多媒体数据压缩基本算法

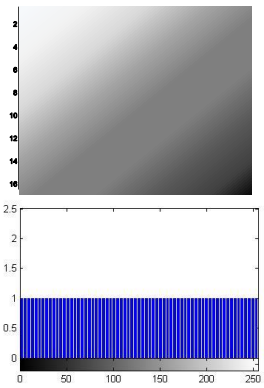
- 1 概述
- 2 失真量度
- 3 比率失真理论
- 4 变换编码
- 5 量化
- 6小波编码和小波包
- 7 EZW和SPIHT

Ze-Nian Li： c8： 有损压缩算法

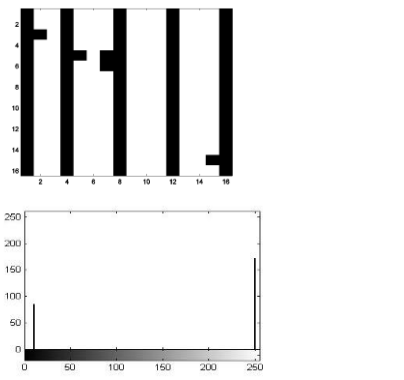
林福宗：c7小波和小波变换 c8小波图像压缩

1 概述

- 当图像直方图相对平坦时，对图像数据采用无损压缩技术，压缩率很低；



$$\eta = H(S) = \sum_{i=1}^{255} p_i \log_2 \frac{1}{p_i} = 8$$



$$\eta = \frac{1}{2}\log_2 3 + \frac{2}{3}\log_2 \frac{3}{2} = 0.92$$

1 概述

- 当图像直方图相对平坦时，对图像数据采用无损压缩技术，压缩率很低；
- 多媒体应用中需要高压缩率
  - 有损压缩方法

无损压缩和有损压缩

- **无损压缩**:使用压缩后的数据进行重构(或者叫做还原，解压缩)，重构后的数据与原来的数据完全相同

- **有损压缩**:使用压缩后的数据进行重构，重构后的数据与原来的数据有所不同，但不影响人对原始资料表达的信息造成误解。

4

2 失真量度（Distortion Measures）

- 失真量度是一个说明在某种失真标准下一个近似值与原值的接近程度的数学量。
- 最常用的有三种：
  - 均方差 MSE
    （像素的平均差异）

$$\sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - y_n)^2$$

  - 信噪比
    （相对于信号的误差大小）

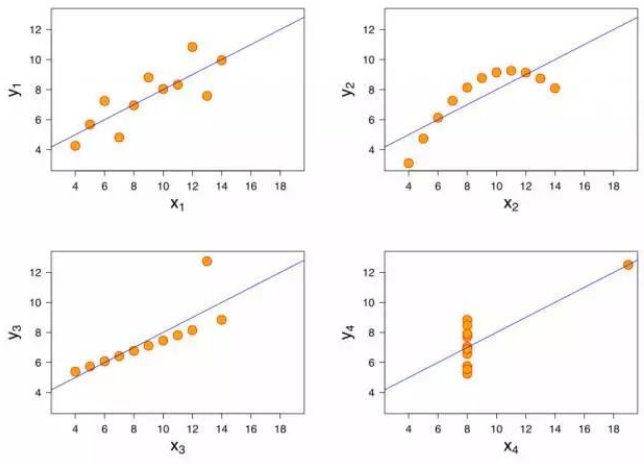
$$SNR = 10\log_{10}\frac{\sigma_x^2}{\sigma_d^2}$$

  - 峰值信噪比
    （相对于信号峰值的误差大小）

$$PSNR = 10\log_{10}\frac{x_{peak}^2}{\sigma_d^2}$$

安斯库姆四重奏

- 四组数据的基本统计特性一致



✓ 包含11个点
✓ 相同的均值
✓ 相同的中值
✓ 相同的方差

图表则截然不同

相同均方差，不同噪声的主观质量的差异MSE=225

(a) Original "Lena" image, 512 by 512, 8bits/pixel

(b) Impulsive Salt-Pepper Noise

(c) Additive Gaussian Noise

(d) Multiplicative Speckle Noise

(e) Mean Shift

(f) Contrast Stretching

(g) Blurring

(h) JPEG Compression

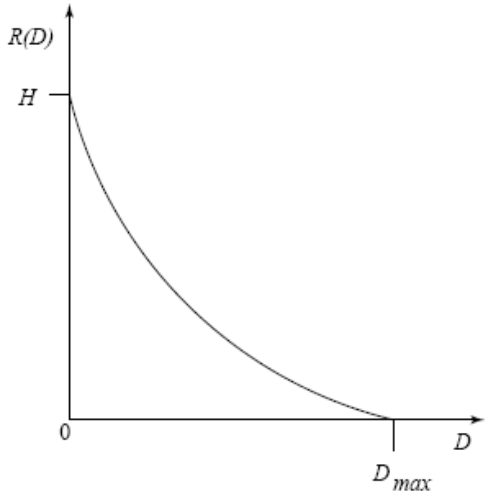
3 率失真理论 Rate-Distortion

信息率：重现源信号所需要的平均比特数（信息量）

R(D)： 率失真函数

1、D=0是没有缺损时的最小比特率，即源数据的熵H；

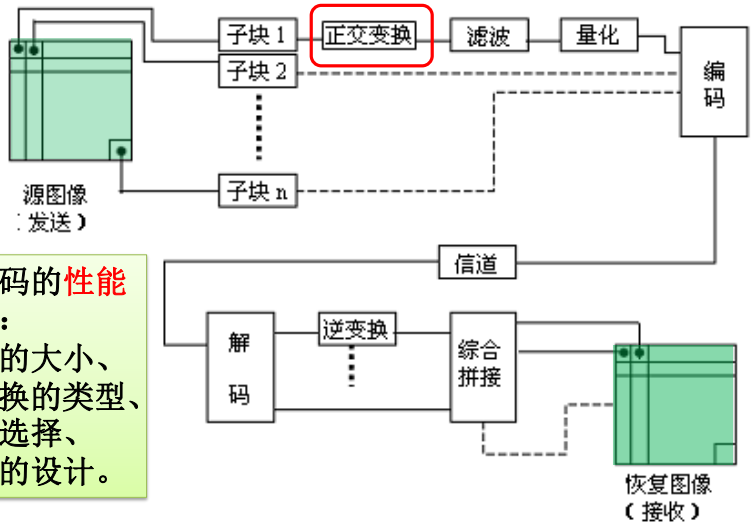
2、而在R(D)=0时，失真达到最大值。



4 变换编码

- 统计分析发现，绝大部分图像信号在空间域中像素之间的相关性是很大的。它们经过正交变换以后，其能量主要集中在低频部分；而且经过正交变换后的变换系数之间的相关性大大降低。
- 变换编码：不是直接对空间域图像信号编码，而是首先将空间域图像信号映射变换到另一个正交矢量空间，产生一批变换系数，然后对这些变换系数进行编码处理。
- 空间域的一个N×N个像素组成的像块经过正交变换后，在变换域变成了同样N×N的变换系数块。

