# **【流媒體】H264—MP4格式及在MP4文件中提取H264的SPS、PPS及码流**

**SkySeraph Apr 1st 2012**

Email：skyseraph00@163[.com](mailto:zgzhaobo@gmail.com)

## **一、MP4格式基本概念**

MP4格式对应标准MPEG-4标准(ISO/IEC14496)

## **二、MP4封装格式核心概念**

**1**  MP4封装格式对应标准为 ISO/IEC 14496-12（信息技术 视听对象编码的第12部分: ISO 基本媒体文件格式/Information technology Coding of audio-visual objects Part 12: ISO base media file format）

**附**-- 标准免费下载： Freely Available Standards    <http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html>

**2**  MP4封装格式是基于QuickTime容器格式定义，媒体描述与媒体数据分开，目前被广泛应用于封装h.264视频和ACC音频，是高清视频/HDV的代表。

**3** MP4文件中所有数据都封装在box中（对应QuickTime中的atom），即MP4文件是由若干个box组成，每个box有长度和类型，每个box中还可以包含另外的子box（称container box）。

　一个MP4文件首先会有且只有一个“ftyp”类型的box，作为MP4格式的标志并包含关于文件的一些信息；之后会有且只有一个“moov”类型的box（Movie Box），它是一种container box，子box包含了媒体的metadata信息；MP4文件的媒体数据包含在“mdat”类型的box（Midia Data Box）中，该类型的box也是container box，可以有多个，也可以没有（当媒体数据全部引用其他文件时），媒体数据的结构由metadata进行描述。

**4** **MP4中box存储方式为大端模式**。一般，标准的box开头会有四个字节的box size。

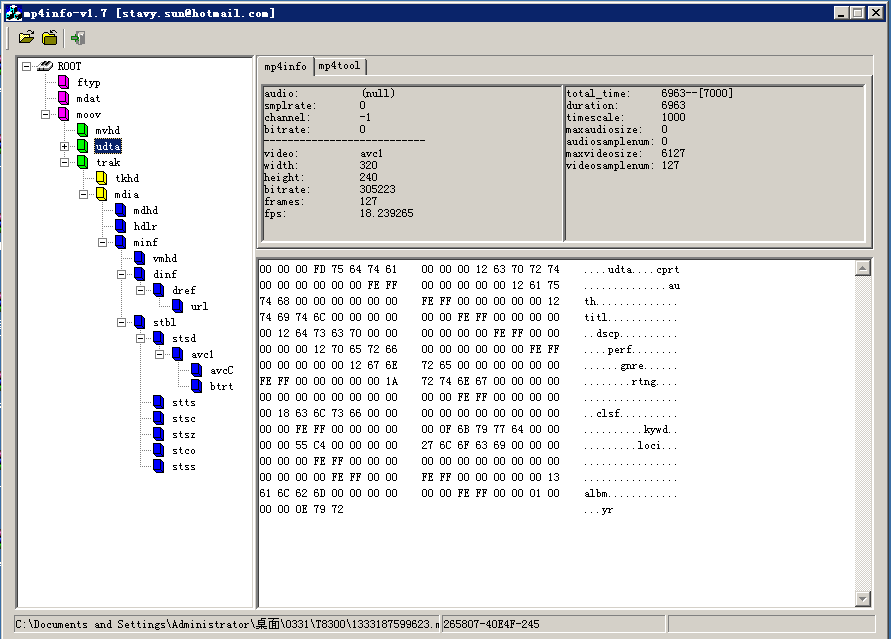
### ****5** 几个名词**

|  |  |
| --- | --- |
| ***track*** | 表示一些sample的集合，对于媒体数据来说，track表示一个视频或音频序列。 |
| ***hint track*** | 特殊的track，并不包含媒体数据，包含的是一些将其他数据track打包成流媒体的指示信息。 |
| ***sample*** | 对于非hint   track来说，video sample即为一帧视频，或一组连续视频帧，audio sample即为一段连续的压缩音频，它们统称sample。  对于hint   track，sample定义一个或多个流媒体包的格式。 |
| ***sample table*** | 指明sampe时序和物理布局的表。 |
| ***chunk*** | 一个track的几个sample组成的单元。 |

