# 图 象 处 理

（第4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9章　图象编码基础
- 第10章　图象变换编码
- 第11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，也常被称为图象压缩


# 第10章  图象变换编码

- ✓ 一大类图象编码方法
- ✓ 主要步骤：
  - 先将图象进行变换
  - 再借助变换结果进行编码
  - 最后使用时需要解码
- ✓ 主要消除象素间冗余和心理视觉冗余，是信息损失型编码
- ✓ 效率和性能与所采用图象变换密切相关


# 第10章 图象变换编码

10.1 可分离和正交图象变换

10.2 离散余弦变换

10.3 正交变换编码

10.4 小波变换

10.5 小波变换编码

# 10.1 可分离和正交图象变换

➢ **2-D变换**

一般形式

正向变换核

变换后函数

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

变换核与

原始函数及

变换后函数无关

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

原始函数

反向变换核


# 10.1 可分离和正交图象变换

- **可分离**（傅里叶变换是一个例子）

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

一个2-D变换可分解成两个1-D变换（图4.2.3）

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

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


# 10.1 可分离和正交图象变换

➢ **对称**（傅里叶变换是一个例子）

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

$h_1$与$h_2$的函数形式一样

正变换

$$\frac{1}{N}\exp[-j2\pi(ux+vy)/N]$$

$$=\frac{1}{\sqrt{N}}\exp[-j2\pi ux/N]\frac{1}{\sqrt{N}}\exp[-j2\pi vy/N]$$

反变换

$$\frac{1}{N}\exp[j2\pi(ux+vy)/N]$$

$$=\frac{1}{\sqrt{N}}\exp[j2\pi ux/N]\frac{1}{\sqrt{N}}\exp[j2\pi vy/N]$$

# 10.1 可分离和正交图象变换

- 可分离且对称

变换结果 → $T = AFA$ ← 对称变换矩阵（指向两个 $A$）；图象矩阵（指向 $F$）

反变换

反变换矩阵 → $BTB = BAFAB$

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

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


# 10.1 可分离和正交图象变换

## ➢ 正交变换

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

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(10.1.1)和式(10.1.2)构成正交变换对


# 10.2 离散余弦变换

## 1. 变换定义

一种可分离、对称、正交的变换

➢ **1-D离散余弦变换（1D−DCT）**

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

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

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


# 10.2　离散余弦变换

- 2-D离散余弦变换（2D-DCT）

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

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

- 变换核的可分离性和对称性

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

**例10.2.1：**　大部分能量集中在（低频）部分系数

# 10.2　离散余弦变换

## 2.　变换计算

借助离散傅里叶变换$\mathcal{F}$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　{P.223}


