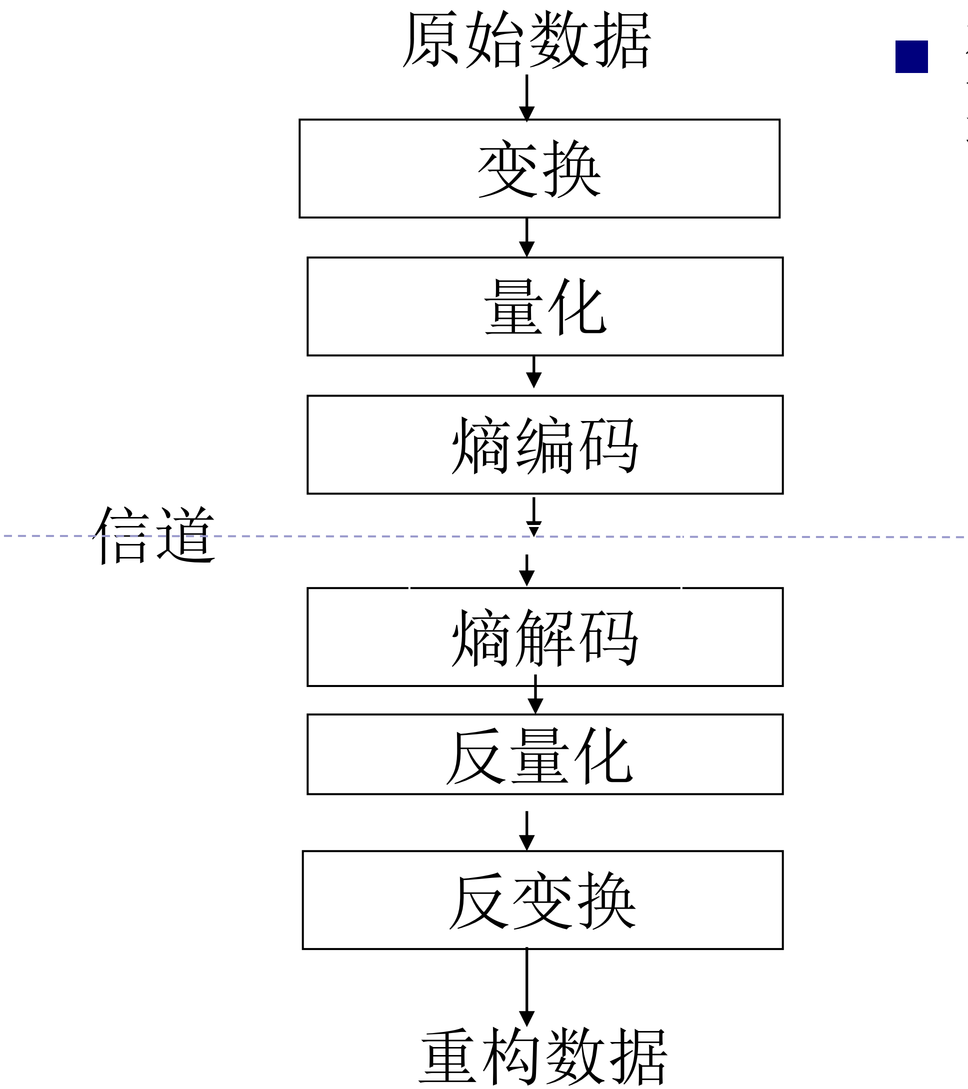
第六章　　变换编码

主要内容

- 6.1 变换编码基本原理
- 6.2 离散正交变换
  - **6.2.1 基本概念**
  - **6.2.2 KL变换**
  - **6.2.3 DCT变换**
- **6.3 JPEG图像压缩标准**

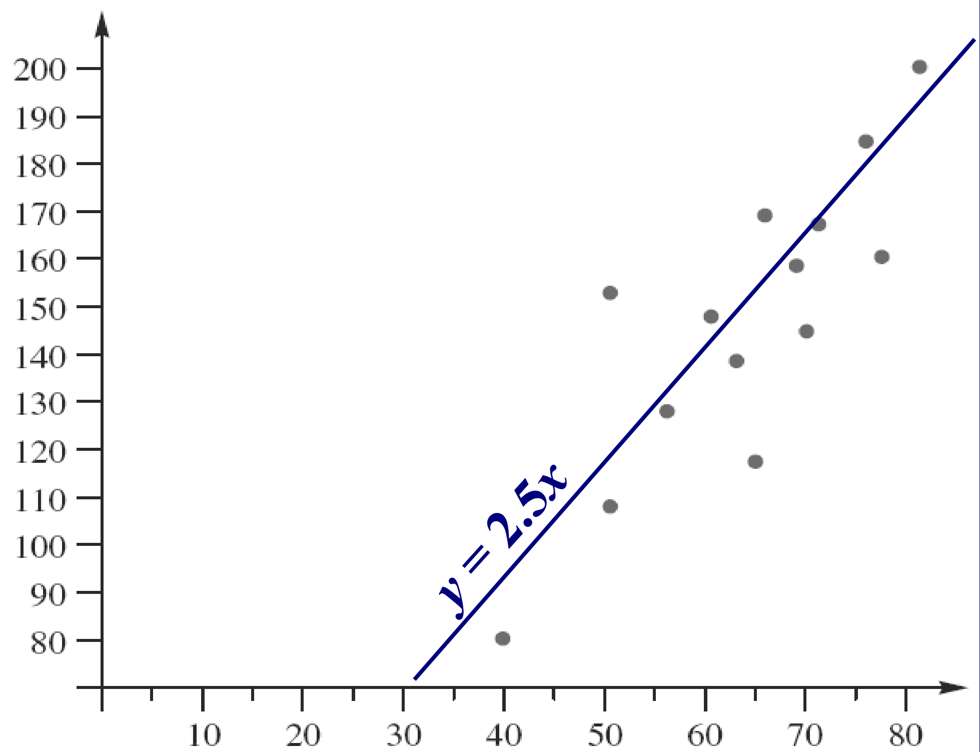
6.1 变换编码的基本原理



- 变换编码：将输入信号变换，得到一些系数表示
  - **去相关**：将输入信号去相关，使得在量化时，对各系数单独量化（标量量化），而不会损伤过多效率（与矢量量化相比）
  - **稀疏化**：将原始信号的能量压缩到尽可能少的系数→对原始信号只用少数幅值较大的系数表示
  - **可逆**：可以重构输入信号

例1 变换的几何意义

Height	Weight
65	170
75	188
60	150
70	170
56	130
80	203
68	160
50	110
40	80
50	153
69	148
62	140
76	164
64	120



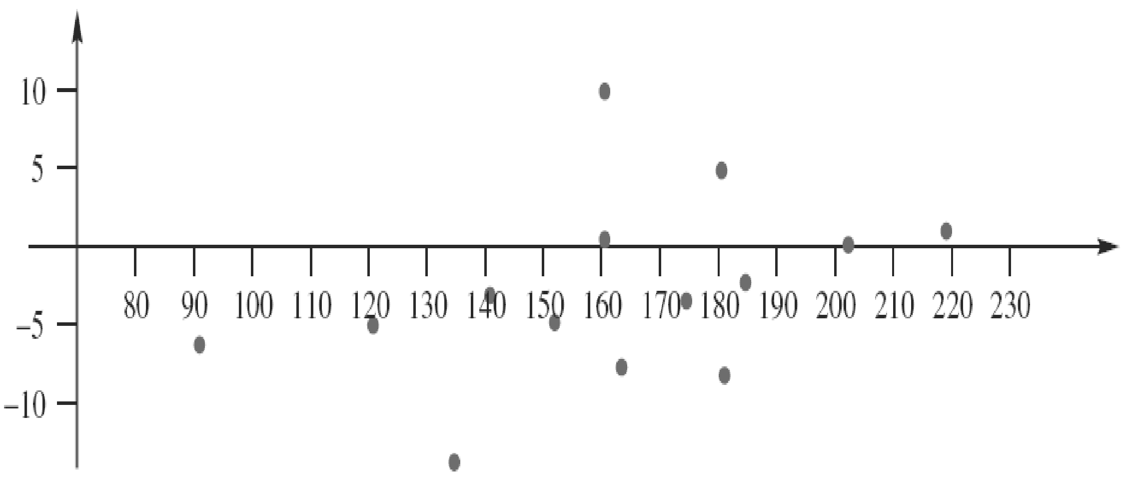
例1 变换的几何意义

线性变换A是将R2中的单位正方形变成R2中以变换核矩阵中行向量a和b为邻边的平行四边形。变换核矩阵的行列式就是变换前后的面积比，也称为伸缩因子。

- 考虑（可逆）变换 $\theta = AX$

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}, \mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}, \mathbf{A} = \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix}, \phi = \arctan 2.5$$

First Coordinate	Second Coordinate
182	3
202	0
162	0
184	−2
141	−4
218	1
174	−4
121	−6
90	−7
161	10
163	−9
153	−6
181	−9
135	−15



$$\mathbf{A} = \begin{bmatrix} 0.37139068 & 0.92847669 \\ -0.92847669 & 0.37139068 \end{bmatrix}$$

例1 变换的几何意义

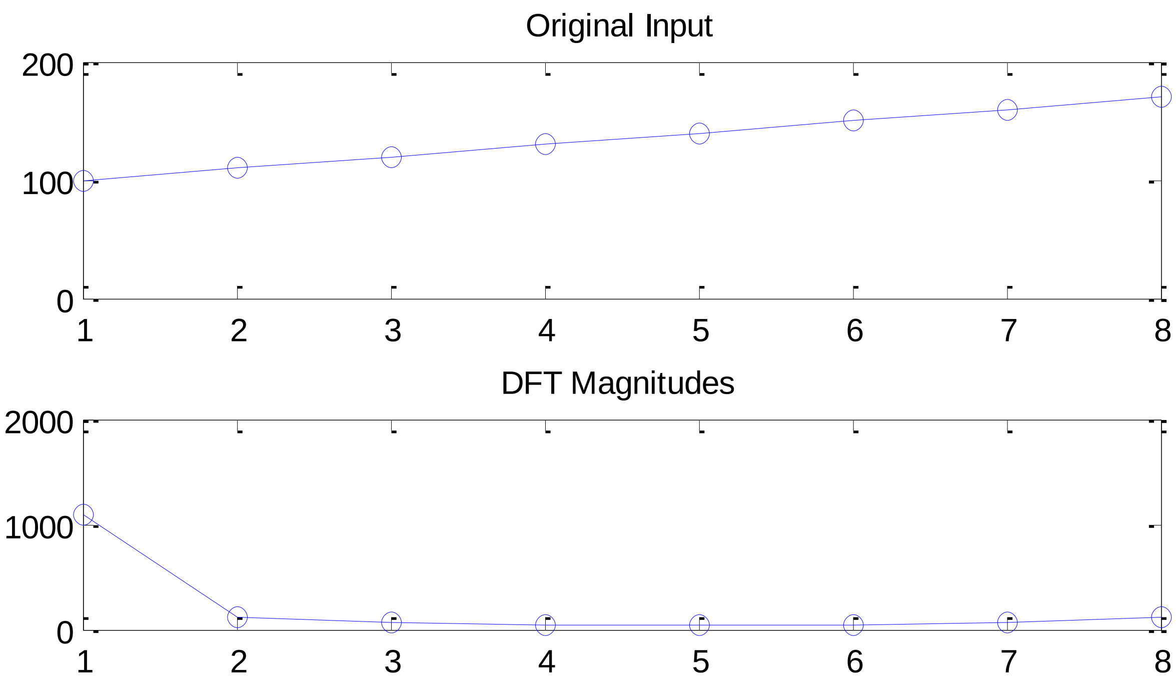
抛弃第二维，再反变换得到重构数据

Height	Weight
65	170
75	188
60	150
70	170
56	130
80	203
68	160
50	110
40	80
50	153
69	148
62	140
76	164
64	120

Height	Weight
68	169
75	188
60	150
68	171
53	131
81	203
65	162
45	112
34	84
60	150
61	151
57	142
67	168
50	125

$$X' = A^{-1}\theta \qquad A^{-1} = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}$$

例2 变换的物理意义- 从信号分解的角度



- 变换编码：能量的重新分配
  - 对于缓变信号，其频谱的高频分量迅速衰减，能量集中于直流和个别低频分量中
  - 高频分量与携带主要能量的低频分量相关性不大。即使舍去也差别不大。

6.2.1 离散正交变换的基本概念

- 令相邻的n个信号样本看作在n维线性空间中的一个列向量

$X=(x_1,x_2,\ldots,x_n)^T$

对它进行线性变换

$Y=AX$

称A为变换矩阵，Y是X的一个线性变换。

如果矩阵A具有性质：$AA^T=A^TA=I$

正交变换既是一个保长度变换，又是一个保夹角变换

则称A为正交矩阵，从X到Y的变换为正交变换

对于正交变换，反变换可以唯一的得到复原信号

$X=A^TY$

6.2.1 离散正交变换的基本概念

- 对正交变换总的能量保持：

$$\boldsymbol{y} = \mathbf{A}\boldsymbol{x} \qquad \|\boldsymbol{y}\|^2 = \boldsymbol{y}\boldsymbol{y}^T = \boldsymbol{x}^T\mathbf{A}^T\mathbf{A}\boldsymbol{x} = \|\boldsymbol{x}\|^2$$

- 总的能量保持，但是通常能量在各系数上分布并不均匀
- X的协方差矩阵：

$$\phi_X = E\{[X - E(X)][X - E(X)]^T\} = \begin{bmatrix} \phi_{11} & \phi_{12} & \cdots \phi_{1N} \\ \phi_{21} & \phi_{22} & \cdots \phi_{2N} \\ \vdots & & \\ \phi_{N1} & \phi_{N2} & \cdots \phi_{NN} \end{bmatrix}$$

$$\text{其中}, \phi_{ij} = E\{[x_i - E(x_i)][x_j - E(x_j)]\} = \phi_{ji}$$

6.2.1 离散正交变换的基本概念

- Y的协方差矩阵：

$$\begin{aligned}\phi_Y &= E\left\{[Y-E(Y)][Y-E(Y)]^T\right\} \\ &= E\left\{A[X-E(X)][X-E(X)]^T A^T\right\} \\ &= A\phi_X A^T\end{aligned}$$

为使得变换后系数相互独立，即要令Y的协方差矩阵中只存在对角线元素。

对于一个实对称矩阵Φ，必存在一个正交矩阵Q，使得

$$Q\phi Q^{-1} = diag[\lambda_1\ \lambda_2 \cdots \lambda_N] = \Lambda$$

式中,对角阵$\Lambda = diag[\lambda_1\ \lambda_2 \cdots \lambda_N]$的$N$个对角元$\lambda_1\ \lambda_2 \cdots \lambda_N$是$\phi$的特征根

