

# Experiment—Image Compress

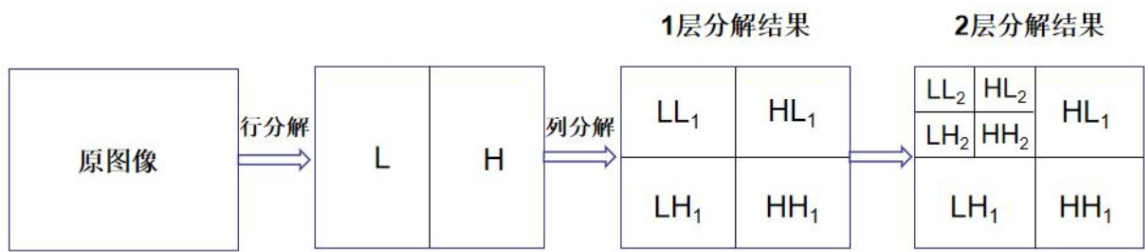
PB18010496 杨乐园

## Introduction

文字、图形、视频都可以存储为图像信息，但计算机处理这些多媒体信息需要大量的存储空间，在网络多媒体技术的应用中，为了兼顾图像的质量和处理的速度的，高保真、大压缩比的图像压缩技术是必要的。

长期以来，图像压缩编码利用离散余弦变换（DCT）作为主要的变换技术，并成功的应用于各种标准，比如JPEG、MPEG-1、MPEG-2。但是，在基于DCT图像变换编码中，人们将图像分为88像素或者1616像素的块来处理，从而容易出现方块效应与蚊式噪声。相比较而言，小波变换则是全局变换，在时域和频域都有良好的局部优化性能。小波变换将图像的像素解相关的变换系数进行编码，比经典编码的效率更高，而且几乎没有失真，在应用中易于考虑人类的视觉特性。目前，基于小波变换的图像压缩方法已经逐步取代DCT或者其他子带编码技术，从而成为新的图像压缩的主要技术之一。小波变换在信号的高频部分可以取得较好的时间分辨率；在信号的低频部分，可以取得较好的频率分辨率，从而能有效地从信号（如语音、图像等）中提取信息，达到数据压缩的目的。

与一维信号不同的是，图像是二维信号。对一张像素为 $m \times n$ 的图像上每点 $(x, y)$ 都有唯一的灰度值 $f(x, y)$ ,  $x = 1, \dots, m, y = 1, \dots, n$ 与之对应，从而对图像的分解一般使用张量积小波，可以看做是一元小波分别对图像的行与列进行分解。如下图所示：



首先将原图像的每一行分解为低频部分 $L$ 与高频部分 $H$ ，再对每一列进行分解，得到的1层分解将图像分为4个部分： $LL_1$ 是平滑逼近， $LH_1$ 是垂直分量， $HL_1$ 是水平分量， $HH_1$ 是对角分量；2层分解是作用在 $LL_1$ 上，又得到4个分量；多层分解以此类推。从而对彩色( $RGB$ )图像只需对三个通道分别进行分解即可。

而图像的压缩可以理解对三组细节系数的阈值处理。一般来讲，为了提高压缩性能，需要在分三个方向做阈值处理。这种思想类似于将图像的多余细节视为噪声，因此其本质就是图像的降噪，但在实际应用中又是简单有效的。对一张原始图像 $X$ ，假设其压缩图像为 $\widetilde{X}$ ，通常可以用分解系数中置为0的系数百分比来模拟压缩比，用保留能量百分比（即 $\|X\|_{L_2} / \|\widetilde{X}\|_{L_2}$ ）来模拟保真性能。

从而基于小波变换的图像压缩基本步骤如下：

- 1). 用小波对图像图层分解并提取分解结构中的低频与高频系数。
- 2). 各频率成分重构。
- 3). 对第一层低频信息压缩。
- 4). 对第二层低频信息压缩。

