追随技术WithPassion

第四章 视频编码基础

1. 压缩码流

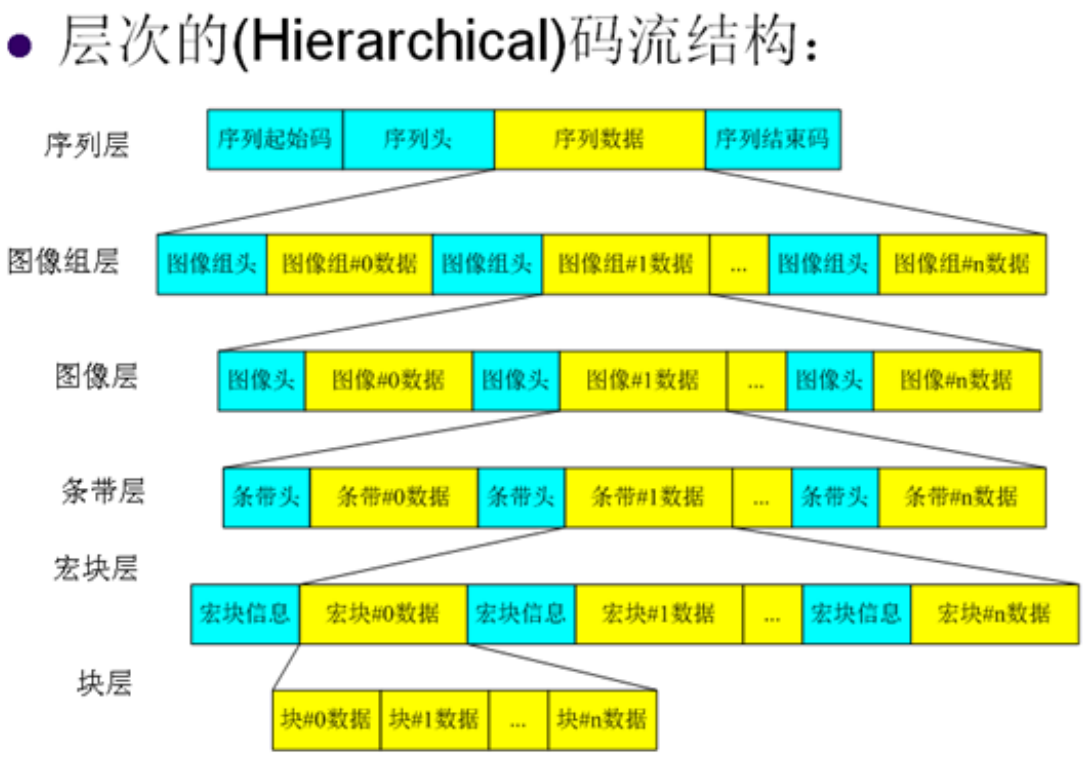
- 语法：码流中各个元素的位置关系
  - 01001001…
  - 图像编码类型(01)，宏块类型(00)，编码系数1001等
- 语义：每个语法元素所表达的意义。
  - 例如：图像编码类型

2. 编码层次

I帧	00
P帧	01
B帧	10

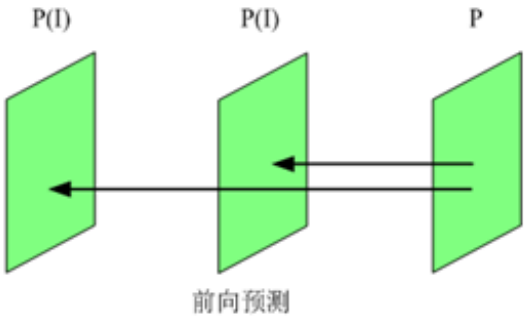
- 序列（Sequence）
- 图像组（Group of Pictures，GOP）
- 图像（Picture）
- 条带（Slice）
- 宏块（Macroblock，MB）
- 块(Block)

3. 码流结构

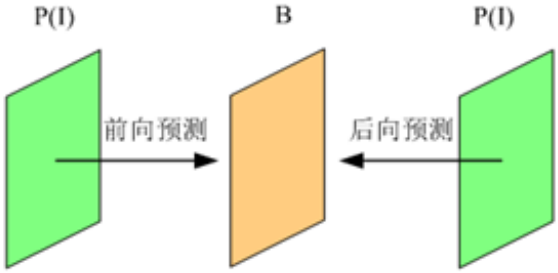


3. PB帧编码

- 当前帧：将要编码的图像。
- 参考帧：预测块所在的图像。
- 前向预测帧（P帧）



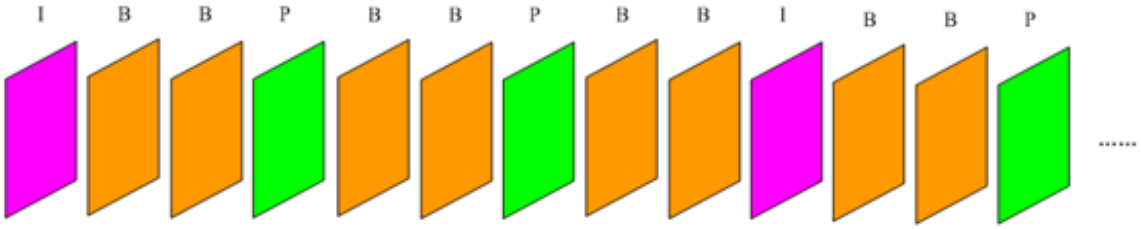
- 双向预测帧（B帧）



4. 序列编码对象

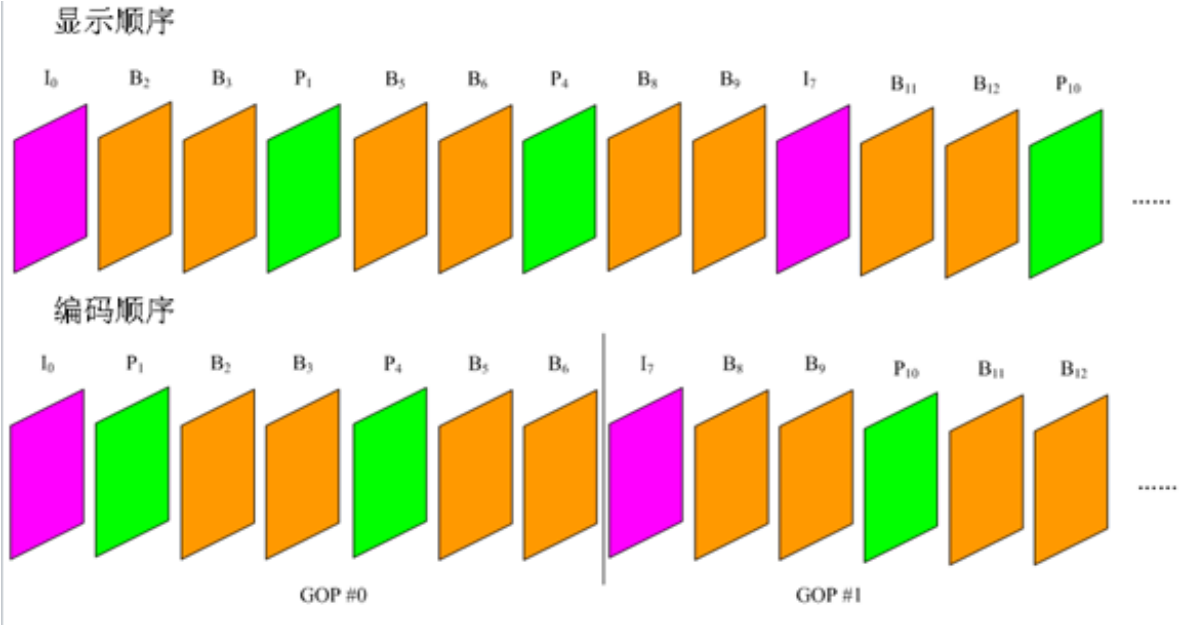
（1）IBBP序列

- IBBP序列编码结构



- 序列：一段连续编码的并具有相同参数的视频图像。
- 序列起始码：专有的一段比特串，标识一个序列的压缩数据的开始
  - MPEG-2的序列起始码为十六进制数000001(B3)。
- 序列头：记录序列信息
  - 档次（Profile），级别（Level），宽度，高度，是否是逐行序列，帧率等。
- 序列结束码：专有的一段比特串，标识该序列的压缩数据的结束
  - MPEG-2的序列结束码为十六进制数000001(B7)。

5. 图像组编码对象



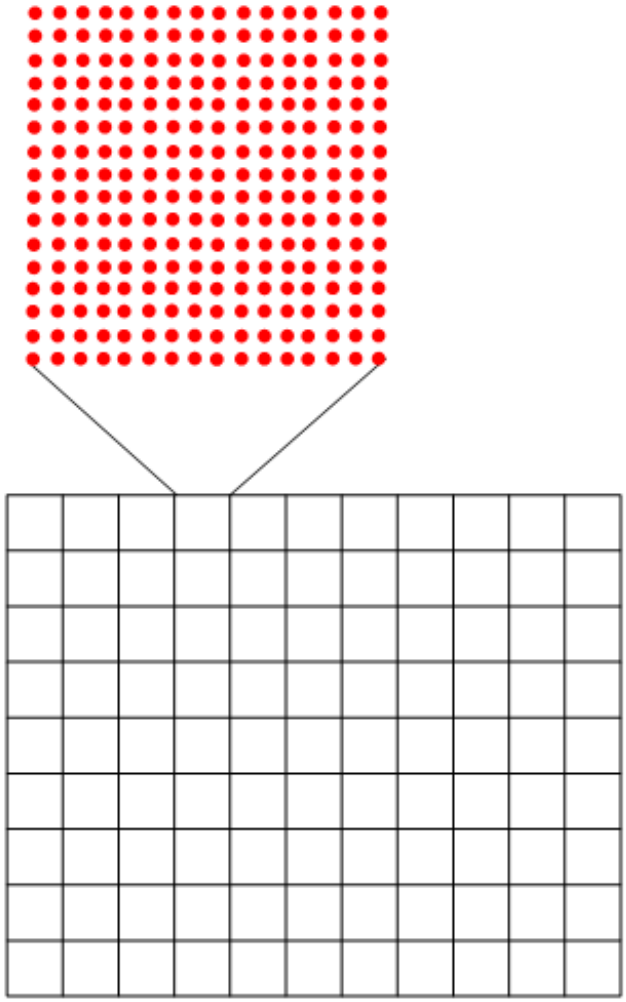
6. 图像编码结构

- 图像：
- 图像起始码：专有的一段比特串，标识一个图像的压缩数据的开始
  - MPEG-2的图像起始码为十六进制数000001(00)。
- 图像头：记录图像信息
  - 图像编码类型，图像距离，图像编码结构，图像是否为逐行扫描。

7. 图像分块编码

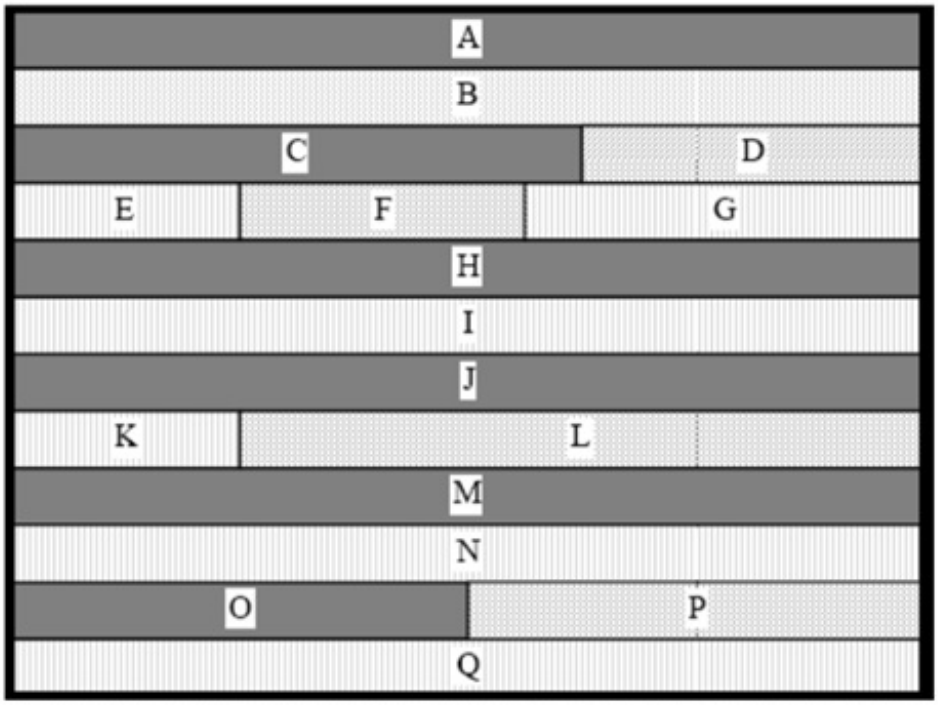
8. 条带编码结构

- 条带：多个宏块的组合。
- 条带起始码：专有的一段比特串，标识一个条带的压缩数据的开始
    - MPEG-2的条带起始码为十六进制数000001(0~AF)。
- 条带头：记录当前图像的相关信息
    - 条带位置，条带量化参数，宏块编码技术标识等。



