相机的标定还是很重要的，特把用过的工具和方法进行一次汇总，以便加深理解，希望和大家多多讨论。（https://shop117107320.taobao.com/index.htm?spm=2013.1.w5002-18156179672.2.727749cc34SL4O里面有双目三维重建、单目标定、双目标定的程序哦，都是经过博主测试）

————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————

**1.MATLAB camera calibration toolbox工具箱**

下载和学习网址：

[http://www.vision.caltech.edu/bouguetj/calib\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/" \t "_blank)

标定测试集的下载网址：

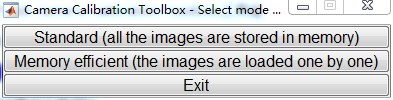
[http://www.vision.caltech.edu/bouguetj/calib\_doc/htmls/example5.html](http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/example5.html" \t "_blank)

我只各使用了13张图片

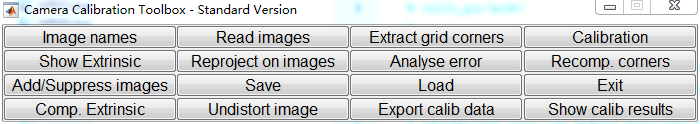
将TOOLBOX\_calib放入MATLAB的toolbox目录下，工作目录设为含标定图集的目录。

1.1单目标定

运行calib\_gui.m文件，选择添加到路径，界面：



选择第一个后，界面：

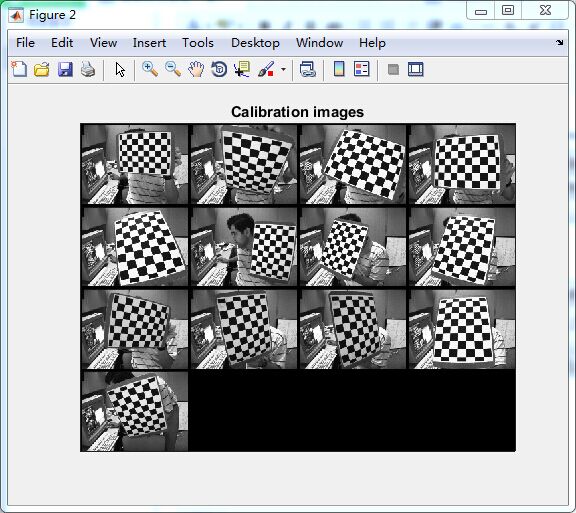


点击Image names或read images后

提示basename，输入left

提示imageformat，输入j

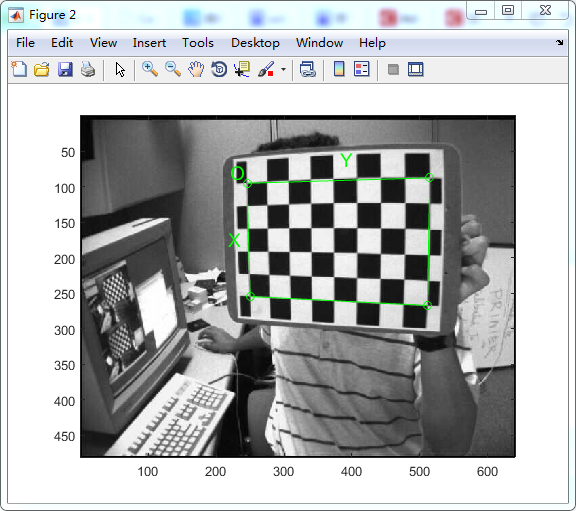
输出读取到的图片集：



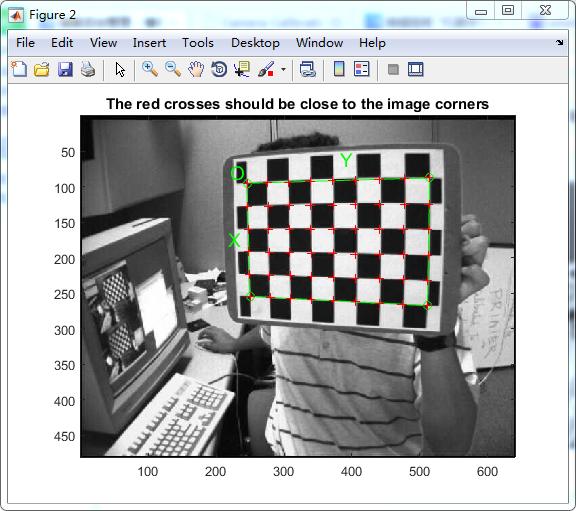
点击extract grid corners(提取角点)

