

数字图像与视频处理

卢官明 唐贵进 崔子冠 编著

机械工业出版社  
CHINA MACHINE PRESS



“十三五”江苏省高等学校重点教材(编号: 2017-2-029)  
高等院校通信与信息专业规划教材

# 数字图像 与视频处理

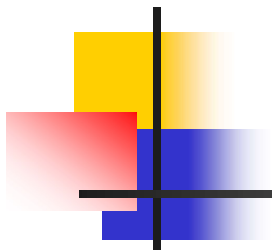
DIGITAL IMAGE AND VIDEO PROCESSING



卢官明 唐贵进 崔子冠 编著

机械工业出版社  
CHINA MACHINE PRESS

教学资源下载网址  
<http://www.cmpedu.com>



# 《图像与视频处理》

崔子冠

通信与信息工程学院

cui zg@njupt. edu. cn



## 第5章 数字图像与视频压缩编码原理

---

- 5.1 数字图像与视频压缩编码概述
- 5.2 熵编码
- 5.3 预测编码
- 5.4 变换编码
- 5.5 MATLAB编程实例



## 5.1 数字图像与视频压缩编码概述

---

- 5.1.1 数字图像与视频压缩的**必要性和可能性**
- 5.1.2 数字图像与视频压缩编码的**主要方法及其分类**

## 5.1.1 数字图像与视频压缩的必要性和可能性

- 图像与视频压缩的必要性：
- 图像格式为720x576，25fps，YUV4:2:2，8bit量化的数码率为：
- $(720 \times 576 + 360 \times 576 + 360 \times 576) \times 25 \times 8 \text{bit} = 165.888 \text{Mbit/s} = 20.736 \text{MB/s}$  ；  
650MB的光盘只能存储31s的视频。
- 因此，需要对图像和视频进行有效的压缩
  - 1) 节约存储空间
  - 2) 提高传输效率
- 数据压缩的理论基础是信息论。从信息论的角度来看，压缩就是去掉数据中的冗余，即保留不确定的信息，去掉确定的信息（可推知的），也就是用一种更接近信息本质的描述来代替原有冗余的描述。
- 在一般的图像和视频数据中，主要存在以下几种形式的冗余。

## 5.1.1 数字图像与视频压缩的必要性和可能性

- 1) 空间冗余：也称为空域冗余，是一种与像素间相关性直接联系的数据冗余。

例：图像中包含许多规则物体，它们的亮度、饱和度及颜色可能都一样，因此，图像在空间上具有很强的相关性。例如Lena图像的脸部和肩部。



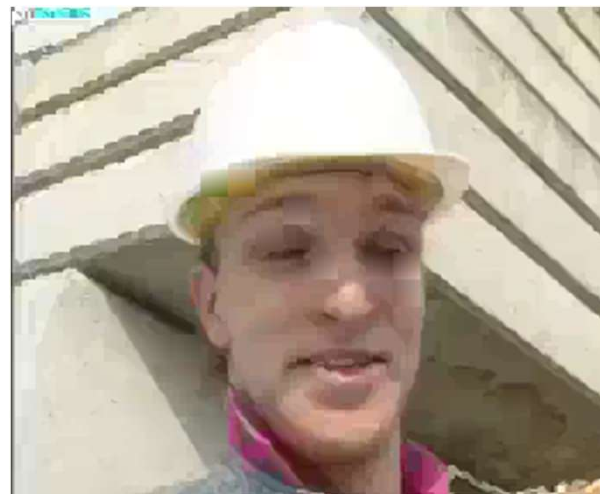


## 5.1.1 数字图像与视频压缩的必要性和可能性

### 1) 空间冗余:



原始帧



预测帧



残差帧

## 5.1.1 数字图像与视频压缩的必要性和可能性

- 2) 时间冗余：也称为时域冗余，它是针对视频序列图像而言的。
- 视频序列每秒有25~30帧图像，相邻帧之间的时间间隔很小；同时实际生活中的运动物体具有运动一致性，使得视频序列图像之间有很强的相关性。



$t$



$t+1$



## 5.1.1 数字图像与视频压缩的必要性和可能性

### ■ 3) 统计冗余（编码表示冗余、符号冗余）

- **信源熵**：如果将信源所有可能事件的信息量进行平均，就得到了信源熵(entropy)。熵就是平均信息量。

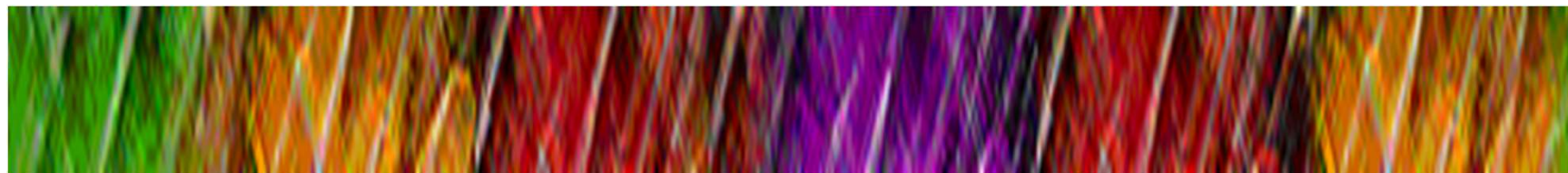
$$H(X) = E\{I(x_j)\} = \sum_{j=1}^n P(x_j) \cdot I(x_j) = -\sum_{j=1}^n P(x_j) \cdot \log_2 P(x_j)$$

- 当  $x_j$  等概率时， $H(X)$  最大。
- 当  $x_j$  非等概率时， $H(X)$  不是最大，就存在冗余。

采用可变长编码技术，对出现概率大的符号用短码字表示，对出现概率小的符号用长码字表示，则可去除符号冗余，从而节约码字，这就是熵编码的思想。如五笔输入法。

## 5.1.1 数字图像与视频压缩的必要性和可能性

- **4) 结构冗余：**在有些图像的部分区域内有着很相似的纹理结构，或是图像的各个部分之间存在着某种关系，例如自相似性等，这些都是结构冗余的表现。

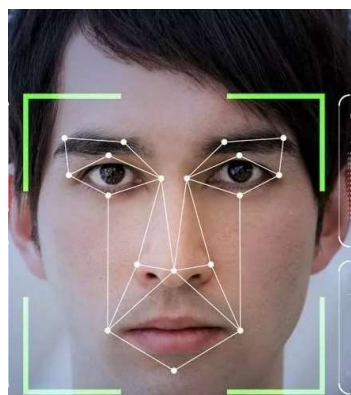
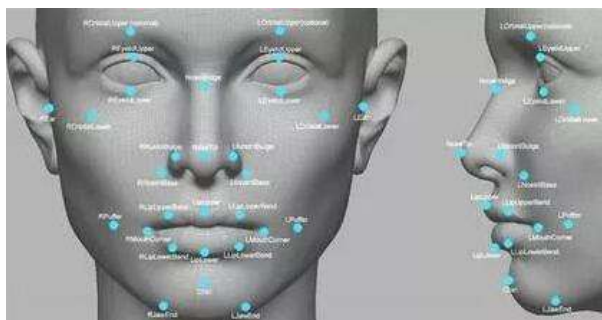


**分形图像编码**的基本思想就是利用了结构的自相似性。

## 5.1.1 数字图像与视频压缩的必要性和可能性

- **5) 知识冗余：**在某些特定的应用场合，编码对象中包含的信息与某些先验的基本知识有关。例如：人脸的图像有同样的结构：嘴的上方有鼻子，鼻子上方有眼睛，鼻子在中线上等等；

可以利用这些先验知识为编码对象建立模型。通过提取模型参数，**对参数进行编码**而不是对图像像素值直接进行编码，可以达到非常高的压缩比。这是**模型基编码**（或称知识基编码、语义基编码）的基本思想。





## 5.1.1 数字图像与视频压缩的必要性和可能性

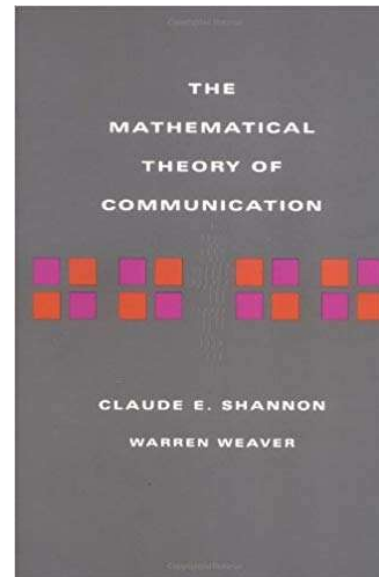
- **6) 人眼的视觉冗余**
  - **视觉冗余度是相对于人眼的视觉特性而言的。**
    - **HVS对亮度比对色度敏感；**
    - **对低频比对高频敏感；**
    - **对静止图像比对运动图像敏感；**
    - **对水平垂直线条比对斜线敏感；等。**
  - **压缩视觉冗余的核心思想是去掉那些相对人眼而言是看不到的或可有可无的图像数据。对视觉冗余的压缩通常反映在各种具体的压缩编码过程中。**
- **综上撰述，图像与视频压缩编码的目的，是在保证重建图像质量一定的前提下，以尽量少的比特数来表征图像或视频信息。**

### 5.1.2 数字图像与视频压缩编码的主要方法及其分类

- 1948年C.E. Shannon（香农）发表了经典论文《通信的数学原理》首次提出信息率-失真函数，并于1959年确立率失真理论，奠定了信源编码的理论基础。



香农, C.E.



A Mathematical Theory of Communication(C Shannon,1948) 被引量: 5.7万



### 5.1.2 数字图像与视频压缩编码的主要方法及其分类

---

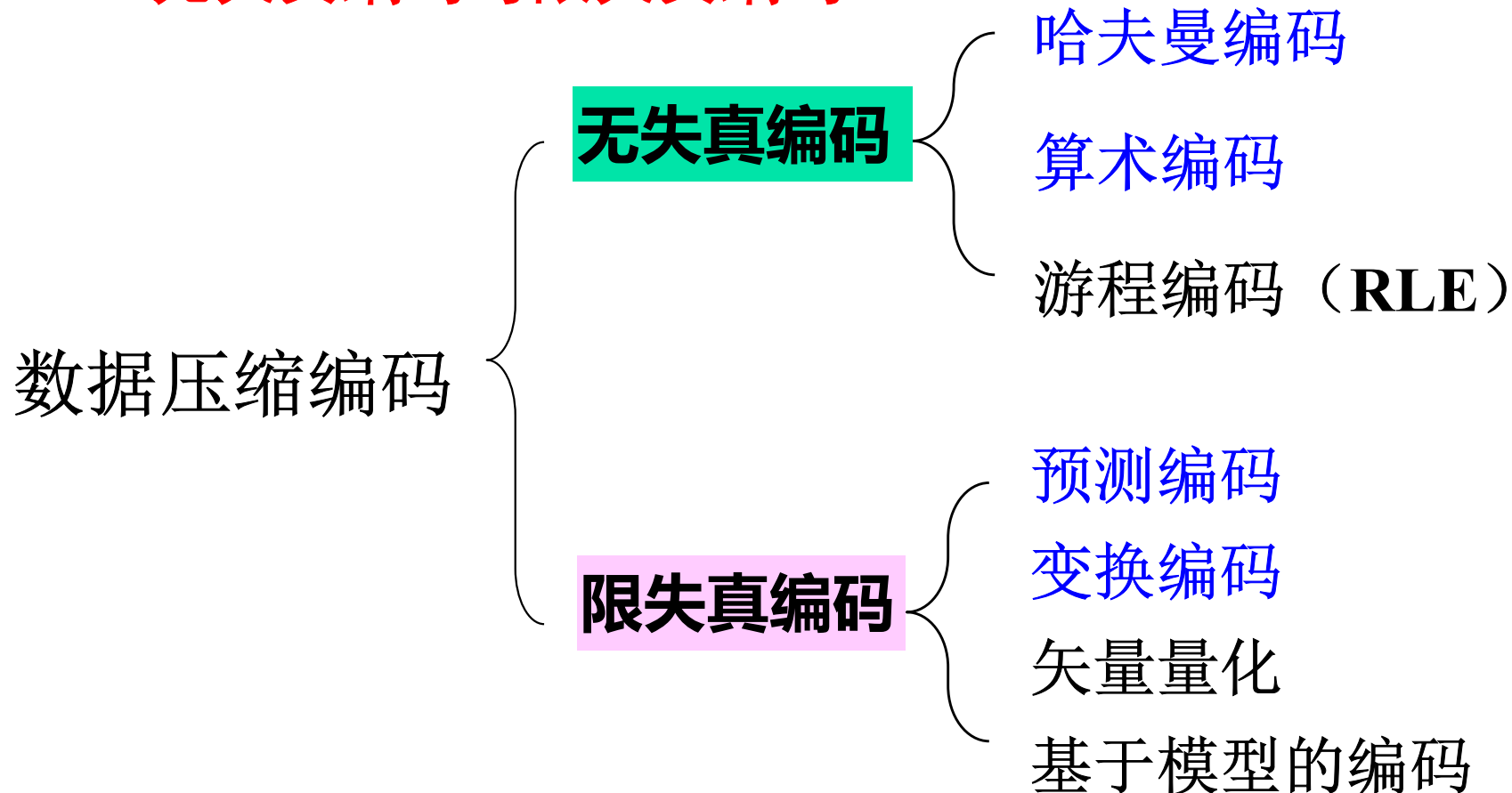
- **1952年D.A. Huffman**（哈夫曼）提出了最优变长编码方法。即哈夫曼编码。
- **1958年Graham**研究了差分脉冲编码调制（**DPCM**）方法。
- **20世纪60年代**开始探索变换编码，包括**K-L变换**、**DFT变换**、**Walsh-Hadamard变换**、斜变换、**DCT变换**等。
- **1986年后**，小波变换（**Wavelet**）开始流行，具有多分辨率和多尺度分析的优点。



## 5.1.2 数字图像与视频压缩编码的主要方法及其分类

- 从信息论的角度出发，根据解码后还原的数据是否与原始数据完全相同，数字图像编码可分为：

- 无失真编码与限失真编码





## 5.1.2 数字图像与视频压缩编码的主要方法及其分类

- **1) 无失真编码**
- 无失真编码又称无损编码、信息保持编码，熵编码。
- 熵编码是纯粹基于信号统计特性的一种编码方法，它利用信源概率分布的**不均匀性**，通过变长编码来减少信源数据冗余，解码后还原的数据与压缩编码前的原始数据完全相同**而不引入任何失真**。
- 熵编码的压缩比较低，可达到的最高压缩比受到信源熵的理论限制，一般为2：1到5：1。
- 最常用的熵编码方法有：1) 哈夫曼(Huffman)编码、2) 算术编码和3) 游程编码(Run-Length Encoding, RLE)等。
- 常用于不希望有失真的图像压缩，如指纹图像、医学图像等。

## 5.1.2 数字图像与视频压缩编码的主要方法及其分类

The screenshot shows a music player interface for the song "夜夜夜夜" (Night Night Night Night) by 郭宴 (Guo Yan). The album is "宴乐·声色" (Yan Le · Sheng Se). The interface includes a song cover, lyrics, and a playback bar.

