Computer Vision: Representation and Recognition

Assignment 2

161180038, 广进, guangjin1998@gmail.com

April 30, 2019

1 Canny Edge Detector (30 points)

1.1 Will the rotated edge be detected using the same Canny edge detector?

假设之前的点为 (x, y)，对应函数为 $f(x, y)$，旋转之后点为 (x', y')，对应函数为 $g(x', y') = f(x, y)$，经过旋转有如下关系：

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \Rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} \tag{1.1}$$

则有，

$$\begin{aligned} \frac{\partial g(x', y')}{\partial x'} &= \frac{\partial f(x, y)}{\partial x'} \\ &= \frac{\partial f(x, y)}{\partial x} \frac{\partial x}{\partial x'} + \frac{\partial f(x, y)}{\partial y} \frac{\partial y}{\partial x'} \\ &= \frac{\partial f(x, y)}{\partial x} cos(\theta) + \frac{\partial f(x, y)}{\partial y} (-sin(\theta)) \end{aligned}$$

$$\begin{aligned} \frac{\partial g(x', y')}{\partial y'} &= \frac{\partial f(x, y)}{\partial y'} \\ &= \frac{\partial f(x, y)}{\partial x} \frac{\partial x}{\partial y'} + \frac{\partial f(x, y)}{\partial y} \frac{\partial y}{\partial y'} \\ &= \frac{\partial f(x, y)}{\partial x} sin(\theta) + \frac{\partial f(x, y)}{\partial y} cos(\theta) \end{aligned}$$

从而有，

$$\begin{aligned}(\frac{\partial g(x',y')}{\partial x'})^2 &= (\frac{\partial f(x,y)}{\partial x}cos(\theta)+\frac{\partial f(x,y)}{\partial y}(-sin(\theta)))^2 \\ &= (\frac{\partial f(x,y)}{\partial x}cos(\theta))^2+(\frac{\partial f(x,y)}{\partial y}sin(\theta))^2-2\frac{\partial f(x,y)}{\partial x}\frac{\partial f(x,y)}{\partial y}sin(\theta)cos(\theta)\end{aligned}$$

$$\begin{aligned}(\frac{\partial g(x',y')}{\partial y'})^2 &= (\frac{\partial f(x,y)}{\partial x}sin(\theta)+\frac{\partial f(x,y)}{\partial y}cos(\theta))^2 \\ &= (\frac{\partial f(x,y)}{\partial x}sin(\theta))^2+(\frac{\partial f(x,y)}{\partial y}cos(\theta))^2+2\frac{\partial f(x,y)}{\partial x}\frac{\partial f(x,y)}{\partial y}sin(\theta)cos(\theta)\end{aligned}$$

故有 the magnitude of its derivative：

$$\sqrt{(\frac{\partial g(x',y')}{\partial x'})^2+(\frac{\partial g(x',y')}{\partial y'})^2}=\sqrt{(\frac{\partial f(x,y)}{\partial x})^2+(\frac{\partial f(x,y)}{\partial y})^2} \tag{1.2}$$

Therefore, the rotated edge will be detected using the same Canny edge detector.

1.2 how to adjust the threshold (up or down) to address both problems

Canny 算法中减少假边缘数量的方法是采用双阈值法[1]。选择两个阈值，根据高阈值得到一个边缘图像，这样一个图像含有很少的假边缘，但是由于阈值较高，产生的图像边缘可能不闭合，未解决这样一个问题采用了另外一个低阈值。在高阈值图像中把边缘链接成轮廓，当到达轮廓的端点时，该算法会在断点的 8 邻域点中寻找满足低阈值的点，再根据此点收集新的边缘，直到整个图像边缘闭合。

Long edges are broken into short segments separated by gaps: 是因为介于高阈值和低阈值中间没有足够的候选者，无法产生闭合边。所以应该将低阈值降低以有更多候选者。