# 10.3　正交变换编码

10.3.1　正交变换编码系统

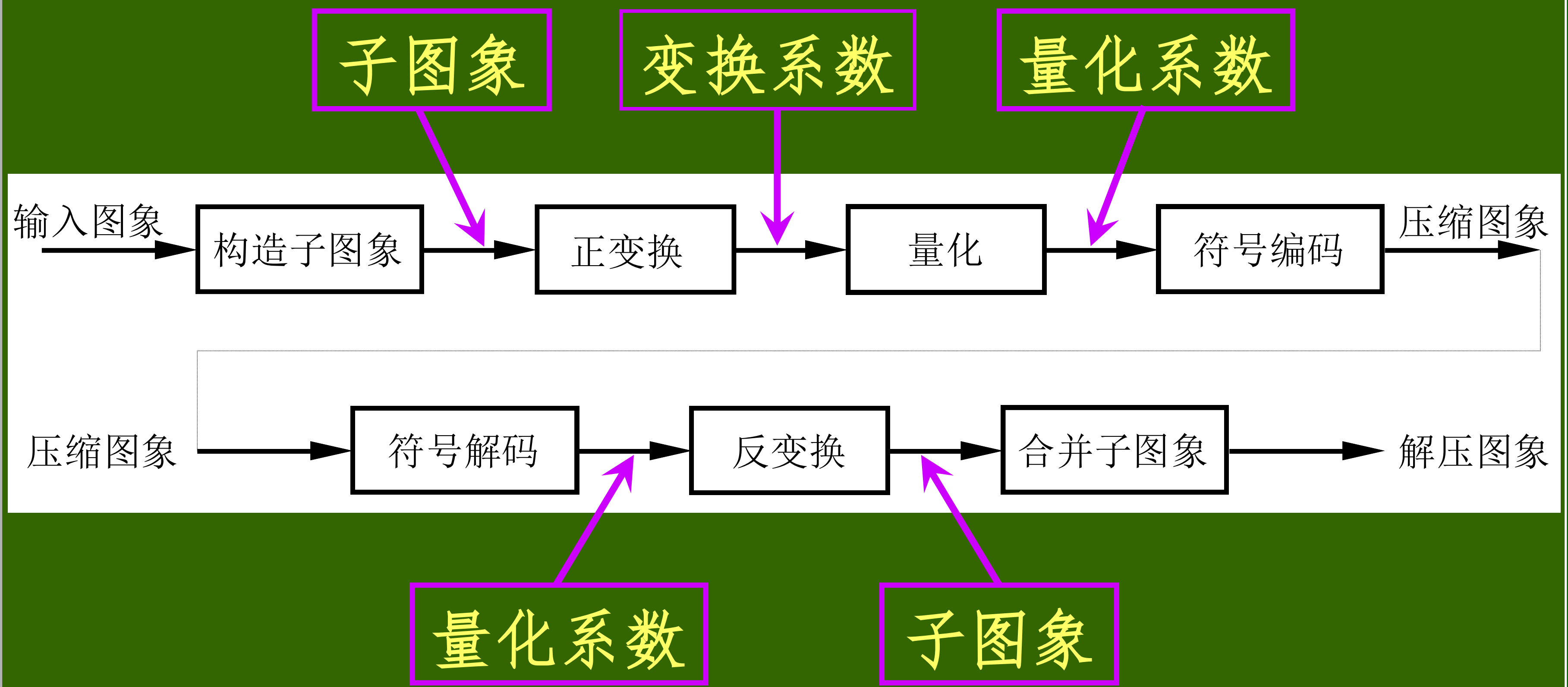
10.3.2　子图象尺寸选择

10.3.3　变换选择

10.3.4　比特分配

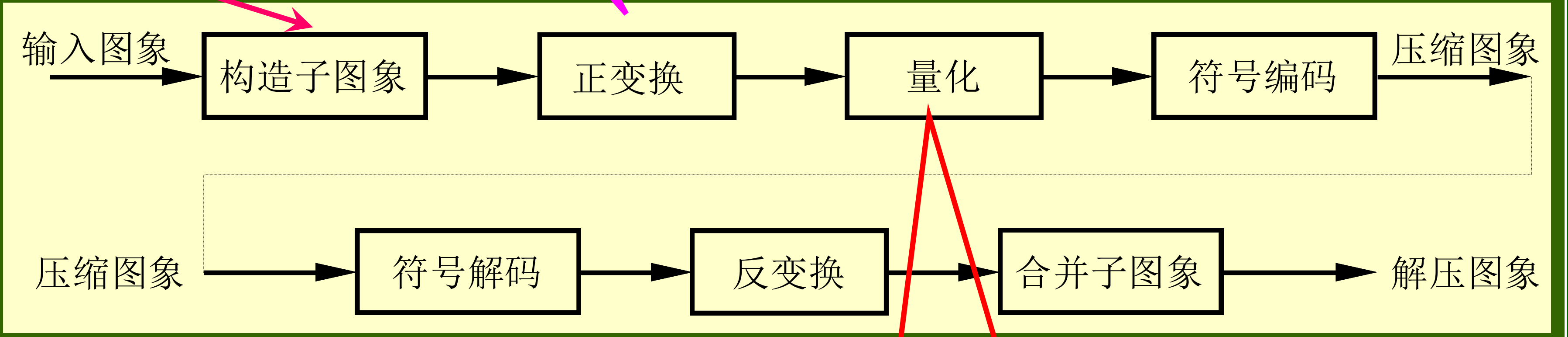
# 10.3.1　正交变换编码系统

## 系统框图




# 10.3.1 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


# 10.3.2　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

➢ （考虑）两个条件：

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：$8 \times 8$和$16 \times 16$

# 10.3.3 变换选择

矩阵形式：

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

模板函数：

$$m(u,v)=\begin{cases}0 & \text{如果}T(u,v)\text{满足特定的截断准则}\\ 1 & \text{其他情况}\end{cases}$$

把
贡献最少
的基本函
数消除掉

截断近似：

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

误差是所有被截除变换系数的方差之和


# 10.3.4　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）

**保留系数的两个准则**

① 最大方差准则，⇒ 分区编码

② 最大幅度准则，⇒ 阈值编码

# 10.3.4　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位置的系数，即分区

保留大方差的系数

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 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 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 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 |

根据方差分配比特

# 10.3.4 比特分配

## 2. 阈值编码

根据子图象特性自适应选择保留（最大）系数

将系数排队，与阈值比较确定去舍（游程/变长码）

根据位置保留系数

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 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 |

之字形扫描排矢量

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 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 |


