

**课 程 实 验 报 告**

**课程名称： 信息隐藏**

**专业班级： 信息安全1602班**

**姓 名： 郭倜维**

**学 号： U201614848**

**指导教师： 凌贺飞**

**报告日期： 2019.06.27**

**计算机科学与技术学院**

目录

[一、任务 1](#_Toc12534033)

[二、算法介绍 1](#_Toc12534034)

[三、具体实现步骤 1](#_Toc12534035)

[3.1 水印嵌入过程 1](#_Toc12534036)

[3.2 水印提取过程 1](#_Toc12534037)

[四、核心源码分析 1](#_Toc12534038)

[4.1水印嵌入 1](#_Toc12534039)

[4.2水印提取 1](#_Toc12534040)

[4.3高斯白噪声攻击 2](#_Toc12534041)

[4.7缩放攻击 2](#_Toc12534042)

[4.8放缩攻击 2](#_Toc12534043)

[4.9旋转攻击 2](#_Toc12534044)

[五、实验测试 2](#_Toc12534045)

[5.1图像信息 2](#_Toc12534046)

[5.1.1载体图像 2](#_Toc12534047)

[5.1.2水印图像 2](#_Toc12534048)

[5.2可选攻击 2](#_Toc12534049)

[5.3测试结果 2](#_Toc12534050)

[六、结论 2](#_Toc12534051)

# 任务

实现LSB水印和量化索引水印算法。

信息载体：每个人自己的一张外景照片；

水印信息：每个人将学号、姓名按上、下两列写在白纸上，然后手机拍摄，转换为黑白图片，作为水印信息；

对信息处理进行鲁棒性测试，以及对水印容量进行分析。

# 算法介绍

## 2.1 LSB算法

LSB 最低有效位法(Least Significant Bit；LSB)是信息隐藏中最基本的方法。它基于人类视觉系统无法觉察细微变化来实现其掩蔽效果；图像像素中的最低位通常表示的是图像的细节信息，很难觉察，正因为如此该位经常成为了信号处理和压缩时被抛弃的部分。而LSB算法就是利用的这点，在实现过程将秘密信息隐藏在图像像素的最低位。该算法具有计算速度快且容易实现的特点。

位平面定义为由所有像素的二进制描述的各个位构成的平面，每个位平面都是二色图像（黑白）。

本次实验已经拥有了原有的图片，所以进行水印检测时可以输入原有图片做匹配处理。

关于添加水印，首先将水印图片扩展到和原图相同的尺寸，然后将对应像素点上的水印像素值和原图像素值进行加和运算；这个加和运算是在RGB通道实现的，需要注意的是水印图是黑白二值bmp图片，其所有像素点为(0,0,0)或(255,255,255)；最后形成加过水印图像的像素点；流程图如下图1.

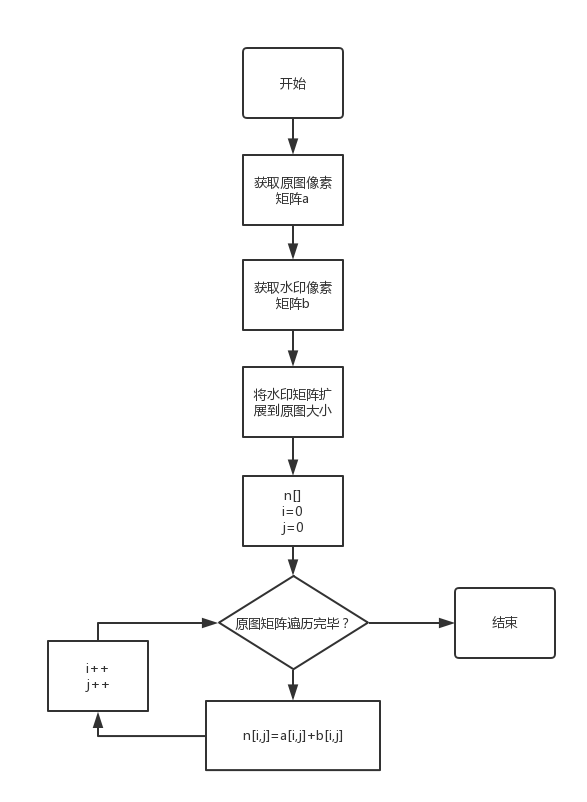


图1 LSB算法流程图

关于提取水印，首先将加水印的图片的像素值和原图像素值相减后即可提取水印。

## 2.2 量化索引算法

量化索引调制是根据要嵌入的水印信息，利用不同的量化器对载体信息进行量化，嵌入前后的图像信息在不同的区间上，量化后的信息就是带有水印的信息。扩频水印是将水印信息直接附加性的嵌入到载体信息中打到嵌入的效果。这样做势必影响水印的容量。而该方法通过调制修改的方法嵌入水印，有较大的嵌入容量。

