

第4步通过将每一块的(0,0)值(左上角元素)以它与前一块中相应元素相差的量替换而减小。由于这些元素是各自所在块的平均,它们应该变化得比较缓慢,所以采用差值可以将这些元素中的大部分缩减为较小的值。对于其他元素不计算差值。(0,0)值称为DC分量,其他值称为AC分量。

第5步是将64个元素线性化并且对线性化得到的列表进行行程长度编码。从左到右然后从上到下地对块进行扫描不能将零集中在一起,所以采用了Z字形的扫描模式,如图7-9所示。在本例中,Z字形模式最终在矩阵的尾部产生了38个连续的0,这一串0可以缩减为一个计数表明有38个0。

现在我们得到一个代表图像的数字列表(在变换空间中),第6步将采用Huffman编码对列表中的数字进行编码以用于存储或传输。

JPEG看来似乎十分复杂,这是因为它确实很复杂。尽管如此,由于它通常可以获得20:1或更好的压缩效果,所以获得广泛的应用。解码一幅JPEG图像需要反过来运行上述算法。JPEG大体上是对称的:解码一幅图像花费的时间与编码基本相同。

|     |    |    |   |   |   |   |   |
|-----|----|----|---|---|---|---|---|
| 150 | 80 | 20 | 4 | 1 | 0 | 0 | 0 |
| 92  | 75 | 16 | 3 | 1 | 0 | 0 | 0 |
| 28  | 19 | 13 | 2 | 1 | 0 | 0 | 0 |
| 3   | 2  | 2  | 1 | 0 | 0 | 0 | 0 |
| 1   | 0  | 0  | 0 | 0 | 0 | 0 | 0 |
| 0   | 0  | 0  | 0 | 0 | 0 | 0 | 0 |
| 0   | 0  | 0  | 0 | 0 | 0 | 0 | 0 |
| 0   | 0  | 0  | 0 | 0 | 0 | 0 | 0 |

图7-9 量化值传送的顺序

### 7.3.2 MPEG标准

最后,我们讨论问题的核心:MPEG(Motion Picture Experts Group,运动图像专家组)标准。这是用于压缩视频的主要算法,并于1993年成为国际标准。MPEG-1(第11172号国际标准)设计用于视频录像机质量的输出(对NTSC制式为 $352 \times 240$ ),它使用的位率为1.2 Mbps。MPEG-2(第13818号国际标准)设计用于将广播质量的视频压缩至4 Mbps到6 Mbps,这样就可以适应NTSC或PAL制式的广播频道。

MPEG的两个版本均利用了电影中存在的两类冗余:空间冗余和时间冗余。空间冗余可以通过简单地用JPEG分别对每一帧进行编码而得到利用。互相连续的帧常常几乎是完全相同的,这就是时间冗余,利用这一事实可以达到额外的压缩效果。数字便携式摄像机使用的数字视频(Digital Video, DV)系统只使用类JPEG的方案,这是因为只单独对每一帧进行编码可以达到更快的速度,从而使编码可以实时完成。这一论断的因果关系可以从图7-2看出:尽管数字便携式摄像机与未压缩电视相比具有较低的数据率,但是却远不及MPEG-2。(为了使比较公平,请注意DV便携式摄像机以8位对亮度、以2位对每一色度进行采样,使用类JPEG编码仍然存在5倍的压缩率。)

对于摄像机和背景绝对静止,而有一两个演员在四周缓慢移动的场景而言,帧与帧之间几乎所有的像素都是相同的。此时,仅仅将每一帧减去前一帧并且在差值图像上运行JPEG就相当不错。然而,对于摇动或缩放摄像机镜头的场景而言,这一技术将变得非常糟糕。此时需要某种方法对这一运动进行补偿,这正是MPEG要做的事情;实际上,这就是MPEG和JPEG之间的主要差别。

MPEG-2输出由三种不同的帧组成,观看程序必须对它们进行处理,这三种帧为:

- 1) I帧:自包含的JPEG编码静止图像。
- 2) P帧:与上一帧逐块的差。
- 3) B帧:与上一帧和下一帧的差。

I帧只是用JPEG编码的静止图像,沿着每一轴还使用了全分辨率的亮度和半分辨率的色度。在输出流中使I帧周期性地出现是十分必要的,其原因有三。首先,MPEG可以用于电视广播,而观众收看是随意的。如果所有的帧都依赖于其前驱直到第一帧,那么错过了第一帧的人就再也无法对随后的帧进行解码,这样使观众在电影开始之后就不能再进行收看。第二,如果任何一帧在接收时出现错误,那么进一步的解码就不可能再进行。第三,没有I帧,在进行快进或倒带时,解码器将不得不计算经过的每一帧,只有这样才能知道快进或倒带停止时帧的全部值。有了I帧,就可以向前或向后跳过若干帧直到找到一个I帧并从那里开始观看。由于上述原因,MPEG每秒将I帧插入到输出中一次或两次。

与此相对照,P帧是对帧间差进行编码。P帧基于宏块(macroblock)的思想,宏块覆盖亮度空间中 $16 \times 16$ 个像素和色度空间中 $8 \times 8$ 个像素。通过在前一帧中搜索宏块或者与其只存在轻微差异的宏块实现