一种基于混沌和 SVD 的图像水印及其实现

王傲

15300240004

数字水印技术

COMP130093.01

指导教师：肖晓春

目录：

内容摘要：

本文介绍了一种基于混沌加密、DWT 变换和 SVD 分解的数字水印算法，利用变换域和多种方法保证鲁棒性的同时，用奇异值嵌入尽可能的减少了信息的嵌入量。同时，对相关背景和实现方法做了介绍，测试了相应的结果并进行了统计，证明了算法的有效和可靠。

关键字：混沌 3-DWT SVD 分解 Arnold 置换

1. 绪论

1.1 背景

随着计算机、网络和通信技术的快速发展，信息安全的保护问题日益突出。基于计算机和网络的信息交换为多媒体数字作品的使用和传播提供了便利的途径，然而数字作品极易被非法复制和篡改的特性，使得数字作品的版权保护成为一个迫切需要解决的实际问题。数字水印技术利用数字作品中普遍存在的冗余数据与随机性，向数字作品中加入不易察觉但可以判定区分的秘密信息水印，从而起到保护数字作品版权或完整性的作用。这种被嵌入的水印可以是文字、图片和其他种类的多媒体信息。数字水印的算法多种多样，大致可以分为在时空域和变换域上进行两类。数字水印应满足不可感知性、鲁棒性和稳健性等要求。

1.2 文章内容

本文综合本学期课程所学知识，介绍了一种基于混沌加密、DWT 变换和 SVD 分解的数字水印算法，在保证鲁棒性的同时利用奇异值尽可能的减少了信息的嵌入量，对相关背景和实现方法做了介绍，利用 Matlab 进行实现并测试了相应的结果，进行了统计。

2. 基本原理

2.1 时空域和变换域

图像的水印技术根据水印嵌入的方式可以大致分为两类:时空域技术(水印被直接嵌入图像的亮度、色度等值上)和变换域技术(将图像做某种数学变换并将水印嵌入于变换系数中)。从目前的情况来看, 变换域方法使用的比较普遍，因为变换域方法通常具有很好的

鲁棒性，对图像压缩、常用的图像滤波以及噪声均有一定的抵抗力。绝大多数变换域方法均采用了酉变换，如离散余弦变换 DCT、离散傅立叶变换 DFT 和离散小波变换 DWT 等等。

根据理论和实践的结果，基于变换域的数字水印技术拥有对各种攻击更高的鲁棒性，同时水印的不可见性也更好，因此大多数数字水印的实施方案都是基于变换域的，常见的如 DFT、DCT、DWT 等等。在此基础上，出现了其他很多侧重不同方面的数字水印实施方案，比如利用伪随机性加密、消除水印统计特征、抵抗裁剪和旋转等攻击等等。

2.2 混沌

混沌现象是一种复杂的而在非线性动力的系统中常见的、确定性的、类似随机的过程，这种过程既非周期又不收敛，并且对初始值有极其敏感的依赖性。混沌序列具有以下优点：1. 形式简单，只需要混沌映射参数和初始条件就可方便生成；2. 初始条件敏感性，一般不同的初始值迭代得到的混沌轨迹序列都不相同，很难从一段有限长度的序列推断混沌序列的初始条件；3. 确定性，相同初始值的混沌动力系统的相应轨迹相同；4. 具有白噪声的统计特性，可以用于需要噪声调制的众多应用场合。

由于混沌序列的这些优点，所以在数字水印技术中，可以将其与数字水印结合。二维水印图像的混沌加密就是将图像经过混沌模型发生变换，产生较大的差异。由于混沌的特性，这种加密比较简单，生成的结果确定，消除了原始图像的统计特征，并且攻击者很难对图像进行复原，而拥有密钥的使用者可以对图像进行逆变换获得原始图像。

2.3 奇异值分解

对任意 m×n 阶实矩阵 A，存在 m×m 阶正交矩阵 U 和 n×n 阶正交矩阵 V，使得

$$A = UEV^T \tag{1}$$

其中 E 为 m×n 阶的对角矩阵。E 对角线上元素为矩阵 A 的奇异值，表明了 A 的重要特征，且按大小顺序排列。

图像奇异值分解的主要特征有：1. 快速将大量的信号能量压缩到很少的系数中。奇异值所表现的是图像的固有特征而非视觉特性，为水印的不可见性提供了保障；2. 数字图像的奇异值（或者特征值）很稳定，经过常规的图像处理后，奇异值的变化很微小。因此，将水印嵌入到奇异值中有很好的鲁棒性，同时能降低嵌入的信息量，减少对载体的影响。

