

数字图像处理

第4章 图像编码

信息科学研究所

5.6 变换编码

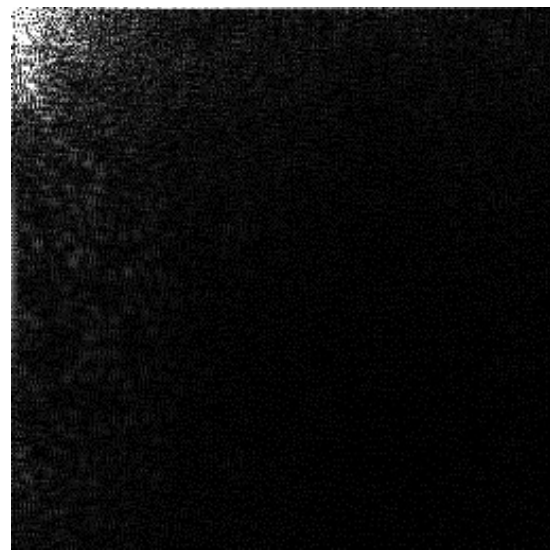
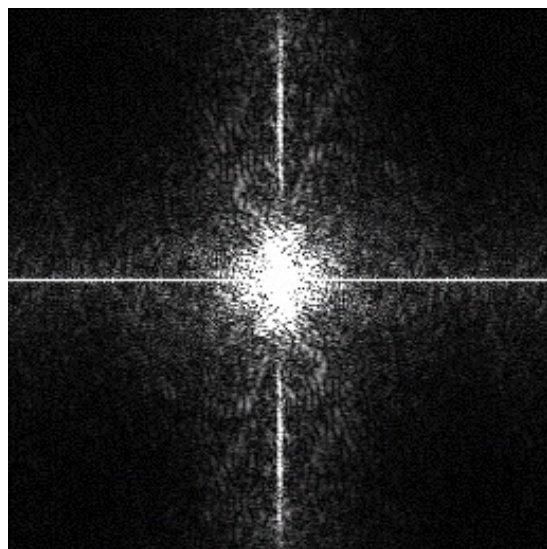
图像编码中另一类有效的方法是变换编码。变换编码的通用模型如图所示：



图5—42 图像变换编码模型

变换编码主要由映射变换、量化及编码几部分操作组成。映射变换是把图像中的各个像素从一种空间变换到另一种空间，然后针对变换后的信号进行量化与编码操作。在接收端，首先对接收到的信号进行译码，然后再进行反变换以恢复原图像。

映射变换的关键在于能够产生一系列更加有效的系数，对这些系数进行编码所需的总比特数比对原始图像进行编码所需要的总比特数要少得多，从而使数据得以压缩。



映射变换的方法很多。广义地讲，前面讨论的预测编码法也可称为是预测变换。它是将信号样值的绝对值映射为相对样元的差值，只是根据实用技术上的习惯，没有把它归入变换编码的范畴。

图像变换编码基本可分为两大类:

特殊映射变换编码法

函数变换编码法

5.6.1 几种特殊的映射变换编码法

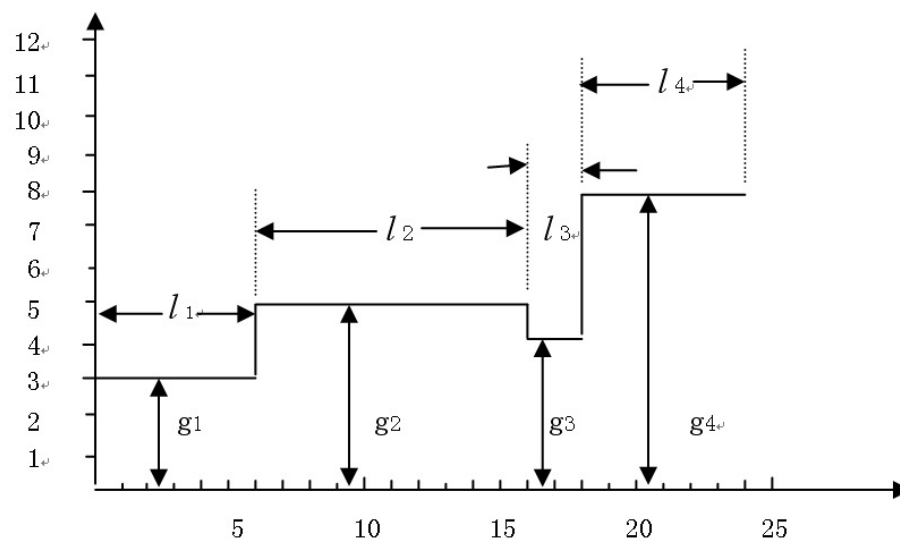
特殊映射变换编码法包括行程编码，轮廓编码等。它们特别适用于二值图像的编码。这类图像包括业务信件、公文、气象图、工程图、地图、指纹卡片及新闻报纸等。当然在编码技术上同样可分为精确编码和近似编码两类。

精确编码可以不引入任何畸变，在接收端可以从编码比特流中精确恢复出原始图像。近似编码会引入畸变，但这种方法却可以在保证可用性的前提下获得较高的压缩比。下面通过具体例子说明这种变换编码法的基本概念。