提示角点拾取窗大小，我选择了默认的5\*5

弹出角点拾取窗口界面，拾取4个边界点：



提示输入方格的x向和y向的实际尺寸，这里的棋盘格实际大小为30mm\*30mm，以及X向和Y向的方格个数，输入5和8，设置后得到角点提取的效果：



后续的12张图片进行同样的操作。

（后续MATLAB自己出了个标定工具箱camera calibrator，以及opencv、halcon的角点都是自动提取，稍后详细说明）

完成后点击calibration进行初始标定

标定的初始化如下：

Calibration parameters after initialization:

Focal Length: fc = [ 532.49170 532.49170 ]  
Principal point: cc = [ 319.50000 239.50000 ]  
Skew: alpha\_c = [ 0.00000 ] => angle of pixel = 90.00000 degrees  
Distortion: kc = [ 0.00000 0.00000 0.00000 0.00000 0.00000 ]

得到的标定结果如下：

Calibration results after optimization (with uncertainties):

Focal Length: fc = [ 534.97637 536.20121 ] +/- [ 3.71984 3.92117 ]  
Principal point: cc = [ 343.26051 233.07723 ] +/- [ 4.11643 4.58569 ]  
Skew: alpha\_c = [ 0.00000 ] +/- [ 0.00000 ] => angle of pixel axes = 90.00000 +/- 0.00000 degrees  
Distortion: kc = [ -0.32075 0.27300 0.00104 -0.00074 0.00000 ] +/- [ 0.02040 0.07372 0.00103 0.00126 0.00000 ]  
Pixel error: err = [ 0.55364 0.23137 ]

这里对这5个参数进行说明

（详见链接[http://www.vision.caltech.edu/bouguetj/calib\_doc/htmls/parameters.html](http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/parameters.html" \t "_blank)）

（1）fc为焦距的像素尺寸，计算为F/DX，F/DY，F为几何焦距尺寸，DX,DY为像元尺寸。

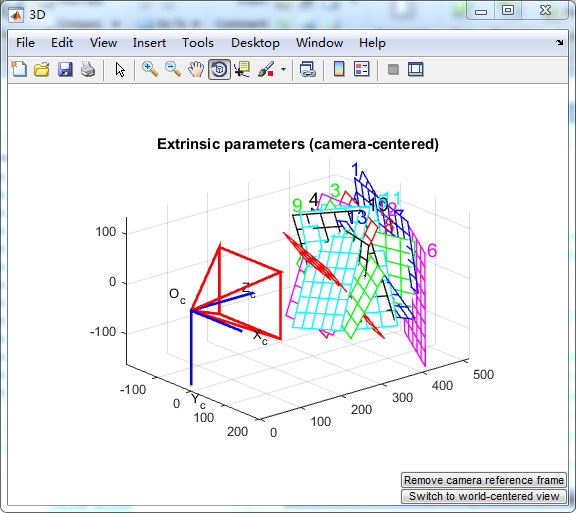
（2）cc为光心在图像坐标系下的图像坐标

（3）alpha\_c倾斜系数，反映X和Y像素轴的夹角

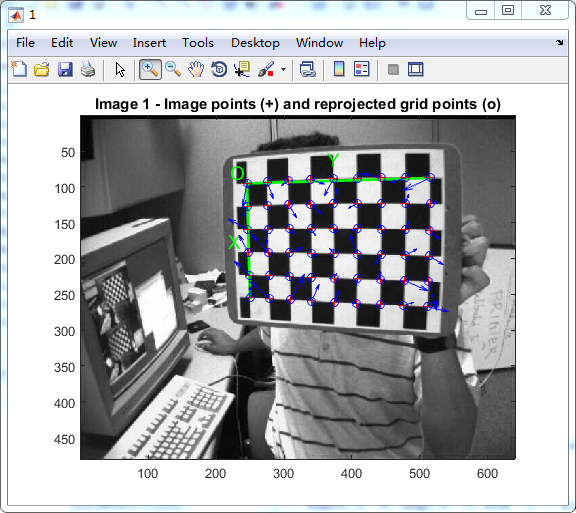
（4）kc畸变系数，这里有5个，前两个和最后一个为径向畸变，第三和第四个为切向畸变，一般第五个为0，切向畸变也很小

