

前段时间略忙，因此一直计划要总结的很多东西都没来得及写，这两天趁着空闲时间写上一篇。以后等时间充裕了再补充一些内容。本文总结一下学习视频质量评价技术的方法。视频质量评价是我研究生阶段主要的工作，包括发表的论文，毕业设计，进行的实验等等。在这一方面在国内做的研究并不是很多，因此我打算总结一下学习的经验，方便看过本文的人了解视频质量评价技术。

## 0. 背景

首先应该了解一下质量评价都是干什么用的。这个其实本身也困扰了我一段时间。进入实验室之后，我也没有向师哥师姐咨询过这个问题。后来自己摸索了一段时间，大致知道视频质量评价应用的方面。

### 0.1. 视频编码器（编码标准）性能评测

编码器（或者是编码标准）性能评测是视频质量评价使用最为广泛的一项应用。

视频编码器的作用是将输入的像素数据编码为输出的码流数据，最终的意义在于压缩视频的数据量。视频编码过程是一个有损压缩的过程，输出的视频码率（比特率）越低，视频的质量越低。虽然大体的趋势是一样的，即编码的视频的质量和视频的码率成正比的关系，但是不同的编码器之间的性能还是有区别的（即使采用相同的视频编码标准，比如同样是基于 H.264 标准的编码器，不同编码器之间性能也有区别）。一个优秀的视频编码器和和其他视频编码器相比，就是能在相同的码率下，编码获得质量更高的视频。或者在编码获得相同质量的前提下，使用更小的码率。可以通过视频质量评价的客观算法，对比得出视频编码器孰优孰劣。例如这方面的研究可以参考文章：

[MSU出品的 H.264编码器比较（2011.5）（包含VP8以及XviD）](#)

[MSU 出品的 H.264 编码器比较（2012.5）](#)

除了对比视频编码器之外，也可以对比视频编码标准的性能。视频编码标准之间的差距一般来说相对比较大，越晚提出的编码标准，性能一般越强。例如可以参考文章：

[视频编码方案之间的比较（HEVC，H.264，MPEG2等）](#)

详细的介绍可以查看第 5 章的内容。

### 0.2. 视频质量监测

视频质量监测是视频质量评价另一种使用比较广泛的应用。

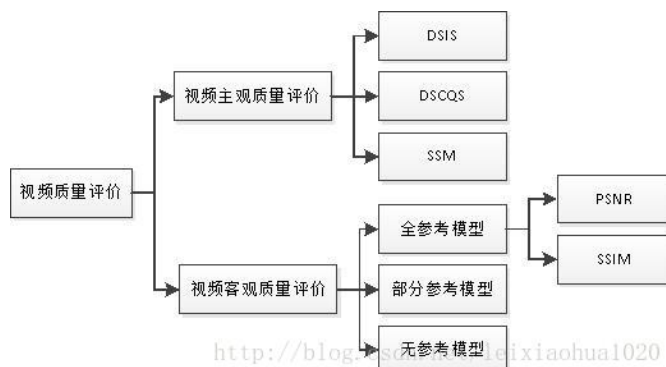
该方面主要用于实时监测分析视频流。例如监测 CMMB，数字电视，IPTV，互联网视频等这些信号的质量。当这些信号出现异常的时候，可以反馈给服务提供商，使之及时采取相应的措施，修复视频传输过程中出现的问题。例如这方面的研究可以参考文章：

[IPTV的QoE评价方法](#)

详细的介绍可以查看第 5 章的内容。

## 1. 视频质量评价分类

通常，人们将数字视频质量评估的方法分为两大类：视频主观质量评价（Subjective Quality Assessment，SQA）和视频客观质量评价（Objective Quality Assessment，OQA）。直接让观测者对视频质量做出直观判断的主观质量评估方法是最为准确的评估方法，但该方法相对复杂且其结果易受多种因素影响，因此，在实际应用中通常使用客观的、易于实现的视频客观质量评估方法。根据对原始视频的依赖程度，客观质量评估法又可以分为三种类型：全参考（Full Reference，FR），部分参考（Reduced Reference，RR）和无参考（No Reference，NR）。视频质量评价的分类如下图所示。



