## **可视计算与交互概论Lab1报告**

王蕙钰 2100018733

**Task 1: Image Dithering**

**1-1 Uniform Random**

实现思路：首先给每个像素加上[-0.5,0.5]中均匀分布的随机扰动作为白噪音，再使用Threshold算法变成黑白。

效果图：



**1-2 Blue Noise Random**

实现思路：通过给每个像素叠加给定的蓝噪音图像对应的像素添加蓝噪音，再使用Threshold算法变成黑白。

效果图：



**1-3 Ordered**

实现思路：使用笔记7.2.2中的模式，用3\*3的有规律的黑白像素分布代替原图中的一个灰度像素。先遍历input的每个像素，再根据像素亮度依次生成output的3\*3黑白像素分布。

效果图：



**1-4 Error Diffuse**

实现思路：生成input副本，从左到右、从上到下依次遍历input副本每个像素，对于每个像素都先将其二极化，再根据第七章笔记第61页给出的弗洛伊德-斯坦伯格抖动算法的原则将误差赋给其周围的四个像素，更新该四个像素在input副本中的值。最后输出input副本作为output。

效果图：



**Task 2: Image Filtering**

2-1图像模糊：

实现思路：每个像素变为自己所在的3\*3像素区的平均值。（如果在角落则为2\*2，边则为2\*3或3\*2像素区的平均）。

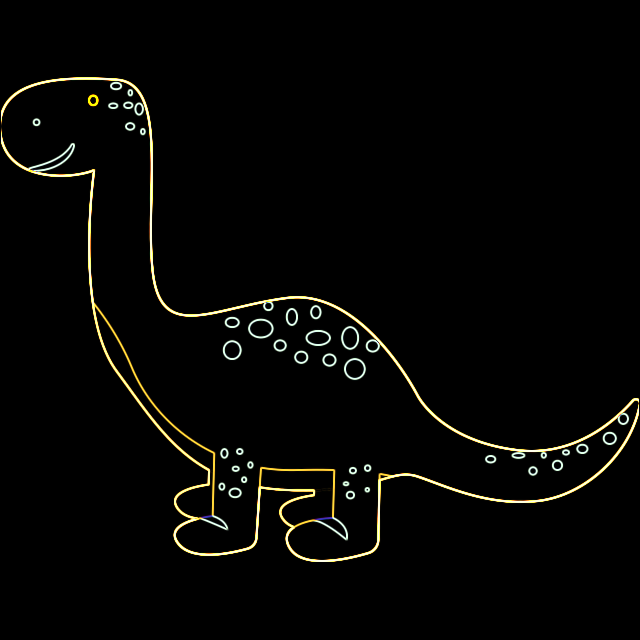
效果图：



2-2图像边缘提取：

实现思路：遍历每个像素，根据第七章讲义7.3.2构造Gx和Gy作为滤波器与3\*3区域点乘，再计算梯度的模长。

效果图：



**Task 3: Image Inpainting**

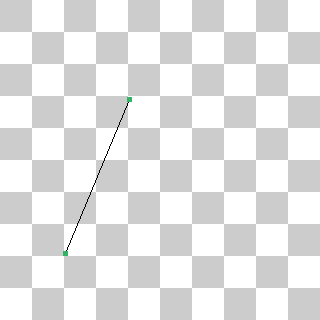
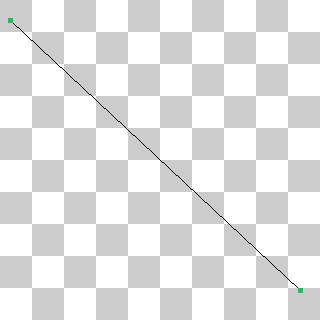
实现思路：填补编辑量g的边界值，即计算边缘处g的值，保证图像内外的连贯性。g为背景未名湖与飞机的差值。

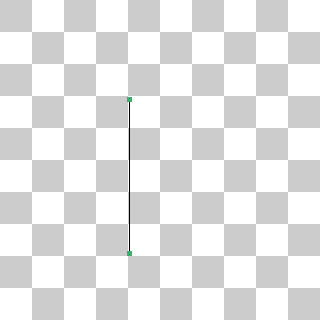
效果图：

**Task 4: Line Drawing**

实现思路：使用布雷森汉姆直线算法，先单独考虑斜率不存在的情况，再写出斜率为0-1的情况，并在此基础上通过调换x、y，或者用-x代替x等方式变幻，写出其他的情况。

效果图：

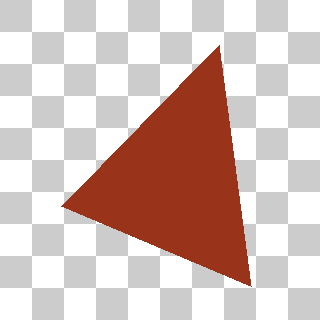
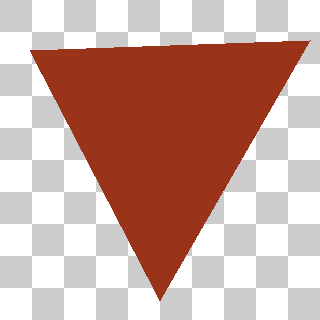


****

**Task 5: Triangle Drawing**

实现思路：使用扫描线算法。先给三个顶点按照y值排序，以中间点所在的平行为x轴的线将三角形划分为两部分，依次画。每一部分都使用DDA算法计算两端，再填充中间。特殊的三角形如有边平行于x轴的情况则单独考虑。

效果图：



**Task 6: Image Supersampling**

实现思路：每一个像素为对应的rate\*rate个像素点的平均。首先求出rate\*rate像素的位置，再通过插值的方式求出每个像素的值以应对像素坐标不为整数的情况，最后求平均。

效果图：



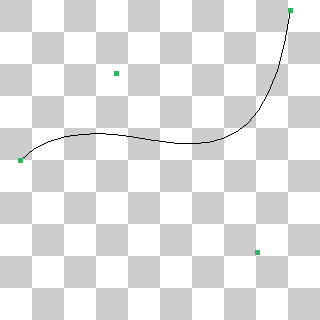
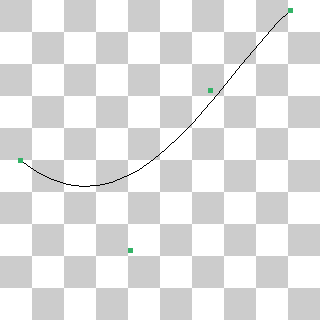
****

（从左到右，从上到下依次为rate = 1，4，8，12时的结果）

**Task 7: Bezier Curve**

实现思路：使用递归的方法，定义一个函数，对相邻的两个控制点做线性插值，直到只剩下一个点为止，该点即为生成的贝塞尔曲线上参数为 t 时对应点的位置。

效果图：

****