

## 图像处理与内容分析作业 2: Canny 边缘检测算法

whh allesgutewh@gmail.com

### I. 算法设计

#### A. Canny 边缘检测算法

Canny 边缘检测算法是一种最优的阶梯型边缘检测算法, 图像边缘检测要满足两个条件: 一是能有效的抑制噪声、二是必须尽量精确确定边缘的位置。canny 边缘检测算子是根据对信噪比与定位乘积进行测度, 得到的最优化逼近算子, 属于先平滑后求导的方法。

Canny 边缘检测的过程可以分为以下四步:

1. **去噪** 用高斯滤波器平滑图像;
2. **获取边缘梯度大小及方向** 用一阶偏导的有限差分来计算梯度的幅值和方向;
3. **防止重复检测** 对梯度幅值进行极大值抑制, 保证一条边缘只有一次检测;
4. **检测并连接边缘** 使用双阈值算法检测强边缘及与强边缘联通的弱边缘。

#### B. 高斯平滑

高斯平滑能够去除图像的细节部分, 图像噪声可大大减少, 而图像的边缘信息不会损失什么。高斯滤波使用的高斯核是有  $x$ 、 $y$  两个维度的高斯函数, 且两个维度标准差一般相同, 高斯平滑函数表达式为

$$H(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

, 确定高斯核后将其与图像进行卷积最终得到滤波后的图像为  $G(x, y) = f(x, y) * H(x, y)$ 。

#### C. 梯度计算

使用一阶差分卷积计算梯度, 卷积模板为

$$H_1 = \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} \quad H_2 = \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$$

得到 Y 方向和 X 方向梯度分别为

$$\phi_y(m, n) = G(m, n) * H_1(m, n)$$

$$\phi_x(m, n) = G(m, n) * H_2(m, n)$$

对应每个像素点处梯度幅值为

$$\phi(m, n) = \sqrt{\phi_y^2(m, n) + \phi_x^2(m, n)}$$

梯度方向为

$$\theta_\phi(m, n) = \tan^{-1} \frac{\phi_y(m, n)}{\phi_x(m, n)}$$

#### D. 非极大值抑制

全局梯度并不足以确定边缘, 故需要保存局部梯度最大的点, 而抑制非极大值, 即 non-maxima supression(NMS)。

NMS 的目的在于消除边缘检测带来的杂散响应, 起到将边缘“瘦身”的作用。基本方法是将当前像素梯度强度与沿正负梯度方向上的相邻像素的梯度强度进行比较, 若其最大 (即为极值), 则保留该像素为边缘点, 若不是最大, 则对其进行抑制, 不将其作为边缘点。在实际操作中, 将梯度的方向  $\theta$  近似为  $\frac{\pi}{4}$  的整数倍, 得到五种数值 0, 1, -1,  $\pm$ , 分别对应图 1 中的区域 0, 1, 3, 2。

比较梯度指示区域与对应中心元素的数值大小, 若中心元素的值不是最大值则将该元素在 NMS 矩阵中置为零, 否则保留原数值。

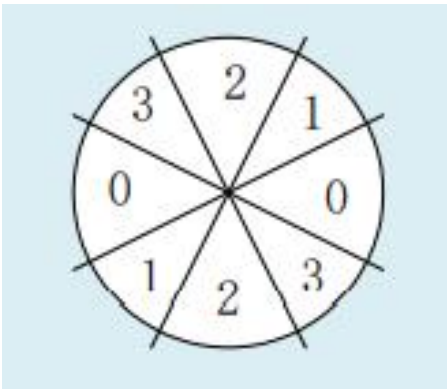


Figura 1. 邻域扇区示意图

E. 双阈值算法

NMS 后的矩阵仍可能存在假边缘端，解决此问题的典型方法是对其使用一个阈值，将低于该阈值的所有值置零，但若仅使用单个阈值可能会造成边缘不连续，故使用双阈值阀，假设所设置的双阈值分别为  $\tau_1, tau_2(\tau_1 < \tau_2)$ ，对所有小于  $\tau_1$  的值置 0，所有大于  $\tau_2$  的值置 1，对  $(\tau_1, \tau_2)$  范围内的值作判断，若其八个相邻像素点中有大于  $\tau_2$  的，则置为 1 否则置为 0。