### 1.1. 视频主观质量评价

视频主观质量评估就是选择一批非专家类型的受测者，让他们在一个特定的受控环境中，连续观看一系列的测试序列大约 10 至 30 分钟，然后采用不同方法让他们对视频序列的质量进行评分，最后求得平均判分（Mean Opinion Score，MOS），并对所得数据进行分析。测试环境中的受控因素包括：观看距离、观测环境、测试序列的选择、序列的显示时间间隔等。根据不同的测试环境、测试目的，有不同的主观评价方法，可参考下文：

### 视频主观质量评价方法

主观评价一般需要专业的实验环境，以及大量的受测人员，因此成本比较大。

主观评价相关标准：

### ITU-R BT.1788建议书对多媒体应用中视频质量的主观评估方法

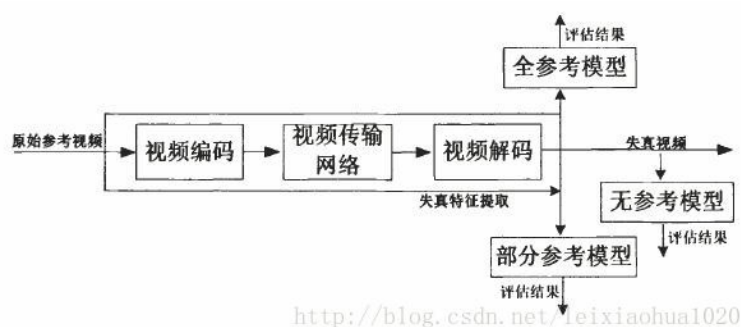
主观质量评价实验软件：

### MSU Perceptual VideoQuality tool

## 1.2. 视频客观质量评价

由于视频主观质量评估过于复杂且其结果易受多种因素影响，因此，在实际应用中需要一种客观的、易于实现的视频客观质量评估方法。因为受测者对视频质量的主观感受最为真实和准确，所以在相同的视频系统和相同的测试序列条件下，要求客观评估的结果应与主观评估的结果具有好的一致性或相关性。

对于一个视频客观质量评估方法来说，关键的是找出一个或几个最合适的视频质量量度来衡量视频质量的好坏。根据失真视频与其相应的原始参考视频的比较程度，视频客观质量评估方法分为三大类：**全参考方法** (Full Reference)、**部分参考方法** (Reduced Reference)、**无参考方法** (No Reference)。三种方法如下图所示。



由图可以看出，全参考方法只适用于已知原始视频的编码器端，可以用它来定量测量编码器的输出视频质量，它主要用于编码器的设计和对不同编码器的性能进行评估和优化。部分参考方法需要部分原始视频或者需要通过传输额外的一些信息。特别是无参考方法，在判断视频质量时不需要来自原始参考视频的任何信息，仅仅通过对失真视频空域和频域的处理分析来提取失真视频的特征，从而得到视频质量。适用于带宽有限，传输成本较高，不可能提供原始参考视频序列的无线和 IP 视频业务，可以用来实时或非实时的监控传输信道的服务质量（QoS）和视频系统的性能，并通过质量反馈来调节编解码器或信道的参数，保证有良好的恢复视频质量。

在这里主要介绍全参考质量评价模型，这种模型已经十分成熟。几种全参考视频质量评价模型可参考下文：

### 全参考客观视频质量评价方法（MSE，PSNR，SSIM）

### 全参考视频质量评价方法（PSNR，SSIM）以及相关数据库

全参考视频质量评价的计算工具有下列两个：

**开源视频质量评价工具：Evalvid**。该工具是开源的，基于命令行的，适用于批处理。

**视频质量评价工具：MSU Video Quality Measurement Tool**。该工具是商业的，图形界面的，界面相对比较友好，但不适用于大批量的处理。

下面列举几张图片，说明全参考视频质量评价模型的客观值和主观感受之间的关系。测试序列选自 VQEG 数据库，内容为一个松鼠拿着松子走路，时长为 10 秒，时间复杂度和空间复杂度相对较低。使用 x264 编码为不同的码率后分别计算了 PSNR 和 SSIM。截图所示为视频序列的第 50 帧。

