## 基于混沌理论的盲水印算法研究

专 业： 信息安全 学 号：8001715013

学生姓名：陈梦雅 指导教师：：刘凌峰

**目录**

[基于混沌理论的盲水印算法研究 - 1 -](#_Toc20035)

[摘 要 - 3 -](#_Toc12584)

[一、绪论 - 4 -](#_Toc3177)

[二、虫口模型和混合光学双稳模型 - 6 -](#_Toc17992)

[2.1、混沌原理 - 6 -](#_Toc28532)

[2.2、虫口模型 - 6 -](#_Toc648)

[2.2.1、原理 - 6 -](#_Toc9394)

[2.2.2、代码实现 - 7 -](#_Toc22125)

[logistic.m - 7 -](#_Toc14660)

[logistic\_picture.m - 9 -](#_Toc9882)

[2.3、混合光学双稳模型 - 12 -](#_Toc25397)

[2.3.1、原理 - 12 -](#_Toc3326)

[2.3.2、代码实现optical\_bistability.m - 12 -](#_Toc26756)

[2.4、利用混沌模型实现完全随机选择 - 14 -](#_Toc9280)

[2.4.1、原理 - 14 -](#_Toc15193)

[2.4.2、代码实现chaos.m - 14 -](#_Toc14230)

[2.4.3、混沌算法流程图 - 16 -](#_Toc23799)

[三、水印算法 - 18 -](#_Toc17017)

[3.1、对图像亮度值的分析 - 18 -](#_Toc25902)

[3.1.1、原理 - 18 -](#_Toc18130)

[3.1.2、代码实现compareBright.m - 18 -](#_Toc14927)

[3.1.3、亮度分析流程图 - 19 -](#_Toc30088)

[3.2、水印嵌入算法 - 21 -](#_Toc9728)

[3.2.1、原理 - 21 -](#_Toc17568)

[3.2.2、代码实现hide.m - 21 -](#_Toc5464)

[3.2.3、水印嵌入流程图 - 24 -](#_Toc10067)

[3.2、水印提取算法 - 25 -](#_Toc1836)

[3.3.1、原理 - 25 -](#_Toc32698)

[3.3.2、代码实现extract.m - 25 -](#_Toc372)

[3.3.3、水印提取流程图 - 26 -](#_Toc24329)

[四、测试 - 28 -](#_Toc28466)

[4.1、中值滤波 - 28 -](#_Toc4205)

[4.1.1、代码实现median.m - 28 -](#_Toc4679)

[4.1.2、相关性检测plotmedian.m - 29 -](#_Toc3079)

[4.2、锐化滤波 - 31 -](#_Toc20748)

[4.2.1线性锐化 - 31 -](#_Toc6815)

[4.2.1.1、代码实现sharpL.m - 31 -](#_Toc9651)

[4.2.2非线性锐化 - 32 -](#_Toc30717)

[4.2.2.1代码实现sharpS.m - 32 -](#_Toc26046)

[4.3、马赛克攻击 - 33 -](#_Toc28944)

[4.3.1.代码实现mosaic.m - 34 -](#_Toc2567)

[4.3.2、相关性检测plotmosaic.m - 35 -](#_Toc9292)

[4.4、加噪攻击 - 36 -](#_Toc9263)

[4.4.1、代码实现noiseadd.m - 36 -](#_Toc32224)

[4.4.2、相关性检测plotaddnoise.m - 37 -](#_Toc21619)

[4.5、jpeg压缩 - 39 -](#_Toc5811)

[4.5.1、代码实现jpeg.m - 39 -](#_Toc27447)

[4.5.2、相关性检测plotjpeg.m - 40 -](#_Toc5202)

[4.6、模糊处理 - 42 -](#_Toc31680)

[4.6.1、代码实现blurringL.m - 42 -](#_Toc22827)

[4.6.2、相关性检测plotblurring.m - 44 -](#_Toc4601)

[4.7、旋转、改变大小和裁剪 - 46 -](#_Toc4047)

[4.7.1、代码实现rotate\_resize\_crop.m - 46 -](#_Toc2877)

[4.8、测试算法流程图 - 48 -](#_Toc12559)

[4.9、测试总结 - 49 -](#_Toc21107)

[五、结论 - 51 -](#_Toc25646)

[六、致谢 - 52 -](#_Toc509)

[七、参考文献（References） - 53 -](#_Toc20864)

# 摘 要

本文提出了一种基于虫口模型和混合光学双稳模型的盲水印算法。就像图像色彩的轻微变化不易被人眼察觉一样,图像亮度的轻微改变也同样对视觉效果没有太大影响，所以在像素亮度值中找冗余就是该文研究的内容之一，再基于虫口模型或混合光学双稳模型随机选出插入水印信息的像素点位置，来进行水印信息的插入和提取，对插入了水印信息的图像进行攻击测试，测试结果表明，该水印具有较好的鲁棒性与良好的不可感知性。

**关键词：**混沌模型；亮度分量；伪随机；盲水印

**Research on blind watermarking algorithm based on Chaos Theory**

Abstract

A blind watermarking algorithm based on insect model and hybrid optical bistable model is proposed. Just as the slight change of image color is not easy to be perceived by the human eye, the slight change of image brightness also has little effect on the visual effect, so finding redundancy in the pixel brightness value is one of the contents of the study, and then based on the insect mouth. The model or hybrid optical bistable model randomly selects the pixel position of the inserted watermark information to insert and extract the watermark information, and attacks the image with the watermark information inserted. The test results show that the watermark is robust and good imperceptibility.

**Keyword:**Chaotic Model; Brightness Component; Pseudorandom ;

Blind watermarking;

# 一、绪论

随着手机、计算机、多媒体的普及，每天需要进行处理的视频、图像数不胜数，相应的软件、机构不断兴起，对于视频、图像的版权保护也变得越来越重要，数字水印技术也因此得以迅速发展。文献[1]结合图像置乱技术和混沌理论,使得水印的鲁棒性和安全性大大提高；文献[2]提出了小波深层变换域的低频子图，并利用混沌置乱和二值操作的方法，来进行水印的嵌入，该方法对多种攻击都表现出了很强的鲁棒性；文献[3]以幻方为基础进行水印的置乱，使得计算量减小，但是缺点是：安全性不高；文献[4]其实和文献[3]原理大致相同，只是嵌入水印时使用的是伪随机，安全性不高；文献[5]提出一种双混沌动力系统，基于混沌序列的初值敏感性和迭代不重复性，使得轨迹难以预测，使得水印的安全性大大提高；文献[6]使用复合离散混沌映射和双混沌映射来进行图像水印的之乱，嵌入和提取采用DCT变换，安全性大大提高；文献[8]产生了基于多种混论理论的随机数；文献[9]提出利用简单低维度的混沌虫口模型，对水印信息中的每一个字符进行反复迭代，使安全性大大的提高。文献[10]将水印信息进行量化，根据对比度的不同，实现了水印信息的盲提取；文献[11]基于奇异值的稳定性，通过奇异值的最大值来进行水印的嵌入；文献[12]基于两种变换，即傅叶斯变换和对数极坐标变化，实现了水印的插入和盲提取；文献[13]针对有雾图像，将其转化为YUV图像，利用YUV图像的亮度分量，分别得到散射光和大气光，成功将有雾图像清晰化。文献[14]利用LSB隐写算法，对相邻像素点的最低位进行分析，检测出了秘密信息的存在及其大小。文献[15]通过分析RGB图像和YUV图像的特点，提出了一种对YUV图像的颜色空间和亮度空间进行运动检测的算法，克服了因亮度变化和阴影所带来的影响。文献[16]结合混沌序列和盲水印算法，成功抵抗高斯噪声和椒盐噪声，但该算法对剪切、旋转、滤化等攻击表现出来的鲁棒性不高。文献[17]针对图像的二维矢量地图数据，通过分析两个分量长度的规律实现水印的提取与嵌入，提出了一种鲁棒性高的盲水印算法。文献[18]利用块的均值、位的修改及特征串的，在灰度级上，总结出了一种盲水印算法；文献[19]结合两种混沌序列和提升小波在彩色图像盲水印上的研究，提出了一种抗裁剪的盲水印算法。文献[20]提出了基于双密匙的混沌盲水印算法，其中一个密匙用来产生随机数子，另一个密匙用来随机置乱，该算法表现出来很高的安全性。本文通过将载体图像转化为YUV模型的图像，因图像的亮度就是在YUV模型中的Y亮度空间体现的，使Y亮度空间的值落在（0,1）区间内，并将亮度空间分为128个单位，经实验，在某一个像素点进行加减一个单位，引起的亮度变化人眼不易察觉，这在一定条件下可以认为是一种亮度冗余，这就为我们的水印插入提供了可能，然后我们在基于混沌模型，随机的选择要插入信息的像素点，进行水印信息的插入和提取。

# 二、虫口模型和混合光学双稳模型

2.1、混沌原理

初值敏感性是混沌数据重要的特征之一，所谓初值敏感性，简单来说就是，我们假设V是一个紧密空间，对连续映射f:V->V,无论m，n离的多么近，在f的作用下，二者的轨道都可能分开很大的距离。如图2，图3，即使初值相差0.0001，得到的混沌序列也相差较远。

除此之外，在f的作用下，任一点的作用域必须将随机遍布整个空间V，而且具有周期点，这就表明混沌系统，具有一定的规律可循，而并不是像普通随机数那样，数据一片混乱，因此我们可以利用混沌模型特有的混沌方程来产生很好的随机数。

2.2、虫口模型

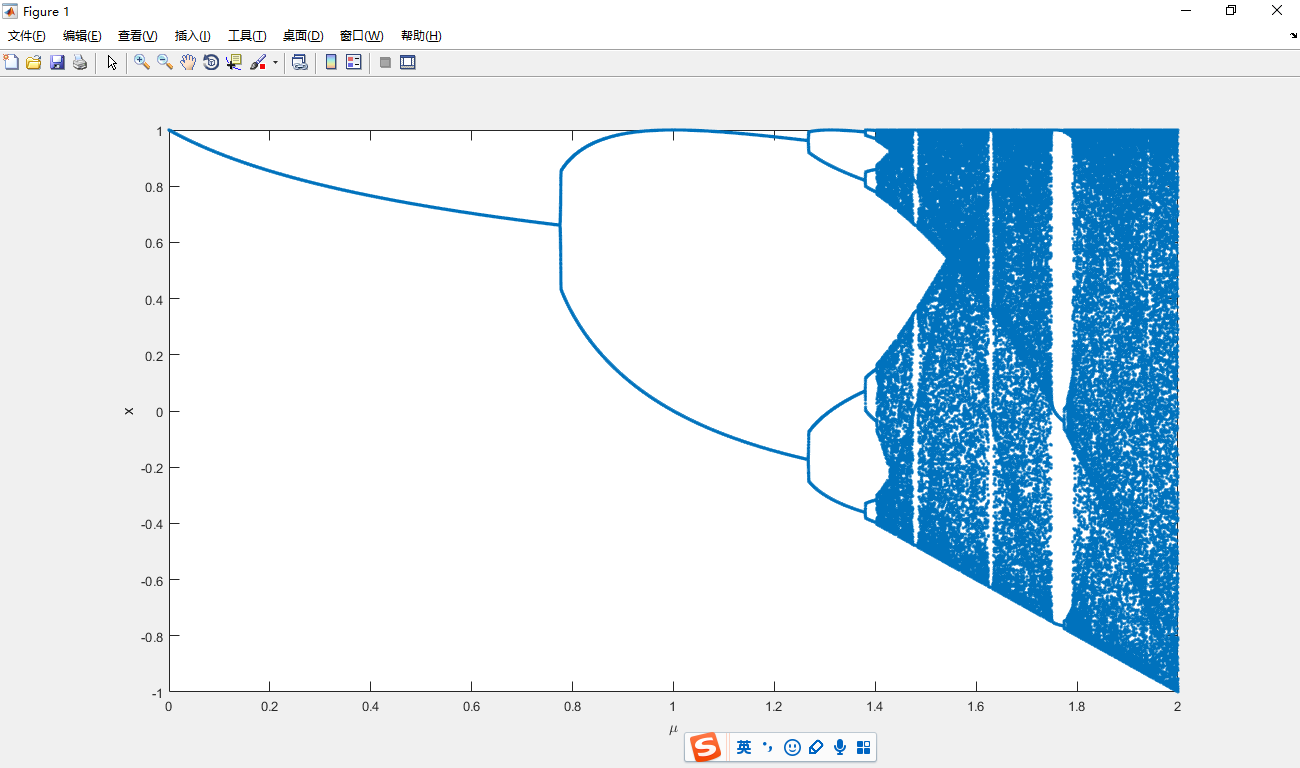


图1、虫口模型

2.2.1、原理

之所以叫虫口模型是因为昆虫的繁殖可以形象的描述该模型，设Xn+1为子代数量，Xn为亲代数量，假设这二者之间的关系只和年份n相关，且n只能取整数。如果用一个函数关系表示，如下所示：

Xn+1=f(Xn)

其中最简单的虫口模型就是Logistic方程。

Xn+1=Xn(p-qXn)其中p为昆虫年增长率，qXn可以理解为因物竞天择等因素而引起的昆虫死亡。

为了数学上的方便，令p=q=μ,则：

Xn+1=μXn(1-Xn)

μ取合适的值时，方程产生的随机数将具有混沌特性。

我们根据参数μ的取值讨论如下:

在此之前，我们先了解两个概念：吸引不动子点和排斥不动点：其中吸引不动点表示在该不动点的导数必定小于1，如果从该不动点出发进行反复的迭代，那么可以收敛到该不动点；而排斥不动点表示的是在该不动点的导数必定大于1，并且从该点出发反复迭代，并不能收敛到该不动点。

0<μ<=1

在范围内变化曲线十分简单,如图4及图5所示，横坐标为0的点为吸引不动点，除此之外，没有了多余的不动点，曲线最终会降到0，也就虫口模型而言，虫子最终的数目会趋于0。

1<μ<=3

在该范围内变化曲线也比较简单,不动点0,1-1/μ为仅有的两个周期点,且0为排斥不动点,1-1/μ为吸引不动点。

3<μ<=4

由图4和图5可知，系统由倍周期通向混沌，所以我们μ的取值应在该范围确定，才能使方程具有混沌特性。

μ>4

系统的动力学形态更复杂,在此就不展开了。

2.2.2、代码实现

#### logistic.m

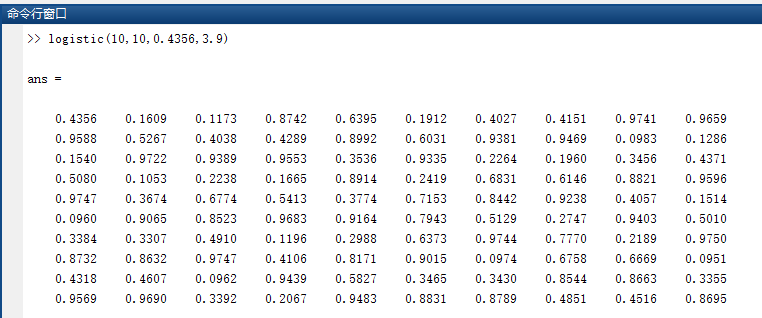
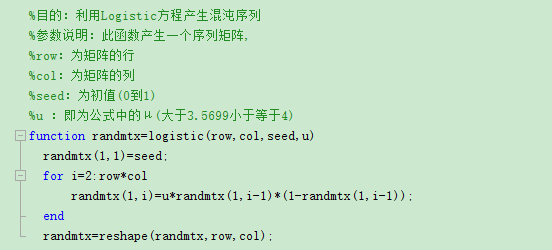