1、一维行程编码

一维行程编码的概念

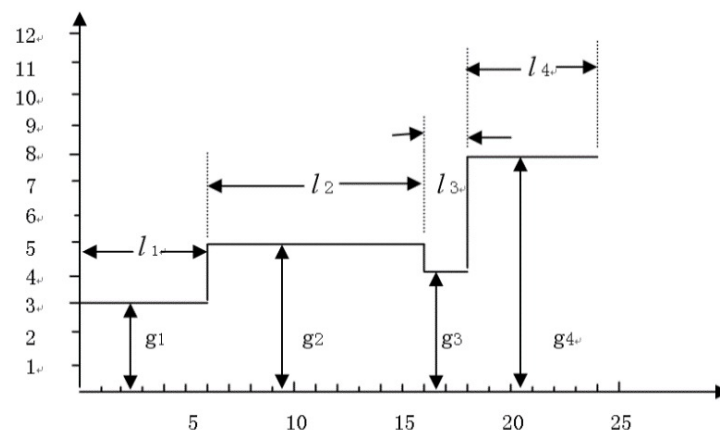
如图所示：



假如沿某一扫描行的像素为 $x_1, x_2, x_3, \dots, x_N$ ，它们所具有的灰度值可能为 g_1, g_2, g_3, g_4 。在编码之前，可以首先把这些像素映射为成对序列 (g_1, l_1) , (g_2, l_2) , (g_3, l_3) 和 (g_4, l_4) 。

有些图像，如二值图像，连续取同一灰度级的像素很多，对映射后的序列进行编码会大大压缩比特率。

图5—42所示的例子可映射成表5—14所示的序列对。在这个例子中有8级灰度，24个像素。如果对 x_i 编码，总的比特数至少要 $24 \times 3 = 72\text{bit}$ 。



如果对表5—14的序列对编码，灰度值用3位码，行程长度用4位码，每对参数用7位码，共4对，总比特数只要28bit就够了。可见压缩率是很可观的。

表 5—14 序列对

i	g_i	l_i
1	3	6
2	5	10
3	4	2
4	8	6

行程编码可分为行程终点编码和行程长度编码。如果行程终点由扫描行的开始点算起，并且由到达行程终点的像素个数来确定，就称为行程终点编码。如果行程终点由这一终点与前一终点的相对距离确定，就称为行程长度编码。

对于二值图像来说采用行程长度编码，甚至不需要传送灰度信息。假定某一扫描线含有3个白色像素，其后是2个黑色像素，接着又是10个白色像素。这样，在行程长度编码中，只传送行程长度3、2和10就可以了。每个行程长度告诉沿扫描线的下一个边界点的相对位置。

2、二维行程编码

二维行程编码也叫预测微分量化器(**Predictive Differential Quantizer, PDQ**)。其基本算法如图5—43所示。

“消失”

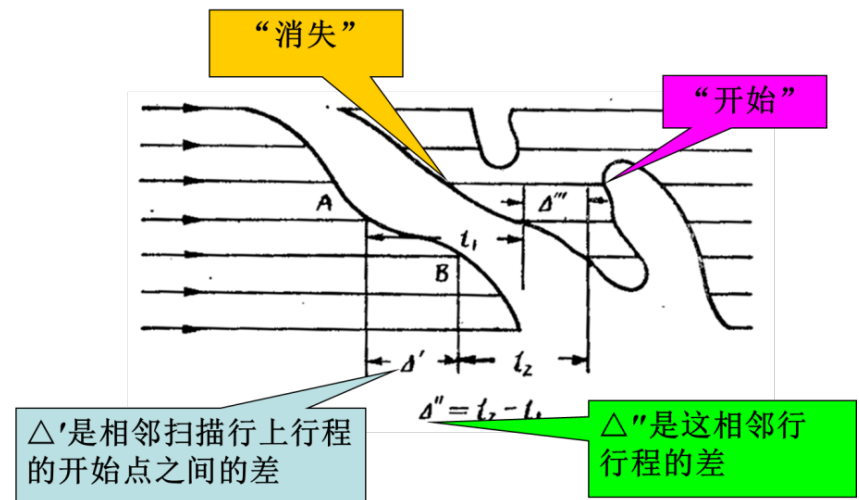
“开始”



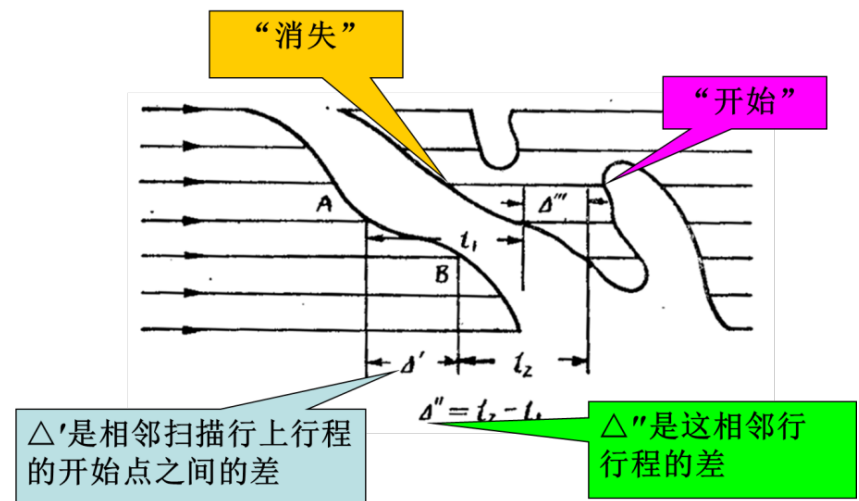
Δ' 是相邻扫描行上行程的开始点之间的差

Δ'' 是这相邻行程的差

预测微分量化器是将图像元素阵列变换为整数对 Δ' 和 Δ'' 的序列。 Δ' 是相邻扫描行上行程的开始点之间的差。图中 Δ' 是 A 点和 B 点的差。 Δ'' 是这相邻行行程的差。对应于 A 起始点的行程为 l_1 ，对应于 B 起点的行程为 l_2 ，因此， $\Delta'' = l_2 - l_1$ 。



对于图中的暗面积还要有一个“开始”和“消失”的标记。这样就**把一幅图像的像素阵列按相继扫描行变换为 Δ' 、 Δ'' 、开始、消失四个参量的序列**，然后便可对这四个参量来编码。预测微分量化器利用了扫描线间的相关性，有更大的压缩潜力。