（5）err像素误差，可以看到这里小于一个像素尺寸

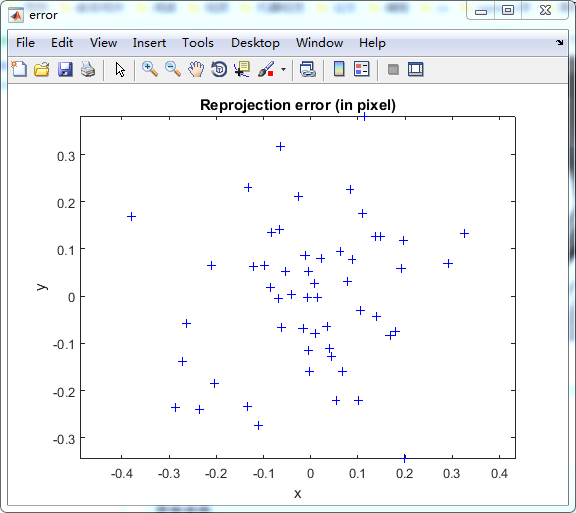
点击show extrinsic，得到相机和标定板的相对位置关系：



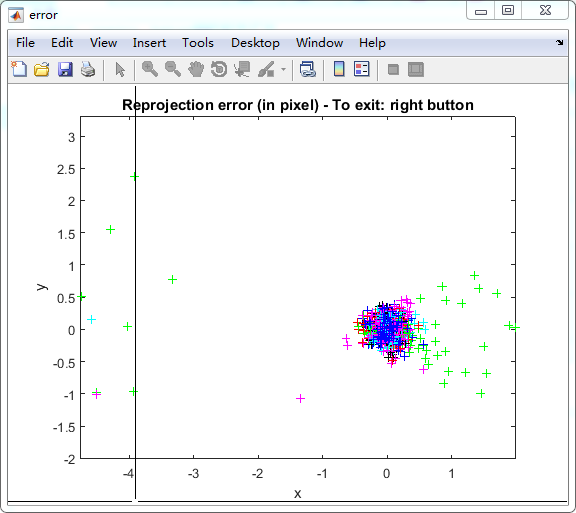
点击reproject on images，根据当前标定的结果和棋盘世界尺寸，得到投影到图像上的角点图



误差图



点击analyse error，对误差进行分析



点击图上点得到相关信息，一种颜色代表一副图,随机点击绿色点

Selected image: 3  
Selected point index: 18  
Pattern coordinates (in units of (dX,dY)): (X,Y)=(5,6)  
Image coordinates (in pixel): (449.80,357.74)  
Pixel error = (0.85299,0.66929)  
Window size: (wintx,winty) = (5,5)

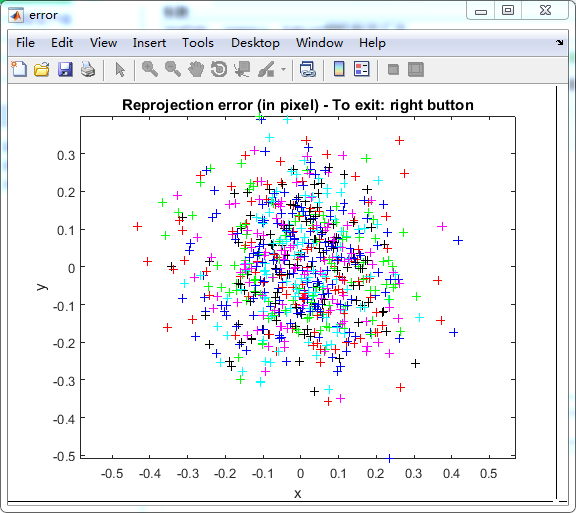
初次标定的效果还算满意，进行一次recomp.corners，对角点进行校正，一路空格。

然后再次标定，得到结果

Calibration results after optimization (with uncertainties):

Focal Length: fc = [ 533.09010 533.21564 ] +/- [ 1.20347 1.26428 ]  
Principal point: cc = [ 342.48318 233.86838 ] +/- [ 1.34457 1.48320 ]  
Skew: alpha\_c = [ 0.00000 ] +/- [ 0.00000 ] => angle of pixel axes = 90.00000 +/- 0.00000 degrees  
Distortion: kc = [ -0.29000 0.10044 0.00121 -0.00016 0.00000 ] +/- [ 0.00646 0.02256 0.00032 0.00041 0.00000 ]  
Pixel error: err = [ 0.13682 0.13975 ]

效果好了很多



点击save保存标定结果Calib\_Results.mat，以供后续的双目标定。

个人感觉此方法标定结果效果更优，可以将MATLAB标定结果转为opencv可处理的xml文件后在vs下进行编程。

工具界面后两排都是一些简单操作，我也简单介绍下

第三排是基本操作：

加减图、保存标定结果、载入标定结果、退出

第四排是小步操作：

获取单图外参、矫正图像、输出标定数据、显示标定结果

这里说下前两个，后两个就是点来看看，不说了

第一个comp.extrinsic是获取单图片的外参，这一步需要相机的标定结果，根据结果获得相机和该图的位置关系。

点击后得到comp.extrinsic后

输入图像的全名，不带格式，第二个提示为格式

然后操作如上，完成后得到以棋盘为世界坐标的相机-世界坐标的外参

Extrinsic parameters:

Translation vector: Tc\_ext = [ 70.077058 -137.151501 379.099782 ]  
Rotation vector: omc\_ext = [ -0.341265 -2.948410 0.719537 ]  
Rotation matrix: Rc\_ext = [ -0.971247 0.194743 -0.136943  
0.235933 0.864274 -0.444260  
0.031840 -0.463796 -0.885370 ]  
Pixel error: err = [ 0.11455 0.17418 ]

参数说明：

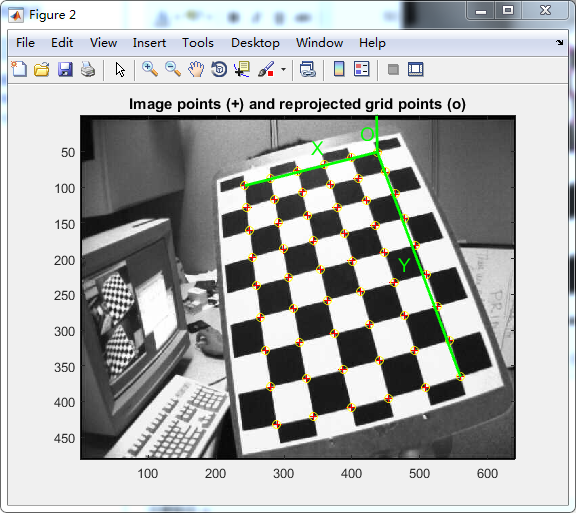
（1）Tc\_ect为平移矢量，这里相机模型不再详述

（2）omc\_ext为旋转矢量

（3）Rc\_ext为旋转矩阵，**Rc\_ext = rodrigues(omc\_ext)**.这里进行了 **rodrigues变换**

（4）err为像素误差

世界坐标系见图，Z轴的标注被吃了：



第二个为undistort image

得到矫正后的图像，并会保存在标定图集的路径下。

经过以上步骤，单目标定结束

点击exit退出

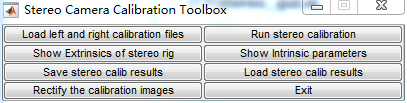
然后标定右相机，做好将左右图集用两个文件夹保存，得到标定结果文件。对两个结果文件重命名，加标注left和right

——————————————————————————————————————————————————————————————————————————

1.2双目标定

 打开**stereo\_gui.m**

操作界面：



点击load left and right calibration files

载入左右标定文件，根据提示输入

然后运行run stereo calibration，进行全局优化的双目标定（？为正负号）

Stereo calibration parameters after optimization:

Intrinsic parameters of left camera:

Focal Length: fc\_left = [ 533.56169 533.59560 ] � [ 0.92519 0.94450 ]  
Principal point: cc\_left = [ 342.53093 234.76465 ] � [ 1.35047 1.34582 ]  
Skew: alpha\_c\_left = [ 0.00000 ] � [ 0.00000 ] => angle of pixel axes = 90.00000 � 0.00000 degrees  
Distortion: kc\_left = [ -0.28866 0.09366 0.00120 -0.00014 0.00000 ] � [ 0.00681 0.02389 0.00031 0.00037 0.00000 ]

Intrinsic parameters of right camera:

Focal Length: fc\_right = [ 536.89174 536.47377 ] � [ 0.97335 0.96952 ]  
Principal point: cc\_right = [ 327.39289 249.94526 ] � [ 1.44008 1.30625 ]  
Skew: alpha\_c\_right = [ 0.00000 ] � [ 0.00000 ] => angle of pixel axes = 90.00000 � 0.00000 degrees  
Distortion: kc\_right = [ -0.29010 0.10622 -0.00051 0.00010 0.00000 ] � [ 0.00531 0.00958 0.00025 0.00062 0.00000 ]