注：PSNR 取值 20-40，值越大，视频质量越好。SSIM 取值 0-1，值越大，视频质量越好。



原视频（像素格式：YUV420P；分辨率：640x360）



码率 = 144 Kbps（每像素比特数 = 0.025）；PSNR=29.622；SSIM=0.81





码率 = 288 Kbps (每像素比特数 = 0.050) ; PSNR=32.89 ; SSIM=0.906



码率 = 432 Kbps (每像素比特数 = 0.075) ; PSNR=34.98 ; SSIM=0.941



码率 = 576 Kbps (每像素比特数 = 0.100) ; PSNR=36.59 ; SSIM=0.958

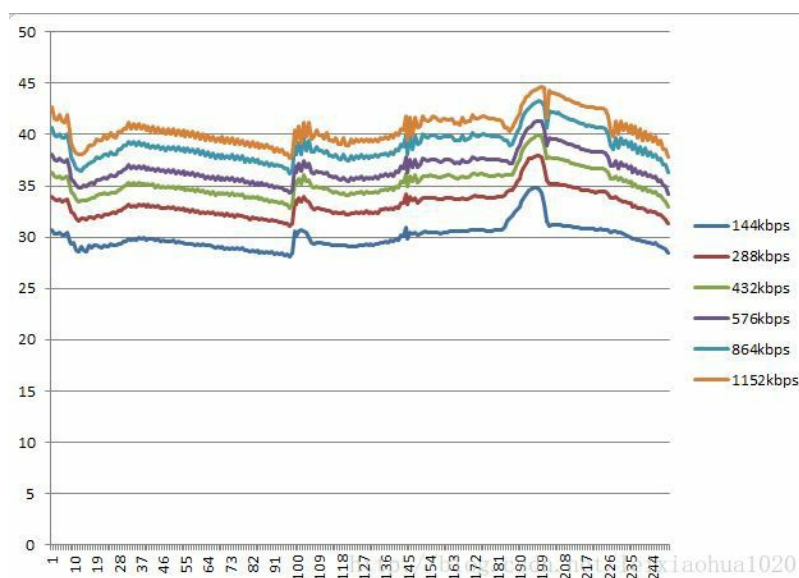


码率 = 864 Kbps （每像素比特数 = 0.150 ）； PSNR=38.823 ； SSIM=0.974

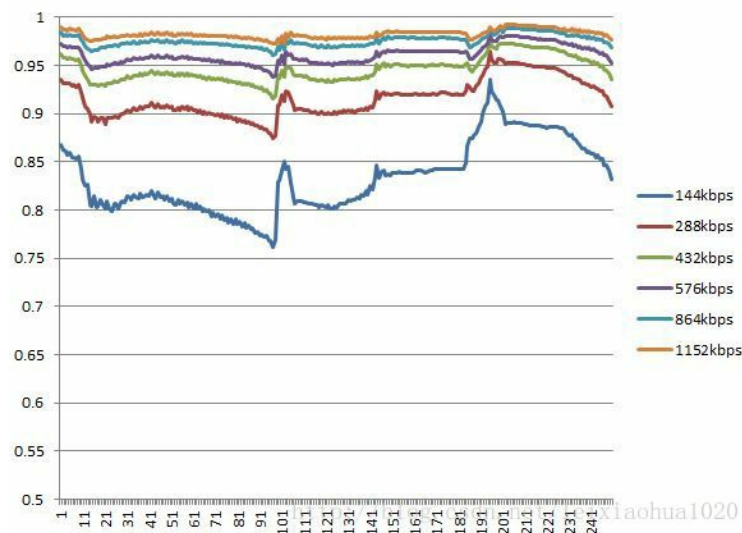


码率 = 1 152 Kbps （每像素比特数 = 0.200 ）； PSNR=40.586 ； SSIM=0.982

整个序列一共 250 帧，所有帧的 PSNR 如下图所示：



所有帧的 SSIM 如下图所示：



## 2. 质量评价涉及到的其它知识

视频质量评价技术中，涉及到以下几方面的知识。

### 2.1. 视频数据的处理

视频质量评价研究的过程中，需要对视频数据进行编码，解码，像素格式转换等操作，以适应各种计算工具的需要。在此简单总结一下常用的工具和命令。除了视频编码所需的特定的编码器（例如 JM，x264，xvid 等等）之外，需要以下几个工具：

