第七章 数字水印模型

随着数字技术的发展,Internet 应用日益广泛,利用数字媒体因其数字特征极易被复制、篡改、非法传播以及蓄意攻击,其版权保护已日益引起人们的关注。近年来国际上提出了一种新型的版权保护技术——数字水印(Digital Watermark) 技术。利用人的听觉、视觉系统的特点,在图像、音频、视频中加入一定的信息,使人们很难分辨出加水印后的数字作品与原始数字作品的区别,而通过专门的检验方法又能提取出所加信息,以此证明原创作者对数字媒体的版权。数字水印技术通过将数字、序列号、文字、图像标志等信息嵌入到媒体中,在嵌入过程中对载体进行尽量小的修改,以达到最强的鲁棒性,当嵌入水印后的媒体受到攻击后仍然可以恢复水印或者检测出水印的存在。数字水印技术,是指在数字化的数据内容中嵌入不明显的记号。被嵌入的记号通常是不可见或不可察觉的,但是通过一些计算操作可以被检测或被提取。水印与原数据(如图像、音频、视频数据) 紧密结合并隐藏其中,成为不可分离的一部分。数字水印技术是一种保护数字作品版权的技术。数字水印主要应用领域包括:原始数据的真伪鉴别、数据侦测与跟踪、数字产品版权保护等。数字水印不仅要实现有效的版权保护,而且加入水印后的图像必须与原始图像具有同样的应用价值,也就是说数字产品不会因为加入了水印而变得不可用。因此,数字水印主要有以下特点:

① 不可见性(Invisibility) :加有水印后的图像不能有视觉质量的下降,与原始图像对比,很难发现二者的差别。

② 鲁棒性(Robustness) :加入图像中的水印必须能够承受施加于图像的变换操作(如加入噪声、滤波、有损压缩、重采样、D/A 或 A/D 转换等) ,不会因变换处理而丢失,水印信息经提取后应清晰可辨。

③ 安全性(Security) :数字水印应能抵抗各种蓄意攻击,必须能够惟一地标识原始图像的相关信息,任何第三方都不能伪造他人的水印信息。

图 7.1 给出了数字水印嵌入的一般模型。与信息隐秘一样,我们约定以下几个名词作为今后对水印系统各阶段作品的称谓:将未加水印的作品称为原始作品或原始图像,将随机生成的水印信息称为水印模板,将嵌入水印的作品直接称为加有水印的作品或加有水印的图像。

考虑到数字水印的出现本身也就是为了保护通信安全,水印系统在某种意义上说就是一个特殊的通信系统。本章我们先将结合通信系统的一般结构来诠释数字水印的生成和嵌入策略,并重点以实验实现的方式帮助大家理解数字水印技术。本章

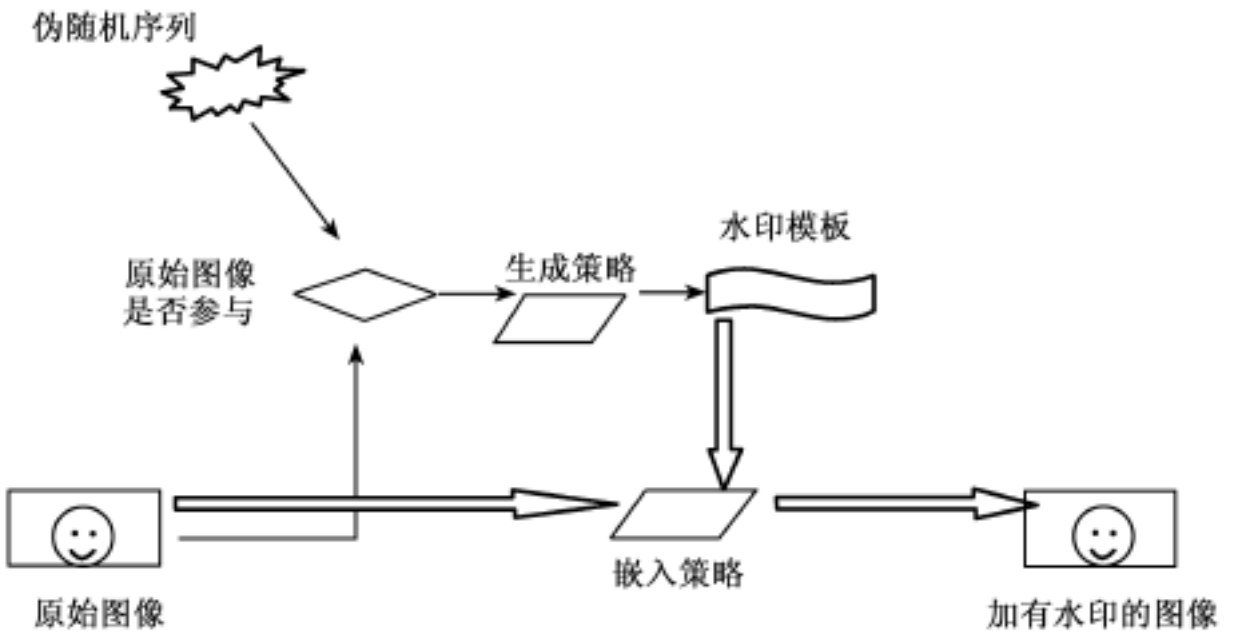


图 7.1　数字水印嵌入模型

所选择的实验有三个，分别代表了数字水印技术的不同种类和发展方向。在本章最后，我们将对水印检测策略作总结性分析。

7.1　水印的通信系统模型

7.1.1　数字通信系统

为了了解水印和传统通信系统的异同，需要简单地回顾一下传统的通信系统。首先介绍的是通信系统的基本构成，我们对此要有所了解，以便进一步把传统模型扩展为水印模型。

数字通信系统可由图 7.2 中的模型加以概括。发送端信息源（也称信源）的作用是把各种消息转换成原始电信号。为了使这个原始信号适合在信道中传输，由发送设备对原始信号完成某种变换，然后再送入信道。信道是指信号传输的通道。在接收端，接收设备的功能与发送设备的相反，它能从接收信号中恢复出相应的原始信号，而受信者（也称信宿）是将复原的原始信号转换成相应的消息。通信系统传输的消息是多种多样的，可以是符号、文字、话音、数据、图像等。

从图 7.2 可以发现在数字信号的实际应用中通常将发送设备分解为三部分：调制器、信源编码器和加密器。调制器把信息映射为取自特定字符集的符号序列。信源编码器把符号序列转化为可以在信道中传输的模拟信号。信号 x 经过编码后开始在信道上传输。信道是有噪声的，因此接收到的信号（通常用 y 表示）往往和发送信号 x 不相同。这应归结为加性噪声作用的结果。可以这样说，信号 x 在信道的传输过程中被添加了一个随机噪声信号 n。在信道接收端，接收到的信号 y 进入解调器、

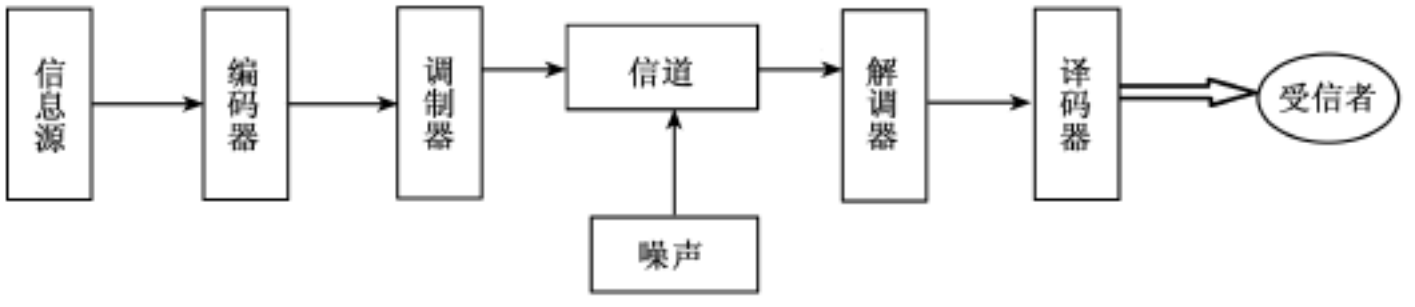


图 7.2　数字通信系统的基本模型

译码器,进行编码过程的逆过程,同时试图纠正传输错误。在这里,我们对通信系统模型加入了密码系统是考虑到安全传输,防止第三方有意或无意的攻击。在信息传输前,密码系统使用密钥对信息或明文进行加密。传输的是将密文编码调制的信号。接收端接收经解调译码得到的密文并使用相同或相关的密钥对其解密得到明文消息。我们用式(7.1)来表示信道的数学模型。

$$y = kx + n \tag{7.1}$$

其中,k 依赖于信道的特性,对 x 是一种干扰,通常称其为乘性干扰,而 n 表示加性干扰。加性干扰的概率分布一般独立于有用信号的概率分布。加性干扰虽然独立于有用信号,但它始终干扰有用信号。

在数字水印系统中,对于原始作品和水印信息有两种不同的认识。当我们关注原始作品时,往往将水印信息作为原始作品信号的加性噪声;而当我们关注水印信息本身在通信系统中的地位时,则往往将原始作品当做纯加性噪声。在今后的讨论中,我们会混合使用上面的两种思想,希望大家不要混淆。

7.1.2　水印系统的基本模型

这里谈到的水印系统分类是根据水印生成和嵌入策略为依据的。这里谈到的水印系统的基本模型是一个盲嵌入(Blind Embedding)水印模型。根据其水印检测器工作方式的不同,我们又将水印系统基本模型分为“含辅助信息检测器水印模型”和“盲检测水印模型”两类。

我们把需要原始作品的检测器称为含辅助信息检测器(informed detection)。它还可以指只需要原始作品的部分信息而不是整个作品信息的检测器。相反地,把不需要任何有关原始作品信息的检测器称为盲检测器(Blind Detection)。在决定是否能用于某个应用时,水印系统是采用盲检测器还是采用含辅助信息检测器十分重要。含辅助信息检测器只能用于那些可以获得原始作品的应用中。

不管采用的是含辅助信息的检测器还是盲检测器,盲嵌入水印(基本水印模型)的嵌入过程都包括两个基本步骤:首先是水印的生成。将信息映射为水印 wa,使它必须和原始载体 Po 类型一致,格式相同。当给图像添加水印时,基于某种水印生成策略产生一个与原始图像大小相同的二维像素模板。接下来是水印嵌入,把 wa 加

到载体作品 Po 上便产生水印作品。在嵌入了水印后，假设水印作品 Pw 还经历了某些处理过程，其结果相当于在作品中加入了噪声。处理作品的方式包括压缩和解压缩，在模拟信道上广播，图像或语音增强等。对方的攻击行为也可以包括在处理过程中。

图 7.3 显示了简单的含辅助信息检测器的水印系统。它是由基本数字通信框架映射而来的。此系统采用了简单的含辅助信息的检测器，也就是说接收方在检测水印时需要原始作品的参与。其检测过程也包含两个步骤：首先，从接收到作品 Pwn 中减去未加水印的载体作品，得到带噪声的水印模型 wn。然后在水印解码器中使用水印密钥对其进行解码。因为嵌入时加入的载体作品在检测时被完全减去，附加模板 wa 和 wn 的惟一差别是由噪声引起的。

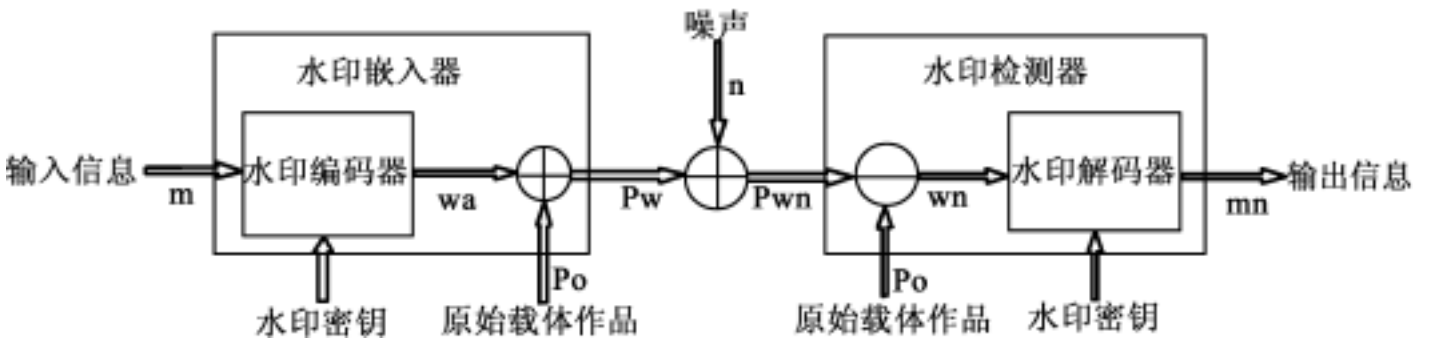


图 7.3　含辅助信息检测器的水印系统

在实际应用中，水印检测时是可以要求原始的未加水印的作品参与的。例如，在拷贝跟踪应用中，为了发现持有非法发行特定副本的人，通常由原始作品的拥有者或可信第三方来进行检测。他们当然拥有不带水印的原始作品版本。这样就能将原始作品和非法副本一起提供给检测器。由于用水印副本减去原始版本就可以得到单独的水印，因此这样经常会显著地提高检测器的性能。原始版本还可以用于同步，以抵消在水印副本中可能有的时间或几何失真。

但在其他一些应用中，检测器必须能够在无法获得原始作品的情况下工作。如一个拷贝控制的应用，就必须为每个顾客的录制设备分配一台检测器。但此时如果使用含辅助信息检测器，就不得不把未加水印的内容发送下去。这不仅不切实际，还会使水印系统的作用失效。

在有些有关数字水印的文献中，将采用含辅助信息检测器的系统通常称为私有水印系统，而那些用盲检测器的系统则称为公有水印系统。一般而言，仅在私有水印应用领域中可获得原始作品，因此含辅助信息检测是不能用于公有水印应用的。图 7.4 显示了盲检测器的水印系统。

在如图 7.4 所示的水印盲检测器中，我们并不知道未加水印的载体作品的具体形式，因此不能在解码前从已加水印的作品中减去它。这种情况可以和图 7.3 的情况对比，前者受水印被加入的单个噪声影响，后者受原始作品和噪声信号共同影响。

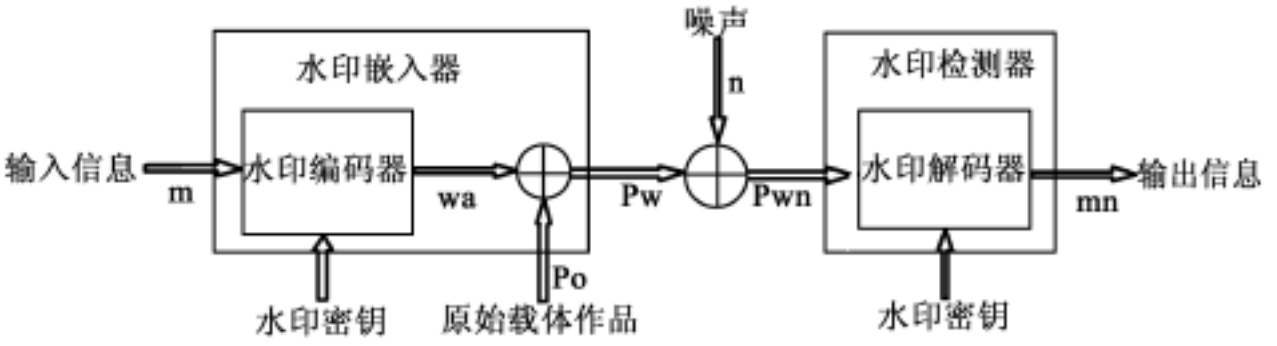


图 7.4　盲检测器的水印系统

因此,可把接收到的水印作品 Pwn 看成加性噪声作用于附加模板 wa 后的结果,此时整个水印检测器被视为信道解码器。

7.1.3　水印作为发送端带边信息的模型

尽管符合基本模型的水印系统的生成和嵌入策略都比较容易实现,但这种模型并不适用于所有的水印算法,因为它的限制是编码水印要独立于载体作品。对于水印生成策略,综合现有各算法,我们发现:数字水印的最原始来源是一个随机序列,该序列单独或者与原始图像一起通过一定的算法就能得到数字水印。这时,水印模板的生成就分为‘有原始信号参与而获得’和‘无原始信号参与而获得’两类。所谓‘有原始信号参与’是指随机序列要与原始信号的某些特征序列一起通过一定的运算得到水印信号,或者随机序列要进行一个数学变换以取得与原始信号相同的某些特征。所谓‘不需要原始信号参与’是指直接对随机序列进行变换得到水印图像,不用考虑原始信号的任何特征。“有原始信号参与”的水印也就是所谓的含辅助信息嵌入的水印。

图 7.5 和图 7.6 分别描述了盲嵌入器水印系统和 wa 依赖于 Po 的水印模型(即“有原始信号参与”的水印模型)。图 7.6 和图 7.4 所示的模型几乎相同,惟一的区别是在这个模型中,水印编码器接收 Po 作为另一个输入。这一改变使得编码器可以赋予 Pw 任何值,只需让 wa = Pw - Po。进一步考虑到载体作品是传输信道中噪声过程(Po + n)的一部分,因此,这个新模型实际上是发送端携带有带边信息的通信系统

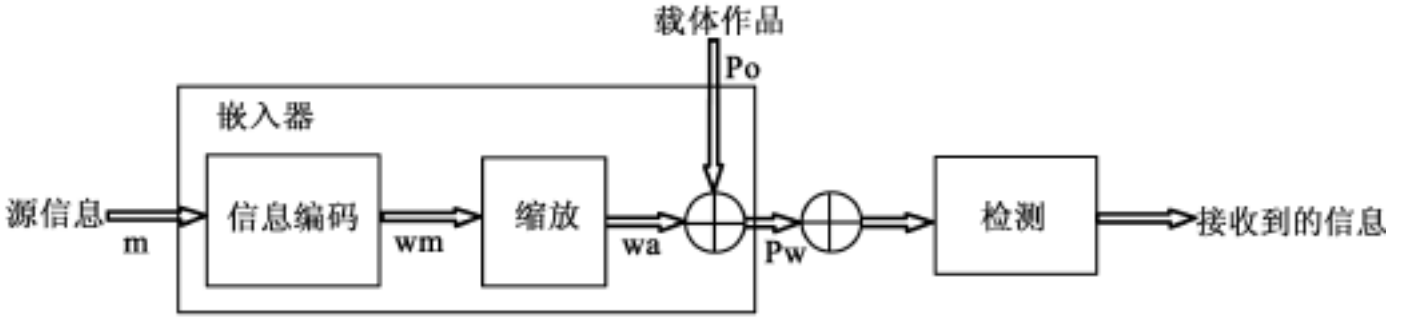


图 7.5　盲嵌入器水印系统

的一个实例。可以这样说,新模型中的嵌入器能够有效地利用信道噪声,尤其是载体作品 Po 自身的一些信息。

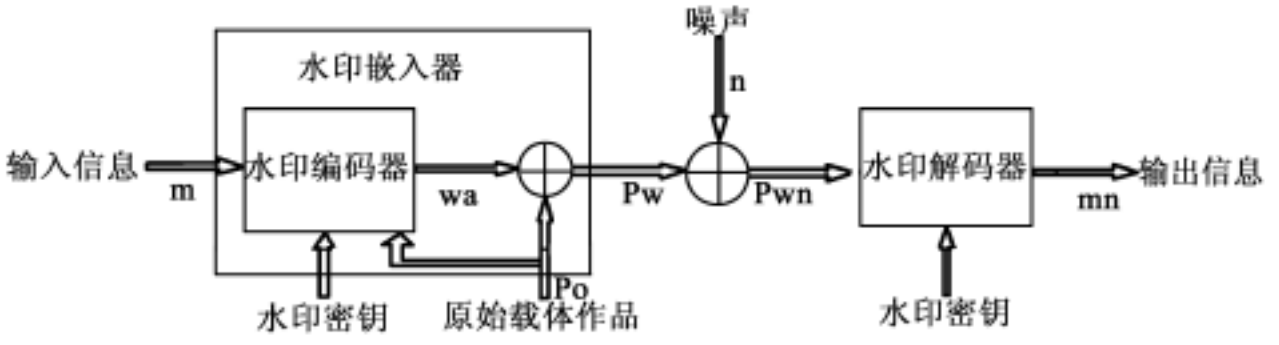


图 7.6　水印作为发送端边带信号的模型

这种通信模型最初是由 Shannon(香农) 提出的。自从他引进了带边信息通信这个概念后,很多科学研究者发现就一些信道类型而言,发送端或接收端是否能得到带边信息并不重要,重要的是要排除它的干扰。近来很多研究人员开始将边带信号的通信理论应用到水印上。

最后,我们在图 7.7 中将水印模型作一个总结。对于水印系统,我们依照其生成策略、嵌入策略以及检测策略将其分为 4 类。

水印模型
- 基本模型
  - 非盲检测　(细胞自动机水印)
  - 盲检测　(基本模型水印)
- 水印作为发送端带边信息的模型
  - 非盲检测　(W—SVD 水印)
  - 盲检测

图 7.7　水印模型分类

7.2　水印基本模型的实验实现

此实验实现一个盲嵌入算法。检测算法采用线性相关性作为检测标准,这是一个应用相当广泛的检测标准。为方便起见,只嵌入 1 比特信息,因此,m 或者等于 0,或者等于 1。此实验选取灰度图像作为载体文件。下面具体描述实验算法。

首先,我们采用一个与图像载体大小相同的像素亮度数组 wr。wr 利用水印密钥随机生成。原始水印 wm 对 1 比特信息 m 进行编码,wm 等于 + wr 或者 - wr,这取决于 m = 1 还是 m = 0。

wm 以输入参数 α为尺度进行缩放后生成水印。α如何取值决定着在水印的不可见性与鲁棒性之间如何取舍折中。因此,盲嵌入算法如下:

$$wm = \begin{cases} wr, & m = 1 \\ -wr, & m = 0 \end{cases} \tag{7.2}$$

$$wa = \alpha(wm) \tag{7.3}$$

$$Pw = Po + wa \tag{7.4}$$

可以看到,这种水印算法实际上就是对图像进行简单加噪处理。所加入的水印对图像的破坏是比较大的,在一定意义上说它不具有任何实际应用的能力。选择这个算法也仅仅是为了在最简单、最直观的条件下反映水印基本模型的特点。

