



Research Institute for Future Media Computing Institute of Computer Vision
未来媒体技术与研究所 计算机视觉研究所



多媒体系统导论

Fundamentals of Multimedia System

授课教师：朱映映教授

Email: zhuyy@szu.edu.cn

第十二讲

MPEG Video Coding

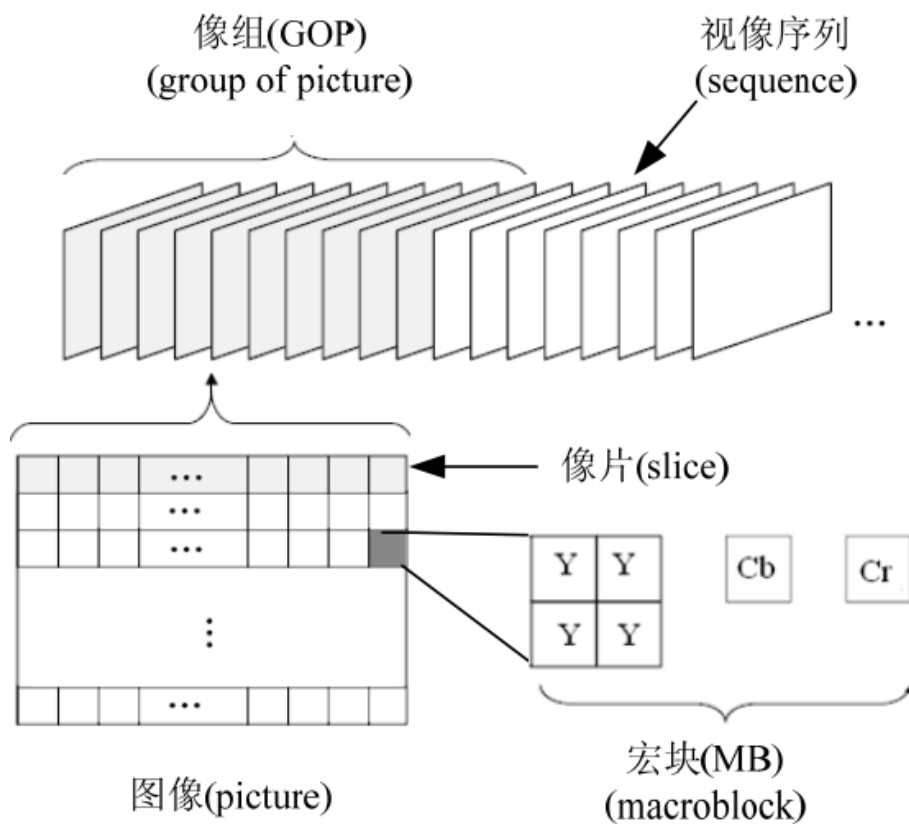
第11章

1. 简介

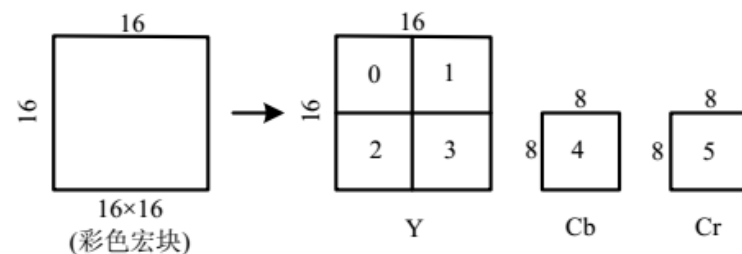
- ◆ MPEG-1视像(MPEG-1 Video)压缩视像数据的基本方法可以归纳成两个要点：
 - 1) 在空间方向上，采用与JPEG类似的算法来去掉空间冗余数据
 - 2) 在时间方向上，采用移动补偿(motion compensation)算法来去掉时间冗余数据
- ◆ MPEG专家组为此开发了两项重要技术
 - 1) 定义了视像数据的结构
 - 2) 定义了三种类型的图像

2. 视像数据结构

- ◆ 把视像片段看成由一系列静态图像(picture)组成的视像序列(sequence); 一个视频序列由一个或多个GOP组成
- ◆ 把视像序列分成许多像组(group of picture, GOP); 一个GOP包含一张或多张图像, 其中一张为I帧。
- ◆ 把像组中的每一帧图像分成许多像片(slice), 每个像片由16行组成;
- ◆ 把像片分成 16×16 像素的宏块(macroblock, MB);
- ◆ 把宏块分成若干个 8×8 像素的图块(block), 见图1(a);
- ◆ 使用子采样格式为4:2:0时, 一个宏块由4个亮度(Y)图块和两个色度图块(Cb和Cr)组成, 见图1(b)。



(a) 视像数据的组织



(方框中的数字为图块编号)

(b) 宏块的结构(4:2:0)

图1 视像数据结构

3. 三种类型图像

MPEG专家组定义了三种类型的图像，然后采用三种不同的算法分别对它们进行压缩

- 帧内图像I (intra- picture), 简称为 I 图像或 I 帧(I picture / I-frame)

包含内容完整的图像，用于为其他帧图像的编码和解码作参考，因此也称为关键帧

- 预测图像P (predicted picture), 简称为 P 图像或 P 帧(P-picture / P-frame)

指以在它之前出现的帧内图像 I 作参考图像的图像，对预测图像 P 进行编码就是对它们之间的差值进行编码

3. 三种类型图像

- 双向预测图像B (bidirectionally-predictive picture), 也称双向插值图像B (bidirectionally interpolated picture), 简称为 B图像或 B帧(B picture/B-frame)
以在它之前和之后的帧图像(I 和 P)作参考的图像, 对B进行编码就是对帧内图像 I 和预测图像 P的差值分别进行编码

