的情况下标清数字广播电视的客观感知视频质量的测量。

1 引言

该RRNR-TV测试涉及ITU-R BT.601-6建议书中定义的图像和两种模型：减少参考（RR）和无参考（NR）。RR模型接入源视频的带宽有限，而NR模型则无法接入源视频。

各种实验中的HRC包括仅对伪影进行编码和带有传输误差的编码。研究的编码方法是MPEG-2和H.264（MPEG-4第10部分）。按照1.0至5.5 Mbit/s的不同比特率运行MPEG-2编码器。按照1.0至3.98 Mbit/s的比特率运行H.264编码器。每个实验包括12个源序列，其中两个为秘密源。每个实验包括34个HRC和156个处理后的视频序列（PVS）。这些PVS中，40个包含传输误差，116个仅包含编码。

1.1 应用

如ITU-T P.911建议书附件B所述，本建议书提供了TV3至MM5B视频类别的视频质量评估。本建议书中描述的评估模型的应用包括但又不限于：

- 1 在源头潜在的实时、运行中的质量监测；
- 2 当邻信道可用于从源视频序列提取的特征时，远程目标质量监测；
- 3 用于对采用视频压缩和解压缩技术（单程或一系列此类技术）的存储或传输系统进行监测的质量测量；
- 4 视频系统的实验室测试。

1.2 局限性

本建议书中描述的评估模型不能用来完全取代主观性测试。两种仔细设计并开展的主观性测试（即在两个不同的实验室）之间的相关值通常在0.95至0.98的范围内。如果将本建议书用于比较不同的编码集，建议采用量化的方法（如ITU-T J.149建议书中的方法）来判断模型在那种特定条件下的精确性。

本建议书模型采用了可显示最高2秒的帧凝固的客观视频质量进行验证。

本建议书模型并未采用测量具有不断增加时延的客观视频质量（即在帧凝固之后不丢弃失真的视频）进行验证。

应注意到，如果出现了产生未在本评估中包括的伪影的新编码和传输技术，客观评估模型可能会产生错误的结果。此时，需要进行主观性评估。

2 参考文件

下列ITU-T建议书和其他参考文献的条款，通过在本建议书中的引用而构成本建议书的条款。在出版时，所指出的版本是有效的。所有的建议书和其他参考文献都面临修订，使用本建议书的各方应探讨使用下列建议书和其他参考文献最新版本的可能性。当前有效的ITU-T建议书清单定期出版。

本建议书中引用某个独立文件，并非确定该文件具备建议书的地位。

2.1 规范性参考文件

ITU-R BT.500-12建议书 – 主观评价电视图像质量的方法。

ITU-T P.910建议书（2008年） – 多媒体应用的主观性视频质量评价方法。

ITU-T P.911建议书（1998年） – 多媒体应用的主观性音视频质量评价方法。

ITU-T J.143建议书（2000年）– 数字有线电视中客观感知视频质量测量的用户需求。

ITU-T J.244建议书（2008年）– 具有固定空间和时间域偏差并具有普遍增益和偏移的视频传输系统的全参考和减少参考情况下的校准方法。

2.2 资料性参考文件

ITU-T J.149建议书（1998年）– 多媒体应用的主观性音视频质量评价方法。

ITU-T J.144建议书（2001年）– 数字有线电视在全参考情况下客观感知视频质量的测量方法。

ITU-T P.931建议书（1998年）– 多媒体通信时延同步和帧速率测量。

ITU-T J.148建议书（2003年）– 客观感知多媒体质量模型的要求。

ITU-T H.261建议书（1993年）– p x 64 kbits音视频业务的视频编解码器。

ITU-T H.263建议书（1996年）– 低比特率通信的视频编码。

ITU-T H.263建议书（1998年）– 低比特率通信的视频编码（H.263+）。

ITU-T H.264建议书（2003年）– 通用音视频业务的高级视频编码。

VQEG – 标清电视减少参考和无参考客观模型的验证，第一阶段，2009年。

3 定义

3.1 在别处定义的术语：

本建议书采用了以下在别处定义的术语：

3.1.1 主观性评价（图像）（ITU-T J.144建议书）：可选引用定义。

3.1.2 客观感知测量（图像）（ITU-T J.144建议书）：可选引用定义。

3.1.3 支持者（ITU-T J.144建议书）：可选引用定义。

3.2 本建议书定义的术语

本建议书定义了以下术语：

3.2.1 不规则帧重复（Anomalous frame repetition）：定义为HRC为响应异乎寻常或不平常的事件而不断地输出一个单一帧的事件。不规则帧重复包括但又限于以下类型的事件：传输信道中的错误、经由传输信道的时延变化、影响解码器性能的有限计算机资源以及影响视频信号显示的有限计算机资源。

3.2.2 连续跳帧（Constant frame skipping）：定义为HRC根据固定且小于原帧速率的有效帧速率输出内容更新的帧的事件。

3.2.3 有效帧速率（Effective frame rate）：定义为每秒单帧的数量（即总帧数减去重复帧数）。

3.2.4 帧速率 (Frame rate) : 定义为每秒单帧的数量 (即总帧数 – 重复帧数)。

3.2.5 计划帧速率 (Intended frame rate) : 定义为对于视频序列的某种显示, 实际存储的每秒视频帧数量。帧速率须固定不变。包含25帧/秒的BetacamSP®录像带和包含25帧/秒的VQEG FR-TV第一阶段符合625线的YUV文件是固定不变计划帧速率的两个例子; 这两者都拥有25帧/秒的计划帧速率。

3.2.6 实时网络条件 (Live network conditions) : 定义为由于实时网络条件而给数字视频比特流带来的误差。

3.2.7 跳跃暂停 (Pausing with skipping) : 定义为当视频暂停一段时间, 然后在损失一些视频信息之后从新开始的事件。在跳跃暂停中, 系统的时间延时将变化约一个平均系统时延, 有时增加, 有时减少。跳跃暂停的一个示例是一对IP视频电话, 很大的网络流量导致IP视频电话显示短暂凝固; 当IP视频电话显示继续时, 一些内容已经失去。固定帧跳跃和可变帧跳跃是跳跃暂停的子集。一个包含跳跃暂停的处理视频序列将约是与相关原始视频序列相同的长度。

3.2.8 无跳跃暂停 (Pausing without skipping) : 定义为视频暂停一段时间, 然后在不损失任何视频信息的情况下重新开始的事件。因此, 整个系统的时间时延必须增加。

3.2.9 刷新率 (Refresh rate) : 定义为显示更新的速率。

3.2.10 仿真传输误差 (Simulated transmission errors) : 定义为在高度可控的环境下对数字视频比特流施加的误差。其示例包括仿真的分组损耗率和仿真比特误差。

3.2.11 源帧速率 (SFR) : 是原始源视频序列的计划帧速率。源帧速率是固定的。对于VQEG RRNR-TV测试, SFR或者是25帧/秒, 或者是30帧/秒。

3.2.12 传输误差 (Transmission errors) : 定义为视频传输上施加的任何误差。误差的示例类型包括仿真传输误差和实时网络条件。

3.2.13 可变帧跳跃 (Variable frame skipping) : 定义为HRC按照随着时间变化的有效帧速率输出带内容更新的帧的时间。系统的时间时延将随着时间增加或降低, 变化幅度约为一个平均系统时延。一个包含可变帧跳跃的经处理视频序列将大约是与相关原始视频序列相同的时长。

4 缩略语与首字母缩写词

本建议书采用了以下缩略语和首字母缩写词:

ACR	绝对种类定级 (参见ITU-T P.910建议书)
ACR-HR	带隐藏参考的绝对种类定级 (参见ITU-T P.910建议书)
AVI	音频视频交织
DMOS	差值平均主观评分
FR	全参考
FRTV	全参考电视
HRC	假定参考电路

NR	无（零）参考
PSNR	峰值信噪比
PVS	经处理视频序列
RMSE	均方根误差
RR	减少参考
SFR	源帧速率
SRC	源参考信道或电路
VQEG	视频质量专家组
YUV	色空间

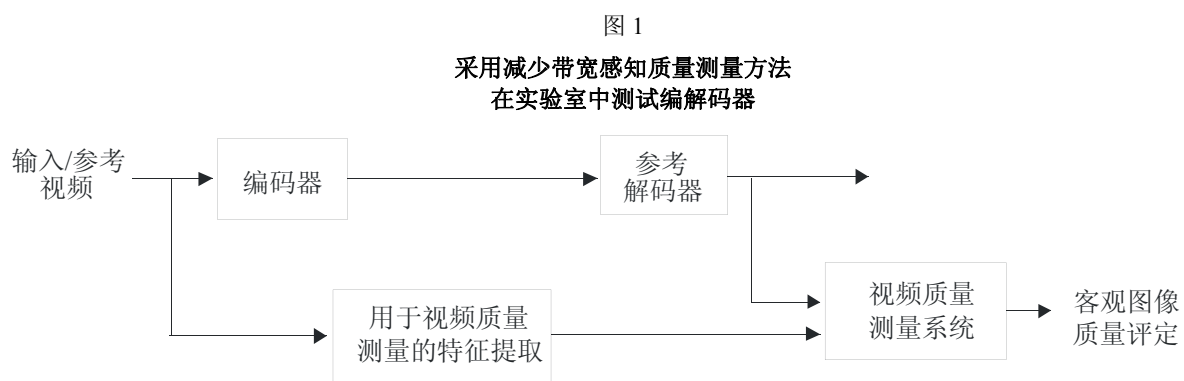
5 惯例

无。

6 减少参考测量方法的描述

用于客观测量感知视频质量的减少参考双端测量方法通过比较在系统输入端从未失真的输入（或参考）提取的特征与系统输出端的降级信号来评估系统的性能（见图1）。

图1说明了采用减少参考方法在实验室中测试编解码器的示例。



BT.1885-01

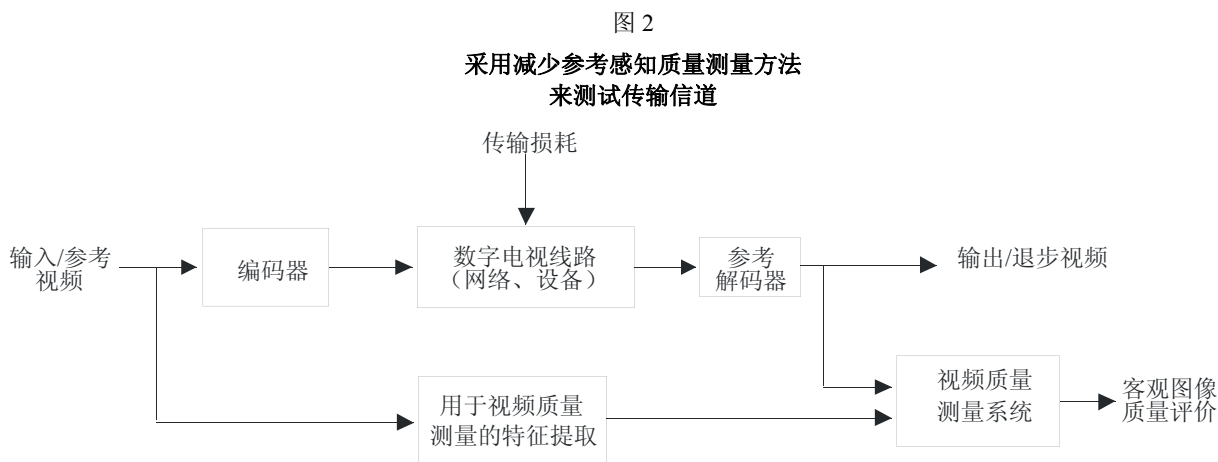
比较输入和输出端信号可能需要时间或空间的校正过程，后者用来补偿一切垂直或水平图像偏移或剪切。也可能需要对亮度和色度信道中的任何偏移或增益差异进行校正。然后，通常通过采用人体视觉的感知模型计算客观图像质量评价。

校正和增益调整称为规范化。需要该过程是因为绝大多数减少参考方法都是实际上在逐个像素的基础上比较从参考图像和已处理图像提取的特征。计算峰值信噪比（PSNR）即是个示例。在视频中移除了时变的静态变化，测试中系统所造成的动态变化则作为质量评价计算的一部分进行测量。ITU-T J.244和ITU-T J.144建议书提供了在客观质量评估之前报告规

范化视频所需数值的标准方法。在建议书附件中描述的视频质量规格包括了相关的规范化方法。如果其它的规范化方法可以提供所要求的规范化精确度，它们也可用于附件中的视频质量规格。

由于视频质量规格通常基于人类视觉反映的近似估计，而不是测量具体的编码伪像，原则上它们同等适用于模拟系统和数字系统。原则上，它们也适用于模拟和数字系统混合的或将数字压缩系统串联起来的电路中。

图2显示了采用减少参考方法测试传输信道的示例。



BT.1885-02

在这种情况下，在传输线路中各个点提供了参考解码器，即如图2所示，解码器可位于网络中的某一点或如图1所示，直接位于编码器的输出端。如果数字传输线路是透明的，那么在源头的客观图像质量评价测量等同于在线路中任意连续点的测量。

一般认为，全参考方法提供了感知图像质量测量的最佳精确性。该方法已证明可用于根据ITU-T P.910建议书所述的ACR-HR方法进行的高度相关的主观性评价。

7 视频质量专家小组的结论

在一个称为“视频质量专家小组（VQEG）的非正式小组内进行了感知视频质量测量的研究，该小组向ITU-T第9和第12以及无线电通信第6研究组报告。近期完成的VQEG的RRNR-TV测试评估了用于ITU-T 601-6图像格式的拟议减少参考感知视频质量测量算法的性能。

根据目前的证据，在当前阶段，ITU-T可以建议六种RR方法（Model_A 15k、Model_A 80k、Model_A 256k、Model_C 80k、Model_B 80k（仅525线）、Model_B 256k（仅525线））。

这些模型的技术描述可分别参见附件A至附件C。应注意，附件的排序不分先后，不意味着质量预测性能的高低。

表1和表2显示了VQEG RRNR-TV测试中重要的测试。对于525线格式，从统计数据上看，此种模型（Model_A 15k、Model_A 80k、Model_A 256k、Model_C 80k）要好于PSNR且在统计数据上，两种模型（Model_B 80k、Model_B 256k）与PSNR相当。应注意到，PSNR是由NTIA采用校准限值穷举搜索的方法计算的。对于625线格式，在统计数据上，四种模型（Model_A 15k、Model_A 80k、Model_A 256k、Model_C 80k）相当，但在统计数据上要优于PSNR。

表 1

525线格式的有效测试

525线格式	比较最优	比较PSNR	相关性
Model_A 15k	1	1	0.906
Model_A 80k	1	1	0.903
Model_A 256k	1	1	0.903
Model_C 80k	1	1	0.882
Model_B 80k	0	1	0.795
Model_B 256k	0	1	0.803
PSNR_NTIA	0	1	0.826

注 1 – “比较最佳”中的“1”表示在统计数据上，该模型与表现最优的模型相当。“0”表示在统计数据上，该模型与性能最优模型并不相当。“比较PSNR”中的“1”表示在统计数据上，该模型与性能最优模型相当。“0”表示在统计数据上该模型与性能最优模型并不相当。

表 2

625格式的有效测试

625线格式	比较最优	比较PSNR	相关性
Model_A 15k	1	1	0.894
Model_A 80k	1	1	0.899
Model_A 256k	1	1	0.898
Model_C 80k	1	1	0.866
PSNR_NTIA	0	1	0.857

注 1 – “比较最佳”中的“1”表示在统计数据上，该模型与表现最优的模型相当。“0”表示在统计数据上，该模型与性能最优模型并不相当。“比较PSNR”中的“1”表示在统计数据上，该模型与性能最优模型相当。“0”表示在统计数据上该模型与性能最优模型并不相当。

表3和表4提供了VQEG RRNR-TV测试中各模型性能的详细信息。

表 3

VQEG RRNR-TV测试中模型性能的信息描述
(525线格式)

525线格式	相关性	RMSE	OR
Model_A 15k	0.906	0.418	0.385
Model_A 80k	0.903	0.423	0.378
Model_A 256k	0.903	0.424	0.378
Model_B 80k	0.795	0.598	0.667
Model_B 256k	0.803	0.587	0.647
Model_C 80k	0.882	0.465	0.513
PSNR_NTIA	0.826	0.556	0.571

表 4

VQEG RRNR-TV测试中模型性能的信息描述
(625线格式)

625线格式	相关性	RMSE	OR
Model_A 15k	0.894	0.524	0.468
Model_A 80k	0.899	0.513	0.462
Model_A 256k	0.898	0.516	0.468
Model_C_80k	0.866	0.585	0.583
PSNR_NTIA	0.857	0.605	0.564

