实验二：信息隐藏技术

综合评分：

### 实验目的

隐写分析以及变换域隐写技术

### 实验内容

#### 针对LSB隐写的卡方分析

* 1. 实现针对LSB隐写的卡方分析
  2. 分析实验性能

#### 针对LSB隐写的RS分析

1. 实现针对LSB隐写的RS分析
2. 分析实验性能

#### JPEG压缩算法

1. 分析JPEG压缩算法的主要流程

#### Jsteg隐写算法

1. 实现Jsteg隐写算法
2. 分析实验性能

#### F3隐写算法

1. 实现F3隐写算法
2. 分析实验性能

### 实验工具及平台

■ Windows+Matlab □ 其它：（请注明）

### 实验涉及到的相关算法

1、与实验内容选择的项目对应；

2、请使用流程图、伪代码、NS图或文字方式描述，不要贴代码

#### 卡方隐写分析

卡方隐写分析主要利用了LSB隐写后图像的**值对效应**。它需要LSB隐写满足如下的条件：

1. 嵌入信息中0、1的分布**较为均匀**，即各为50%左右。由于信息嵌入到载体之前通常需要经过加密操作，因此这一点是容易满足的。
2. 图像需要有**较多的像素点**被嵌入信息。当嵌入信息较少时，卡方分析的效果并不精确。

卡方分析的原理是：若设表示图像载体中灰度值为的像素数量，如果载体图像没有使用LSB隐写算法，那么和的值通常相差较大，而LSB隐写方法将秘密信息取代图像的最低位，由于秘密信息通常是加密过的，因此可以看成0、1分布均匀的比特流。在嵌入过程中只存在而不存在的变换，因此使得和的值趋于一致，我们能够借助改变的统计特性判断图像是否经过隐写。

我们首先定义，由LSB隐写算法的性质我们可以知道在**嵌入前后该值是不变的**。

由**中心极限定理**，我们有

因此服从卡方分布。

结合卡方分布的密度计算函数我们可以计算出载体被隐写的可能性为：

当的值接近于1时，我们可以推断出载体图像中含有秘密信息。

#### RS隐写分析

RS隐写分析基于隐写前后**图像平滑度**的变化来检测秘密信息。与卡方隐写分析相比，它具有如下优势：

1. 能够检测**随机LSB替换**隐写。
2. 能够比较精确的估计隐写信息的长度。

对于大多数正常图像而言，其采样点具有**较强的相关性**，而秘密信息由于经过压缩或者加密处理，其**相关性较弱**，因此当使用LSB算法将秘密信息嵌入到载体图像的最低位之后，像素灰度值之间的相关性在一定程度上会受到破坏。RS使用如下的函数来衡量一个图像的平滑度，其中是使用**Zig-zag方式**扫描排成的一个像素向量。

值越小，说明图像块的混乱程度越小，图像块的空间相关性越强。

在RS中定义了在上的如下三个操作函数：

**在LSB中我们只使用了**操作。

RS分析的流程如下：

1. 首先将图像分成若干个像素组。
2. 选取掩码算子
3. 对像素组进行操作：
4. 根据**平滑度函数**比较与的大小，并归类为如下三个组别：
   1. **正则组(Regular)**：
   2. **奇异组(Singular)**：
   3. **不变组(Unusable)**：
5. 计算
   1. 为作用下正则组占所有像素组的比例
   2. 为作用下正则组占所有像素组的比例
   3. 为作用下奇异组占所有像素组的比例
   4. 为作用下奇异组占所有像素组的比例

如果待检测图像没有经过LSB，那么无论用还是，从统计上来说，一般都会**同等程度的增加图像块的混乱程度**，也就是说**近似等于；也近似等于，并且；**。而如果是经过LSB隐写的图像，即相当于对部分像素使用过操作，其结果就会有显著的不同，一般来说会有如下的式子成立：

除了判断载体是否含有秘密信息，RS隐写分析算法还能够估计隐写容量，其原理是利用了**LSB隐写仅用了翻转造成使用或翻转去处理载体图像时，参数呈现不对称性**。

定义如下参数(其中表示嵌入率为时对应的值)