9. 条带编码对象

- 图像的条带划分

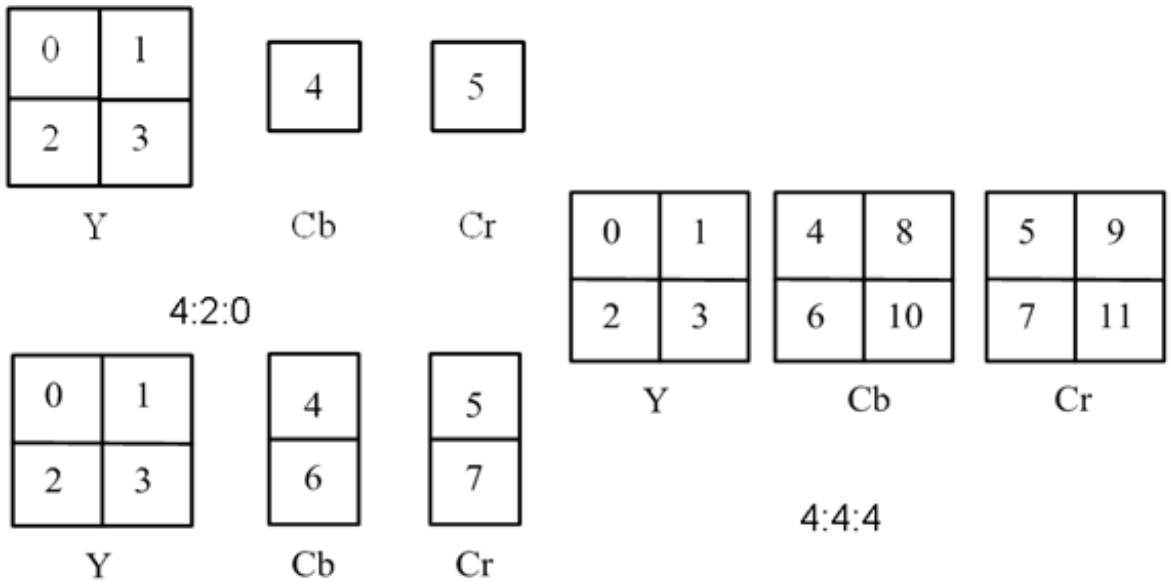


10. 宏块编码结构

- 宏块：16x16的像素块（对亮度而言）。

- 宏块内容：宏块编码类型，编码模式，参考帧索引，运动矢量信息，宏块编码系数等。

11. 宏块编码对象

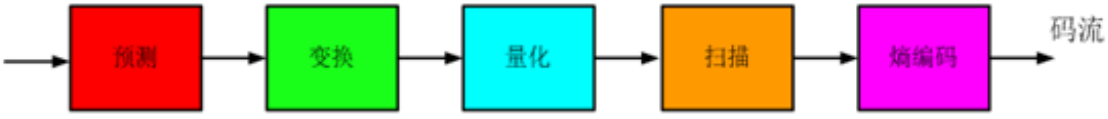


12. 块编码结构

- 8x8或4x4块的变换量化系数的熵编码数据。
- CBP (Coded Block Patten)：用来指示块的变换量化系数是否全为零。
    - 对于YUV(4:2:0)编码，CBP通常6比特长，每个比特对应一个块，当某一块的变换量化系数全为零时，其对应比特位值为0，否则为1。
- 每个块的变换量化系数的最后用一个EOB (End of Block)符号来标识。

13. 视频编解码关键技术

- 预测：通过帧内预测和帧间预测降低视频图像的空间冗余和时间冗余。
- 变换：通过从时域到频域的变换，去除相邻数据之间的相关性，即去除空间冗余。
- 量化：通过用更粗糙的数据表示精细的数据来降低编码的数据量，或者通过去除人眼不敏感的信息来降低编码数据量。
- 扫描：将二维变换量化数据重新组织成一维的数据序列。
- 熵编码：根据待编码数据的概率特性减少编码冗余。



14. 预测

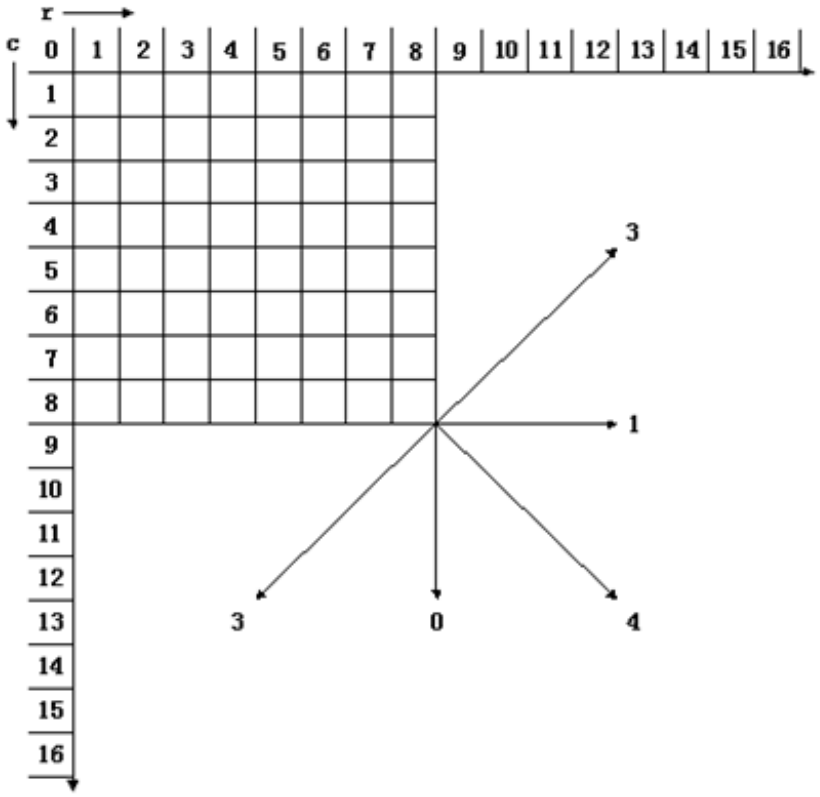
- 空间预测：利用图像空间相邻像素的相关性来预测的方法。
    - 帧内预测技术：利用当前编码块周围已经重构出来的像素预测当前块
    - Intra图像编码（I帧）

- 时间预测：利用时间上相邻图像的相关性来预测的方法。
    - 帧间预测：运动估计（Motion Estimation，ME），运动补偿（Motion Compensation，MC）
    - Inter图像编码：前向预测编码图像（P帧），双向预测编码图像（B帧）

15. 帧内预测

- I帧图像的每个宏块都采用帧内（Intra）预测编码模式。
- 宏块分成8x8或者4x4块，对每个块采用帧内预测编码，称作Intra8x8或者Intra4x4。
- 帧内预测有多个预测方向：水平，垂直，左下，右上。
- 帧内预测还有直流（DC）预测。
- 色度块预测还有平面预测。

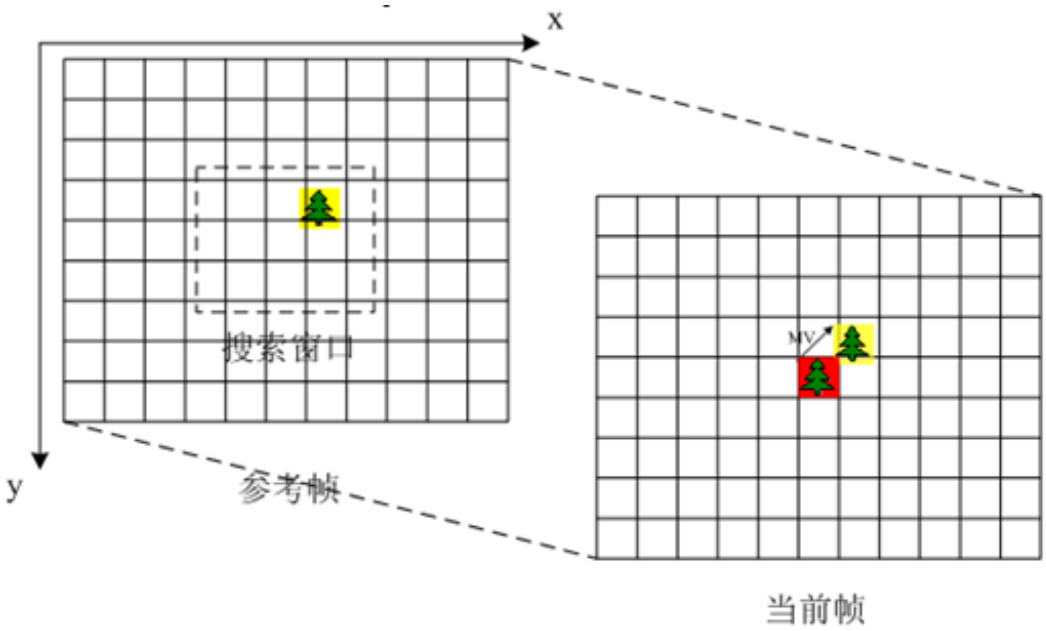
- 帧内预测的四个预测方向：



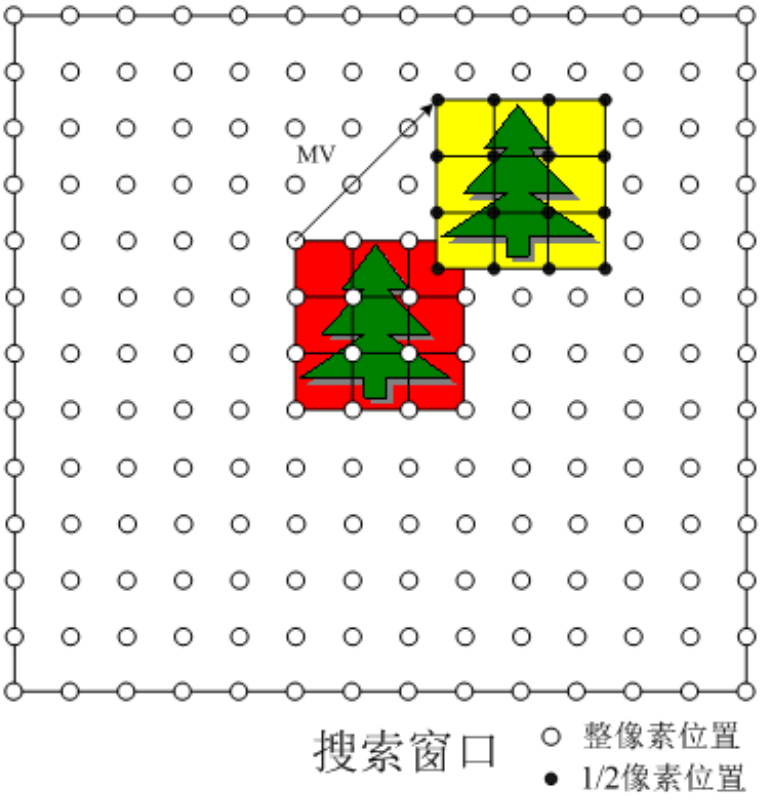
16. 帧间预测

- 块基运动估计：为待预测块在参考帧上找到最佳的预测块，并记录预测块在参考帧上的相对位置。

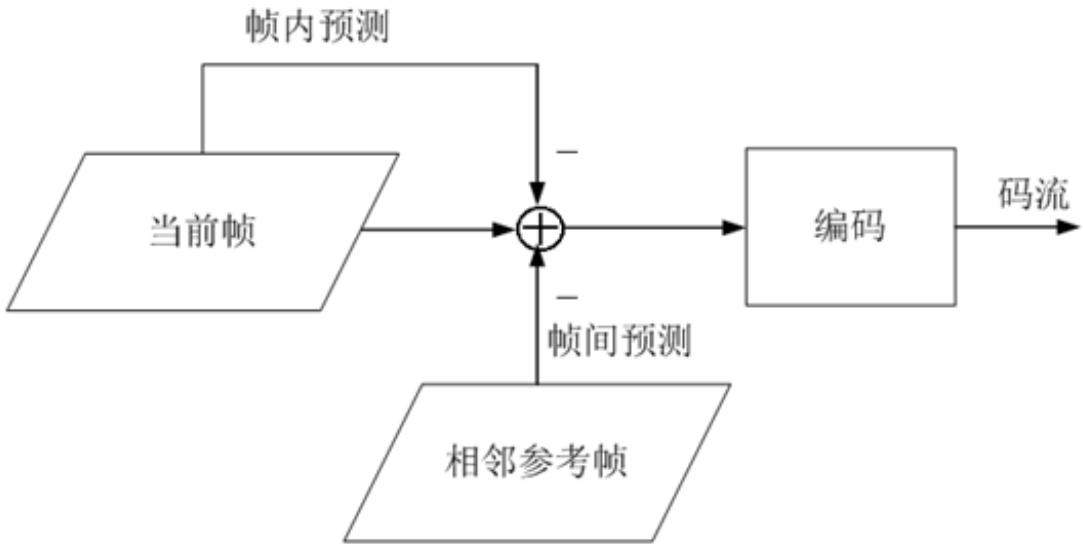
- 运动矢量（MV）：参考帧上的预测块与当前帧上的的待预测块的相对位置。
  - MV有两个分量：（x，y）



