图 象 处 理

（第3版）

章毓晋

清华大学电子工程系 100084 北京

第3单元　图象编码

- 第8章　图象编码基础
- 第9章　图象变换编码
- 第10章　其他图象编码方法

图象编码的目的是在保证一定视觉质量的前提下减少数据量（从而也减少图象传输所需的时间），这也可看作使用较少的数据量来获得较好的视觉质量。图象编码以信息论为基础，以压缩数据量为主要目的，所以图象编码也常被称为图象压缩。

第9章 图象变换编码

- 9.1 可分离和正交图象变换
- 9.2 离散余弦变换
- 9.3 正交变换编码
- 9.4 小波变换
- 9.5 小波变换编码

9.1 可分离和正交图象变换

2-D变换

一般形式

正向变换核

$$T(u,v)=\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}f(x,y)h(x,y,u,v)$$

变换核与

原始函数及

变换后函数无关

$$f(x,y)=\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}T(u,v)k(x,y,u,v)$$

反向变换核

9.1 可分离和正交图象变换

可分离（傅里叶变换是一个例子）

$$\boxed{h(x,y,u,v)=h_1(x,u)h_2(y,v)}$$

1个2-D变换可分解成2个1-D变换

$$T(x,v)=\sum_{y=0}^{N-1}f(x,y)h_2(y,v) \qquad T(u,v)=\sum_{x=0}^{N-1}T(x,v)h_1(x,u)$$

对称

$$\boxed{h(x,y,u,v)=h_1(x,u)h_1(y,v)}$$

(h_1与h_2的函数形式一样)

9.1 可分离和正交图象变换

可分离且对称

对称变换矩阵

变换结果

$$T = AFA$$

图像矩阵

反变换

$$BTB = BAFAB$$

反变换矩阵

$$F = BTB \qquad B = A^{-1}$$

$$\hat{F} = BAFAB \qquad B \neq A^{-1}$$

9.1 可分离和正交图象变换

正交变换

考虑变换矩阵： $\boldsymbol{B} = \boldsymbol{A}^{-1}$ $\quad \boldsymbol{F} = \boldsymbol{BTB}$

酉矩阵（*代表共轭 ）： $\boldsymbol{A}^{-1} = \boldsymbol{A}^{*\mathrm{T}}$

如果$\boldsymbol{A}$为实矩阵，且： $\boldsymbol{A}^{-1} = \boldsymbol{A}^{\mathrm{T}}$

则$\boldsymbol{A}$为正交矩阵

式(9.1.1)和式(9.1.2)构成正交变换对

9.2 离散余弦变换

1. 变换定义

一种可分离、正交、对称的变换

1-D离散余弦变换（1D−DCT）

$$C(u) = a(u)\sum_{x=0}^{N-1} f(x)\cos\left[\frac{(2x+1)u\pi}{2N}\right] \qquad u = 0, 1, \cdots, N-1$$

$$f(x) = \sum_{u=0}^{N-1} a(u)C(u)\cos\left[\frac{(2x+1)u\pi}{2N}\right] \qquad x = 0, 1, \cdots, N-1$$

$$a(u) = \begin{cases} \sqrt{1/N} & u = 0 \\ \sqrt{2/N} & u = 1,\ 2,\ \cdots,\ N-1 \end{cases}$$

9.2　离散余弦变换

2-D离散余弦变换（2D-DCT）

$$C(u,v)=a(u)a(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}f(x,y)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

$$f(x,y)=\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}a(u)a(v)C(u,v)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

+ 讨论可分离性和对称性

$$h(x,y,u,v)=h_1(x,u)h_2(y,v) \qquad h(x,y,u,v)=h_1(x,u)h_1(y,v)$$

9.2 离散余弦变换

2. 变换计算

借助离散傅里叶变换的实部计算来进行

$$C(u)=a(u)\left\{\exp[-\mathrm{j}\pi u/(2N)]\mathcal{F}[g(x)]\right\} \qquad u=0,\ 1,\ \cdots,\ N-1$$

$$g(x)=\begin{cases} f(2x) & x=0,1,\cdots,N/2-1 \\ f[2(N-1-x)+1] & x=N/2,N/2+1,\cdots,N-1 \end{cases}$$