3. 实现方法

3.1 一些实现方法

现有的图像数字水印的实现方法有很多，其中很多是在变换域的基础上，结合其他现有技术来实现具有更好鲁棒性、更好不可见性的水印的。文献[3]采用 SVD 的特性降低虚警率，采用小波变换算法提高鲁棒性，通过两者的结合产生两个零水印信息，既保证了高鲁棒性又保证了低虚警率；文献[6]提出了一种新的非常鲁棒的基于奇异值分解(SVD)的数字水印方法，并将该方法与 Cox 的扩展谱方法进行比较，得到了很好的结果。

本文基于文献[1],着重介绍一种基于混沌加密和 SVD 的数字图像水印算法，采用有意义的二值图像作为嵌入水印，首先对原始图像进行三维小波分解，根据各个子图的特点，选取近似子图、水平子图和垂直子图作为嵌入区域。之后，对加密后的水印和嵌入区域进行分块，并对每一块进行 SVD 分解，将水印的 S 对角矩阵分别嵌入到对应的嵌入区域中，完成水印的三次嵌入。在水印提取时，通过比较提取出的三个水印的每一位，得到第四个水印的对应位，最后通过比较四个水印的归一化相关系数 NC 选择最终的提取水印。

3.2 水印的混沌加密预处理算法

文献[1]利用混沌映射产生一个混沌序列，将其转换成为与二值水印大小相同的二值矩阵，并与水印进行异或运算，实现水印的预处理加密。为了进一步提升安全性，更好的消除水印的统计特征，首先对水印图片进行一次置乱。

具体的预处理加密过程如下：

1. 对 m×n 的原始二值水印图像进行 Arnold 置乱，其中 Arnold 置乱的迭代为：

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & b \\ a & ab+1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} mod(N)$$

其中 x_n，y_n 是图片矩阵中像素点的位置

2. 利用混沌映射产生长为(m×n)/4+k 的序列，其中 k 为人为选取的任一大于 1 的整数，是密钥

3. 将产生的混沌序列转换成与水印相同大小的二值矩阵，其中具体转换过程如下：

1). 对照下表，将混沌序列的元素转换成为 4 位二进制序列：

表 1　混沌序列与二进制序列的转换关系

混沌区间	二进制	混沌区间	二进制
[7/8,1]	1111	[−1/8,0]	0111
[6/8,7/8]	1110	[−2/8,−1/8]	0110
[5/8,6/8]	1101	[−3/8,−2/8]	0101
[4/8,5/8]	1100	[−4/8,−3/8]	0100
[3/8,4/8]	1011	[−5/8,−4/8]	0011
[2/8,3/8]	1010	[−6/8,−5/8]	0010
[1/8,2/8]	1001	[−7/8,−6/8]	0001
[0,1/8]	1000	[−1,−7/8]	0000

图 1 混沌序列与二进制序列转换表

2). 将通过混沌序列得到的二进制串进行拼接

3). 将二进制串转换为 m×n 的矩阵

4). 将置乱后的水印与产生的混沌二值矩阵进行异或操作，得到加密后的水印

3.3 水印的嵌入算法

水印的主要嵌入步骤如下：

1. 对原始图像 I 进行 3-DWT 变换，获得 LL3、LH3 和 HL3 作为嵌入区域

2. 将加密后的水印和 LL3、LH3、HL3 三部分分别进行 8×8 分块

3. 对每一块进行奇异值分解，得到 Sw_{ij}、Sa_{ij}、Sh_{ij}、Sv_{ij} 4 个对角矩阵和对应的 U、V 矩阵，其中，Sw_{ij} 为水印各块的对角矩阵，Sa_{ij}、Sh_{ij}、Sv_{ij} 分别为 LL3、LH3、HL3 各块的对角矩阵

