

阿洲的程式教學

關於Qt、OpenCV、影像處理演算法

找邊緣(Canny)

Sobel或Laplace都是基於微分的邊緣檢測算法，只有當雜訊極少，或者先用平滑濾波抑制雜訊後，才能得到理想結果。

在邊緣檢測中，抑制雜訊和邊緣精確定位很難同時滿足，當我們通過平滑濾波去除噪音的同時，也增加了邊緣定位的不確定性，而提高邊緣檢算子對邊緣敏感度的同時，也增加了對雜訊的敏感度，這邊介紹Canny邊緣檢測算子，Canny在抗雜訊和精確定位間有不錯的效果，OpenCV提供Canny()函式，來進行Canny邊緣檢測，以下為流程，OpenCV已將步驟封裝好，使用時只要呼叫Canny()即可。

1. 去雜訊：使用5×5的高斯濾波，我們根據下面的高斯函數，以 $\sigma=1.3$ 帶入後得到高斯濾波各個參數。

$$\text{高斯函數: } H_{ij} = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(i-k-1)^2 + (j-k-1)^2}{2\sigma^2}\right)$$

$$\text{高斯濾波: } \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

2. 記錄梯像素梯度方向和強度：以Sobel運算子計算水平和垂直梯度(G_x 、 G_y)，數學上為兩者平方後相加，得到梯度強度(G)，實際上由於運算效率上的考量，OpenCV預設將 G 設為 G_x 和 G_y 的絕對值相加， θ 為梯度方向，我們將其分類到0、45、90、135度之一，也就是假設此點我們計算的 θ 為0~22.5或157.5~180，便將此點 θ 分類為0。

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} \quad G = \sqrt{G_x^2 + G_y^2}$$

$$\theta = \arctan\left(\frac{G_y}{G_x}\right)$$

3. 非最大抑制：採用梯度找邊緣，邊緣會比較模糊，這方法讓邊緣定位較精確，我們比較每個像素，和他正負梯度方向的像素，要是這個像素的梯度強度最大，就保留此像素的值，否則設為0，也就是假設有某個像素的梯度方向為朝上或下，則和它上面及下面像素比較梯度強度，如果不是最大就設為0。
4. 判斷邊界：我們依據輸入的上下兩個閾值，通常上下閾值的比例，大約在2：1到3：1之間，由閾值判斷此像素是否為邊緣，分以下三個判斷依據：
 - a、要是此像素梯度強度大於上閾值，則此像素為邊緣。
 - b、如果此像素梯度強度小於下閾值，此像素不為邊緣。
 - c、如果此像素梯度強度介於上下閾值，如果此像素周圍，有像素的梯度強度大於上閾值，則此像素為邊緣，否則不為邊緣。

OpenCV canny

`void Canny(InputArray image, OutputArray edges, double threshold1, double threshold2, int apertureSize=3, bool L2gradient=false)`

- `src`：輸入圖，單通道8位元圖。
- `dst`：輸出圖，尺寸、型態和輸入圖相同。
- `threshold1`：第一個閾值。
- `threshold2`：第二個閾值。
- `apertureSize`：Sobel算子的核心大小。
- `L2gradient`：梯度大小的算法，預設為false。