图2、初值为0.4356

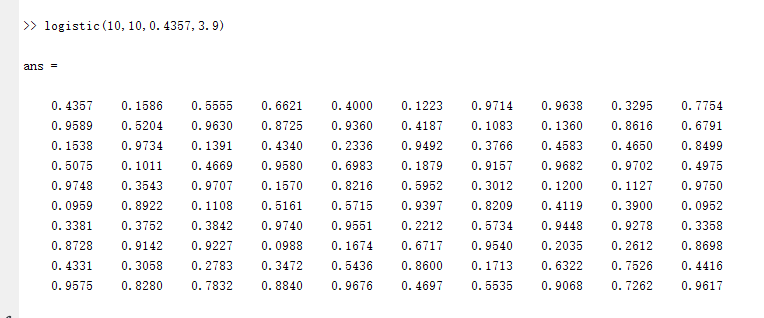
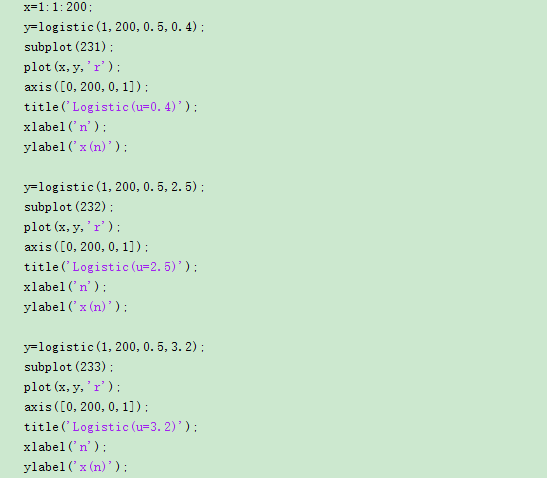
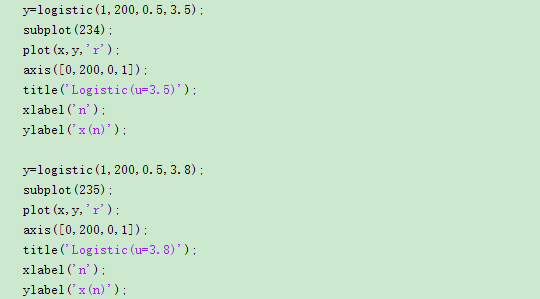


图3、初值为0.4357

#### logistic\_picture.m







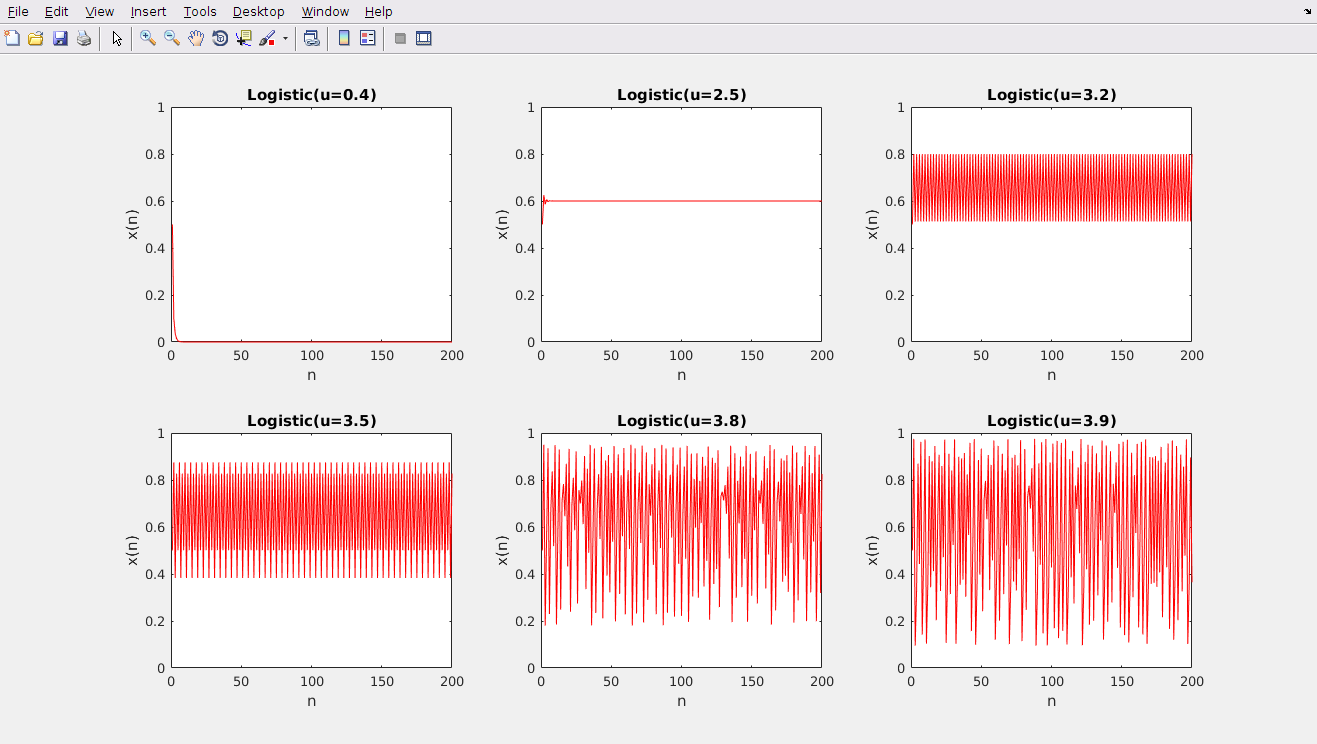
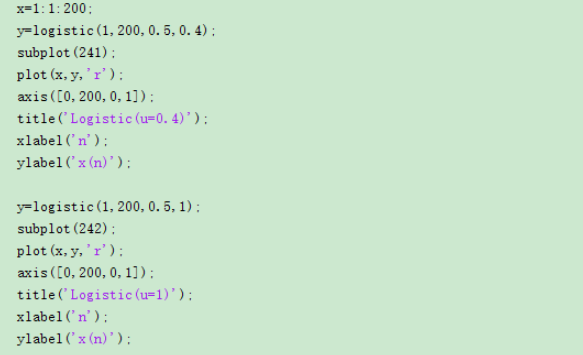
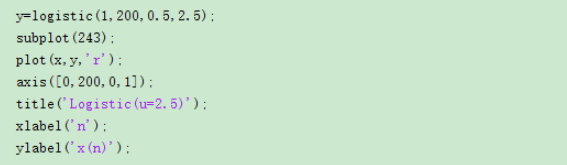
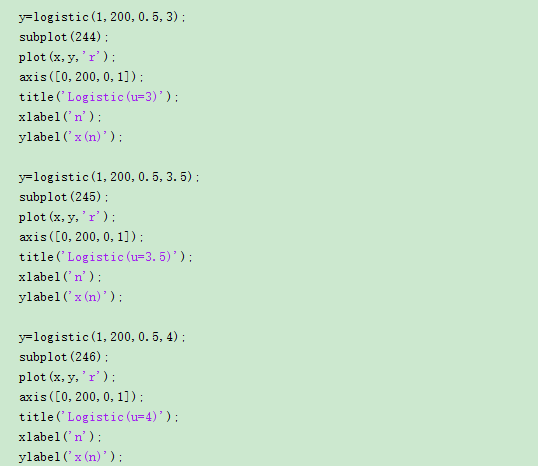


图4、μ随机取不同值











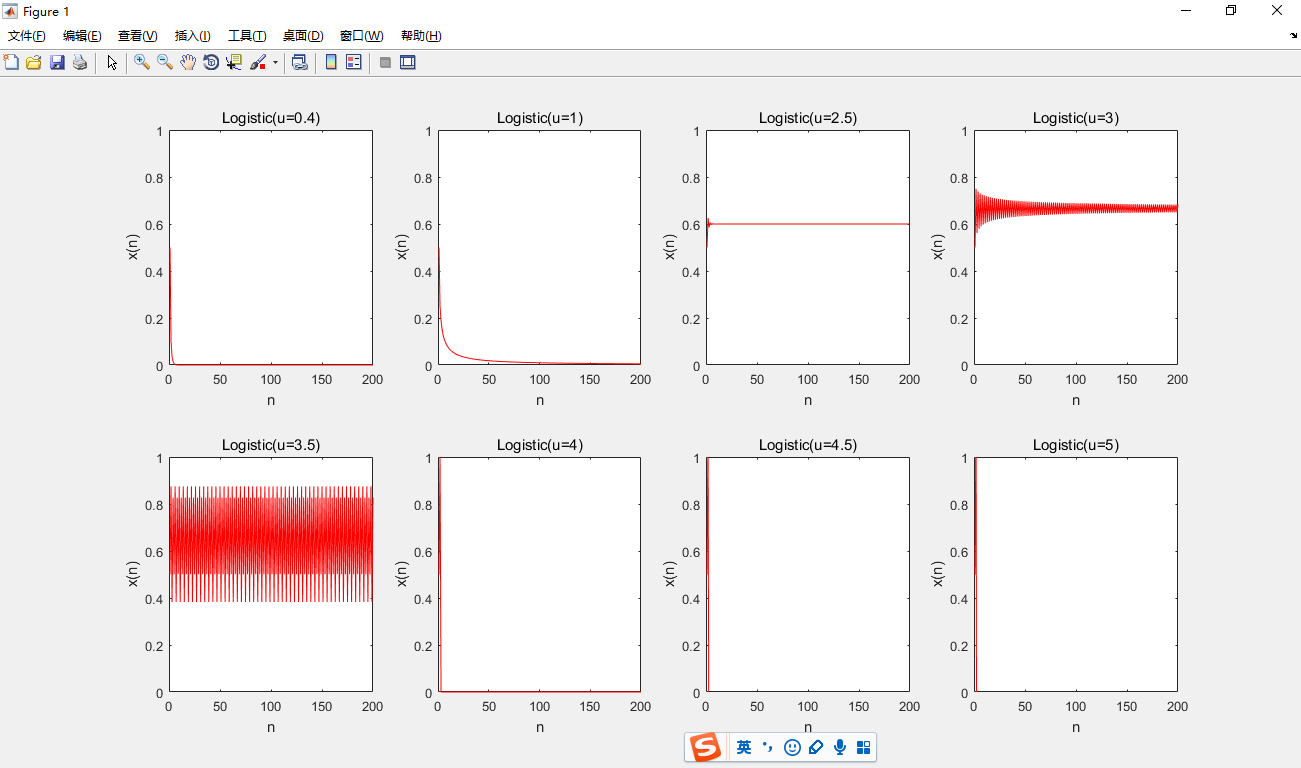


图5、μ按顺序取不同值

2.3、混合光学双稳模型

2.3.1、原理

光学双稳模型源自光学的双稳性，而光学双稳系统在适当的条件下能够表现出丰富的混沌运动现象，为我们随机数的产生提供了条件。

混合光学双稳模型是光学双稳模型中的较简单的一种，其中方程的形式为:

Xn+1=Asin(Xn-Xb))

这里有两个参数A,Xb,分别取4和2.5,这时，方程产生的随机数将进行处于混沌状态，假如我们给予其不同的初值，那么我们将得到不同的混沌序列值。

2.3.2、代码实现optical\_bistability.m



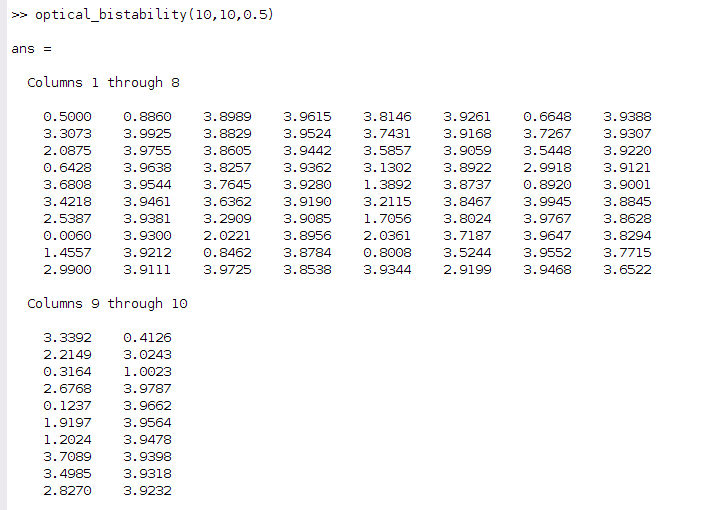


图6、混合光学双稳模型随机值

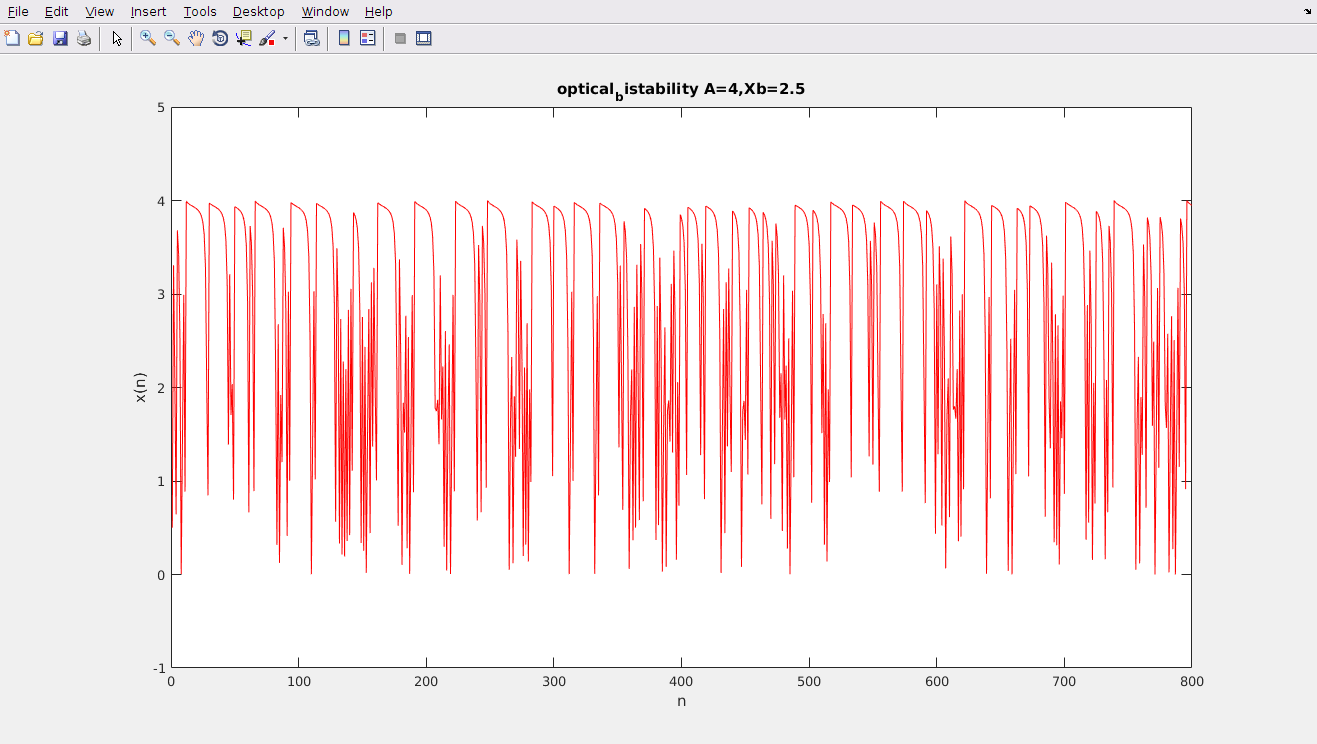


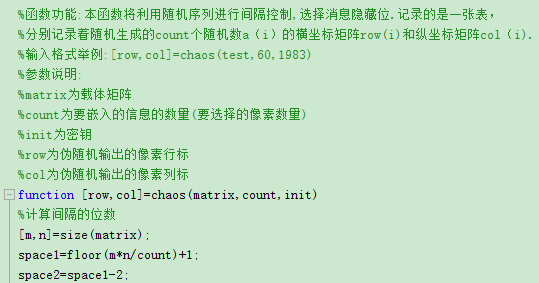
图7、混合光学双稳模型变化图

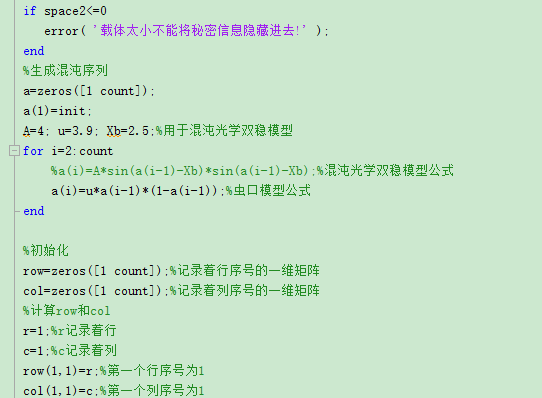
2.4、利用混沌模型实现完全随机选择

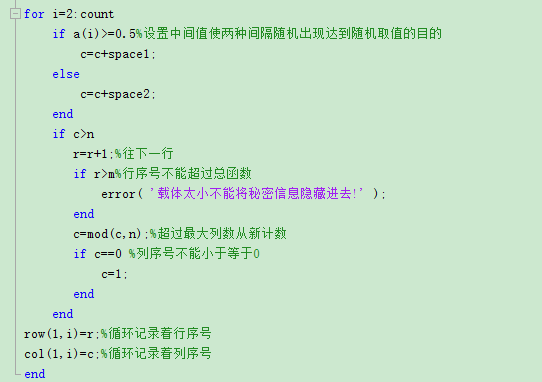
2.4.1、原理

利用混沌系统随机挑选像素点，其实和随机挑选随机点并无太大不同，只需将随机算法改为混沌序列产生算法，然后需要计算出合理的两个间隔和一个中间值，若混沌方程产生的随机数大于该中间值，那么从出发点相隔隔第一个间隔，得到第一个像素点，存取其横坐标，纵坐标；若产生的随机数小于该中间值，那么从出发点相隔第二个间隔，得到第二个像素点，存取其横、纵坐标；按照该规律，循环得到要求数量的随机像素点。利用混沌序列随机性，使得间隔不会均匀分布，而是随机分布，利用间隔大小和中间值的可控性，通过调整间隔大小和中间值，使得每个像素点都能被取得，挑选出符合要求数量的随机像素点。