4. 根据水印嵌入公式，将水印分别嵌入到 LL3、LH3、HL3 这三个子图中：

$$SI_{k(i,j)}(m,n) = S_{k(i,j)}(m,n) + \alpha_k Sw_{i,j}(m,n)$$

其中，$S_{k(i,j)}$表示 Sa_{ij}、Sh_{ij}、Sv_{ij}；$SI_{k(i,j)}$表示 Sa_{ij}、Sh_{ij}、Sv_{ij} 嵌入后的对角矩阵；α_k 表示嵌入强度。为了保证水印的不可见性和鲁棒性，三个区域的嵌入强度不同。同时，为了后面能够提取水印图片，需要记录相应的 U 和 V 矩阵。

5. 利用式(1)，对水印和 LL3、LH3、HL3 进行奇异值重构，将各块还原为一个整体

6. 逆小波变换得到嵌入水印的图像 I_w

可以看到，在预处理时先对水印图像进行了置乱和加密，在消除水印图片统计特征的同时利用混沌进行了加密，更好的保证了水印的安全。而嵌入的过程主要使用了小波变换和奇异值分解，将水印的奇异值嵌入其余三个部分的奇异值，利用奇异值的特性，更好的

保证了水印的鲁棒性。此外，嵌入共进行了三次，将三个部分都嵌入了水印，可以进一步提升鲁棒性。

3.4 水印的提取算法

水印的提取算法是嵌入算法的逆运算，需要原始图像。

提取的具体步骤为：

1. 将待提取图像和原始图像进行 3-DWT 分解，得到 $LL3_w$、$LH3_w$、$HL3_w$ 和 LL3、LH3、HL3

2. 将 6 个部分分别进行 8×8 分块，对各块进行 SVD，得到各对角阵 S_k'、S_k，S_k' 为待提取水印图像得到的奇异值，S_k 为原始图像得到的奇异值

3. 利用公式

$$Sw'(i,j) = (S_k'(i,j) - S_k(i,j))/\alpha_k$$

得到三组图像的水印的奇异值对角阵

4. 根据式(1)，利用嵌入时记录的 U 和 V 矩阵，对水印的各块进行奇异值重构，组合成完整的水印

5. 根据水印加密的逆运算，解密出水印

6. 根据 3 组水印，生成第 4 组水印：若对于矩阵的某一比特位，3 组中有 2 组或以上均为 1，则第 4 组水印的该比特位记为 1

7. 计算四组水印的归一化相关系数 NC 的值，将 NC 值大的一组作为提取出来的水印

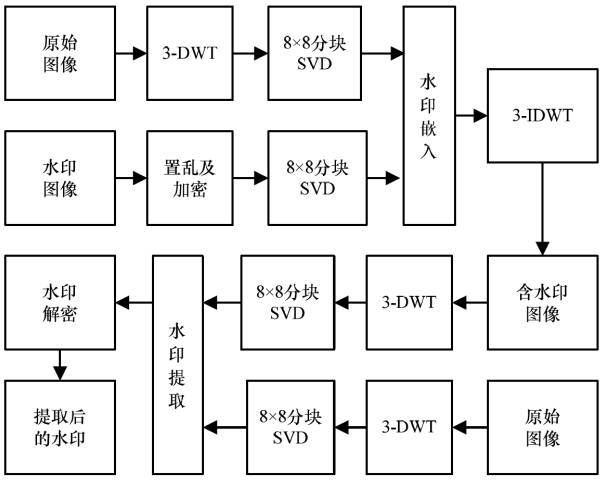

图 2 该水印算法的流程

3.5 算法分析

从算法的实现过程可以看出，这种图像水印的算法兼顾了鲁棒性和不可见性。

这种算法提升鲁棒性的方法有：1. 嵌入前对水印图片进行预处理，通过置乱和混沌加密，在消除水印统计特征的同时保证了水印的安全；2. 嵌入时，将水印分三次嵌入，提取时则分三次提取，并且根据三个水印的情况计算出第四个水印的值，选出与原始水印NC值最大的作为提取出来的水印；3. 嵌入时将水印SVD分解的奇异值嵌入载体图片的奇异值，更好的保留了主要信息，提升了鲁棒性；4. 可以人为选择鲁棒性系数α，调整鲁棒性和不可见性的平衡；5. 加入了人为选定的密钥。

