



CPS Mark Detector

2017/11/01 Phisten



Outline

Introduction

Example

Setup

Calibration

Method



Introduction

CPS Mark Detector

- 可辨識50種Mark
- 定位距離約20cm~200cm
- 可辨識Mark的角度
- 以OpenCV3.2.0實作

CPS Mark Detector Demo Project

Show Mark Info

Test OpenCV2.4.10

CPS Mark Detector

OpenCV 3.2.0

OpenCV 2.4.10



Example - 初始化圖卡檢測執行緒

```
#include <thread>
```

```
#include "CPS_Mark.h"
```

```
const int CameraIndex = 0; //檢測圖卡使用的相機索引
```

```
CPS_Mark markDetector0; //圖卡檢測類別
```

```
//建立執行緒並初始化攝影機開始檢測圖卡
```

```
thread CPS_0(&CPS_Mark::DetectCardProcess, &markDetector0, CameraIndex);
```



Example - 取得圖卡資訊

```
//判斷圖卡檢測程式是否執行中
if( markDetector0.IsRunning() )
{
    //取得圖卡資訊
    CPS_Mark::MarkInfo mark0( markDetector0.GetMarkInfo() );
    //若有抓到圖卡(Id > 0) 則輸出其 編號,距離,角度,中心點XY座標
    if (mark0.Id > 0)
    {
        printf("ID=%d, Dist=%d Angle=%d Center:(%d,%d)\n",
            mark0.Id, mark0.Distance, mark0.Angle, mark0.CenterX, mark0.CenterY);
    }
}
```



Setup

本程式提供Visual Studio 2013的範例程式, 示範如何引用本程式的DLL取得圖卡資訊
範例程式內包括引用本程式DLL的Demo專案 與 本程式原始碼兩個專案

Demo專案內有使用OpenCV 2.4.10 x86,僅為測試輸出與示範OpenCV多版本整合,與本程式無關
本程式圖卡識別專案使用OpenCV 3.2.0 x86實作
所有必要檔案均已附夾在範例程式資料夾內,不須額外安裝OpenCV 3.2.0 x86

1.請在您的專案設定內新增本程式原始碼之路徑

[C/C++ > 一般 > 其他include目錄]

ex: C:\dev\project\CPS\CPS_MarkLib\CPS_Mark

2.將範例程式資料夾內 (CPS\CPS_MarkLib\Debug) 下述的DLL複製到您的專案執行檔目錄內 主要DLL (CPS_Mark.dll)與OpenCV 3.2.0 x86的DLL(opencv_*.dll)

若需自行編譯原始碼可依照 CPS\Path.txt 文件進行設定



Calibration

由於不同相機的影像變形情況不同以及圖卡實際列印的大小改變會影響距離計算
本系統架設時須先依相機拍攝圖卡的影像校正程式內的參數
校正時相機需正攝圖卡圖卡與相機的距離建議在40~100公分的範圍內

下列兩項為需要校正的數值：

- `double baseDist1 = 45;`
此參數為圖卡與相機實際的距離（單位可自訂,此例為公分）
- `double basePixelHeight1 = 197; //10.9cm mark`
此參數為圖卡被相機拍攝在影像中的高（單位為像素）

Method - 相機與圖卡距離計算

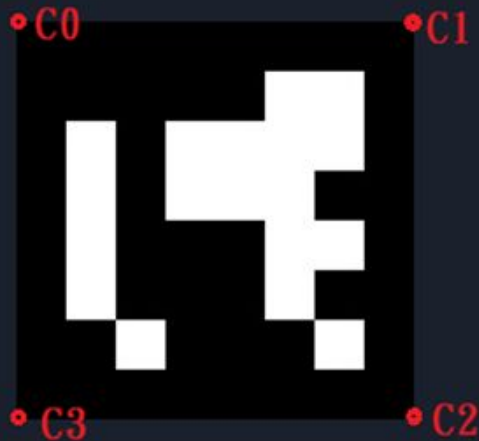
依據右圖標示的四點(C0~C3)計算圖卡縮放比
進行相機與圖卡現實距離的計算

$$\text{Height} = (\text{C2.Y} + \text{C3.Y} - \text{C0.Y} - \text{C1.Y}) / 4$$

BaseDistance = 校正時圖卡距離

BaseHeight = 校正時圖卡像素高度

$$\text{Distance} = \text{BaseDistance} * \text{BaseHeight} / \text{Height}$$



Method - 影像Y軸旋轉角度計算

依據右圖標示的四點(C0~C3)
進行相機視線與圖卡法線夾角的計算
夾角在45度以內時精度較好

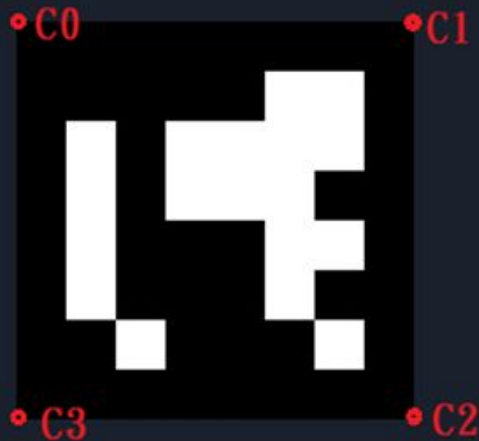
$$\text{Width} = (C1.X + C2.X - C3.X - C4.X) / 4$$