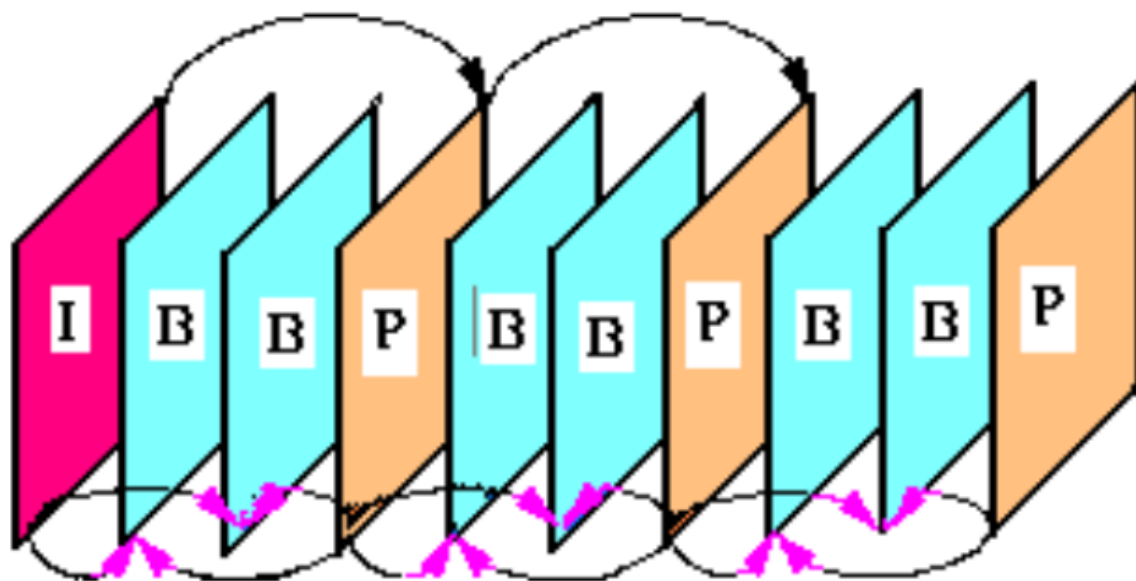


图2 MPEG专家组定义的三种图像

3.1 帧图像的编排顺序

◆ MPEG-1编码器允许选择

- 帧内图像 I 出现的频率和位置。通常的频率为2 Hz
- 在两帧图像 I 之间或在图像 I 和 P 之间选择图像 B 的数目
- 图像I、 P和B的数目主要是根据节目的内容来确定。

例如, 对于快速运动的图像, 帧内图像I的频率可以选择高一些, 双向预测图像B的数目可以选择少一些; 对于慢速运动的图像, 帧内图像I的频率可以低一些, 而双向预测图像B的数目可以选择多一些, 这样可保证视像的质量。

◆ 一个I、 P和B的典型编排顺序见图3

- 编码参数: I 的距离 $N=15$, P 的距离 $M=3$
- 在视像解码时, 因B 需 I和 P做参考, 故在解码之前需重新组织帧图像数据流的输入顺序, 其方案见图4

