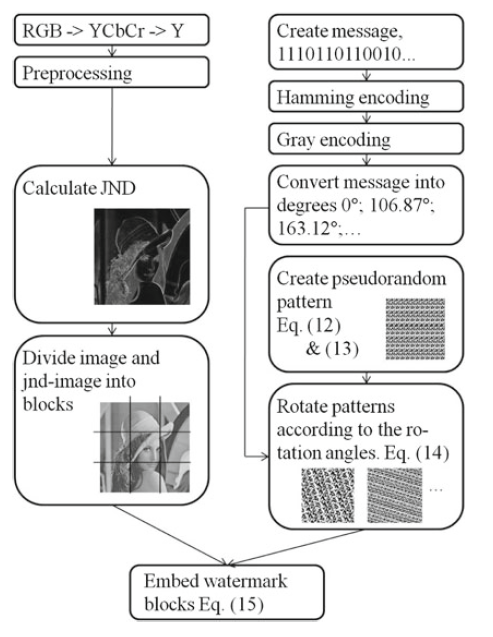
最后一周工作总结

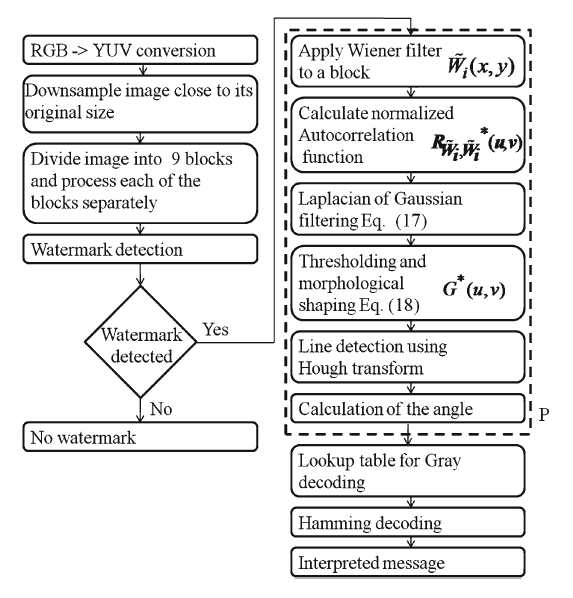
1. 仿真水印算法

首先简单介绍下水印的算法：

1. **嵌入算法（随机块旋转指定角度，嵌入图像）**



1. RGB图像转化为YCbCr图像，之后提取Y分量，之后都是对Y分量操作的。
2. 预处理，为了破坏原图像存在的周期性。
3. 提取图像的JND，JND可以简单理解为像素点能让人眼识别不出最大的阈值。
4. 将图像分成九块，八块嵌入水印信息，一块是参考块。
5. 随机产生26个随机{0,1}集合,之后进行汉明码编码产生32位水印信息。并且以四位计算出一个角度，一共有八个角度。
6. 随机产生28\*7随机块，不断重复随机块，直到能覆盖图像块，之后按照步骤(5)产生的角度进行旋转。
7. 旋转之后的随机块，同时结合JND，嵌入到原图像块中。
8. **提取算法（依据自相关函数）**