## **三、MP4封装格式结构图**

### **1 实例样本**

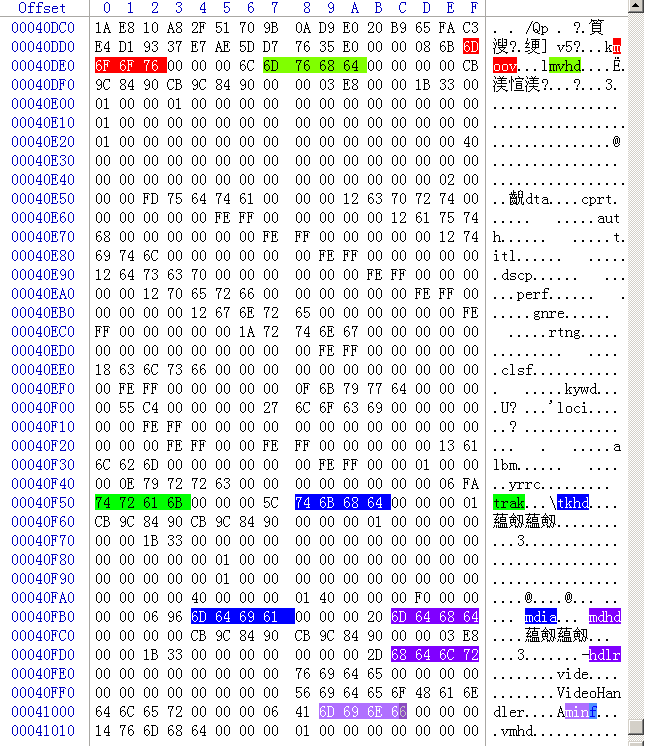
来源于Android MediaRecoder视频录制，平台为华为T8300和TCL968，**用**[mp4info](http://dl.dbank.com/c01jrsr9fq)**查看如下**：



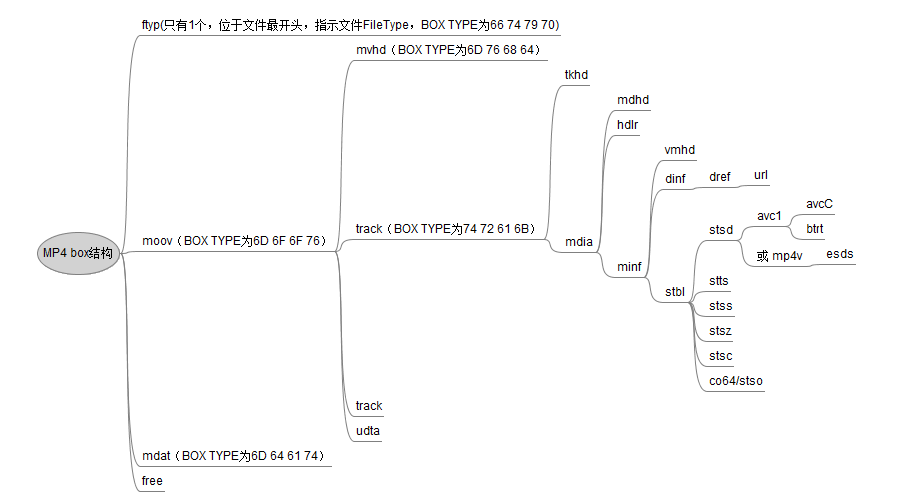
### **用[EsEYE](http://dl.dbank.com/c0dnp0ne0j)查看如下：**

