卡方检测实验报告

李浩然

PB13210007

# 实验目的

* 理解卡方检测的原理
* 实现卡方检测算法，考察其检测能力

# 实验原理

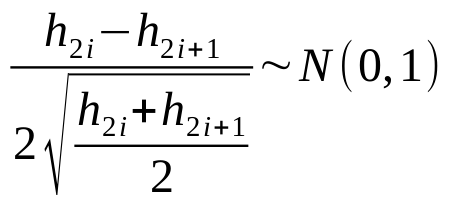
假设：

1. 需要隐写的内容经过加密处理，呈现完全随机的01比特位分布。
2. 图像中的灰度值变化梯度明显。
3. 只改变图像中像素灰度值的最低位。
4. 将从第一个像素开始连续修改图像将其最低位替换成隐写内容的对应比特直到隐写内容结束。

由第三点可知：原先灰度值为h的像素，经过替换以后只会变为h^1或者保持h不变，而同时，灰度值为h^1的像素经过替换后会变成h或保持不变。（^为抑或运算）

再结合第一点假设可以得到结论：经过隐写后灰度值为h和h^1的像素个数会十分接近。

设灰度值为i的像素个数是hi，则以上结论表示h2i与h2i+1之间的差是一个在[-1, 1]上的伯努利分布分布的累加。所以就有：



检验一个分布符合标准正态分布，可以利用卡方检测法：多个正态分布的平方和符合卡方分布。于是可以通过求得上述随机量对于所有满足灰度值0~255限制的i的平方和，从而可以得到改分布符合正态分布的概率。

# 实验环境

* Matlab 2014a
* Windows 10

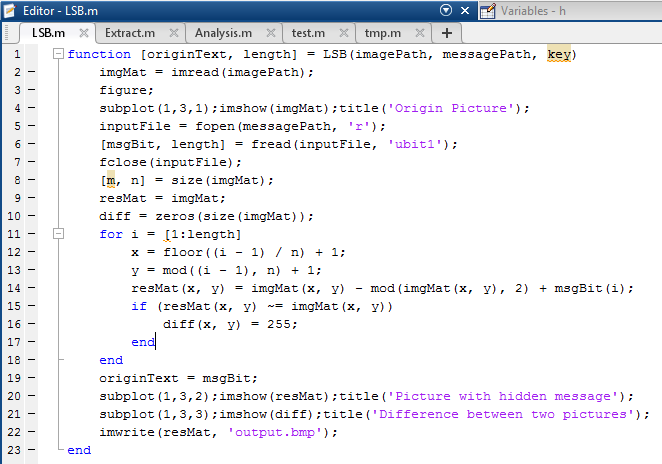
# 实验内容

本实验主要分为三个部分：实现LSB隐写，实现卡方检测。

## 实现LSB隐写

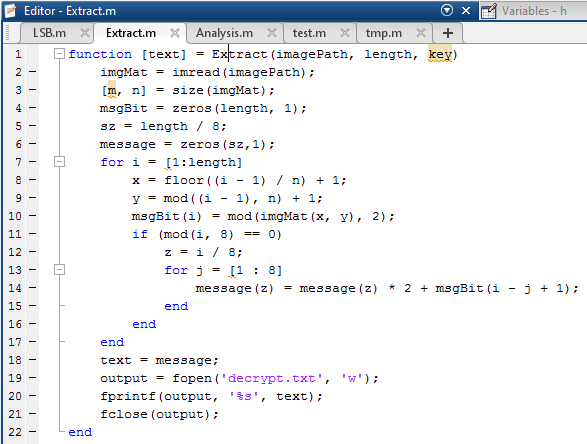
隐写算法流程：

1. 读入bmp格式图片，将其每个像素点的灰度值存入矩阵中(imread)。（第2行）
2. 读入需要隐写的信息，将其转化(dec2bin)为比特流。（第5, 6行）
3. 从(1, 1)像素开始，依次灰度值的最低有效位上写入隐写信息的比特流。（第11到18行）
4. 将载有隐写信息的新矩阵输出成新的图片。（第22行）



提取隐写信息算法流程：

1. 读入载有信息的图片。（第2行）
2. 顺序取出前length个像素的灰度值的最低有效位组成信息比特流。（第7到10行）
3. 将比特流每八位分一组转化为原始信息。（第11到16行）