$g(x)$的前半部分是$f(x)$的偶数项，$g(x)$的后半部分是$f(x)$的奇数项的逆排

9.3 正交变换编码

9.3.1　正交变换编码系统

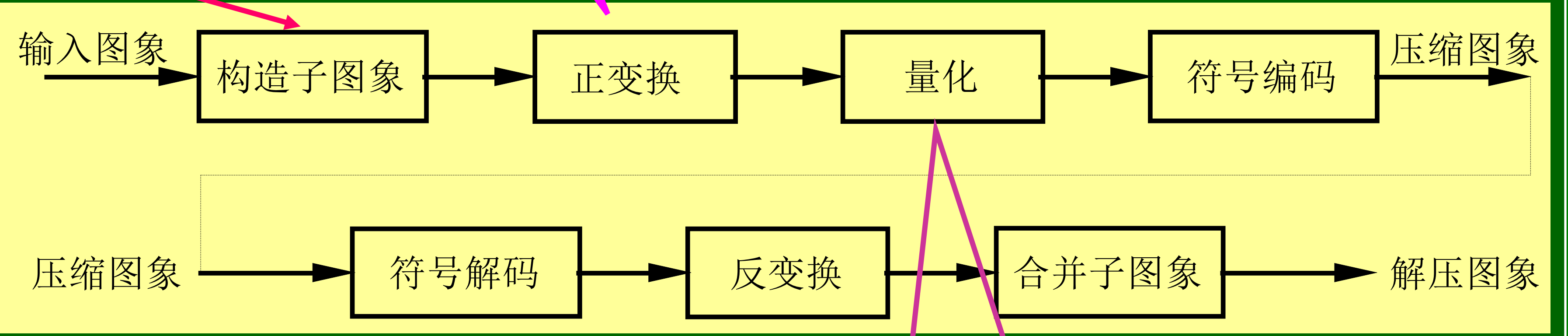
9.3.2　子图象尺寸选择

9.3.3　变换选择

9.3.4　比特分配

9.3.1 正交变换编码系统

- 图象分解：减少变换的计算复杂度
- 图象变换：解除每个子图象内部象素之间的相关性，或者说将尽可能多的信息集中到尽可能少的变换系数上



压缩不是在变换中而是在量化变换系数时取得的

9.3.2 子图象尺寸选择

- 影响变换编码误差和计算复杂度

  （压缩量和计算复杂度都随子图象尺寸的增加而增加）

- 两个条件/考虑：

  ① 相邻子图象之间的相关（冗余）减少到某个可接受的水平；

  ② 子图象的长和宽都是2的整数次幂

最常用的子图象尺寸：8×8和16×16

9.3.4 比特分配

比特分配：对变换子图象的系数截断、量化和编码的全过程

截断误差

① 截除的变换系数的数量和相对重要性

② 用来表示所保留系数的精度（量化）

保留系数的2个准则

① 最大方差准则，称为分区编码

② 最大幅度准则，称为阈值编码

9.3.4 比特分配

1. 分区编码

具有最大方差的变换系数带有最多的图象信息

事先确定模板并保留一定的系数，即分区

1	1	1	1	1	0	0	0
1	1	1	1	0	0	0	0
1	1	1	0	0	0	0	0
1	1	0	0	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

(a)

8	7	6	4	3	2	1	0
7	6	5	4	3	2	1	0
6	5	4	3	3	1	1	0
4	4	3	3	2	1	0	0
3	3	3	2	1	1	0	0
2	2	1	1	1	0	0	0
1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0

(b)

图 9.3.4 典型的分区模板和分区比特分配（有阴影的系数为保留的系数）

9.3.4 比特分配

2. 阈值编码

根据子图象特性自适应选择保留系数

将系数排队，与阈值比较确定去舍（游程/变长码）

1	1	1	1	0	1	0	0
1	1	1	1	0	0	0	0
1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	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

(a)

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

(b)

图 9.3.5 典型的阈值模板和取阈值系数序列（有阴影的系数为保留的系数）

9.3.4 比特分配

2. 阈值编码

随子图象不同而保留不同位置的变换系数

➢ 常用三种对变换子图象取阈值（即产生式(9.3.3)所示模板函数）的方法：

(1) 对所有子图象用一个全局阈值

压缩的程度随（不同）子图象而异

(2) 对各个子图象分别用不同的阈值

舍去同数量系数，码率是个常数

9.3.4 比特分配

2. 阈值编码

(3) 根据子图象中系数的位置选取阈值

将取阈值和量化结合起来

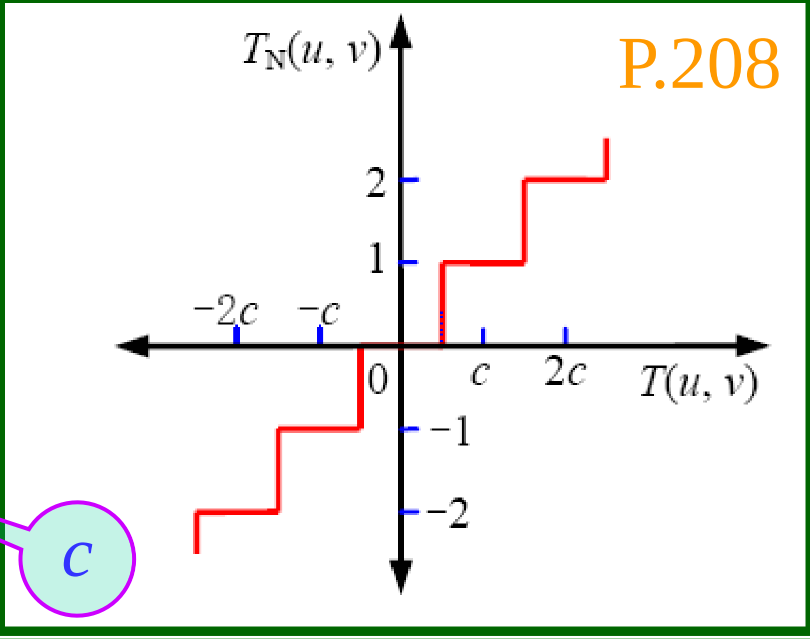
$$\hat{\boldsymbol{F}} = \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} T(u,v) m(u,v) \boldsymbol{H}_{uv}$$