**Song Information:**

- 专辑: 宴乐·声色
- 歌手: 郭宴

**Lyrics:**

慢慢的拼凑 慢慢的拼凑  
拼凑成一个完全不属于真正的我  
你也不必牵强再说爱我  
反正我的灵魂已片片凋落  
慢慢的拼凑 慢慢的拼凑  
拼凑成一个完全不属于真正的我  
我不愿再放纵  
我不愿每天每夜每秒飘流

**Similar Songs:**

喜欢这首歌的人也听

郭宴  
3.2万

**Comments:**

留言评论 16条

期待你的精彩评论~

**Audio Quality Options:**

- 无损音质-免费
- 超品音质-320K
- 高品音质-128K
- 流畅音质-WMA

**Playback Bar:**

03:30 / 05:02 无损

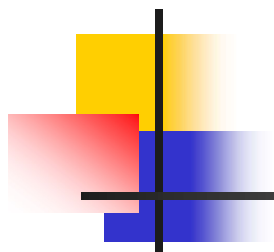
1.0x [Volume Icon] [Equalizer Icon] [HIFI] [Lyrics Icon] [Heart Icon] [Share Icon] 28



## 5.1.2 数字图像与视频压缩编码的主要方法及其分类

### ■ 2) 限失真编码

- 限失真编码也称有损编码、非信息保持编码、熵压缩编码。解码后数据与原始数据之间**存在失真**。
- 限失真编码方法利用了人类视觉的感知特性，允许压缩过程中损失一部分信息，虽然在解码时不能完全恢复原始数据，但是如果把失真控制在视觉阈值以下或控制在可容忍的限度内，**则不影响人们对图像的理解，却换来了高压缩比**。在限失真编码中，允许的失真愈大，则可达到的压缩比愈高。
- 常见的限失真编码方法有：预测编码、变换编码、矢量量化、基于模型的编码等。



## 第5章 数字图像与视频压缩编码原理

- 5.1 数字图像与视频压缩编码概述
- **5.2 熵编码**
- 5.3 预测编码
- 5.4 变换编码
- 5.5 MATLAB编程实例



## 5.2 熵编码

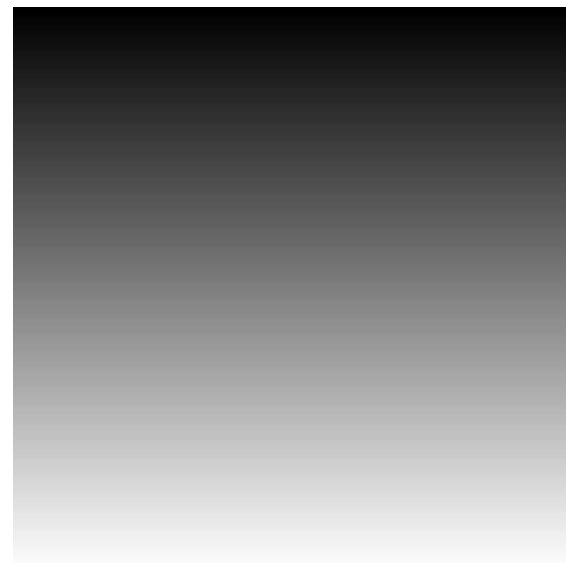
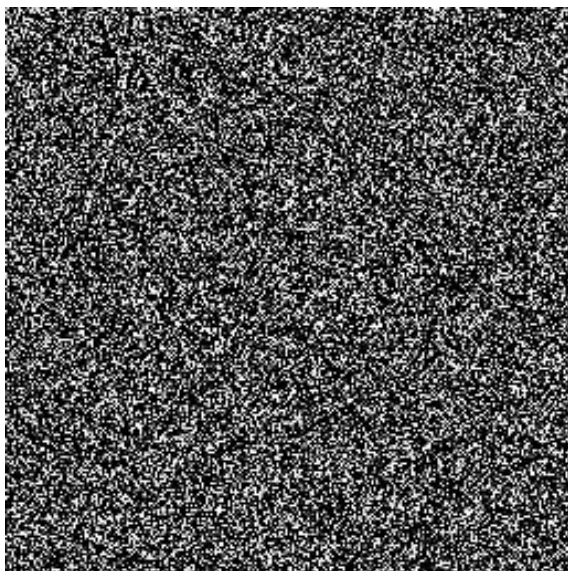
---

- **熵编码的基本原理**就是去除图像信源在空间和时间上的相关性，去除图像信源像素值的概率分布不均匀性，使编码码字的平均码长接近信源的熵而不产生失真。
- 由于这种编码完全基于图像的统计特性，因此，有时也称其为**统计编码**。
  - 1) 游程编码(Run-Length Encoding, RLE)
  - 2) 哈夫曼(Huffman)编码
  - 3) 算术编码



## 5.2.1 图像的信源熵

- 图像的统计特性：即图像的灰度值在统计意义上的分布特性。
- 图像多种多样，但实际有意义的图像其内部像素是有相关性的，而不是随机的。
- 对于256x256大小，每像素用8bit表示的灰度图像，其图像总数为： $(2^8)^{256 \times 256}$
- 如 `I=randn(256), imshow(I);`





## 5.2.1 图像的信源熵

- 在信息论中：信源 $X$ 可发出的符号为 $a_i$ ，发出符号 $a_i$ 的概率为 $p(a_i)$ ，则**符号 $a_i$ 出现的自信息量为**：
  - $I(a_i) = -\log(p(a_i))$
- **对无记忆信源 $X$** ，即 $a_i$ 的出现相互独立，则 $X$ 发出一符号序列的概率=各符号出现概率之积，即该序列出现的信息量为各符号的自信息量之和，**而每个符号的平均信息量为**：

$$H(X) = -\sum_{i=1}^m P(a_i) \cdot \log_2 P(a_i)$$

- $H(X)$ 为信源 $X$ 的熵(Entropy)，单位是bit/符号，**其含义是信源 $X$ 发出任意一个符号的平均信息量。**



## 5.2.1 图像的信源熵

- 但实际上都是有记忆信源 $X$ ，即 $a_i$ 出现的概率与之前出现的符号相关；常简化为 $N$ 阶马尔可夫过程，即 $a_i$ 出现的概率只与之前出现的 $N$ 个符号相关，而与再之前的 $N+1$ 等符号无关；
- 此时一般将这 $N$ 个相关的符号序列当作一个新的符号 $B_i(N)$ 进行分析。
- 对图像而言：
  - 以像素为单位，并考虑多个像素之间的相关性来得到高阶熵，则高阶熵小于低阶熵。
  - 即高阶熵考虑了更多像素之间的相关性。



## 5.2.1 图像的信源熵

- **图像信源熵**：是图像压缩编码的一个理论极限，表示无失真编码所需的**比特率的下限**。
- **比特率**：编码表示一个像素所需的平均比特数，单位是**bpp**，即**bit per pixel**。
- **熵编码**：要求编码输出的平均码长，只能大于等于信源熵，否则压缩编码过程就会丢失信息。
- **信源压缩编码的目的**：就是在一定信源概率分布条件下，尽可能使编码码字的平均码长接近信源熵，减少冗余。
- **图像熵编码的基本原理**：去除图像信源在空间和时间上的相关性，利用图像像素值概率分布的不均匀性，来压缩图像数据。



## 5.2.2 游程编码

- 游程编码（RLE），也称行程编码或游程（行程）长度编码，其基本思想是将具有相同数值（例如，像素的灰度值）的、连续出现的信源符号构成的符号序列用其数值及串的长度表示。
- 以图像编码为例，灰度值相同的相邻像素的连续长度（像素数目）称为连续的游程，又称游程长度，简称游程。

原像素值串：[100,100,100,100,100,100, 100,100,100,100]

RLE编码后：[100,10]



## 5.2.2 游程编码

- 以二值图像为例进行说明。

- 二值图像是指图像中的像素值只有两种取值，即“0”和“1”，**因而在图像中这些符号会连续地出现**，将连“0”这一段称为“0”游程，而连“1”的一段则称为“1”游程，它们的长度分别表示为 **$L(0)$ 和 $L(1)$** ，往往“0”游程与“1”游程会交替出现，即第一游程为“0”游程。第二游程为“1”游程。第三游程又为“0”游程。

- 以一个具体的二值序列为例进行说明。

- 已知一个二值序列**00101110001001.....**，根据游程编码规则，可知其游程序列为**21133121.....**。





## 5.2.2 游程编码

---

- 显然，平均游程长度越长，游程编码的效率越高。
- 由于需要保证一个游程内所有像素值要相同，因此游程编码不太适合多值的灰度图像，因为灰度级越多，越难产生长游程。
- 因此游程编码多用于二值图像或经过处理的变换系数编码。



## 5.2.3 哈夫曼编码

- **Huffman**于1952年提出一种编码方法，完全依据**符号出现概率**来构造异字头（前缀）的平均长度最短的码字，有时称之为**最佳编码**。
- **哈夫曼（霍夫曼）编码**是一种可变长度编码（Variable Length Coding, VLC），各符号与码字一一对应，是一种分組码。
- **变字长编码的最佳编码定理：**
  - 对出现概率大的符号以短字长编码，概率小的符号长字长编码；
  - 若码字长度严格按照符号出现概率的大小逆序排列，则平均码字长度一定小于其它任何符号排列方式。
  - **如五笔字型编码，依据汉字使用概率。**

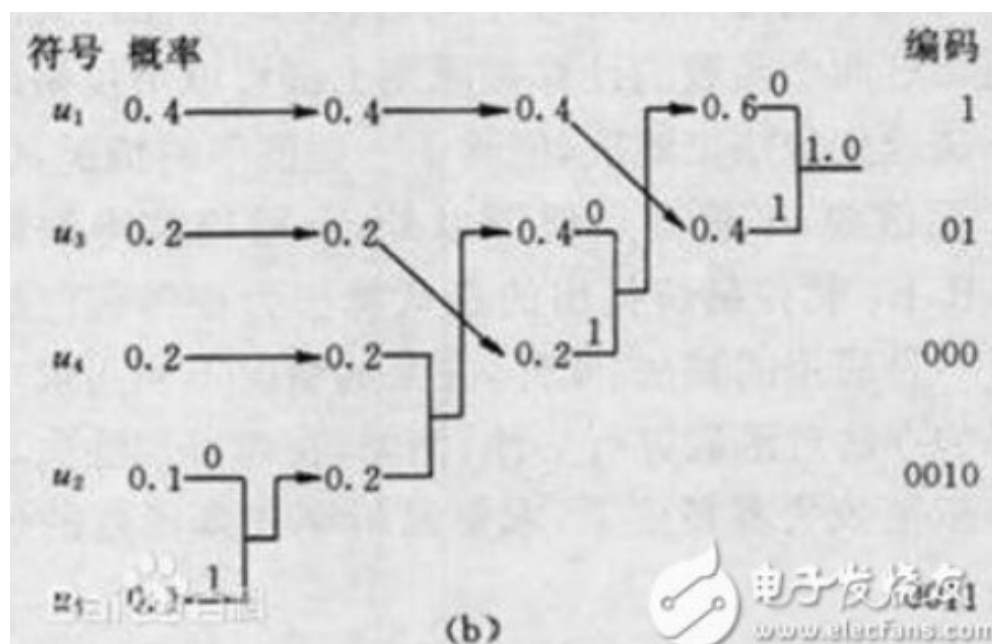
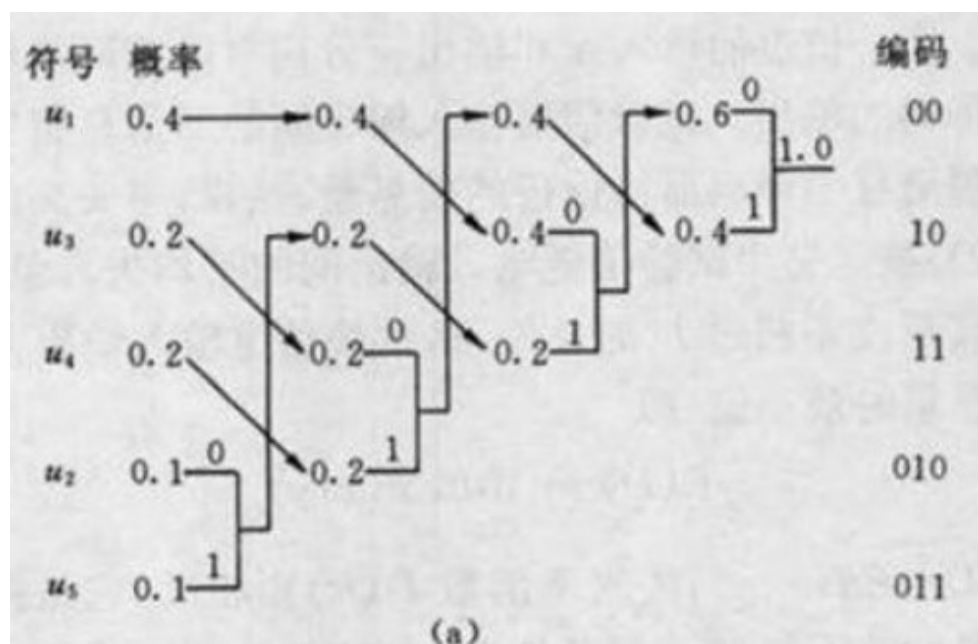


## 5.2.3 哈夫曼编码

### ■ 哈夫曼编码过程：

- 1) 将待编码的 $N$ 个信源符号按出现的概率由大到小排列，给排在最后的两个符号的**最末一位码**各赋一个二进制码元，如对概率大符号赋0，对概率小的符号赋1；
  - 此步只确定了出现概率最小的两个符号的最末一位码元，这两个符号有相同的码长，码字只有最末一位不同，前面各位均相同，要由后续步骤来确定。
- 2) 把最后两个符号的概率相加，求出的和作为一个新符号的出现概率，再按步骤1) 的方法，对排在前面的 $N-2$ 个符号和新符号重新排序，重复步骤1) 的编码过程；
- 3) 重复步骤2) ，直到最后只剩下两个概率值为止；
- 4) **分配码字。**
  - 码字分配从最后一步反向进行，即每个符号的哈夫曼编码结果就是从根结点经中间结点到达叶结点的路径上遇到的二进制码元的序列。