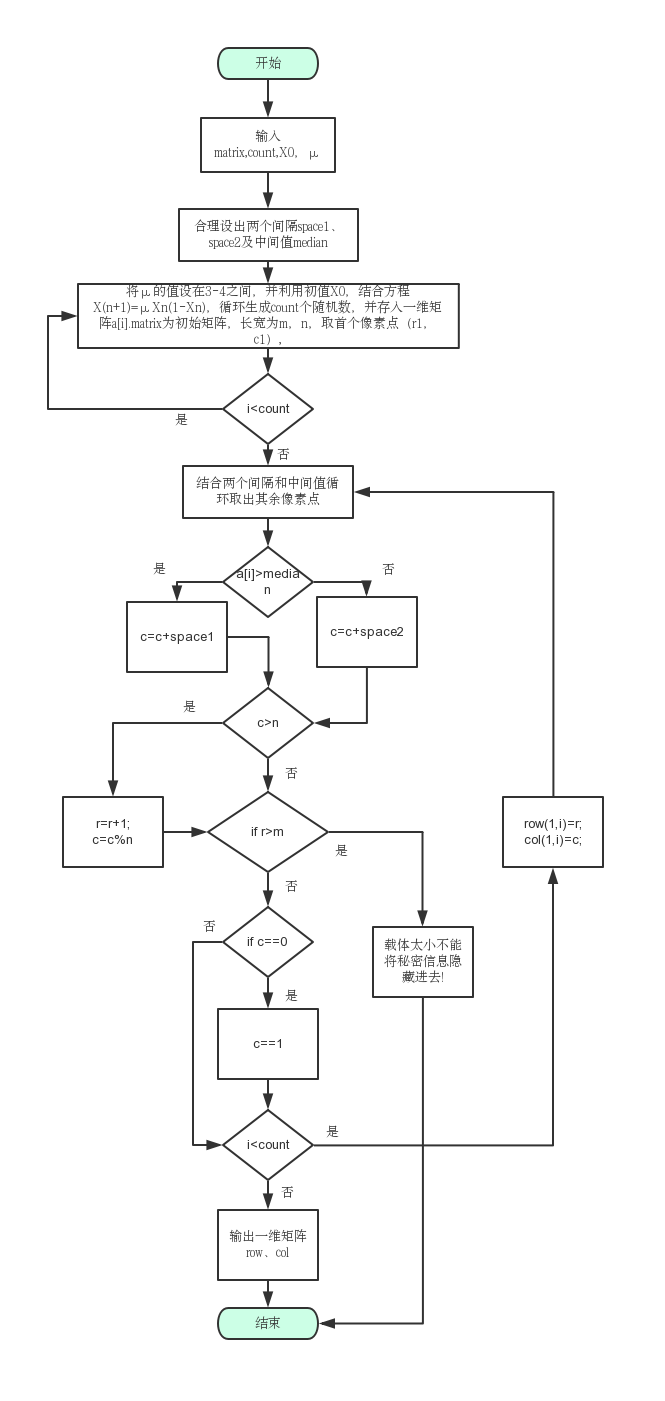
2.4.2、代码实现chaos.m







2.4.3、混沌算法流程图



流程图1、混沌算法流程

# 三、水印算法

3.1、对图像亮度值的分析

3.1.1、原理

一幅图像的亮度必须用YCbCr( YUV)模型中的Y分量才能具体体现,YUV图像中Y为亮度分量。

如果图片是RGB格式图片，像素点 P( r, g, b) 的每一个分量经过double（P）/255后，保证分量的值在( 0, 1) 区间的。

如果图片是YUV格式图片，像素点 P( Y, U, V) 的每一个分量经过double（P）/65535后，保证分量的值在( 0, 1) 区间的。

YUV图像转化为RGB图像、RGB图像转化为YUV图像，可以通过matlab中的两个函数进行转换，分别是YUV=rgb2ycbcr(a)、result=ycbcr2rgb(YUV)。经过转换之后，我们应尽量将分量的值分布在( 0, 1) 之间。

那么我们怎么将抽象的亮度概念进行具体化呢？

我们可以作出如下的定义。

定义：我们将YUV图像的Y亮度分量层提取出来，将其某个像素点提高一个亮度即在该像素点相应的Y值上加1/128=0.0078，若将其某个像素点降低一个亮度，则只需要在该像素点相应的Y值上减0.0078，即我们将亮度分量又划分为128个单位。

接下来将某一副图像的部分像素点的Y值加上一个亮度单位，观察情况如何。

3.1.2、代码实现compareBright.m





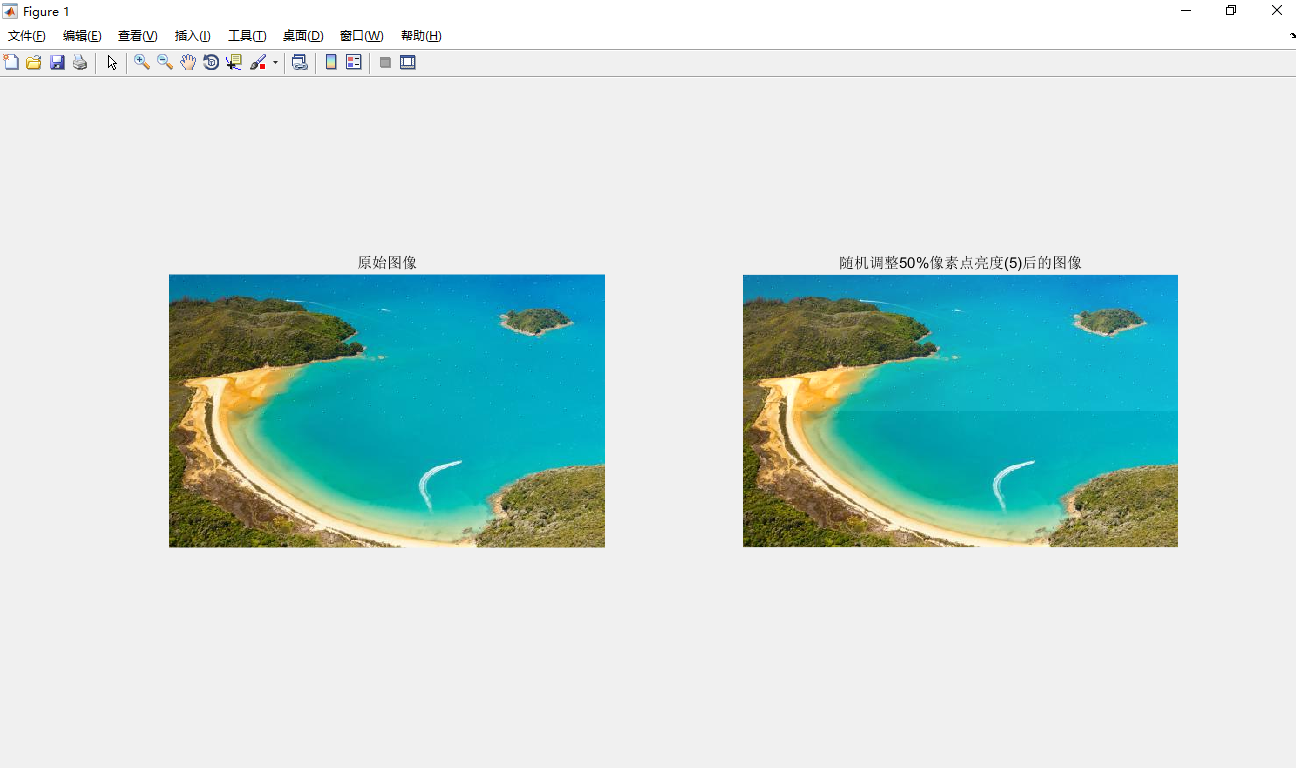
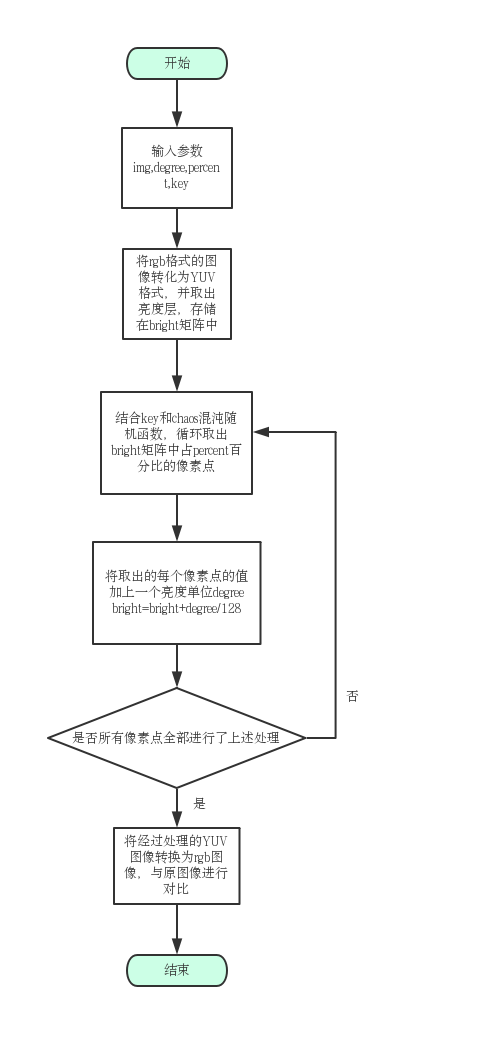


图8、亮度分析

3.1.3、亮度分析流程图



流程图2、亮度分析流程

3.2、水印嵌入算法

3.2.1、原理

通过分析我们清楚地认识到对图像亮度的轻微改变与对图像色彩轻微改变一样，都较难引改变人眼所看到的视觉效果，这就在水印嵌入时，给我们提供了非常实用的秘密信息隐藏空间。

我们将RGB图像转换为YUV图像后，选择亮度空间层，存入一个二维矩阵cover当中，并将水印信息化为二进制，也存入一个一维矩阵msg，再基于虫口模型或混合光学双稳模型随机在二维矩阵中选择像素点及下一个相邻像素点。

假如msg的第k比特对应的的亮度分量隐藏位为cover(m,n)，且一个亮度单位为degree

If(msg(k) == 0)

cover(m,n)- degree；

cover(m,n)+degree;

Else

cover(m,n) + degree；

cover(m,n) - degree;

具体代码为：

for i=1:count

if msg(i,1)==0

bright(row(i),col(i))=bright(row(i),col(i))-degree;

bright(row(i)+1,col(i)+1)=bright(row(i)+1,col(i)+1)+degree;

else

bright(row(i),col(i))=bright(row(i),col(i))+degree;

bright(row(i)+1,col(i)+1)=bright(row(i)+1,col(i)+1)-degree;

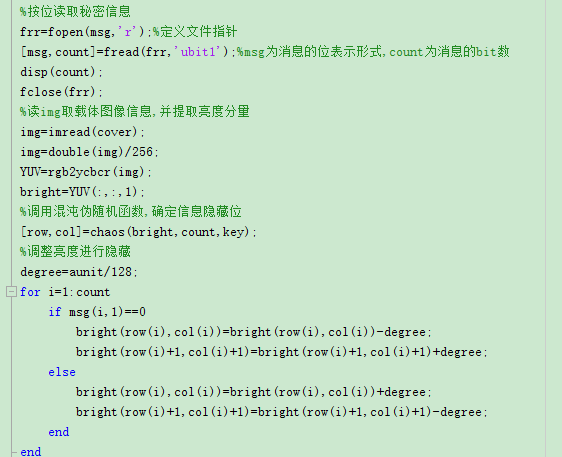
end

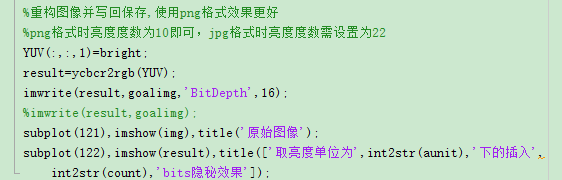
end

因为我们插入的水印信息已经转化为了二进制的0/1，且两相邻像素点之间的亮度值相差不大，我么可以根据水印信息中的二进制中的0/1, 决定我们随机取出的两随机点的大小，并合理设置degree的值，就可以明显比较两相邻像素点的大小，来进行水印的隐藏。

3.2.2、代码实现hide.m







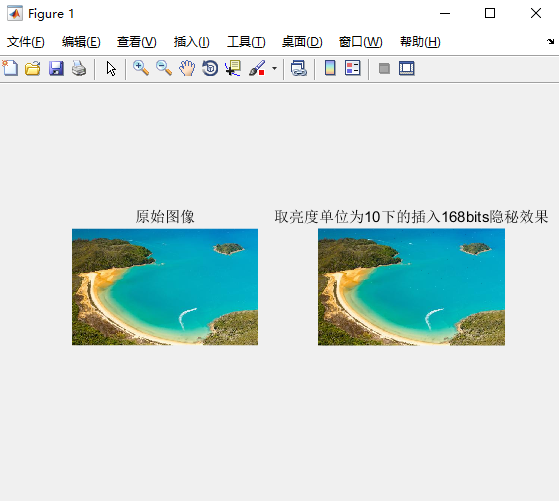
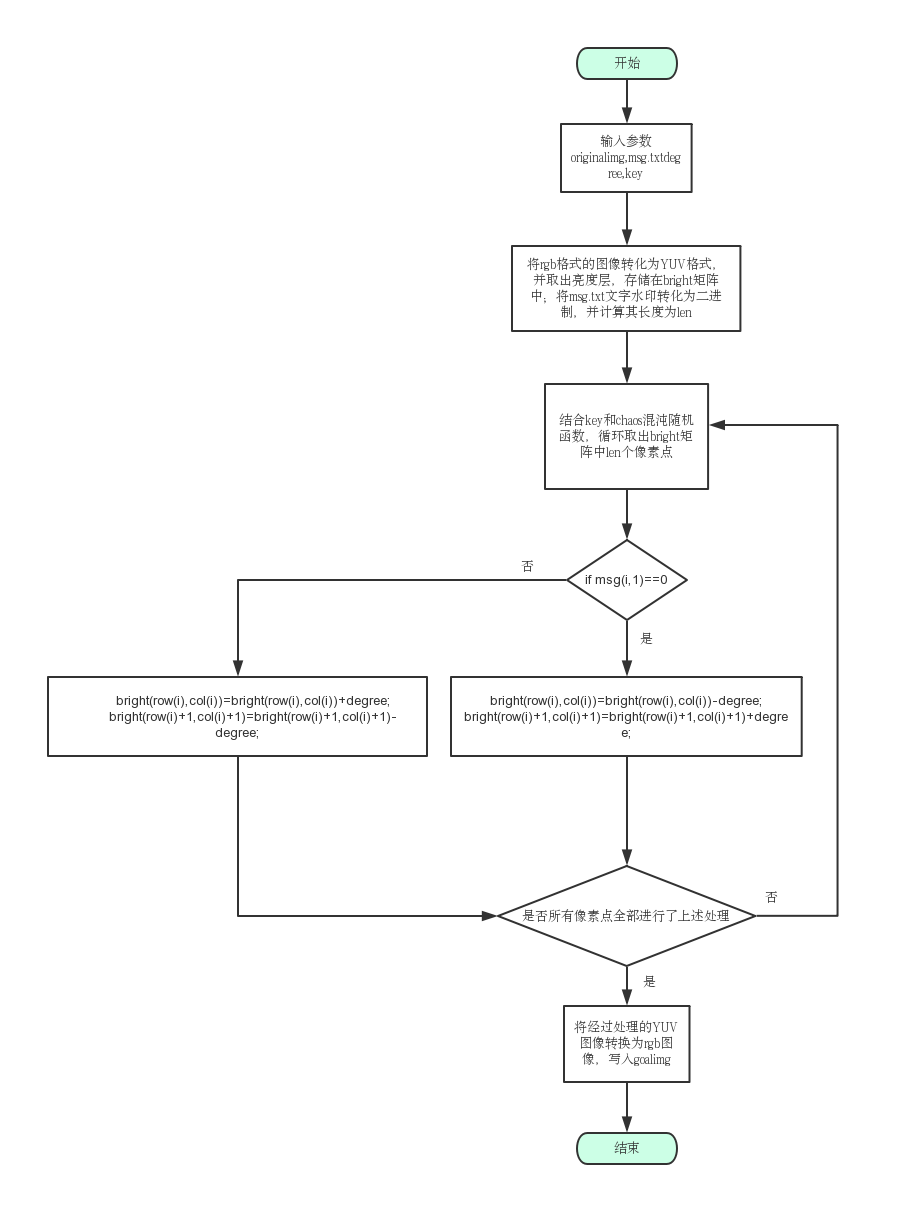


