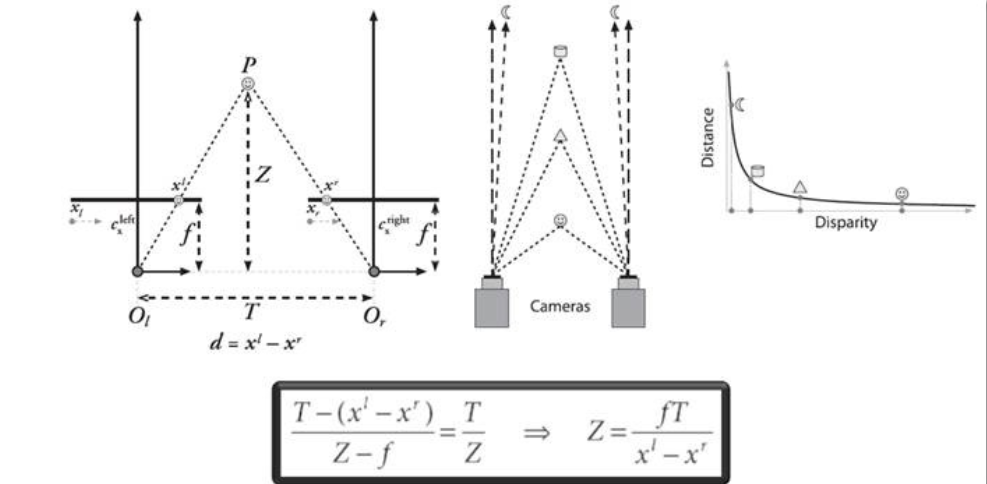
# 基于Matlab和OpenCV的双目测距

摘要：双目测距的原理是利用左右两个摄像机拍摄同一物体形成的视差来确定物体距摄像机的距离。这其中需要通过标定得出的参数包括内参（焦距fc, 主点Principal point, 径向畸变Radial Distortion, 切向畸变Tangential Distortion、 歪曲畸变skew）和外参（都是以Camera2相对于Camera1的描述，包括旋转矩阵Rotation 、平移矩阵Translation、基础矩阵Fundamental、特征矩阵Essential）。标定完成之后，先使用标定得到的畸变参数构建畸变矩阵并使用cv::stereoRectify()进行立体校正，最后再用立体匹配，获得图像视差图。整个过程最大的问题在于如何有效校正误差和提高效率上。

## 一、双目测距概述



该公式中f、T通过标定过得，xl-xr通过立体匹配获得。由于该公式忽略了实际测量中可能产生的一切误差，包括正向畸变、切向畸变、图形畸变，所以实际操作中，在标定结束了还需要做立体匹配，使两相机的极线真正平行，减少误差。

## 二、关于器材

1. 为了保证双目相机左右镜头帧率同步，不建议购买单独的网络摄像机。应购买使用同一主板上的两镜头，比如<https://item.taobao.com/item.htm?spm=a1z09.2.0.0.lO4pVl&id=526168710243&_u=g1qsbvok3189>
2. 使用90度及以下的无畸变镜头，可以使skew误差降低至0。



