编码原理以及I帧B帧P帧

# 一、前言

H264是新一代的编码标准，以高压缩高质量和支持多种网络的流媒体传输著称。在编码方面，它的理论依据是：参照一段时间内图像的统计结果表明，在相邻几幅图像画面中，一般有差别的像素只有10%以内的点，亮度差值变化不超过2%，而色度差值的变化只有1%以内。所以对于一段变化不大图像画面，我们可以先编码出一个完整的图像帧A，随后的B帧就不编码全部图像，只写入与A帧的差别，这样B帧的大小就只有完整帧的1/10或更小，B帧之后的C帧如果变化不大，我们可以继续以参考B的方式编码C帧，这样循环下去。这段图像我们称为一个序列（序列就是有相同特点的一段数据），当某个图像与之前的图像变化很大，无法参考前面的帧来生成，那我们就结束上一个序列，开始下一段序列，也就是对这个图像生成一个完整帧A1，随后的图像就参考A1生成，只写入与A1的差别内容。

在H264协议里定义了三种帧，完整编码的帧叫I帧，参考之前的I帧生成的只包含差异部分编码的帧叫P帧，还有一种参考前后的帧编码的帧叫B帧。

H264采用的核心算法是帧内压缩和帧间压缩，帧内压缩是生成I帧的算法，帧间压缩是生成B帧和P帧的算法。

# 二、序列的说明

在H.264中图像以序列为单位进行组织，一个序列是一段图像编码后的数据流，以I帧开始，到下一个I帧结束。

一个序列的第一个图像叫做 IDR 图像（立即刷新图像），IDR 图像都是 I 帧图像。H.264 引入 IDR 图像是为了解码的重同步，当解码器解码到 IDR 图像时，立即将参考帧队列清空，将已解码的数据全部输出或抛弃，重新查找参数集，开始一个新的序列。**这样，如果前一个序列出现重大错误，在这里可以获得重新同步的机会。IDR图像之后的图像永远不会使用IDR之前的图像的数据来解码。**

一个序列就是一段内容差异不太大的图像编码后生成的一串数据流。当运动变化比较少时，一个序列可以很长，因为运动变化少就代表图像画面的内容变动很小，所以就可以编一个I帧，然后一直P帧、B帧了。当运动变化多时，可能一个序列就比较短了，比如就包含一个I帧和3、4个P帧。

# 三、三种帧的说明

## 1、I帧

**I帧**：

帧内编码帧，I帧表示关键帧，你可以理解为这一帧画面的完整保留；解码时只需要本帧数据就可以完成（因为包含完整画面）

**I帧特点：**

1）它是一个全帧压缩编码帧。它将全帧图像信息进行JPEG压缩编码及传输；

2）解码时仅用I帧的数据就可重构完整图像；

3）I帧描述了图像背景和运动主体的详情；

4）I帧不需要参考其他画面而生成；

5）I帧是P帧和B帧的参考帧(其质量直接影响到同组中以后各帧的质量)；

6）I帧是帧组GOP的基础帧(第一帧)，在一组中只有一个I帧；

7）I帧不需要考虑运动矢量；

8）I帧所占数据的信息量比较大。

## 2、P帧

**P帧：**

前向预测编码帧。P帧表示的是这一帧跟之前的一个关键帧（或P帧）的差别，解码时需要用之前缓存的画面叠加上本帧定义的差别，生成最终画面。（也就是差别帧，P帧没有完整画面数据，只有与前一帧的画面差别的数据）

**P帧的预测与重构：**

P帧是以I帧为参考帧，在I帧中找出P帧“某点”的预测值和运动矢量，取预测差值和运动矢量一起传送。在接收端根据运动矢量从I帧中找出P帧“某点”的预测值并与差值相加以得到P帧“某点”样值，从而可得到完整的P帧。

