

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 边缘检测算法的流程，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。

2. 算法流程

2.1 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 $I_1(x,y)$ （原图）和 $I_2(x,y)$ （目标图）

step2: 计算 I_1 与 I_2 的灰度分布 $h_1(I_1), h_2(I_2)$ 以及灰度累积分布 $C_1(I_1), C_2(I_2)$

step3: 创建 I_1 关于 I_2 的查找表 $T[i] = \arg \min_j |C_1(I_1(i)) - C_2(I_2(j))|$ （其中 i 为 I_1 中灰度

等级 i ($0 \leq i \leq 255$), j 为 I_2 中的灰度等级)

step4: 根据 step3 构建的查找表重构 I_1 , 即 $I_1(\arg I_1(x,y) == i) := j$ ($:=$ 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

2.2 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 $I(x,y)$ ，使用高斯滤波器去噪，得到 $I_1(x,y) = I(x,y) * G(u,v)$

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：

$$H_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$H_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

其中 H_x, H_y 分别为 x 方向和 y 方向的 sobel 算子，与 $I_1(x,y)$ 进行卷积运算，得：

$$\phi_x(x,y) = I_1(x,y) * H_x$$

$$\phi_y(x,y) = I_1(x,y) * H_y$$

分别运算梯度幅值与角度：

$$\phi(x,y) = L(\phi_x, \phi_y)$$

其中 L 常用 L1 范数 ($\phi(x,y) = |\phi_x| + |\phi_y|$)，L2 范数 ($\phi(x,y) = \sqrt{\phi_x^2 + \phi_y^2}$)

$$\theta(x, y) = \arctan \frac{\phi_y}{\phi_x}$$

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 $\theta(x, y)$ 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：

$$\text{sector}(\theta(x, y)) = n, \text{ where } n \times 45^\circ - 22.5^\circ \leq \theta \leq n \times 45^\circ + 22.5^\circ \text{ or } n \times 45^\circ + 157.5^\circ \leq \theta \leq n \times 45^\circ + 202.5^\circ, n = 0, 1, 2, 3$$

在对角度进行近似编码后，在 ϕ 矩阵中进行同方向梯度非极大值抑制。在以 i, j 为中心的 $m \times m$ 大小矩形分块 $W_{ij}(u, v)$ （对应 sector 分块 $S_{ij}(u, v)$ ）内，有：

$$W_i(u, v) = 0 \text{ if } W_i(u, v) \neq \max W_i(\arg_{f, g}(S_i(f, g) = S_i(u, v)))$$

此时得到经过非极大值抑制后的梯度矩阵 $\phi_{nms}(x, y)$

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 $\text{Edge}(x, y)$ ，其取值范围为 0（非边缘），1（弱边缘），2（强边缘），有：

$$\text{Edge}(x, y) = 2 \text{ if } \phi(x, y) > \max \phi \times TH$$

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：

$$\text{Edge}(x, y) = 1 \text{ if } \max \phi \times TL < \phi(x, y) < \max \phi \times TH \text{ and } \exists \text{Edge}(u, v) \geq 2$$