图9、水印嵌入图

3.2.3、水印嵌入流程图



流程图3、水印嵌入流程

3.2、水印提取算法

3.3.1、原理

水印的提取是根据相邻两个像素点的亮度值在原图像中相差不大的原理，进行水印的插入，将前一个像素的亮度分量减（加）去一个亮度单位，又将后一个像素的亮度分量加（减）一个亮度单位，那么两个像素点的亮度分量就能明显比较大小。

假如msg的第k比特对应的的亮度分量隐藏位为cover(m,n)，且一个亮度单位为degree

If(cover(m,n)<cover(m,n+1))

回写0；

Else

回写1；

具体代码为：

for i=1:key1

if bright(row(i)+1,col(i)+1)-bright(row(i),col(i))>=0

fwrite(frr,0,'ubit1');%回写0

else

fwrite(frr,1,'ubit1');%回写1

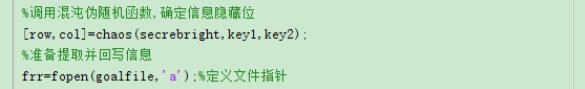
end

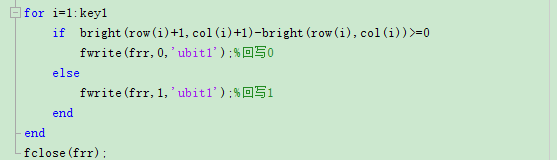
end

进行0/1的回写之后，就可以还原水印信息，达到提取水印的目的。

3.3.2、代码实现extract.m







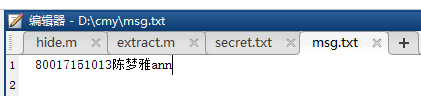


图10、水印信息

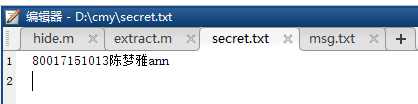
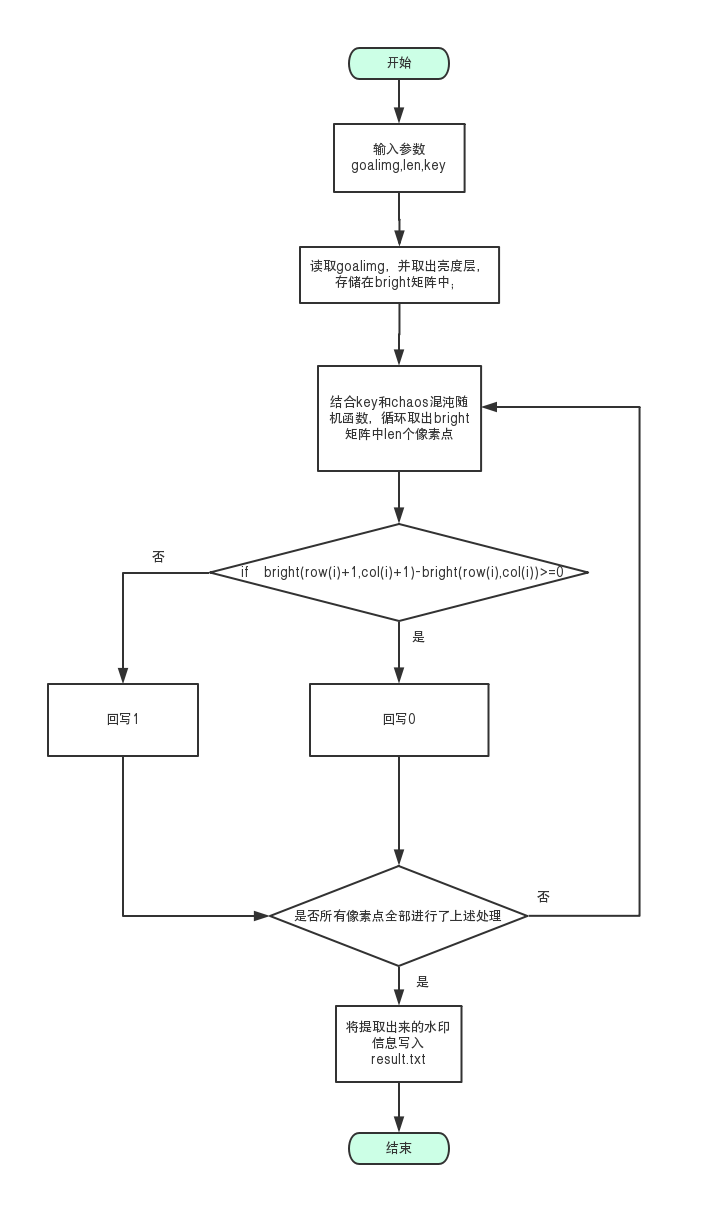


图11、提取水印

3.3.3、水印提取流程图



流程图4、水印提取流程

# 四、测试

锐化滤波可分为线性和非线性，核心在于算子的不同，因算子唯一，无法绘制相关性系数曲线。而对图像的旋转，改变大小，剪切，每次攻击后要更改算法才能得到相应的相关系数，且为空域算法，无法抵抗该攻击，所以未绘制相关性系数曲线。

## 4.1、中值滤波

中值滤波是图像处理中常见的滤波类型，用来去除噪声，可分为一维和二维，本文采用的时二维中值滤波，原理是将模板中的数据从小到大排列，取出中值，并用该中值取代模板中心的值，达到滤波的效果。

### 4.1.1、代码实现median.m



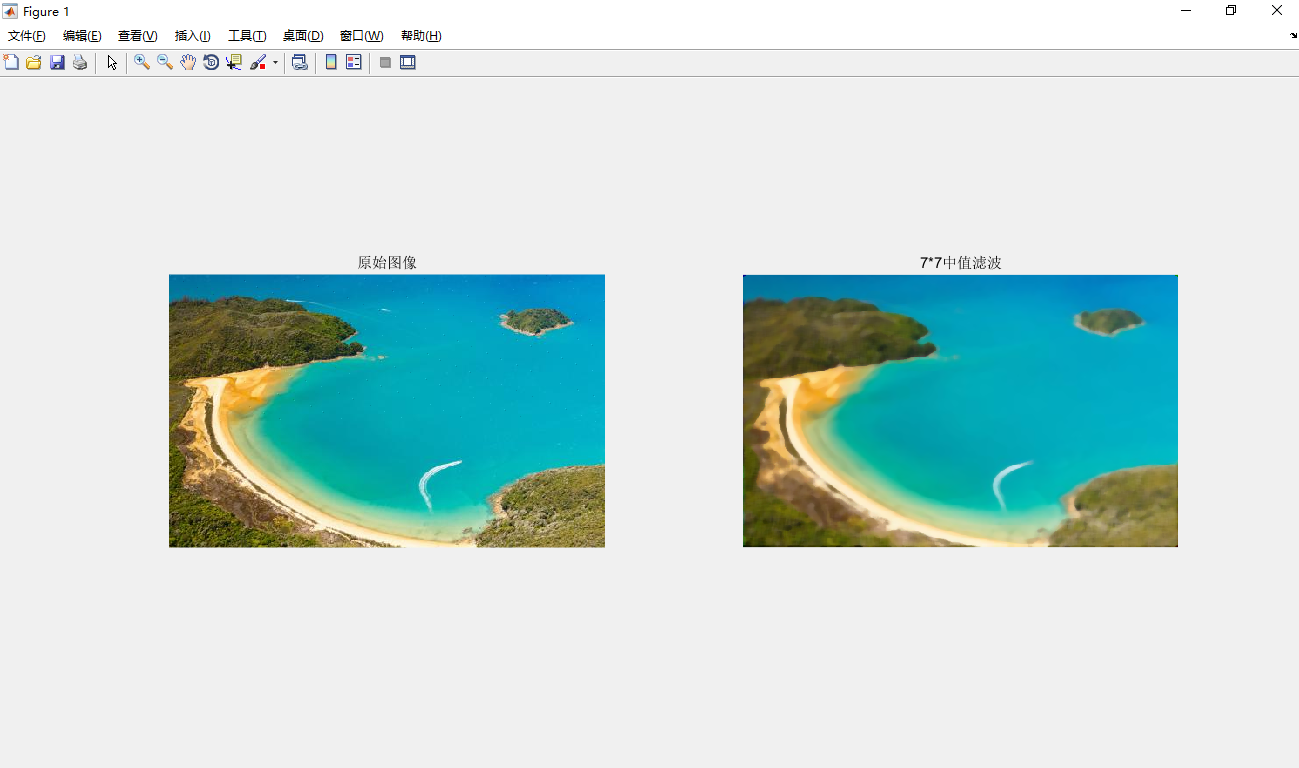


图12、中值滤波攻击

### 4.1.2、相关性检测plotmedian.m





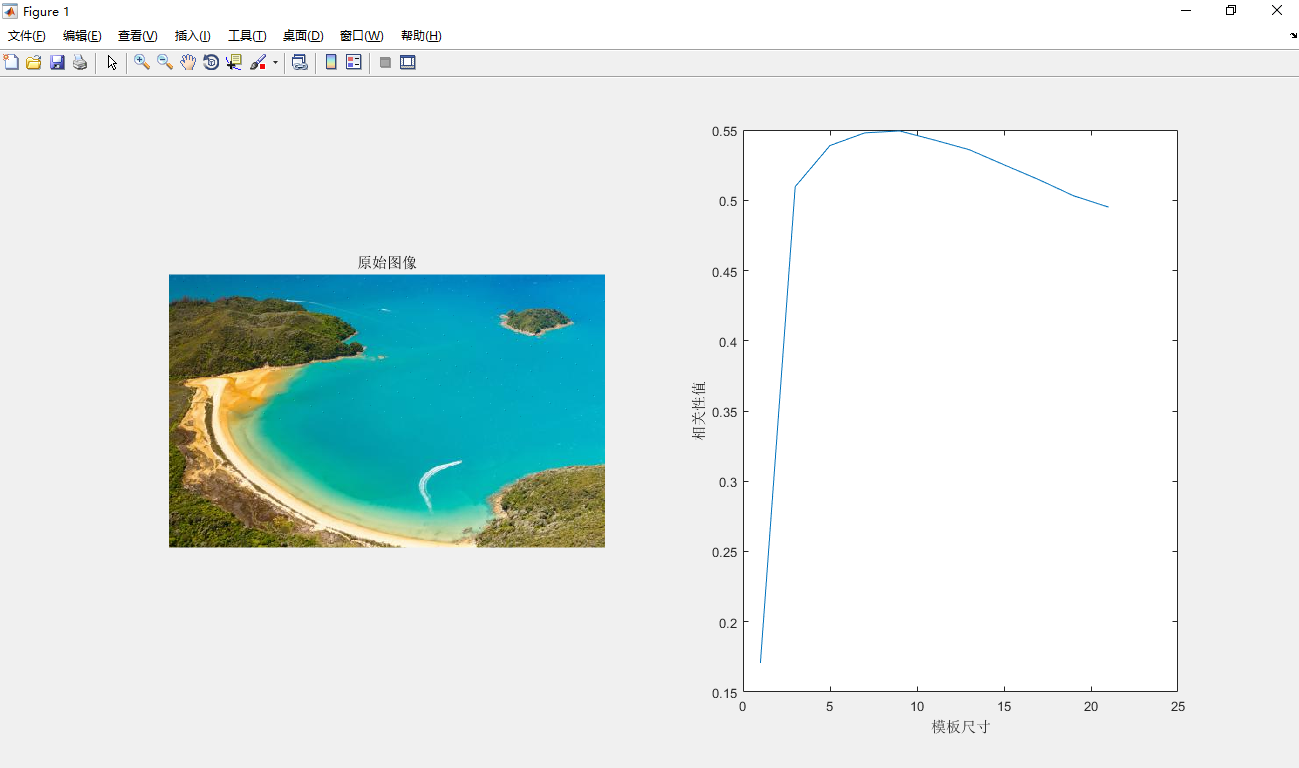


图13、加入DCT变换的中值滤波相关性值曲线

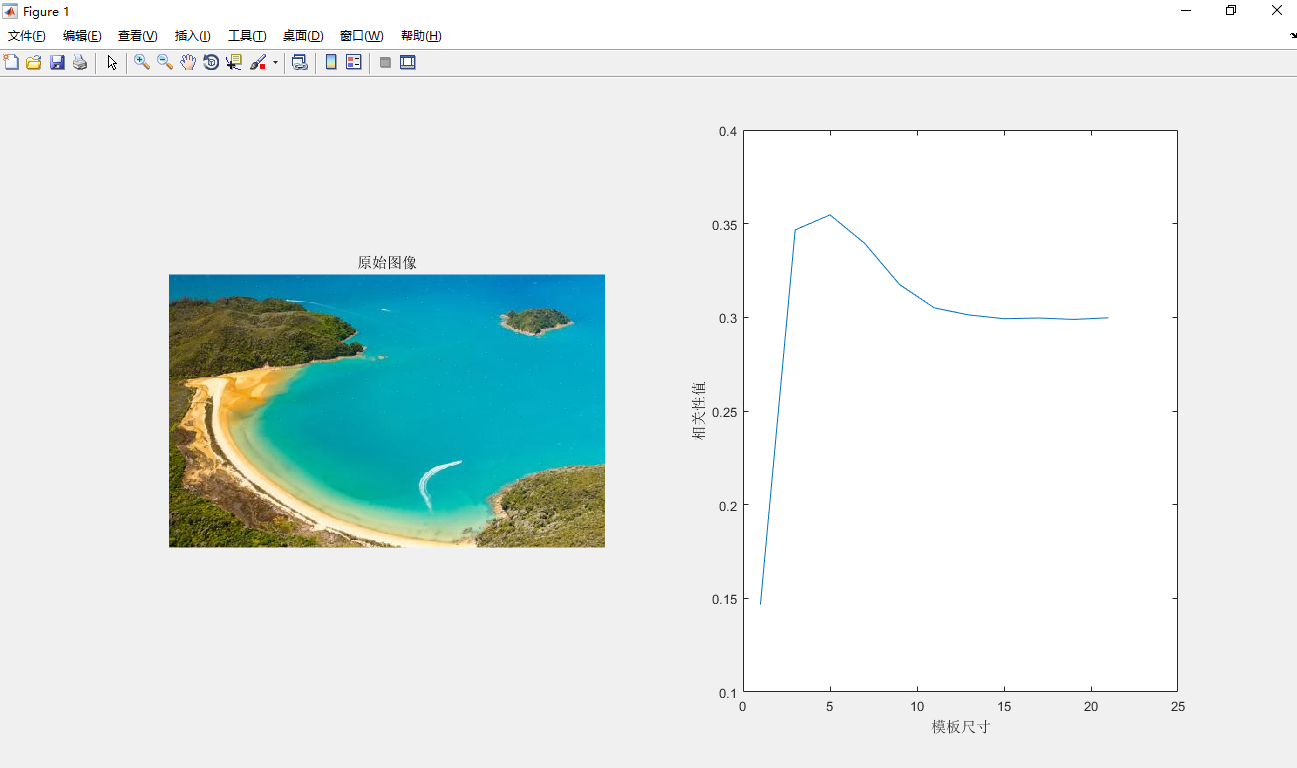


图14、未加入DCT变换的中值滤波相关性值曲线

随着模板尺寸的增大，检测到的相关性值先增大后平缓减小，其中当模板尺寸为8时，对水印的破坏最小，抵抗攻击的能力最强。

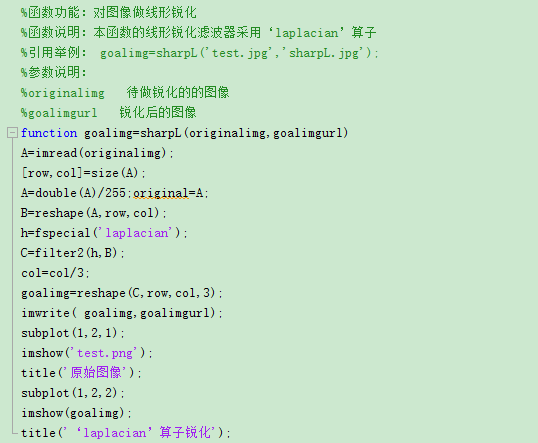
## 4.2、锐化滤波

锐化是空域滤波增强的一种方法，对于不同的滤波增强来说, 它们的共同点是将图像与模板进行卷积, 不同的是模板不同,我们可以把图像抽象成为一个大矩阵，每一个像素点就是矩阵中的一个元素，而模板就是一个小矩阵，其中图像的长和宽最后时模板的长和宽的整数倍，将模板作用在图像上,进行反复迭代卷积运算，将图像对应模板大小的值全改为进行反复迭代卷积运算后的得到的值。

