论文摘要

小波变换是近十来年发展起来的一种新的信号分析与处理的域变换技术。它把信号在不同尺度上进行小波展开，因而更适合于处理突变和非平稳信号。目前国际上已经形成了基于该技术的静态图像压缩标准——JPEG2000。本文首先简单介绍了这一标准的内容，然后详细介绍了小波变换数学理论，主要内容有正交小波的多尺度分析，小波变换的快速 Mallat 算法，基于快速提升算法的第二代小波等，接着又介绍了基于小波变换的二维图像编解码技术——集分割算法（SPIHT），它是来源于零树编码的一种更为有效的标志位编码方法。之后论文介绍了整个算法的仿真验证方法，小波变换部分主要用 MATLAB 仿真，SPIHT 编解码部分主要用 VC++仿真，并对编解码过程中获得的原始数据进行了分析，提出了改进的方法。最后介绍了编解码系统的 DSP 实现。

关键词：小波变换　JPEG2000　SPIHT　DSP

Abstract

In recent decades, there has been developed a new kind of domain transform technology in signal analysis and processing—wavelet transform. It uses wavelets to spread the signal in various scales, so it is fit for singularity and unsmoothed signal processing. At the present time, based on this technology, the world has developed a static image-encoding standard--JPEG2000. This page first introduces the standard in brief, and then introduces the wavelet transform theory in detail. The main content has: orthogonal wavelet's multi-resolution analysis, wavelet transform's fast Mallat algorithm, the second generation wavelet transform based on the lifting steps and so on. And then based on this transform, the paper introduces the two-dimension image encoding and decoding technology--SPIHT(Set Partitioning In Hierarchical Trees), which is derived from the EZW(Embedded Zerotrees of Wavelet coefficients) technology while it is even more effective symbol bit encoding technology. Following this it introduces the simulation of the whole algorithm. Wavelet transform is mainly simulated by MATLAB. And SPIHT is simulated by the VC++. The paper analyzes the initial data and brings forward some improved methods. Last, it introduces the system realization in DSP.

Keywords: wavelet transform JPEG2000 SPIHT DSP

第一章 绪论

1.1 引言

视觉是人类感知外部世界最重要的手段，据统计，在人类获取的信息中，视觉信息占 60%，图像正是人类获取信息的主要途径，因此和视觉紧密相关的图像处理技术的潜在应用范围自然十分广阔。而伴随着个人电脑和 Internet 的广泛普及，数字图像处理技术迅速发展了起来，随着人们对图像质量的要求越来越高，图像的信息量越来越大，而网络带宽和计算机处理速度是有限的，因此对数字图像处理技术的要求也就越来越高了，不但要求算法简洁，快速而且要求对图像的压缩率要高。

1.2 课题的提出和意义

近 20 年来，随着多媒体技术的发展，出现了各种各样的静止图像压缩技术，如：JPEG、TIFF、PNG、HDF 和 PCX 等，其中最成功的当推 JPEG 标准。JPEG 标准是由 ISOIEC 联合技术委员会下属工作组：联合图像专家组(JPEG，Joint Photographic Experts Group)制定的。由于 JPEG 优良的品质，目前网站上 80%的图像都是采用 JPEG 标准。然而，随着多媒体应用领域的快速增长，传统 JPEG 压缩技术已无法满足人们对数字化多媒体图像资料的要求，如：网上 JPEG 图像只能一行一行地下载，直到全部下载完毕，才可以看到整个图像，如果只对图像的局部感兴趣也只能将整个图片下载下来再处理；JPEG 格式的图像文件体积仍然嫌大；JPEG 格式属于有损压缩，当被压缩的图像上有大片近似颜色时，会出现马赛克现象；同样由于有损压缩的原因，许多对图像质量要求较高的应用传统 JPEG 无法胜任。为了弥补传统标准的不足，适应 21 世纪图像压缩的需要，早在 1997 年，ISO/ITU-T 组织下的 IECJTC1/SC29/WG1 小组便开始着手制定新的静止图像压缩标准——JPEG2000。与传统 JPEG 基于离散余弦不同，JPEG2000 基于离散小波变换，它不仅在压缩性能方面明显优于 JPEG，还具有很多 JPEG 无法提供或无法有效提供的新功能，比如，同时支持有损和无损压缩、大幅图像的压缩、渐进传输、感兴趣区编码、良好的鲁棒性、码流随机访问等。JPEG2000 的所有这些特点，使得它的应用领域非常广泛，尤其在 Internet 传输、无线通信、医疗图像等领域将具有诱人的应用前景。

本论文对与 JPEG2000 相关的核心技术——小波变换和基于小波变换的编解码技术作了深入研究，然后用 MATLAB 对小波变换进行了仿真，用 VC++对基于小波变换的 SPIHT 编解码算法进行了仿真，并对编解码过程中所获得的数据进行了分析，提出了一些改进方法，最后介绍了系统的 DSP 实现。

第二章　JPEG2000 图像压缩标准

JPEG2000 静态图像压缩标准分为下列 7 个部分：

（1） JPEG2000 图像编解码系统；

（2） 扩充（给(1)的核心定义添加更多的特征和完善度）；

（3） 运动 JPEG2000；

（4） 一致性；

（5） 参考软件（包含 Java 和 C 实现）；

（6） 复合图像文件格式（用于文件扫描和传真应用程序）；

（7） 对（1）的最小支持（技术报告）。

其中(1)是完全被认可的 ISO 标准，定义了核心压缩技术和最小文件格式，(2)—(6)定义压缩和文件格式的扩充。下面对第(1)部分做进一步的说明。

一个典型的 JPEG2000 PART1 的压缩过程如图 2.1 所示。

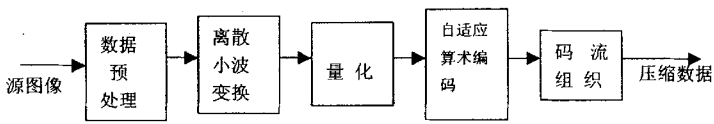


图 2.1　JPEG2000 压缩过程

2.1　数据预处理

PART1 的预处理一般包括三种操作：区域划分，降低量级，分量变换。

2.1.1　区域划分

区域划分是指将图像划分为大小相等的若干区域，对每一区域独立进行压缩处理。区域划分的目的在于要降低压缩过程所需的内存资源，如果内存足够，这一步可以忽略。PART1 要求划分的区域是互不重叠的，因为这种划分是最简单的，但由此而产生的一个缺点就是边缘象素环境信息的缺乏，这个问题可以采用边界镜像扩展技术来解决。

2.1.2　降低量级

降低量级是将采样精度为 P 的无符号整数减去 2^{P-1}，使原来范围为 $[0,2^P-1]$ 的样本移位到 $[-2^{P-1},2^{P-1}-1]$ 这个关于 0 对称的范围内。这一步在简化对数值溢

出等问题处理的同时，不会影响编码的效率。

2.1.3 分量变换

分量变换指对具有多个分量的图像先通过某种变换降低这几个分量之间的相关性，提高压缩效率。目前 PART1 中主要是对 RGB 分量采用 ICT 或 RCT，将色彩信息转换为频道信息。

ICT(IrreversibleColorTransform)定义如下

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.16875 & -0.33126 & 0.500 \\ 0.500 & -0.41869 & -0.08131 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

RCT(ReversibleColorTransform)定义如下：

$$Y = \left[\frac{-R+2G+B}{4}\right], \quad U = R-G, \quad V = B-G$$

解压缩时需进行相应的逆变换以恢复原分量的值。

2.2 离散小波变换(DWT)

DWT 是 JPEG2000 的核心之一，关于这部分理论会在第三章作详细论述，这里只作概要说明。

预处理后的数据将进行离散小波变换，以进一步降低数据之间的相关性。与 JPEG 采用的离散余弦变换(DCT)相比，DWT 具有很好的局部性，能够针对不同类型特点的图像中的不同区域采用不同的空-频分辨率，从而有可能取得更好的压缩比，而且它还可以提供实现无损压缩的机制。简单来讲，一维 DWT 即是对源信号进行了一系列的高通和低通滤波，并在每次滤波后将数据采样频率降为原来的一半，以保证每次小波变换后得到的系数与源信号数目相同。每次的低通滤波输出保存了源信息的低频信息，它是一个以更低分辨率对源信号的再现，集中了源信号中的大部分能量；而高通滤波输出保存的则是源信号的高频信息，如边界、材质等，其中所含能量很少。一次低通滤波后的信号往往还存在着大量的相关性，为提高压缩性能，仍需要对它再次滤波，直至信号之间相关性达到可以忽略的程度为止。高通滤波后的信号由于能量很小，再对它进行滤波往往是不划算的，因而一般不再对它滤波。这种滤波方式被称为 dyadic 分解，它是 JPEG2000PART1 唯一支持的分解方式。二维 DWT 是对一维 DWT 的简单扩充，通过将行信号和列信号与高通滤波器 $g(n)$ 和低通滤波器 $h(n)$ 进行不同的组合，源图像被划分为 4 个子带。其中唯一的一个低频子带仍可以继续分解。

2.3 量化

JPEG2000 的量化与 JPEG 量化基本相同：总体上都是采用均匀量化；不同子

带的量化步长一般不同。量化器步长是因子带而异的，目前有两种考虑因素：一种是人类视觉对子带信号的敏感性(HVS 的属性)；另一种是依据不同子带的均方误差对重建后图像的总均方误差的贡献大小来决定量化步长。量化器步长的传输也有两种方式：一种是类似于 JPEG 中的 q-table，显式地传给解码器；另一种仅传输某子带的量化器步长 Δb，其它子带步长由 Δb 计算出。

2.4 自适应算术编码(第一层编码)和码流组织(第二层编码)

优化截取的嵌入式块编码(EBCOT)算法，它是基于小波变换的嵌入式编码的方法之一，是 JPEG2000 的另一个核心技术。其主要思想是基于 Shapiro 提出的 EZW（嵌入式零树编码），本文将在第三章着重论述 EZW 的改进算法 SPIHT 算法。

所谓“基于小波变换的嵌入式编码”是指编码器将等待编码的、经过小波变换后的比特流按重要性不同进行排序，提供多个满足不同目标码率或失真度的截断点，使得解码器方能根据目标码率或失真度的要求在某一截断点结束解码，提供相应质量的图像。在进行块编码时，JPEG2000 强调多截断点的支持，越多的截断点，表明图像可提供更多的质量选择。对码流的组织，EBCOT 也有专门的论述。码流的组织(第二层编码)就是将上述截断的数据进行打包，并附加相关的标志信息，从而实现 JPEG2000 对多失真度的支持。

2.5 JPEG2000 的主要技术优势

JPEG2000 相对以往的 JPEG 标准有一个很大的飞跃。以前彩色静态图像编码采用 JPEC；二值静态图像采用 JBIC；低压缩率编码采用 JPEC-LS。多种方式同时存在，而 JPEG2000 则将上述方式统一起来，成为各种图像的通用编码方式。不仅如此，JPEG2000 还有许多原先的标准所不可比拟的优点。

1) 高压缩率

JPEG2000 格式的图片压缩比可在现在的 JPEG 基础上再提高 10%～30%，而且压缩后的图像显得更加细腻平滑。

2) 渐进传输

JPEG2000 格式的图像支持渐进传输(Progressive Transmission)，就是先传输图像轮廓数据，然后再逐步传输其他数据来不断提高图像质量。这样你就不需要像以前那样等图像全部下载后才决定是否需要，而是先直接快速浏览图像概貌，之后再决定是否需要更细致的图像。

3) 感兴趣区域压缩

JPEG2000 另一个极其重要的优点就是 ROI(Region of Interest，即感兴趣区域)。我们可以对一幅图像中我们感兴趣的部分采用低压缩比以获取较好的图像效果，而对其他部分采用高压缩比以节省存储空间。

4) 同时支持有损压缩和无损压缩

JPEG 标准无法在同一个压缩模式中同时提供有损和无损压缩，而在 JPEG2000 标准中，能在同一个算法中对图像进行有损和无损压缩，并且能实现非常大图像压缩。

5) 其它技术优势

① 考虑了人的视觉特性：

增加了视觉权重和掩膜，在不损害视觉效果的情况下大大提高了压缩效率；

② 码流的随机访问和处理：

这一特征允许用户在图像中随机地定义感兴趣区域，使得这一区域的图像质量高于其它图像区域；码流的随机处理允许用户进行旋转、移动、滤波和特征提取等操作。

③ 容错性：

在码流中提供容错性有时是必要的，例如在无线等传输误码很高的通信信道中传输图像时，没有容错性是让人不能接受的。

④ 开放的框架结构：有利于在不同的图像类型和应用领域优化编码系统。

⑤ 基于内容的描述：基于内容的描述在 JPEG2000 中是压缩系统的特性之一。

第三章　小波变换数学理论

3.1　小波变换概论

传统傅立叶分析只能对信号进行时域或频域单独进行分析，时域上有限的信号在频域是无穷的，频域内有限的信号在时域里是无穷的，其时频窗如图 3.1 所示：

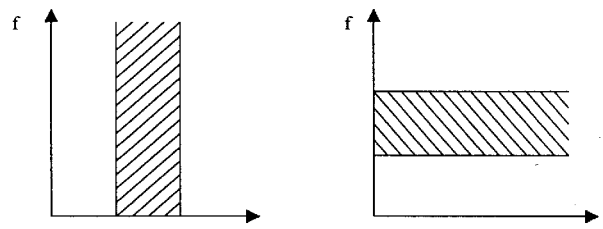


图 3.1 傅氏分析

为了解决傅氏分析的上述局限性，提出了短时傅立叶分析，其时频窗如图 3.2 所示，即在时域和频域内同时进行分析，但只能作均匀分析，对信号中的畸变部分（时域或频域内的），则由于分辨率不够而无法做出精确分析。

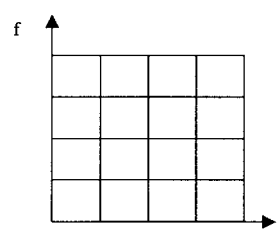


图 3.2 短时傅立叶分析

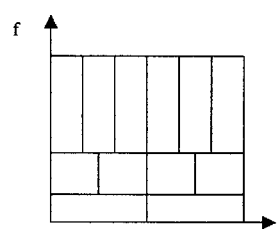


图 3.3 小波分析

小波分析解决了以上的所有问题，它不仅能在时域和频域内同时分析，而且还能自动调整分辨率，其时频窗如图 3.3 所示，采用不同尺度的小波，即频率或高或低的小波对信号的不同时刻作相关，这样，在时域内信号的奇变部分可采用短时高频小波进行分析，对应时频图的上部，而对时域内的缓变部分则采用长时低频小波进行分析，对应时频图的下部，可根据不同信号调整分辨率，因而小波分析被誉为数学上的“显微镜”。

3.2 连续小波变换

3.2.1 定义

连续小波变换的数学表达式：

$$W_f(a,b)=\frac{1}{\sqrt{a}}\int_R f(t)\overline{\psi}(\frac{t-b}{a})dt,\quad a>0 \tag{3.1}$$

$f(t),\psi(t)$是平方可积的，且$\psi(t)$的傅氏变换$\Psi(\omega)$满足条件：

$$C_\psi=\int_R|\Psi(\omega)|^2/\omega\,d\omega<\infty \tag{3.2}$$

被称为小波函数或小波母函数，a为尺度因子，b为平移因子。

从上述定义可以看出，小波变换也是一种积分变换，是将一个时间函数变换到时间－尺度相平面上，使得能够提取函数的某些特征。而上述两参数a,b是连续变化的，故称上述变换为连续小波变换。

如果令$\psi_{a,b}(t)=\frac{1}{\sqrt{a}}\psi(\frac{t-b}{a})$，则连续小波变换也可采用内积来表示：

$$W_f(a,b)=<f(t),\psi_{a,b}(t)> \tag{3.3}$$

从这种写法可以粗略地解释小波函数的含义：由于数学上内积表示两个函数“相似”的程度，所以由式（3.3）可看出，小波变换$W_{f(a,b)}$表示$f(t)$与$\psi_{a,b}(t)$的“相似”程度。当a增大时（$a>1$），表示用伸展了的$\psi(t)$波形去观察整个$f(t)$；反之，当a减小时（$0<a<1$），则以压缩了的$\psi(t)$波形去衡量$f(t)$的局部，见图3.4。所以，随着尺度因子的从大变到小（$0<a<+\infty$），$f(t)$的小波变换可以反映$f(t)$从概貌到细节的全部信息。从这个意义上说，小波变换是一架“变焦镜头”，它既是“望远镜”，又是“显微镜”，而a就是“变焦旋钮”。

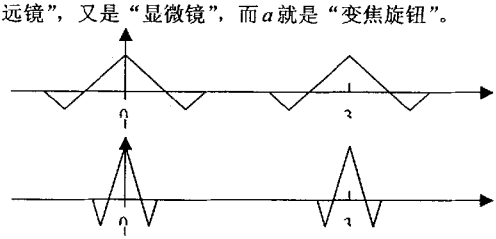


图3.4　尺度因子a和移位因子b的作用

3.2.2 允许小波

与傅氏变换类似，也存在小波反变换。设 $f(t)\in L^2(R)$,则

$$f(t)=\frac{1}{C_\psi}\int_0^{+\infty}\int_{-\infty}^{+\infty}\frac{1}{a^2}W_f(a,b)\cdot\psi_{a,b}(t)dbda \tag{3.4}$$

为使小波反变换有意义，小波函数需要满足 $0<C_\psi<+\infty$，即

$$0<\int_R\frac{|\Psi(\omega)|^2}{\omega}d\omega<\infty \tag{3.5}$$

该式被称为小波的容许条件，由此可推出

$$\Psi(0)=\int_{-\infty}^{+\infty}\psi(t)dt=0 \tag{3.6}$$

由（3.5）式可知 $\psi(t)$ 应具有快速衰减性，由式（3.6）可知 $\psi(t)$ 应具有波动性。可以想象 $\psi(t)$ 的图像是快速衰减的振动曲线，这就是 $\psi(t)$ 称为小波的原因。