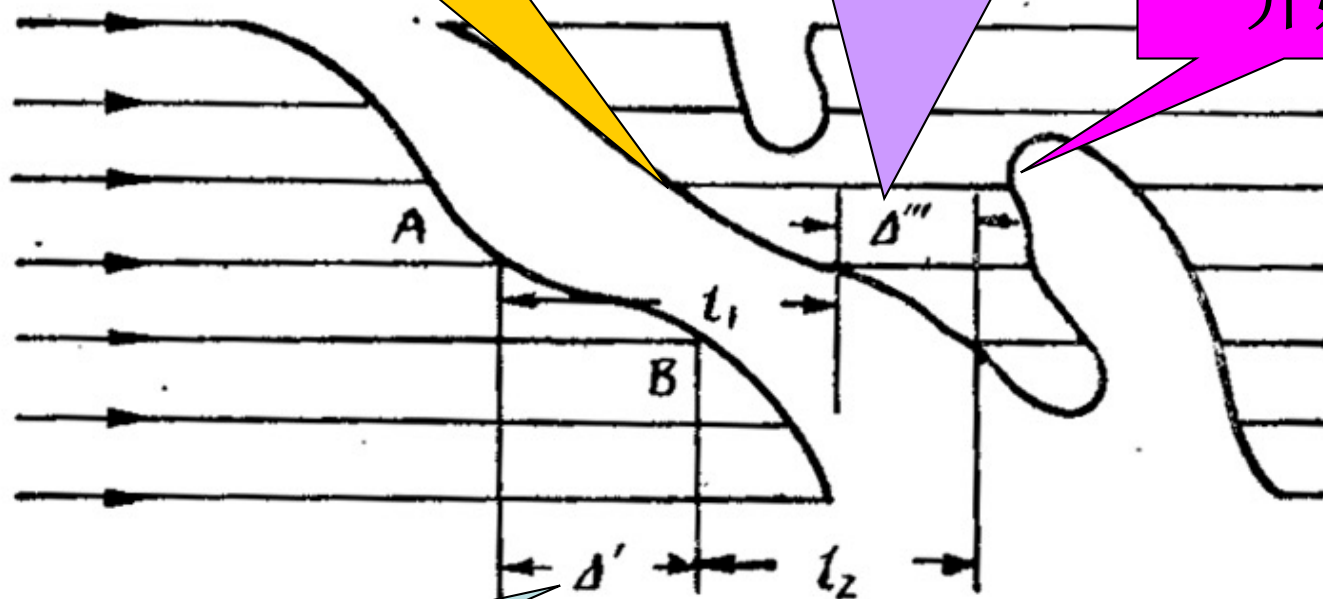


另外一种方法叫做双重增量编码(**Double Delta coding, DDC**)。它是对 Δ' 和 Δ'' 进行编码而不是对 Δ' 和 Δ'' 编码。 Δ'' 是前一扫描行的暗区后边界与相继扫描行暗区后边界的差。实验证明, 这种方法的压缩比预测微分量化器更大。

“消失”

Δ''' 是前一扫描行的暗区后边界与相继扫描行暗区后边界的差

“开始”



Δ' 是相邻扫描行上行程的开始点之间的差

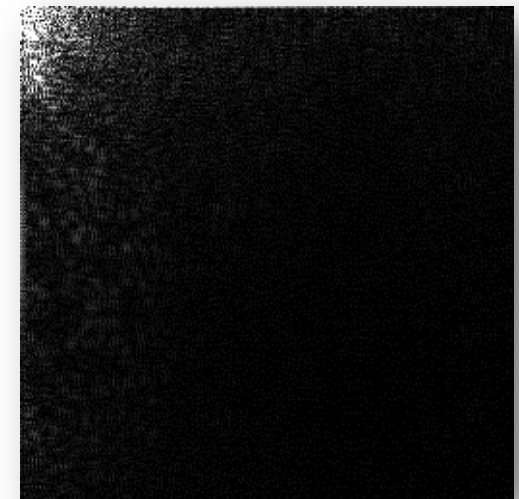
$$\Delta'' = l_2 - l_1$$

当图像中有少数大的暗区时二维行程编码更有效，
对于有许多小暗区的图像来说，一维行程编码更有效。



6.2 正交变换编码

变换编码中另一类方法是正交变换编码（或称函数变换编码），其基本原理是通过正交变换把图像从空间域转换为能量比较集中的变换域，然后对变换系数进行编码，从而达到缩减比特率的目的。