：嵌入率

那么估计值即为下述方程绝对值较小的解：

**这是因为和与嵌入比例成线性关系，和是的二次曲线关系**。

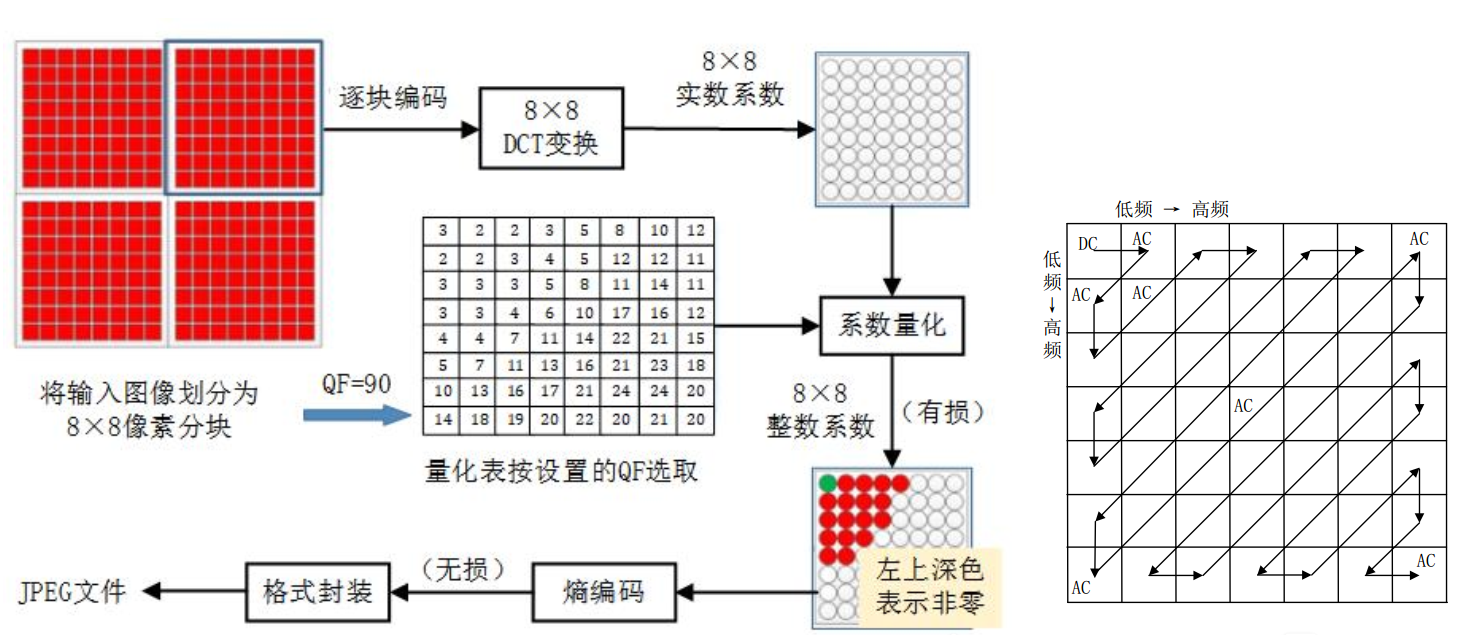
#### JPEG压缩算法

**JPEG(Joint Photographic Experts Group)**是一种针对照片影像而广泛使用的**有损压缩**标准方法

其压缩流程如下：

1. 色彩空间转换：由RGB转换为YUV。
2. 缩减取样：减少UV的成分。
3. 离散余弦变换：将图片中的每个区域(Y、U、V)划分成的子区域，每一个子区域使用DCT转换到频域空间。在频域空间中，左上角的数值称为**DC系数**，其他的63个值称为**AC系数**。
4. 量化：将的子矩阵中的值依次量化矩阵中对应的值并向下取整。该操作是**有损**的。本次实验中要分析的隐写算法针对的均是**量化后的系数矩阵**。
5. 熵编码：使用编码的方式(通常是霍夫曼编码)压缩载体的大小，该步骤是**无损的**。

上述流程可以用下图来表示：



其解码流程即为压缩流程的逆过程。

#### Jsteg隐写算法

Jsteg算法针对的是**量化后的系数矩阵**。其本质和LSB类似，**主要区别在于它不将秘密信息隐藏在系数0、1中**。

Jsteg算法的嵌入步骤如下：

1. 解码JPEG图像，得到量化后的DCT系数，判断该DCT系数**是否等于正负1或0**，若等于则跳过该DCT系数，否则，执行下一步。
2. 判断DCT系数的LSB是否与要嵌入的秘密信息比特相同，若相同，则不对其进行修改，否则执行下一步。
3. 用秘密信息比特替换DCT系数的LSB，将修改后的DCT系数重新编码得到隐秘JPEG图像。

Jsteg算法不使用0以及正负1进行信息的隐藏，其原因主要有以下两点：

* DCT系数中0的比例最大（一般可达到60%以上，取决于图像质量和压缩因子），压缩编码是利用大量出现连零实现的，如果改变DCT系数中“0”的话，不能很好的实现压缩。
* DCT系数中的1若变成0，由于接受端**无法区分未使用的0和嵌入消息后得到的0**，从而无法实现秘密信息的提取。