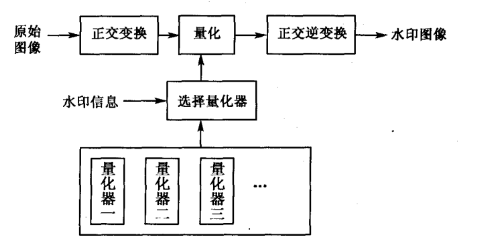


图2 图像水印嵌入模型

经过频域DCT变换后的原始图像可以用最大的直流系数DC和低频交流洗漱AC分量来表示，系数大多数都集中在低频分量区域，高频系数之相对要小得多，所以可以通过适当的量化，进队部分低频系数进行编码，实现图像的压缩编码。

抖动调制，首先将水印信息或由水印信息确定的抖动量添加到载体上，然后对其进行量化从而嵌入水印信息；水印提取时一般采用最小距离解码的方法。

下面分别介绍水印嵌入和水印提取的基本原理。

水印嵌入的原理可用下式描述：



其中参数如下：x表示原始数据；x\_w表示嵌入水印之后的数据；∆表示量化步长；m表示水印信息;d(m)是与m相对应的抖动量;Q\_∆^m(∙)是与m相对应的量化器。

当采用最小距离解码器来提取水印信息时，如果接收到的数据距离量化器0得到的数据最近，解码器认为嵌入的水印信息为0；否则，解码器认为嵌入的水印信息为1，即：



其中参数表如下y表示解码器接收到的数据； 是提取的水印信息;y[0]和y[1]分别表示采用量化器0和量化器1对接收到的数据y进行抖动调制所得到的量化值。

其具体提取过程如图3所示。

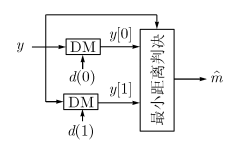


图3 抖动调制提取水印算法框架

由此可见，量化步长∆是影响抖动调制水印系统性能的关键因素之一，但非自适应抖动调制的量化步长与载体无关。

## 2.3 攻击

本次实验所用的攻击方式有加大噪声、滤波、压缩、光照变化强度、旋转等。其中一部分采用Python2.7实现，一部分采用MATLAB实现。

加大噪声：即在图中随即加入黑点。

滤波：即在尽量保留图像细节特征的条件下对目标图像的噪声进行抑制。

压缩：将图片缩小后再放大会原来大小，使得部分像素值信息损失。

光照变化强度：调整亮度。

旋转：几何变换。

# 三、具体实现步骤

## 3.1 水印嵌入过程

### 3.1.1 LSB算法-水印嵌入

1）将水印图片扩展到和原图相同的尺寸。

2）将对应像素点上的水印像素值和原图像素值做加和，形成加过水印图像的像素点。

### 3.1.2 量化索引算法-水印嵌入

1）获取二维水印序列

水印序列和水印图的性质有关，如果水印图是彩色的，水印序列则由W[height][width][0]产生；如果水印图是灰度的，则直接由W[height][width]产生水印序列；如果没有此输入项，则会生成一个随机序列来作为水印序列。再将水印序列二值化，即为0或1。

2）获取二维载体图像序列

选择的是彩色图像，则需要先将其转化为灰度图像。

3）水印嵌入

先根据JPEG量化表并乘以系数0.1得到量化步长∆以及抖动量d（+-delta/4）；然后将载体信息划分为8\*8的小块并进行DCT变换，最后根据公式将水印信息嵌入到每一块的载体信息上。这个过程需要分块进行，最后需要进行块上的DCT逆变换才能得到目标图像。

