**音视频基础知识**

**H264码流中 SPS PPS**

**SPS** **0x07**

在H.264标准协议中规定了多种不同的NAL Unit类型，其中类型7表示该NAL Unit内保存的数据为Sequence Paramater Set。

SPS即Sequence Paramater Set，又称作**序列参数集**。SPS中保存了一组编码视频序列(Coded

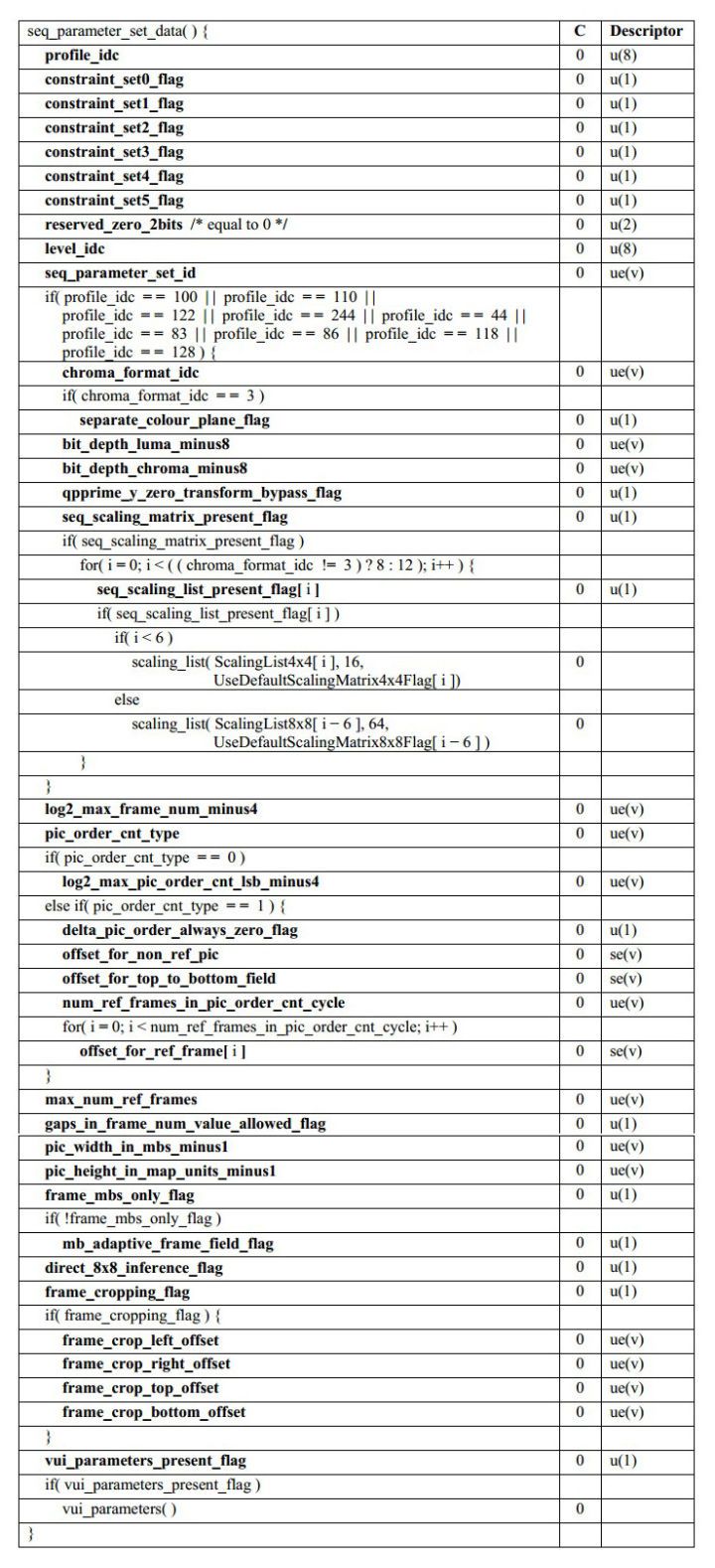
video sequence)的全局参数。所谓的编码视频序列即原始视频的一帧一帧的像素数据经过编码之后的结构组成的序列。而每一帧的编码后数据所依赖的参数保存于图像参数集中。一般情况SPS和PPS的NAL Unit通常位于整个码流的起始位置。但在某些特殊情况下，在码流中间也可能出现这两种结构，主要原因可能为：