提取步骤和嵌入步骤类似，只是将第2步变成信息的提取。

#### F3隐写算法

F3隐写算法主要是为了解决Jsteg中0、1系数不嵌入信息的问题。该算法的主要步骤和Jsteg类似，只是在隐写规则上稍有区别：

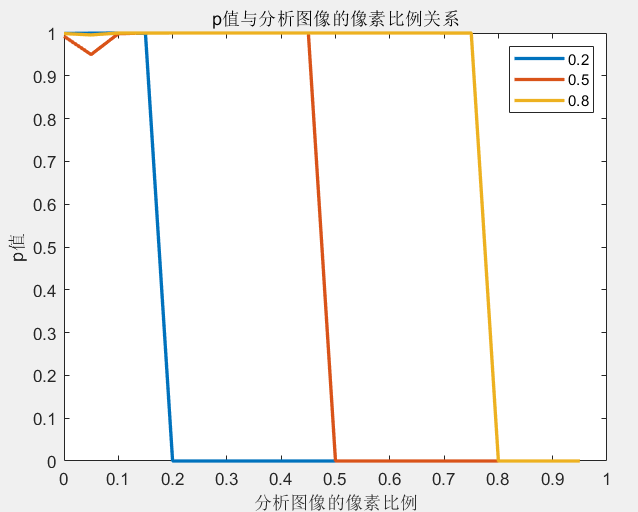
1. 每个非零的DCT系数用于隐藏1bit信息，为0的系数不负载秘密信息。
2. 如果秘密信息与DCT的LSB相同，便不作改动；如果不同，**将DCT系数的绝对值减小1，符号不变**。
3. 当原始值为正负1且要嵌入秘密信息为0时，将这个位置**归0并视为无效**，在下一个DCT系数上重新嵌入。

### 实验分析

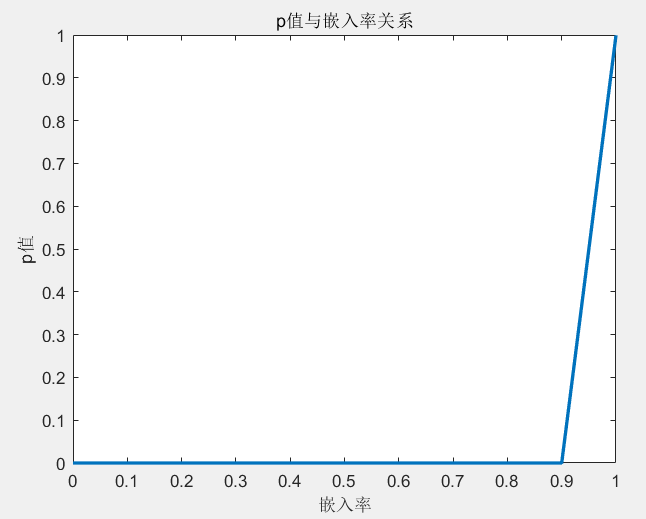
1. 请尽量使用曲线图、表等反映你的实验数据及性能
2. 对照实验数据从理论上解释原因
3. 如无明显必要，请不要大量粘贴实验效果图

#### 卡方隐写分析

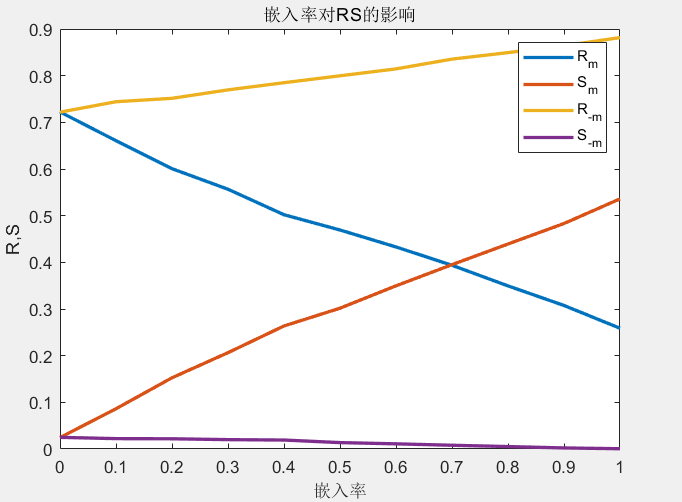
第一张图是当嵌入率为0.2、0.5、0.8时p值与分析的图像像素比例之间的关系，这里采用的是顺序的LSB隐写。可以发现，**当嵌入的比率较少时，如果对整张图片进行分析得到的效果并不好；而如果分析的像素中全部使用了LSB隐写，那么卡方分析能够得到较好的结果**。



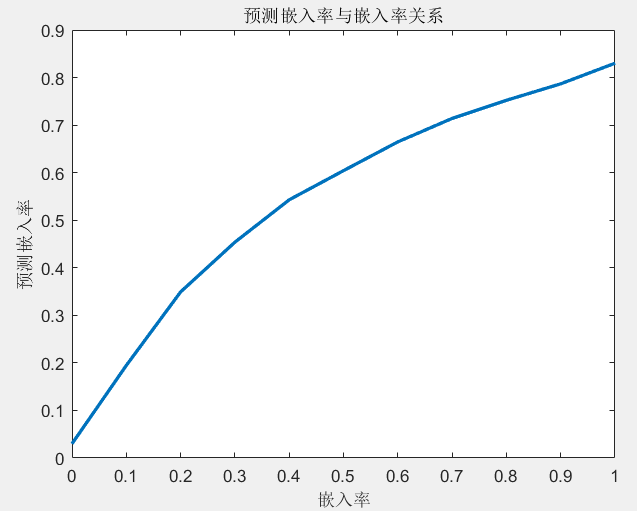
下面的图是对整个图像使用卡方分析时其p值和嵌入率的关系，可以发现只有在嵌入率较高的时候(大于90%)，卡方分析才有较好的结果