4.1 变换编、解码过程示意图



变换编码的性能取决于：
子图像的大小、
正交变换的类型、
样本的选择、
量化器的设计。

4.2 正交变换类型

- 从均方误差最小和主观图像质量两个观点来看，最好的变换类型是离散K-L变换。 但由于离散K-L变换的基核向量是不固定的，一般没有快速算法，因此只宜作理论分析和试验。
- 在数字信号处理技术中，傅里叶变换是应用最为广泛的一类正交变换，它不仅具有物理含义明确的优点，而且可以使用快速算法FFT来减少运算量。但它应用在图像编码中时也有两个明显的弱点：一是要进行复数运算；二是收敛速度较慢。

正交变换类型

- 离散余弦变换DCT与K-L变换压缩性能和误差比较接近，而且DCT也具有多种快速算法，因而在图像压缩编码中被广泛的应用。
- **DCT变换具有如下特点：**
  - DCT变换计算复杂度适中。
  - DCT变换域系数矩阵能量集中在直流和低频区。
  - DCT的直流系数近似满足瑞利分布，交流系数近似满足拉普拉斯分布。
  - DCT系数相关性很小。

例如

- 在JPEG图像压缩算法中，首先将输入图像划分为8×8的方块，然后对每一个方块执行二维离散余弦变换，最后将变换得到的量化的DCT系数进行编码和传送，形成压缩后的图像格式。
- 在接收端，将量化的DCT系数进行解码，并对每个8×8方块进行二维IDCT，最后将操作完成后的块组合成一幅完整的图像。

离散余弦变换 DCT

对N * N的像素矩阵进行DCT变换的公式如下：

离散余弦变换(Discrete Cosine Transform,DCT)公式：

$$DCT(i,j)=\frac{1}{\sqrt{2N}}C(i)C(j)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}\{Pixel(x,y)\cdot Cos[\frac{(2x+1)\cdot i\cdot\pi}{2N}]\cdot Cos[\frac{(2y+1)\cdot j\cdot\pi}{2N}]\}$$

反向离散余弦变换(Inverse Discrete Cosine Transform,IDCT)公式：

$$Pixel(x,y)=\frac{1}{\sqrt{2N}}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\{C(i)C(j)\cdot DCT(i,j)\cdot Cos[\frac{(2x+1)\cdot i\cdot\pi}{2N}]\cdot Cos[\frac{(2y+1)\cdot j\cdot\pi}{2N}]\}$$

其中：

$$C(x)=\begin{cases}\frac{1}{\sqrt{2}} & (x=0)\\ 1 & (x>0)\end{cases}$$

但是：按照上述基本公式写出的程序实现存在一个严重的问题——时间复杂度太高

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。

二维DCT变换

- 正交变换的矩阵表示

  若以 f 表示图像矩阵，以C表示变换核矩阵，以F表示变换系数矩阵，则利用两次矩阵乘法可表示二维线性变换如下：

$$\mathbf{F} = C f C^{T}$$

- 变换矩阵C应与图像矩阵 **f** 的阶数相同，变换系数矩阵F与图像矩阵 **f** 的阶数也相同。
- 发送端以 **F** 形式发送，接收端收到这种形式的信号后，采用相应的逆变换把 **F** 反变换成 **f** ，所用的变换矩阵是A的逆矩阵C^{-1}和C^{T}的逆矩阵$(C^{T})^{-1}$：

$$C^{-1}F(C^{T})^{-1} = C^{-1}CfC^{T}(C^{T})^{-1} = IfI = f$$

余弦变换矩阵C(Cosine Transform Matrix)满足：

$$C(i,j)=\frac{1}{\sqrt{N}} \quad if \quad i=0$$

$$C(i,j)=\sqrt{\frac{2}{N}}Cos[\frac{(2j+1)\cdot i\cdot \pi}{2N}] \quad if \quad i>0$$

且C的转置矩阵为C_t(即$C(i,j)=C_t(j,i)$)，则函数DCT可表示为：

$$DCT = C\times 象素矩阵\times C_t$$

(注：上面的乘号表示矩阵乘法)

- 实现上面的替代公式的程序代码的时间复杂度就大大降低了。进一步的改进还包括将余弦函数由浮点运算改为整数运算、改进傅立叶变换技术等。

离散余弦变换的矩阵算法举例：

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

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

$$C(i,j)=\frac{1}{\sqrt{N}} \quad if \quad i=0$$

$$C(i,j)=\sqrt{\frac{2}{N}}Cos[\frac{(2j+1)\cdot i\cdot\pi}{2N}] \quad if \quad i>0$$

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

$$C(1,0)=\sqrt{\frac{2}{N}}Cos[\frac{(2j+1)\cdot i\cdot\pi}{2N}]=\sqrt{\frac{1}{2}}Cos[\frac{\pi}{8}]=0.653281$$

$$C(1,1)=\sqrt{\frac{2}{N}}Cos[\frac{(2j+1)\cdot i\cdot\pi}{2N}]=\sqrt{\frac{1}{2}}Cos[\frac{3\pi}{8}]=0.270598$$

$$=\begin{bmatrix}2&0&-2&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}$$

变换编码

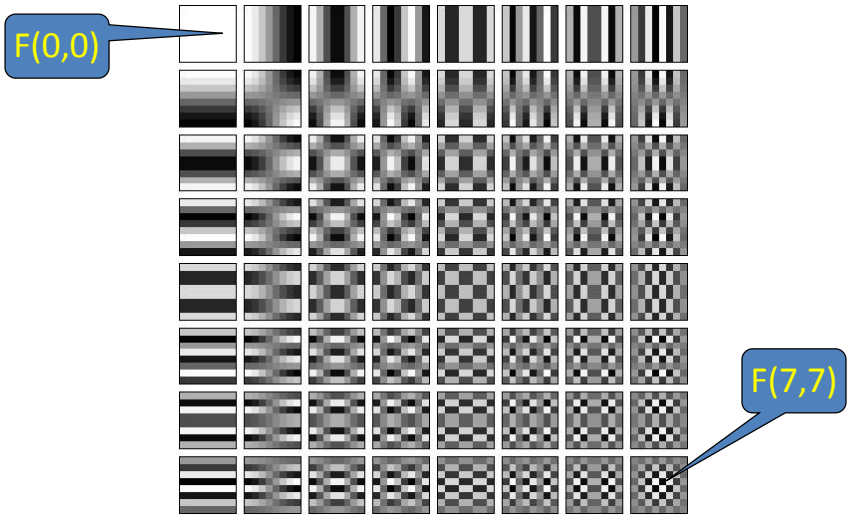
- 不同系数的物理意义：

F(0,0)表示像素块的直流分量；其它交流系数表示不同空间频率的相位和幅度。

例如，F(0,1)表示像素块内水平方向半周余弦空间频率的相位和幅度。F(1,0)表示像素块内垂直方向半周余弦空间频率的相位和幅度。

- 基图像（basis）
  - 当64个系数中只有一个系数为1，其余全部为0时，相应的64个像素值所组成的基本图象
  - 任何像块都可表示成64个基图像的不同比例的组合。

– Two dimension DCT basis functions (8 x 8)