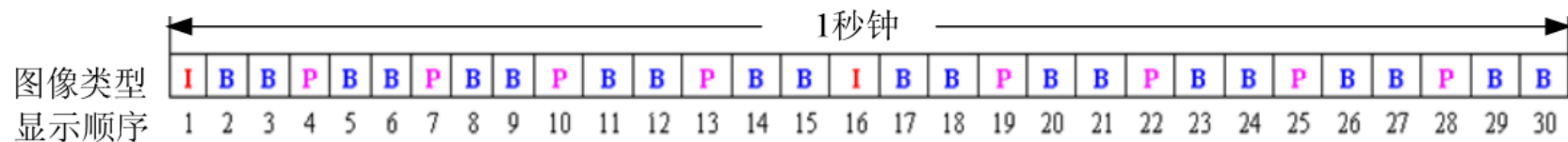


图3 MPEG帧图像的编排示例2

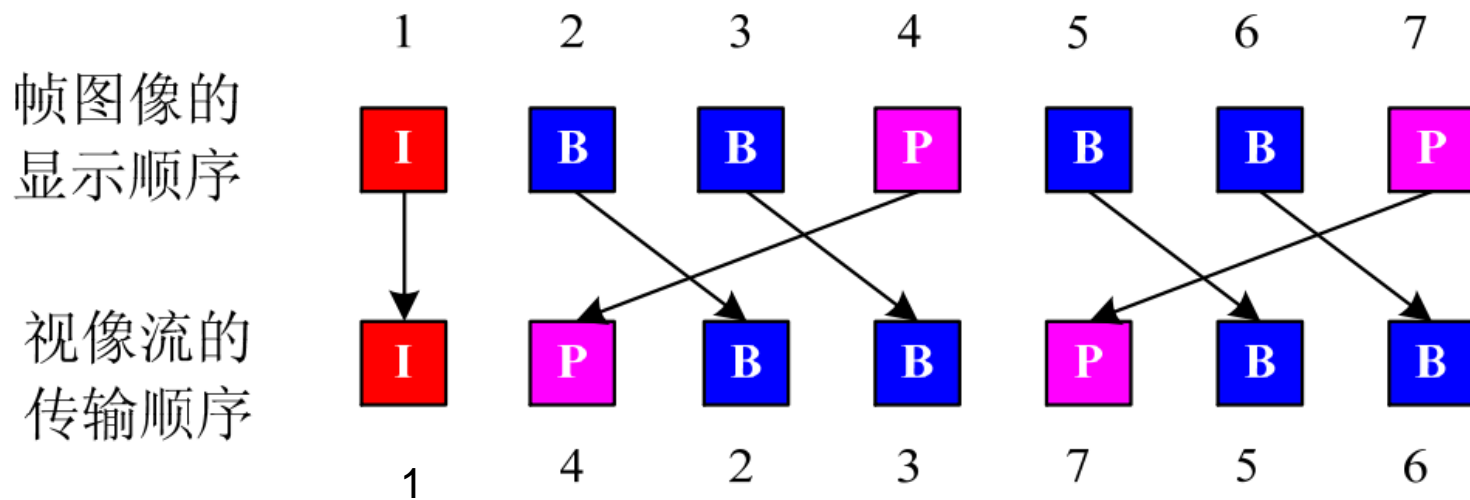
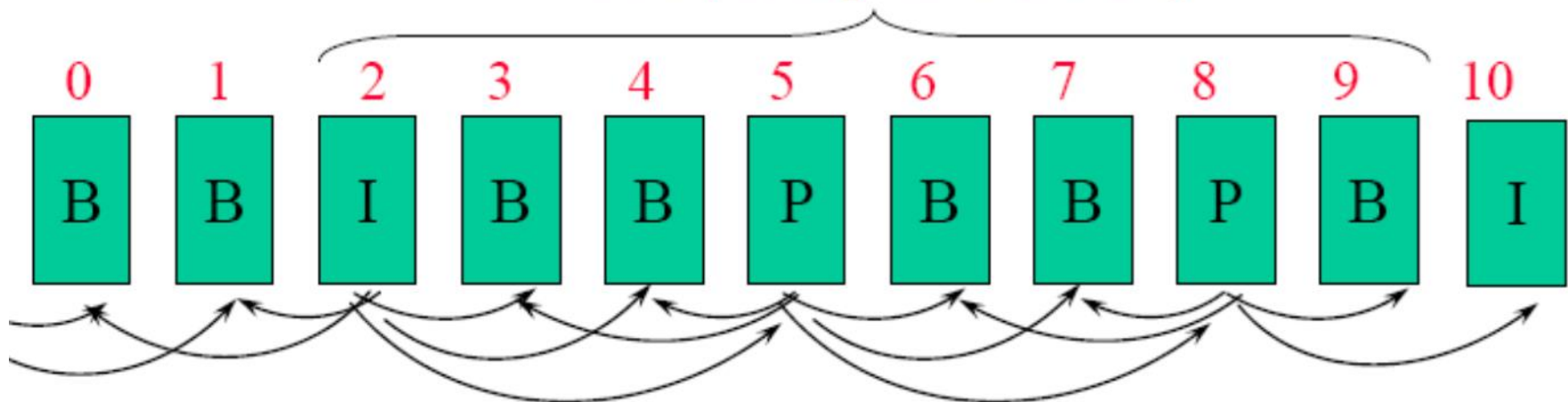


图4 显示顺序与传输顺序

GOP(Group of Pictures)



➤ 画面的显示顺序是：

B B I B B P B B P B I
0 1 2 3 4 5 6 7 8 9 10

➤ 画面的编码顺序是：

I B B P B B P B B I B
2 0 1 5 3 4 8 6 7 10 9

因此, 每一个GOP中的画面在编码前和解码后都必须重排序

图5 MPEG帧图像的编排示例

3.2 帧内图像I的压缩编码算法

- ◆ 不参照过去的帧和将来的帧，采用与JPEG类似的压缩算法以减少空间的冗余数据，见图6

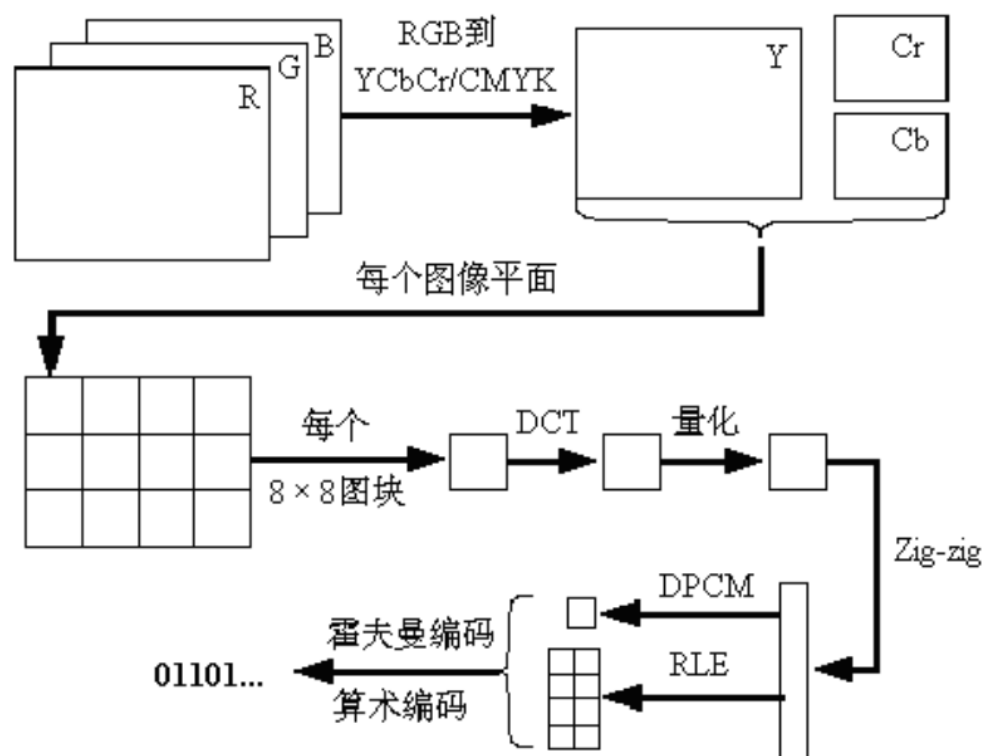


图6 帧内图像 I 的压缩编码算法框图

3.2 帧内图像I的压缩编码算法

- ◆ 如果视像就是用RGB空间表示的视像，则首先把它转换成YCrCb空间表示的视像
- ◆ 每个图像平面分成 8×8 像素的图块，对每个图块进行离散余弦变换(DCT)，变换后产生的交流分量系数经过量化之后按照Zig-zag的形状排序。DCT得到的直流分量系数经过量化之后用差分脉冲编码(DPCM)，交流分量系数用行程长度编码RLE，然后再用霍夫曼(Huffman)编码或者用算术编码