运行结果：

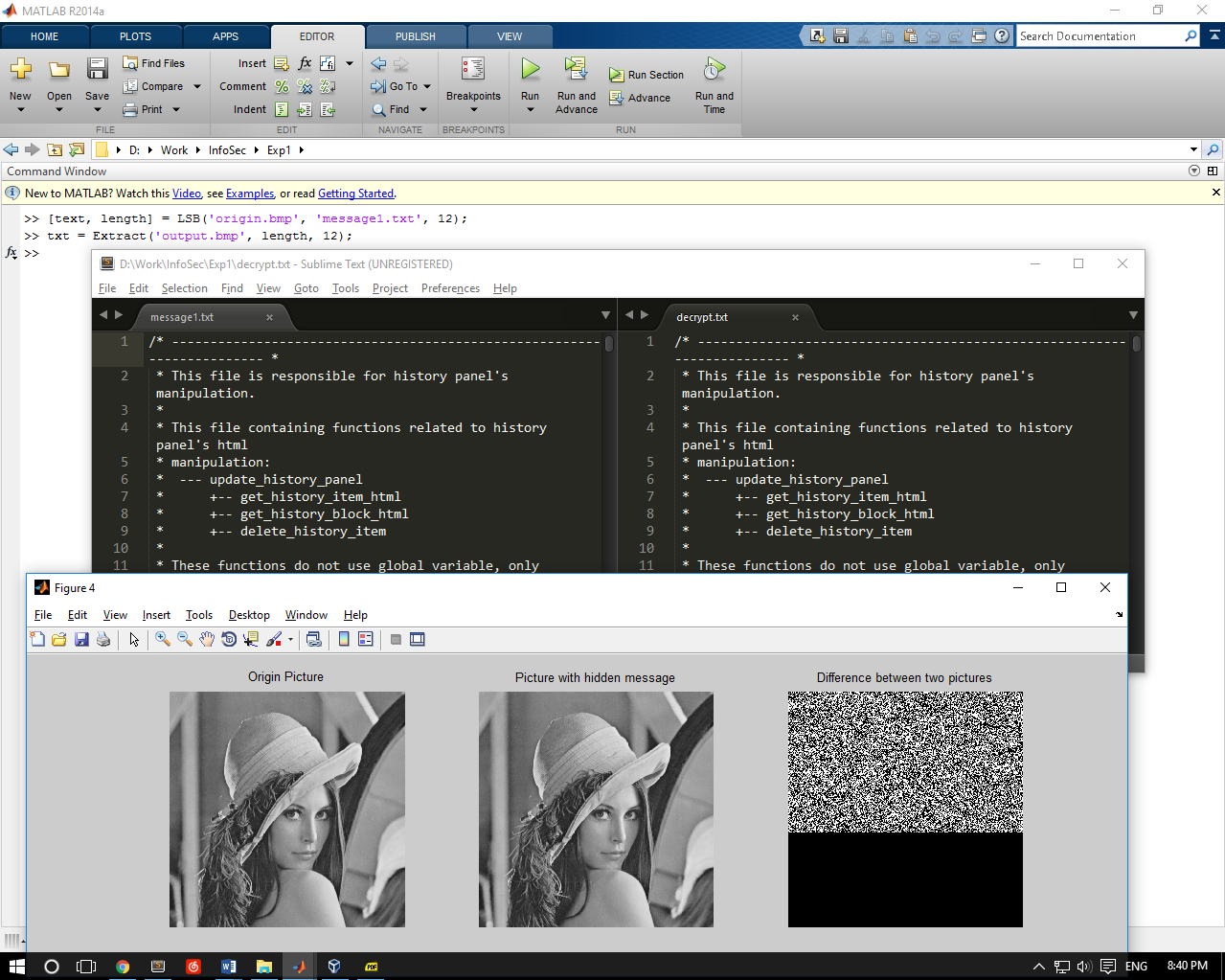
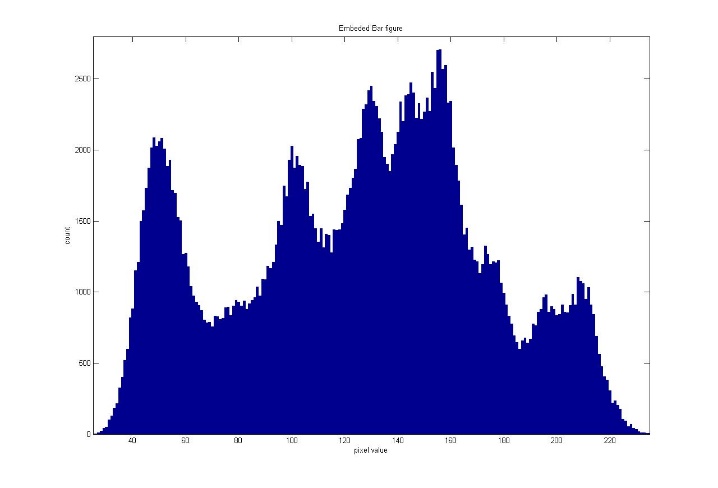
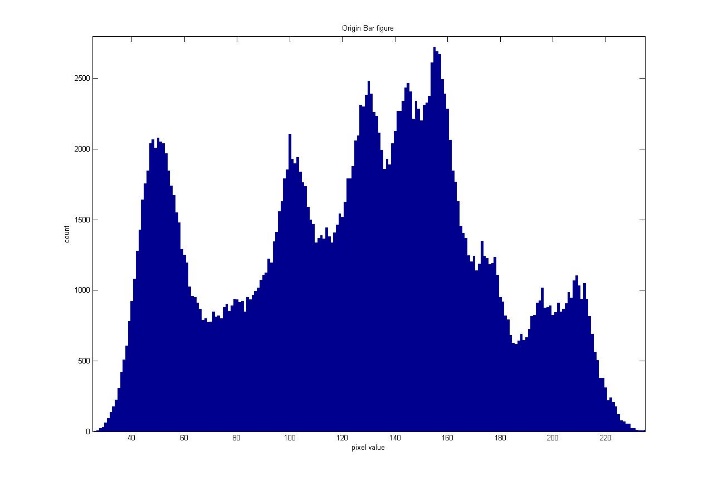