任何像块都可表示成64个基图像的不同比例的组合。

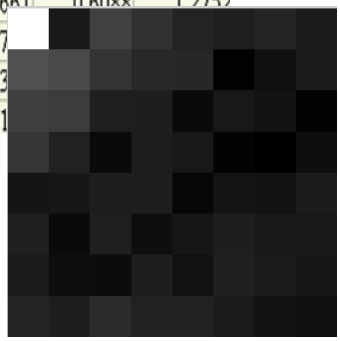
一个真实的编码和解码过程

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

8×8原始图像块

一个真实的编码和解码过程

1259.625	-1.0333	-12.0809	-5.2029	2.125	-1.6724	-2.708	1.3238
-22.5904	-17.4842	-6.2405	-3.1574	-2.8557	-0.0695	0.4342	-1.1856
-10.9493	-9.2624	-1.5758	1.5301	0.2029	-0.9419	-0.5669	-0.0629
-7.0816	-1.9072	0.2248	1.4539	0.8963	-0.0799	-0.0423	0.3315
-0.625	-0.8381	1.4699	1.5563	-0.125	-0.661	0.6088	1.2752
1.7541	-0.2029	1.6205	-0.3424	-0.7755	1.47		
-1.2825	-0.36	-0.3169	-1.4601	-0.49	1.73		
-2.5999	1.5519	-3.7628	-1.8448	1.8716	1.21		



经过DCT变换

	1	2	3	4	5	6	7	8
1	139.0000	144.0000	149.0000	153.0000	155.0000	155.0000	155.0000	155.0000
2	144.0000	151.0000	153.0000	156.0000	159.0000	156.0000	156.0000	156.0000
3	150.0000	155.0000	160.0000	163.0000	158.0000	156.0000	156.0000	156.0000
4	159.0000	161.0000	162.0000	160.0000	160.0000	159.0000	159.0000	159.0000
5	159.0000	160.0000	161.0000	162.0000	162.0000	155.0000	155.0000	155.0000
6	161.0000	161.0000	161.0000	161.0000	160.0000	157.0000	157.0000	157.0000
7	162.0000	162.0000	161.0000	163.0000	162.0000	157.0000	157.0000	157.0000
8	162.0000	162.0000	161.0000	161.0000	163.0000	158.0000	158.0000	158.0000

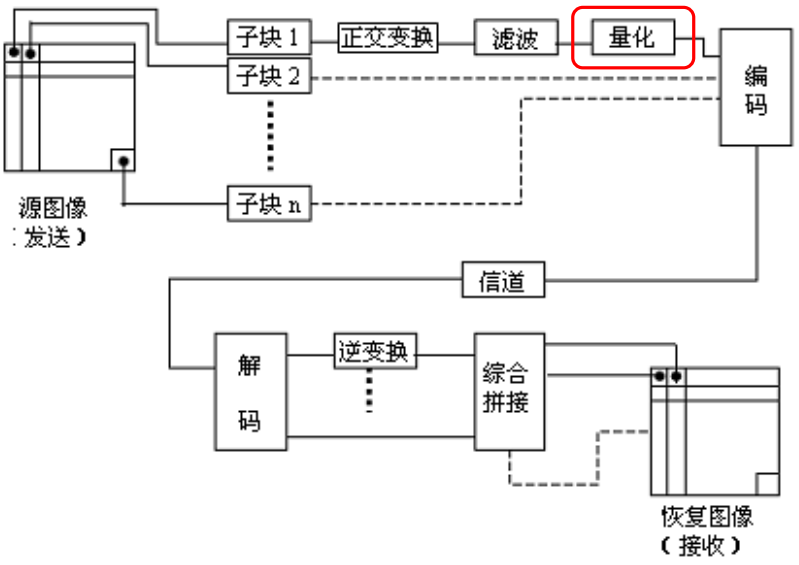
反DCT后的矩阵

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

(a) 8X8的原始图像块

4.4 离散余弦变换的几点说明

- DCT变换避免了复数运算。由于图像矩阵是实数矩阵，那么它的DCT也是实数
- DCT是正交变换，其变换矩阵是正交阵，变换核是可分离的。
- DCT有快速算法
- DCT与IDCT具有相同的变换核，因此具有相同的变换矩阵，即正变换与逆变换公用同一个算法模块
- DCT具有更强的信息集中能力，能将最多的信息放到最小的系数上去。



5、量化

- **量化是压缩的核心**
- 每个量化器都可以由编码器端的输入范围划分和解码器端的输出值域唯一确定。
- 分类
  - 标量量化器：输入值和输出值是标量
    - **均匀**标量量化器
    - **非均匀**标量量化器
  - 矢量量化器：输入值和输出值是矢量

5.1 均匀标量量化器

- 判定边界：
  - 输入值域划分为等距的区间的区间端点
- 步长：
  - 区间的长度 △
- 输出：
  - 每个区间对应的重现值，取区间的中点值。

- 均匀量化器只对均匀分布信源是最佳的。

均匀量化

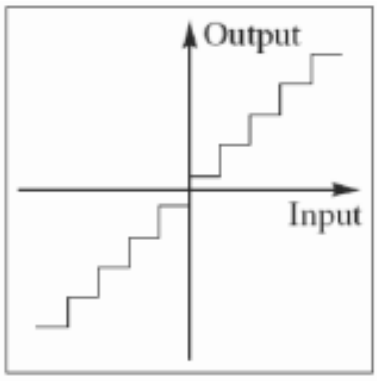
假设：输入值的范围
$[-X_{max}, X_{max}]$

- 输出值$Y=\{y_1, y_2, \cdots, y_M\}$

则：

- 量化器的比率**$R= \log_2 M$**
  (即对M个值进行编码所需位数)
- 步长 **$\triangle=2X_{max}/M$**
- 粒度失真：量化器引起的量化误差

$$SQNR = 6.02n(dB)$$



均匀量化器

5.2 非均匀量化

- 对于输入源不是均匀分布的，均匀量化就可能失去作用
- 在源密集分布的区域增加判定级数量可以有效降低粒度失真。
- ① Lloyd-Max量化器
  - 亦称为PDF-最佳量化器

Probability density function

$$D = \sigma_q^2 = \int_{-\infty}^{\infty} (x - Q(x))^2 f(x)\,dx = \sum_{i=1}^{M} \int_{b_{i-1}}^{b_i} (x - y_i)^2 f(x)\,dx$$

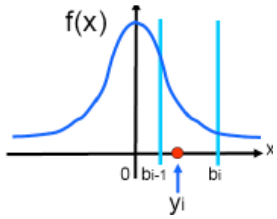
  - 给定M，最佳的b_i，y_i使得MSQE最小，满足

$$\frac{\partial D}{\partial y_i} = 0, \quad \frac{\partial D}{\partial b_i} = 0$$

5.2 非均匀量化

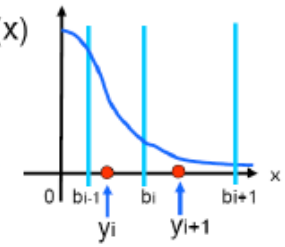
$$\frac{\partial D}{\partial y_i}=0 \Rightarrow y_i=\frac{\int_{b_{i-1}}^{b_i} xf(x)dx}{\int_{b_{i-1}}^{b_i} f(x)dx}$$