- **例5-别为**



## 5.2.3 哈夫曼编码

- 例5-1 设有离散无记忆信源，符号 $x_1, x_2, x_3, x_4, x_5$ 出现的概率分别为0.4, 0.2, 0.2, 0.1, 0.1，其哈夫曼编码的结果如下：

符号 $x_i$	概率 $p(x_i)$	码字 $W_{i1}$	码长 $K_{i1}$	码字 $W_{i2}$	码长 $K_{i2}$
$x_1$	0.4	00	2	1	1
$x_2$	0.2	10	2	01	2
$x_3$	0.2	11	2	000	3
$x_4$	0.1	010	3	0010	4
$x_5$	0.1	011	3	0011	4

## 5.2.3 哈夫曼编码

- 编法a的平均码长为

$$\bar{K}_1 = \sum_{i=1}^5 p(x_i)K_i = 0.4 \times 2 + 0.2 \times 2 \times 2 + 0.1 \times 3 \times 2 = 2.2$$

- 编法b的平均码长为

$$\bar{K}_2 = \sum_{i=1}^5 p(x_i)K_i = 0.4 \times 1 + 0.2 \times 2 + 0.2 \times 3 + 0.1 \times 4 \times 2 = 2.2$$

- 两种编法平均码长相同，所以编码效率相同

$$H(X) = -\sum_{i=1}^5 p(x_i) \log_2 p(x_i) = 2.12$$

$$\eta = \frac{H(X)}{\bar{K}} = \frac{2.12}{2.2} = 96.4\%$$





## 5.2.3 哈夫曼编码

### ■ 哈夫曼编码的特点:

- 1) 哈夫曼编码的算法是确定的，但编码结果并不唯一。
  - 如对大或小的概率用1表示；
  - 排序过程中两个概率相同时；
- 2) 其依据信源符号的概率分布，因此其编码效率取决于信源的统计特性。
  - 当信源符号概率相等时，编码效率最低；
  - 概率分布很不均匀时，编码效率高；当概率分布服从 $2^{-n}$ 时，编码效率为100%，平均码长能降到信源熵。
- 3) 没有错误保护功能。
- 4) 是可变长编码，码字字长参差不齐，硬件实现不太方便。
- 5) 编码会形成哈夫曼码表，在解码时，必须对照此码表才能正确解码。



## 5.2.4 算术编码

- **算术编码**也利用信源概率分布特性，但编码原理与哈夫曼编码不同，不是按符号编码，即不是为每个符号分别确定码字，而是从整个符号序列出发，用一个算术码字表示整个信源符号序列。
- 算术编码将被编码的信源符号序列映射为实数半开区间  $[0, 1)$  中的一个小区间，其长度等于该序列的概率。
- 这个小区间随着信源符号序列中每一个信源符号的加入逐步减小，每次减小的程度取决于当前加入的信源符号的先验概率。



## 5.2.4 算术编码-方法1（要求掌握）

- 算术编码过程是在  $[0, 1)$  区间上划分子空间的过程。
- 算术编码步骤如下：
  - ① **初始化**：编码器将**当前区间**  $[low, high)$  设置为  $[0, 1)$ 。
  - ② 对每一个信源符号，分配一个初始编码子区间  $[symbol\_low, symbol\_high)$ ，其长度与符号出现的概率成正比。当输入符号序列时，编码器在当前区间内按照每个信源符号初始编码子区间的划分，选择对应于当前输入符号的子区间，**使它成为新的当前区间**  $[low, high)$ 。
  - ③ 重复第②步，最后输出**当前区间**  $[low, high)$  **左端点值**  $low$  就是该序列的编码结果。

## 5.2.4 算术编码-方法2（了解即可）

- 算术编码首先假设一个概率模型，然后用这些概率来缩小表示信源集的区域。
- 在算术编码的初始阶段，设置两个专用寄存器C和A来存储符号到来之前子区间的状态参数，令寄存器C的值为子区间的起始位置，寄存器A的值为子区间的宽度，该宽度恰好是已输入符号串的概率。再设L和H分别为编码字符的初始编码区间的低端和高端值。

初始时， $C=0$ ， $A=1$ 。

当新的符号到来时， $C \leftarrow C + A \times L$ ，  
 $A \leftarrow A \times (H - L)$ 。

算术编码的结果落在子区间 $[C, C+A)$ 之内。

- 输入的符号串中，符号出现的概率越大，则对应的子区间越宽，就可用长度较短的码字表示；符号出现概率越小，对应子区间就越窄，则需要较长的码字来表示。



## 5.2.4 算术编码

---

- **例5-2:** 假设信源符号为 $X=\{A:00, B:01, C:10, D:11\}$ , 这些符号的概率分别为 $P(X)=\{0.1, 0.4, 0.2, 0.3\}$ 。根据这些概率, 可把间隔 $[0, 1)$  分成4个子间隔:  $[0, 0.1)$ ,  $[0.1, 0.5)$ ,  $[0.5, 0.7)$  和  $[0.7, 1)$ 。
- 如果输入的符号序列为**CADACDB**, 求其算术编码。

## 5.2.4 算术编码-方法1（要求掌握）

### ■ 算术编码步骤如下：

- ① **初始化：**设置当前区间的 $low=0, high=1, length=1$ 。
- ② 对每一个信源符号，使用如下递推公式更新 $[low, high)$ ；

$$\begin{cases} low = low + length \times symbol\_low \\ high = low + length \times symbol\_high \end{cases}$$

当前区间的长度：

$$length = high - low$$

1) 输入C：

$$\begin{cases} low = low + length \times symbol\_low = 0 + 1 \times 0.5 = 0.5 \\ high = low + length \times symbol\_high = 0 + 1 \times 0.7 = 0.7 \end{cases}$$

$$length = high - low = 0.7 - 0.5 = 0.2$$

2) 输入CA：递推计算；

## 5.2.4 算术编码-方法2（了解即可）

编码间隔

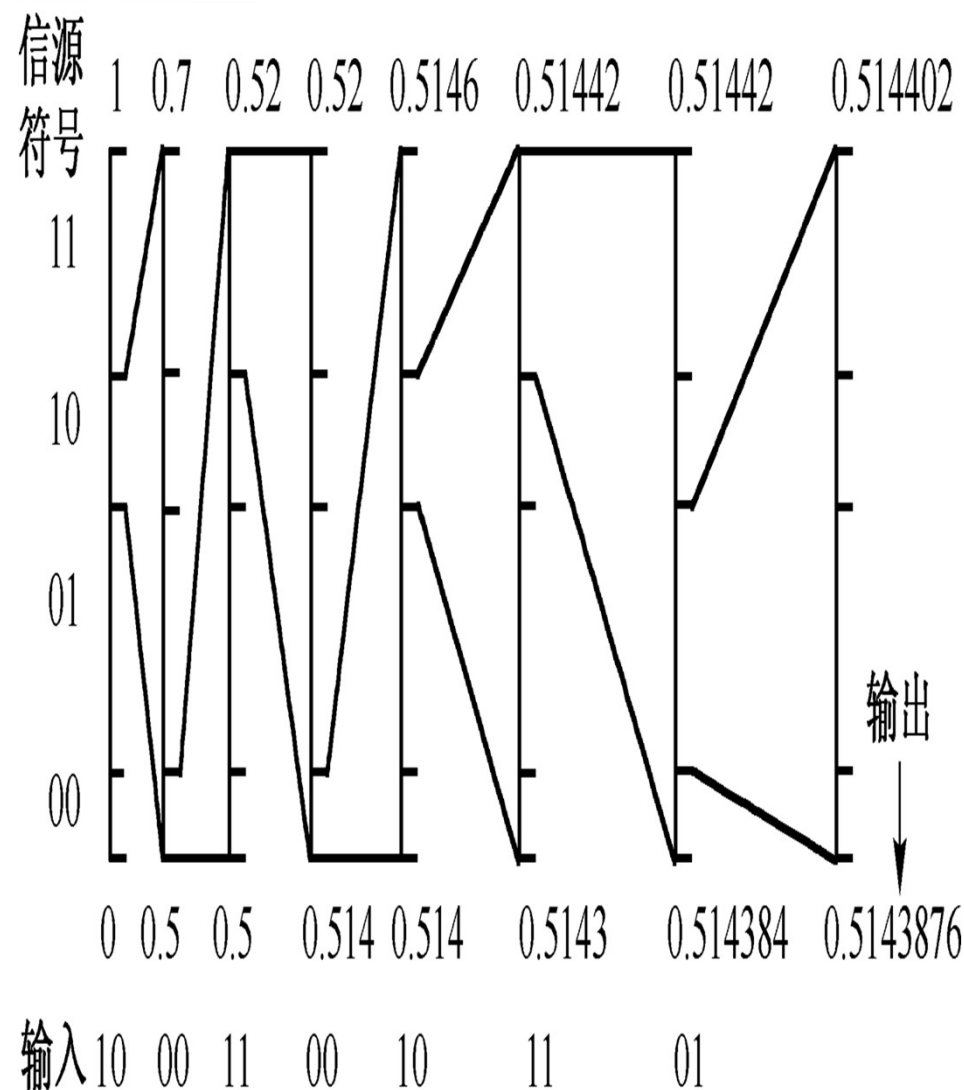
如果二进制消息序列的输入为10 00  
11 00 10 11 01, 则

(1) 首先输入的符号是**10**, 其编码范围是 **$[0.5, 0.7)$** , 即 **$L=0.5$** ,  
 **$H=0.7$** 。因此 **$C=0+1\times 0.5=0.5$** ,  
 **$A=1\times 0.2=0.2$** 。

(2) 由于消息中第二个符号**00**的编码范围是 **$[0, 0.1)$** , 因此  
 **$C=0.5+0.2\times 0=0.5$** ,  **$A=0.2\times 0.1=0.02$** 。

(3) 第3个符号**11**的编码范围是 **$[0.7, 1]$** , 因此 **$C=0.5+0.02\times 0.7=0.514$** ,  
 **$A=0.02\times 0.3=0.006$** 。

依此类推, 编码第4个符号00时,  
 **$C=0.514+0.006\times 0=0.514$** ,  
 **$A=0.006\times 0.1=0.0006$** , ...消息的编码输出可以是最后一个间隔中的任意数。





## 5.2.4 算术编码-方法1

### ■ 算术解码步骤如下：

- ① 由于0.5143876落在 $[0.5, 0.7)$ 区间内，可知第一个符号为C。

$$\frac{0.5143876 - 0}{1} = 0.5143876 \in [0.5, 0.7) \Rightarrow C$$

- ② 递推解码后序符号，直到结果为0结束。

$$\frac{0.5143876 - 0.5}{0.2} = 0.071938 \in [0, 0.1) \Rightarrow A$$

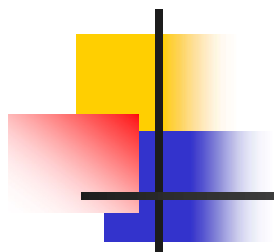
- ③ 以此类推：

$$\frac{0.071938 - 0}{0.1} = 0.71938 \in [0.7, 1) \Rightarrow D$$



## 5.2.4 算术编码

- 算术编码的主要特点有：
  - 1) 当信源符号的出现概率比较均匀时，**算术编码的效率比哈夫曼编码高**。
  - 2) 算术编码的实现比哈夫曼编码复杂。
- 算术编码是一种相对较新的编码，它在许多方面比哈夫曼编码优越：
  - 1) 算术编码按照分数比特逼近熵，而哈夫曼编码是按照整数比特逼近熵的；
  - 2) 算术编码可以有效地从模型中分离出来，而哈夫曼编码是与统计模型强相关的。



## 第5章 数字图像与视频压缩编码原理

- 5.1 数字图像与视频压缩编码概述
- 5.2 熵编码
- **5.3 预测编码**
- 5.4 变换编码
- 5.5 MATLAB编程实例



## 5.3 预测编码

- 从信息论观点来看，能够完全被预测（预测误差为**0**）的信号是不带任何信息的，因而不需要传送。
- 预测编码的基本原理就是利用图像数据的相关性，利用已编码传输的像素值对当前需要传输的像素值进行预测，然后对当前像素的实际值与预测值的差值（即预测误差）进行编码传输，而不是对当前像素值本身进行编码传输，以去除图像数据中的空间相关冗余或时间相关冗余。
- 在接收端，将收到的预测误差的码字解码后与预测值相加，得到当前像素值。



## 5.3 预测编码

---

- **预测编码**：根据某一**模型**，利用信号以往的样本值对新样本值进行预测，对**预测误差**进行编码。
- 对于**相关性较强**的信号，如果建立**合适的模型**，**预测误差**的幅值将远远小于原始信号，从而可以用较少的**量化级**对其误差信号进行**量化**，得到较大的数据压缩效果。



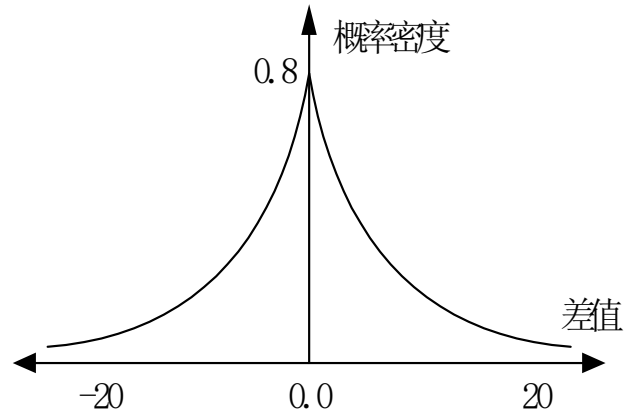
## 5.3 预测编码

- **帧内预测编码：**对于静止图像，由于相邻像素具有很强的相关性，这样当前像素的灰度（颜色）值可用前面已经出现的像素值进行预测，得到一个预测值，**对实际值与预测值的差值进行编码。**
- **帧间预测编码：**对于视频信号，图像帧间的相关性具有很强的相关性，通过帧间预测，对残差图像编码。
- **预测编码是当今主流技术并且还会流行于未来。**