6.2.1 正交变换编码的基本概念

正交变换编码的基本原理如图所示。编码器由预处理、正交变换、量化与编码几部分组成，译码器由译码、反变换及后处理组成。

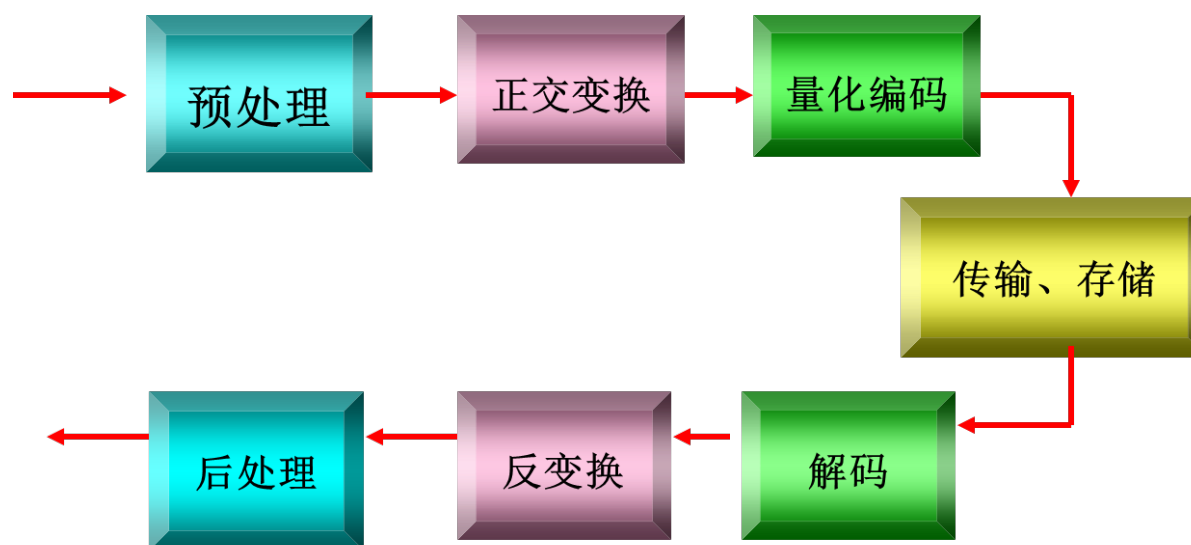


图5—50 正交变换编码原理框图

在编码操作中，模拟图像信号首先送入预处理器，将模拟信号变为数字信号。然后把**数字信号分块进行正交变换**，通过正交变换使空间域信号变换到变换域，然后对变换系数进行量化和编码。

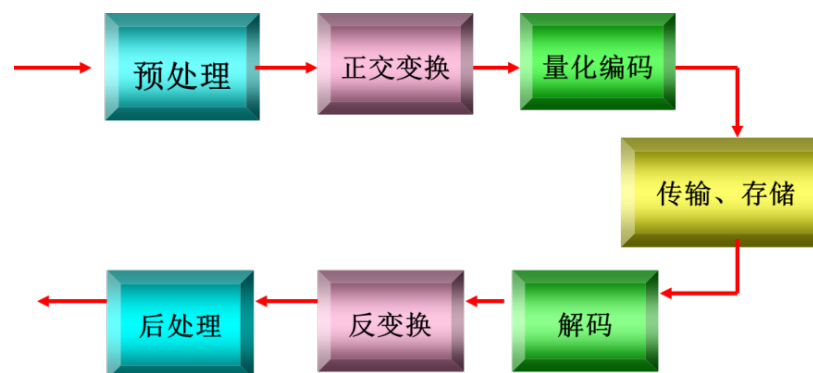


图5—50 正交变换编码原理框图

在译码端，首先将收到的码字进行译码，然后进行反变换以使变换系数恢复为空间域样值，最后经过处理使数字信号变为模拟信号显示。

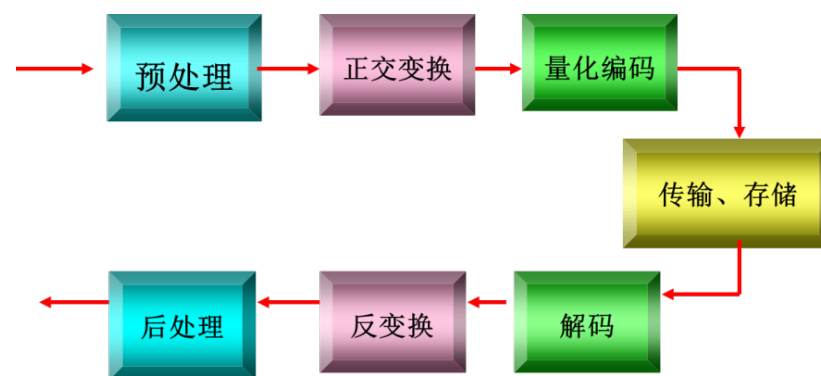


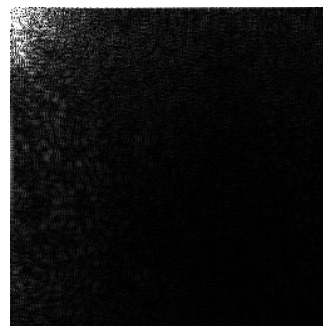
图5—50 正交变换编码原理框图

正交变换编码之所以能压缩数据率，主要是它有如下一些性质：

1) 正交变换具有熵保持性质。这说明通过正交变换不丢失信息，因此，可以用传输变换系数来达到传送信息的目的。

2) 正交变换有能量保持性质。这就是第三章提到的各种正交变换的帕斯维尔能量保持性质。它的意义在于：只有当有限离散空间域能量全部转移到某个有限离散变换域后，有限个空间取样才能完全由有限个变换系数对于基础矢量加权来恢复。

3) **能量重新分配与集中**。这个性质使我们有可能采用熵压缩法来压缩数据，也就是在质量允许的情况下，舍弃一些能量较小的系数，或者对**能量大的谱点分配较多的比特，对能量较小的谱点分配较少的比特**，从而带来较大的压缩比。



4) **去相关特性**。正交变换可以使高度相关的空间样值变为相关性很弱的变换系数。换句话说，正交变换有可能使相关的空间域转变为不相关的变换域。这样就使存在于相关性之中的冗余度得以去除。

综上所述，由于正交变换的性质，图像的空间域可变为能量保持、集中且不相关的变换域。用变换系数代替空间样值编码传送时，只需对变换系数中能量比较集中的部分加以编码，这样就能使数字图像传输或存贮时所需的码率得到压缩。