# 10.3.4　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(10.3.3)所示模板函数）的方法：

(1) 对所有子图象使用同一个全局阈值

压缩的程度随（不同）图象而异

(2) 对各个子图象分别使用不同的阈值

舍去同数量系数，码率是个常数

# 10.3.4 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

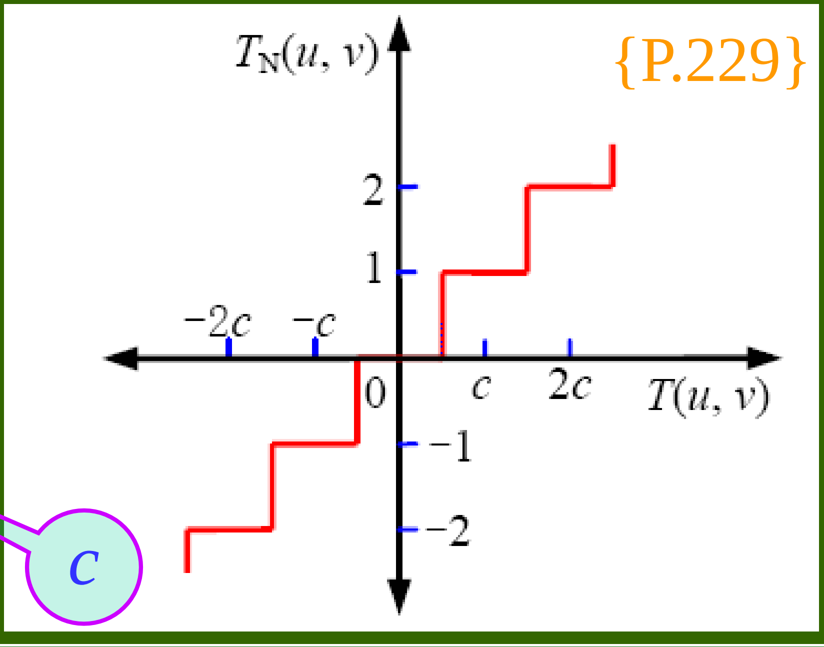
将取阈值和量化结合起来

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

归一化 $T_{\mathrm{N}}(u,v) = \mathrm{round}\left[\dfrac{T(u,v)}{N(u,v)}\right]$

解除归一化 $T_{\mathrm{A}}(u,v) = T_{\mathrm{N}}(u,v)N(u,v)$

$C$




# 10.4 小波变换

10.4.1 小波变换基础

10.4.2 1-D小波变换

10.4.3 快速小波变换

10.4.4 2-D小波变换

# 10.4.1 小波变换基础

## 1. 序列展开

对偶函数

$$f(x)=\sum_k a_k u_k(x) \qquad a_k=\langle u_k'(x), f(x)\rangle=\int u_k'^*(x) f(x)dx$$

$a_k$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

$$\langle u_j(x), u_k(x)\rangle=0 \quad j\neq k$$

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


# 10.4.1 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

# 10.4.1 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 $\boxed{u(x) = \sum_k h_u(k)\sqrt{2}u(2x-k)}$

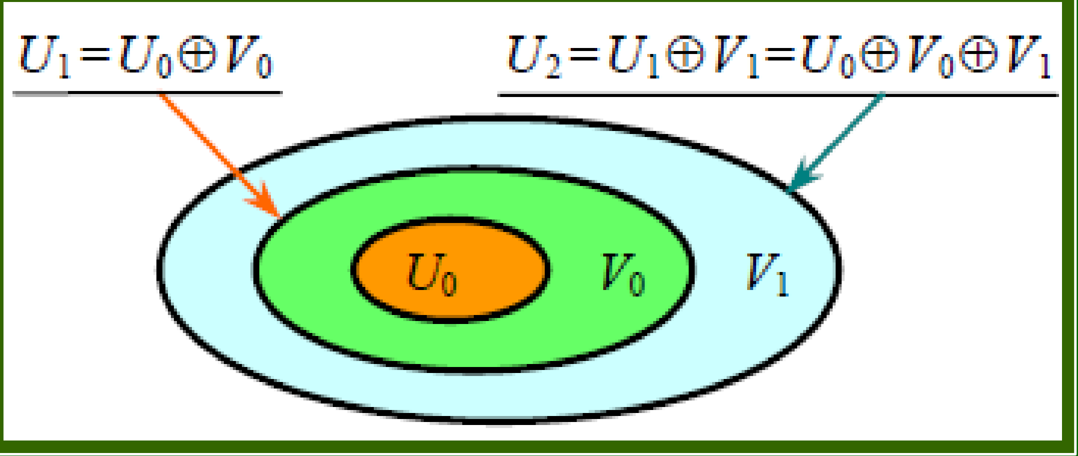
  任何一个子空间的展开函数都可用其下一个分辨率（1/2分辨率）的子空间的展开函数来构建

## 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（$\oplus$表示空间的并）

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

- 在$U_{j+1}$中，$U_j$的补是$V_j$



# 10.4.1 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

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

- 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$f(x)$
- $U_j$中所有缩放函数与$V_j$中所有小波函数正交

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

## 3. 小波函数

- 一个小波函数可表示成其下一个分辨率的各位置缩放函数的加权和：

$$\boxed{v(x)=\sum_{k}h_v(k)\sqrt{2}u(2x-k)}$$

- 缩放函数系数$h_u(k)$和小波函数系数$h_v(k)$具有如下联系：

$$\boxed{h_v(k)=(-1)^k h_u(1-k)}$$

• 分析滤波器

• 合成滤波器


# 10.4.1 小波变换基础

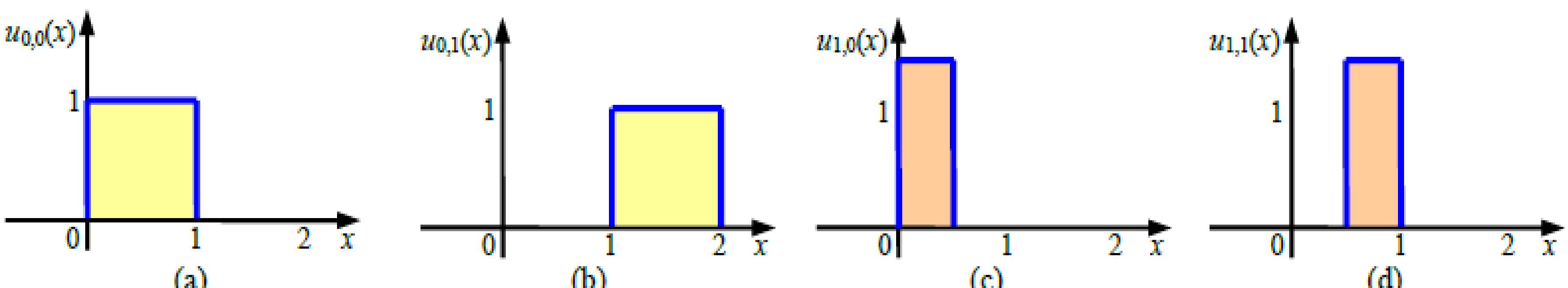
## 4. 缩放函数和小波函数示例

最简单的正交归一化小波（$\Leftarrow$ 哈尔变换）

单位高度和单位宽度的缩放函数

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

正交归一化 (10.4.16)