**P帧特点：**

1）P帧是I帧后面相隔1~2帧的编码帧；

2）P帧采用运动补偿的方法传送它与前面的I或P帧的差值及运动矢量(预测误差)；

3）解码时必须将I帧中的预测值与预测误差求和后才能重构完整的P帧图像；

4）P帧属于前向预测的帧间编码。它只参考前面最靠近它的I帧或P帧；

5）P帧可以是其后面P帧的参考帧，也可以是其前后的B帧的参考帧；

6）由于P帧是参考帧，它可能造成解码错误的扩散；

7）由于是差值传送，P帧的压缩比较高。

## 3、B帧

**B帧：**

双向预测内插编码帧。B帧是双向差别帧，也就是B帧记录的是本帧与前后帧的差别（具体比较复杂，有4种情况），换言之，要解码B帧，不仅要取得之前的缓存画面，还要解码之后的画面，通过前后画面的与本帧数据的叠加取得最终的画面。B帧压缩率高，但是解码时CPU会比较累。

**B帧的预测与重构：**

B帧以前面的I或P帧和后面的P帧为参考帧，“找出”B帧“某点”的预测值和两个运动矢量，并取预测差值和运动矢量传送。接收端根据运动矢量在两个参考帧中“找出(算出)”预测值并与差值求和，得到B帧“某点”样值，从而可得到完整的B帧。

**B帧特点：**

1）B帧是由前面的I或P帧和后面的P帧来进行预测的；

2）B帧传送的是它与前面的I或P帧和后面的P帧之间的预测误差及运动矢量；

3）B帧是双向预测编码帧；

4）B帧压缩比最高，因为它只反映参考帧间运动主体的变化情况，预测比较准确；

5）B帧不是参考帧，不会造成解码错误的扩散。

**注：**I、B、P各帧是根据压缩算法的需要，人为定义的，它们都是实实在在的物理帧。一般来说，I帧的压缩率是7（跟JPG差不多），P帧是20，B帧可以达到50。可见使用B帧能节省大量空间，节省出来的空间可以用来保存多一些I帧，这样在相同码率下，可以提供更好的画质。

# 四、压缩算法的说明

h264的压缩方法：

1.分组：把几帧图像分为一组(GOP，也就是一个序列)，为防止运动变化,帧数不宜取多。

2.定义帧：将每组内各帧图像定义为三种类型,即I帧、B帧和P帧；

3.预测帧：以I帧作为基础帧,以I帧预测P帧,再由I帧和P帧预测B帧；

4.数据传输：最后将I帧数据与预测的差值信息进行存储和传输。

帧内（Intraframe）压缩也称为空间压缩（Spatial compression）。当压缩一帧图像时，仅考虑本帧的数据而不考虑相邻帧之间的冗余信息，这实际上与静态图像压缩类似。帧内一般采用有损压缩算法，由于帧内压缩是编码一个完整的图像，所以可以独立的解码、显示。帧内压缩一般达不到很高的压缩，跟编码jpeg差不多。

帧间（Interframe）压缩的原理是：相邻几帧的数据有很大的相关性，或者说前后两帧信息变化很小的特点。也即连续的视频其相邻帧之间具有冗余信息,根据这一特性，压缩相邻帧之间的冗余量就可以进一步提高压缩量，减小压缩比。帧间压缩也称为时间压缩（Temporal compression），它通过比较时间轴上不同帧之间的数据进行压缩。帧间压缩一般是无损的。帧差值（Frame differencing）算法是一种典型的时间压缩法，它通过比较本帧与相邻帧之间的差异，仅记录本帧与其相邻帧的差值，这样可以大大减少数据量。

顺便说下有损（Lossy ）压缩和无损（Lossy less）压缩。无损压缩也即压缩前和解压缩后的数据完全一致。多数的无损压缩都采用RLE行程编码算法。有损压缩意味着解压缩后的数据与压缩前的数据不一致。在压缩的过程中要丢失一些人眼和人耳所不敏感的图像或音频信息,而且丢失的信息不可恢复。几乎所有高压缩的算法都采用有损压缩,这样才能达到低数据率的目标。丢失的数据率与压缩比有关,压缩比越小，丢失的数据越多,解压缩后的效果一般越差。此外,某些有损压缩算法采用多次重复压缩的方式,这样还会引起额外的数据丢失。