6.2.2 变换编码的数学模型

由正交变换编码的基本概念可知，编码过程是在变换域上进行。在这个基础上可以建立以下数学模型：

设图像信源为向量 $[X]'$

$$[X]' = [X_0, X_1, X_2 \cdots X_{N-1}] \quad (5-95)$$

变换后输出向量 $[Y]'$

$$[Y]' = [Y_0, Y_1, Y_2, \cdots, Y_{N-1}] \quad (5-96)$$

取正交变换为 $[T]$ ，那么 $[X]$ 与 $[Y]$ 之间的关系为

$$[Y] = [T] [X] \quad (5-97)$$

由于[T]是正交矩阵，所以

$$[T][T]' = [I] = [T]^{-1} [T] \quad (5-98)$$

这里[I]为单位矩阵， $[T]'$ 是[T]的转置， $[T]^{-1}$ 是[T]的逆。所以有：

$$[X] = [T]' [Y] \quad (5-99)$$

也就是说在编码端可利用变换得到[Y]，在译码端可用反变换来恢复[X]。

如果在传输或存贮中只保留 M 个分量, $M < N$, 则可由 $[Y]$ 的近似值 $[Y]^0$ 来恢复 $[X]$ 。

$$[Y]^0 = [Y_0, Y_1, Y_2, \dots, Y_{M-1}]$$

$$[X]^0 = [T]'^0 [Y]^0$$

当然 $[X]^0$ 是 $[X]$ 的近似值。但只要选取得当, 仍可保证失真在允许的范围内。

关键问题在于选取什么正交变换 $[T]$ ，才能既得到最大的压缩率，又不造成严重失真。因此，先研究一下由正交变换得到 $[Y]$ 的统计特性。 $[Y]$ 的统计特性中最为重要的是协方差矩阵。下面讨论一下正交变换后得到的 $[Y]$ 的协方差矩阵采用何种形式。

设图像信号是N维向量

$$[X]' = [X_0, X_1, X_2, \cdots, X_{N-1}] \quad (5-101)$$

当然，[X]的统计特性可以测得：

[X] 的协方差矩阵

$$[C_X] = E \left\{ ([X] - [\bar{X}])([X] - [\bar{X}])' \right\} \quad (5-102)$$

式中 $[C_X]$ 是[X]的协方差矩阵， $[\bar{X}]$ 是[X]的均值，
E是求数学期望值。

又设变换系数向量为

$$[Y]' = [Y_0, Y_1, Y_2, \cdots, Y_{N-1}] \quad (5-103)$$

$[C_Y]$ 为 $[Y]$ 的协方差矩阵，所以

$$[C_Y] = E \left\{ ([Y] - [\bar{Y}])([Y] - [\bar{Y}])' \right\} \quad (5-104)$$

式中 $[\bar{Y}]$ 是 $[Y]$ 的均值。

由正交变换的定义，有

$$[Y] = [T][X]$$

$$[X] = [T]'[Y]$$

因此

$$\begin{aligned}[C_Y] &= E\{([Y] - [\bar{Y}])([Y] - [\bar{Y}])'\} \\&= E\{[T][X] - [T][\bar{X}])([T][X] - [T][\bar{X}])'\} \\&= E\{[T]([X] - [\bar{X}])([X] - [\bar{X}])'[T]'\} \\&= [T]E\{([X] - [\bar{X}])([X] - [\bar{X}])'\}[T]' \\&= [T][C_X][T]'\end{aligned}$$

即

$$[C_Y] = [T][C_X][T]' \quad (5-105)$$

式(5—105)说明，变换系数的协方差矩阵可以通过空间域图像的协方差矩阵的二维变换得到。由此得出结论：变换系数的协方差矩阵决定于变换矩阵 $[T]$ 和空间域图像的协方差矩阵 $[C_X]$ 。而 $[C_X]$ 是图像本身所固有的，因此关键在于寻求合适的 $[T]$ 。

如果 $[C_y]$ 是对角矩阵，就说明系数间的相关性完全解除了，即解除了包含在相关性中的冗余度，为无失真压缩编码打下了基础。另外，还希望对角形矩阵中元素的能量尽量集中，以便使舍去若干系数后造成的误差不致于太大，这样，就为熵压缩编码提供了条件。

综上所述，变换编码要解决的关键问题是合理地寻求变换矩阵[T]。

6.2.3 最佳变换问题

在研究各种变换矩阵 $[T]$ 的过程中，自然要比较它们的优劣，因此，**就有一个比较准则问题**。下面讨论最佳变换问题。

1) 最佳变换应满足下面两个条件

1). 能使变换系数之间的相关性全部解除

2). 能使变换系数的方差高度集中

第一个条件希望变换系数的协方差矩阵为对角矩阵；

第二个条件希望对角形矩阵中对角线上的元素能量主要集中在前 M 项上，这样就可以保证在去掉 $N-M$ 项后的截尾误差尽量小。

2) 最佳的准则

常用的准则仍然是均方误差准则。均方误差由下式表示

$$\begin{aligned}\bar{e}^2 &= \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} e^2(x, y) \\ &= \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} [g(x, y) - f(x, y)]^2\end{aligned}\quad (5-106)$$

式中 $f(x,y)$ 代表原始图像， $g(x,y)$ 为经编、译码后的恢复图像。均方误差准则就是使 $\bar{e}^2$ 最小，对应变换就是最佳变换。

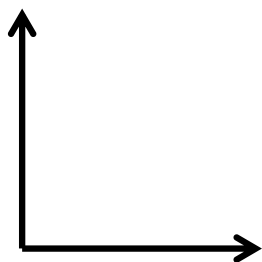
3) 均方误差准则下的最佳变换

均方误差准则下的最佳变换也叫K-L变换。

设 T 是一正交变换矩阵

$$T' = [\phi_1 \ \phi_2 \ \cdots \ \phi_N] \quad (5-107)$$

这是一个 $N \times N$ 矩阵，其中 ϕ_i 是一个 N 维列向量(基向量，投影基)。这个矩阵是正交的，因此

$$T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$


$$\varphi_i' \varphi_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \quad (5-108)$$

显然 $\mathbf{T}' \mathbf{T} = \mathbf{I}$

另外，设有一数据向量

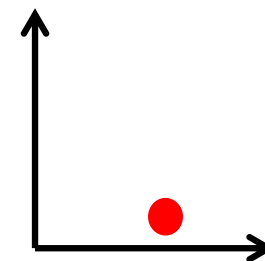
$$\mathbf{X}' = [X_1 \ X_2 \ \cdots \ X_N]$$