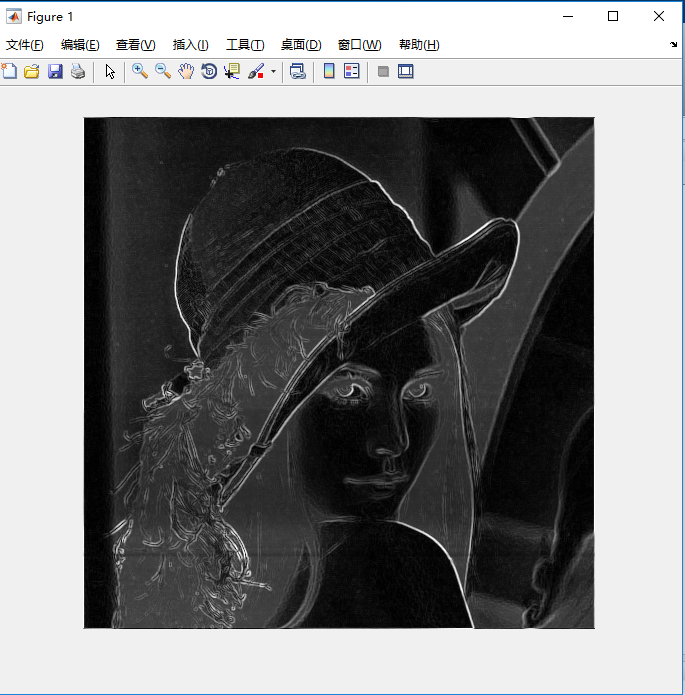
1. 原图像减去维纳滤波之后的图像，得到的就是我们嵌入的信息。
2. 将图像分成九块，分块操作。
3. 对图像每块进行自相关，然后进行高斯变换去噪。
4. 对步骤(3)得到的矩阵，提取峰值。
5. 通过hough直线检测，检测到直线的角度
6. 将步骤(5)得到的角度，转换为二进制信息，之后再通过汉明码检错得到水印信息。
7. 仿真结果展示