Q中的第i个行向量是ϕ的第i个特征根λ_i所对应的满足归一化正交条件的特征向量

即：$\phi q_i = \lambda_i q_i$，$\quad q_i^T q_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$

6.2.1 离散正交变换的基本概念

$$Q\phi Q^{-1} = diag[\lambda_1\ \lambda_2 \cdots \lambda_N] = \Lambda$$

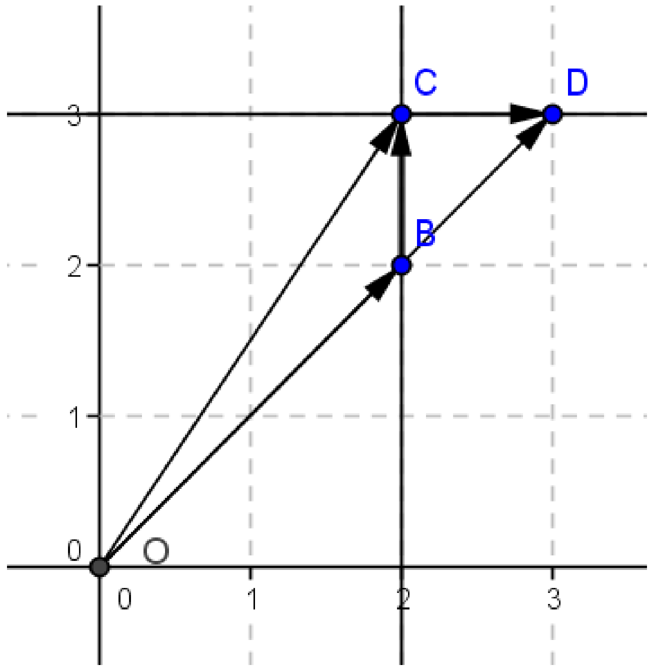
式中,对角阵$\Lambda = diag[\lambda_1\ \lambda_2 \cdots \lambda_N]$的$N$个对角元$\lambda_1\ \lambda_2 \cdots \lambda_N$是$\phi$的特征根

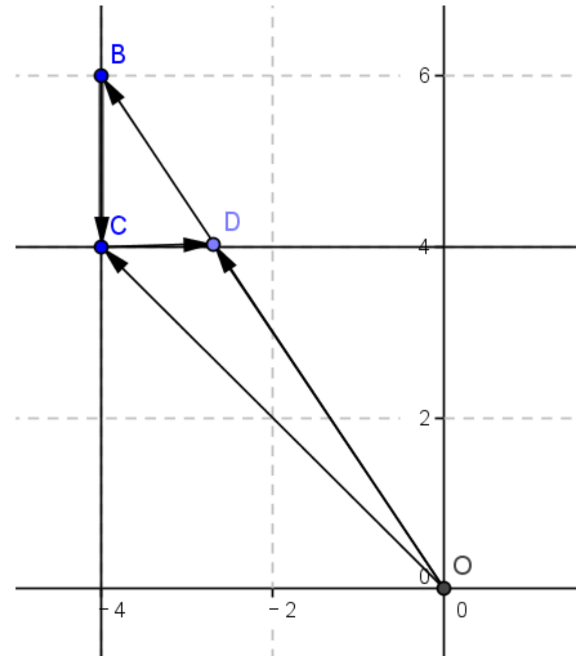
Q中的第i个行向量是ϕ的第i个特征根λ_i所对应的满足归一化正交条件的特征向量

即：$\phi q_i = \lambda_i q_i, \quad q_i^T q_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$

- 对于确定的方阵A，如果存在向量**x**与数值λ，使得A**x**=λ**x**，则定义λ为方阵A的特征值，称**x**为方阵A的特征向量；
- 向量**x**经过矩阵A的变换等价于经过因子λ的伸缩变换，所以特征向量和特征值可以如下概述：一个矩阵在对任意向量进行变换时，可能会存在一种特殊情形，即一些方向上的向量在变换后还是在此方向（和原向量共线），这些向量就称为这个矩阵的特征向量，对这些特征向量来说，变换矩阵完全可以用一个伸缩因子代替，这个伸缩因子就称为对应的特征值

6.2.1 离散正交变换的基本概念





向量OB先沿y方向剪切变换变为OC，可得到这一过程的变换矩阵为 $\begin{pmatrix} 1 & 0 \\ 0.5 & 1 \end{pmatrix}$，然后沿x方向剪切变换得到OD，这一过程的矩阵为 $\begin{pmatrix} 1 & \frac{1}{3} \\ 0 & 1 \end{pmatrix}$

6.2.1 离散正交变换的基本概念

- 只要选择正交矩阵Q作为变换核矩阵，其行向量是X的协方差矩阵的特征向量，则变换后的矢量Y的协方差矩阵为：

$$
\begin{aligned}
\phi_Y &= E\left\{[Y - E(Y)][Y - E(Y)]^T\right\} \\
&= E\left\{[QX - E(QX)][QX - E(QX)]^T\right\} \\
&= QE\left\{[X - E(X)][X - E(X)]^T\right\}Q^T \\
&= Q\phi_X Q^T = \Lambda
\end{aligned}
$$

则变换后各分量之间的相关性被全部去除

6.2.2 K-L变换

- 以矢量X的协方差矩阵的归一化正交特征向量(q_i)所构成的正交矩阵Q，对该矢量所做的正交变换
  $Y=QX$称作Karhunen-Loeve变换，即KL变换

即：

- ✓ 要实现KLT，必须要先知道输入信号的协方差矩阵，然后再求出协方差矩阵的特征值和特征向量。
- ✓ 用特征向量构成该输入信号矢量的正交变换核矩阵。

A为n阶方阵，若存在数λ和n维非0列向量x使

$Ax = \lambda x$

则λ为x的特征值，x为A对应于特征值λ的特征向量

方程组形式：（$A - \lambda I)x = 0$

如果特征值和特征向量存在，则上述方程必有非0解x，即要求系数矩阵$A - \lambda I$的秩$< n$,所以系数行列式$|A - \lambda I| = 0$,该式是关于λ的n次多项式，即矩阵A的特征多项式

一个特征值可对应无穷多个不同的特征向量。某一特征向量只对应某一特征值。所以方阵A的不同特征值所对应的特征向量线性无关。

求法：

⑴$|A - \lambda I| =\lambda$的多项式，求出特征值

（2）把每个特征值代入（$A - \lambda I)x = 0$，求出齐次线性方程组的一个基础解系的全部解ξ_1，$\xi_2,\ldots,\xi_s$,则$\sum_{i=1}^{s} k_i \xi_i$是A对应于特征值λ的全部特征向量，其中$k_1, k_2, \ldots, k_s$不全为0。

6.2.2 K-L变换

- 例4：若已知随机信号X的协方差矩阵为$\phi_X = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$求正交矩阵$Q$。

解：(1)求特征值

$$令\begin{vmatrix} 1-\lambda & 1 & 0 \\ 1 & 1-\lambda & 0 \\ 0 & 0 & 1-\lambda \end{vmatrix} = 0, 按\lambda_1 > \lambda_2 > \lambda_3次序可解出: \lambda_1 = 2, \lambda_2 = 1, \lambda_3 = 0$$

(2)求特征向量

$$将q_i = [q_{i1}, q_{i2}, q_{i3}]^T代入\phi q_i = \lambda_i q_i, 解3个方程组:$$

$$由\begin{bmatrix} q_{11}+q_{12} \\ q_{11}+q_{12} \\ q_{13} \end{bmatrix} = 2\begin{bmatrix} q_{11} \\ q_{12} \\ q_{13} \end{bmatrix}, \quad 得q_{11} = q_{12} = a, q_{13} = 0, 即q_1 = \begin{bmatrix} a \\ a \\ 0 \end{bmatrix}$$

6.2.2 K-L变换

$$由\begin{bmatrix} q_{21}+q_{22} \\ q_{21}+q_{22} \\ q_{23} \end{bmatrix} = 2\begin{bmatrix} q_{21} \\ q_{22} \\ q_{23} \end{bmatrix}, \quad 得 q_{21}=q_{22}=0, q_{23}=b, 即 q_2=\begin{bmatrix} 0 \\ 0 \\ b \end{bmatrix}$$

$$由\begin{bmatrix} q_{31}+q_{32} \\ q_{31}+q_{32} \\ q_{33} \end{bmatrix} = 2\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad 得 q_{31}=q_{32}=c, q_{33}=0, 即 q_3=\begin{bmatrix} c \\ -c \\ 0 \end{bmatrix}$$

- 待定实常数可由归一化正交条件解得：

$$a=\frac{\sqrt{2}}{2}, b=1, c=\frac{\sqrt{2}}{2}$$

- 得到归一化正交矩阵：

$$Q=[q_1 \quad q_2 \quad q_3]^T=\frac{\sqrt{2}}{2}\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & \sqrt{2} \\ 1 & -1 & 0 \end{bmatrix}$$

$$Q\phi_X Q^T=\frac{1}{2}\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & \sqrt{2} \\ 1 & -1 & 0 \end{bmatrix}\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & \sqrt{2} \\ 1 & -1 & 0 \end{bmatrix}^T=\begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

6.2.2 K-L变换

- KLT：基函数为输入信号的协方差矩阵的特征向量
- KLT产生的系数不相关（输出信号协方差矩阵为对角阵）
- KLT达到最佳的能量集中
- KLT取决于信号的统计性质
  - 没有快速计算方法
- →寻找结构化的变换，使得其性能接近KLT

6.2.3 DCT变换

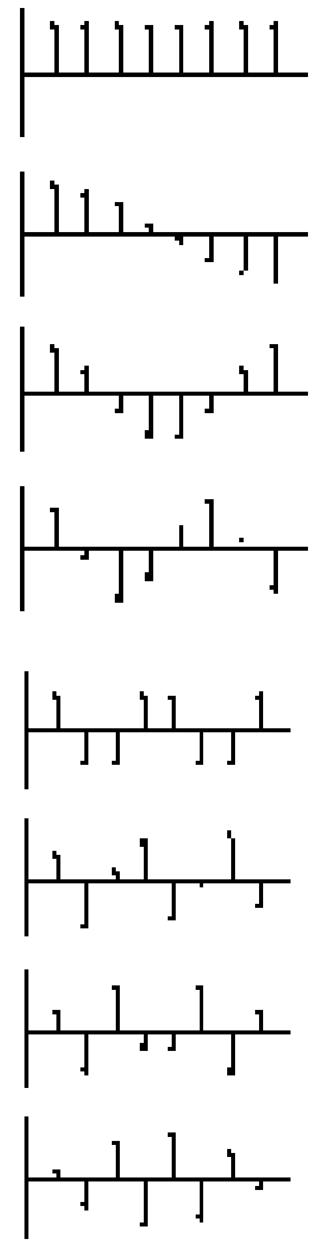
- 离散DCT变换的变换核矩阵为

$$C = \frac{1}{\sqrt{N}}\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \sqrt{2}\cos\frac{\pi}{2N} & \sqrt{2}\cos\frac{3\pi}{2N} & \cdots & \sqrt{2}\cos\frac{(2N-1)\pi}{2N} \\ \sqrt{2}\cos\frac{2\pi}{2N} & \sqrt{2}\cos\frac{6\pi}{2N} & \cdots & \sqrt{2}\cos\frac{2(2N-1)\pi}{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \sqrt{2}\cos\frac{(N-1)\pi}{2N} & \sqrt{2}\cos\frac{(N-1)3\pi}{2N} & \cdots & \sqrt{2}\cos\frac{(N-1)(2N-1)\pi}{2N} \end{bmatrix}$$

- 设信号序列 $X_0=X_1=X_2=X_3=X_4=X_5=X_6=X_7=1$，对该序列进行 1×8DCT变换：
- **Y=CX　　X=C^TY**
- 正交变换的实质可看作基底函数的分解

$$Y(K)=\sqrt{\frac{2}{8}}\bullet C(K)\sum_{n=0}^{7} X(n)\cos\frac{(2n+1)K\pi}{16}$$

$$=\frac{1}{2}\begin{bmatrix} 8\times\frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}=\begin{bmatrix} 2\sqrt{2} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}=\begin{bmatrix} Y_0 \\ Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ Y_5 \\ Y_6 \\ Y_7 \end{bmatrix}$$



6.2.3 DCT变换

- 二维DCT变换

$$F(u,v)=\frac{2}{N}C(u)C(v)\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} f(i,j)\cos\frac{(2i+1)u\pi}{2N}\cos\frac{(2j+1)v\pi}{2N}$$

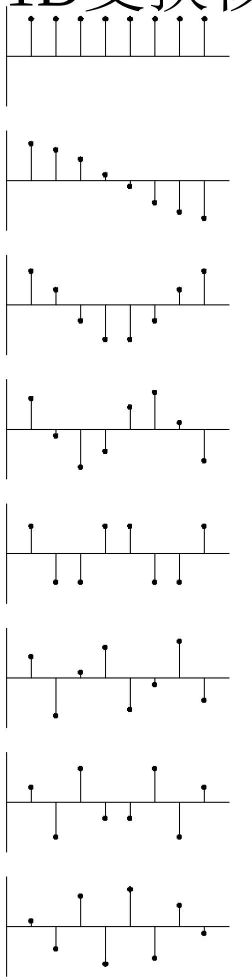
，u,v = 0,1,…,N-1

$$C(u),C(v)=\begin{cases}\frac{1}{\sqrt{2}} & ,\quad u,v=0\\ 1 & ,\quad u,v=1,2,\ldots,N-1\end{cases}$$