$$\Downarrow$$

$$T_{\mathrm{N}}(u,v) = \mathrm{round}\left[\frac{T(u,v)}{N(u,v)}\right]$$

$$T_{\mathrm{A}}(u,v) = T_{\mathrm{N}}(u,v) N(u,v)$$

c



9.4 小波变换

9.4.1 小波变换基础

9.4.2 1-D小波变换

9.4.3 快速小波变换

9.4.4 2-D小波变换

9.4.1 小波变换基础

1. 序列展开

对偶函数

$$f(x)=\sum_k a_k u_k(x) \qquad a_k=\langle u'_k(x), f(x)\rangle=\int u'^{*}_k(x)f(x)d$$

a_k是实数，称为展开系数，$u_k(x)$是实数，称为展开函数

(1) 展开函数构成空间U的正交归一化基，$u_k(x) = u'_k(x)$

$$\langle u_j(x),u_k(x)\rangle=\delta_{jk}=\begin{cases}0 & j\neq k\\ 1 & j=k\end{cases} \qquad a_k=\langle u_k(x), f(x)\rangle$$

(2) 展开函数仅构成空间U的正交基，但没有归一化

$$\langle u_j(x),u_k(x)\rangle=0 \quad j\neq k \qquad \langle u_j(x),u'_k(x)\rangle=\delta_{jk}=\begin{cases}0 & j\neq k\\ 1 & j=k\end{cases}$$

9.4.1 小波变换基础

1. 序列展开

$$f(x)=\sum_k a_k u_k(x)$$

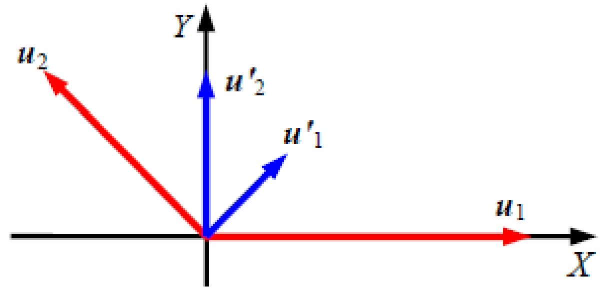


图 9.4.1 双正交性示例

基：展开函数的集合$\{u_k(x)\}$

函数空间：由所有函数 $f(x)$构成

双正交基：（几何矢量解释，例9.4.1）

例：双正交基 $u_1=[2\ \ 0]^{\mathrm{T}}$， $u_2=[-1\ \ 1]^{\mathrm{T}}$

对偶基为$u'_1=[1/2\ \ 1/2]^{\mathrm{T}}$，$u'_2=[0\ \ 1]^{\mathrm{T}}$

$\langle u_1,u'_1\rangle=1$ $\langle u_2,u'_1\rangle=0$ $\langle u_1,u'_2\rangle=0$ $\langle u_2,u'_2\rangle=1$

9.4.1 小波变换基础

2. 缩放函数

- 用展开函数作为缩放函数，并对其进行平移和2进制缩放

$$u_{j,k}(x) = 2^{j/2} u(2^j x - k)$$

- k确定了$u_{j,k}(x)$沿X-轴的位置，j确定了$u_{j,k}(x)$沿X-轴的宽度（所以$u(x)$也称为尺度函数），系数$2^{j/2}$控制$u_{j,k}(x)$的幅度
- 给定一个初始 j（下面常取为0），就可确定一个缩放函数空间U_j，U_j的尺寸随 j 的增减而增减

9.4.1 小波变换基础

2. 缩放函数

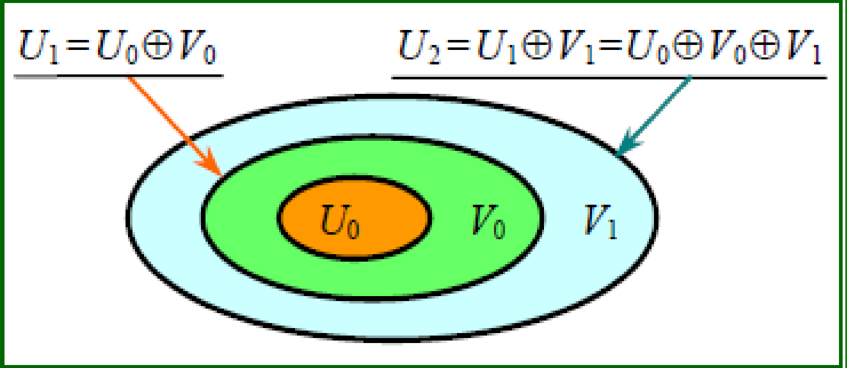
- 各个缩放函数空间U_j，$j = -\infty, \ldots, 0, 1, \ldots, \infty$是嵌套的，即$U_j \subset U_{j+1}$，$U_j$中的展开函数可以表示成$U_{j+1}$中展开函数的加权和