图 7.8 是随机生成的水印模板。图 7.9 是原始图像与加有水印的图像。编写函数 basic_wm.m 完成实验,函数代码如下:

```
% 文件名: basic_wm.m
% 程序员: 李巍
% 编写时间: 2004.3.20
% 函数功能: 本函数是一个实现基本通信水印模型的函数
% 输入格式举例:
% [watermarkimage, watermark] = basic_wm('lenna.jpg', 1, 2000, 0.9)
% 参数说明:
% input 为原始载体图像文件名
% m 是 1 比特水印信息
% seed 为随机序列种子
% alpha 是尺度参数
% watermarkimage 嵌入水印后的图像
% wm 是生成水印
function [watermarkimage, wm] = basic_wm(input, m, seed, alpha)
Po = imread(input);
[row, col] = size(Po);
Po = double(Po)/255;
% 生成一个服从均匀分布的随机矩阵
rand('seed', seed);
wr = rand(row, col)/10;
% 将 1 比特水印信息嵌入随机矩阵
if m == 0
    wm = -wr;
else wm = wr;
end
wa = zeros(row, col);
wa = alpha* wm;
Pw = Po + wa;
% 显示结果
```

```
watermarkimage = Pw;
subplot( 131) ; imshow( Po) ; title( '原始图像') ;
subplot( 132) ; imshow( wm) ; title( '水印模板') ;
subplot( 133) ; imshow( watermarkimage) ; title( '加有水印的图像') ;
```

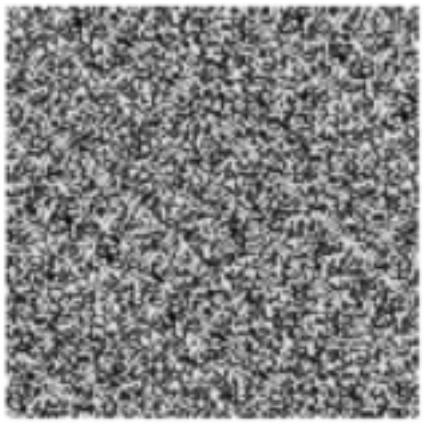


图 7.8　水印模板

原始图像

加有水印的图像

图 7.9　水印的嵌入(α = 1)

为了检测水印, 必须在有噪声的情况下检测出 ±wr 信号, 该噪声由 Po 和 n 引起。在检测时, 计算收到的图像 P 和 wr 之间的线性相关性值:

$$Z_{lp}(P, wr) = \frac{1}{N} P \cdot wr = \frac{1}{N} \sum_{x,y} P[x, y]\, wr[x, y] \tag{7.5}$$

其中, P[x, y] 和 wr[x, y] 分别表示 P 和 wr 中(x, y) 点的像素值, N 是图像的像素数量。

如果 P = Po + wa + n, 那么

$$Z_{lp}(P, wr) = \frac{1}{N}(Po \cdot wr + wa \cdot wr + n.\, wr) \tag{7.6}$$

假设 Po 和 n 满足高斯分布, 那么 Po · wr 和 n · wr 的值几乎可以肯定会很小。另一方面, wa · wr = ±αwr · wr 的幅度则要大得多。因此, 对于嵌入水印信息且m = 0

的作品有 $Z_{lp}(P, wr) \approx -\alpha wr \cdot wr/N$。

在大部分应用中，如果接收到的作品很可能不包含水印，那么我们会希望检测器输出一条特殊信息。如果检测器接收到信号 $P = Po + n$，根据上述分析可知，$Z_{lp}(P, wr)$ 的幅值将非常小。因此，可以通过设置 $Z_{lp}(P, wr)$ 值的阈值 τ_p 来判断作品是否包含水印。如果 $|Z_{lp}(P, wr)| < \tau_p$，检测器就能作出没有水印的判断。

因此，检测器的输出 mn 有：

当 $Z_{lp}(P, wr) > \tau_p$ 时，mn = 1；

当 $Z_{lp}(P, wr) < -\tau_p$ 时，mn = 0；

当 $-\tau_p \leq Z_{lp}(P, wr) \leq \tau_p$ 时，无水印。

阈值 τ_p 越低，未加水印的作品越有可能被当做水印作品。检测的函数为 detectbasic_wm. m，函数代码如下：

```
% 文件名: detectbasic_wm. m
% 程序员: 李巍
% 编写时间: 2004. 3. 20
% 函数功能: 本函数是一个检测基本通信水印的函数
% 输入格式举例:
% Mn = detectbasic_wm( watermarkimage, 2000, 0. 7)
% 参数说明:
% watermarkimage 嵌入水印后的图像数据矩阵
% seed 为随机序列种子
% tlp 是检测阈值
% mn 是检测结果
function Mn = detectbasic_wm( watermarkimage, seed, tlp)
P = watermarkimage;
[ row, col] = size( P) ;
% 生成一个服从均匀分布的随机矩阵
rand( 'seed', seed) ;
wr = rand( row, col) * 3;
% 计算线性相关 Zlp
Zlp = 0;
for j = 1: col
    for i = 1: row
        Zlp = Zlp + P( i, j) * wr( i, j) ;
    end
end
Zlp = Zlp /( row* col) ;
```

```
% 根据检测阈值判断检测结果
if Zlp > tlp
        mn = 1;
elseif  - tlp > Zlp
        mn = 0;
else mn =  - 1; % 无水印
end
```

表 7.1 是定义不同的检测阈值时水印检测的结果。

表 7.1　　水印检测(m = 1)

判 定 阈 值	检 测 结 果	
0.2	1	检测有水印
0.3	1	
0.5	- 1	检测无水印
0.6	- 1	

7.3　W-SVD 数字水印算法

7.3.1　W-SVD 数字水印算法描述

W-SVD 数字水印算法是美国 Syracuse 大学数学系和美国空军实验室通信遥感部联合于 1998 年发布的。该算法属于小波变换域数字水印算法, 具有良好的水印不可见性和鲁棒性等特点。尽管这一算法并不十分完美, 而且也不具备现代数字水印自适应和盲检测的要求, 但无论如何, W-SVD 的经典性是无庸置疑的。从其发布到现在, 已经有不少学者从各个方面对其进行了改进, 使得这种水印从实验到具体应用一步一步迈进。我们选择此算法也正是看重了其良好的教学和实验特性, 大家有必要重点掌握。

完整的 W-SVD 数字水印算法包括水印生成、水印嵌入和水印检测三个部分。W-SVD 水印的检测我们将在后面专门叙述, 其生成和嵌入策略描述如下:

假设要加入水印的图像为 M, 其归一化后的尺度 level 下的低频系数记为 CA = wavetrans(M, level) 。对 CA 作奇异值(单值) 分解, 得到: $CA = U\Sigma V^T$

其中有:

$$U = \begin{pmatrix} u_1 \\ \cdots \\ u_n \end{pmatrix} \qquad V = \begin{pmatrix} v_1 \\ \cdots \\ v_n \end{pmatrix} \qquad \Sigma = \begin{pmatrix} \sigma_1 & & \\ & W & \\ & & \sigma_n \end{pmatrix}$$

其中, U 和 V 是正交矩阵, 即 $UU^T = VV^T = I$, I 是单位矩阵。Σ是对角矩阵, 即除主对角线外为 0 的矩阵。对于任意的图像矩阵, 这种奇异值(单值) 分解都是可以成立的。这为我们对任意图像加水印提供了保证。

按照 U, V 和 Σ的特点, 随机生成这样三个矩阵:

$$\overline{U} = \begin{bmatrix} \text{存存} \\ \cdots \\ \text{存存} \end{bmatrix} \qquad \overline{V} = \begin{bmatrix} \text{聊聊} \\ \cdots \\ \text{聊聊} \end{bmatrix} \qquad \text{斑斑} = \alpha \begin{bmatrix} \text{斑斑}_1 & & \\ & W & \\ & & \text{斑斑}_n \end{bmatrix}$$

$\overline{U}$ 和 $\overline{V}$ 也要是正交矩阵, 这一点可以利用随机矩阵的 QR 分解(正交—三角分解) 来实现。α是我们第一个要关心的参数, 它的取值决定了水印强度。

作为噪声的水印模板是这样产生的。用种子控制的随机矩阵 $\overline{U}$ 和 $\overline{V}$ 的后 d 列(行) 来替换原始低频系数分解矩阵 U 和 V 的后 d 列(行) , 得到矩阵 U 和 V。d 是一个由比例因子 d/n 决定的整数。n 是 U 或 V 的列(行) 数, d/n 是一个比例, 这是我们要关心的第二个参数, 其矩阵如下:

$$U = \begin{bmatrix} u_1 \\ \cdots \\ u_{n-d} \\ \text{存存}_{n-d+1} \\ \cdots \\ \text{存存} \end{bmatrix} \qquad V = \begin{bmatrix} v_1 \\ \cdots \\ v_{n-d} \\ \text{聊聊}_{n-d+1} \\ \cdots \\ \text{聊聊} \end{bmatrix}$$

由 U, V 和 $\overline{\Sigma}$ 进而构成完整的水印模板 waterCA:

$$\text{waterCA} = U\overline{\Sigma}V^T$$

这就是全部的水印生成策略。可以发现, 与前一节基本模型不同的是, 水印模板并不是单纯的由独立于原始图像的随机噪声构成, 而是与原始图像有密切相关的联系。得到水印模板后, 将其加入到原始图像的低频系数中, 就完成了水印的嵌入。即:

$$CA_w = CA + \text{waterCA}$$

因为 CA 是归一化的低频系数, waterCA 也是由 CA 获得的。在重构图像时应对 CA_w 作适当放缩, 恢复到原始图像低频系数的数量级, 完成图像重构。

图 7.10 是上述算法的流程图。

结合图 7.10, 我们具体来看一下算法中值得注意的几个问题。

(1) 小波分解提取低频系数

对于二维图像的小波低频系数, 在第三章中我们已经从原理到实现上给予了详细的说明。这里要强调的是, 图像的小波低频系数是与分解尺度有密切联系的。

(2) 低频系数矩阵的奇异(单) 值分解

与小波分解一样, 矩阵奇异值分解也是构成本算法的灵魂。我们从数学上来简

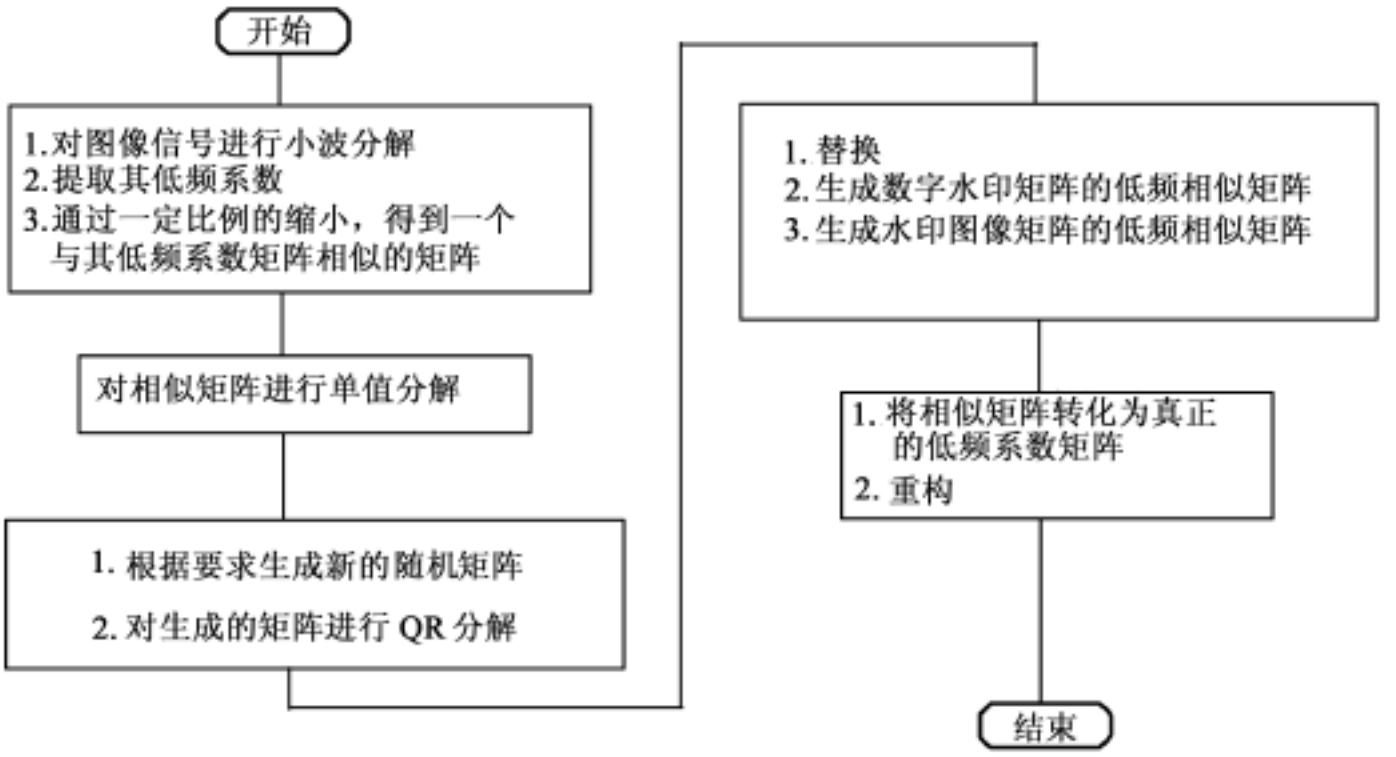


图 7.10　W-SVD 算法流程图

单说明一下什么是奇异值分解。

定义 7.1　对于 N× N 矩阵 A, 有 N 个标量 λ_N(N = 1, 2, …, N) 满足:

$$| A - \lambda_N I | = 0$$

则称这一组 λ_N 为矩阵 A 惟一的特征值。

定义 7.2　当矩阵的每个对角元素都减去其特征值时, 矩阵将变为奇异阵。

定义 7.3　矩阵 A 的秩等于其非 0 特征值的个数。

定义 7.4　如果存在这样一个 N× 1 的向量 V_K, 有:

$$AV_K = \lambda_K V_K$$

则称 V_K 为 A 的与特征值 λ_K 对应的一个特征向量。A 一共有 N 个特征向量。

定义 7.5　(矩阵奇异值分解)　对于任意 M× N 矩阵 B, 都可以写成 $B = U\Sigma V^T$, 其中 U 和 V 分别是 M× M 和 N× N 的正交矩阵。Σ是 M× N 的对角矩阵, 其对角元包含了 B 的奇异值。具体地说, U 的各列是矩阵 BB^T 的特征向量, V 的各列是矩阵 B^TB 的特征向量。$\Sigma = U^TBV$ 为奇异值矩阵。这种变换又称矩阵 SVD 变换。

下面, 我们以一个简单的 5× 4 的矩阵 B 来具体看一下其 SVD 变换。

$$B = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \\ 17 & 18 & 19 & 20 \end{bmatrix} = U\Sigma V^T$$

其中，$U = \begin{bmatrix} -0.096548 & -0.76856 & 0.62318 & -0.0010772 & -0.10789 \\ -0.24552 & -0.48961 & -0.53833 & 0.23524 & 0.5957 \\ -0.39448 & -0.21067 & -0.45389 & 0.057371 & -0.76857 \\ -0.54345 & 0.06827 & 0.030023 & -0.81617 & 0.18157 \\ -0.69242 & 0.34721 & 0.33901 & 0.52463 & 0.09918 \end{bmatrix}$

$$\Sigma = \begin{bmatrix} 53.52 & 0 & 0 & 0 \\ 0 & 2.3634 & 0 & 0 \\ 0 & 0 & 4.9168e-015 & 0 \\ 0 & 0 & 0 & 7.112e-016 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$V = \begin{bmatrix} -0.44302 & 0.70974 & 0.43191 & -0.33683 \\ -0.47987 & 0.26405 & -0.82694 & 0.12717 \\ -0.51673 & -0.18164 & 0.35814 & 0.75613 \\ -0.55358 & -0.62734 & 0.036886 & -0.54648 \end{bmatrix}$$

矩阵 SVD 分解在一定程度上可以用来进行图像压缩。对于图像分解得到的奇异值矩阵 Σ，将其较小的一些对角元清为 0，再进行 SVD 反变换即可完成图像的有损压缩。

在 W-SVD 水印算法中，我们的水印模板就是根据经过 SVD 分解后的原始图像的低频小波系数矩阵生成的。在这里，我们尤其要注意对奇异值对角矩阵 Σ的分析。

(3) 应关注的参数

在 W-SVD 算法中，有两个参数是我们要重点关注的：强度因子 α和水印模板生成因数 d/n。理论上，α是作用在一个随机对角矩阵上的，其整体 琢用以代替原始的奇异值矩阵 Σ。我们应该通过试验分析这几者的关系，找出为什么 α是强度因子的原因以及 α的适当取值。参数 d/n 表明了随机正交矩阵 琁和 琢代替原始分解矩阵 U 和 V 的比例。我们应该通过试验分析 d/n 与水印性能的关系。

此外，算法中使用的小波、小波分解的尺度和随机数种子都是我们要关心的。

7.3.2 W-SVD 算法实现

参照图 7.10，我们来具体分步实现 W-SVD 算法。

(1) 图像小波分解及低频系数归一化

我们首先对图像进行小波分解提取低频系数，由于这一低频系数矩阵中的值不在[0,1]内，为了后面操作的方便，我们将其进行如下放缩：

$$CAsimilar = (1/(CAmax - CAmin)) \times (CA - CAmin)$$

这样就得到了归一化后的原始图像低频矩阵。

(2) SVD 变换

将归一化后得到的系数矩阵进行单值分解。在 MATLAB 中，SVD 变换是通过内

置函数 svd. m 完成的。该函数调用形式如下:

[U, sigma, V] = svd(B)

B 为要分解的矩阵, U, sigma, V 就是前面分析过的 U, Σ和 V。

图 7. 11 是 lenna R 层作 2 尺度低频的系数矩阵的相似矩阵单值分解结果。

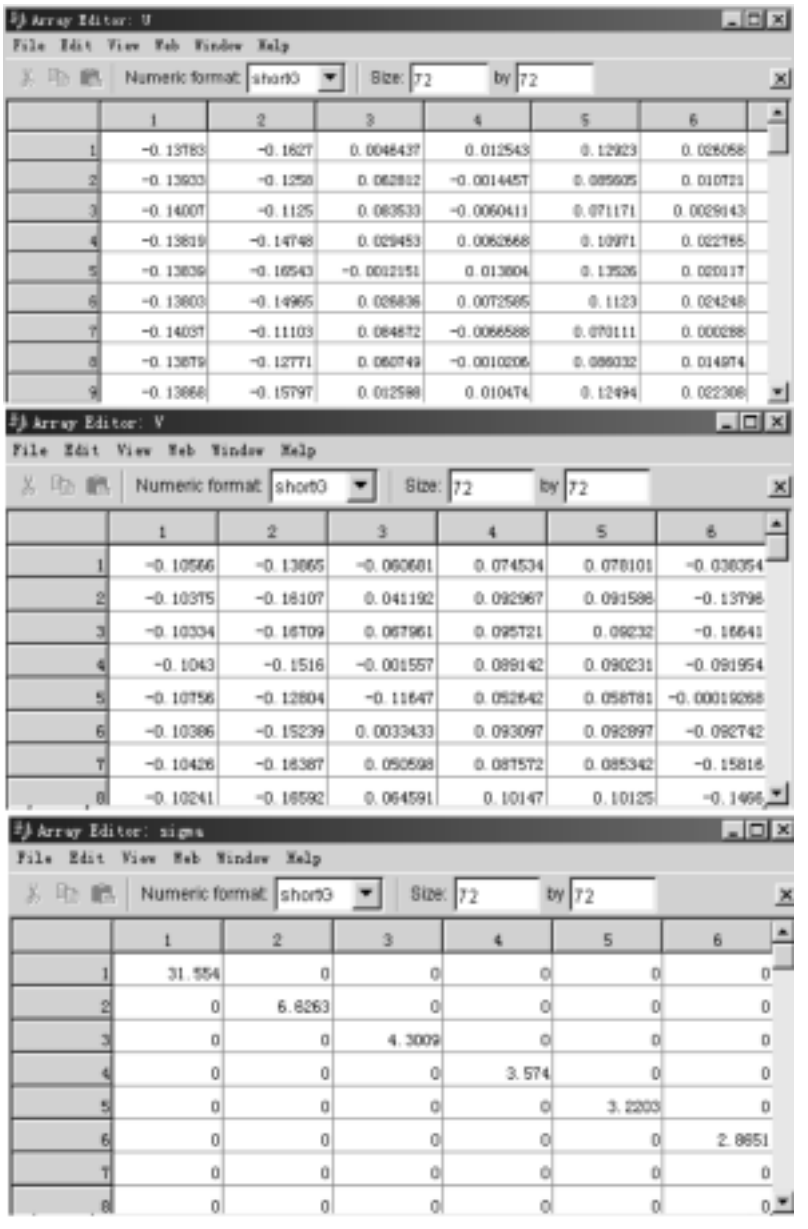


图 7. 11　lenna 低频系数的 SVD 分解

(3) 正交随机矩阵的生成

值得注意的是, 在 W-SVD 算法中, 需要生成 珊和 琦两个随机的正交矩阵。生成随机矩阵很容易, 而生成的矩阵若是正交的则必须通过一定数学变换来完成。这里, 使用的是矩阵的 QR 分解得到随机正交矩阵的。

对于一个 M× N 矩阵 C, 其 QR 分解将得到一个 M× M 的正交矩阵 Q 和一个 M× N的矩阵 R。当 M = N 时, R 是上三角矩阵, 这也是对于任何矩阵都适用的。

在 MATLAB 中, QR 变换是通过内置函数 qr. m 完成的, 该函数调用形式如下:

[Q, R] = qr(C)

其中, C 为要分解的矩阵, Q 和 R 是分解的结果。

所以, 当 C 是随机生成的矩阵时, 得到的 Q_C 就是一个随机正交矩阵, 相应的 R_C 则废弃不用。图 7.12 是按照图 7.11 中 U 的尺寸随机生成的 72× 72 的矩阵以及其 QR 分解的结果。

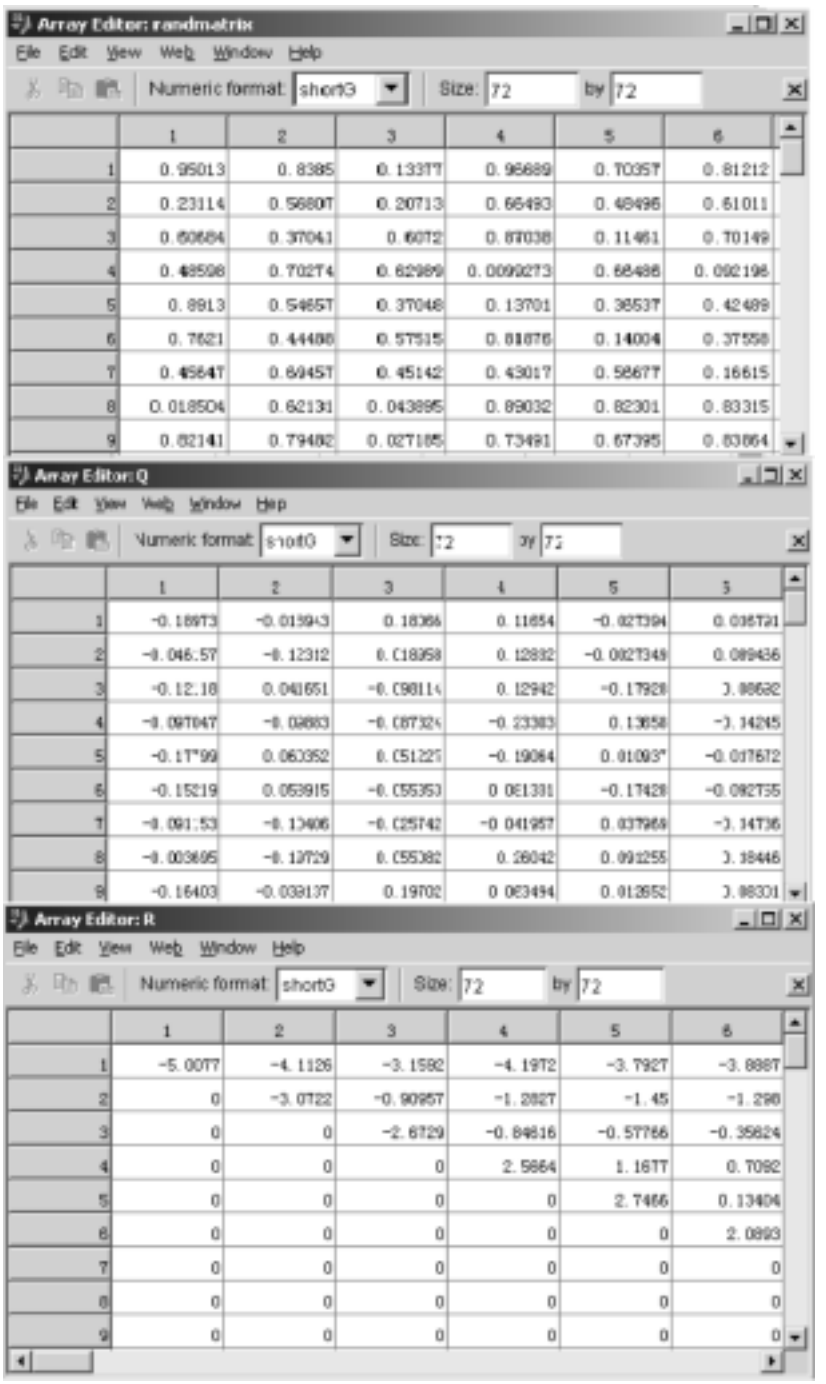


图 7.12　随机矩阵的 QR 分解

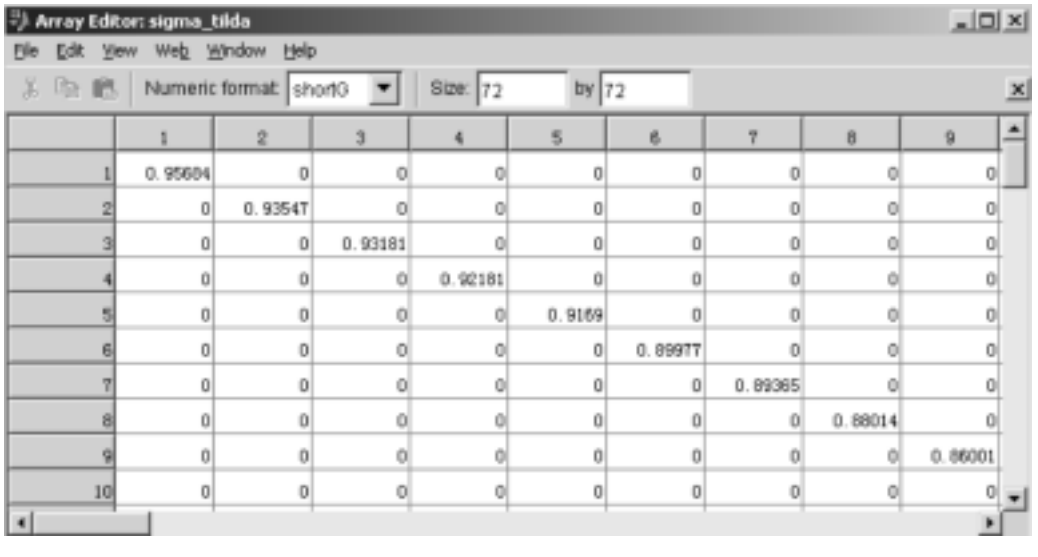
(4) 随机对角矩阵的生成

算法中的随机对角矩阵是通过以下一组命令完成的:

sigma_tilda = alpha* diag(flipud(sort(rand(d,1))))

即首先调用 rand 函数随机生成一个 d 行 1 列的向量, 再用 sort 和 flipud 函数对其从大到小排列, 最后用 diag 函数完成对角化。上式中 alpha 就是强度因子 α, d 是

根据 d/n 计算出的要替换的行数，sigma_tilda 就是算法中的 Σ。图 7.13 是按照图 7.11中 sigma 的尺寸随机生成的 sigma_tilda。此时强度因子 α取为 1。



	1	2	3	4	5	6	7	8	9
1	0.95604	0	0	0	0	0	0	0	0
2	0	0.9354T	0	0	0	0	0	0	0
3	0	0	0.93181	0	0	0	0	0	0
4	0	0	0	0.92181	0	0	0	0	0
5	0	0	0	0	0.9169	0	0	0	0
6	0	0	0	0	0	0.8997T	0	0	0
7	0	0	0	0	0	0	0.89365	0	0
8	0	0	0	0	0	0	0	0.88014	0
9	0	0	0	0	0	0	0	0	0.86001
10	0	0	0	0	0	0	0	0	0

图 7.13　随机生成的对角矩阵

(5) 替换

我们用图 7.14 说明替换的过程:

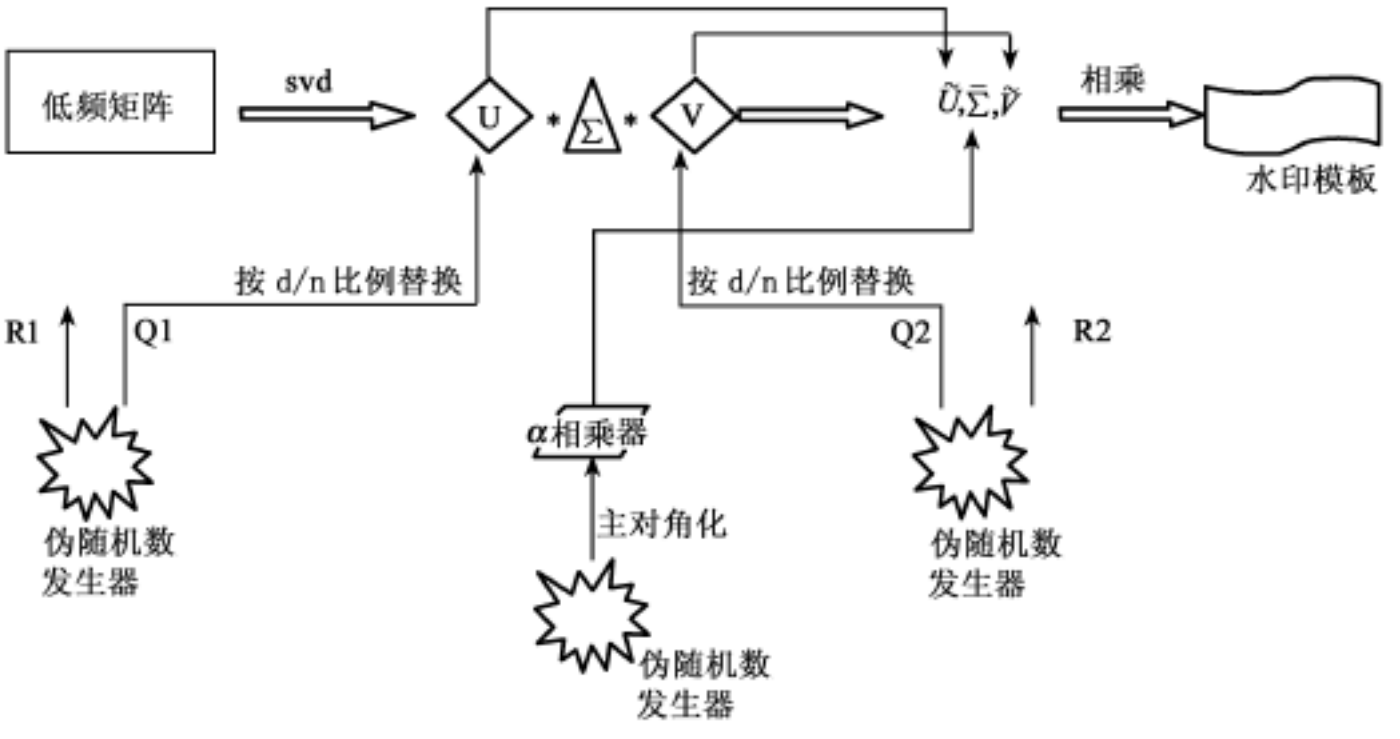


图 7.14　水印模板的生成

(6) 图像的补充

考虑到随机对角矩阵的生成，矩阵的 QR 分解以及编写函数的通识性等多方面的问题，我们在处理图像时都是先将其补充成行列相等的正方形矩阵。

(7) MATLAB 精度对实验的影响

我们已经不止一次地遇到过 MATLAB 精度对实验结果造成影响的例子。为了后面实验分析的正确, 建议大家使用 16 位的 png 格式图像。

编写函数 wavemarksvd. m 完成 W-SVD 数字水印算法的实验。函数代码如下:

```
% 文件名: wavemarksvd. m
% 程序员: 郭迟
% 编写时间: 2003. 10. 7
% 函数功能: 本函数将完成 W-SVD 模型下数字水印的嵌入
% 输入格式举例:
[ watermarkimagergb, watermarkimage, waterCA, watermark, correlationU, correlationV] = wavemarksvd( 'c: \lenna. jpg', 'c: \test. png', 1983, 'db6', 2, 0. 1, 0. 99)
% 参数说明:
% input 为输入原始图像
% seed 为随机数种子
% wavelet 为使用的小波函数
% level 为小波分解的尺度
% alpha 为水印强度
% ratio 为算法中 d/n 的比例
% watermarkimagergb 为加有水印的结果
% watermarkimage 为单层加水印的结果
% waterCA 为加有水印模板的低频分解系数
% watermark2 为由水印模板直接重构得到的水印形态, 仅便于直观认识, 本身无意义。
% correlationU, correlationV 为替换正交矩阵后与未替换的正交矩阵的相关系数
function
[ watermarkimagergb, watermarkimage, waterCA, watermark2, correlationU, correlationV] = wavemarksvd( input, goal, seed, wavelet, level, alpha, ratio)
% function watermark = wavemarksvd( input, goal, seed, wavelet, level, alpha, ratio)
% 读取原始图像
data = imread( input) ;
data = double( data) /255;
datared = data(:,:, 1) ; % 在 R 层加水印
% 对原始图像的 R 层进行小波分解记录原始大小, 并将其补成正方形
[ C, Sreal] = wavedec2( datared, level, wavelet) ;
[ row, list] = size( datared) ;
standard1 = max( row, list) ;
new = zeros( standard1, standard1) ;
```

```
if row < = list
        new( 1∶ row,∶ )  = datared;
else
        new(∶, 1∶ list)  = datared;
end
% 正式开始加水印
% 小波分解, 提取低频系数
[ C, S]  = wavedec2( new, level, wavelet) ;
CA = appcoef2( C, S, wavelet, level) ;
% 对低频系数进行归一化处理
[ M, N]  = size( CA) ;
CAmin = min( min( CA) ) ;
CAmax = max( max( CA) ) ;
CA = ( 1 /( CAmax - CAmin) ) *  ( CA - CAmin) ;
d = max( size( CA) ) ;
% 对低频率系数单值分解
[ U, sigma, V]  = svd( CA) ;
% 按输出参数得到要替换的系数的数量
np = round( d*  ratio) ;
% 以下是随机正交矩阵的生成
rand( 'seed', seed) ;
M_V = rand( d, np)  - 0. 5;
[ Q_V, R_V]  = qr( M_V, 0) ;
M_U = rand( d, np)  - 0. 5;
[ Q_U, R_U]  = qr( M_U, 0) ;
% 替换
V2 = V; U2 = U;
V(∶, d - np + 1: d)  = Q_V(∶, 1: np) ;
U(∶, d - np + 1: d)  = Q_U(∶, 1: np) ;
sigma_tilda = alpha*  flipud( sort( rand( d, 1) ) ) ;
correlationU = corr2( U, U2) ; % 计算替换的相关系数
correlationV = corr2( V, V2) ;
% 生成水印
watermark = U*  diag( sigma_tilda, 0) *  V';
% 重构生成水印的形状, 便于直观认识, 本身无意义
watermark2 = reshape( watermark, 1, S( 1, 1) *  S( 1, 2) ) ;
```

```
waterC = C;
waterC( 1, 1: S( 1, 1) *  S( 1, 2) )  = watermark2;
watermark2 = waverec2( waterC, S, wavelet) ;
% 调整系数生成嵌入水印后的图像
CA_tilda = CA + watermark;
over1 = find( CA_tilda > 1) ;
below0 = find( CA_tilda < 0) ;
CA_tilda( over1)  = 1;
CA_tilda( below0)  = 0; % 系数调整, 将过幅系数与负数修正
CA_tilda = ( CAmax-CAmin) *  CA_tilda + CAmin; % 系数还原到归一化以前的范围
% 记录加有水印的低频系数
waterCA = CA_tilda;
if row <= list
        waterCA = waterCA( 1: Sreal( 1, 1) ,: ) ;
else
        waterCA = waterCA(: , 1: Sreal( 1, 2) ) ;
end
% 重构
CA_tilda = reshape( CA_tilda, 1, S( 1, 1) *  S( 1, 2) ) ;
C( 1, 1: S( 1, 1) *  S( 1, 2) )  = CA_tilda;
watermarkimage = waverec2( C, S, wavelet) ;
% 将前面补上的边缘去掉
if row <= list
        watermarkimage = watermarkimage( 1: row,: ) ;
else
        watermarkimage = watermarkimage(: , 1: list) ;
end
watermarkimagergb = data;
watermarkimagergb(: ,: , 1)  = watermarkimage;
imwrite( watermarkimagergb, goal, 'BitDepth', 16) ; % 通过写回修正过幅系数
watermarkimagergb2 = imread( goal) ;
% figure( 1) ;
% subplot( 321) ; imshow( watermark2* 255) ; title( '水印形态图') ;
% subplot( 323) ; imshow( data) ; title( '原始图像') ;
% subplot( 324) ; imshow( watermarkimagergb2) ; title( '嵌入水印后的 RGB 图像') ;
% subplot( 325) ; imshow( datared) ; title( 'R 层图像') ;
```

% subplot(326) ; imshow(watermarkimage) ; title(嵌入水印后的 R 层图像) ;

图 7. 15 ～图 7. 17 是将不同参数代入函数得到的水印效果。

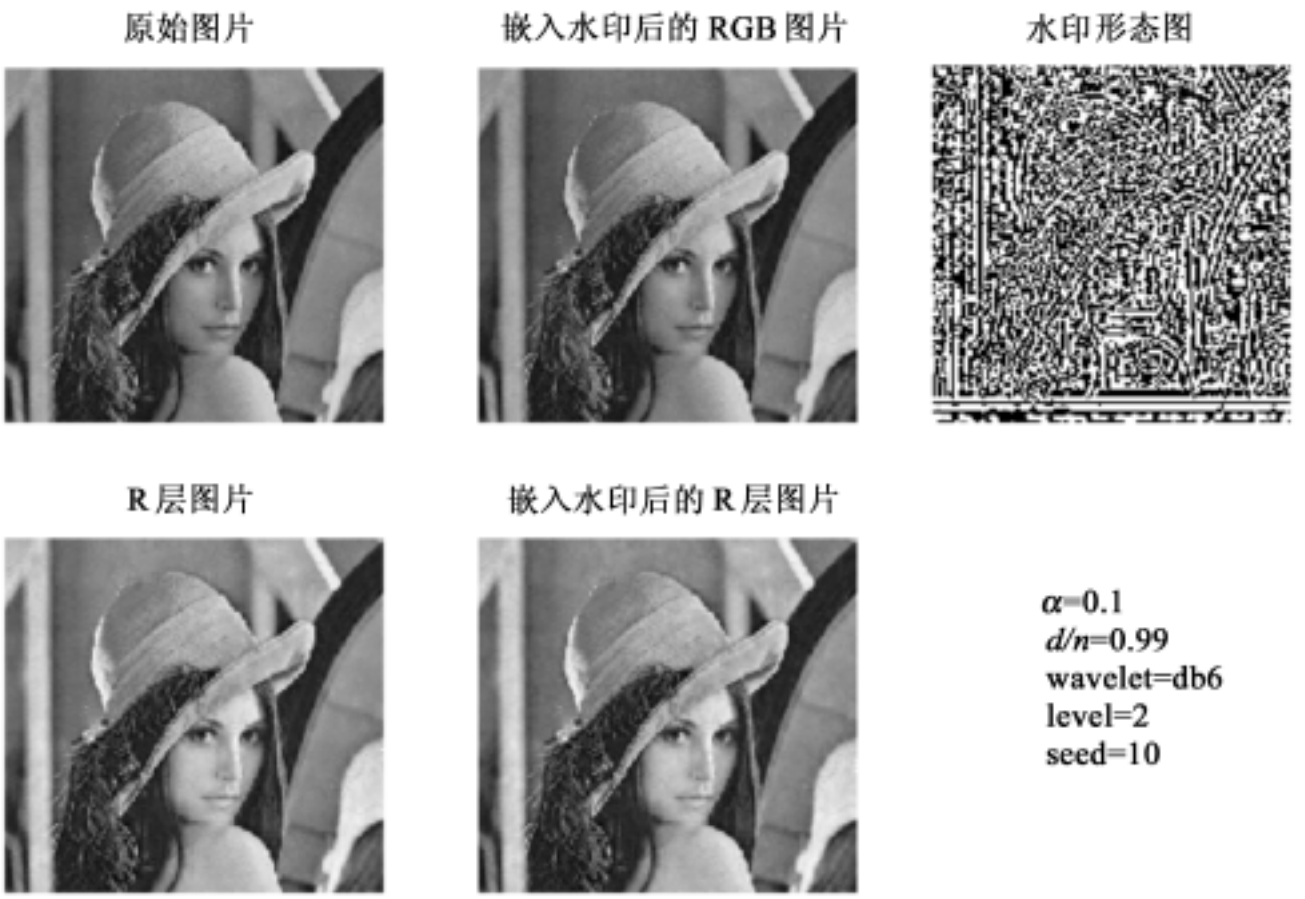


图 7. 15　实验结果 1

图 7. 16　实验结果 2

图 7.17　实验结果 3

7.3.3　W-SVD 水印的检测和检测阈值的确定

W-SVD 算法采用非盲检测手段对图像进行检测。其思路为: 利用原始图像生成一个理论上存在的水印模板(原始水印) , 从待测图像中提取可能存在的水印模板(待测水印) , 继而计算两者的相关性。当两者高度相关时, 我们认为待测图像含有水印; 反之则检测不出水印。水印的检测模型如图 7.18 所示。

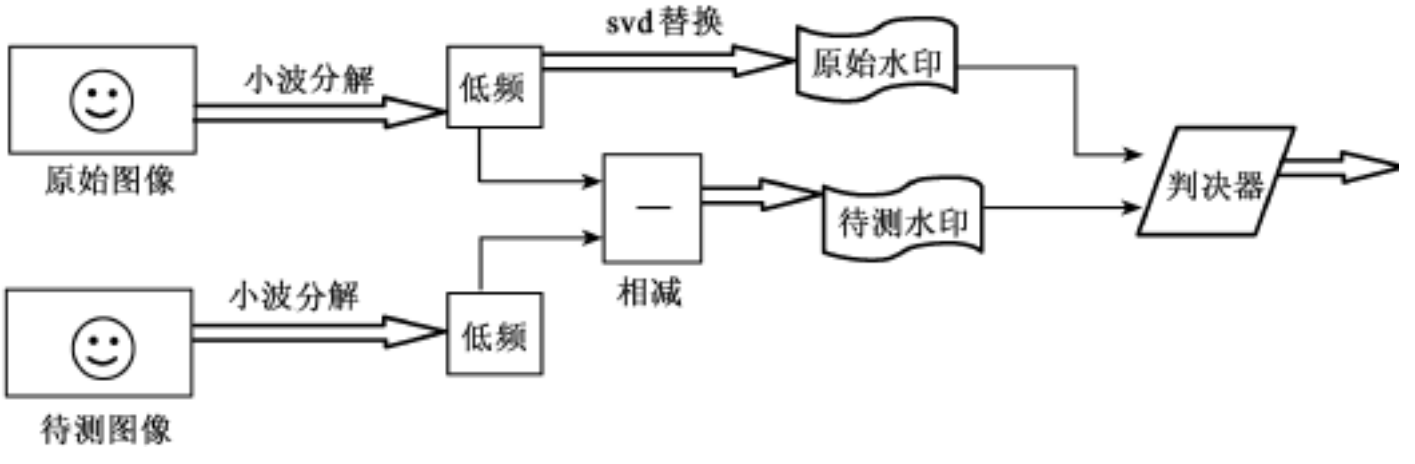


图 7.18　W-SVD 水印的检测

前文已述, 由于数字水印一般是一种具有特定性质但不具备可读性的随机信号, 所以我们不能像信息隐秘中的机密信息一样采用“ 提取 ”的方式加以识别。我们对水印的识别是通过检测的手段实现的。我们要检测作品 N 是否含有水印 W, 则需要

