PMP 文件格式解析

陈小川

著名的 jonny 设计了 PMP 格式，让大家可以使用 PSP 来欣赏到高质量的影音。但随着版本的推进，出现了 PMP1.0, PMP2.0, PMP-AVC, 这些名词和概念可能让大家感到很困惑，本文将对 PMP 格式做一个详细的分析

【PMP 格式】

PMP 是 jonny 设计的视频音频封装格式，如同 AVI 一样，只是一种封装的容器格式。

PMP 格式现在只有两个版本，一个是 1.0, 一个是 2.0

【PMP 1.0】

PMP1.0 是一个比较简陋的版本，只支持封装 MP4V 流(xvid, divx)和 MP3 流(而且要求是 CBR 的,采样率为 44100)。

从文件头开始，顺序结构如下：

CODE:

4 字节：一定是"pmpm"，pmp 的标志

4 字节：版本号,为 0，表示版本是 1.0

4 字节：视频帧总数

4 字节：视频宽度

4 字节：视频高度

4 字节：视频 scale

4 字节：视频 rate ，注意：视频的帧率 fps = rate/scale

4 字节：最大视频帧的大小

视频帧总数×4 个字节：每帧视频的索引，每个索引 4 个字节，最低一个 bit 位表示是否关键帧，其余的 31 位 bit 表示帧的大小。

视频数据：字节数，由上面的索引计算可以得出。

4 个字节：音频帧的数据大小（由于采用的是 cbr 模式，所有的音频帧数据大小一样，但关键帧的大小比普通帧大小多一个字节）

4 个字节：音频帧的总数

音频帧总数×1 个字节：每帧音频的索引，每个索引 1 个字节，关键帧为 1，普通帧为 0，注意：每帧的实际大小=音频帧的数据大小+索引值。

音频数据：字节数，由上面的索引计算得出。

从上面的分析来看，PMP1.0 格式有很大的不足，没有视频和音频的标志位，也就是说，固定死了视频和音频的编码格式，无法封装各式的流，同时，由于视频和音频数据是非交错存储，播放程序在回放的时候，文件指针在来回地移动，读视频帧的时候移到前面，读音频帧又移到后面。

【PMP 2.0】

PMP2.0 开始，jonny 估计意识到 1.0 的不足，重新设计了文件的格式结构，个人认为

这个改变很不错：

从文件头开始，顺序结构如下：

CODE:

4 字节：一定是"pmpm"，pmp 的标志

4 字节：版本号,为 1，表示版本是 2.0

4 字节：视频格式标志，这是一个改进，为支持封装各种视频流提供保证，0 表示 MP 4V 流(xvid，divx)，1 表示 AVC 流(PMP-AVC 其实就是 PMP2.0 格式，只不过封装了 AVC 流)

4 字节：视频帧总数

4 字节：视频宽度

4 字节：视频高度

4 字节：视频 scale

4 字节：视频 rate ，注意：视频的帧率 fps = rate/scale

4 字节：音频格式标志，同样为了以后支持封装各种音频提供保证，现在只支持 mp3 流，该值为 0；

4 字节：包含的音频流数量，为支持多音轨封装提供了保证，如果一个 pmp 中封装了两条音轨，该值就为 2

4 字节：每帧视频附带的音频帧的最大数，由于 PMP2.0 采用了视频音频交错存储的方式，一帧视频和相应的几帧音频放在一起，这是一个最大值；

4 字节：音频 scale,默认为 1152

4 字节：音频 rate,默认为 44100

4 字节：音频是否立体声，0 表示单声，1 表示立体声

视频帧总数×4 个字节：每帧视频的索引，每个索引 4 个字节，最低一个 bit 位表示是否关键帧，其余的 31 位 bit 表示帧的大小（注意，这里的帧大小和 1.0 格式的帧大小不一样，看下面的解释）

视频音频混和数据：这里，jonny 做了一个比较有意思的设计，把 1 个视频帧和其相应的音频帧混合起来，当做一个数据帧；那究竟怎么个混合法呢，我们分析一下：

首先计算每个视频帧的时间戳和每个音频帧的时间戳，videotime = videonum / videofps ，audiotime = audionum / audiofps（videonum 和 audionum 都从 0 开始)

那么第一个视频帧的时间戳就是 0/videofps =0,而第二个视频帧的时间戳就是 1/videofps ,然后，把第一个视频帧和所有时间戳小于 1/videofps 的音频帧接在一起，成了一个数据帧，如此类推下去；

然后在每个数据帧的前面再加上 n 个字节， n=1＋4＋4＋4＋4×每音轨被混合的帧数×音轨数,

其中这 n 个字节的意义如下：

1 字节：本数据帧中，每音轨被混合的帧数；

4 字节：被混合的第一个音频帧和视频帧之间的时间差；

4 字节：被混合的最后一个音频帧和视频帧之间的时间差；

4 字节：视频帧的大小;

4×每音轨被混合的帧数×音轨数：每个被混合的音频帧的大小