- 用$h_u(k)$表示缩放函数系数，因为$u(x) = u_{0,0}(x)$

多分辨率细化方程 $\boxed{u(x) = \sum_k h_u(k)\sqrt{2}u(2x-k)}$

任何一个子空间的展开函数都可用其下一个分辨率（1/2分辨率）的子空间的展开函数来构建

3. 小波函数



- 用$v(x)$表示小波函数

$$v_{j,k}(x)=2^{j/2}v(2^j x-k)$$

- 与$v_{j,k}(x)$对应的空间为V_j

$$f(x)=\sum_k a_k v_{j,k}(x)$$

- 空间U_j，U_{j+1}和V_j有如下关系（⊕表示空间的并）

$$U_{j+1}=U_j\oplus V_j$$

- 在U_{j+1}中，U_j的补是V_j

9.4.1 小波变换基础

3. 小波函数

- 每一个V_j空间是与其同一级的U_j空间和上一级的U_{j+1}空间的差

$$U_{j+1} = U_j \oplus V_j$$

- 如果考虑把j取到趋近$-\infty$，则有可能仅用小波函数，而完全不用缩放函数来表达所有的$f(x)$

- U_j中所有$u_{j,k}(x)$与V_j中所有$v_{j,k}(x)$是正交的

$$\left\langle u_{j,k}(x), v_{j,k}(x) \right\rangle = 0$$

9.4.1 小波变换基础

4. 缩放函数和小波函数示例

哈尔变换的基本函数是最简单的正交归一化小波

单位高度和单位宽度的缩放函数

$$u(x)=\begin{cases}1 & 0\le x<1\\ 0 & \text{其他}\end{cases}$$

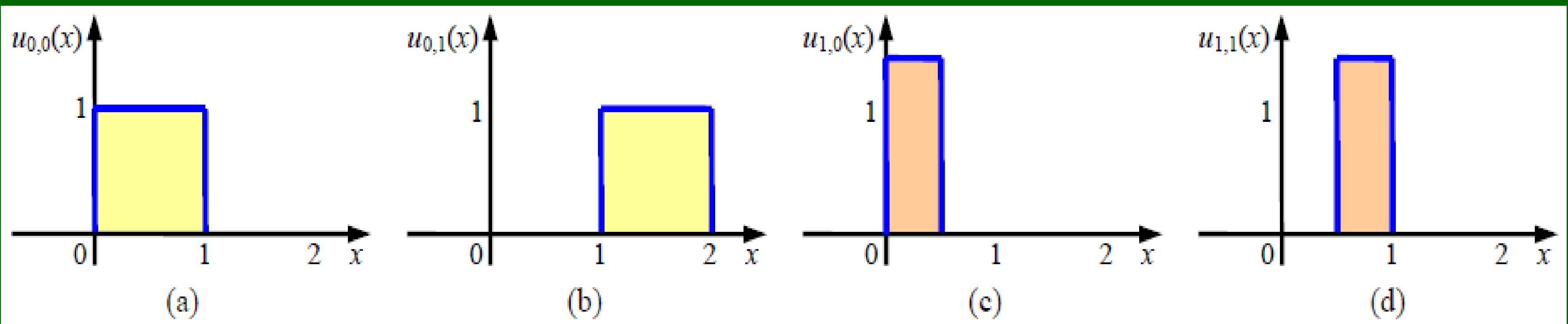


图 9.4.3 U_0 和 U_1 中的缩放函数

4. 缩放函数和小波函数示例

- 随着 j 的增加，缩放函数变窄变高

图9.4.4：仅用 j =0的缩放函数不够，还需要 j =1的缩放函数

$$f(x)=u_{1,1}(x)+0.5[u_{1,2}(x)+u_{1,3}(x)]+0.75[u_{1,5}(x)+u_{1,6}(x)]$$

$f(x)$是属于U_1的，

而不是属于U_0的

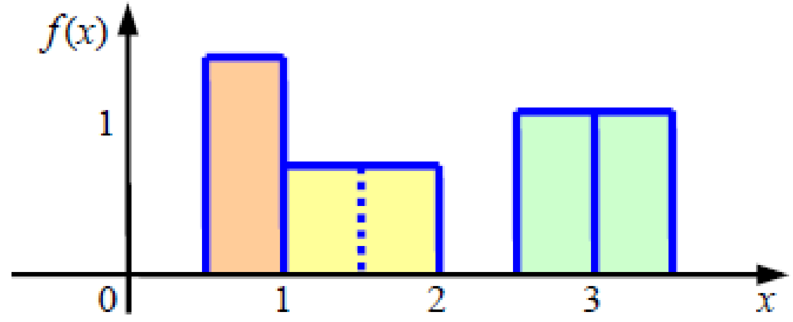


图 9.4.4 用缩放函数表示属于 U_1 的 $f(x)$

4. 缩放函数和小波函数示例

哈尔小波函数 $v(x)=\begin{cases} 1 & 0\le x<0.5 \\ -1 & 0.5\le x<1 \\ 0 & \text{其它} \end{cases}$

