

阿洲的程式教學

關於Qt、OpenCV、影像處理演算法

霍夫找圓(HoughCircles)

我們用和霍夫直線偵測同樣的概念，進行霍夫圓形偵測，圓方程式為 $(x-a)^2 + (y-b)^2 = r^2$ ，其中 (a, b) 為圓心座標， r 為圓的半徑，用這個三維數據組，讓 (a, b) 在影像座標內不斷改變位置，找出所有可能的半徑 r ，最後當這三維數據組的點數，超過我們定的閾值時就判斷為圓。

因為傳統的霍夫圓偵測是三維空間上的計數，基於效率上的考量，而且維度變多，精確定位局部峰值變得困難，OpenCV的霍夫圓偵測使用以下兩個步驟：

1. 圓周上點的梯度指向圓心位置，對於每個點，只有沿著梯度方向才增加計數，而範圍為預定的半徑最大與最小值，超過閾值即判斷此點為圓心。
2. 對圓心和點的距離進行計數，最大值就是此圓的半徑。

OpenCV 偵測圓

```
void HoughCircles(InputArray image, OutputArray circles, int method, double dp, double minDist, double param1=100, double param2=100, int minRadius=0, int maxRadius=0)
```

- **image**：輸入圖，8位元單通道圖。
- **circles**：以`vector< Vec3f >`記錄所有圓的資訊，每個`Vec3f`紀錄一個圓的資訊，包含3個浮點數資料，分別表示 x 、 y 、 $radius$ 。
- **method**：偵測圓的方法，目前只能使用`CV_HOUGH_GRADIENT`。
- **dp**：偵測解析度倒數比例，假設 $dp=1$ ，偵測圖和輸入圖尺寸相同，假設 $dp=2$ ，偵測圖長和寬皆為輸入圖的一半。
- **minDist**：圓彼此間的最短距離，太小的話可能會把鄰近的幾個圓視為一個，太大的話可能會錯過某些圓。
- **param1**：圓偵測內部會呼叫`Canny()`尋找邊界，**param1**就是`Canny()`的高閾值，低閾值自動設為此值的一半。

- **param2**：計數閾值，超過此值的圓才會存入**circles**。
- **minRadius**：最小的圓半徑。
- **maxRadius**：最大的圓半徑。

以下我們示範如何**HoughCircles()**找影像中的圓，並用自行撰寫的**drawCircle()**將找到的圓畫出：

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

void calcCircles(const Mat &input, vector<Vec3f> &circles);
void drawCircle(Mat &input, const vector<Vec3f> &circles);

int main(){
    Mat img = imread("input.jpg",CV_LOAD_IMAGE_GRAYSCALE);
    Mat result = imread("input.jpg",CV_LOAD_IMAGE_COLOR);