FFmpeg：视频处理。

主页地址：<http://ffmpeg.org>

YUV Player Deluxe：看 YUV 文件用。注意如果把 YUV 文件的宽和高信息写到文件名中，形如“xxx\_[width]x[height].yuv”，例如“src01\_640x360.yuv”，则该播放器会自动以 640x360 分辨率打开该 YUV 文件，还是很智能的。

主页地址：<http://www.yuvplayer.com/>

视频播放器：看压缩码流文件用。像终极解码，暴风影音，QQ 影音这些都可以。

#### 2.1.1 编码

一般情况下可以使用 FFmpeg。绝大多数的视频编码器都被 FFmpeg 支持，因此使用起来比较方便。而且对于不同的编码器，FFmpeg 都使用统一的命令行格式，命令行参数相对比较容易记忆。当然特殊情况下还是需要独立的编码器，比如说用于学术研究的 JM，HM 这样的软件，或者是比较新的编码器。下面举几个例子。

下面两条命令是使用 FFmpeg 调用相应的库进行编码的命令。

FFmpeg 调用 x264 编码 YUV420P 视频序列（码率 800kbps）：

```
1. ffmpeg -s 640x360 -i src01_640x360.yuv -b:v 800k -vcodec libx264 src01_640x360_800k.h264
```

FFmpeg 调用 xvid 编码 YUV420P 视频序列（码率 800kbps，封装为 AVI）：

```
1. ffmpeg -s 640x360 -i src01_640x360.yuv -b:v 800k -vcodec libxvid src01_640x360_800k.avi
```

可以看出上面两个命令参数格式基本上是一致的。

下面两条命令是使用独立的编码器进行编码的命令。功能上和上两条命令是一样的。

使用 x264 编码 YUV420P 视频序列（码率为 800kbps）：

```
1. x264 --bitrate 800 --input-res 640x360 -o src01_640x360_800k.h264 src01_640x360.yuv
```

使用 xvid 编码 YUV420P 视频序列（码率 800kbps，封装为 AVI）：

```
[plain]
1. xvid_encraw -i src01_640x360.yuv -w 640 -h 360 -csp i420 -bitrate 800 -o src01_640x360.avi
```

可以看出上面两个命令行的参数格式差别还是比较大的，相对难记忆一些。

专用于学术研究的软件模型如 JM，HM 这些一般是不被 FFmpeg 支持的，只能独立运行。如下面的例子。

使用 HM（HEVC）编码 YUV420P 视频序列：

```
[plain]
1. tappencoder -c encoder_lowdelay_P_main.cfg -i src01_640x360.yuv -wdt 640 -hgt 360 -b src01_640x360_800k.hm10
```

注：这里需要注意的是 tappencoder 后面附加的参数如 -i，-wdt 这些参数可以覆盖掉原先写在配置文件 xx.cfg 中的参数。

### 2.1.2 解码

一般情况下都可以使用 FFmpeg。和编码器一样，绝大多数的视频解码器都被 FFmpeg 支持，因此使用起来比较方便。少数情况下也可以使用独立的解码器。下面举几个例子。

下面两条命令是使用 FFmpeg 进行解码的命令。

FFmpeg 解码 H.264 码流为 YUV420P：

```
[plain]
1. ffmpeg -i src01_640x360_800k.h264 src01_640x360_800k.yuv
```

FFmpeg 解码 MPEG-4 码流（AVI 封装）为 YUV420P：

```
[plain]
1. ffmpeg -i src01_640x360_800k.avi src01_640x360_800k.yuv
```