附件 A

模型A：延世大学减少参考方法

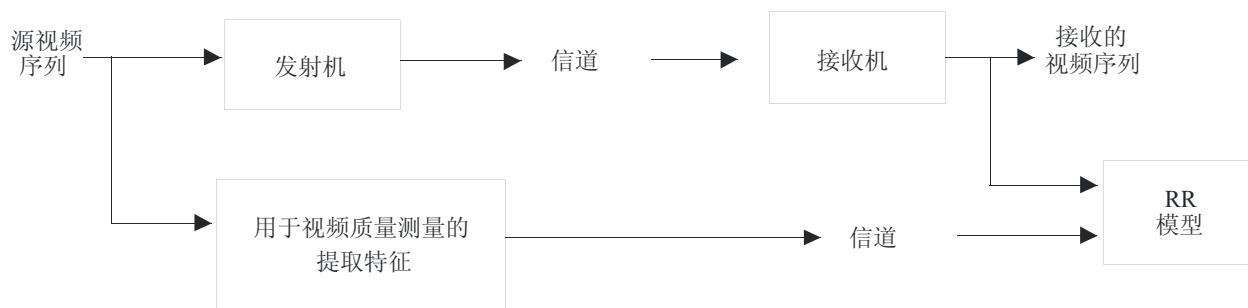
1 引言

尽管PSNR已作为客观视频质量测量而广泛使用，也有人报告该方法并不能很好地表现感知视频质量。通过分析人类如何感知视频质量，发现人类视觉系统对边缘周围的退步降级较为敏感。也就是说，尽管PSNR值很高，当一段视频的边缘像素变得模糊时，评估人倾向于给视频打出低分。根据以上发现，制定了主要测量边缘退步降级的减少参考模型。

图3显示了降低参考模型是如何工作的。从源视频序列提取将用来在一个监测点测量视频质量的特征并进行传输。表5显示了特征的邻信道带宽，已在VQEG RRNR-TV测试中对其进行了测试。

图 3

降低参考模型的方框图



BT.1885-03

表 5

邻信道带宽

视频格式	测试带宽
525 格式	15 kbit/s, 80 kbit/s, 256 kbit/s
625 格式	15 kbit/s, 80 kbit/s, 256 kbit/s

2 EPSNR减少参考模型

2.1 边缘PSNR

RR模型主要测量边缘的退步。在模型中，首先将边缘探测算法应用于源视频序列，以确定边缘像素。然后，通过计算平均方根误差来测量这些边缘像素的退步。根据此平均方根误差，计算边缘PSNR（EPSNR）。

人们可以采用任意的边缘探测算法，尽管结果会有细微的差别。例如，人们可以使用任意的梯度算子来定位边缘像素。已经建议了多个梯度算子。在许多边缘探测算法中，首先采用梯度算子计算水平梯度图像 $g_{horizontal}(m,n)$ 和垂直梯度图像 $g_{vertical}(m,n)$ 。然后，可按照以下公式计算幅度梯度图像 $g(m,n)$ ：

$$g(m,n) = |g_{horizontal}(m,n)| + |g_{vertical}(m,n)|$$

最后对幅度梯度图像 $g(m,n)$ 应用阈值运算，来发现边缘像素。也就是说，幅度梯度超过门限值的像素视为边缘像素。

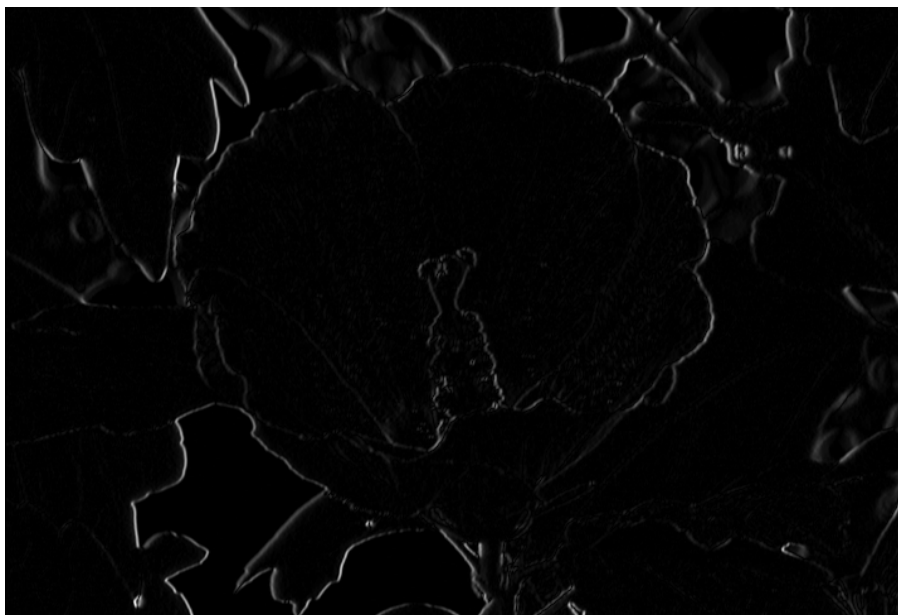
图4至图8说明了上述过程。图4显示了一个源图像。图5显示了一个水平梯度图像 $g_{horizontal}(m,n)$ ，该图像通过对图4的源图像应用水平梯度算子获得。图6显示了一个垂直梯度图像 $g_{vertical}(m,n)$ ，该图是通过对图4源图像应用垂直梯度算子获得。图7显示了幅度梯度图像（边缘图像），图8显示了通过对图7的幅度梯度图像采用阈值后获得二进制边缘图像（掩模图像）。

图 4
源图像 (原始图像)



BT.1885-04

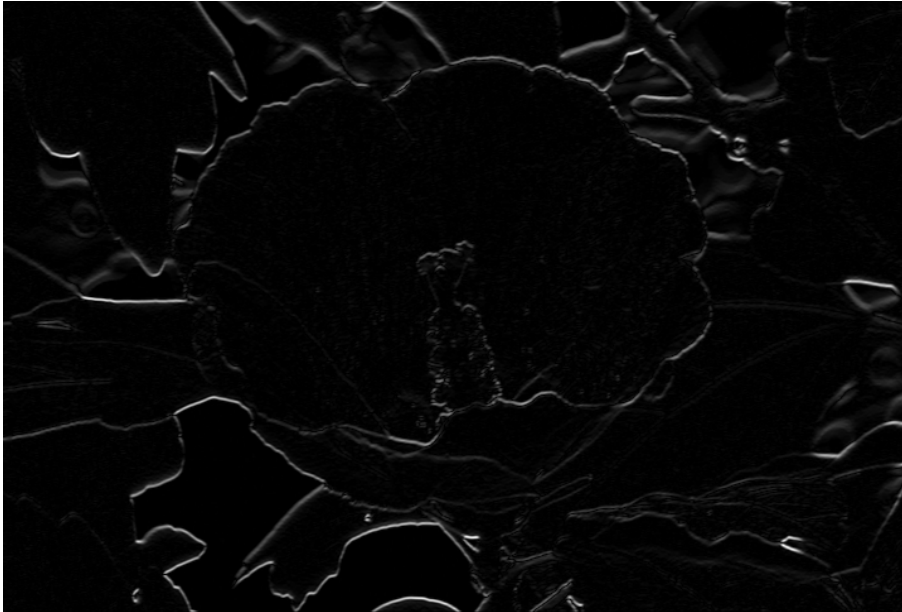
图 5
通过对图4的源图像应用水平梯度算子
获得的水平梯度图像



BT.1885-05

图 6

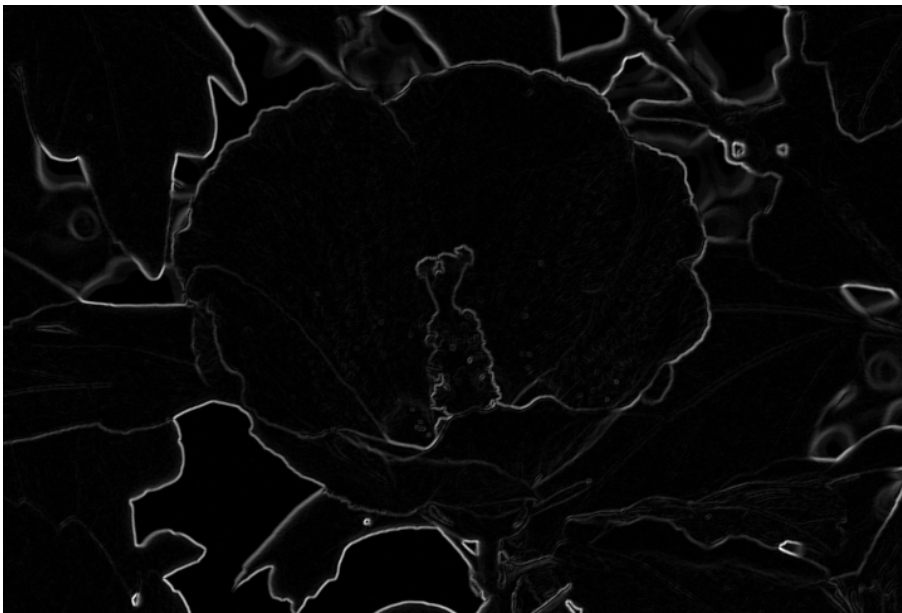
通过对图4的源图像应用垂直梯度算子
获得的垂直梯度图像



BT.1885-06

图 7

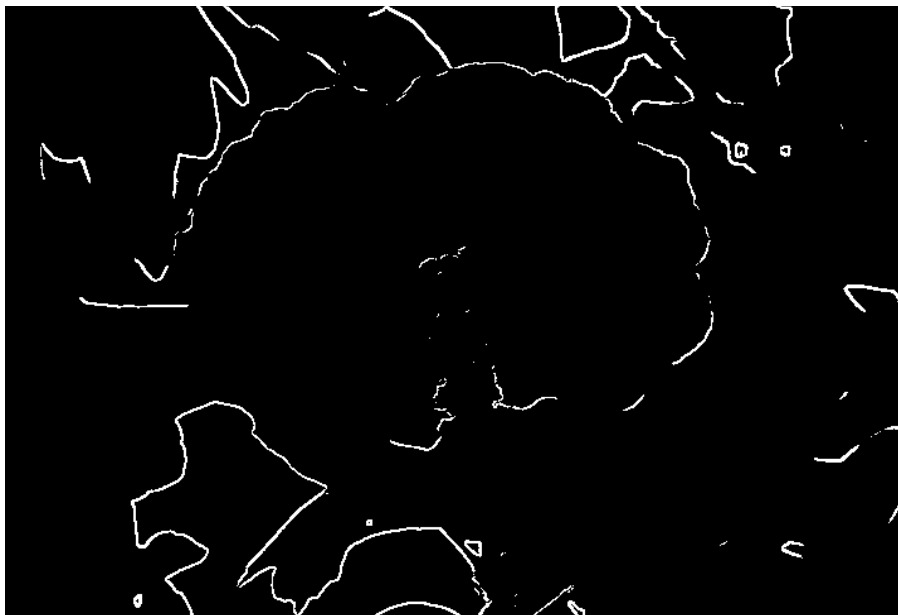
幅度梯度图像



BT.1885-07

图 8

通过对图7的幅度梯度图像应用门限之后
获得的二进制边缘图像（掩模图像）



BT.1885-08

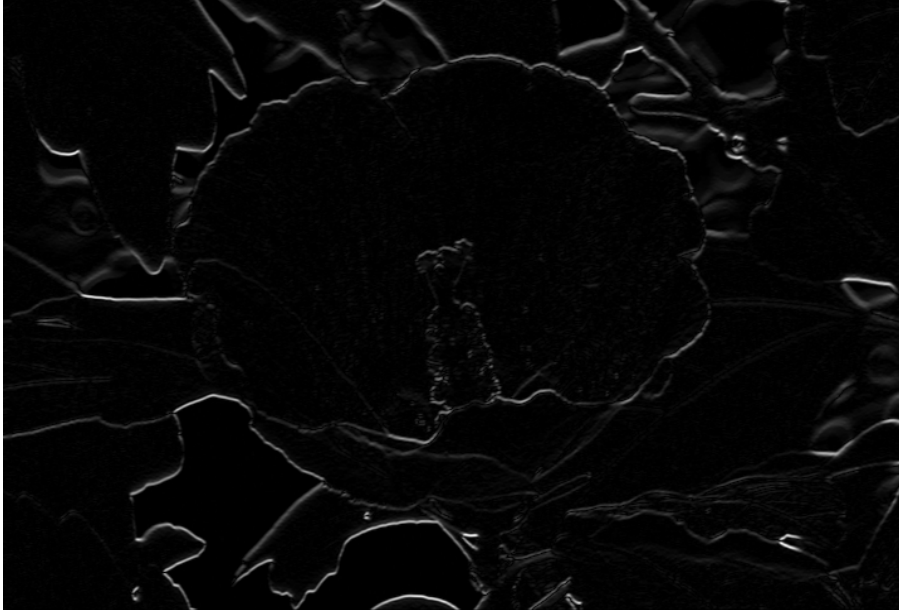
或者，人们也可采用经过修订的程序寻找边缘像素。例如，人们可以首先对源图像应用垂直梯度算子，获得垂直梯度图像。然后，对垂直梯度图像应用水平梯度算子，获得经过修正的连续梯度图像（水平和垂直梯度图像）。最后，可对经过修正的连续梯度图像进行门限运算，寻找边缘像素。也就是说，超过某个门限值的经过修正的连续梯度图像的像素视为边缘像素。图9至图12显示了经过修正的程序。图9显示了通过对图4的源图像应用垂直梯度算子后获得的垂直梯度图像 $g_{vertical}(m,n)$ 。图10显示了通过对图9的垂直梯度图像应用水平梯度算子后获得的经修正连续梯度图像（水平和垂直梯度图像）。图11显示了通过对图10的经修正连续梯度图像应用门限后获得的二进制边缘图像（掩模图像）。

应注意到，两种方法均可视为一种边缘探测算法。人们可以根据视频的性质和压缩算法选择任意一种边缘探测算法。但是，某些方法的效果可能要优于其它方法。

由此，在模型中首先应用边缘探测算子，获得边缘图像（见图7和图10）。然后，通过对边缘图像应用门限获得掩模图像（二进制边缘图像）（见图8和图11）。也就是说，数值小于门限 t_e 的边缘图像的像素设为零且数值等于或大于门限的像素设为非零的值。图8和图11显示了一些掩模图像。由于视频可视为帧或画面的序列，上述程序可适用于视频的每一帧或每一个画面。由于模型可用于以画面为基础的视频或以帧为基础的视频，将采用“图像”（image）的术语来表示一帧或一个画面。

图 9

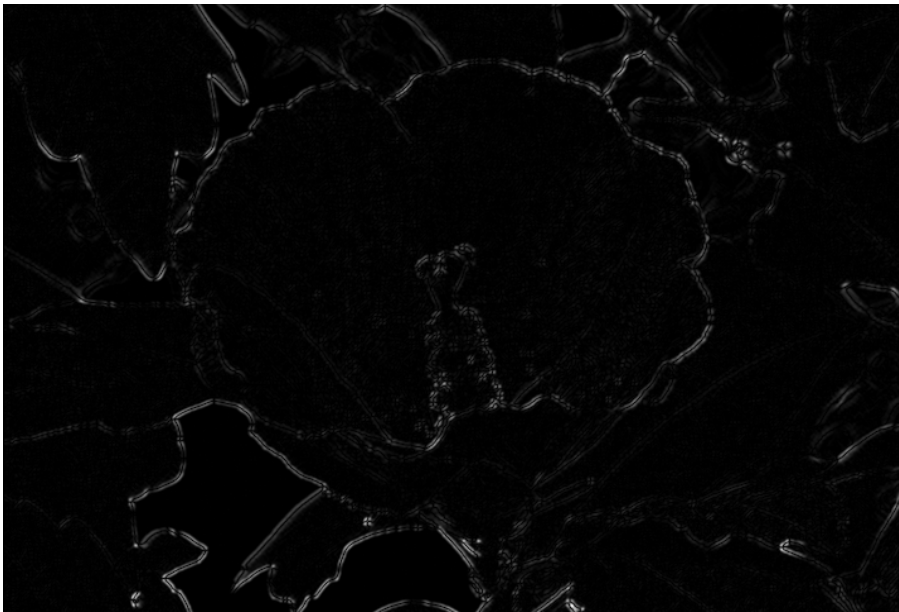
通过对图4源图像应用垂直梯度算子后
获得的垂直梯度图像



BT.1885-09

图 10

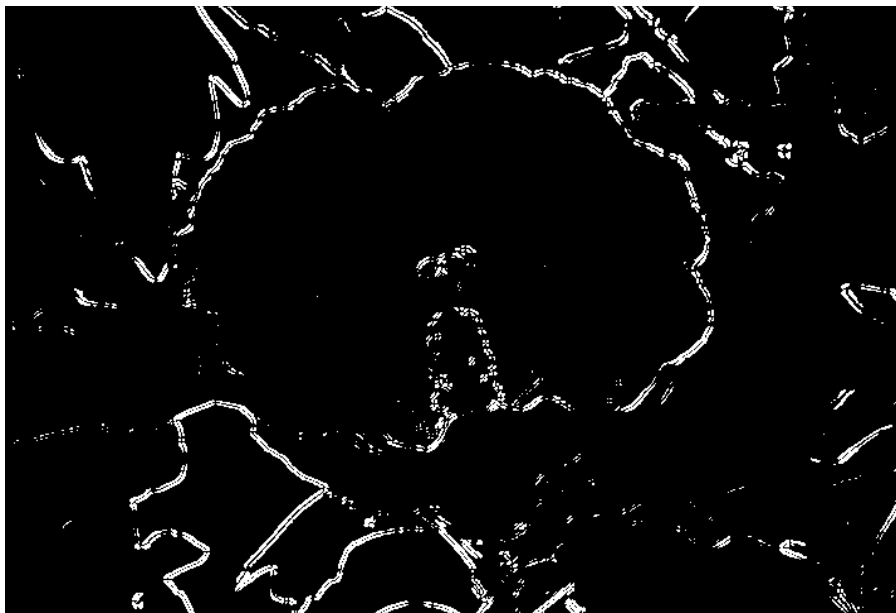
通过对图9的垂直梯度图像应用水平梯度算子后
获得的修正连续梯度图像（水平和垂直梯度图像）