Some spurious edges appear: 是因为假边有一部分误以为是必须要的，应该通过提高高阈值来抑制假边。

最简单方法[2]：使用平均值或者中位数。令 high threshold 为 1.33 倍的平均值/中位数，low threshold 为 0.67 倍的平均值/中位数

[1] Canny 边缘检测算法原理及其 VC 实现详解 (一) https://blog.csdn.net/likezhaobin/article/details/6892176

[2] Canny Edge Detection Auto Thresholding http://www.kerrywong.com/2009/05/07/canny-edge-detection-auto-thresholding/

2 Difference-of-Gaussian (DoG) Detector (30 points)

本部分代码请见 DoG.ipynb，不过图示部分均在 pdf 有展示。

2.1 2nd derivative with respect to x

The 1-D Gaussian is

$$g_{sigma}(x) = \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{x^2}{2\sigma^2})$$

1st derivative with respect to x is

$$\begin{aligned} g'_{sigma}(x) &= \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{x^2}{2\sigma^2}) * (-\frac{x}{\sigma^2}) \\ &= -\frac{x}{\sqrt{2\pi}\sigma^3} exp(-\frac{x^2}{2\sigma^2}) \end{aligned}$$

2nd derivative with respect to x is

$$\begin{aligned} g''_{sigma}(x) &= -\frac{1}{\sqrt{2\pi}\sigma^3} exp(-\frac{x^2}{2\sigma^2}) - \frac{x}{\sqrt{2\pi}\sigma^3} exp(-\frac{x^2}{2\sigma^2}) * (-\frac{x}{\sigma^2}) \\ &= \frac{1}{\sqrt{2\pi}\sigma^3} (\frac{x^2}{\sigma^2} - 1) exp(-\frac{x^2}{2\sigma^2}) \end{aligned}$$

use Python to plot it (use = 1)

