

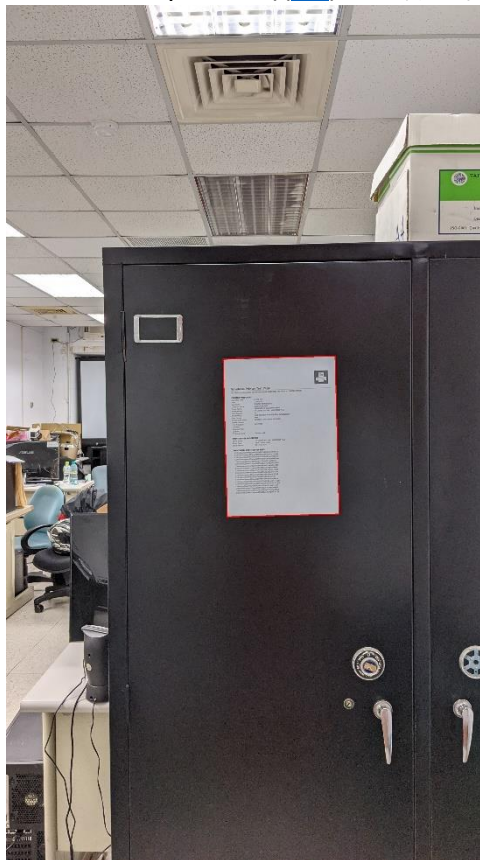
主旨：

影像測距

說明：

影像測距是利用拍到的照片，推測出物品大約的距離。一般是使用兩顆鏡頭，通過一系列的運算後，計算出物品與鏡頭的距離，中間牽扯不少攝影相關的知識包含鏡頭視角，焦距等，本次作業簡化算法，利用固定長度的物品，去推算其距離。

將圖片讀入後，將原圖轉成灰階圖後使用高斯模糊降噪，利用 Canny 尋找邊緣，再利用 OpenCV 內建的尋找輪廓(findcontours)，並使用(contourArea)找到面積最大的輪廓，再用函式(approxPolyDP)來逼近邊緣成為類四邊形，最後使用(minAreaRect)找出最佳邊框(圖一)，求出的點可以畫出矩形，並使用第一點跟第四點的距離(boxpoint[0]跟 boxpoint[3])([ref](#))當作像素寬度。



(圖一)

請使用 data.jpg 求出焦距參數後，套用至所有的測資檢測。

焦距參數的方法為：

(像素寬度*已知距離)/ 已知寬度

已知寬度 = 210.0

已知距離 = 1500.0

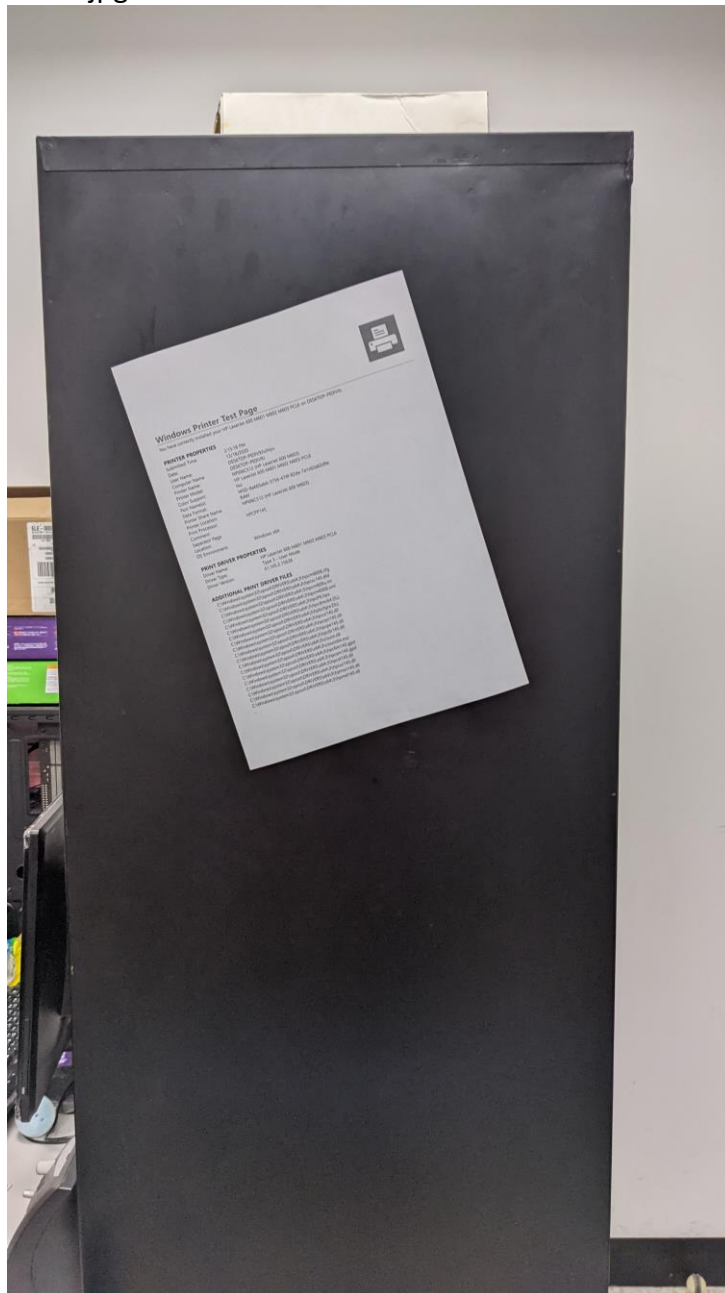
計算距離的方法為：

(已知寬度*焦距參數)/像素寬度

Example:

Input:

case1.jpg



Output:

822.299

截止時間：

2020.01.19 23:59

繳交方式：

批改系統、Portal

作業系統：

Ubuntu 16.04

程式語言：

C or C++ (gcc version 9.3.0)

Command：

./hw4.exe 1.jpg

Provide data :

mailto: tinin@saturn.yzu.edu.tw

title:[LAHW4] SID 測資提供

注意事項：

1. cv::RotatedRect Class Reference
https://docs.opencv.org/3.4/db/dd6/classcv_1_1RotatedRect.html
2. GaussianBlur(matrix, matrix, Size(5, 5), 0);
3. Canny(matrix, out_ matrix, 35, 125);
4. findContours(matrix, contours, CV_RETR_LIST,
CV_CHAIN_APPROX_SIMPLE);
5. approxPolyDP(contours, contours_poly, 3, true);
6. 傳值方式

```
int main(int argc, char* argv[])
{
    ifstream fin;
    fin.open(argv[1]);
}
```
7. 取到小數後 4 位，接受 0.0001 誤差
8. 不要有 system("pause");
9. 不接受遲交