Camera Calibration Toolbox使用教程

首先说明一下为什么需要做Camera Calibration. 现实的摄像机都是采用光学透镜聚光成像的，并且所用的透镜由于加工工艺限制不可能是严格的抛物面, 同时, 感光芯片也透镜之间也非严格平行. 因此我们使用的摄像机所产生的画面实际上是存在扭曲和偏移的. 如果直接使用原始的摄像机画面进行测距, 势必造成误差. 因此需要进行相机的标定, 通过标定后获取消除画面中的扭曲和偏移, 再用来进行激光测距的相关操作.

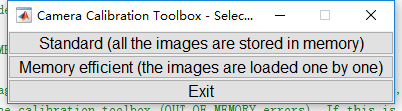
相机的标定需要使用标定图像. 用需要标定的相机在不同位置不同角度以不同姿态拍摄黑白棋盘格标定板得到标定图像, 至少需要3张, 以10至20张为宜. 黑白棋盘格标定板参考文件中的checkboard.pdf, 可以直接用A4纸打印出来.

关于更多相机标定的原理和详细内容可以浏览下面的官网.

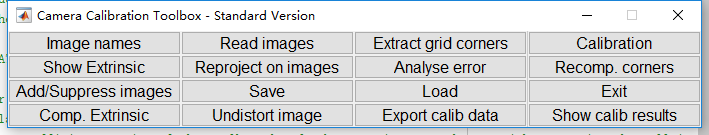
从官网(http://www.vision.caltech.edu/bouguetj/calib\_doc/)上获取Camera Calibration Toolbox. 同时上面说明了如何开始, 也提供了一些Calibration example.

下载完成并解压. 将拍摄好的不同角度的棋盘格照片拷贝到目录下. 注意命名成序列形式如chess01.jpg, chess02.jpg, …, chess20.jpg

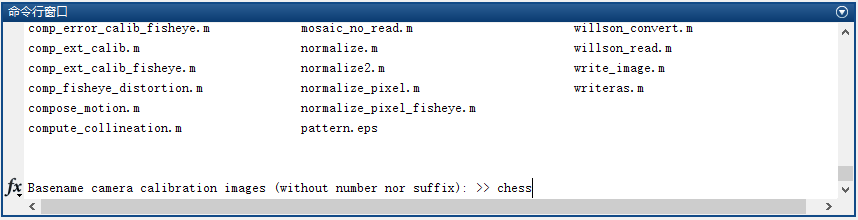
打开matlab, 更改工作区到目录下, 打开calib\_gui.m或calib.m(官网上Getting started有详细说明). 打开calib\_gui.m出现如下界面, 选择Standard即可.



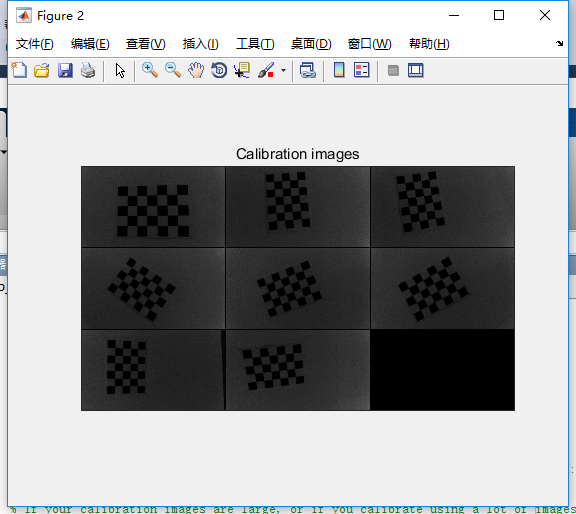
进入如下二级界面.

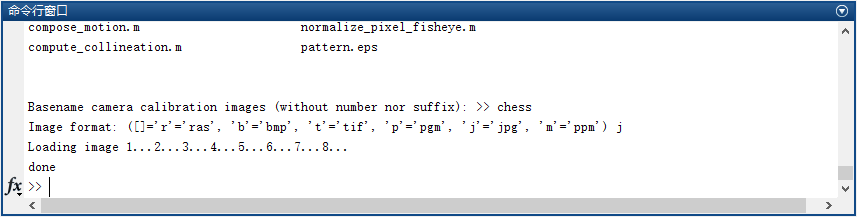


首次使用需要先设置图像序列的Basename. 点击Image names, 在命令行窗口输入图像序列的公共名, 如上例则输入chess. 如图.

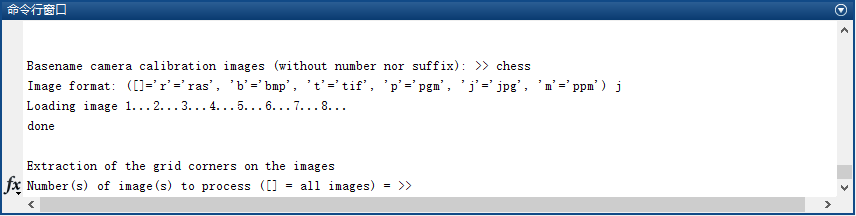


回车之后需要选择图像格式, 我用的例子是jpg, 所以是j. 之后程序自动读取目录下所有符合命名和格式的图像. 这里我只有8张图.