## 5.3.1 图像差值信号的统计特性

**1) 帧内统计特性：**对一幅（帧）图像内部像素进行的统计分析。

对于常见的大多数图像，相邻（两行或两列）两个像素的差值的统计分布将集中在零附近。

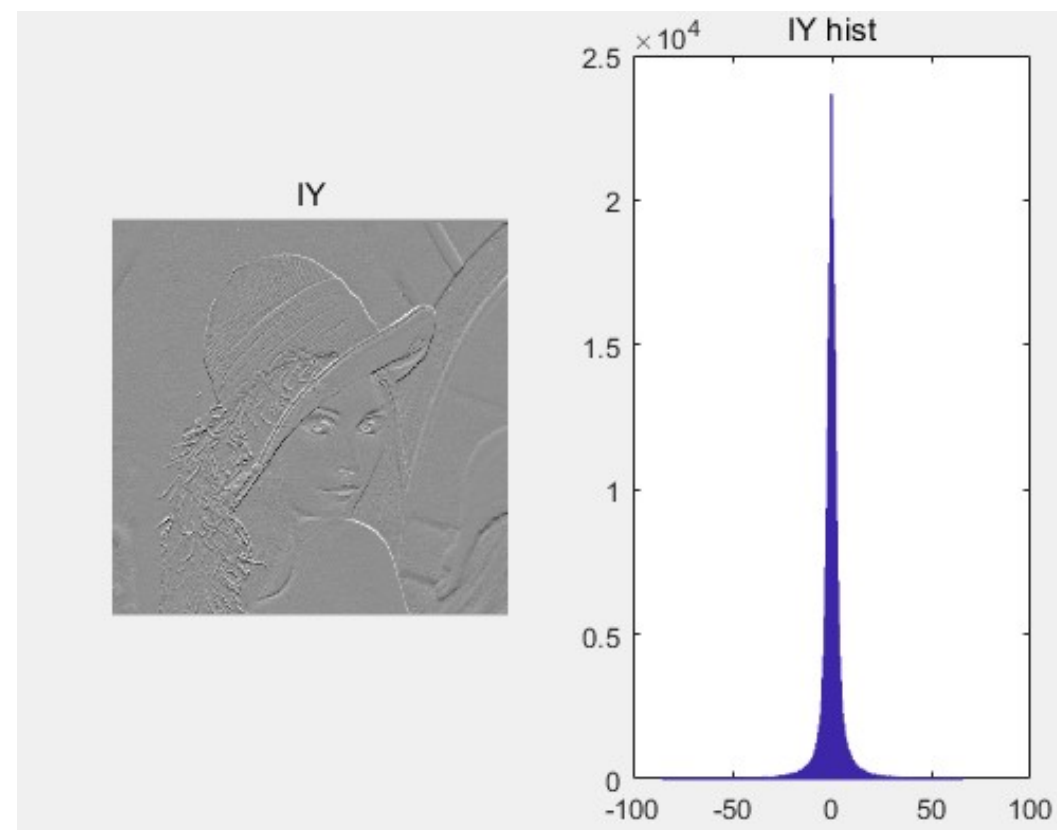
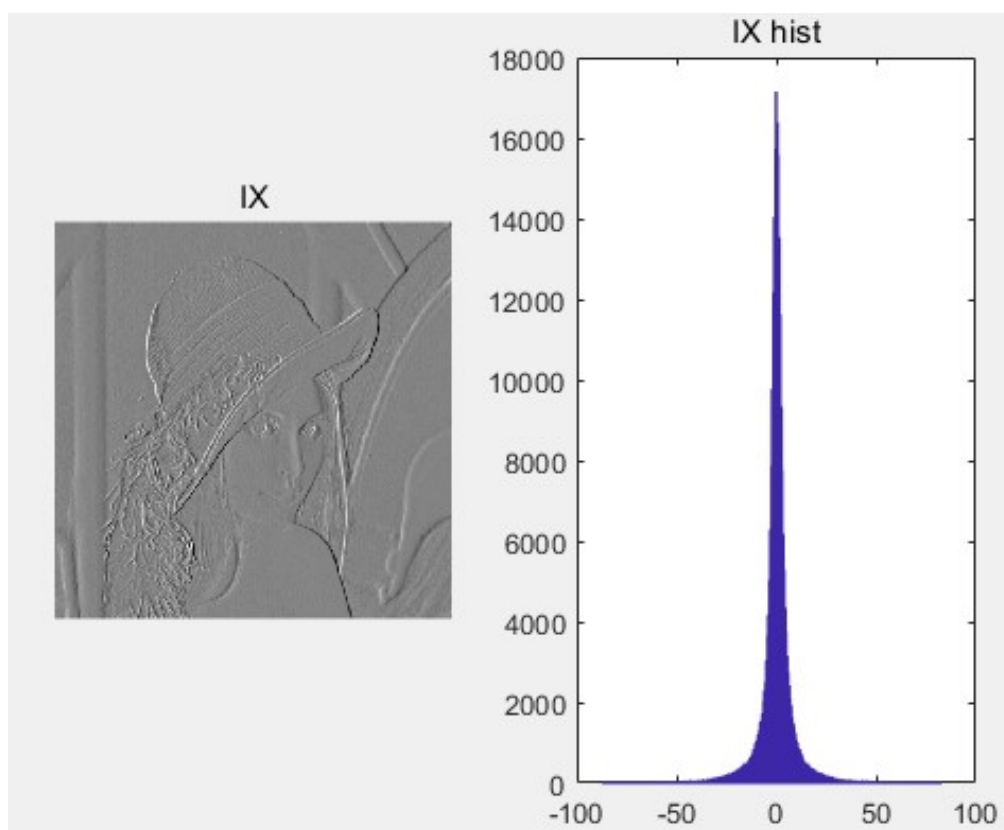


大量的图像数据统计表明，差值信号绝对值的80%~90%落在总数为256个量化层中的16~18个量化层范围内。



## 5.3.1 图像差值信号的统计特性

如Lena图像水平与垂直差分图像的统计特性

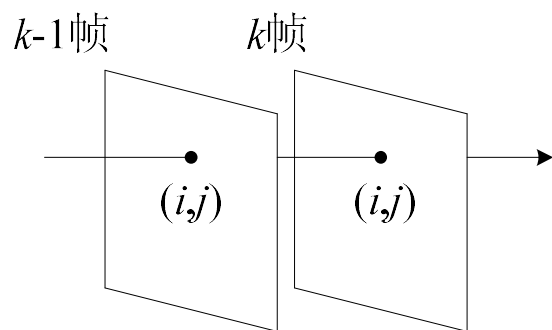


## 5.3.1 图像差值信号的统计特性

**2) 帧间统计特性：**对相邻帧对应位置的像素进行的统计分析。

对视频图像，相邻帧对应位置像素之间的时间间隔很小，很有可能表示的是场景中的同一点。

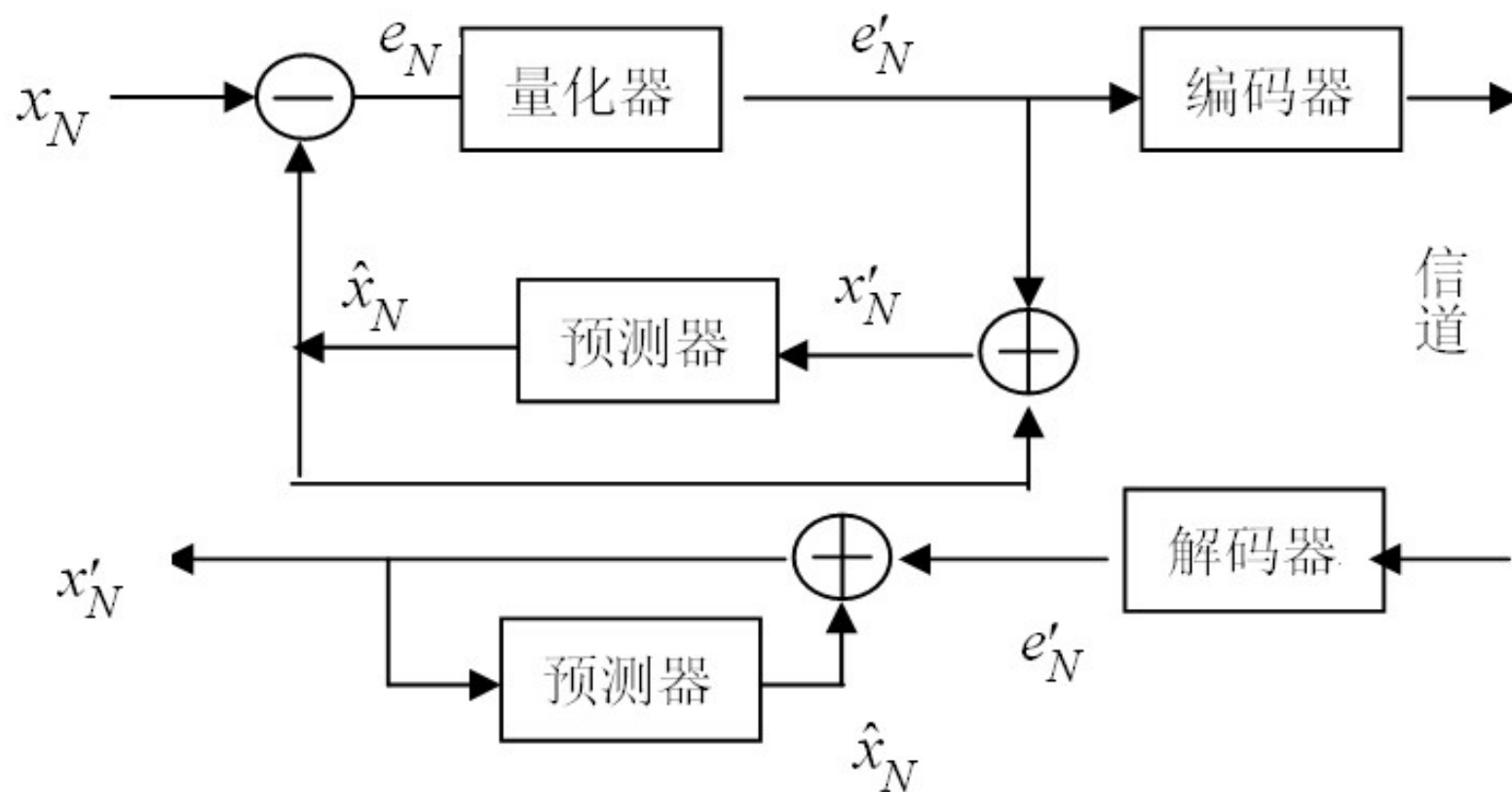
帧间差定义为： $d_k(i, j) = f_k(i, j) - f_{k-1}(i, j)$



## 5.3.2 帧内预测编码

### ■ 1. DPCM系统的基本原理

**DPCM**(Differential Pulse Code Modulation, 差分脉冲编码调制)



## 5.3.2 帧内预测编码

### ■ 2.预测模型

设  $t_N$  时刻之前的样本值  $x_1, x_2, \dots, x_{N-1}$  与预测值之间的关系呈现某种函数形式;

∞ 线性预测编码器

$$\hat{x}_N = \sum_{i=1}^{N-1} a_i x_i$$

∞ 非线性预测编码器



## 5.3.2 帧内预测编码

在图像数据压缩中，常用如下几种**线性预测**方案：

- ※ **1) 前值预测**，即  $\hat{x}_N = x_{N-1}$
- ※ **2) 一维预测**，即采用同一扫描行中前面已知的若干个样值来预测。
- ※ **3) 二维预测**，即不但用同一扫描行中的前面几个样值，而且还要用以前几行扫描行中样值来预测。



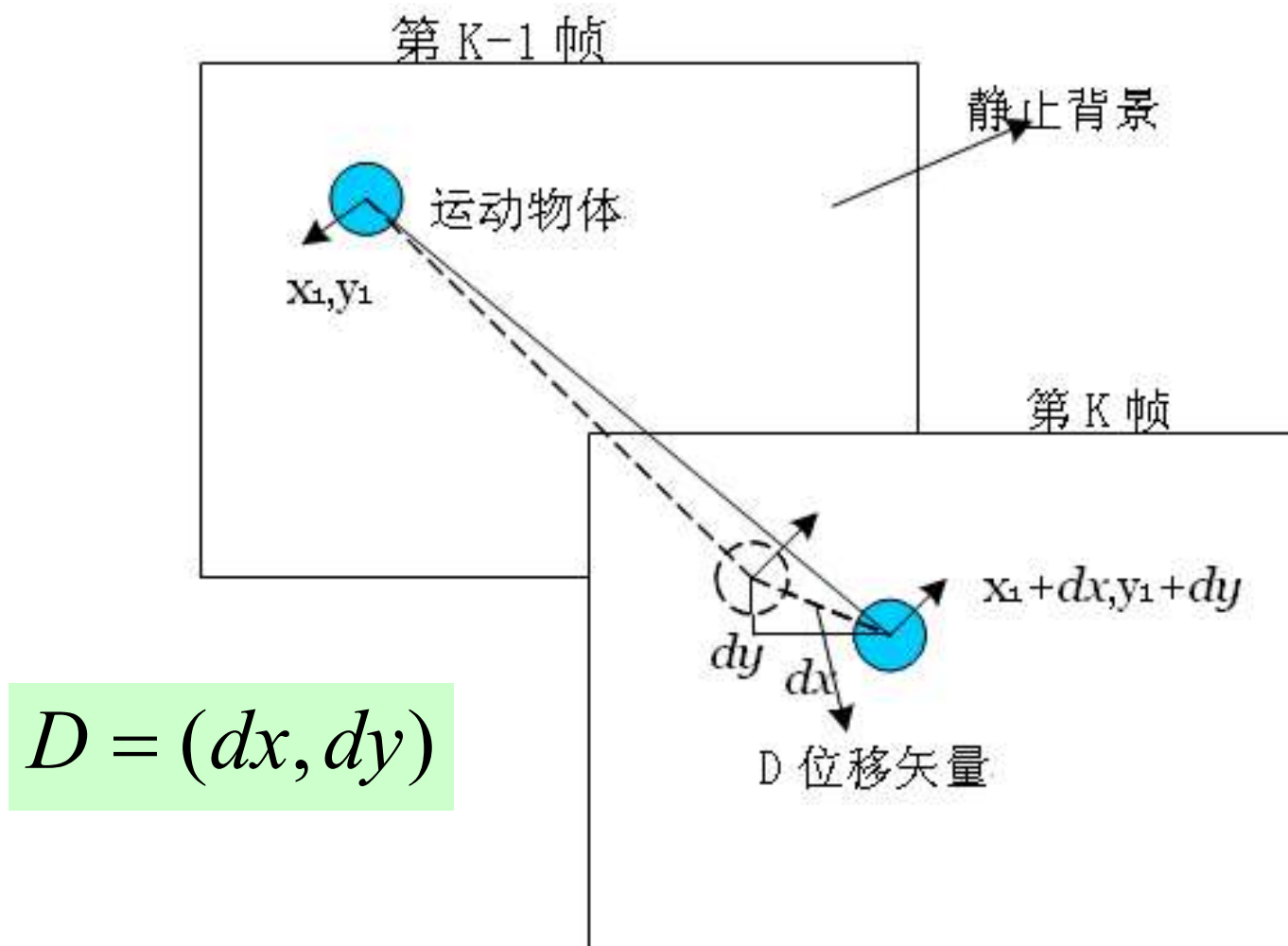
### 5.3.3 帧间预测编码

**序列图像在时间上的冗余情况可分为如下几种：**

- ❧ ①**对于静止不动的场景**，当前帧和前一帧的图像内容是完全相同的。
- ❧ ②**对于运动的物体**，只要知道其运动规律，就可以从前一帧图像推算出它在当前帧中的位置。
- ❧ ③**摄像机**对着场景的横向移动、焦距变化等操作会引起整个图像的平移、放大或缩小。对于这种情况，只要**摄像机的运动规律**和镜头改变的参数已知，图像随时间所产生的变化也是可以推算出来的。