## Threshold selection

**首先，为什么要使用阈值？**这是由于信号在空间上(或者时间域)是有一定连续性的，因此在小波域，有效信号所产生的小波系数其模值往往较大；而高斯白噪声在空间上(或者时间域)是没有连续性的，因此噪声经过小波变换，在小波域仍然表现为很强的随机性，通常仍认为是高斯白噪声的。那么就得到这样一个结论：在小波域，有效信号对应的系数很大，而噪声对应的系数很小。刚刚已经说了，噪声在小波域对应的系数仍满足高斯白噪声分布。如果在小波域，噪声的小波系数对应的方差为 $\sigma$ ，那么根据高斯分布的特性，绝大部分(99.99%)噪声系数都位于 $[-3\sigma, 3\sigma]$ 区间内(切比雪夫不等式， $3\sigma$ 准则)。因此，只要将区间 $[-3\sigma, 3\sigma]$ 内的系数置零(这就是常用的硬阈值函数的作用)，就能最大程度抑制噪声的，同时只是稍微损伤有效信号。将经过阈值处理后的小波系数重构，就可以得到去噪后的信号。

常用的软阈值函数，是为了解决硬阈值函数“一刀切”导致的影响(即：模小于 $3\sigma$ 的小波系数全部切除，大于 $3\sigma$ 全部保留，这势必会在小波域产生突变，导致去噪后结果产生局部的抖动，类似于Fourier变换中频域的阶跃会在时域产生拖尾)。软阈值函数将模小于 $3\sigma$ 的小波系数全部置零，而将模大于 $3\sigma$ 的做一个比较特殊的处理：即大于 $3\sigma$ 的小波系数统一减去 $3\sigma$ ，小于 $-3\sigma$ 的小波系数统一加 $3\sigma$ 。经过软阈值函数的作用，小波系数在小波域就比较光滑了，因此用软阈值去噪得到的图象看起来很平滑。

**其次，比较硬阈值函数去噪和软阈值函数去噪区别。**硬阈值函数去噪所得到的峰值信噪比( $PSNR$ )较高，但是有局部抖动的现象；软阈值函数去噪所得到的 $PSNR$ 不如硬阈值函数去噪，但是结果看起来很平滑，原因就是软阈值函数对小波系数进行了较大的“改造”，小波系数改变很大。因此各种各样的阈值函数就出现了，其目的应该就是要使大的系数保留，小的系数被剔除，而且在小波域系数过渡要平滑。

如何估计小波域噪声方差 $\sigma$ ，这个只需：把信号做小波变换，在每一个子带利用Robust Estimator估计就可以(可能高频带和低频带的方差不同)。Robust Estimator估计就是将子带内的小波系数模按大小排列，然后取最中间那个，并除以0.6745就得到噪声在某个子带内的方差 $\sigma$ 。利用这个 $\sigma$ ，然后选种阈值函数，就可以去去噪了，在Matlab有实现api可使用。

**常见的阈值选择方法。**有：固定阈值估计、极值阈值估计、无偏似然估计以及启发式估计等。

1. 无偏风险估计阈值( $rigrsure$ ):

1). 把信号 $s(i)$ 中的每一个元素取绝对值，在从小到大排序，然后将各个元素取平方，从而得到新的信号序列： $f(k) = (sort(|s|))^2, k = 0, 1, \dots, N-1$ 。

2). 若阈值为 $f(k)$ 的第 $k$ 个元素的平方根，即 $\lambda_k = \sqrt{f(k)}, k = 0, 1, \dots, N-1$ ，则该阈值产生的风险即为： $rish(k) = \frac{N-2k + \sum_{i=1}^k f(i) + (N-k)f(N-k)}{N}$ 。

3). 根据所得到的风险曲线 $rish(k)$ ，记起最小风险点所对应的值为 $k_{min}$ ，那么 $rigrsure$ 阈值定义为 $\lambda_k = \sqrt{f(k_{min})}$ 。