经正交变换后

$$\mathbf{Y} = \mathbf{T} \mathbf{X}$$

$$\mathbf{X} = \mathbf{T}' \mathbf{Y}$$

$$\mathbf{T} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



这里 $\mathbf{Y}' = [Y_1 \ Y_2 \ \cdots \ Y_N]$

而 $\mathbf{X} = \mathbf{T}' \mathbf{Y} = [\phi_1 \ \phi_2 \ \cdots \ \phi_N] \mathbf{Y}$

于是

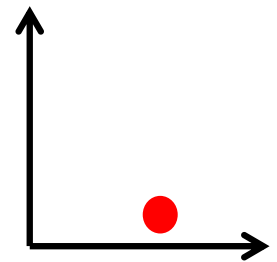
$$\mathbf{X} = Y_1 \phi_1 + Y_2 \phi_2 + \cdots + Y_N \phi_N = \sum_{i=1}^N Y_i \phi_i \quad (5-111)$$

$$X = Y_1\varphi_1 + Y_2\varphi_2 + \cdots + Y_N\varphi_N = \sum_{i=1}^N Y_i\varphi_i$$

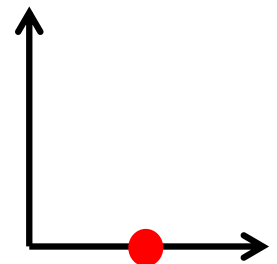
为了压缩数据，在恢复 X 时不取完整的 N 个 Y 分量，而仅取 M 个分量，其中 $M < N$ 。这样其中 M 个分量构成一个子集，即

$$[Y_1 \ Y_2 \ \cdots \ Y_M]$$

$$T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



$$T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



用前 M 个分量去估计 X ，其余的用常量 b_i 来代替。于是可得到

$$\overset{o}{X} = \sum_{i=1}^M Y_i \phi_i + \sum_{i=M+1}^N b_i \phi_i \quad (5-112)$$

这里 $\overset{o}{X}$ 是 X 的估计。

$\mathbf{X}$ 的值与 $\bar{\mathbf{X}}^0$ 的误差为

$$\begin{aligned}\Delta\mathbf{X} &= \mathbf{X} - \bar{\mathbf{X}}^0 \\ &= \mathbf{X} - \sum_{i=1}^M Y_i \varphi_i - \sum_{i=M+1}^N b_i \varphi_i \\ &= \sum_{i=M+1}^N (Y_i - b_i) \varphi_i\end{aligned}$$

设 $\mathcal{E}$ 为均方误差，则

$$\mathcal{E} = E \left\{ \|\Delta\mathbf{X}\|^2 \right\} = E \left\{ (\Delta\mathbf{X})^T (\Delta\mathbf{X}) \right\}$$

将 ΔX 代入 ε

$$\varepsilon = E \left\{ \sum_{i=M+1}^N \sum_{j=M+1}^N (Y_i - b_i)(Y_j - b_j) \phi_i^T \phi_j \right\}$$

由上述的正交条件可简化为 $\phi_i' \phi_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$

$$\varepsilon = \sum_{i=M+1}^N E \{ (Y_i - b_i)^2 \}$$

根据最小均方误差准则，使 ε 最小就要正确选取 ϕ_i 及 b_i 。为了求得最佳的 ϕ_i 和 b_i ，可分两步来求：

第一步把 ε 对 b_i 求导并令其等于零，即

$$\begin{aligned}\frac{\partial \varepsilon}{\partial b_i} &= \frac{\partial}{\partial b_i} E\{(Y_i - b_i)^2\} \\ &= -2[E\{Y_i\} - b_i] = 0 \\ b_i &= E\{Y_i\}\end{aligned}\tag{5—116}$$

又因为

$$Y_i = \phi_i' X$$

所以

$$b_i = \phi_i' E\{X\} = \phi_i' \bar{X}$$

$$\bar{X} = E\{X\}$$

将
$$\begin{aligned} Y_i &= \phi_i' X. \\ b_i &= \phi_i' \bar{X} \end{aligned}$$
 代入 ε , 则

$$\begin{aligned} \varepsilon &= \sum_{i=M+1}^N E\{(Y_i - b_i)(Y_i - b_i)'\} \\ &= \sum_{i=M+1}^N \phi' E\{(X - \bar{X})(X - \bar{X})'\} \phi_i \end{aligned} \quad (5-118)$$

因为 $E\{(X - \bar{X})(X - \bar{X})'\} = [C_X]$

所以
$$\varepsilon = \sum_{i=M+1}^N \phi_i' [C_X] \phi_i \quad (5-119)$$

第二步求最佳化的 ϕ_i 。为了求得最佳的 ϕ_i ，不仅要找出 ϕ_i 使 ε 最小，而且还要满足 $\phi_i' \phi_i = 1$ 的正交条件。因此，可用求条件极值的拉格朗日乘数法法则，在求 ϕ_i 的条件极值时做一个新函数 $\hat{\varepsilon}$ 。

$$\begin{aligned}
\hat{\varepsilon} &= \varepsilon - \sum_{i=M+1}^N \beta_i \left[\varphi_i' \varphi_i - 1 \right] \\
&= \sum_{i=M+1}^N \left\{ \varphi_i' C_X \varphi_i - \beta_i \left[\varphi_i' \varphi_i - 1 \right] \right\}
\end{aligned}$$

(5—120)

对 ϕ_i 求导，并注意到

$$\frac{\partial}{\partial \phi_i} \left[\phi_i' \mathbf{C}_X \phi_i \right] = 2 \mathbf{C}_X \phi_i \quad (5-121)$$

$$\frac{\partial}{\partial \phi_i} \left[\phi_i' \phi_i \right] = 2 \phi_i \quad (5-122)$$

所以

$$\frac{\partial \hat{\varepsilon}}{\partial \phi_i} = 2\mathbf{C}_X \phi_i - 2\beta_i \phi_i = 0 \quad (5-123)$$

