**图像处理与内容分析-作业1**

**516021910700 郭远帆**

1. **摘要**

直方图规定化是指将一张图像的灰度直方图（原图）与另一张图像的灰度直方图（目标图）匹配的过程。Canny边缘检测是一种空间域锐化的图像边缘检测算法，其特点在于使用双阈值法提取强边缘与弱边缘并将其连通。本报告介绍了直方图规定化与Canny边缘检算法的流程，并进行了测试与分析。直方图规定化与Canny边缘检测均使用Matlab实现，

1. **算法流程**
   1. **直方图规定化**

直方图规定化的算法流程如下:

**step1:** 将读取的彩色图像转换为灰度图,设为 （原图）和 （目标图）

**step2:** 计算与的灰度分布*,*以及灰度累积分布

**step3:** 创建关于的查找表 = （其中i为中灰度等级i (0<=i<=255),j为中的灰度等级）

**step4:** 根据step3构建的查找表重构,即 (:=为赋值符号)

本实验中还将结果进行了可视化，与原图进行对比

* 1. **Canny边缘检测算法**

Canny边缘检测算法的流程如下:

**step1:**

图像预处理，将得到的彩色图像转换为灰度图*(x,y)*，使用高斯滤波器去噪，得到*(u,v)*

**step2:**

使用sobel算子计算x方向与y方向的二阶偏导。sobel算子的模板如下:

其中分别为x方向和y方向的sobel算子，与进行卷积运算，得:

分别运算梯度幅值与角度:

其中L常用L1范数()，L2范数()

**step3:**

第三步为非极大值抑制，即我们希望在全局梯度的基础上提取局部梯度的最大值，从而获得更加精确的结果。

首先需要对进行分段，将其分为3\*3邻域内的4个扇区,这实际上是一种编码算法，即以每45°角以及其对角线（共90°）作为一个编码，这样一共获得四个扇区,有:

*, n = 0,1,2,3*

在对角度进行近似编码后，在矩阵中进行同方向梯度非极大值抑制。在以i,j为中心的m\*m大小矩形分块（对应sector分块）内，有:

此时得到经过非极大值抑制后的梯度矩阵

**step4:**

第四步即采用双阈值法确定以及连接边缘，也是该算法的核心步骤。对于给定的归一化阈值TL（弱阈值）,TH（强阈值）有:

首先确定强边缘，定义边缘矩阵Edge(x,,y), 其取值范围为0（非边缘）,1(弱边缘),2(强边缘),有:

此时大于强阈值的点被确定为强边缘点。接下来确定弱边缘:

确定弱边缘的思想为: 若某点梯度大于弱阈值而小于强阈值，且其邻域（可分为4邻域与8邻域）内有至少一个强边缘，则确定为弱边缘:

其中, 即(x,y)的邻域。

得到边缘点后即可输出结果。

1. **实验结果与分析**

**3.1 直方图规定化**

采用五对彩色图片进行实验, 得到实验结果如下:

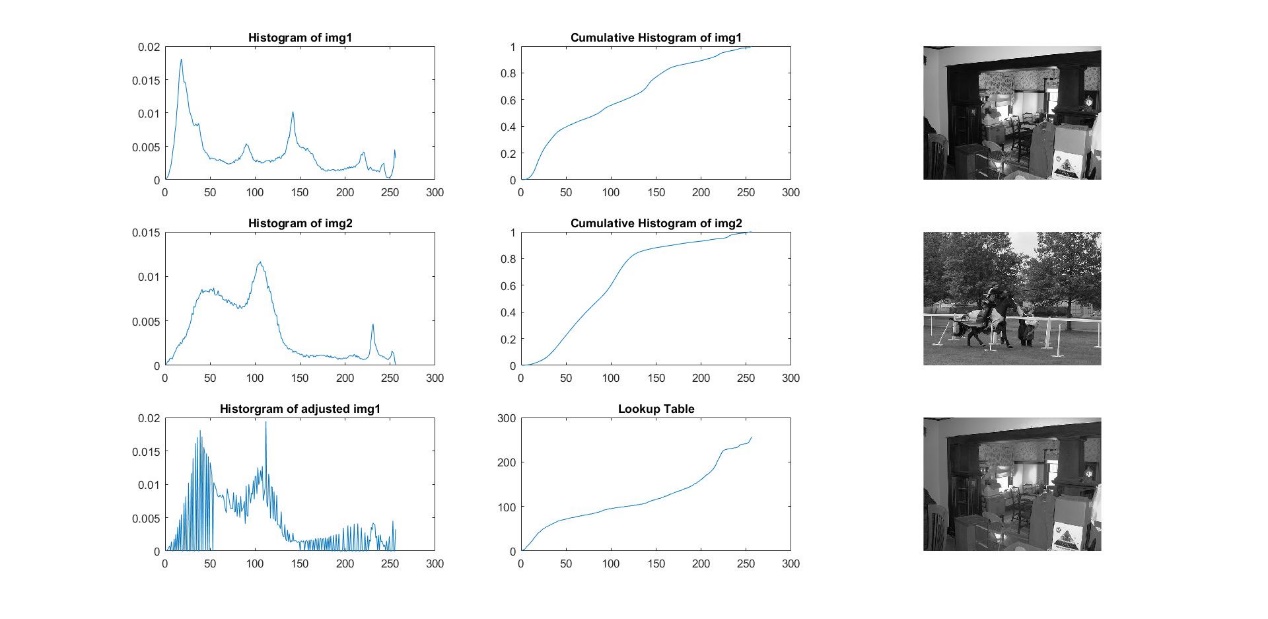


Figure 1

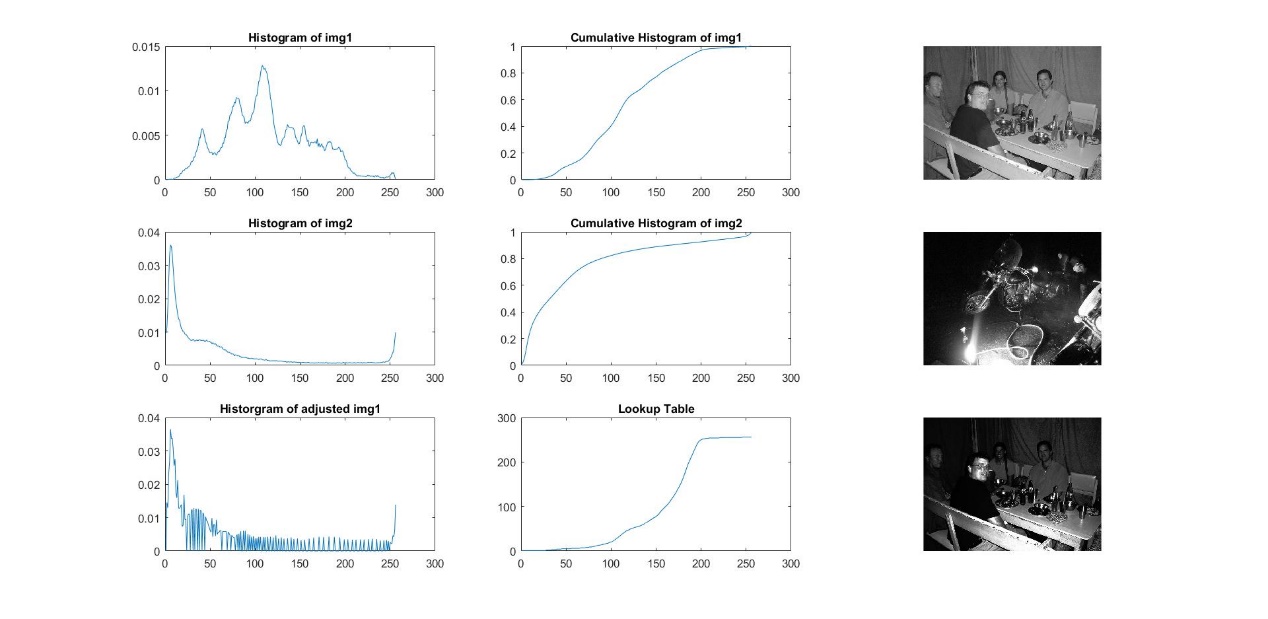


Figure 2

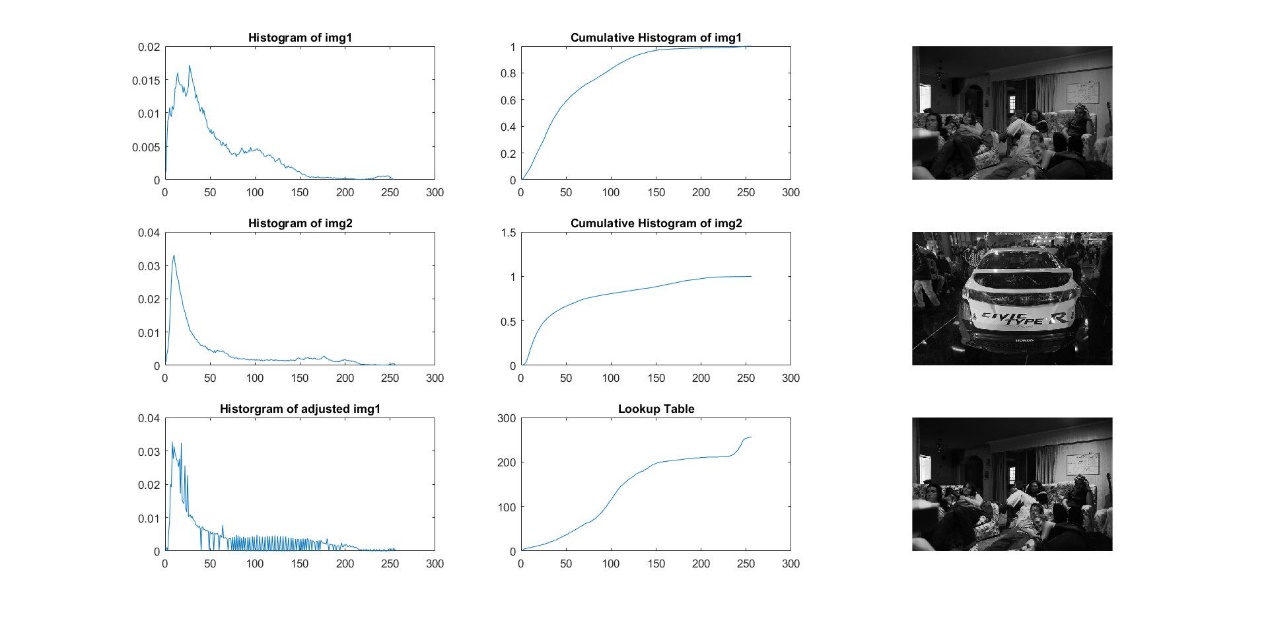