3.2.3 窗口宽度与海森堡(Heissenberg)测不准原理

与窗口傅立叶变换类似，在小波变换中，可称 $\psi_{a,b}(t)$ 是窗函数，小波变换的时—频窗表现了小波变换的时—频局部化能力，因此可定义小波变换的窗口中心与窗口宽度。

定义：设 $\psi(t)\in L^2(R)$ 是小波函数，其对应的时窗中心 t_ψ^*，时窗半径 $\mathrm{V}t_\psi$，频窗中心 ω_ψ^*，频窗半径 $\mathrm{V}\omega_\psi$ 分别为：

$$t_\psi^*=\frac{\int_R t|\psi(t)|^2\,dt}{\|\psi_{a,b}(t)\|^2} \tag{3.7}$$

$$\mathrm{V}t_\psi=\frac{[\int_R(t-t_\psi^*)^2|\psi(t)|^2\,dt]^{\frac{1}{2}}}{\|\psi(t)\|} \tag{3.8}$$

$$\omega_\psi^*=\frac{\int_R\omega|\Psi(\omega)|^2\,d\omega}{\|\Psi(\omega)\|^2} \tag{3.9}$$

$$\mathrm{V}\omega_\psi=\frac{[\int_R(\omega-\omega_\psi^*)^2|\Psi(\omega)|^2\,d\omega]^{\frac{1}{2}}}{\|\Psi(\omega)\|} \tag{3.10}$$

由于小波变换中的窗函数$\psi_{a,b}(t)$是小波函数$\psi(t)$平移和缩放的结果，所以，记$\psi_{a,b}(t)$对应的有关分量分别为t^*，$\mathrm{V}t$，ω^*，$\mathrm{V}\omega$。可推出他们之间的关系式：

$$t^* = at_\psi^* + b \tag{3.11}$$

$$\mathrm{V}t = a\mathrm{V}t_\psi \tag{3.12}$$

$$\omega^* = \frac{1}{a}\omega_\psi^* \tag{3.13}$$

$$\mathrm{V}\omega = \frac{1}{a}\mathrm{V}\omega_\psi \tag{3.14}$$

由(3.11)式可看出，$\psi_{a,b}(t)$的时窗中心是$\psi(t)$的时窗中心的a倍后再平移b个单位；由式(3.13)可看出$\psi_{a,b}(t)$的频窗中心是$\psi(t)$的频窗中心的$\frac{1}{a}$倍；由式(3.12)可看出$\psi_{a,b}(t)$的时窗宽度是$\psi(t)$的时窗宽度的a倍；由式(3.14)可看出$\psi_{a,b}(t)$的频窗宽度是$\psi(t)$的频窗宽度的$\frac{1}{a}$倍。这样对于固定的b，随着a的增大（a>1），小波变换的时窗就增宽，而频窗就变窄。

虽然$\psi_{a,b}(t)$的时窗、频窗的中心、宽度随着a,b在变化，但在时－频相平面上，时窗和频窗所形成的区域（即窗口）的面积为

$$2\mathrm{V}t \cdot 2\mathrm{V}\omega = 4a\mathrm{V}t_\psi \cdot \frac{1}{a}\mathrm{V}\omega_\psi = 4\mathrm{V}t_\psi \cdot \mathrm{V}\omega_\psi \tag{3.15}$$

它是不随a,b的变化而变化的。

综上所述，易知在时－频相平面上，小波变换$W_f(a,b)$的时频窗是面积相等但长宽不同的矩形区域，这些窗口的长、宽是相互制约的，它们都受参数a的控制，而尺度参数a是与小波变换$W_f(a,b)$所反映原信号$f(t)$的频率成份相关的量：当a较小时，$W_f(a,b)$反映的是$t=b$时附近的高频成份的特性；当a较大时，$W_f(a,b)$反映的是$t=b$时附近的低频成份的特性，所以我们由如下的对应：

$$a：小->大 \Leftrightarrow 频率\omega：大->小 \tag{3.16}$$

因此，时频相平面上的窗口分布如图 3 所示。“扁平”状的时频窗是符合信号低频成份的局部时频特性的，而“瘦窄”状的时频窗是符合信号高频成份的局部时频特性的。

3.3 离散小波变换

3.3.1 连续小波变换的冗余

由式(3.4)所示的小波变换的反演公式可知，$f(t)$可由它的小波变换$W_f(a,b)$精确地重建，它也可看成将$f(t)$按“基”$\psi_{a,b}(t)$的分解，系数就是$f(t)$的小波变换，但是“基”$\psi_{a,b}(t)$的参数a,b是连续变化的，所以$\psi_{a,b}(t)$之间不是线性无关的，也就是说，它们之中有“富余”的，这就导致$W_f(a,b)$之间有相关性，数学表述如下：

对于$a=a_1,b=b_1$，有

$$\begin{aligned} W_f(a_1,b_1) &= \int_R f(t)\overline{\psi}_{a_1,b_1}(t)dt \\ &= \int_R \frac{1}{C_\psi}\left(\int_0^{+\infty}\int_{-\infty}^{+\infty}\frac{1}{a^2}W_f(a,b)\cdot\psi_{a,b}(t)dbda\right)\overline{\psi}_{a_1,b_1}(t)dt \\ &= \frac{1}{C_\psi}\int_0^{+\infty}\int_R\frac{1}{a^2}W_f(a,b)\left(\int_R\psi_{a,b}(t)\cdot\overline{\psi}_{a_1,b_1}(t)dt\right)dbda \\ &= \int_0^{+\infty}\int_R\frac{1}{a^2}W_f(a,b)k_\psi(a,a_1,b,b_1)dbda, \end{aligned} \tag{3.17}$$

其中

$$k_\psi(a,a_1,b,b_1)=\frac{1}{C_\psi}\int_R\psi_{a,b}(t)\cdot\overline{\psi}_{a_1,b_1}(t)dt. \tag{3.18}$$

上式（3.17）说明(a_1,b_1)处的小波变换$W_f(a_1,b_1)$可以由半平面$0<a<+\infty$，$-\infty<b<+\infty$上的点(a,b)处的小波变换$W_f(a,b)$表出，系数是$k_\psi(a,a_1,b,b_1)$，它称为小波变换的再生核。若$\psi_{a,b}(t)$与$\psi_{a_1,b_1}(t)$正交时，由式(3.18)知$k_\psi(a,a_1,b,b_1)=0$。此时(a,b)处的小波变换$W_f(a,b)$对$W_f(a_1,b_1)$的“贡献”为零，而那些使$\psi_{a,b}(t)$与$\psi_{a_1,b_1}(t)$不正交的(a,b)处的小波变换就要对(a_1,b_1)处的小波变换做“贡献”。所以，要使各点小波变换之间没有相关，需要在函数族$\{\psi_{a,b}(t)\}$中寻找相互正交的基函数。解决的办法是将$\psi_{a,b}(t)$的参数a,b离散化，以图能够找到相互正交的基函数。

这样，需要引入离散小波变换的概念。

3.3.2 参数的离散化与离散小波变换的概念

为达到上述目的，我们将对小波函数$\psi_{a,b}(t)$中的a,b进行离散化。

先看尺度参数a的离散化。通行的做法是取$a=a_0^j, j=0,\pm1,\pm2\mathrm{L}$，此时相应的小波函数是$a_0^{-\frac{j}{2}}\psi(a_0^{-j}(t-b)), j=0,\pm1,\pm2\mathrm{L}$.

再将位移参数b离散化为$k=0,\pm1,\pm2\mathrm{L}$，得离散小波函数：

$$\psi_{j,k}(t)=a_0^{-\frac{j}{2}}\psi(a_0^{-j}(t-b)), j,k\in Z, a_0>0\text{，是常数}$$

$f(t)$得离散小波变换为：

$$W_f(j,k)=\int_R f(t)\overline{\psi}_{j,k}(t)dt \tag{3.19}$$

离散小波变换是尺度－位移相平面的规则分布的离散点上的函数，与连续小波变换相比，少了许多点上值，但是有如下两个结论：

（1） 离散小波变换$W_f(j,k)$包含了函数$f(t)$的全部信息，或者说，由$W_f(j,k)$可重构原函数$f(t)$；

（2） 任意函数$f(t)$都可以以$\psi_{j,k}(t)$为“基”表示出来。

当取$a_0=2$时，就成为离散二进小波变换：

$$\psi_{j,k}(t)=\frac{1}{2^{j/2}}\psi(\frac{t}{2^j}-k). \quad j,k\in Z \tag{3.20}$$

函数$f(t)\in L^2(R)$对应的二进小波变换便可被写成

$$W_f(j,k)=<f(t),\psi_{j,k}(t)>=\frac{1}{2^{j/2}}\int_{-\infty}^{+\infty}f(t)\overline{\psi}(\frac{t}{2^j}-k)dt \tag{3.21}$$

3.4 正交小波变换

一般来说离散小波变换的信息量仍然是有冗余的。从数值计算及数据压缩的角度，我们仍希望减少它们的冗余度；另外，离散小波框架一般不是$L^2(R)$的正交基。本节要做的事情就是寻找信息量没有冗余的小波和$L^2(R)$的正交基，如何构造它们。

3.4.1 正交小波变换的定义

若式（3.20）中$\psi(t)\in L^2(R)$是一个可容许小波，若其二进伸缩平移系$\psi_{j,k}(t)$，$j,k\in Z$构成$L^2(R)$的标准正交基，则称$\psi(t)$为正交小波，也称$\psi_{j,k}(t)$是正交小波函数，称相应的离散小波变换式（3.21）为正交小波变换。

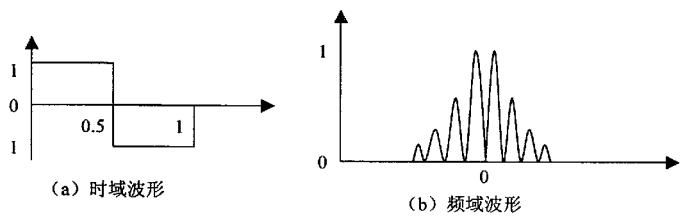
让我们来看一个正交小波的非常简单的例子，这就是数学家 A.Haar 在上世纪初提出的 Haar 系，它取小波母函数$h(t)$为

$$h(t)=\begin{cases}\dfrac{1}{2^{j/2}}, & 2^j k\le t<(2k+1)2^{j-1},\\ -\dfrac{1}{2^{j/2}}, & (2k+1)2^{j-1}\le t\le(k+1)2^j,\\ 0, & \text{其他.}\end{cases}\tag{3.22}$$

其频域表达式为：

$$\mathrm{H}(\omega)=\frac{1-2e^{-\frac{i\omega}{2}}+e^{-i\omega}}{i\omega}\tag{3.23}$$

它们的图形见图 3.5。



（a）时域波形　　（b）频域波形

图 3.5　Haar 小波

取它的二进伸缩平移系

$$h_{j,k}(t)=\frac{1}{2^{j/2}}h\left(\frac{t}{2^j}-k\right).\quad j,k\in Z\tag{3.24}$$

易知有

$$h(t)=\begin{cases}\dfrac{1}{2^{j/2}}, & 2^j k\le t<(2k+1)2^{j-1},\\ -\dfrac{1}{2^{j/2}}, & (2k+1)2^{j-1}\le t\le(k+1)2^j,\\ 0, & \text{其他.}\end{cases}\tag{3.25}$$

容易验证 $\{h_{m,n}(t)\}_{m,n\in Z}$ 构成 $L^2(R)$ 的一个标准正交基。

然而 Haar 系的小波函数是不连续的，且它在频域随 ω 的衰减速度仅为 $\frac{1}{\omega}$，因此，它不能满足对基的光滑性的要求，频域的局部性也较差，在实际应用中很少用它，但它的结构简单，很方便用来说明问题，常用于理论研究中。

3.4.2 构造正交小波的多尺度分析

3.4.2.1 多尺度分析（Muli-Resolution Analysis）

简称 MRA，也称多分辨率分析，是 S.Mallat 在 1988 年提出的。

定义：我们称满足下述条件的 $L^2(R)$ 中的一列子空间 $\{V_j\}_{j\in Z}$ 及一个函数 $\varphi(t)$ 为一个正交多尺度分析，记为 $(\{V_j\}_{j\in Z},\varphi(t))$：

（1）$V_j\subseteq V_{j-1}, j\in z,$ (3.26)

（2）$f(t)\in V_j \Leftrightarrow f(2t)\in V_{j-1},$ (3.27)

（3）$\mathop{\mathrm{I}}_{j\in z} V_j=\{0\},\ \mathop{\mathrm{U}}_{j\in z} V_j=L^2(R),$ (3.28)

（4）$\varphi(t)\in V_0$，且 $\{\varphi(t-k)\}_{k\in z}$ 是 V_0 的标准正交基，称 $\varphi(t)$ 是此多尺度分析的尺度函数或父函数。

由性质（2）、（4）可知，对于任何 $f(t)\in V_0$，有 $f(\frac{t}{2^j})\in V_j$，且容易验证，函数系 $\{2^{-\frac{j}{2}}\varphi(2^{-j}t-k)\}_{k\in z}$ 构成了 V_j 的一组标准正交基。

下面我们先讨论如何由 $\{V_j\}_{j\in z}$ 构造空间 $L^2(R)$ 的正交分解 $L^2(R)=\oplus_{j\in z} W_j$。

由于 $V_j\in V_{j-1}$，记 W_j 是 V_j 在 V_{j-1} 中的正交补空间，即 $V_{j-1}=V_j+W_j$，$W_j\perp V_j$，$j\in z$，这样得到空间列 $\{W_j\}_{j\in z}$，显然当 $m,n\in z, m\neq n$ 时，有 $W_m\perp W_n$。另外，

$$\begin{aligned}V_{j-1}&=V_j+W_j=V_{j+1}+W_{j+1}+W_j\\&=V_{j+2}+W_{j+2}+W_{j+1}+W_j=\mathrm{L}\\&=V_{j+s}+W_{j+s}+W_{j+s-1}+W_{j+s-2}\mathrm{L}+W_{j+1}+W_j,\end{aligned}$$

令 $s\to+\infty$，得到：$V_{j-1}=\oplus_{m=j}^{+\infty} W_m$，令 $j\to-\infty$，得到：

$$L^2(R)=\oplus_{m=-\infty}^{+\infty} W_m \tag{3.29}$$

我们也称W_j为尺度为j的小波空间，V_j为尺度为j的尺度空间。由于W_j是V_j在V_{j-1}中的正交补，因此，W_j也称V_j在V_{j-1}的细节空间。

有上述过程可知，由$\{V_j\}_{j\in z}$可确定$\{W_j\}_{j\in z}$。

下面，我们再来看如何由$\varphi(t)$求出W_j的标准正交基。

我们注意到W_j中函数也有对伸缩变换的特点：若非零函数$f(t)\in W_j \Leftrightarrow f(2t)\in W_{j-1}$。

正是因为W_j对伸缩变换的上述特点，我们寻找W_j的标准正交基的问题归结为找W_0的标准正交基即可。我们希望W_0的标准正交基是由一个函数$\psi(t)$的平移系构成。在通过$\varphi(t)$寻找$\psi(t)$之前，我们先来研究它们应满足的性质，以及它们之间的关系。

3.4.2.2 尺度函数和小波函数的性质

并不是任何函数的平移系都是标准正交系，下面定理说明这样的特性函数的条件。

定理 1：设$\varphi(t)\in L^2(R)$，令$\varphi_{0,k}(t)=\varphi(t-k)$，则$\{\varphi_{0,k}(t)\}_{k\in z}$是标准正交系

$$\Leftrightarrow \sum_{k\in z}|\Phi(\omega+2k\pi)|^2\equiv 1 \tag{3.30}$$

其中$\Phi(\omega)$是$\varphi(t)$的傅立叶变换。

定理 2：设$\{V_k\}_{k\in z}$和尺度函数$\varphi(t)$构成$L^2(R)$的一个多尺度分析，则有$\{h_k\}_{k\in z}$，使

$$\frac{1}{\sqrt{2}}\varphi(\frac{t}{2})=\sum_{k\in z}h_k\varphi(t-k) \tag{3.31}$$

其中，显然有

$$h_k=\frac{1}{\sqrt{2}}\int_R\varphi(\frac{t}{2})\overline{\varphi}(t-k)dt. \tag{3.32}$$

在式（3.31）两端取傅立叶变换，得到

$$\sqrt{2}\Phi(2\omega)=\sum_{k\in z}h_k e^{-k\omega i}\Phi(\omega), \quad 即 \quad \Phi(2\omega)=\frac{1}{\sqrt{2}}\sum_{k\in z}h_k e^{-k\omega i}\Phi(\omega),$$

记 $$H(\omega)=\frac{1}{\sqrt{2}}\sum_{k\in z}h_k e^{-k\omega i} \tag{3.33}$$

即得 $$\Phi(2\omega)=H(\omega)\Phi(\omega) \tag{3.34}$$

式（3.33）是$\{h_k\}_{k\in z}$得傅立叶变换或频域形式，也称为共轭滤波器，称h_k为滤波器系数，而式（3.31）称为双尺度方程，式（3.34）是双尺度方程得频域形式。

