

多媒体数据安全实验一

图像基本隐写与分析方法

实验课总体要求

◆ 考勤要求:

- 1.每次实验课需检查相关完成度
- 2.有科研等任务提前离开的同学需请假说明，下次实验课检查相关任务

实验课总体要求

◆ 关于抄袭(态度问题)

1.实验报告和别人完全一样

2.实验结论和别人一字不差

3.实验图片和数据和别人一样

后果：抄袭者和被抄袭者报告均被打回重写，并且直接给一个均分

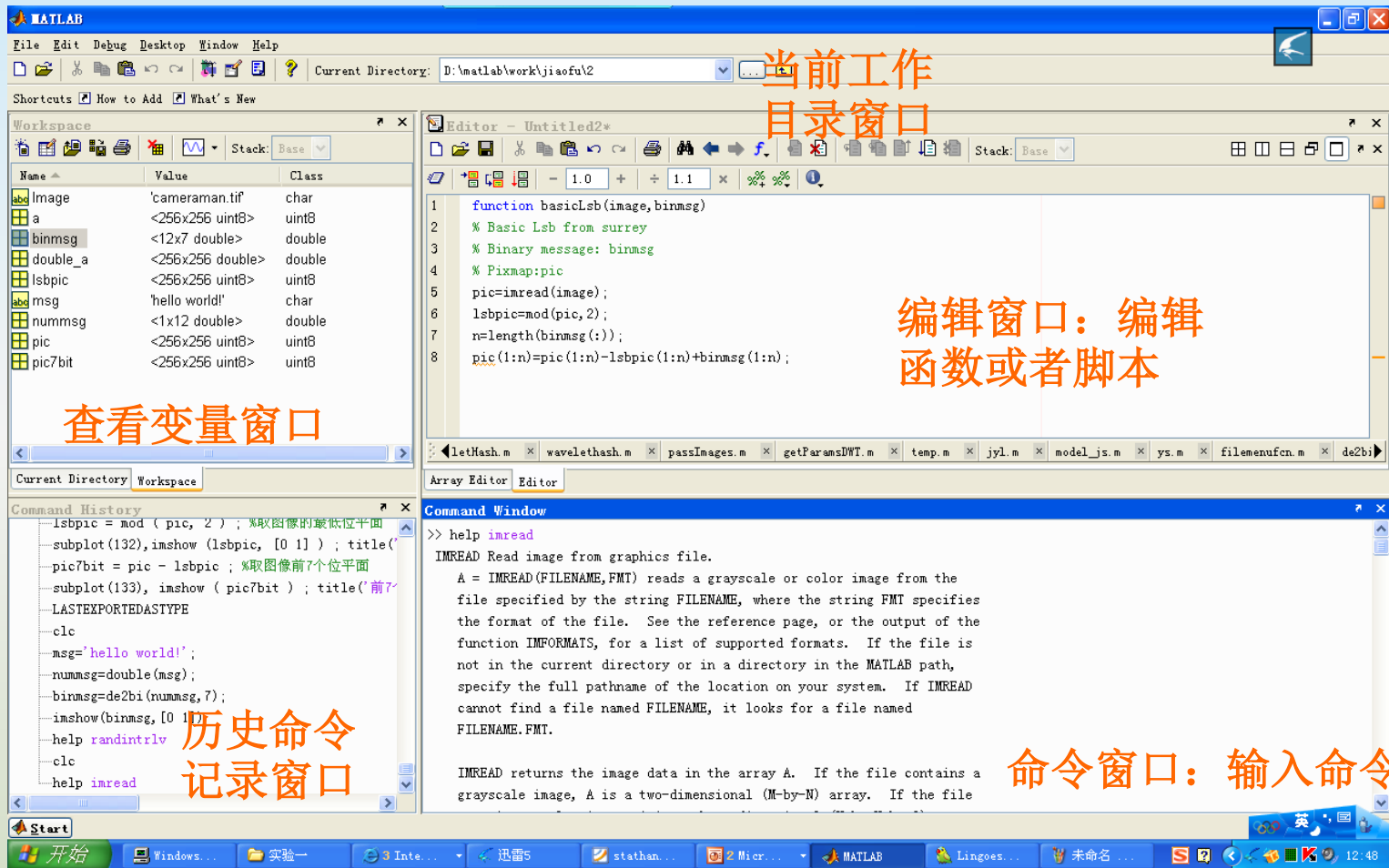
要求：代码可以参考别人的，但要跑自己的数据

实验1 内容

图像基本隐写与分析方法

- ◆ 一、Matlab基本操作（提前自学）
- ◆ 二、LSB替换隐写
- ◆ 三、卡方隐写分析

一、MATLAB基本操作



一、MATLAB基本操作

◆ MATLAB 变量名:

变量名以字母打头，后面最多跟19个字母或数字，如x, y, ae3和d3er45等都是合法的，不能使用内部函数或命令名作为变量名。Matlab中的变量名区分大小写，ab与AB表示两个不同的变量。

Matlab中可以直接使用变量而不必事先声明。

◆ 一些常用命令:

dir: 列出当前目录下的所有文件

clc: 清除命令窗口

clear all: 清除环境变量

help: 帮助查询命令，help ** (**代表要查询的命令名)

一、MATLAB基本操作

◆ 熟悉MATLAB的矩阵操作

常用的矩阵函数

函数名称	函数功能
<code>zeros(m,n)</code>	m行n列零矩阵
<code>eye(n)</code>	n阶方阵
<code>ones(m,n)</code>	m行n列全1矩阵

具体实例： `>> X=zeros(m,n); %m,n 为矩阵行列`

一、MATLAB基本操作

◇ 矩阵取值：

索引： $Z(1,2)$

序号： $Z(4)$

冒号表达式： $1:10$ $1:2:10$ $Z(1:2,3:4)$ $Z(:,end-1:end)$

逻辑，关系运算：

```
>> a= rand(3,4)
```

```
>> I= a>0.5 ;
```

```
>> b=a(I)
```


一、MATLAB基本操作

◆ 熟悉随机数操作

rand: 均匀分布随机矩阵

rand %无变量输入时只产生一个随机数

y=rand(n) %生成n*n随机矩阵，其元素在(0,1)内

y=rand(m,n) %生成m*n的随机矩阵

randn: 正态分布随机矩阵

randint: 整数随机分布矩阵

randint(m,n,rg)%第三个参数可缺省,生成一个m行n列的元素，元素值在[0,rg-1]之间

一、MATLAB基本操作

◆ 熟悉随机数操作

具体实例：

```
>> rand('seed', num); %rng
```

```
>>rand('seed',1032);%设置一个种子， 设置后下面的随机数是一定的
```

```
>>rand(n,m); %生成m*n随机矩阵， 其元素在(0,1)内
```

```
>>rand(5,4);
```

```
>>randint(n,m,rg); %元素在[0,rg-1]内
```

```
>>randint(5,4,4);
```

一、MATLAB基本操作

◆ 熟悉一般文件的导入操作

具体实例：

```
>> fid=fopen(filepath,'r');
```

```
>> fid=fopen('C:\1.txt','r');
```

```
r , w , a , r+ .....
```

```
>>[msg,msg_len]=fread (fid,len,'ubit1');
```

```
help fopen ;help fread
```

一、MATLAB基本操作

◆ 熟悉图像的导入操作

`imread(filename,fmt);` %读取图片数据，参数`fmt`指示图片格式，如‘`bmp`’，‘`jpg`’等，缺省的话系统自动识别图像文件格式。

具体实例：

```
>>data1=imread('c:\woman.bmp');
```

```
>>data2=imread('.\Lenna.jpg');
```

一、MATLAB基本操作

◆ 熟悉图像RGB模型下的色彩操作

具体实例：RGB图像分层

```
>> imageR=data2(:,:,1);%提取红色
```

具体实例：RGB图像合并

```
>>%对R层进行处理
```

```
>> Mix=cat(3,imageR,imageG,imageB);%色彩合成
```

一、MATLAB基本操作

◆ 熟悉显示图像实验结果操作

`subplot(mnp)`:前两个参数表示将画面分为 $m*n$ 个子图像,
`p`表示子图像序号,排序顺序为从左至右,从上至下。

`imshow(x)`:显示图像函数

具体实例:

```
>>subplot(121),imshow(data1);title('woman'),  
    subplot(122),imshow(data2);title('lena');
```