即y_i为区间$[b_{i-1},b_i]$的质心



$$\frac{\partial D}{\partial b_i}=0 \Rightarrow b_i=\frac{y_i+y_{i+1}}{2}$$

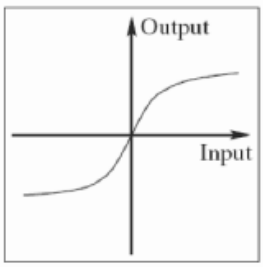
b_i为y_i和y_{i+1}的中点→最近邻量化器



②压缩扩展量化器

- 输入通过一个压缩函数G映射后由一个均匀量化器进行量化。

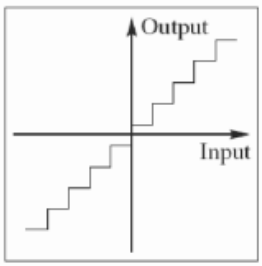
X



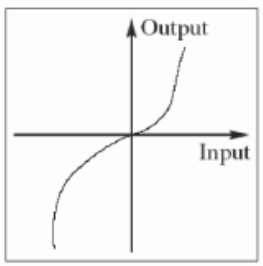
Compressor

G

μ律和A律



均匀量化器



Expander

G^{-1}

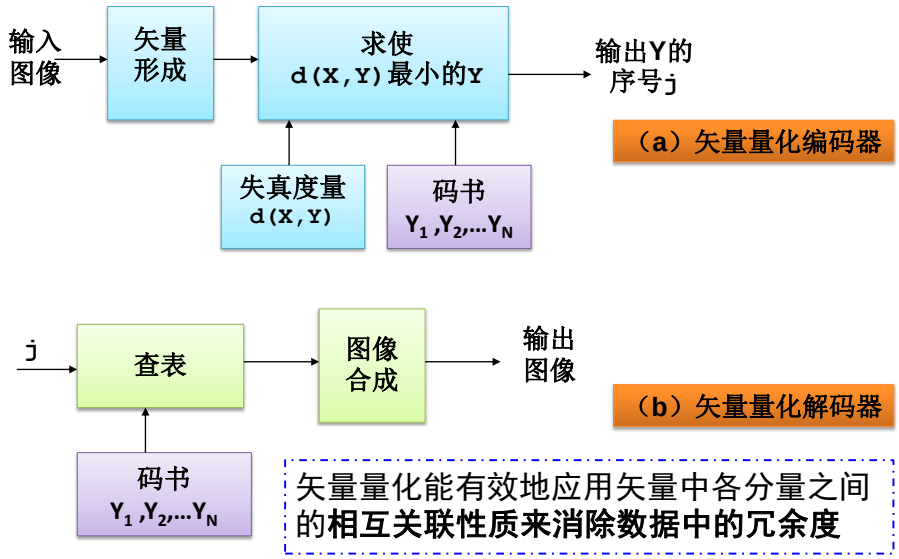
X^

③矢量量化（VQ）

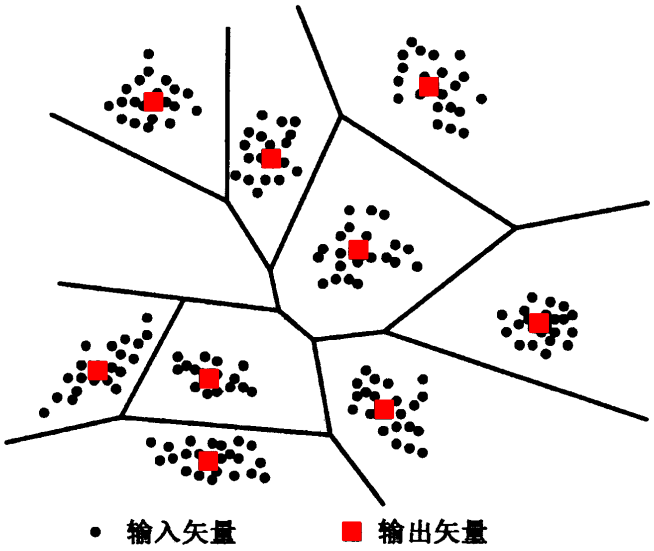
- 将标量量化扩展到多个维度
- 用一个包含n个分量的码向量（**code vector**）表示n维空间某个区域的向量。这些码向量的集合构成了矢量量化量化器中的码本（**codebook**）。
- **VQ技术目前已在语音、图象压缩等领域得到广泛应用**
- 矢量量化是香农信息论在信源编码理论方面的发展，它的理论基础是香农的率失真理论：

在给定失真D条件下，所能够达到的最小速率R(D)
或
在给定速率R条件下，所能够达到的最小失真D(R)

矢量量化编解码



矢量量化举例



一个真实的编码和解码过程（带有量化）

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

8×8原始图像块

一个真实的编码和解码过程（带有量化）

1259.625	-1.0333	-12.0809	-5.2029	2.125	-1.6724	-2.708	1.3238
-22.5904	-17.4842	-6.2405	-3.1574	-2.8557	-0.0695	0.4342	-1.1856
-10.9493	-9.2624	-1.5758	1.5301	0.2029	-0.9419	-0.5669	-0.0629
-7.0816	-1.9072	0.2248	1.4539	0.8963	-0.0799	-0.0423	0.3315
-0.625	-0.8381	1.4699	1.5563	-0.125	-0.661	0.6088	1.2752
1.7541	-0.2029	1.6205	-0.3424	-0.7755	1.4759	1.041	-0.993
-1.2825	-0.36	-0.3169	-1.4601	-0.49	1.7348	1.0758	-0.7613
-2.5999	1.5519	-3.7628	-1.8448	1.8716	1.2139	-0.5679	-0.4456

经过DCT变换

一个真实的编码和解码过程（带有量化）

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

量化矩阵：即每个系数上对应的量化阶

一个真实的编码和解码过程（带有量化）

	1	2	3	4	5	6	7	8
1	1264	-22	-10	-16	0	0	0	0
2	0	-12	-14	0	0	0	0	0
3	-14	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
9								

反量化后的矩阵

一个真实的编码和解码过程（带有量化）

141.8569	148.595	156.6824	160.7568	160.2795	158.7249	158.8517	159.9608
144.1185	150.5069	158.0707	161.7008	161.0487	159.6155	160.0186	161.3437
147.7166	153.4588	160.0552	162.8643	161.8893	160.6803	161.5942	163.3182
151.2823	156.1802	161.5125	163.249	161.852	160.936	162.5172	164.7626
153.692	157.6761	161.6403	162.2157	160.362	159.7631	162.0665	164.8762
154.5789	157.7187	160.4189	159.9216				
154.3887	156.8822	158.615	157.2968				
153.9715	156.1153	157.3245	155.562				

反DCT后的矩阵

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

(a) 8X8的原始图像块

6 小波编码和小波包

- 小波变换（wavelet transform）: 采用一组称为小波的基函数来表示信号，可以在时域频域上得到很好的分辨率。
- 一维函数的Fourier变换定义为

$$\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} f(t)e^{j\omega t}\,dt$$

- 逆变换为：

$$\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} F(\omega)e^{-j\omega t}\,d\omega$$

小波函数

- 小波（wavelet），即小区域的波。

- 小波函数确切定义为：设$\psi(t)$为一平方可积函数（即能量有限的信号空间），也即$\psi(t) \in L^2(R)$，若其傅里叶变换满足条件：

$$C_\psi = \int_R \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega < \infty$$

- 则$\psi(t)$为一个基本小波或小波母函数，并称上面为小波函数的可容许条件。

1-42

① CWT及其反变换

- 母小波ψ(t)是满足下列特性的实或复的连续函数：
  - 函数均值为0
  - 属于L^2
  - 相容性条件

$$\int_{-\infty}^{+\infty} \psi(t)\,dt = 0$$

$$\int_{-\infty}^{+\infty} |\psi(t)|^2\,dt < +\infty$$

- 一旦选择了小波ψ(t)，函数f(t)的连续小波变换：

$$W(a,b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{+\infty} f(t)\,\psi^*\left(\frac{t-b}{a}\right) dt$$

CWT及其反变换.