将原始作品 M 用策略 K 加入水印 W, 然后用同样的策略 K 从 N 中提取我们认为是 W 的 W′, 继而计算两者的相关性。当量化的相关性值大于一个特定值时, 我们就认为 W′= W, 即 N 作品含有 W; 反之则不然。这里需要说明的是, 为什么我们不是直接比较 W 是否等于 W, 而是计算其相关性? 这是因为即使 N 含有 W, 当 N 经历重构、存储、传输、分解等过程后, 其水印与原始水印因加性噪声的原因仍然会存在很大的差异。我们只能通过一定的方式来消除这种差异给水印识别上带来的错误, 而不能简单地比较两者是否相同。

图 7.19 是我们将 lenna 加入原始水印的低频系数与嵌入水印后的图像提取的低频系数作差后的三维投影, 绝大多数差值相对 0 平面有一定的距离。尤其在边缘部分, 这种差异更是不可忽略的, 这还仅仅是将图像存储后又马上提取的结果。一旦图像在信道中传输, 这种差异还要大。所以, 即使两者生成时使用的一切参数都相同, 也不可能使得原始水印和待测水印是完全相同的。

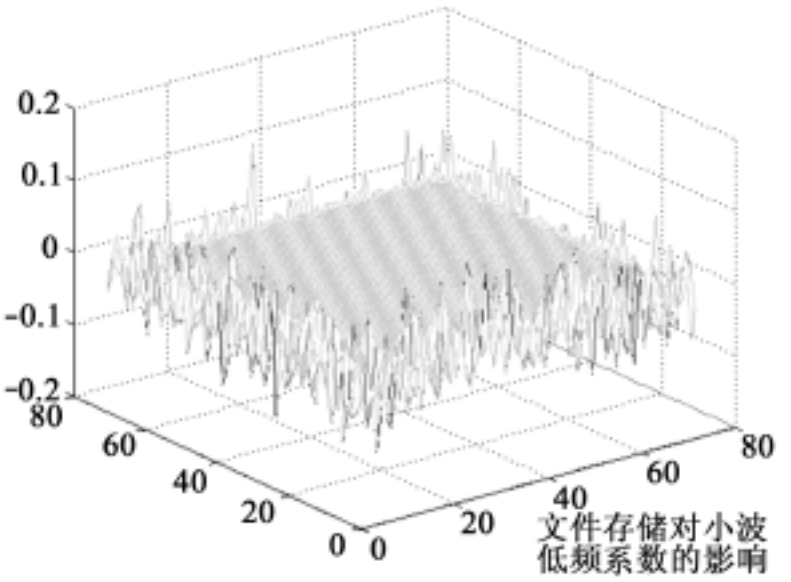


图 7.19　水印的检测只能用相关的方式完成

计算相关系数的方法很多, 这里我们选择以下两种:

方法一: 常规检测直接相关性值 d

$$d = \frac{|<W, W'>|}{\|W\| \cdot \|W'\|} \tag{7.7}$$

其中, W 和 W′分别表示图 7.18 中的原始水印和待测水印。$<W, W'> = \sum_{i=1}^{M}\sum_{j=1}^{N} w_{ij} w_{ij}'$, M 和 N 为水印模板的大小。$\|W\| = \sqrt{<W, W>}$, $\|W'\| = \sqrt{<W', W'>}$。

方法二: DCT 域相关性值 $\hat{d}$

$$\hat{d} = \frac{|<\hat{W}, \hat{W}'>|}{\|\hat{W}\| \cdot \|\hat{W}'\|} \tag{7.8}$$

^表示经过 DCT 后的系数矩阵。实验证明, 只需要采用有限点的 DCT 系数便可以完

成检验,且效果良好。这里取的是 1024 点(32× 32) DCT 系数。

理论上讲,当待测图像确有水印时,无论是 d 还是 $\hat{d}$ 都将是 1。但由于实际信号在传输中不可避免地受到信道的影响,所以这里求得的相关性值很难达到 1。

编写函数 wavedetect. m 完成检测实验,函数代码如下:

```
% 文件名: wavedetect. m
% 程序员: 郭迟
% 编写时间: 2003. 10. 7
% 函数功能: 本函数将完成 W-SVD 模型下数字水印的检测
% 输入格式举例:
[ corr_coef, corr_DCTcoef] = wavedetect( 'c: \test. png', 'c: \lenna. jpg', 1983, 'db6',2,0.1,0.99)
% 参数说明:
% input 为输入原始图像
% seed 为随机数种子
% wavelet 为使用的小波函数
% level 为小波分解的尺度
% alpha 为水印强度
% ratio 为算法中 d/n 的比例
% corr_coef, corr_DCTcoef 分别为不同方法下检测出的相关性值
function [ corr_coef, corr_DCTcoef] = wavedetect( test, original, seed, wavelet, level, alpha, ratio)
dataoriginal = imread( original) ;
datatest = imread( test) ;
dataoriginal = double( dataoriginal) /255;
datatest = double( datatest) /65535;
% 请大家注意这里的两个分母,这是与图像文件格式有关的
dataoriginal = dataoriginal(:,:, 1) ;
datatest = datatest(:,:, 1) ;
% 提取加有水印的图像的小波低频系数
[ watermarkimagergb, watermarkimage, waterCA, watermark2, correlationU, correlationV] = wavemarksvd( original, 'temp. png', seed, wavelet, level, alpha, ratio) ;
% 提取待测图像的小波低频系数
[ C, S] = wavedec2( datatest, level, wavelet) ;
CA_test = appcoef2( C, S, wavelet, level) ;
% 提取原始图像的小波低频系数
[ C, S] = wavedec2( dataoriginal, level, wavelet) ;
```

```
realCA = appcoef2( C, S, wavelet, level) ;
% 生成两种水印
realwatermark = waterCA-realCA;
testwatermark = CA_test-realCA;
% 计算相关性值
corr_coef = trace( realwatermark ' * testwatermark) /( norm( realwatermark, 'fro') *
norm( testwatermark, 'fro') ) ;
% DCT 系数比较
DCTrealwatermark = dct2( waterCA-realCA) ;
DCTtestwatermark = dct2( CA_test-realCA) ;
DCTrealwatermark = DCTrealwatermark( 1: min( 32, max( size( DCTrealwater-
mark) ) ) , 1: min( 32, max( size( DCTrealwatermark) ) ) ) ;
DCTtestwatermark = DCTtestwatermark( 1: min( 32, max( size( DCTtestwater-
mark) ) ) , 1: min( 32, max( size( DCTtestwatermark) ) ) ) ;
DCTrealwatermark( 1, 1) = 0;
DCTtestwatermark( 1, 1) = 0;
corr_DCTcoef = trace( DCTrealwatermark ' * DCTtestwatermark) /( norm( DCTrealwa-
termark, 'fro') * norm( DCTtestwatermark, 'fro') ) ;
```

图 7.18 所示的判决器最终要根据一个检测阈值(Test_threshold) 来决定水印的有无。我们在[0, 20] 内取 20 个种子生成 20 种原始水印并计算出待测水印的 d 和 $\hat{d}$, 最终分析获得相应的检测阈值。具体的算法流程图, 如图 7.20 所示。

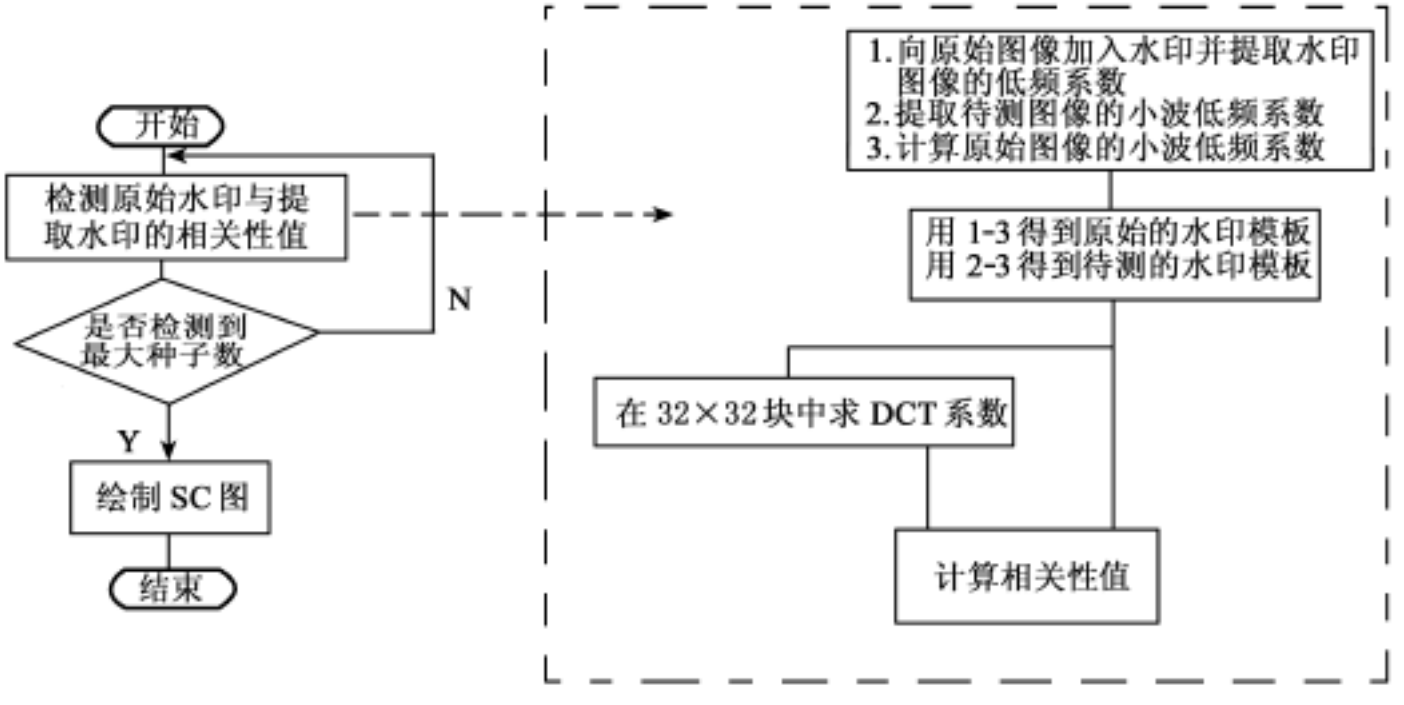


图 7.20　求取适当阈值的流程图

编写函数 plotcorr_coef. m 完成上述实验。函数代码如下:

```
% 文件名: plotcorr_coef. m
% 程序员: 郭迟
% 编写时间: 2003. 10. 7
% 函数功能: 这是一个绘制 SC 图的函数
% 输入格式举例:
[ corr_Wcoef, corr_Dcoef] = plotcorr_coef( 'c: \test. png', 'c: \lenna. jpg', 20, 'db6', 2, 0. 1, 0. 99)
% 参数说明:
% test 为待测图像
% original 为原始图像
% testMAXseed 为实验使用的最大随机数种子
% wavelet 为使用的小波函数
% level 为小波分解的尺度
% alpha 为水印强度
% ratio 为算法中 d/n 的比例
% corr_Wcoef, corr_Dcoef 分别为利用不同种子检测出的相关性值的集合
function
[ corr_Wcoef, corr_Dcoef] = plotcorr_coef( test, original, testMAXseed, wavelet, level, alpha, ratio)
corr_Wcoef = zeros( testMAXseed, 1) ;
corr_Dcoef = zeros( testMAXseed, 1) ;
s = 1;
for i = 1: testMAXseed
    [ corr_coef, corr_DCTcoef] = wavedetect( test, original, i, wavelet, level, alpha, ratio) ;
    corr_Wcoef( s) = corr_coef;
    corr_Dcoef( s) = corr_DCTcoef;
    s = s + 1;
end
subplot( 211) ; plot( abs( corr_Wcoef) ) ;
title( '常规检测阈值分析') ;
xlabel( '种子') ;
ylabel( '相关性值') ;
subplot( 212) ; plot( abs( corr_Dcoef) ) ; 待测
title( 'DCT 变换后检测阈值分析') ;
xlabel( '种子') ;
```

ylabel(相关性值);

下面分别是通过以上函数 20 次常规检测的水印系数相关性值(如表 7.2 所示) 和 DCT 变换后检测的相关性值(如表 7.3 所示) 。其中第 10 次使用的种子是正确的。

表 7.2　　常规检测水印系数相关性值

常规系数相关性值	1	2	3	4	5
	0.038151	0.035859	0.030678	0.059559	0.010753
	6	7	8	9	10
	0.060545	0.037593	0.03337	0.046426	0.83212
	11	12	13	14	15
	0.036774	0.028876	0.038522	0.065498	0.061114
	16	17	18	19	20
	0.043419	0.033113	0.015824	0.034676	0.044406

表 7.3　　DCT 变换后检测的相关性值

DCT 系数相关性值	1	2	3	4	5
	0.042142	0.027408	0.013771	0.007516	0.063773
	6	7	8	9	10
	0.022848	0.0014088	0.026152	0.018246	0.7829
	11	12	13	14	15
	0.016097	0.010609	0.012322	0.063708	0.018965
	16	17	18	19	20
	0.077649	0.0028074	0.065668	0.0070119	0.0068167

经检测可以发现, 当待测水印的种子与原始水印的种子一致时, 计算出的相关性值明显大于其他的值。图 7.21 是表 7.2, 表 7.3 的图形表示, 我们可以得到检测阈值为 $T = 0.1$ 或 $\hat{T} = 0.1$。

绘制“种子—相关性值图”(SC 图) 是我们分析水印系统的一个重要手段。一般来说, 取尽可能多的种子作为横坐标, 绘制出的 SC 图越能说明问题。考虑到我们实验设备的有限, 这里只取 20 个种子。

由于我们的水印模板最初是由随机数种子控制的, 所以这个种子在一定程度上构成了 W-SVD 的惟一密钥(我们专门在后面有关于 W-SVD 密钥的讨论) 。那么从 SC 图上可以看到, 当检测水印时使用的种子与实际种子不一致时, 得到的相关性值是非常小的。只有检测时的种子与嵌入水印时的种子一致时, 才有可能检测出水印。下面我们归纳一下 SC 图的作用:

第一, 通过绘制 SC 图可以肯定我们使用的检测策略是正确的。因为只有当 SC 图出现明显的且惟一的峰值时, 才说明在该峰值对应的种子下嵌有数字水印。如果

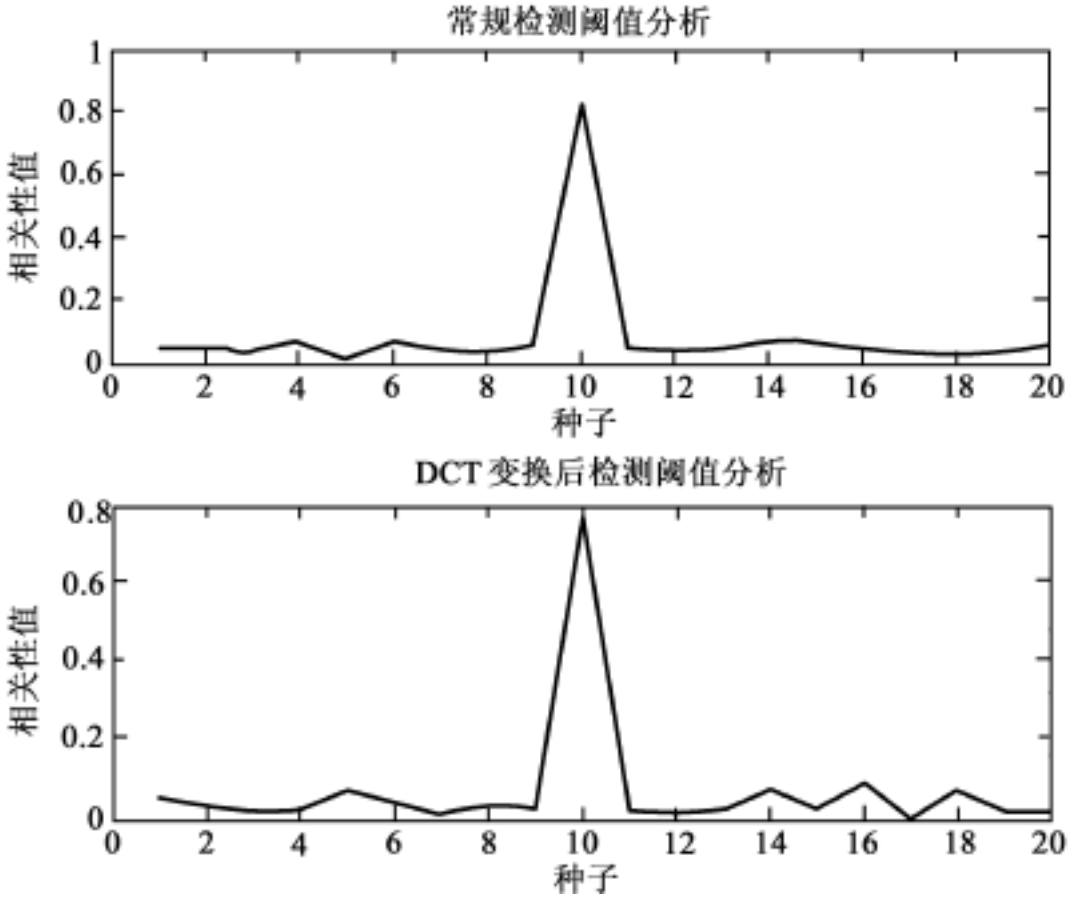


图 7.21　种子—相关性值图

对于一个确有水印的图像绘制的 SC 图没有明显的峰值，则表明所选择的检测手段是不理想的。

第二，SC 图可以给我们选择检测阈值提供依据。在数字水印的检测判定中，不可避免地会出现两类错误：虚警错误和漏警错误。前者是指将没有水印的图像判定为有水印，后者则是将有水印的图像判定为无水印。这在版权保护中是我们尽可能要回避的。当检测阈值选取过大时，就会造成漏警概率过大；而当检测阈值选取过小时，就会造成虚警概率过大。这都是不理想的。有关虚警错误和漏警错误的分析我们将在第九章专门谈到，这里仅提到它们的基本概念。SC 图可以直观地给我们提供检测阈值的恰当范围。在选择不同参数完成水印嵌入后，应该适当地通过大样本的 SC 图绘制，选择一个合适的检测阈值。例如，前面我们就确定了检测阈值为 0.1。

第三，检测阈值的作用除了判定一个作品有无水印外，在实验阶段还给我们对水印系统的鲁棒性测试提供了保证。在水印系统的鲁棒性测试中，我们看一个系统是不是鲁棒的，就是要看其水印图像在经过一定手段的攻击后，其所嵌入的水印能否还被检测出来，这时就需要检测阈值作为判决的根据，而检测阈值是与 SC 图有关系的。

第四，SC 图还可以给我们提供判定其他参数与水印鲁棒性的关系的依据。当我们改变一些参数（指除种子之外的参数）进行水印检测时，如果 SC 图在相应的种子上仍然出现明显的峰值，则说明这个（些）参数对水印鲁棒性和安全性的影响不大；SC 图越不理想，说明该参数对水印鲁棒性和安全性的影响越大。

第五, SC 图是攻击水印系统的有效手段。当我们不知道某一图像作品是否含有水印或不知其具体的水印种子时, 就可以通过绘制 SC 图来穷举密钥。一旦得到了水印种子, 我们就可以方便地攻击水印了。比如上面的例子, 假设我们不知道水印种子, 仅经过 20 次(甚至还可以更少) 穷举, 就得到了水印种子为 10。当然, 在实际中, 我们不可能取这么小的数作为种子, 但为了应付 SC 图穷举, 这一数至少也应该是一个足够大的数才行, 否则水印系统是不安全的。

7.3.4　W-SVD 水印系统性能分析

下面, 我们从水印鲁棒性、安全性和不可见性三个方面来分析 W-SVD 的性能。水印性能是与具体使用的参数不可分的。这里我们重点考虑的是以下五个参数:

① α;

② d/n;

③ 使用的小波;

④ 小波分解的尺度;

⑤ 随机数种子。

实验所使用的基本参数组为:

① db6 小波;

② 2 尺度分解;

③ α=0.1;

④ d/n = 0.99;

⑤ seed = 10。有变化的将特别注明。

(1) 算法各参数与水印生成的关系

从图 7.15 ~图 7.17 上我们可以分析得到:

① 参数 d/n 表示的是随机数矩阵替换原始低频系数正交矩阵 U, V的比例。显然应有 d/n∈[0, 1] 。考虑到水印应该有良好的惟一性以及水印所拥有的信息量应尽可能多, 所以 d/n 的取值应尽量大一些。同时, d/n 取值越小, 表示原图像特征系数被替换得越少, 水印形态图与原始图像越相像。