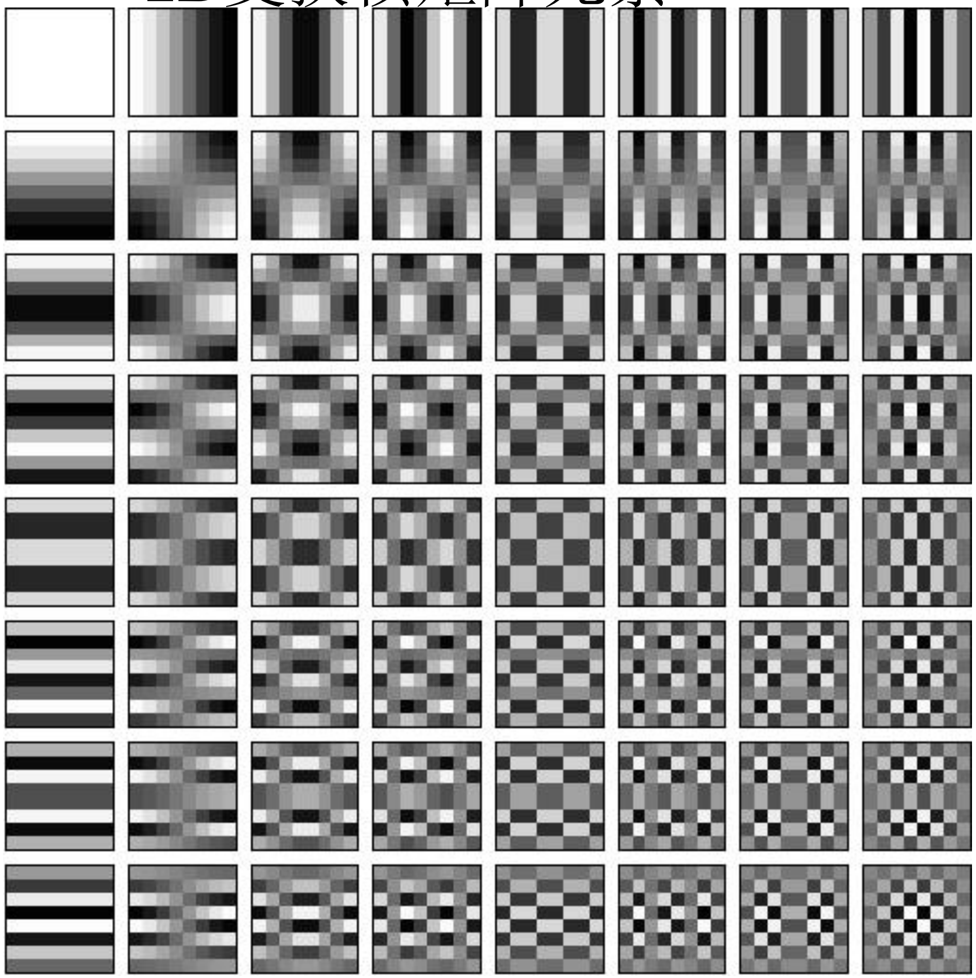
变换核可分离，先进行垂直方向的8*1DCT变换，再进行水平方向的8*1DCT变换。

6.2.3 DCT变换

1D变换核矩阵元素



2D变换核矩阵元素



例5： DCT变换

已知： $f(x,y)=\begin{bmatrix}0&0&0&0\\0&1&1&0\\0&1&1&0\\0&0&0&0\end{bmatrix}$ 用矩阵算法求其DCT。

$F(u,v)=CfC^T$

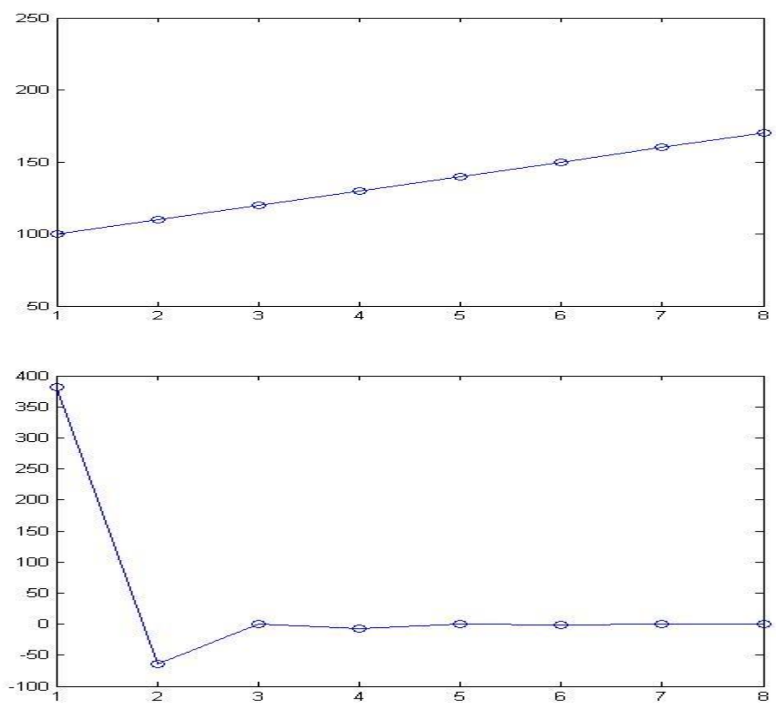
$$=\begin{bmatrix}0.5&0.5&0.5&0.5\\0.65&0.27&-0.27&-0.65\\0.5&-0.5&-0.5&0.5\\0.27&-0.65&0.65&-0.27\end{bmatrix}\begin{bmatrix}0&0&0&0\\0&1&1&0\\0&1&1&0\\0&0&0&0\end{bmatrix}\begin{bmatrix}0.5&0.65&0.5&0.27\\0.5&0.27&-0.5&-0.65\\0.5&-0.27&-0.5&0.65\\0.5&-0.65&0.5&-0.27\end{bmatrix}$$

例6：空域量化 vs. 变换域量化

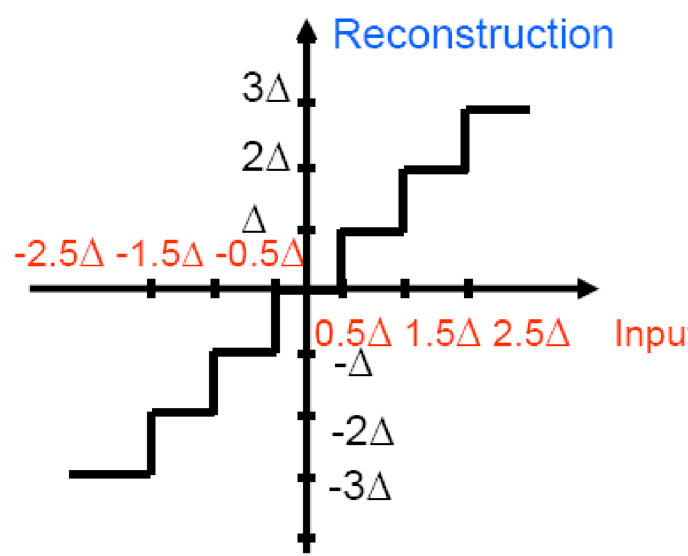
- $x = [100\ 110\ 120\ 130\ 140\ 150\ 160\ 170]^T$
- 8点DCT变换：

  y = [381.84 -64.42 0 -6.73 0 -2.01 0 -0.5070]

  能量主要集中在前两个系数



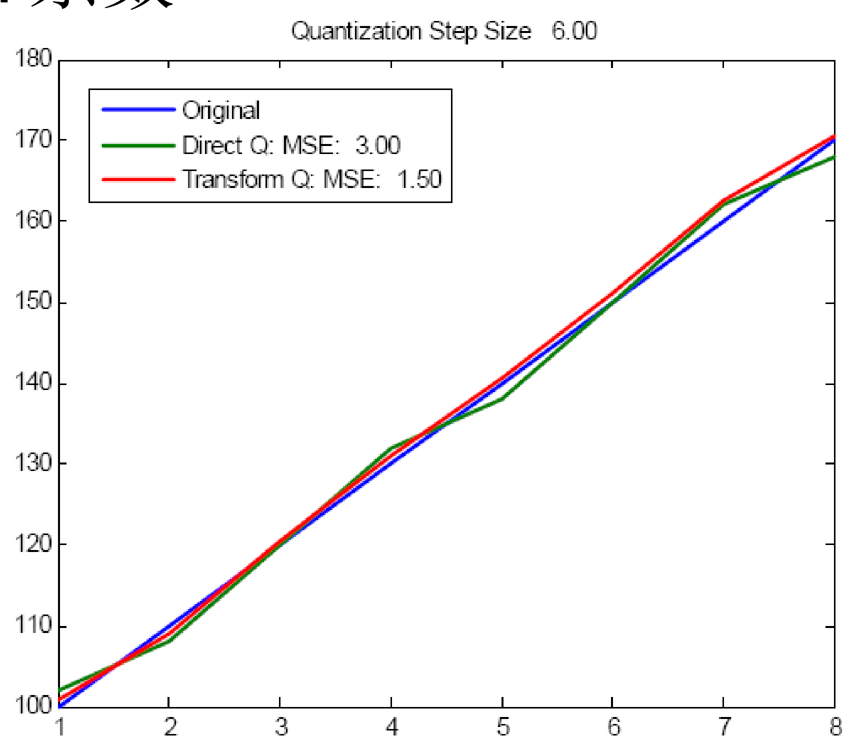
7水平的中平量化器



例6：空域量化 vs. 变换域量化

- 方案1：直接对原始数据进行量化
- 方案2：对DCT系数进行量化
  - △=6，量化后的DCT系数：[64 -11 0 -1 0 0 0 0]
  - 3个非0 DCT系数

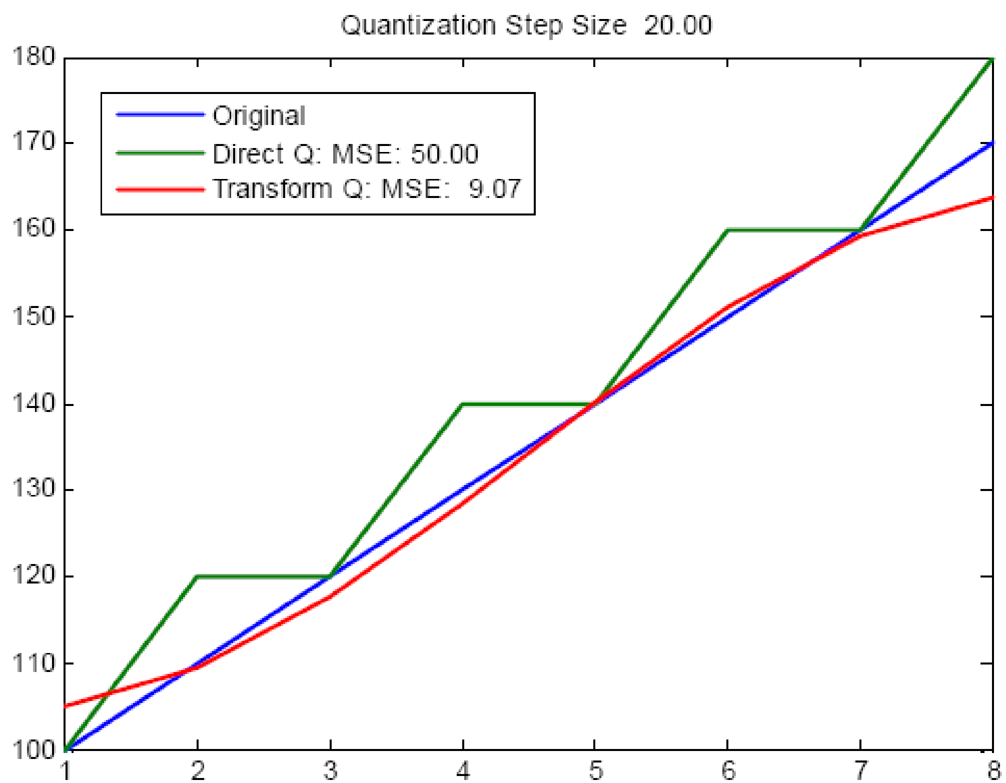
MSE:
w/o DCT: 3.0
w/ DCT: 1.5



例6：空域量化 vs. 变换域量化

- △=20， 2个非0 DCT系数：[19 -3 0 0 0 0 0 0]
  - DCT系数重构效果仍然很平滑
- 直接方法开始产生块/mosaic效应

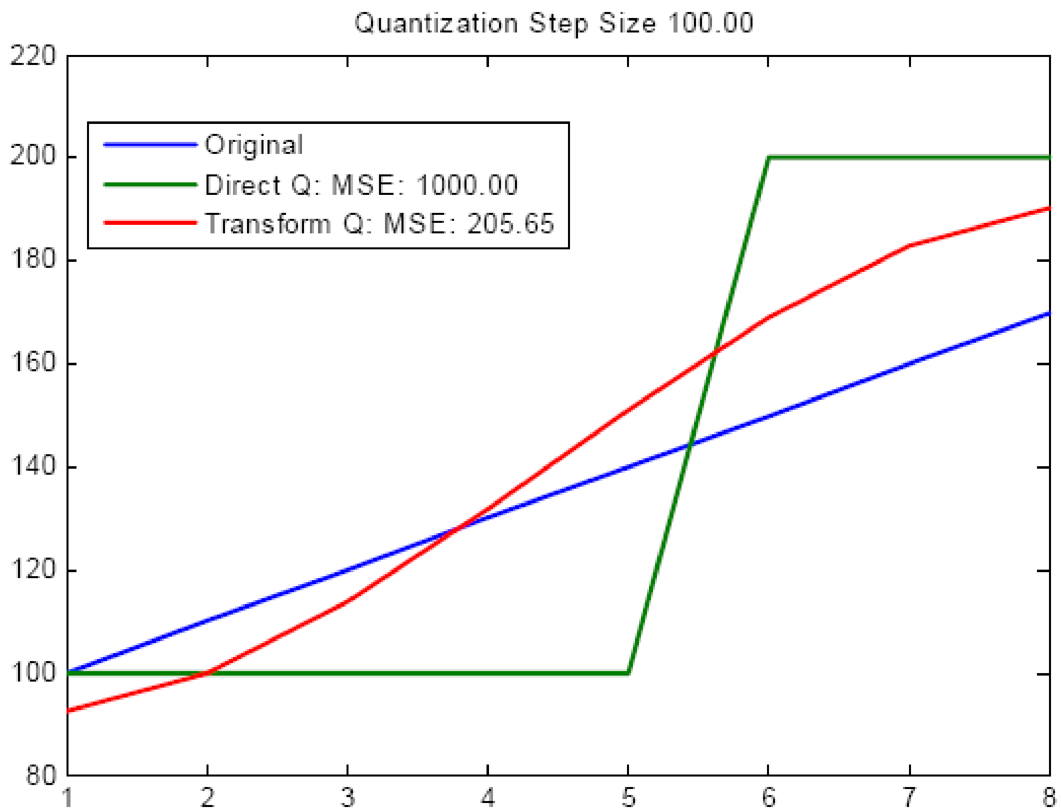
MSE:
w/o DCT: 50.0
w/ DCT: 9.07



例6：空域量化 vs. 变换域量化

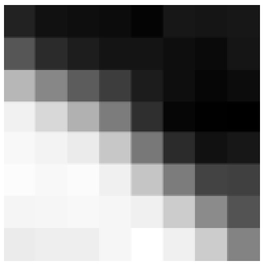
- △=100， 2个非0 DCT系数：[4 -1 0 0 0 0 0 0]
  - DCT系数重构效果仍然平滑
- 直接方法产生的块/mosaic效应更多