- 一般，定义小波基函数

其中b是平移参数

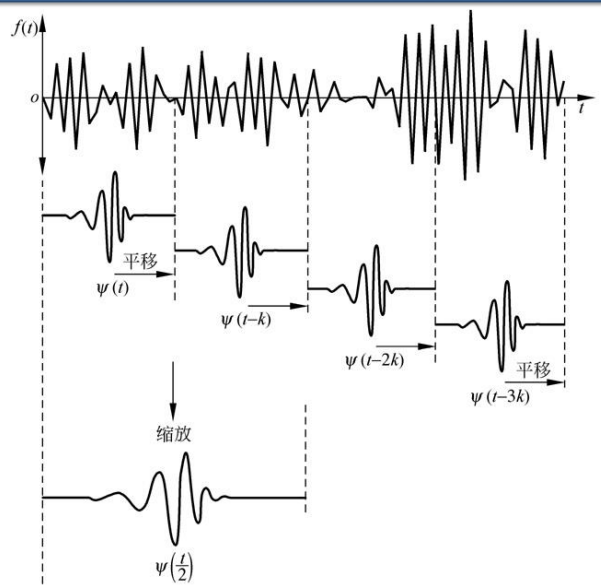
a是尺度（或伸缩）参数

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}}\psi\left(\frac{t-b}{a}\right)$$

- 逆CWT定义为：

$$f(t) = \frac{1}{C}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\frac{1}{|a|^2}W(a,b)\,\psi_{a,b}(t)\,da\,db$$

平移和缩放



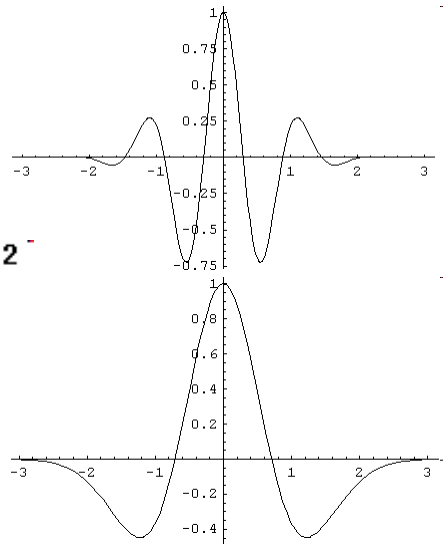
两个小波

- Morlet小波

$$\psi(t) = e^{-t^2} \cos\left(\pi t \sqrt{\frac{2}{\mathrm{Log2}}}\right)$$

- 墨西哥草帽小波

$$\psi(t) = (1 - 2t^2)\, e^{-t^2}$$



② Haar变换

- Haar变换基于Haar函数$h_k(x)$, $(0\le x\le 1,\ k=0,1,\ldots,N-1,\ N=2^n)$
- 设$k=2^p+q-1$,
  其中$0\le p\le n-1$
  当p=0时，q=0或1
  当p>0时，$1\le q\le 2^p$,
- 定义: $h_0(x) = 1/\mathrm{Sqrt}[N]$

$$h_k(x) = h_{p,q}(x) = 1/\mathrm{Sqrt}[N] * \begin{cases} 2^{p/2} & \frac{q-1}{2^p} \le x < \frac{q-1/2}{2^p} \\ -2^{p/2} & \frac{q-1/2}{2^p} \le x < \frac{q}{2^p} \\ 0 & \text{其它} \end{cases}$$

N×N阶Haar变换矩阵A_N 的i行j列的元素为：$h_i(j/N)$, i,j =0,1,…,N-1

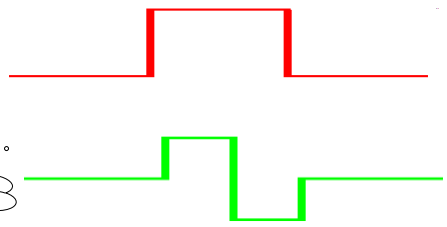
Haar变换

- Haar变换将函数f(x)分解为以下的无穷和

$$f(t) = \sum_{k=-\infty}^{\infty} c_k \phi(t-k) + \sum_{k=-\infty}^{\infty} d_{j,k} \psi(2^j t - k)$$

$$\phi(t) = \begin{cases} 1, & 0 \le t < 1 \\ 0, & \text{其它} \end{cases}$$

$$\psi(t) = \begin{cases} 1, & 0 \le t < 0.5 \\ -1, & 0.5 \le t < 1 \end{cases}$$



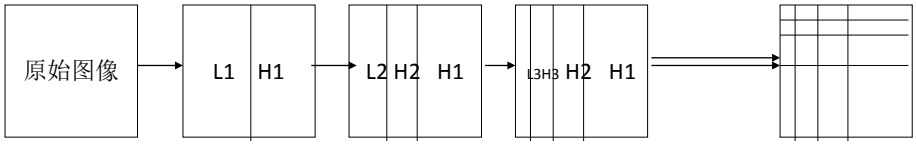
一个例子

- 设有8个整数{1, 2, 3, 4, 5, 6, 7, 8}
- 首先计算4个平均值和4个差值(低分辨率和细节)：
  {(1+2)/2, (3+4)/2, (5+6)/2, (7+8)/2, (1-2)/2, (3-4)/2, (5-6)/2, (7-8)/2}
  = {3/2, 7/2, 11/2, 15/2, -1/2, -1/2, -1/2, -1/2}
- 对4个平均值重复上述过程，分成两个平均值和两个差值：
  {10/4, 26/4, -4/4, -4/4, -1/2, -1/2, -1/2, -1/2}
- 对前两个分量继续进行：
  {36/8, -16/8, -4/4, -4/4, -1/2, -1/2, -1/2, -1/2}

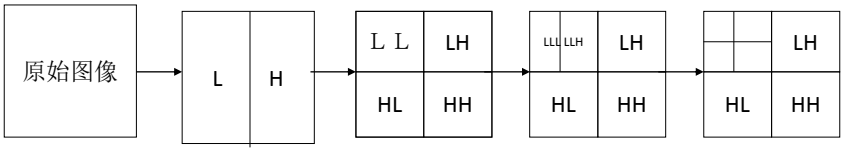
- 这些过程把序列分解为不同分辨率的表示，由于差值较小或可以忽略，就有可能进行有效的压缩。