## 3.2 水印提取过程

### 3.1.1 LSB算法-水印提取

将加水印的图片的像素值和原图像素值相减后即可提取水印。

### 3.1.1 LSB算法-水印提取

1）获取二维待检测图像序列；

2）将载体信息划分为8\*8的块进行DCT变换，根据公式设置两个量化器y0, y1,分别将0,1嵌入到每一块中的8\*8载体信息上；

3）根据公式进行比较，选取数据更接近的那个值作为提取的水印信息；

4）写入0/1矩阵中。

# 四、核心源码分析

## 4.1 LSB算法

1）水印嵌入

%读取原始图像

carner=imread('gray1.bmp');

%显示原始图像

subplot(3,3,1),imshow(carner),title('原始载体图像')

%最低位置为0

carner1=bitand(carner,254);

%读取水印图像

secret=imread('gray3.bmp');

%显示水印图像

subplot(3,3,2),imshow(secret),title('原始水印图像')

%水印图像格式转换

secret=uint8(secret);

%获取水印图像大小

[x,y,z]=size(secret);

%水印图像与载体图像整合

result=carner1;

result(1:x,1:y,1:z)=result(1:x,1:y,1:z)+secret;

%计算并打印PSNR值

psnr(result,carner)

%显示加入水印的图像

subplot (3,3,3),imshow (result),title('隐藏后的载体图像')

%保存结果

imwrite (result, 'result.bmp');

2）水印提取

%读取加入水印的图像

hide=imread('result.bmp');

%显示加入水印的图像

subplot (3,3,4),imshow (hide),title('加入水印图像')

%获取最低有效位

resulth=bitand(hide,1);

resulth=resulth(:,:,1);

%格式转换

resulth=logical(resulth);

%显示并保存图像

subplot (3,3,5),imshow (resulth),title('水印图像')

imwrite(resulth,'water.bmp');

%读取原始水印图像，并显示

secret2=imread('gray3.bmp');

subplot(3,3,6),imshow(secret2),title('原始水印图像')

%获取水印图像大小

[xx,yy,zz]=size(secret2);

%裁剪水印图像

resulth2=resulth(1:xx,1:yy,1);

%格式转换

resulth2=logical(resulth2);

%显示并保存最终水印

subplot(3,3,7),imshow(resulth2),title('提取大小水印图像')

imwrite(resulth2,'water2.bmp');

## 4.2 量化索引算法

1）水印嵌入

jpegStart = [16,11,10,16,24,40,50,61;

12,12,14,19,26,58,60,55;

14,13,16,24,40,57,69,56;

14,17,22,29,51,87,80,62;

18,22,37,56,68,109,103,77;

24,35,55,64,91,104,113,92;

49,64,78,87,103,121,120,101;

72,92,95,98,112,100,103,99];

G = 0.5;%G

jpegD = round(jpegStart \* G);

%插入水印

picture =imread('IMG\_20190522\_112430.jpg');

picture0 =picture;

subplot(2,5,1),imshow(picture),title('原始的载体图像')

pic=picture(:,:,1);

subplot(2,5,2),imshow(pic),title('原始的载体的r')

secret = double(imread('gray3.bmp'));

subplot(2,5,3),imshow(secret),title('原始的水印图像')

[p\_x,p\_y,p\_z]=size(pic);

[s\_x,s\_y,s\_z]=size(secret);

p\_x8=floor(p\_x/8);

p\_y8=floor(p\_y/8);

blockCell = cell(p\_x8,p\_y8);

ChangedMatrix = pic;

%将图像进行分块

for i=1:p\_x8

for j=1:p\_y8

blockCell{i,j}=pic(((i-1)\*8+1):(8\*(i-1)+8),(8\*(j-1)+1):(8\*(j-1)+8));

end

end

%对分块后的图像进行DCT变换

for i=1:p\_x8

for j=1:p\_y8

blockCell{i,j}=dct2(blockCell{i,j});

end

end

% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%将水印图像嵌入到原始图像中

x = 1;

for i=1:s\_x

for j=1:s\_y

tempx = mod(x,64);

x = mod(x+1,64);

d0=jpegD(mod(round(tempx/8),8)+1,mod(tempx,8)+1)/4;

step =jpegD(mod(round(tempx/8),8)+1,mod(tempx,8)+1);