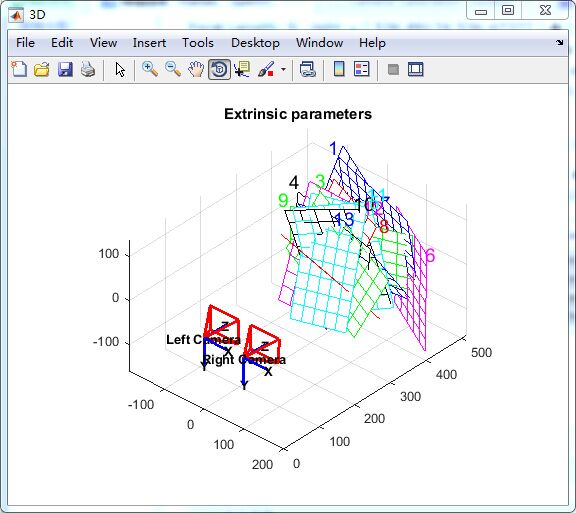
Extrinsic parameters (position of right camera wrt left camera):

Rotation vector: om = [ 0.00715 0.00427 -0.00351 ] � [ 0.00298 0.00336 0.00032 ]  
Translation vector: T = [ -99.81327 1.10181 -0.14312 ] � [ 0.15469 0.12424 0.54609 ]

 om表示右摄像头相对于左摄像头的旋转角度，即以左相机为参考系

 T表示右摄像头相对于左摄像头的平移量，也以左相机为参考系

得到两相机的相对位置

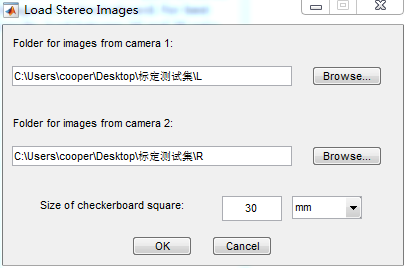


点击save stereo calibration results保存双目标定结果，即完成了MATLAB calibration toolbox工具箱的双目标定。

**————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————**

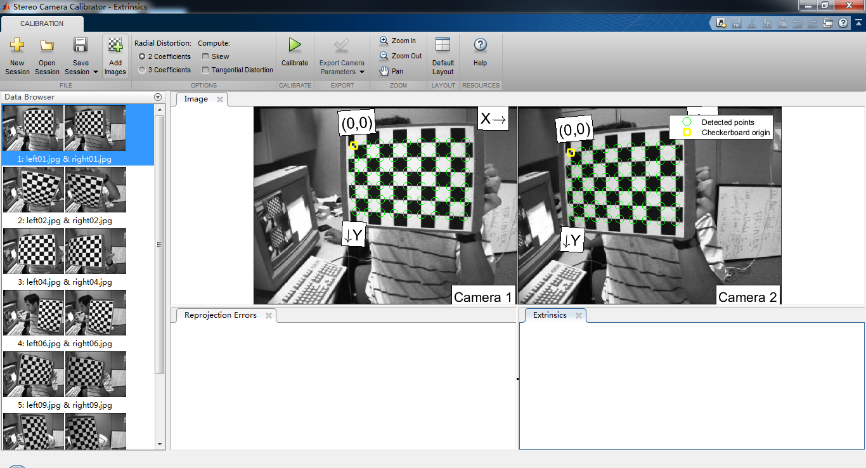
**2.MATLAB 更新的应用stereo camera calibrator**

本人使用MATLAB2015版



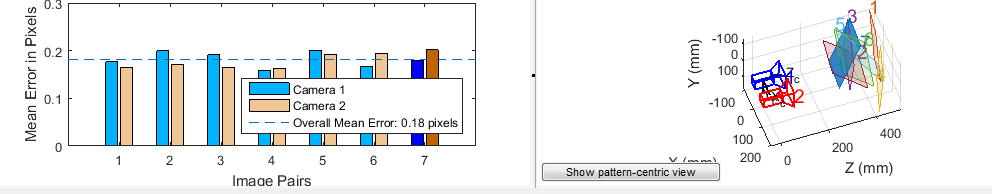
然后自动提取角点，得到对应图像对。这里该工具箱需要保证标定板的X向和Y向的方格数要有一个为偶数，一个为奇数。