即

$$\mathbf{C}_X \phi_i = \beta_i \phi_i \quad (5-124)$$

由线性代数理论可知

$$\beta_i \varphi_i - C_X \varphi_i = 0$$

(5—125)

$$(\beta_i I - C_X) \varphi_i = 0$$

显然, β_i 就是 C_X 的特征根, ϕ_i 就是 C_X 的特征向量。

若 C_X 是对称矩阵, 就可找到一个变换 $[T]$, 使 C_Y 成为对角形矩阵。

4) 最佳变换的实现方法

由上面的分析可见， K - L 变换中的变换矩阵 $[T]$ 不是一个固定的矩阵，它由信源的 C_X 确定。当给定一信源时，可用如下几个步骤求得 $[T]$ ：

a) 给定一幅图像后，首先要统计其协方差矩阵 C_X

b) 由 C_X 求 λ 矩阵，即 $[\lambda E - C_X]$ 。

由 $[\lambda E - C_X]$ 求得其特征根，进而求得每一个特征根所对应的特征向量；

c) 由特征向量求出变换矩阵 $[T]$ ；

d) 用求得的 $[T]$ 对图像数据进行正交变换。

经过上面运算就可以保证 $[\mathbf{C}_Y] = [\mathbf{T}][\mathbf{C}_X][\mathbf{T}]'$

是一个对角形矩阵。这个 $[\mathbf{T}]$ 就是 K - L 变换中的变换矩阵。

通过上面的讨论不难看出，图像不同， C_X 就不同，因此 $[T]$ 也就不同。为了得到最佳变换，每送一幅图像就要重复上述四个步骤，找出 $[T]$ 后再进行正交变换操作，所以运算相当繁琐，而且没有快速算法。此外， $K-L$ 变换在数学推导上总能实现，

但用硬件实现就不那么容易了。因此， $K-L$ 变换的实用性受到很大限制，一般多用来作变换性能的评价标准。当然，寻求 $K-L$ 变换的简洁算法也是许多人研究的课题。下面举一个简单的例子说明 $K-L$ 变换的具体实现方法。

例：已知某信源的协方差矩阵为 C_X ，求最佳变换矩阵 $[T]$ 。

$$C_X = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

写出 λ 矩阵

$$[\lambda E - C_X] = \begin{bmatrix} \lambda - 1 & -1 & 0 \\ -1 & \lambda - 1 & 0 \\ 0 & 0 & \lambda - 1 \end{bmatrix}$$

$$\begin{aligned}
 f(\lambda) &= \begin{vmatrix} \lambda - 1 & -1 & 0 \\ -1 & \lambda - 1 & 0 \\ 0 & 0 & \lambda - 1 \end{vmatrix} \\
 &= (\lambda - 1)^3 - (\lambda - 1) = 0
 \end{aligned}$$

求得

$$\lambda_1 = 2, \lambda_2 = 1, \lambda_3 = 0$$

求 $\lambda_1 = 2, \lambda_2 = 1, \lambda_3 = 0$ 的特征向量

$$\lambda_1 = 2:$$

$$(\lambda_1 E - C_X)X = 0$$

$$\begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = 0$$

$$\begin{cases} X_1 - X_2 = 0 \\ -X_1 + X_2 = 0 \\ X_3 = 0 \end{cases}$$

所以其基础解系为 $(1 \ 1 \ 0)$

归一化后为 $\left(\frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \quad 0\right)$

同理 $\lambda_2 = 1$ 时特征向量为 $(0 \quad 0 \quad 1)$

$\lambda_3 = 0$ 时特征向量为 $\left(\frac{1}{\sqrt{2}} \quad \frac{-1}{\sqrt{2}} \quad 0\right)$

由上面的结果可求得 $[T]$

$$[T] = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \end{bmatrix} \quad [T]^T = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \end{bmatrix}$$

$[T]$ 便是*K-L*变换的变换矩阵。

6.2.4 准最佳变换

最佳变换的性能固然好，但实现起来不容易。因此，在实践中更加重视一些所谓的准最佳变换。什么是准最佳变换呢？最佳变换的核心在于经变换后能使 $[C_Y]$ 成为对角形矩阵形式。如果能找到某些固定的变换矩阵 $[T]$ ，使变换后的 $[C_Y]$ 接近于对角形矩阵，那也比较理想。

在线性代数理论中知道，任何矩阵都可相似于一个约旦(Jordan)形矩阵，这个约旦形矩阵就是准对角形矩阵，其形状如式(5—127)所示。

$$\begin{bmatrix} \lambda_0 & & & & \\ \mathbf{O} & \lambda_1 & & & \mathbf{O} \\ \vdots & \mathbf{1} & \lambda_2 & & \\ \vdots & & \mathbf{O} & \ddots & \\ \vdots & & & \ddots & \lambda_{N-2} \\ \mathbf{O} & \mathbf{O} & \cdots & \mathbf{1} & \lambda_{N-1} \end{bmatrix} \quad (5-127)$$

其主对角线上是特征值，在下对角线上仅有若干个 1，这也就比较理想了。从相似变换理论可知，总可以找到一个非奇异矩阵 T ，使得

$$T^T A T = B \quad (5-128)$$

这个 T 并不是唯一的。

在第三章讨论过的正交变换都具有 T 的性质，是常用的准最佳变换。尽管它们的性能比 $K-L$ 变换稍差，但它们的变换矩阵 T 是固定的，这给工程实现带来了极大的方便。因此，首先付诸于实用的是这些准最佳变换法。