- 分像素运动估计
  - 最佳的预测块不在整像素位置，而在分像素位置；
  - 1/2，1/4，1/8像素插值得到分像素值。



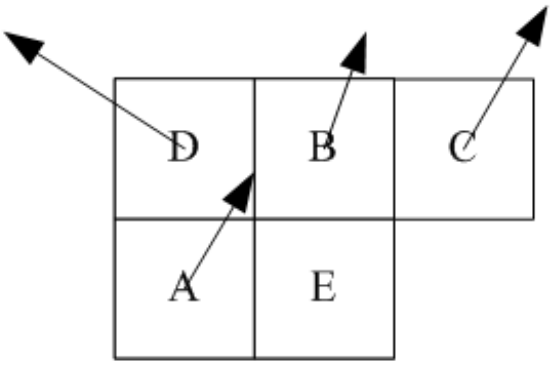
- 帧间预测流程：



- 运动补偿：给定MV和参考帧，为待解码块从参考帧上获取预测块。
- 运动矢量编码
  - MV预测：用当前块的周围可得到邻块的运动矢量来预测当前块的运动矢量
  - 运动矢量差（MV difference，MVD）：实际运动矢量与预测运动矢量的差，即：
  - 运动矢量差采用变长编码。

$$MVD(x) = MV(x) - MVP(x)$$
$$MVD(y) = MV(y) - MVP(y)$$

17. 预测残差



- 预测残差：待预测的原始图像块减去预测的图像块所得的结果

$$e(x,y)=P_i(x,y)-\hat{P}_i(x,y)=P_i(x,y)-P_{i-1}(x+\triangle x,y+\triangle y)$$

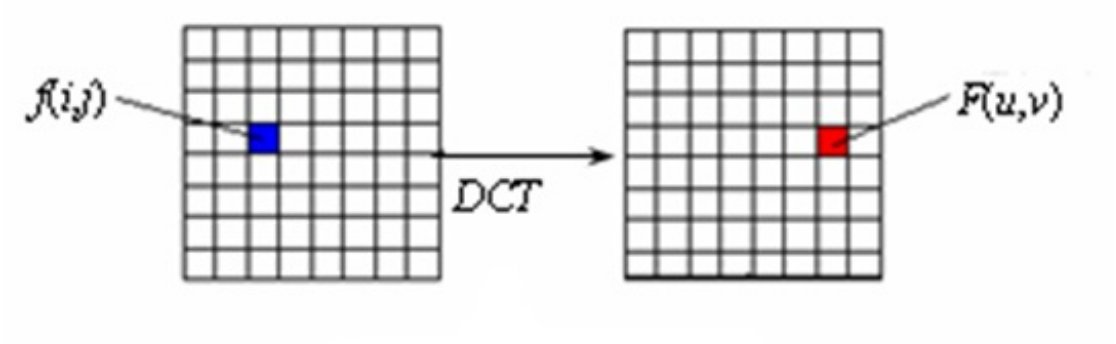
其中 $MV=(\triangle x,\triangle y)$

88	84	83	84	85	86	83	82		84	82	83	81	85	86	83	81	-4	-2	0	-3	0	0	0	-1
86	82	82	83	82	83	83	81		82	82	81	83	82	83	83	81	-4	0	-1	0	0	0	0	0
82	82	84	87	87	87	81	84		83	82	84	87	87	87	81	88	1	0	0	0	0	0	0	4
81	86	87	89	82	82	84	87	-	81	85	86	88	82	82	84	87	= 0	-1	-1	-1	0	0	0	0
81	84	83	87	85	89	80	81		81	84	85	87	85	89	84	81	0	0	2	0	0	0	4	0
81	85	85	86	81	89	81	85		82	85	81	84	81	89	81	83	1	0	-4	-2	0	0	0	-2
82	81	86	83	86	89	81	84		81	87	86	83	86	89	81	84	-1	6	0	0	0	0	0	0
88	88	90	84	85	88	88	81		88	82	87	84	87	89	84	81	0	-6	-3	0	2	1	-4	0

P_i $\hat{P}_i$ e

18. 变换编码

- 变换编码：通过变换将空域信号转换为频域信号来去除空间信号的冗余信息，减少编码数据。
- 二维离散余弦变换
  - 4x4变换，8x8变换



- 二维离散余弦变换

例：

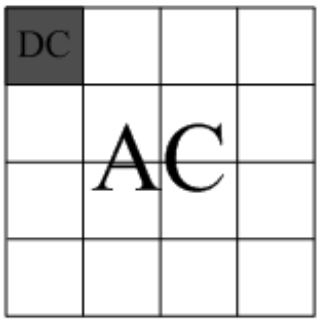
139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	162	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
162	162	161	163	162	157	157	157
162	162	161	161	163	158	158	158

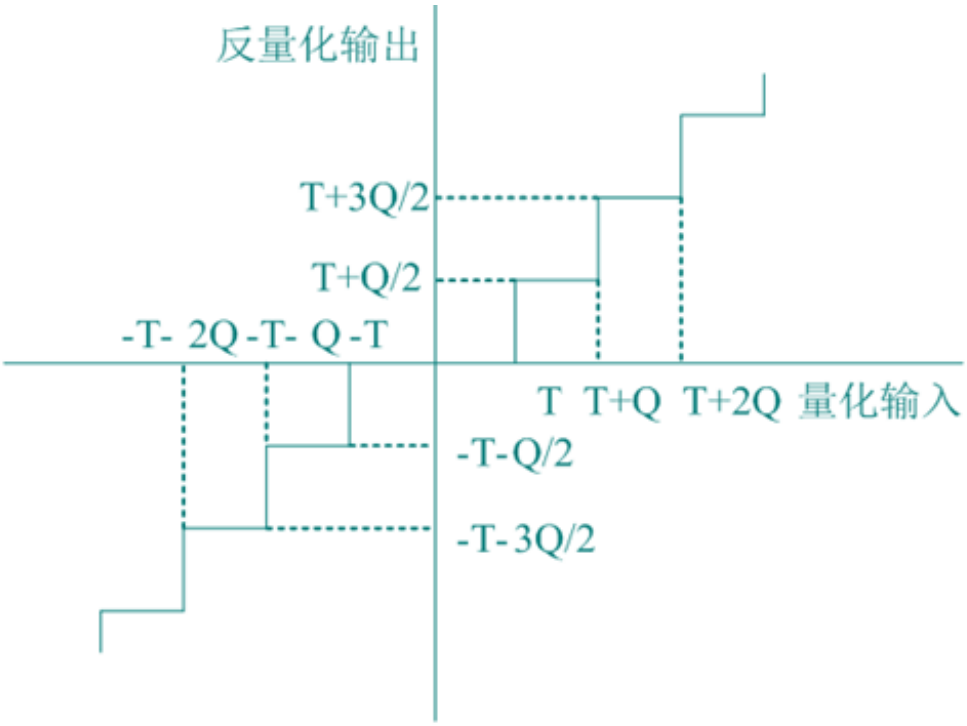
235.6	−1.0	−12.1	−5.20	2.1	−1.7	−2.7	1.3
−22.6	−18.5	−6.2	−3.2	−2.9	−0.1	0.4	−1.2
−10.9	−9.3	−1.6	1.5	0.2	−0.9	−0.6	−0.1
−7.1	−1.9	0.2	1.5	0.9	−0.1	0.0	0.3
−0.6	−0.8	1.5	1.6	−0.1	−0.7	0.6	1.3
1.8	−0.2	−1.6	−0.3	−0.8	1.5	1.0	−1.0
−1.3	−0.4	−0.3	−1.5	−0.5	1.7	1.1	−0.8
−2.6	1.6	−3.8	−1.8	1.9	1.2	−0.6	−0.4

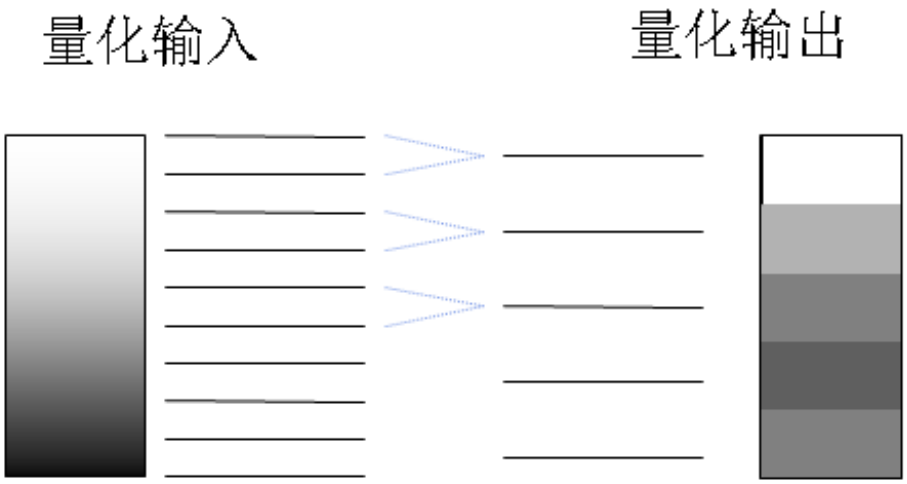
- 变换系数：直流（DC）系数，交流（AC）系数

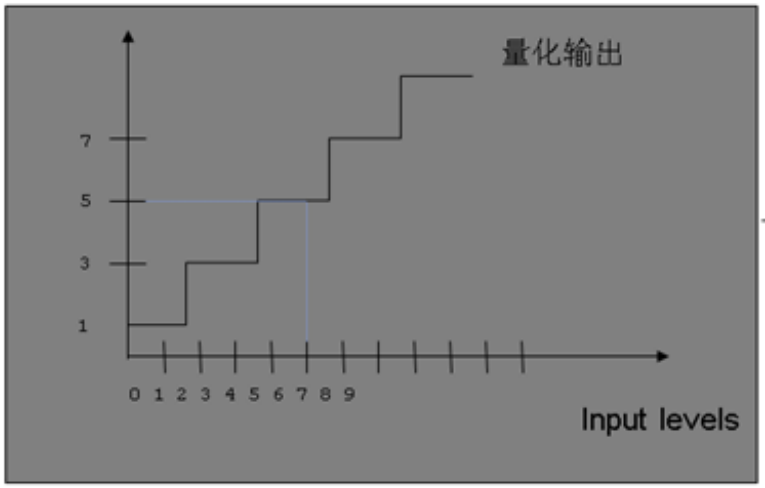
19. 量化

- 量化原理：将含有大量的数据集合映射到含有少量的数据集合中。

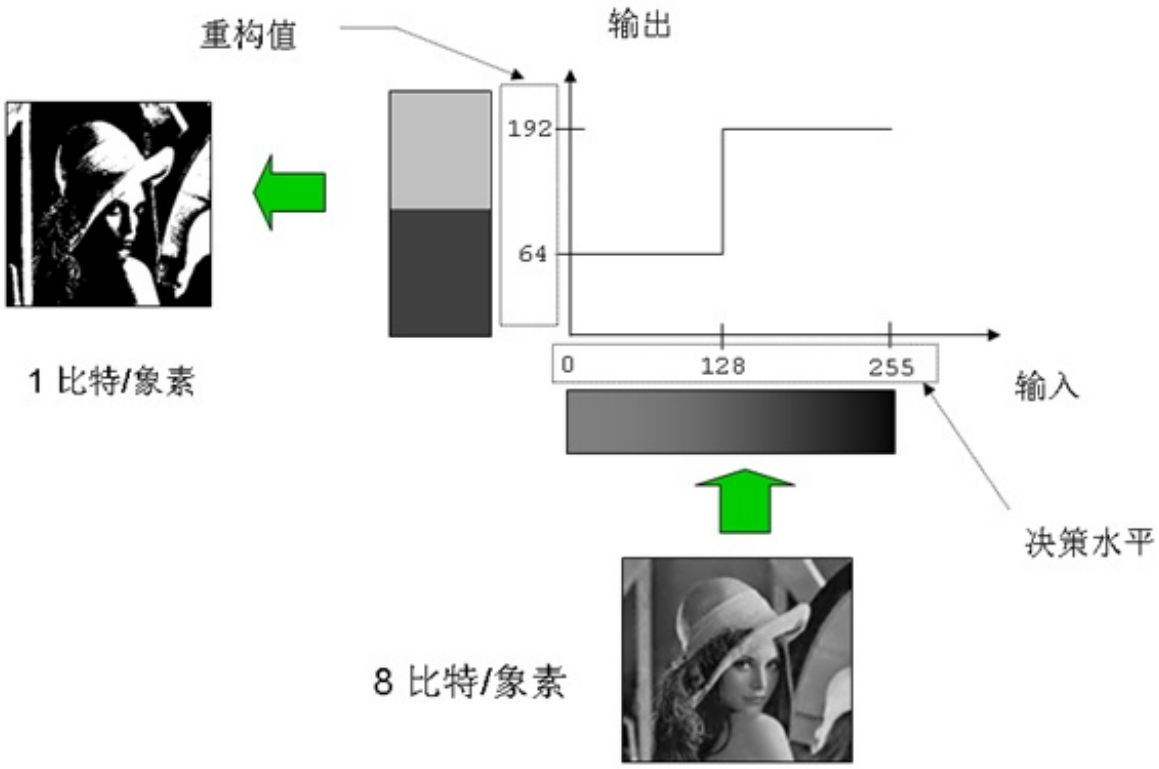


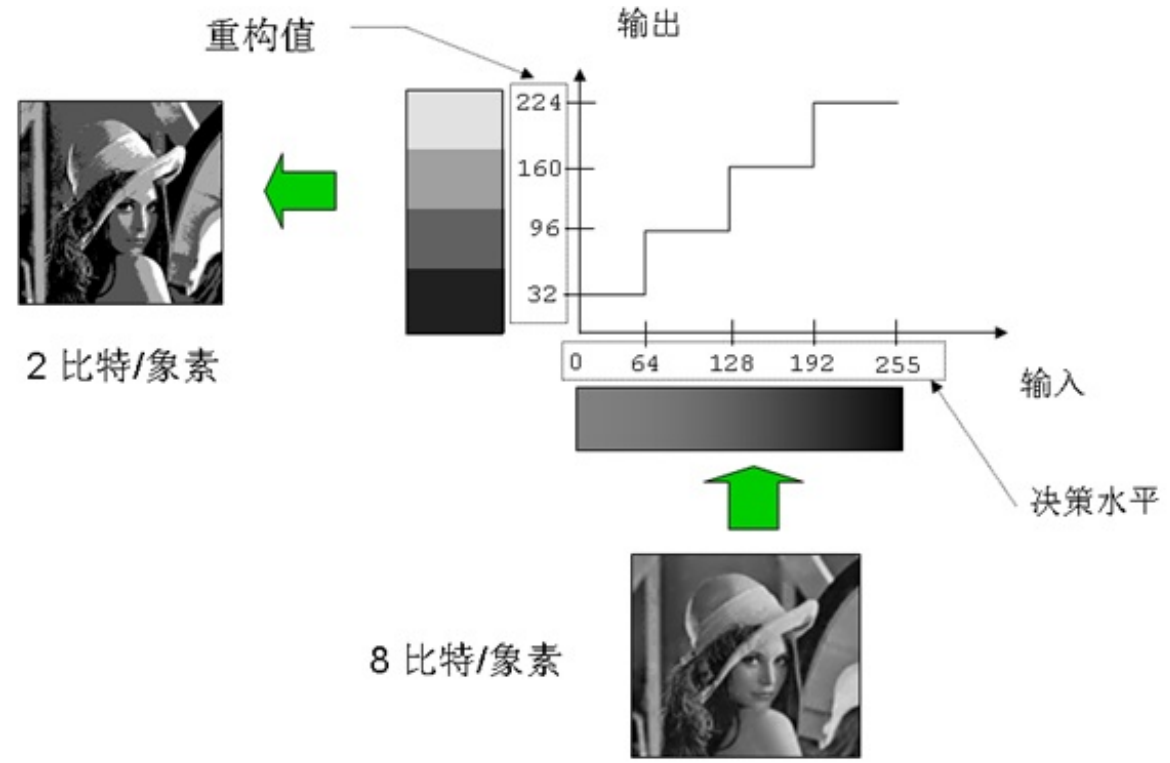





量化输出
7
5
3
1
0 1 2 3 4 5 6 7 8 9
Input levels


熵编码


重构值
输出
192
64
0
128
255
输入
决策水平
1 比特/象素
8 比特/象素



- 一般情况下量化后高频部分包含大量的零系数

量化表

235.6	−1.0	−12.1	−5.20	2.1	−1.7	−2.7	1.3
−22.6	−18.5	−6.2	−3.2	−2.9	−0.1	0.4	−1.2
−10.9	−9.3	−1.6	1.5	0.2	−0.9	−0.6	−0.1
−7.1	−1.9	0.2	1.5	0.9	−0.1	0.0	0.3
−0.6	−0.8	1.5	1.6	−0.1	−0.7	0.6	1.3
1.8	−0.2	−1.6	−0.3	−0.8	1.5	1.0	−1.0
−1.3	−0.4	−0.3	−1.5	−0.5	1.7	1.1	−0.8
−2.6	1.6	−3.8	−1.8	1.9	1.2	−0.6	−0.4