结果只检测到7组



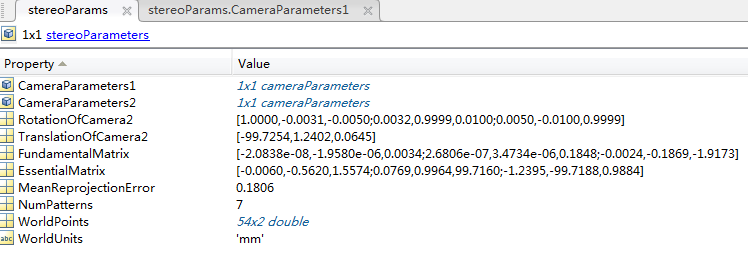
该工具设长轴为X轴

点击标定，得到reprojection errors和extrinsics



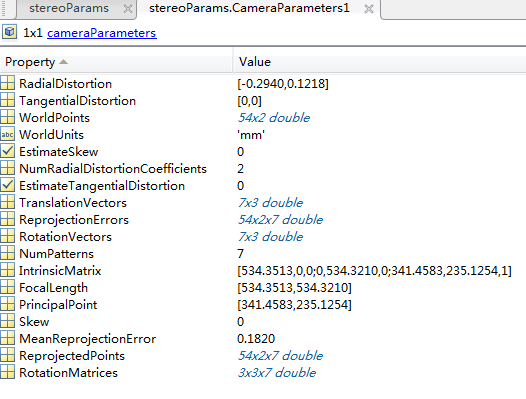
点击输出标定参数

得到标定的结果



可以看到有相机内参数、两相机的旋转平移矩阵、基础矩阵、本质矩阵、平均投影误差、标定数量、角点数、单位

点击cameraParameters，查看左相机的内参数



径向畸变，切向畸变，世界点，单位，扭曲系数，径向畸变参数，切向畸变参数，7张图的平移矢量，7张图的投影误差，7张图的旋转矢量，图片数量，内参数矩阵，焦距的像素尺寸，主点的像素坐标，倾斜系数，平均投影误差，投影点，7张图的旋转矩阵

生成mat文件运行得到标定结果

Camera 1 Intrinsics  
-------------------  
Focal length (pixels): [ 534.3513 +/- 0.4050 534.3210 +/- 0.3959 ]  
Principal point (pixels):[ 341.4583 +/- 0.5920 235.1254 +/- 0.5816 ]  
Radial distortion: [ -0.2940 +/- 0.0028 0.1218 +/- 0.0092 ]

Camera 2 Intrinsics  
-------------------  
Focal length (pixels): [ 537.2223 +/- 0.4180 536.8958 +/- 0.4019 ]  
Principal point (pixels):[ 325.8371 +/- 0.6577 251.4986 +/- 0.5598 ]  
Radial distortion: [ -0.2894 +/- 0.0018 0.1048 +/- 0.0040 ]

Position And Orientation of Camera 2 Relative to Camera 1  
---------------------------------------------------------  
Rotation of camera 2: [ 1.0000 +/- 0.0012 0.0032 +/- 0.0014 0.0050 +/- 0.0001 ]  
Translation of camera 2 (mm): [ -99.7254 +/- 0.0708 1.2402 +/- 0.0580 0.0645 +/- 0.2536 ]