$$\text{Height} = (C2.Y + C3.Y - C0.Y - C1.Y) / 4$$

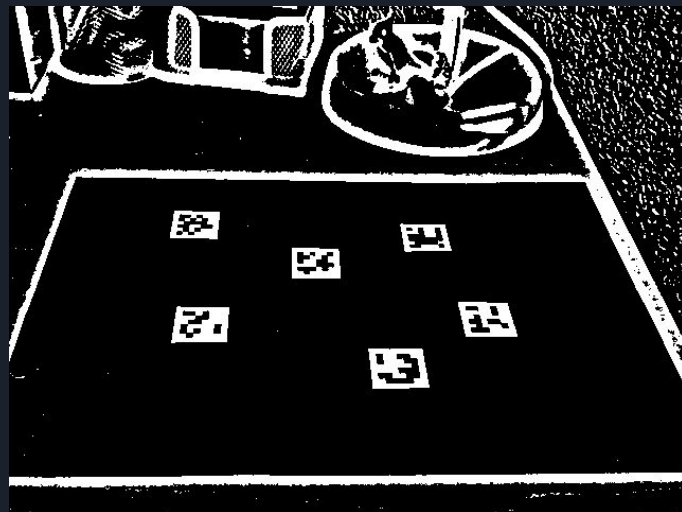
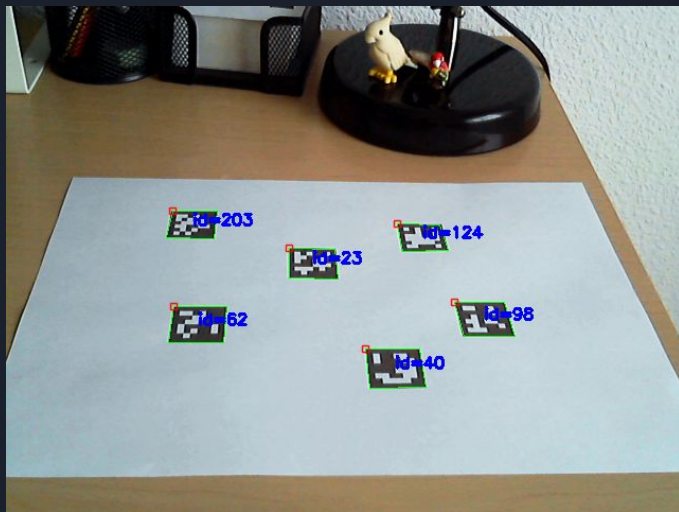
$$\text{Angle} = \arccos(\text{Width} / \text{Height})$$

圖卡向右旋轉時輸出角度為負向左時為正
if (leftBoard > rightBoard)

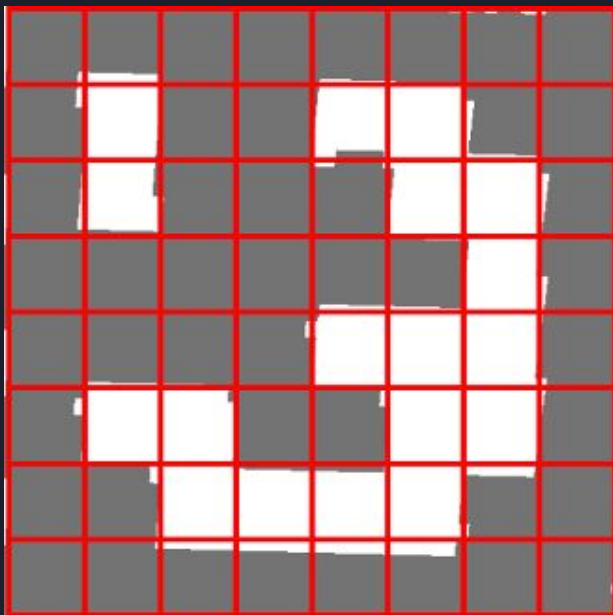
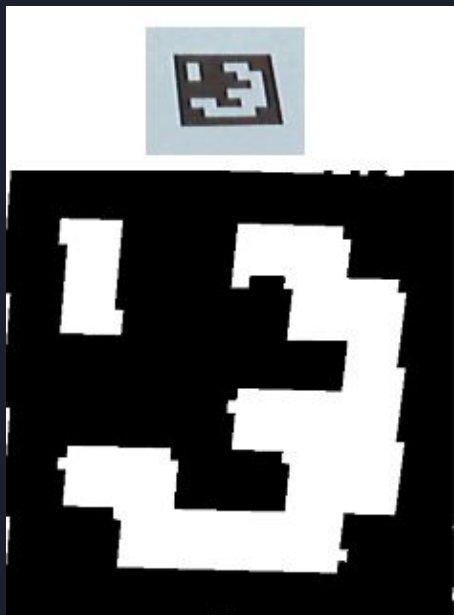
Angle *= -1



Method - 圖卡識別原理



Method - 圖卡識別原理



Method - 圖卡識別原理