可以看出上面两个命令参数格式基本上是一致的。

专用于学术研究的软件模型一般是不被 FFmpeg 支持的，只能独立运行。如下面的例子。

使用 HM（HEVC）解码 HEVC 码流为 YUV420P：

```
[plain]
1. tappdecoder -b src01_640x360.hm10 -o src01_640x360.yuv
```

### 2.1.3 像素格式转换

推荐使用 FFmpeg。FFmpeg 中的 libswscale 对像素格式转换的支持很好，使用起来很方便。下面举几个例子。

封装过的像素数据（AVI）转换为 YUV420P：

```
[plain]
1. ffmpeg -i src01.avi src01.yuv
```

封装过的像素数据（BMP）转换为 YUV420P：

```
[plain]
1. ffmpeg -i src01.bmp src01.yuv
```

封装过的像素数据（AVI）转换为 YUV422P。在这里需要注意，如果不指定 -pix\_fmt 的话，FFmpeg 默认将 YUV 数据转换为 YUV420P 格式的。如果需要其它 YUV 格式的数据的话，一定要指定 -pix\_fmt。

```
[plain]
1. ffmpeg -i src01.avi -pix_fmt yuv422p src01_422.yuv
```



注：查看 FFmpeg 支持的所有的格式，可以使用命令：

```
[plain]
1.  ffmpeg -pix_fmts
```

YUV 格式之间互相转换，例如 YUV422P 转换为 YUV420P，同时将图像分辨率从 1920x1080 下变换为 1280x720。在这里需要注意，YUV 是纯像素数据，并不包含画面的宽高，取样格式等信息，因此转换的时候需要自己指定。

```
[plain]
1.  ffmpeg -s 1920x1080 -pix_fmt yuv422p -i src01_1920x1080_422.yuv -s 1280x720 -pix_fmt yuv420p src01_1280x720.yuv
```

YUV422P 转换为封装过的像素数据（BMP）。同理，YUV 需要指定宽高，取样格式。

```
[plain]
1.  ffmpeg -s 1920x1080 -pix_fmt yuv422p -i src01_1920x1080.yuv src01_1920x1080.bmp
```

YUV420P 转换为 YUV422P 并且进行封装（AVI）：

```
[plain]
1.  ffmpeg -s 1920x1080 -i src01_1920x1080.yuv -vcodec rawvideo -pix_fmt yuv422p src01_1920x1080.avi
```

注：像素数据之间转换的时候，有一个问题需要注意。就是 YUV 和 RGB 相互转换的过程中，像素取值范围的问题。有两种取值范围：0-255 或 16-235。对于精确度要求非常高的实验，需要研究一下这方面的设置。

## 2.1.4 批处理

当需要批量进行视频的编码，解码，像素转换，计算 PSNR 等等工作的时候，通常可以将所有的命令一行一行地写入到一个记事本中，保存后将记事本的后缀“.txt”改为“.bat”，双击运行之后即可批量执行所有的命令。

## 2.2. 实验结果的处理

视频质量评价研究过程中经常需要进行数据处理,除了使用 Excel 之外,使用最多的就是 Matlab 中的拟合工具了,包含曲线拟合工具 cftool 以及曲面拟合工具 sftool。在 Matlab 主界面命令窗口中输入“cftool”或者“sftool”就可以打开相应的工具。

这两个工具都是图形界面的，使用起来比较方便。曲线拟合工具 cftool 使用可参考：

[Matlab的曲线拟合工具箱cftool使用简介](#)

## 2.3. 视频序列复杂度

视频编码和序列的复杂度是息息相关的。序列的复杂度分为时间复杂度和空间复杂度。视频画面细节越多，空间复杂度越大；视频内容运动越剧烈，时间复杂度越大。视频复杂度越大，在同等画质的前提下，编码需要越多的数据量。可以根据序列的时间复杂度和空间复杂度筛选合适的测试序列。时间复杂度和空间复杂度可以参考：