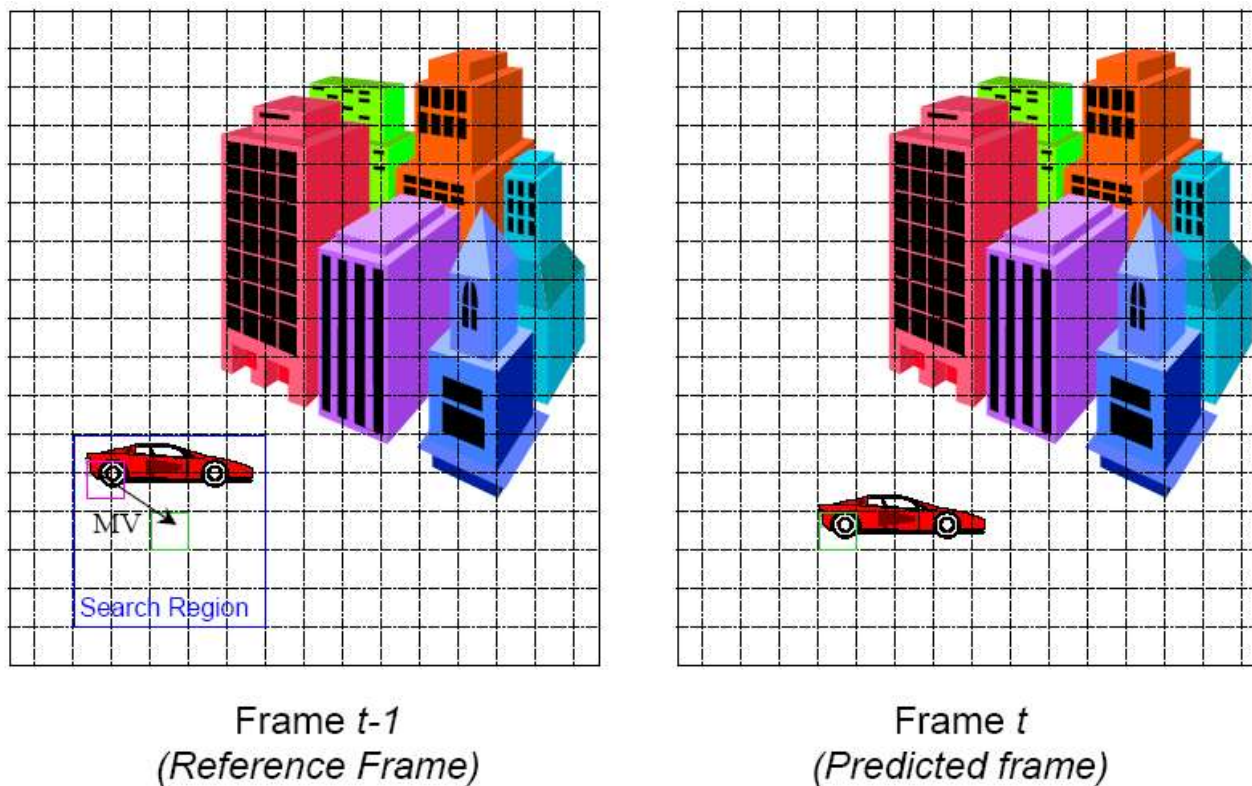
## 5.3.3 帧间预测编码

### 1) 运动补偿预测





### 5.3.3 帧间预测编码



◆ **运动估计 (ME)**，是对运动物体的位移进行估计，即对运动物体从前一帧到当前帧位移的大小和方向进行估计，得到MV。

◆ **运动补偿预测 (MC)**，根据求出的MV，找到当前帧的像素是从前一帧哪个位置移动过来的，得到当前像素的预测值。



## 5.3.3 帧间预测编码

---

### ■ 运动估计方法：

#### ➤ 1) 像素递归法：（目前较少使用）

- 根据像素间亮度的变化和梯度，通过递归修正的方法来估计每个像素的运动矢量。
- 接收端在与发送端同样的条件下，用与发送端相同的方法进行运动估值。
- 像素递归法估计精度高，可以满足运动补偿帧内插的要求。
- 但接收端较复杂，不利于一发多收（如数字电视广播等）的应用。



## 5.3.3 帧间预测编码

---

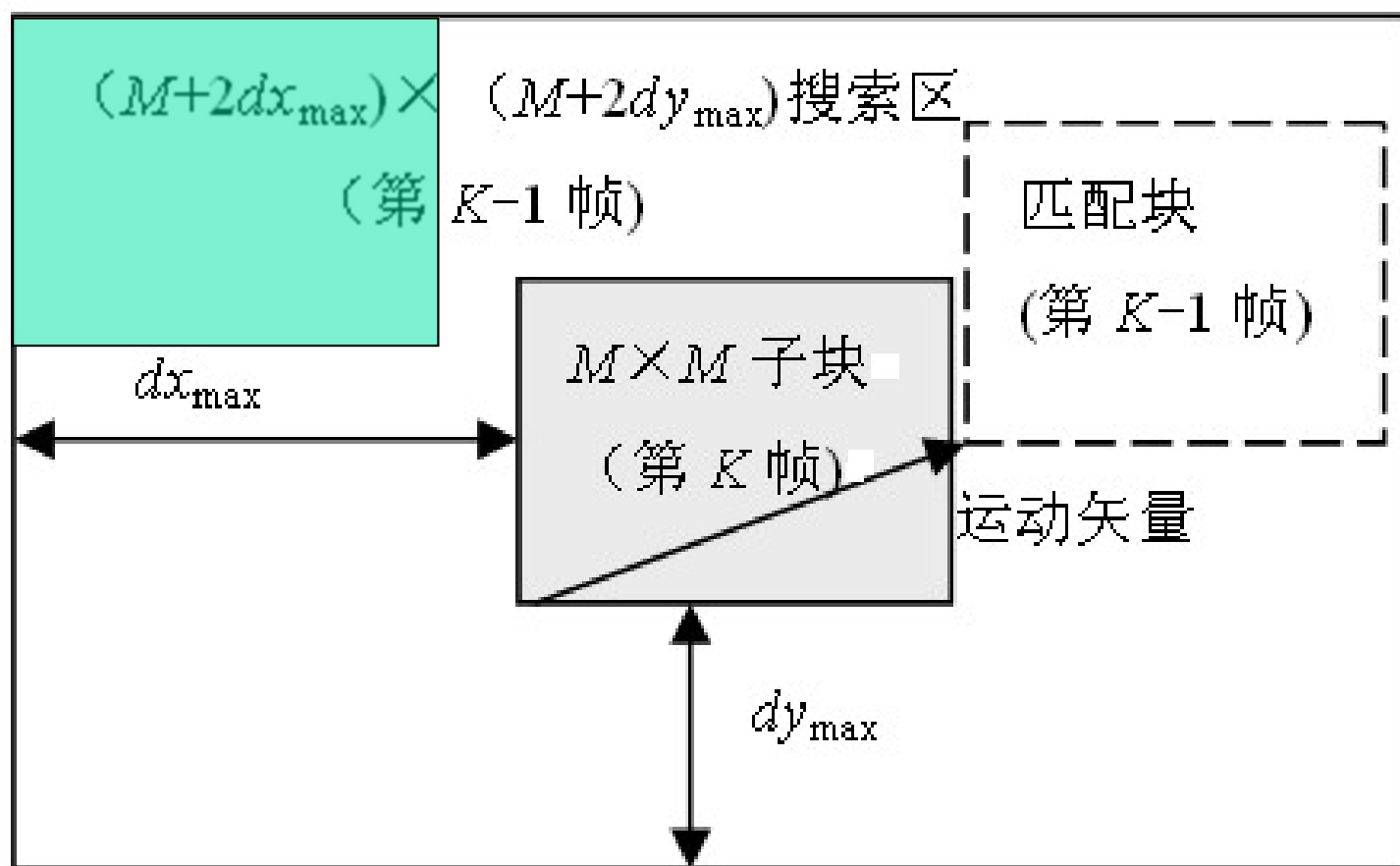
### ■ 运动估计方法：

#### ➤ 2) 块匹配算法：（目前常用技术）

- 块匹配算法对当前帧图像的每一子块，在上一帧（第K-1帧）的一定范围内搜索最优匹配，并认为本图像子块就是从上一帧最优匹配块位置处平移过来的。
- 块匹配算法虽然作了一定假设（假设位于同一图像子块内的所有像素都作相同的运动，且只作平移运动），但满足了计算复杂度和实时实现的要求。

### 5.3.3 帧间预测编码

#### ➤ 块匹配算法(BMA):





# 块匹配算法(BMA)

## ➤ 块大小对ME精度的影响

- **块大时**，一个方块可能包含多个作不同运动的物体，块内各像素作相同平移运动的假设难以成立，影响估计精度。
- **若块太小**，则估计精度容易受噪声干扰的影响，不够可靠，而且传送运动矢量所需的附加比特数过多，不利于数据压缩。
- 一般可用 $4\times 4$ 、 $8\times 8$ 、 $16\times 16$ 、 $32\times 32$ 、 $64\times 64$ 像素的块作为匹配单元。

# 块匹配算法(BMA)

➤ 最优匹配准则，决定什么样的子块是匹配的，即最相似的。

- 1) **绝对差均值** (MAD, Mean Absolute Difference) **最小准则**

$$MAD(i, j) = \frac{1}{MM} \sum_{m=1}^M \sum_{n=1}^M |S_K(m, n) - S_{K-1}(m+i, n+j)|$$

- 2) **均方误差** (MSE, Mean Squared Error) **最小准则**

$$MSE(i, j) = \frac{1}{MM} \sum_{m=1}^M \sum_{n=1}^M [S_K(m, n) - S_{K-1}(m+i, n+j)]^2$$

- 3) **归一化互相关函数最大准则**

$$NCCF(i, j) = \frac{\sum_{m=1}^M \sum_{n=1}^N f_k(m, n) f_{k-1}(m+i, n+j)}{\sqrt{\sum_{m=1}^M \sum_{n=1}^N f_k^2(m, n)} \cdot \sqrt{\sum_{m=1}^M \sum_{n=1}^N f_{k-1}^2(m+i, n+j)}}$$



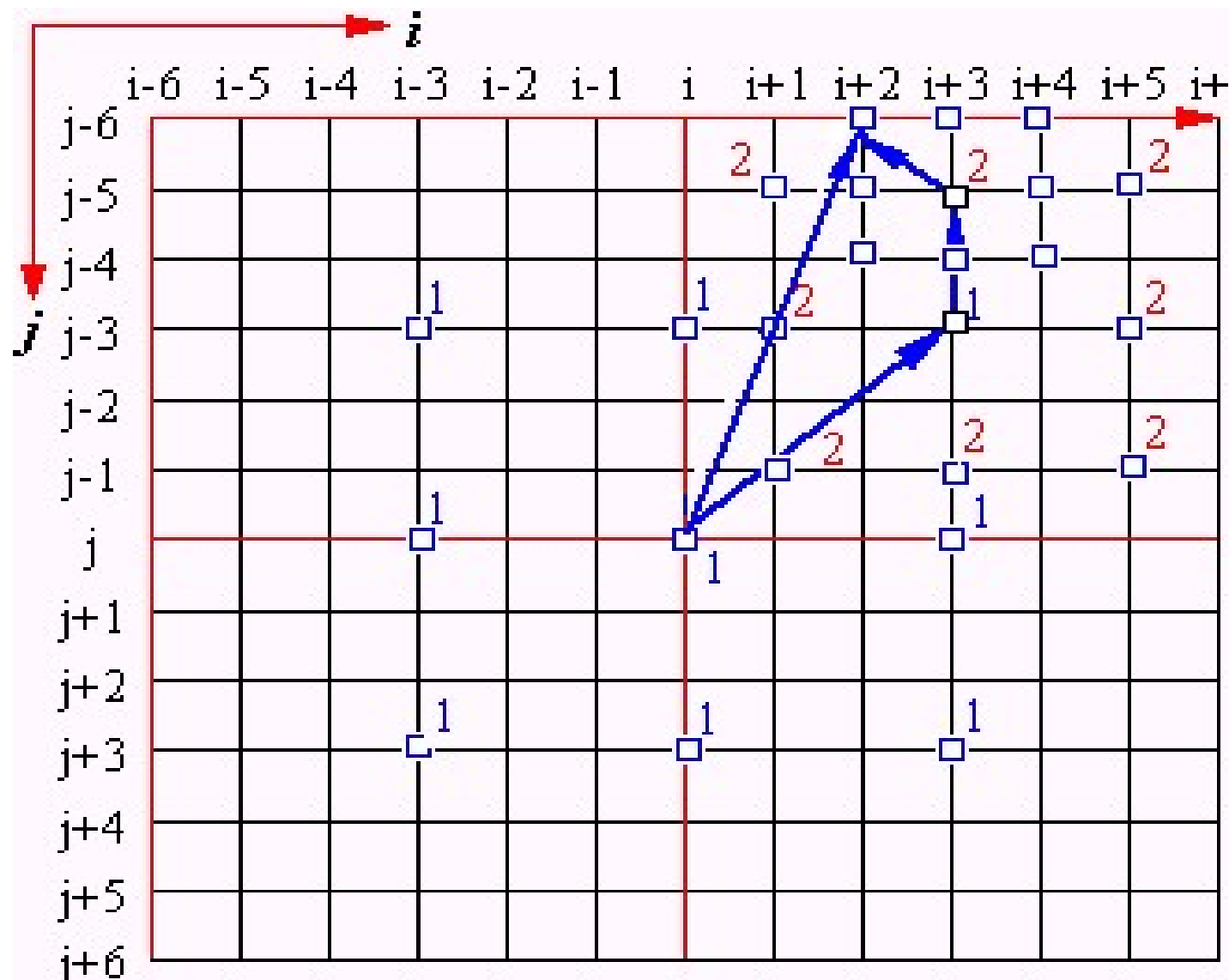
# 块匹配算法(BMA)