3.3 预测图像P的压缩编码算法

◆ 算法原理

- 预测图像P的编码以宏块(MB)为基本编码单元，一个宏块定义为像素的图块，一般取 16×16
- 预测图像P使用两种类型的参数表示
 - 当前要编码的图像（时刻2）宏块与参考图像（时刻1）的宏块之间的差值
 - 宏块的移动矢量(motion vector, MV)，见图7

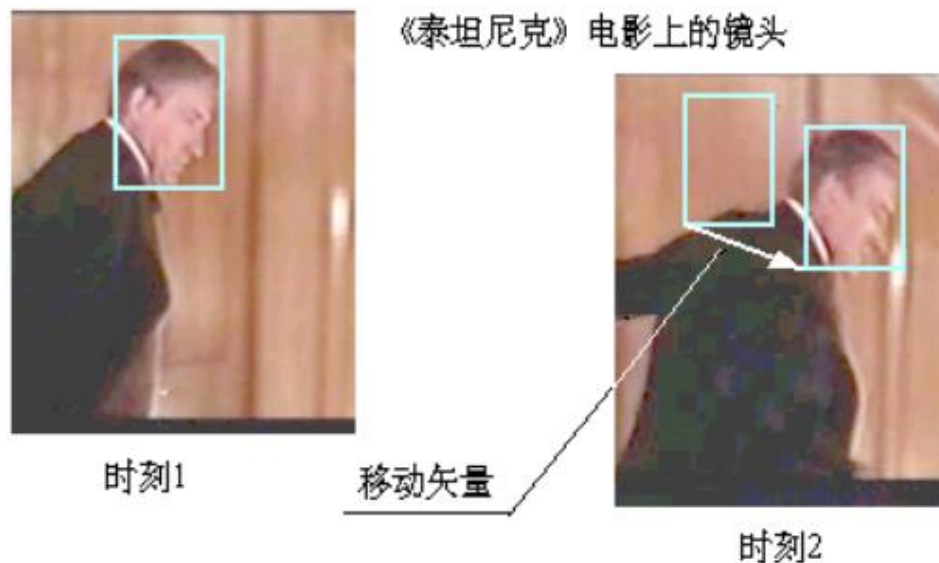


图7 移动矢量

3.3 预测图像P的压缩编码算法

(1)求解差值的方法(见图8)

- ◆ 假设编码宏块 M_{PI} 是参考宏块 M_{RJ} 的最佳匹配块，它们的差值就是这两个宏块中相应的像素值之差。
- ◆ 对所求得的差值进行彩色空间转换，然后使用4:1:1或4:2:0格式采样。对采样得到的Y, Cr和Cb分量值，仿照JPEG压缩算法对差值进行编码。
- ◆ 对计算出的移动矢量进行DCT变换和霍夫曼编码