其中 $(u, v) \in \delta(x, y)$ ，即 (x, y) 的邻域。

得到边缘点后即可输出结果。

3. 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：

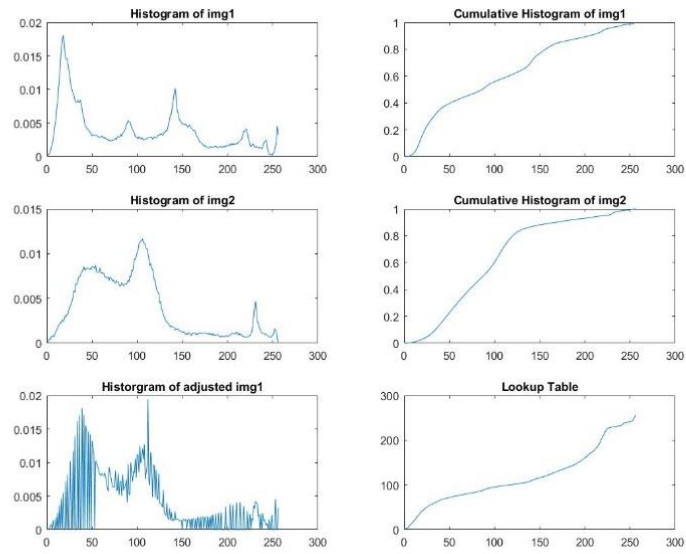


Figure 1

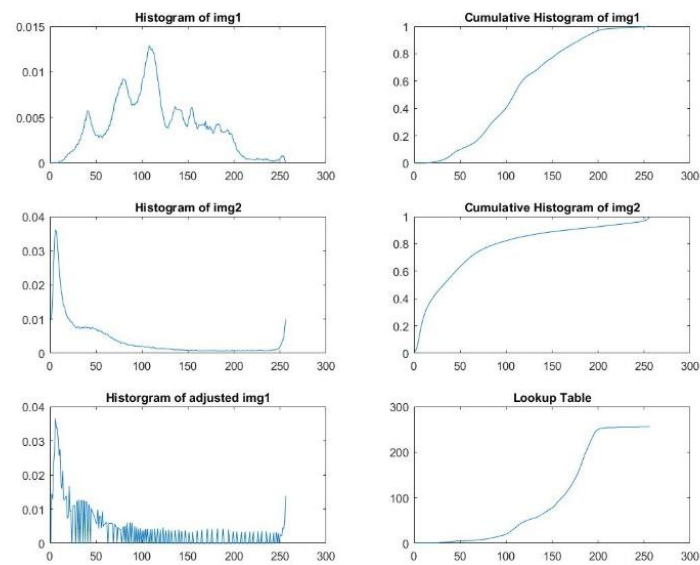


Figure 2

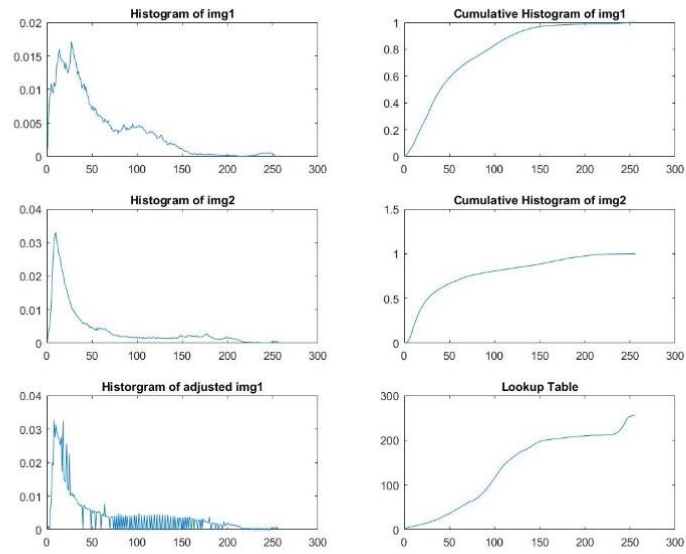


Figure 3

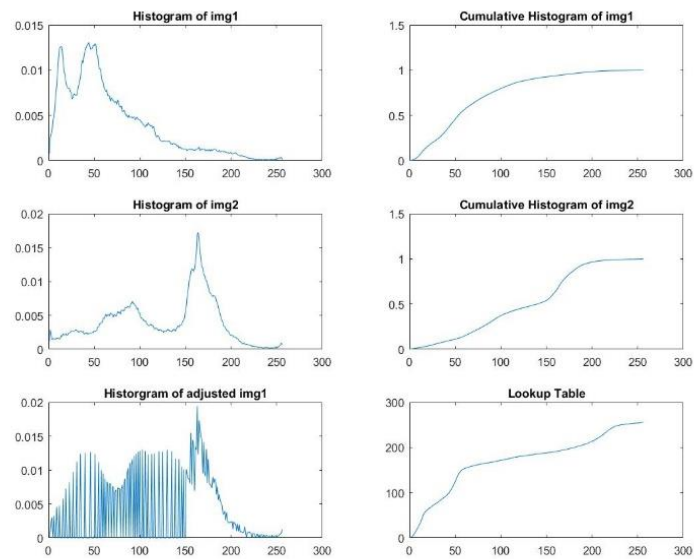


Figure 4

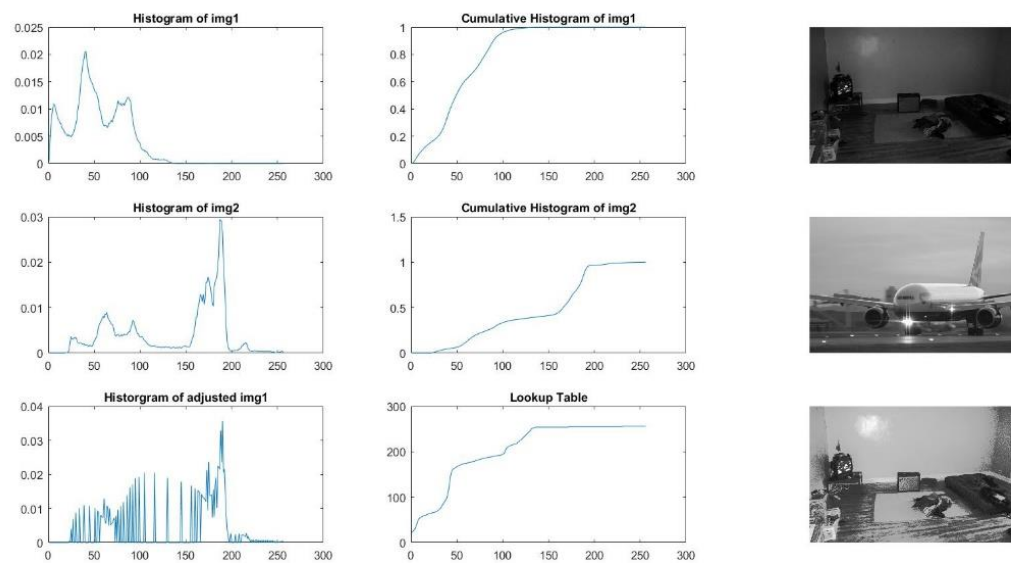


Figure 5