定理 3：设$\{h_k\}$是一个以$\varphi(t)$为尺度函数得多尺度分析的滤波器系数，$H(\omega)$是它的频域形式，则

（1）$|H(\omega)|^2+|H(\omega+\pi)|^2=1$，

（2）若$\{h_k\}$满足$\sum_k|h_k|<+\infty$且$\Phi(\omega)$连续，$\Phi(0)\neq 0$，则$H(0)=1$。

将$H(0)=1$代入（3.33）可得：

$$\sum_{k\in z}h_k=\sqrt{2}<+\infty \tag{3.35}$$

定理 4：设$\{V_k\}_{k\in z}$和尺度函数$\varphi(t)$构成$L^2(R)$的一个多尺度分析，$\{W_k\}_{k\in z}$是由它所确定的小波空间，若$\psi(t)\in W_0$，则有$\{g_k\}_{k\in z}$，使

$$\frac{1}{\sqrt{2}}\psi(\frac{t}{2})=\sum_{k\in z}g_k\varphi(t-k), \tag{3.36}$$

其中 $$g_k=\frac{1}{\sqrt{2}}\int_R\psi(\frac{t}{2})\cdot\overline{\varphi}(t-k)dt, \tag{3.37}$$

与式（3.34）类似，式（3.36）的频域形式为：

$$\Psi(2\omega)=G(\omega)\Phi(\omega) \tag{3.38}$$

其中 $$G(\omega)=\frac{1}{\sqrt{2}}\sum_{k\in z}g_k e^{-k\omega i} \tag{3.39}$$

式（3.36）也称为双尺度方程。式（3.31）和式（3.36）这两个双尺度方程是多尺度分析赋予尺度函数$\varphi(t)$和小波函数$\psi(t)$的最基本性质。

对于给定的多尺度分析$(\{V_k\}_{k\in z},\varphi(t))$，如何找到$W_0$中的一个函数$\psi(t)$，有下定理。

定理 5：设给定的多尺度分析的尺度函数$\varphi(t)$，则

（1）$\psi(t)\in W_0$的充要条件是

$$H(\omega)\overline{G}(\omega)+H(\omega+\pi)\overline{G}(\omega+\pi)=0 \tag{3.40}$$

（2）
$$|G(\omega)|^2+|G(\omega+\pi)|^2=1 \tag{3.41}$$

（3）式（3.40）和（3.41）是函数系$\{\psi(t-k)\}_{k\in z}$构成W_0的标准正交基的充要条件。

定理5告诉我们满足式（3.40）和（3.41）的函数$\psi(t)$可构成W_0的标准正交基，下面的定理说明这样的$\psi(t)$可以构成$L^2(R)$的标准正交基。

定理6：设$\psi(t)\in L^2(R)$，且满足式（3.40）和（3.41），则$\psi(t)$的伸缩平移系

$$\psi_{j,k}(t)=2^{-j/2}\psi(2^{-j}t-k),\quad j,k\in z \tag{3.42}$$

构成$L^2(R)$的标准正交基。

下述定理说明如何由$\varphi(t)$构造$\psi(t)$标准正交基。

定理7：对于已给的多尺度分析$(\{V_k\}_{k\in z},\varphi(t))$，取$G(\omega)=e^{-i\omega}\overline{H}(\omega+\pi)$，则由式（3.38）所确定的函数$\varphi(t)$的伸缩平移系$\{2^{-j/2}\psi(2^{-j}t-k)\}_{j,k\in z}$构成$L^2(R)$的标准正交基，其中$H(\omega)$由式（3.33）所确定。

3.4.2.3 由多尺度分析构造正交小波基

至此，我们得到一个从多尺度分析$(\{V_k\}_{k\in z},\varphi(t))$，求得$L^2(R)$的小波函数$\psi(t)$，进而

求得$L^2(R)$的小波基$\{2^{-j/2}\psi(2^{-j}t-k)\}_{j,k\in z}$的步骤：

（1） 由$\varphi(t)$及式$\Phi(2\omega)=H(\omega)\Phi(\omega)$，确定$H(\omega)$；

（2） 由$H(\omega)$及式$G(\omega)=e^{-i\omega}\overline{H}(\omega+\pi)$，确定$G(\omega)$；

（3） 由$G(\omega)$，$\Phi(\omega)$及式$\Psi(2\omega)=G(\omega)\Phi(\omega)$，确定$\Psi(\omega)$；

（4） 由$\Psi(\omega)$和逆傅立叶变换确定$\psi(t)$。

上述步骤基本是在频域内进行的，也可在时域内进行：

（1） 由$\varphi(t)$及公式$h_k=\frac{1}{\sqrt{2}}\int_R\varphi(\frac{t}{2})\overline{\varphi}(t-k)dt$.确定$\{h_k\}_{k\in z}$；

（2） 由式$g_n=\overline{h}_{1-n}(-1)^{1-n}(n\in z)$确定$\{g_n\}_{n\in z}$； （3.43）

（3） 由$\psi(t)$，$\{g_n\}_{n\in z}$及式$\frac{1}{\sqrt{2}}\psi(\frac{t}{2})=\sum_{k\in z}g_k\varphi(t-k)\Psi(\omega)$确定$\psi(t)$。

3.4.3 Mallat 算法

3.4.3.1 函数的正交小波分解和多尺度逼近

由一个给定的多尺度分析$(\{V_j\}_{j\in z},\varphi(t))$可确定一个小波函数$\psi(t)$和相应的小波空间$\{W_j\}_{j\in z}$，且$L^2(R)=\bigoplus_{j,k\in z}W_j$。因此，对任何$f(t)\in L^2(R)$，有

$$f(t)=\sum_{j,k\in z}d_{j,k}\psi_{j,k}(t) \tag{3.44}$$

其中$d_{j,k}=<f(t),\psi_{j,k}(t)>$，$j,k\in z$由离散小波变换的定义知，$\{d_{j,k}\}_{j,k\in z}$就是$f(t)$的离散小波变换。由于此时$\{\psi(t)_{j,k}\}_{j,k\in z}$是$L^2(R)$的正交基，故$\{d_{j,k}\}_{j,k\in z}$是$f(t)$的正交小波变换，式（3.44）就是它的重构公式，也称为$f(t)$的正交小波分解。

若记$d_{j,k}\psi_{j,k}(t)=g_i(t)$，则$g_i(t)\in W_j$，而$W_j\oplus V_j=V_{j-1}$，$V_j$的频率范围恰是$V_{j-1}$的一半，且是$V_{j-1}$中的低频表现部分，所以，$W_j$的频率表现在$V_j$与$V_{j-1}$之间的部分，它表现的是一个有限频带。所以，通常$V_j$表现了$V_{j-1}$的“概貌”，$W_j$表现了$V_{j-1}$的“细节”。由于$W_j$的频带是互不重叠的，所以$W_j$表现的是不同频带中的“细节”。此时式（3.44）可写成

$$f(t)=\sum_{j\in z}g_i(t) \tag{3.45}$$

它说明，任何一个函数$f(t)\in L^2(R)$，可以分解成不同频带的细节之和。随着j的不同，这些频带是互不重叠，且充满整个频率空间的。这样，正交离散小波变换$d_{j,k}$的时－频窗是互不重叠、相互邻接的，它们形成了对时－频平面的一种剖分。

在实际中，任何函数$f(t)\in L^2(R)$，它实际上只有有限的细节，因为物理仪器

记录下的信号总是只有有限的分辨率，我们可以假设 $f(t)\in V_0$，将有最精细的细节的函数空间记为 V_0

由于

$$\begin{aligned} V_0 &= V_1 + W_1 = V_2 + W_2 + W_1 \\ &= \mathrm{L}\ \mathrm{L} \\ &= V_J + W_J + W_{J-1} + \mathrm{L} + W_1 \end{aligned}$$

所以 $f(t) = f_J(t) + g_J(t) + g_{J-1}(t) + \mathrm{L} + g_1(t)$，

即
$$f(t) = \sum_{k\in z} c_{J,k}\varphi_{J,k}(t) + \sum_{i=1}^{J}\sum_{k\in z} d_{j,k}\psi_{j,k}(t)\,, \tag{3.46}$$

其中
$$c_{J,k} = <f(t), \varphi_{J,k}(t)>,\ k\in z\,, \tag{3.47}$$

$$d_{i,k} = <f(t), \psi_{i,k}(t)>,\ k\in z\,, \tag{3.48}$$

式（3.46）中的第一项 $f_J(t)$，是 $f(t)$ 在尺度 J 下的一种逼近，J 越大，逼近程度越差，是 $f(t)$ 的第 J 级“模糊像”，它表示的是 $f(t)$ 的频率不超过 2^{-J} 的成份；而第二项中的 $g_i(t)$，是 $f(t)$ 的频率在 2^{-i} 到 2^{-i+1} 之间的细节成份。式（3.46）对所有的 $J(J\geq 1)$ 成立，也就是说，我们可得到不同尺度 J 下的逼近式。我们称

$$f_J(t) = \sum_{k\in z} c_{J,k}\varphi_{J,k}(t) \tag{3.49}$$

为 $f(t)$ 的尺度为 J 的连续逼近，称其系数（3.47）为 $f(t)$ 的离散逼近，称

$$g_i(t) = \sum_{k\in z} d_{i,k}\psi_{i,k}(t) \tag{3.50}$$

为 $f(t)$ 在尺度 i 下（或频率 2^{-i}）的连续细节，称其系数（3.48）为 $f(t)$ 的离散细节。

3.4.3.2 快速算法

当 $\varphi(t), \psi(t)$ 已确定时，要想得到函数 $f(t)\in L^2(R)$ 的多尺度逼近 $f_j(t)$，只需知道 $\{c_{J,k}\}_{k\in z}$，同样，要想得到 $f(t)$ 在尺度 j 下的细节，只需知道 $\{d_{j,k}\}_{k\in z}$，这也是我们将它们称为离散逼近和离散细节的原因。而 $\{c_{J,k}\}_{k\in z}$ 与 $\{d_{j,k}\}_{k\in z}$ 的计算对于

j有传递关系，见下两式：

$$c_{j+1,k}=\sum_{n\in z}c_{j,n}\bar{h}_{n-2k},\quad k\in z \tag{3.51a}$$

$$d_{j+1,k}=\sum_{n\in z}c_{j,n}\bar{g}_{n-2k},\quad k\in z \tag{3.51b}$$

式（3.51）和（3.52）便是 Mallat 快速算法。从两式中可看出，只要知道双尺度方程中的传递系数$\{h_n\}_{n\in z}$和$\{g_n\}_{n\in z}(g_n=(-1)^{1-n}\bar{h}_{1-n})$，就可由$\{c_{0,n}\}_{n\in z}$计算出$\{c_{1,n}\}_{n\in z}$，$\{d_{1,n}\}_{n\in z}$，再由$\{c_{1,n}\}_{n\in z}$计算出$\{c_{2,n}\}_{n\in z},\{d_{2,n}\}_{n\in z}$，由$\{c_{2,n}\}_{n\in z}$计算出$\{c_{3,n}\}_{n\in z},\{d_{3,n}\}_{n\in z}$,L ,其过程可由图 5 示意。

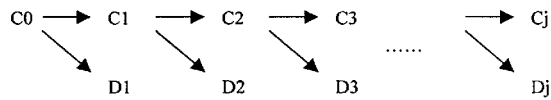


图 3.6 分解算法示意图

以上是快速分解公式，同样由$\{c_{j+1,n}\}_{n\in z}$和$\{d_{j+1,n}\}_{n\in z}$可以重构出$\{c_{j,n}\}_{n\in z}$的快速算法。

$$c_{j,k}=\sum_{n\in z}c_{j+1,n}h_{k-2n}+\sum_{n\in z}d_{j+1,n}g_{k-2n},\quad k\in z \tag{3.52}$$

这即是由$\{c_{j+1,n}\}_{n\in z}$和$\{d_{j+1,n}\}_{n\in z}$重构$\{c_{j,n}\}_{n\in z}$的重构公式，它可用图 6 示意：

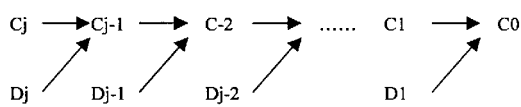


图 3.7 重构算法示意图

3.5 Mallat 算法滤波器卷积实现的 MATLAB 仿真分析

3.5.1 小波分析、综合与卷积的关系

公式（3.51）将较粗略尺度 j+1 上的 DWT 系数与较精确尺度 j 上的 DWT 系数联系起来。为方便观察起见，将分析综合公式改写为：

$$c_{j+1}[k]=\sum_{n=-\infty}^{\infty}c_j[n]h[n-2k] \tag{3.51'a}$$

$$d_{j+1}[k]=\sum_{n=-\infty}^{\infty}c_j[n]g[n-2k] \tag{3.51'b}$$

$$c_j[k]=\sum_{n=-\infty}^{\infty}c_{j+1}[n]h[k-2n]+\sum_{n=-\infty}^{\infty}d_{j+1}[n]g[k-2n] \tag{3.52'}$$

分解公式类似于傅立叶变换中的卷积运算：

$$x[k]*h[k]=\sum_{n=-\infty}^{\infty}x[n]h[k-n]$$

常规卷积与分解公式之间的相似性表明，尺度函数系数 h_k 和小波函数系数 g_k 可被看成是滤波器的脉冲响应。所以 DWT 分析是一种形式的滤波。实际上，DWT 分析与简单的卷积只有两方面不同：首先，脉冲响应的采样点是倒置的，用 h_{n-k} 代替了 $h[k-n]$；其次，标号 k 是乘 2 的。分析公式的这个乘 2 的标号意味着卷积后每逢第二个采样点要被去掉。

公式（3.51）的实现效率很高。对于一个长度为 N 的信号，DFT 需要用 N^2 次乘法和 N^2 次加法，FFT 的加法和乘法次数均为 $N\log_2 N$，而 DWT 只需 N 次乘法和 N 次加法。

公式（3.52）为重构公式或称综合公式，该公式将较精细尺度 j 上的 DWT 系数和较粗略尺度 $j+1$ 上的 DWT 系数联系起来。它实现的是逆 DWT(IDWT)功能。综合公式中的各项与常规卷积非常相似，因此也暗含着滤波作用，唯一不同的是标号 n 乘 2，其效果是脉冲函数的采样点隔一个丢弃一个，或者说相当于在 $c_{j+1,n}$ 和 $d_{j+1,n}$ 的采样点之间插入 0。

如上所述，系数 h_k 和 g_k 可以看成是一对滤波器的脉冲响应。它表明对于所有的小波族来说，h_k 的作用是一个低通滤波器，而 g_k 的作用是一个高通滤波器。通过求它们的离散傅里叶变换的幅度响应，通常可以得到滤波器的波形。图 3.8 显示了 Daubechies-4 小波族中两个互补性很好的滤波器。它们增益曲线的交点是在它们最大值的 $3dB$ 处。这就意味着这两个滤波器将频谱等分成了两个频率段。当对一个信号滤波时，低通滤波器的输出包含了信号的低频分量，而高通滤波器的输出是信号的高频分量，滤波器特性之所以互补是因为尺度函数系数 h_k 和小波函数系数 g_k 有如（3.43）式所示关系，重写如下：

$$g_n=\overline{h}_{1-n}(-1)^{1-n}(n\in z)。$$

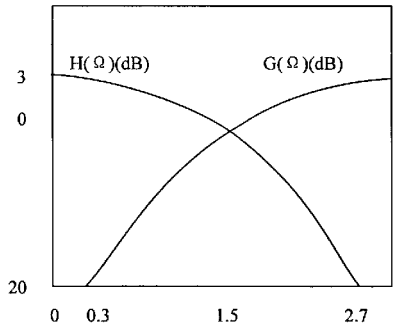


图 3.8　Daubechies-4 族的低通和高通滤波器

对于许多小波函数族来说，分析和综合所使用的是相同的小波和尺度函数滤波器 h_k 和 g_k。然而，对于双正交小波（biorthogonal wavelet）族，综合滤波器与分析滤波器是不同的。两套滤波器的系数由下面两个公式建立联系的：

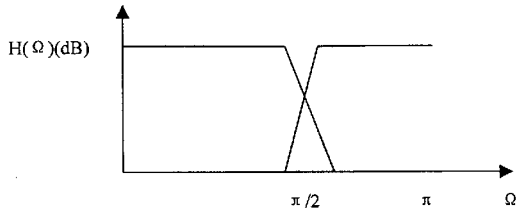
$$h_k = (-1)^k g_{N-1-k} \tag{3.54}$$

$$g_k = (-1)^k h_{N-1-k} \tag{3.55}$$

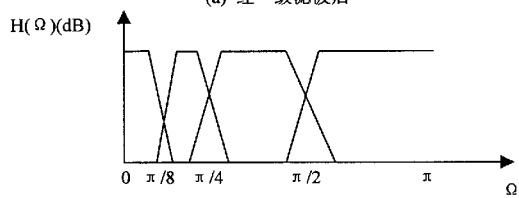
下面从频域角度理解小波的分解和综合公式。小波分析的思想是识别信号中的细节程度。由上面的分析引入了滤波器的思想，即第一个脉冲函数 h_k 定义了尺度函数，具有低通特性。第二个脉冲函数 g_k 定义了小波函数。具有高通特性。

图 3.8 显示了 Daubechies-4 族的两个滤波器的波形。在小波分析的第一步，待分析的信号通过这个低通和高通滤波器分解为两部分。高通滤波器拾取小的细节，低通滤波器拾取信号的其他分量。结果如图 3.8 在频域所示的那样，信号被分解成两半：一个由尺度函数平移所确定的低通部分和一个由小波函数平移所确定的高通部分。对信号的低通部分持续进行这个过程，用低通和高通滤波器来对它进行滤波。每一步都会找出信号的进一步的细节，只要细节存在，就可以继续进行这种对低频带的子分解过程。

图 3.9（b）显示了三个分析级。对于低频分量来说，好的频率分辨率是最重要的。对低通部分再滤波是为了获得最低频部分最佳的频率分辨率，尽管在较高频率部分得到的频率分辨率较差。图 3.9（b）中的滤波器的宽度表明了它们的频率分辨率：滤波器越窄，所能分辨出的频率就越精细。