6.2.5 各种准最佳变换的性能比较

从运算量大小以及压缩效果这两个方面来比较各种正交变换，其性能比较如表所示。

	正交变换类型	运 算 量	对视频图像实时处理的难易程度
劣 ↑ ↓ 优	K-L	求 $[C_X]$ 及其特征值，特征矢量。 矩阵运算要 N^2 次实数加法和 N^2 次实数乘法	极难做到
	DFT	$N\log_2 N$ 次复数乘法以及 $N\log_2 N$ 次复数加法	较复杂
	FDCT	$\frac{3N}{2}\log_2(N-1)+2$ 次实数加法 以及 $N\log_2 N - \frac{3N}{2} + 4$ 次实数乘法	采用高速 CMOS/SOS 大规模集成电路，能做到实时处理
	ST	当 $N=4$ 时，共需8次加或减以及6次乘法	可能实时
	DWHT	$N\log_2 N$ 次实数加或减	可利用一般高速 TTL，ECL 数字集成电路做到实时
	HT	$2(N-1)$ 次实数加或减	与 DWHT 相同

6.2.6 编码

变换为压缩数据创造了条件，压缩数据还要靠编码来实现。通常所用的编码方法有二种：

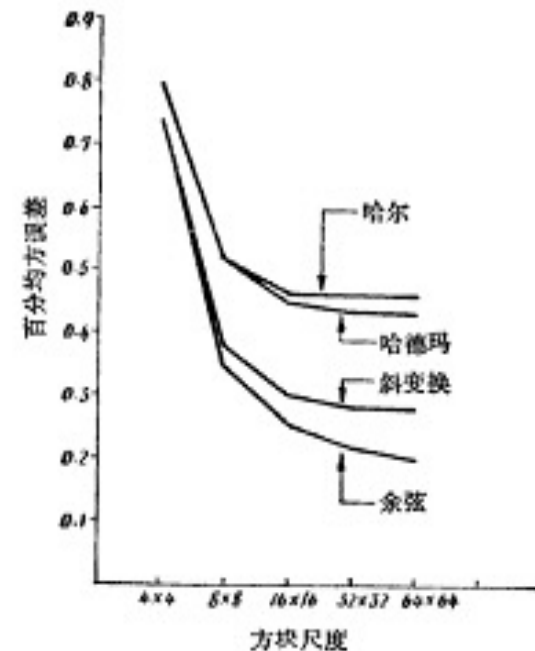
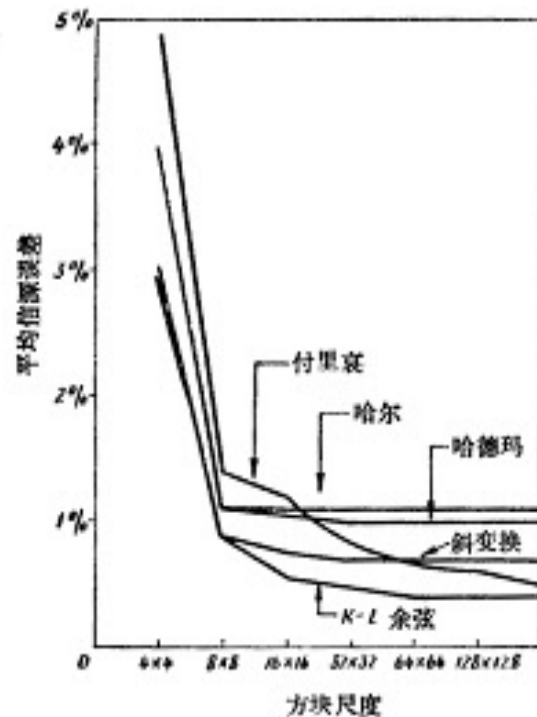
一是区域编码法，

二是门限编码法。

1) 区域编码法

这种方法的**关键在于选出能量集中的区域**，在编码过程中就可以选取这一区域的系数进行编码传送，而其他区域的系数可以舍弃不用。在译码端可以对舍弃的系数进行补零处理。这样由于保持了大部分图像能量，在恢复图像中带来的质量劣化并不显著。

在区域编码中，区域抽样和区域编码的均方误差均与方块尺度有关。图5—52示出了区域抽样的均方误差与方块尺度的关系。图5—53则示出了区域编码均方误差与方块尺度的关系。



区域编码的显著缺点是一旦选定了某个区域就固定不变了，有时图像中的能量也会在其他区域有较大的数值，舍掉它们会造成图像质量较大的损失。

2) 门限编码

这种采样方法不同于区域编码法，不是选择固定的区域，而是事先设定一个门限值 T 。如果系数超过 T 值，就保留下来并进行编码传送。如果系数值小于 T 值就舍弃不用。这种方法有一定的自适应能力。它可以得到比区域编码好的图像质量。但这种方法也有缺点，那就是超过门限值的系数的位置是随机的。

因此，在编码中除对系数值编码外，还要有位置码。这两种码同时传送才能在接收端正确恢复图像。所以，其压缩比有时会有所下降。

6.2.7 图像编码的国际标准

在图像编码中，目前的国际标准是：

①、静止图像：**JPEG (Joint Photographic Expert Group)**：“联合图片专家组”

1991年提出的 **ISO CD 10916**建议草案。

这个建议规定了具体的编码方法及质量要求，即：

* **8 × 8 DCT + Huffman 编码**（基本）；

* 自适应算术编码（扩展）；无失真预测，帧内预测及Huffman；

②、可视电话 / 会议电视：

* CCITT H. 261标准，1988年提出 $P \times 64\text{Kbit/s}$ 。

P为可变系数，对于可视电话，建议 $P=2$ ，会议电视建议 $P \geq 6$ 。编码方法可采用混合编码法，即采用DCT变换，运动补偿DPCM及Huffman编码等方法。

③、 MPEG (Motion Picture Expert Group 运动图像专家组)

CCITT的ISO CD 11172 号建议, MPEG1指标是压缩PCM一次群(1.5Mb/s--2Mb/s), 采用DCT、运动补偿、帧内、帧间预测等方法。

具体有MPEG1、MPEG2、MPEG4、MPEG7等。