Figure4中将原始和隐写后的图片不同的像素表为白色。嵌入率为60%~70%。

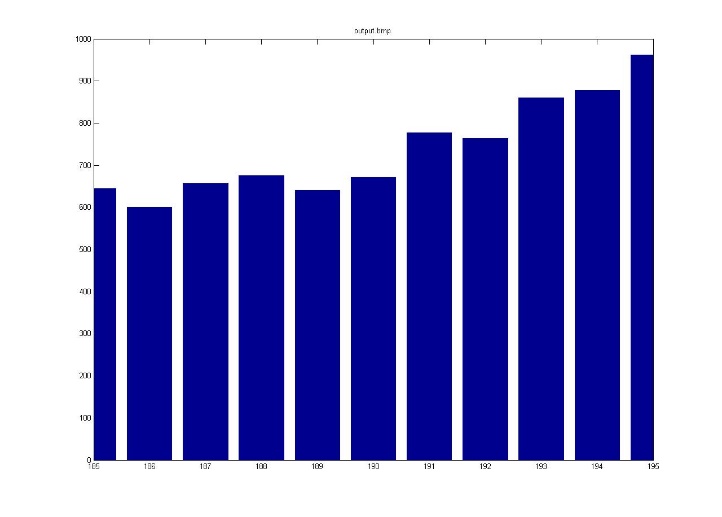
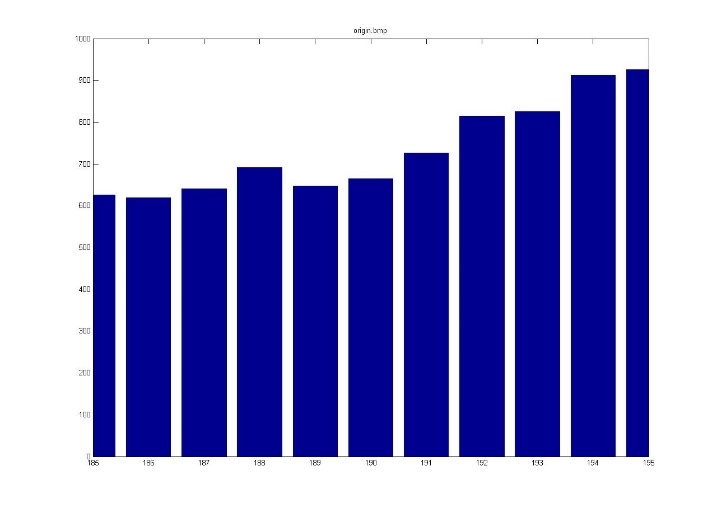
从文本的比较中可以看出，信息被完整恢复。

画出两张图片的像素灰度值柱状图：



左图是嵌入之前的柱状图，右边是嵌入之后的直方图。

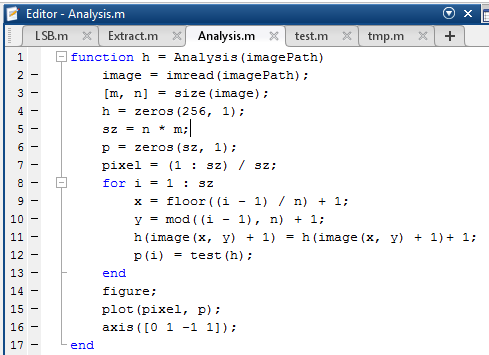
因为图像的灰度值变化比较缓和，所以嵌入的影响不好观测，但如果将灰度值在[185, 195]区间的部分放大查看可得：



