程，具有一定的參考價值，感興趣的小夥伴們可以參考一下

相機標定：簡單的說，就是獲得相機引數的過程。引數如：相機內參數矩陣，投影矩陣，旋轉矩陣和平移矩陣等

什麼叫相機引數？

簡單的說，將現實世界中的人、物，拍成一張影象（二維）。人或物在世界中的三維座標，和影象上對應的二維座標間的關係。表達兩種不同維度座標間的關係用啥表示？用相機引數。

相機的成像原理

先來看一下，相機的成像原理：

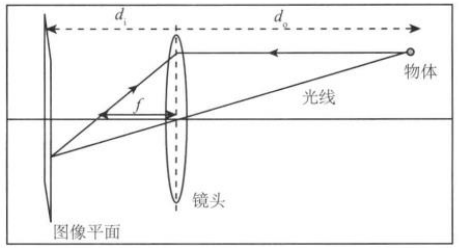
聊一聊OpenCV相

軟體程式設計 · 發表 2018-01-10

NT

馬_

這篇文章主要為大家詳細介紹了OpenCV相機標定的相關資料，即獲得相機引數的過



如圖所示，這是一個相機模型。將物體簡化看成一個點。來自物體的光，通過鏡頭，擊中影象平面（影象感測器），以此成像。d0是物體到鏡頭的距離，di時鏡頭到影象平面的距離，f是鏡頭的焦距。三者滿足以下關係。

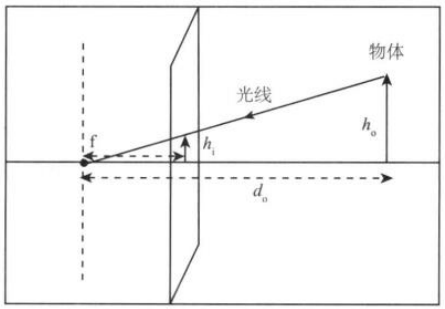
$$\frac{1}{f} = \frac{1}{d_0} + \frac{1}{d_i}$$

現在，簡化上面的相機模型。

將相機孔徑看成無窮小，只考慮中心位置的射線，這樣就忽視了透鏡的影響。然後由

於d0遠遠大於di，將影象平面放在焦距處，這樣物體在影象平面上成像為倒立的影像（沒有透鏡的影響，只考慮從中心的孔徑進入的光線）。這個簡化的模型就是針孔攝像機模型。然後，我們在鏡頭前，將影象平面放在焦距距離的位置，就可以簡單獲得一個筆直的影象（不倒立）。當然，這只是理論上的，你不可能將影象感測器從相機裡拿出來，放在鏡頭前面。實際應用中，針孔攝像機應該是將成像後的影象倒過來，以獲得正立的影象。

到此，我們獲得了一個簡化的模型，如下圖：



h0是物體的高，hi是影象上物體的高，f是焦距（距離），d0是影象到鏡頭的距離。四者滿足如下關係：

$$h_i = f \frac{h_0}{d_0}$$

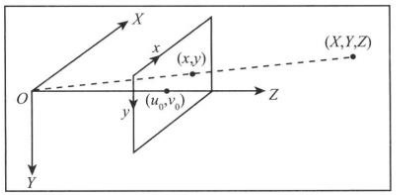
（1）

物體在影象中的高度hi，和d0成反比。也就是說，離鏡頭越遠，物體在影象中越小，離得越近越大（好吧，這句話是廢話）。

但通過這個式子，我們便能夠預測三維中的物體，在影象（二維）中的位置。那麼怎麼預測？

相機標定

如下圖所示，根據上面簡化的模型，考慮三維世界中的一個點，和其在影象（二維）中的座標關係。



（X，Y，Z）為點的三維座標，（x，y）為其通過相機成像後在影象（二維）上的座標。u0和v0是相機的中心點（主點)，該點位於影象平面中心（理論上是這樣。但實際的相機會有幾個畫素的偏差）現在只考慮y方向上，由於需要將三維世界中的座標，轉換為影象上的畫素（影象上的座標，實際上是畫素的位置），需要求y方向上焦距等於多少個畫素（用畫素值表示焦距），Py表示畫素的高，焦距f（米或毫米）。垂直畫素表示的焦距為