MSE:
w/o DCT: 1000
w/ DCT: 205



例6：空域量化 vs. 变换域量化

- 输入数据：



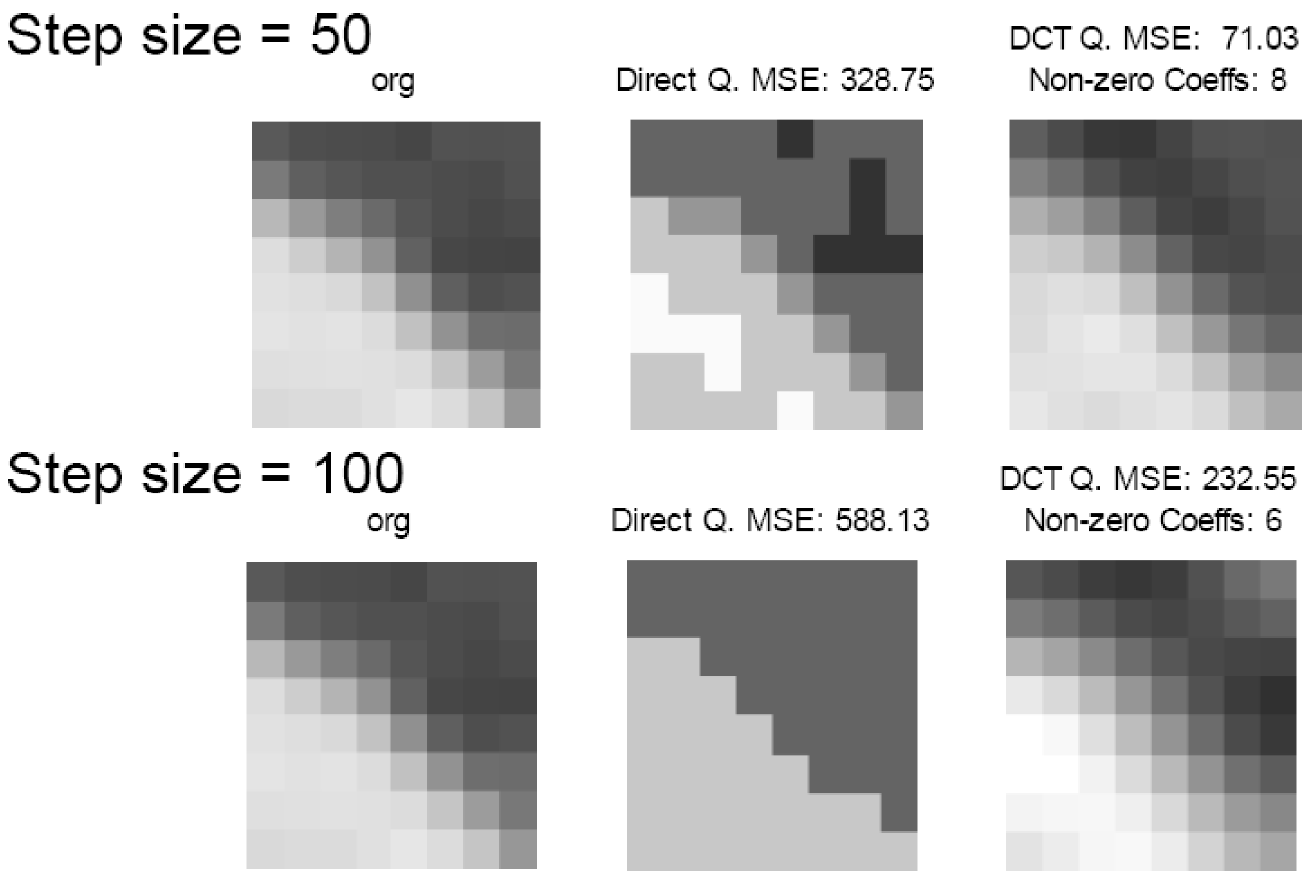
89	78	76	75	70	82	81	82
122	95	86	80	80	76	74	81
184	153	126	106	85	76	71	75
221	205	180	146	97	71	68	67
225	222	217	194	144	95	78	82
228	225	227	220	193	146	110	108
223	224	225	224	220	197	156	120
217	219	219	224	230	220	197	151

- 2-D DCT变换系数（取整）：

1155	259	-23	6	11	7	3	0
-377	-50	85	-10	10	4	7	-3
-4	-158	-24	42	-15	1	0	1
-2	3	-34	-19	9	-5	4	-1
1	9	6	-15	-10	6	-5	-1
3	13	3	6	-9	2	0	-3
8	-2	4	-1	3	-1	0	-2
2	0	-3	2	-2	0	0	-1

大多数能量集中在左上角

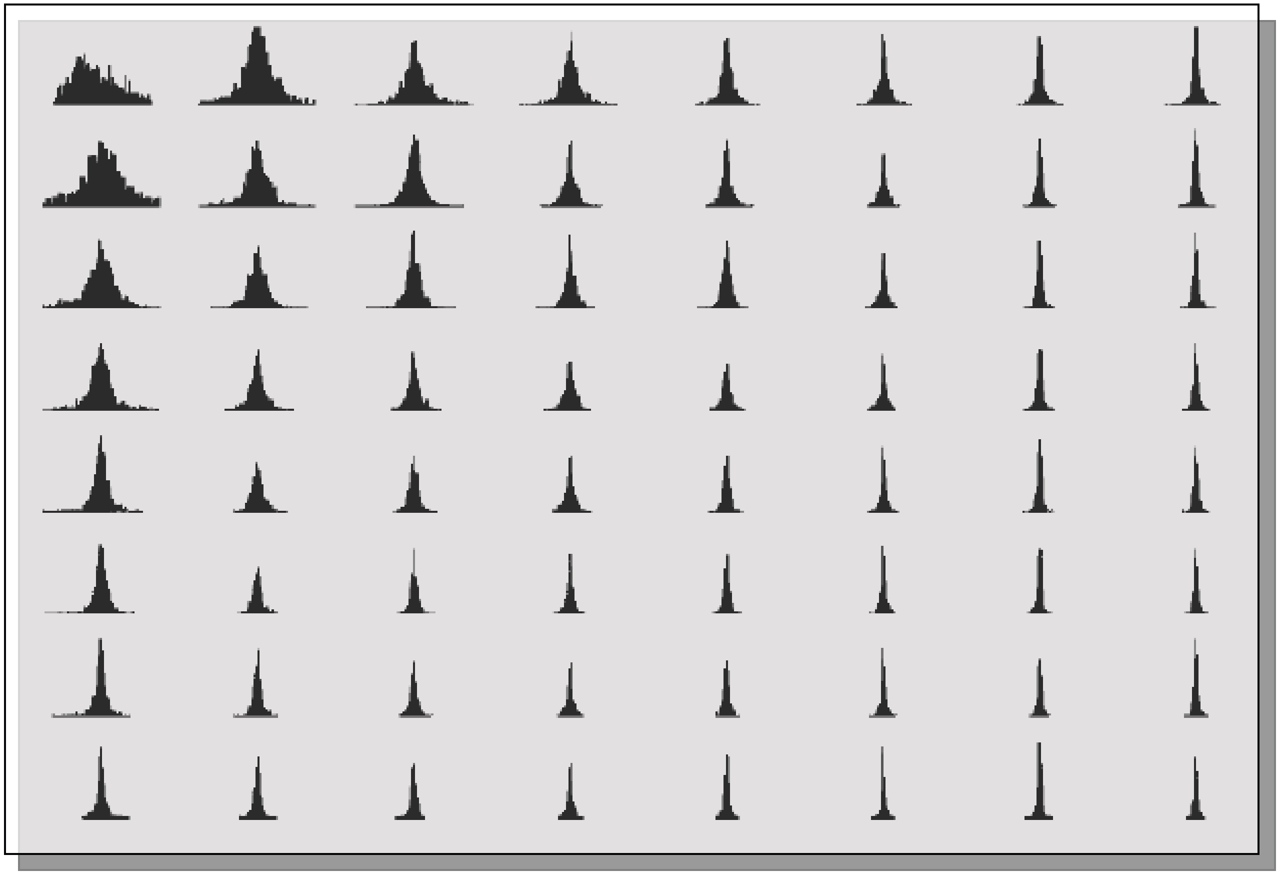
例6：空域量化 vs. 变换域量化



- 在变换域量化通常能得到更好的结果
- 我们还可以做得更好
  - 对不同的子带采取不同的量化步长

DCT系数的分布

Histograms for 8x8 DCT coefficient amplitudes measured for test image *[Lam, Goodman, 2000]*

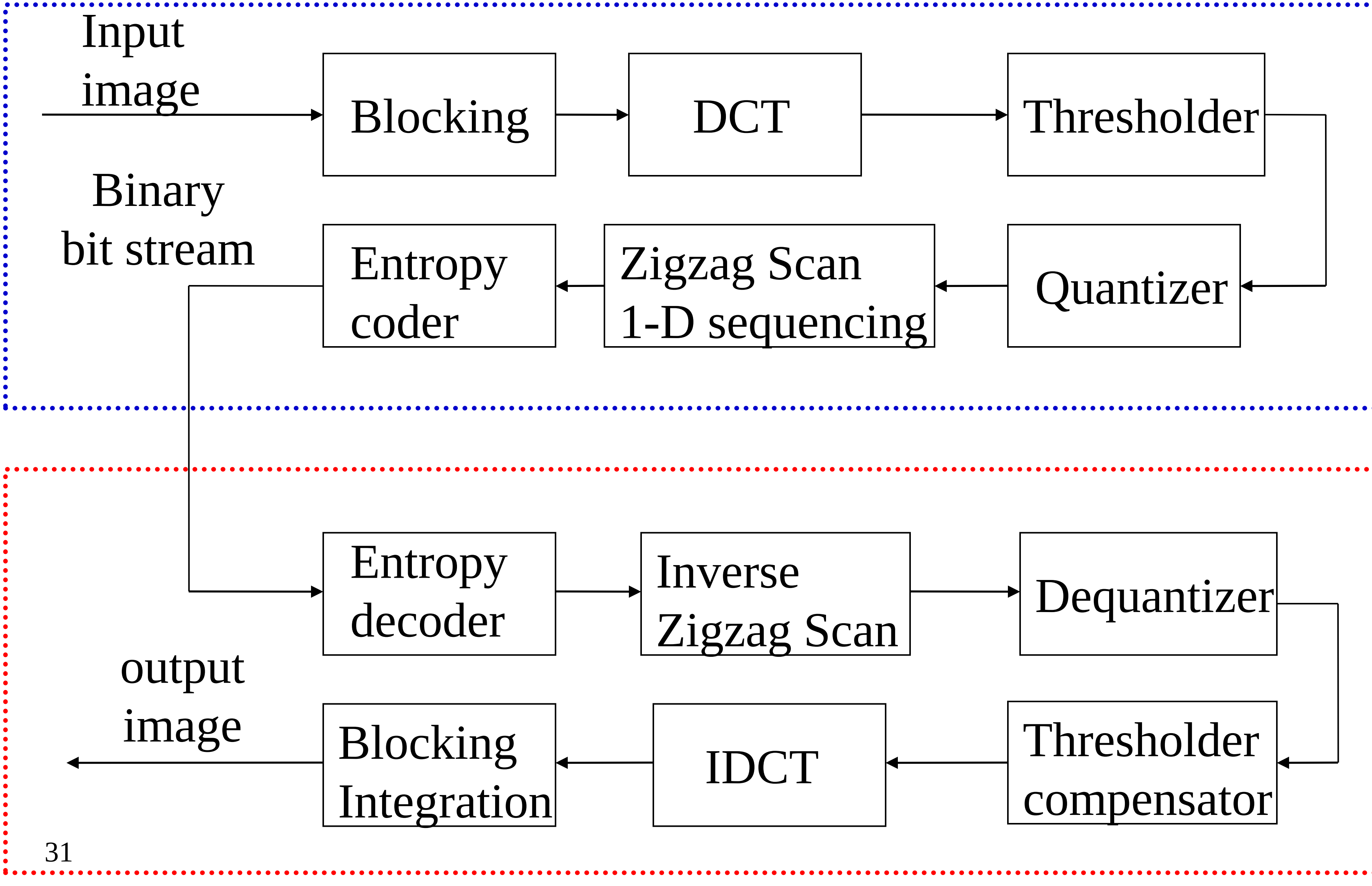


Test image
Bridge

AC coefficients: Laplacian PDF

DC coefficient distribution similar to the original image

A Typical Transform Coder



6.3 JPEG图像压缩标准

- JPEG：Joint Photographic Experts Group
- 正式名称：ISO/IEC JTC1/SC29/WG10

International Organization for Standardization

International Electrotechnical Commission

Joint ISO/IEC Technical Committee (Information Technology)

Subcommittee 29 (Coding of Audio, Picture, Multimedia and Hypermedia Information)

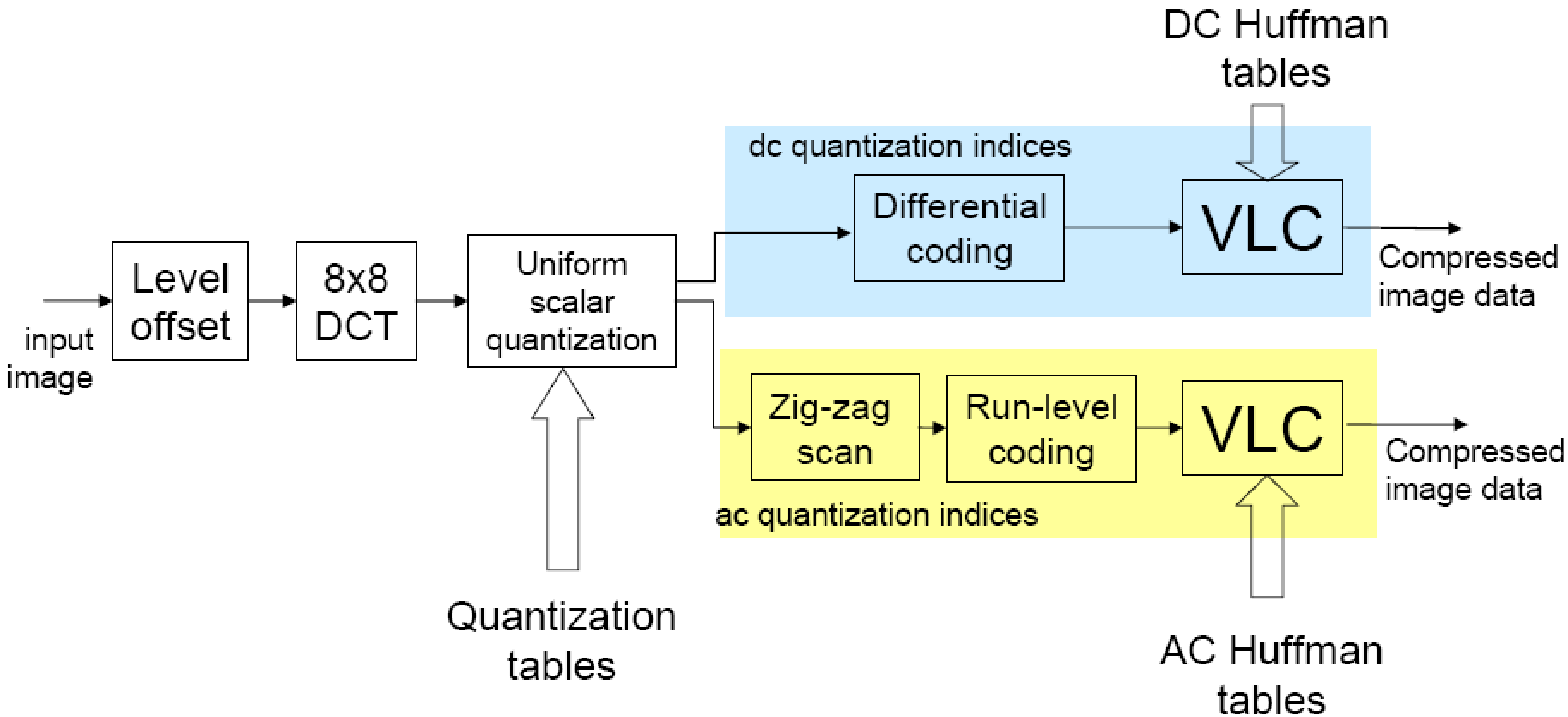
Working Group 10(JBIG, JPEG)