BT.1885-10

图 11

通过对图10的修正连续梯度图像应用门限后
获得的二进制边缘图像（掩模图像）



BT.1885-11

2.2 从源视频序列中选择特征

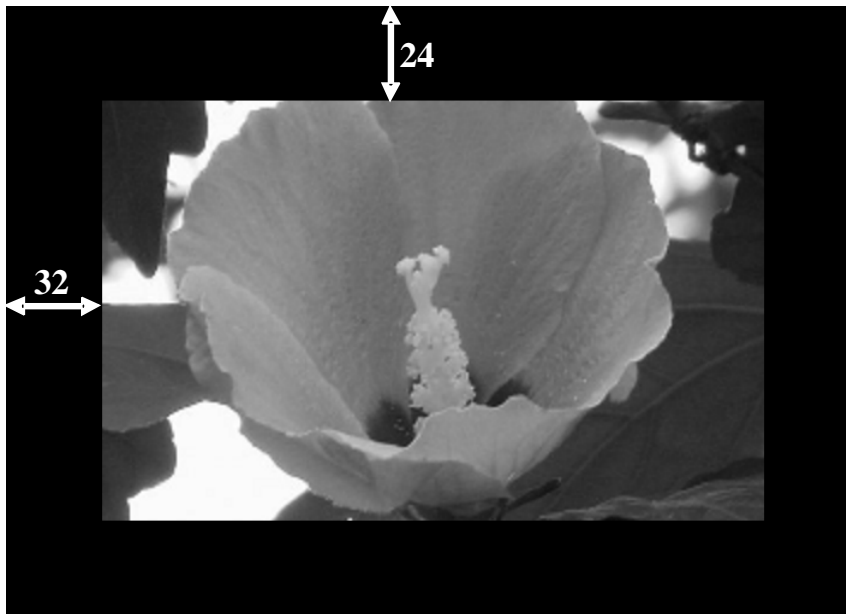
既然模型是一种RR模型，需要从源视频序列的每一个图像中提取一系列特征。在EPSNR RR模型中，从每个图像选取一定数量的边缘像素。然后，位置和像素值进行编码和传输。但是，对于某些视频序列，当采用固定的门限值时，边缘像素的数量可能非常少。在最差情况下，可能为零（空白图像或频率很低的图像）。为了解决该问题，如果一个图像的边缘像素数量小于某个给定值，用户可以降低门限值，直至边缘像素的数量大于某个给定值。或者，人们可以选择与水平和垂直梯度图像最大值相对应的边缘像素。当在一帧中没有边缘像素（如空白图像）时，可以随机选择所需的像素数量或跳过该帧。例如，如果需要从每帧中选取10个边缘像素，人们可以根据水平和垂直梯度图像的值对其进行排序并选择最大的10个值。但是，该程序可能会在相同的位置产生多个边缘像素。为解决该问题，可以首先选择数次水平和垂直梯度图像所需数量的像素，然后在选定的水平和垂直梯度图像像素中选择所需数量的边缘像素。在VQEG RRNR-TV测试进行测试的模型中，在大量边缘像素中随机选择所需数量的边缘像素。通过对梯度图像应用门限运算获得大量边缘像素。

在EPSNR RR模型中，在对选定像素位置应用高斯低通滤波器（LPF）之后对位置和边缘像素值进行编码。尽管在VQEG RRNR-TV测试中采用了高斯LPF (5×3)，根据视频格式的不同也可采用不同的低通滤波器。注意到在编码过程中，可能会进行剪切。为了避免在被剪切的区域内选择边缘像素，模型在中间区域选取边缘像素（见图12）。表6显示了剪切之后的大小。表6还显示了对一个边缘像素的位置和像素值进行编码所需的比特数量。

表 6
每个边缘像素的比特要求

视频格式	大小	剪切后的大小	位置比特	像素值比特	每像素的总比特
525	720 × 486	656 × 438	19	8	27
625	720 × 576	656 × 528	19	8	27

图 12
剪切和中间区域的示例



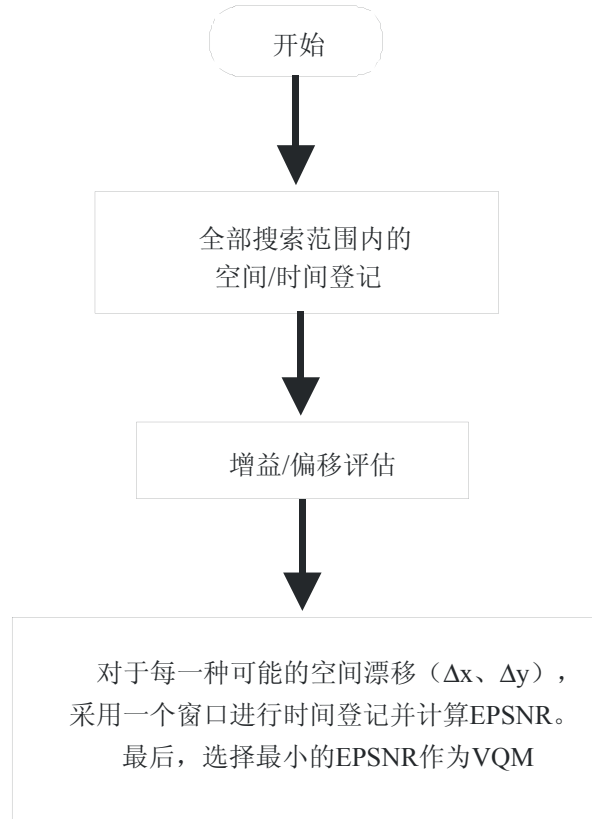
BT.1885-12

模型根据允许的带宽从每帧中选取边缘像素（见表5）。表7显示了可传输用于测试带宽的每帧边缘像素数量。

表 7
每帧边缘像素的数量

视频格式	15 kbit/s	80 kbit/s	256 kbit/s
525	16	74	238
625	20	92	286

图 13
模型的流程图



BT.1885-13

2.3 空间/时间登记和增益/偏移调整

在计算源视频序列的边缘像素与经处理视频序列（即在接收机端收到的视频序列）边缘像素之间的差异之前，模型首先进行空间/时间登记和增益/偏移调整。采用了ITU-T J.244建议书（附件B）的校准方法。要传输ITU-T J.244建议书（附件B）的增益和偏移特征，在VQEG RRNR-TV测试中使用了30%的可用带宽。由于视频序列是交织的，采用了三次的校准方法：在偶数半帧、奇数半帧和组合帧中。如果偶数半帧误差（PSNR）和奇数半帧误差之间的差别大于某个门限值，即采用PSNR较小的注册结果（x-shift、y-shift）。否则，则采用组合帧的注册结果。在VQEG RRNR-TV测试中，门限设为2 dB。

在监测点，经处理视频序列应与从源视频序列提取的边缘像素相校正。但是，如果邻信道带宽较小，只能使用源视频序列的一些边缘像素（见图14）。因此，如果采用单帧进行空间注册，那么空间注册就可能不准确（见图15）。为解决该问题，模型采用用于时间登记的窗口。模型构建了包含多个相邻帧，以寻找最优时间漂移的窗口，而不是采用经处理视频序列的单一帧。图16说明了此程序。按照以下公式计算窗口内的平均方根误差：

$$MSE_{window} = \frac{1}{N_{win}} \sum (E_{SRC}(i) - E_{PVS}(i))^2$$

其中：

MSE_{window} ：窗口平均方根误差

$E_{SRC}(i)$ ：窗口内在经处理视频序列中有对应像素的边缘像素

$E_{PVS}(i)$ ：与边缘像素对应的经处理视频序列的一个像素

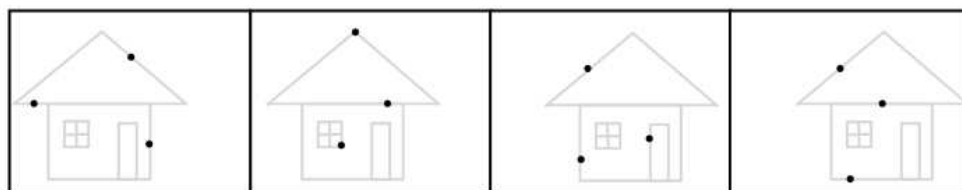
N_{win} ：用来计算 MSE_{window} 的边缘像素总数。

该窗口平均方根误差用作经处理视频序列的一帧与原视频序列对应帧之间的差别。

可以通过考虑经处理视频序列的属性来决定窗口大小。对于典型的应用，建议为二秒的窗口。或者，可以采用各种大小的窗口且可以采用提供最小均方根误差的最优结果。此外，可使用不同的窗口中心来考虑由于传输误差而导致的跳帧（见图20）。

图 14

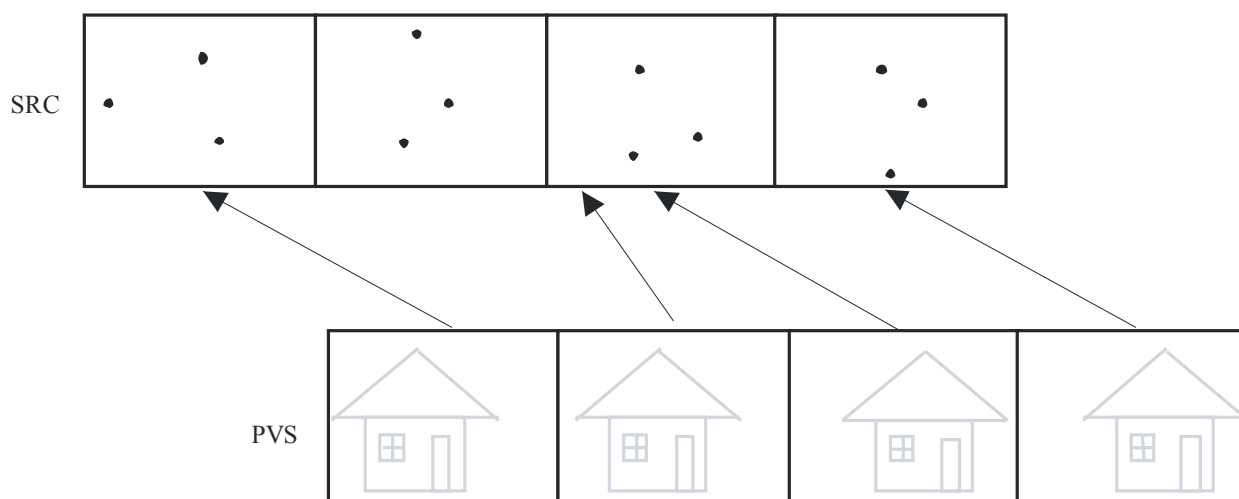
源视频序列边缘像素的选择



BT.1885-14

图 15

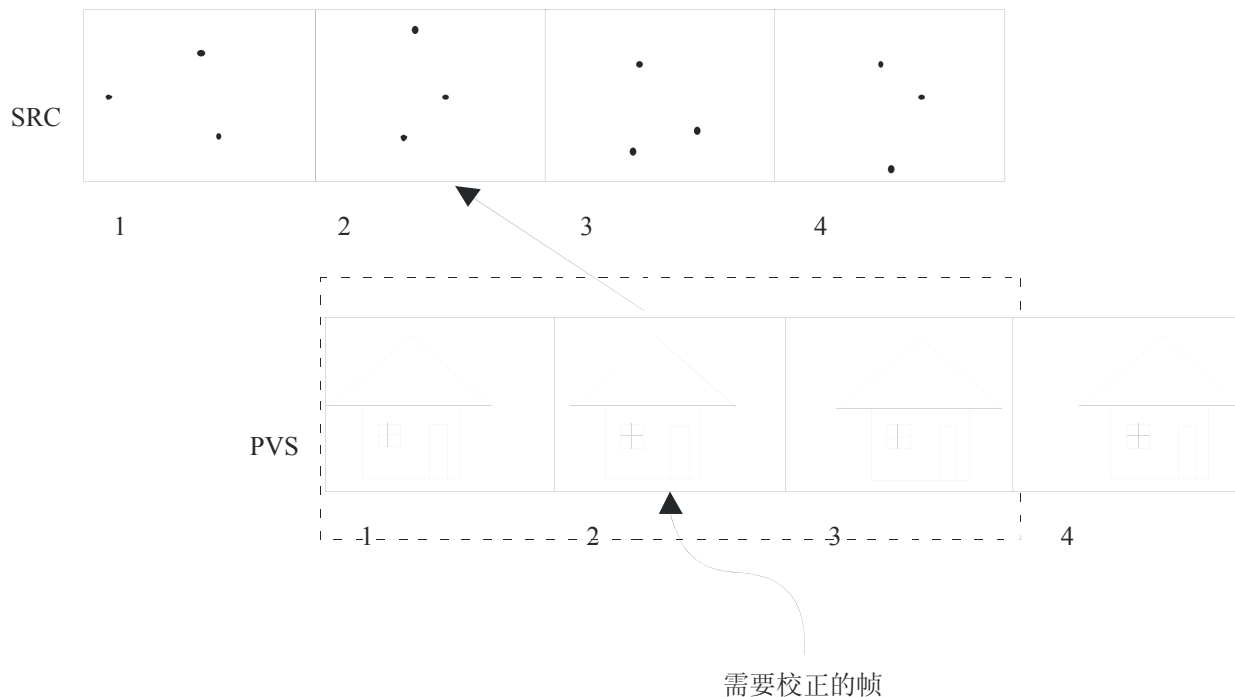
将经处理视频序列与源视频序列的边缘像素相校正



BT.1885-15

图 16

采用一个窗口将经处理视频序列与边缘像素相校正

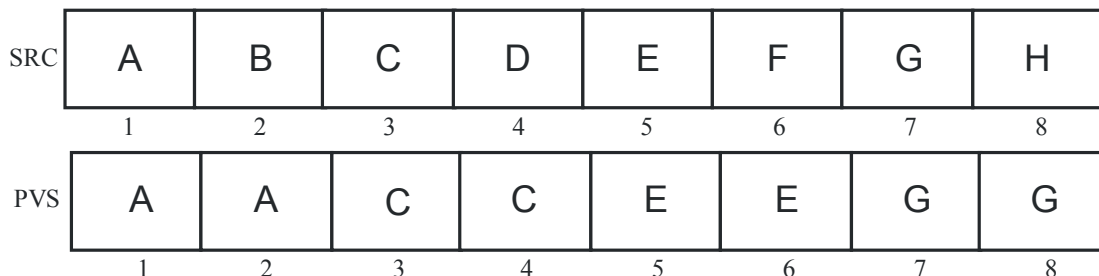


BT.1885-16

当源视频序列用高压缩比进行编码时，编码器可能会降低每秒的帧数且经处理视频序列有重复帧（见图17）。在图17中，经处理视频序列没有与源视频序列某些帧（第2、4、6、8帧）相对应的帧。在这种情况下，模型在计算平均方根误差时不使用重复帧。也就是说，模型采用每个重复块的第一帧（有效帧）进行时间登记。因此，在图18中，只使用了窗口中的三帧（第3、5、7帧）用于时间注册。