#### RS隐写分析

第一张图是嵌入率从0增加到1时对R、S值的影响。可以发现当嵌入率为0时我们有，随着嵌入率的逐渐增加我们可以发现以及的值迅速增加，而和的值逐渐降低，这导致了，并且在**嵌入率较小时，上述变化仍然较大**，这就保证了RS分析即使在嵌入率较小时仍然可以使用，克服了卡方分析的不足。 

下图是预测嵌入率与嵌入率之间的关系，可以看出RS分析预测出的结果与实际的结果相近。



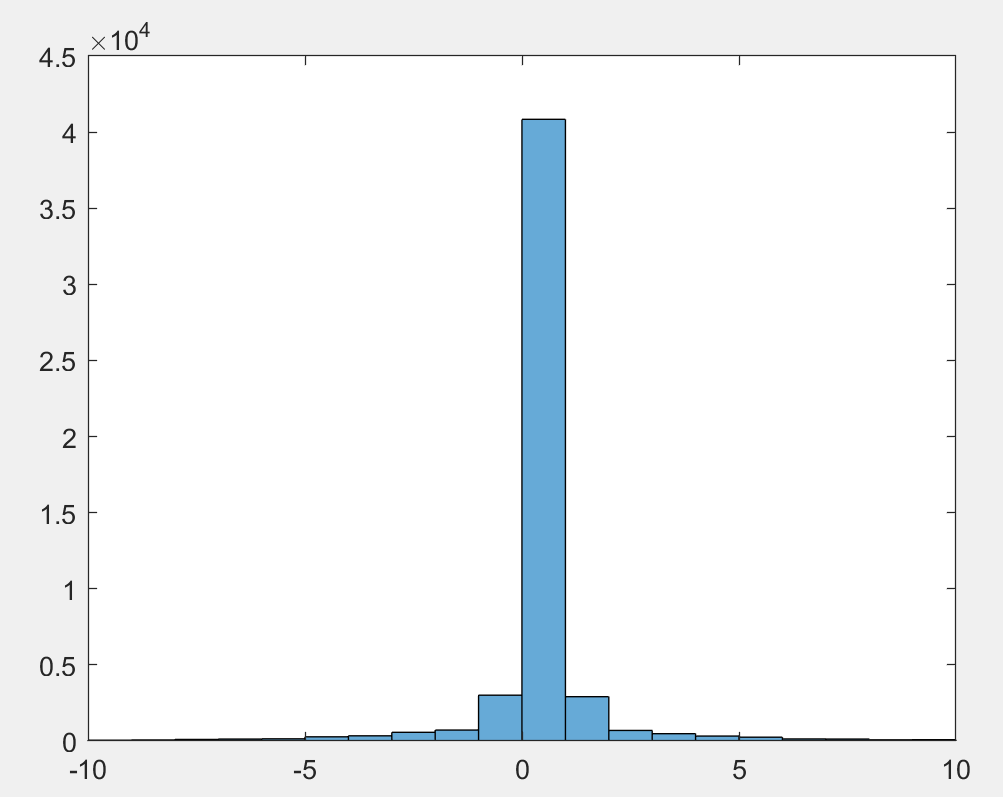
#### Jsteg隐写

Jsteg是在**量化后的DCT系数矩阵**中使用LSB隐写(除0、1外)，下图是原始的载体图像(左)和其量化后的DCT系数矩阵(右)：