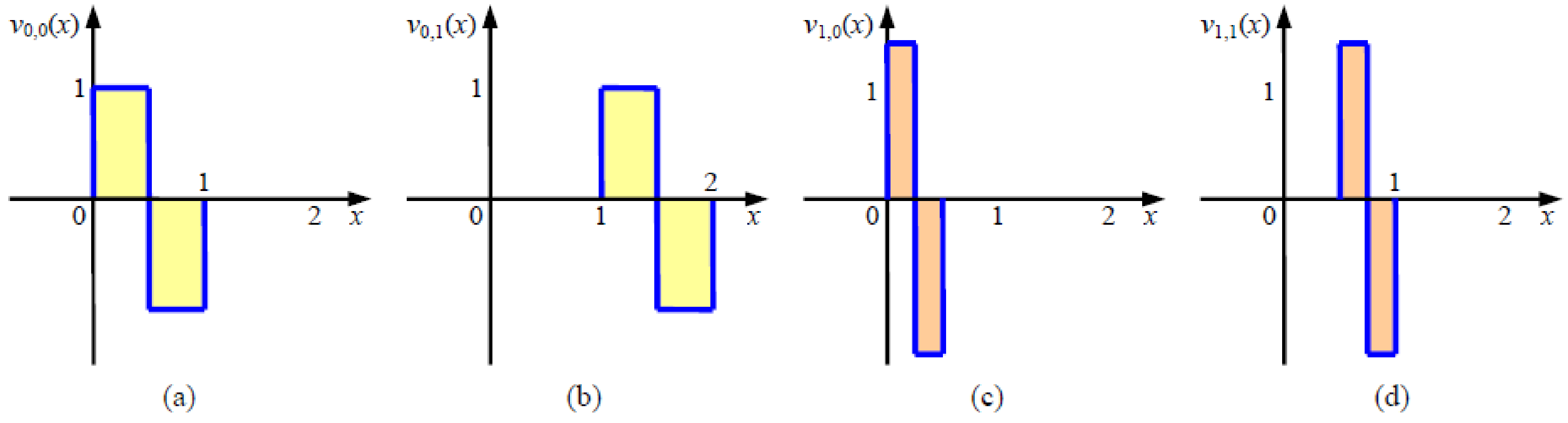


图 9.4.5 V_0 和 V_1 中的小波函数

例9.4.3

9.4.2 1-D小波变换

1. 小波序列展开

对给定的函数 $f(x)$，可以用$u(x)$和$v(x)$对它进行展开

$$f(x) = \sum_k a_0(k) u_{0,k}(x) + \sum_{j=0}^{\infty} \sum_k d_j(k) v_{j,k}(x)$$

$a_0(k)$：缩放系数

$$a_0(k) = \left\langle f(x), u_{0,k}(x) \right\rangle = \int f(x) u_{0,k}(x) \mathrm{d}x$$

$d_j(k)$：小波系数

$$d_j(k) = \left\langle f(x), v_{j,k}(x) \right\rangle = \int f(x) v_{j,k}(x) \mathrm{d}x$$

9.4.2 1-D小波变换

2. 离散小波变换

如果$f(x)$是一个离散序列，展开得到的系数称为$f(x)$的离散小波变换（DWT）

$$f(x)=\frac{1}{\sqrt{M}}\sum_{x}W_u(0,k)u_{0,k}(x)+\frac{1}{\sqrt{M}}\sum_{j=0}^{\infty}\sum_{x}W_v(j,k)v_{j,k}(x)$$

近似系数 $$W_u(0,k)=\frac{1}{\sqrt{M}}\sum_{x}f(x)u_{0,k}(x)$$

细节系数 $$W_v(j,k)=\frac{1}{\sqrt{M}}\sum_{x}f(x)v_{j,k}(x)$$

9.4.3 快速小波变换

- 在尺度 j 上的系数 $W_u(j, k)$ 和 $W_v(j, k)$ 都可用在尺度 j+1的近似系数 $W_u(j+1, k)$ 分别与缩放矢量 h_u 和小波矢量 h_v 卷积再进行亚抽样得到