Haar变换的应用

- 把它推广到二维图像：标准分解和金字塔分解
  - 标准分解
    - 每行进行小波变换，然后对每列进行

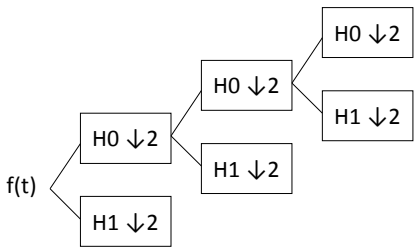


  - 金字塔分解
    - 在行列之间交替进行小波变换



滤波器组

- 滤波器是用滤波器系数h(0), h(1), h(2), …定义的线性算子(x输入,y输出)：
  $y(n) = \sum h(k)x(n-k) = h*x$ (卷积)
- 滤波器组的思想是：由一个低通滤波器H0和一个高通滤波器H1组成一个滤波器组，低通滤波器利用卷积去除输入信号x中的高频分量，而让低频通过，高通滤波器的作用正好相反，它们共同将输入分成频带。



滤波器组系数

- 给定一组滤波器，有N抽头的前向和反向滤波器各两个：H0、H1和F0、F1(N是偶数)，其系数记为h0, h1,f0,f1。这些量必须满足的一组条件是：
  - 归一化：矢量h0是归一化的，即长度为1
  - 正交性：$1 \le i \le N/2$，h0由2i个元素形成的矢量与同一个h0后2i个元素形成的矢量正交
  - 矢量f0是h0的逆
  - 矢量h1与f0除了奇数下标的元素相反外都相同
  - 矢量h0与f1除了偶数下标的元素相反外都相同

③ DWT

- 用矩阵乘法描述离散小波变换，以Daubechies D4为例：

$$W = \begin{pmatrix} c_0 & c_1 & c_2 & c_3 & 0 & 0 & \cdots & 0 \\ c_3 & -c_2 & c_1 & -c_0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & c_0 & c_1 & c_2 & c_3 & \cdots & 0 \\ 0 & 0 & c_3 & -c_2 & c_1 & -c_0 & \cdots & 0 \\ \vdots & \vdots & & & & \ddots & & \\ 0 & 0 & \cdots & 0 & c_0 & c_1 & c_2 & c_3 \\ 0 & 0 & \cdots & 0 & c_3 & -c_2 & c_1 & -c_0 \\ c_2 & c_3 & 0 & \cdots & 0 & 0 & c_0 & c_1 \\ c_1 & -c_0 & 0 & \cdots & 0 & 0 & c_3 & -c_2 \end{pmatrix}.$$

- 满足条件：
  - $c_0^2+c_1^2+c_2^2+c_3^2=1$
  - $c_2c_0+c_3c_1=0$

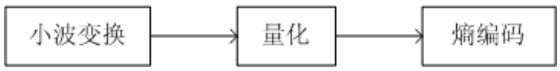
$$\left\{\frac{1+\sqrt{3}}{4\sqrt{2}}, \frac{3+\sqrt{3}}{4\sqrt{2}}, \frac{3-\sqrt{3}}{4\sqrt{2}}, \frac{1-\sqrt{3}}{4\sqrt{2}}\right\}.$$

其它常用小波的滤波器系数

- Beylkin
- Coifman 1-tap
- Coifman 2-tap
- …
- Coifman 5-tap
- Daubechies 4-tap
- …
- Daubechies 20-tap
- Symmlet 4-tap
- …
- Symmlet 10-tap
- Vidyanathan

7 EZW编码 和SPIHT

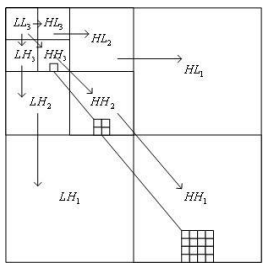
EZW：嵌入式零树小波图像压缩技术

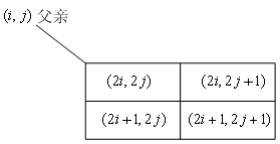


小波图象编码的一般结构

如何组织小波系数及其位置信息？

如何处理小波系数及其位置信息？



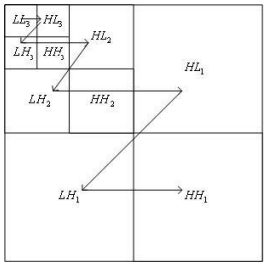


63	−34	49	10	7	13	−12	7
−31	23	14	−13	3	4	6	−1
15	14	3	−12	5	−7	3	9
−9	−7	−14	8	4	−2	3	2
−5	9	−1	47	4	6	−2	2
3	0	−3	2	3	−2	0	4
2	−3	6	−4	3	6	3	6
5	11	5	6	0	3	−4	4

小波树状结构（1992，Lewis and Knowles）

几个重要的概念: 重要系数、不重要系数、零树根、孤立点

EZW编码



63	−34	49	10	7	13	−12	7
−31	23	14	−13	3	4	6	−1
15	14	3	−12	5	−7	3	9
−9	−7	−14	8	4	−2	3	2
−5	9	−1	47	4	6	−2	2
3	0	−3	2	3	−2	0	4
2	−3	6	−4	3	6	3	6
5	11	5	6	0	3	−4	4

通过多遍扫描编码多分辨图像，其中每一遍扫描包含以下的处理步骤：

1.选择阈值

2. 主扫描

3. 辅扫描

4. 重新排序

5. 输出编码信号

EZW编码

1.选择阈值

对于*L*级小波变换，EZW算法应用一系列的阈值 T_0，T_1,…, T_{L-1} **来确定小波系数的重要性，其中** $T_i = T_{i-1}/2$，i **为扫描次数，**i ＝1，2，….，L－1。

初始阈值的选择方法如下：

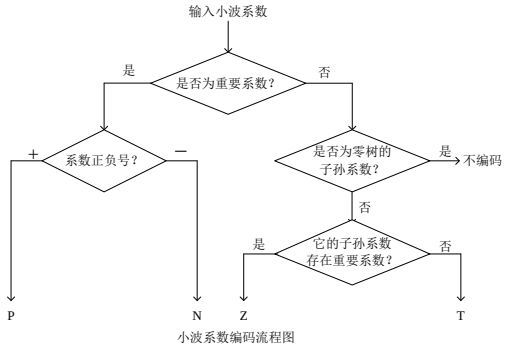
$$T_0 = 2^{\lfloor \log_2 Max\{|c_{i,j}|\} \rfloor}$$

63	−34	49	10	7	13	−12	7
−31	23	14	−13	3	4	6	−1
15	14	3	−12	5	−7	3	9
−9	−7	−14	8	4	−2	3	2
−5	9	−1	47	4	6	−2	2
3	0	−3	2	3	−2	0	4
2	−3	6	−4	3	6	3	6
5	11	5	6	0	3	−4	4

$T_0 = 32$

EZW编码

2. 主扫描



小波系数编码流程图

P_1	N_2	P_3	T_6	Z_{13}	Z_{14}	×	×
Z_3	T_4	T_7	T_8	Z_{15}	Z_{16}	×	×
T_9	Z_{10}	×	×	×	×	×	×
T_{11}	T_{12}	×	×	×	×	×	×
×	×	Z_{17}	P_{18}	×	×	×	×
×	×	Z_{19}	Z_{20}	×	×	×	×
×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×

D_1：PNZTPTTTTZTTZZZZZPZZ

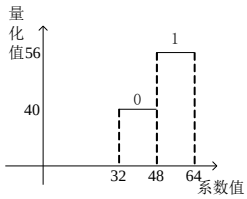
在扫描过程中，用一个主扫描表记录这些输出符号。当一个系数的输出符号为T时，它的所有子孙系数就不再扫描，并用×表示。