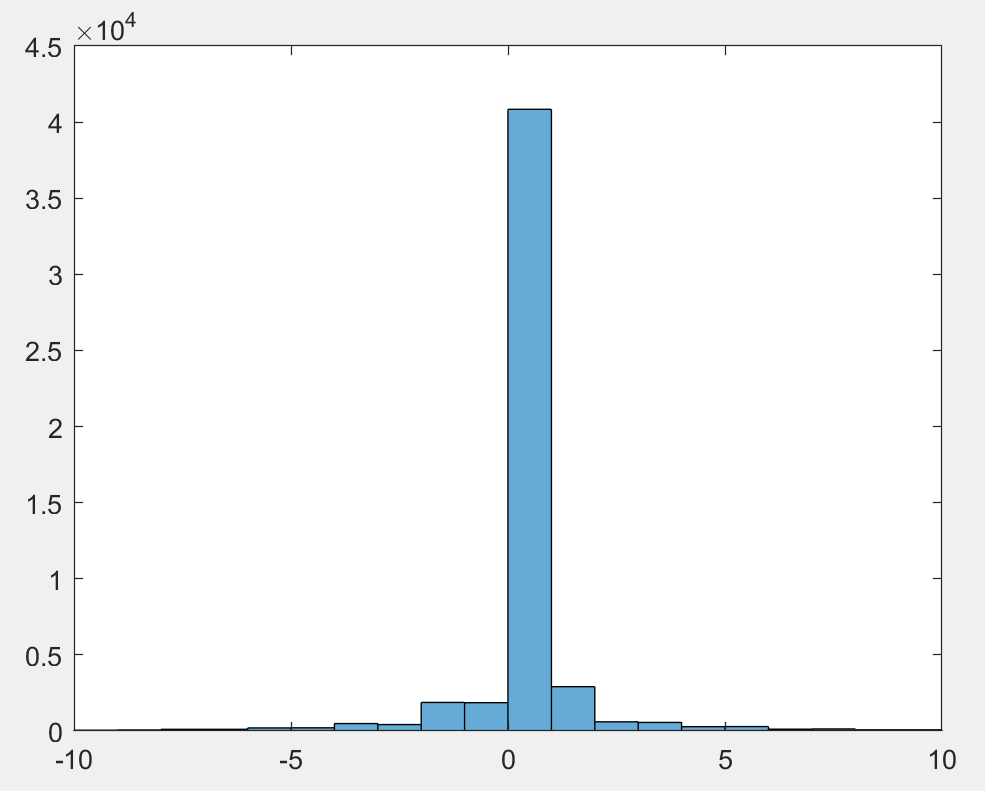


观察下面正常载体量化后的DCT系数直方图，我们可以发现如下特点：

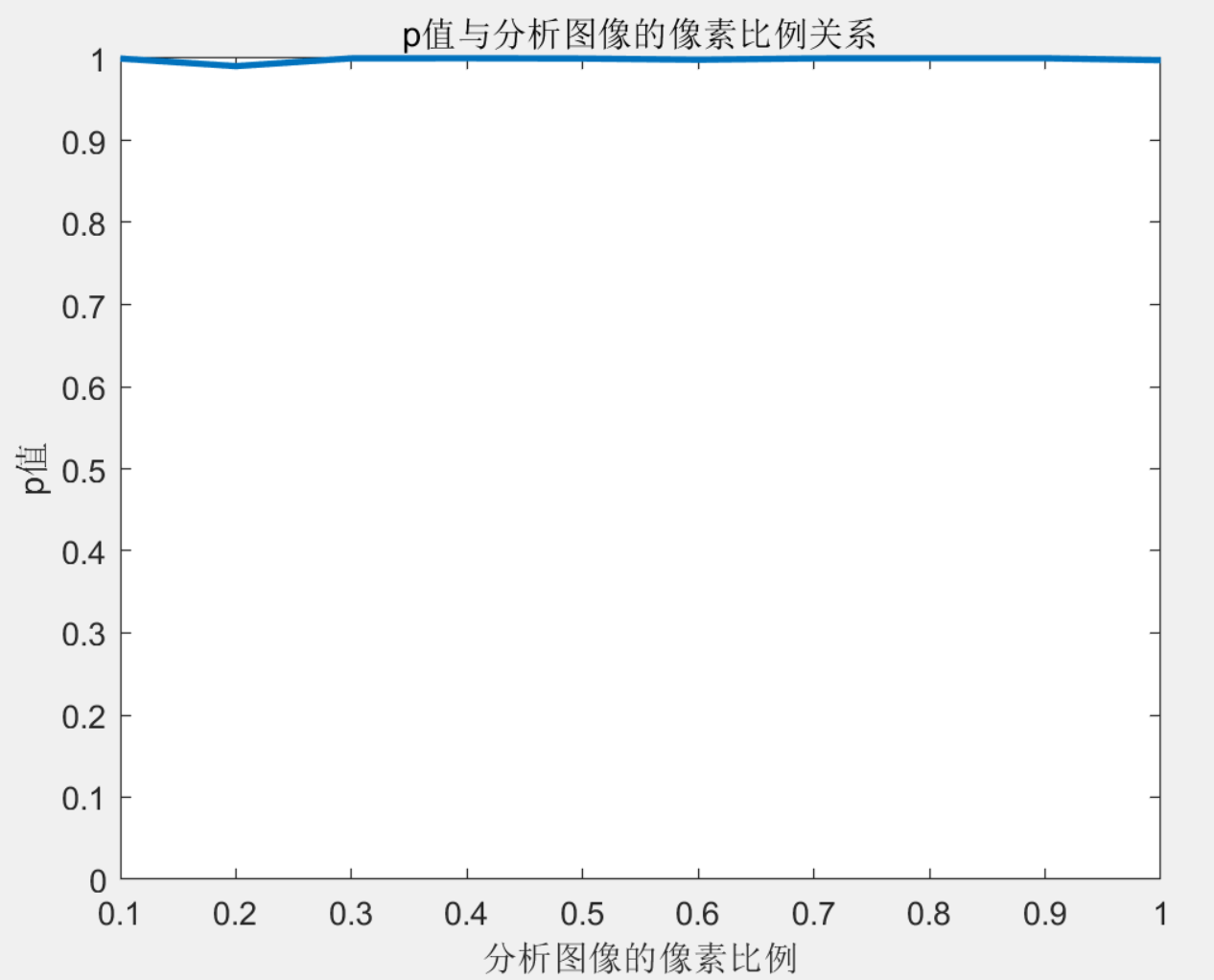
1. 系数0出现的频率最高，且随着系数绝对值的增加，出现的频率逐渐降低。
2. 随着系数绝对值的增加，出现的频率减少的越来越慢。这点可以理解为如果用一条曲线将直方图中的每个bar的中心点连接起来，则边缘部分会更加平滑，即下降的更缓慢。
3. 两侧分布近似对称。