② 对于参数 α, 其值越大, 通过其相乘得到的随机对角矩阵 珔就越大, 继而得到水印模板的数据就越大, 对原始图像低频系数改变得就越多。

③ 与参数 d/n 和 α一样, 利用不同的小波基分解和同一小波不同尺度下的分解生成的水印在形态、与原始图像的相关性、信息容量和随机性等各方面也不同。图 7.22 是 db6 小波在不同尺度(1 ~6) 分解下得到的水印形态图。所谓‘ 水印形态图 ”是指以水印模板直接作为图像重构的低频系数而获得的重构图像。可以发现, 当分解尺度越大时, 水印形态与原始图像越相似, 也就是说实际上的水印模板的信息越来越少了。图 7.23 左侧是利用其他 dbN 小波(db1 ~db4) 在 2 尺度下分解生成的水印; 右侧给出了使用 db4 与 db6 小波生成的水印在各个对应的系数索引上频率系数的差值(W_{db4} - W_{db6})

的三维投影,显然二者的差异是巨大的。这里生成水印的其他参数均相同。

图 7.22　不同尺度下的水印形态

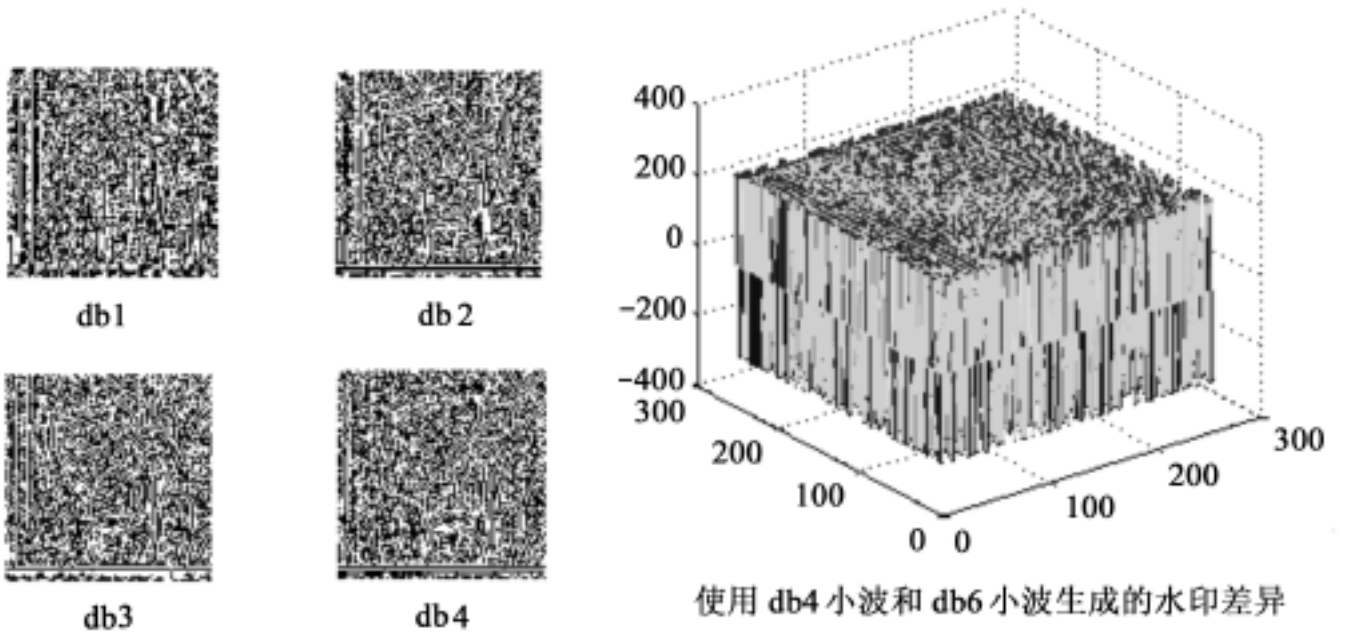


图 7.23　使用不同小波基得到的水印形态图

(2) 算法各参数与水印鲁棒性的关系

在鲁棒性测试中,我们将使用到 JPEG 压缩、图像模糊化、图像中值滤波、图像马赛克处理这 4 种攻击方法。具体攻击手段的原理以及相应分析曲线的绘制方法,我们将在第九章讨论,这里我们仅仅是用它们来判定 W-SVD 的性能。

1) α与鲁棒性的关系

我们先固定 d/n =0.99,让 α分别取 0.1 和 0.5 生成水印并嵌入图像。通过绘制 SC 图,我们将检测阈值固定在 0.1。

图 7.24 是“ W-SVD 抗 JPEG 压缩攻击曲线 ”。该图表明,当 α取不同值时,水印作品的抗 JPEG 压缩能力是不同的,也就是说鲁棒性是不同的。直观地看,尽管当 α取 0.1 和 0.5 生成的水印作品的抗 JPEG 压缩能力大体相当(在 5% 压缩率时能被检测出来) ,但 α=0.5 时的曲线美观一些,其与坐标轴共同围成的面积大一些。在这种情况下,发生检测错误的概率明显地低于 α=0.1 时的那一组曲线。

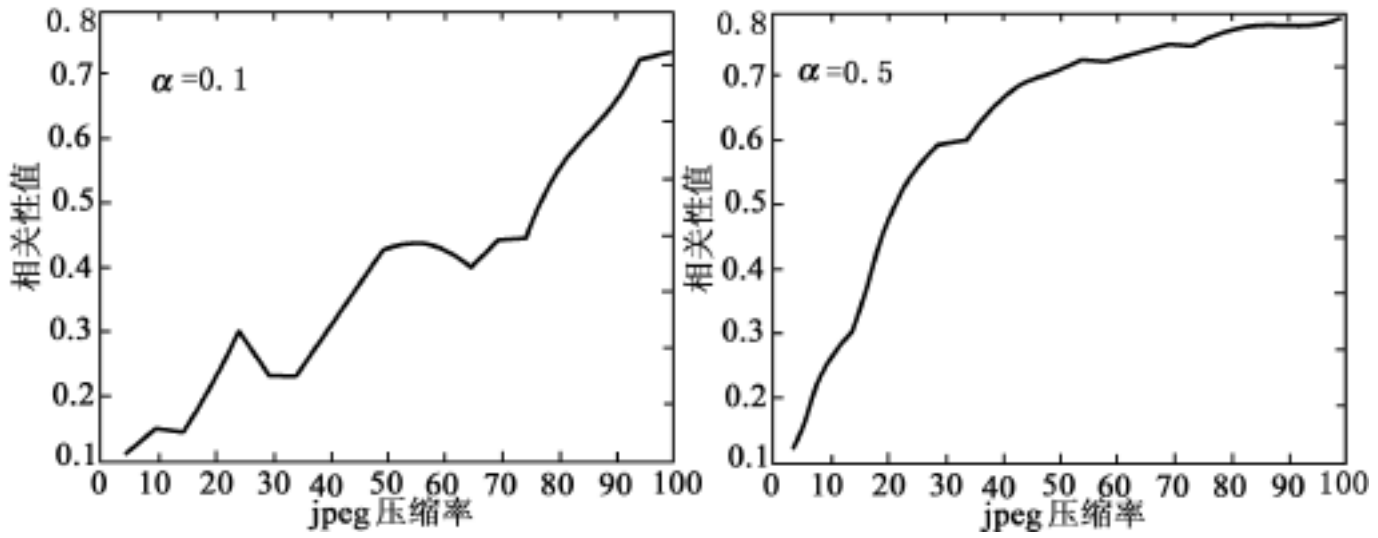


图 7.24　W-SVD 抗 JPEG 压缩攻击曲线

图 7.25 是“ W-SVD 抗模糊处理攻击曲线 ”。该图表明,当 α取不同值时,水印作品的抗模糊处理能力是不同的,也就是说鲁棒性是不同的。很明显,当 α=0.1 时,其水印图像在经过大约 3 次模糊处理后就无法检测出水印,也就意味着水印被攻击掉了。而当 α=0.5 时,这一指标明显提高。

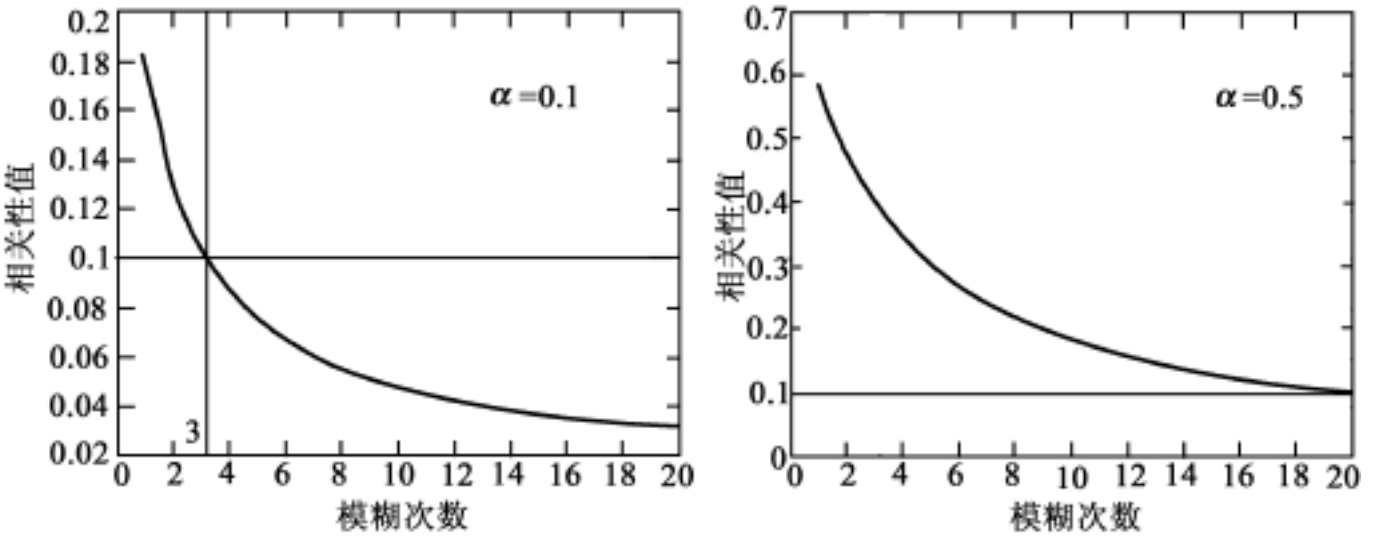


图 7.25　W-SVD 抗模糊处理攻击曲线

图 7.26 和图 7.27 分别是“ W-SVD 抗中值滤波攻击曲线 ”和“ W-SVD 抗马赛克处理攻击曲线 ”。当 α= 0.1 时, 其水印图像在经过 9× 9 的中值滤波或 6× 6 的马赛克处理后就无法检测出水印, 也就意味着水印被攻击掉了。而当 α= 0.5 时, 这些指标都明显提高。

综上所述, 参数 α是与水印鲁棒性密切相关的。事实上, α的取值略微改变一点, 水印性能都会发生很大变化。在实际中, 我们应该慎重选择 α。

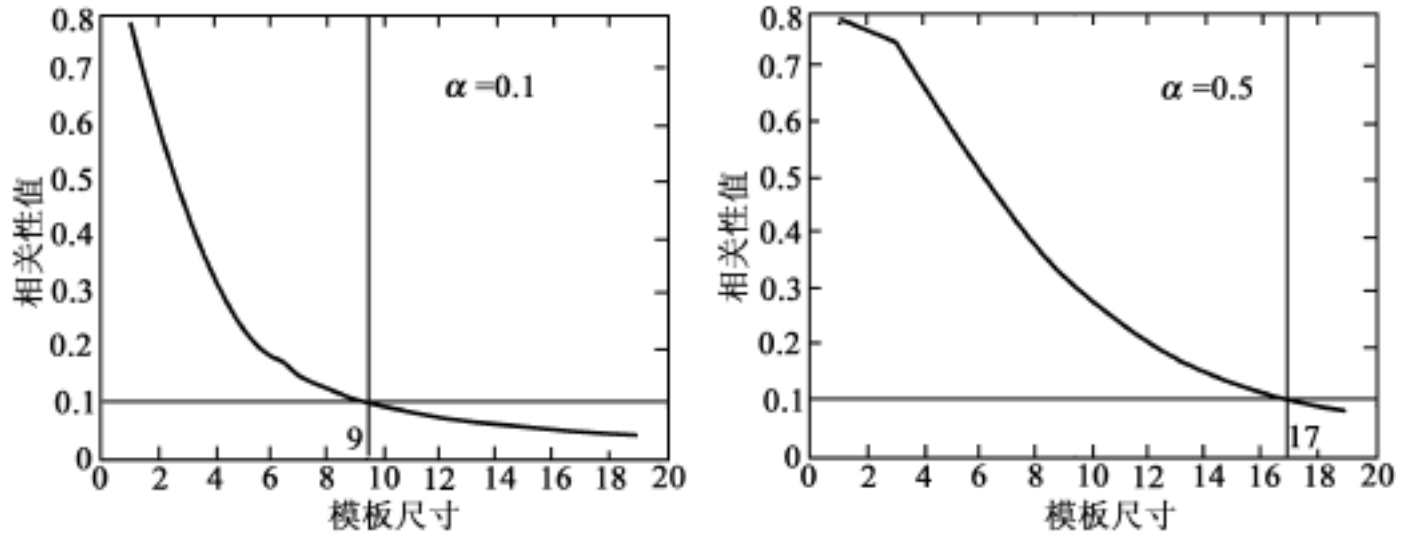


图 7.26　W-SVD 抗中值滤波攻击曲线

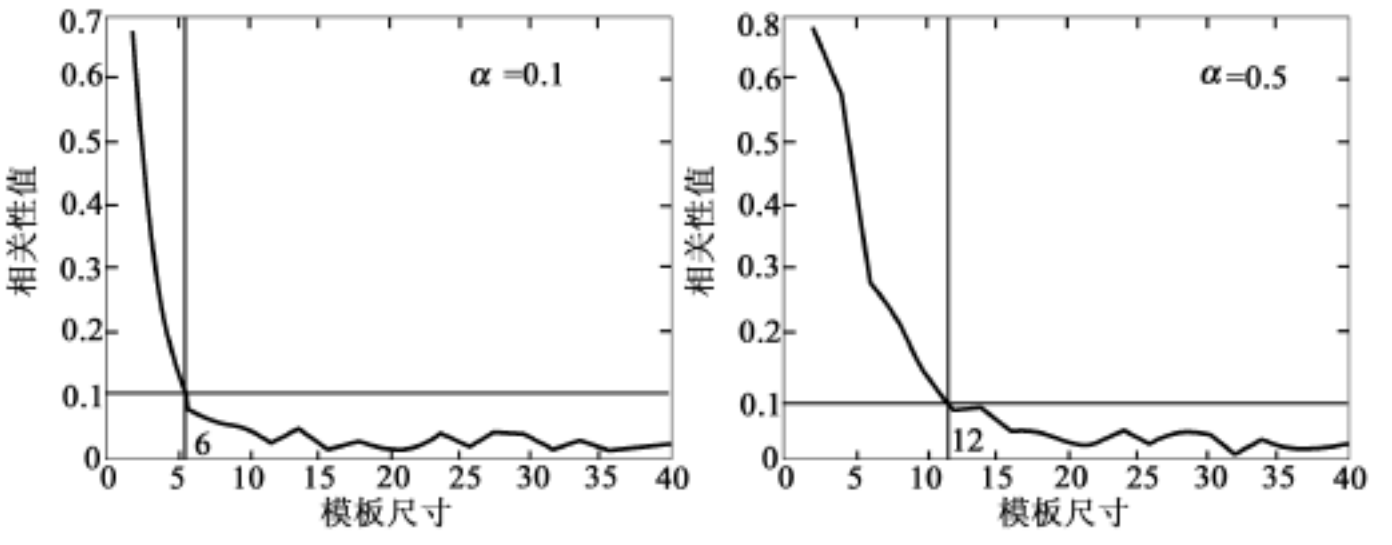


图 7.27　W-SVD 抗马赛克处理攻击曲线

2) d/n 与鲁棒性的关系

我们先通过 SC 图来说明参数 d/n 在水印检测中的影响。

图 7.28 是 d/n = 0.5 和 d/n = 0.1 时的 SC 图。通过它们和图 7.21(d/n = 0.99) 进行比较可以发现, 由于替换率变小, 检测已经很难实现。当 d/n = 0.1 时, 其 SC 图上已无法找到明显的峰值数据, 所以也就谈不上检测阈值了。因而, 该参数也是越大则水印检测的效果越好, 对检测越有利。

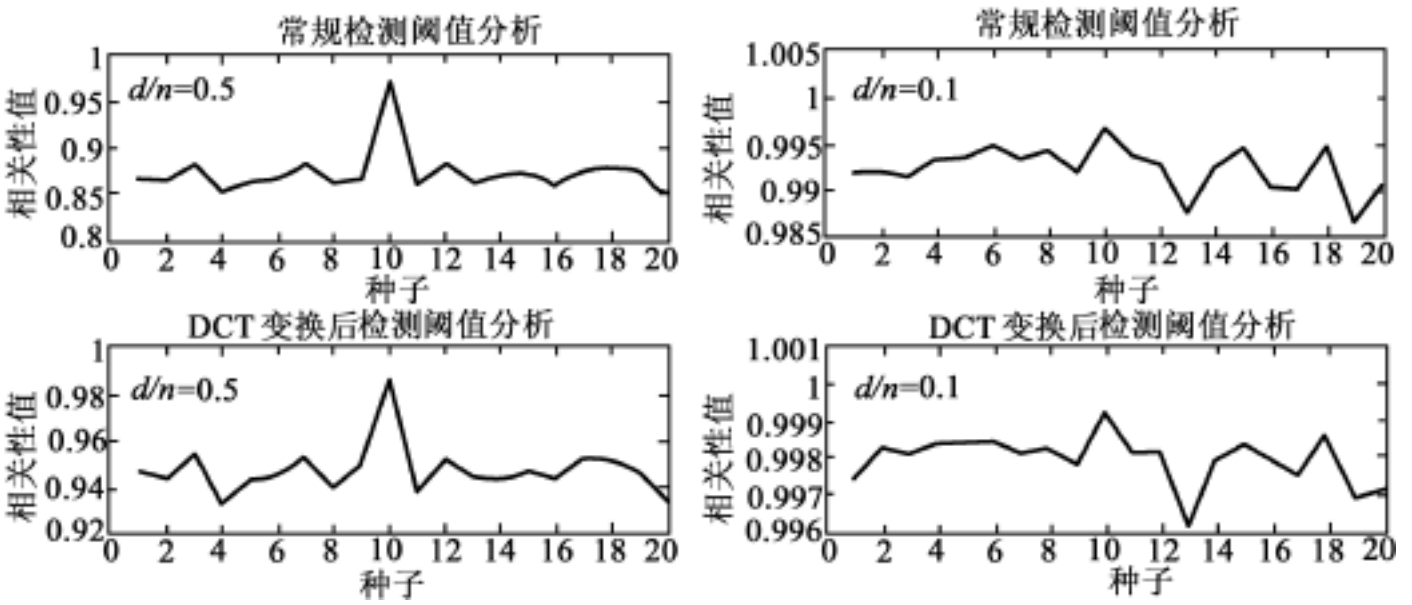


图 7.28　不同 d/n 下的 SC 图

对照图 7.28,由于 SC 图中峰值数据与一般数据的差越来越小,我们定义的检测阈值只有不断提高。对于 d/n =0.5 这种情况,我们就必须将检测阈值定为 0.9。同时,由于 SC 图峰值不明显,这时的检测将会造成一定的检测错误。我们仍利用前面鲁棒性检测的手段对水印进行攻击,图 7.29 是 d/n =0.5 时的“ W-SVD 抗 JPEG 压缩攻击曲线 ”(左图) 和“ W-SVD 抗中值滤波攻击曲线 ”(右图) 。可以发现,由于阈值很高,其水印作品的抗攻击能力明显下降了。

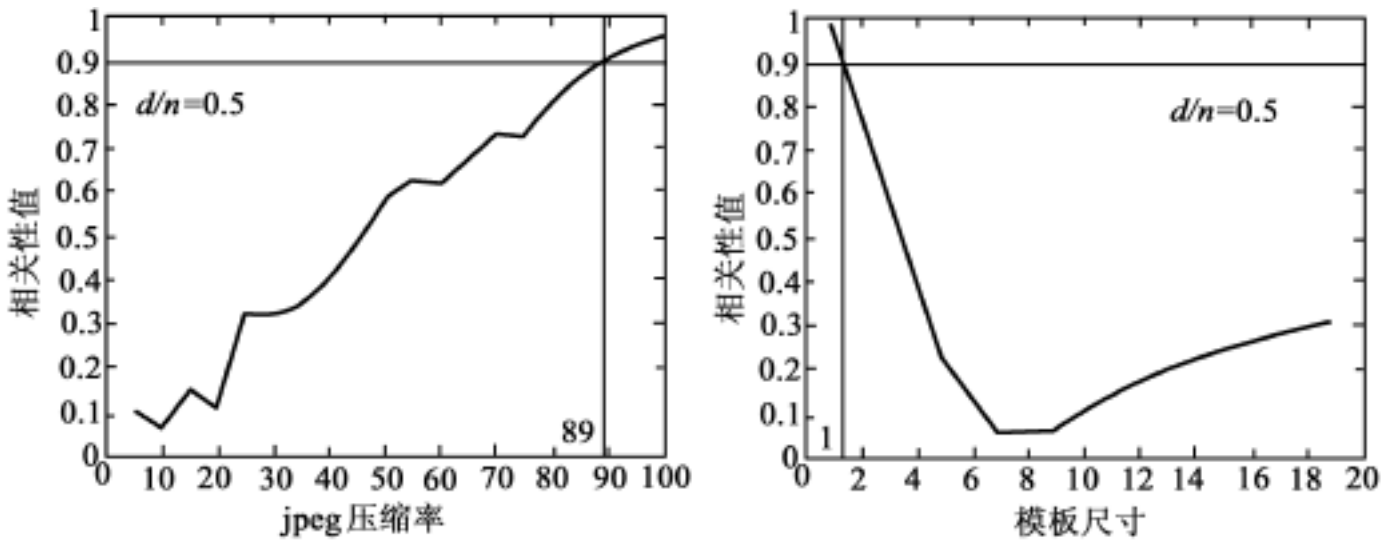


图 7.29　d/n = 0.5 时的抗攻击曲线

3) 小波分解尺度与鲁棒性的关系

通过绘制 SC 图发现,随着小波分解尺度的增加,水印的检测越来越困难,SC 图峰值在下降,检测难度和检测错误都在增加。结合二维小波分解的 Mollat 算法我们简单分析一下。