d1=d0+step/2;

if secret(i,j)==0

blockCell{i,j}(4,4)=blockCell{i,j}(5,5)-(step\*round((blockCell{i,j}(5,5)-blockCell{i,j}(4,4)+d0)/step)-d0);

else

blockCell{i,j}(4,4)=blockCell{i,j}(5,5)-(step\*round((blockCell{i,j}(5,5)-blockCell{i,j}(4,4)+d1)/step)-d1);

end

end

end

%执行逆ＤＣＴ逆变换，并将更改后的图像数据存储到矩阵ChangedMatrix中

for i=1:p\_x8

for j=1:p\_y8

blockCell{i,j}=idct2(blockCell{i,j});

ChangedMatrix(((i-1)\*8+1):(8\*(i-1)+8),(8\*(j-1)+1):(8\*(j-1)+8))=blockCell{i,j};

end

end

subplot(2,5,4),imshow(ChangedMatrix),title('加入水印的载体的r')

picture(:,:,1)=ChangedMatrix;

psnr(picture,picture0)

subplot(2,5,5),imshow(picture),title('加入水印载体')

imwrite (picture, 'result.bmp');

2）水印提取

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%提取水印

picture = imread('result.bmp');

subplot(2,5,6),imshow(picture),title('插入水印的图像')

[p\_x,p\_y,p\_z]=size(picture);

p\_x8=floor(p\_x/8);

p\_y8=floor(p\_y/8);

BlockCell=cell(p\_x8,p\_y8);

ChangedMatrix = picture(:,:,1);

subplot(2,5,7),imshow(ChangedMatrix),title('加入水印的载体的r')

secret = double(imread('gray2.bmp'));

[s\_x,s\_y,s\_z]=size(secret);

WMsize=[s\_x,s\_y];

WaterMatrix=zeros(WMsize);

% 抖动调制算法，QIM的提取水印算法

% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%将图像进行分块，每块为8\*8的子块

for i=1:p\_x8

for j=1:p\_y8

BlockCell{i,j}=ChangedMatrix(((i-1)\*8+1):(8\*(i-1)+8),(8\*(j-1)+1):(8\*(j-1)+8));

end

end

%对分块后的图像进行DCT变换

for i=1:p\_x8

for j=1:p\_y8

BlockCell{i,j}=dct2(BlockCell{i,j});

end

end

% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%将水印图像从更改后的图像中提取出来

thisx=1;

for i=1:s\_x

for j=1:s\_y

tempx = mod(thisx,64);

thisx = mod(thisx+1,64);

d0=jpegD(mod(round(tempx/8),8)+1,mod(tempx,8)+1)/4;

step =jpegD(mod(round(tempx/8),8)+1,mod(tempx,8)+1);

d1=d0+step/2;

x=(BlockCell{i,j}(5,5)-BlockCell{i,j}(4,4))-(step\*round((BlockCell{i,j}(5,5)-BlockCell{i,j}(4,4)+d0)/step)-d0);

y=(BlockCell{i,j}(5,5)-BlockCell{i,j}(4,4))-(step\*round((BlockCell{i,j}(5,5)-BlockCell{i,j}(4,4)+d1)/step)-d1);

if abs(x)<abs(y)

WaterMatrix(i,j)=0;

else

WaterMatrix(i,j)=1;

end

end

end

resultWM = logical(WaterMatrix);

subplot (2,5,8),imshow (resultWM),title('水印图像')

## 4.3 LSB鲁棒性测试

%读取加入水印的图像

hide=imread('result.bmp');

%显示加入水印的图像

subplot (2,2,1),imshow (hide),title('加入水印图像')

%读取原始水印图像，并显示

secret2=imread('gray3.bmp');

subplot(2,2,2),imshow(secret2),title('原始水印图像')

%获取水印图像大小

[xx,yy,zz]=size(secret2);

%裁剪水印图像

hide = imnoise(hide,'salt & pepper',0.1);%添加噪声

subplot(2,2,3),imshow(hide),title('添加噪声后')

%获取最低有效位

resulth=bitand(hide,1);

resulth=resulth(:,:,1);

%格式转换

resulth=logical(resulth);

resulth2=resulth(1:xx,1:yy,1);

%格式转换

resulth2=logical(resulth2);