- 与CCITT （现为ITU）学习组VIII联合
- 工作开始于1986年
- 于1992年形成国际标准ISO/IEC 10918-1 和CCITT建议 T.81
- 广泛用于图像交换、WWW、数字图像

6.3 JPEG图像压缩标准

- JPEG规定了4种运行模式，以满足不同需要：
  - 基于DPCM的无损编码模式：压缩比可达2:1
  - 基于DCT的有损顺序编码模式：压缩比可达10:1以上
  - 基于DCT的递增编码模式
  - 基于DCT的分层编码模式

基本(baseline) JPEG编码器



- Huffman编码可以用自适应的二进制算术编码代替（很少产品支持）
  - 编码效率提高10%，但算法更复杂
  - 专利

颜色空间

- JPEG标准本身并没有规定具体的颜色空间，只是对各分量分别进行编码
- 实现中通常将高度相关RGB颜色空间转换到相关性较小的YUV颜色空间
- RGB→YUV（8bit/pixel）

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}$$

- YUV→RGB

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.164 & -0.002 & 1.596 \\ 1.164 & -0.391 & -0.813 \\ 1.164 & 2.018 & -0.001 \end{bmatrix} \begin{bmatrix} Y-16 \\ U-128 \\ V-128 \end{bmatrix}$$

颜色空间

- 颜色空间转换的整数实现

RGB to YUV:

$$Y = ((\ 66 * R + 129 * G + \ 25 * B + 128) >> 8) + 16$$
$$U = ((-38 * R \ - 74 * G + 112 * B + 128) >> 8) + 128$$
$$V = ((112 * R \ - 94 * G \ - 18 * B + 128) >> 8) + 128$$

YUV to RGB:

$$C = Y - 16$$
$$D = U - 128$$
$$E = V - 128$$

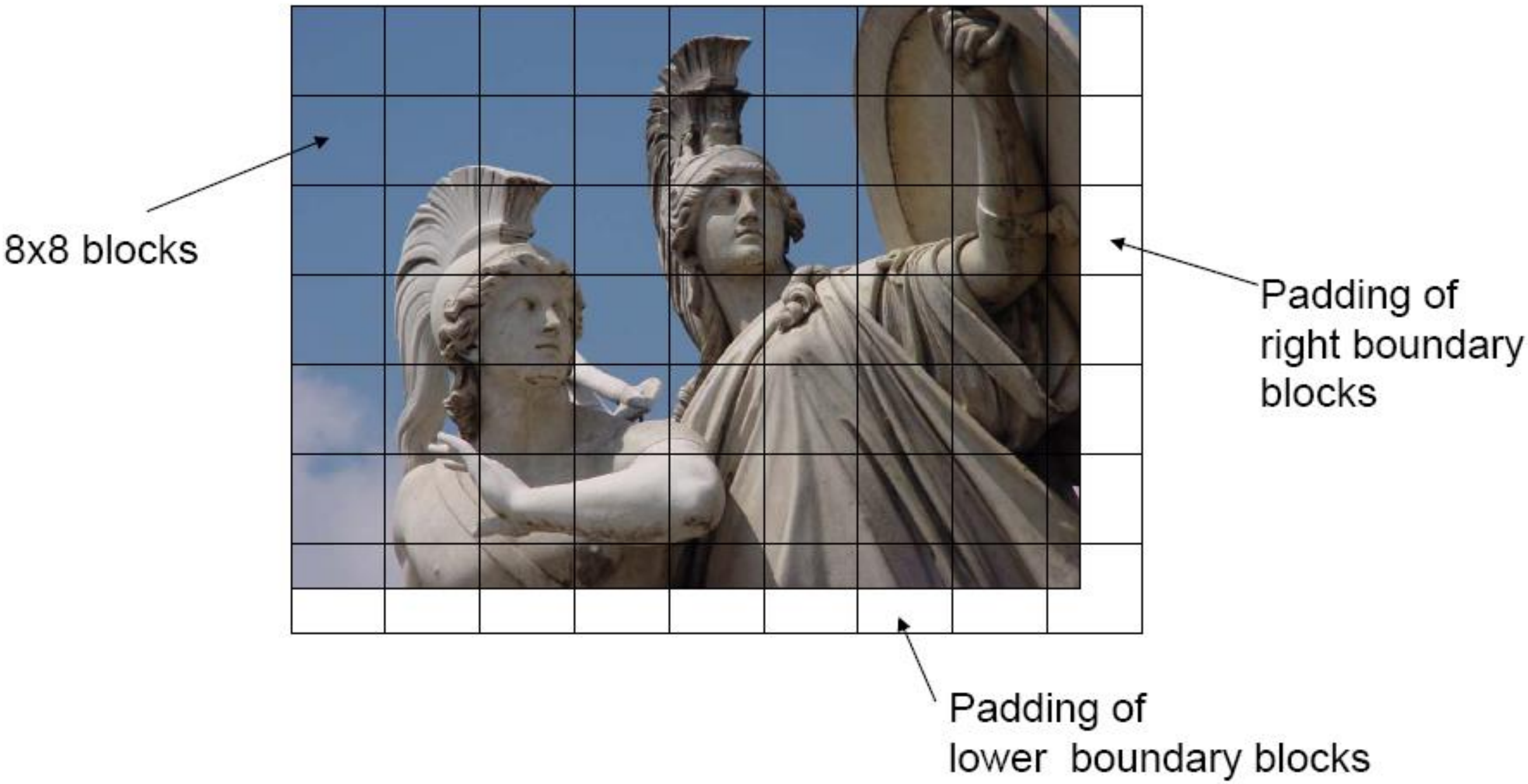
$$R = clip((298 * C + 409 * E + 128) >> 8)$$
$$G = clip((298 * C - 100 * D - 208 * E + 128) >> 8)$$
$$B = clip((298 * C + 516 * D + 128) >> 8)$$

clip( ): clipping the data to [0, 255].

颜色空间

- 图像的主要信息包括在Y通道
  - U、V更平滑→容易压缩
- 人眼对色度分量不敏感
  - 对色度分量可以进行下采样：如4：2：2

JPEG：8×8的块编码

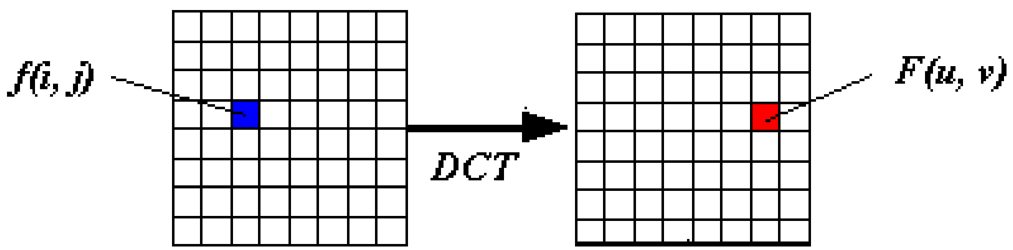

8x8 blocks
Padding of
right boundary
blocks
Padding of
lower boundary blocks

零偏置（Level Offset）

- 对于灰度级是2^n的像素，通过减去2^{n-1}，将无符号的整数值变成有符号数
  - 对于n=8，即将0~255的值域，通过减去128，转换为值域在-128~127之间的值
- 目的：使像素的绝对值出现3位10进制的概率大大减少

DCT变换

- 对每个单独的彩色图像分量，把整个分量图像分成8×8的图像块，如图所示，并作为两维离散余弦变换DCT的输入



DCT变换使用下式计算

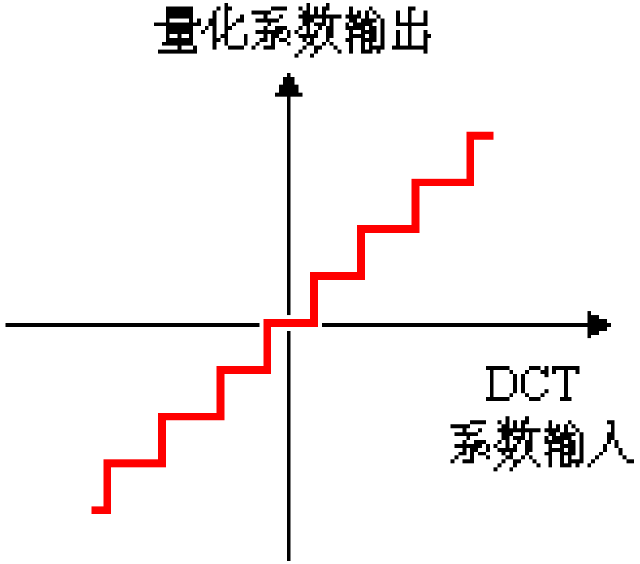
$$F(u,v)=\frac{1}{4}C(u)C(v)\left[\sum_{i=0}^{7}\sum_{j=0}^{7}f(i,j)\cos\frac{(2i+1)u\pi}{16}\cos\frac{(2j+1)v\pi}{16}\right]$$

逆变换使用下式计算

$$f(i,j)=\frac{1}{4}C(u)C(v)\left[\sum_{u=0}^{7}\sum_{v=0}^{7}F(u,v)\cos\frac{(2i+1)u\pi}{16}\cos\frac{(2j+1)v\pi}{16}\right]$$

其中 $\begin{cases} C(u),C(v)=1/\sqrt{2} & u,v=0 \\ C(u),C(v)=1 & otherwise \end{cases}$

量化

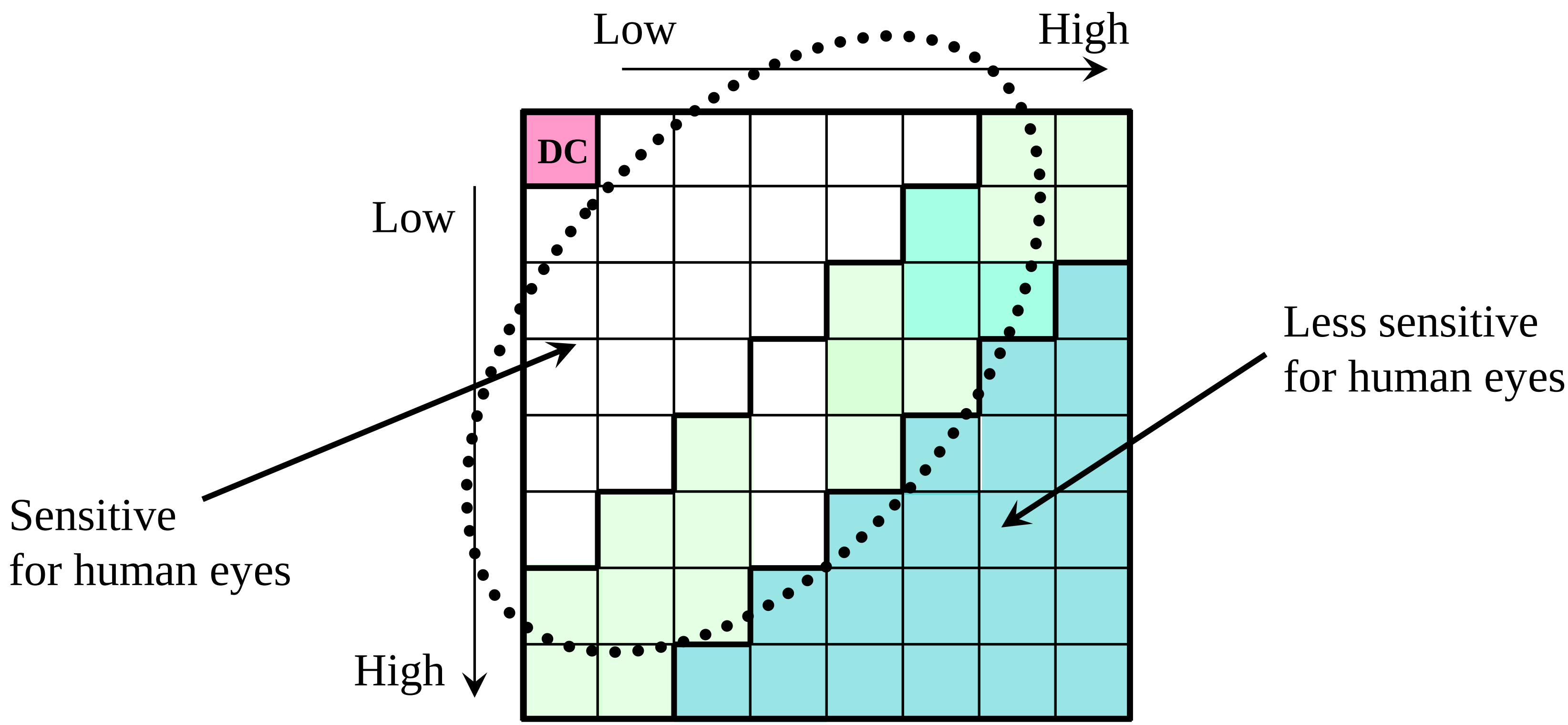


- 中平型均匀量化器
- 量化步距是按照系数
  - 所在的位置
  - 颜色分量

来确定

- 因为人眼对亮度信号比对色差信号更敏感，因此使用了两种量化表：亮度量化值和色差量化值
- 根据人眼的视觉特性（对低频敏感，对高频不太敏感）对低频分量采取较细的量化，对高频分量采取较粗的量化
  - 如果原始图象中细节丰富，则去掉的数据较多，量化后的系数与量化前差别
  - 反之，细节少的原始图象在压缩时去掉的数据少些