解码器需要在码流中间开始解码；

编码器在编码的过程中改变了码流的参数（如图像分辨率等）；

在做视频播放器时，为了让后续的解码过程可以使用SPS中包含的参数，必须对其中的数据进行解析。包含视频流的宽、高、帧率等信息。

H.264标准协议中规定的SPS格式位于文档的7.3.2.1.1部分。



其中的每一个语法元素及其含义如下：

1. profile\_idc

标识当前H.264码流的profile。我们知道，H.264中定义了三种常用的档次profile：

基准档次：baseline profile;

主要档次：main profile;

扩展档次：extended profile;

在H.264的SPS中，第一个字节表示profile\_idc，根据profile\_idc的值可以确定码流符合哪一种档次。判断规律为：

profile\_idc = 66 → baseline profile;

profile\_idc = 77 → main profile;

profile\_idc = 88 → extended profile;

在新版的标准中，还包括了High、High 10、High 4:2:2、High 4:4:4、High 10 Intra、High 4:2:2 Intra、High 4:4:4 Intra、CAVLC 4:4:4 Intra等，每一种都由不同的profile\_idc表示。

另外，constraint\_set0\_flag ~ constraint\_set5\_flag是在编码的档次方面对码流增加的其他一些额外限制性条件。

1. level\_idc

标识当前码流的Level。编码的Level定义了某种条件下的最大视频分辨率、最大视频帧率等参数，码流所遵从的level由level\_idc指定。

1. seq\_parameter\_set\_id

表示当前的序列参数集的id。通过该id值，图像参数集pps可以引用其代表的sps中的参数。

1. log2\_max\_frame\_num\_minus4

用于计算MaxFrameNum的值。计算公式为MaxFrameNum = 2^(log2\_max\_frame\_num\_minus4 +4)。MaxFrameNum是frame\_num的上限值，frame\_num是图像序号的一种表示方法，在帧间编码中常用作一种参考帧标记的手段。

1. pic\_order\_cnt\_type

表示解码picture order count(POC)的方法。POC是另一种计量图像序号的方式，与frame\_num有着不同的计算方法。该语法元素的取值为0、1或2。

1. log2\_max\_pic\_order\_cnt\_lsb\_minus4

用于计算MaxPicOrderCntLsb的值，该值表示POC的上限。计算方法为MaxPicOrderCntLsb = 2^(log2\_max\_pic\_order\_cnt\_lsb\_minus4 + 4)。

1. max\_num\_ref\_frames

用于表示参考帧的最大数目。

1. gaps\_in\_frame\_num\_value\_allowed\_flag

标识位，说明frame\_num中是否允许不连续的值。

1. pic\_width\_in\_mbs\_minus1

用于计算图像的宽度。单位为宏块个数，因此图像的实际宽度为:

frame\_width = 16 × (pic\\_width\\_in\\_mbs\_minus1 + 1);

1. pic\_height\_in\_map\_units\_minus1

使用PicHeightInMapUnits来度量视频中一帧图像的高度。PicHeightInMapUnits并非图像明确的以像素或宏块为单位的高度，而需要考虑该宏块是帧编码或场编码。PicHeightInMapUnits的计算方式为：

PicHeightInMapUnits = pic\\_height\\_in\\_map\\_units\\_minus1 + 1;

1. frame\_mbs\_only\_flag

标识位，说明宏块的编码方式。当该标识位为0时，宏块可能为帧编码或场编码；该标识位为1时，所有宏块都采用帧编码。根据该标识位取值不同，PicHeightInMapUnits的含义也不同，为0时表示一场数据按宏块计算的高度，为1时表示一帧数据按宏块计算的高度。按照宏块计算的图像实际高度FrameHeightInMbs的计算方法为：