2. 固定阈值( $sqtwolog$ ):

$$\lambda = \sqrt{2 \log(N)}.$$

3. 启发式阈值( $heasure$ ):

$$crit = \sqrt{\frac{1}{N} \left( \frac{\ln N}{\ln 2} \right)^2}, \quad eta = \frac{\sum_{i=1}^N |S_i|^2 - N}{N}$$

如果 $eta < crit$ ，则选用 $sqtwolog$ 阈值；否则选取 $sqtwolog$ 阈值和 $rigrsure$ 阈值中的较小者作为本准则选定的阈值。

4. 极值阈值( $minimaxi$ ):

$$\lambda = \begin{cases} 0.3936 + 0.1829 \left( \frac{\ln N}{\ln 2} \right), & N > 32 \\ 0, & N \leq 32 \end{cases}$$

一般来讲，极值阈值估计和无偏似然估计方法比较保守，当噪声在信号的高频段分布较少时，这两种阈值估计方法效果较好可以将微弱的信号提取出来。而固定阈值估计和启发式阈值估计去噪比较彻底，在去噪时显得更为有效，但是也容易把有用的信号误认为噪声去掉。

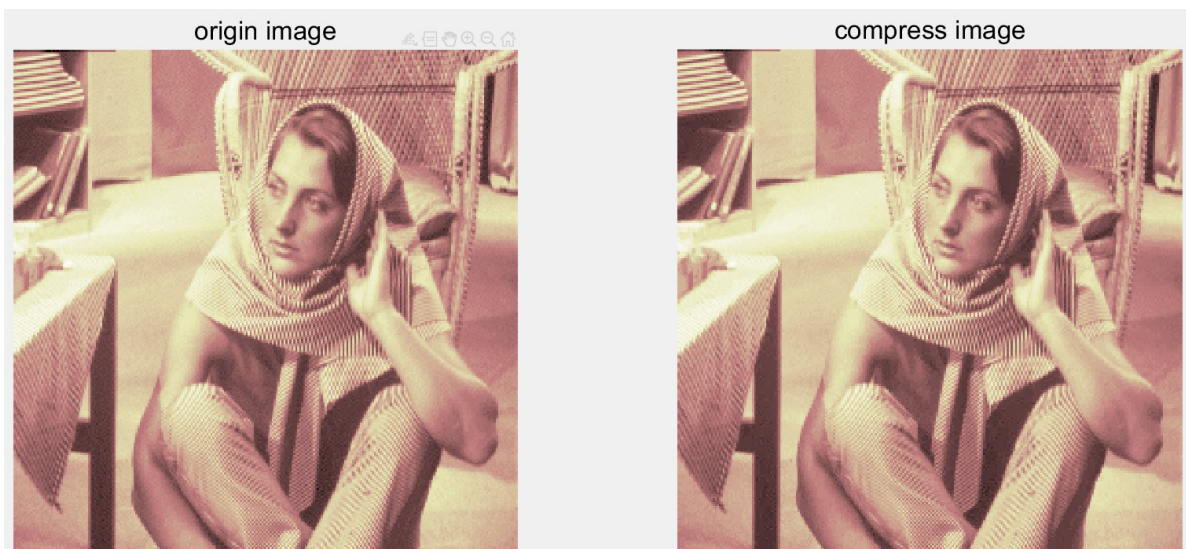
## Example implementation

接下来我们实现书中一些示例，并给出具体的代码解释：

示例9.5，利用Haar小波对图像压缩：

```
load wbarb;whos;
%对图像X用Daubechies-1阶消失矩小波基函数实现2层分解
[C, S] = wavedec2(X,2,'db1');
%利用ddencmp()函数自动生成小波压缩的阈值选取方案，用小波分解
%thr为阈值，sorth为选择阈值的方式（sorth=s为软阈值，=h为硬阈值）
%keepapp=0/1决定是否对近似分量进行阈值处理，=1则低频系数不进行阈值量化处理，反之，则低频系数进行阈值量化
[thr,sorth,keepapp] = ddencmp('cmp','wv',X);
%对每层采取同一个阈值进行处理
%XC是压缩后的信号，[CXC,LXC]是XC的小波分解结构
%PERF0和PERFL2是恢复和压缩L^2的范数百分比，是用百分制表明降噪或压缩所保留的能量成分
[Xcomp, CXC, LXC, PERF0, PERFL2] = wdencmp('gbl',C,S,'db1',2,thr,sorth,keepapp);
colormap(map);
subplot(121);image(X);title('origin image','FontSize',20);
axis square
axis off
subplot(122);image(Xcomp);title('compress image','FontSize',20);
axis square
axis off
```

从而输出结果如下：



```
>> compress1
```

Name	Size	Bytes	Class	Attributes
BPP	1x1	8	double	
C	1x65536	524288	double	
CR	1x1	8	double	
CXC	1x65536	524288	double	
LXC	4x2	64	double	
PERF0	1x1	8	double	
PERFL2	1x1	8	double	
S	4x2	64	double	
X	256x256	524288	double	
Xc	512x512	2097152	double	
Xcomp	256x256	524288	double	
keepapp	1x1	8	double	
map	192x3	4608	double	
sorh	1x1	2	char	
thr	1x1	8	double	