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	113	100	103	99

15	0	−1	0	0	0	0	0
−2	−1	0	0	0	0	0	0
−1	−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	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

- 量化对主观质量的影响

原图像

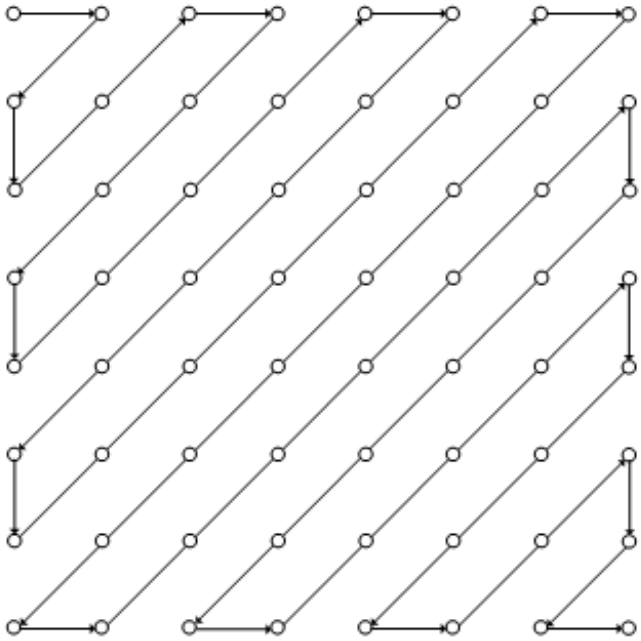
$P_{x,y}$

量化后图像

$((P_{x,y}+8)/16)*16$

20. 扫描

- 扫描：将二维数据转换为一维的数据序列。

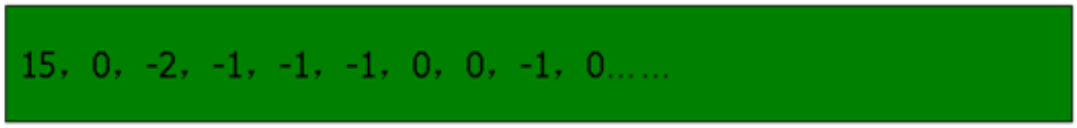


8x8 Zigzag扫描示意图

例：

15	0	−1	0	0	0	0	0
−2	−1	0	0	0	0	0	0
−1	−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	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

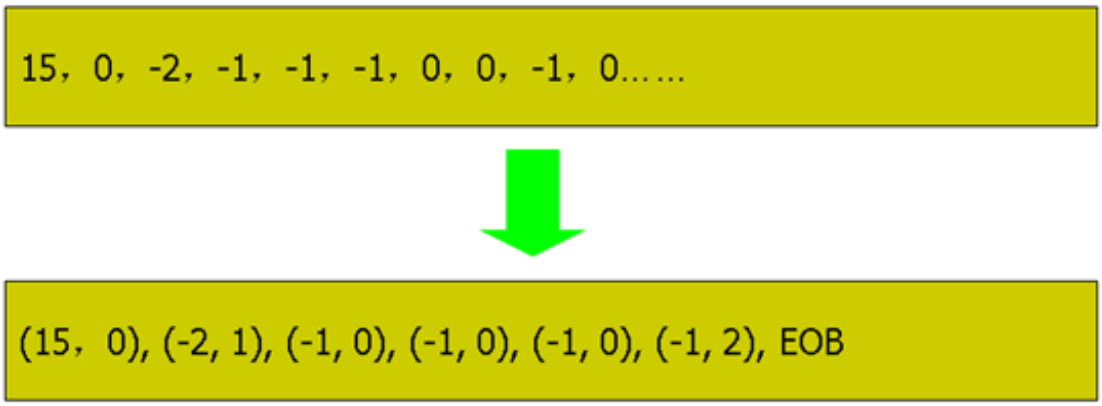
8x8变换量化系数



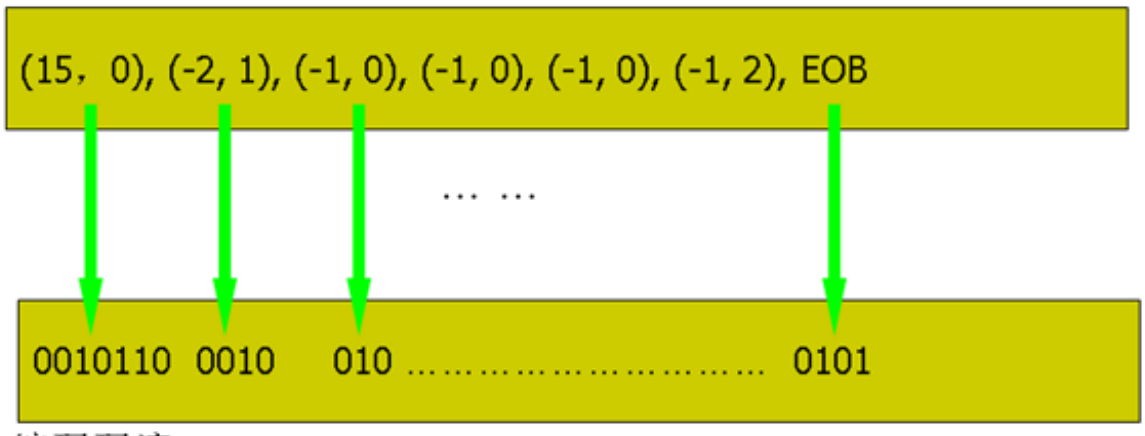
扫描后的变换量化系数

21. 熵编码

- 熵编码：根据符号出现的概率，对经常出现的符号分配较短的码字，对不常出现的符号分配较长的码字。
- Level-Run编码：用数据中非零值和其前面非零值之间出现零值的个数重新描述量化系数序列为（Level，Run）二元组序列



- 变长编码
  - 将Level-Run编码后的（level，run）变长编码成最终的比特串。



编码码流

22. 码率控制

- 受到缓冲区，带宽的限制，编码码率不能无限制的增长，因此需要通过码率控制来将编码码流控制在目标码率范围内。
- 一般通过调整量化参数的手段控制码率
    - 帧级控制
    - 条带级控制
    - 宏块级控制
- 码率控制考虑的问题
    - 防止码流有较大的波动，导致缓冲区发生溢出，
    - 同时保持缓冲区尽可能的充满，让图像质量尽可能的好而且稳定
- CBR（Constant Bit Rate）
    - 比特率稳定，但图像质量变化大
- VBR（Variable Bit Rate）
    - 比特率波动大，但图像质量稳定
- 码率控制算法
    - 码率分配
    - 码率控制
- 码率控制属于非标准技术
    - 编码端有，解码端没有

第5章 预测

1. 预测技术

- 目的：去除空间冗余和时间冗余。
- 视频存在大量的空间冗余和时间冗余
    - 空间冗余：用帧内预测编码去除
        - 基于块的帧内预测
    - 时间冗余：用帧间预测编码去除
        - 基于块匹配（Block Matching）的帧间预测
- 预测后得到去除大部分空间或时间冗余的残差