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

**图 10.4.3 $U_0$ 和 $U_1$ 中的缩放函数**

$$\boxed{u_{1,0}(x)=2^{1/2}u_{0,0}(2x)}$$

# 10.4.1 小波变换基础

## 4. 缩放函数和小波函数示例

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

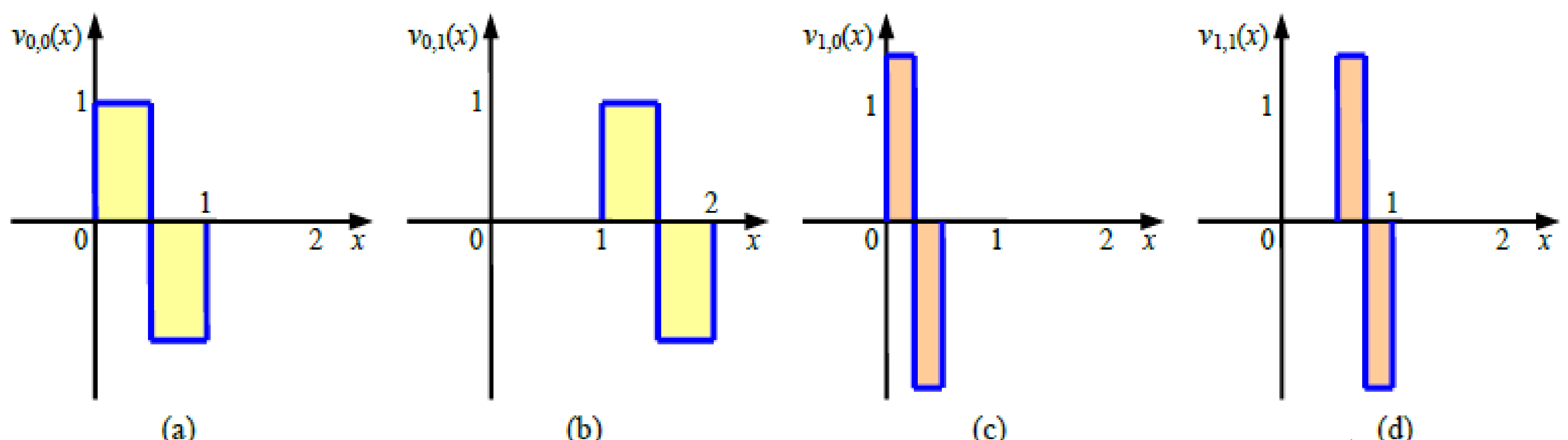



图 10.4.5　$V_0$和 $V_1$中的小波函数

**{例10.4.3}**

# 10.4.1　小波变换基础

## 4.　缩放函数和小波函数示例

母小波函数 ⇒ 小波基函数

平移和二进制缩放可表示为 $v_{j,k}(x) = v\left(\dfrac{x+k}{2^j}\right)$

对它们的双重求和就可表达需要的信号$f(x)$

$$f(x) = \sum_{j=0}^{J} \sum_{k=0}^{K} a_{j,k} v_{j,k}(x)$$

对属于$U_{j+1}$的$f(x)$，可结合使用$U_j$中的缩放函数和$V_j$中的小波函数来表达。将$f(x)$分解成两部分：

$$f(x) = f_{\mathrm{a}}(x) + f_{\mathrm{d}}(x)$$

## 1.　小波序列展开

对给定函数 $f(x)$，可用 $u(x)$ 和 $v(x)$ 对它进行展开

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

$a_0(k)$：缩放系数

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

$d_j(k)$：小波系数

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


# 10.4.2　1-D小波变换

## 2.　离散小波变换

如果$f(x)$是一个离散序列，展开所得到的系数称为$f(x)$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　$$W_v(j,k)=\frac{1}{\sqrt{M}}\sum_{x}f(x)v_{j,k}(x)$$