1. 该镜头不能使用普通的单接口usb数据线连接，否则在Opencv下会导致镜头无法正常打开。应使用类似于移动硬盘数据线的数据线连接。
2. 棋盘纸

下载地址：<http://docs.opencv.org/2.4/_downloads/pattern.png>

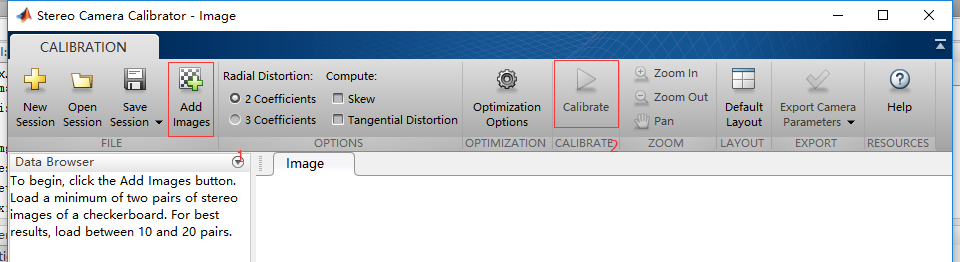
注意A4打印时由于打印机会缩放，所以需要自己再实际测量一遍格子之间的间隔。

## 三、相机标定

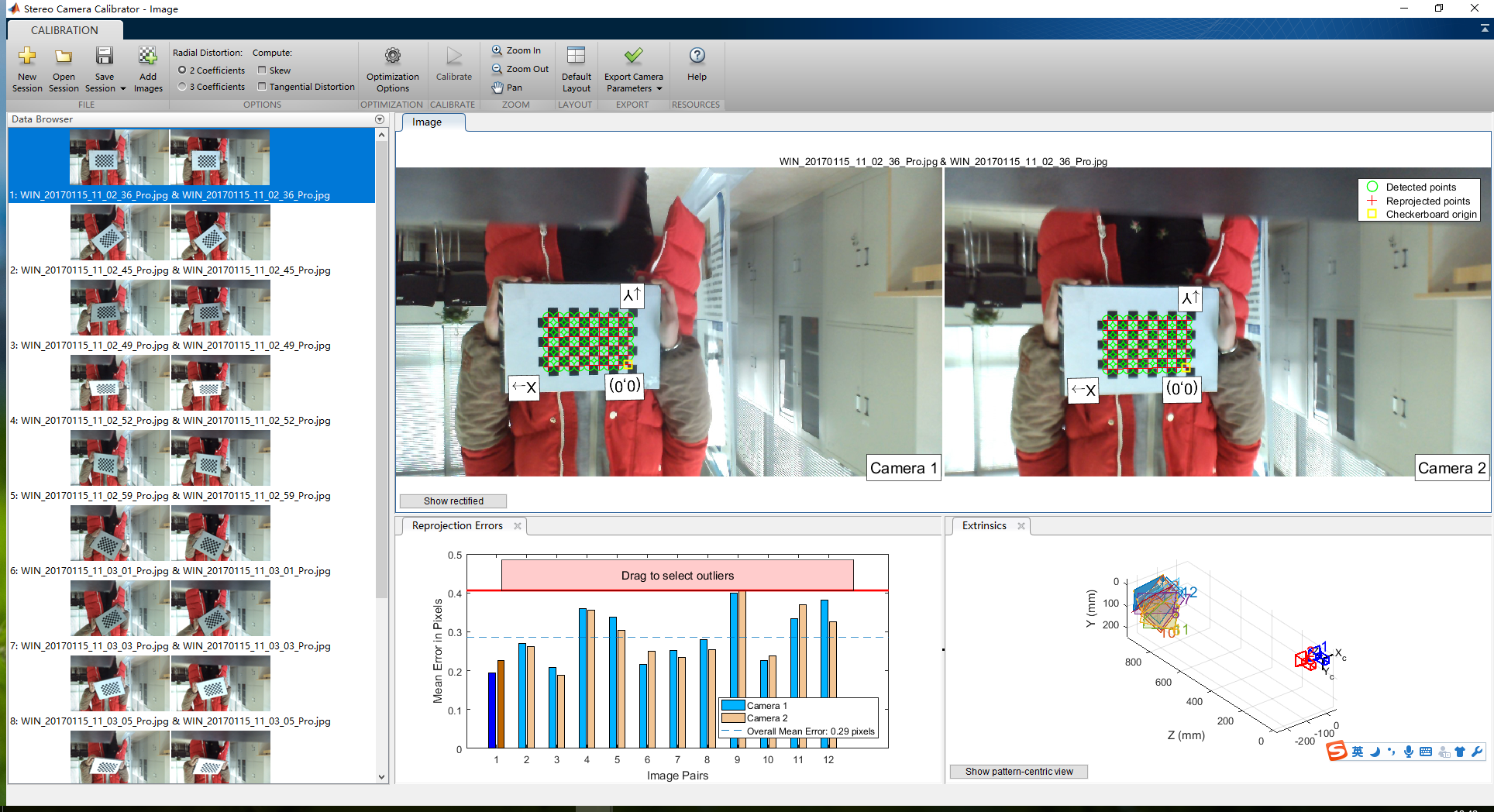
### 3.1基于Matlab的标定和参数解读

#### 3.11 Matlab标定步骤

Matlab（2016a）下的标定比较简单。依次选择 应用程序->Stereo Camera Calibrator ->Add Image->Calibrate(如下图)。

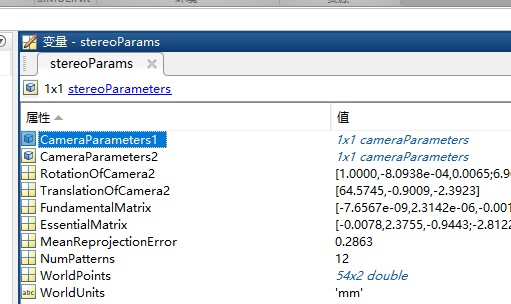


标定结果如下：



再点击Export Camera Parameters，将所有数据导出到Matlab工作区间。

图片如下：



#### 3.12 参数解读

此处只解读对本项目有意义的参数，关于所有参数的解释可以参考官方文档：

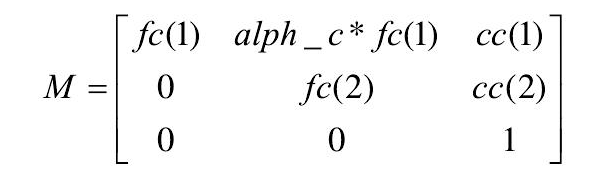
<http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/parameters.html>