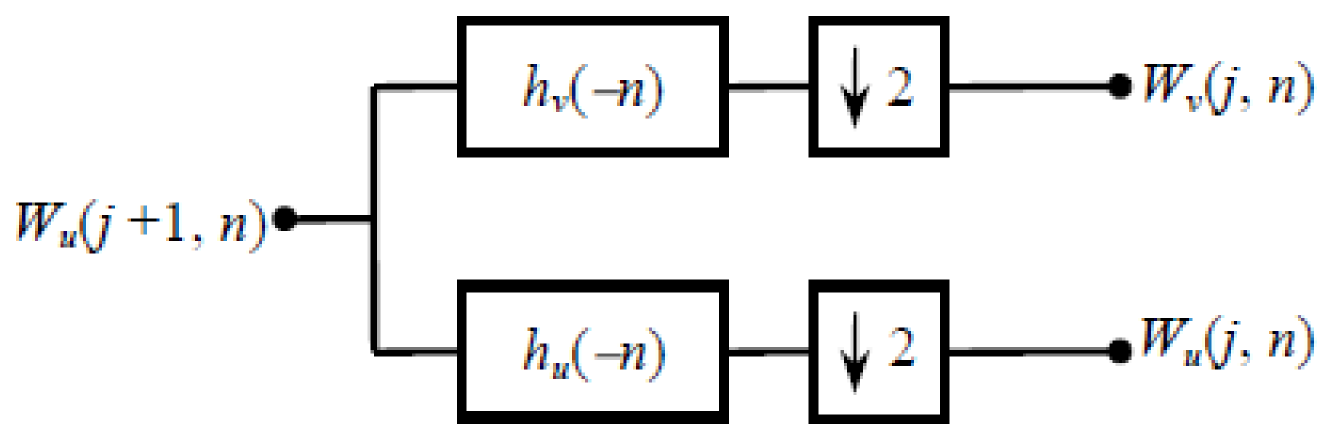


图 9.4.7　正变换的分析方框图

9.4.4 2-D小波变换

1. 2-D变换函数

需要1个2-D缩放函数$u(x, y)$和3个2-D小波函数$v^{H}(x, y)$，$v^{V}(x, y)$，$v^{D}(x, y)$，每一个都是1-D缩放函数u和对应的小波函数v的乘积

可分离的缩放函数 $u(x,y)=u(x)u(y)$

水平边缘 $v^{H}(x,y)=v(x)u(y)$

垂直边缘 $v^{V}(x,y)=u(x)v(y)$

沿对角线的变化 $v^{D}(x,y)=v(x)v(y)$

9.4.4 2-D小波变换

- 缩放和平移的基函数

$$u_{j,m,n}(x,y)=2^{j/2}u(2^{j}x-m,\ \ 2^{j}y-n)$$

$$v^{(i)}_{j,m,n}(x,y)=2^{j/2}v^{(i)}(2^{j}x-m,\ \ 2^{j}y-n)\quad (i)=\{\mathrm{H,V,D}\}$$

- 离散小波变换

$$W_{u}(0,m,n)=\frac{1}{\sqrt{MN}}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)u_{0,m,n}(x,y)$$

$$W_{v}^{(i)}(j,m,n)=\frac{1}{\sqrt{MN}}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)v^{(i)}_{j,m,n}(x,y)\quad (i)=\{\mathrm{H,\ V,\ D}\}$$

9.5 小波变换编码

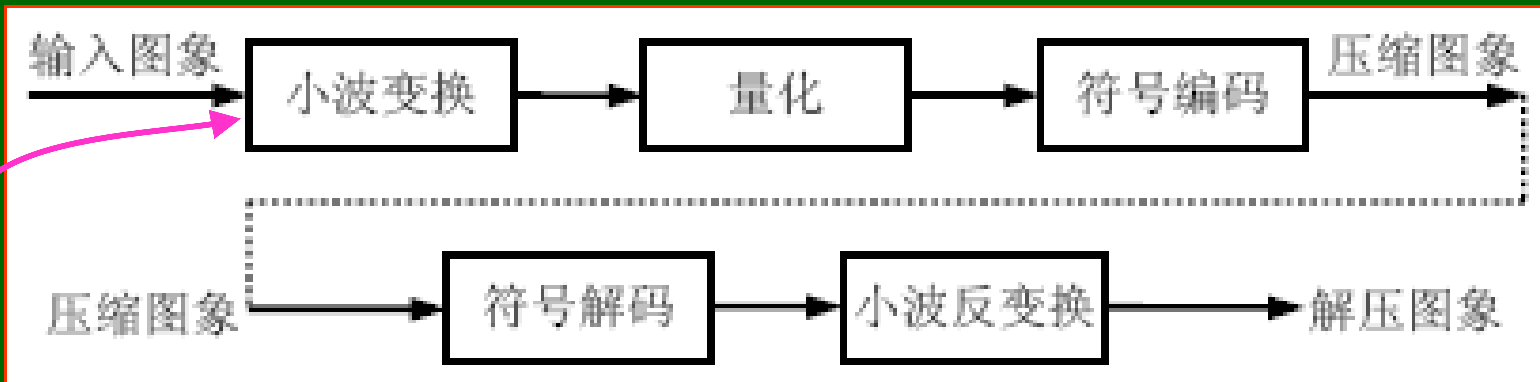
{在JPEG-2000及MPEG-4和H.264中都得到了应用}

9.5.1 小波变换编解码系统

9.5.2 基于提升小波的编码

9.5.1 小波变换编解码系统

➢ 小波变换编码也是一种变换编码方式



- 与采用正交变换（如DCT）的编解码系统不同，小波变换编解码系统中没有图象分块的模块
- 小波变换的计算效率很高，且本质上具有局部性
- 小波变换编码不会产生使用DCT变换在高压缩比时出现的块效应