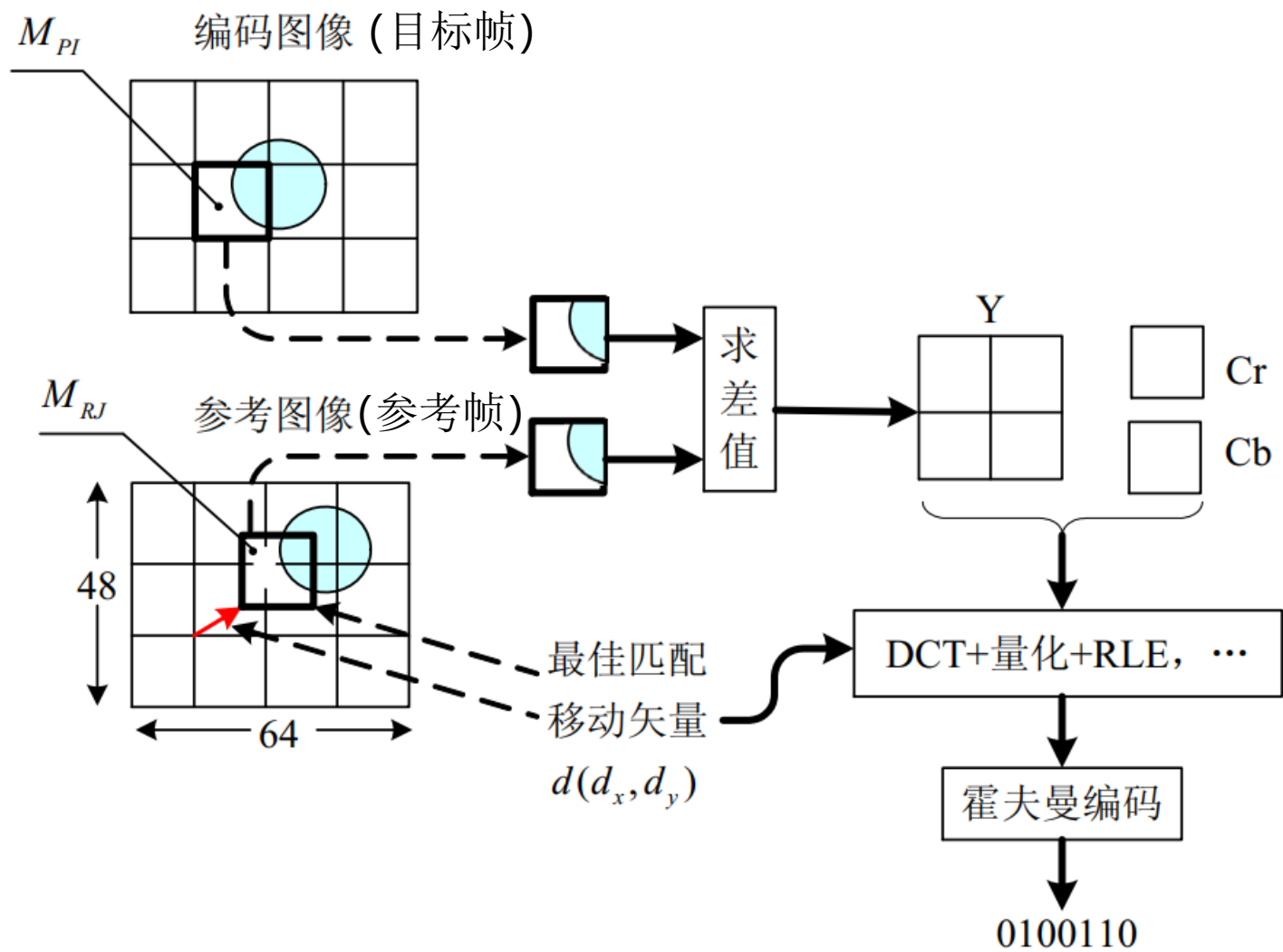


图8 预测图像P的压缩编码算法框图

3.3 预测图像P的压缩编码算法

(2)求解运动向量

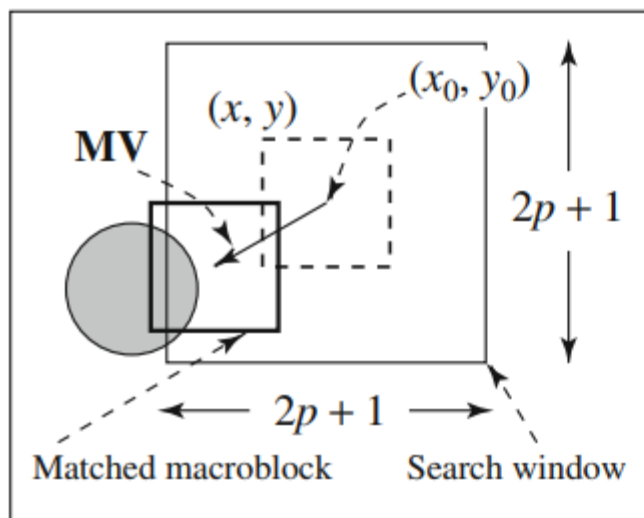
- ◆ 在求两个宏块差值之前，需要找出预测编码图像中的编码宏块 M_{P_l} 相对于参考图像中的参考宏块 M_{R_j} 所移动的距离和方向，即运动向量。
- ◆ 求解的目标是找到一个向量 (i, j) 作为运动向量 $MV=(u, v)$ ，使 $MAD(i, j)$ 取最小值。 MAD (Mean Absolute Difference)用于表示两个宏块的差

$$(u, v) = [(i, j) \mid MAD(i, j) \text{ is minimum, } i \in (-p, p), j \in (-p, p)]$$

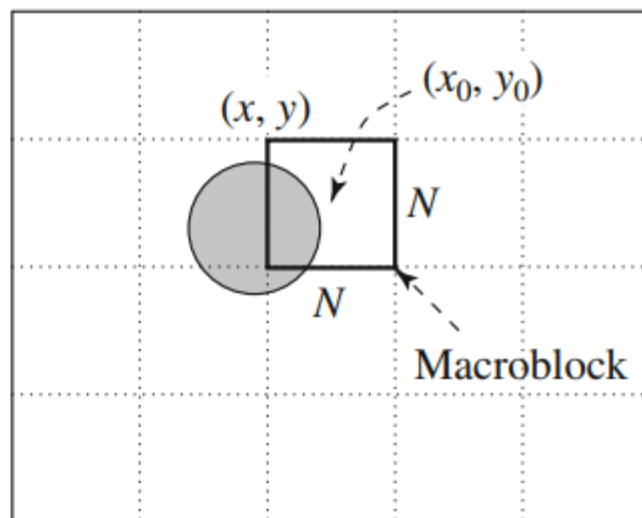
For convenience, we use the upper left corner (x, y) as the origin of the macroblock in the Target frame. Let $C(x + k, y + l)$ be pixels in the macroblock in the Target (current) frame and $R(x + i + k, y + j + l)$ be pixels in the macroblock in the

Reference frame, where k and l are indices for pixels in the macroblock and i and j are the horizontal and vertical displacements, respectively. The difference between the two macroblocks can then be measured by their *Mean Absolute Difference (MAD)*, defined as

$$MAD(i, j) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} |C(x + k, y + l) - R(x + i + k, y + j + l)|$$



参考帧



目标帧

图9 宏块和运动向量

3.3 预测图像P的压缩编码算法

(3) 搜索算法

- ◆ 顺序搜索 (full search)

- 将参考帧中整个 $(2p+1) * (2p+1)$ 的窗口中的每一个宏块逐个像素与目标帧 中的宏块进行比较。
- 获取一个宏块的运动向量的复杂度为 $(2p+1) * (2p+1) * N^2 * 3$, 计算复杂度为 $O(p^2 N^2)$

Procedure 10.1 (Motion vector: Sequential search)