# 10.4.3　快速小波变换

- 离散小波变换在尺度 $j$ 的细节系数是离散小波变换在尺度 $j$+1的近似系数的函数

$$W_v(j,k)=\sum_n h_v(n-2k)W_u(j+1,n)$$

小波矢量

- 离散小波变换在尺度 $j$ 的近似系数也是离散小波变换在尺度 $j$+1的近似系数的函数

$$W_u(j,k)=\sum_n h_u(n-2k)W_u(j+1,n)$$

缩放矢量

- 给出了相邻尺度间离散小波变换系数的联系

# 10.4.3　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

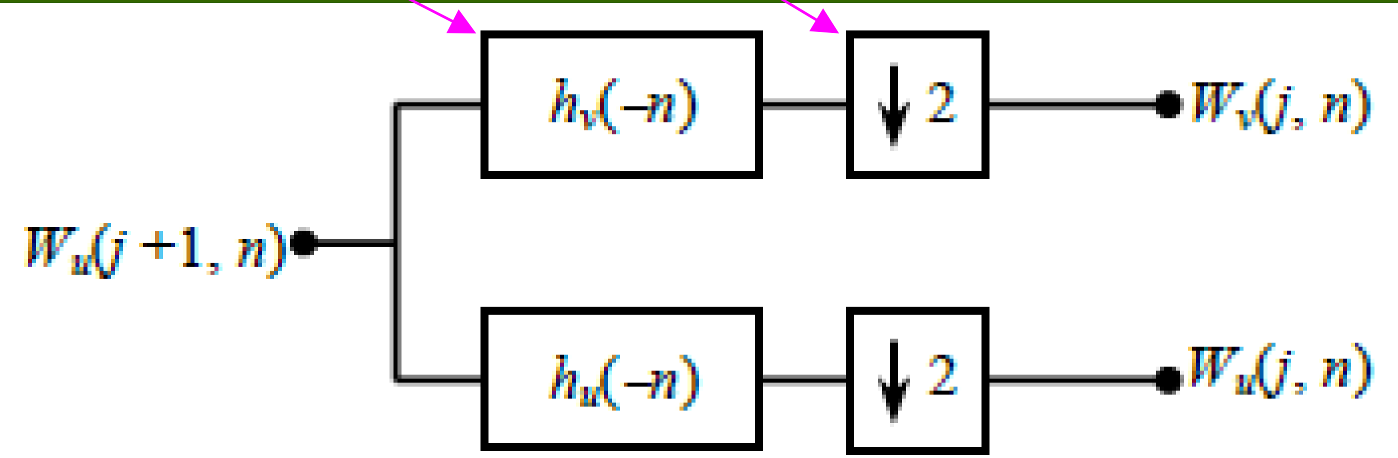


图 10.4.7　正变换的分析方框图

# 10.4.3　快速小波变换

循环计算多级尺度间的离散小波变换系数

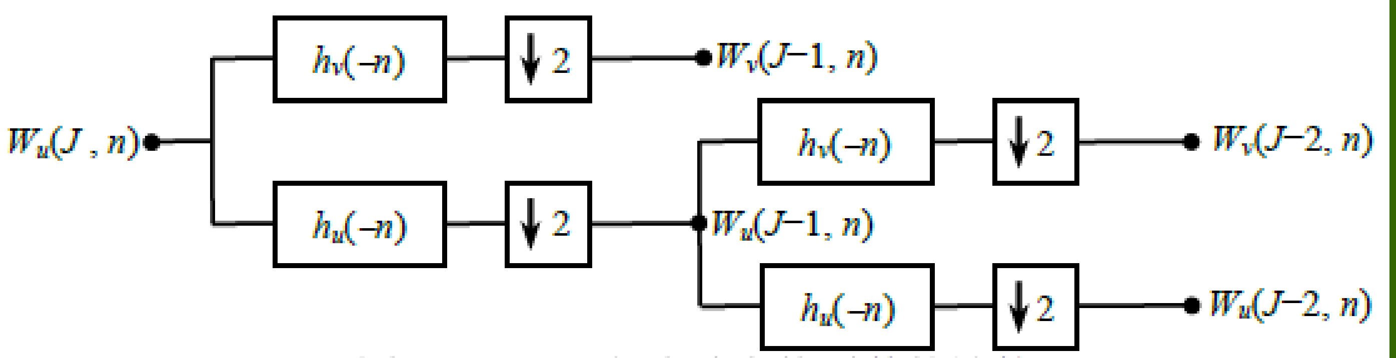



图 10.4.8　两级小波变换系数的计算

尺度空间$U_J$先被第一级计算分解为两半，即小波子空间$V_{J-1}$和尺度子空间$U_{J-1}$

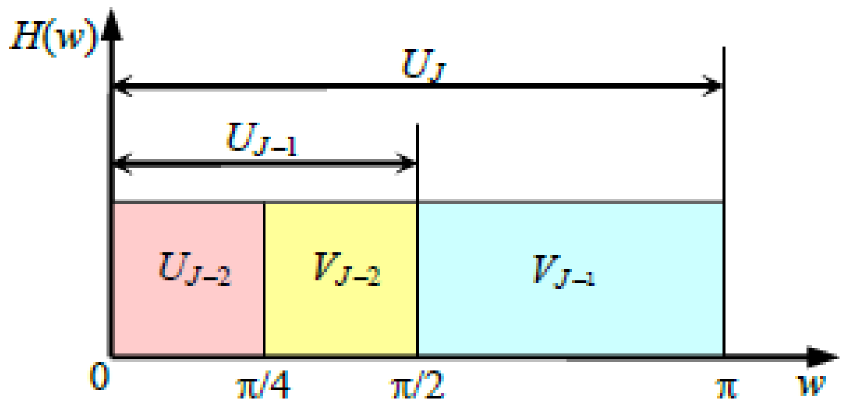



图 10.4.9　频谱空间两级计算效果示意

第 2 级计算再将尺度子空间$U_{J-1}$分成小波子空间$V_{J-2}$以及尺度子空间$U_{J-2}$，……

# 10.4.3　快速小波变换

## 快速小波反变换（$FWT^{-1}$）

利用近似系数$W_u(J, n)$和细节系数$W_v(J, n)$重建$f(x)$也有快速的算法（利用正变换中的缩放矢量和小波矢量以及在尺度$j$上的近似系数和细节系数来产生在尺度$j$+1上的近似系数）