锐化也可分为线性锐化和非线性锐化，但是原理是相同的。若模板为[(1,1),(2,2)],图像中某一区域为[...,(3,3),(4,-4),...],该区域进行卷积运算K=1\*3+1\*3+2\*4+2\*(-4)=6，则图像的该区域为[...,(6,6),(6,6),...]。

### 4.2.1线性锐化

#### 4.2.1.1、代码实现sharpL.m



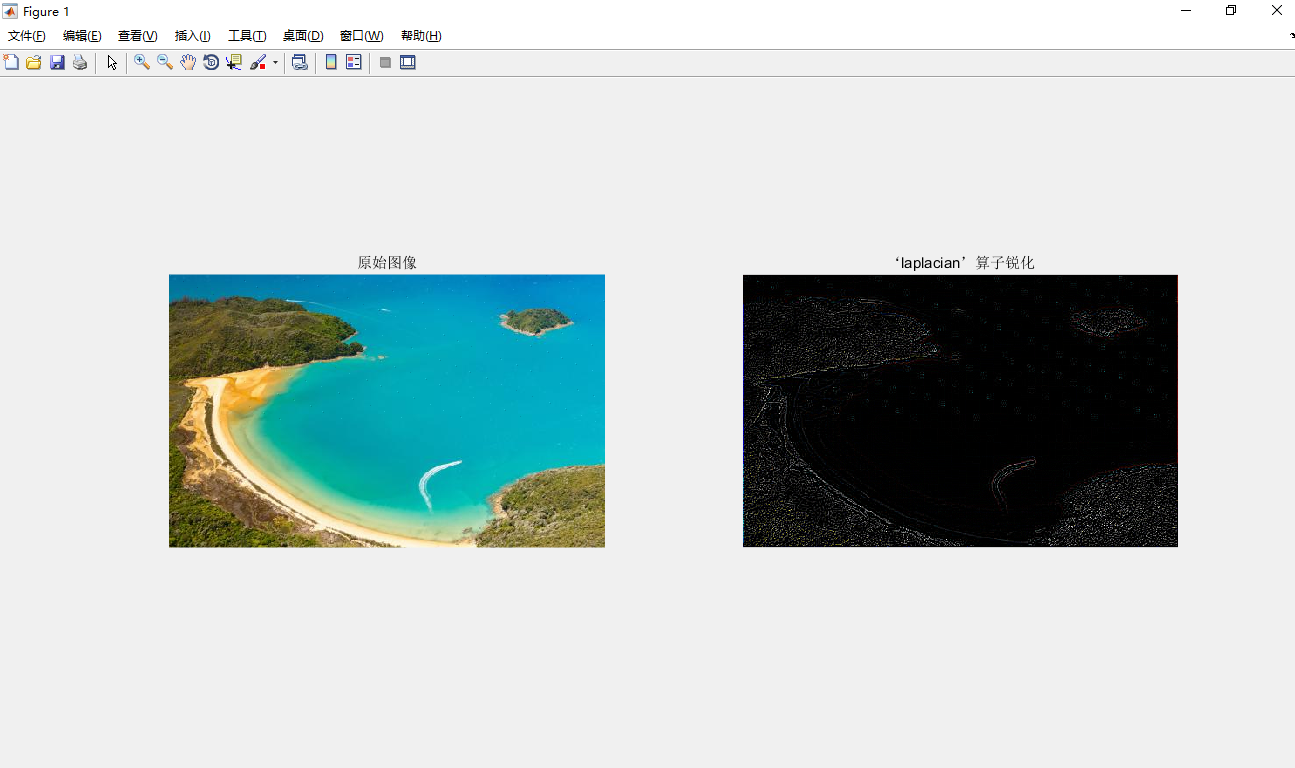
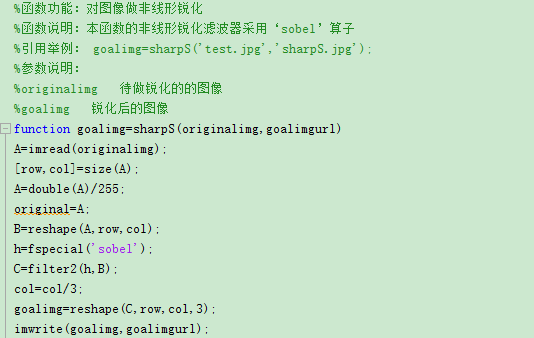
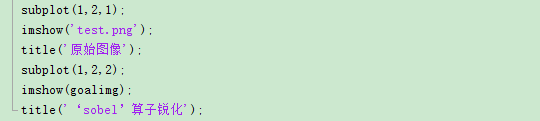


图15、线性锐化攻击

### 4.2.2非线性锐化

#### 4.2.2.1代码实现sharpS.m





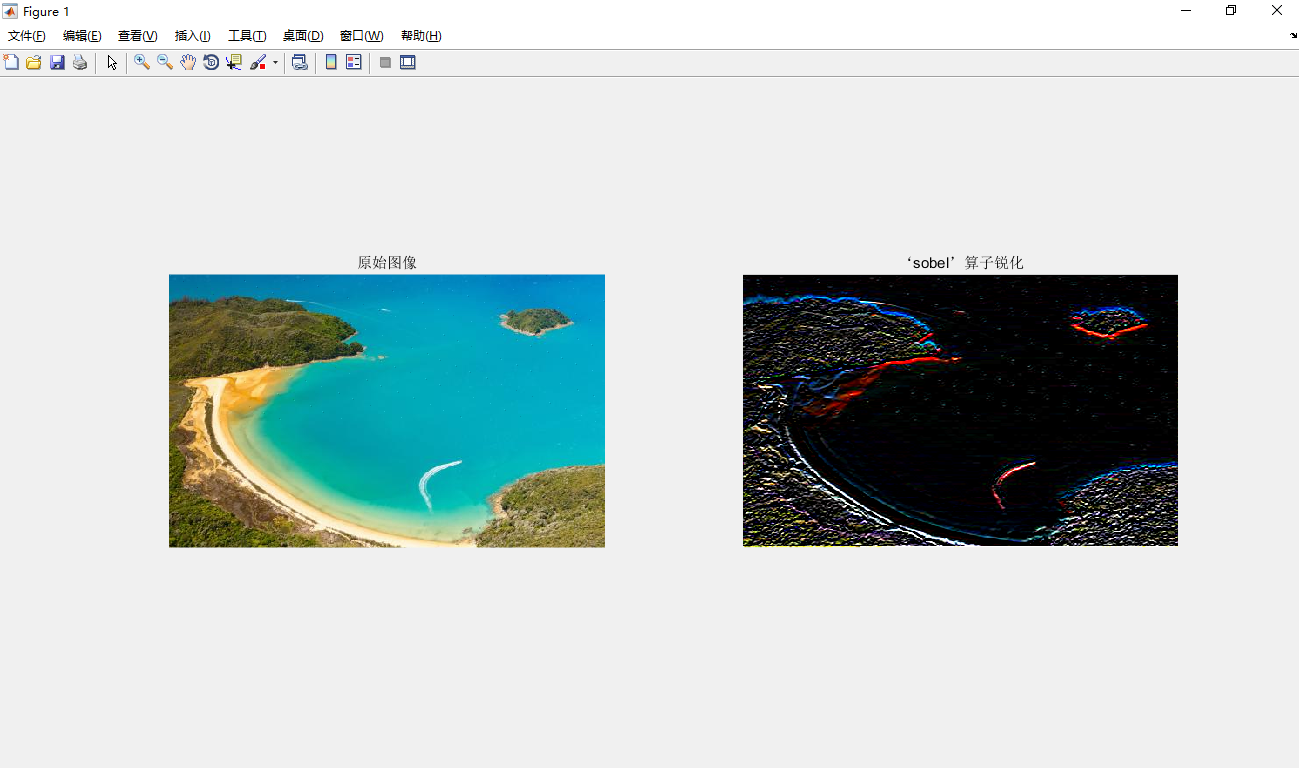


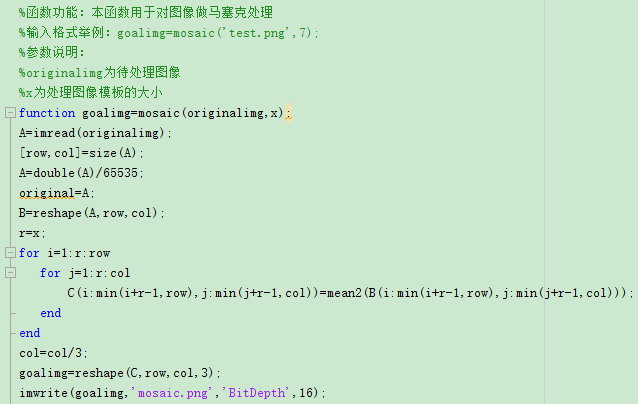
图16、非线性锐化攻击

进行锐化滤波时，几乎改变了所有像素点的值，也就更改了隐藏了秘密信息的亮度值，所以对图像进行锐化时，无法抵抗锐化滤波攻击。

## 4.3、马赛克攻击

有时我们不想要图片的某一部分被看到，常在该部位打上马赛克，或者有时因为存储介质被损坏，都会出现马赛克现象，所以马赛克攻击也是常见的无意攻击之一，而本文中的马赛克攻击和上述的中值滤波和锐化滤波类似，原理很简单, 选定模板尺寸，将该模板应用到图像中，将图像中该模板中所有的值累加取平均值，然后用该平均值取代模板中的所有值。如[13,14,15],经马赛克攻击后，为[14,14,14].