## ➤ 最优匹配点的搜索方法

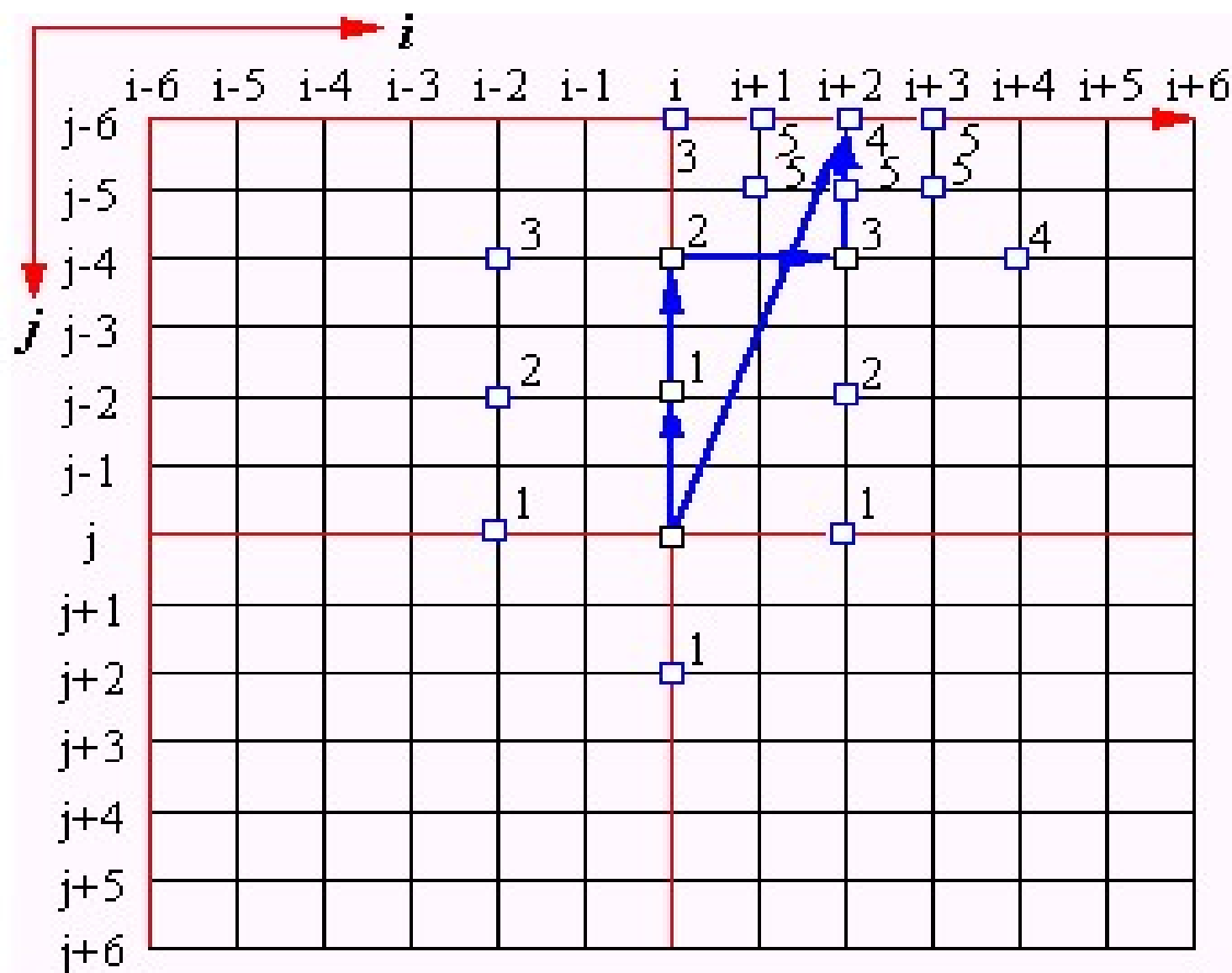
- 1) 穷尽搜索 (full search, 也称**全搜索**) 。
- 2) **快速搜索**: 其算法共同之处在于它们把使准则函数 (例如, MAD) 趋于极小的方向视同为最小失真方向, 并假定准则函数在偏离最小失真方向时是单调递增的, 即认为它在整个搜索区内是 $(i,j)$ 的单极点函数, 有唯一的极小值, 而快速搜索是从任一猜测点开始沿最小失真方向进行的。
- 3) **分级搜索**: 先通过对原始图像滤波和亚采样得到一个图像序列的低分辨率表示, 再对所得低分辨率图像进行全搜索。由于分辨率降低, 使得搜索次数成倍减少, 这一步可以称为**粗搜索**。然后, 再以低分辨率图像搜索的结果作为下一步**细搜索**的起始点。经过粗、细两级搜索, 便得到了最终的运动矢量估值。



## BMA常用搜索算法——三步搜索法：



## BMA常用搜索算法——二维对数搜索法:





# 第5章 数字图像与视频压缩编码原理

---

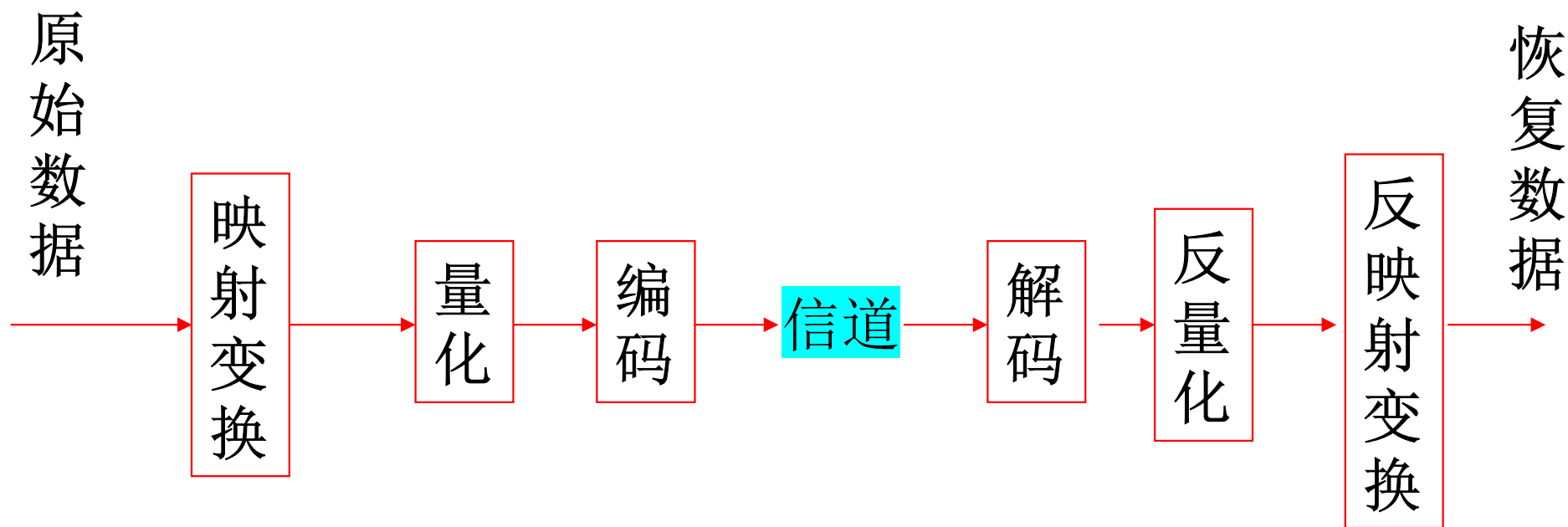
- 5.1 数字图像与视频压缩编码概述
- 5.2 熵编码
- 5.3 预测编码
- **5.4 变换编码**
- 5.5 MATLAB编程实例



## 5.4.1 变换编码的基本原理

- **预测编码** 希望通过对信源建模尽可能精确地预测数据，然后对预测误差进行编码，常在时间域或空间域进行。
- **变换编码的思路**：将原始数据从时间域或空间域“变换”到另一个更为紧凑表示、适合于压缩的**变换域（频域）**，从而得到比预测编码更高效率的数据表示（压缩）。

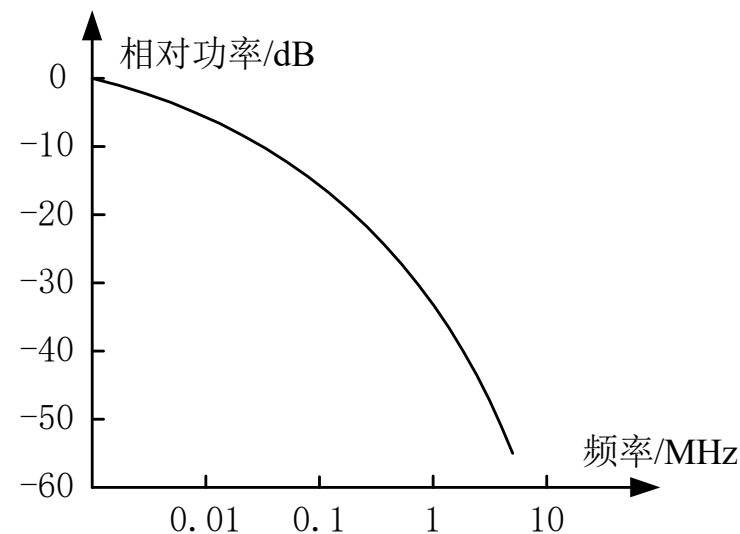
# 变换编码的通用模型



## 5.4.1 图像的频率域统计特性

- 在频率域，图像表现为不同频率分量系数的分布。
- 按照空间域与频率域的对应关系，**空间域的强相关性，即图像中的平坦区域，反应在频率域就是能量集中在低频部分**；即图像在频率域中呈现低通特性。

下图为对大量不同视频进行统计平均所得的实验结果。



从图中可以看出，电视信号的绝大部分能量集中于直流和低频部分。



## 5.4.2 变换编码的基本原理

- 变换编码不直接对空间域图像数据进行编码，而是首先将空间域图像数据映射变换到**另一个正交向量空间（变换域）**，得到一组变换系数，然后对这些**变换系数**进行量化和编码。
- 变换编码系统通常包括**正交变换**、**变换系数选择**和**量化编码**3个模块。
- 为了保证平稳性和相关性，同时也为了减少运算量，在变换编码中，一般在发送端的编码器中，**先将一帧图像划分成若干个 $N \times N$ 像素的图像块**，然后对每个图像块逐一进行变换编码，最后将各个图像块的编码比特流复合后再传输。在接收端，对收到的变换系数进行相应的逆变换，再恢复成图像数据。



## 5.4.2 变换编码的基本原理

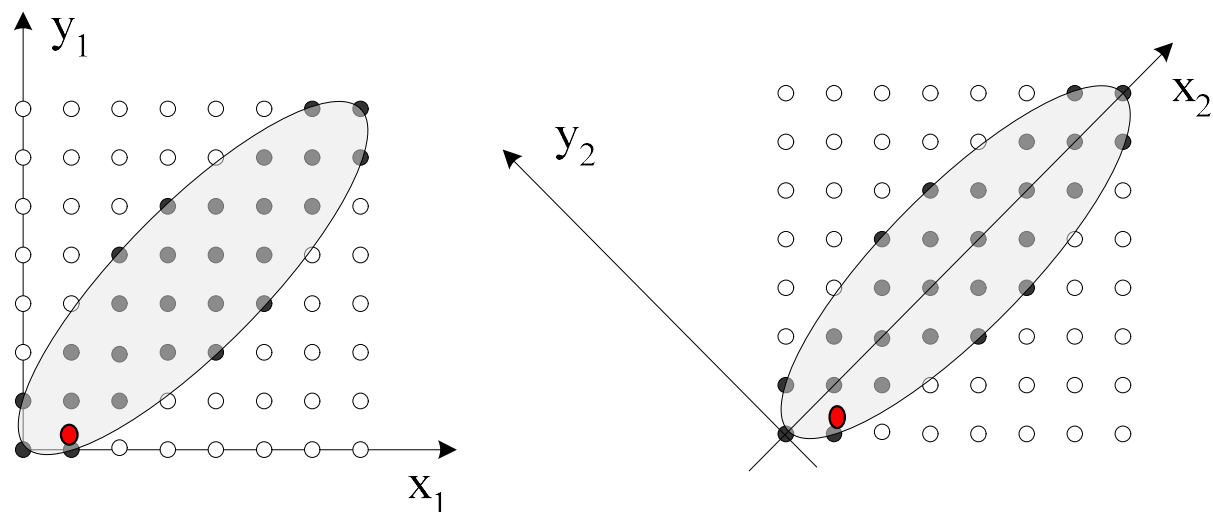
- **注意：正交变换本身并不能压缩数据，它只是把信号从一个域变换到另一个域，但由于变换后系数之间的相关性明显降低，为在变换域进行有效的压缩创造了条件。**
- 如图像块像素之间有很强的相关性，能量分布比较均匀；在正交变换后，变换系数间的相关性基本去除，近似独立，且图像的大部分能量集中在直流和少数低频系数上，通过选择保留其中一些对重建图像质量重要的变换系数，丢弃一些不重要的系数，再进行适当的量化和熵编码，就可以有效的压缩图像的数据量。