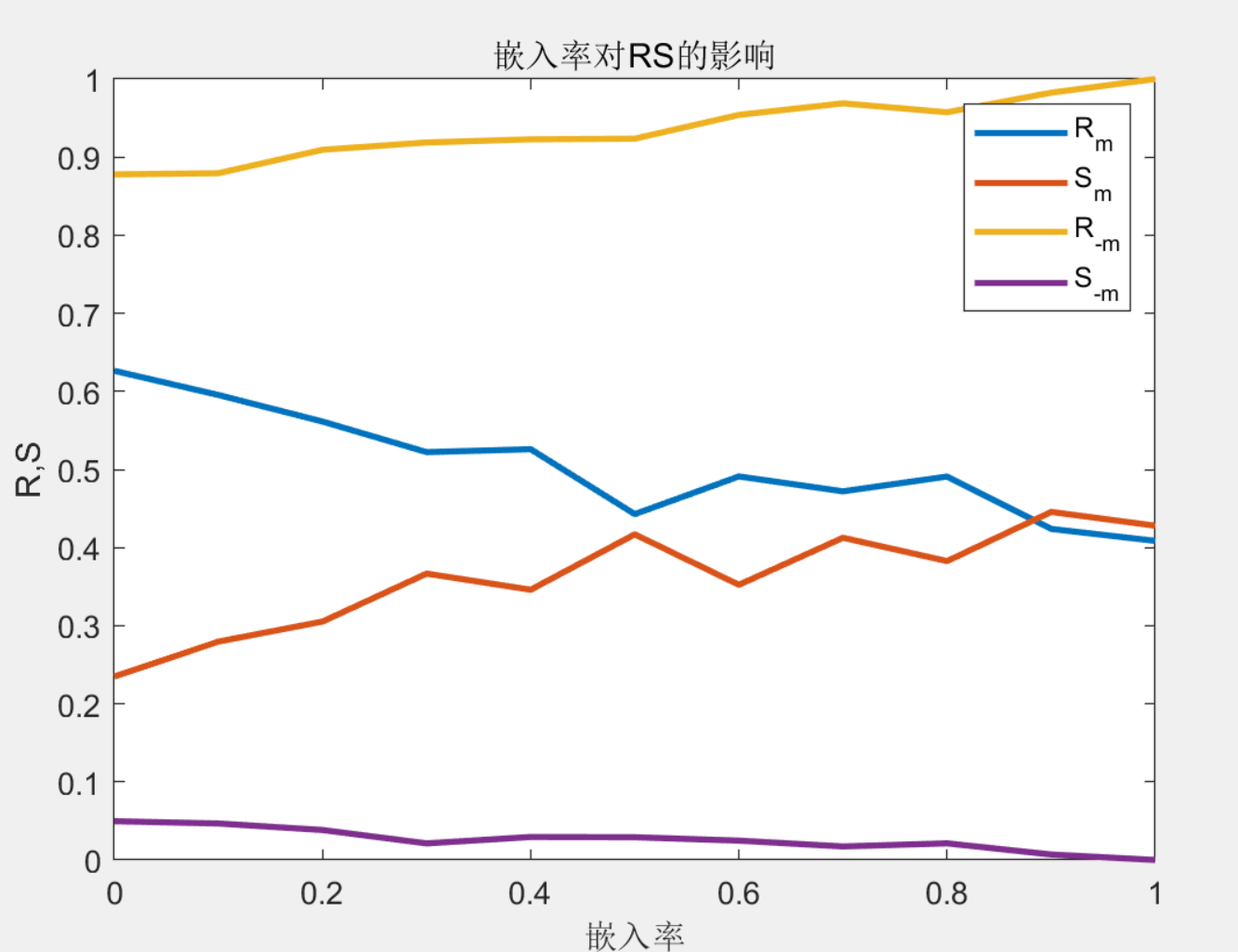
而隐藏后的DCT系数矩阵会出现明显的**值对效应**(除(0,1)外)：



由于Jsteg这种效应，因此我们可以使用卡方分析来进行检测，需要注意的是，在使用卡方分析的时候我们需要忽略**(0, 1)**对。下面是除0、1外均嵌入信息时使用卡方分析的结果，其中横坐标代表分析的比例，纵坐标代表p值，可以发现卡方分析对Jsteg隐写非常有效。