$$f_y=f/p_y$$

根據式子（1），只考慮y方向。我們三維世界中得點，在影象（二維）中y的座標。

$$y = \frac{f_y Y}{Z} + v_0$$

同理，得到x的座標。

$$x = \frac{f_X X}{Z} + u_0$$

現在，將上圖中的座標系的原點O，移動到影象的左上角。由於（x，y）是關於（u0，v0）的偏移，上面表示影象（二維）中點的座標的式子不變。將式子以矩陣的形式重寫，得。

$$s\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

其中，等式左邊的第一個矩陣，叫做“相機內參數矩陣”，第二個矩陣叫（投影矩陣）。

更為一般的情況，開始時的參考座標系不位於主點（中心點），需要額外兩個引數“旋轉向量”和“平移向量”來表示這個式子，這兩個引數在不同視角中是不一樣的。整合後，上述式子重寫為。

$$s\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_1 & r_2 & r_3 & t_1 \\ r_4 & r_5 & r_6 & t_2 \\ r_7 & r_8 & r_9 & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

校正畸變

通過相機標定，獲得了相機引數後，可以計算兩個對映函式（x座標和y座標），它們分別給出了沒有畸變的影象座標。將畸變的影象重新對映成為沒有畸變的影象。

程式碼：

做相機標定時，一般用標定板(棋盤）拍攝一組影象，利用這些影象提取角點，通過角點在影象中得座標和三維世界中的座標（通常自定義3維座標），計算相機引數。

```
std::vector<cv::P
//提取標定影象角點，
 cv::findChessboar
 boardSize, //角點
 imageConers);
```

函式calibrateCamera完成相機標定工作。

```
cv::calibrateCame
 imagePoints, //二
 imageSize,//影象大
 camerMatirx,//相機
 disCoeffs,//投影矩
 rvecs, //旋轉
 tvecs,//平移
flag //標記opencv找
);
```

計算畸變引數，去畸變

```
//計算畸變引數
cv::initUndistort
  cv::Mat(), cv:
```

```
    map1, //x對映函
    map2 //y對映函式
    );
  //應用對映函式
  cv::remap(image,
  undistorted, //去
  map1, map2, cv::I
```

現在整合程式碼。

示例：

標頭.h

```
#include<opencv
#include<opencv2
#include<opencv2
#include<opencv2
#include <opencv
#include<string>
#include<vector>
class CameraCal
{
private:
  //世界座標
  std::vector <
  //影象座標
  std::vector <s
  //輸出矩陣
  cv::Mat camerM
  cv::Mat disCoe
  //標記
  int flag;
  //去畸變引數
  cv::Mat map1,
```

```
    //是否去畸變
    bool mustInitU

    ///儲存點資料
    void addPoints
    {
      imagePoints
      objectPoints
    }
  public:
    CameraCalibrat
    //開啟棋盤圖片
    int addChessbo
    {
      std::vector<
      std::vector<
      //輸入角點的
      for (int i =
      {
        for (int j
        {
          objectCo
        }
      }
      //計算角點在影
      cv::Mat imag
      int success
      for (int i =
      {
        image = cv
        //找到角點
        bool found
        cv::corner
          imageCor
          cv::Size
          cv::Size
          cv::Term
```

```
          30, 0.1
        if (image
        {
          addPoint
          success
        }
        //畫出角點
        cv::drawCh
        cv::imshow
        cv::waitKe
      }
      return succe
    }

    //相機標定
    double calibra
    {
      mustInitUnd
      std::vector<
      //相機標定
      return cv::c
        camerMatri
    }
    ///去畸變
    cv::Mat remap(
    {
      cv::Mat undi
      if (mustInit
      {
        //計算畸變
        cv::initUn
          cv::Mat(
        mustInitUn
      }
      //應用對映函
      cv::remap(in
      return undis
```