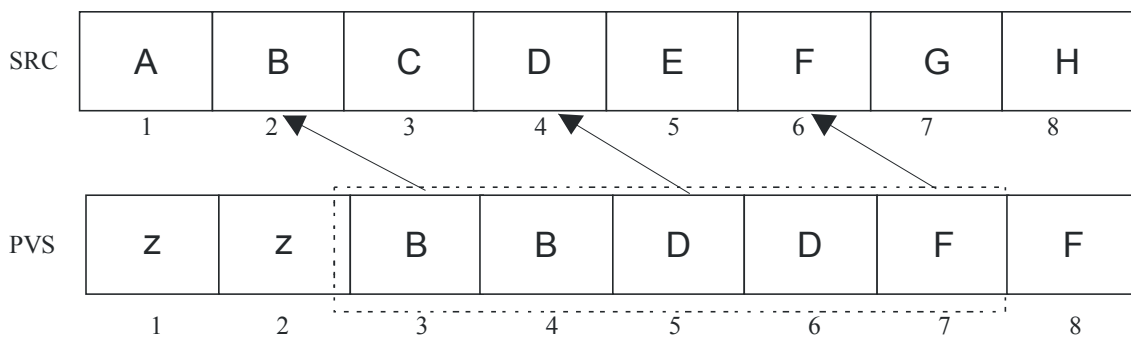
图 17

重复帧的示例



BT.1885-17

图 18
重复帧的转交



BT.1885-18

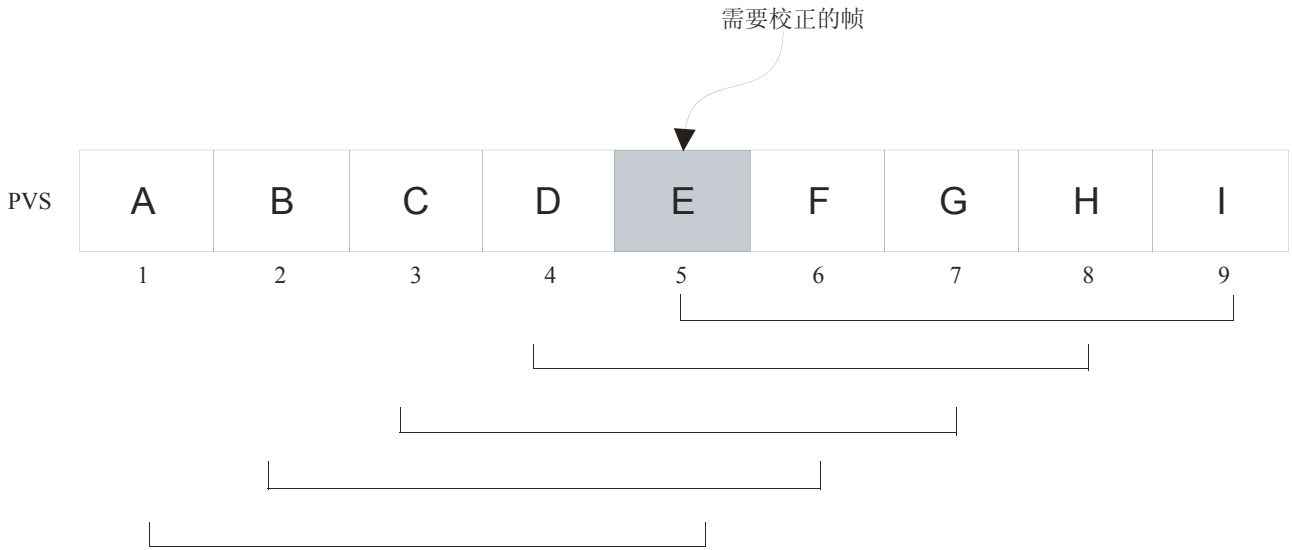
有可能使经处理视频序列具有不规则的帧重复，这可能导致采用窗口的时间登记方法产生不准确的结果。为解决该问题，如图21所示，在采用窗口进行时间登记之后，有可能在给定值的范围内（如 ± 1 ）本地调整每一帧。然后，采用提供最小MSE的本地调整来计算EPSNR。

图 19
各种大小的窗口



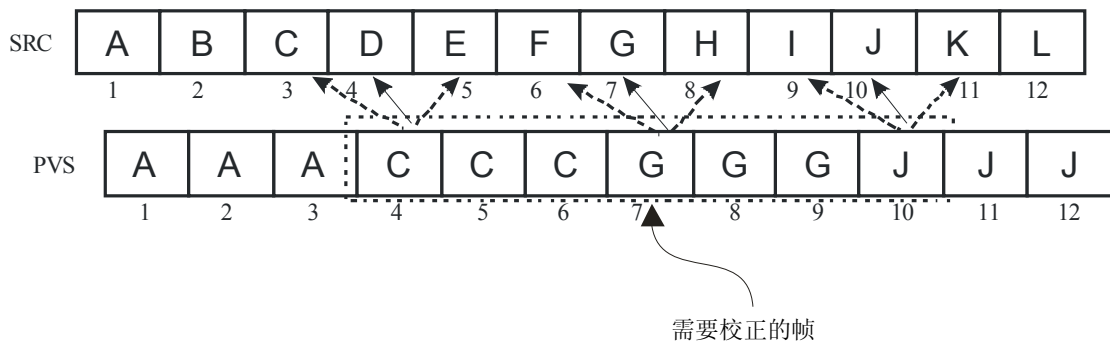
BT.1885-19

图 20
窗口中心



BT.1885-20

图 21
采用窗口的时间注册的本地调整



BT.1885-21

2.4 计算EPSNR和后处理

在执行时间注册之后，计算源视频序列边缘像素与经处理视频对应像素之间的平均差异，这可以理解为经处理视频序列的边缘均方根误差 (MSE_{edge})。最后，按照以下公式计算EPSNR (边缘PSNR)：

$$EPSNR = 10 \log_{10} \left(\frac{P^2}{MSE_{edge}} \right)$$

其中 p 为图像的峰值。

1 帧凝固

由于帧速率降低，可能出现帧重复且由于传输误差，可能出现帧凝固，这将降低感知视频质量。为了解决这种效应，模型在计算EPSNR之前应用以下的调整：

$$MSE_{frozen_frame_considered} = MSE_{edge} \times \frac{K \times N_{total_frame}}{N_{total_frame} - N_{total_frozen_frame}}$$

其中：

$MSE_{frozen_frame_considered}$: 将帧重复和帧凝固考虑在内的均方根误差

N_{total_frame} : 帧总数, $N_{total_frozen_frame}$

K : 常数。

在VQEG RRNR-TV测试里测试的模型中, K 设为1。

2 高频和快速运动

如果视频序列包含大量高频和快速运动, 对于相同的MSE, 感知质量趋向于增加。为考虑这种效应, 以下定义了归一化的帧差异 (NFD) 和归一化的高频能量 (NHFE) :

$$NFD = \frac{\text{帧差异}}{\text{每像素的平均能量}}$$

其中 $FD = \frac{1}{N_F} \sum_i \sum_{k=1}^{height} \sum_{j=1}^{width} (Frame_i[j,k] - Frame_{i-1}[j,k])^2$ 且 N_F 为求和中采用的帧数量。注意到为了在计算平均帧差异时排除场景变化, 假定了八秒的视频序列, 在计算FD时排除了三个最大的帧差异。在采用了2D傅立叶变换之后, 通过计算高频能量 (见图22) 来计算归一化的高频能量 (NHFE) :

$$NHFE = \frac{\text{平均高频能量}}{\text{每像素的平均能量}}$$

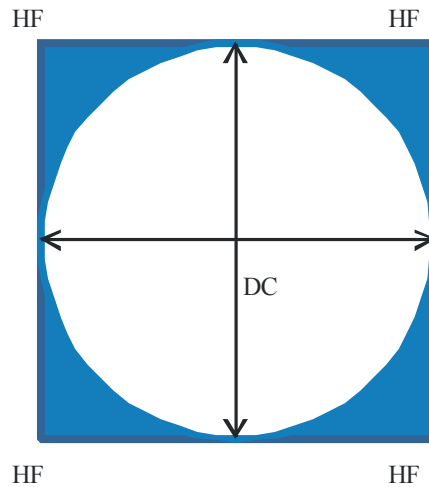
最后, 采用了以下公式:

```
IF(SNFD > 0.35 && SNHFE > 2.5) {
    IF(EPSNR < 20) EPSNR = EPSNR+3
    ELSE IF(EPSNR < 35) EPSNR = EPSNR+5
}
ELSE IF((SNFD > 0.2 && SNHFE > 1.5) || (SNFD>0.27) && SNHFE > 1.3)) {
    IF(28 < EPSNR < 40) EPSNR = EPSNR + 3
    IF(EPSNR > 40) EPSNR = 40
}
```

其中SNFD为源NFD且SNHFE为源NHFE。注意到SNFD和SNHFE由SRC计算得来且作为特征数据的一部分进行传输 (每个1字节)。

图 22

计算归一化的高频能量 (NHFE)。
高频能量由阴影区域计算获得



BT.1885-22

3 模糊

为了考虑模糊效应，采用了以下公式：

```

IF (NHFE/SNHFE < 0.5)
    IF (EPSNR > 26)      EPSNR = 26
ELSE IF (NHFE/SNHFE < 0.6)
    IF (EPSNR > 32)      EPSNR = 32
ELSE IF (NHFE/SNHFE < 0.7)
    IF (EPSNR > 36)      EPSNR = 36
ELSE IF (NHFE/SNHFE > 1.2)
    IF (EPSNR > 23)      EPSNR = 23
ELSE IF (NHFE/SNHFE > 1.1)
    IF (EPSNR > 25)      EPSNR = 25

```

其中NHFE为PVS NHFE。

4 遮挡

为考虑遮挡效应，计算了平均列差异。假定了8模，按照以下公式计算第*i*个帧的遮挡分数：

$$Blk[i] = \frac{\text{最大的列差异}}{\text{第二大列差异}}$$

通过平均帧遮挡分数来计算最终的遮挡 (*Blocking*) 分数：

$$\text{遮挡} = \frac{1}{\text{帧数}} \sum_i Blk[i]$$

最终，采用以下公式：

```

IF (BLOCKING > 1.4) {
    IF (20 ≤ EPSNR < 25) EPSNR = EPSNR - 1.086094 * BLOCKING - 0.601316
    ELSE IF (EPSNR < 30) EPSNR = EPSNR - 0.577891 * BLOCKING - 3.158586
    ELSE IF (EPSNR < 35) EPSNR = EPSNR - 0.223573 * BLOCKING - 3.125441
}

```


5 最大凝固帧

传输误差可能会导致长时间的凝固帧。要考虑长时间凝固的帧，采用了以下公式：

```
IF(MAX_FREEZE > 22 AND EPSNR>28) EPSNR = 28
ELSE IF(MAX_FREEZE > 10 AND EPSNR>34) EPSNR = 34
```

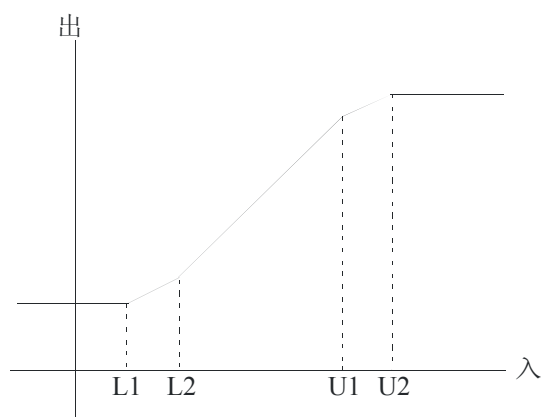
其中MAX_FREEZE是凝固帧的最长持续时间。注意到如果视频序列不是8秒，应采用不同的门限。

6 分段线性拟合

当EPSNR超过特定值时，感知质量就饱和了。在这种情况下，有可能设定EPSNR的上限。而且，当需要EPSNR和DMOS（差值平均主观评分）之间的线性关系时，人们可以应用图23所示的一种分段线性方程。在VQEG RRNR-TV测试的模型中，上限设为48且下限设为15。

图 23

EPSNR和DMOS之间
线性关系的分段线性方程



BT.1885-23

用于视频质量客观测量的EPSNR减少参考模型基于边缘退步。模型可实时采用中等的计算能力予以实施。模型非常适合具有邻信道且需要进行实时视频质量监控的应用。

附件 B

模型B：NEC减少参考方法

本附件提供了RR模型的全功能描述。在RR模型中，向客户端传输的是活动值，而不是单个给定尺寸像素块的像素值。在源参考信道（SRC）和经处理视频序列（PVS）之间活动差异的基础上评估视频质量。与活动差异有关的视觉加权将提高预测的准确率。

此模型不需要很大计算量的空间和增益与偏移注册。而且，可分别在服务器和客户端30线节目和250线节目上进行实施。因此，它适合用于可从低复杂性和易于实施方面获益最多的广播业务中的实时视频质量监控。

1 摘要

在RR模型中，向客户端传输的是活动值，而不是单个给定尺寸像素块的像素值。在源参考信道（SRC）和经处理视频序列（PVS）之间活动差异的基础上评估视频质量。与活动差异有关的视觉加权将提高预测的准确率。

此模型不需要很大计算量的空间和增益与偏移注册。而且，可分别在服务器和客户端30线节目和250线节目上进行实施。因此，它适合用于可从低复杂性和易于实施方面获益最多的广播业务中的实时视频质量监控。

2 定义

活动（Activity） – 每个亮度值与给定尺寸块的平均亮度值之间的平均绝对差别。

块（Block） – $M \times N$ （M列N行）的像素阵列。

帧（Frame） – 一个完整的电视图像。

增益（Gain） – 假定参考电路（HRC）适用于单个平面像（如亮度、色度）所有像素的乘法换算系数。亮度信号的增益通常称为对比度。

假定参考电路（HRC） – 编解码器或数字视频传输系统等被测试的视频系统。

亮度（Y） – 视频信号中主要承载亮度信息（即图像的黑色和白色部分）的部分。

全球电视制式委员会（NTSC） – 525线模拟彩色视频复合系统。

偏移或电平偏移（Offset或level offset） – 假定参考电路（HRC）对单个平面像（如亮度、色度）所有像素适用的一种附加因子。亮度信号的偏移通常称为亮度。

峰值信噪比 (PSNR) – 信号的最大可能功率与破坏噪音功率之间的比率。

相位变化线 (PAL) – 625线模拟彩色视频复合系统。

光栅扫描 (Raster scan) – 一种二维矩形图像映射为一维图像的方法，这样一维图像的第一个输入来自于从左向右扫描的二维图像的第一顶行，图像的第二、三等行与此类似地继续下去（向下），每行从左向右扫描。

减少参考 (RR) – 一种采用从原始或经处理视频流中提取的低带宽特征的视频质量测量方法，与采用全参考视频并要求完全了解原始和经处理视频流的方法相反。减少参考方法对于端到端进行中的质量监控具有优势，因为减少参考的信息可以容易的在无所不在的电信网络中传输。

感兴趣区域 (ROI) – 一种用来表示画面或视频帧特定子区域的图像点阵（以矩形坐标确定）。

场景 (Scene) – 一系列视频帧。

空间注册 (Spatial registration) – 用来评估并校正经处理视频序列相对于原始视频序列的空间偏移的一种程序。

时间注册 (Temporal registration) – 用来评估并校正经处理视频序列相对于原始视频序列的时间偏移（即视频时延）的一种方法。

视频质量量度 (VQM) – 一套综合视频衰减的测量方法。VQM作为一个单独的数量进行报告且标称输出范围从零到一，其零表示没有感知衰减且一表示最大的感知衰减。

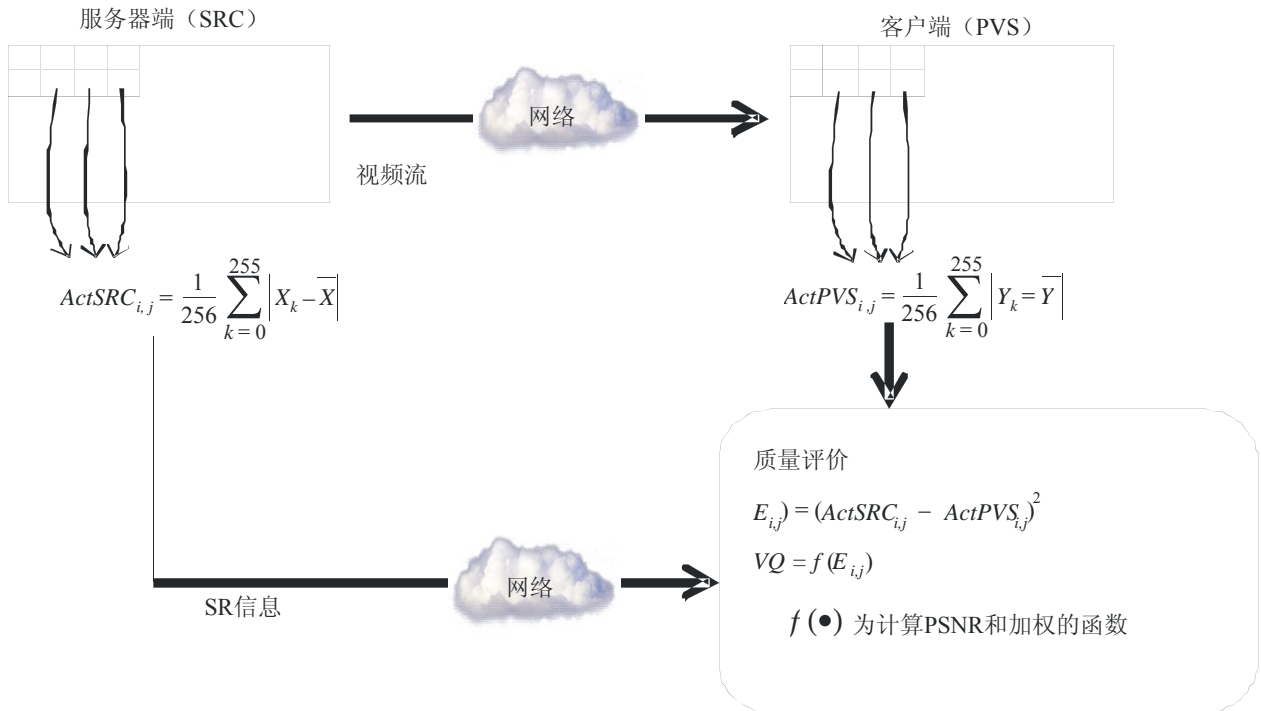
3 视频质量量度计算综述

RR模型将单个给定尺寸像素块的活动值传输给客户端。该值表示块中亮度值的变化。图1总结了RR模型。如图24所示，在SRC和PVS之间活动差异的基础上评估视频质量。此外，对更高评估精确度采用了与活动差异有关的视觉加权。按照以下步骤进行视频质量评估：