BEGIN

$min_MAD = LARGE_NUMBER;$ /* Initialization */

 for $i = -p$ to p

 for $j = -p$ to p

 {

$cur_MAD = MAD(i, j);$

 if $cur_MAD < min_MAD$

 {

$min_MAD = cur_MAD;$

$u = i;$ /* Get the coordinates for **MV**. */

$v = j;$

 }

 }

END

3.3 预测图像P的压缩编码算法

◆ 2D对数搜索 (logarithmic search)

基本思想是在不同大小的搜索范围内进行搜索，每次迭代搜索范围都在缩小，直到找到最佳的运动向量。搜索次数减少，容易漏掉小的移动，计算复杂度为 $O(\log p \cdot N^2)$

Procedure 10.2 (Motion vector: 2D-Logarithmic search)

BEGIN

offset = $\lceil \frac{p}{2} \rceil$;

Specify nine macroblocks within the search window in the Reference frame,
they are centered at (x_0, y_0) and separated by offset horizontally and/or vertically;

WHILE last \neq TRUE

{

Find one of the nine specified macroblocks that yields the minimum MAD ;

if offset = 1 then last = TRUE;

offset = $\lceil \text{offset}/2 \rceil$;

Form a search region with the new offset and new center found;

}

END

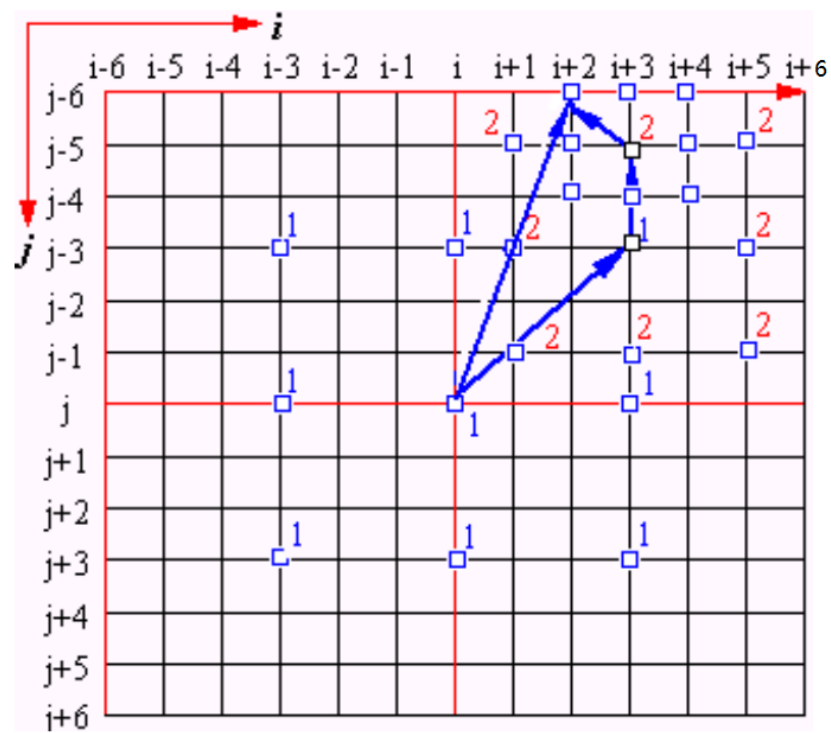
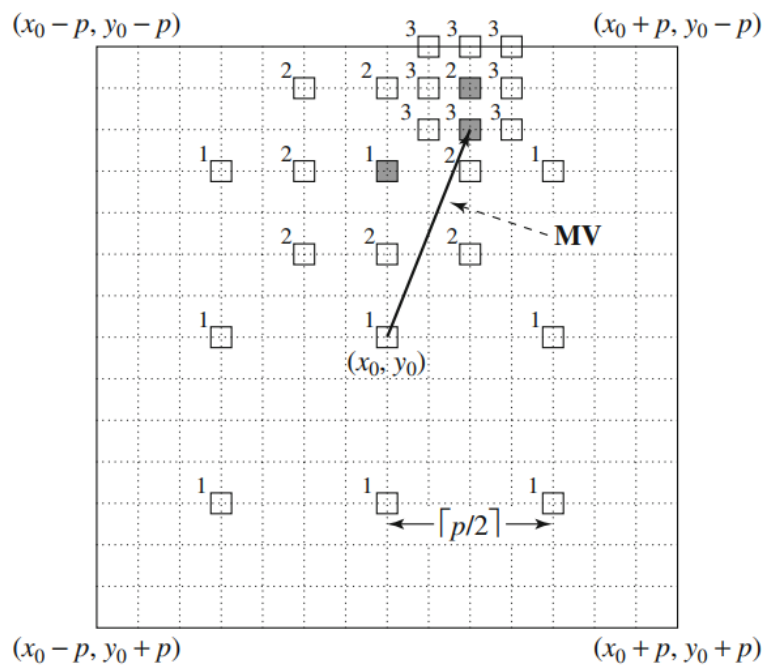