FrameHeightInMbs = ( 2 − frame\_mbs\_only\_flag ) \* PicHeightInMapUnits

1. mb\_adaptive\_frame\_field\_flag

标识位，说明是否采用了宏块级的帧场自适应编码。当该标识位为0时，不存在帧编码和场编码之间的切换；当标识位为1时，宏块可能在帧编码和场编码模式之间进行选择。

1. direct\_8x8\_inference\_flag

标识位，用于B\_Skip、B\_Direct模式运动矢量的推导计算。

1. frame\_cropping\_flag

标识位，说明是否需要对输出的图像帧进行裁剪。

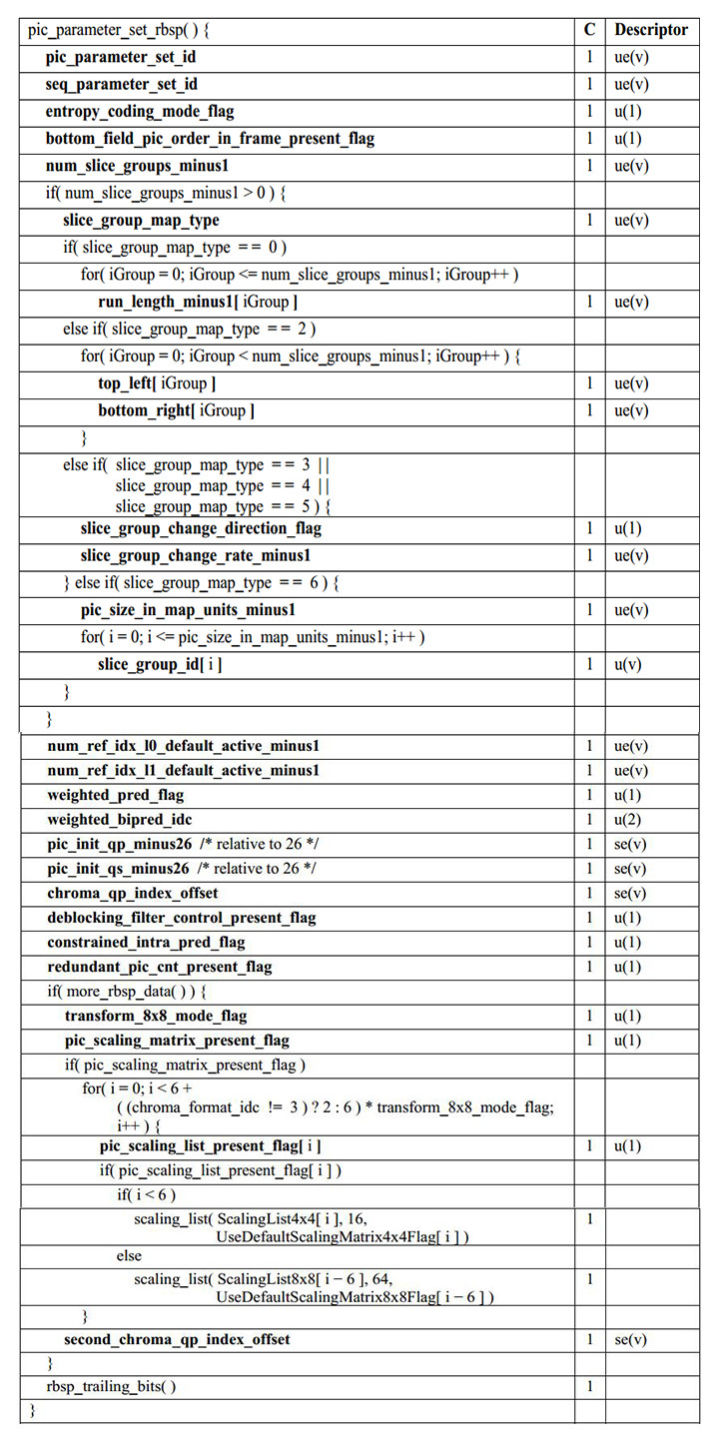
1. vui\_parameters\_present\_flag

标识位，说明SPS中是否存在VUI信息

**PPS 0x08**

H.264中另一重要的参数集合为图像参数集Picture Paramater Set(PPS)。通常情况下，PPS类似于SPS，在H.264的裸码流中单独保存在一个NAL Unit中，只是PPS NAL Unit的nal\_unit\_type值为8；而在封装格式中，PPS通常与SPS一起，保存在视频文件的文件头中。

在H.264的协议文档中，PPS的结构定义在7.3.2.2节中。



其中的每一个语法元素及其含义如下：

1. pic\_parameter\_set\_id

表示当前PPS的id。某个PPS在码流中会被相应的slice引用，slice引用PPS的方式就是在Slice header中保存PPS的id值。该值的取值范围为[0,255]。