[视频特性TI（时间信息）和SI（空间信息）的计算工具：TlandSI](#)

## 3. 质量评价做的比较好的机构

### 3.1 . VQEG（视频质量专家组）

ITU-T 和 ITU-R 的研究小组联合在一起,在 1997 年成立了视频质量专家组 (VideoQuality Experts Group，VQEG)，专门从事视频质量评估的研究和相关标准的制定。属于视频质量评价方面的国际组织。



中文简介：<http://blog.csdn.net/leixiaohua1020/article/details/11729745>

首页：<http://www.its.bldrdoc.gov/vqeg/vqeg-home.aspx>



## 3.2 .MSU Video Group （ MSU 视频组）

莫斯科大学视频组。出了很多视频质量评价方面的报告，以及这方面的软件。



出品的工具 MSU VQMT （视频质量评价工具）：<http://blog.csdn.net/leixiaohua1020/article/details/11729257>

首页：<http://www.compression.ru/video/index.htm>

## 4. 视频质量评价资源

### 4.1. 视频测试序列库

CDVL （ TheConsumer Digital Video Library ）：<http://www.cdvl.org/>

包含了大量的未压缩的原始视频序列，以及一些受损序列。一般情况下这个网站上的视频序列就够用了。

### 4.2 . 图像测试序列库

这些库不但包含了图像，而且包含了主观质量评价的结果。

[TID2008 database](#)

[CSIQ database](#)

[LIVE database](#)

注：该网站可以作为参考 <http://sse.tongji.edu.cn/linzhang/IQA/IQA.htm>

## 5. 质量评价的已有的研究

在此列出一些自己看过并且感觉比较有价值的论文，已经记录为中文。

### 5.1. 编码器（编码标准）性能评测

[HEVC, VP9, x264性能对比](#)

[限制码率的视频编码标准比较（包括MPEG-2, H.263, MPEG-4, 以及 H.264）](#)

[视频编码国家标准AVS与H.264的比较（节选）](#)

[MSU 出品的 H.264 编码器比较（2012.5）](#)

[视频编码方案之间的比较（HEVC, H.264, MPEG2等）](#)

[MSU 出品的 H.264 编码器比较（2011.5）（包含VP8以及XviD）](#)

### 5.2. 学术研究

#### 5.2.1. 压缩编码与视频质量

[流媒体视频质量评价（单刺激连续质量评价方法）](#)

[一种H.264高清视频的无参考视频质量评价算法（基于QP和跳过宏块数）](#)

[HDTV（1920x1080）码率和视频质量关系的研究 1（前期准备）](#)

[HDTV（1920x1080）码率和视频质量关系的研究 2（实验结果）](#)

[H.264无参考视频质量评价方法（使用了基于遗传编程方法的符号回归）](#)

[基于HEVC的UHD（超高清 4K）视频的主观质量评价](#)

[H.264视频质量评价方法（基于视频内容）](#)

[基于HEVC的UHD（超高清4K）视频质量评价](#)

视频质量，分辨率以及码率之间的关系

视频质量，分辨率，码率之间的关系 2

H.264视频质量评价算法（基于偏最小二乘法回归）

基于运动特征的视频质量评价方法（基于H.264）

IPTV的QoE评价方法

5.2.2其他因素（除了压缩编码之外的因素）与视频质量

Web服务中延时对QoE（体验质量）的影响

网络中延迟对主观感受的影响

基于RTMP的实时流媒体的QoE分析

图像大小，分辨率与人眼主观感受之间的关系（与压缩编码无关）

视频帧率对人眼主观感受的影响

视频帧率对人眼主观感受的影响 2

视频帧率和分辨率对QoE的影响

版权声明：本文为博主原创文章，未经博主允许不得转载。<https://blog.csdn.net/leixiaohua1020/article/details/16359465>

文章标签：[视频](#) [学习](#) [视频质量](#) [图像](#) [编码](#)

个人分类：[视频质量评价](#)

所属专栏：[视频质量评价](#)

此PDF由spygg生成,请尊重原作者版权!!!  
我的邮箱:liushidc@163.com