(a) 经一级滤波后



(b) 经三级滤波后

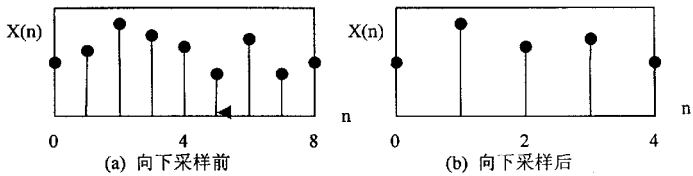
图 3.9 DWT 滤波器形状

3.5.2 离散二进小波变换的计算

上节从频域对 DWT 的考察可为式（3.51'）和（3.52'）提出的小波分析和综合公式提供一种可行的解释。小波分析式（3.51'）实现滤波和向下采样。式（3.51'a）指出，将系数 $c_j[n]$ 与翻转的低通脉冲响应 $h[-k]$ 进行卷积实现滤波，再将结果进行向下采样获得 $c_{j+1}[k]$。在式（3.51'b）中，滤波由系数 $c_j[n]$ 与翻转的高通脉冲响应 $g[-k]$ 进行卷积来实现，再将滤波输出向下采样得到系数 $d_{j+1}[k]$。在这两个分析公式中，上节定义的并示于图 3.10 的向下采样，通过每两点丢弃一点的方法使采样频率降低一半。表现为标号 k 乘以 2，以产生向下采样作用。图 3.12a 显示了一个描述 DWT 分析公式的框架。其中 HP 指的是与高通滤波器 $g[-k]$ 相卷积，而 LP 指与低通滤波器 $h[-k]$ 相卷积。向下采样用符号 $\downarrow 2$ 来表示。

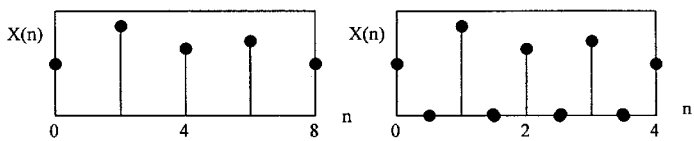
上节把综合描述为向上采样，然后再进行滤波，以及低频与高频分量相加。小波综合公式（3.52'）完成的是滤波任务，将系数 $c_{j+1}[n]$ 和 $d_{j+1}[n]$ 与脉冲响应 $h[k]$ 和 $g[k]$ 分别进行卷积。向上采样的作用室友标号 n 乘 2 产生的，其效果是再每一对 $c_{j+1}[k]$ 和 $d_{j+1}[k]$ 采样点之间加入零。图 3.11 说明了这一点。图 3.12b 显示

了小波综合的过程：系数$d_{j+1}[k]$先向上采样再进行高通滤波，$c_{j+1}[k]$先向上采样再进行低通滤波，这两部分相加得到系数$c_j[k]$。



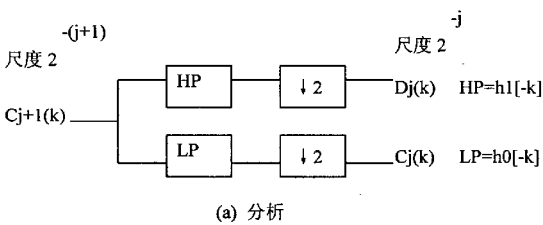
(a) 向下采样前　　(b) 向下采样后

图 3.10 从时域角度看 1/2 向下采样

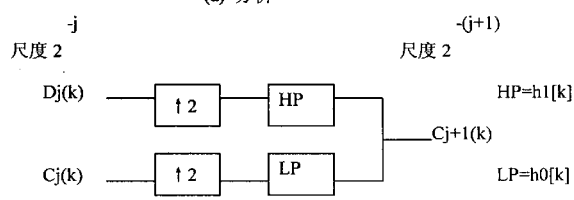


(a) 向上采样前　　(b) 向上采样后

图 3.11 从时域角度看 1/2 向上采样



(a) 分析



(b) 综合

图 3.12　DWT 分析与综合

尽管 DWT 分析公式（3.51'a）将一个尺度上的系数与下一个较精细尺度上的系数联系起来，但仍然存在这样一个问题：如何开始分析？换句话说，在第一步中使用的一组$c_j[k]$是什么？$c_j[k]$应当是用V_j子空间的尺度函数描述$f(t)$的系

数，如下式：

$$f(t) \approx f_j(t) = \sum_{k=-\infty}^{\infty} c_j[k] 2^{-j/2} \varphi(2^{-j}t - k)$$

从上式可以看出，正确的程序是用某个足够小尺度的尺度函数来描述一组信号的采样点，然后用其系数$c_j[k]$作为 DWT 分析的起点。然而实际上，几乎总是用信号的采样点本身（缩放$2^{-j/2}$倍之后）作为第一组系数。之所以要这样做，是因为在最精细的尺度上，尺度函数（以及小波函数）看上去类似脉冲函数，因为它们的脉冲宽度很窄而且接近单位高度。如果一个数字信号可以用脉冲函数的和来表示，也就可以近似地用非常精细地类似脉冲函数地尺度函数的和来表示了。举例来说，一个在V_3子空间的函数可以写成如下形式：

$$\begin{aligned} f(t) = \mathrm{L} &+ c_3[0] 2^{-3/2} \varphi(2^{-3}t) + c_3[1] 2^{-3/2} \varphi(2^{-3}t - 1) \\ &+ c_3[2] 2^{-3/2} \varphi(2^{-3}t - 2) + c_3[3] 2^{-3/2} \varphi(2^{-3}t - 3) + \mathrm{L} \end{aligned}$$

因为任何一个数字信号都可以写为脉冲函数和：

$$f[n] = \mathrm{L} + f[0]\delta[n] + f[1]\delta[n-1] + f[2]\delta[n-2] + f[3]\delta[n-3] + \mathrm{L}$$

随着尺度函数在更精细的尺度上越来越像脉冲函数，就越来越有理由使信号的采样值$f[k]$等于$c_j[k]2^{-j/2}$，即对于子空间V_j中的函数采样点$f[k]$，使

$$c_j[k] = 2^{j/2} f[k] \tag{3.56}$$

一般可以用这样的近似得到可接收的结果：在由标号 j 给定的某个足够精细的尺度上，尺度函数系数可以用信号的采样值乘以$2^{j/2}$来代替。

无论多少级的 DWT 分析都可以实现。由于向下采样，低通 DWT 系数$c_{j+1}[k]$的数目是$c_j[k]$的 1/2。假设，最大的尺度是 j=0，那么2^N个采样点的信号长度允许最多 N 级分析。第 N 级也就是最后一级产生 DWT 系数$c_0[k]$。对于高通的 DWT 系数$d_j[k]$也一样。例如，在V_8中具有 256 个采样点的信号$f(t)$，经过第一级分析生成 128 个低通采样点（即 DWT 系数$c_7[k]$），以及 128 个高通采样点（即 DWT 系数$d_7[k]$）。下一级，低通系数$c_7[k]$进一步被分析产生 64 个采样点的低通系数$c_6[k]$和 64 个采样点的高通系数$d_6[k]$。第 8 级用两个低通系数$c_1[k]$生成单一的低通系数$c_0[k]$和单一的高通系数$d_0[k]$。

DWT 综合可以从任何尺度上的 $d_j[k]$ 和 $c_j[k]$ 系数开始。根据式(3.52')，DWT 系数 $c_0[k]$ 和 $d_0[k]$ 用来重新生成系数 $c_1[k]$；接下来，再将 $c_1[k]$ 和 $d_1[k]$ 合并生成系数 $c_2[k]$，然后将 $c_2[k]$ 和 $d_2[k]$ 合并生成 $c_3[k]$。由于向上采样的原因，每一步综合的结果，系数 $c_{j+1}[k]$ 的数量是 $c_j[k]$ 的 2 倍。因为已知 $c_0[k], d_0[k], d_1[k], d_2[k]$ 就能最终计算出 $c_3[k]$，所以可以用这组系数对子空间 V_3 中的任何信号进行编码。

3.6 第二代小波－基于快速提升的算法

小波变换的 Matlab 算法中，信号经过卷积运算后再按隔二取一的周期进行采样，这意味着卷积运算中有一半是无用功，另外一个缺点是无法即时用离散小波变换系数替换对应点的源信号数据，因此要占用更多的内存。为此 Sweldens W. 提出了小波变换的快速提升算法，成为第二代小波的关键技术。

它的基本思想是先将源信号划分为奇数信号集合和偶数信号集合，然后对它们交替进行预测和更新的操作，最后偶序列对应于低频分量，奇序列对应于高频分量。

基于提升小波的滤波的另一个特点就是计算简单，不论是分解或重构都不涉及复杂的内积运算，而且它还提供了无损压缩的能力。对于基于卷积的滤波，即使是采用整数型的(5，3)滤波器组，随着分解层数的加深，由于计算机无法为完整的表示滤波系数提供足够的精度，从而出现信息丢失；而基于提升小波的滤波则可以和量化器结合提供一个整数到整数的压缩框架，以实现无损压缩。

小波提升分为三个步骤：分裂、预测和更新，如图 3.19 所示。

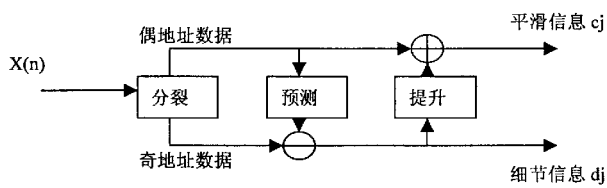


图 3.19 提升形式

首先将信号分解成奇数和偶数的样本，然后交替地使用对偶预测（用一个滤波器对偶数样本进行滤波，然后用奇数样本减去滤波后地结果）和提升（用一个滤波器对奇数样本进行滤波，然后用偶数样本减去滤波后的结果），经过 M 次的预测与提升以后，偶数样本就相应于低通小波系数，技术样本相应于高通小波系数。最后经过一次以 K 为比例因子的伸缩变换。

下面由 Mallat 算法出发推导提升格式。

记 $x(n)$，$h(n)$ 和 $g(n)$ 的偶部和奇部分别为：

$$x_e(n)=x(2n),\quad x_o(n)=x(2n+1)$$

$$h_e(n)=h(2n),\quad h_o(n)=h(2n+1)$$

$$g_e(n)=g(2n),\quad g_o(n)=x(2n+1)$$

于是，根据 Mallat 算法，有

$$x_{low}(k)=\sum_n x(n)h(n-2k)=\sum_n x_e(n)h_e(n-k)+\sum_n x_o(n)h_o(n-k) \tag{3.57}$$

取 Z 变换可得

$$X_{low}(z)=X_e(z)H_e(z^{-1})+X_o(z)H_o(z^{-1}) \tag{3.58}$$

同样地，有

$$X_{high}(z)=X_e(z)G_e(z^{-1})+X_o(z)G_o(z^{-1}) \tag{3.59}$$

用矩阵的形式同一表示，有

$$\begin{bmatrix}X_{low}(z)\\X_{high}(z)\end{bmatrix}=\begin{bmatrix}H_e(z^{-1}) & H_o(z^{-1})\\G_e(z^{-1}) & G_o(z^{-1})\end{bmatrix}\begin{bmatrix}X_e(z)\\X_o(z)\end{bmatrix}=[P_p(z^{-1})]^T\begin{bmatrix}X_e(z)\\X_o(z)\end{bmatrix} \tag{3.40}$$

其中，$P_p(z)=\begin{bmatrix}H_e(z) & G_e(z)\\H_o(z) & G_o(z)\end{bmatrix}$。于是，Mallat 算法也可以用图 3.20 所示的等价算法来代替。用提升格式来实现小波变换就是要用若干个提升过程来实现图 3.20 算法中的向量滤波 $P(z^{-1})$。以 CDF9/7 双正交小波变换为例，来说明这个实现过程。

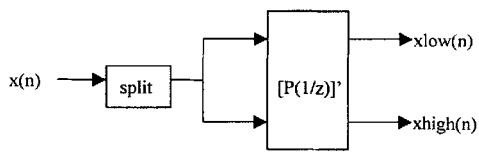


图 3.20　小波变换（Mallat 算法的等价算法

CDF9/7 双正交小波变换的低通和高通滤波器的冲击响应分别为：

$$h(0)=h_0=0.8527$$

$$h(1)=h(-1)=h_1=0.3774$$

$$h(2)=h(-2)=h_2=-0.11062$$

$$h(3)=h(-3)=h_3=-0.02385$$

$$h(4)=h(-4)=h_4=0.03783$$

可以推导出

$$H_e(z) = h_4 z^{-2} + h_2 z^{-1} + h_0 + h_2 z + h_4 z^2$$
$$H_o(z) = h_3 z^{-1} + h_1 + h_1 z + h_3 z^2 \tag{3.41}$$

由 h 值根据公式可得 g 值：

$$g(-2) = g_{-2} = 0.06454 \qquad g(-1) = g_{-1} = -0.0406$$
$$g(0) = g_0 = -0.41809 \qquad g(1) = g_1 = 0.78849$$
$$g(2) = g_2 = -0.41809 \qquad g(3) = g_3 = -0.04069$$
$$g(4) = g_4 = 0.06454$$

同样可以推导出

$$G_e(z) = g_4 z^{-2} + g_2 z^{-1} + g_0 + g_{-2} z$$
$$G_o(z) = g_3 z^{-1} + g_1 + g_{-1} z \tag{3.42}$$

将滤波器多项式矩阵 $P(z)$ 进行因式分解，生成一系列三角矩阵的连乘积，其结果如下式：

$$\begin{aligned} P(z) &= \begin{bmatrix} H_e(z) & G_e(z) \\ H_o(z) & G_o(z) \end{bmatrix} \\ &= \begin{bmatrix} 1 & \alpha(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \beta(1+z) & 1 \end{bmatrix} \begin{bmatrix} 1 & \gamma(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \delta(1+z) & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix} \end{aligned} \tag{3.43}$$

其中，比例因子 K 和提升系数 α、β、γ 和 δ 分别为

$$\left.\begin{aligned} \alpha &= h_4 / h_3 \approx -1.586134343 \\ \beta &= h_3 / r_1 \approx -0.052980118 \\ \gamma &= r_1 / s_0 \approx 0.882911075 \\ \delta &= s_0 / t_0 \approx 0.443506852 \\ \mathrm{K} &= t_0 \approx 1.230174105 \end{aligned}\right\} \tag{3.44}$$

$$\left.\begin{aligned} r_0 &= h_0 - 2h_4 h_1 / h_3 \\ r_1 &= h_2 - h_4 - h_4 h_1 / h_3 \\ s_0 &= h_1 - h_3 - h_3 r_0 / r_1 \\ t_0 &= r_0 - 2r_1 \end{aligned}\right\} \tag{3.45}$$

根据式(3.43)可得多项式矩阵 $[P(z^{-1})]^T$ 的因式分解结果。

$$[P(z^{-1})]^T = \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix} \begin{bmatrix} 1 & \delta(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \gamma(1+z) & 1 \end{bmatrix} \begin{bmatrix} 1 & \beta(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \alpha(1+z) & 1 \end{bmatrix}$$

假定一维输入信号 $\chi = (x_n)\, n \in Z$，首先把它进行抽样提取，分成互不相交的两部分：偶下标抽样 $\chi_e = (x_{2n})\, n \in Z$ 和奇下标抽样 $\chi_o = (x_{2n+1})\, n \in Z$，然后在其首尾边界处进行镜像扩展，输出信号用 $\gamma = (y_n)\, n \in Z$ 表示，根据式（3.40）可得

CDF9/7 小波滤波器的提升实现结构，如图 3.21 所示。

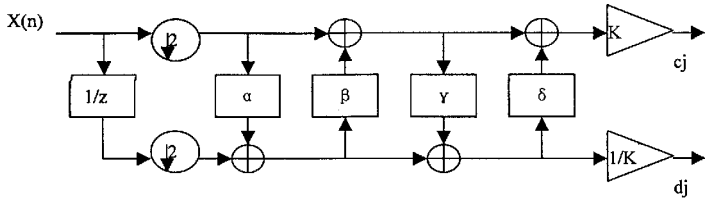


图 3.21　CDF9/7 小波滤波器的提升实现结构

第四章　SPIHT 图像编解码

4.1　简介

众所周知，图像压缩特别是非可逆或有损压缩，编码效率与计算复杂度会同步增长，这是由信息论中的信源编码原理决定的，即某种信源编码技术当计算量达到无限时效率接近最优。然而，近年来由 Shapiro 提出的内嵌零树编码(EZW)以及由 Said 和 Pearlman 在此基础上提出的改进算法——基于零树的集分割算法(SPIHT)打破了这种限制。这种技术不但在性能上与最复杂的编码算法相比毫不逊色，而且其执行速度也相当快，另外它还具有内嵌编码的特征，能够终止在所需的码率或图像质量处，有利于逐渐浮现式的图像传输。

4.2　内嵌编码

内嵌编码具体采用渐进传输技术来实现的，即编码器将待编码的比特流按重要性的不同进行排序，重要的比特先编码，根据目标码率或失真度大小要求随时结束编码；同样，对于给定码流解码器也能够随时结束解码，并可以得到相应码流截断处的目标码率的恢复图像。

假定将初始图像表示为一系列的象素值 $p_{i,j}$，(i,j) 是象素坐标。为简单起见，我们用粗体 **P** 来代表这种二维数组。编码就是对这种二维数组进行操作：

$$\mathbf{c}=\Omega(\mathbf{P}) \tag{4.1}$$

$\Omega(\cdot)$ 具体代表将图像 **P** 进行离散小波变换后再进行整体分集子带传输操作。二维数组 c 与 **P** 有相同的尺寸，数组内每一个元素 $c_{i,j}$ 称为在坐标 (i,j) 处的传输系数。为了进行编码，需要将 $c_{i,j}$ 转换为定点二进制格式，通常用 16 位或更少的位数来表示。