可以看出右图中一对灰度值的差更小。

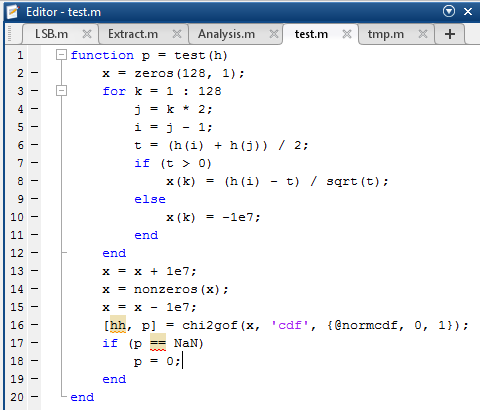
## 实现卡方检测

按嵌入顺序便利图片中的每个像素，统计灰度值的频数。每统计一个像素求一次拟合概率。

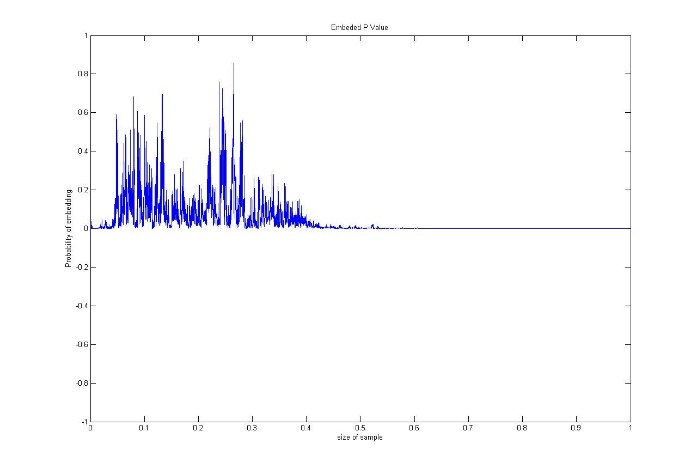
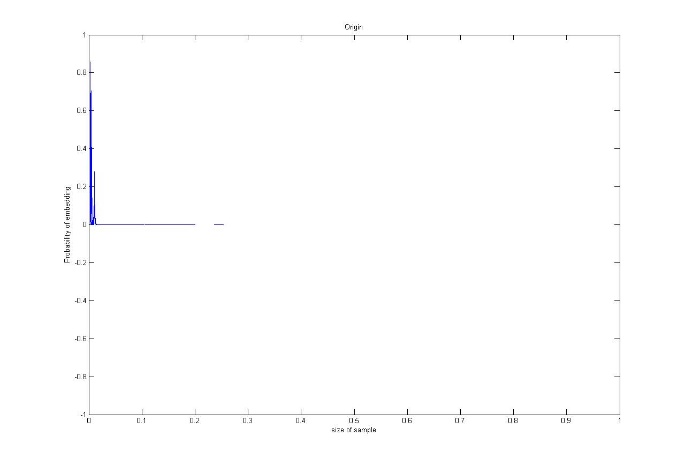


求标准正态分布的拟合度调用自定义的test函数，将频数分成两个一对，求出正态分布变量x，调用Matlab提供的chi2gof接口得到拟合程度。

代码如下：



程序执行结果：



左图是为嵌入时的卡方检测结果，右图是嵌入后的卡方检测结果。可以看出嵌入之前概率趋近于0，潜入以后，在大约60%的地方收敛到零。

# 实验分析

1. 原图像的选择

假设2中提到，图像中，相邻灰度值需要有较大的频数差。这也是卡方检测成立的前提。但实际情况中，图像的灰度值一般都变化得比较平缓。变化剧烈的图像一般包含不同的灰度值较少，卡方检测的可信度也不高。另外，灰度值只有256大小的取值空间，导致卡方检测的样本容量太小，可信度不高。

1. 嵌入信息的选择、

假设1提出，嵌入的信息具有随机性，所以一般是加密过的密文。如果直接嵌入明文，虽然不安全，但是明文中，常用字符用8位表示为ascii码的时候会存在较大的冗余，这样就基本上不能满足随机性，也就无法进行卡方检测。

# 改进设计

1. 在信息中等间隔插入一定数量的0：引入01数量的偏差，这个偏差会体现在标准正态分布沿x轴上的平移，能够抵御仅考虑标准正态分布的卡方检测。  
   缺点：降低了有效信息量。
2. 随机位置嵌入信息：假设随机的位置占全部像素的p部分。那么剩下的 (1-p) 像素的灰度值分布还是符合原先的分布。并且假设随机的位置均匀分布在图像的所有像素中，那么顺序卡方检测无论进行到什么时候，所检察到的分布都是p\*h0+(1-p)\*h1。  
   缺点：降低了信息嵌入效率，传输生成随机位置的密钥需要额外开销。
3. 选择画面变化丰富的图片：这种图片的灰度值分布一般比较均匀，嵌入信息以后发生的改变不明显。  
   缺点：符合上述条件，又适合情境的图片可能不容易获得。