- 1 在服务器端计算SRC每个 16×16 亮度像素块的活动值。然后将所有活动值传送给客户端。块的活动值定义为单个数值及其平均值之间的平均绝对差别。
- 2 在客户端计算与PVS有关的对应活动值。
- 3 在客户端，首先根据其均方误差评估每个块，即SRC和PVS活动值之间的均方差别。
- 4 在包含高空间频率成分、具体色彩、大型帧间差以及场景变化的块中对均方误差应用视觉加权。
- 5 按照与PSNR计算相同的方式，从加权均方误差的总和中获得临时的视频质量评分。
- 6 修订评分，以反映由于浓淡不均和本地衰减而造成的视觉重大降级。最后，修订的评分代表着RR模型中PVS的测量视频质量。

图 24

在活动差别的基础上进行的视频质量评价



BT.1885-24

4 详细算法

4.1 服务器端

- 1 每帧中，从视频序列的顶部开始，每秒之后将每个帧中SRC的亮度像素划分为 16×16 像素块。在第一个一秒钟，SRC信息不传送，因为人类视觉系统很难在第一帧之后的场景中发现视频质量退步。
- 2 在除帧边缘以外的每一个块中，计算活动值（SRC-活动： $ActSRC_{i,j}$ ）。图25描述了计算并传送其活动值的块。SRC活动按照以下公式计算：

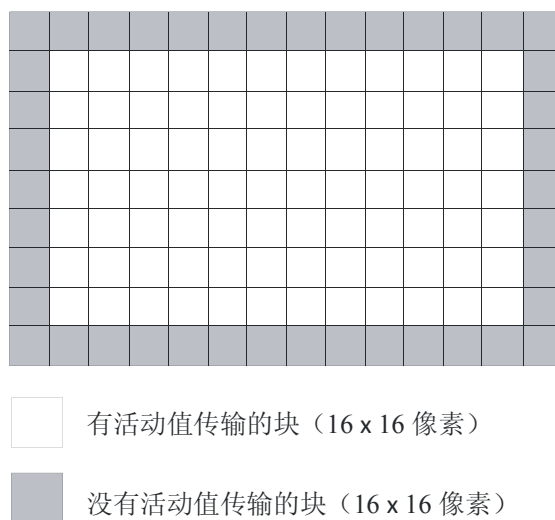
$$ActSRC_{i,j} = \frac{1}{256} \sum_{k=0}^{255} |X_k - \bar{X}|$$

其中 X_k 为SRC给定尺寸块中的亮度值， \bar{X} 为其平均值， i 为帧号， j 为帧中的块编号。

- 3 用每块八个比特表示的活动值按照光栅扫描的顺序，从视频序列顶端的一秒之后开始传送给客户端。对于256 kbit/s速率的SRC信息传输，在所有帧中传输活动值。当速率降低到80 kbit/s时，每隔四帧传送活动值。

图 25

有和没有活动值传输的块



BT.1885-25

4.2 客户端

4.2.1 活动值的均方误差计算

- 1 每帧中，从视频序列的顶部开始，每秒之后将每个帧中PVS的亮度像素划分为 16×16 像素块。
- 2 在除帧边缘以外的每一个块中，计算活动值 (PVS-活动: $ActPVS_{i,j}$)。256 kbit/s速率的SRC信息在所有帧中计算活动值。当SRC信息的速率降为80 kbit/s时，每隔四帧计算活动值。

$$ActPVS_{i,j} = \frac{1}{256} \sum_{k=0}^{255} |Y_k - \bar{Y}|$$

其中

Y_k : PVS给定尺寸块中的亮度值

\bar{Y} : 其平均值

i : 帧编号

j : 帧中块编号。

3. SRC活动和PVS活动之间的均方误差根据以下公式计算

$$E_{i,j} = (ActSRC_{i,j} - ActPVS_{i,j})^2$$

4.2.2 均方误差的视觉加权

三种加权，即空间频率差别的加权、具体色区差别的加权以及亮度帧间差别的加权适用于 $E_{i,j}$ ，以考虑人类视觉系统的特性。

1 空间频率差别的加权

加权系数 W_{SF} 和门限 Th_{SF} 用于此加权。（ W_{SF} 和 Th_{SF} 的数值见表8。）

$$E_{i,j} \leftarrow \begin{cases} E_{i,j} \times W_{SF}, & ActPVS_{i,j} > Th_{SF} \\ E_{i,j}, & \text{其它情况} \end{cases}$$

2 具体色区差别的加权

对于给定块及其周边的八个块，如果像素数量（ $NumROI Pixels$ ）在 $48 \leq Y \leq 224$ 、 $104 \leq Cb \leq 125$ 且 $135 \leq Cr \leq 171$ 大于一个门限，采用加权系数 W_{CR} 和门限 Th_{CR} 来进行以下加权。

$$E_{i,j} \leftarrow \begin{cases} E_{i,j} \times W_{CR}, & NumROI Pixels > Th_{CR} \\ E_{i,j}, & \text{其它情况} \end{cases}$$

W_{CR} 和 Th_{CR} 的值见表8。

3 亮度帧间差别的加权

计算给定块与前一帧中该块之间亮度的平均绝对差别（ $MAD_{i,j}$ ）。 $MAD_{i,j}$ 定义为：

$$MAD_{i,j} = \frac{1}{256} \sum_{k=0}^{255} |Y_k - Y'_k|$$

其中 Y_k 是PVS 16×16 像素块中的亮度值，而 Y'_k 是前一帧中相同位置的亮度值。

采用加权系数 W_{MAD1} 、 W_{MAD2} 和门限 Th_{MAD1} 、 Th_{MAD2} 进行以下加权。

$$E_{i,j} \leftarrow \begin{cases} E_{i,j} \times W_{MAD1}, & MAD_{i,j} > Th_{MAD1} \\ E_{i,j} \times W_{MAD2}, & MAD_{i,j} \leq Th_{MAD2} \\ E_{i,j}, & \text{其它情况} \end{cases}$$

W_{MAD1} 、 W_{MAD2} 、 Th_{MAD1} 和 Th_{MAD2} 的值见表8。

4.2.3 发现场景变化时的加权

每个帧的平均 $MAD_{i,j}$ ($MADAve_i$) 按照以下公式计算：

$$MADAve_i = \frac{1}{M} \sum_{j=0}^{M-1} MAD_{i,j}$$

其中 M 为一帧中块的数量。

如果 $MADAve_i$ 大于门限 Th_{SC} ，则视为发生了场景变化。当发现场景变化时， E_{ij} 设为0，对于场景变化之后的15帧。

$$\text{场景变化} = \begin{cases} \text{真}, & MADAve_i > Th_{sc} \\ \text{假} & \text{其它情况} \end{cases}$$

$$E_{i,j} \Leftarrow \begin{cases} E_{i,j} \times W_{SC} & \text{场景变化15帧之后} = \text{真} \\ E_{i,j}, & \text{其它情况} \end{cases}$$

W_{SC} 和 Th_{SC} 的值见表8。

4.2.4 在活动均方误差基础上的PSNR

根据以下公式，在活动差别的基础上计算PSNR：

$$VQ = 10 \times \log_{10} \frac{255 \times 255}{E_{Ave}}$$

$$E_{Ave} = \frac{1}{N \times M} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} E_{i,j}$$

其中 N 和 M 为用于PSNR计算的帧和块的数量。

4.2.5 浓淡不均伪影的加权

加权系数 W_{BL} 、门限 Th_{BL} 和浓淡不均水平信息 BL_{Ave} 用于该加权。（ W_{BL} 和 Th_{BL} 的值见表8。）

$$VQ \Leftarrow \begin{cases} VQ \times W_{BL}, & BL_{Ave} > Th_{BL} \\ VQ, & \text{其它情况} \end{cases}$$

通过以下步骤计算 BL_{Ave} ：

第1步：计算PVS中的 8×8 像素块的活动值。如图26所示，根据以下公式计算水平相邻块（ $ActBlock_1$ 、 $ActBlock_2$ ）两个活动值的平均值（ Act_{Ave} ）：

$$Act_{Ave} = \frac{1}{2} (ActBlock_1 + ActBlock_2)$$

第2步：计算两个块之间沿着边界的亮度值的绝对差别。如图26所示， $Y_{1,0}$ 和 $Y_{2,0}$ 表示左右块沿着边界的亮度值。绝对亮度差别的平均值 $DiffBound$ 表示为以下公式：

$$DiffBound = \frac{1}{8} \sum_{i=0}^7 |Y_{1,i} - Y_{2,i}|$$

第3步：浓淡不均水平（ $BL_{i,j}$ ）定义为 $DiffBound$ 和 Act_{Ave} 之间的比值，即：

$$BL_{i,j} = \frac{DiffBound}{Act_{Ave} + 1}$$

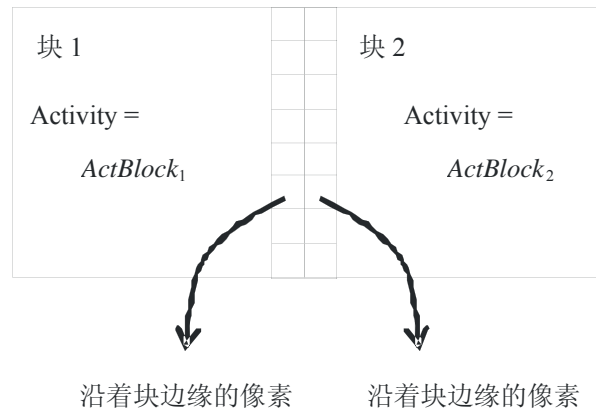
第4步：通过以下公式计算 BL 的平均值：

$$BL_{Ave} = \frac{1}{N \times M} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} BL_{i,j}$$

对于帧当中最右边的块， $BL_{i,j}$ 值设为零。如果 BL_{Ave} 大于一个预设的门限，则认为视频序列包括很大水平的浓淡不均并对计算的视频质量值应用加权。

图26

用于计算浓淡不均水平的像素和活动值



$$(Y_{1,0}, Y_{1,1}, Y_{1,2}, Y_{1,3}, Y_{1,4}, Y_{1,5}, Y_{1,6}, Y_{1,7}) \quad (Y_{2,0}, Y_{2,1}, Y_{2,2}, Y_{2,3}, Y_{2,4}, Y_{2,5}, Y_{2,6}, Y_{2,7})$$

BT.1885-26

4.2.6 本地衰减伪像的加权

该加权采用了加权系数 W_{LI} 、门限 Th_{LI} 和本地衰减 LI 。（ W_{LI} 和 Th_{LI} 的值见表8。）

$$VQ \leftarrow \begin{cases} VQ \times W_{LI}, & LI < Th_{LI} \\ VQ & \text{其它情况} \end{cases}$$

LI 通过以下步骤计算。采用活动变化的差别来发现传输误差造成的本地衰减。

- 1 对于给定块及其周边的八个块，计算SRC ($ActVar_{SRC}$) 和PVS ($ActVar_{PVS}$) 的活动变化且这些变化值的绝对差别通过下式计算：

$$\Delta ActVar = |ActVar_{SRC} - ActVar_{PVS}|$$

- 2 计算每帧的该绝对差别值的平均值。
- 3 LI 作为平均的最大值 ($\Delta ActVar_{Max}$) 与最小值 ($\Delta ActVar_{Min}$) 的比率进行计算。

$$LI = \begin{cases} \Delta ActVar_{Min} / \Delta ActVar_{Max} & \Delta ActVar_{Max} \neq 0 \\ 1 & \Delta ActVar_{Max} = 0 \end{cases}$$

VQ 表示视频质量评分。

4.2.7 加权的参数

表8显示了加权的参数值。这些数值由实验数据集的初步实验决定。

表8
加权的参数

加权运算类型	参数值	
空间频率加权	W_{SF}	0.36
	Th_{SF}	25
特定色彩加权	W_{CR}	4.0
	Th_{CR}	175
帧间差别加权	W_{MAD1}	0.06
	Th_{MAD1}	17
	W_{MAD2}	25
	Th_{MAD2}	13
场景变化探测	W_{SC}	0.0
	Th_{SC}	35
浓淡不均加权	W_{BL}	0.870
	Th_{BL}	1.0
本地衰减加权	W_{LI}	0.870
	Th_{LI}	1.67

4.2.8 注册

1 空间注册

RR模型并不需要任何空间注册，这是因为均方误差是在与基于像素值基础上的空间偏移相比更加强健的活动值的基础上计算的。

2 增益和偏移注册

RR模型并不需要增益和偏移注册。活动值内在地没有偏移（即DC分量）并对增益不敏感。

3 时间注册

PVS序列划分为一秒的子序列。对于每个子序列，根据最高至 ± 2 SRC帧的五个时延变化计算活动的均方根误差。最后将均方根误差的最小值作为此子序列的均方根误差。导致此最小均方根误差的时延调整作为时间登记。

5 样本代码

此处提供了RR模型的C语言样本代码。

5.1 服务器和客户端的通用代码

```
// Calculate the activity value
unsigned int CalcActivitybyRect(unsigned char * lpBuffer, int nWidth, int iRectWidth, int iRectHeight)
{
    // lpBuffer: Luminance Frame Buffer
    // nWidth: Frame Buffer Width
    // iRectWidth: Width of the rectangle to calculate an activity value.
    // iHeightWidth: Height of the rectangle to calculate an activity value.
    unsigned int i, j, nTmp, nSum;
    unsigned char *pSrc;

    pSrc = lpBuffer; nSum = 0;
    for (j = 0; j < iRectHeight; j++){
        for (i = 0; i < iRectWidth; i++){
            nSum += pSrc[i];
        }
        pSrc += nWidth;
    }
    nSum /= (iRectWidth*iRectHeight);

    pSrc = lpBuffer; nTmp = 0;
    for (j = 0; j < iRectHeight; j++){
        for (i = 0; i < iRectWidth; i++){
            nTmp += abs(pSrc[i] - nSum);
        }
        pSrc += nWidth;
    }
    return nTmp/iRectWidth/iRectHeight;
}
```

5.2 服务器端

```
// Server side
int nStart = 30; // the frame number to start transmission (30 or 25)
int nMaxFrame = 240; // the number of total video frames (240 or 200)
int nFrameIncrement = 1; // 1 for 256kbps, 4 for 80kbps
void ReadOneFrame(unsigned char, int, unsigned char *, int, int); // function to read one frame data
int nRim = 16 // 16 or 32 (use 32 to avoid the problem in HRC9)

// nWidth: Frame Buffer Width
// nHeight: Frame Buffer Height
// lpSrc: Frame Buffer
for(int nFrame = nStart; nFrame < nMaxFrame; nFrame+=nFrameIncrement){
    ReadOneFrame(SRC_file_name, nFrame, lpSrc, nWidth, nHeight);
    for (j= 16; j<nHeight-32; j+=16) {
        for (i= nRim; i<nWidth- nRim; i+=16) {
            lpOrg = lpSrc + i + j * nWidth;
            nActSrc = CalcActivitybyRect(lpOrg, nWidth, 16, 16);
            // OutputSRCInfo(nActSrc); // Output or transmission the SRC information
        }
    }
}
```