## 5.4.2 变换编码的基本原理

### ■ 正交变换的物理意义：

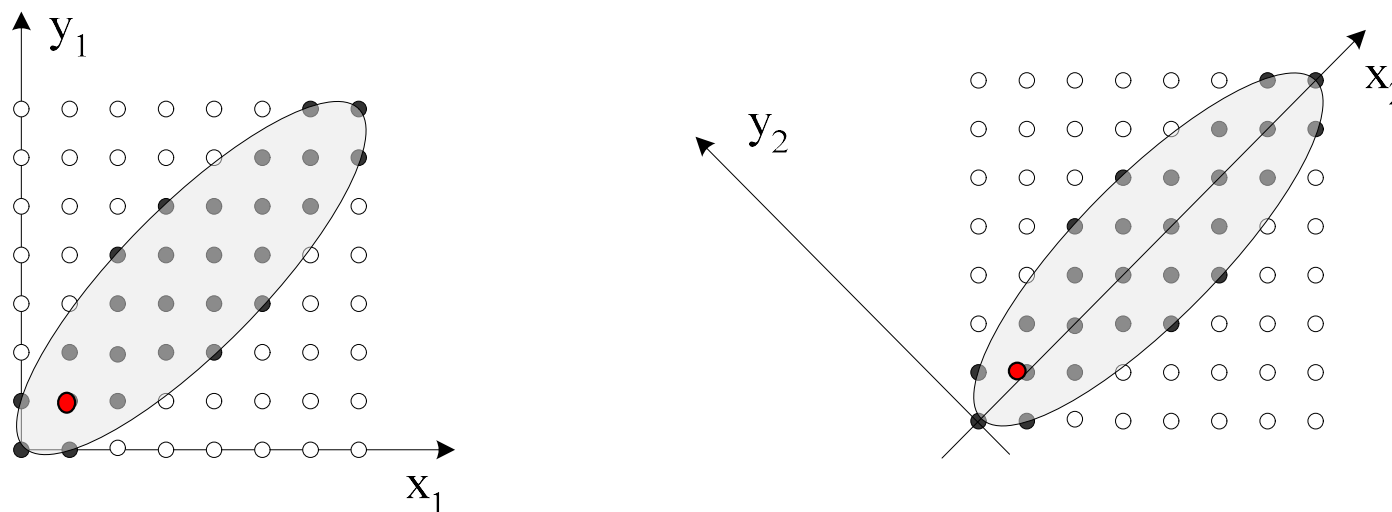
如最简单的1x2图像块  $(x_1, x_2)$ ，每个像素有8个灰度级，则此图像块共有  $8 \times 8 = 64$  种。



- 1) 由于图像相关性很强，即  $x_1, x_2$  的值很大概率是相似的，因此小图像在对角线处出现的概率很大。如  $(1, 1)$ 。
- 2) 正交变换后，相当于坐标轴旋转，同一像素在新坐标系下为  $(\sqrt{2}, 0)$ ，去掉了原像素值之间的相关性，现在只需要编码一个数即可。信号本身没有变化，但能量更为集中。

## 5.4.2 变换编码的基本原理

- 正交变换的物理意义：



图像正交变换实现数据压缩的**物理本质在于**：经过多维坐标系中适当的坐标旋转和变换，能够把接近均匀散布在各个坐标轴上的原始图像数据，变换到新的适当坐标系中，集中在少数坐标上，因此可用较少的编码比特来表示一副子图像，实现高效率的压缩编码。



### 5.4.3 正交变换基的选择

---

- 选择不同的正交基向量，可以得到不同的正交变换。  
常用的正交变换包括：
  - 1) 离散傅里叶变换 (DFT)
  - 2) 离散余弦变换 (DCT)
  - 3) Karhunen—Loeve变换 (K-L)
  - 4) 沃尔什-哈达玛变换 (WHT) 、
  - 5) 离散小波变换 (DWT)
  - .....

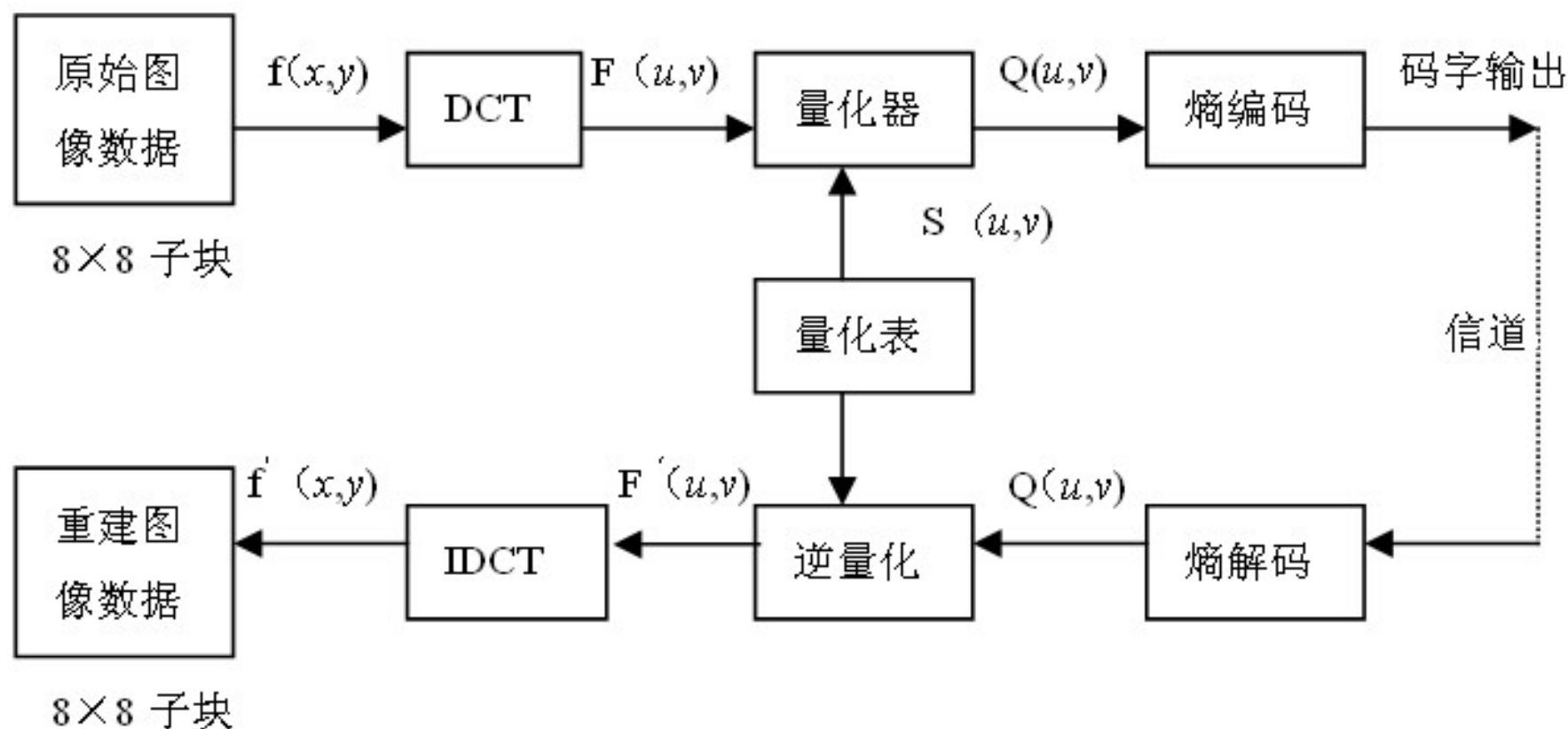


### 5.4.3 正交变换基的选择

- K-L变换能使变换后协方差矩阵为**对角阵**，并且有**最小均方误差（MSE）**，因此称为在MSE最小准则下的**最佳变换**。
- 由于K-L变换是取原图像各子块的**协方差矩阵的特征向量**作为变换基向量，因此K-L变换的变换基是**不固定**的，且与编码对象的统计特性有关，**没有快速算法**，计算复杂性高，使得K-L变换的应用不现实。
- 对大多数图像信源来说，DCT的性能最接近K-L变换，同时其变换基向量是固定的，且有快速算法，故DCT广泛应用于图像/视频压缩。

## 5.4.4 基于DCT的图像编码

### ■ 基于DCT的编码和解码原理





## 5.4.4 基于DCT的图像编码

### $8 \times 8$ 二维DCT变换

$$F(u, v) = \frac{1}{4} C(u) C(v) \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16}$$

### $8 \times 8$ 二维DCT反变换

$$f(x, y) = \frac{1}{4} \sum_{u=0}^7 \sum_{v=0}^7 C(u) C(v) F(u, v) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16}$$

当  $u = v = 0$  时,  $C(u) = C(v) = \frac{1}{\sqrt{2}}$

当  $u, v$  为其他值时  $C(u) = C(v) = 1$

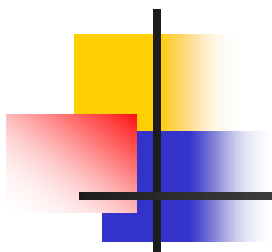


## 5.4.4 基于DCT的图像编码

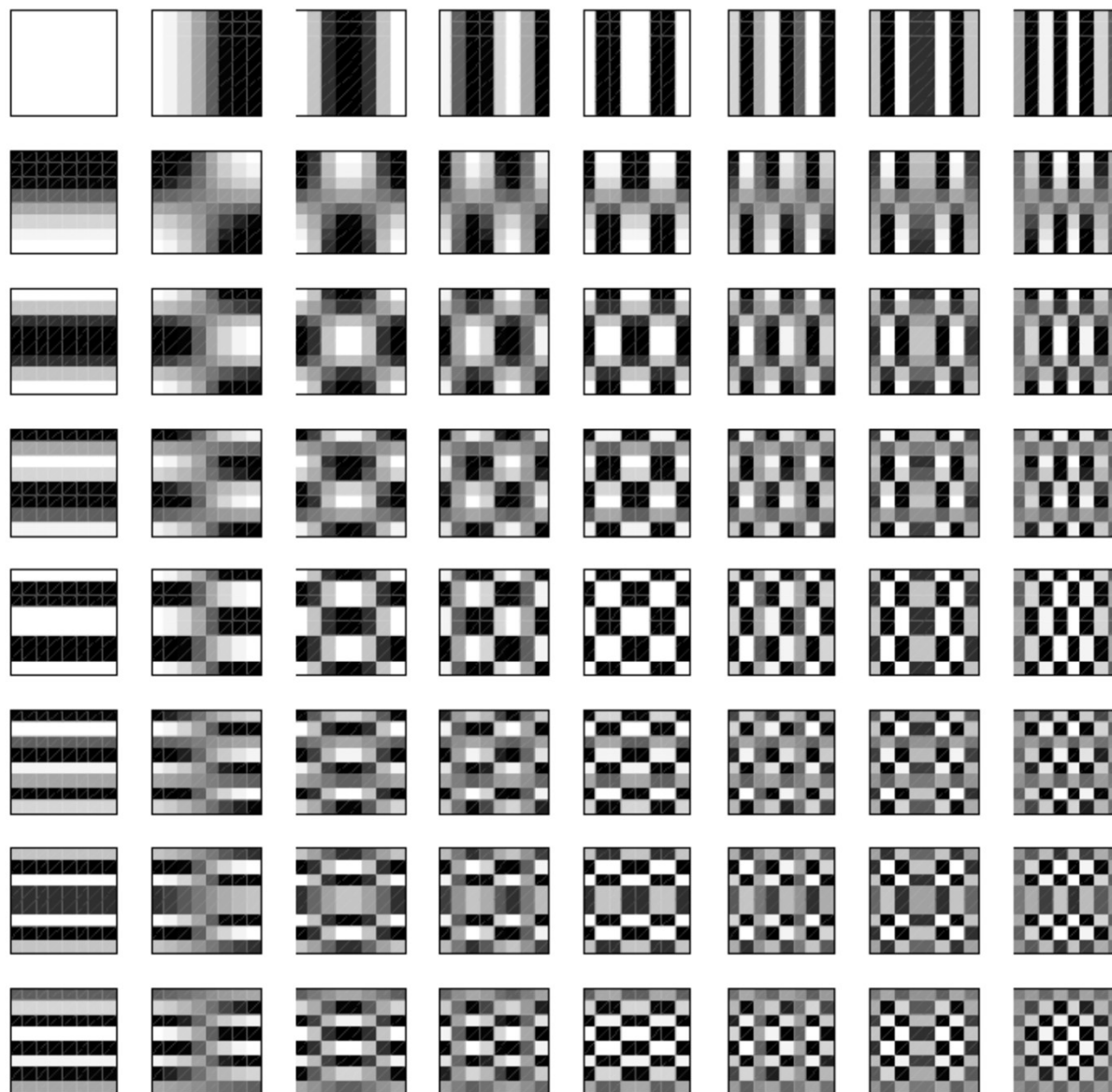
**$8 \times 8$ 二维DCT反变换的变换核函数为**

$$C(u)C(v)\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16}$$

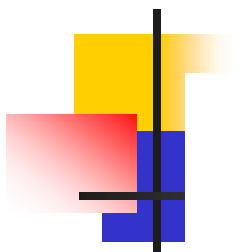
按 $u$ ， $v$ 分别展开后得到64个 $8 \times 8$ 像素的图像块组，称为**基图像**。