Figure 3

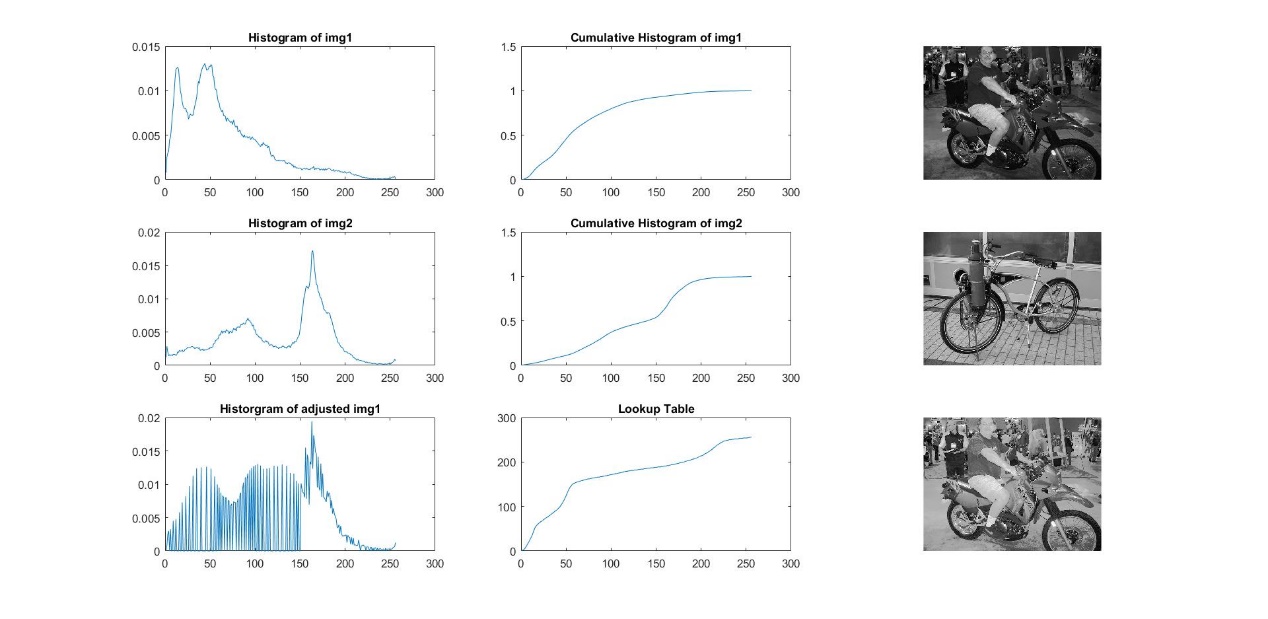


Figure 4

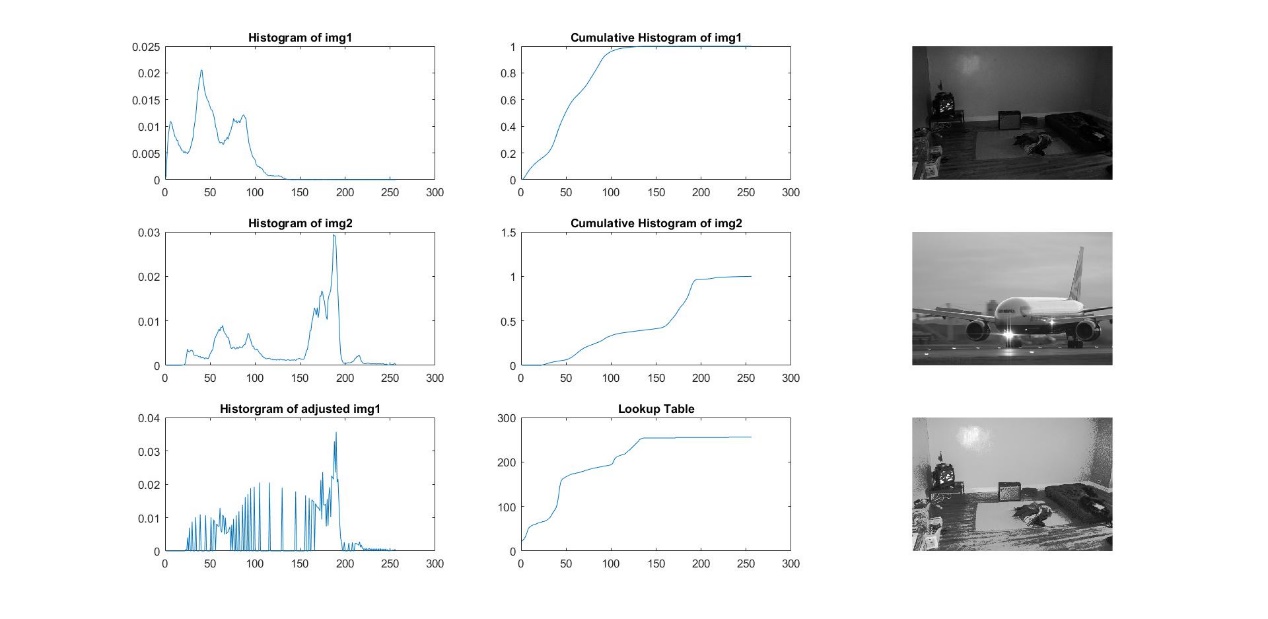


Figure 5

从以上五张图片测试结果来看，直方图规定化实现了灰度分布匹配的过程，算法的实现结果无误。

**3.2 Canny 边缘检测**

采用了一张彩色图片进行测试，在TH（强阈值） =0.2，TL（弱阈值） = 0.1 的情况下分别进行8邻域连接和4邻域连接，得到以下两个实验结果:

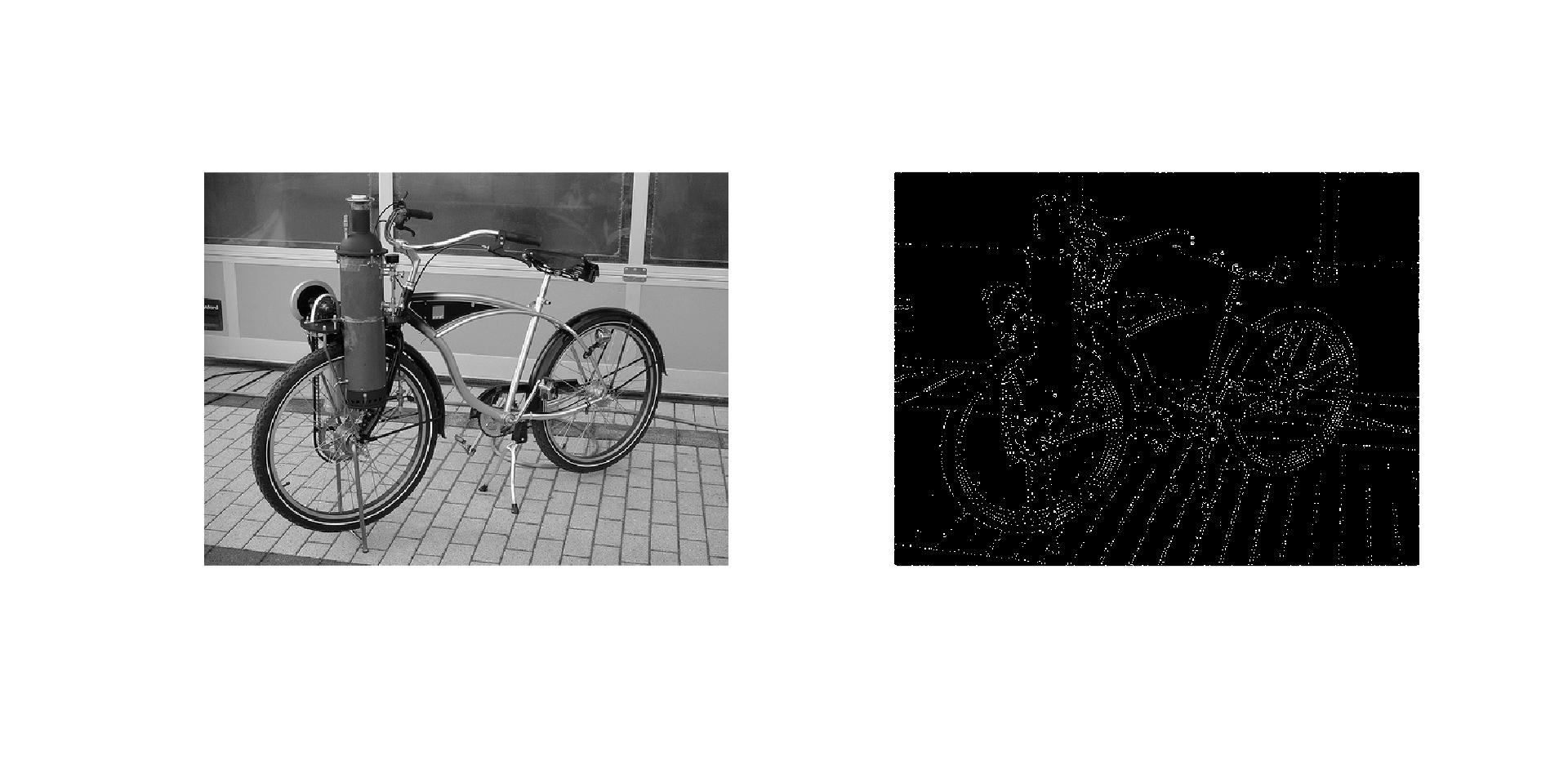


Figure 6 4邻域连接

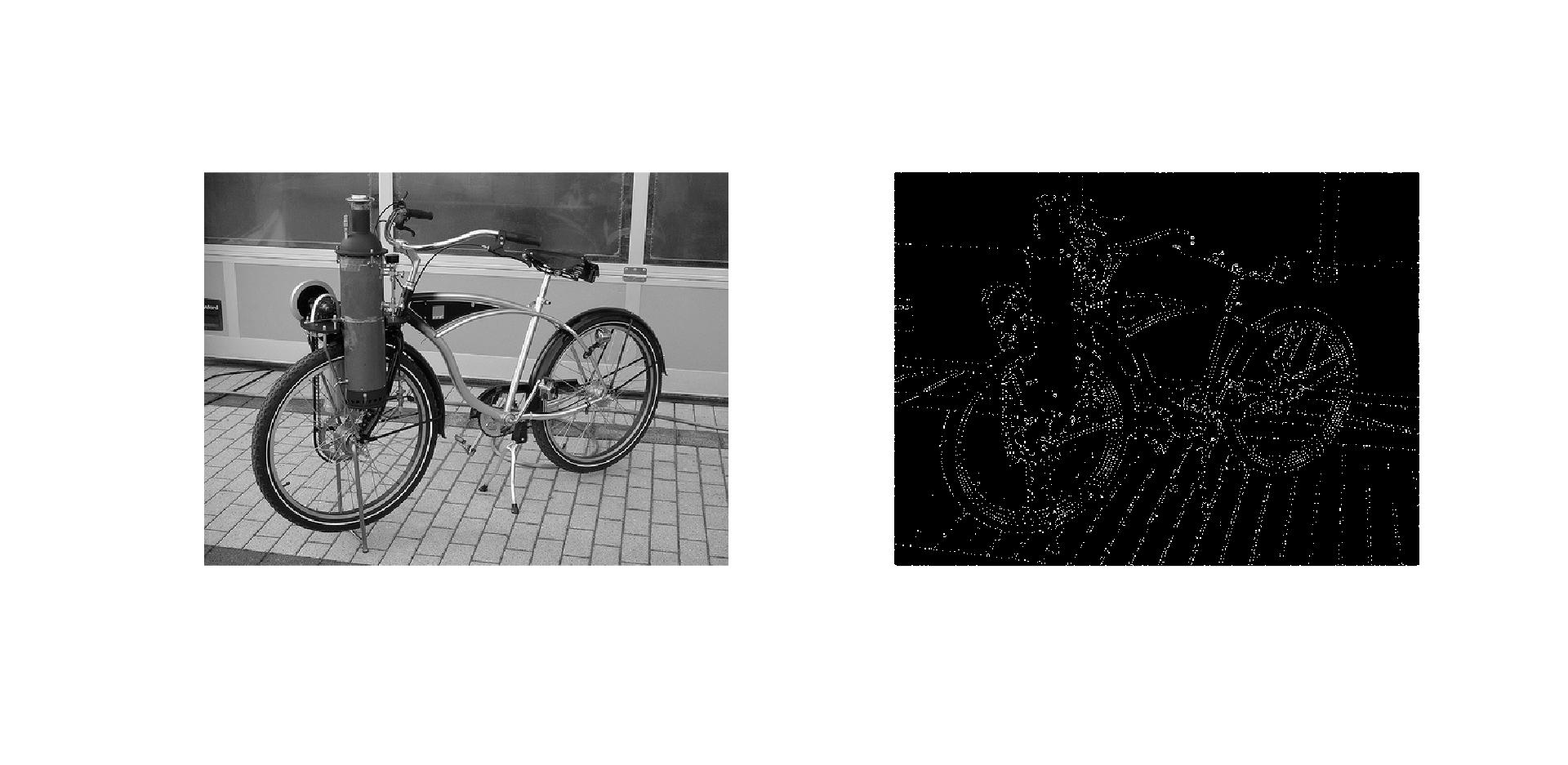


Figure7 8邻域连接

在本图的测试中没有体现出4邻域连接与8邻域连接差距，实际上在本地进行其它图片测试时会发现8邻域连接得到的边缘更加细致和完整。

1. **结果与讨论**
2. **附录**

**CodeList1 His\_specialization.m**

%read image and convert into grey

%img1 is the original image

img1 = imread('./image./9.jpg');

%img2 is the target image

img2 = imread('./image./10.jpg');

img1 = rgb2gray(img1);

img2 = rgb2gray(img2);

im1 = img1;

im2 = img2;

% histogram

hist1 = zeros(1,256);

hist2 = zeros(1,256);

for i = 1:256

hist1(i) = sum(sum(img1==i-1));

hist2(i) = sum(sum(img2==i-1));

end

%Normalization

hist1 = hist1/sum(hist1);

hist2 = hist2/sum(hist2);

%cumulative histogram of img1 and img2

cu\_hist1 = cumsum(hist1);

cu\_hist2 = cumsum(hist2);

I = zeros(1,256);

%build look up table by looking arg min(|cu(I\_i)-cu(I\_j)|

for i = 1:256

I(i) = min(find(abs(cu\_hist2-cu\_hist1(i))==min(abs(cu\_hist2-cu\_hist1(i)))));

%map img1

img1(find(im1==i-1)) = I(i);

end

%calculate histogram of adjusted image

hist3 = zeros(1,256);

for k = 1:256

hist3(k) = sum(sum(img1==k-1));

end

hist3 = hist3/sum(hist3);

%Result Visualization

subplot(331);

plot(hist1);

title('Histogram of img1');

subplot(332);

plot(cu\_hist1);

title('Cumulative Histogram of img1');

subplot(333);

imshow(im1);

subplot(334);

plot(hist2);

title('Histogram of img2');

subplot(335);

plot(cu\_hist2);

title('Cumulative Histogram of img2');

subplot(336);

imshow(im2);

subplot(337);

plot(hist3);

title('Historgram of adjusted img1');

subplot(338);

plot(I);

title('Lookup Table');

subplot(339);

imshow(img1);