图10 2D对数搜索法

3.3 预测图像P的压缩编码算法

◆ 分层搜索

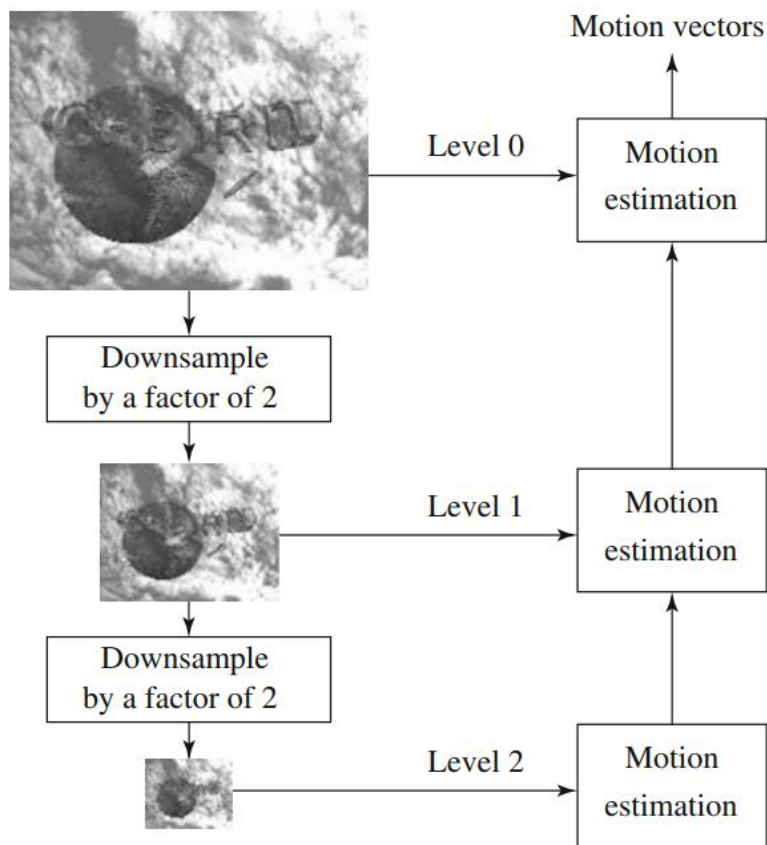
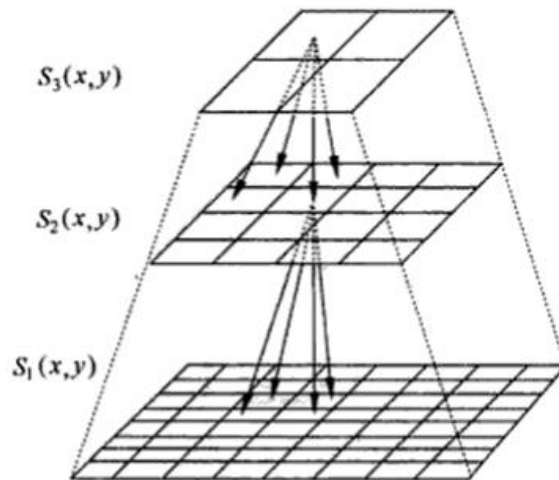


图11 分层搜索法



Procedure 10.3 (Motion vector: Hierarchical search)

BEGIN

// Get macroblock center position at the lowest resolution level k , e.g., level 2.

$x_0^k = x_0^0 / 2^k$; $y_0^k = y_0^0 / 2^k$;

Use Sequential (or 2D Logarithmic) search method to get initial estimated $\mathbf{MV}(u^k, v^k)$ at level k ;

WHILE last \neq TRUE

{

Find one of the nine macroblocks that yields minimum MAD

at level $k - 1$ centered at

$(2(x_0^k + u^k) - 1 \leq x \leq 2(x_0^k + u^k) + 1, 2(y_0^k + v^k) - 1 \leq y \leq 2(y_0^k + v^k) + 1)$;

if $k = 1$ then last = TRUE;

$k = k - 1$;

Assign (x_0^k, y_0^k) and (u^k, v^k) with the new center location and motion vector;