### **IMG_257**

### ****用****[winhex](http://dl.dbank.com/c039tjw92a)****分析如下：****



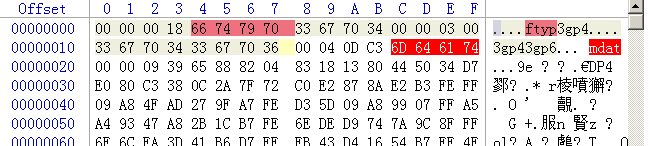
### **2 box结构图**



接下来对h264编码中有用的几个进行阐述，其它不再描述。

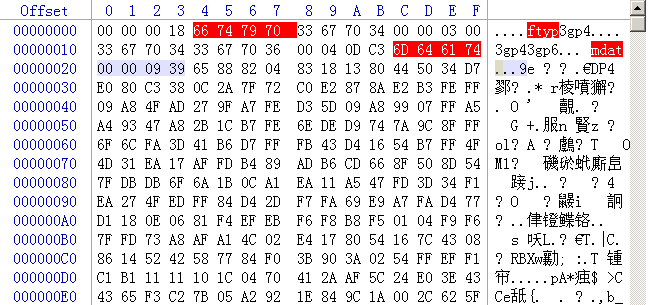
### **3 ftyp（file type box）**

如下图所示，开始的四字节**00 00 00 00 18**表示该box的size为24字节（含头），然后**66 74 79 70**是ftyp的BOX TYPE，其它是一些格式兼容等相关信息。



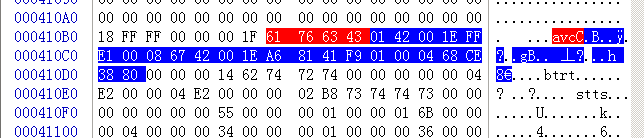
### **4  mdat**

如下图所示，BOX YPE为6D 64 61 74 ，紧接着的00 00 09 39表示sliece长度



### **5 avcC**

如下图所示，红色为BOX TYPE



## **四、MP4文件中h264的 SPS、PPS获取**

**1  【参考依据】**ISO/IEC 14496-15 （[下载](onenote:" \l "MP4\\3GP%E6%A0%BC%E5%BC%8F%E5%8F%8A%E5%9C%A8MP4\\3GP%E4%B8%AD%E6%8F%90%E5%8F%96H264 SPS%E3%80%81PPS%E5%8F%8A%E7%A0%81%E6%B5%81&section-id={E638D2FB-5F30-4C31-B3BA-1C3AD2FD3685}&page-id={B00D8D97-2C79-4EDA-9F2A-302633EE73A7}&object-id={E3EDD90B-A967-4E36-A2B4-644084FA5F07}&79&base-path=E:\\My Document\\OneNote\\%E5%AD%A6%E4%B9%A0%E7%AC%94%E8%AE%B0\\2012-3.one)）

**2  【综述】**在H264中，SPS和PPS存在于NALU header中，而在MP4文件中，SPS和PPS存在于AVCDecoderConfigurationRecord， 首先要定位avcC.

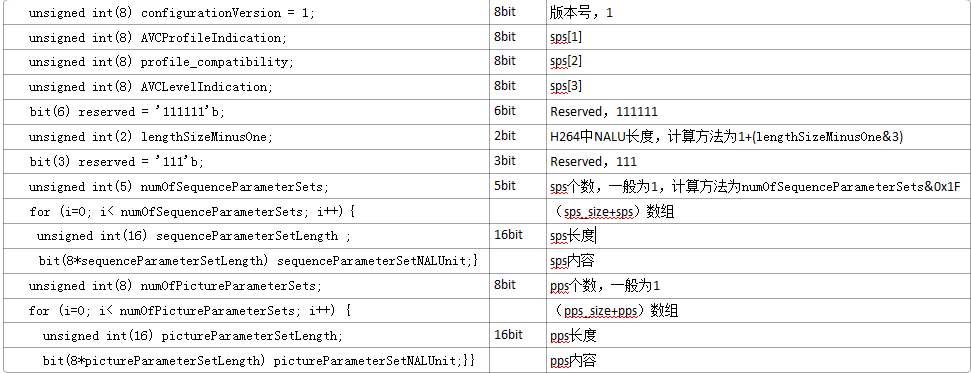
**3  【定义】**

**①参数集：**一组很少改变的，为大量VCL NALU 提供解码信息的数据。

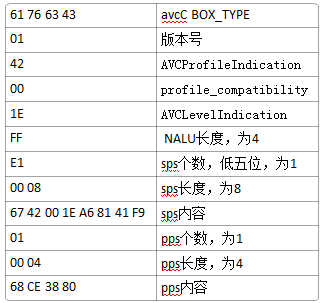
**序列参数集SPS作用于一系列连续的编码图像**，而**图像参数集PPS作用于编码视频序列中一个或多个独立的图像。**