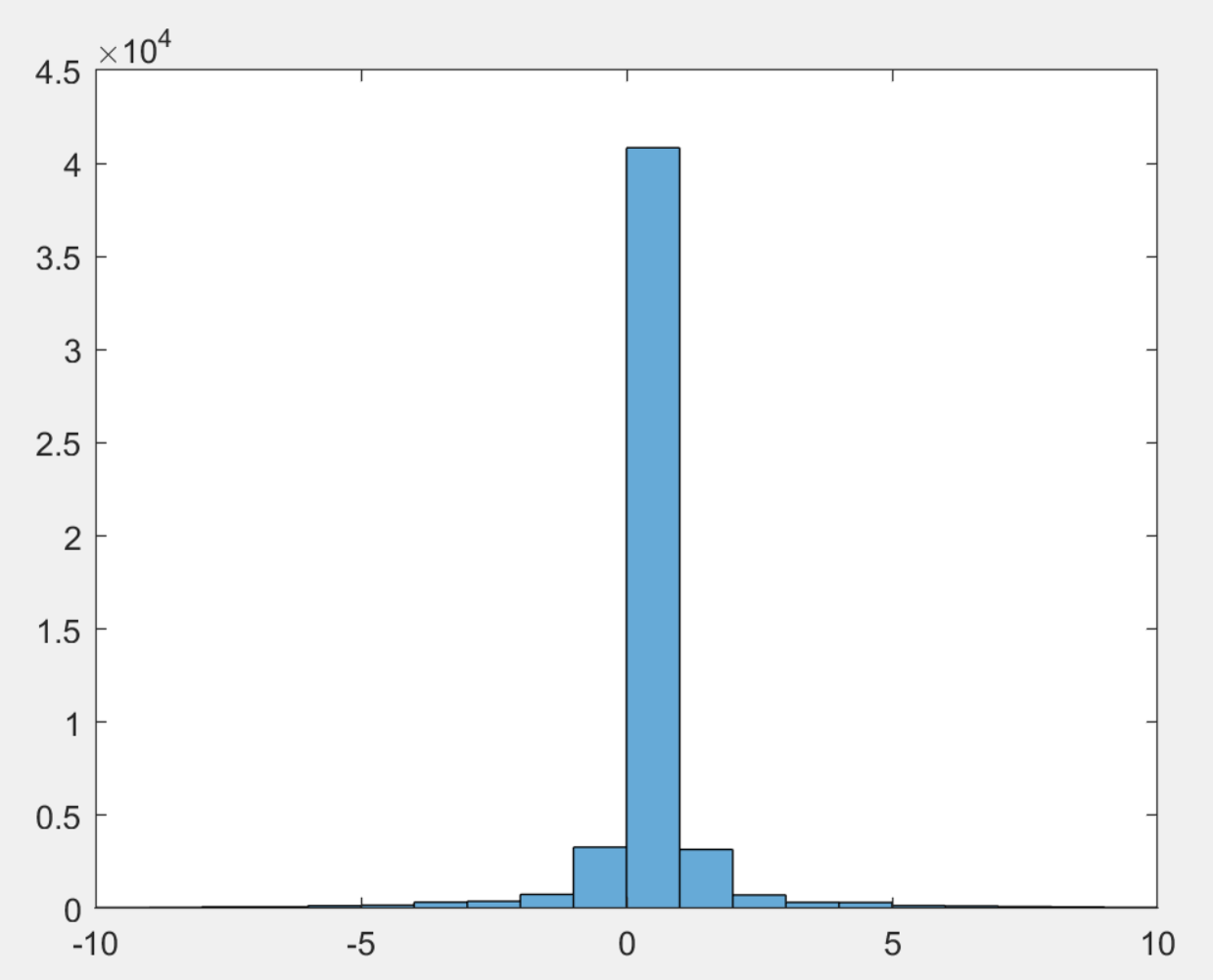


因为不在0、1嵌入信息，而由直方图可知，0、1占据了量化后的DCT系数矩阵的大部分，因此使用RS分析可能并不能很好的对该算法进行检测，下图是RS随嵌入率变化的折线图，可以发现尽管和在空域中相同的变化趋势，但是变化的幅度却比较有限