同时，这个算法考虑到了水印的不可见性。提升不可见性的方法有：1. 在DWT变换域进行嵌入，并且选择高层的DWT分解域作为嵌入位置；2. 嵌入时，只嵌入水印分解后的奇异值，在保留最重要信息的同时尽可能减少了插入的信息容量。

不过，这种算法也存在着缺点。首先，这种算法提取水印时需要原始图片，是一种非盲水印；并且，提取时需要嵌入时保存的 U 和 V 矩阵，比较麻烦。此外，由于这种算法使用的是3-DWT，因此可以嵌入的容量非常小。实验中，使用512×512的灰度载体图片，最大只能嵌入64×64的二值图像。不过，这两个问题均是由于算法希望在提升鲁棒性的同时兼顾不可见性所造成的，可以理解。

4. 实验结果

4.1 代码实现

为了更好的测试这个水印算法的性能，通过Matlab进行了实现，并与课程中实现的基于DCT的数字水印算法在各个方面进行了性能验证和对比。

在实验的过程中，发现了这个水印算法的一些特性。首先，对于Arnold置乱，在实验中发现它是有一定周期的，比如将参数a设为3，参数b设为5，则其周期为12，即迭代12轮后图片变回原始图片。其次，根据论文中描述，选用Chebyshev映射生成混沌序列，序列的值均在-1和1之间，并且若给定的初始值和长度相同，则结果相同，而且完全无法看出规律性。将水印图片与生成的二值图像取两次异或，则恢复原始图片。

实验中选择 db1 作为小波基。使用 SVD 分解并嵌入时，需要加上鲁棒性系数 α，测试中发现 α 大于 5 以上时才能保证较好的准确率（NC 值在 0.996 以上），并且对原始图片的影响很小，因此后面的实验均以略高于 5 的 α 作为系数。

4.2 鲁棒性测试

论文中给出了一些测试的结果。为了自行测试这种图像水印算法的鲁棒性，将其与本学期课程中实现的基于 DCT 的水印算法进行对比，对比较常见的攻击方式进行测试并计算提取出的水印，攻击方式包括 JPEG 压缩、椒盐噪声、高斯噪声和剪切。以归一化相关性系数 NC 作为衡量指标。本算法中 α 分别取 5.1，5.11 和 5.12，DCT 中 α 取 0.1。下面是试验结果（结果为与原始水印图像的 NC 值）：

算法 / 攻击方式	本文水印算法	基于 DCT 的水印算法
JPEG 压缩，30	0.9764	0.6998
JPEG 压缩，60	0.9978	0.9860
JPEG 压缩，90	0.9999	1.0
椒盐噪声 0.02	0.9346	0.9337
高斯噪声 0.01	0.9287	0.8774
截取图左上 1/4	0.9260	0.9854