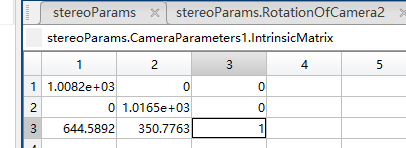
我们需要所有的误差参数来做立体校正

1. 内参



内参矩阵：



M与Matlab中的IntrinsicMatrix为转置关系，如图：

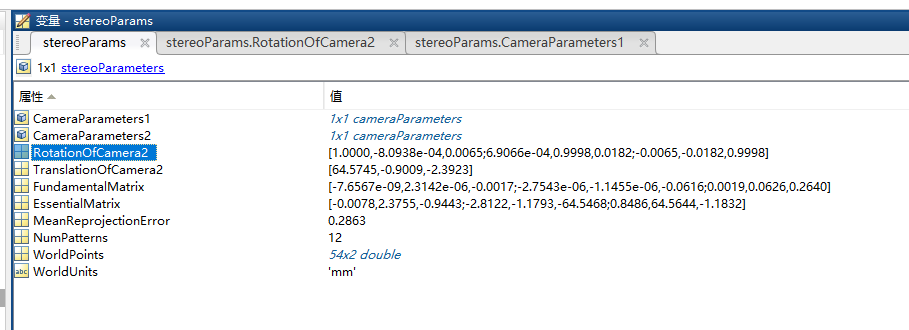
其中fc(1)为FocalLength[0](图中为1.0082e+03),fc(2) FocalLength[1](图中为1.10165e+03), alph\_c为skew值, cc为Principal point(主点)。

畸变系数矩阵：

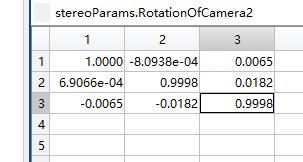
径向畸变3个（k1,k2,k3）和切向畸变2个(p1,p2)

distCoeffs向量 = [RadialDistortion[0], RadialDistortion[1], TangentialDistoration[0], TangentiaDistoration[1], 0]

1. 外参



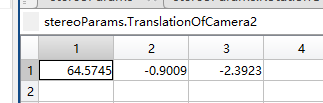
**旋转矩阵（RotationOfCamera2）:**



旋转矩阵在理想状态下就是单位矩阵，若不是，则表明有误差，需要进行极线校正。

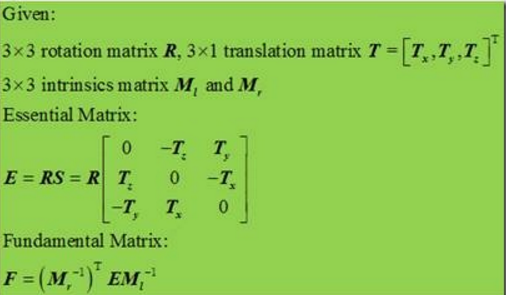
**平移矩阵（TranslationOfCamera2）:**

相机2在相机1的坐标系中的位置，理想状态下应该是（x,0,0）的形式，若y，z不为0，则两相机极线不平行，需要极线校正。



**特征矩阵（EssentialMatrix）和基础矩阵（FundamentalMatrix）**

特征矩阵包含了两个摄像头相对位置关系，基础矩阵包含了两个摄像头相对位置关系和各自内参信息。它们的解法如下：



实现原理可以参考以下博客：

<http://blog.csdn.net/chenyusiyuan/article/details/5963256>

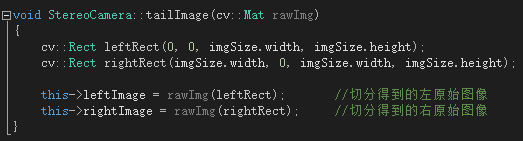
#### 3.13 注意事项

1. 由于双目相机拍摄到的照片已经将左右照片合在一起了，但是Matlab只能识别单一的左照片或右照片。因此需要做裁剪，并且裁剪得到的左右图片必须大小完全一致。建议使用Adobe Lightroom做批量裁剪（此教程自行百度），也可以使用Opencv做裁剪（后面叙述）。
2. 若在Reprojection Errors窗口中发现某一列的误差值太大，应删除该图片重新标定，但为保证精确，总图片不应少于10张。
3. 不建议使用另一个Matlab的标定工具箱，Calib\_gui（calib），因为我在使用中总是出错，错误原因未知。