人眼的对亮度敏感性


Low
High
Low
High
DC
Less sensitive
for human eyes
Sensitive
for human eyes

建议量化表

- 基于人的生理感知阈值实验

- Luminance

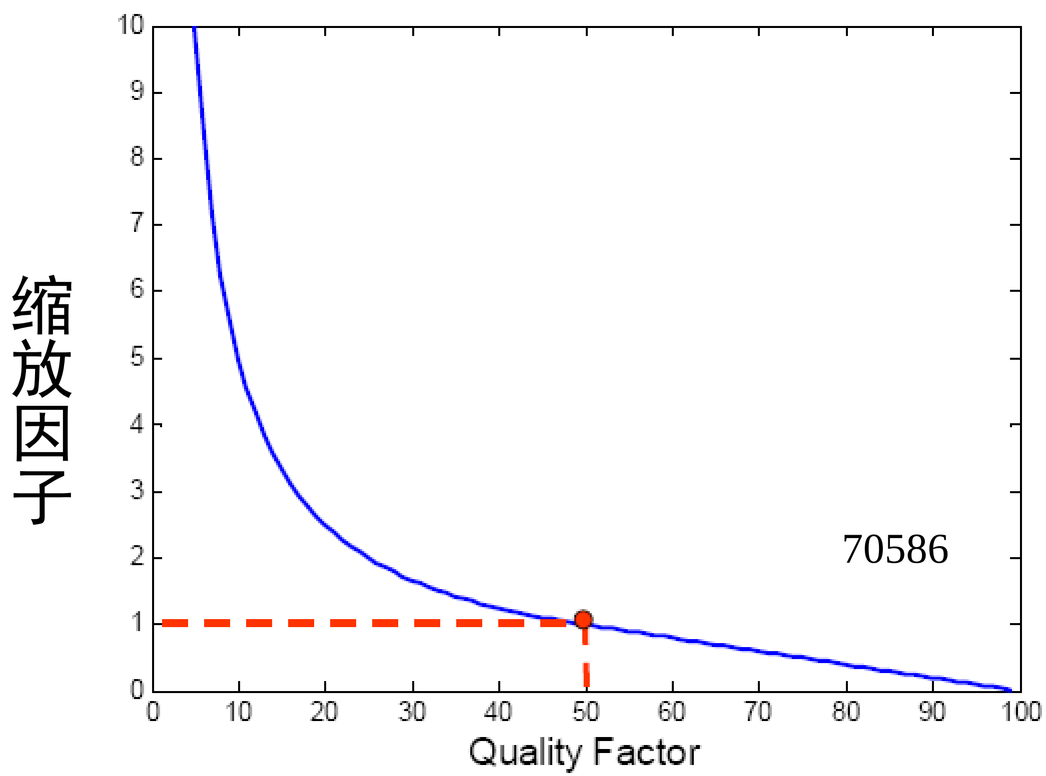
16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	36	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

- Chrominance, subsampled 2:1

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

量化表缩放

- 真正的量化表=缩放因子×基本量化表
  - 质量因子≤ 50：缩放因子= 50 / 质量因子;
  - 质量因子> 50：缩放因子 = 2 – 质量 因子/ 50



Quality Factor	Scaling
10	5.0
20	2.5
50	1.0
75	0.5

. 4

不同质量因子的图像示例

- GIF：258898 bytes

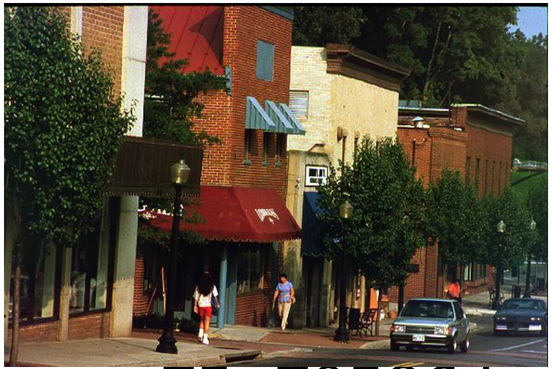
http://www.cs.sfu.ca/CC/365/mark/material/cgi-bin/whichjpeg.cgi

9438 bytes

10: 15325 bytes

25: 29360 bytes

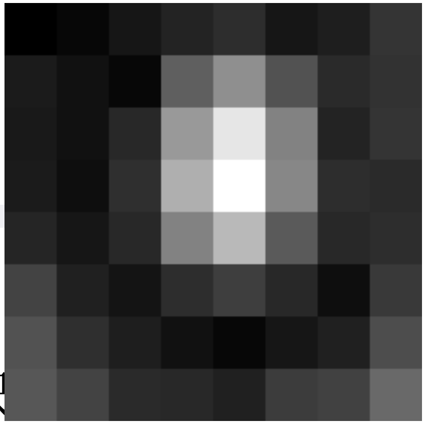
50: 46295 bytes



75: 70586 bytes

100: 326321bytes

例：JPEG编码过程



用8x8的JEPG Baseline标准，压缩并重构下列亮度块

52	55	61	66	70	61	64	73
63	59	66	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	68	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

例：JPEG编码过程

1、 0偏置转换后

-76	-73	-67	-62	-58	-67	-64	-55
-65	-69	-62	-38	-19	-43	-59	-56
-66	-69	-60	-15	16	-24	-62	-55
-65	-70	-57	-6	26	-22	-58	-59
-61	-67	-60	-24	-2	-40	-60	-58
-49	-63	-68	-58	-51	-65	-70	-53
-43	-57	-64	-69	-73	-67	-63	-45
-41	-49	-59	-60	-63	-52	-50	-34

正向DCT变换（N = 8）后变成

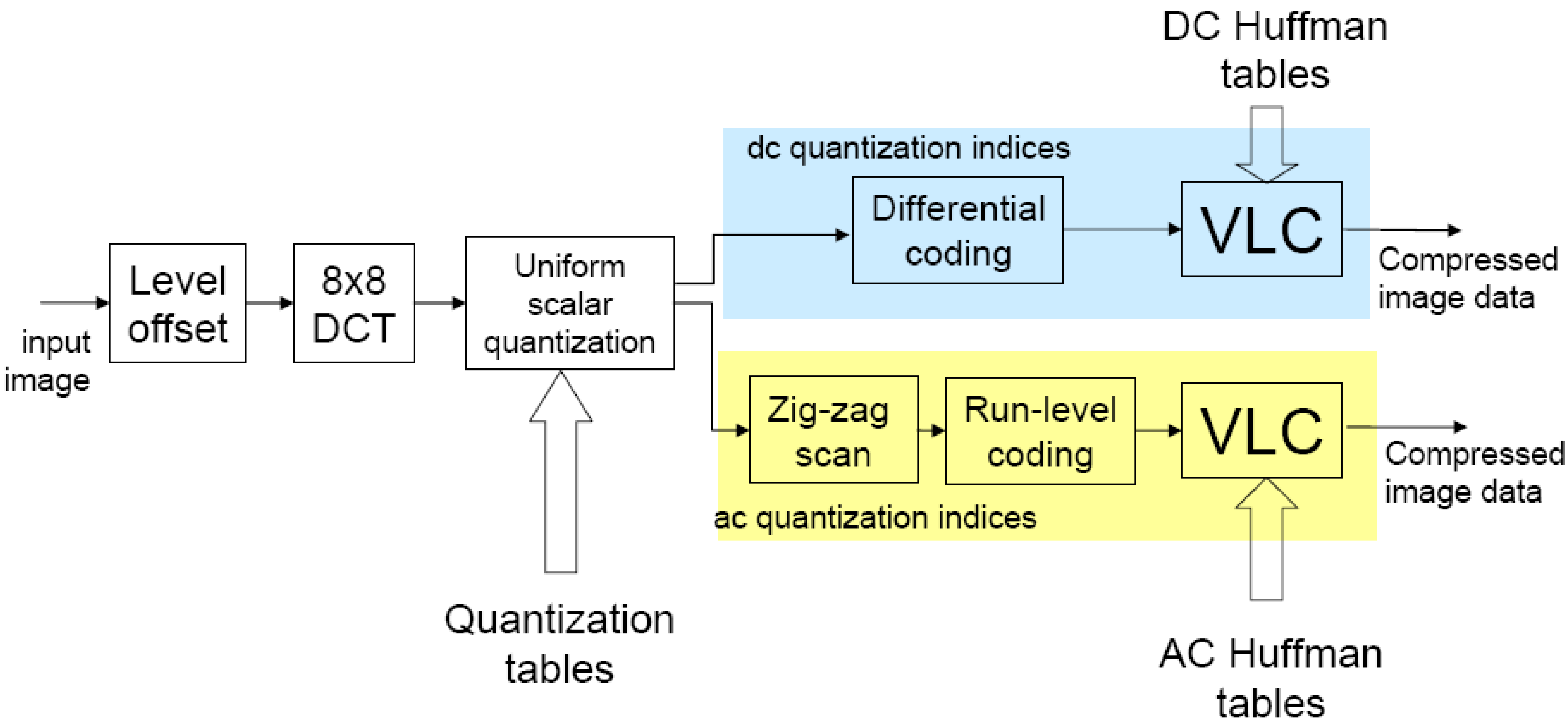
-415	-29	-62	25	55	-20	-1	3
7	-21	-62	9	11	-7	-6	6
-46	8	77	-25	-30	10	7	-5
-50	13	35	-15	-9	6	0	3
11	-8	-13	-2	-1	1	-4	1
-10	1	3	-3	-1	0	2	-1
-4	-1	2	-1	2	-3	1	-2
-1	-1	-1	-2	-1	-1	0	-1

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	36	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

量化变换后的数组

-26	-3	-6	2	2	0	0	0
1	-2	-4	0	0	0	0	0
-3	1	5	-1	-1	0	0	0
-4	1	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

基本(baseline) JPEG编码器



DC系数的差分编码

- 8×8图像块经过DCT变换之后得到的DC直流系数有两个特点
  - 系数的数值比较大
  - 相邻8×8图像块的DC系数值变化不大：冗余

- 根据这个特点，JPEG算法使用了差分脉冲调制编码(DPCM)技术，对相邻图像块之间量化DC系数的差值DIFF进行编码：

$$DIFF_k = DC_k - DC_{k-1}$$

DC系数的差分编码

- 对DIFF用Huffman编码：分成类别，类似指数Golomb编码
  - 类别ID：一元码编码
  - 类内索引：采用定长码

Ranges	Range Size	DC Cat. ID	AC Cat. ID
0	1	0	N/A
-1, 1	2	1	1
-3, -2, 2, 3	4	2	2
-7, -6, -5, -4, 4, 5, 6, 7	8	3	3
-15, …, -8, 8, …, 15	16	4	4
-31, …, -16, 16, …, 31	32	5	5
-63, …, -32, 32, …, 63	64	6	6
…	…	…	…
[-32767, -16384], [16384, 32767]	32768	15	15

DC系数的差分编码

$$DIFF_k = DC_k - DC_{k-1}$$

DC Cat.	Prediction Errors	Codeword
0	0	010
1	-1, 1	011x
2	-3, -2, 2, 3	100xx
3	-7, -6, -5, -4, 4, 5, 6, 7	00xxx
4	-15, …, -8, 8, …, 15	101xxxx
5	-31, …, -16, 16, …, 31	110xxxxx
6	-63, …, -32, 32, …, 63	1110xxxxxx
…	…	…

xxx: fixed length index within each category.

例：DC=8，上一DC=5，则DIFF=8-5=3
类别ID=2，类内索引=3，则码流=10011

AC系数的Z字扫描

- 由于经DCT变换后，系数大多数集中在左上角，即低频分量区，因此采用Z字形按频率的高低顺序读出，可以出现很多连零的机会。可以使用游程编码。尤其在最后，如果都是零，给出EOB (End of Block)即可。

Zig-zag 扫描

231	-74	-12	-1	-1	-2	-1	0
-102	-5	1	0	1	0	0	0
-13	7	0	0	0	0	0	0
-4	0	0	-1	0	0	0	0
0	-1	-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

EOB (End Of Block)

EOB is transmitted instead of zeros

231	-74	-12	-1	-1	-2	-1	0
-102	-5	1	0	1	0	0	0
-13	7	0	0	0	0	0	0
-4	0	0	-1	0	0	0	0
0	-1	-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

EOB

AC系数的游程编码

- 在JPEG和MPEG编码中规定为：(run, level)
  - 表示连续run个0，后面跟值为level的系数
  - 如：0，2，0，0，3，0，-4，0，0，0，-6，0，0，5，7
  - 表示为(1, 2)，(2, 3)，…
- 编码：
  - Run: 最多15个，用4位表示RRRR
  - Level：类似DC
    - 分成16个类别，用4位表示SSSS表示类别号
    - 类内索引
  - 对(RRRR, SSSS)联合用Huffman编码
  - 对类内索引用定长码编码

AC系数的Huffman编码

RRRR/SSSS	Base codeword	RRRR/SSSS	Base Codeword	…	RRRR/SSSS	Base codeword
EOB	1010	-	-	…	ZRL	1111 1111 001
0/1	00	1/1	1100	…	15/1	1111 1111 1111 0101
0/2	01	1/2	11011	…	15/2	1111 1111 1111 0110
0/3	100	1/3	1111001	…	15/3	1111 1111 1111 0111
0/4	1011	1/4	111110110	…	15/4	1111 1111 1111 1000
0/5	11010	1/5	11111110110	…	15/5	1111 1111 1111 1001
…	…	…	…	…	…	…

- **ZRL：表示16个0**
- **0，2，0，0，3，0，-4，0，0，0，-6，0，0，5，7,…**

解码重构

- 与编码相反
  - 解码Huffman数据
  - 解码DC差值
  - 重构量化后的系数
  - DCT逆变换
  - 丢弃填充的行/列
  - 反0偏置
  - 对丢失的CbCr分量差值（下采样的逆过程）
  - YCbCr → RGB

解码重构

- 量化后的系数（已通过DC差重构DC系数）

$$\begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -3 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- 乘以量化表，得到 $\begin{bmatrix} -416 & -33 & -60 & 32 & 48 & -40 & 0 & 0 \\ 0 & -24 & -56 & 19 & 26 & 0 & 0 & 0 \\ -42 & 13 & 80 & -24 & -40 & 0 & 0 & 0 \\ -56 & 17 & 44 & -29 & 0 & 0 & 0 & 0 \\ 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$