   如果解码器没能正确接收到这两个参数集，那么其他NALU 也是无法解码的。因此它们一般在发送其它 NALU 之前发送，并且使用不同的信道或者更加可靠的传输协议（如TCP）进行传输，也可以重复传输。

**②关于AVCDecoderConfigurationRecord结构定义为**



**4  【实例分析】** 数据如上avcC图所示，现在对数据进行详细分析



所以，提取的SPS和PPS分别为67 42 00 1E A6 81 41 F9和68 CE 38 80

## **五、MP4文件中的H264 data /NALU slice**

**1  【参考】**H264官方文档（[下载](http://dl.dbank.com/c0ng2niprl)） + 毕书—新一代视频压缩编码标准（[下载](http://dl.dbank.com/c0fry4rukk)）

**2  【综述】**

**①**在MP4格式文件中，H264 slice并不是以00 00 00 01来作分割，而是存储在mdat box中。

**②** H264基本码流由一些列的NALU组成。原始的NALU单元组成：[start code] + [NALU header] + [NALU payload]

|  |  |  |  |
| --- | --- | --- | --- |
| start   code | 1字节 | 00 00 01 或 00 00 00 01 | 需要添加的 |
| NALU header | 1字节 | 如下3 | 通过mdat定位 |

**③** H264基本码流结构分两层：视频编码层VCL和网络适配层NAL，这样使信号处理和网路传输分离

|  |  |
| --- | --- |
| VCL | 负责高效视频内容表示 |
| NAL | 以网络所要求的恰当方式对数据进行打包和发送 |

**3  【定义】 NALU header**

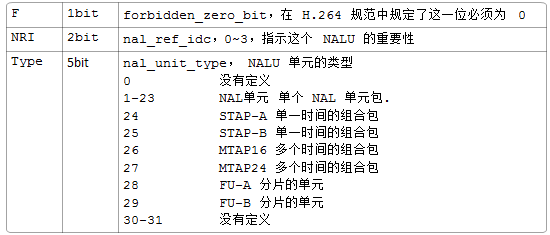
+---------------+

|0|1|2|3|4|5|6|7|

+-+-+-+-+-+-+-+-+

|F|NRI| Type |

+---------------+



**特别的，当值为7和8分别为SPS和PPS。**

**毕书（**[下载](http://dl.dbank.com/c0fry4rukk)**）（P191）上的定义为：**



**4  【实例分析】数据分析，**数据如上图mdat所示

|  |  |
| --- | --- |
| 6D 64 61 74 | mdat   BOX TYPE |
| 00 00   09 39 | silce长度，2361 |

接下来的65就是NALU header，可以由65&0x1F来求的后五个bit，从而得知此slice为I frame

**注意**，mdat与silce之间有可能存在若干占位符，我在TCL手机测试时就出现了连续的00的占位符，这样后面用H264硬编码时会比较麻烦一点。

## **Ref/Related**

1 相关资料和工具在文中链接下载

2 <http://www.52rd.com/Blog/wqyuwss/559/4/>

3 <http://blog.csdn.net/szu030606/article/details/5943279>

4 <http://blog.csdn.net/k1988/article/details/5654631>

5 <http://www.cppblog.com/czanyou/archive/2008/11/26/67940.html>

6 <http://krdai.info/blog/sps-pps-in-mp4-format.html>

7 <http://www.cnitblog.com/zouzheng/archive/2007/04/04/25155.html>

8 <http://bbs.chinavideo.org/viewthread.php?tid=10273>