可以看出和calibration toolbox的功能基本一致，角点的提取更加智能化，但提取的效果不是很理想，13张图只检测到了7张，使用上会更加方便

（这里reject的图片过多，怎么提高检测数量希望向大家请教）

————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————

**3.opencv相机标定**

opencv相机标定的参考很多，这里就不再给出链接，自行百度

大概步骤如下：

（1）opencv标定需要对图像的大小，角点的个数，方格的实际尺寸，图像的数量，棋盘的世界坐标等等进行初始化

（2）使用findChessboardCorners进行角点的提取

（3）使用cornerSubPix提取角点子像素级精度

（4）保存每幅图片的角点和世界坐标点

（5）使用calibrateCamera进行单相机的标定

（6）使用 stereoCalibrate进行双目标定

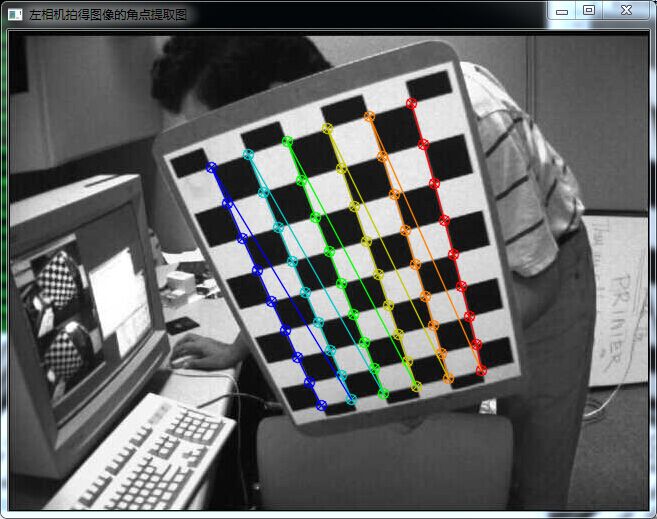
这里对calibrateCamera的标定标志位进行注释

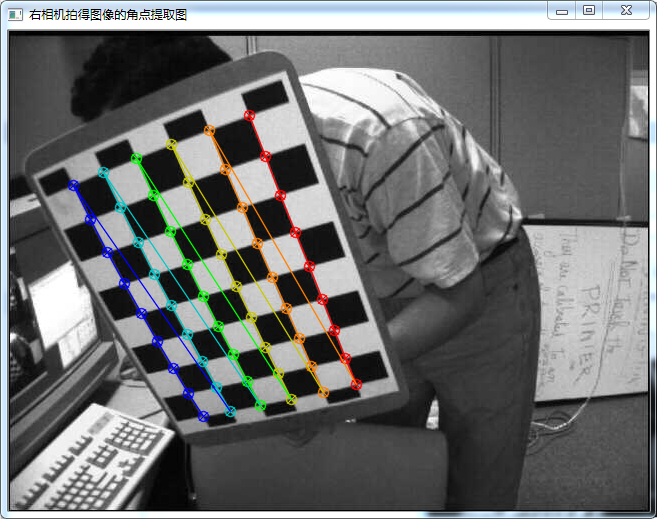
//标志位  
  
/\*CV\_CALIB\_USE\_INTRINSIC\_GUESS -内参数矩阵包含fx，fy，cx和cy的初始值。否则，(cx, cy)被初始化到图像中心（这儿用到图像大小），焦距用最小平方差方式计算得到。注意，如果内部参数已知，没有必要使用这个函数，使用cvFindExtrinsicCameraParams2则可。

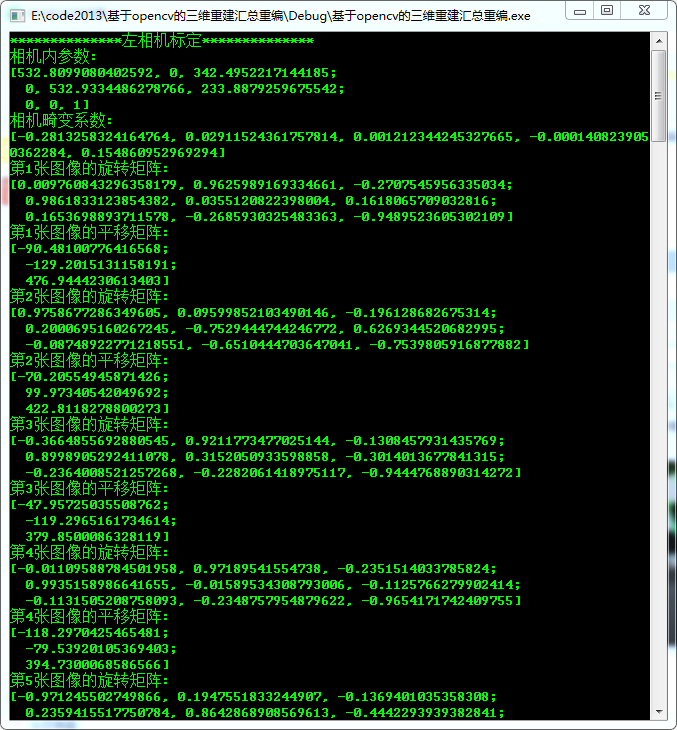
CV\_CALIB\_FIX\_PRINCIPAL\_POINT - 主点在全局优化过程中不变，一直在中心位置或者在其他指定的位置（当CV\_CALIB\_USE\_INTRINSIC\_GUESS设置的时候）。

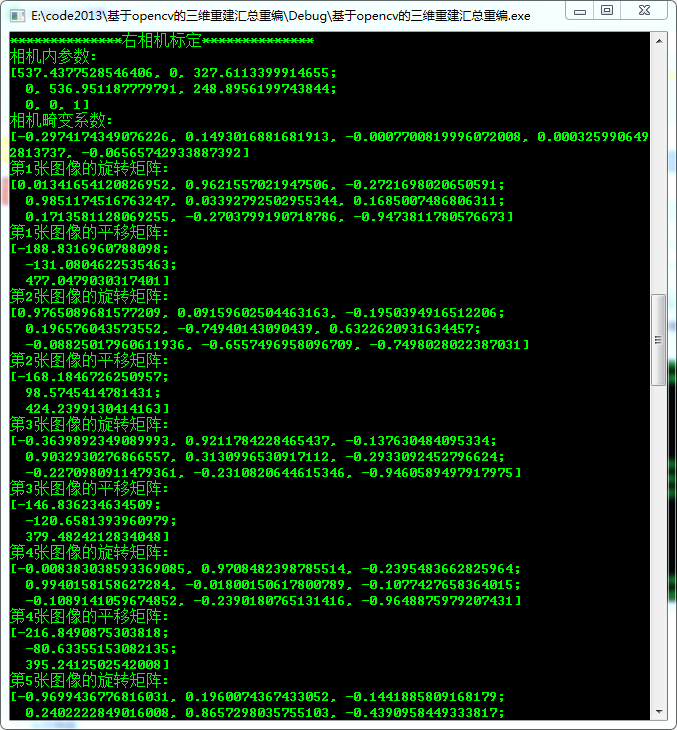
CV\_CALIB\_FIX\_ASPECT\_RATIO - 优化过程中认为fx和fy中只有一个独立变量，保持比例fx/fy不变，fx/fy的值跟内参数矩阵初始化时的值一样。在这种情况下， (fx, fy)的///实际初始值或者从输入内存矩阵中读取（当CV\_CALIB\_USE\_INTRINSIC\_GUESS被指定时），或者采用估计值（后者情况中fx和fy可能被设置为任意值，只有比值被使用）。