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：

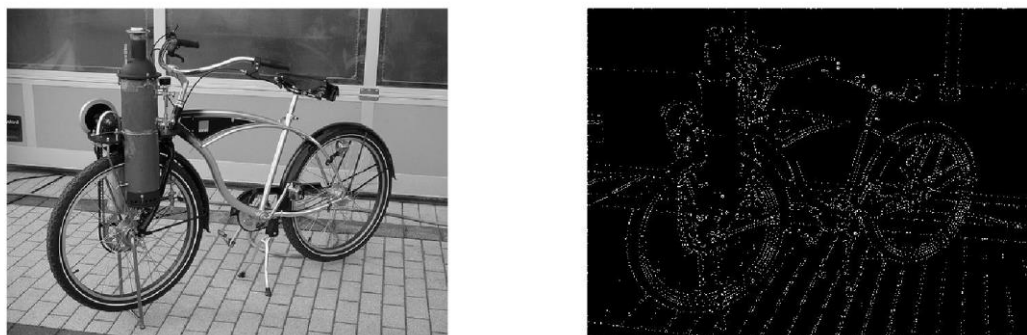


Figure 6 4 邻域连接

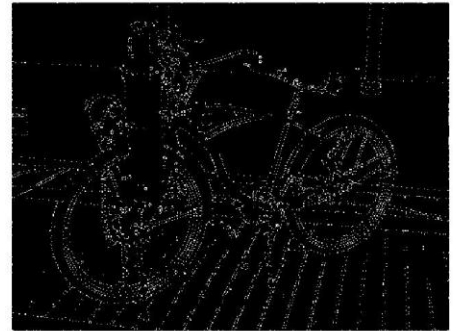


Figure7 8 邻域连接

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。

4. 结果与讨论

在 opencv 中，直方图规定化和 canny 边缘检测均有相关实现，但自己将其实现才能够明白这些图像增强算法的原理。Matlab 是一门十分方便的语言，在本次实验中使用 Matlab 实现两个程序均没有遇到太大的困难。在实现 canny 边缘检测算法过程中，有几个疑惑需要在更多图片的测试中得到解决：