1. seq\_parameter\_set\_id

表示当前PPS所引用的激活的SPS的id。通过这种方式，PPS中也可以取到对应SPS中的参数。该值的取值范围为[0,31]。

1. entropy\_coding\_mode\_flag

熵编码模式标识，该标识位表示码流中熵编码/解码选择的算法。对于部分语法元素，在不同的编码配置下，选择的熵编码方式不同。例如在一个宏块语法元素中，宏块类型mb\_type的语法元素描述符为“ue(v) | ae(v)”，在baseline profile等设置下采用指数哥伦布编码，在main profile等设置下采用CABAC编码。

标识位entropy\_coding\_mode\_flag的作用就是控制这种算法选择。当该值为0时，选择左边的算法，通常为指数哥伦布编码或者CAVLC；当该值为1时，选择右边的算法，通常为CABAC。

1. bottom\_field\_pic\_order\_in\_frame\_present\_flag

标识位，用于表示另外条带头中的两个语法元素delta\_pic\_order\_cnt\_bottom和delta\_pic\_order\_cn是否存在的标识。这两个语法元素表示了某一帧的底场的POC的计算方法。

1. num\_slice\_groups\_minus1

表示某一帧中slice group的个数。当该值为0时，一帧中所有的slice都属于一个slice group。slice group是一帧中宏块的组合方式，定义在协议文档的3.141部分。

1. num\_ref\_idx\_l0\_default\_active\_minus1、num\_ref\_idx\_l0\_default\_active\_minus1

表示当Slice Header中的num\_ref\_idx\_active\_override\_flag标识位为0时，P/SP/B slice的语法元素num\_ref\_idx\_l0\_active\_minus1和num\_ref\_idx\_l1\_active\_minus1的默认值。

1. weighted\_pred\_flag

标识位，表示在P/SP slice中是否开启加权预测。

1. weighted\_bipred\_idc

表示在B Slice中加权预测的方法，取值范围为[0,2]。0表示默认加权预测，1表示显式加权预测，2表示隐式加权预测。

1. pic\_init\_qp\_minus26和pic\_init\_qs\_minus26

表示初始的量化参数。实际的量化参数由该参数、slice header中的slice\_qp\_delta/slice\_qs\_delta计算得到。

1. chroma\_qp\_index\_offset

用于计算色度分量的量化参数，取值范围为[-12,12]

1. deblocking\_filter\_control\_present\_flag

标识位，用于表示Slice header中是否存在用于去块滤波器控制的信息。当该标志位为1时，slice header中包含去块滤波相应的信息；当该标识位为0时，slice header中没有相应的信息。

1. constrained\_intra\_pred\_flag

若该标识为1，表示I宏块在进行帧内预测时只能使用来自I和SI类型宏块的信息；若该标识位0，表示I宏块可以使用来自Inter类型宏块的信息。

1. redundant\_pic\_cnt\_present\_flag

标识位，用于表示Slice header中是否存在redundant\_pic\_cnt语法元素。当该标志位为1时，slice header中包含redundant\_pic\_cnt；当该标识位为0时，slice header中没有相应的信息。