第*i*次主扫描结束后，将输出符号为P或N的系数的相应位置加标记或将这些系数置为零，以免在下次主扫描时再对它们编码。

EZW编码

3. 辅扫描

对主扫描表进行顺序扫描，对其中输出符号为P或N的小波系数进行量化。



系数量化器

表5.1 第一次辅扫描量化表

系数幅值	量化符号	重构幅值
63	1	56
34	0	40
49	1	56
47	0	40

量化符号组成的位流为S_1 ： 1010

4. 重新排序

为便于设置第 $i+1$ 次扫描所用的量化间隔，以提高解码的精度，对输出符号为P或N的数据重新排序。

$$\{63-P,34-N,49-P,47-P\} \Rightarrow \{63-P,49-P,34-N,47-P\}$$

EZW编码

5. 输出编码信息

编码器输出两类信息：

一类是给**解码器**的信息，包括阈值、主扫描表和辅扫描表；

$T_0 = 32$，D_1 ：PNZTPTTTTZTTZZZZZPZZ；　S_1：1010

第二类是用于**下次扫描**的信息，包括阈值及第**4**步中重新排序过的重要系数序列。

$T_0 = 32$，$\{63-P,49-P,34-N,47-P\}$，**小波图像数据。**

EZW编码

第二次编码：

设置新阈值：$T_1 = T_0 / 2 = 16$

主扫描：

⊗	⊗	⊗	T_3	Z_{14}	Z_{15}	×	×
N_1	P_2	T_4	T_5	Z_{16}	Z_{17}	×	×
T_6	T_7	T_{10}	T_{11}	×	×	×	×
T_8	T_9	T_{12}	T_{13}	×	×	×	×
×	×	×	⊗	×	×	×	×
×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×

D_2：NPTTTTTTTTTTTZZZZ

辅扫描：

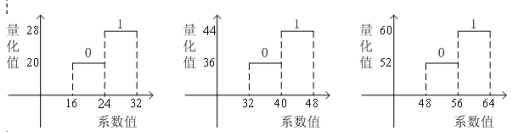


表5.2 第二次辅扫描量化表

系数幅值	量化符号	重构幅值
63	1	60
49	0	52
34	0	36
47	1	44
31	1	28
23	0	20

S_2 ：100110

重新排序：$\{63-P,49-P,47-P,34-N,31-N,23-P\}$

EZW编码

第二次编码输出结果：

a）为解码器提供的信息

$T_1=16$，D_2 ：NPTTTTTTTTTTTZZZZ； S_2：100110

b）为下一次扫描的信息

$T_1=16$，$\{63-P,49-P,47-P,34-N,31-N,23-P\}$，小波图像数据。

表5.3 二次编码的输出结果

T_0	32
D_1/S_1	PNZTPTTTTZTTZZZZZPZZ/1010
D_2/S_2	NPTTTTTTTTTTTZZZZ/100110

EZW解码

解码过程的主要步骤包括：接收编码器发送的解码信息后，设置阈值，构造逆量化器。解读位流中包含的位置信息和小波系数信息。

第一次解码

解码器接收到的信息：32/ PNZTPTTTTZTTZZZZZPZZ/1010

重要的小波系数与其量化符号有如下的对应关系：

D_1	P	N	Z	T	P	T	T	T	T	Z	T	T	Z	Z	Z	Z	Z	P	Z	Z
S_1	1	0			1													0		

56	−40	56	0	0	0	×	×
0	0	0	0	0	0	×	×
0	0	×	×	×	×	×	×
0	0	×	×	×	×	×	×
×	×	0	40	×	×	×	×
×	×	0	0	×	×	×	×
×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×

EZW解码

第二次解码

解码器接收到的信息：16/NPTTTTTTTTTTTZZZZ / 100110

其中S_2的前4位表示第一次解码时得到的S_1中的量化符号，它们的重构值依次为 $(56,-40,56,40)$

第二次解码过程由两步组成：

1）应用新的量化器，提高第一次解码得到的重要系数的重构精度。

$(56,-40,56,40)$ ⇒ $(60,-36,52,44)$

2）求解在第一次解码时尚未恢复的系数。

由系数输出符号组成的位流与 S_2中后两位量化符号间的对应关系如下：

D_2	N	P	T	T	T	T	T	T	T	T	T	T	Z	Z	Z	Z
	1	0														

EZW解码

小波零树算法具有显著编码性能的原因：

（1）离散小波变换

（2）零树编码

（3）累进逼近

（4）自适应算法编码。

60	−36	52	0	0	0	×	×
−28	20	0	0	0	0	×	×
0	0	0	0	×	×	×	×
0	0	0	0	×	×	×	×
×	×	×	44	×	×	×	×
×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×

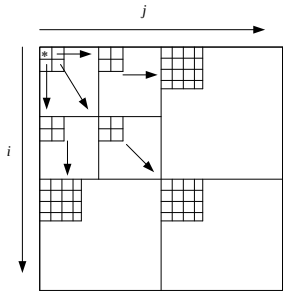
第二次解码后的结果

EZW编解码算法的实现:

SPIHT编码

SPIHT算法是EZW算法的改进算法.

SPIHT算法采用与EZW算法相似的零树结构，但它在系数子集的分割和重要信息的传输方式上采用了独特方法，能够在实现幅值大的系数优先传输的同时，不显式传送系数的排序信息。其基本依据是：任何排序算法的执行路径都是使用分支点的比较结果进行定义的，如果编码器和解码器使用相同的排序算法，则对于编码器输入的系数比较结果，解码器通过执行相同的路径就可获得排序信息。



63	−34	49	10	7	13	−12	7
−31	23	14	−13	3	4	6	−1
15	14	3	−12	5	−7	3	9
−9	−7	−14	8	4	−2	3	2
−5	9	−1	47	4	6	−2	2
3	0	−3	2	3	−2	0	4
2	−3	6	−4	3	6	3	6
5	11	5	6	0	3	−4	4

SPIHT编码

分集规则

$$S_n(X)=\begin{cases}1 & \text{若}\max_{(i,j)\in X}\{|c_{i,j}|\}\geq 2^n \\ 0 & \text{其他}\end{cases} \qquad \begin{matrix}S_n(X)=1 \\ S_n(X)=0\end{matrix} \qquad \begin{matrix}X\ \text{是重要的} \\ X\ \text{是不重要的}\end{matrix}$$

$O(i,j)$ ： 节点(i,j)所有孩子的坐标集；

$D(i,j)$ ： 节点(i,j)所有子孙的坐标集；

H ： 所有树根的坐标集。

$L(i,j)$ ： 节点(i,j)所有非直系子孙的坐标集；

$$L(i,j)=D(i,j)-O(i,j)$$

一般地，

$$O(i,j)=\{(2i,2j),(2i,2j+1),(2i+1,2j),(2i+1,2j+1)\}$$

63	−34	49	10	7	13	−12	7
−31	23	14	−13	3	4	6	−1
15	14	3	−12	5	−7	3	9
−9	−7	−14	8	4	−2	3	2
−5	9	−1	47	4	6	−2	2
3	0	−3	2	3	−2	0	4
2	−3	6	−4	3	6	3	6
5	11	5	6	0	3	−4	4