- IDCT:

$$\begin{bmatrix} -68 & -65 & -73 & -70 & -58 & -67 & -70 & -48 \\ -70 & -72 & -72 & -45 & -20 & -40 & -65 & -57 \\ -68 & -76 & -66 & -15 & 22 & -12 & -58 & -61 \\ -62 & -72 & -60 & -6 & 28 & -12 & -59 & -56 \\ -59 & -66 & -63 & -28 & -8 & -42 & -69 & -52 \\ -60 & -60 & -67 & -60 & -50 & -68 & -75 & -50 \\ -54 & -46 & -61 & -74 & -65 & -64 & -63 & -45 \\ -45 & -32 & -51 & -72 & -58 & -45 & -45 & -39 \end{bmatrix}$$

- +128：

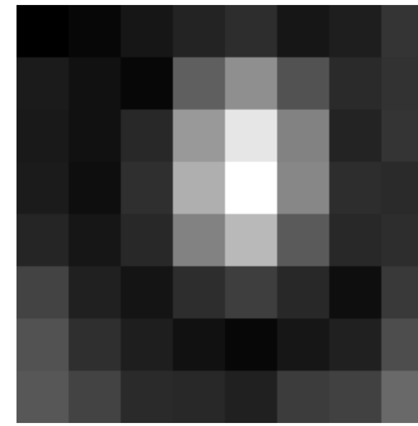
$$\begin{bmatrix} 60 & 63 & 55 & 58 & 70 & 61 & 58 & 80 \\ 58 & 56 & 56 & 83 & 108 & 88 & 63 & 71 \\ 60 & 52 & 62 & 113 & 150 & 116 & 70 & 67 \\ 66 & 56 & 68 & 122 & 156 & 116 & 69 & 72 \\ 69 & 62 & 65 & 100 & 120 & 86 & 59 & 76 \\ 68 & 68 & 61 & 68 & 78 & 60 & 53 & 78 \\ 74 & 82 & 67 & 54 & 63 & 64 & 65 & 83 \\ 83 & 96 & 77 & 56 & 70 & 83 & 83 & 89 \end{bmatrix}$$

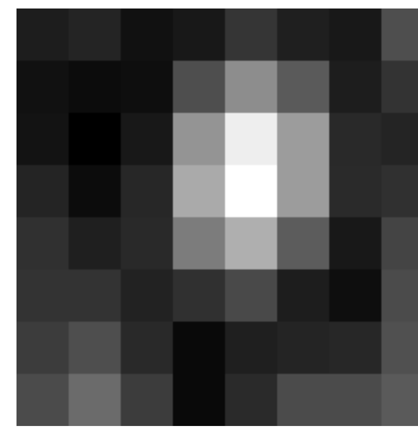
- 重构误差：

$$
\begin{bmatrix}
-8 & -8 & 6 & 8 & 0 & 0 & 6 & -7 \\
5 & 3 & -1 & 7 & 1 & -3 & 6 & 1 \\
2 & 7 & 6 & 0 & -6 & -12 & -4 & 6 \\
-3 & 2 & 3 & 0 & -2 & -10 & 1 & -3 \\
-2 & -1 & 3 & 4 & 6 & 2 & 9 & -6 \\
11 & -3 & -1 & 2 & -1 & 8 & 5 & -3 \\
11 & -11 & -3 & 5 & -8 & -3 & 0 & 0 \\
4 & -17 & -8 & 12 & -5 & -7 & -5 & 5
\end{bmatrix}
$$

- 每个像素大约为5的平均绝对误差
  - 误差在左下角比较明显

原图





重构图

示例图像

0.56 bits/pixel

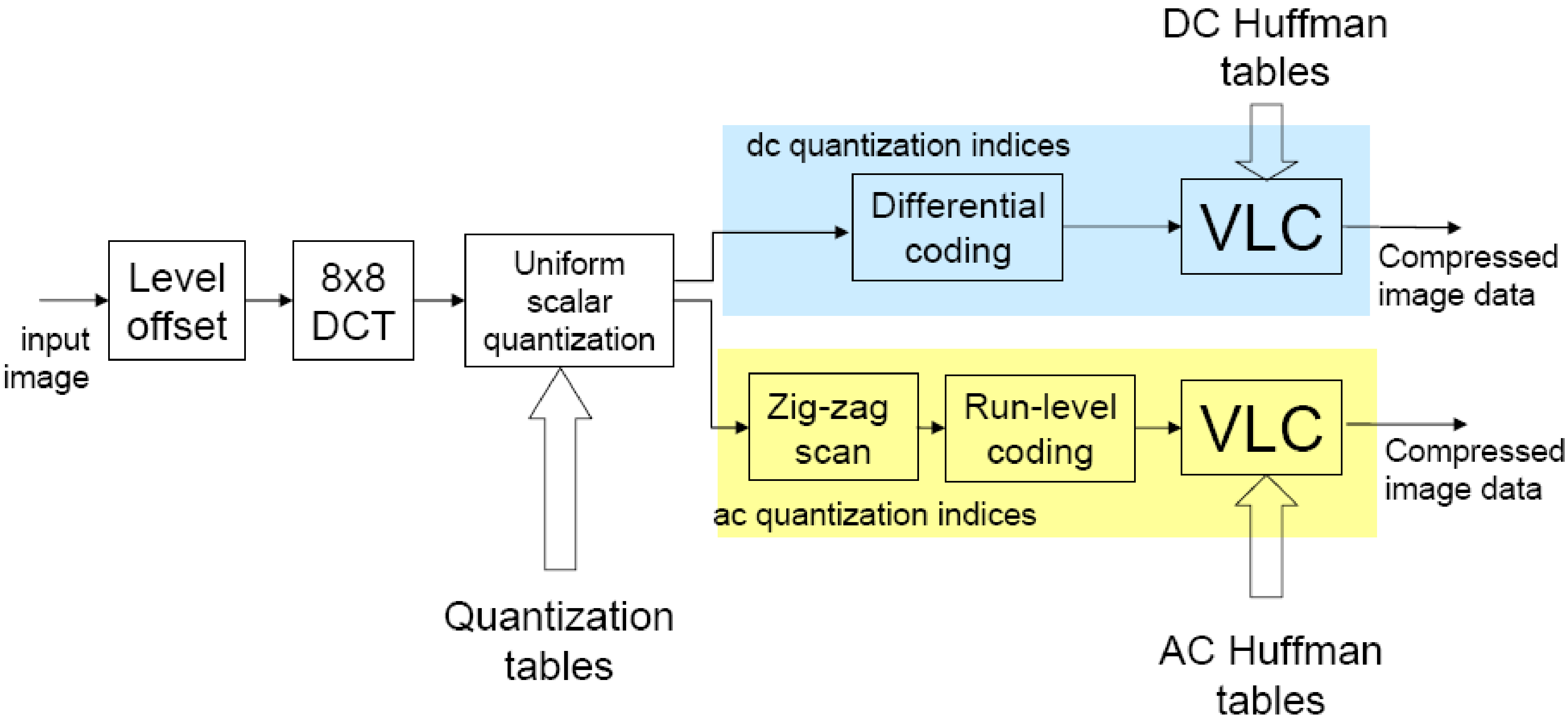
0.14 bits/pixel

示例图像

0.07 bits/pixel

0.035 bits/pixel

JPEG Baseline 系统分析



- 两张量化表 – 亮度和色度
- 一般四张Huffman码表
  – 亮度DC，亮度AC，色度DC，色度AC

如何保存

JPEG标准中的语法结构

- 定义了一系列标记（Markers）
  - 均以0xFF开始，后跟1字节的标记标识符和2字节的标记长度以及该标记所对应的payload
  - 标记长度部分高位在前，低位在后，不包含该标记的头两个字节
  - 熵编码部分的数据在0xFF后由编码器插入0x00，解码器解码时跳过此字节不予处理
  - SOI(Start Of Image)和EOI(End Of Image)标记没有payload

Common JPEG markers

Short name	Bytes	Payload	Name	Comments
SOI	0xFFD8	*none*	Start Of Image	
SOF0	0xFFC0	*variable size*	Start Of Frame (Baseline DCT)	Indicates that this is a baseline DCT-based JPEG, and specifies the width, height, number of components, anc subsampling (e.g., 4:2:0).
SOF2	0xFFC2	*variable size*	Start Of Frame (Progressive DCT)	Indicates that this is a progressive DCT-based JPEG, and specifies the width, height, number of components, subsampling (e.g., 4:2:0).
DHT	0xFFC4	*variable size*	Define Huffman Table (s)	Specifies one or more huffman tables.
DQT	0xFFDB	*variable size*	Define Quantization Table(s)	Specifies one or more quantization tables.
DRI	0xFFDD	2 bytes	Define Restart Interval	Specifies the interval between RST*n* markers, in macroblocks.
SOS	0xFFDA	*variable size*	Start Of Scan	Begins a top-to-bottom scan of the image. In baseline DCT JPEG images, there is generally a single scan. Prc JPEG images usually contain multiple scans. This marker specifies which slice of data it will contain, and is ir followed by entropy-coded data.
RST*n*	0xFFD*n*	*variable size*	Restart	Inserted every *r* macroblocks, where *r* is the restart interval set by a DRI marker. Not used if there was no DRI low 4 bits of the marker code, cycles from 0 to 7.
APP*n*	0xFFE*n*	*variable size*	Application-specific	For example, an Exif JPEG file uses an APP1 marker to store metadata, laid out in a structure based closely
COM	0xFFFE	*variable size*	Comment	Contains a text comment.
EOI	0xFFD9	*none*	End Of Image	

JPEG文件的解析

- 由若干个必不可少的标记顺序连接构成整个文件
- 一定以0xFFD8开始，即表示图像开始SOI (Start of Image)标记
- 一定以0xFFD9结束，表示图像结束EOI(End of Image)标记

二、APP0 Segment (0xFFE0)		
2字节	APP0块的长度	
5字节	“JFIF”+”0”	
1字节	<Major version>	
1字节	<Minor version>	
1字节	<Units for the X and Y densities>	Units=0无单位 Units=1 DPI Units=2 点/厘米
2字节	<X density>	水平方向像素密度
2字节	<Y density>	垂直方向像素密度
1字节	<X thumbnail>	缩略图水平像素数目
1字节	<Y thumbnail>	缩略图垂直像素数目
3n	<Thumbnail RGB bitmap>	缩略RGB位图，n为像素数

三、APPn标记，数值0xFF E1~0xFFEF

- N=1~15,数值对应0xE1~0xEF
- APPn长度（Length）
- 应用细节信息（Application specific information）
- EXIF文件格式中多写入0xFF E1

四、量化表DQT，数值0xDB

- 一般为两个量化表，即亮度和色度各一张
- 以0xFFDB开始
  - 量化表长度，一般为00 43（或00 84）
  - 量化表信息（1字节）
    - Bit 0~3　QT号（只能取值为0~3，否则错误）
    - Bit 4~7　QT精度（0为8比特，否则表示16比特）
  - 量化表的实际数据
    - 量化表中的数据按照Z字形保存量化表内8x8的数据

五、帧图像开始SOF0,数值0xFF C0

SOF长度(2字节)	0x00 11
精度(1字节)	每个颜色分量每个像素的位数（通常为8）
图像高度(2字节)	以像素数表示图像的高度
图像宽度(2字节)	以像素数表示图像的宽度
颜色分量数(1字节)	通常为3
对每个颜色分量：	颜色索引ID(1字节,从01,02,03)
	Sample factor(1字节，高四位水平因子，低四位垂直因子)
	量化表号(1字节)

六、一个或多个Huffman表，数值0xFF C4

- Huffman表的长度
- 类型（AC或DC）
- 索引（Index）
- 位表（bit table）
- 值表（value table）

Huffman表存储方式举例说明

- FF C4 00 1D 00 00 03 01 01 01 01 01 01 01 00 00 00 00 00 00 04 05 06 03 02 01 00 09 07 08
- 红色部分 为哈夫曼表ID和表类型，其值0x00表示此部分数据描述的是DC直流0号表。
- 蓝色部分（16个字节）为不同位数的码字的数量。这16个数值实际意义为：没有1位的哈夫曼码字；2位的码字有3个；3位-9位的码字各有1个；没有9位或以上的码字。
- 绿色部分（10个字节）为编码内容。由蓝色部分数据知道，此哈夫曼树有0+3+1+1+1+1+1+1+1=10个叶子结点，即本字段应该有10个字节。这段数据表示10个叶子结点按从小到大排列，其权值依次为04、 05、 06、 03、 02、 01、00、09、 07、 08 （16进制）
- 见JPEG和Trace文件。

Huffman表存储方式举例说明

- FF C4 00 3E 10 00 01 02 05 03 03 03 02 05 03 04 02 02 02 01 05 00 01 03 02 04 05 11 21 22 31 61 06 12 A1 32 41 62 13 51 23 42 71 81 91 15 52 63 07 14 33 53 16 43 08 B1 34 C1 24 D1 09 72 F0 A2
- 红色部分（1字节）为哈夫曼表ID和表类型，其值0x10表示此部分数据描述的是AC交流0号表。
- 蓝色部分（16字节）为不同位数的码字的数量。这16个数值实际意义为：没有1位的哈夫曼码字……。
- 绿色部分（3E-16-2-1=43字节）为编码内容。由蓝色部分数据知道，此哈夫曼树有（绿色数据相加）=43个叶子结点，即本字段应该有43个字节。这段数据表示43个叶子结点按从小到大排列，其权值依次为（16进制）
- 看具体JPEG和Trace文件。

建立huffman树/表

- 在读出哈夫曼表的数据后，就要建立哈夫曼树。具体方法为：

1）第一个码字必定为0。
如果第一个码字位数为1，则码字为0；
如果第一个码字位数为2，则码字为00；
如此类推。

2）从第二个码字开始，
如果它和它前面的码字位数相同，则当前码字为它前面的码字加1；
如果它的位数比它前面的码字位数大，则当前码字是前面的码字加1后再在后边添若干个0，直至满足位数长度为止。

建立huffman树/表

- 利用第一个例子:
- FF C4 00 1D 00 00 03 01 01 01 01 01 01 01 00 00 00 00 00 00 04 05 06 03 02 01 00 09 07 08

序号	码长	码字	权值
1	2	00	04
2	2	01	05
3	2	10	06
4			
5	3	110	03
6	4	1110	02
7	5	11110	01
8	6	111110	00
9	7	1111110	09
10	8	11111110	07
	9	111111110	08

码字: 加一(补**0**)

第一个码字必定为**0**，当前码长有两位

01表示只有一个码长只有**1**个码字

由**00 03**得到前三个码字码长都为**2**

权值顺序写入

建立huffman树/表（直流）

- 刚才建立的是DC(0)huffman表，其权值就是解码时**再需要读入的bit位数**。这个再次读入的位数通过查表得到真正的码值。

  例如：0110101011

根据刚建立的huffman表分解： 01,10101011.