表 1 实验结果统计

统计结果的柱状图如下：

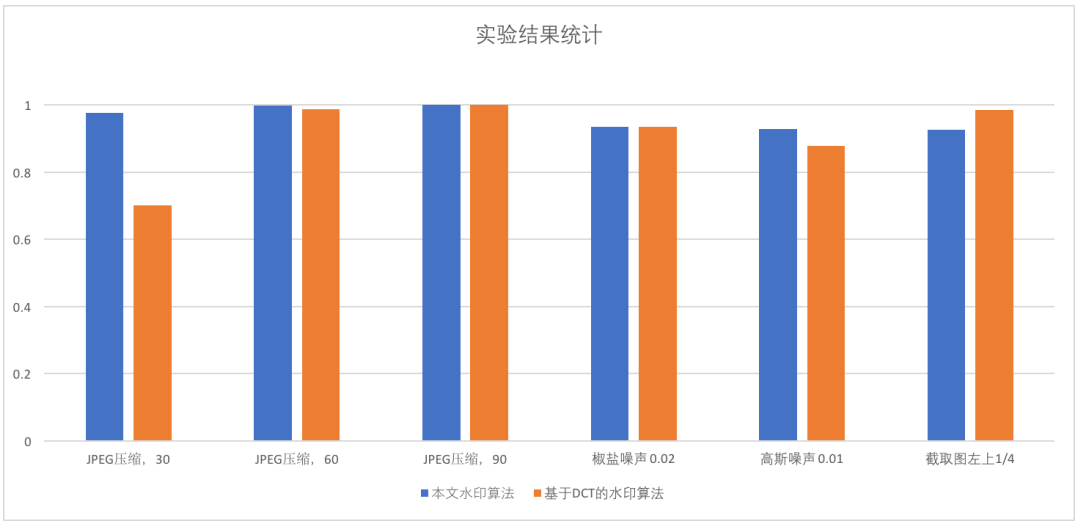

图 3 实验结果统计柱状图

根据获得的实验结果，以 NC 值为评判依据，两种算法针对几种攻击的鲁棒性相差不是很大。其中，对于不同压缩质量的 JPEG 压缩，本文的算法均优于基于 DCT 的算法；两者对于椒盐噪声攻击的鲁棒性相差不多，而对于高斯噪声，本文算法略优于基于 DCT 的算法；基于 DCT 的算法对图片剪切的鲁棒性要高于本文算法。总体来说，本文算法的鲁棒性略优于基于 DCT 的水印算法。

实验中两者取 α 的原则是不影响图片质量的前提下，不加以攻击时能够正确提取出水印的最小值。两者 α 的值相差的比较大，不过方法不同，α 的值没有可比性。

论文中的实验做的更加全面，给出的实验结果如下：

表 2　本文算法与文献[10]算法的 *NC* 值比较结果

攻击类型	本文算法	文献[10]算法
JPEG 压缩(70%压缩比)	1.000 0	-
JPEG 压缩(50%压缩比)	1.000 0	-
JPEG 压缩(30%压缩比)	0.986 2	-
JPEG 压缩(15%压缩比)	0.927 3	-
JPEG 压缩(95%压缩比)	-	0.985 3
JPEG 压缩(75%压缩比)	-	0.728 3
JPEG 压缩(55%压缩比)	-	0.586 0
JPEG 压缩(15%压缩比)	-	0.421 8
高斯噪声(0.01)	0.928 2	0.784 6
椒盐噪声(0.02)	0.930 7	0.723 5
均值滤波(3×3)	0.990 3	0.857 4
剪切(10%)	0.898 3	0.863 5
剪切(20%)	0.843 7	0.783 0

表 3　本文算法与文献[11]算法的 *NC* 值比较结果

攻击类型	本文算法	文献[11]算法
JPEG 压缩(70%压缩比)	1.000 0	0.973 4
剪切 1/4	0.805 4	0.813 2
高斯噪声	0.928 2	0.755 2

表 4　其他攻击下本文算法的 *NC* 值

攻击类型	*NC* 值
加性噪声(20%)	0.885 0
加性噪声(40%)	0.901 3
加性噪声(60%)	0.900 9
加性噪声(100%)	0.919 4
放大 512- 768-512	1.000 0
缩小 512-408-512	0.922 5
缩小 512-256-512	0.954 9

图 4 论文中的实验结果

从论文给出的结果，可以看出这种水印算法的优越性。在设计上，这种算法同时考虑了水印的鲁棒性和不可见性，并且通过实验的结果得到了证明。

5. 结论

本文介绍了一种基于混沌加密、DWT 变换和 SVD 分解的数字水印算法，利用变换域和多种方法保证鲁棒性的同时，利用奇异值嵌入尽可能的减少了信息的嵌入量。同时，对相关背景和实现方法做了介绍，测试了相应的结果并进行了统计，证明了算法的有效和可靠。

6. 参考文献

[1]薛胜男,陈秀宏.基于混沌加密和 SVD 的数字图像水印算法[J].计算机工程,2012,38(19):107-110.

[2]陈河山,吕珍珍,罗伟.一个基于离散混沌加密的数字水印算法[J].计算机科学,2014,41(12):48-52.

[3]陈伟琦,李倩.基于 DWT-SVD 的图像双零水印算法[J].计算机工程与科学,2014,36(10):1991-1996.

[4]纪震,肖薇薇,王建华,张基宏.基于混沌序列的多重数字图像水印算法[J].计算机学报,2003(11):1555-1561.

[5]张志明,王磊.基于混沌加密的 DCT 域图像水印算法[J].计算机工程,2003(17):9-10+39.

[6]刘瑞祯,谭铁牛.基于奇异值分解的数字图像水印方法[J].电子学报,2001(02):168-171.

[7]孙圣和,陆哲明.数字水印处理技术[J].电子学报,2000(08):85-90.