在渐进传输技术里，解码器将要重构的二维向量 $\hat{c}$ 初始化为 0，然后根据接收到的编码流逐渐提高数组元素值的分辨率，之后将形成的系数矩阵进行小波反变换获得重构图像：

$$\hat{P}=\Omega(\hat{c}) \tag{4.2}$$

渐进传输技术的主要目标是选择最重要信息——使失真度降低到最小的系数——最先传输。选择的标准是均方误差（MSE：Mean Squared-Error）失真来量度：

$$D_{mse}(P-\hat{P})=\frac{\|P-\hat{P}\|}{N}=\frac{1}{N}\sum_i\sum_j(p_{i,j}-\hat{p}_{i,j})^2 \tag{4.3}$$

N 是图像象素数量。更进一步，由于变换 $\Omega(\cdot)$ 不会改变欧几里得均方值，也就是

说有：

$$D_{mse}(P-\hat{P}) = D_{mse}(c-\hat{c}) = \frac{1}{N}\sum_i\sum_j (c_{i,j}-\hat{c}_{i,j})^2 \qquad (4.4)$$

上式清楚表明当传输系数 $c_{i,j}$ 的精确值传送到解码器后，MSE 与 $|\hat{c}_{i,j}|^2/N$ 成正比。

这意味着量级大的系数应该首先传输，因为它们有更多的信息量。也即将 $|\hat{c}_{i,j}|^2$ 的值根据它的二进制表示来排序，将系数中所有的 SMB（Most Significant Bits）首先传输。这种方法在渐进传输中被称为位平面编码。

下面提供结合这两种思想——系数按量级排序及 SMB 最先传输——的渐进传输方案的一种实现。为简化起见，假定排序信息已经详细地传给解码器。以后会介绍一种更有效率的排序信息的编码方法。

图 4.1 显示了系数按幅值排序的链表的二进制表示。

Sign		s	s	s	s	s	s	s	s
Msb	4	1	1	0	0	0	0	0	0
	3	--	>	1	1	0	0	0	0
	2	--	--	--	>	1	1	0	0
	1	--	--	--	--	--	>	1	0
Lsb	0	--	--	--	--	--	--	>	1

图 4.1 系数按幅值排序的二进制表示

每一列代表一个系数。顶行表示系数的符号，行号从底向上递增，底行代表 lsb。所有位于同一行的位含有相同的信息量，因此最有效的系数传输方法是从顶行开始按顺序传送每一行的位，正如图中箭头所示。另外，还要传输系数在 $[2^n, 2^{n+1})$ 范围内的数量 μ_n。在表中有 $\mu_4 = 2, \mu_3 = 2, \mu_1 = 1, etc.$

上面所描述的渐进传输可用以下算法来实现。

算法 1：

1）初始化：输出 $n = \left\lfloor \log_2(\max_{(i,j)}\{|c_{i,j}|\}) \right\rfloor$ 到解码器；

2）排序过程：输出 μ_n，所有范围在 $[2^n, 2^{n+1})$ 的象素坐标 $\eta(k)$ 和符号位；

3）细化过程：对于所有 $|c_{i,j}| \geq 2^{n+1}$ 的系数，输出第 n 层重要位的值；

4）步长更新：n 自减，调回步骤 2。

上述算法可以截止在任何所需的码率和失真度要求下。通常，如果幅值相当小的象素片也被传输的话就会重构出质量相当高的图像。

这种编码算法采用了均匀标量量化，也许有人会觉得不如其他的使用非均匀

或/和矢量量化的算法。而事实并非如此，排序信息使这种简单的量化方法非常有效。另一方面，如果采用其他量化方法在排序过程中大片象素的位预测会消耗大量的计算时间，大大增加了算法的复杂性。

4.3 集分割排序算法

改进上述算法的一个主要特点是不传输有关排序的详细信息。它是基于这样一种事实，即任何算法的执行过程被定义为不断比较它的各个分支点，并输出选择的结果。因此，如果编码和解码有相同的排序算法，则在解码器收到这种幅值比较的结果后即可复制编码器的执行路径，并由此恢复排序信息。

而在排序算法设计上的一个重要特点是不需要对系数进行排序。事实上算法只需简单地选择幅值介于$[2^n, 2^{n+1})$之间的系数，在每次排序循环中 n 递减。给定 n，如果$|c_{i,j}| \geq 2^n$我们说该系数是重要的，否则是不重要的。

排序算法将像素点集分成各种子集T_m并对其执行幅值检测

$$\max_{(i,j)\in T_m} \{|c_{i,j}|\} \geq 2^n \tag{4.5}$$

如果解码器收到的检测结果是“no”（子集是不重要的），则T_m里 的所有系数都是不重要的。如果结果是“yes”（子集是重要的），之后根据某种与编码器里的一致的规则将改集合T_m进一步分成新的子集$T_{m,l}$，然后对新的集合进行重要性检测。这种集合划分操作一直持续到对重要集合的所有单个象素都进行了幅值检测以期辨认出每一个重要的系数。

为了弄清幅值比较和信息位的关系我们使用函数

$$S_n(T) = \begin{cases} 1, & \max\limits_{(i,j)\in T_m} \{|c_{i,j}|\} \geq 2^n \\ 0, & otherwise \end{cases} \tag{4.6}$$

来确认坐标集 T 的重要性。

4.4 空间定位树

通常，图像能量的大部分集中在低频区。因此当从子带金字塔的顶层移到底层时，差异就会越来越小。更进一步的观察发现在子带之间有空间自相似性，如果我们顺着相同的空间定位从金字塔的顶层往下移的话，系数就会表现出更强的幅值有序性。例如，分布于金字塔最高层的大块的低幅值区域在较低层的相应的空间区域的幅值也低。

这种被称为空间定位树的树结构，自然定义了金字塔层次间的关系。图 4.2 显示了拥有四条递归切分子带的金字塔结构的空间定位树的定义。树内每一节点相应于一个像素并以像素坐标定位。节点或者没有后继节点或者有四个后继节

点。金字塔最高层的像素是根结点，它们的分支规则不同，仅有三个兄弟节点。像素主要分成以下四类坐标集：

- H：空间定位树所有根结点的坐标集合（塔形结构最高层的节点）；
- $O(i,j)$：点(i,j)的直接后继（儿子）的坐标集合；
- $D(i,j)$：点(i,j)的所有后继（子孙）的坐标集合；
- $L(i,j)$：点(i,j)除直接后继外的所有后继的坐标集合（孙子），

$L(i,j)=D(i,j)-O(i,j)$。

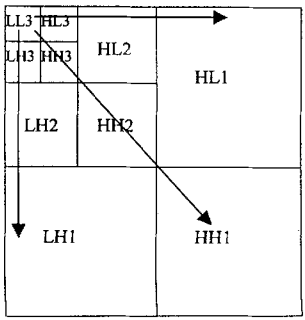


图 4.2 空间定位树的层次关系

比如，除了塔形结构的最高层和最低层，有

$$O(i,j)=\{(2i,2j),(2i,2j+1),(2i+1,2j),(2i+1,2j+1)\}. \tag{4.7}$$

我们在排序算法中使用空间定位树来分割子集。子集分割规则简述如下：

$$\begin{cases} H(i,j)=\{(i,j)\}+D(i,j) \\ D(i,j)=O(i,j)+L(i,j) \\ L(i,j)=\sum_{k,l} D(k,l),\ \ (k,l)\in O(i,j) \end{cases} \tag{4.8}$$

图 4.3 直观显示了上述集合的定义。

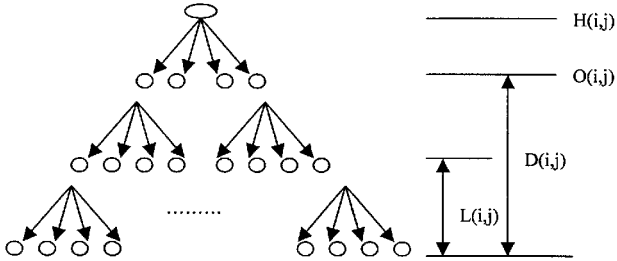


图 4.3 算法中集合定义的直观显示

4.5 编码算法

4.5.1 SPIHT 编码算法

既然用以测试子集重要性的顺序很重要，因此在实际操作中这种重要信息被存成三个顺序链表，即：

- LSP（List of Significant Pixels）：重要像素列表；
- LIP（List of Insignificant Pixels）：不重要像素列表；
- LIS（List of Insignificant Sets）：不重要集合列表，表内集合又可分为两类即 A 类（$D(k,l)$）或 B 类（$L(i,j)$）。

链表内每一条目是像素坐标，LSP 与 LIP 内的是单一像素坐标，LIS 的是坐标的集合。

在排序过程中（见算法 1），LIP 内的像素一在上次循环中不重要的一被测试，那些在新的循环中重要的被移到 LSP。类似的，集合在 LIS 内按顺序连续测试，如果是重要的，它就移出 LIS 并被分割。那些拥有不止一个元素地新集合被重新加到 LIS 链表末尾。而单一元素集合被加到 LIP 或 LSP 的末尾，依赖于它们是否是重要像素。LSP 内包含了在下一步幅值细化过程中将要访问的像素坐标。

下面是新算法的完整流程。它本质上与算法 1 一样，只不过在排序过程中使用了集合分割策略。

算法 2：

1）初始化：输出 $n=\left\lfloor \log_2(\max_{(i,j)}\{|c_{i,j}|\}) \right\rfloor$；令 LSP 为空列表，将 $(i,j)\in H$ 的点坐标加入 LIP 和 LIS 中，后者里的并同时被置为 A 类；

2）排序过程：

2.1）对 LIP 内的每一像素（i，j）作：

2.1.1）输出 $S_n(i,j)$；

2.1.2）如果 $S_n(i,j)=1$，将（i，j）移到 LSP，并输出 $c_{i,j}$ 的符号；

2.2）对 LIS 内的每一元素（i，j）作：

2.2.1）如果该元素属于 A 类则：

- 输出 $S_n(D(i,j))$；
- 如果 $S_n(D(i,j))=1$，则：
  - 对每一个 $(k,l)\in O(i,j)$ 作：
    - 输出 $S_n(k,l)$；
    - 如果 $S_n(k,l)=1$ 则将 (k,l) 加到 LSP 并输出 $c_{k,l}$ 的符号；
    - 如果 $S_n(k,l)=0$ 则将 (k,l) 加到 LIP 的末尾；
  - 如果 $L(i,j)\neq\varnothing$ 则把 (i,j) 移到 LIS 的末尾，并标记为 B 类；否则，将 (i,j) 从 LIS 内删除；

2.2.2）如果像素是 B 类，则：

- 输出 $S_n(L(i,j))$；
- 如果 $S_n(L(i,j))=1$，则
  - 将每一个 $(k,l)\in O(i,j)$ 加到 LIS 的末尾并标记为 A 类；
  - 将 (i,j) 从 LIS 内删除。

3）细化过程：对每一个 LSP 内的元素 (i,j)，输出 $|c_{i,j}|$ 的第 n 层重要位的值；

4）量化步长更新：n 减 1，返回 2）。

算法 2 中，码率可以精确控制因为传输信息是由单一的位构成。编码器可用（4.4）式来评价失真度降低的程度，并在指定的失真度下结束编码。

注意在算法 2 中，编码器传输了所有基于重要性数据 S_n 的分支条件，而 S_n 只有在知道了系数 $c_{i,j}$ 的信息后才能计算出来。而解码器在对重要系数值排序时复制了编码器的执行步骤，因此为获得所需的解码算法，我们只需简单地将算法 2 中的“输出”换为“输入”即可。比较算法 2 和算法 1 可以看出，当重要系数的坐标被加到 LSP 的末尾时，排序信息 $\eta(k)$ 就被同时恢复了出来。但必须注意的

是一旦解码器输入了编码数据，它的三个控制链表（LSP，LIP 和 LIS）就同编码器输出编码数据时所使用的完全一样了，那就意味着从相同的执行步骤中完全恢复了排序信息。很容易看出在这种方案中编码器和解码器有相同的计算复杂性。

解码器要做的额外工作是更新图像信息。对于某一 n 值当某一坐标移入 LSP 时，它的值是介于$[2^n, 2^{n+1})$范围内的。因此解码器使用该信息加上继之而来的符号位，来形成系数值$\hat{c}_{i,j} = \pm 1.5 \times 2^n$。类似的，在幅值细化过程中，解码器将系数$\hat{c}_{i,j}$加或减$2^n$。在这种方式下，失真率在排序和幅值细化过程中会逐步降低。

如果再加入其它编码方法，比如通过将输出再作熵编码，算法 2 的效率会进一步提高，但这样会消耗大量的编解码时间。

4.5.2 游程编码与哈夫曼编码

本实验通过对 SPIHT 编码数据的具体分析，发现数据中存在大量连续 FF 和 00 数据，因此先对该数据进行了游程编码，即记录数据中连 0 和连 1 的个数数据的压缩方式。比如二进制数据串 110001001 的游程码是 23121。而游程码中大量数据都是不超过一个字节的小数据，因此采用一个字节来记录 0、1 游程的长度，而如果超过一个字节的话，如果是两个字节就在其前附加两个字节的 0 数据，三个字节就在其前附加三个字节的 0 数据，因为游程码中不会出现 0 或 1 的游程长度是 0 的数据，因此多字节的标志数据 0 不会引入解码歧义。

游程编码后再进行哈夫曼编码，它是一种无损最优编码方案。

哈夫曼码是一种最优前缀码，即任一字符的代码都不是其他字符代码的前缀，因此在构造哈夫曼二叉树时，字符只能是叶，而不能是中间结点，而为了达到最优，就要充分利用叶，而不能有冗余，因此该二叉树必须是完全二叉树，即任一结点均有两个儿子结点。在构造哈夫曼二叉树时用到了贪心算法。

第五章　系统仿真及 DSP 实现

基于上述分析，本论文提出了一套基于小波变换的 SPIHT 图像编解码系统。为了验证系统的可行性及性能，采用 MATLAB 对系统中的小波变换部分进行了仿真，因为 MATLAB 具有强大的矩阵处理功能，丰富的矩阵操作函数简化了对一维信号矩阵和二维图像矩阵的操作。然后用 C++语言对小波变换后的系数进行编解码，即实现系统中第二部分的仿真，以大幅降低数据量，C++语言中的链表类正好用来实现 SPIHT 算法中的三个控制链表（LSP，LIP 和 LIS），方便的位操作命令适合处理算法中基于位平面的编解码功能。最后将仿真验证后的系统用 DSP 来实现。

下面先介绍一下用 MATLAB 对系统中的小波变换进行仿真的过程。

5.1　离散二进小波变换的 MATLAB 仿真分析

先进行一维信号的仿真，二维图像可分解为行和列的一维信号的小波变换，故其仿真是基于一维信号的仿真的。

5.1.1　一维 DWT 的 MATLAB 仿真

由第三章的讲解可知，小波变换主要采用 Mallat 算法分解，具体编程可采用两种算法来实现：基于卷积的滤波法和基于提升的滤波法。下面先以正弦白噪声信号为例讲解基于卷积的滤波法。

所谓正弦白噪声信号就是在正弦波形上叠加白噪声的信号，MATLAB 仿真库中有该信号的模型，只需应用命令（load noissin;），即可将它载入 MATLAB 工作环境中。

然后调用用MATLAB的m文件编写的离散二进小波变换函数

[c,l]=mydwtn(x,n,type)

来实现对该信号的小波变换，其中x是待变换的信号，n是小波变换的级数，type是采用的小波类型，变换后的系数存放在c中，变换后各层系数的个数存放在l中。

下面具体说明该函数的实现过程。

尺度函数和小波函数系数值可采用迭代法推导出，一些典型小波族的尺度函数系数和小波函数系数已被发表。尺度函数系数与小波函数系数有如下关系：

$$g[k]=(-1)^k h[N-1-k]$$

N 是滤波器的长度，等于系数 $h[k]$ 中采样点数。

Haar 小波族：　$h[0]=1/\sqrt{2}$，　$h[1]=1/\sqrt{2}$

$g[0]=1/\sqrt{2}$，　$g[1]=-1/\sqrt{2}$

Daubechies-4 小波族：

$$h[0]=0.4830 \quad h[1]=0.8365 \quad h[2]=0.2241 \quad h[3]=-0.1294$$

$$g[0]=-0.1294 \quad g[1]=-0.2241 \quad g[2]=0.8365 \quad g[3]=-0.4830$$

用MATLAB语言描述如下：

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [c,l]=mydwtn(x,n,type)
switch type
case 'db4',
    Ld=[0.4830,0.8365,0.2241,-0.1294];
    Hd=[-0.1294,-0.2241,0.8365,-0.4830];
    Lf=4;
case 'db6',
    Ld=[0.3327 0.8069 0.4599 -0.1350 -0.0854 0.0352];
    Hd=[0.0352 0.0854 -0.1350 -0.4599 0.8069 -0.3327];
    Lf=6;
otherwise,
    Ld=[1/sqrt(2),1/sqrt(2)];
    Hd=[1/sqrt(2),-1/sqrt(2)];
    Lf=2;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

其中Ld是尺度函数h[k]的系数，相当于一低通滤波器，Hd是小波函数g[k]的系数，相当于一高统滤波器，Lf是滤波器系数的个数N。函数中采用了一个switch语句来选择采用type参数所对应的小波族的系数，默认采用haar小波族。

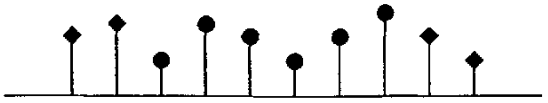
3.5.2小节指出离散小波变换相当于将系数$c_j[n]$与翻转的低通/高通脉冲响应$h[-k]/g[-k]$进行卷积实现滤波，故需再对系数Ld和Hd进行翻转处理：

```
    Ld_=Ld(Lf:-1:1);
    Hd_=Hd(Lf:-1:1);
```