一、MATLAB基本操作

◆ 熟悉显示图像实验结果操作

`subplot(mnp)`:前两个参数表示将画面分为 $m*n$ 个子图像，`p`表示子图像序号，排序顺序为从左至右，从上至下。

`imshow(x)`:显示图像函数

具体实例：读取图像库任意一幅图像，转换为灰度图像和二值图像显示。

转换为灰度图像：`X=rgb2gray(image)`

转换为二值图像：`img_b=im2bw(img,0.3)`，0.3为阈值，阈值不同转换的图像也不尽相同。

一、MATLAB基本操作

◆ 熟悉取图像大小的操作

具体实例： `>>[row,col]=size(data1);`

◆ 熟悉MATLAB基本绘图操作

`plot(x,y['属性值'])`以x、 y为轴绘制曲线,可以设置绘制属性

在同一幅图上绘制多幅图像：`plot(x1,y1,x2,y2,.....)`

或者使用hold on即：`plot(x1,y1);`

`hold on;`

`plot(x2,y2);`

`hold off;`

一、MATLAB基本操作

◆ 熟悉MATLAB的信号变换操作

离散傅里叶(DFT)、离散余弦(DCT)、离散小波变换(DWT)是图像、音频信号常用基础操作，时域信号转换到不同的变换域后会导致不同程度的能量集中。

其中正变换为`dct2(data1)`；逆变换为`idct2(data1)`

一、MATLAB基本操作

◆ 熟悉MATLAB的图像分块操作

$B = \text{blkproc}(A, [m, n], \text{fun}, P1, P2, \dots)$

其中A为原始信号矩阵，[m,n]为分块大小，fun为对每一分块x的操作规则，Pi是fun中调用的参数。

具体实例： $\gg T = \text{dctmtx}(8);$

$\gg \text{result} = \text{blkproc}(\text{data1}, [8\ 8], 'P1 * x * P2', T, T');$

一、MATLAB基本操作

◆ 熟悉MATLAB脚本编程的基本语法

一般形式为： <M文件名>.m

运行方式：在命令窗口直接输入文件名，按enter结束；
在编辑窗口打开菜单debug，单击run

例：绘制图像lena.bmp

```
clc;
```

```
clear;
```

```
[x,map]=imread(strcat(pn,fn),'bmp');%x代表像素值，  
map代表色谱
```

```
imshow(x);
```

一、MATLAB基本操作

◆ 熟悉MATLAB的函数编写

一般形式：function<因变量>=<函数名>(<自变量>)

例：读入并绘制图像

```
function y=a(resource)
```

```
x=imread(resource);
```

```
imshow(x);
```

在命令窗口输入a('lena.bmp');

一、MATLAB基本操作

◆ 关于符号：

matlab命令中的符号一定要是英文输入法中的符号。

◆ 关于类型

一般为uint8和double，用imread读入的是uint8类型，进行数值计算时要转化为double。uint8的范围是[0 255],double的范围是[0 1];

```
a=imread('c:\woman.bmp'); %读入图像
```

```
imshow(a); %正常显示图像
```

```
double_a=double(a); %转化为double类型
```

```
imshow(double_a); %显示白屏
```

```
a = uint8(double_a); %转化为uint8类型
```

一、MATLAB基本操作

- ◆ 命令的末尾输入分号可以使命令的输出不显示在命令窗口，用clc命令清理命令窗口，clear命令清理变量。
- ◆ 不懂函数用法时用 help命令获得相关说明文档。help ***
(***代表函数名)

二、LSB替换隐写

◆ LSB (Least Significant Bites) , 最低有效位。

将图像各个分量的LSB清0, 如下:

```
>> data = imread('lenna.bmp');  
>> data1 = bitand(data,254);%bin2dec  
>> data2 = bitand(data,127);
```