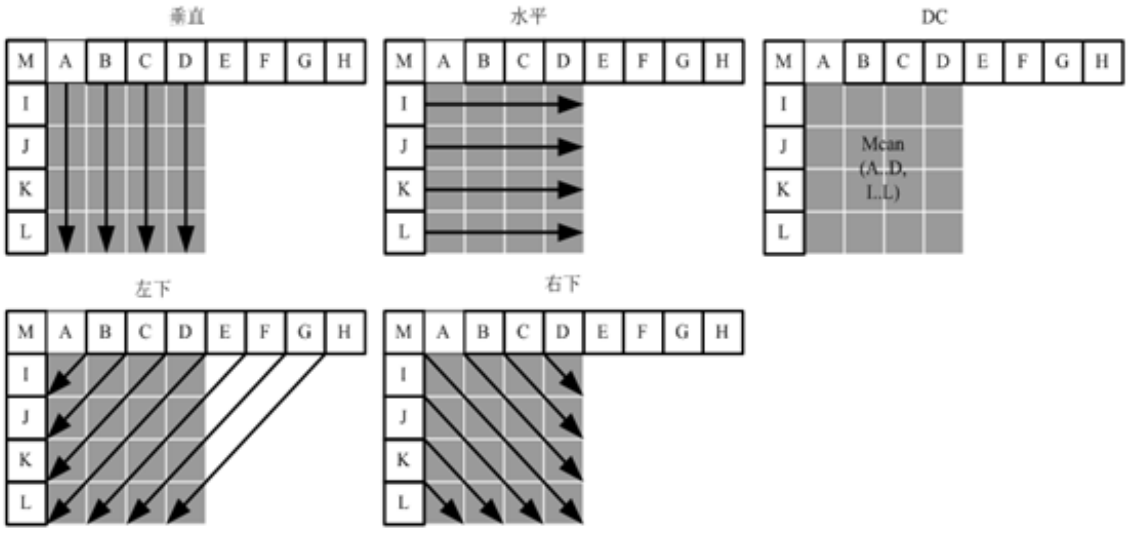
2. 空间冗余

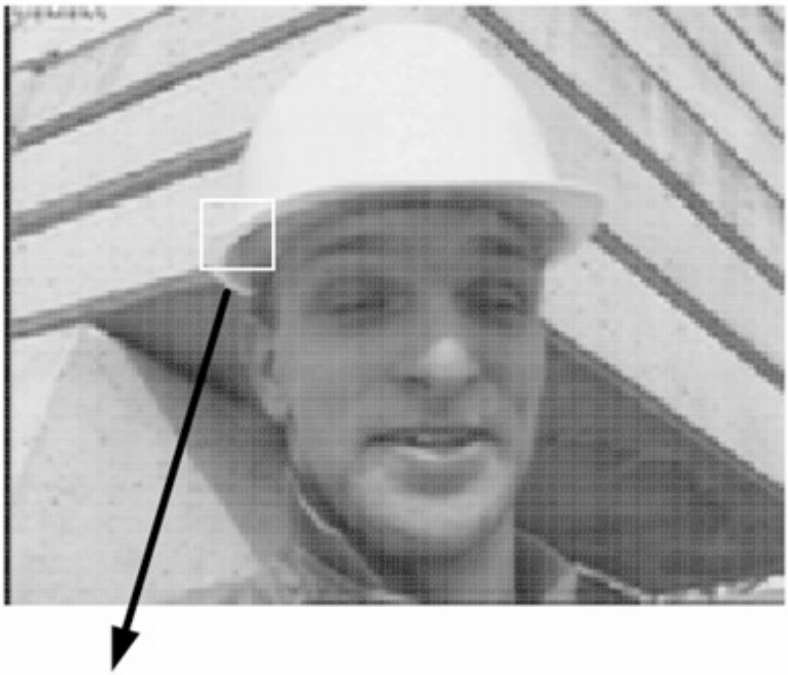
- 图像空间相邻像素具有很强的相关性。
- 帧内预测技术去除空间冗余

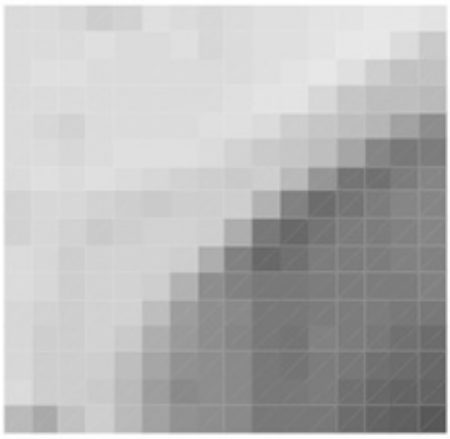
原始图像

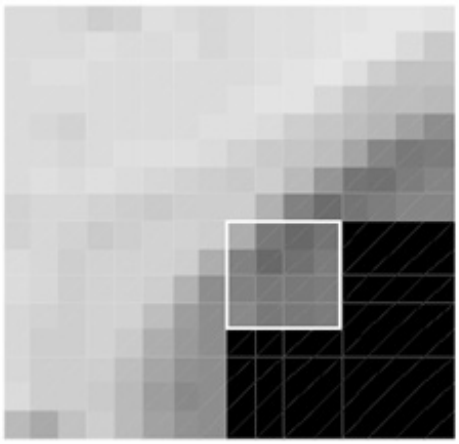
帧内预测图像

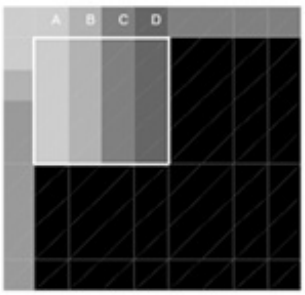
3. 亮度预测模式



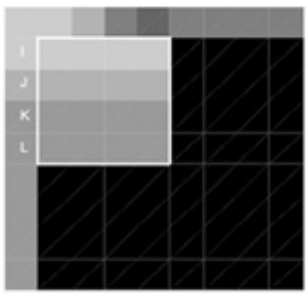




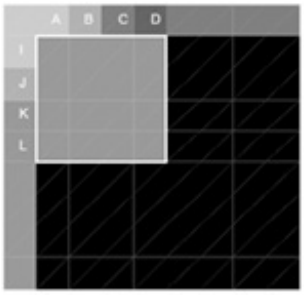




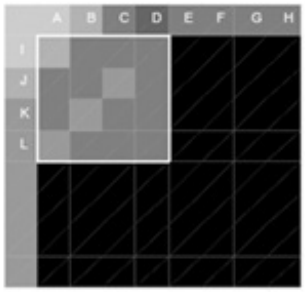
垂直



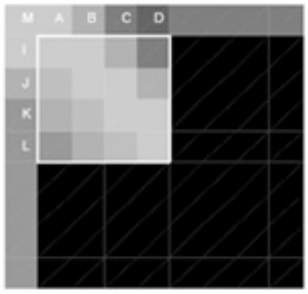
水平



DC



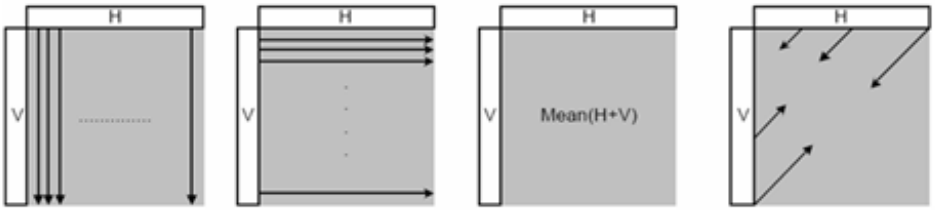
左下



右下

4. 色度预测模式

- 常用的色度预测模式：



垂直　水平　DC　平面

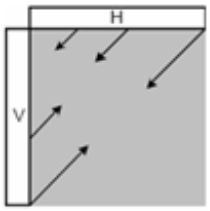
平面模式的计算：

$$pred[x,y] = Clip((a + b\times(x-7) + c\times(y-7) + 16) >> 5)$$

$$a = 16\times(p[-1,15] + p[15,-1])$$

$$b = (5\times H + 32) >> 6$$

$$c = (5\times V + 32) >> 6$$

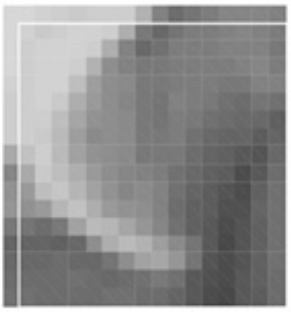


平面

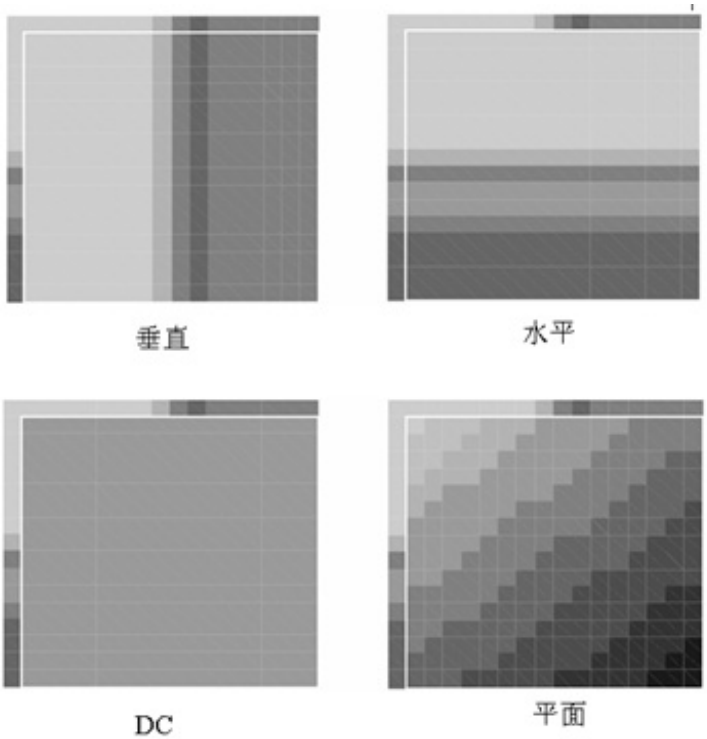
$$H = \sum_{x'=0}^{7}(x'+1)\times(p[8+x',-1] - p[6-x',-1])$$

$$V = \sum_{y'=0}^{7}(y'+1)\times(p[-1,8+y'] - p[-1,6-y'])$$

$$Clip(p)\begin{cases} p, & 0 \le p \le 255 \\ 0, & p < 0 \\ 255, & p > 255 \end{cases}$$



最优的预测模式为平面模式



垂直　水平　DC　平面

5. 时间冗余

- 视频图像在时间上有较强的相关性，即存在时间冗余

- 去除时间冗余的编码技术
  - 运动估计（Motion Estimation，ME）
    - 为待编码块搜索最相似的预测块
    - 记录运动矢量（Motion Vector，MV)
    - 记录预测残差：$E = F_i - \overline{F}_{i-1}$
  - 运动补偿（Motion Compensation，MC）
    - 根据运动矢量获取预测块
    - 根据预测残差计算重构块：$F_i = E + \overline{F}_{i-1}$


第66帧


第69帧


两帧之差


运动补偿后两帧之差

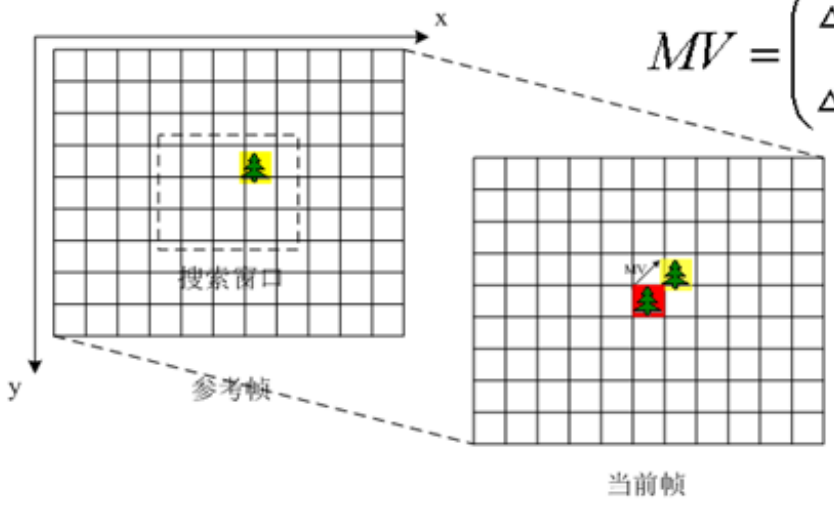
6. 运动模型

- 平移：$\begin{cases}\hat{x}=x+\Delta x\\ \hat{y}=y+\Delta y\end{cases}$ （最常用的一种模型）
- 仿射：$\begin{cases}\hat{x}=a_{00}x+a_{01}y+\Delta x\\ \hat{y}=a_{10}x+a_{11}y+\Delta y\end{cases}$
- 双线性：$\begin{cases}\hat{x}=a_{00}x+a_{01}y+a_{02}xy+\Delta x\\ \hat{y}=a_{10}x+a_{11}y+a_{12}xy+\Delta y\end{cases}$

（1）平移

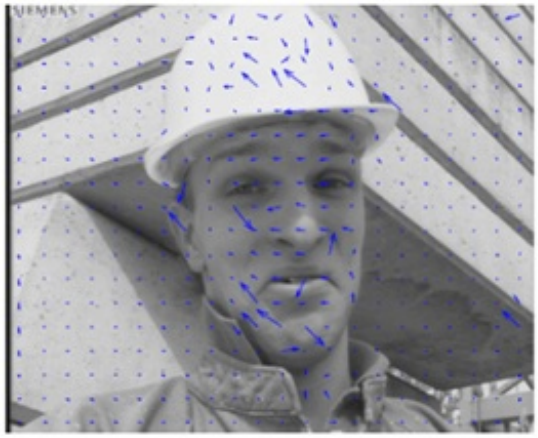
- 平移：$\begin{cases}\hat{x}=x+\Delta x\\ \hat{y}=y+\Delta y\end{cases}$
  - 适用于块匹配的运动估计和运动补偿
  - 运动矢量表达简单

$$MV=\begin{pmatrix}\Delta x\\ \Delta y\end{pmatrix}$$



第66帧

第69帧



运动矢量情况

运动补偿图像

7. 匹配准则

- 匹配准则：衡量待预测块和预测块之间的相似度。
  - 绝对值差和（Sum of Absolute Differences，SAD）：

$$SAD(\triangle x,\triangle y)=\sum_{x=k}^{x=k+n}\sum_{y=j}^{y=j+m}\left|P_i(x,y)-P_{i-1}(x+\triangle x,y+\triangle y)\right|$$

  - 平方差和（Sum of Squared Differences，SSD）：

$$SSD(\triangle x,\triangle y)=\sum_{x=j}^{x=k+n}\sum_{y=j}^{y=j+m}\left(P_i(x,y)-P_{i-1}(x+\triangle x,y+\triangle y)\right)^2$$

- 简化的SAD：

$$SAD(\triangle x,\triangle y)=\sum_{x=k}^{x=k+n}\sum_{y=j}^{y=j+m}\left|P_i(x,y)-P_{i-1}(x+\triangle x,y+\triangle y)\right|$$

$$\geq\left|\sum_{x=k}^{x=k+n}\sum_{y=j}^{y=j+m}P_i(x,y)-P_{i-1}(x+\triangle x,y+\triangle y)\right|=\left|\sum_{x=k}^{x=k+n}\sum_{y=j}^{y=j+m}P_i(x,y)-\sum_{x=k}^{x=k+n}\sum_{y=j}^{y=j+m}P_{i-1}(x+\triangle x,y+\triangle y)\right|$$

- 简化的SSD：

$$SSD(\triangle x,\triangle y)=\sum_{x=k}^{x=k+n}\sum_{y=j}^{y=j+m}(P_i(x,y)-P_{i-1}(x+\triangle x,y+\triangle y))^2$$

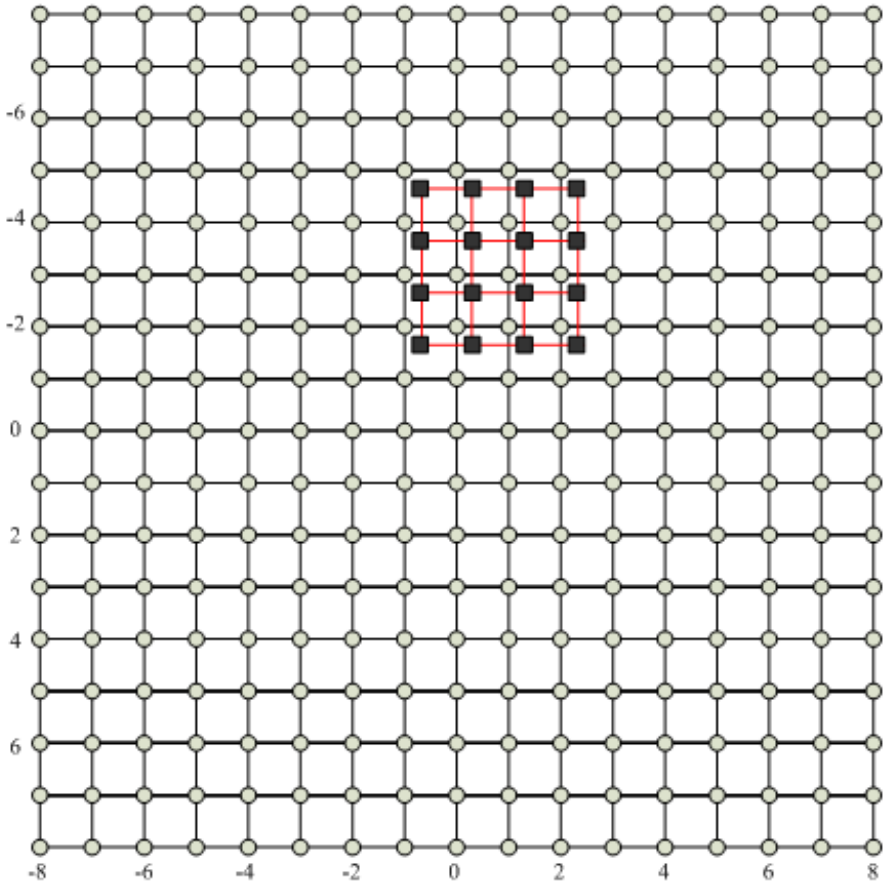
$$\leq\sum_{x=k}^{x=k+n}\sum_{y=j}^{y=j+m}P_i(x,y)^2-P_{i-1}(x+\triangle x,y+\triangle y)^2=\left(\sum_{x=k}^{x=k+n}\sum_{y=j}^{y=j+m}P_i(x,y)^2-\sum_{x=k}^{x=k+n}\sum_{y=j}^{y=j+m}P_{i-1}(x+\triangle x,y+\triangle y)^2\right)$$