前已指出，可以用信号的采样点本身（缩放$2^{-j/2}$倍之后）作为第一组小波分析的系数。即：

```
    N=nextpow2(length(x));
    Cj=2^(-N/2).*x;
```

在信号的边界部分，由于缺乏周边信号的信息，需先对信号进行扩展。具体的扩展方式受到滤波器的影响，一般采取镜像扩展的方式，如图 5.1 所示。



注：圆头线是源信号，方头线是扩展信号

图 5.1　对一维信号的边界镜像扩展

程序如下：

```
L0=length(Cj);
while length(Cj)<L0+Lf-1
    L=length(Cj);
    if L==1
        L=2;
    end
    Cjj=Cj(1,L-1:-1:1);
    Cj=[Cj,Cjj];
end
Cj=Cj(1,1:L0+Lf-1);
```

接下来采用卷积滤波之后向下采样来实现信号的小波变换：

```
Cj_1=conv(Ld_,Cj);      %low filter coef
Cj_1=Cj_1(1,1:L0+Lf-1);
Dj_1=conv(Hd_,Cj);      %high filter coef
Dj_1=Dj_1(1,1:L0+Lf-1);
%get one every other for high filter coef
c(2^j+1:2^(j+1))=Dj_1(Lf:2:length(Dj_1));
%get every layers' number
l(i+1)=2^j;
i=i-1;
%get one every other for next layer's low filter coef
Cj=Cj_1(Lf:2:length(Cj_1));
if(j==m)
    c(1:2^j)=Cj_1(Lf:2:length(Cj_1));      %Layer 0 's low filter coef
    l(i+1)=2^j;                  %Layer 0 's low filter coef number
end
```

小波综合的过程是先对系数二倍向上采样，再分别进行高低通滤波，之后将滤波结果相加。程序如下：

```
Cj=c(1:2^m);                %original recon coef
for j=0:N-1                 %reconstruction cycle
    Cjt=Cj;
    for i=1:l(j+2)   %set zero every other
        Dj(2*i-1)=c(2^j+i);
        Dj(2*i)=0;
        Cj(2*i-1)=Cjt(i);
```

```
        Cj(2*i)=0;
    end
    %reconstruction signal
    Cj=conv(Lr,Cj)+conv(Hr,Dj);
    %cut out the last length(filter)-1 number
    Cj=Cj(1:2^(j+1));
end
x=2^(N/2).*Cj;
```

初始正弦白噪声信号波形如图5.2所示：

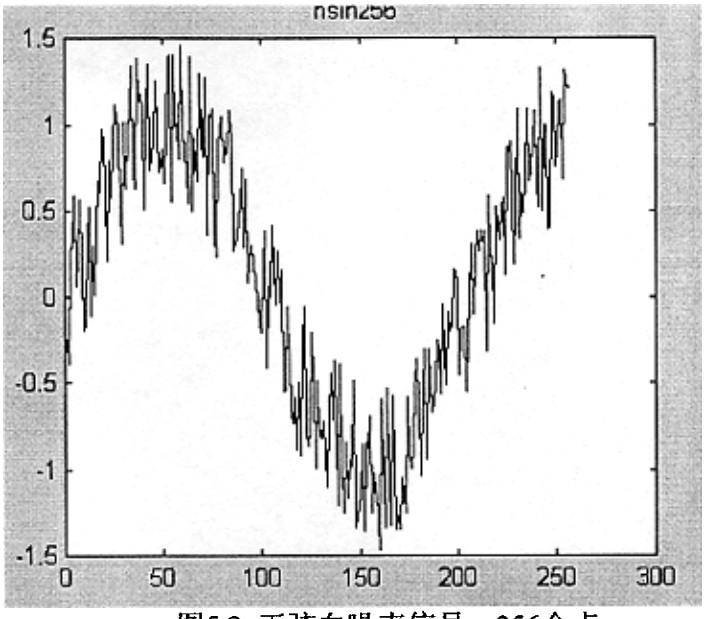


图5.2 正弦白噪声信号 256个点

进行 Haar 小波变换后再反变换恢复出的图像如图 5.3 所示。

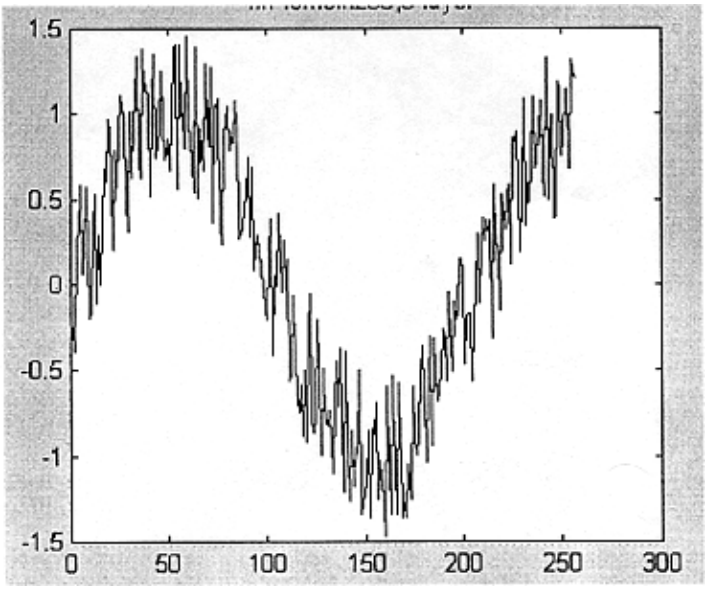


图 5.3 Haar 小波变换恢复出的信号

由图可以看出，恢复出的图像与原图基本上没有差异。

虽然 Haar 小波能很好的恢复出原信号，但是由图 3.5 所示可知，Haar 系的小波函数是不连续的，频域的局部性较差，也即小波变换后的系数相关性很大，在实际应用中很少用，只是因为它结构简单故仅用于理论研究中。

5.1.2 二维 DWT 的 MATLAB 仿真

二维信号（如数字图像）需要用二维小波处理。二维 DWT 在图像的行和列上分析其灰度级，将水平、垂直和对角细节分开来。图 5.4 说明了一个二维 DWT 的分析过程。

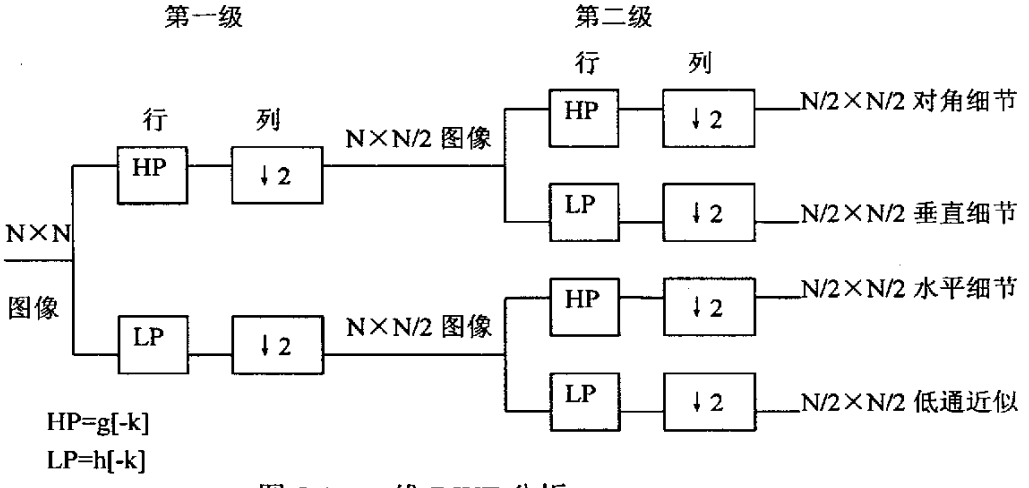


图 5.4 二维 DWT 分析

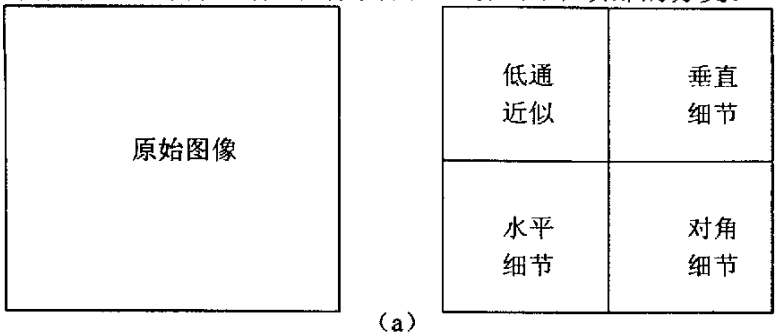
第一步，对 $N\times N$ 图像中的每一行进行低通和高通滤波。因为每一行的图像就是一个一维信号，所以滤波可以采用将其与 $h[-k]$ 和 $g[-k]$ 进行一维卷积的方式来实现。与通常一样，二分之一的向下采样将滤波后的采样点每隔一个去掉一个，就相当于在 $N\times N$ 图像块中每隔一列去掉一列，这样就分别得到一个低通 $N\times\frac{N}{2}$ 的图像和一个高通 $N\times\frac{N}{2}$ 的图像。

第二步，滤波后图像的每一列与 $h[-k]$ 和 $g[-k]$ 分别进行一维卷积。向下采样将两次滤波后的图像的每一列采样点隔一个去掉一个，相当于每隔一行去掉一行。这样图 5.4 所示树的每一个分支都生成了一幅 $\frac{N}{2}\times\frac{N}{2}$ 的图像。

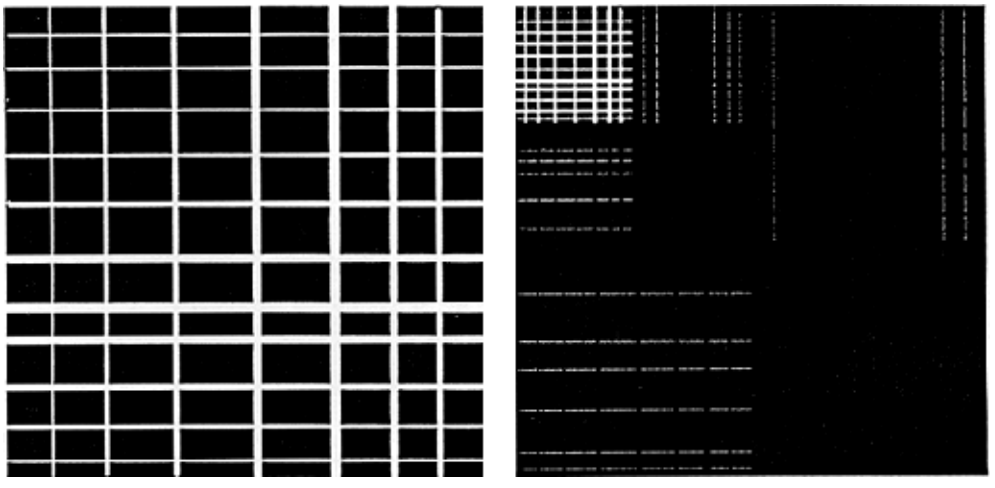
图 5.4 中的每一个分支在 $N\times N$ 图像分析过程中起着不同的作用。开始进行的低通滤波使得每一行的灰度值变得模糊。若其后再进行列的低通滤波，那么所得的结果就是整个图像的低通近似，它在图中是底部的第四个分支。如果行的低通滤波之后跟着列的高通滤波，那么行与行之间的变化，即水平细节就表现出来了，示于图中的第三个分支。

图 5.4 所示树的顶部分支，表示的是原始图像先经过高通滤波而不是低通滤

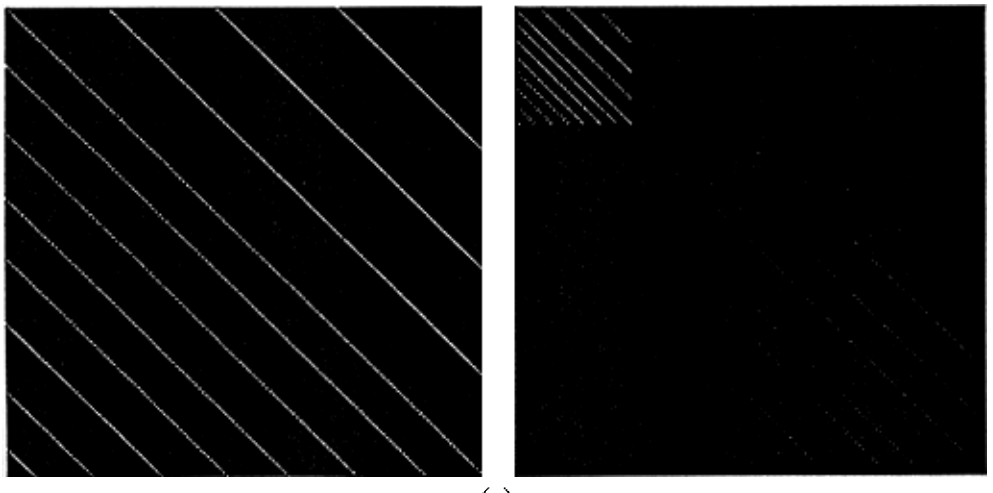
波。开始对原始航的高通滤波突出了任意给定行中相邻像素之间的变化。接下来列的低通滤波模糊了两行之间的变化，最终只是提供了垂直细节，记为图中的第二个分支。当行的高通之后跟着列的高通时，突出了既非水平也非垂直的变化。这种操作顺序给出原始图像的对角细节。对应图中顶部的分支。



(a)



(b)



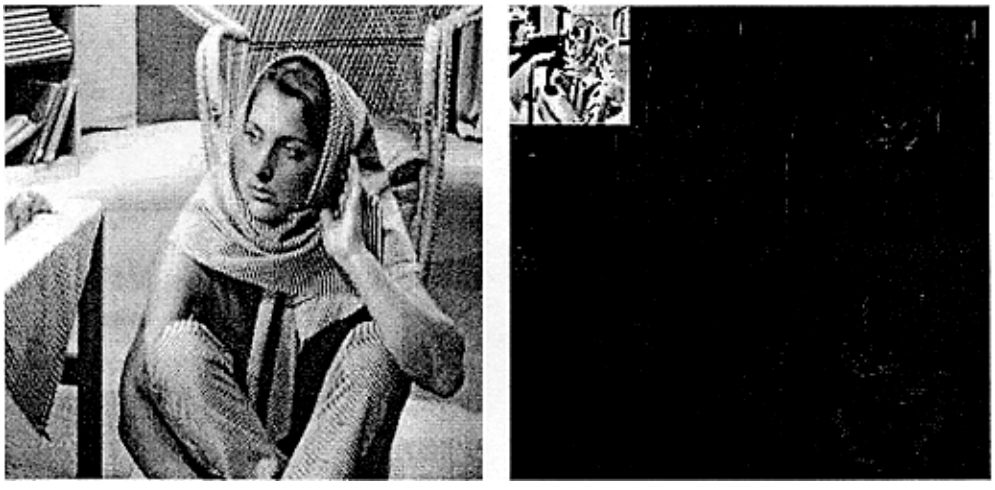
(c)

图 5.5　DWT 图像分析

图 5.5a 显示了低通近似和图像细节的一种方便排列形式。如上所述，每个滤波后图像的行和列都是原图像的一半。图 5.5b,c 是用 CDF9/7 小波族进行二维

DWT 的例子。（b）所示的图像主要是由水平和垂直元素组成的，实际上没有对角分量。而图（c）中的图像包含有很强的对角分量，对水平和垂直方向的细节只表现为很小的响应。滤波树的水平、垂直和对角细节部分，类似于一维情况下的高通系数集$d_j[k]$。低通近似部分类似于一维情况下的低通系数集$c_j[k]$，因此当需要继续分析时，这是被进一步滤波的部分。像一维小波分析那样，每一分析级都从滤波后的图像中提取新的细节层。若对$\frac{N}{2}\times\frac{N}{2}$的低通近似进行与原始图像相同的过程，则会产生 4 个$\frac{N}{4}\times\frac{N}{4}$的子图象：一个低通部分，以及水平、垂直和对角细节。分析可以一直进行下去，直到获得的子图像只包含一个像素为止。

图 5.6 是用 CDF9/7 小波族对一个 256×256 的图像做两级 DWT 分析的情况。在第二级中，第一级得到的低通近似本身被分成 4 个子图像，提取出了细节，留下一个新的低通近似。低通滤波的效果很容易从围巾、桌布的花纹，藤椅的方格，远处罗列的书籍这些细节的丢失看出。第二级低通近似也可以继续变换，以生成 4 幅新的子图像。正如前面所说的，这种变换可以一直进行下去知道子图像只包含一个像素为止。在图（5.5）的例子中，垂直细节对应于分析块的右上角，水平细节对应于左下角，而对角细节对应于右下角，它们都表现出了较强的外观。



a　原始图像　　b　二维 DWT 分析图像

图 5.6　两级二维 DWT 分析

当数字图像需要通过二维 DWT 子图像重建时，就要用向上采样和卷积的办法将细节与低通近似组合起来，如图 5.7 所示。对于二维 DWT 综合来说，向上采样指的时在每一个已存在的行之后加入一个零行（第一级），或是在每一个已存在的列之后加入一个零列（第二级）。在第一级中，向上采样后的子图像的列与脉冲相应 h[k]和 g[k]进行卷积；在第二级中，向上采样后的和图像的行与相同的脉冲相应进行卷积。