1. 计算梯度时使用 L1 范数和 L2 范数有什么区别？可否使用其它表示方法？
2. 计算偏导的模板除了 sobel 算子，其它算子的效果如何？
3. 填充弱边缘时，四邻域和八邻域的差别有多大？

在提交报告后，在后续过程中将持续对自己设计的算法进行设计与完善，并解决自己提出的几个问题。

5. 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('Histogram of adjusted img1');

subplot(338);
plot(l);
title('Lookup Table');

subplot(339);
imshow(img1);
```

CodeList2 Canny.m

```
%read Image and transfer into gray image
img = imread('./image/8.jpg');
subplot(121);
gray = rgb2gray(img);
imshow(gray);
%Thres hold low/high
TL = 0.1;
TH = 0.2;
%step1: Gaussian filter
sigma = 1;
Im1 = imgaussfilt(gray,sigma);

%% step2: Get Gradient
%using Sobel
Dy = [-1,-2,-1;0,0,0;1,2,1];
Dx = [-1,0,1;-2,0,2;-1,0,1];
phiy = conv2(Im1,Dy);
phix = conv2(Im1,Dx);
phi = sqrt(phix.^2+phiy.^2);
```



```

%Calculating theta
theta = zeros(size(phi,1),size(phi,2));
for i = 1:size(phi,1)
    for j = 1:size(phi,2)
        if phix(i,j) >= 0
            theta(i,j) = atan(phiy(i,j)/phix(i,j))*180/pi;%in degree
        else
            theta(i,j) = atan(phiy(i,j)/phix(i,j))*180/pi+180;
        end
    end
end

%% step3: NMS
% Make theta into section
theta(theta>-22.5 & theta<=22.5) = 0;
theta(theta>22.5 & theta<=67.5) = 1;
theta(theta>67.5 & theta<=112.5) = 2;
theta(theta>112.5 & theta<=157.5) = 3;
theta(theta>157.5 & theta<=202.5) = 0;
theta(theta>202.5 & theta<=247.5) = 1;
theta(theta>247.5 | theta<=-67.5) = 2;
theta(theta>-67.5 & theta<=-22.5) = 3;

is_max = zeros(size(phi,1),size(phi,2));
% 3*3 NMS Window
for x = 2:size(phi,1)-1
    for y = 2:size(phi,2) - 1
        phi_near = phi(x-1:x+1,y-1:y+1);
        phi_near = phi_near.*(theta(x-1:x+1,y-1:y+1)==theta(x,y));
        is_max(x,y) = phi(x,y)==max(max(phi_near));
    end
end

%Mapping!
phi = is_max.*phi;
%% step4:Thres
Edge = zeros(size(phi,1),size(phi,2));
phi_max = max(max(phi));
TH = TH*phi_max;
TL = TL*phi_max;
%4 neibour| 8 neibour
Nei4 = [0,1,0;1,0,1;0,1,0];
Nei8 = [1,1,1;1,0,1;1,1,1];
%High thres mapping
Edge(phi>TH) = 2;
%Low thres mapping

```

```
for i = 2:size(phi,1)-1
    for j = 2:size(phi,2)-1
        if phi(i,j)<TL
            Edge(i,j) = 0;
        elseif phi(i,j)>=TL && phi(i,j)<TH
            Edge(i,j) = sum(sum((Edge(i-1:i+1,j-1:j+1).*Nei8>2))>0);
        end
    end
end
```

```
Edge = Edge(2:end-1,2:end-1);
Edge(Edge>0) = 1;
%Make output
Out = Edge.*double(gray);
Out(Out>0) = 255;
subplot(122);
imshow(Out);
```