L2gradient分兩種找梯度方式：

true 時梯度大小為 $\sqrt{(dI/dx)^2 + (dI/dy)^2}$

false 時梯度大小為 $|dI/dx| + |dI/dy|$

以下示範Canny0的使用，dst2為dst1黑白倒轉，改呈現比較習慣的黑色邊緣：

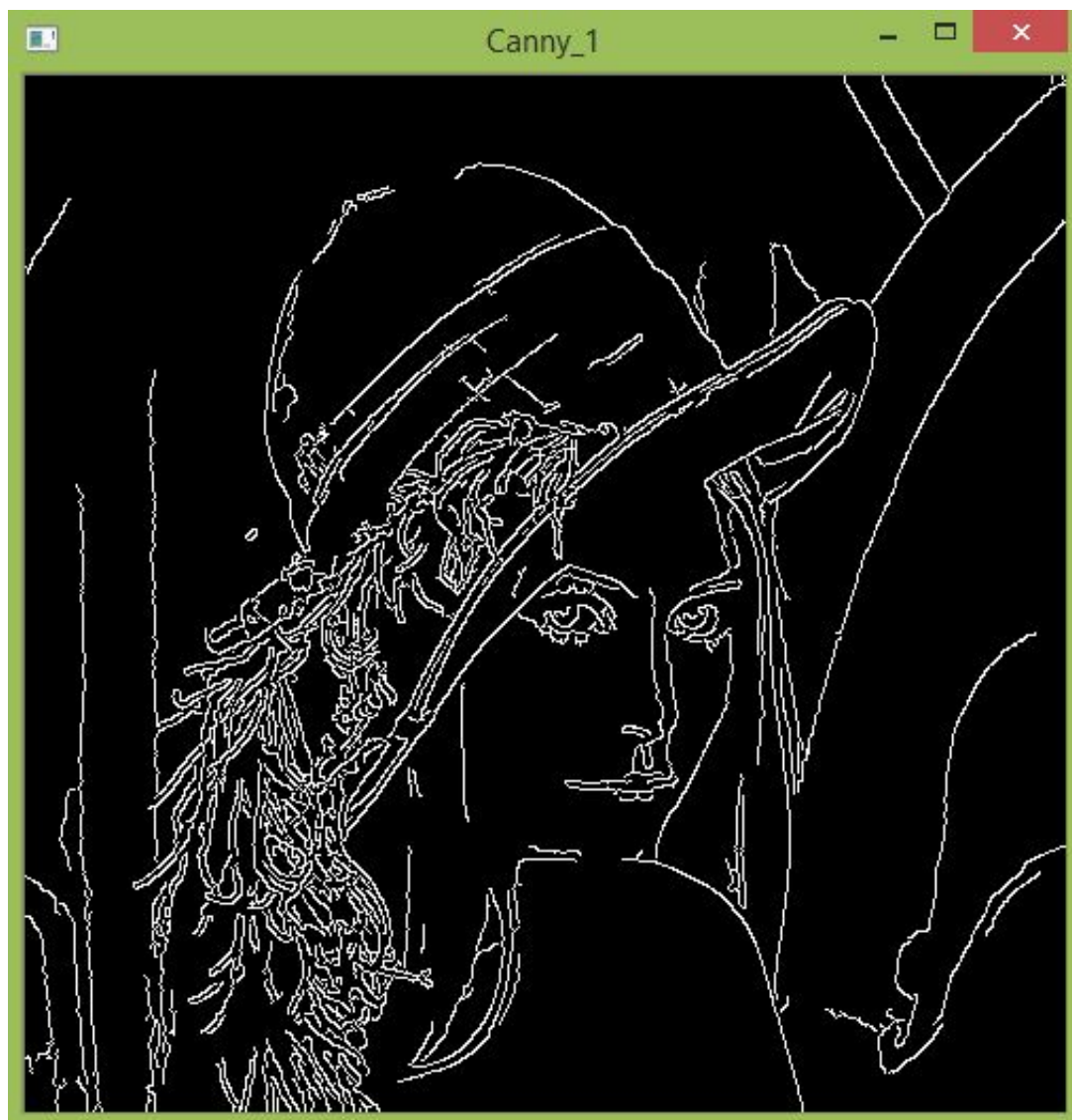
```
#include <stdio>
```

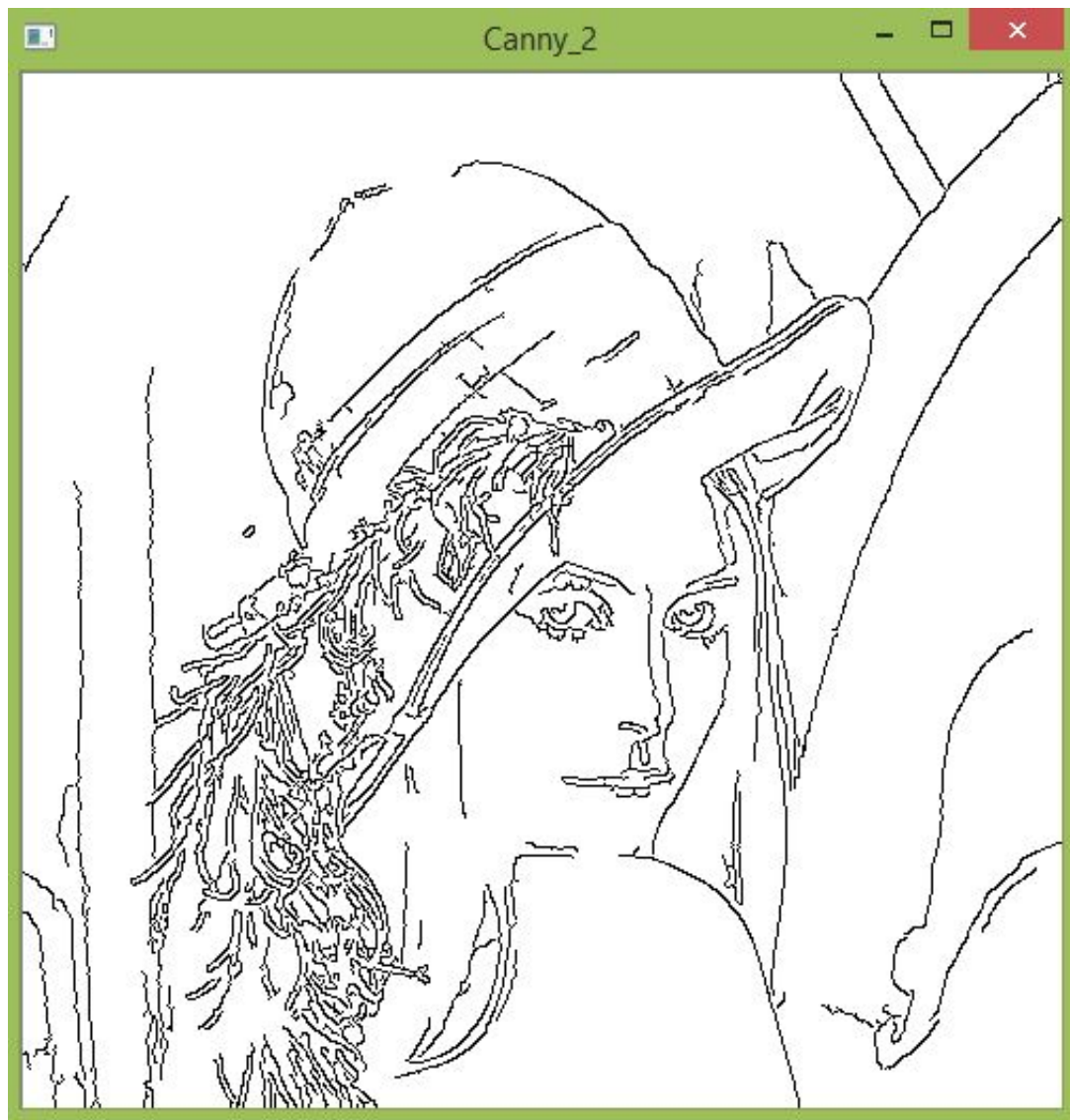
```
#include <opencv2/opencv.hpp>
using namespace cv;

int main(){
    Mat src = imread("lena.jpg", CV_LOAD_IMAGE_GRAYSCALE);
    GaussianBlur(src, src, Size(3,3), 0, 0);
    Mat dst1, dst2;
    Canny(src, dst1, 50, 150, 3);
    threshold(dst1, dst2, 128, 255, THRESH_BINARY_INV); //反轉影像，讓
    邊緣呈現黑線
    imshow("origin", src);
    imshow("Canny_1", dst1);
    imshow("Canny_2", dst2);
    waitKey(0);

    return 0;
}
```







[回到首頁](#)

[回到OpenCV教學](#)

參考資料：

[OpenCV 教程](#)

📅 2015-11-30 👤 阿宅 📁 OpenCV, 邊緣 🔖 Canny

0 Comments

猴子遇到0與1! 程式學習筆記

1 Login ▾

♥ Recommend

🔗 Share

Sort by Best ▾



Start the discussion...

Be the first to comment.

ALSO ON 猴子遇到0與1! 程式學習筆記

Qt主窗口(Top Level Window)

1 comment • 6 months ago

mike — 喔喔

文件對話框(QFileDialog)

1 comment • 6 months ago

楊政穎 — dialog.cpp 裡面的 QString s
=
QFileDialog::getOpenFileName(this, tr

✉ Subscribe

D Add Disqus to your site Add Disqus Add

🔒 Privacy

自豪的採用 WordPress