### 4.3.1.代码实现mosaic.m





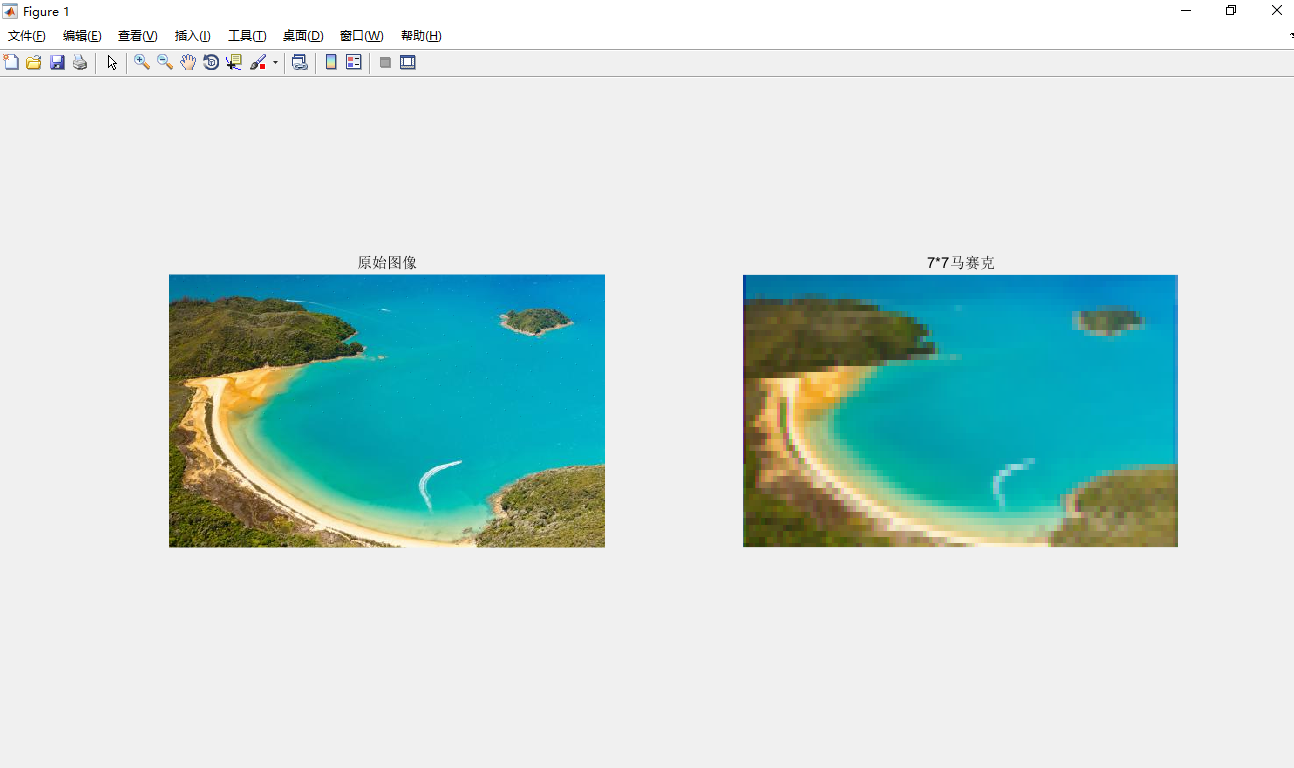


图17、马赛克攻击

### 4.3.2、相关性检测plotmosaic.m





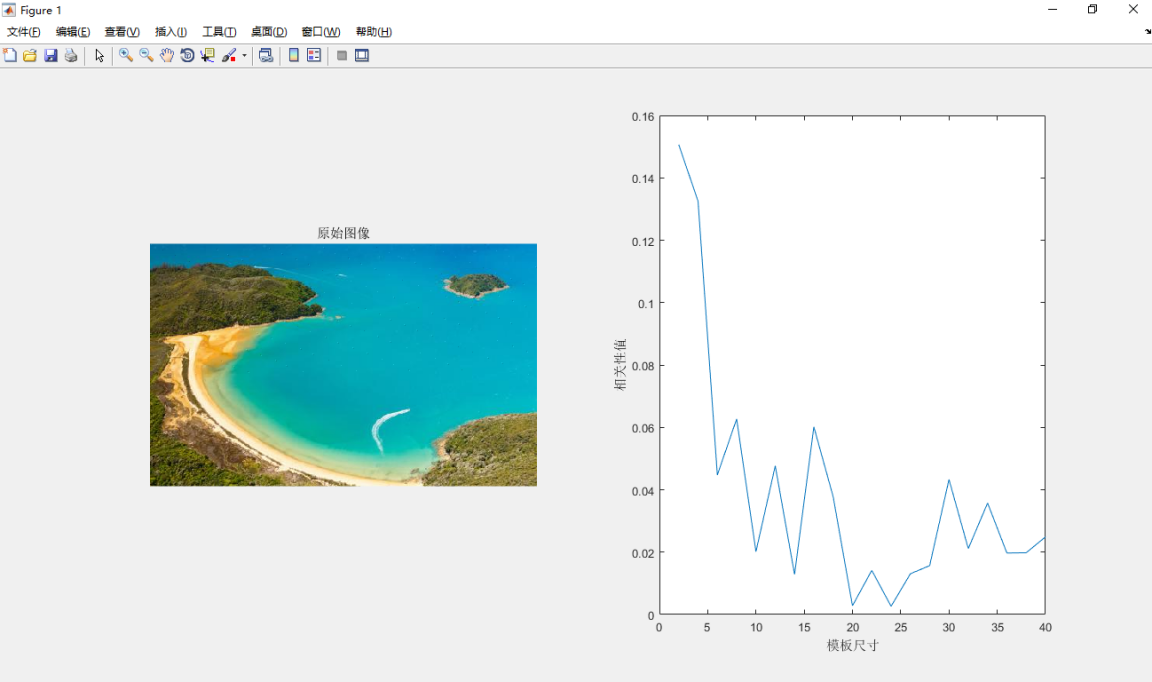


图18、加入DCT变换的马赛克攻击相关性值曲线

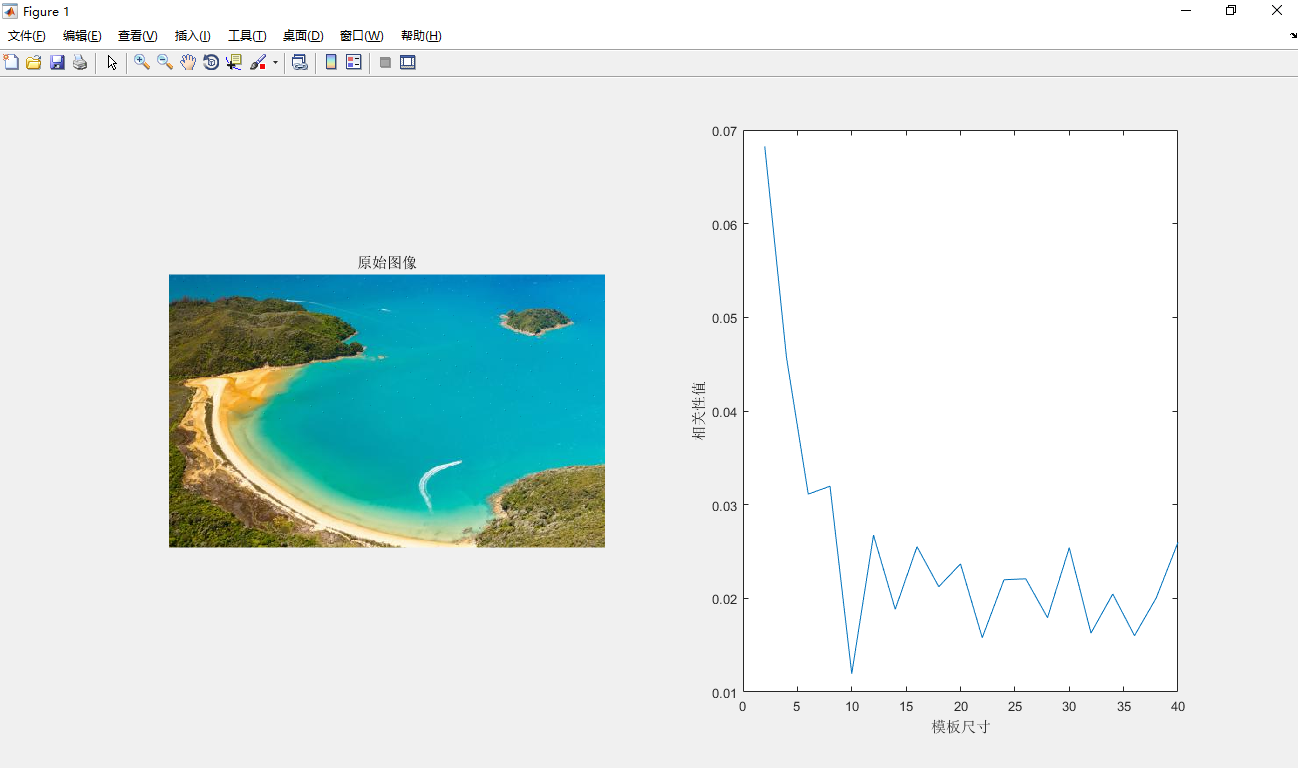


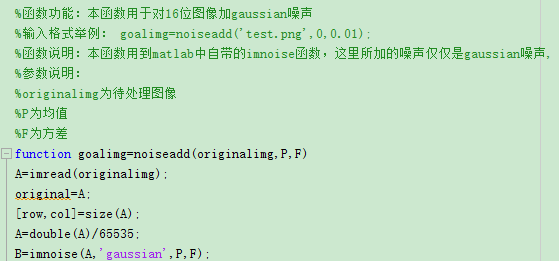
图19、未加入DCT变换的马赛克攻击相关性值曲线

随着模板尺寸的增大，检测到的相关性越小，说明对水印的破坏越大，当模板尺寸大小维持在2以内时，抵抗攻击的能力最强，但是相关性检测的值却在0.1以下，所以不足以抵挡马赛克攻击的。

## 4.4、加噪攻击

图像在传播过程中，噪声攻击无疑就是最容易受到的攻击之一，它对嵌入的水印无疑也会造成一定的影响，下文以加入高斯噪声为例。

### 4.4.1、代码实现noiseadd.m



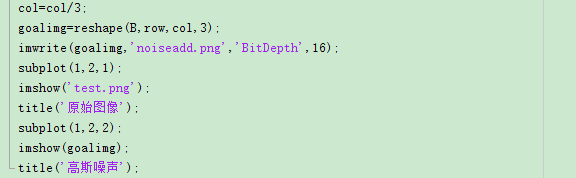
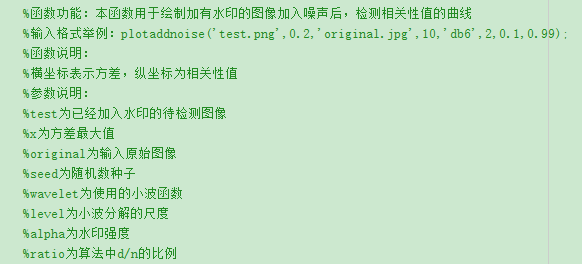
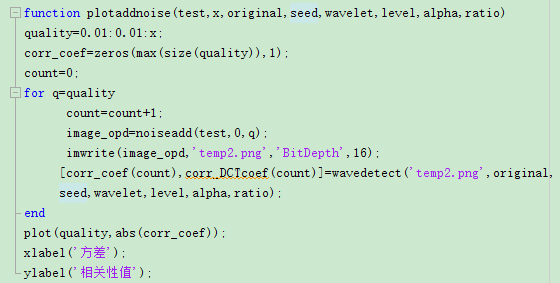




图20、加噪攻击

### 4.4.2、相关性检测plotaddnoise.m





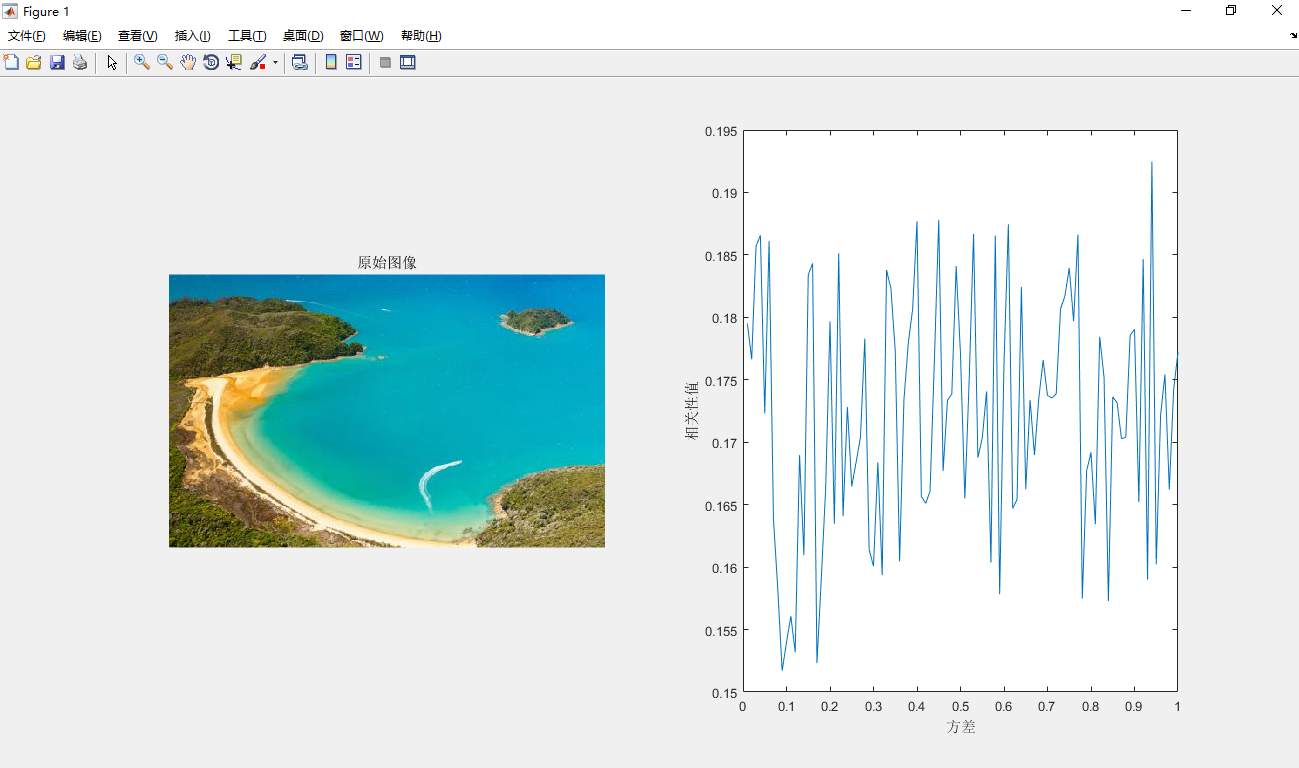


图21、加入DCT变换的加噪攻击相关性值曲线

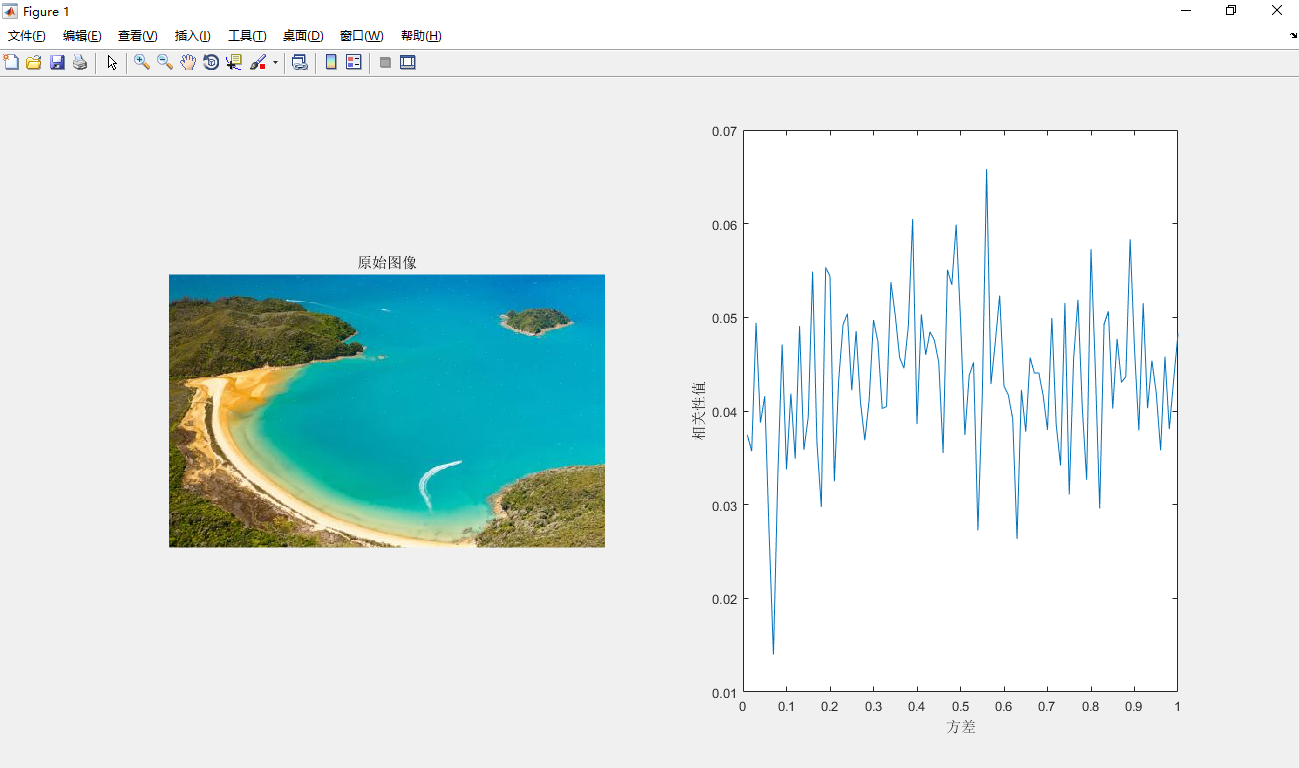


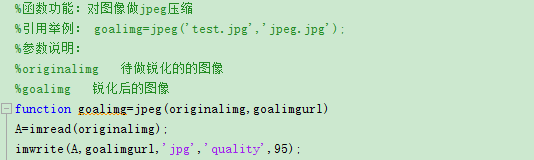
图22、未加入DCT变换的加噪攻击相关性值曲线

如图22，随着方差的增大，相关性始终在0.17周围变化，但相关性较小，对水印破坏度较大，经测试，无法抵抗加噪攻击。

## 4.5、jpeg压缩

jpeg压缩是图像处理中最普遍的方法，典型的无意攻击，本文利用写入函数imwrite( data, ′1 . jpg′, ′jpg′, ′quality′, compress) 进行jpeg压缩攻击，其中compress代表压缩比。

### 4.5.1、代码实现jpeg.m





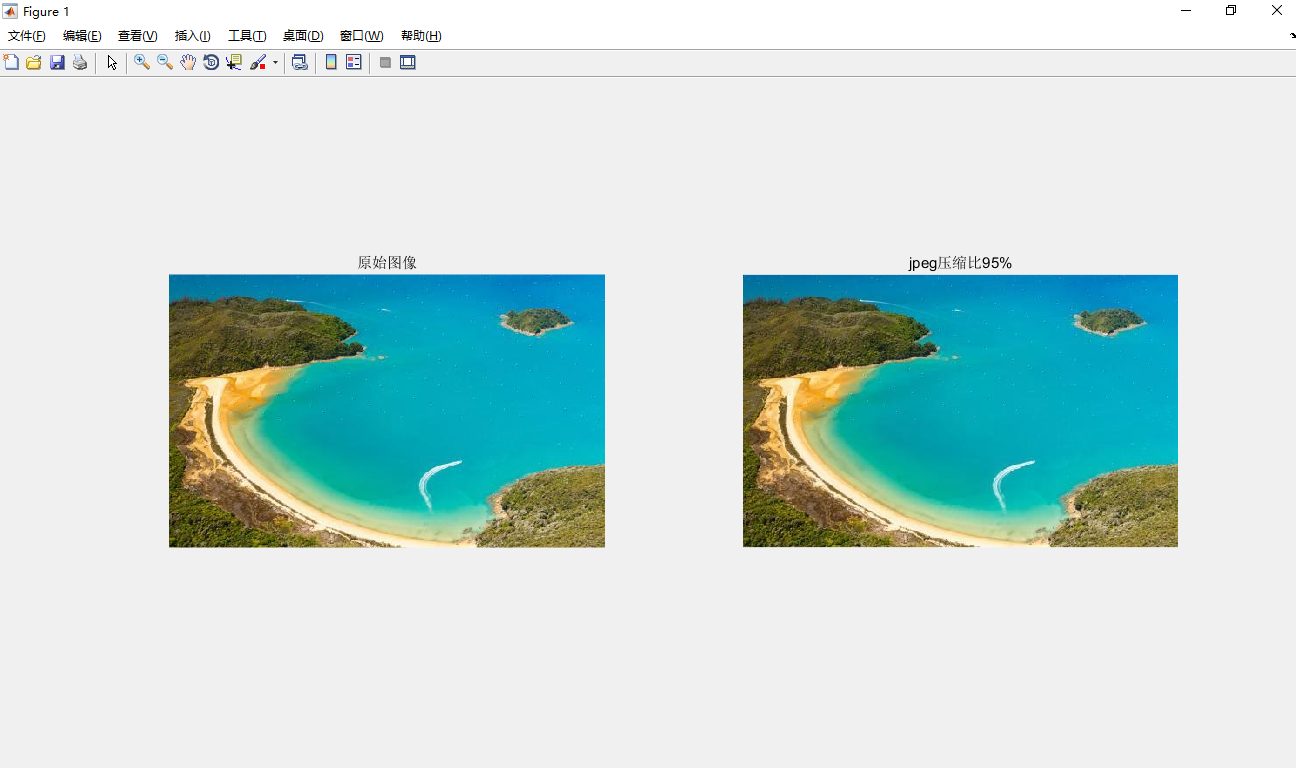
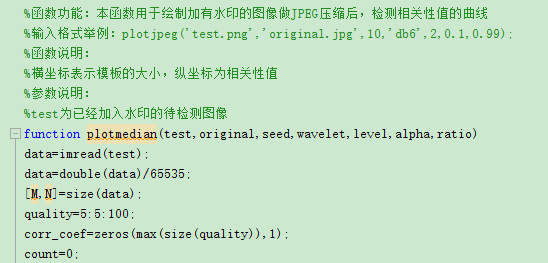
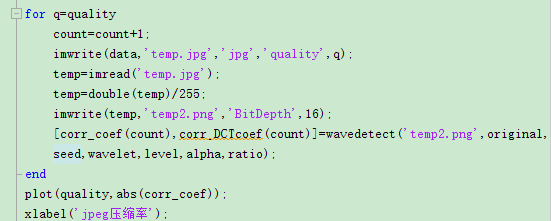