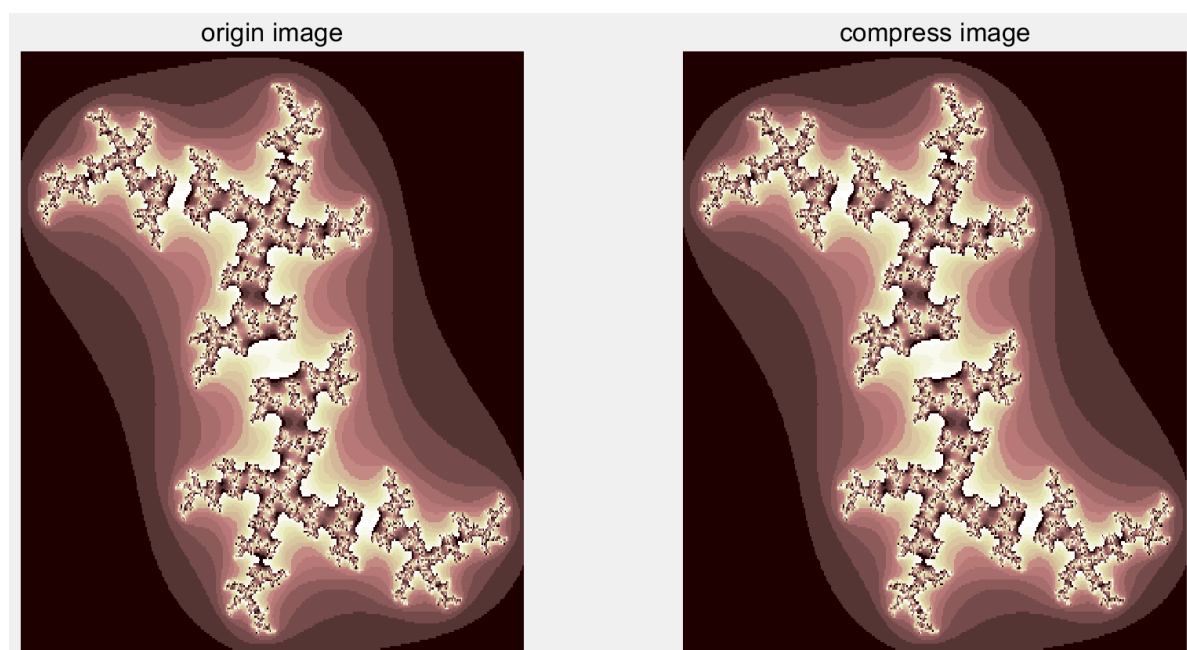
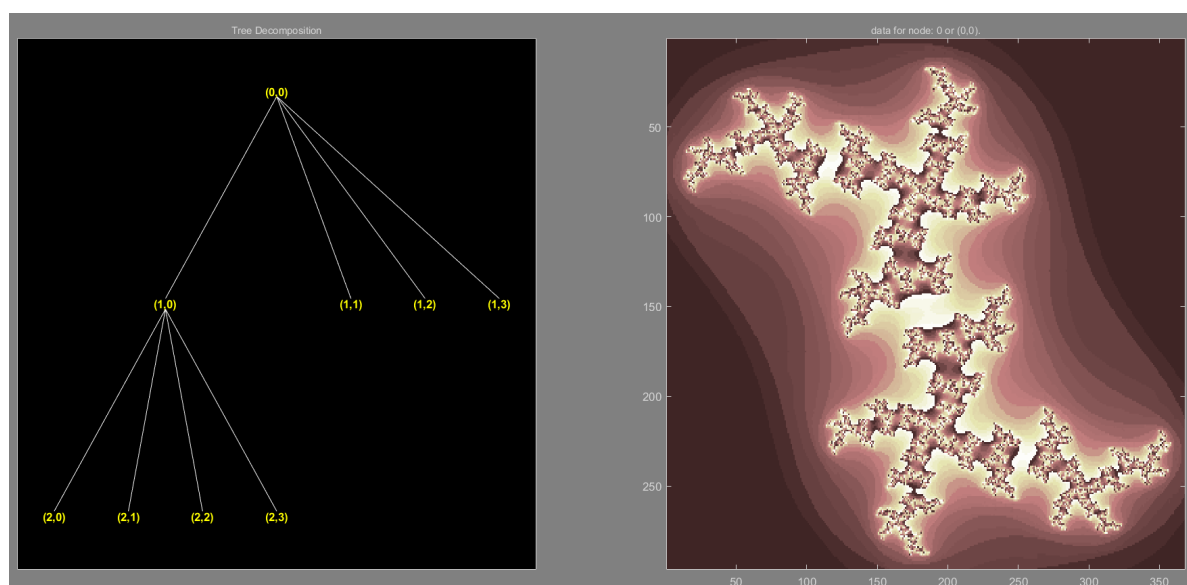
上述示例采取的是全局同一阈值，在大规模图像处理中，全局阈值处理是一个通常的选择，但这种方式并不够精细。对单一图像，分层、分方向处理更能体现图像固有的时频局部特性，但需要先进行分析以获得足够的关联信息。即便如此，小波分解仍不够灵活，分解出来的小波树只有一种模式，不能完全地体现出时频局部化信息。因此，实际的压缩算法多采用小波包算法，而小波树的确定则是根据不同的信息论推测，以达到分解系数表达的信息密度最高。需要说明的一点，对高频成分很多的图像（如指纹图像），小波包的分解细节信息的特点尤其能发挥出优势。正是因为这一点，美国联邦调查局*FBI*的指纹库就是采用基于小波包的压缩算法*WSQ*。实际应用中，为了提高机器实现效率，一般采用特定的双正交小波，利用其滤波器分布规则的特点，用移位操作实现滤波操作。

示例9.6，使用*Daubechies4*小波2层分解，全局默认硬阈值处理压缩图像：

```
clc;clear;close all;
load julia;whos;
%crit为使用小波包进行分解时所选取的熵函数类型
[thr,sorh,keepapp,crit] = ddencmp('cmp','wp',X);
%返回输入X压缩后的XD。输出参数TREED是XD的最佳小波包分解树；
%PERFL2和PERF0是恢复和压缩L2的能量百分比。PERFL2=100*(X的小波包系数范数/X的小波包系数)^2；
%函数使用由字符串CRIT定义的熵和阈值参数thr*2实现最佳分解。
[Xcomp,treed,PERF0,PERFL2] = wpdencmp(X,sorh,2,'db4',crit,thr*2,keepapp);
n=size(map,1);
colormap(pink(n));
subplot(121);image(wcodemat(X,n));title('origin image','FontSize',20);
axis off
subplot(122);
image(wcodemat(Xcomp,n));title('compress image','FontSize',20);
axis off
plot(treed)
```