9.5.1 小波变换编解码系统

小波变换编码需考虑的几个因素

1. **小波选择**

   如：哈尔小波、双正交小波

2. **分解层数选择**

   影响小波编码计算的复杂度和重建误差

3. **量化设计**

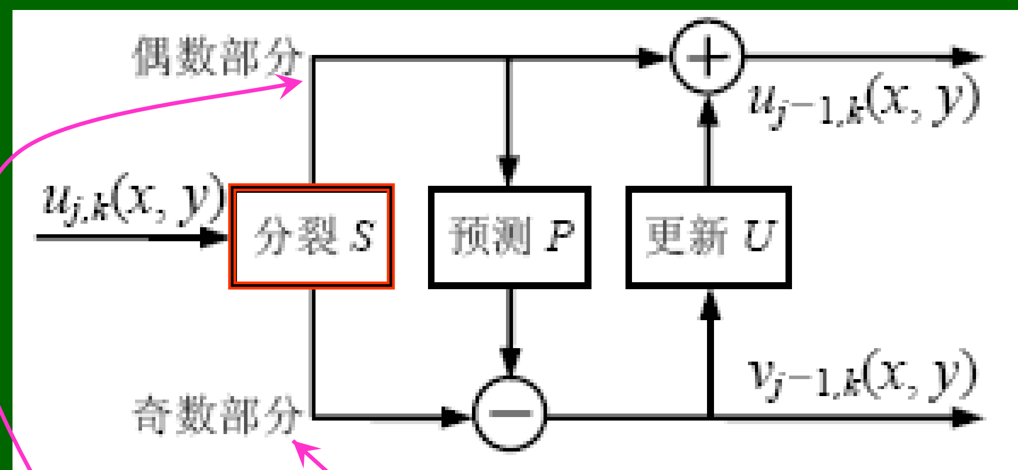
   对小波编码压缩和重建误差影响最大

   需在不同尺度间调整量化间隔 {例：P.327}

9.5.2 基于提升小波的编码

可以在当前位置实现整数到整数的变换，运算速度快且节约内存。它包括三个步骤：

1. 分裂（split）

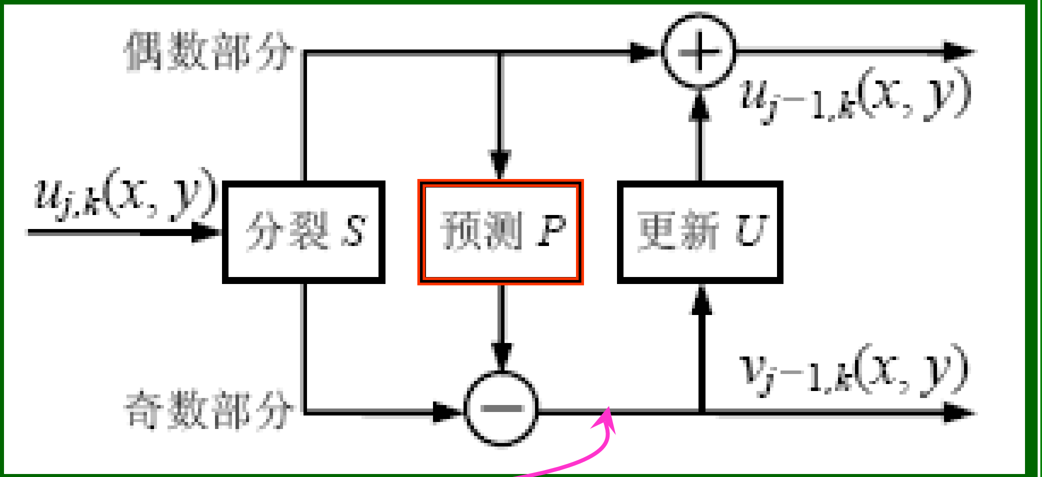


将图象数据分解成偶数部分和奇数部分

$$S\,[u_{j,k}(x, y)] := [u_{j-1,k}(x, y), v_{j-1,k}(x, y)]$$
$$= [u_{j-1,2k}(x, y), u_{j-1,2k+1}(x, y)]$$