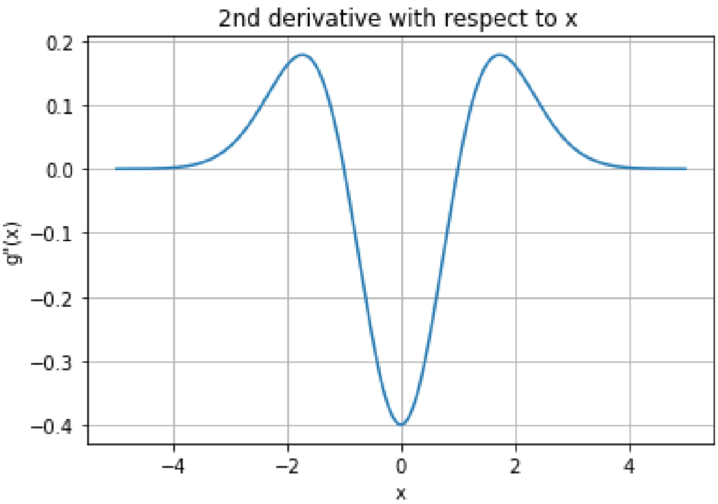
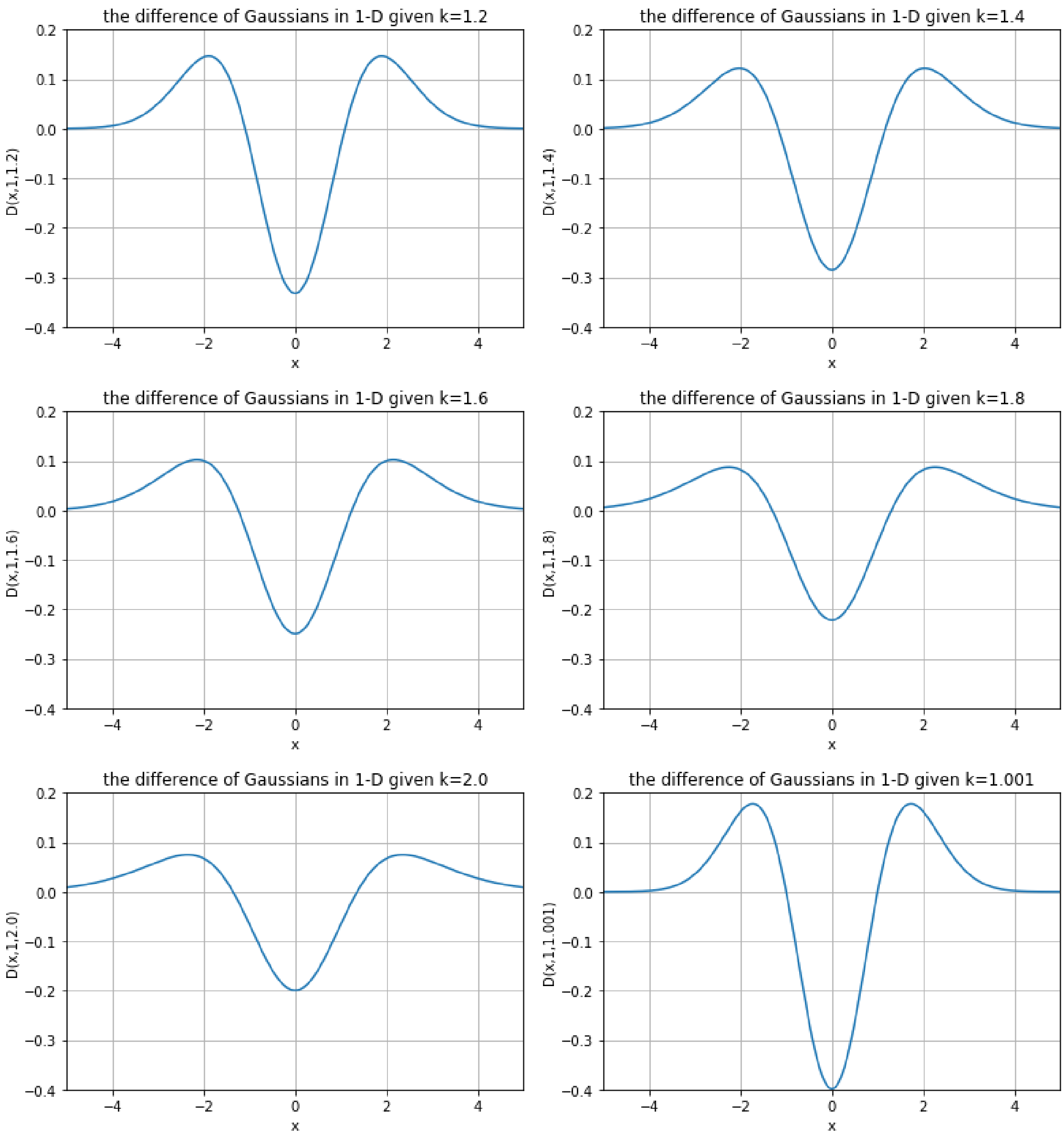


Figure 1: 2nd derivative with respect to x

2.2 plot the difference of Gaussians in 1-D

Use Python to plot them (use = 1, k = 1.2, 1.4, 1.6, 1.8, 2.0), and k = 1.2 gives the best approximation to the 2nd derivative with respect to x.



Morever, We can see that 1st derivative with respect to σ is

$$
\begin{aligned}
\frac{\partial g_{sigma}}{\partial \sigma} &= \frac{1}{\sqrt{2\pi}}(-\frac{1}{\sigma^2})exp(-\frac{x^2}{2\sigma^2}) + \frac{1}{\sqrt{2\pi}\sigma}exp(-\frac{x^2}{2\sigma^2})(-\frac{x^2}{2})(-2\frac{1}{\sigma^3})) \\
&= \frac{1}{\sqrt{2\pi\sigma^2}}(\frac{x^2}{\sigma^2} - 1)exp(-\frac{x^2}{2\sigma^2}) \\
&= \sigma\frac{\partial^2 g_{sigma}}{\partial^2 x}
\end{aligned}
$$

When $\sigma = 1$, $\frac{\partial g_{sigma}}{\partial \sigma} = \frac{\partial^2 g_{sigma}}{\partial^2 x}$, so $k \to 1$ gives the best approximation to the 2nd derivative with respect to x. And we can see k=1.001 is better than k=1.2

2.3 The 2D equivalents of the plots above are rotationally symmetric. To what type of image structure will a difference of Gaussian respond maximally?