**嵌入水印过程**

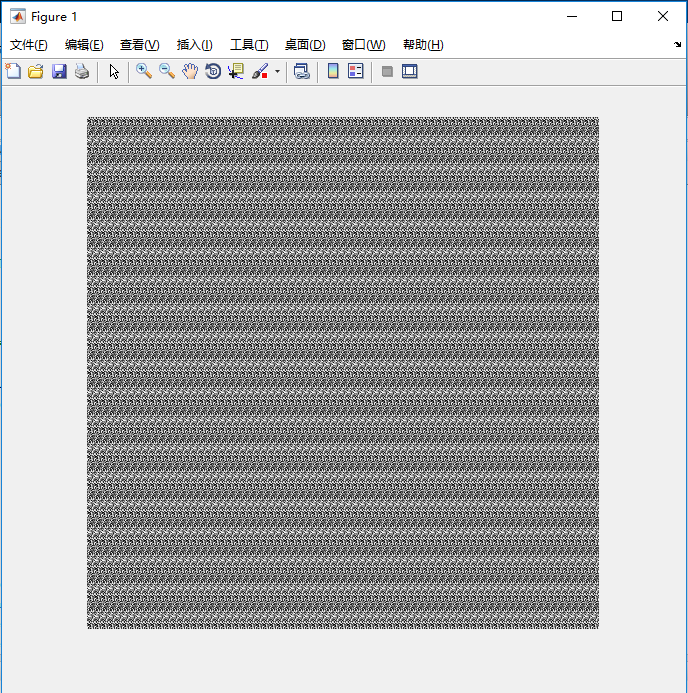
原图像：



图像的JND：



嵌入的随机矩阵：



嵌入水印图像：

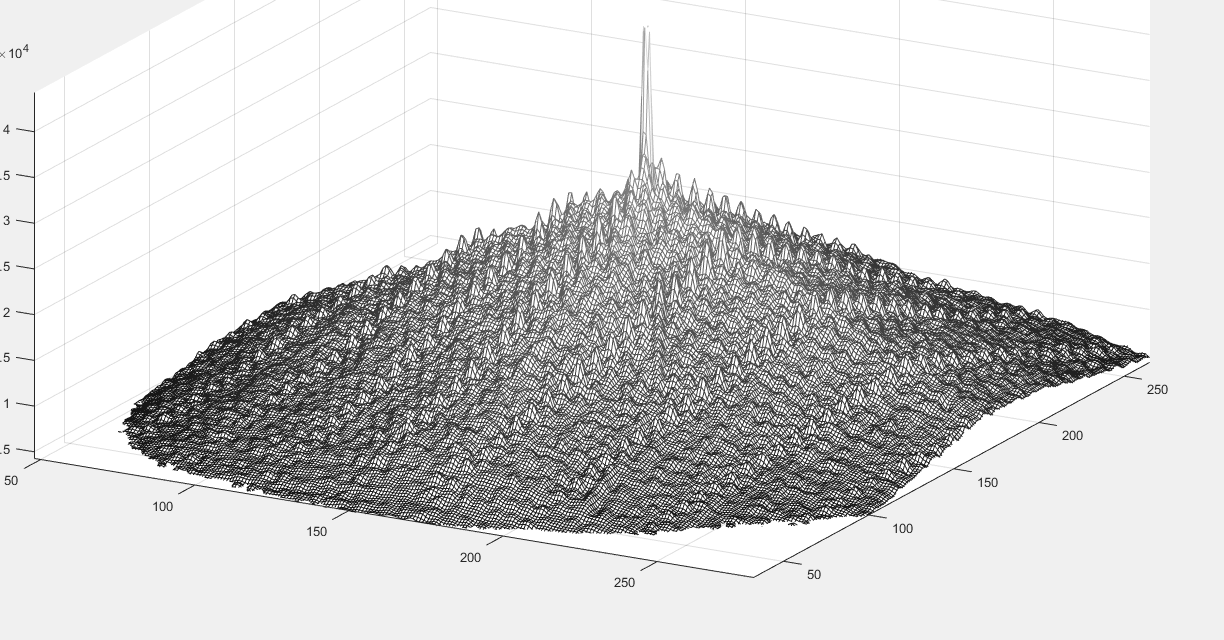


打印之后，相机拍摄的图像：

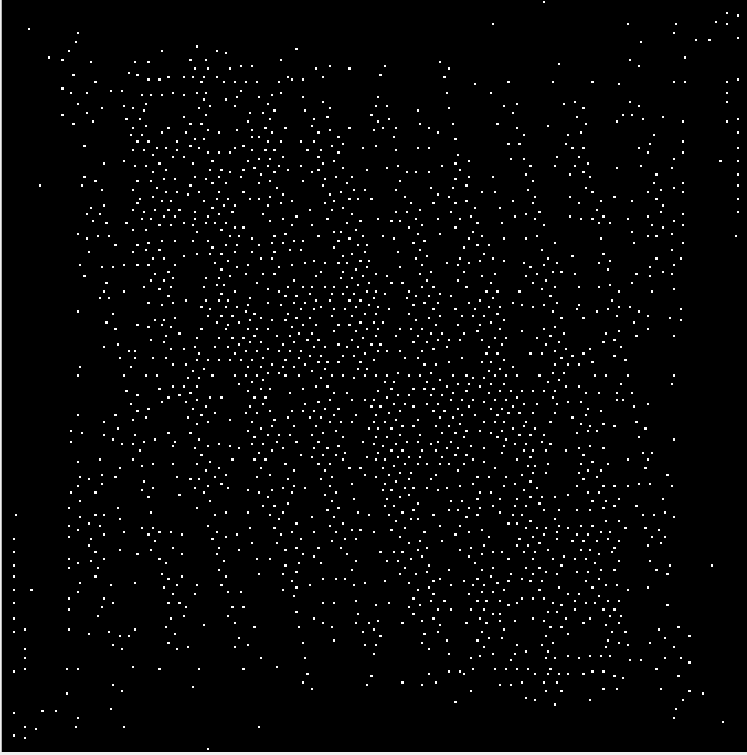


**提取水印过程**

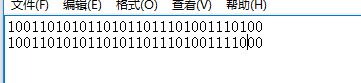
自相关之后的矩阵，从图中看出极值点是在几条平行的直线上



提取极值点之后的图像，从图中能够看到直线



提取水印效果：



提取出的32位中出现了两位的错误，也就是检测九块中出错了一块。

1. 不足之处
2. 嵌入水印的效果图不是特别理想，只要特别仔细看，有些块能看出一些倪端。
3. 拍摄图片之后，提取水印过程不是自动化，此仿真中我需要手动将感兴趣部分抠出。
4. 判断图像是否有水印方面还没有考虑周到，需要之后去更加准确判断。
5. 实验次数太少，之后需要大量的实验来验证此方法的鲁棒性。
6. 后期工作
7. 调整参数，直到水印嵌入之后的图像能够有较好隐藏水印的功能，同时保证鲁棒性较高。
8. 提取水印的过程中，将我去阴影和图像矫正步骤加入，实现程序的自动化。
9. 图像矫正还未实现，之后需要加快速度将此实现。
10. 将matlab程序转化为基于MFC的程序。