图23、JPEG攻击

### 4.5.2、相关性检测plotjpeg.m





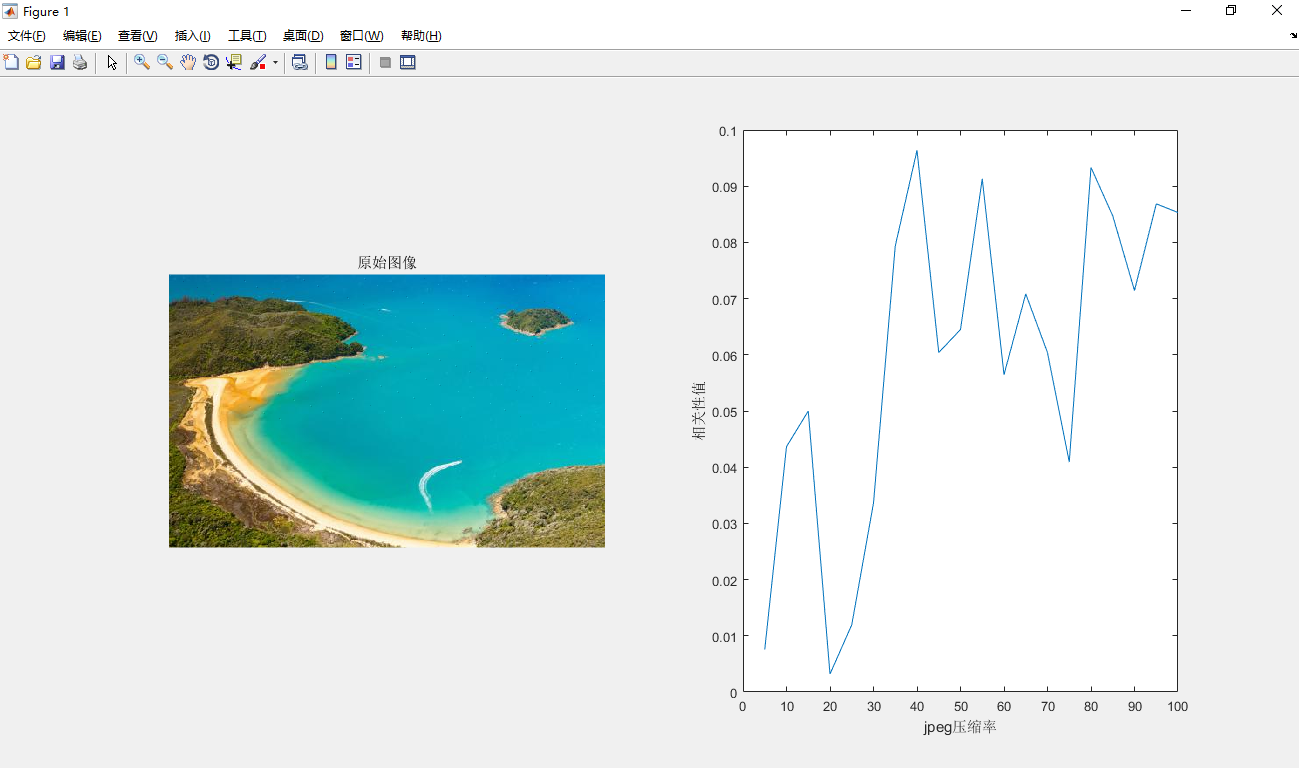


图24、加入DCT变换的JPEG压缩攻击相关性值曲线

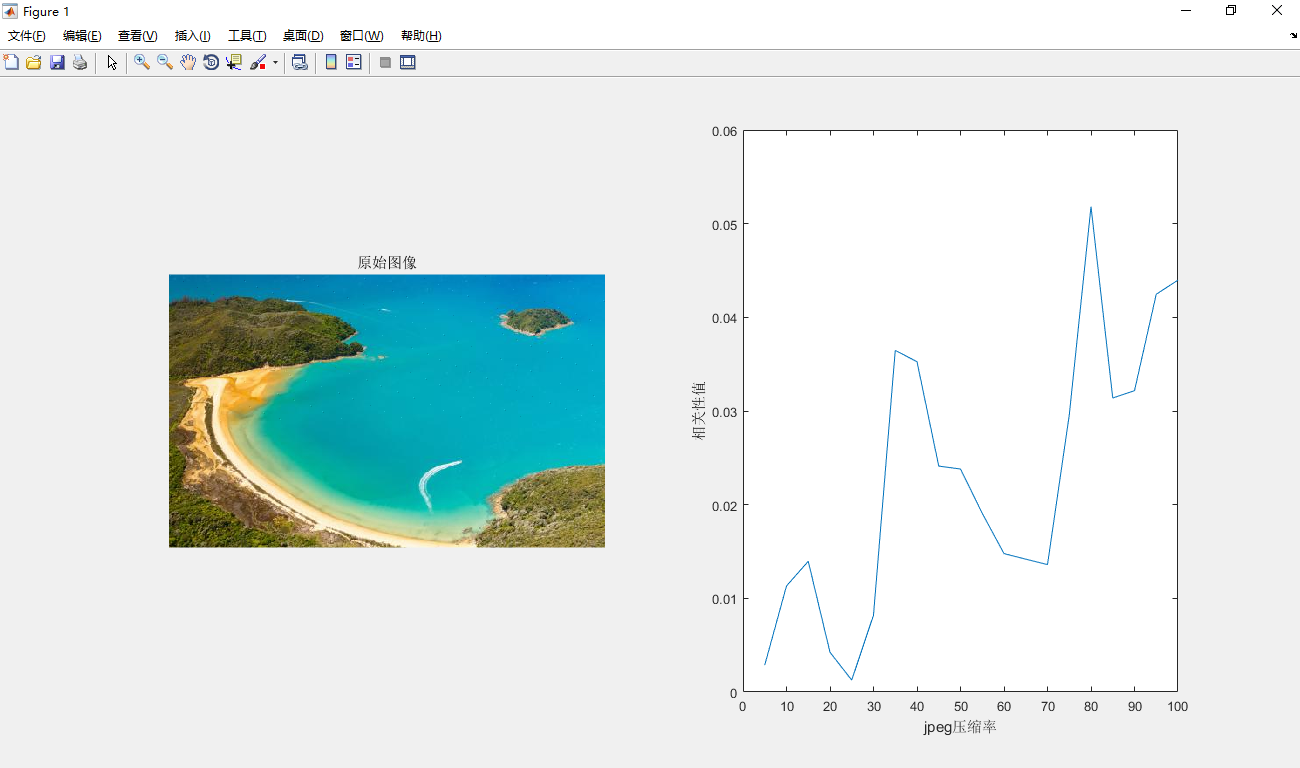


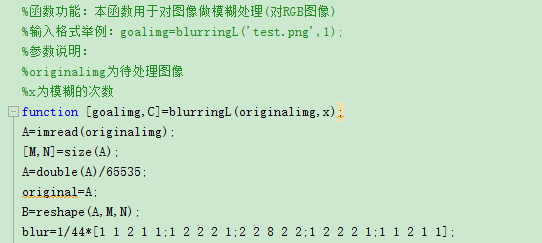
图25、未加入DCT变换的JPEG压缩攻击相关性值曲线

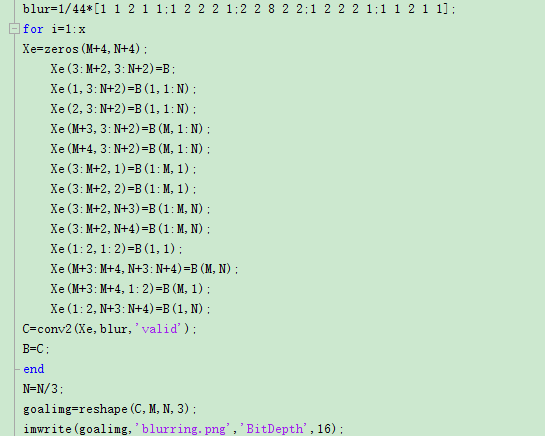
压缩的越厉害，对图像的破坏越大，压缩率大于0.95，当压缩比接近1时，经实验是可以抵挡jpeg压缩攻击的

## 4.6、模糊处理

模糊处理图像处理中不可或缺的方法。这种方法的原理非常简洁, 利用一个二维的卷积就可以完成, 本文使用MATLAB 中的conv2 函数来实现这个卷积。

### 4.6.1、代码实现blurringL.m







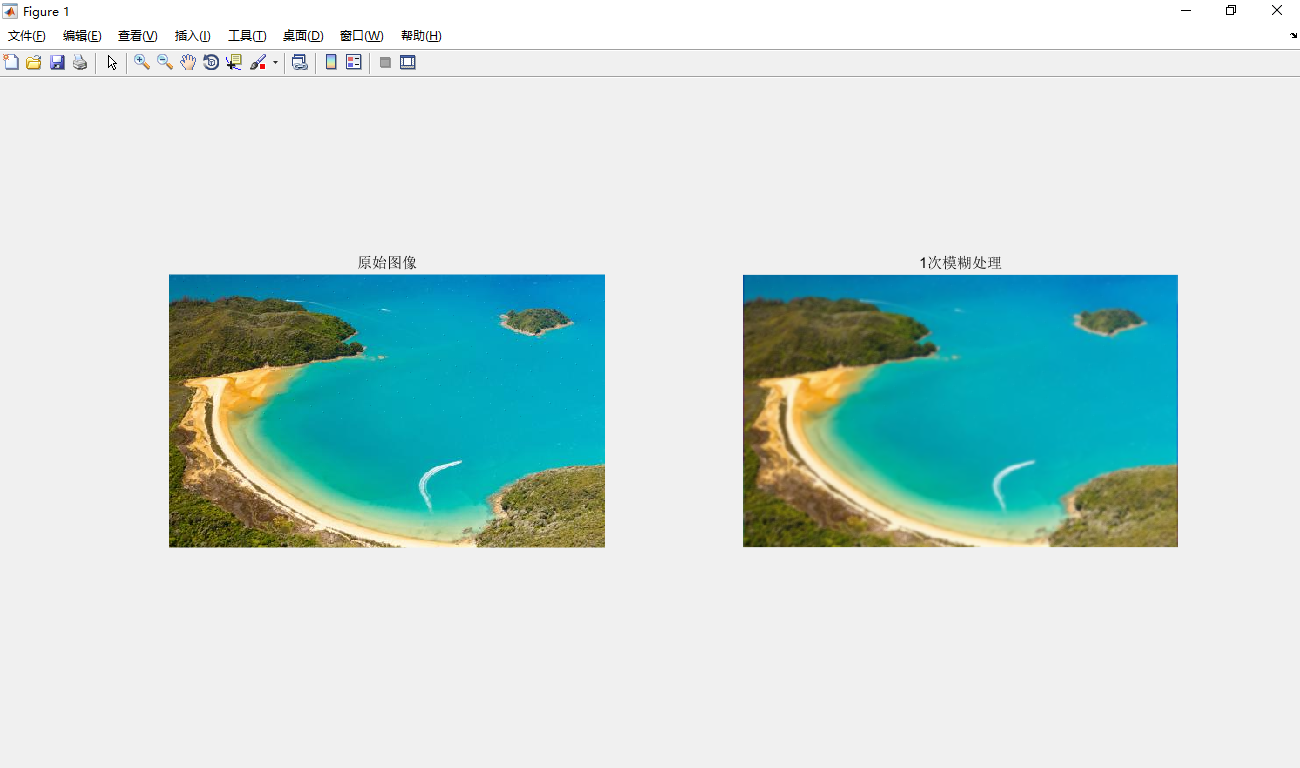
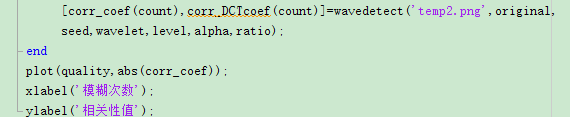