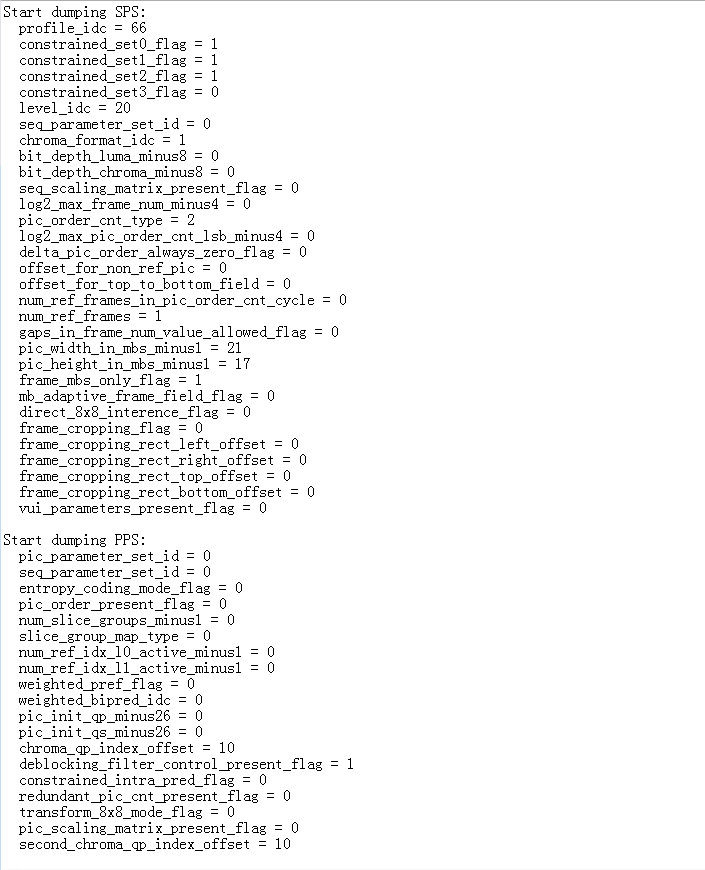
**解析SDP中包含的 H.264的 SPS 和PPS串**

用RTP传输H264的时候,需要用到sdp协议描述,其中有两项:Sequence Parameter Sets (SPS) 和Picture Parameter Set (PPS)需要用到,那么这两项从哪里获取呢?答案是从H264码流中获取.在H264码流中,都是以"0x00 0x00 0x01"或者"0x00 0x00 0x00 0x01"为开始码的,找到开始码之后,使用开始码之后的第一个字节的低5位判断是否为7(sps)或者8(pps), 及data[4] & 0x1f == 7 ||data[4] & 0x1f == 8.然后对获取的nal去掉开始码之后进行base64编码,得到的信息就可以用于sdp.sps和pps需要用逗号分隔开来。

SDP中的H.264的SPS和PPS串，包含了初始化H.264解码器所需要的信息参数，包括编码所用的profile，level，图像的宽和高，deblock滤波器等。