2. 预测（predict）

保持偶数部分不变并用偶数部分来预测奇数部分，然后用奇数部分与预测值的差（称为细节系数）替代奇数部分

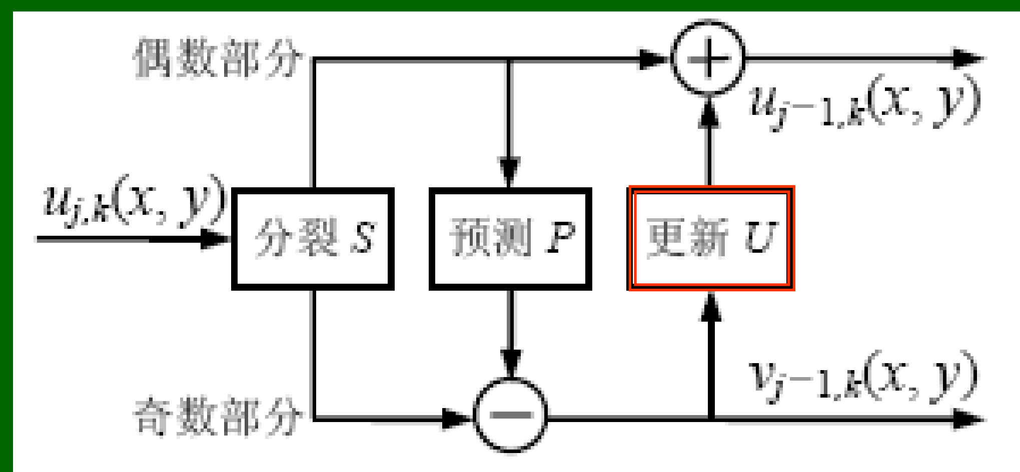


$$v_{j-1,k}(x, y) := v_{j-1,k}(x, y) - P\,[u_{j-1,k}(x, y)]$$

9.5.2 基于提升小波的编码

3. 更新（update）

构造一个作用于细节函数的算子U，并叠加到偶数部分上以获得近似图象，这里要保持原始图象的一些特性



$$u_{j-1,k}(x, y) := u_{j-1,k}(x, y) + U[v_{j-1,k}(x, y)]$$

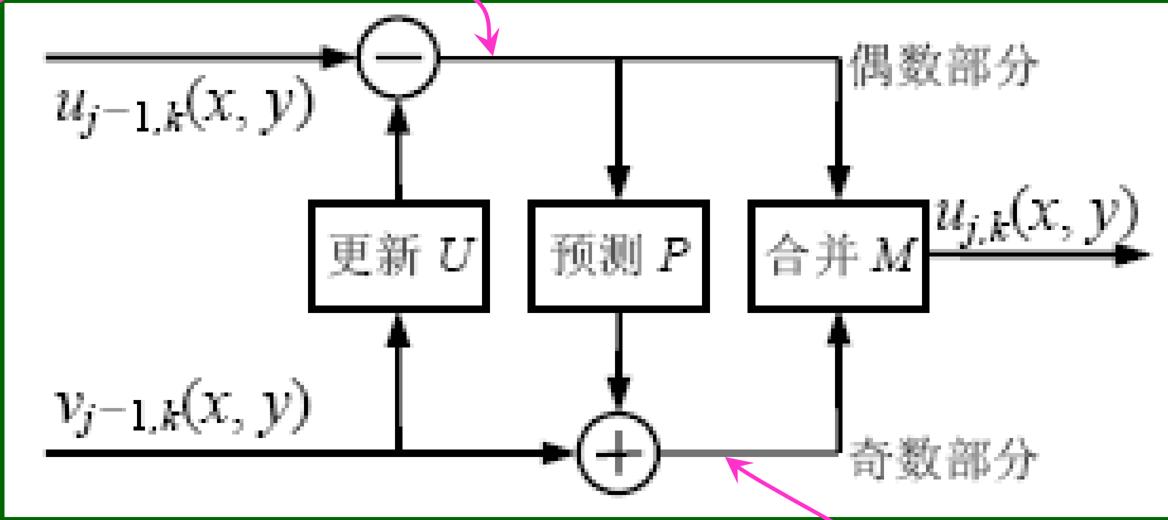
$$Q[u_{j-1,k}(x, y)] = Q[u_{j,k}(x, y)]$$

9.5.2 基于提升小波的编码

重建过程

三个运算：

（M →合并）



(1) $u_{j-1,k}(x, y) := u_{j-1,k}(x, y) - U\,[v_{j-1,k}(x, y)]$

(2) $v_{j-1,k}(x, y) := v_{j-1,k}(x, y) + P\,[u_{j-1,k}(x, y)]$

(3) $u_{j,k}(x, y) := M\,[u_{j-1,k}(x, y), v_{j-1,k}(x, y)]$

联　系　信　息

- ☞ 通信地址：北京清华大学电子工程系
- ☞ 邮政编码：100084
- ☞ 办公地址：清华大学，罗姆楼，6层305室
- ☞ 办公电话：(010) 62798540
- ☞ 传真号码：(010) 62770317
- ☞ 电子邮件：zhang-yj@tsinghua.edu.cn
- ☞ 个人主页：oa.ee.tsinghua.edu.cn/~zhangyujin/