输出结果如下：左侧为压缩过程中使用的最优小波树，使得零系数百分比最高。





Name	Size	Bytes	Class	Attributes
X	296x368	871424	double	
map	64x3	1536	double	

上述仅考虑图像本身像素点之间的相关性的算法成为一代图像数据压缩法。在多媒体系统的应用领域中，人眼作为图像信息的接收端，其视觉对于边缘急剧变化不敏感（即视觉掩盖效应），以及人眼对于图像的亮度信息敏感，而对颜色分辨弱等因素，使得高压缩比的情况下，解压缩的图像依然有着满意的主观质量。由Kunt等人提出的第二代图像数据压缩算法，就充分考虑了人类视觉生理心理特征，侧重于将原始图像在领域内做多层分解，然后对这些信息表示灵活地有选择地编码，可得到较高的压缩比和很小的失真度。基于图像渐进式编码（嵌入式零树小波编法EZW、多级树集合分裂算法SPIHT、集合分裂嵌入块编码器SPECK）、基于行的熵编码、嵌入式最优截断(EBCOT)编码是目前国际上最为流行的三种基于小波变换的图像编码方法。前面学习的消失矩和信号压缩有着密切的关系。一方面，一个小波的消失矩阶数越高，压缩性能越好；另一方面，消失矩阶数的增长会导致滤波器系数的个数成倍增加，从而影响压缩图像的能量集中性质变差。大量的图像压缩试验证实，Cohen与Daubechies等人于1992年发现的消失矩为4的双正交9-7小波（又称CDF9-7小波）具有最好的信息压缩性质，并在JPEG2000国际静态图像压缩标准中被推荐使用。

示例9.7，对比同样压缩12次情况下，采用不同编码方式和小波变换得到的压缩效果：

```

clc;clear;close all;load porche
colormap(pink(255))
subplot(2,2,1); image(X);
axis square,axis off;
title('origin image','FontSize',20)

```

%返回压缩比CR与Bit-Per-Pixel ratio, 采用EZW方法, Haar小波

```

[CR,BPP] = wcompress('c',X,'mask.wtc','ezw','maxloop',12,'wname','haar');
Xc = wcompress('u','mask.wtc');
subplot(2,2,2);image(Xc);
axis square,axis off;
title(['EZW - haar'],['ratio: ' num2str(CR,'%1.2f %'),'BPP: '
num2str(BPP,'%3.2f')]], 'FontSize',20)

```

%采用EZW方法, bior4.4小波

```

[CR,BPP] = wcompress('c',X,'mask.wtc','ezw','maxloop',12,'wname','bior4.4');
Xc = wcompress('u','mask.wtc');
colormap(pink(255))
subplot(2,2,3); image(Xc);
axis square,axis off;
title(['EZW - Bio4.4'],['ratio: ' num2str(CR,'%1.2f %'),'BPP: '
num2str(BPP,'%3.2f')]], 'FontSize',20)

```

%采用SPIHT方法, bior4.4小波

```

[CR,BPP] = wcompress('c',X,'mask.wtc','spiht','maxloop',12,'wname','bior4.4');
Xc = wcompress('u','mask.wtc');
colormap(pink(255))
subplot(2,2,4); image(Xc);
axis square,axis off;
title(['SPINT - Bio4.4'],['ratio: ' num2str(CR,'%1.2f %'),'BPP: '
num2str(BPP,'%3.2f')]], 'FontSize',20)

```

输出结果如下:

origin image



EZW - Haar

ratio: 13.44 %,BPP: 1.08



EZW - Bio4.4

ratio:11.31 %,BPP: 0.90



SPINT - Bio4.4

ratio: 3.60 %,BPP: 0.29