以 256× 256 图像为例,1 尺度分解得到的水印模板为 133× 133,2 尺度分解得

到的水印模板为 72× 72, 到 4 尺度时, 水印模板只有 26× 26 大小了。尽管在 level = 4 时添加的水印位于图像的非常低频的小波系数中, 但由于本身数据太少, 水印能量也远不如 level = 2 时的大, 所以造成了检测相关性值的降低。

图 7.31 是取分解尺度为 1, 其余参数等于基本参数时的一组 W-SVD 抗攻击曲线, 对照图 7.30, 取检测阈值仍为 0.1。

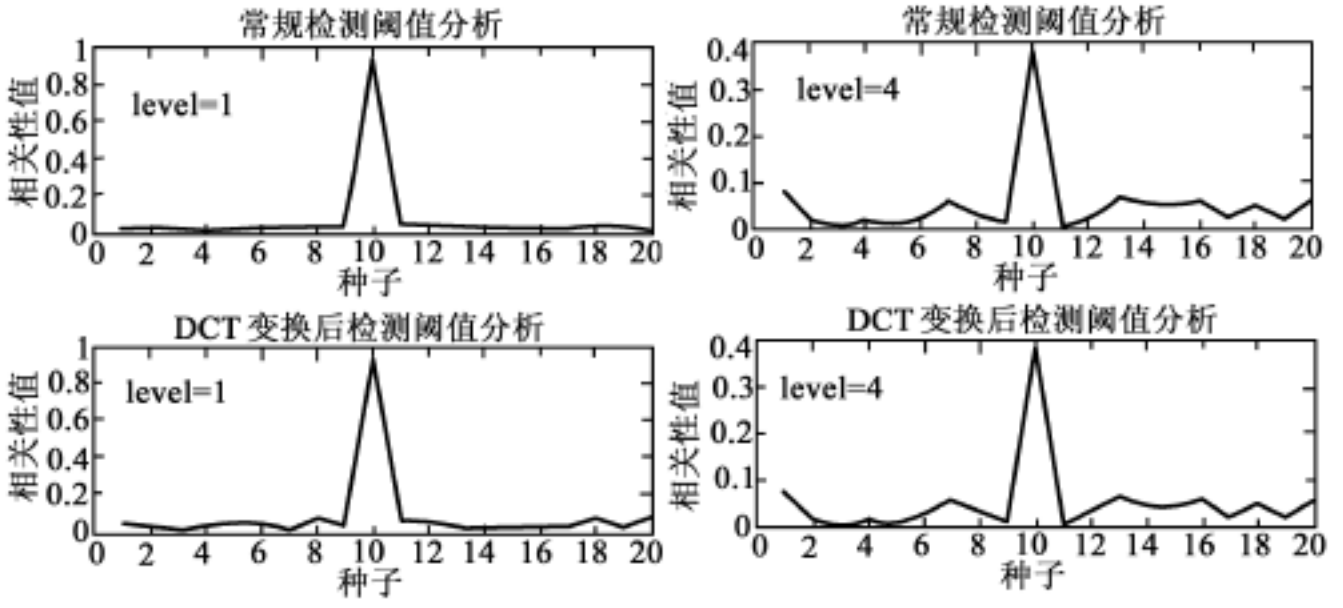


图 7.30　分解尺度与水印检测

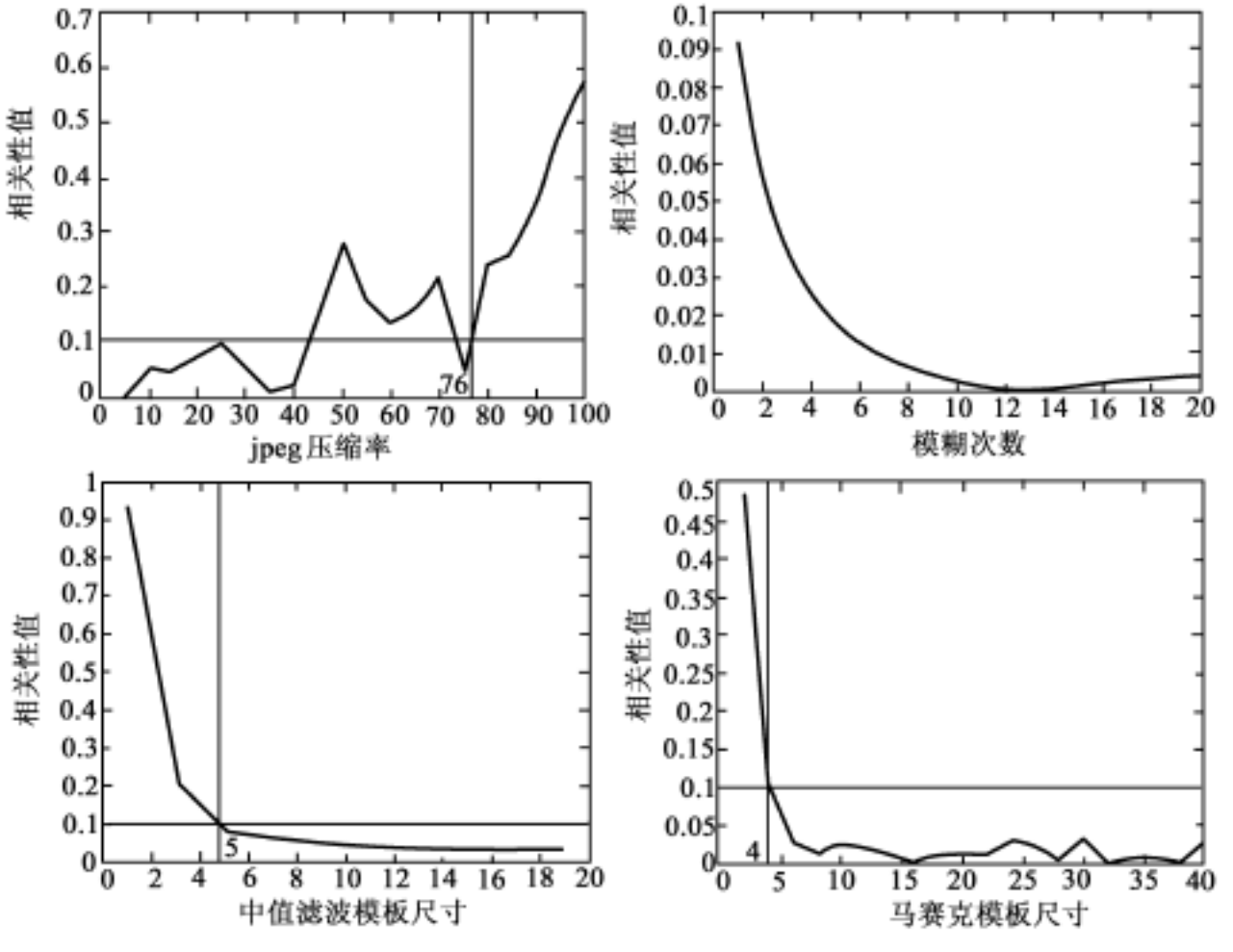


图 7.31　1 尺度下的水印性能

与图 7.24 ～图 7.27 对比不难发现,1 尺度下的水印性能不如 2 尺度下的水印健壮。这是因为水印嵌入的位置越是低频系数,水印越不易受到攻击。但结合前面的分析,水印模板本身也应该有一定的数据量。

① 小波分解的尺度越大,与之相关的水印信息越少,检测越困难。

② 小波分解的尺度越大,水印越能嵌入到图像的高能量部分(低频部分),水印鲁棒性越强。

综合以上考虑,我们认为在 W-SVD 中取小波分解尺度为 2 ～3 是合适的。

(3) 算法各参数与水印安全性的关系

Kerckhoffs 准则认为:一个安全保护系统的安全性不是建立在它的算法对于对手来说是保密的,而是应该建立在它所选择的密钥对于对手来说是保密的。数字水印的安全性也应该遵循这一原则。

将密钥控制引入水印系统是必不可少的。但一个很明显的问题就是:在水印系统中如何来划定算法与密钥的界线。对于这个问题,当前学者的看法也不一致,一部分学者认为对所要加入的信息进行加密,直接引入密码学中的密钥为密钥;另一部分学者则认为水印嵌入的位置和相关参数应该是密钥而不是算法的一部分。

以 W-SVD 为例,水印模板生成时的随机数种子毫无疑问是密钥的一部分,那么其他几个参数(小波、分解尺度、α和 d/n)能否或有无必要也构成密钥呢?

在实际生活中,版权持有者显然希望自己的作品所加入的水印是安全的,并且拥有较高的检测成功率;而对于盗版者来说,其意愿当然是相反的。设立数字水印技术的目的本身就是为了保护知识产权。站在社会公益的立场上,我们当然希望数字水印只能通过本人或可信任的第三方才能被检测到,从而增加盗版的难度。为了在技术上解决这个问题,提高数字水印的安全性,我们就要求生成水印的各项参数最好是保密的。以使用基本参数的水印((a) db6 小波,(b) 2 尺度分解,(c) α=0.1,(d) d/n=0.99,(e) seed=10)为检测对象,图 7.32 ～图 7.35 依次表明了在不知道小波基、分解尺度、α和 d/n 的情况下随意定义这些参数对水印进行穷举检测绘制的 SC 图,显然除了在不知道参数 α的情况下仍可以检测以外,其余的检测均是失败的。

如果把这些参数都看做为广义的密钥,密钥空间明显扩大了。以攻击者穷举攻击为例,穷举攻击的难度明显增加了。从广义的密钥角度去理解,小波基、分解尺度、α、d/n、随机序列种子都可以看做是密钥的一部分。这里我们的观点是与上述后一部分学者一致的,因为这在实际中是有利于保护水印作品安全性的。

(4) 算法各参数与水印不可见性的关系

我们将在第八章进行有关水印不可见性的详细讨论。这里,我们仅用峰值信噪比(Peak Signal to Noise Ratio, PSNR)来衡量一下算法各参数对水印不可见性的影响。与前面一样,以下分析中未明确说明的参数都是基本参数。

1) α与不可见性的关系

直接观察图 7.15 ～图 7.17 不难看出,随着水印强度因子 α的不断增大,水印对

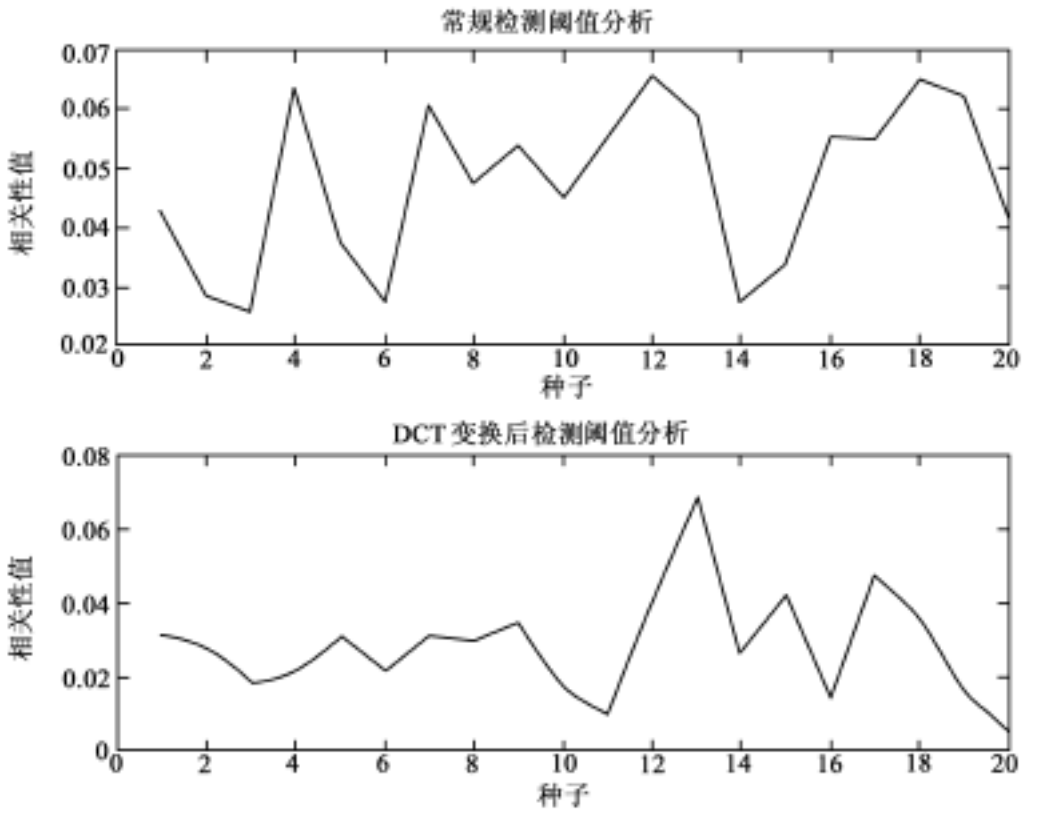


图 7.32　用 db5 小波检测水印

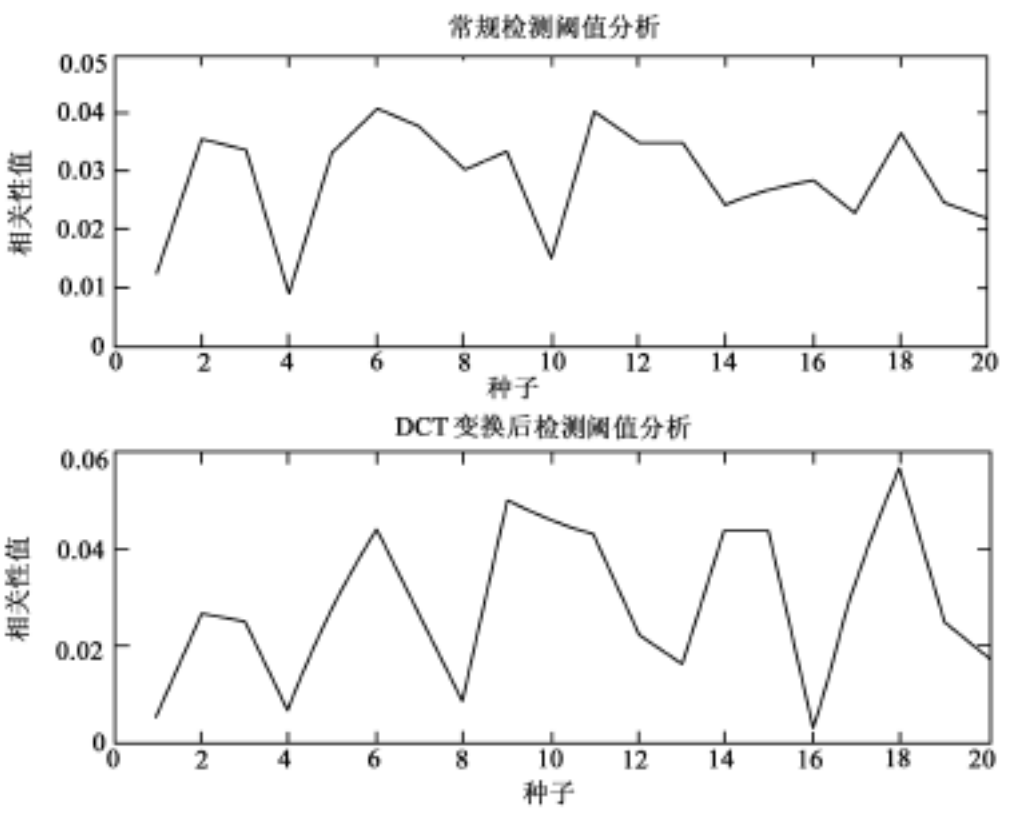


图 7.33　用 1 尺度检测水印

原始图像的破坏也越来越大, 水印的不可见性降低。表 7.4 列出了 α从 0.1 ~0.5 时水印图像的 PSNR。

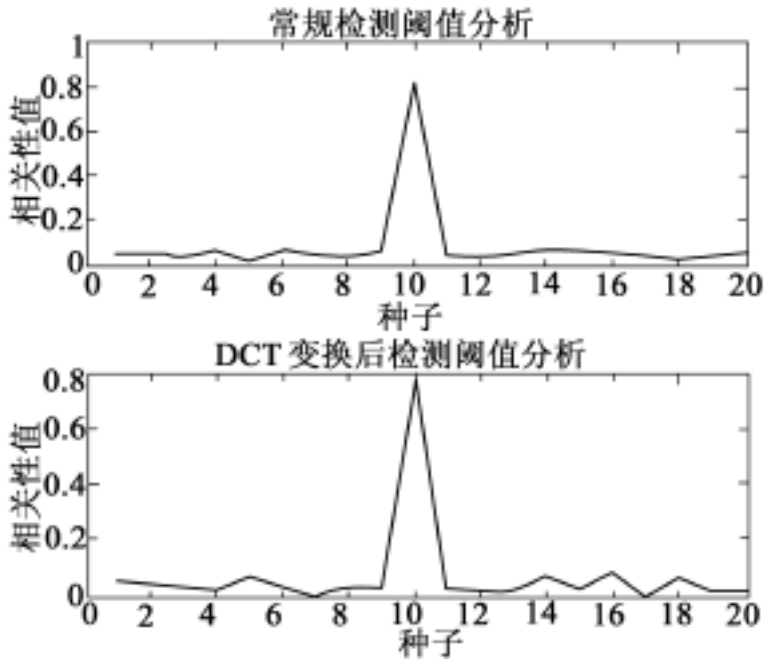


图 7.34　用 α = 0.5 检测水印

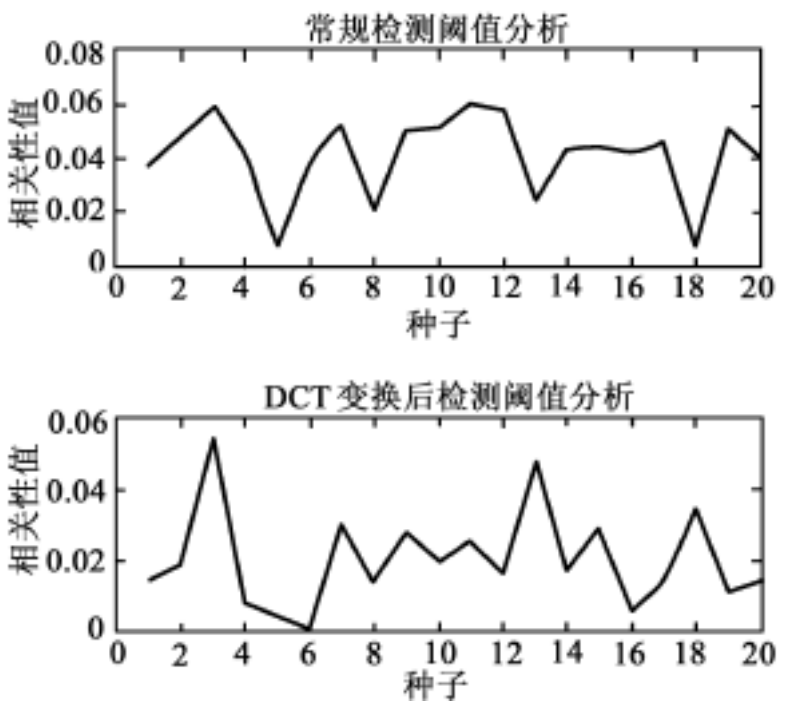


图 7.35　用 d/n = 0.95 检测水印

表 7.4　　α与水印不可见性

水印强度	PSNR(dB)
α=0.1	55.422
α=0.2	49.1822
α=0.3	45.4978
α=0.4	43.2266
α=0.5	41.4967

2) d/n 与不可见性的关系

表 7.5 列出了 d/n =0.9 ~d/n =0.5 时的水印图像 PSNR。

表 7.5　　d/n 与水印不可见性

替换比例	PSNR(dB)
d/n =0.9	54.1509
d/n =0.8	54.1141
d/n =0.7	54.2718
d/n =0.6	54.5203
d/n =0.5	54.1651

可以看到, d/n 对水印不可见性影响不大。

3) 分解尺度与不可见性的关系

表 7.6 列出了 level =1 ~level =5 时的水印图像 PSNR。

表 7.6　　分解尺度与水印不可见性

小波分解尺度	PSNR(dB)
level =1	57.6184
level =2	55.422
level =3	54.0429
level =4	53.8587
level =5	51.3474

可以看到, 随着小波分解尺度的增加, 水印更集中在图像能量高的部分, 对图像的感知质量造成的影响越来越大。

7.4　混沌细胞自动机数字水印

7.4.1　细胞自动机与水印生成

在计算机学科内, 我们把能和其他细胞相互作用具有相同的可计算能力的细胞数组称为细胞自动机。这里的细胞一般是指一些特定区域的特定数据。细胞自动机能根据一定的算法和规则设计出相应的邻居关系, 从而根据邻居的当前状态来改变自己的状态。将细胞自动机的思想运用到数字水印的生成策略上, 可以得到很好的

水印模板。

我们以投票规则(Vote Rule) 为例, 简要地说明利用细胞自动机生成水印的方法。水印模板的生成分为以下几个步骤:

① 产生随机数模板。

② 根据一定的判定规则, 将随机矩阵转化为二值矩阵。

③ 将二值矩阵代入到细胞自动机, 得到经细胞自动机处理的水印模板“凝聚模式”。

④ 将“凝聚模式”的模板经过平滑处理, 得到最终的水印模板。

编写函数 cellauto. m 完成水印生成的任务。函数代码如下:

```
% 文件名: cellauto. m
% 程序员: 李巍
% 编写时间: 2004. 3. 20
% 函数功能: 这是一个细胞自动机及相应的 vote 和 smooth 函数
% 输入格式举例: watermark = cellauto( 72, 72, 1983, 20)
% 参数说明:
% row, col 为要求得到的水印模板大小
% seed 为随机数种子
% do_num 为细胞自动机处理次数
function watermark = cellauto( row, col, seed, do_num)
% 生成随机模板
rand( 'seed', seed) ;
chaoticrand = rand( row, col) > 0. 5; % 转二值矩阵
chaotic = chaoticrand;
% 扩大边界等待处理
temp = zeros( row + 2, col + 2) ;
temp( 2: row + 1, 2: col + 1) = chaotic;
% 细胞自动机处理
for i = 1: do_num
    % 边界补充
    temp( 1, 2: col + 1) = temp( row + 1, 2: col + 1) ;
    temp( row + 2, 2: col + 1) = temp( 2, 2: col + 1) ;
    temp( 2: row + 1, 1) = temp( 2: row + 1, col + 1) ;
    temp( 2: row + 1, col + 2) = temp( 2: row + 1, 2) ;
    temp( 1, 1) = temp( row + 1, col + 1) ;
    temp( row + 2, col + 2) = temp( 2, 2) ;
    temp( 1, col + 2) = temp( row + 1, 2) ;
```

```
        temp( row + 2, 1)  = temp( 2, col + 1) ;
        % vote1 规则
        cell1 = temp( 1: row, 1: col) ;
        cell2 = temp( 1: row, 2: col + 1) ;
        cell3 = temp( 1: row, 3: col + 2) ;
        cell4 = temp( 2: row + 1, 1: col) ;
        cell5 = temp( 2: row + 1, 2: col + 1) ;
        cell6 = temp( 2: row + 1, 3: col + 2) ;
        cell7 = temp( 3: row + 2, 1: col) ;
        cell8 = temp( 3: row + 2, 2: col + 1) ;
        cell9 = temp( 3: row + 2, 3: col + 2) ;
        temp( 2: row + 1, 2: col + 1)  = ( cell1 + cell2 + cell3 + cell4 + cell5 + cell6 +
cell7 + cell8 + cell9)  > 4;
    end
    chaoticcell = temp( 2: row + 1, 2: col + 1) ;
    % 平滑处理
    chaotic2 = chaoticcell;
    avg = fspecial( 'average',3) ;
    for j = 1: do_num
        chaotic2 = filter2( avg, chaotic2) ;
    end
    scale = max( max( chaotic2) ) ;
    chaotic2 = chaotic2 /scale;
    % 水印生成
    watermark = ( chaotic2 - mean2( chaotic2) *  ones( row, col) ) ;
    subplot( 131) ; imshow( chaoticrand) ; title( '随机模式') ;
    subplot( 132) ; imshow( chaoticcell) ; title( '细胞模式') ;
    subplot( 133) ; imshow( watermark) ; title( '平滑模式( 水印)') ;
```

图 7.36 是采用上述自动机完成 20 次运算后得到的水印模板各阶段形态。图 7.37 是得到的水印的矩阵表示形式。

下面我们就水印生成的几个问题作一个具体说明。

(1) 投票规则的运作

将前面程序中使用的细胞自动机运作算法记为 vote1, 对应该程序, 我们以一个 5 × 5 的矩阵 A 来具体观察一下该细胞自动机运作一次的情况。

矩阵 A 如下:

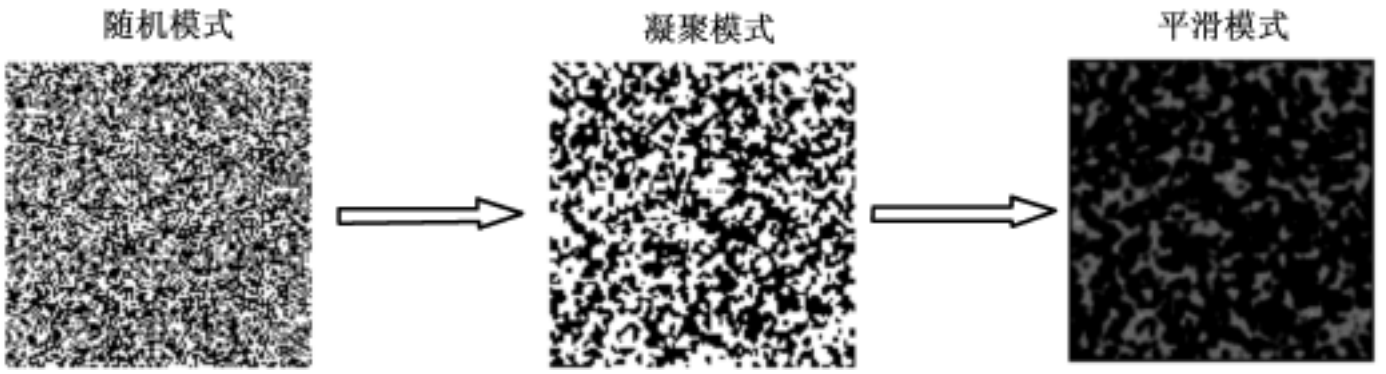


图 7.36　细胞自动机水印各阶段形态

Array Editor: watermark

File Edit View Web Window Help

Numeric format: shortG　Size: 256 by 256

	1	2	3	4	5	6
1	-0.45955	-0.44399	-0.44222	-0.45107	-0.46367	-0.47429
2	-0.44131	-0.41454	-0.41145	-0.42658	-0.44818	-0.46632
3	-0.43355	-0.40186	-0.39789	-0.41534	-0.44046	-0.4615
4	-0.43384	-0.4017	-0.39649	-0.41244	-0.43638	-0.45641
5	-0.43473	-0.40152	-0.39299	-0.40489	-0.42562	-0.44339
6	-0.42731	-0.38614	-0.36997	-0.37637	-0.39475	-0.41195
7	-0.40574	-0.34548	-0.31533	-0.31471	-0.33255	-0.35219
8	-0.37082	-0.28064	-0.22961	-0.21898	-0.23611	-0.25963
9	-0.33029	-0.20475	-0.12765	-0.10213	-0.11448	-0.13955

图 7.37　细胞自动机水印模板

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

生成一个临时矩阵 temp, temp 的中心部分就是原始矩阵 A。

$$temp = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

将 temp 的边界进行补充, 得到如下矩阵:

$$\text{规定 temp(边界补充)} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

以下是分别从经过边界补充的 temp 矩阵中截取的 9 个子矩阵:

$$\text{cell1} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \text{cell2} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\text{cell3} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \text{cell4} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\text{cell5} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \text{cell6} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

$$\text{cell7} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad \text{cell8} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{cell9} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

将这 9 个子矩阵相加, 得到一个和矩阵:

$$\text{summation} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 4 & 2 & 0 & 2 & 4 & 1 \\ 0 & 2 & 2 & 1 & 2 & 2 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 2 & 2 & 1 & 2 & 2 & 0 \\ 1 & 4 & 2 & 0 & 2 & 4 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

将这个和矩阵的各元大于 4 的写成 1, 小于等于 4 的写成 0, 就完成了 1 次细胞自动处理。

事实上, 细胞自动机的运作规则有很多种, 比如 Life, Brain, Aurora 和 Vote 等等。仅就投票规则而言, 除了我们上面提供的一种方式外, 我们再给出一种常用的基于投票原理的细胞自动机运作算法(Vote2) , 供大家参考。

算法 Vote2 以一个计数值 p 记录二值矩阵中每 3× 3 邻域内 1 的个数, 如果 p 小于 5, 则将这个 3× 3 邻域中心的数值设置为 0, 否则设置为 1。假设二值矩阵 old 为 n× n, 则相应的经过细胞自动机 Vote2 处理得到新二值矩阵的每一个数值 new(i, j) 。用伪 C 代码描述如下:

```
for ( i = 2; i <= n - 1; i ++ )
for( j = 2; j <= n - 1; j ++ )
   {
   p = 0;
   for( ii = i - 1; ii <= i + 1 ; ii ++ )     // 考察 3× 3 邻域内 1 的个数
      for( jj = j - 1; jj <= j + 1 , j ++ )
         {
         if( old( ii, jj)  == 1)
            p ++ ;
         {
   if( p < 5)     // 根据 p 的投票数确定细胞自动机输出
      new( i, j)  = 0;
   else
      new( i, j)  = 1;
}
```

当然, 就投票规则而言, 大家也可以自己定义新的算法, 只要能保证随机模式下的水印模板有效的转变为凝聚模式下的水印模板就行了。

(2) 平滑处理的方法

我们这里采用的平滑处理方法是图像的均值滤波。有关均值滤波的原理我们将在第九章介绍。

MATLAB 中的 fspecial. m 函数可以根据生成的参数得到一个滤波的冲击响应核, 其调用方式为:

h = fspecial('average', hsize)

其中, 'average 表示均值滤波, hsize 是冲击响应模板的大小, 默认值为 3× 3。

(3) 混沌细胞自动机

所谓"混沌细胞自动机"实际上就是指水印最初的随机数矩阵不是用一般的随机数发生器获得的, 而是用具有混沌特性的随机数发生器获得的。我们在第二章专门介绍了具有混沌特性的随机数发生器。以混合光学双稳模型为例(函数 randCS. m), 将生成的随机数模板按照特定的判定规则调整为二值模板, 接下来的步骤就与前面使用线性同余发生器的步骤一样了。这种特定的判定规则可以描述为:

```
if ( x_i > = 2A/3)
        S_i = 1;
else
        S_i = 0;
```

其中, A 是混沌模型的既定参数。

7.4.2 水印的嵌入和检测策略

单纯获得细胞自动机水印是不够的。接下来的问题就是: 水印往哪嵌入。一般来说, 这时的水印可以往 DCT 系数或小波低频系数中嵌入。下面我们分别来简单实现这些方法。

(1) DCT 域嵌入策略

正如前面所述, 本水印算法的水印生成策略和水印嵌入策略是不相关联的。嵌入的方式为:

E(image) = IDCT(DCT(image) + δ* watermark.)

考虑到原始图像 DCT 系数的大小, 我们必须用参数 δ对水印强度进行控制。图 7.38 和图 7.39 分别是 δ=0.01 和 δ=5 时嵌入水印后的 lenna 图像。可以发现, 当 δ较大时, 原图像的低频部分首先被破坏了。

下面给出 DCT 域嵌入此水印的函数, 函数代码如下:

```
% 文件名: dctwatermark. m
% 程序员: 李巍
% 编写时间: 2004. 3. 20
% 函数功能: 本函数是一个嵌入水印的函数
% 输入格式举例:
% [ corr_coef, corr_DCTcoef] = dctwatermark( 'lenna. jpg', 'lenna1. jpg', jpg, 1024, 3,0.2)
% 参数说明:
```

```
% original 为原始图像文件名
% goal 为嵌入水印图像
% permission 为图像文件格式
% seed 为随机序列种子
% alpha 是尺度参数
% do_num 参数是进行投票选择的次数
function[ watermark, datared, datadct, datared2 ]  = dctwatermark ( orignal, goal, permission, seed, do_num, alpha)
data = imread( orignal, permission) ;
data = double( data) /255;
datared = data(:,:, 1) ;
[ row, col]  = size( datared) ;
datadct = dct2( datared) ;
% 调用函数 cellauto
[ chaoticrand, chaoticcell, watermark]  = cellauto( row, col, seed, do_num) ;
dataadd = datadct + alpha*  watermark;
datared2 = idct2( dataadd) ;
data(:,:, 1)  = datared2;
% 显示结果
subplot( 131) ; imshow( datared2) ; title( 'R 层图像') ;
% subplot( 132) ; imshow( data) ; title( '加入水印后的图像')
imwrite( data, goal, permission) ;
```

图 7.38　δ = 0.01 的水印图像

图 7.39　δ = 5 的水印图像

我们采用类似 W-SVD 检测算法对水印进行检测, 函数代码如下:

```
% 文件名: wavedetect2. m
```

```
% 程序员: 李巍
% 编写时间: 2004. 3. 20
% 函数功能: 本函数是一个检测 DCT 水印的函数
% 输入格式举例:
% corr_coef = wavedetect2( 'lenna1. jpg', jpg, 'lenna. jpg', jpg, 1024, 3, 0. 1)
% 参数说明:
% test 为待检测的 RGB 图像文件名
% permission1 为待检测的 RGB 图像文件格式
% original 为原始图像文件名
% permission2 为原始图像文件格式
% seed 为随机序列种子
% do_num 参数是进行投票选择的次数
% alpha 是尺度参数
% corr_coef 是检测相关值
function corr_coef = wavedetect2( test, permission1, original, permission2, seed, do_num, alpha)
dataoriginal = imread( original, permission2) ;
datatest = imread( test, permission1) ;
dataoriginal = dataoriginal( :, :, 1) ;
[ m, n] = size( dataoriginal) ;
datatest = datatest( :, :, 1) ;
% 提取加有水印的图像的 DCT 系数
waterdct = dct2( datatest) ;
% 提取原始图像的 DCT 系数
waterdcto = dct2( dataoriginal) ;
% 生成两种水印
realwatermark = cellauto( m, n, seed, do_num) ;
testwatermark = ( waterdct-waterdcto) / alpha;
% 计算相关性值
corr_coef = trace( realwatermark' * testwatermark) / ( norm( realwatermark, 'fro') * norm( testwatermark, 'fro') ) ;
```

图 7. 40 是水印检测的 SC 曲线。

最后, 我们简单地考察一下参数 δ与水印鲁棒性的关系。我们采用的攻击手段是 JPEG 压缩。由于我们是将水印直接加往 DCT 全部系数中, 可以想像, 该算法抵抗 JPEG 压缩的能力是很低的。图 7. 41 和图 7. 42 证实了我们的推断。虽然图 7. 42 中显示可以容忍的压缩率为 63%, 但事实上当 δ= 0. 1 时图像已经被所嵌入的水印部

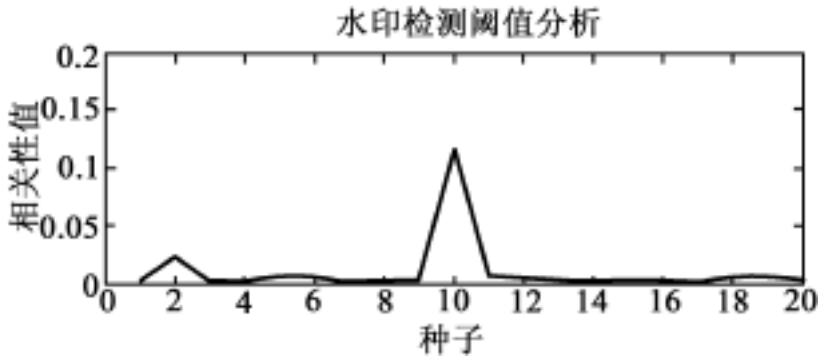


图 7.40　水印检测的 SC 曲线

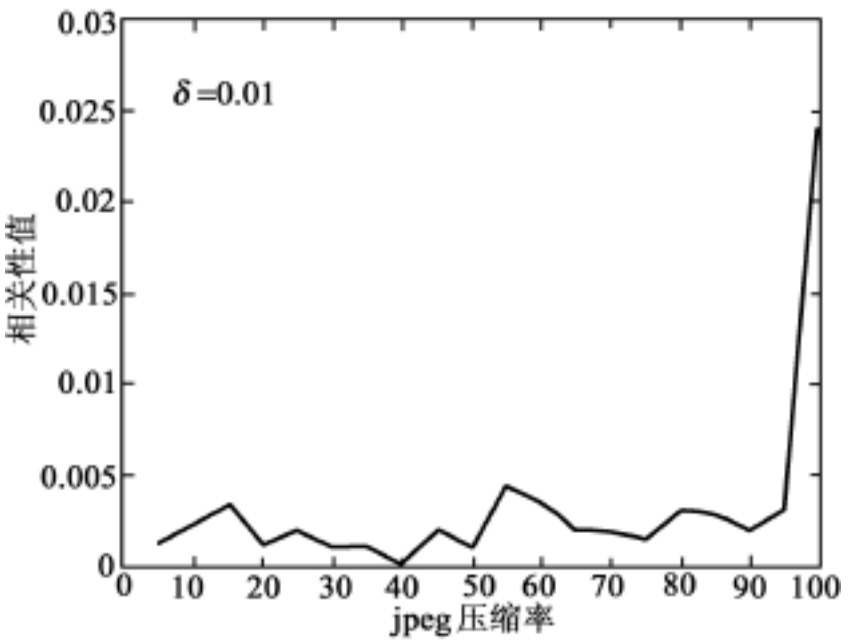


图 7.41　水印抗 JPEG 压缩曲线 1

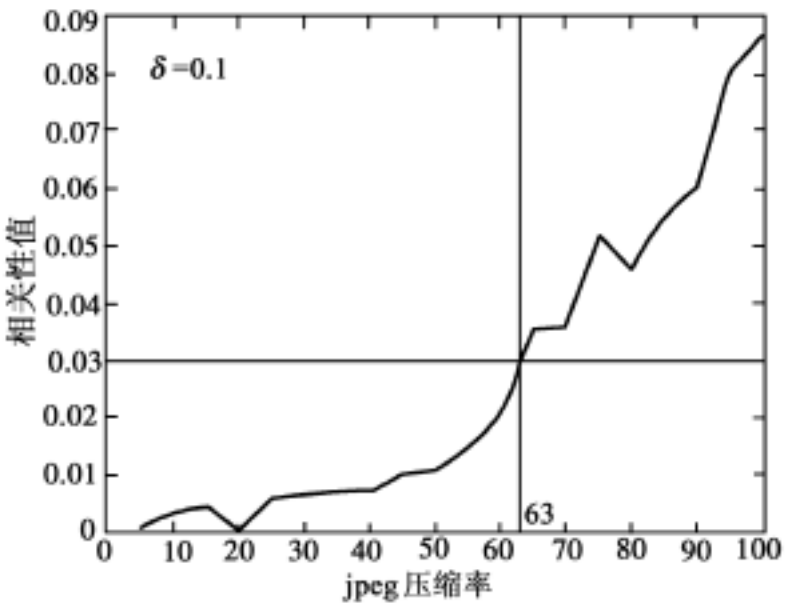


图 7.42　水印抗 JPEG 压缩曲线 2

分破坏了(与图 7.39 情况相似)。所以,只能认为 δ值与 W-SVD 算法中的 α一样,起

了一个水印强度因子的作用,它的具体取值读者可自己去分析。

(2) DWT 域嵌入策略

细胞自动机水印还可以往图像小波分解的低频系数中嵌入。嵌入的方法与前面的 DCT 方法和 W-SVD 都有类似的地方。在 W-SVD 中,水印模板是通过 SVD 分解替换构造的,这里将这些部分统一用细胞自动机水印替换就可以了,其余的步骤大体相同。读者可自行完成并与 W-SVD 对照,分析这种水印算法的性能。

7.5 数字水印的几何解释

这一节我们主要从几何模型上认识一下数字水印。假设有一个 N 维空间,这个空间中的每一个元素代表一件数字作品,这个空间就被称为媒体空间。维度 N 表示原始作品所需的样本数。对于单色图像,N 是像素点总数。对于 RGB 图像,N 是像素点总数的 3 倍。对于连续的时序内容(如音频或视频),假设水印嵌入到信号的固定长度的一段内,并且内容已经过时间采样,因此,对音频来说,N 是段内的样本数,对视频来说,N 是段内的帧数与每帧像素点数的乘积(如果是彩色的还要再乘以 3)。媒体空间在一定方式下的投影或变形得到的子空间又被称为标志空间。

以 8 比特灰度图像为例,其每点像素值都映射为 0 ~255 之间的一个整数。这表明实际作品的集合是有限的(尽管非常大),它们在媒介空间构成一个离散格点空间。格点之间或区域以外的点不代表有效的数字作品。因为量化间隔往往很小,区域范围往往很大,通常忽略媒介空间的离散性,把它看做是连续的(也就是说,即使是那些不在格点上的点,也都代表了可实现的作品)。

(1) 未加水印作品的分布

在不同作品中嵌入或检测到水印的可能性是不同的。诸如音乐和自然图像等作品的内容具有独特的统计分布,我们必须把这些分布考虑进去。

当估计水印系统的虚警率和有效性等特性时,建立作品内容的先验分布模型对于研究系统很重要。我们既可以用格点集上的概率分布来表示数字作品的分布,也可以用媒介空间上所有点的概率密度函数来表示此分布。

有文献指出,对于未加水印作品格点集的概率分布,其中最简单的模型是假设作品满足椭圆高斯分布。而对大部分媒介来说,更精确的模型满足拉普拉斯或广义高斯分布。事实上,未加水印作品格点集的概率分布取决于具体应用。不同种类作品具有的多种不同分布会造成某些问题。如果我们根据某种分布来估计水印检测器的虚警率,然后在另外一个应用领域使用该检测器,而这个应用中未加水印的作品服从的分布又不同,那么估计值就会不精确。尤其对于那些严格要求虚警率的应用(如拷贝控制),问题将会非常严重。

(2) 可接受保真度的区域

想像有这么一幅图像 Po,改变其中一个像素点的值,哪怕只是亮度增加 1 或减

少 1, 也会在媒介空间形成一个新的向量。然而光凭感觉, 新图像和原图像是无法区分的。显然有很多这样的图像, 我们可以想像在 Po 周围有那么一个区域, 其中每个向量都表示一幅和原作品凭感觉无法区分的新图像。媒介空间内的这类区域叫做可接受保真度的区域, 区域中每个向量都表示和载体作品无法区分的新作品。

(3) 检测区域

对应于给定信息 m 和水印密钥 k, 检测区域是媒介空间中的一组作品, 检测器解码输出的结果表明这组作品包含水印信息 m。和可接受保真度的区域相似, 检测区域通常是由检测器输入和嵌入信息 m 的模板之间的相关性值以及检测阈值定义的。

在 7.2 节的水印基本模型实验算法中, 检测度量是线性相关性值 $Z_{lp}(p, wr)$。为了刻画检测区域的形状, 首先看到接收的作品和参考模板之间的线性相关 $p \cdot wr/N$ 等于其的欧式长度和两者之间夹角余弦的乘积再除以 N。因为 wr 的欧式长度是常数, 故该度量等价于 N 维向量 p 在 N 维向量 wr 上的正交投影。该度量值超过阈值 τ_p 的所有点都位于某个与 wr 垂直的平面的一侧。这些点构成的半空间表示 m = 1 的检测区域。同样, m = 0 的检测区域在由 $-\tau_p$ 决定的平面一侧。