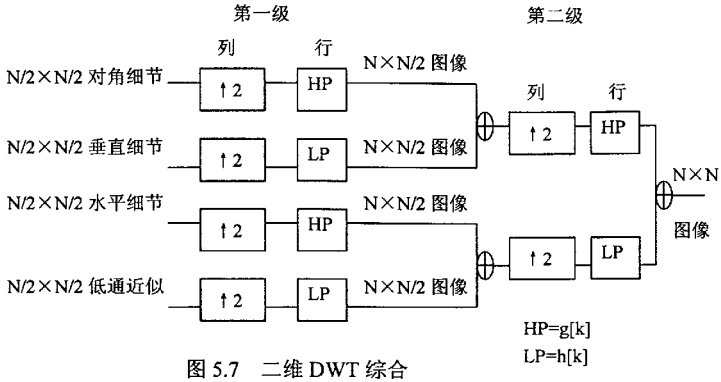


图 5.7　二维 DWT 综合

5.1.3　二维 DWT 分析综合提升算法的 MATLAB 仿真

由上节的二维 DWT 分析综合图和图 3.21 的提升结构可以编出图像小波变换提升实现的 MATLAB 的仿真程序，小波变换取 CDF9/7 双正交小波基，该小波基具有线性相位，因而有着极好的图像压缩性能，是应用 最广泛的小波基之一。

以下是提升实现的二维 DWT 分析的 MATLAB 函数。

其中，x和n是输入参数，c和s是输出参数。x是待处理的二维图像矩阵，n是进行小波变换的分解级数，c是小波变换后的系数，s是每级变换的点数。

c 矩阵中的数据符合下面的数据排列方式：

A(0)	V(0)	… V(n-1)
H(0)	D(0)	
… H(n-1)		… D(n-1)

s 向量中的数据内容为[A(0), Det(0), Det(1), Det(2), …, Det(n-1)]。

```
function [c,s]=mydwtlift(x,n)

%下面是CDF9/7小波族的提升系数：
alf=-1.586134342;
bet=-0.052980118;
gam=0.882911075;
sit=0.443506852;
K=1.230174105;
%下面语句用于获得图像矩阵的行列尺寸和行点数的以二为底的指数。
```

```
[row,col]=size(x);
N=nextpow2(row);
%下面获得分解的最低一级的级数
m=N-n;
if m<0
    m=0;
end
%下面进行二维矩阵的提升小波变换
%变换的大循环
for j=N-1:-1:m
    %对行进行小波提升变换
    for i=1:(col/2)

        t=x(1:col,2*i-1)*alf;
        x(1:col,2*i)=x(1:col,2*i)+t;

        t=x(1:col,2*i)*bet;
        x(1:col,2*i-1)=x(1:col,2*i-1)+t;

        t=x(1:col,2*i-1)*gam;
        x(1:col,2*i)=x(1:col,2*i)+t;

        t=x(1:col,2*i)*sit;
        x(1:col,2*i-1)=x(1:col,2*i-1)+t;

        %低通滤波后的数据
        xx(1:col,i)=x(1:col,2*i-1)*K;
        %高通滤波后的数据
        xx(1:col,i+row/2)=x(1:col,2*i)/K;
    end
    %将低通高通系数合并以供下面的列变换
    x(1:col,1:col)=xx(1:col,1:col);

    %对列进行提升变换
    for i=1:row/2

        t=x(2*i-1,1:row)*alf;
        x(2*i,1:row)=x(2*i,1:row)+t;

        t=x(2*i,1:row)*bet;
        x(2*i-1,1:row)=x(2*i-1,1:row)+t;

        t=x(2*i-1,1:row)*gam;
        x(2*i,1:row)=x(2*i,1:row)+t;
```

```
        t=x(2*i,1:row)*sit;
        x(2*i-1,1:row)=x(2*i-1,1:row)+t;

        xx(i,1:row)=x(2*i-1,1:row)*K;
        xx(i+row/2,1:row)=x(2*i,1:row)/K;
    end

    x(1:row,1:row)=xx(1:row,1:row);

    %获得s向量
    s(j-m+2)=2^j;
    row=row/2;
    col=col/2;

    if j==m
        s(1)=2^j;
    end
end
c=x;
```

以下是提升实现的二维综合DWT的MATLAB仿真函数。综合是分析的逆过程。

```
function [x]=myrwtlift(c,s)


N=nextpow2(s(length(s))*2);   %get the number of the deepest decompse layers
m=nextpow2(s(1));

%lifting wavelet cof
alf=-1.586134342;
bet=-0.052980118;
gam=0.882911075;
sit=0.443506852;
K=.230174105;

for j=m:N-1              %reconstruction cycle

    l=s(j-m+2).*2;   %col and row nember of the mth layer

    for i=1:l/2        %low reconstruction with lifting wavelet
        c(i,1:l)=c(i,1:l)/K;
        c(i+l/2,1:l)=c(i+l/2,1:l)*K;

        t=c(i+l/2,1:l)*sit;
```

```
        c(i,1:l)=c(i,1:l)-t;

        t=c(i,1:l)*gam;
        c(i+l/2,1:l)=c(i+l/2,1:l)-t;

        t=c(i+l/2,1:l)*bet;
        c(i,1:l)=c(i,1:l)-t;

        t=c(i,1:l)*alf;
        c(i+l/2,1:l)=c(i+l/2,1:l)-t;

        cc(2*i-1,1:l)=c(i,1:l);
        cc(2*i,1:l)=c(i+l/2,1:l);
    end

    c(1:l,1:l)=cc(1:l,1:l);

    for i=1:l/2          %row reconstruction with lifting wavelet
        c(1:l,i)=c(1:l,i)/K;
        c(1:l,i+l/2)=c(1:l,i+l/2)*K;

        t=c(1:l,i+l/2)*sit;
        c(1:l,i)=c(1:l,i)-t;

        t=c(1:l,i)*gam;
        c(1:l,i+l/2)=c(1:l,i+l/2)-t;

        t=c(1:l,i+l/2)*bet;
        c(1:l,i)=c(1:l,i)-t;

        t=c(1:l,i)*alf;
        c(1:l,i+l/2)=c(1:l,i+l/2)-t;

        cc(1:l,2*i-1)=c(1:l,i);
        cc(1:l,2*i)=c(1:l,i+l/2);
    end

    c(1:l,1:l)=cc(1:l,1:l);

end
    x=c;
```

5.2 SPIHT 编解码系统的 C++仿真

为了对小波变换和 SPIHT 编码算法有更直观的认识，下面从一幅图像中选取 8×8 尺寸的像素矩阵进行处理，分析每一步处理后数据的变化。

初始图像矩阵如图 5.8 所示。

0	10	20	30	40	50		
146	145	144	144	146	136	130	116
152	146	150	150	148	140	134	123
146	150	154	144	146	144	132	127
150	150	150	144	146	143	134	133
148	148	146	150	144	144	133	133
150	146	150	153	146	136	133	127
150	148	146	141	140	142	137	133
150	146	148	144	140	140	136	137

图 5.8 图像 lady.bmp 中的一部分数据 8×8

由上图可以看出图像幅值分布比较随机，没有什么规律。

对上数据进行离散小波变化后的数据如图 5.9 所示，小波变换的级数是 3 级。

0	10	20	30	40			
52A	0CD	075	04E	035	038	02D	023
0F4	025	074	05C	03B	03D	035	02E
079	074	019	012	037	03C	034	031
075	06D	011	017	036	033	037	031
03C	03E	039	035	008	00B	00A	009
03B	036	037	035	009	00A	009	00B
039	03C	036	031	008	00B	004	006
038	03A	035	034	009	00A	009	00B

图 5.9 三级小波变换后的系数数据

由上图数据可以看出，幅值大的数据主要分布于左上交，这与小波变换后的数据分布特点是一致的，即第一级是低频图像概貌数据，集中了图像大部分的能量，其余各级是分辨率不同的图像的高频细节数据，反映图像水平、垂直、和倾斜方向的纹理信息，故大数据呈线状分布，总能量比较少，便于对数据进行不同分辨质量的压缩。

对上数据进行 SPIHT 编码后的数据见图 5.10

0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
09	F9	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
FF	FF	FF	3F	FE	FF	FF	FF	FF	FF	FF	3F	02	FC	FF	FF
FF	FF	FF	FF	28	3C	00	00	00	C0	FF	BF	0A	02	2E	00
00	C0	FF	7F	97	F9	58	3F	3A	7D	40	0C	F1	1C	89	6E
C5	AA	BF	F7	4E	5F	34	A9	6E	1E	1F	9B	46	77	5D	3A
7D	98	B9	31												

图 5.10 SPIHT 编码数据

从上图数据可以看出，连 1 数据和连 0 数据都较多，可对其作进一步的游程编码。编码后的数据见图 5.11。

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
:	09	01	01	02	93	03	3D	03	01	08	36	03	01	01	01	04
:	04	20	10	01	01	01	01	01	01	05	01	07	03	01	01	18
:	11	01	03	01	01	02	02	02	05	03	02	01	01	01	06	03
:	01	01	03	02	01	01	05	07	01	03	02	04	01	03	04	02
:	03	03	01	02	01	03	01	01	03	01	02	01	01	01	01	03
:	02	01	01	01	01	01	01	01	07	01	04	01	04	01	03	02
:	01	01	05	01	01	03	01	01	02	02	01	02	01	01	01	01
:	01	01	03	01	02	02	04	03	05	03	02	01	02	02	01	01
:	02	03	01	01	03	01	03	01	01	01	03	01	01	02	01	01
:	03	02	01	01	05	04	02	02	02	02	03	01	02	03	02	

图 5.11　游程编码数据

在对上数据作进一步的哈夫曼编码，见图 5.12。

0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
4B	F4	1B	BE	7B	AF	D3	78	9E	6A	3E	0C	CE	63	8E	DE
F6	DD	2F	E7	82	07	18	6D	DB	63	72	4A	C0	D5	FE	7C
8A	CE	1D	27	1E	73	57	5D	0F							

图 5.12　huffman 编码数据

注意上述整个编码都是无失真的，也即通过一系列的解码过程可以完全恢复出原始图像。如果对原始图像进行某一等级分辨率的压缩后可大大降低数据量。

5.3　系统整体性能分析及改进

以下通过对一幅具体图像的处理来显示系统整体性能。所用图像是 258 色（每像素 8 位），512×512 像素的灰度图。小波变换采用基于提升的 CDF9/7 双正交小波基，进行 5 级分解。见图 5.13。

由于编码的渐进传输性，可以从同一编码文件中获得一系列的失真率。失真率用峰值信噪比（PSNR，Peak Signal to Noise Ratio）来衡量：

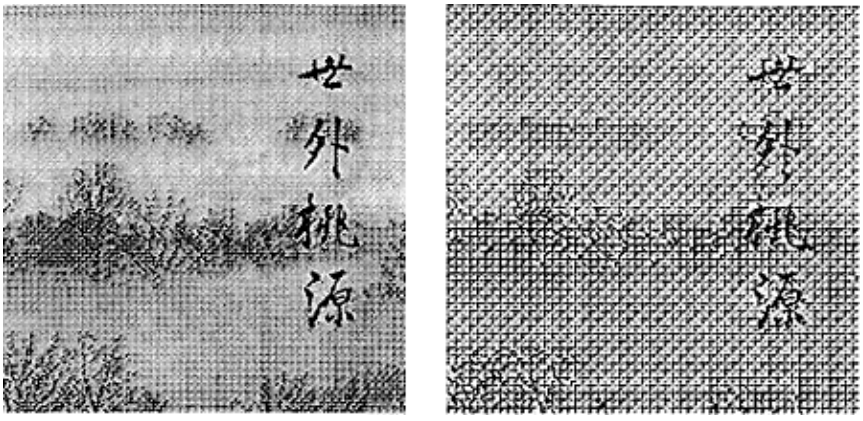
$$PSNR = 10\log_{10}(\frac{255^2}{MSE})\ dB$$

MSE 代表均方误差，定义见（4.4）式。

码率采用 bpp（bit per pixel）来衡量，即用最后压缩的数据量（bit 数）除以图像总的像素数，得到平均每像素的 bit 数。为显示方便，截取了原图中的 220×220 个像素显示如下。

(1) 原图　rate＝8bpp

(2) PSNR=29.2　rate＝2.27bpp

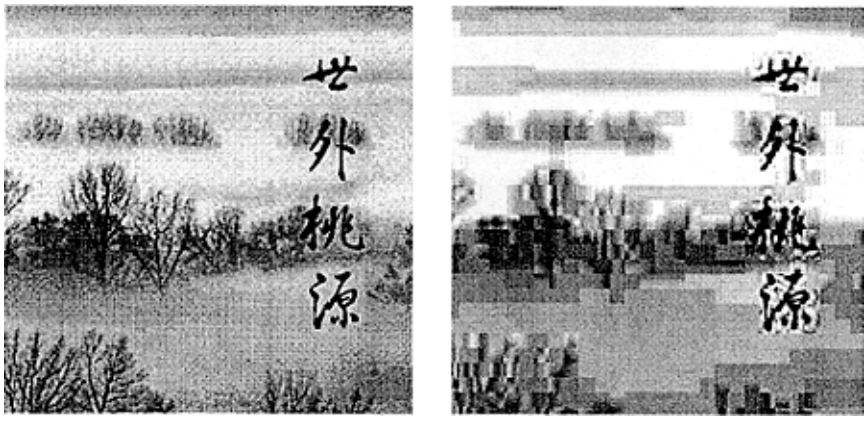


(3) PSNR=13.2　rate=0.483　　(4) PSNR=6.5　rate=0.32

图 5.13　不同 PSNR 的 SPIHT 压缩后的码率

由上面一系列不同PSNR的解码恢复图可以看到，这种压缩方法对高频图像，即变化比较大的条纹信息，压缩效果比较好，特别是文字信息，当 PSNR 降低到 6db 时，文字还是清晰可辨，而图像中仅能依稀可辨树林的轮廓。由图还可看出，PSNR 在 13db 以下的图像质量比较差，而码率并没有降低多少，这是由游程编码和哈夫曼编码的特点决定的。

图 5.14 比较了在相同 PSNR 下本系统和 JPEG 系统对图像处理的结果，由图可以看出当 PSNR 降低到一定限度时，本系统编码的图像中没有像 JPEG 编码中的马赛克效应，这是因为系统采用的 DWT 是针对整块区域的，而 JPEG 中的 DCT 是针对某个 8×8 块的，所以系统可以有效消除 JPEG 不可避免的“马赛克”现象。关于 DCT 和 DWT 的差异，有一个形象的比喻：同是压缩一堵墙，DCT 是将每块砖敲碎了重新组装再放回原处，而 DWT 是将整堵墙敲碎了再重新组织。由此可以看出 DWT 的优势。



(a) PSNR=23.1　rate=1.2bpp　　(b) PSNR=23.1　bpp=0.2

图 5.14 相同 PSNR 的本系统和 JPEG 的比较

由图 5.14 还可看出，虽然在相同 PSNR 下的本系统图像效果要远比 JPEG 压缩的图像好得多，但压缩率却比 JPEG 低得多。下面综合比较一下本系统与 JPEG 的性能。

下表是两种方案的性能比较，其中压缩方式一列中的 JPEG 代表 JPEG 压缩，MYDWT 代表本文提出的压缩方案。

Bpp	压缩方式	PSNR(db)
5.0	JPEG	49.8
	MYDWT	51.0
4.1	JPEG	44.8
	MYDWT	42.5
2.3	JPEG	37.8
	MYDWT	30.3
1.4	JPEG	30.5
	MYDWT	23.4

从上表可以看出，本方案在整体性能上较 JPEG 差，即在相同码率下，本方案的 PSNR 要比 JPEG 的低一些，表现在图像上就是图像会更模糊一些。或者说在相同 PSNR 下，本方案的码率要比 JPEG 的高。为提高性能可考虑采用二进制算术编码来代替游程和哈夫曼编码。

但是本方案也有 JPEG 所不具备的优势，除了消除“马赛克”效应外，采用基于小波变换的图像压缩，在对变换系数进行 SPIHT 编码后，可在该编码文件的任何位置截断，获得不同的码率和 PSNR，进而实现图像逐渐浮现式的显示，非常有利于网络上图像的传输，这样用户就可以在很短的时间内看到图像的概貌，然后决定是否继续下载数据以使图像更加清晰。

需要指出的是该方案虽具备一定的渐进传输的性能，但与 JPEG2000 相比，嵌入式性能还是较弱，JPEG2000 则在小波变换的基础上采用更为复杂精细的小波块分割算法，即优化截取的嵌入式块编码（EBCOT，Ebedded Block Coding with Optimized Truncation）算法，从而实现了丰富的功能，比如基于感兴趣区域编码（ROI，Region Of Interested），即对一幅图像中我们感兴趣的部分采用低压缩比以获取较好的图像效果，而对其他部分采用高压缩比以节省存储空间，这样就可以通过点击 ROI 部分以获得更高的分辨率，看到图像的细节部分。限于篇幅本论文不再对它继续作深入介绍了。有兴趣的同学可以接着往下做。

5.4 系统的 DSP 实现

5.4.1 DSP 介绍

DSP 即数字信号处理器，是专门用于完成各种实时数字信号处理的器件。目前美国 TI 公司的 DSP 产量最大，占世界 DSP 总产量的 60%。TMS320 系列由定点型、浮点型和多处理器型组成。TMS320 的结构是为实现信号实时处理专门设计的。其中的 TMS32054X 系列 DSP 特有的指令可以高效执行编写的通信软件，不尽功耗低，而且在内核电路中，已经将逻辑电路、模拟组件、大容量存储器以及 RISC 等集成化。这里主要介绍 TMS320C5402。

主要特点：

5.4.1.1 硬件结构

TMS320C5402 处理器硬件资源丰富，主要由三大部分组成：CPU、外围设备和存储器。

CPU：采用哈佛结构，程序和数据分开存储，先进的多总线结构，有 1 个程序总线，3 个数据总线，4 个地址总线，取指和读数可以同时进行，从而提高速度；40 位的算术逻辑单元（ALU），包括一个 40 位的桶形移位器和 2 个独立的 40 位的累加器；一个独立的 17×17 位并行硬件乘法器，乘法指令在单周期内完成，优化卷积、数字滤波、FFT、相关、矩阵运算等算法中的大量重复乘法，还可耦合到一个专用的加法器，作非流水的单周期乘法/累加运算（MAC）；比较、选择、存储单元（CSSU）,供 Viterbi 运算中作加法/比较选择运算；作为指数编码器，在单周期内计算一个 40 位累加器值的指数；有两个地址发生器，包括 8 个辅助寄存器和两个辅助寄存器算术单元。