}

END

3.4 双向预测图像B的压缩编码算法

- ◆ 对在它前后帧的像素值之差进行编码，见图12。具体计算方法与预测图像P的算法类似。
- ◆ 双向预测图像B不传播编码误差。

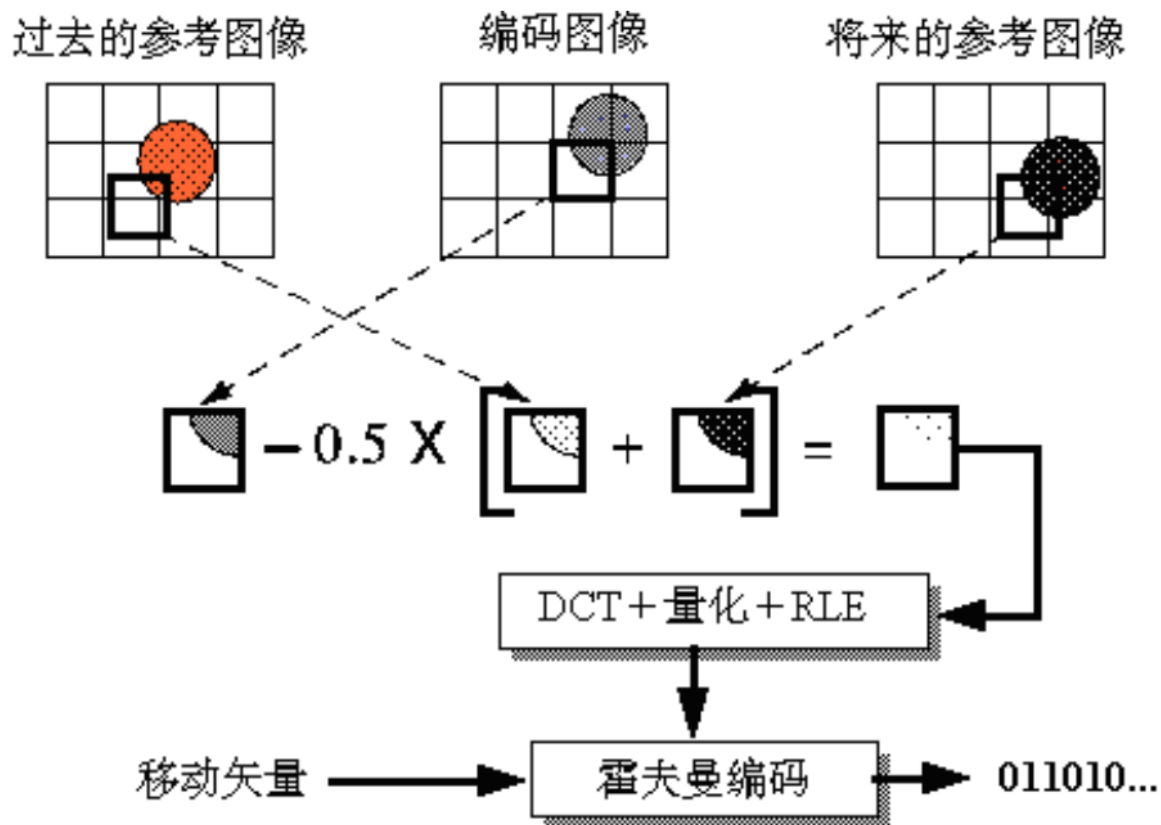


图12 双向预测图像B的压缩编码算法框图

3.5 运动向量搜索方法计算代价对比

◆ 运动向量搜索算法效率对比

Table 1 Comparison of Computational Cost of Motion Vector Search based on examples

运动向量搜索方法计算代价对比

Search Method	<i>OPS_per_second</i> for 728×480 at 30 fps	
	$p = 15$	$p = 7$
Sequential search	29.89×10^9	7.00×10^9
2D Logarithmic search	1.25×10^9	0.78×10^9
3-level Hierarchical search	0.51×10^9	0.40×10^9

3.6 视像数据流的结构

- ◆ 数据位流的组织关系到如何设计解码器，如无统一规范，设计的解码器就不能通用。
- ◆ 按层次结构组织，一个视像序列(video sequence)分成6层，见图13
 - (1) 序列层(sequence)
 - (2) 像组层(group of pictures, GOP)
 - (3) 图像层(picture)
 - (4) 像片层(slice)
 - (5) 宏块层(macroblock, MB)
 - (6) 图块(block)层

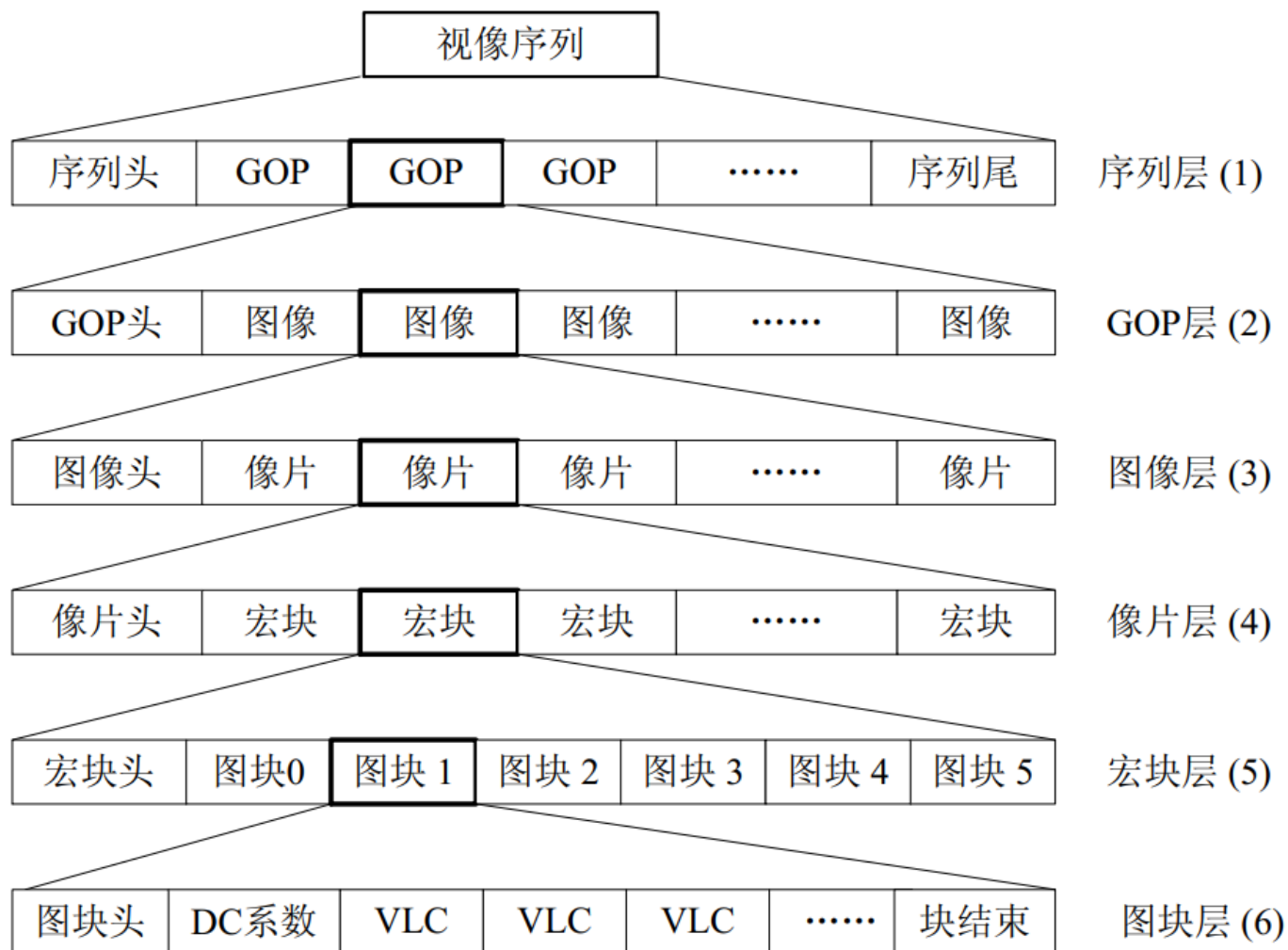


图13 MPEG-1视像序列的结构