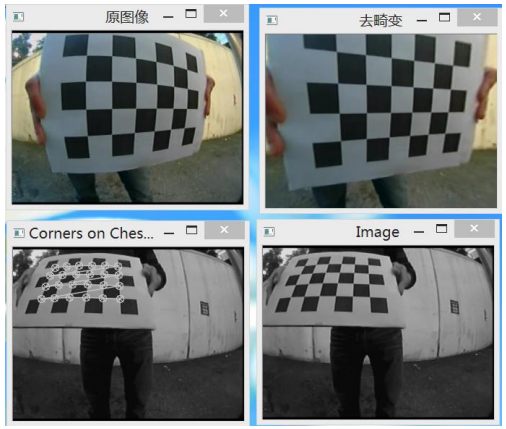
```
    }
    //常成員函式，獲
    cv::Mat getCam
    cv::Mat getDis
};
```

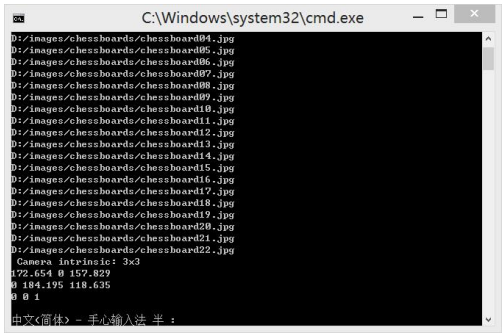
源.cpp

```
#include"標頭.h"
#include<iomanip
#include<iostrea
int main()
{
  CameraCalibrat
  cv::Mat image;
  std::vector<st
  cv::namedWindo
  for (int i = 1
  {
    ///讀取圖片
    std::strings
    s << "D:/ima
    std::cout <<

    filelist.pus
    image = cv::
    cv::imshow("
    cv::waitKey(
  }
  //相機標定
  cv::Size board
  Cc.addChessboa
  Cc.calibrate(i

  //去畸變
  image = cv::im
```

```
29      cv::Mat uImage=
30      cv::imshow("原
31      cv::imshow("去
32      //顯示相機內參數
33      cv::Mat camera
34      std::cout << "
35      std::cout << ca
36      std::cout << ca
37      std::cout << ca
38
39      cv::waitKey(0)
40  }
```

實驗結果：





標籤： 軟體程式設計 C語言

您可能也會喜歡...

Python使用OpenCV進行標定

Android實現呼叫系統相簿與相機設定頭像並儲存在本地及伺服器

Android中通過訪問本地相簿或者相機設定使用者頭像例項

Android 相機相簿許可權設定方法

Python+樹莓派+YOLO打造一款人工智慧照相機

android 7自定義相機預覽及拍照功能

Android 用 camera2 API 自定義相機

Android 自定義相機及分析原始碼

Spring Boot實戰之netty-socketio實現簡單聊天室(給指定使用者推送訊息)

Android自定義照相機的例項

iOS開發-自定義相機例項(仿微信)

Android中關於自定義相機預覽介面拉伸問題

Android自定義相機實現定時拍照功能

Android使用系統自帶的相機實現一鍵拍照功能
Android自定義相機實現自動對焦和手動對焦
Android自定義相機介面的實現程式碼
Android自定義照相機Camera出現黑屏的解決方法
Android實用控制元件自定義逼真相機光圈
Android自定義照相機詳解
View
Android實現從本地相簿/相機拍照後裁剪圖片並設定頭像

首頁
Python教學

看以看到，相機內參數矩陣為

172.654 、0、157.829
0、184.195、118.635
0 、0 、1

以上就是本文的全部內容，希望對大家的學習有所幫助，也希望大家多多支援itread01.com。