图26、模糊处理攻击

### 4.6.2、相关性检测plotblurring.m





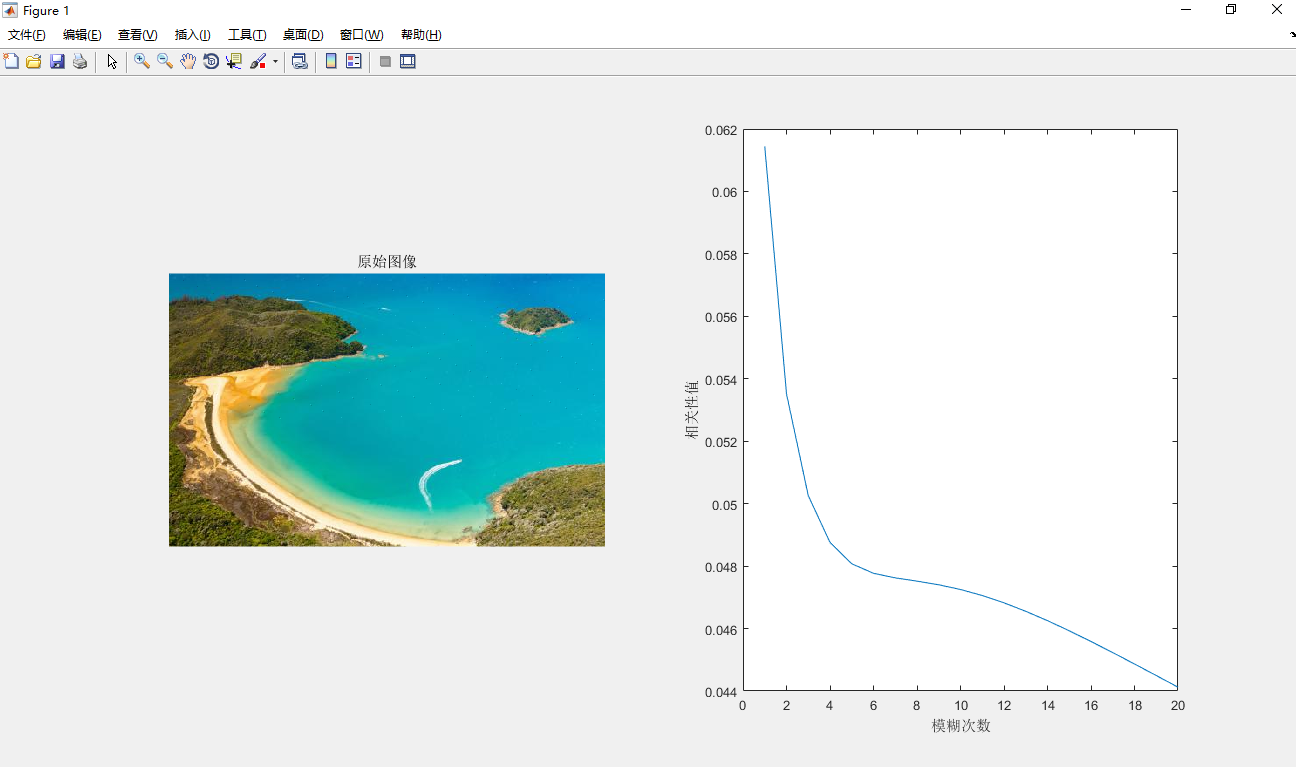


图27、加入DCT变换的模糊处理攻击相关性值曲线

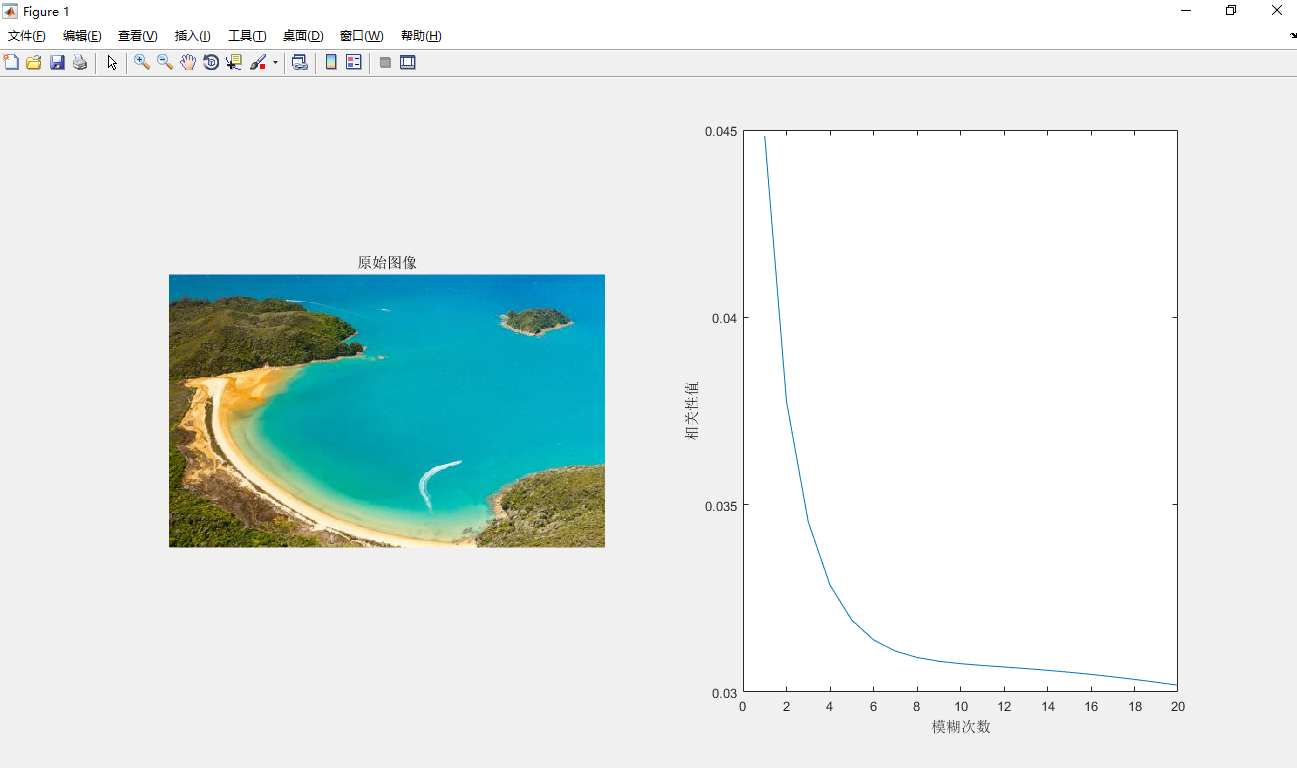


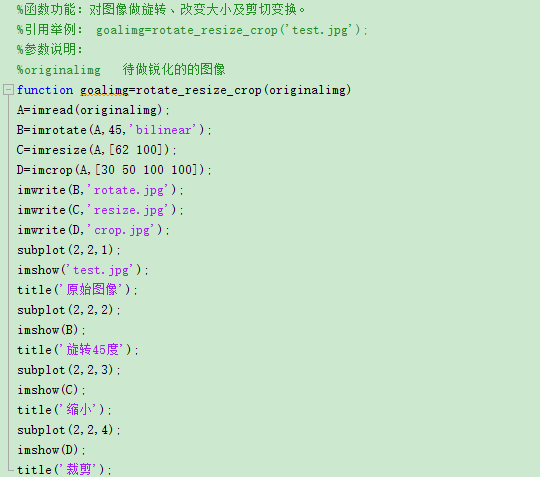
图28、未加入DCT变换的模糊处理攻击相关性值曲线

随着模糊次数的增大，检测到的相关性值越小，对水印的破坏越大，当模糊次数在2以内时，但因为相关性检测时值在0.1以下，相关性值太小，所以无法抵挡模糊处理攻击。

## 4.7、旋转、改变大小和裁剪

在图像的处理中,经常会对图像进行旋转，改变尺寸和裁剪，跟模糊处理一样，matlab也提供了相关的函数，分别为imcrop、imrotate、imresize。

### 4.7.1、代码实现rotate\_resize\_crop.m



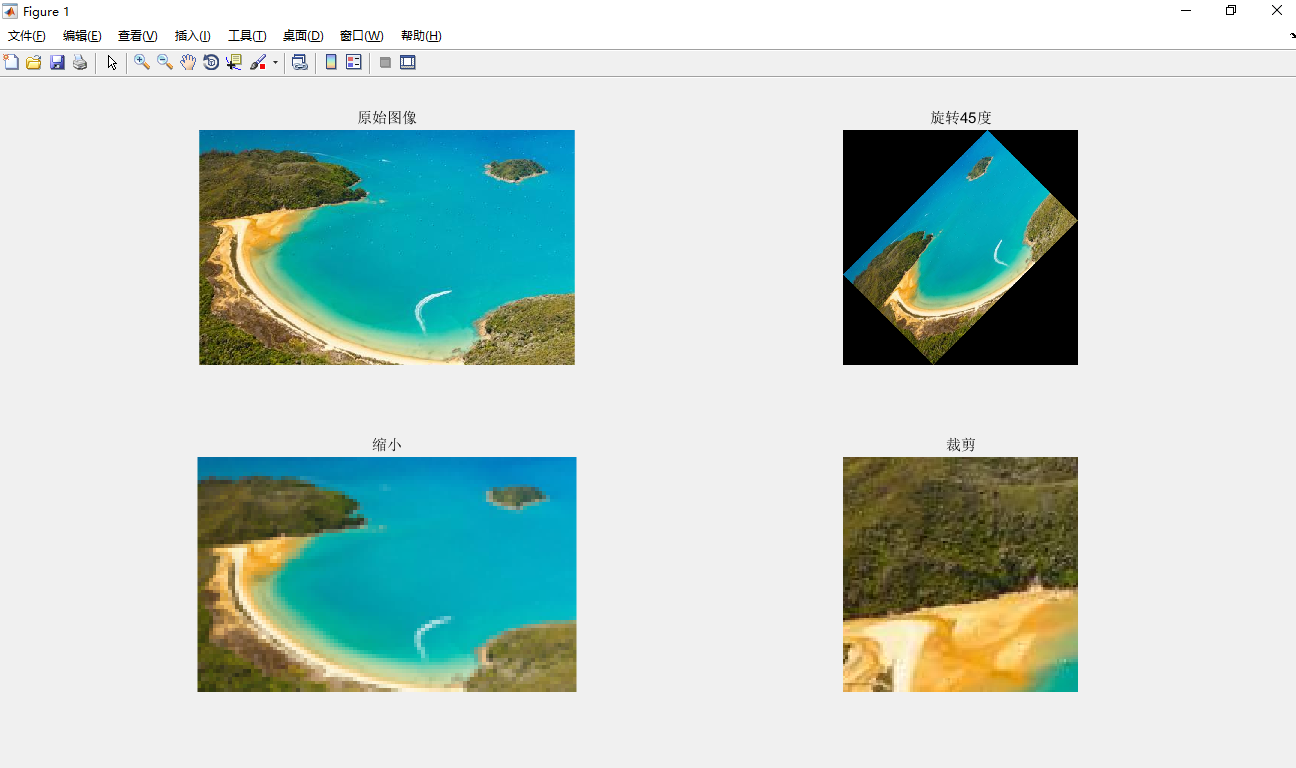
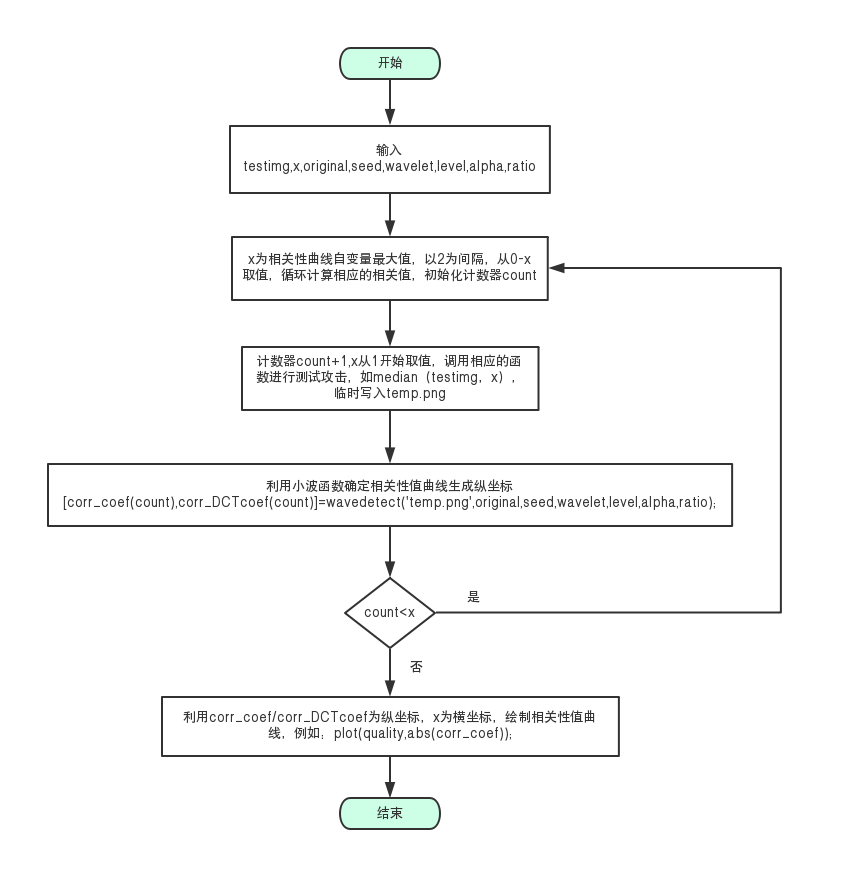


图29、旋转、剪切、改变大小攻击

因为算法为空域隐藏，隐藏的信息分布在整张图片中，旋转操作会改变像素点的位置，裁剪操作只能得到部分的像素点，改变大小会增加或减小像素点的数目，所以无法抵挡旋转、裁剪、改变大小攻击。

## 4.8、测试算法流程图



流程图5、测试流程

## 

## 4.9、测试总结

|  |  |  |
| --- | --- | --- |
| 攻击 | 能够抵抗 | 原因 |
| 中值滤波 | 部分抵抗 | 中值滤波作为常见的滤波之一，在去除噪声的同时，可能会造成无意的攻击，正如相关性检测曲线显示的那样，随着模板尺寸的增大，检测到的相关性值先增大后平缓减小，其中当模板尺寸为8时，相关性值达到最大，对水印的破坏最小，抵抗攻击的能力最强，经检测，该点能够抗住攻击。 |
| 锐化滤波 | 不能抵抗 | 锐化滤波跟中值滤波一样，都要采用模板，但不同的是，中值滤波利用模板中的中值替换掉模板中心的值，而锐化滤波利用的是将模板和原图像中对应模板大小的值，与模板中的值一一进行卷积后，将卷积后的值之和替换掉原图像中对应模板中的全部值，这无疑会大大破坏图像，无法抵抗该攻击。 |
| 马赛克攻击 | 不能抵抗 | 在相关性曲线中，随着模板尺寸的增大，相关性值再[1,5]范围内先急剧减小，后上下波动，这表明刚开始随着模板尺寸的增大，抵抗该攻击的能力越弱，到了5之后，水印信息已经完全被破坏，所以无法抵抗该攻击。 |
| 加噪攻击 | 不能抵抗 | 随着方差的增大，相关性始终在0.17周围变化，但相关性较小，这表明水印信息刚开始就应经被破坏，无法抵抗该攻击 |
| JPEG压缩 | 部分抵抗 | 由相关性值曲线可知，当压缩比低于20%时，曲线在相关性值较小的区域来回波动，当压缩比在20%-40%范围时，相关性值急剧增大，当压缩比大于40%时，曲线在相关性值较大的区域来回波动，这表明，压缩比越小压缩的越厉害，当低于20%时，水印已经被完全破坏，经实验，当jpeg大于95%时，是可以抵抗该攻击的。 |
| 模糊处理 | 不能抵抗 | 随着模糊次数的增大，检测到的相关性值在[2,6]范围内急剧减小，说明水印的信息正在被急剧破坏，模糊次数在6之后，则水印信息被完全破坏，所以无法抵抗该攻击。 |
| 旋转、剪切、改变大小 | 不能抵抗 | 这三种操作都是在空域上对图片的破坏，裁剪操作只能得到部分的像素点，旋转操作会改变像素点的位置，改变大小会增加或减小像素点的数目，所以无法抵挡旋转、裁剪、改变大小攻击。 |

# 五、结论

本文基于虫口模型和混合光学双稳模型实现图片盲水印的插入和提取，基于模型而推导出来了相应的公式，将公式中的参数赋予合适的参数值后，产生的数值序列具有了混沌特性，利用该特性，伪随机的选出图片中隐藏水印的像素点，将要插入的水印文字信息，化为二进制序列，然后利用亮度的轻微改变，如加减一个亮度单位，人眼不易察觉和图片中两相邻像素点亮度相差不大的特性，当二进制位为0时，选出的两个相邻像素点左者减去一个单位，右者加上一个单位；当二进制位为1时，选出的两个相邻像素点左者加上一个单位，右者减去一个单位；因此在水印的提取时，无需原始图片，只需比较两相邻像素点大小，即可将0/1信息写入文件，还原为文字信息，实现了水印的盲提取。

# 六、致谢

在本次毕设过程中，不论是开题报告，还是外文翻译，亦或是文献综述，还有最后论文的敲定，都离不开老师的耐心指导，同时还要感谢每天陪我一起学习一起进步的同学，是他们在我毕业设计时耐心指导我，并及时指出我的错误，使我能及时的得到反馈并发现问题、解决问题，使我能够顺利的结稿。感激之情无以言表，再次向帮组我的老师和同学致谢。

七、参考文献（References）

[1] 唐林海, 梁丽香, 周长敏. 基于混沌理论的数字图像水印算法研究[J]. 数字化用户, 2013, 19(4):103.

[2] 朱从旭, 陈志刚. 基于DWT域的混沌置乱二值图像数字水印新算法[J]. 小型微型计算机系统, 2005, 26(7):1241-1245.

[3] 王建哲, 姜昱明. 一种基于DCT变换的数字水印技术[J]. 计算机工程与应用, 2002, 38(6):104-105.

[4] 梁敏, 陈万军. 基于离散余弦变换的图像数字水印算法[J]. 西北民族大学学报：自然科学版, 2003, 24(4):41-44.

[5] 李红达, 冯登国. 基于复合离散混沌动力系统的序列密码算法[J]. 软件学报, 2003, 14(5).

[6] 孙燕飞, 王凤英, 周运明, et al. 基于双混沌系统的数字水印技术[J]. 微电子学与计算机, 2005,22(8):114-116.

[7] 王丽娜,郭迟,李鹏.信息隐藏技术实验教程.武汉:武汉大学出版社,2004.

[8] I. Ozturk and R. Kilic, A novel method for producing pseudo random numbers from differential equation-based chaotic systems, Nonlinear Dynamics, 2015, 80: 1147-1157.

[9] Baptista M S . Cryptography with chaos[J]. Physics Letters A, 1998, 240(1-2):50-54.

[10]孙锐, 孙洪, 姚天任. 一种空域的数字图像盲水印方案[J]. 华中科技大学学报（自然科学版）, 2002, 30(2):81-83.

[11]张志宏, 陈凤祥, 王伟. 一种基于奇异值分解的盲水印[J]. 计算机仿真, 2005, 22(8):115-117.

[12]李黎, 张明敏, 潘志庚. 一种抗几何变换的图像盲水印算法[J]. 浙江大学学报(工学版), 2004, 38(2):141-144.

[13]王昕, 孙莹莹, 杜旭. 基于YUV 颜色空间的图像去雾算法[J]. 吉林大学学报(信息科学版), 2016, 33(6).

[14]张雅琴, 杜新辉, 刘宏凯. 基于图像相邻像素差值的隐写分析[J]. 微计算机信息, 2008, 24(9):80-82.

[15]马国峰, 杨俊红, 周兵. 基于YUV颜色空间的视频运动检测[J]. 计算机工程与设计, 2008, 29(14):3700-3702.

[16]高亮, 徐伟, 孟高峰. 一种新的基于混沌序列的空间域盲水印算法[J]. 计算机工程与应用, 2006, 42(1):106-108.

[17]王超, 王伟, 王泉, et al. 一种空间域矢量地图数据盲水印算法[J]. 武汉大学学报·信息科学版, 2009, 34(2): 163-165. WANG Chao，WANG Wei，WANG Quan，QIN Qianqing. A Watermarking Algorithm for Vector Maps in Spatial Domain. GEOMATICS AND INFORMATION SCIENCE OF WUHAN UNIVERS, 2009, 34(2):163-165.

[18]高亮, 孟高峰, 徐伟. 一种基于混沌序列的灰度级盲水印算法[J]. 西南民族大学学报（自然科学版）, 2005, 31(5):823-827.

[19]赵玉霞. 基于混沌系统与提升小波的抗剪切攻击的彩色图像盲水印算法[J]. 工程图学学报, 2010, 31(4):81-86.

[20]王树梅, 赵卫东, 王志成. 一种基于混沌变换的自适应盲水印[J]. 计算机仿真, 2007, 24(11).