%显示并保存最终水印

subplot(2,2,4),imshow(resulth2),title('提取大小水印图像')

imwrite(resulth2,'water2.bmp');

# 五、实验测试

## 5.1实验环境

笔记本系统：macOS Mojave

虚拟机：Parallel Desktop 14

实验系统：Windows 10

软件：Matlab 2017

## 5.2 LSB嵌入和提取水印

1）嵌入水印

原始载体图像、原始水印图像和加入水印后的载体图像如下图4、5和6所示，同时输出PSNR值为51.2005。



图4 原始载体图像



图5 原始水印图像



图6 隐藏水印后的图像

2）提取水印

从水印载体图中提取出来的水印图片如7所示。



图7 提取的水印图像

## 5.3 量化索引嵌入和提取水印

## 1）嵌入水印

原始载体图像、原始水印图像和加入水印后的载体图像如下图8、9和10所示，同时输出PSNR值为63.4004。

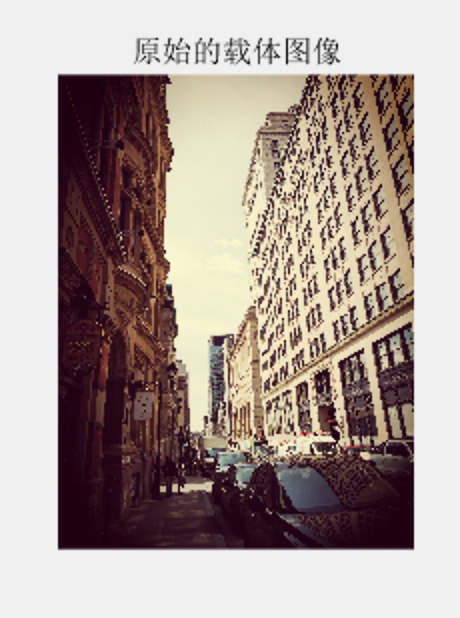


图8 原始载体图像



图9 原始水印图像

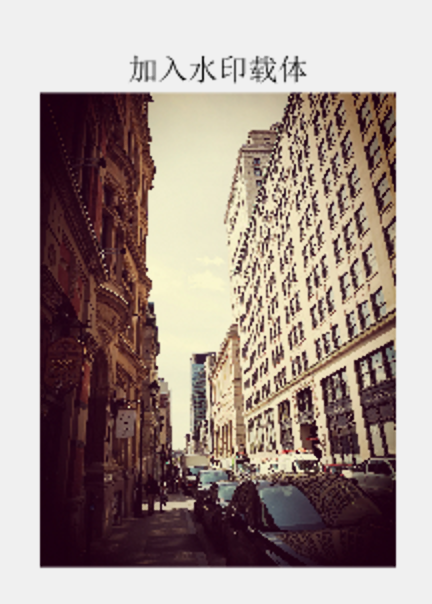


图10 加入水印后的图片

2）提取水印

对加入水印后的图片进行水印提取，结果如图11。



图11 提取的水印图像

## 5.4 LSB鲁棒性测试

添加噪声后使用LSB算法提取出的水印图片如图12.



图12 添加噪声后提取水印结果

# 六、结论

这次实验是我第一次接触matlab这个软件，但是这次实验没有用到matlab，不过使用起来还是比较顺手的。

第一个实验是使用LSB算法给图片添加水印，实现起来不是很困难，主要是使用了很多matlab自带的函数，包括显示图片、添加噪声和获取图片信息等，这个过程依靠网络查询也解决了。

第二次实验相比较第一次实验来说难度有了一定的提高，在了解了理论知识的基础上还需要掌握它并进行应用；所以也遇到了不少问题，不过基本上解决了。比如在实验的时候发现在亮度分量上每一位上直接嵌入水印会有比较大的影响，所以量化调制嵌入的时候将原来的图转化为灰度图后再进行嵌入操作，这样会使得嵌入前后的图像差别较小。

总的来说这次实验让我了解并实现了对图片进行添加水印和提取的操作，这个作为我们进入信息隐藏领域的敲门砖，可以帮助我进一步了解信息隐藏；当然，这次实验还是有很多不足，自己的技术还有待提高，所以希望以后可以学到更多这个方面的知识。