5.3 客户端

```
// Client Side
int nStart = 30; // the frame number to start transmission (30 or 25)
int nMaxFrame = 240; // the number of total video frames (240 or 200)
int nFrameIncrement = 1; // 1 for 256kbps, 4 for 80kbps
int nFrameRate = 30; //30 or 25
void ReadOneFrame(unsigned char, int, unsigned char **, int, int); // function to read one frame data
void ReadRRData(unsigned char, int, unsigned char *); // function to read RR-data

// nWidth: Frame Buffer Width
// nHeight: Frame Buffer Height
// lpPvsByte[3]: Frame Buffer (0:Y, 1:Cb, 2:Cr)
```

```

// lpRRData: RR-data Buffer
// double ddActivityDifference[]: Store the activity-difference
// double ddActivityVariance[]: Store the activity-variance
// double ddBlock[]: Store the blockiness level
// int nSceneChange: Scene change detection

for(int nTemporalAlign = -2; nTemporalAlign <=2; nTemporalAlign++){ // Changing temporal alignment
    for(int nFrame = 0; nFrame < nMaxFrames; nFrame++){
        if(nFrame+nTemporalAlign >= nMaxFrames || nFrame+nTemporalAlign < 0){
            continue;
        }
        ReadOneFrame(PVS_file_name, nFrame+nTemporalAlign, lpPvsByte, nWidth, nHeight);
        if(((nFrame-(nFrameRate+nStart)) % nFrameIncrement) == 0
            && nFrame >= nStart ){
            ReadRRData(RR_file_name, nFrame, lpRRData);
            ddActivityDifference[nTemporalAlign+2][nFrame]
                = RRCalcObjectiveScore(lpPvsByte, lpRRData, nWidth, nHeight);
            ddActivityVariance[nTemporalAlign+2][nFrame] = gnActVar;
        }else{
            ddActivityDifference[nTemporalAlign+2][nFrame] = 0.0;
            ddActivityVariance[nTemporalAlign+2][nFrame] = 0.0;
        }
        // Blockiness Level
        if(nTemporalAlign ==0){
            ddBlock[nFrame] = BlockinessLevelEstimation(lpPvsByte[0], nWidth, nHeight);
        }
        // Pixel copy for inter-frame difference calculation
        memcpy(lpPrev, lpPvsByte[0], sizeof(char)*nWidth*nHeight);
        if(nSceneChange){
            nSceneChange--;
        }
    }
}

double ddSum[8][5]; // Sum of the Activity-difference for each second
double ddActVarSum[8][5]; // Sum of the Activity-variance for each second
double ddActVarMax[8][5]; // Maximum of the Sum of the Activity-variance
double ddActVarMin[8][5]; // Minimum of the Sum of the Activity-variance
int nnMin[8];
int nnNumFrames[8][5];
#define LARGENUMBER 100000
for(int nTemporalAlign = -2; nTemporalAlign <=2; nTemporalAlign++){
    for(int j=0;j<8;j++){ // for each one second
        nnNumFrames[j][nTemporalAlign+2] = 0;
        ddActVarMax[j][nTemporalAlign+2] = 0.0;
        ddActVarMin[j][nTemporalAlign+2] = LARGENUMBER;
        ddActVarSum[j][nTemporalAlign+2] = 0.0;
        ddSum[j][nTemporalAlign+2] = 0.0;
        for(int i=nFrameRate*j;i< (j+1)*nFrameRate; i++){
            if(ddActivityDifference[nTemporalAlign+2][i]){
                ddSum[j][nTemporalAlign+2] += ddActivityDifference[nTemporalAlign+2][i];
                nnNumFrames[j][nTemporalAlign+2]++;
            }
            ddActVarSum[j][nTemporalAlign+2] += ddActivityVariance[nTemporalAlign+2][i];
            if(ddActivityVariance[nTemporalAlign+2][i]){
                if(ddActivityVariance[nTemporalAlign+2][i] >
                    ddActVarMax[j][nTemporalAlign+2]){
                    ddActVarMax[j][nTemporalAlign+2] =
                        ddActivityVariance[nTemporalAlign+2][i];
                }
                if(ddActivityVariance[nTemporalAlign+2][i] <
                    ddActVarMin[j][nTemporalAlign+2]){
                    ddActVarMin[j][nTemporalAlign+2] =
                        ddActivityVariance[nTemporalAlign+2][i];
                }
            }
        }
    }
}
}

```

```

// Local Impairment Level Calculation
double dSum = 0.0;
double dActMax = 0.0;
double dActMin = LARGENUMBER;
int nNumFrames = 0;
for(int j=1; j<8; j++){
    double dMin = LARGENUMBER;
    double dMinSum = LARGENUMBER;
    for(int nTemporalAlign = -2; nTemporalAlign <=2; nTemporalAlign++){
        if(ddActVarSum[j][nTemporalAlign+2] < dMin){
            dMin = ddActVarSum[j][nTemporalAlign+2];
            dMinSum = ddSum[j][nTemporalAlign+2];
            nnMin[j] = nTemporalAlign+2;
        }
    }
    dSum += dMinSum;
    nNumFrames += nnNumFrames[j][nnMin[j]];
    if(ddActVarMax[j][nnMin[j]] > dActMax){
        dActMax = ddActVarMax[j][nnMin[j]];
    }
    if(ddActVarMin[j][nnMin[j]] < dActMin){
        dActMin = ddActVarMin[j][nnMin[j]];
    }
}
double dTransError = dActMax/dActMin;

// Blockiness Level Calculation
double dBlockinessLevel = 0.0;
for(int i=0; i<nMaxFrames; i++){
    dBlockinessLevel += ddBlock[i];
}
dBlockinessLevel = dSumBlock / (double)(nMaxFrames-nFrameRate);

// Calculating the Video Quality Score
if(nNumFrames && nNumberOfBlocks && dSum){
    dSum = dSum / (double)(nNumFrames)/(double)nNumberOfBlocks;
    dSum = 10*log10(255.0*255.0/dSum); //PSNR based on the activity difference
    if(dBlockinessLevel > dBlokinessTh){
        dSum /= nBlockinessWeighting; // Weighting for blockiness level
    }
    if(dTransError > nErrorTh){
        dSum /=nErrorWeightin; // Weighting for transmission errors
    }
}
return dSum;
}

-----
// Calculating the MAD value
unsigned int CalcMAD(unsigned char *lpSrc, unsigned char *lpSrc2, int nWidth, int nHeight)
{
    // lpSrc: Frame Buffer of the current frame
    // lpSrc2: Frame Buffer of the previous frame
    unsigned int nSum = 0;
    for (y = 0; y < nHeight; y++) {
        for (x = 0; x < nWidth; x++) {
            nSrc = lpSrc[x + y*nWidth];
            nSrc2 = lpSrc2[x + y*nWidth];
            nSum += abs(nSrc - nSrc2);
        }
    }
    return nSum/nWidth/nHeight;
}

// Calculating a mean squared error with weightings
double RRCalcObjectiveScore(unsigned char *lpBuffer[], unsigned char *lpRRData, int nWidth, int nHeight)
{
    int i, j, nActSrc, nActSrc2, nY, nCb, nCr, nYMin, nYMax, nCbMin, nCbMax, nCrMin, nCrMax;
    int nMBX, nMBY, nMB, nStart;
    unsigned int nMAD;

```

```

double e2, dMADFrame;
unsigned char *lpRec, *lpRecCb, *lpRecCr, *lpPrev1;
int nRim = 16 // 16 or 32 (use 32 to avoid the problem in HRC9)

nYMin = 48; nYMax = 224; nCbMin = 104; nCbMax = 125; nCrMin = 135; nCrMax = 171;
nMB = nMBY = nStart = 0;
e2 = dMADFrame = 0.0;

for (j=16+nStart; j<ImageHeight-32; j+=16) {
    nMBX = 0;
    for (i= nRim; i<nWidth- nRim; i+=16) {
        lpRec = lpBuffer[0] + i + j * nWidth;
        lpRecCb = lpBuffer[1] + i/2 + (j/2) * nWidth/2;
        lpRecCr = lpBuffer[2] + i/2 + (j/2) * nWidth/2;
        lpPrev1 = lpPrev + i + j * nWidth;

        nActSrc = lpRRData[nMB]; // SRC activity
        nActSrc2 = CalcActivitybyRect(lpRec, nWidth, 0, 16, 16); // PVS activity
        nActArray[nMB] = (double)nActSrc;
        nActArray2[nMB] = (double)nActSrc2;
        e2 += (double)(nActSrc - nActSrc2)*(nActSrc - nActSrc2); // Mean squared error

        nMAD = CalcMAD(lpRec, lpPrev1, 16, 16, nWidth); // Inter-frame difference
        dMADFrame += (double)nMAD;

        int nNumROIPixels=0;
        for(int jj=-16;jj<32; jj++){
            for(int ii=-16;ii<32; ii++){
                nY = lpRec[ii];
                nCb = lpRecCb[ii/2];
                nCr = lpRecCr[ii/2];
                if(nY >= nYMin && nY <= nYMax
                    && nCb >= nCbMin && nCb <= nCbMax
                    && nCr >= nCrMin && nCr <= nCrMax){
                    nNumROIPixels++;
                }
            }
            lpRec += nWidth;
            if((jj & 1) == 1){
                lpRecCb += nWidth/2;
                lpRecCr += nWidth/2;
            }
        }

        // Weighting for spatial frequency
        if(nActSrc2 > gdwActThHigh){
            e2 *= dActWeightingHigh;
        }else if(nActSrc2 > gdwActThLow){
            e2 *= dActWeightingMiddle;
        }else {
            e2 *= dActWeightingLow;
        }

        // Weighting for specific color region
        if(nNumROIPixels > dwROITh){
            e2 *= dROIWeighting;
        }

        // Weighting for inter-frame difference
        if(nMAD > dwMADThHigh){
            e2 *= dMADWeightingHigh;
        }else if(nMAD > dwMADThLow){
            e2 *= dMADWeightingMiddle;
        }else {
            e2 *= dMADWeightingLow;
        }
        nMB++;      nMBX++;
    }
}
nMBY++;

```

```

}

// Calculating Activity-Variance for Surrounding Nine Blocks.
double nSumActSrc, nSumActPvs, nActVar, nActVar2;
nSumActSrc = nSumActPvs = nActVar = nActVar2 = 0.0;
gnActVar = 0.0;
for (j=1; j<nMBY-1; j++) {
    for (i=1; i<nMBX-1; i++) {
        nSumActSrc = 0.0;
        nSumActPvs = 0.0;
        for(int jj=-1; jj<2; jj++){
            for(int ii=-1; ii<2; ii++){
                nSumActSrc += nActArray[i+ii+nMBX*(j+jj)];
                nSumActPvs += nActArray2[i+ii+nMBX*(j+jj)];
            }
        }
        nSumActSrc /= 9.0;
        nSumActPvs /= 9.0;

        nActVar = 0.0;
        nActVar2 = 0.0;
        for(int jj=-1; jj<2; jj++){
            for(int ii=-1; ii<2; ii++){
                nActVar += (nActArray[i+ii+nMBX*(j+jj)]-nSumActSrc)*(nActArray[i+ii+nMBX*(j+jj)]-nSumActSrc);
                nActVar2 += (nActArray2[i+ii+nMBX*(j+jj)]-
                    nSumActPvs)*(nActArray2[i+ii+nMBX*(j+jj)]-
                    nSumActPvs);
            }
        }
        nActVar /= 9.0;
        nActVar2 /= 9.0;
        gnActVar += abs(nActVar- nActVar2);
    }
}

// Average of the Activity-Variance for the Frame
gnActVar = gnActVar/(double)(nMBY-2)/(double)(nMBY-2);

// Scene change detection
if(dMADFrame/ nMB > 35){
    nSceneChange = 15;
}
if(nSceneChange){
    e2 = 0.0;
}
gnFrame++;

return e2;
}

```

```

// Calculate Blockiness Level
double BlockinessLevelEstimation(unsigned char *lpBuffer, int nWidth, int nHeight)
{
    // lpBuffer: Frame Buffer
    int i, j, nActSrc, nActSrc2, nDiff, nMB;
    unsigned char *lpRec = lpBuffer;
    double dBlock=0.0;

    nMB = 0;
    for (j=0; j<nHeight-16; j+=8) {
        for (i=0; i<nWidth-16; i+=8) {
            lpRec = lpBuffer + i + j * nWidth;
            nActSrc = CalcActivitybyRect(lpRec, nWidth, 0, 8, 8); // Activity of the left block
            nActSrc2 = CalcActivitybyRect(lpRec+8, nWidth, 0, 8, 8); // Activity of the right block
            nActSrc = (nActSrc + nActSrc2)/2; // Average of the activity values
            nDiff = 0;
            for(int jj=0; jj<8; jj++){
                nDiff += abs(lpRec[7+jj*nWidth] - lpRec[8+jj*nWidth]);
            }
            // Difference of the luminance values at the boundary

```

```

    }
    nDiff/= 8;
    dBlock += (double)nDiff/(double)(nActSrc+1);
    nMB++;
  }
}
return (double)dBlock/(double)nMB;
}

```

6 资料性参考文献

- [1] SMPTE 170M, “SMPTE电视标准 – 复合模拟视频信号 – 用于演播室应用的NTSC”, 活动图像和电视工程师学会。
- [2] ITU-T J.143建议书 – 数字有线电视中客观感知视频质量测量的用户需求。

附件 C

模型C: NTIA减少参考方法

1 背景情况

在2003-2004年期间, 美国国家电信和信息管理局 (NTIA) 开发了两套视频质量模型 (VQM)。对于ITU-R BT.601建议书的采样视频, 其减少参考 (RR) 带宽约为12至14 kbits/s [1]。这些模型称为“低带宽VQM”和“快速低带宽VQM”。快速低带宽VQM是低带宽VQM在计算上效率更高的版本。快速低带宽VQM大约要快4倍, 因为它从视频帧中提取的空间特征是首先预平均的, 而不是直接从ITU-R BT.601视频帧中提取空间特征。从亮度Y信道的像素中在随机分采样的基础上获得的计算时间信息 (即图像) 特征, 而不是采用所有三个视频信道 (Y、Cb和Cr) 所有像素, 这额外节省了快速低带宽VQM的计算。多年以来, 我们的VQM软件工具中一直都有VQM, 并可免费用于商业和非商业性应用。这些VQM工具的二进制可执行版本及其相关源代码均可下载[2]。

由于NTIA希望将低带宽和快速低带宽VQM用于减少参考电视 (RRTV) 测试, 以便供视频质量专家组 (VQEG) 进行独立评估, 尽管它们的RR比特率要求相同, NTIA还是选择将其提交给不同的比特率类别。NTIA选择将低带宽VQM提交给256k类别, 将快速低带宽VQM提交给80k类别, 因为低带宽VQM的性能预计要高于快速低带宽VQM。两种VQM都使用了NTIA RR校正算法, 此算法包括在ITU-T J.244建议书中[3]。该校正算法需要大约22至24 kbits/s的RR带宽来生成时延、空间偏移、空间换算、增益和电平偏移等预测。

VQEG RRTV评估测试[4]的一项有意思的结果是, 对于525线和625线的测试, 快速低带宽VQM要胜过低带宽VQM。这项结果很有意思, 因为它意味着从平均帧中提取空间特征要优于从非平均帧中提取特征。对于其他数据集而言, 该结果是否也是这样还需要进一步研究。与此同时, NTIA没有理由不将两种模型进行标准化, 所以本附件仅描述了快速低带宽VQM。

2 引言

此附件介绍了NTIA快速低带宽VQM的描述和参考代码。NTIA快速低带宽VQM采用与NTIA一般VQM类似的方法，这些方法的描述可查阅ITU-T J.144建议书[5]和ITU-R BT.1683建议书[6]。快速低带宽VQM采用比NTIA一般VQM带宽少得多的RR特征，但特征提取和比较过程与两者类似。对于ITU-R BT.601采样视频[1]，快速低带宽VQM采用大约需要12至14 kbits/s传输带宽的RR特征。本附件仅描述了快速低带宽VQM，因为其互补性低带宽校正算法完整记录在ITU-T J.244建议书中[3]。但是，出于完整起见，本附件中的参考代码包括了快速低带宽VQM及其相关低带宽校正算法，参考代码也包含低带宽校正所采用的特征的示例性量化运算（这些量化运算不是ITU-T J.244建议书的一部分）。

3 快速低带宽VQM的描述

3.1 VQM综述

VQM描述将包括三个主要方面：

- 1) 从原始和经处理视频流中提取的低带宽特征；
- 2) 从比较原始和经处理特征流获得的参数；
- 3) 合并各种参数的VQM计算，每种参数测量视频质量的不同方面。

此描述使用了可以获得的技术细节的参考。

3.2 特征描述

3.2.1 特征综述

快速低带宽VQM采用了三种特征：色彩、空间和时间。每种特征类型在其各自域内将感知失真量化。参考代码子程序“model_fastlowbw_features”提供了快速低带宽VQM所使用特征的完整数学描述。

3.2.2 色彩特征

色彩特征与NTIA一般VQM所使用的 f_{COHER_COLOR} 特征相同。这些特征详述于ITU-T J.144建议书的附件D.7.3中。 f_{COHER_COLOR} 特征提供了每个S-T区域中蓝色和红色色度信息（ C_B 、 C_R ）数量的二维向量测度。由此， f_{COHER_COLOR} 特征对色彩失真敏感。NTIA快速低带宽VQM的 f_{COHER_COLOR} 特征提取自30垂直线乘以30水平像素乘以1秒时间的空间-时间（S-T）区域大小（即 $30 \times 30 \times 1$ s），而NTIA一般VQM采用的是 $8 \times 8 \times 1$ 帧的S-T区域尺寸。