如果水印作品在媒介空间中位于可接受保真度的区域和检测区域的交叠部分, 那么该水印嵌入就是成功的。如图 7.43 所示, 图中通过误差图像与原始图像的均方差来确定可接受保真度的区域, 通过接收的作品和参考模板 wr 之间的线性相关检测阈值确定检测区域。图中显示的是媒介空间的二维截面, 其中包括两个向量 Po 和 wr。经过调整, 向量 wr 正好位于横轴上, 因为高维空间内随机选择的向量很可能靠近于 wr 正交的方向, 所以 Po 离纵轴很近。可接受保真度的区域是 N 维球体, 它在二维截面上的投影是一个填充圆。检测区域和媒介空间其他区域的分割平面在二维截面上的投影是一条与 wr 垂直的直线。所有既在可接受保真度的圆形区域内, 同时又位于检测区域边界右侧的点都对应了 Po 的不同版本。这些点在可接受保真度的范围内, 并且检测器也能正确地检测出其中的水印信息。也就是说, 这些点代表着那些已经成功地嵌入了水印信息的作品。

(4) 嵌入分布和嵌入区域

水印嵌入器相当于一个函数, 它把作品、信息或密钥映射为新作品。这个函数通常是确定性的, 给定原始作品 Po、信息 m 和密钥 k, 嵌入器总是输出相同的加了水印的作品 Pw。水印嵌入器实际上包括了水印生成和嵌入两方面的策略。因为原始作品由未加水印作品的分布随机决定, 所以嵌入器的输出结果也可以看成是随机的。嵌入器输出特定作品 Pw 的概率等于从未加水印作品的分布中得到 Po 的概率, 其中 Pw 是 Po 输入嵌入器后的结果。如果几幅未加水印的作品都可以映射为 Pw, 则嵌入器输出 Pw 的概率等于这些未加水印作品的概率之和。我们把 Pw 的概率分布叫做嵌入分布。

通常的水印嵌入策略有以下两种:

一种策略是在一些嵌入策略定义的嵌入分布中, 每幅可能的图像, 甚至包括那些

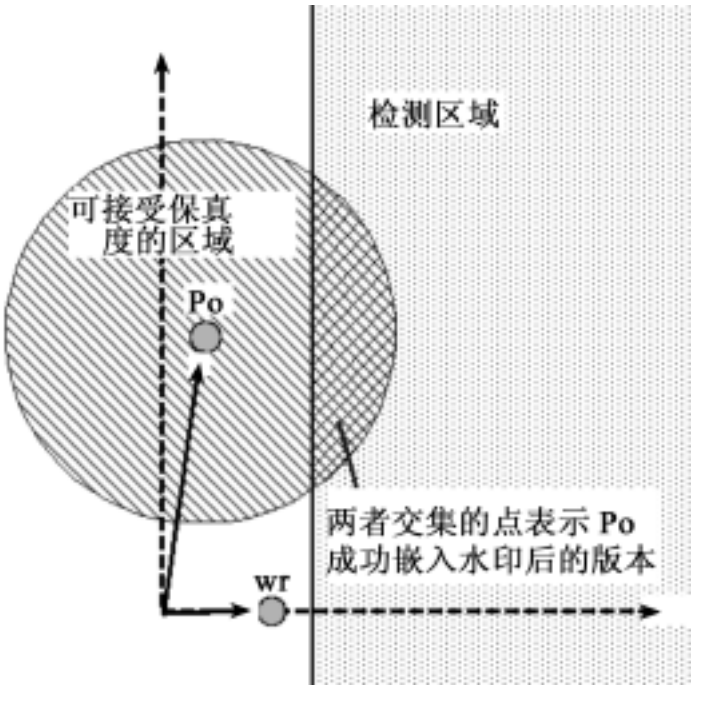


图 7.43　可接受保真度的区域和检测区域

在检测区域以外的图像,都可以由其他图像嵌入水印后生成。仍以 7.2 节的水印基本模型实验算法为例,与每个原始作品 Po 相加的都是和作品无关的相同向量 wa = αwr。图 7.44 说明了这一点。这样的算法存在的问题就是不能保证 100% 的有效性,因为嵌入器输出一幅位于检测区域以外的作品的概率不等于 0。

另一种策略是,其水印的生成和嵌入策略可以保证对于给定的水印模板和原始作品,嵌入器输出的图像总能位于某一特定的平面上。其几何模型见图 7.45。对这类系统而言,讨论给定信息的嵌入区域(也就是嵌入器所有可能输出作品的集合) 是有意义的。当且仅当嵌入区域完全位于检测区域内时,系统才具有 100% 的有效性。

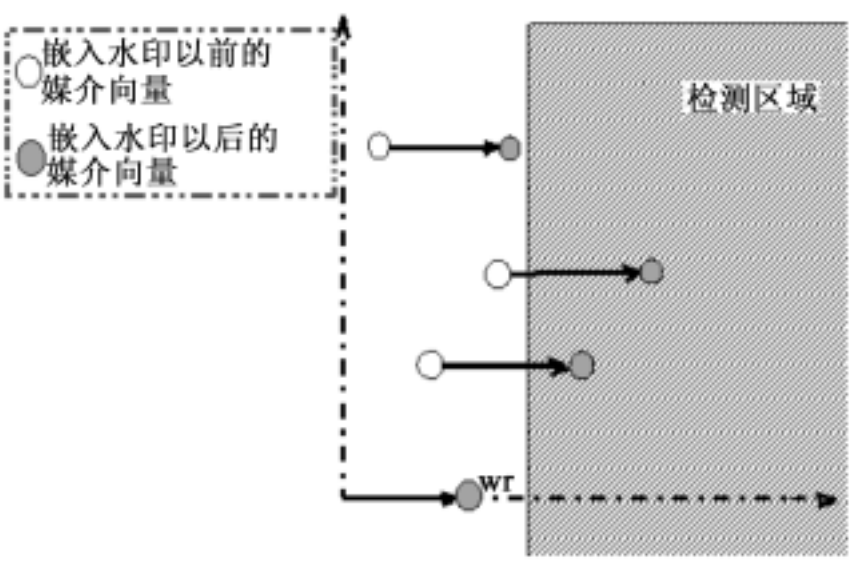


图 7.44　第一种嵌入策略

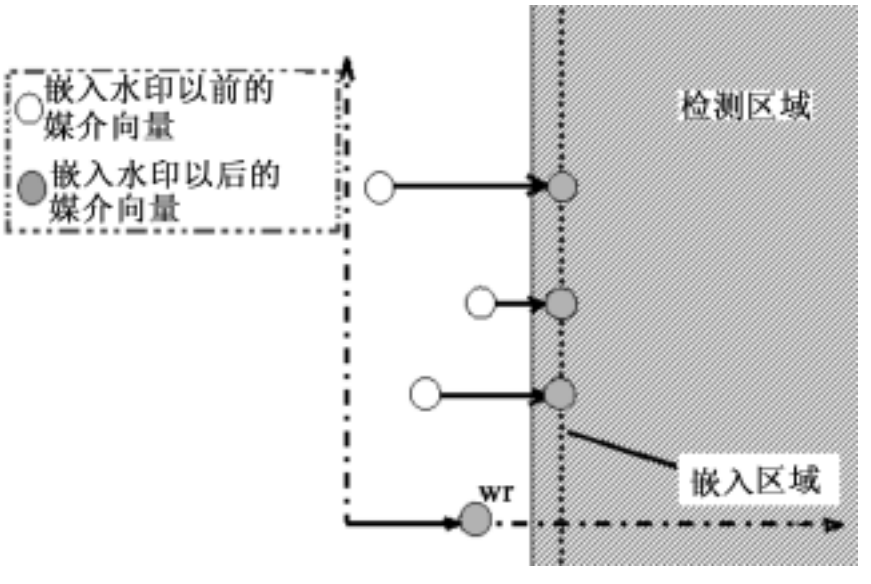


图 7.45　第二种嵌入策略

7.6　水印的相关检测

前面几节介绍了比较常用的几种水印系统,也分别简单地讨论了各种水印的检测。下面将详细地讨论水印的相关检测。

常见的水印检测策略中一般都涉及这样几种相关性的计算:线性相关、归一化相关和相关系数。其中,最基本的是线性相关,在介绍基本模型实验时已经对此有所了解。另一种相关性与它的区别在于计算内积前先对向量进行归一化。如果对向量归一化后得到两个单位向量,就可以计算出两者之间的归一化相关。如果先将两个向量均值减到 0 再计算归一化相关,那么就可以求得向量之间的相关系数。

7.6.1　线性相关

两个向量 V 和 wr 之间的线性相关是两者对应元素的乘积的平均值:

$$Z_{lp}(v, wr) = \frac{1}{N}\sum_{i} v[i]\, wr[i] \tag{7.9}$$

在通信中经常通过计算 $Z_{lp}(v, wr)$ 并将其与预定阈值比较,来检测接收到的信号 v 中是否存在已发送信号 wr。在工程上这个过程又被称为匹配滤波,在加性高斯噪声的作用下,它是信号检测的最优方法。

根据匹配滤波得到的检测区域包括了如图 7.44 所示的阴影平面一侧的所有点。该平面垂直于水印,它和原有的距离由检测阈值决定。如果由高斯分布决定的高维向量几乎和给定的水印是正交的,我们就能了解检测区域的鲁棒性。若参与检测的是一个无关的噪声向量,则噪声向量和检测区域的边界面平行,很少会穿过平面。

从广义上来说,一切检测算法,只要计算作品模板间的线性函数并且将函数值与特定阈值进行比较,都可以认为使用了线性相关检测。

比如 Koch 和 Zhao 提出过一种在图像中嵌入水印的算法。首先对图像的每个 8× 8块进行离散余弦变换(DCT) , 然后从每个块中选择一对 DCT 系数排序后进行分组。算法的主要思路是在每对系数中都嵌入 1 比特水印信息, 具体值取决于组内第一个系数是否大于第二个系数。如果大于就嵌入比特 1, 否则就嵌入比特 0。除了对该比特进行编码的一对系数外, 每个模板的块 DCT 系数都是 0。在不为 0 的那对 DCT 系数中, 第一个系数为 1, 第二个系数为 - 1 。因此, 如果将这些模板的其中之一和图像计算二者块 DCT 系数的相关性, 通过比较结果的正负就可以知道第一个系数是否大于第二个系数。因为 DCT 是线性变换, 如果在空间域对模板和图像进行相关性计算, 也可以得到相同的结果。

下面编写求线性相关的函数 linear_corr. m, 函数代码如下:

```
% 文件名: linear_corr. m
% 程序员: 李巍
% 编写时间: 2004. 3. 20
% 函数功能: 这是一个线性相关性的函数
% 输入格式举例:
% L_corr = linear_corr( wr, c, row, col)
% 参数说明:
% realwatermark 为真实的水印
% testwatermark 为待检测的水印
% row, col 是水印矩阵的维度
% L_corr 为函数计算出的两矩阵的线性相关
function L_corr = linear_corr( realwatermark, testwatermark, row, col)
N = row* col;
corr = 0;
for i = 1: row
    for j = 1: col
        corr = corr + realwatermark( i, j) * testwatermark( i, j) ;
    end
end
L_corr = corr / N;
```

7.6.2 归一化相关

线性相关的一大问题是检测值在很大程度上依赖于从作品中提取的向量幅度。这表明水印对于一些简单处理, 如改变图像的亮度或降低音乐的音量, 不具备鲁棒性, 等等。有文献指出, 线性相关检测器的虚警概率也很难预测。

有学者采用归一化相关的计算方法来改进线性相关的不足。在计算内积之前先

对提取信号模板和参考模板进行归一化，使它们成为单位向量，就可以解决这个问题。即：

$$v' = \frac{v}{|v|}$$

$$wr' = \frac{wr}{|wr|}$$

$$Z_{nc}(v, wr) = \sum_i v[i]\, wr[i] \tag{7.10}$$

称为归一化相关。

由归一化相关的阈值决定的检测区域和由线性相关的阈值决定的检测区域完全不一样。线性相关的检测区域包含了阴影平面一侧的所有点，而与归一化相关对应的是圆锥形的检测区域。这和以下事实是一致的，即两个向量的内积等于它们的欧氏长度与夹角余弦的乘积，表示为：

$$v \cdot wr = |v||wr| \cos\theta \tag{7.11}$$

其中，θ表示 v 和 wr 之间的夹角。因此，两个向量之间的归一化相关就等于它们之间夹角的余弦值。设置内积的阈值等价于设置夹角的阈值，即：

$$\frac{v \cdot wr}{|v||wr|} > \tau_{nc} \quad \theta < \tau_\theta \tag{7.12}$$

其中，

$$\tau_\theta = \cos^{-1}(\tau_{nc}) \tag{7.13}$$

因此，对于给定的参考向量 wr，其检测区域是以 wr 为轴、顶角为 $2\tau_\theta$ 的 N 维圆锥。

图 7.46 描述了参考向量 wr 在阈值 τ_{nc} 下的检测区域。图中显示的是标志空间的截面。参考标志沿着 x 轴方向，y 轴是任一与参考标志正交的方向。阴影部分表示平面上检测结果为包含参考标志 wr 的所有点。这是个 N 维圆锥体的二维截面，对于包含 wr 的所有平面来说形状都相同。也就是说，如果选择了不同的 y 轴方向（只要它仍然与 x 轴方向正交），该区域也不会变化。注意图中显示的阈值相当高，低阈值会使锥形变钝。

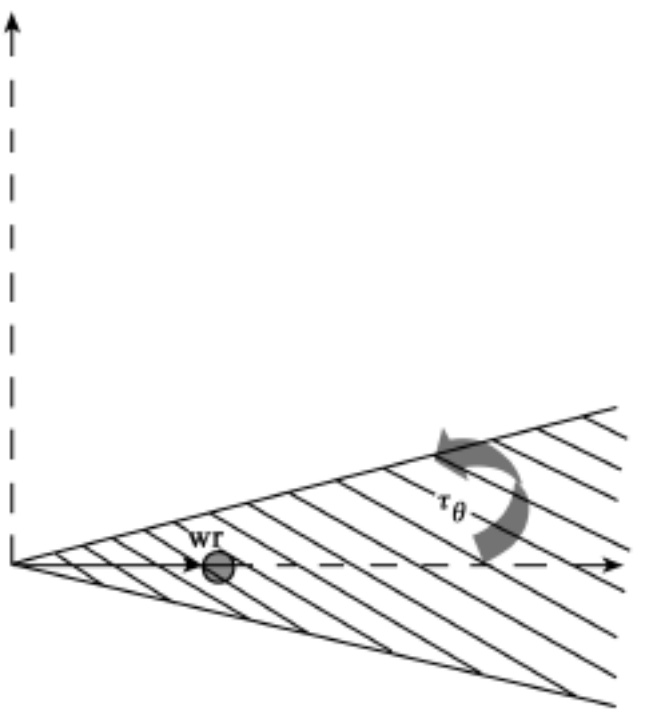


图 7.46　归一化相关的几何解释

虽然有时线性相关可以作为检测度量，但是检测阈值必须根据已提取标志的幅值进行缩放。这等价于使用归一化检测度量。为了说明这一点，首先要注意用已提取向量的幅值去除相关值可以得到相同的检测效果，于是有检测度量：

$$Z_1(v, wr) = \frac{v \cdot wr}{|v|} \tag{7.14}$$

根据上述定义,它和归一化相关之间的惟一区别在于,在此没有对参考标志进行归一化。不过,通常限制参考标志的幅度为常数,由此看来这两者之间至多相差一个常数因子。

下面编写求线性相关的函数 unification_corr. m,函数代码如下:

```
% 文件名: unification_corr. m
% 程序员: 李巍
% 编写时间: 2004. 3. 20
% 函数功能: 这是一个归一化相关函数
% 输入格式举例:
% U_corr = unification_corr( wr, c)
% 参数说明:
% realwatermark 为真实的水印
% testwatermark 为待检测的水印
% U_corr 为函数计算出的两矩阵的归一化相关
function U_corr = unification_corr( realwatermark, testwatermark)
mtimes = trace( realwatermark* testwatermark) ;
modulus = norm( realwatermark, 'fro') * norm( testwatermark, 'fro') ;
U_corr = mtimes /modulus;
```

7.6.3 相关系数

在本节中使用的最后一种相关形式是相关系数。先将两个向量的均值都减到0,然后再计算它们的归一化相关,由此得到向量的相关系数如下:

$$v' = v - \text{聊}$$
$$wr' = wr - \overline{wr}$$
$$Z_{cc}(v, wr) = Z_{nc}(\text{聊}\overline{wr}) \tag{7.15}$$

这就保证了水印对作品中直流量的变化(比如图像中每个像素值都增加一个常数)具备鲁棒性。

归一化相关和相关系数之间具有简单明了的几何关系:N 维空间中两个向量之间的相关系数就是这两个向量投影到(N - 1)维空间上的归一化相关。也就是说,减去向量均值的运算相当于投影到低维空间上。我们可以把它看做向量运算。例如,如果 v 是三维空间内的向量[x, y, z],那么 v - 聊= [x, y, z] - [聊聊聊 。向量[聊聊聊表示坐标系统对角线上距向量 v 最近的点。因此,经过减法运算就得到了和对角线正交的向量。这表明运算得到的向量位于(N - 1)维空间内,该(N - 1)维空间和原 N 维坐标系统的对角线正交。一个简单的二维的例子如图 7.47 所示,两个二维空

间向量减去它们的均值,结果相当于沿对角线方向投影到一维子空间。

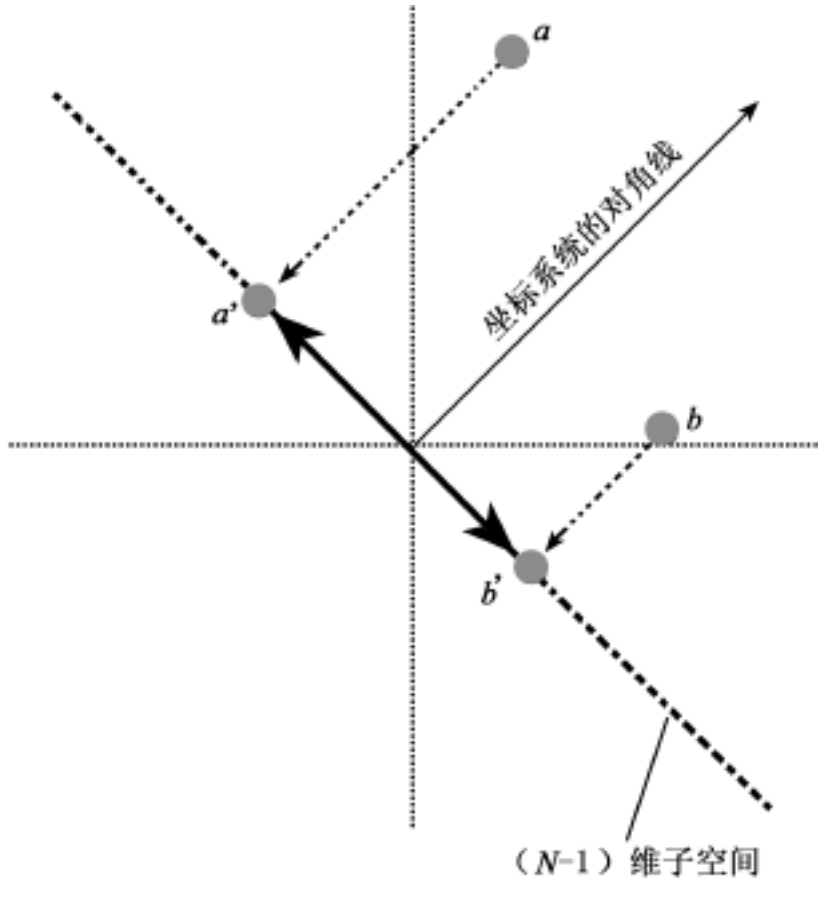


图 7.47　相关系数的几何解释

因为相关系数是归一化相关投影后的结果,所以经常交叉使用这两种度量,考察归一化相关的理论,然后利用相关系数建立系统来说明理论。如果根据已提取向量的样本标准差对线性相关进行缩放,就得到了下列检测度量:

$$Z_2(v, wr) = \frac{v \cdot wr}{S_v} \tag{7.16}$$

这与根据已提取标志的幅度(检测度量 Z_1)尺度变换相比,尽管只有细微差别,但该检测度量属于相关系数而不是归一化相关。已提取向量 v 的样本标准方差 S_v 表示为:

$$\begin{aligned} S_v &= \sqrt{\frac{1}{N}\sum_{i}^{N}(v[i] - \overline{v[i]})^2} \\ &= \frac{1}{\sqrt{N}}|v - \text{聊}| \\ &= \frac{1}{\sqrt{N}}|v'| \end{aligned} \tag{7.17}$$

其中 v 根据式(7.15)定义。于是有:

$$Z_2(v, wr) = \sqrt{N}\frac{v \cdot wr}{|v'|} \tag{7.18}$$

如果限制参考标志的幅度为常数并且均值为0，则这个检测度量等价于相关系数，因为 wr = wr 和 v · wr = v′· wr（因为 v 的均值对于它和一个均值为0 的向量之间的内积没有影响）。

下面编写求线性相关的函数 modulus_corr。函数代码如下：

```
% 文件名: modulus_corr. m
% 程序员: 李巍
% 编写时间: 2004. 3. 20
% 函数功能: 这是一个相关系数函数
% 输入格式举例:
% M_corr = modulus_corr( wr, c, 256, 256)
% 参数说明:
% realwatermark 为真实的水印
% testwatermark 为待检测的水印
% row, col 是水印矩阵的维度
% M_corr 为函数计算出的两矩阵的相关系数
function  M_corr = modulus_corr( realwatermark, testwatermark, row, col)
sumaDif = 0;
for i = 1: row
        for j = 1: col
                sumaDif = sumaDif + ( realwatermark( i, j)  -  testwatermark( i, j) ) ^2;
        end
end
Sv = ( sumaDif /( row*  col) ) ^( 1 /2) ;
M_corr = ( realwatermark*  testwatermark)
```