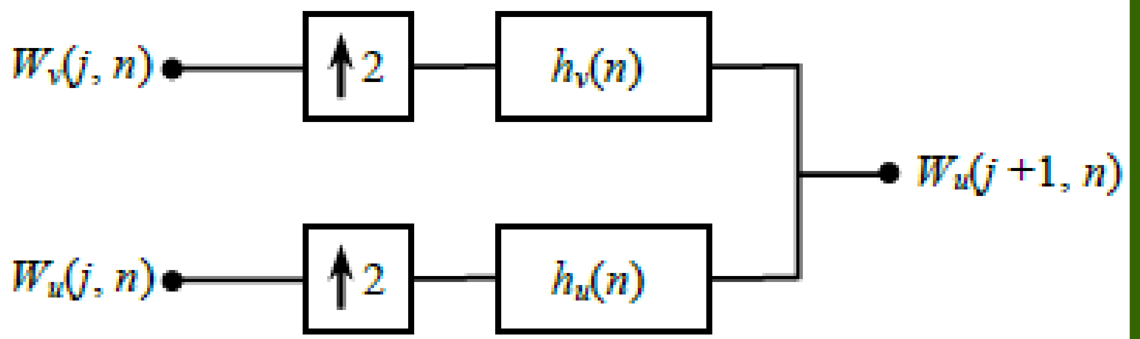



图 10.4.10　反变换的合成方框图


TH-EE-IE

# 10.4.4　2-D小波变换

## 1.　2-D变换函数

需要一个2-D缩放函数$u(x, y)$和三个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：　$u(x,y)=u(x)u(y)$