8. 匹配准则简化

- 简化技术方法
  - 分别计算当前块和预测块的象素值和
  - 根据简化形式，比较当前块和预测块
  - 如果用简化准则对预测块和当前块比较的结果比以前最好的结果差，可以确定预测效果不好，不必对预测块再进行比较。

9. 运动估计

- 去除视频图像的时间冗余
- 运动估计在搜索范围内为当前块寻找匹配最好的预测块
- 全搜索方式的运动估计计算复杂度高



10. 全搜索复杂度分析

- 图像大小：MxM
- 预测块大小：NxN
- 搜索范围：(-R，R)
- 每个搜索点象素比较个数：N^2
- 搜索点个数$(2R+1)^2$
- 在搜索范围内的象素比较个数总和$N^2(2R+1)^2$
- 一帧图像所有块的全搜索象素比较个数总和$N^2(2R+1)^2(M/N)^2=(2R+1)^2M^2$
- 例：M=512，N=4，R=8，帧率：30帧/秒

  $(2R+1)^2M^2$

  $=17^2X512^2$

  = 75759616次/帧

  = 75759616x30次/秒

  =2272788480次/秒

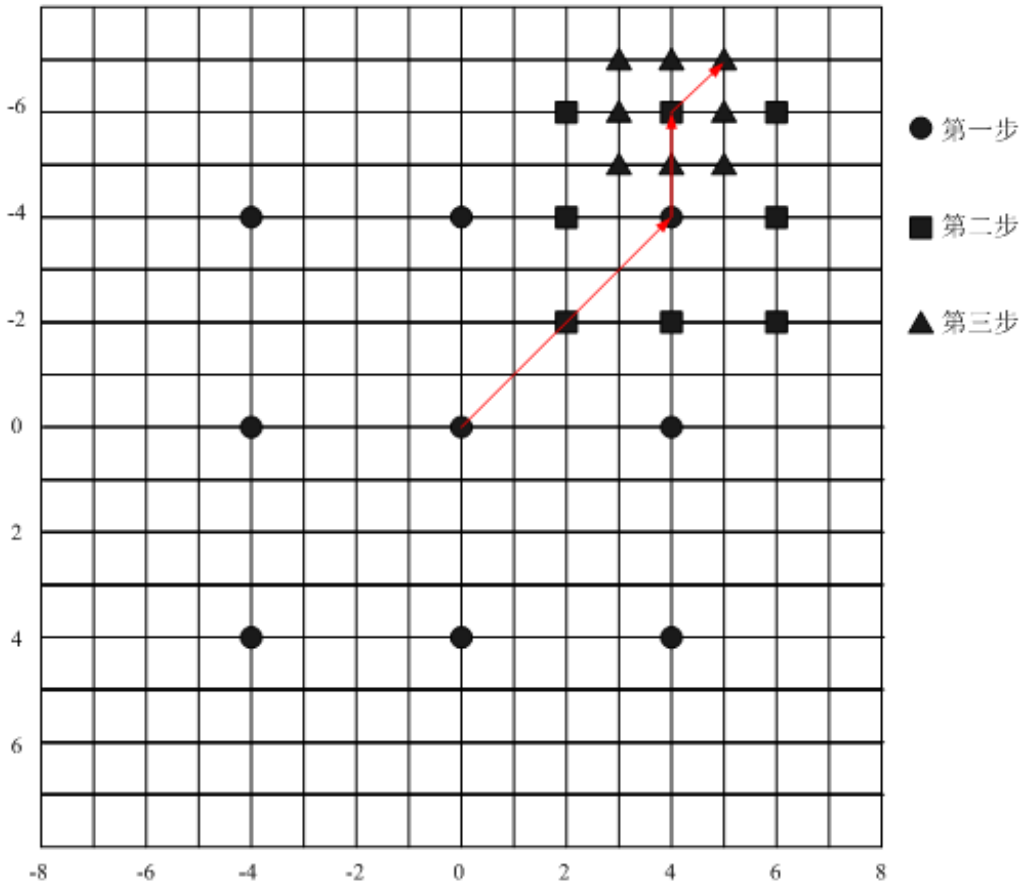
  采用SSD匹配准则：每次象素比较需1个减法，1个乘法，1个加法，则上述全搜索计算每秒需要2272788480x2次加减法和2272788480次乘法操作。

11. 快速运动估计

- 在保持预测精度的同时减少运动估计的搜索次数。
    - 三步搜索（Three Step Search，TSS）
    - 二维Log搜索（2D Logarithmic Search，2DLOG）
    - 正交搜索（Orthogonal Search Algorithm，OSA）
    - 十字搜索（Cross Search Algorithm，CSA）
    - 新三步搜索（New Three Step Search，NTSS）
    - 四步搜索（Four Step Search，FSS）
    - 共轭方向搜索（Conjugate Direction Search，CDS）
    - 梯度下降搜索（Gradient Descent Search，GDS）
    - 层次块搜索（Hierarchical Block Matching Algorithm，HBMA）

12. 三步搜索

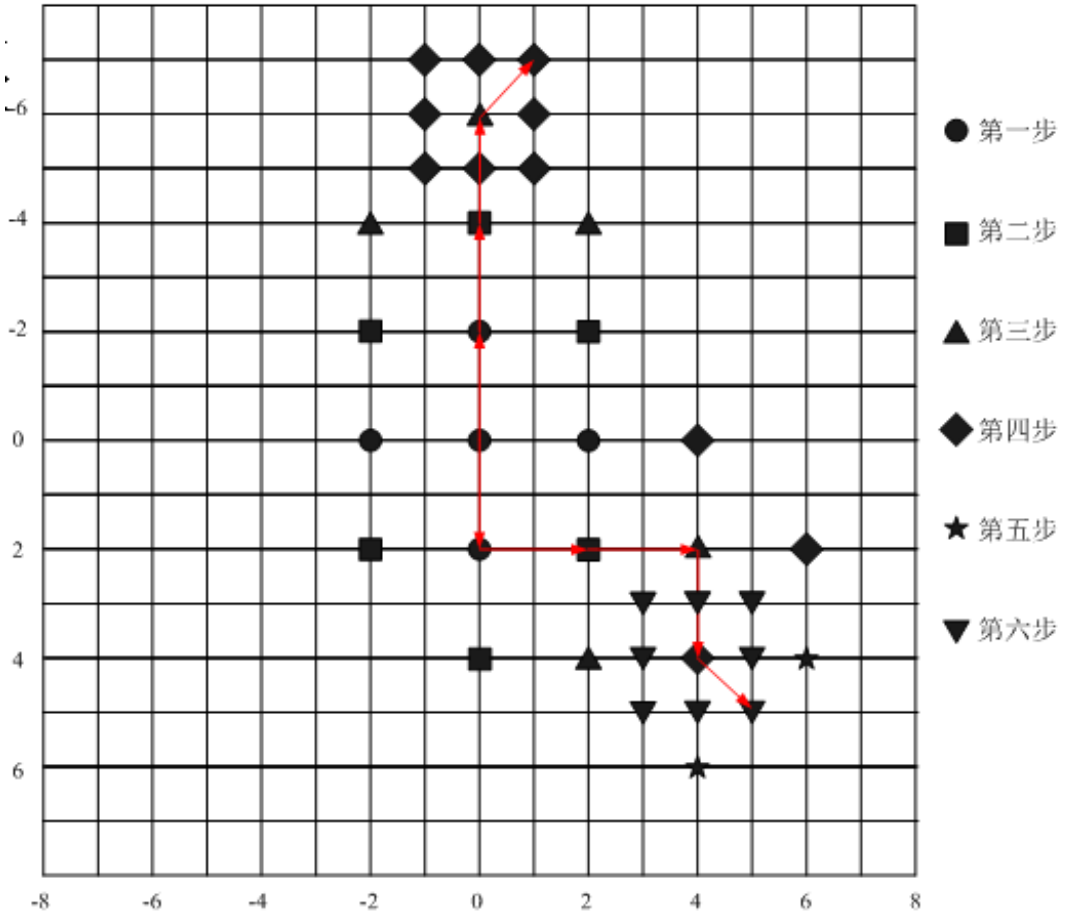
- 由粗到精搜索最优点，初始步长为R/2.
- 第一步：检查起始点和其周围步长为R/2的8个点，将最优点作为第二步的起始点；
- 第二步：以新的起始点为中心检查其周围步长为R/4的8个点，找到最优点作为第三步的起始点；
- 第三步：以新的起始点为中心检查其周围步长为R/8的8个点，找到最优点，如果R/8=1则搜索终止，最优点位置的预测块作为最优的预测块，否则重复该过程直到$R/n^2=1$；
- 三步搜索方法检查点的个数为$1+8\log_2$（d+1），当d=8时，检查点个数为9+8+8=25



13. 二维Log搜索

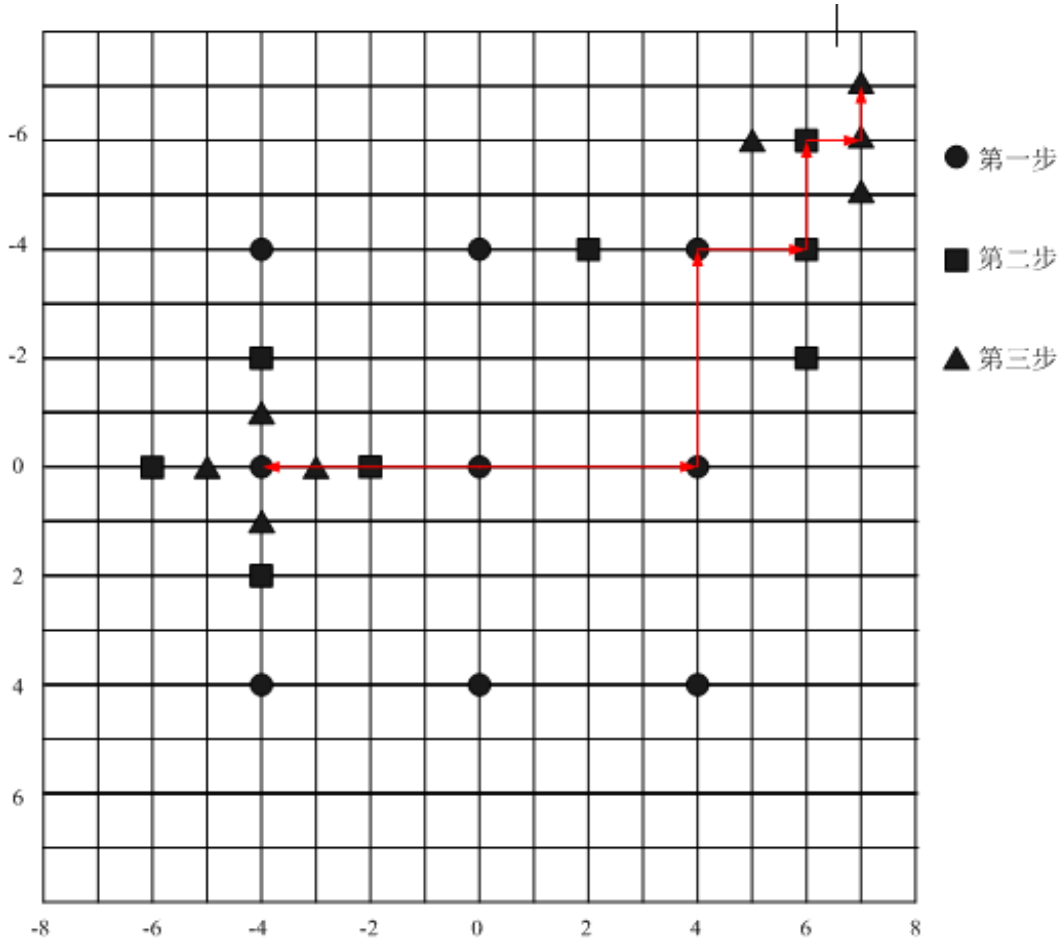
- 每一步采用十字搜索模式

- 如果每一步的最优点为中心点或者搜索窗的边界点，搜索步长减半，否则搜索步长不变
- 当搜索步长为1时，中心点周围的8个点都要检查
- 两个搜索路径一个需要5+3+3+8=19，另外一个需要5+3+2+3+2+8=23