SDP中的SPS和PPS都是**BASE64编码**形式的，不容易理解，需要解析才能看到具体值。

示例：



需要特别提一下这两个参数

pic\_width\_in\_mbs\_minus1 = 21

pic\_height\_in\_mbs\_minus1 = 17

分别表示图像的宽和高，以宏块（16x16）为单位的值减1。因此，实际的宽为 (21+1)\*16 = 352 高为 (17+1)\*16 = 288。

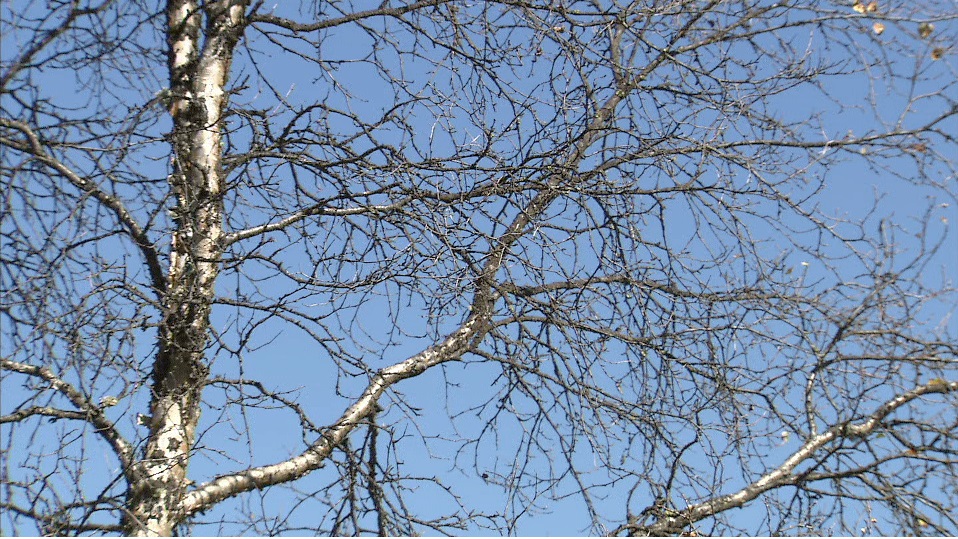
**视频特性TI（时间信息）和SI（空间信息）**

SI表征一帧图像的空间细节量。空间上越复杂的场景，SI值越高。空间复杂度。

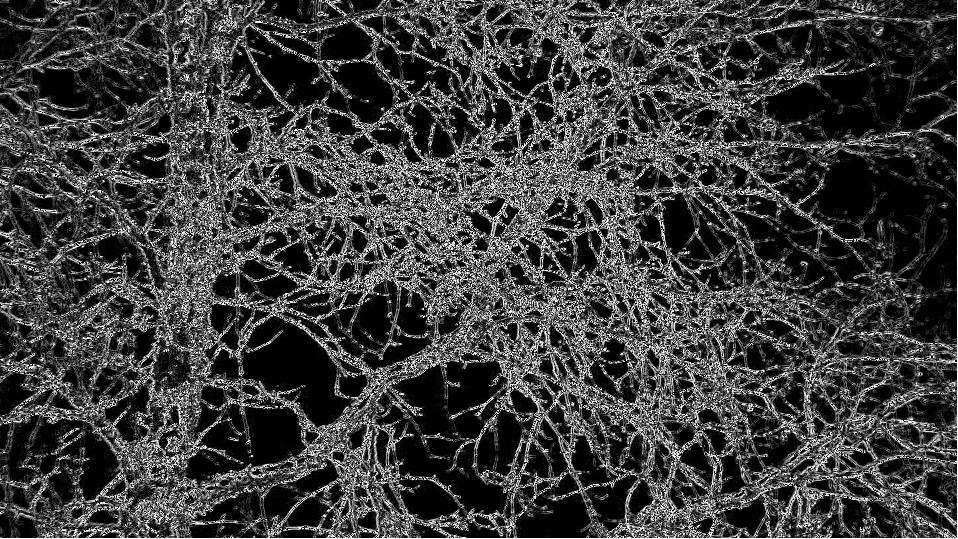
SI计算方法：对第n帧视频进行Sobel滤波（反应空间信息），然后对滤波后图像计算标准差。选这些帧中的最大值为SI。

TI表征视频序列的时间变化量。运动程度较高的序列通常会有更高的TI值。时间复杂度。

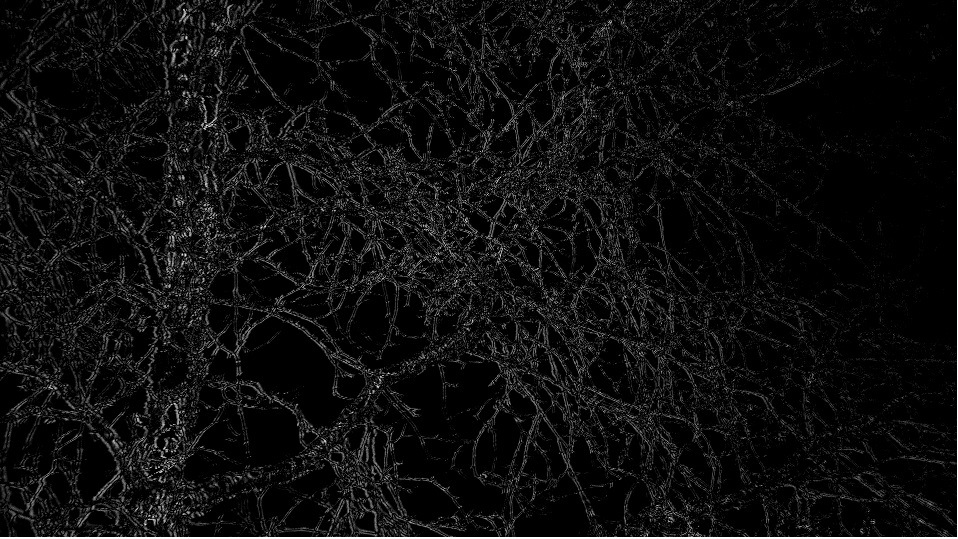
TI计算方法：求n与n-1帧图像的帧差（反应时间信息），然后对帧差图像计算标准差。选这些帧中的最大值为TI。



Sobel滤波后结果（反映空间信息，即空间上的复杂度）



帧差的结果（反映时间信息，即时间上的复杂度）



**工具**

可以直接使用的exe：<http://download.csdn.net/detail/leixiaohua1020/7817721>

源代码（VS2010）：<http://download.csdn.net/detail/leixiaohua1020/7817741>

**附**

**视频参数对比**

**流媒体系统对比**

http://en.wikipedia.org/wiki/Comparison\_of\_streaming\_media\_systems

**封装格式对比**

http://en.wikipedia.org/wiki/Comparison\_of\_container\_formats

**视频编码器对比**

http://en.wikipedia.org/wiki/Comparison\_of\_video\_codecs

**音频编码格式对比**

http://en.wikipedia.org/wiki/Comparison\_of\_audio\_formats

**视频播放器对比**

http://en.wikipedia.org/wiki/Comparison\_of\_video\_player\_software