水平方向小波函数：　$v^{H}(x,y)=v(x)u(y)$

垂直方向小波函数：　$v^{V}(x,y)=u(x)v(y)$

对角方向小波函数：　$v^{D}(x,y)=v(x)v(y)$

## 2. 2-D变换实现和结果

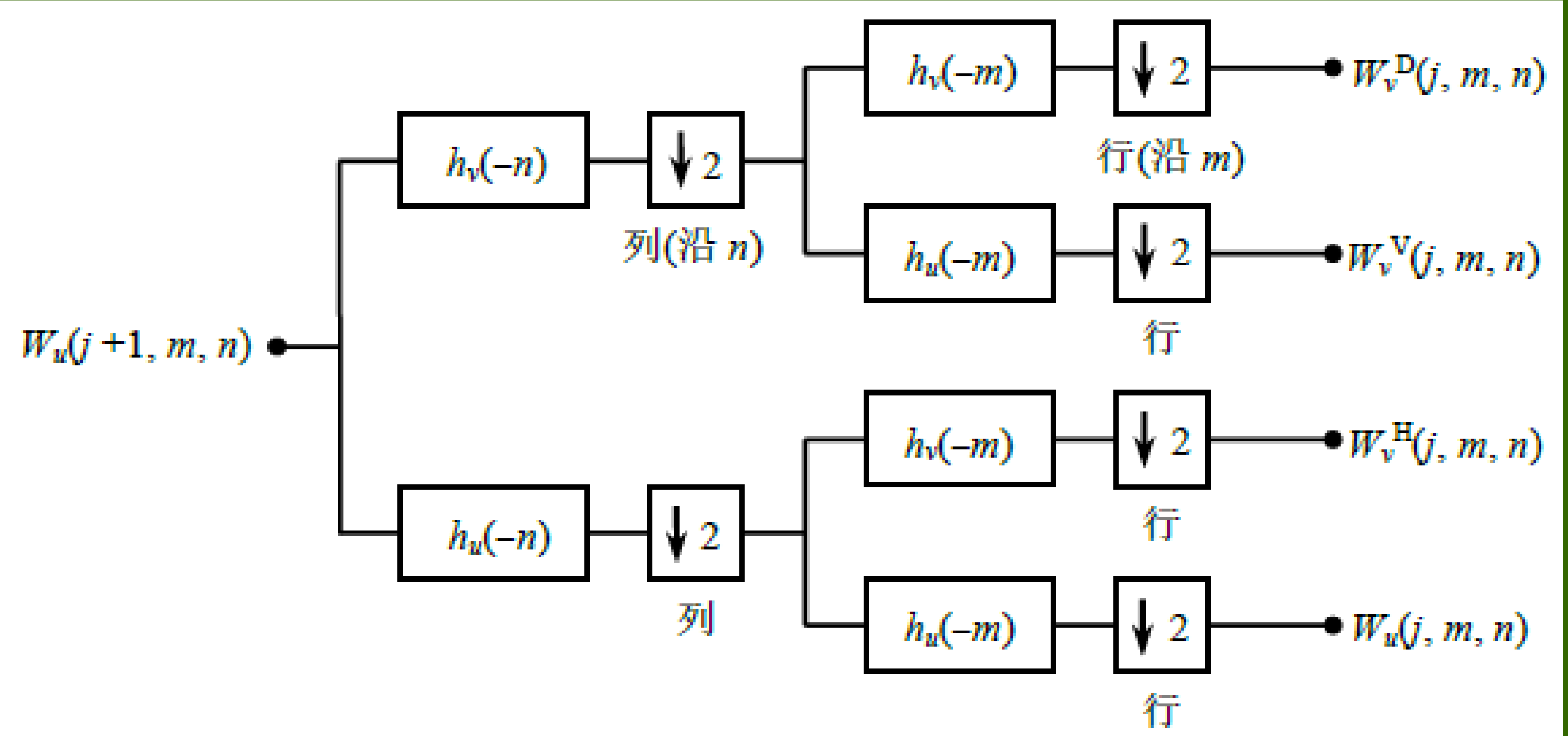



图 10.4.11 2-D 小波变换的方框图


# 10.5　小波变换编码

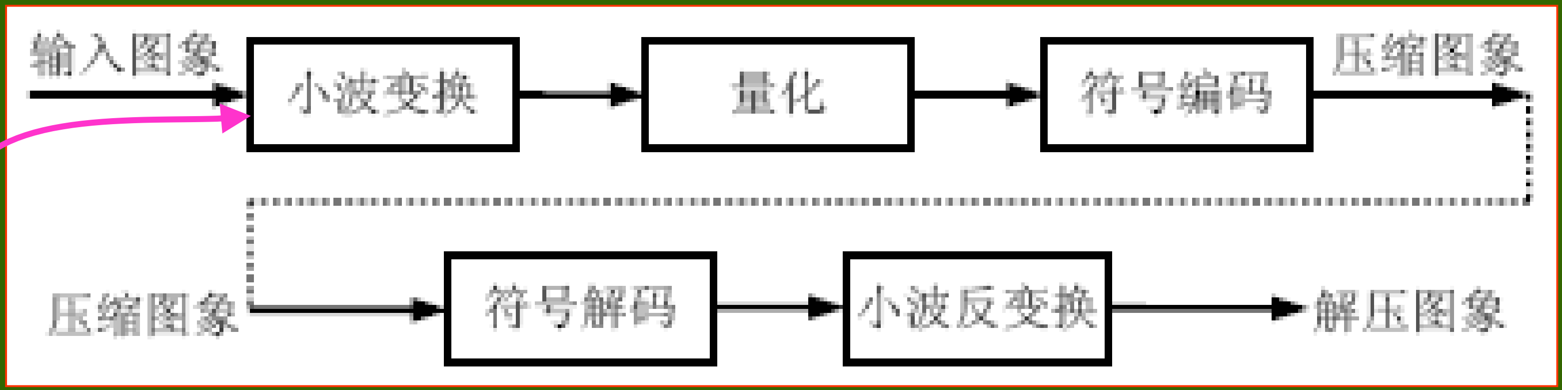
在图象国际标准JPEG-2000以及MPEG-4和H.264、H.265等里面都得到了应用

