视频压缩

在视频压缩中，我们常常把视频帧分为三种：一种是独立的关键帧，叫做I帧，另一种则是需要依赖I帧的p帧，还有一种则称为B帧，双向帧。

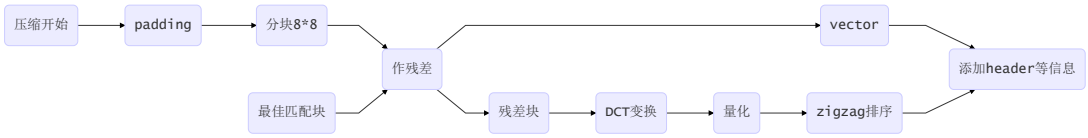
I帧压缩

在处理一般的I帧压缩时，一般情况下，我们需要将RGB图像进行转化，考虑到人眼对图像的色彩信息不太敏感，所以我们讲RGB图像转化成YUV图像，这样可以减少冗余信息，增加压缩率。

这里我们为了简化，直接使用灰度图，就跳过了这个过程。之后的过程就和上述的JPEG压缩过程一致。在此我们不再重复。

P帧压缩

p帧通过降低图像序列中前面已经编码中的时间冗余信息来压缩数据量，也可以叫做预测帧。根据本帧与前一帧的不同点来压缩本帧数据。其工作流程图如下：



在匹配最佳块时，有几种计算方法： MAD、SAD 、SSD。

MAD：平均绝对失真:

$$MAD(x,y)=\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}|f(x+i,y+i)-p(x+i+dx,y+j+dy)|$$

SAD：绝对失真：

$$SAD(x,y)=\sum_{i=1}^{m}\sum_{j=1}^{n}|f(x+i,y+i)-p(x+i+dx,y+j+dy)|$$

SSD：平方误差和：

$$SSD(x,y)=\sum_{i=1}^{m}\sum_{j=1}^{n}|f(x+i,y+i)-p(x+i+dx,y+j+dy)|^2$$

另列一下PSNR（峰值信噪比）的计算公式：

$$PSNR=10log_{10}\sum_{x=0}^{M-1}\sum_{y=o}^{N-1}\frac{255^2MN}{|f(x,y)-g(x,y)|^2}$$

问题关键：找最佳匹配块

为了便于匹配，我们对I帧进行padding

```
i =padarray(i,[R,R],'replicate','both'); % I帧周围填充便于P帧边界搜索
rimage = padarray(rimage,[R,R],'replicate','both');
```

找最佳匹配的过程实际上是一个优化问题，那就应首先明确我们的优化目标:块之间的灰度差最小。

我们可以采用上述的计算方法来明确我们的优化目标，本程序采用SSD方法。

```
        error = SSD(pblock,Iblock);
        %error = SAD(pblock,Iblock);
    for i=-R+1:R-1%位移整数个像素，可用插值 位移非整数个像素
        for j= -R+1:R-1
                Iblock=I((1+8*(row-1)+i+R:8*row+R+i),(1+8*(col-1)+R+j:8*col+R+j)); % 匹配块
                errorNow = SSD(pblock,Iblock);
                if errorNow < error %优化过程
                    vector = [i,j];
                    error = errorNow;
                end
        end
    end
```

这边我们记录下vector 运动向量，以便解压时候确定匹配块。

在匹配时有多种匹配方法。可以有十字型的匹配，或者是8个方向的匹配，这时候 运动向量记录的是方向和像素值。

在我们程序中我们则采用的是坐标记录：以待压缩块为坐标原点。

找到最佳匹配块后，求待压缩块与最佳匹配块的残差，之后压缩过程则与前所述一致。

解压时，需要先解压I帧才能解压p帧。

具体代码可以参照代码文件。

结果分析

When K=1:
for I_frame :
PSNR 45.3306
Original Bit: 4718592 bit
Compressed Bit: 288368 bit
Compression Ratios for I_frame: 16.3631
时间已过 5.371988 秒。
When K=1:

for P_frame :
PSNR: 44.6034
ImageB Bit: 4718592 bit
PFrame Bit: 187400 bit
Compression Ratios: 25.1793

复原的图像如下：

When K=2:
for I_frame :
PSNR 42.1719
Original Bit: 4718592 bit
Compressed Bit: 208272 bit
Compression Ratios for I_frame: 22.6559
时间已过 5.424953 秒。

When K=2:
for P_frame :
PSNR: 41.8451
ImageB Bit: 4718592 bit
PFrame Bit: 184560 bit
Compression Ratios: 25.5667

我们可以加大K可以发现， P帧的压缩率比较稳定，原因是因为我们是对残差进行压缩，不同压缩率的残差块可能相差并不是很显著，而对于I帧来说，K的提高可以明显提高压缩率。

附录:一些遇到的问题

矩阵除法

数值预算时候 /与. / 没有区别

数值与矩阵运算时
数值在前，只能用. /
数值在后，两者没有区别

矩阵与矩阵运算时
$A/B = A * inv(B)$
$A./B$表示对位相除