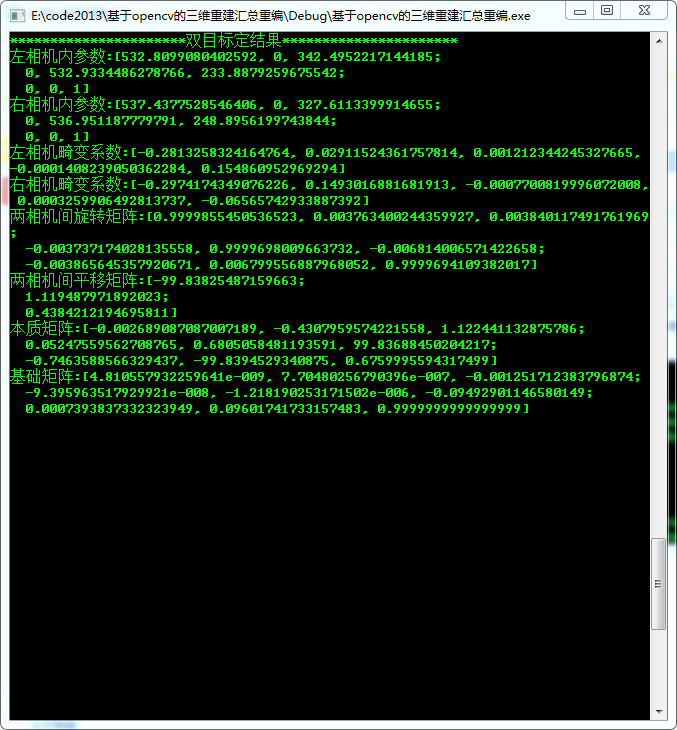
CV\_CALIB\_ZERO\_TANGENT\_DIST – 切向形变参数(p1, p2)被设置为0，其值在优化过程中保持为0。\*/  
  
  
尺寸设为30，单位为mm。  
  
  
  
角点提取的效果如下：











————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————

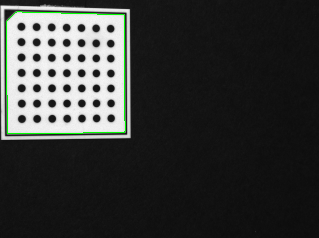
**4.halcon标定**

对于halcon软件，目前才开始学习。只是跑过示例，了解不多

一般步骤：

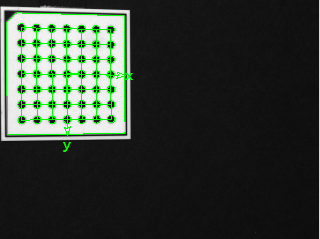
（1）初始化参数

（2）使用 find\_caltab找到标定板区域



（3）使用find\_marks\_and\_pose确定标定点坐标和相机外参

（4）使用disp\_caltab显示投影的标志点，并连线

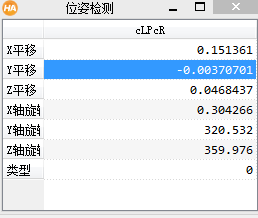


（5）使用binocular\_calibration进行双目标定

binocular\_calibration (X, Y, Z, RowsL, ColsL, RowsR, ColsR, StartCamParL, StartCamParR, StartPosesL, StartPosesR, 'all', CamParamL, CamParamR, NFinalPoseL, NFinalPoseR, cLPcR, Errors)

（6）得到左、右相机的内参数和相对位姿、误差如下：

http://images2015.cnblogs.com/blog/748646/201511/748646-20151110162906587-1614607435.png



errors为0.0154181，这里拍摄的标定图片效果很好，精度很高

以上便是目前常用的几种标定方法。