## 10.5.1　小波变换编解码系统

## 10.5.2　基于提升小波的编码

# 10.5.1　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


# 10.5.1　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

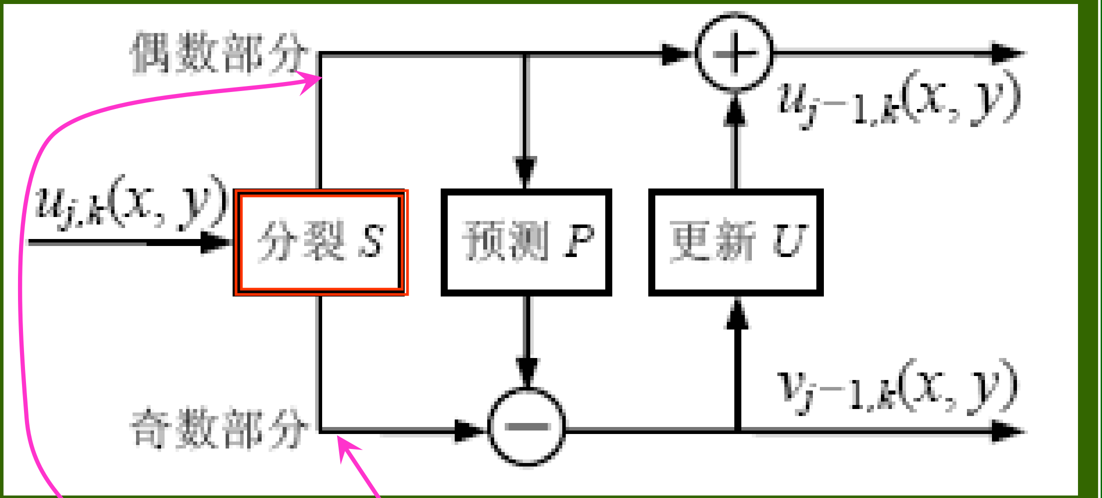
   需在不同尺度间调整量化间隔 {例：P.239}

# 10.5.2　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



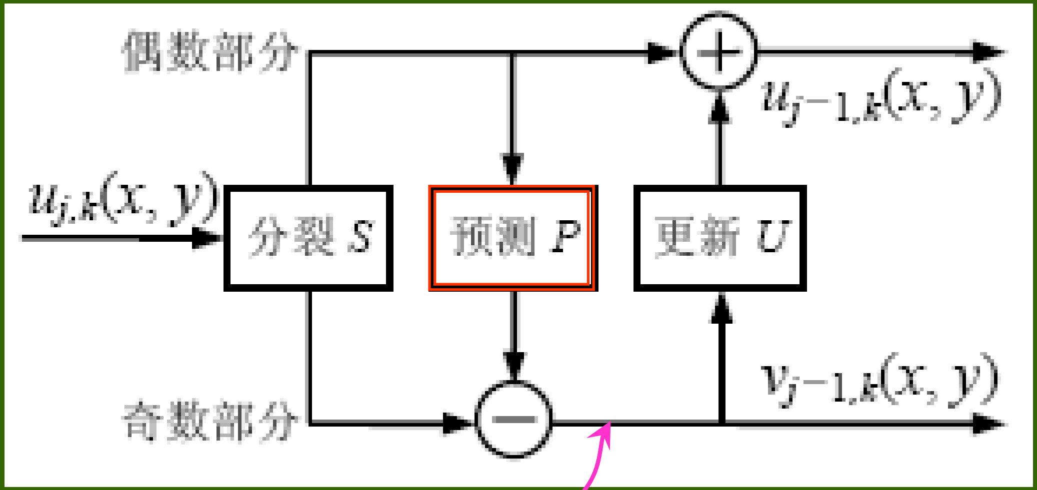

$$S\,[u_{j,k}(x, y)] := [u_{j-1,k}(x, y),\ v_{j-1,k}(x, y)] \qquad (10.5.2)$$

$$= [u_{j,2k}(x, y),\ u_{j,2k+1}(x, y)] \qquad (10.5.3)$$


# 10.5.2　基于提升小波的编码

## 2.　预测（predict）

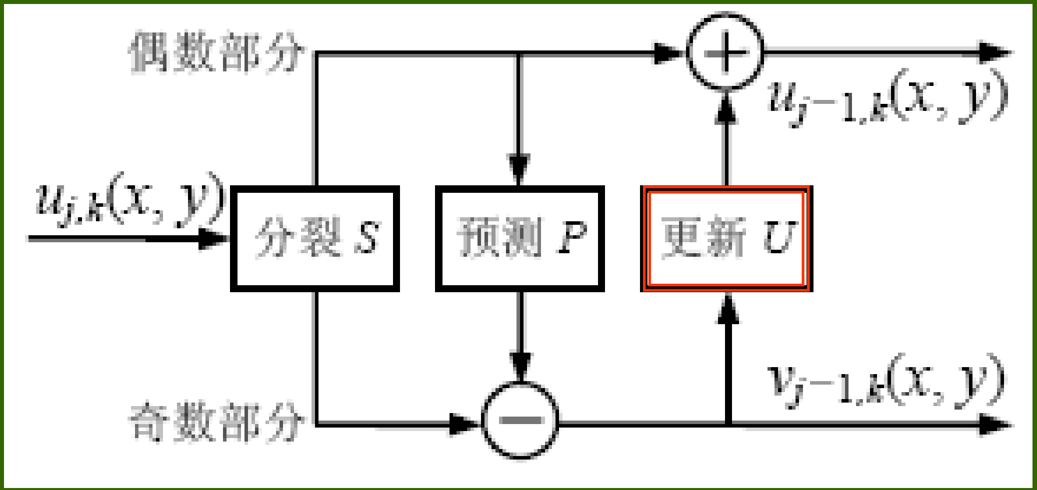
保持偶数部分不变并用偶数部分来预测奇数部分，然后用奇数部分与预测值的差（称为细节系数）替代奇数部分




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

### 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




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

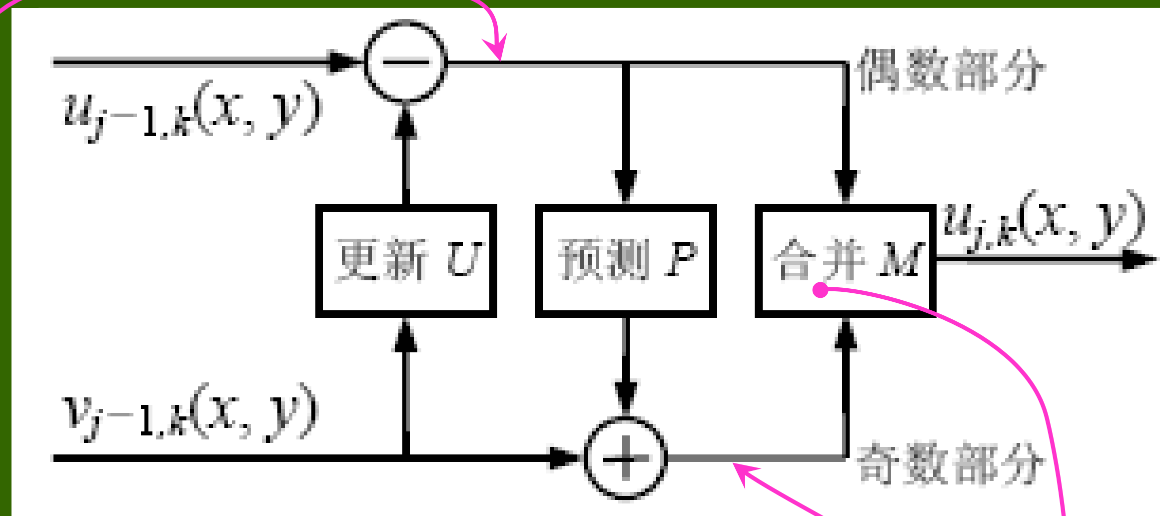
$$Q[u_{j-1,k}(x, y)] = Q[u_{j,k}(x, y)]$$


# 10.5.2 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/