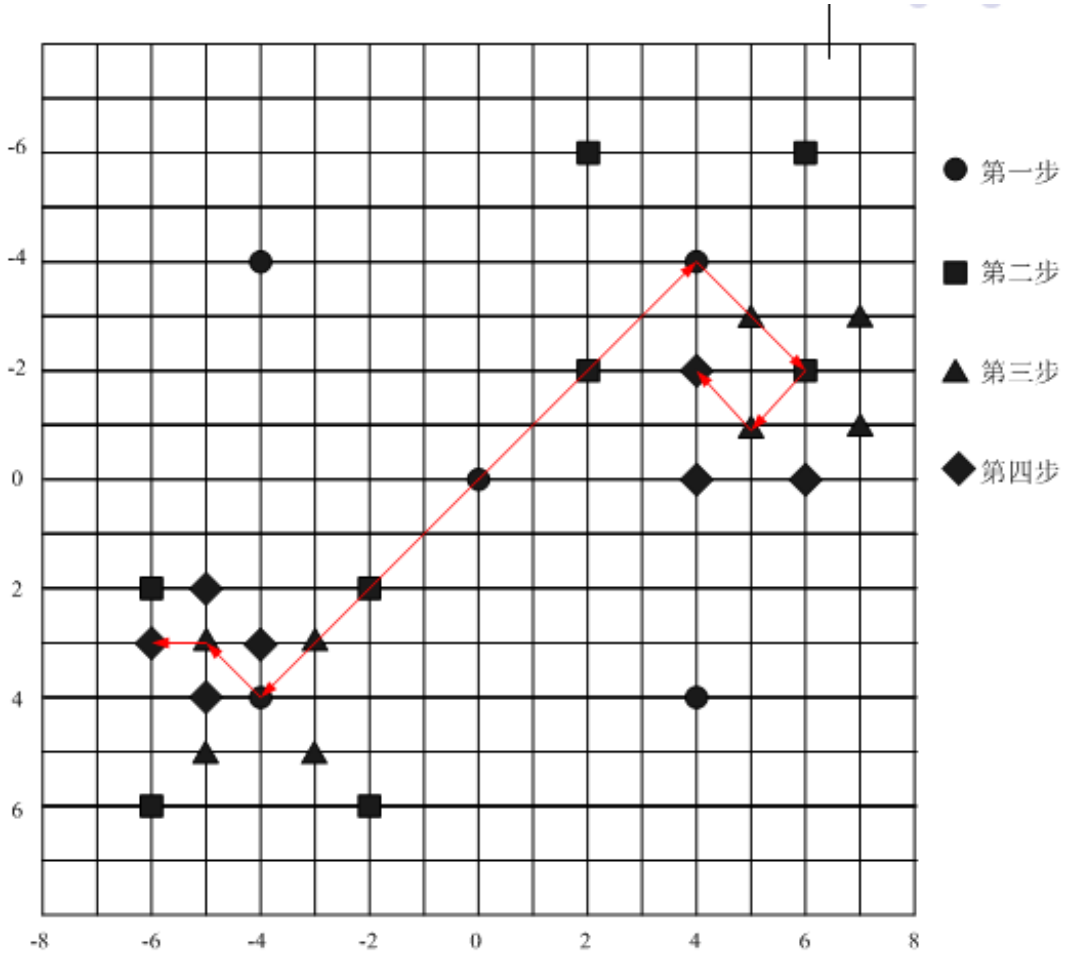
14. 正交搜索

- 起始搜索步长R/2，从起始点开始水平搜索三个点，得到最优点并沿着最优点垂直方向搜索相邻的两个点，得到最优点，以搜索步长为R/4再以同样的方式先水平再垂直搜索，当步长为1时停止搜索
- 搜索方法检查点的个数为$1+4\log_2(d+1)$，当d=8时，检查点个数为3+2+2+2+2+2=13。



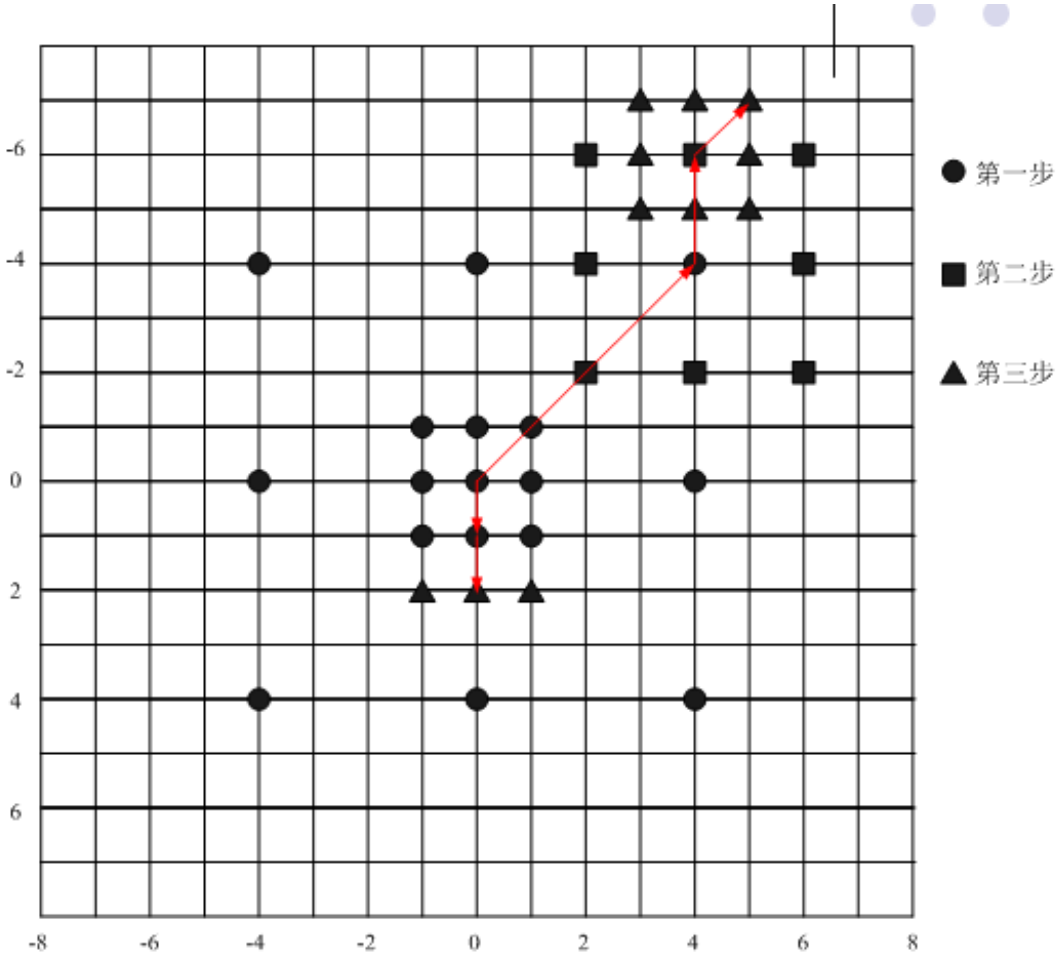
15. 十字搜索

- 起始搜索步长R/2，从起始点开始以"X"形十字搜索，当搜索步长降为1时，如果上一步的最优点为中心点，左上点或右下点，则这一步搜索以"+"形状十字搜索，然后结束搜索，否则还是以"X"形十字搜索，然后结束搜索。
- 十字搜索方法检查点的个数为$1+4\log_2 2d$，当d=8时，检查点个数为5+4+4+4=17



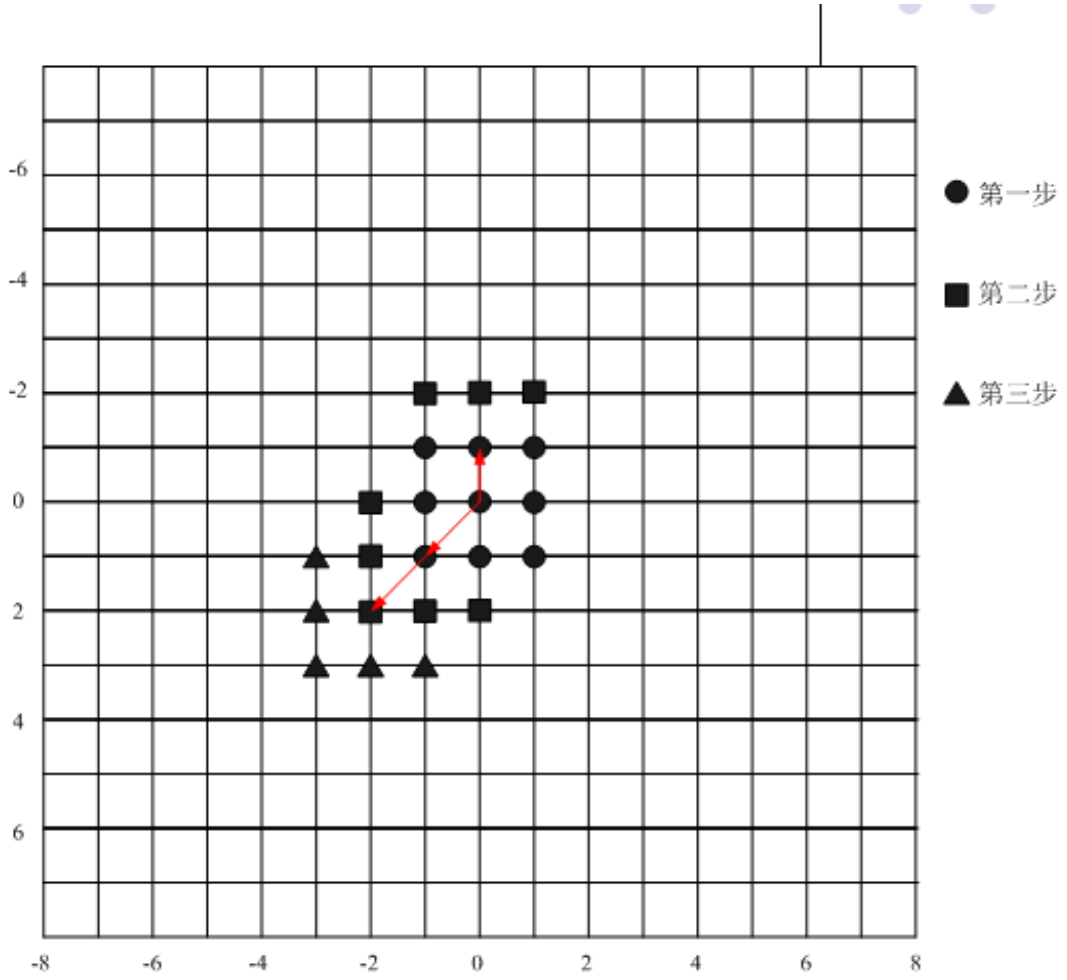
16. 新三步搜索

- 与三步搜索方法不同的是，考虑到运动矢量高的中心分布特点，新三步搜索方法，除了围绕起始点为中心搜索步长为R/2的8个点之外，在起始点周围增加了步长为1的8个搜索点，如果最优点为步长为1的8个搜索点之一，则在最优点邻近的三个点中搜索最优点，然后结束搜索，否则，和三步搜索方法过程一样
- 其中一个搜索路径需要检查点个数为17+3=20，另一个需要17+8+8=33。



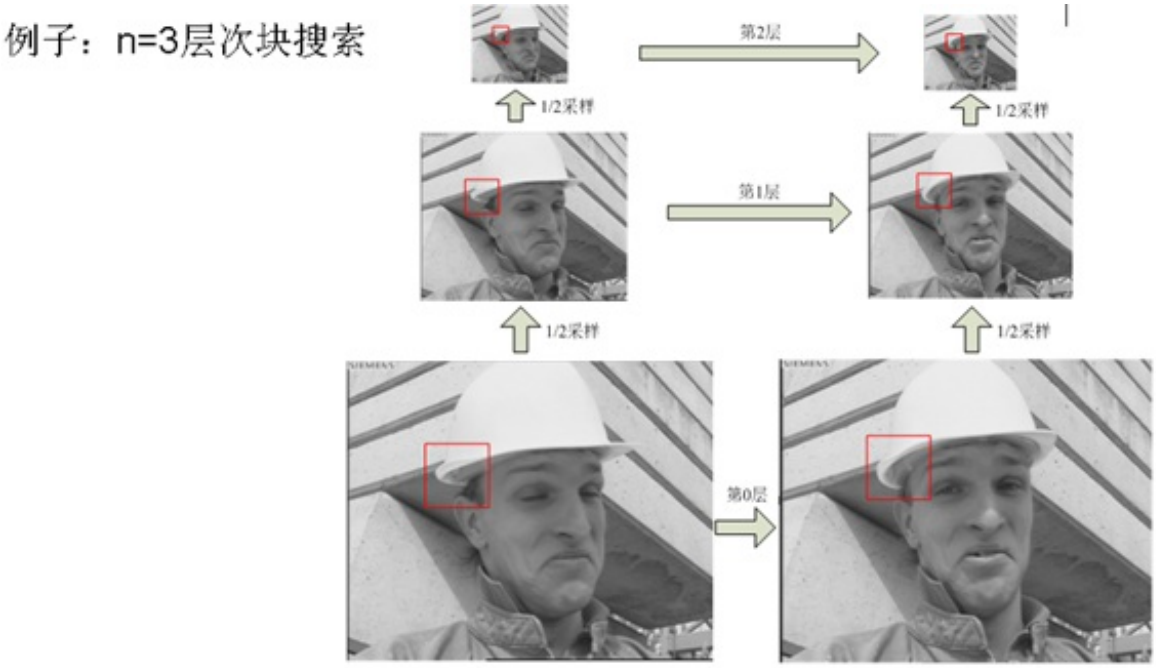
17. 块梯度下降搜索

- 该方法以起始点为中心搜索8个步长为1的相邻点，确定最优点，再以最优点为中心搜索8个步长为1的相邻点，如此循环下去，不限制搜索步骤，但当搜索得到的最优点为中心点或者到搜索窗的边界，搜索终止。



18. 层次块搜索

- 对编码图像和参考图像下采样，分别得到编码图像和参考图像的下采样图像，未经采样处理的编码图像和参考图像属于第0层，一次下采样的编码图像和参考图像属于第1层，对第1层图像再进行下采样得到的编码图像和参考图像属于第2层，依次重复上述过程，得到第n层下采样的编码图像和参考图像。
- 然后在n层下采样参考图像的搜索范围中找到与下采样编码图像块最佳匹配块的MV，该MV作为n-1层的运动估计搜索范围的中心点，依次重复上述过程，直到n=0为止，此时得到的最佳匹配块就是编码图像的预测块，其对应的MV为最终的最优MV。