由下图[3]结合上面 DoG 图像，可以看到，DoG 对中心点负响应最大，周围有一圈正响应。所以如果做卷积，对于黑点（周围白背景且点的范围也要合适）响应最大。

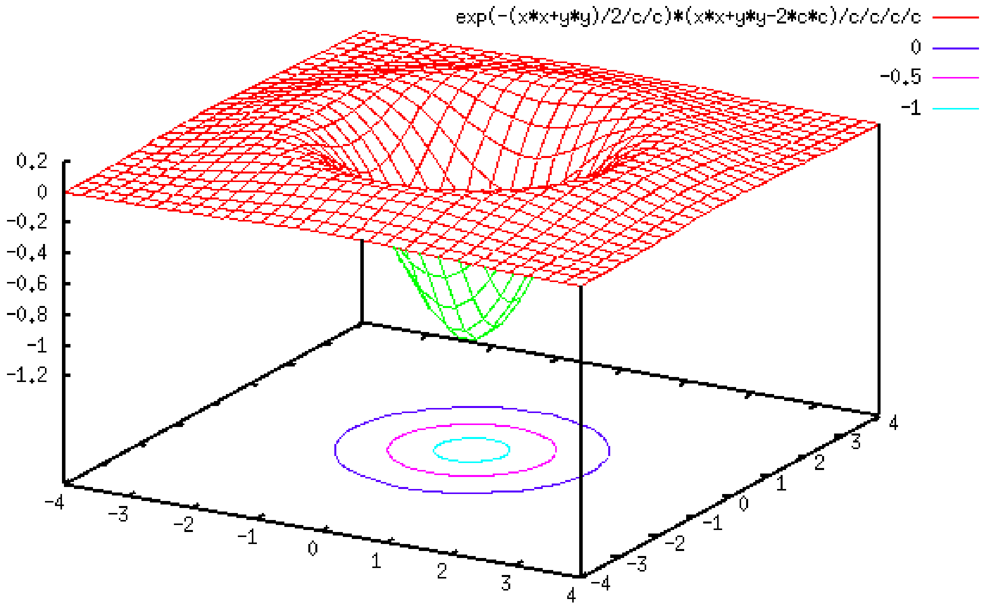


Figure 2: 2D 高斯二阶导

另外，使用 DoG 算子对图像做处理，其极大值和极小值还可以检测角点。

[3]图片来自于 https://blog.csdn.net/pi9nc/article/details/18619893

3 Edge detector(40 points)

本部分代码请见 Edge_detector.py, 详细使用方法请见 README.md

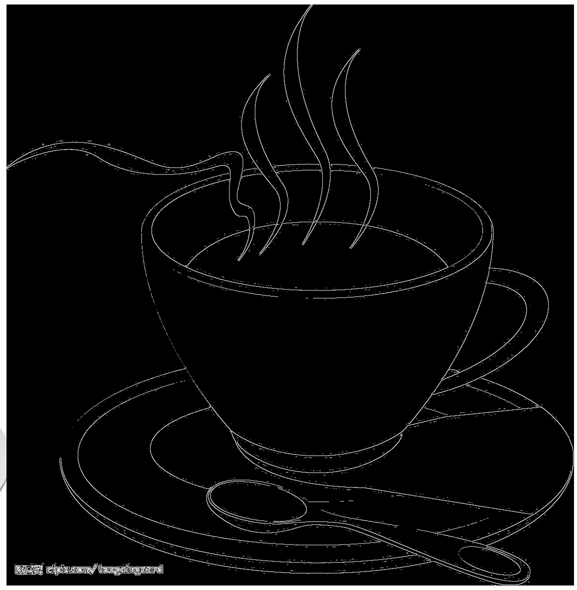
3.1 效果展示

3.1.1 Lenna threshold:0.015

Figure 3: Lenna Edge

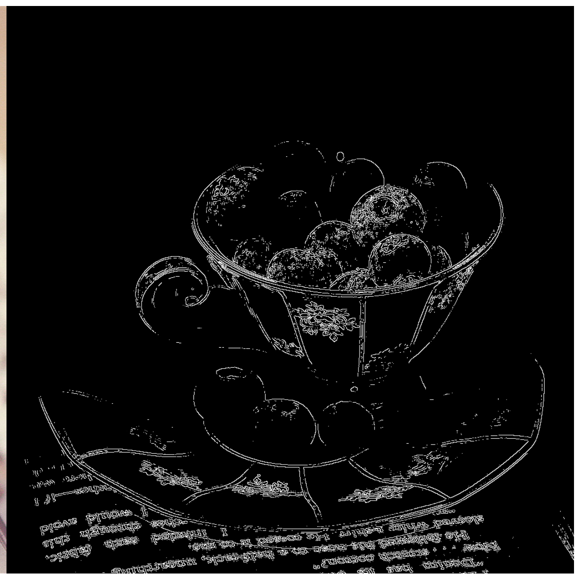
3.1.2 Cup threshold:0.5