II. 实验结果

A. 高斯滤波



Figura 2. 3x3 高斯核



Figura 3. 5x5 高斯核

图 2、3 所示为同一张图片经过相同  $\sigma$  值、不同大小高斯核的高斯滤波后得到的输出图片，通过对比可以得到，高斯核为 5x5 的输出图片相比 3x3 的输出更为模糊，这与理论相符，因为理论上高斯核越大，越

接近真正的高斯分布，图像也越“平滑”。

B. 梯度检测和 NMS

图 4、5 分别展示了图片梯度幅值和梯度矩阵经 NMS 后的输出，可以看到 NMS 后的输出相比梯度幅值线条更细，证明该方法可以有效消除边缘检测带来的杂散响应。



Figura 4. 梯度幅值输出

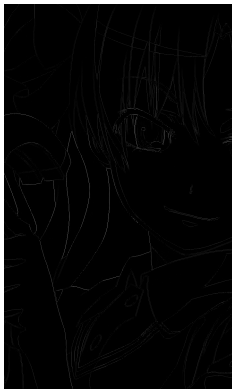


Figura 5. NMS 输出

C. 最终输出



Figura 6. 图片 1 原图



Figura 7. canny 输出 1

图 6、7 展示了该图片原图及经过 canny 边缘检测算法后的输出图像，处理该图片使用的阈值为  $(0.1m, 0.2m)$ ，其中  $m$  为 NMS 输出矩阵中的最大值。从 canny 算法输出的图片中可以比较清晰的看到原图的轮廓，认为对于该图片边缘检测是成功的。

图 8、9 展示了使用相同参数下另一图片及其输出



Figura 8. 图片 2 原图



Figura 9. canny 输出 2

结果，可以看到，在这张图片中人物的面部细节并没有被提取，考虑将阈值降低得到 (0.08m, 0.16m)，得到如图 10 所示的输出图片，可以看到细节有所增加，将阈值进一步降低至 (0.05m, 0.1m) 如图 11 所示，此次输出包含了更加丰富的面部信息，但同时也包含了更多噪声。

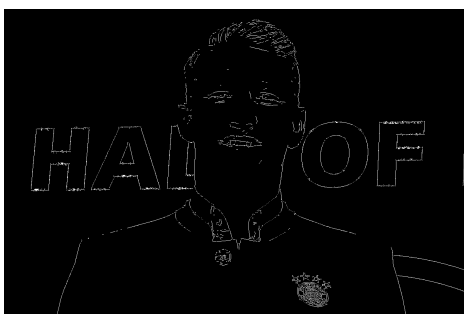


Figura 10. 阈值 (0.08m, 0.16m)

考虑到最后输出的图片和原图的大小一致，而在 canny 算法处理前有将彩色图片转换为灰度图的操作，尝试将边缘对应的彩色图像像素点的色彩还原，对图 10 作此处理后得到如图 12 所示的结果，从结果来看，虽

然有彩色轮廓出现但色彩轮廓能提供的信息也很有限，

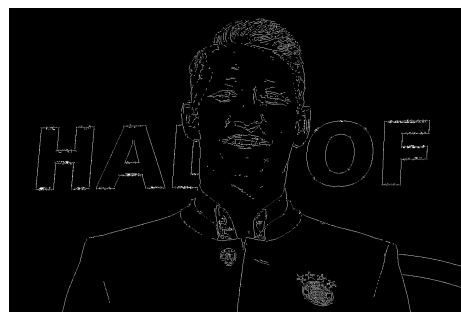


Figura 11. 阈值 (0.05m, 0.1m)

且将图片背景由黑色转换为白色后发现提取到的边缘连续性不如黑色背景，这种差距是人眼视觉特性造成的，而转换后的效果也暴露了该算法的缺陷，也就是仅通过双阈值法并不能完美实现边缘的连续。



Figura 12. 彩色轮廓

### III. 总结与体会

使用 canny 算子检测边缘时，要想取得较为理想的结果，需要针对不同类型的图片使用不同的参数，可以调节的参数很多，比如高斯核的大小和  $\sigma$  值、双阈值、计算梯度的差分卷积矩阵等。

对于边缘检测这一任务，虽然 canny 算法能够获取比较令人满意的结果，但在调节参数过程中也会面临细节不足和噪声增多问题。几年前在机器学习课程中去了解了一些抠图的项目，当时看上去效果很不错，就理所当然的认为边缘检测是一项很成熟的技术，通过这个作业，发现效果并不像想象中那么完美，当时所看到的项目也仅仅是在特定风格的图片上效果较好。此次通过实践认识到，我们的科技还没有想象中那么强大……