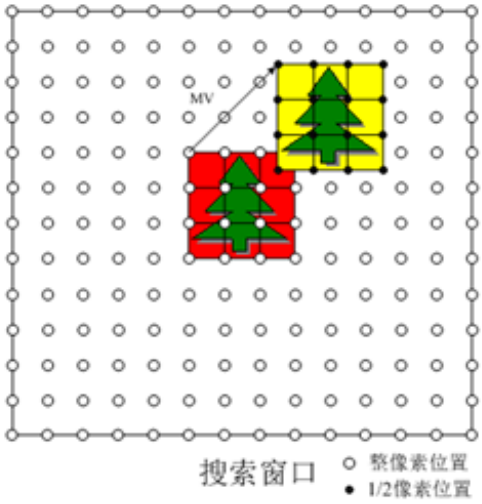
19. 搜索算法复杂度比较

算法	复杂度（最坏情况）	优点	缺点
FS	$(2R+1)^2$	能搜索到最佳的匹配块	非常高的计算代价
TSS	$1+8\log_2(d+1)$	保持预测性能的同时，较低的复杂度	不适合小的运动搜索
NTSS	$1+8\log_2(d+1)+8$	对于小的运动搜索比TSS更有效	对于大的运动搜索，比TSS更复杂
OSA	$1+4\log_2(d+1)$	复杂度比TSS更低	不适合小的运动搜索
CSA	$1+4\log_2 2d$	复杂度比TSS更低	搜索点较少，性能略低
HBMA	由算法决定	适用于高清图像运动估计	比TSS更复杂
GDS	9n(n为搜索步骤次数)	非常适合小运动搜索	对于大的运动搜索非常复杂

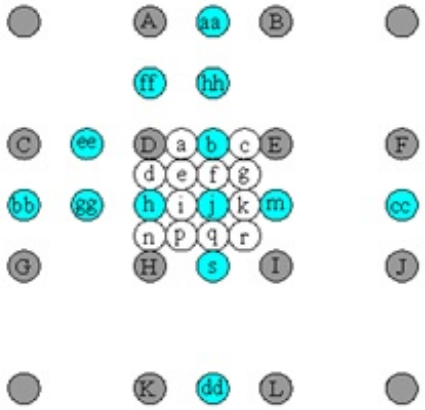
20 . 分像素运动估计与运动补偿

- 时域运动位置更可能在整象素之间，即分像素上。
- 利用相邻的整象素可以估计出分象素的值
  - 常用线性或双线性插值得到分象素的值。
- 分象素运动估计有更高的预测精度，但复杂度也更高，
  - 1/2分象素运动估计，图像存储空间增加4倍，运动矢量需要放大2倍，1/4分象素运动估计，图像存储空间增加16倍，运动矢量需要放大4倍，计算复杂度也成倍增加。

例：最佳预测块的MV不是$\begin{pmatrix} MV(x) \\ MV(y) \end{pmatrix} = \begin{pmatrix} 2 \\ -2 \end{pmatrix}$也不是$\begin{pmatrix} MV(x) \\ MV(y) \end{pmatrix} = \begin{pmatrix} 3 \\ -3 \end{pmatrix}$.....而是 $\begin{pmatrix} MV(x) \\ MV(y) \end{pmatrix} = \begin{pmatrix} 2.5 \\ -2.5 \end{pmatrix}$



21. 分像素插值



$$b = \alpha_1 C + \alpha_2 D + \alpha_3 E + \alpha_4 F$$

$$h = \alpha_1 A + \alpha_2 D + \alpha_3 H + \alpha_4 K$$

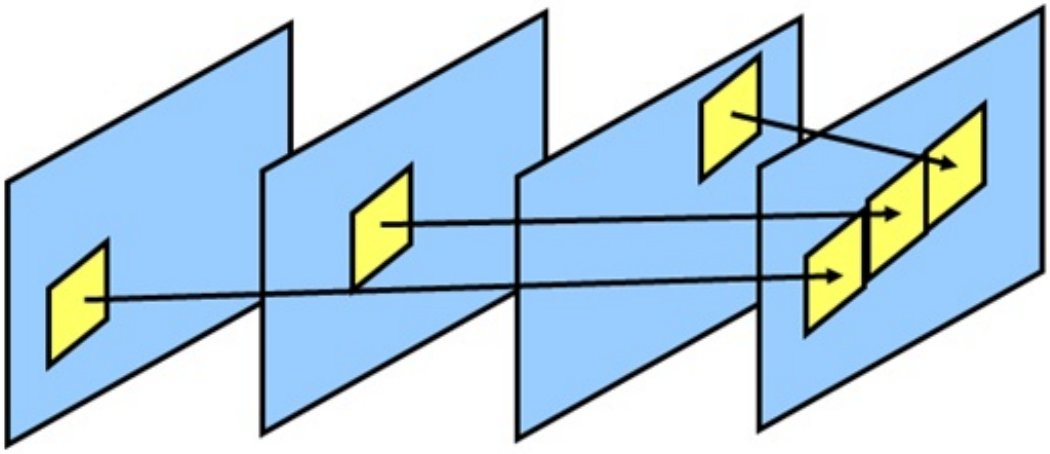
$$j = \alpha_1 bb + \alpha_2 h + \alpha_3 m + \alpha_4 cc$$

$$a = \beta_1 D + \beta_2 b$$

$$f = \beta_1 b + \beta_2 j$$

22. 多参考帧预测

- 有更多的候选图像，搜索更精确的预测块
- 需要更多的参考图像存储空间
- 码流需要标识参考帧索引的语法元素



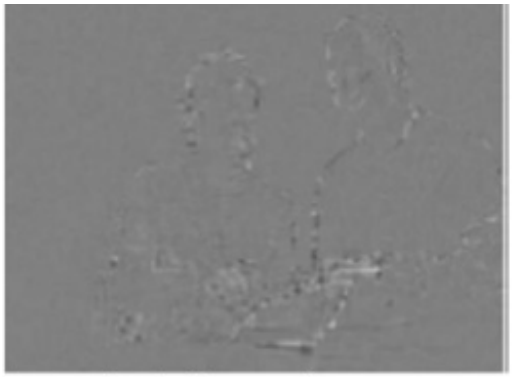
23. 图像分块编码

第1帧

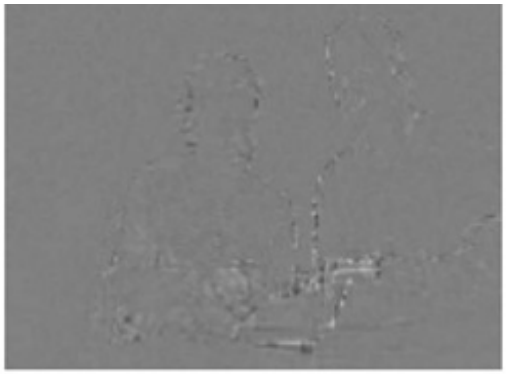
第2帧

预测残差（16x16块ME）

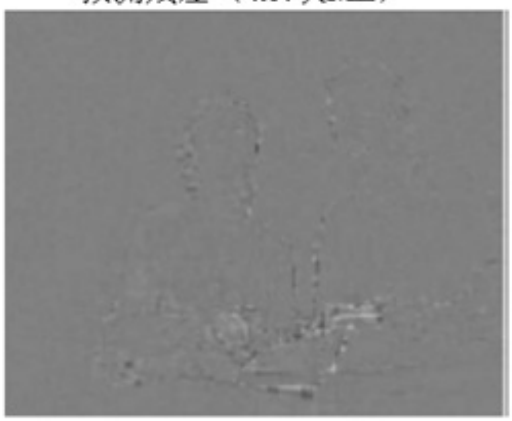
预测残差（8x8块ME）



预测残差（4x4块ME）

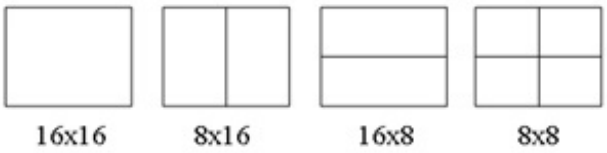


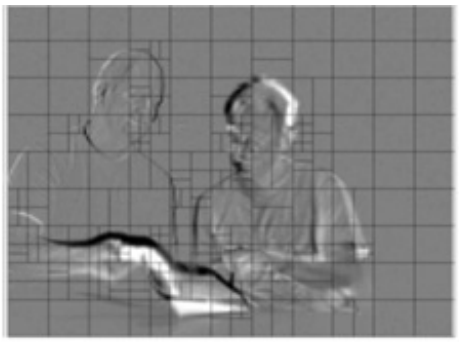
预测残差（4x4块，半象素ME）



预测残差（4x4块，1/4象素ME）

- 视频内容的运动非常复杂，图像分块编码可以更好的提高运动预测精度，提高压缩效率。
- 要在编码块大小和附信息（MV，Mode）编码比特数之间权衡，小的编码块大小会有更好的预测但有更多的附信息比特数。





变块大小编码图像

23. 双向预测编码

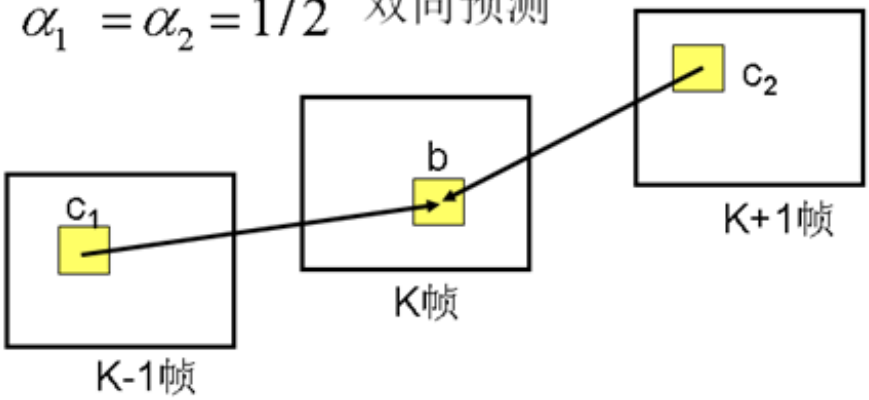
- 双向预测图像（B帧）的时域预测模型：

$$\hat{b} = \alpha_1 \hat{c}_1 + \alpha_2 \hat{c}_2,$$

$\alpha_1 = 1, \alpha_2 = 0$ 前向预测

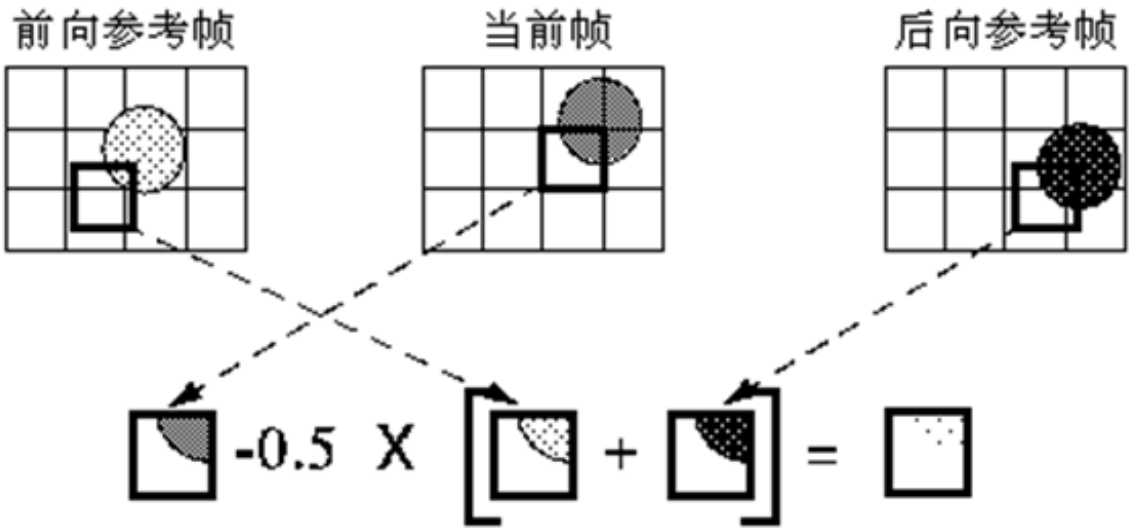
$\alpha_1 = 0, \alpha_2 = 1$ 后向预测

$\alpha_1 = \alpha_2 = 1/2$ 双向预测



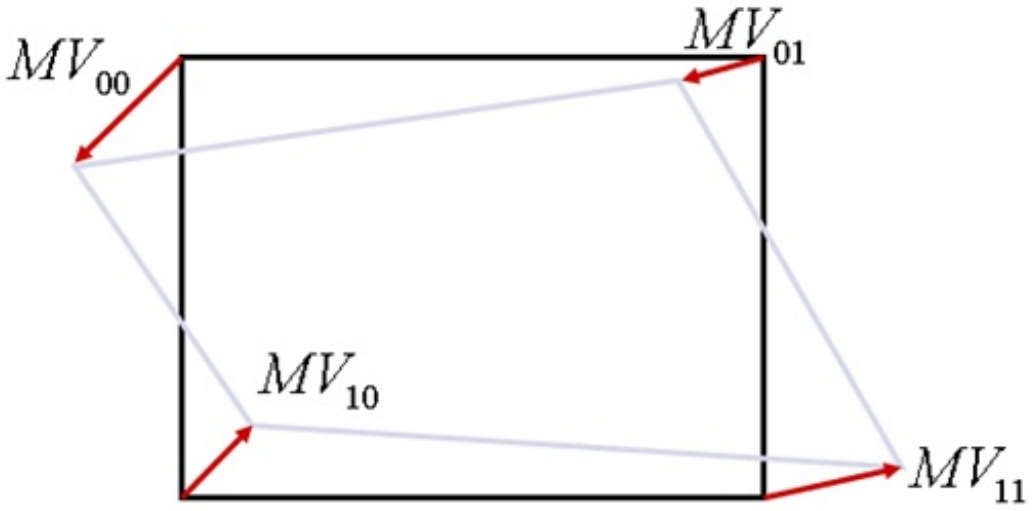
24. B帧有更好的编码效率

- B帧有更好的编码效率
  - 新出现的对象参考将来的帧有更好的预测效果
  - 前后两个预测的平均值可以减少预测方差



25. 全局运动估计

- 基于全局仿射运动模型
- 预测精度不如基于块的运动估计
- MV数目少，适合简单运动场景的运动估计


MV_{00}
MV_{01}
MV_{10}
MV_{11}