存储器：5402 的地址总线为 20 位，最大外部程序空间可扩展到 1M，片内有 16×16 位的 DARAM 和 4K×16 位的 ROM。片内数据 RAM 包括 2 个 8K 的块，这些块交织在一起，使得 CPU 可同时访问数据 RAM 的两个不同块而不会发生冲突。

片内外设：软件可编程等待状态发生器；软件可编程存储器切换；片内 PLL 时钟发生器、片内振荡器或片外时钟；外部总线断开控制，以禁止外部数据总线、地址总线和控制信号；数据总线可以挂起；可编程定时器。

外围设备：包括 DMA 控制器、主机接口（HPI）、中断选择器、串行口、电源、JTAG 仿真口等。

5.4.1.2 软件资源

指令集：TMS32054X 系列的汇编语言指令特别适合于数字信号处理，大部分指令是单周期的。指令集共有 50 多条，可分为 3 类：数据传输类、算术逻辑操作类和程序控制类。数据传输类指令进行数据的寄存器和存储器间、寄存器之间的数据存取操作，在语句中可显示地标志进行 8 位、16 位、32 位地数据存取。算术逻辑操作类指令进行定点的算术和逻辑操作。程序控制类指令控制程序地流程，包括标准跳转指令、条件跳转指令和空闲指令等。定点 DSP 指令集的设计目标是使处理器能够在每个指令周期内完成多个操作，从而提高每个指令周期地计算效率，进而间存贮 DSP 程序地存储器空间见到最小，以节约成本。为此，DSP 处理器地指令集通常都允许程序员在一个指令内说明若干个并行的操作。

流水线操作：流水线操作和长指令字是 TMS320C54X 地高性能特点。其指令执行可分为六个步骤：预取指、取指、解码、访问、读、执行。流水线操作即指以上六个步骤地并行操作。在 n 周期，流水线完全重叠，六个步骤同时在进行。六个步骤的优先级从高到低依次降低。当一条指令的处理已经准备到下一级，但该级尚未准备好接受新的输入时，就产生了所谓地流水线冲突。在这种情况下，优先级高的先操作。流水线冲突大致有三种原因：跳转冲突、寄存器冲突和存储器冲突。以上冲突可以采取一定手段解决，例如通过调整数据结构和数据放置的地址解决存储器冲突。

采用了流水线技术后，在软件设计中为充分利用执行跳转指令时流水线中尚存的资源，可考虑适当改变语句的位置，以达到优化的目的。

寻址方式：TMS320C54X 支持多种寻址方式，如寄存器寻址、直接寻址、间

接寻址、短立即数寻址、长立即数寻址和相对寻址。此外还提供循环寻址，主要用于相关和卷积运算中的存储器寻址。

速度：单周期为 10ns 执行时间，定点指令为 100MIPS。

5.4.2 系统的 DSP 实现

由于 DSP 内部资源所限，比如 5402 内部仅有 16K 的 DARAM，而一幅 256×256，8 位灰度的图像数据量就达 64K，DSP 根本无法一次处理这么多的数据，所以需要将图像进行分割，本实验采用将一幅 512×512 的 8 位灰度图分割为 64 幅 64×64 的小图像，依次送入 DSP 内进行小波变换，再将变换后的数据输出到外部磁盘上。

由于基于提升的小波变换主要用到 6 个无理数对图像数据反复进行加减乘的运算，这就涉及到小数的定点表示及其四则运算问题。该问题决定了基于提升的小波变换算法 DSP 实现的精度和速度，是需要解决的主要问题。下面介绍一下小数的定点表示方法。

5.4.2.1 数的定标

DSP 中表示的数据都是整数，如果想表示小数，就需要人为确定一个小数点的位置，即数的定标。通过设定小数点在 16 位数中的不同位置，就可以表示不同大小和不同精度的小数。数的定标有 Q 表示法和 S 表示法两种。下表列出了一个 16 位数的 Q 表示、S 表示及它们所能表示的十进制数值的范围。

Q 表示	S 表示	十进制数表示范围
Q15	S0.15	-1≤X≤0.9999695
Q14	S1.14	-2≤X≤1.9999390
Q13	S2.13	-4≤X≤3.9998779
……	……	……
Q1	S14.1	-16384≤X≤16383.5
Q0	S15.0	-32768≤X≤32767

从表中可以看出，同样一个 16 位数，若小数点设定的位置不同，它所表示的数也就不同。例如，

十六进制数 0X2000H＝8192，用 Q0 表示

十六进制数 0X2000H＝0.25，用 Q15 表示

当对 DSP 芯片来说，处理方法是完全相同的。

从表中还可看出，不同 Q 所表示的数不仅范围不同，而且精度也不同。Q 越大，数值范围越小，精度越高。例如，Q0 的数值范围是-32768～+32767，精度位 1，而 Q15 的数值范围为-1～0.9999695，精度为 1/32768＝0.00003051。因此对定点数而言，数值范围与精度是一对矛盾，一个变量要想能够表示比较大的范围，就必须以牺牲精度为代价；要想提高精度，则数的表示范围就相应地减小。在实际的定点算法中，为了达到最佳性能，必须充分考虑到这一点。

浮点数与定点数的转换关系可表示为

浮点数(X)转换为定点数(Xq)：$Xq=(int)X*2^{Q}$

定点数(Xq)转换为浮点数(X)：$X=(float)Xq*2^{-Q}$

5.4.2.2 溢出及处理

由于定点数的表示范围是一定的，因此在进行定点数的加法或减法运算时，其结果就有可能出现超过数值表示范围的情况，即出现了溢出。比如两个 int 型变量(在DSP中用一个字来表示)32767和2相加，0x7fff+0x0002=0x8001即-32767

为了避免这种情况发生，一般在 DSP 芯片中可以设置溢出保护功能，这样当溢出时 DSP 芯片会自动将结果设置为最大值或最小值。比如上结果为 0x7fff，避免了从 32769 到－32767 的灾难性后果。

5.4.2.3 定点运算的 DSP 实现方法

在编写 DSP 模拟算法时，一般都是采用高级语言如 C 语言来编写模拟程序。程序中所用的变量一般既有整型数，又有浮点数。由于 DSP 集成开发环境 CCS 支持 C 语言，如果要用定点 DSP 实现，需将高级语言的浮点算法改写为高级语言的定点算法。

将浮点加/减法转化为定点加/减法时最重要的一点就是必须保证两个操作数的定标值一样。若两者不一样，则在做加/减法运算前先进行小数点的调整。为保证运算精度，需使 Q 值小的数调整为与另一个 Q 值一样大。此外，在做加/减法运算时,必须注意结果可能会超过 16 位表示。如果加/减法的结果超出 16 位的表示范围，则必须保留 32 位结果，以保持运算精度。而乘的过程中，可不考虑溢出而只需调整运算中的小数点。一般地，若一个数的整数位为 i 位，小数位为 j 位，另一个数的整数位为 m 位，小数位为 n 位，则这两个数的乘积为(i+m-1)位整数位和(j+n)位小数位。整数位的 i+m 位中有两个符号位所以应减去一位。

5.4.2.4 程序的段组织

TI 公司的汇编器和链接器所创建的目标文件采用公共目标文件格式(Common Object File Format，简称 COFF)。采用这种目标文件格式更利于模块化编程，并且为管理代码段和目标系统存储器提供更强有力和更加灵活的方法。满足 COFF 文件格式的汇编语言或高级语言的程序是基于代码块和数据块的概念来组织的，而不是一条条命令或一个个数据，这使得程序的可读性和可移植性大大增强。这种块被称为 section，汇编器和链接器都提供了有关的命令来创建块和对块进行处理。

块是目标文件中的最小单位，一个块就是最终在存储器映像中占据连续空间的一块代码或数据。目标文件中的每一个块都是相互独立的。一般地，COFF 目标文件中所有的块可以分为两大类，即已初始化块和未初始化块。

1、已初始化块及其用法

- .text

通常包含可执行代码的块；

- .data

通常包含已初始化的数据的块；

上面两个是系统默认的三个块中的两个，建立的块的块名就是“.text”和“.data”。

- .sect “块名”
- .asect “块名”，地址

这两个是用户自定义的块。建立包含代码或已初始化的数据的块，用法类似

于.text 块和.data 块。.sect 块是可重定位的，而.asect 块命令则建立具有绝对地址的块，一般不提倡用。

当汇编器遇到上述命令时，它立即停止汇编至当前块中，且开始将随后的代码或数据汇编至相应的块中。例如，当汇编器首次遇到一个.data 命令，.data 块被建立，后面的语句被汇编到.data 块中，直到遇到其他三个命令中的一个时。如果汇编器在后面又遇到.data 命令，则将这些.data 后面的语句加到已在.data 块中语句的后面，以建立唯一的.data 块可以在存储器中分配一个连续的空间。

2、未初始化块

- .bss　　符号，字数

通常为未初始化的数据保留空间，这是系统默认的三个块中的一个。

- 符号　.usect　“块名”，字数

用户自定义块，用法类似于.bss，在 RAM 中为未初始化变量分配空间。

上两个命令中符号指向由.bss 或.usect 命令保留的第一个字，它对应于为变量保留空间的变量名，可以在其他任何块中被访问，也可用.global 命令定义为全局符号。字数表示空间的大小，而块名则是程序员自己定义的名字。汇编器遇到.bss 和.usect 命令并不结束当前块开始一个新块，它们只是暂时离开当前块。该命令可以出现在一个已初始化块中的任何地方，而不影响已初始化块的内容。

这两个块在目标文件中没有实际内容，只是保留空间而已。程序可以在运行时利用这些空间来建立和存储变量。

当然，如果在程序中未用任何命令来指示，则汇编器将把所有的程序块或数据块统一汇编至.text 块中。

汇编器对块处理完以后交给链接器处理，完成两个功能。首先，它将 COFF 目标文件中的块用来建立程序块或数据快，将输入块组合起来，以建立可执行的 COFF 输出模块。其次，链接器为输出块选择存储器地址。链接器提供两个命令来完成上述功能：MEMORY 和 SECTIONS。MEMORY 命令定义目标系统的存储器，程序员可定义每一块存储器，指定起始地址和长度，SECTIONS 告诉链接器如何组合输入块以及在存储器何处存放输出块，在链接器命令文件（扩展名为.cmd）中确定。

汇编器对每块汇编时都假定其起始地址为 0，每块中所有的重定位符号（标号）都是相对于 0 地址而言的。实际上，并不是所有块在存储器中都是以 0 地址起始的，因此链接器必须对每块进行重定位。

有时某块程序装入某块存储器中但需在另一块存储器中运行，例如一段关键代码装在 ROM 中但需在更高速的 RAM 中运行就是如此。此时需进行运行时重定位，可在 SECTIONS 中将块分配两次：一次设定装入地址，另一次设定运行地址。例如：.text: load=ROM, run=RAM0

另外，DSP 也支持 C 语言的开发。C 编译器对 C 语言程序编译后生成 6 个可以进行重定位的代码和数据块。这些块类似于汇编中的块，也可以用不同的方式分配至存储器以符合不同系统配置的需要，也分为初始化块和未初始化块。

已初始化块：

- .text 块
- .cinit 块：包含初始化变量和常数表。
- .const 块：字符串和 switch 表。在大存储器模式下，常数表也包含在该

块内。

未初始化块：

- .bss 块：保留全局和静态变量空间。
- .stack 块：为系统堆栈分配存储器。用于将变量传至函数以及分配局部变量。
- .sysmem 块：为动态存储器函数 malloc、calloc、realloc 分配存储器空间。

一般.text、.cinit 和.const 块连同汇编语言中的.data 块，可链入到系统的 ROM 或 RAM，而.bss、.stack 和.sysmem 块则应链入 RAM 中。

编译器支持两种存储器模式：小存储器模式和大存储器模式。小模式是默认的。改模式要求程序中所有静态和全局数据必须小于 64K 字，并且.bss 块不能跨越任何的 64K 字地址边界。大模式则无此限制。

在 C 程序运行之前首先必须建立 C 语言运行环境，这项任务由 C 初始化程序完成的，该引导程序是一个名为 c_int00 的函数。它完成以下工作：

（1） 为系统堆栈定义一个名为.stack 的块，并建立初始化堆栈和帧指针。

（2） 将.cinit 块中的数据表拷贝到.bss 块，对全局和静态变量初始化。

（3） 对小模式而言，设置页指针 DP，指向.bss 块中的全局存储器页。

（4） 调用 main 函数，开始运行 C 程序。

由上所分析，可以编写程序的 cmd 文件如下：

```
/*wavelet.cmd*/
-c                      /*ROM 初始化*/
-o  wavelet.out         /*输出文件名为 wavelet.out*/
-m  wavelet.map         /*产生映像文件 wavelet.map*/
wavelet.obj             /*C 目标文件*/
-l  rts.lib             /*链入 C 语言运行支持库*/

MEMORY
{
    RAM:        origin = 0x80,      len = 0x2780
    ROM:        origin = 0x2800,    len = 0x1800
}

SECTIONS
{
    .text:      {} > ROM
    .cinit:     {} > ROM
    .const:     {} > ROM
    .data       {} > ROM
    .bss:       {} > RAM
    .stack:     {} > RAM
    .sysmem:    {} > RAM
}
```

第六章　总结及展望

本论文研究了一种新技术——小波变换在图像压缩领域里的应用。小波变换不同于传统的域变换压缩方式，它对图像整体进行变换，获得一系列不同分辨率的图像概貌信息和细节信息，从而为进一步处理提供很大的余地，本文介绍了一种基于嵌入式零树思想的集分割算法，该算法性能高，计算量小，并且编解码采用同一套算法，大大降低了解码算法的复杂性，是一种很有前途的图像编解码技术。

该技术还可以实现许多新功能，比如实现数据的渐进传输，从而达到图像逐渐凸现式的显示效果；可以对一幅图像中我们感兴趣的部分采用低压缩比以获取较好的图像效果，而对其他部分采用高压缩比以节省存储空间；码流的随即访问特性允许在编码后数据的任何地方进行截断，以获得所需的码率和不同的分辨率等等。这些功能有待于进一步的研究实现。

在系统的 DSP 实现方面，由于时间所限，仅用 DSP 集成开发环境 CCS 的模拟仿真器验证了基于提升的离散小波变换的算法，证明该算法对图像进行小波变换是有效的。由于模拟仿真过程中要频繁的与磁盘进行数据交换，故对一幅图像进行变换需要较长的时间，在赛扬 266 的 CPU，128M 内存的电脑上大约要耗时几百毫秒左右，该时间是没有参考价值的。

此外由于 TMS320C5402 资源是有限的，内部仅有 16K 的 DARAM，CPU 运算速度也不高，下一步工作可考虑用 TMS320C6000 系列的 DSP，搭建硬件平台，扩展外部程序空间存储小波变换和 SPIHT 编解码程序，进行实时硬件仿真。

参考文献

1. JPEG2000 part I：final draft international standard [S] ISO/IEC JTC1/SC29/WG1 N1890-2000
2. An embedded hierarchical image coder using zerotrees of wavelet coefficients [J] J.M.Shapiro. IEEE Trans. On signal Processing 1993.41:3445-3462
3. A new fast and efficient image codec based on set partitioning in hierarchical trees [J] A.Said W.A.Pearlman IEEE Trans.On Circuits & system for video Tech 1996.6
4. 基于内嵌小波变换的遥感图像编码 李云松 吴成柯 著 电子学报第 28 卷第 10 期 2000
5. JPEG2000 中 9/7 离散小波变换二进制系数实现 刘在德 郑南宁 著 西安交通大学学报第 37 卷第 12 期
6. MATLAB 6.0 与科学计算 王沫然 编著 电子工业出版社 2001
7. MATLAB 6.X 图形编程与图像处理 陈杨 陈荣娟 编著 西安电子科技大学出版社 2002
8. MATLAB 6.5 辅助小波分析与应用 飞思科技产品研发中心 编著 2003
9. 离散信号的滤波 王欣 王德隽 著 电子工业出版社 2002
10. 数据结构与算法分析 Clifford A. Shaffer 著 张铭 刘晓丹 译 电子工业出版社 1998
11. 算法设计与分析 王晓东 编著 清华大学出版社 2003
12. 分形与小波 李水根 吴纪桃 编著 科学出版社 2002
13. 数字信号处理基础（加）Joyce Van de Vegte 著 侯正信 王国安 译 电子工业出版社 2003
14. DSP 芯片的原理与开发应用（第三版） 张雄伟 陈亮 编著 电子工业出版社 2003
15. C++程序设计教程 郑莉 刘慧宁 编著 机械工业出版社 2001
16. Visual C++6.0 开发技巧与实例教程 同志工作室 编著 人民邮电出版社 2000

致　谢

本论文在导师张立民副教授的耐心指导和帮助下完成的。张老师对我的学业严格要求，务使精益求精。每当我遇到困难时，张老师又总是耐心的鼓励启发我想方设法排除困难，在此谨向张老师几年来对我的辛勤培养、教导和关心致以最诚挚的感谢和敬意！

在读研期间，我还得到了杨文霞老师、李维详老师、孙桂琳老师和赵腊月老师的热心的指导和帮助，在此也表示衷心的感谢！

同时感谢 EDA 实验室曹慧丽老师和司敏山老师的在实验条件上的大力支持。感谢李章林、王新和林锐同学在我论文完成期间给予的帮助。

借此机会也向我的父母表示衷心的感谢，没有他们几十年如一日对我学业的督促与支持就没有我现在的成绩。

最后，再次感谢所有的老师和同学，祝他们生活幸福，工作顺利，事业有成！