3.2.3 空间特征

空间特征与NTIA一般VQM所使用的 f_{SI13} 和 f_{HVI13} 特征相同。这些特征详述于ITU-T J.144建议书的附件D.7.2.2中。 f_{SI13} 和 f_{HVI13} 特征测量亮度（Y）图像中空间-时间（S-T）子区域中空间梯度的数量和角分布。由此， f_{SI13} 和 f_{HVI13} 特征对模糊和浓淡不均等空间失真敏感。

NTIA快速低带宽VQM的 f_{SI13} 和 f_{HVI13} 特征提取自30垂直线乘以30水平像素乘以1秒时间（即 $30 \times 30 \times 1s$ ）的空间-时间（S-T）区域尺寸，而NTIA一般VQM采用 $8 \times 8 \times 0.2s$ 的S-T区域尺寸。此外，为减少计算，在应用ITU-T J.144建议书附件D.7.2.1中给定的二维 13×13 边缘增强滤波器之前，亮度Y图像的1s首先在时间上进行预平均。

还从预先平均的亮度（Y）图像中提取一个额外的空间特征。该特征为每个 $30 \times 30 \times 1s$ S-T区域平均亮度（Y）电平（此处标记为 f_{MEAN} ）。 f_{MEAN} 特征的目的是为由 f_{SI13} 和 f_{HVI13} 特征测量的空间信息（SI）损耗提供一个亮度电平感知加权运算。将在参数描述节中描述该问题。

3.2.4 时间特征

可以从上述色彩和空间特征集中获得感知视频质量的强力评估。但是，由于提取这些特征的S-T区域横跨多个视频帧（即视频帧的1s），他们倾向于对图像中的简短时间扰动不敏感。此类扰动可能是噪音或数字传输误差造成的；而且尽管在属性上很简短，但它们也可对感知图像质量产生重大影响。由此，采用以时间为基础的RR特征来将时间扰动的感知效应进行量化。此特征测量亮度Y图像平面中的绝对时间信息（ATI）或活动，并按照以下公式计算：

$$f_{ATI} = rms\{rand5\%|Y(t) - Y(t - 0.2s)|\}$$

对于计算效率，Y随机进行二次采样，以便仅包含5%的图像像素（此处表示为 $rand5\%$ 的函数）。从在时间 t 同等进行二次采样的Y图像扣除在 $t - 0.2s$ 时间二次采样的Y图像且将结果的均方根误差（rms）用作ATI的测量。采用ITU-T J.144建议书附件D.8中的约定，这将表示为：

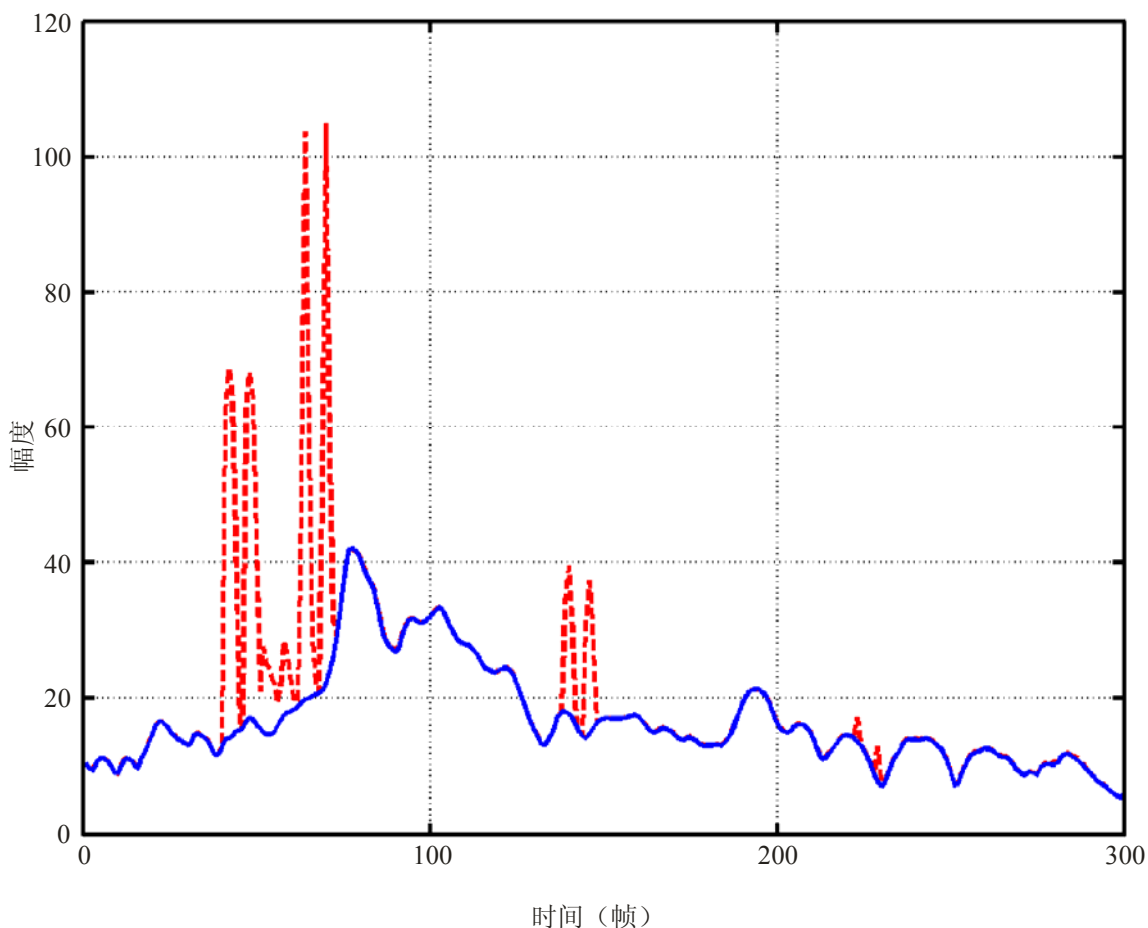
$$f_{ATI} \cong Y_rand5\%_ati0.2s_rms$$

f_{ATI} 特征对时间扰动敏感。对于30 fps的视频，0.2s为6视频帧而对于25 fps视频，0.2s为5视频帧。减去图像0.2s使得特征对帧刷新率至少为5 fps的实时30 fps和25 fps视频系统不敏感。 f_{SI13} 、 f_{HVI13} 和 f_{COHER_COLOR} 特征足以捕捉这些低帧频视频系统的质量方面（这在多媒体应用中很常见）。相对于区分在时间上相隔一帧的两个图像，0.2s的间隔也更接近于人类视觉系统的峰值时间响应。

图27提供了带有数字传输信道中瞬态出发误差的数字视频系统的原始（蓝色实线）和经处理（红色虚线）视频场景的示意图。经处理图像中的瞬态误差产生了 f_{ATI} 特征中的尖状物。传输 f_{ATI} 特征所需的带宽极其低，因为，对于30 fps的视频，它只需要每秒30次的采样。经处理视频中其它类型的相加噪声（如可能由模拟视频系统产生的噪声），在经处理特征流的时间历史中，相对于原始特征流，将作为一个正位DC偏移出现。消除噪音的视频编码系统将产生负的DC偏移。

在从图27所示的 f_{ATI} 特征流中提取瞬态误差参数之前，增加活动尖状物（图27中的红色尖状物）的带宽是有益的。原因是瞬态误差的低矮活动尖状物并不能足以代表这些类型误差的感知影响。增加活动尖状物带宽的一种方法是在计算两个波形之间的误差参数函数之前，对原始和经处理特征流应用最大过滤器。对于基于 f_{ATI} 的误差参数，采用了一个7点宽最大过滤器（将在此处表示为 $max7pt$ 函数），在各个帧产生了自身最大的输出样本以及在各边上3个最近的相邻图形（即较早和较晚时间样本）。

图 27

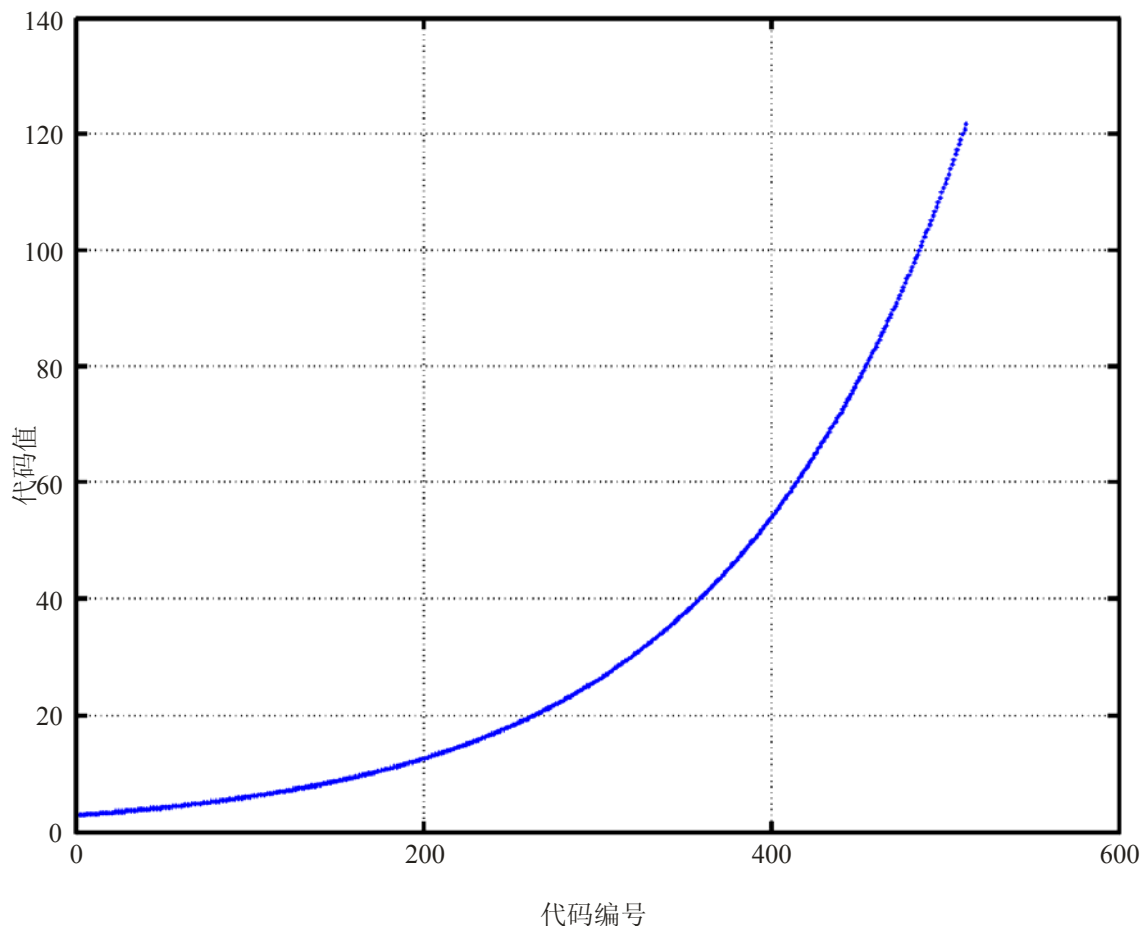
 f_{ATI} 特征的时间历史范例

3.2.5 特征的量化

量化为9比特的精确度对于 Y_{MEAN} 、 f_{SI13} 、 f_{HV13} 和 f_{COHER_COLOR} 特征已经足够，但 f_{ATI} 特征应量化为10比特。要使得对视频质量参数计算的影响最小，应在量化误差与量化信号的幅度呈比例的情况下采用非线性分层设计。很低的数值一律量化为一些截止数值，在此数值以下没有有用的质量评价信息。这样的一种量化器设计将对应的参数计算中的误差降至最低，因为通常这些计算基于经处理和原始特征流的误差比或对数比（参见以下的参数描述节）。

图 28 提供了用于 f_{SI13} 原始特征的9比特非线性量化器。参考编码子程序“model_lowbw_compression”提供了快速低带宽VQM所采用的建议量化器的完整数学描述。如果特征超出了低端和高端（不太可能）上建议的量化器范围，那么从这些特征获得的S-T参数归零，以便它们不影响整体的VQM。

图 28

 f_{SI13} 特征的非线性9比特量化器

3.3 参数描述

3.3.1 参数综述

跟踪视频质量各个感知方面的参数计算涉及到几个步骤。这些步骤可能涉及：

- 对从每个S-T子区域提取的特征应用感知门限；
- 计算经处理特征与对应原始特征之间的误差函数；
- 集中在空间和时间上产生的误差。

对于这些方法及其伴随的参数名称的数学标记法（也将在此处使用的）的详细描述，参见ITU-T J.144建议书的附件D.8。参考代码子程序“`model_fastlowbw_parameters`”提供了快速低带宽VQM参数的完整数学描述。出于简化起见，本节中的参数描述并未考虑特征量化的影响（如处理可能超出所建议量化范围的特征值）。

3.3.2 新方法

本节将总结在RR特征基础上用很低传输带宽（如用于NTIA快速低带宽VQM）被发现可用来改进参数客观主观相关性的新方法（即未包括在ITU-T J.144建议书中的新方法）。应注意到，没有发现可以改进ITU-T J.144建议书附件D.8.2.1中给出的参数误差函数基本形式。两种持续产生最优参数结果的误差函数（用于空间和时间参数）是对数比 $\{\log_{10} [f_p(s,t) / f_o(s,t)]\}$ 和误差比 $\{[f_p(s,t) - f_o(s,t)] / f_o(s,t)\}$ ，其中 $f_p(s,t)$ 和 $f_o(s,t)$ 为从S-T区域内提取的经处理特征和对应原始特征，其空间坐标为 s 且时间坐标为 t 。误差必须分为增益和损耗，因为人类对相加衰减（如遮挡）和相减衰减（如模糊）的反应不同。在应用这两种误差函数之前对特征应用一个更低的感知门限可防止被零相除。

在采用上述误差函数计算S-T参数之后，必须在空间和时间上集中S-T参数，以生成用于视频剪辑的参数值。这种误差集中可发生在多个阶段（例如在空间并随后在时间上）。快速低带宽VQM采用的一种新误差集中方法称为宏块（MB）误差集中。宏块误差集中将连续多个S-T子区域组合起来并对该集合应用误差集中函数。例如，标记为“MB(3,3,2)max”的函数将对由垂直为3、水平为3、时间为2堆积而成的每组18个S-T子区域的参数值进行最大函数运算。对于上述 f_{SI13} 、 f_{HV13} 和 f_{COHER_COLOR} 特征的 $32 \times 32 \times 1s$ 的S-T子区域，每个MB(3,3,2)区域将包含跨越96条垂直线乘以96条水平像素并乘以2秒时间的视频流的一部分。宏块误差集中被发现在跟踪局限于空间和时间的衰减的感知影响方面大有作用。此类局部化衰减通常在质量决定进程中占支配地位。当MB位于S-T样本的中心时，宏块误差集中也可作为一种过滤过程进行实施，以便每个S-T样本由其宏块过滤后的值替换，而不是为每个MB生成单独的输出值。这称为重叠式宏块（OMB）误差集中。

第二种误差集中方法是一种广义Minkowski(P,R)求和,其定义为:

$$Minkowski(P,R) = R \sqrt[R]{\frac{1}{N} \sum_{i=1}^N |v_i|^P}$$

此处, v_i 表示包括在求和中的参数值。例如,该求和可包括在时间上的某个给定时刻的所有参数值(空间集中)或可适用于上述的宏块。Minkowski求和(其中幂 P 等于根 R)被许多开发者用于误差集中的视频质量度。广义Minkowski求和,其中 $P \neq R$,提供了将单个参数在感知质量方面对变化作出反应的额外灵活性。这是在将多个参数综合成感知视频质量的单一评估之前一个必要的步骤,后者通过线性拟合进行。

3.3.3 色彩参数

从 f_{COHER_COLOR} 特征中提取两个参数。其中一个称为 $color_extreme$ 的参数测量可能由于传输误差造成的色块而导致的极端色彩失真。另一个称为 $color_spread$ 的参数提供了色彩误差中的变化或扩展指示。这两种参数使用曼哈坦距离的平方根,而不是采用欧几里得距离来量化失真(如ITU-T J.144建议书附件D.8.2.2那样)。根据ITU-T J.144建议书附件D.8.2.2中的数学表达式,其中 $f_p(s,t)$ 和 $f_o(s,t)$ 表示从经处理和原始视频流S-T区域中提取的二维 f_{COHER_COLOR} 特征,此特征比较函数如下:

$$sqrtmanhat(s,t) = \sqrt{\sum_{C_B, C_R} \left| \frac{f_p(s,t)}{C_B} - \frac{f_o(s,t)}{C_R} \right|^2}$$