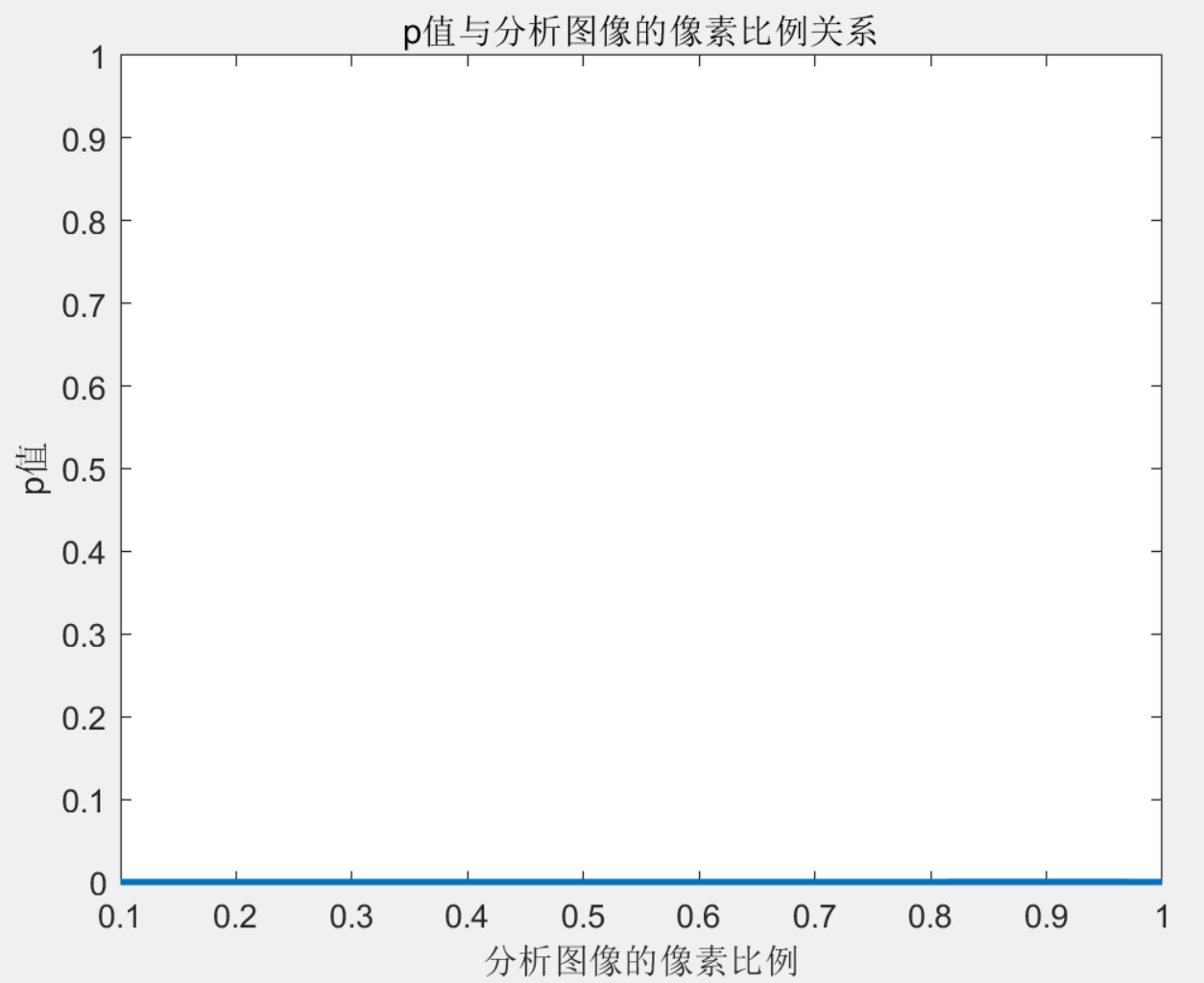
#### F3隐写

下图是使用f3进行隐写得到的DCT系数直方图。



可以发现该F3较好的避免了值对效应，因此抗卡方分析的性能较好，但如果仔细观察的话，可以发现其DCT系数直方图的偶数(例如2、-2)数量略有增加，因此在一定程度上仍然是可以被分析的。

下图是使用卡方分析在每个非0系数均嵌入信息时的结果，可以发现正常的卡方分析对于F3算法的效果较差。



### 拓展思考

#### 使用纠错码克服有损压缩

我们可以发现，在JPEG压缩算法中，**量化**操作是有损的，因此如果直接将信息嵌入量化后的DCT系数矩阵，那么在提取的过程中，由于我们需要再次进行量化操作，因此提取出来的信息并不能保证**完整性**。为了克服该缺点，我们可以首先构造原始数据的纠错码，然后对纠错编码进行传输，这样保证了即使量化后的某些值有精度损失，仍然不影响我们对信息的提取。

下图是在python中使用RS编码的例子，可以发现在Hello, world!中修改了前三个字符之后仍然能够解码出正确的字符串。