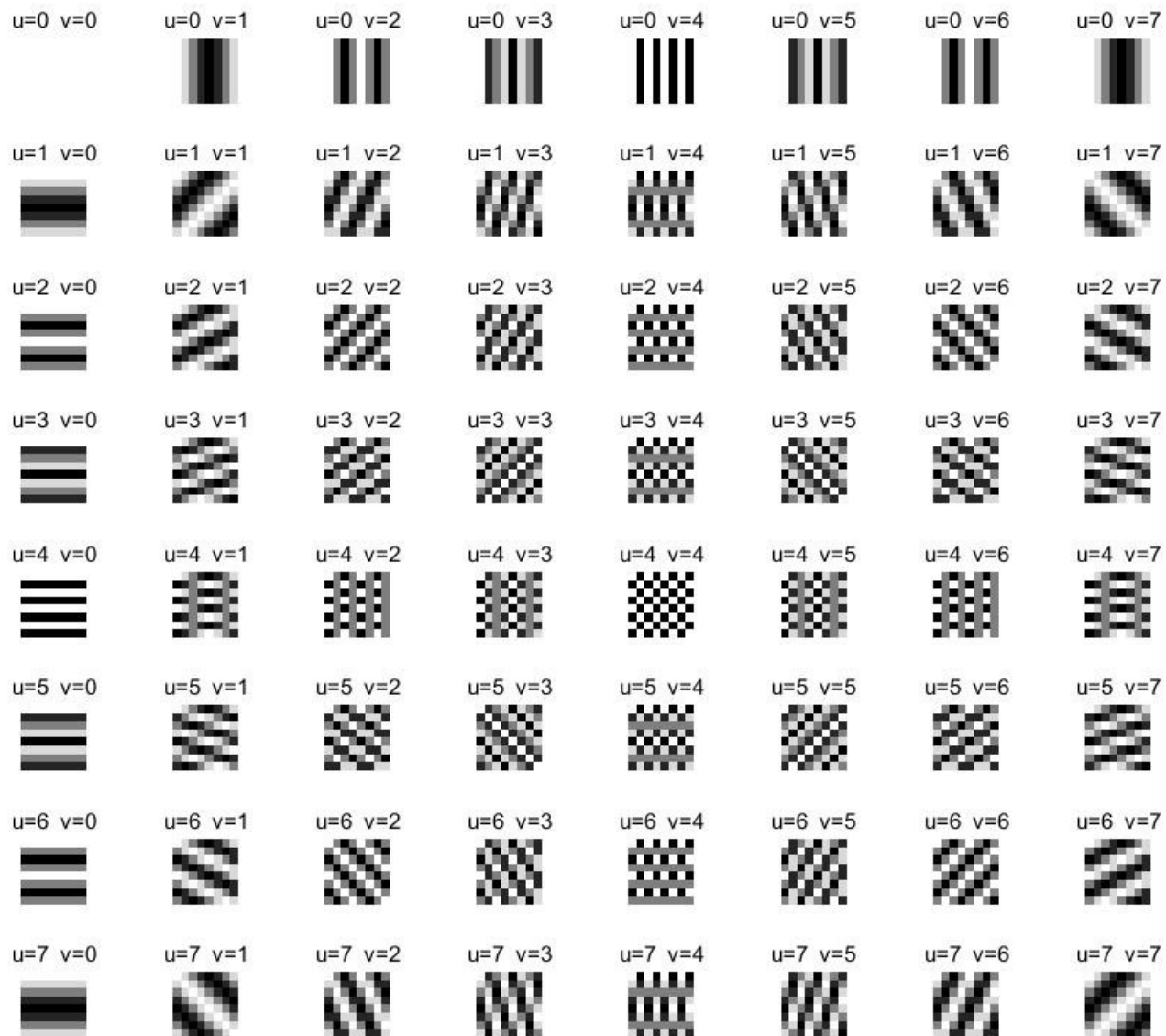
## $8 \times 8$ 二维DCT变换基图像







# 8 × 8二维DFT变换基图像

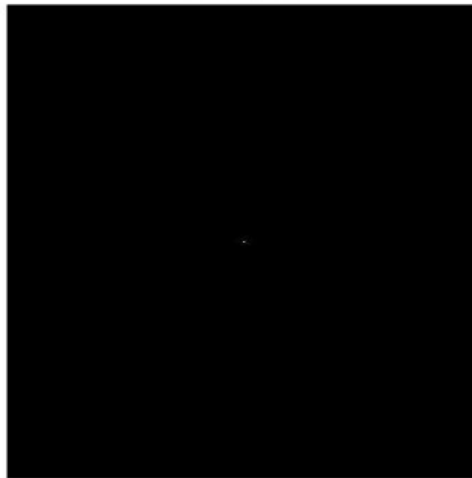


# Lena图像DFT频谱图

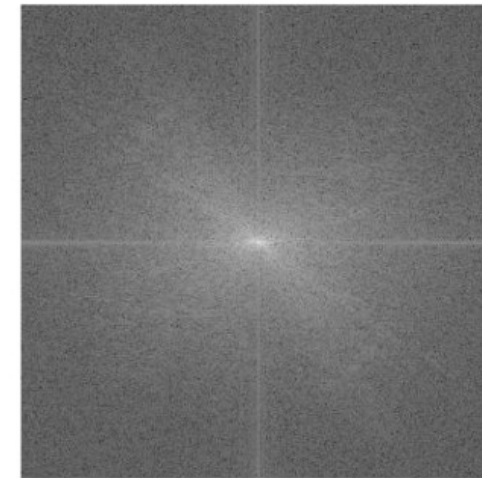
原始图像



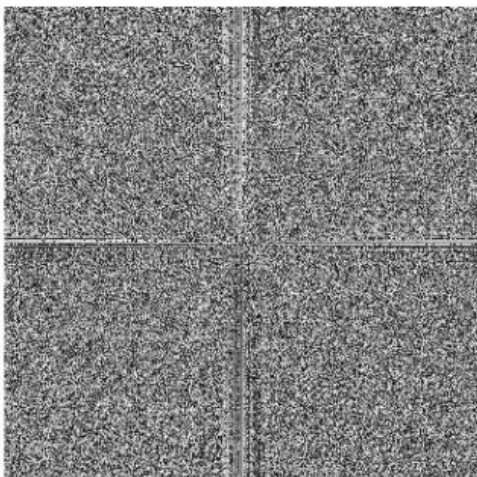
原始频谱图



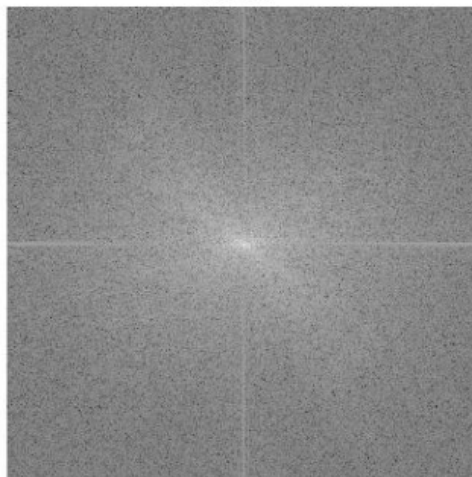
取对数后的频谱图



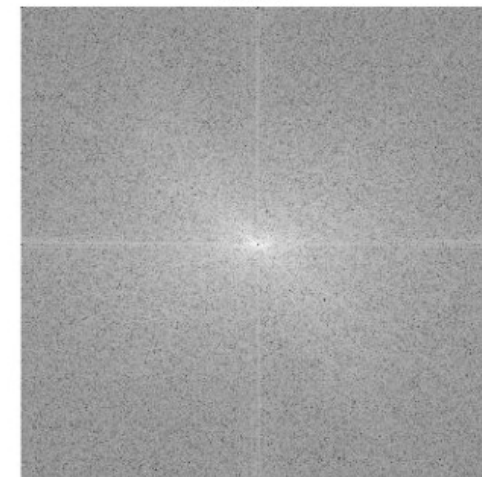
相位图



实部图



虚部图





# Lena图像DFT频谱图-彩色

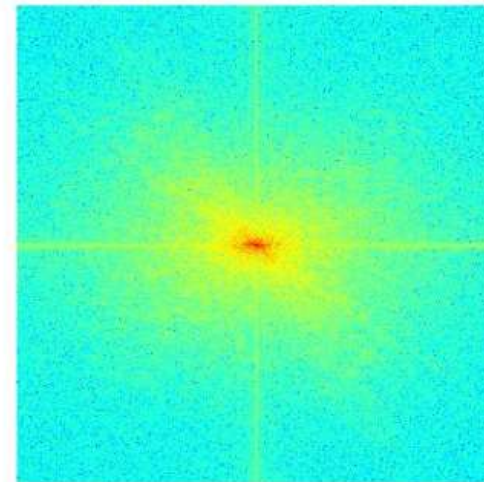
原始图像



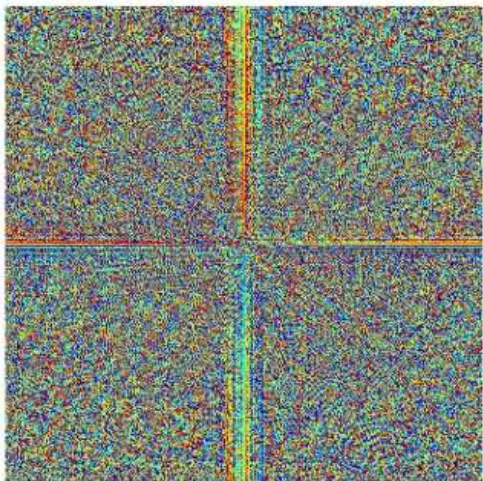
原始频谱图



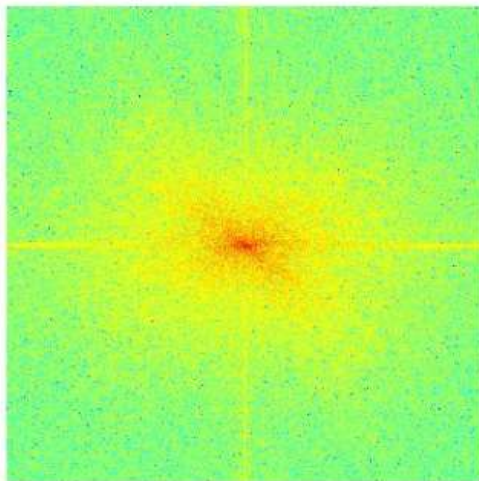
取对数后的频谱图



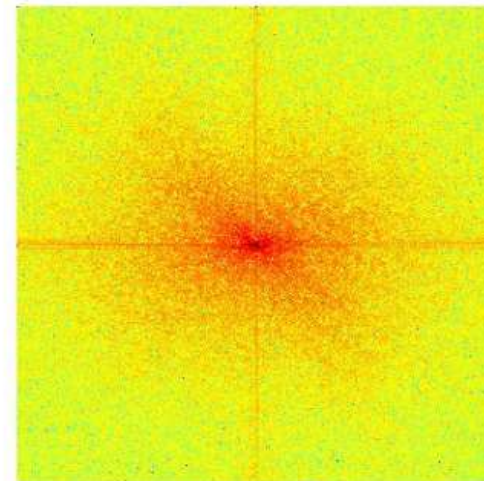
相位图



实部图



虚部图





## 5.4.4 基于DCT的图像编码

### 量化

量化处理是一个多到一的映射，它是造成DCT编解码信息损失的根源。

根据人眼的视觉特性，对不同的变换系数设置不同的量化步长。

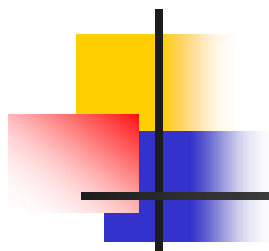
$$Q(u, v) = \text{round}\left[\frac{F(u, v)}{S(u, v)}\right]$$



## JPEG标准中亮度DCT系数的量化步长

表5-4 亮度量化表

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	112	100	103	99



## JPEG标准中色度DCT系数的量化步长

表5-5 色度量化表

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

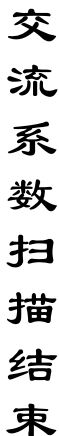


## 5.4.4 基于DCT的图像编码

### 量化步长的确定

- 通过大量实验并根据主观评价效果确定,其值随DCT系数的位置而改变。
- 量化表左上角及其附近区域的数值较大,而右下角及其附近区域的数值较小,且色度量化步长比亮度量化步长要大,符合人眼视觉特性。
- 即人眼对高频分量不太敏感,且对色度信号的敏感度比对亮度信号的敏感度低。

- 







## 变换系数熵编码

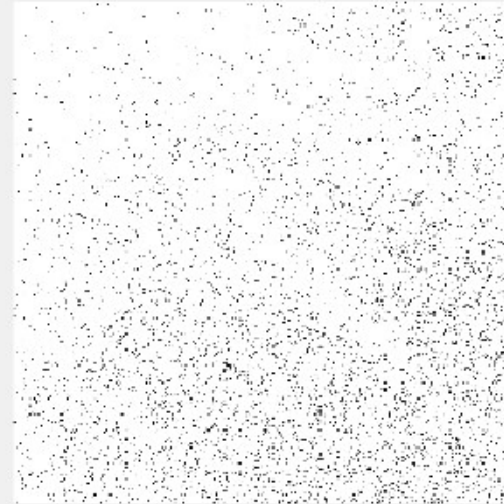
- ◆ **直流分量 (DC)** : 相邻图像子块的直流分量 (图像子块的平均样值) 也存在着相关性, 所以对DC的量化系数用**DPCM**编码较合适, 即对当前块和前一块的DC系数的差值进行编码。
- ◆ **交流分量 (AC)** : 把数值为0的连续长度 (即0游长) 和非0值结合起来构成一个事件 (**Run, Level**) , 然后再对事件 (**Run, Level**) 进行熵编码。

## 例5-6图像压缩示例

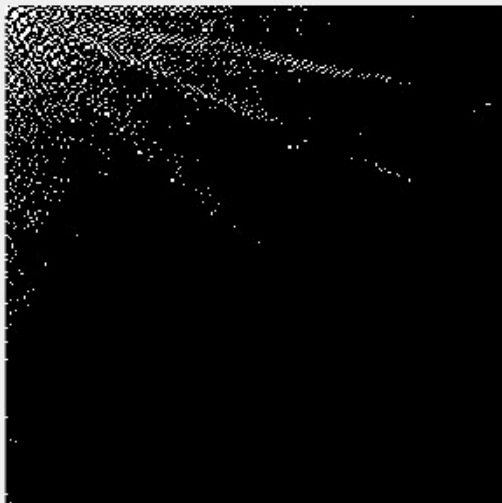
Original Image



DCT of Image



DCT Coefficients thresholded at 50



IDCT of Image after thresholding smaller co-efficients



## 例5-6图像压缩示例



lena



lena\_dct\_compressed



error\_map



## 5.7 习题

1. 为什么要对图像数据进行压缩？其基本原理是什么？图像压缩编码的目的是什么？目前有哪些编码方法？
2. 一个信源包含**6**个符号消息，它们的出现概率分别为**0.3、0.2、0.15、0.15、0.1、0.1**，请对该信源进行哈夫曼编码，并求出码字的平均长度和编码效率。
3. 设有一个信源具有**4**个可能出现的符号 **$X_1$ 、 $X_2$ 、 $X_3$ 、 $X_4$** ，其出现的概率分别为 **$1/2$ 、 $1/4$ 、 $1/8$ 、 $1/8$** 。请以符号序列 **$X_2X_1X_4X_3X_1$** 为例解释其算术编码的过程。
4. 请比较算术编码和哈夫曼编码的特点。
5. 请说明预测编码的原理，并画出**DPCM**编解码器的原理框图。
6. 预测编码是无损编码还是有损编码？为什么？
7. **DCT**本身能不能压缩数据？为什么？请说明**DCT**变换编码的原理。
8. 目前最常用的运动估计技术是什么？其假设的前提条件是什么？块大小的选择与运动矢量场的一致性是如何考虑的？