SPIHT编码

分集规则

1)最初坐标集由 $\{(i,j)\,|\,(i,j)\in H\}$ 和 $\{D(i,j)\,|\,(i,j)\in H$且具有非零子孙$\}$ 组成；

2) 若 $D(i,j)$ 是重要的，则 $D(i,j)$ 分成 $L(i,j)$ 及4个单节点 $(k,l)\in O(i,j)$

3) 若 $L(i,j)$ 是重要的，则 $L(i,j)$ 分成4个集 $D(k,l)$，$(k,l)\in O(i,j)$

63	-34	49	10	7	13	-12	7
-31	23	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2
-5	9	-1	47	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	-4	3	6	3	6
5	11	5	6	0	3	-4	4

有序表

LIP——不重要系数表；

LSP——重要系数表；

LIS——不重要子集表。

每一个表项都使用坐标 (i,j) 标识

在LIS中，坐标 (i,j) 代表 $D(i,j)$ 或者 $L(i,j)$

分别用*(i,j)*D和*(i,j)*L 表示

SPIHT编码的主要步骤

（1）阈值和有序表的初始化

设阈值 $T=2^n$ ，其中 $n=\lfloor \log_2(\max_{(i,j)}\{|c_{i,j}|\})\rfloor$

LSP为空集

$\mathrm{LIP}=\{(i,j)\,|\,(i,j)\in H\}$

$\mathrm{LIS}=\{(i,j)D\,|\,(i,j)\in H$且具有非零子孙$\}$

其中LIP和LIS中小波系数(i,j) 的排列顺序与零树的扫描顺序相同。

63	-34	49	10	7	13	-12	7
-31	23	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2
-5	9	-1	47	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	-4	3	6	3	6
5	11	5	6	0	3	-4	4

例: 小波系数最大幅值为63，故n＝5，阈值 $T=2^5=32$

$\mathrm{LIP}=\{(0,0),(0,1),(1,0),(1,1)\}$

$\mathrm{LIS}=\{(0,1)D,(1,0)D,(1,1)D\}$

$\mathrm{LSP}=\{\ \}$

SPIHT编码的主要步骤

（2）排序扫描

由以下两个大的步骤构成：

1） 顺次检查LIP中的所有小波系数 (i, j)，确定其是否重要

- 如果是重要的系数，则输出“1”及其符号位，其中正、负小波系数的符号位分别采用“1”和“0”表示，然后将该系数从LIP中删除，并添加到有序表LSP的尾部。
- 如果是不重要的系数，则输出“0”。

排序扫描1	输出
T＝32	
%对LIP中的每个表项顺次进行处理	
Is （0,0） significant？yes：	1
	1（符号位）
/将（0，0）从LIP中删除，添加到LSP的尾部/	
LSP＝{（0,0）}	
LIP＝{（0,1），（1,0），（1,1）}	

SPIHT编码的主要步骤

Is （0,1） significant？yes：	1
	0（符号位）
LSP＝{（0,0），（0,1）}	
LIP＝{（1,0），（1,1）}	
Is （1,0） significant？no：	0
Is （1,1） significant？no：	0

（2）排序扫描

2) 对LIS中的每个表项顺次处理，并对D型表项和L型表项分别采用不同的处理方法，具体算法如下：

SPIHT编码的主要步骤

Check the significance of all trees in the LIS according to the type of tree type:

√ For a tree of type D:

° If it is significant, output 1, and code its children:

• If a child is significant, output 1, then a sign bit and add it to the LSP

• If a child is insignificant, output 0 and add it to the end of LIP.

• If the children have descendants, move the tree to the end of LIS as

type L, otherwise remove it from LIS.

° If it is insignificant, output 0.

√ For a tree of type L:

° If it is significant, output 1, add each of the children to the end of LIS as

an entry of type D and remove the parent tree from the LIS.

° If it is insignificant, output 0.

SPIHT编码的主要步骤

三个有序表LIP,LSP,LIS的当前状态信息,即

LSP={（0,0）,（0,1）,（0,2）,（4,3）}

LIP={（1,0）,（1,1）,（0,3）,（1,2）,（1,3）,（2,0）,（2,1）,（3,0）,

（3,1）,（4,2）,（5,2）,（5,3）}

LIS={（1,1）D,（0,1）L,（2,0）D,（3,0）D,（3,1）D}

（3）精细扫描

对于LSP中的每个表项 (i, j)，若 (i, j) 不是在刚刚进行过的扫描过程（2）中新添加的，则输出 $|c_{ij}|$ 的第n个最重要的位，其中 $T = 2^n$是扫描过程中设定的阈值。

例子(续): 由于排序扫描1进行之前，LIS= { },故没有符号位输出。

（4）进行下一次排序扫描和精细扫描

例5.6第一次SPIHT编码后输出的信息

第一次编码过程完成后，编码器输出两类信息：

1) 给解码器的信息，包括域值 2^n 、排序扫描的输出位流 S_n：

1 1 1 0 0 0 1 1 1 0 0 0 1 0 0 0 0 0 0 1 0 1 0 1 1 0 0 0 0、

精细扫描位流及三个有序表的初始化信息，即LIP ,LIS和LSP

LIP＝{（0,0），（0,1），（1,0），（1,1）}
LIS={（0,1）D,（1,0）D,（1,1）D}
LSP={ }

2) 用于下次扫描的信息，包括域值 2^n、三个有序表LIP,LSP,LIS的当前状态信息.

SPIHT解码过程

为获得SPIHT解码器算法，只需将编码器输出的位流、初始值以及三个控制表LIS、LIP、LSP的初始化信息提供给解码器，并执行编码器的相同路径即可。为此，只需将SPIHT编码器代码中的输出（output）改为输入（input）即可。这样，解码器可恢复数据的排序信息。

解码器需要做的另一项工作是更新重构的图像。对于阈值 $T=2^n$，当一个坐标 (i,j) 移到LSP中时，这表明，$2^n \le |c_{i,j}| < 2^{n+1}$ 。再由随后输入的符号位信息，可获得 (i,j) 的重构值 $\hat{c}_{i,j} = \pm 1.5 \times 2^n$ 。进一步根据精细扫描过程中 $|c_{i,j}|$ 二进表示的位信息以获得更精确的重构信息。

SPIHT解码过程举例----例5.7

SPIHT解码器第一次编码为解码器提供的信息如下：

位流 S_5: 1 1 1 0 0 0 1 1 1 0 0 0 1 0 0 0 0 0 0 1 0 1 0 1 1 0 0 0 0

精细扫描表：空集

初始n: n=5

控制表初始化信息：

LIP＝{（0,0），（0,1），（1,0），（1,1）}

LIS= {（0,1）D,（1,0）D,（1,1）D}

LSP={ }

解码器的执行过程：

排序扫描1：

设定阈值T=2^5=32

%对LIP中的每个表项顺次进行处理

%对LIS中每个表项顺次进行处理

精细扫描1：

从精细扫描表读取位数据，修改LSP中相应系数的重构值。

48	−48	48	0	×	×	×	×
0	0	0	0	×	×	×	×
0	0	×	×	×	×	×	×
0	0	×	×	×	×	×	×
×	×	0	48	×	×	×	×
×	×	0	0	×	×	×	×
×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×