### 3.2基于OpenCV的双目标定

#### 3.21 裁剪图像

因为我们使用的相机将左右图片都合成一张了，所以需要在标定前将其裁剪，裁剪代码如下：



其中rawImg是读取的图片，imgSize是类型为Size的变量，比如：

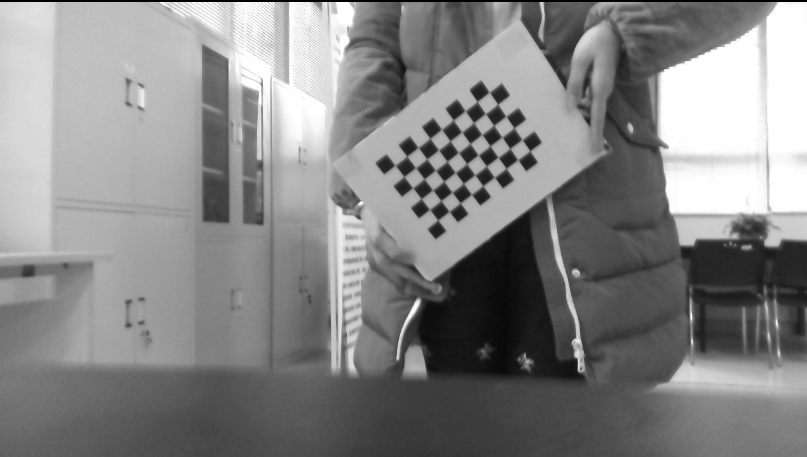
Size imgSize(1280, 720);

#### 3.22 图像预处理

此步骤可以省略，图像预处理的目的是使得白色棋盘格及其边缘在整个图像上更为突出。代码如下



预处理前：



预处理后：

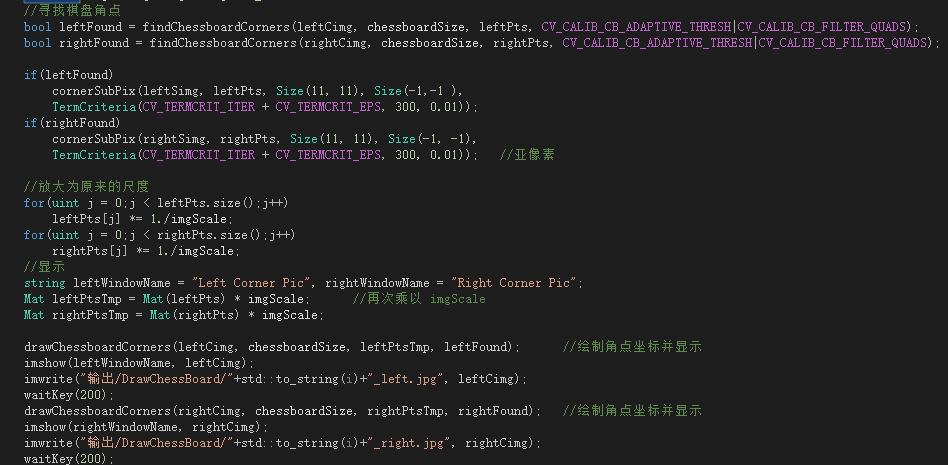


关于预处理前后的误差大小有待进一步讨论，但是无疑的是做了预处理比没做的效果要好。

#### 3.23 寻找角点

角点坐标是标定函数cameraCalibrate（）的第2个输入参数。角点是指黑白方块相接的方块定点部分，主要由函数findChessboardCorners（）完成查找，值得注意的是，该函数传入的图像矩阵必须是8-bit灰度图，所以必须做灰度转换。

寻找角点函数部分代码如下：



#### 3.24 寻找物点

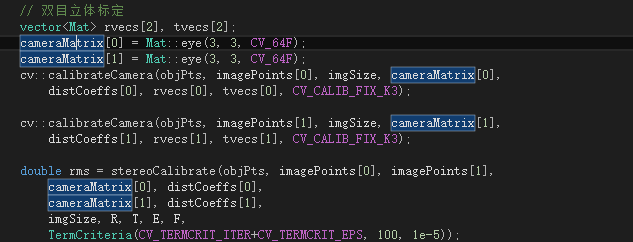
物点，世界坐标系中的点。该坐标类型为vector<vector<Point3f>>， 是标定函数calibrateCamera的第一个参数。其实就是将chessboard中的二维坐标乘以棋盘间隔并加上Z坐标（通常为0）。

代码如下：



#### 标定CalibrateCamera

先对左右镜头分别标定,获得矩阵内参矩阵cameraMatrix、畸变矩阵distCoeffs, 再双目标定，获得旋转矩阵（R），平移矩阵（T），特征矩阵（E），基础矩阵（F）。TermCriteria表示迭代终止的条件。



至此，相机标定完成。

## 四、立体校正

## 七、参考资料

1.寻找角点函数（findChessboardCorners）解析

<http://blog.csdn.net/u011651743/article/details/51099543>

2.标定函数（calibrateCamera）解析

<http://blog.csdn.net/ychl87/article/details/11473593>