图片来自于昵图网[4]

3.1.3 Blueberry and Cup threshold:8

图片来自于 http://www.weimeiba.com[5]

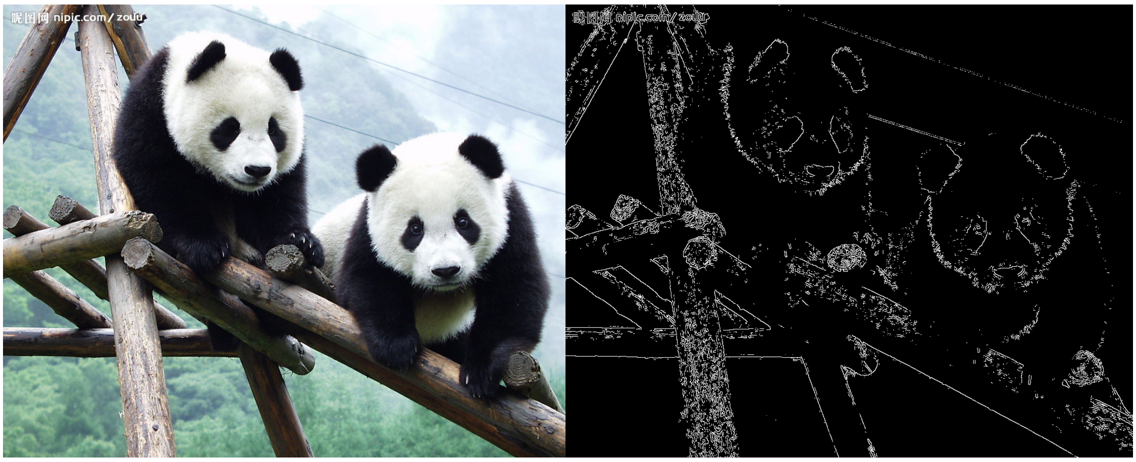



[4] http://pica.nipic.com/2007-11-26/200711262323153_2.jpg

[5] http://old.bz55.com/uploads/allimg/140903/138-140Z3093610.jpg

3.1.4 Pandas threshold:8

图片来自于互动百科、昵图网[6]



3.1.5 Teapot threshold:15

茶壶茶杯图片来自于昵图网[7]

[6] http://a4.att.hudong.com/63/06/16300000291746124581064816436.jpg

[7] http://pic20.nipic.com/20120427/3177520_175320712116_2.jpg