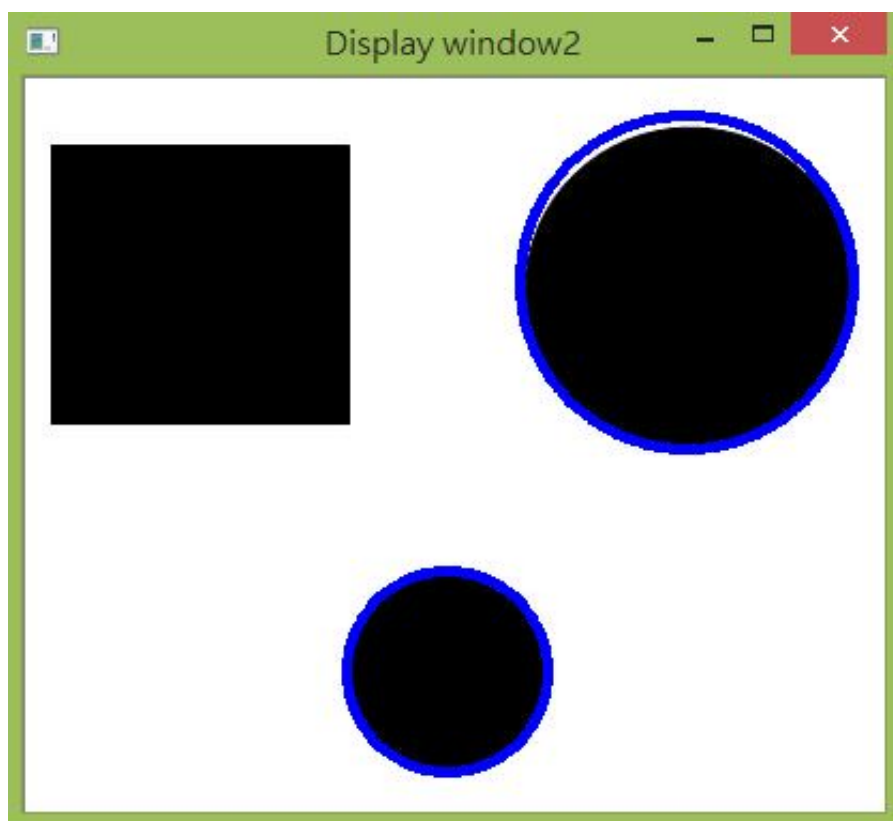
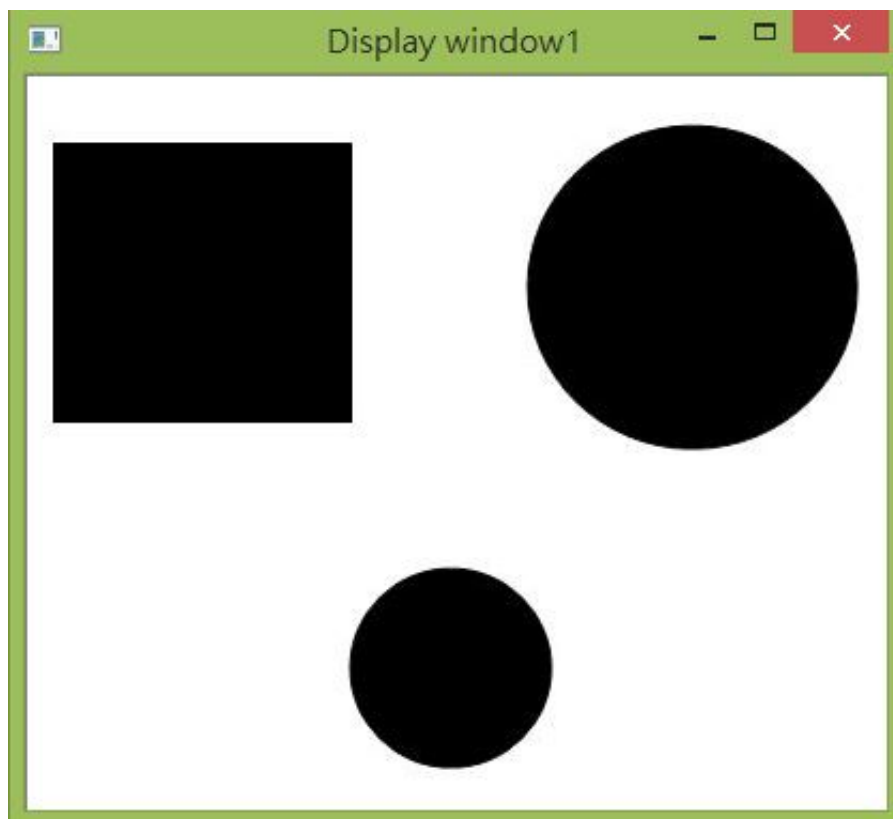
    vector<Vec3f> circles;
    calcCircles(img, circles);
    drawCircle(result, circles);

    namedWindow("Display window1", WINDOW_AUTOSIZE);
    namedWindow("Display window2", WINDOW_AUTOSIZE);
    imshow("Display window1", img);
    imshow("Display window2", result);
    waitKey(0);

    return 0;
}

void calcCircles(const Mat &input, vector<Vec3f> &circles){
    Mat contours;
    Canny(input,contours,50,150);
    HoughCircles(contours, circles, CV_HOUGH_GRADIENT, 2, 50, 200, 100);
}

void drawCircle(Mat &input, const vector<Vec3f> &circles){
    for(int i=0; i<circles.size(); i++){
        Point center(cvRound(circles[i][0]), cvRound(circles[i][1]));
        int radius = cvRound(circles[i][2]);
        circle(input, center, radius, Scalar(255,0,0), 3, 8, 0 );
    }
}
```



📅 2015-11-30 👤 阿宅 📁 OpenCV, 影像分割 🔧 HoughCircles

0 Comments

猴子遇到0與1! 程式學習筆記

 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

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

Qt主窗口(Top Level Window)

1 comment • 6 months ago

mike — 喔喔

 Subscribe Add Disqus to your site Add Disqus Add Privacy

自豪的採用 WordPress