曼哈坦距离测量似乎要好于欧几里得距离测量且需要平方根函数来将参数对质量变化的反应进行量化。根据ITU-T J.144建议书附件D.8中的数学表达式,色彩参数由以下公式给定:

$$color_extreme = color_coher_color_30x30_1s_mean_sqrtmanhat_OMB(3,3,2)_{above99\%} _Minkowski(0.5,1)$$

$$color_spread = color_coher_color_30x30_1s_mean_sqrtmanhat_OMB(3,3,2) _Minkowski(2,4) _90\%$$

然后,包含 $color_extreme$ 和 $color_spread$ 参数最优组合的综合色彩参数($color_comb$)按照以下公式计算:

$$color_comb = 0.691686 * color_extreme - 0.617958 * color_spread$$

然后,在第一段将该正值的 $color_comb$ 参数进行截取,在数学上表示为以下公式(根据ITU-T J.144建议书附件D.8.5的表达式):

$$color_comb = color_comb_clip_0.114$$

此 $color_comb$ 参数包括在VQM计算的线性组合中。

3.3.4 空间参数

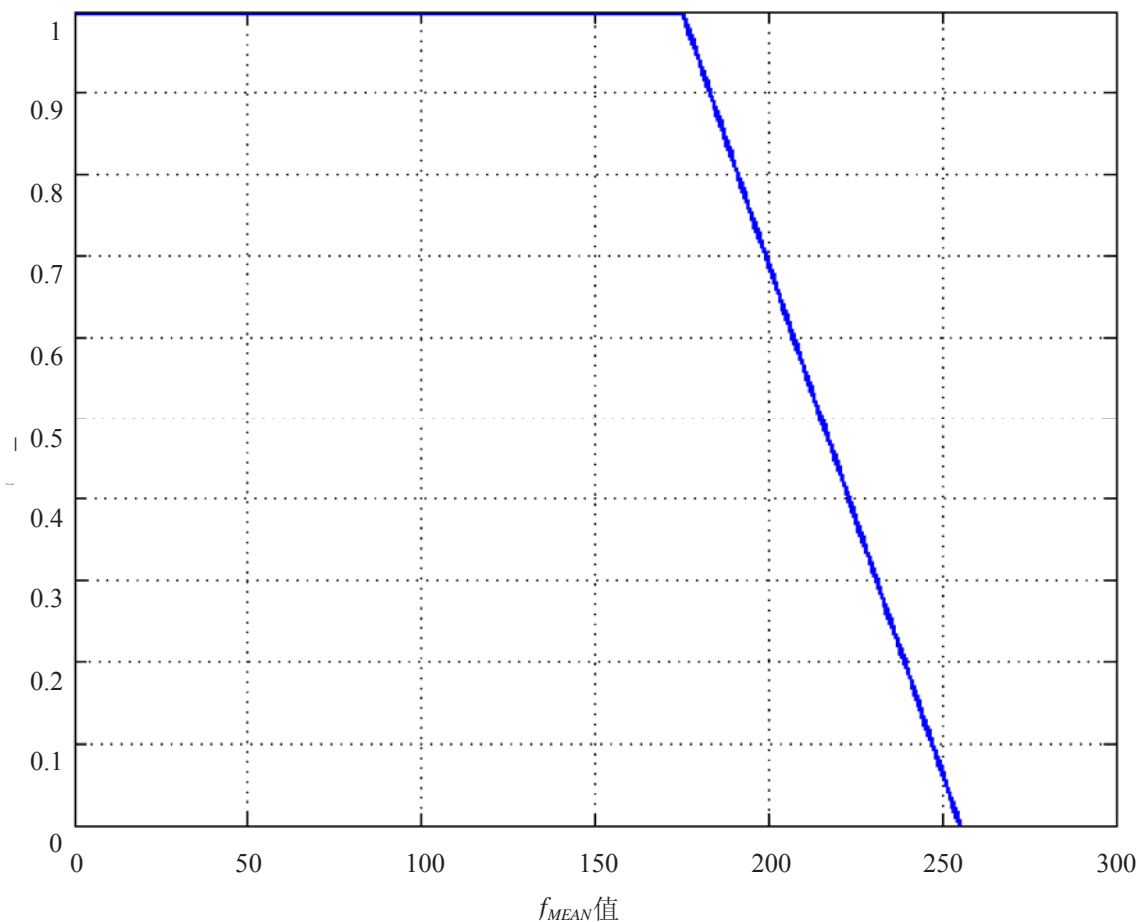
从 f_{SI13} 特征计算两个空间参数，一个参数测量空间信息的损耗 (si_loss)，另一个测量空间信息的增益 (si_gain)。根据ITU-T J.144建议书附件D.8数学表达式，这两种参数由以下公式给定：

$$si_loss = \text{avg1s_Y_sil3_30x30_std_3_ratio_loss_OMB}(3,3,2)\text{Minkoski}(1,2)\text{Minkosk}(1.5,2.5)\text{clip_0.12}$$

$$si_gain = \text{avg1s_Y_sil3_30x30_std_3_log_gain_clip_0.1_above95\%tail_Minkoski}(1.52)$$

随着S-T子区域平均亮度 (Y) 水平的增加 (即根据 f_{MEAN} 特征的测量)，在空间细节上感知变化的能力 (如由 si_loss 所测量的模糊) 则下降。如图29所示，可以对来自每个S-T子区域的 si_loss 值引入加权函数 (Y_weight) 加以解决 (即在每个S-T子区域比率损耗比较函数之后，但在空间和时间折叠函数之前执行 si_loss 数值)。在达到175的平均亮度水平之前，加权函数 Y_weight 等于一 (即完全加权)，然后随着亮度值从175增加到255，它线性下降为零。该中间修正仅适用于 si_loss 数值，而不适用于 si_gain 数值。

图 29

修改 si_loss S-T参数的加权函数 Y_weight 

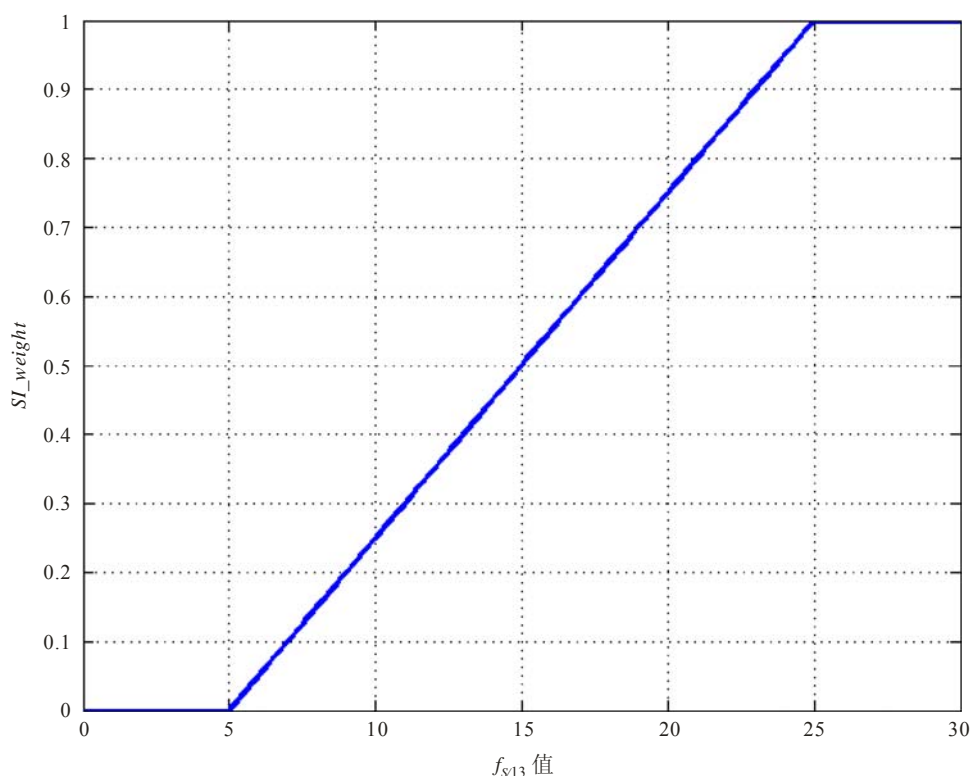
BT.1885-29

从 f_{HV13} 特征计算两种空间参数，一种测量相对水平和垂直 (HV) 空间信息的损耗 (hv_loss)，另一种测量增益 (hv_gain)。根据ITU-T J.144建议书附件D.8中的数学表达式，这些参数由以下公式给定：

$$\begin{aligned}
 hv_loss &= \text{avg1s_Y_hv13_angle0.225_rmin20_30x30_mean_4_ratio_loss_...} \\
 &\quad \text{OMB(3,3,2)below1\%_Minkoski(1,1.5)_clip_0.08} \\
 hv_gain &= \text{avg1s_Y_hv13_angle0.225_rmin20_30x30_mean_4_log_gain_...} \\
 &\quad \text{clip_0.06_OMB(3,3,2)above99\%tail_Minkoski(1.5,3)}
 \end{aligned}$$

未在上述公式中说明的是，图29中所示的 Y_weight 在空间和时间折叠函数之前也适用于每个S-T子区域的 hv_loss 和 hv_gain 数值（分别在 $ratio_loss$ 和 log_gain 计算之后）。对来自每个S-T子区域的 hv_loss 数值应用额外的加权函数（图30所示的 SI_weight ）。这是降低S-T区域 hv_loss 灵敏度而必要的，后者几乎没有空间信息（即 f_{SI3} 原始特征值较低）。

图 30

修改 hv_loss S-T参数的加权函数 SI_weight 

BT.1885-30

可采用在VQM计算一节中所述的VQM破碎函数等函数来碾平空间失真参数（即超出培训数据的过多偏移受到限制或压缩）。

3.3.5 时间参数

从 f_{ATI} 特征计算两个时间参数，一个测量增加的随机噪声（ ati_noise ），另一个测量由传输误差造成的活动扰动。根据ITU-T J.144建议书附件D.8中的数学表达式，这些参数由以下公式给定：

$$ati_noise = Y_rand5\%_ati0.2s_rms_5_ratio_gain_between25\%50\%$$

$$ati_error = Y_rand5\%_ati0.2s_rms_max7pt_12_ratio_gain_above90\%$$

为使 *ati_noise* 和 *ati_error* 参数在时间未校准的情况下更加强健，对属于相对于原始视频最佳评估时间校准±0.4 s 范围内的所有经处理视频的时间校准，计算新参数，然后选择最小参数值。

3.4 VQM计算

与ITU-T J.144建议书附件D中的NTIA一般VQM类似，快速低带宽VQM计算将 f_{HVI3} 特征的两个参数 (*hv_loss* 和 *hv_gain*)， f_{SI13} 特征的两个参数 (*si_loss* 和 *si_gain*) 以及 f_{COHER_COLOR} 特征的两个参数进行线性组合（不包括将两个参数合并为一个单一的色彩失真参数，称为 *color_comb*）。NTIA一般性VQM中的噪声参数用此处所描述的低带宽 f_{ATI} 特征的两个参数替换 (*ati_noise* 和 *ati_error*)。

由此， VQM_{FLB} （快速低带宽VQM的缩写）包括八个参数的线性组合。 VQM_{FLB} 由以下公式给定：

$$\begin{aligned} VQM_{FLB} = & \{ 0.38317338378290 * hv_loss + 0.37313218013131 * hv_gain + \\ & 0.58033514546526 * si_loss + 0.95845512360511 * si_gain + \\ & 1.07581708014998 * color_comb + \\ & 0.17693274495002 * ati_noise + 0.02535903906351 * ati_error \} \end{aligned}$$

整个VQM（在所有的参数叠加之后）在较低的门限0.0上进行截取，以防止出现负的VQM数字。最后，对超过1.0的VQM数值应用允许最大50%超额的破碎函数，亦限制超出培训数据范围的过度失真视频的VQM数值。

如果 $VQM_{FLB} > 1.0$ ，那么 $VQM_{FLB} = (1 + c) * VQM_{FLB} / (c + VQM_{FLB})$ ，其中 $c = 0.5$ 。

通过以上方式计算的 VQM_{FLB} 的数值将大于或等于零且标称最大值为1。对于极度失真的视频场景， VQM_{FLB} 可能会偶尔超过1。

为使得 VQM_{FLB} 在空间未对准的情况下更加强健，计算相对于原始视频最佳评估空间校准正负1像素范围内经处理视频所有空间校准的 VQM_{FLB} ，然后选择最小 VQM_{FLB} 。

4 参考文件

- [1] ITU-R BT.601-6建议书（01/07）– 4:3标准和16:9宽屏幕纵横比数字电视的演播室编码参数。
- [2] 视频质量模型（VQM）软件工具 – 二进制可执行和源代码，可从国家电信和信息管理局（NTIA）http://www.its.bldrdoc.gov/n3/video/VQM_software.php 下载。
- [3] ITU-T J.244建议书（04/08）– 具有固定空间和时间域偏差并具有普遍增益和偏移的视频传输系统的全参考和减少参考情况下的校准方法。

- [4] 多媒体第一阶段验证测试的VQEG最终报告（2008年），“视频质量专家组有关多媒体质量评估目标模型验证第一阶段的最终报告”，视频质量专家组（VQEG），<http://www.its.bldrdoc.gov/vqeg/projects/multimedia>，ITU-T第9研究组，TD923，2005-2008年研究期。
- [5] ITU-T J.144建议书（03/04）–数字有线电视在全参考情况下客观感知视频质量测量方法。
- [6] ITU-R BT.1683建议书（06/04）–完整参考的情况下标清数字广播电视的客观感知视频质量测量方法。

5 实施快速低带宽VQM的参考代码

此参考代码的目的是帮助用户恰当地落实快速低带宽VQM。尽管将MATLAB®代码用于参考代码，但也可采用任意软件代码来生成此处给出的结果。第5部分的每一个小节都包含了在小节标题中命名的函数的MATLAB代码（如将第5.1段的内容保存到一个名为“fastlowbw_ref.m”的文件中）。在不带任何参数的情况下执行fastlowbw_ref，将收到与如何调用程序有关的帮助信息。该代码包含了在较大的视频序列（如1分钟序列）中运行一个较短视频片段（即5至15秒）的灵活性。通过将短视频片段偏移一秒钟并重新计算各个时间偏移的模型的方法，可以实现此任务。尽管以下并未展示这种功能，代码中的意见以及从“model_fastlowbw_parameters.m”返回的变量将提及该能力。这种隐藏的能力对于实施在进行中视频质量监测系统可能是有益的。

当用快速低带宽VQM参考代码（“fastlowbw_ref.m”函数）处理样本测试向量（即视频片段）时，生成包括校准和模型结果的文字文档。对于以下MATLAB函数调用示例，应可获得与以下给出相类似的输出文件（由于快速低带宽VQM的随机程序，结果可能与此处给出的结果略有差别）：



5.Reference
Code.doc

附录

传输误差分析

RRNR-TV项目的VQEG验证测试包括525（NTSA）和625（PAL）制式。每个实验包括12个源序列和156个经处理的视频序列（PVS）。在这156个PVS中，40个包含输出误差，而116个仅包含编码误差。表9和表10显示了带有传输误差的PVS的RMSE和OR。应注意到，RMSE和OR是采用从整个数据获得的回归直线计算的。也就是说，回归直线是采用整个数据计算的。然后，传输误差的RMSE和OR是采用带有传输误差的PVS计算的。

表 9

RRNR-TV验证测试（525制式）的RMSE和OR（TE：传输误差）

525 制式	所有		有TE		无TE	
	RMSE	OR	RMSE	OR	RMSE	OR
Model_A_15k	0.418	0.385	0.574	0.500	0.362	0.293
Model_A_80k	0.423	0.378	0.582	0.475	0.366	0.293
Model_A_256k	0.424	0.378	0.584	0.475	0.367	0.293
Model_B_80k	0.598	0.667	0.768	0.650	0.544	0.586
Model_B_256k	0.587	0.647	0.763	0.600	0.530	0.578
Model_C_80k	0.465	0.513	0.557	0.550	0.440	0.405
Model_C_256k	0.511	0.609	0.584	0.450	0.495	0.578
PSNR_NTIA	0.556	0.571	0.549	0.500	0.568	0.491

表 10

RRNR-TV验证测试（625制式）的RMSE和OR（TE：传输误差）

625 制式	所有		有TE		无TE	
	RMSE	OR	RMSE	OR	RMSE	OR
Model_A_15k	0.524	0.468	0.597	0.450	0.508	0.414
Model_A_80k	0.513	0.462	0.594	0.500	0.494	0.379
Model_A_256k	0.516	0.468	0.593	0.500	0.499	0.379
Model_B_80k	0.887	0.724	0.545	0.500	0.986	0.716
Model_B_256k	0.864	0.744	0.523	0.600	0.962	0.716
Model_C_80k	0.585	0.583	0.282	0.200	0.663	0.647
Model_C_256k	0.657	0.590	0.292	0.175	0.747	0.638
PSNR_NTIA	0.605	0.564	0.338	0.250	0.678	0.517