展示原图:

```
>> subplot(2,2,1) imshow(data);
```

展示最低有效位图:

```
>> subplot(2,2,2) imshow(data1);
```

展示最高有效位图:

```
>> subplot(2,2,3) imshow(data2);
```

二、LSB替换隐写

实验1.1

◇ 顺序选取图像载体像素

嵌入：

- 1) 顺序选取图像载体像素的最低有效位，排列起来作为秘密信息载体
- 2) 将秘密信息按位替换到载体中
- 3) 将载体重新写入图片
- 4) 展示原始图像和隐写图像的局部直方图

```
>> count = imhist(data); 得到直方图计数
```

```
>> subplot(2,2,4); stem(0:20,count(1:21));
```

二、LSB替换隐写

◇ 顺序选取图像载体像素

提取：

- 1) 顺序选取图像载体像素的最低有效位，排列起来作为秘密信息载体
- 2) 提取秘密信息载体的前 n 位， n 为秘密信息长度，即得到秘密信息

二、LSB替换隐写

实验1.2

◆ 随机选取图像载体像素

顺序选取像素点导致图像各部分统计特征不一致。为了解决这一问题，可以随机间隔选取像素序列。与顺序嵌入相比，需要设计随机取点的算法。

嵌入：

1) 用伪装密钥k作为伪随机数发生器的种子，生成伪随机序列，嵌入位置为：

$$j_1 = k_1$$

$$j_i = j_{i-1} + k_i \quad i \geq 2$$

>> rng(1,'twister'); 将种子设定为1，生成器为梅森旋转

>> r4 = randi(10,30,1); 创建 1 到 10 之间的随机整数值数组

二、LSB替换隐写

- 2) 利用伪随机序列选取图像载体像素的最低有效位，排列起来作为秘密信息载体
- 3) 将秘密信息按位替换到载体中
- 4) 展示原始图像和隐写图像的局部直方图

提取：

- 1) 用伪装密钥 k 作为伪随机数发生器的种子，生成伪随机序列，
- 2) 根据伪随机序列，画出图像随机序列位置，选取该序列像素的最低有效位排列起来作为秘密信息载体

三、卡方隐写分析

- ◆ 主要针对图像所有像素点的LSB全嵌入情况
- ◆ 利用数理统计假设检验中的卡方检验模型来分析

三、卡方隐写分析

- ◆ 设图像中灰度值为 j 的像素数量为 h_j ，其中 $0 \leq j \leq 255$ 。如果载体图像未经隐写，像素值为 $2i$ 和 $2i+1$ 的像素点数量会相差很大。
- ◆ 秘密信息在嵌入前往往经过加密，可以看作是0、1随机分布的比特流，而且0，1比特概率各为50%。
- ◆ 如果秘密信息完全替代载体图像的最低位，那么像素值为 $2i$ 和 $2i+1$ 的像素点数量会比较接近，也就是“值对”现象，可以根据这个性质判断图像是否经过隐写。

三、卡方隐写分析

实验1.3:

定量分析载体图像最低位完全嵌入秘密信息的情况:

- ◆ 嵌入信息会改变直方图的分布，由差别很大变得近似相等，但是不会改变像素值为 $2i$ 和 $2i+1$ 的像素点总量，因为样值要么不改变，要么就在 $2i$ 和 $2i+1$ 之间改变。令

$$h_{2i}^* = \frac{h_{2i} + h_{2i+1}}{2} \quad q = \frac{h_{2i} - h_{2i+1}}{2}$$

显然 h_{2i}^* 这个值在隐写前后是不会变的，所以可构造

$$r = \sum_{i=1}^k \frac{(h_{2i} - h_{2i}^*)^2}{h_{2i}^*}$$

服从卡方分布， r 为卡方统计量， r 越小表示含有隐密信息的可能性越大。

三、卡方隐写分析

上式中，k等于 h_{2i} 和 h_{2i+1} 所组成数字对的数量， h_{2i}^* 为0的情况不计在内。结合卡方分布的密度计算函数计算载体被隐写的可能性为：

$$p = 1 - \frac{1}{2^{\frac{k-1}{2}} \Gamma(\frac{k-1}{2})} \int_0^r \exp(-\frac{t}{2}) t^{\frac{k-1}{2}-1} dt$$

>> p= 1 - chi2cdf(r, k-1); % r为卡方统计量，k-1为卡方分布自由度

如果p接近于1，则说明载体图像中含有秘密信息。