码字01 的权值为05.则再读取5位。 01,10101，011

这5位10101进行译码为：21. 表示直流系数为21.

注意：直流系数是差分编码的

直流系数的差分编码

- **直流系数的差分编码**
- 把所有的颜色分量单元按颜色分量（Y、Cr、Cb）分类。每一种颜色分量内，相邻的两个颜色分量单元的直流变量是以差分来编码的。也就是说，通过步骤3解码出来的直流变量数值只是当前颜色分量单元的实际直流变量减去前一个颜色分量单元的实际直流变量。也就是说，当前直流变量要通过前一个颜色分量单元的实际（非解码）直流分量来校正：
- DCn=DCn-1+Diff
- 其中Diff为差分校正变量，也就是直接解码出来的直流系数。但如果当前颜色分量单元是第一个单元，则解码出来的直流数值就是真正的直流变量。

建立huffman树/表（交流）

- 对于交流系数，用交流哈夫曼树/表查得该码字对应的权值。权值的高4位表示当前数值前面有多少个连续的零，低4位表示该交流分量数值的二进制位数，也就是接下来需要读入的位数。
- 例如：权值为0X31.可表示为（3，1）。表明此交流系数前有3个0，而此交流系数的具体值还需要再读入1个bit的码字，才能得到。
- 具体见Trace文件。

解码一个JPEG文件

- Start of Image SOI
  - Set image started to true.
- An APP0 Marker
  - Get the APP0 length.
  - Check if the Identifier is JFIF
  - Skip over any extensions.
  - Get the Version (This will be ignored)
  - Get the Units for X & Y densities (This will be ignored)
  - Get the X density (This will be ignored)
  - Get the Y density (This will be ignored)
  - Get the Thumbnail horizontal pixels
  - Get the Thumbnail vertical pixels
  - Skip over Thumbnail RGB bitmap, if there is one.
  - Skip to the end of this section if we are not already there.

解码一个JPEG文件

- Quantization table DQT
  - Get the Quantization table length.
  - Get the Quantization table number.
    - This will be used to determine which component will use this table. See Start of Frame.
  - Get the 64 entries in the Quantization table.
- Start of Frame SOF0
  - Get the Start of Frame length
  - Check that the Precision is 8 Bits per pixel per color component.
  - Get the Image height
  - Get the Image width
  - Check that the Number of color components is 3.
  - For each component
    - Get An ID (This will be ignored)
    - Get A vertical sample factor (Low Nibble)
    - Get A horizontal sample factor (High Nibble)
    - Get A quantization table# See Quantization table above.

解码一个JPEG文件

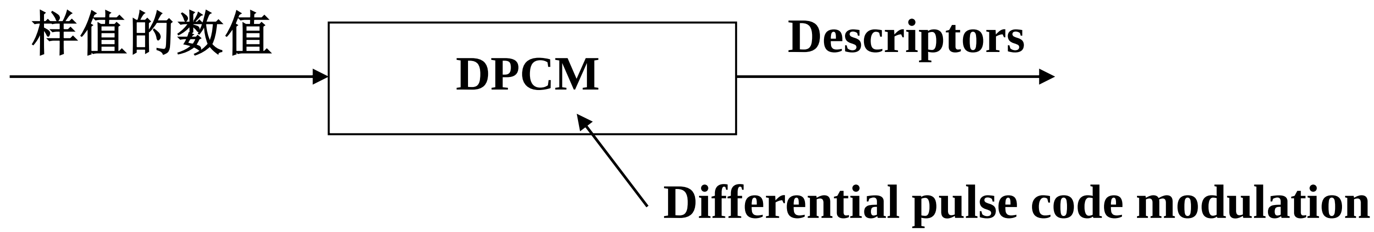
- Huffman table DHT
  - Get the Huffman table length.
  - Get the Type, AC or DC (1 -> AC)
    - Setup table pointer to table of correct type , AC or DC.
  - Get the Index.
    - This will be used to determine which component will use this table. See Start of Scan.
  - Get the Bits table & add up the entries to determine how many entries are in the value table.
  - Get the Value table.

JPEG文件格式http://en.wikipedia.org/wiki/JPEG

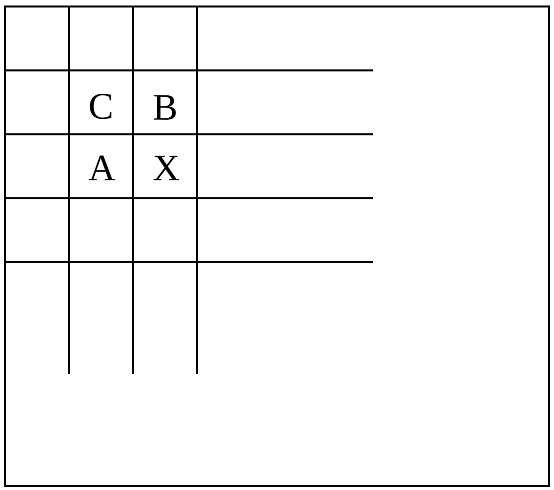
- JPEG Interchange Format' (JIF)
  - specified in Annex B of the standard
  - 考虑所有的模式，过于复杂，很少应用
- JPEG File Interchange Format (JFIF)
  - 是JIF的缩减版本
  - 由C-Cube Microsystems开发
- Exchangeable image file format(EXIF)
  - 由Japan Electronic Industries Development Association开发
  - 保存了丰富的metadata

JPEG 无损模型

采用三邻域二维预测编码和熵编码



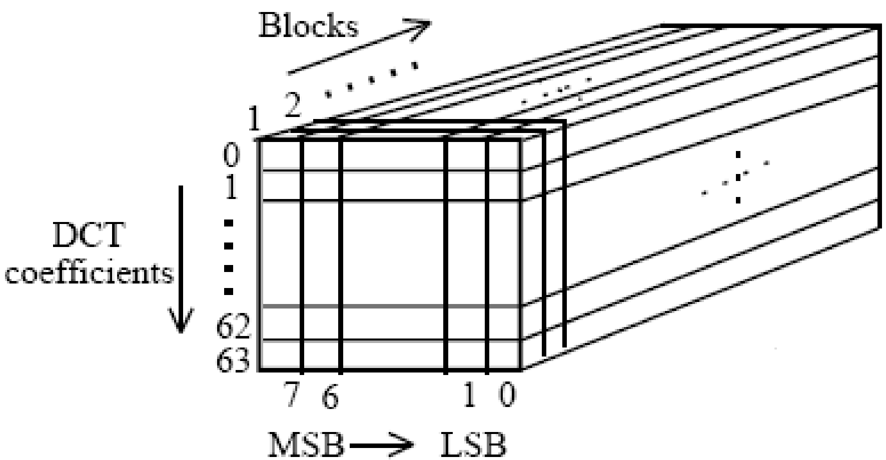
Predictors for lossless coding	
selection value	prediction strategy
0	no prediction
1	A
2	B
3	C
4	A+B-C
5	A+(B-C)/2
6	B+(A-C)/2
7	(A+B)/2

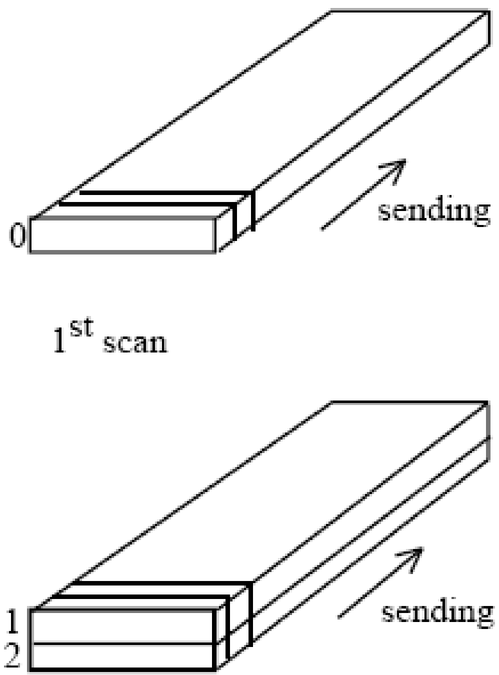


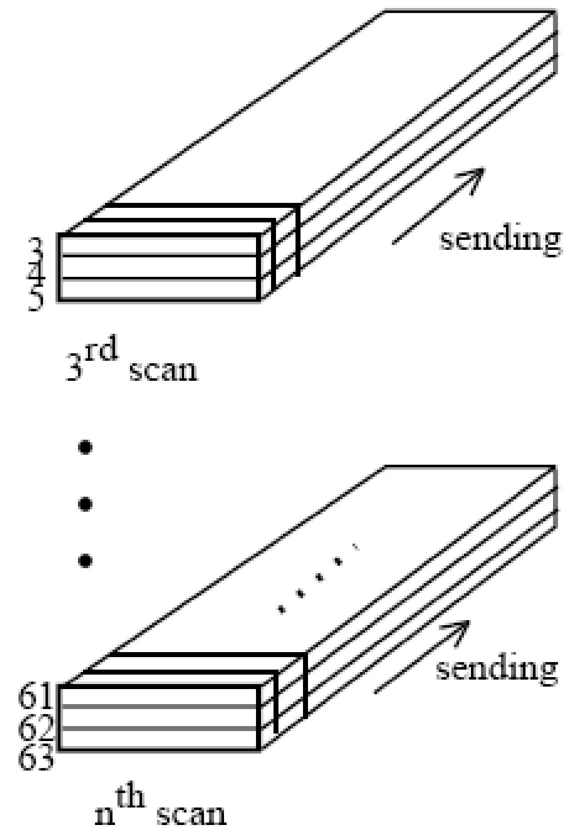
基于DCT的递增编码模式

- 此模式与顺序模式编码步骤基本一致
- 不同之处在于：递增模式每个图像分量的编码要经过多次扫描才完成
  - 第一次扫描只进行一次粗糙的压缩，然后根据此数据先重建一幅质量低的图像
  - 以后的扫描再作较细的扫描，使重建图像质量不断提高，直到满意为止
- 递增模式(Progressive Encoding)分为两种：
  - 按频段累进(Spectral Selection)
  - 按位累进(Successive Approximation)

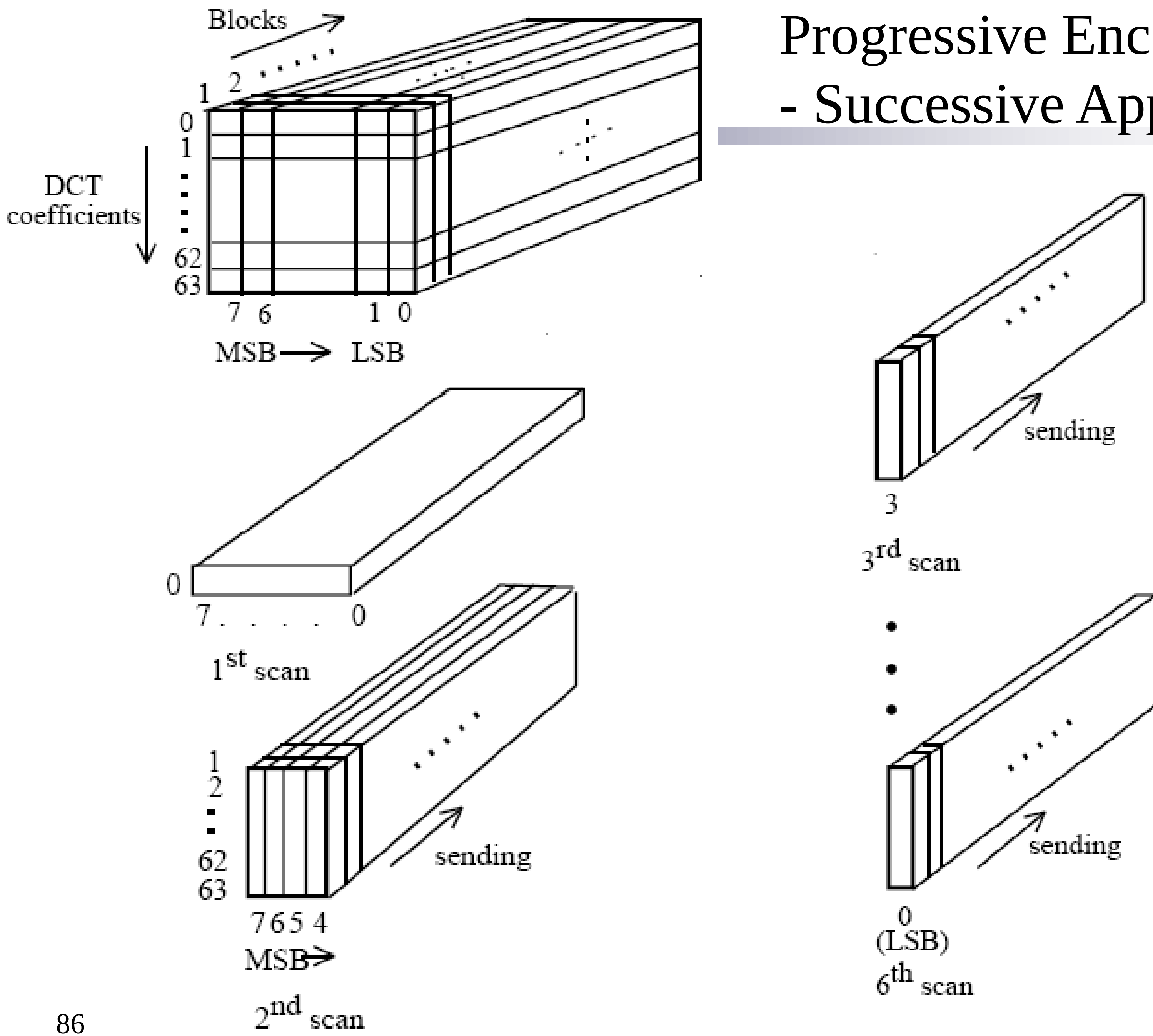
Progressive Encoding - Spectral Selection







Progressive Encoding
- Successive Approximation



基于DCT的分层编码模式

- 1、降低原始图像的空间分辨率
- 2、对已经降低分辨率的图像按照顺序编码模式进行压缩并存储或传输
- 3、对低分辨率图像进行解码，然后用插值法提高图像的分辨率
- 4、将分辨率已经升高的图像作为原图像的预测值，并把它与原图像的差值进行基于DCT的编码
- 5、重复步骤3、4直到图像达到完整的分辨率

Google Guetzli- Perceptually Guided JPEG Encoder 2017