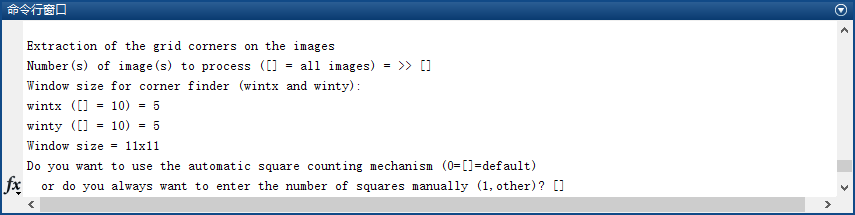




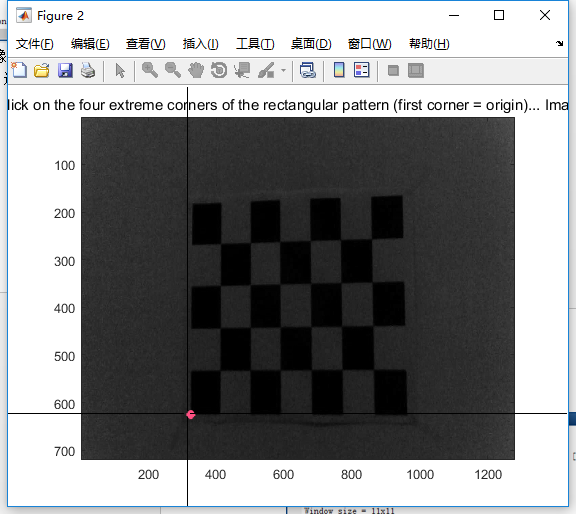
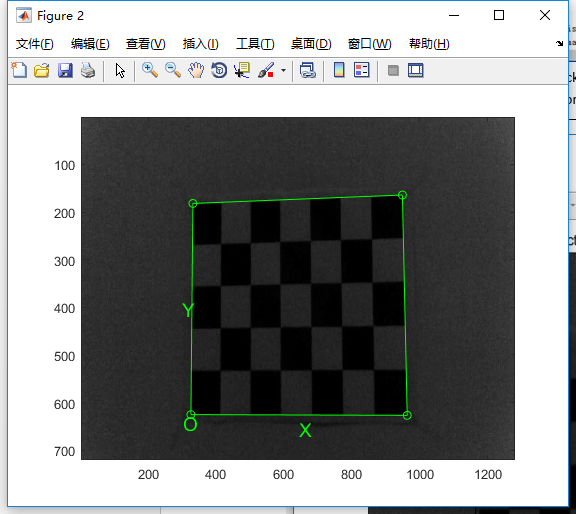
下一步, 提取每一张图中的角点. 点击Extract grid corners. 要求输入处理的图像数量, 这里全部使用这8张, 所以输入8或者如提示输入[].



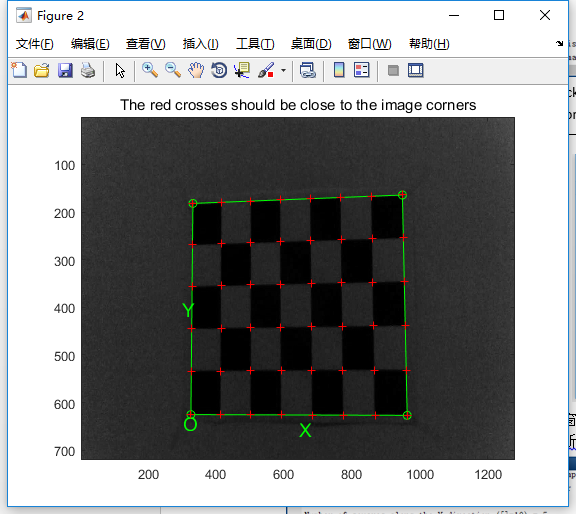
之后要求输入寻找角点的滑动窗口的大小, 统一选择5\*5. 其意义不做赘述, 有兴趣的同学自行翻阅官方文档. 还会做其他询问, 节约时间我们都选择默认[].

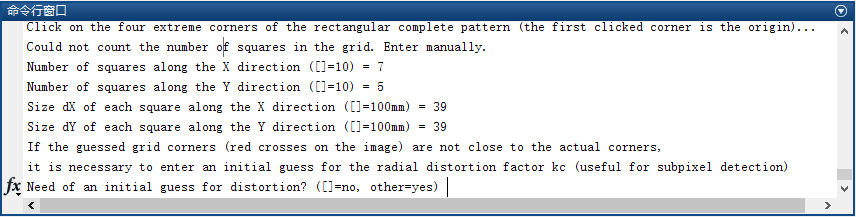


下面开始一张张处理图像, 每一张都提示Click on the four extreme corners of the rectangular complete pattern (the first clicked corner is the origin)...意为要求用户自己点选出棋盘格的四角, 第一个点选角为坐标原点. 如图是点选了第一个点以及点选完成的示例.

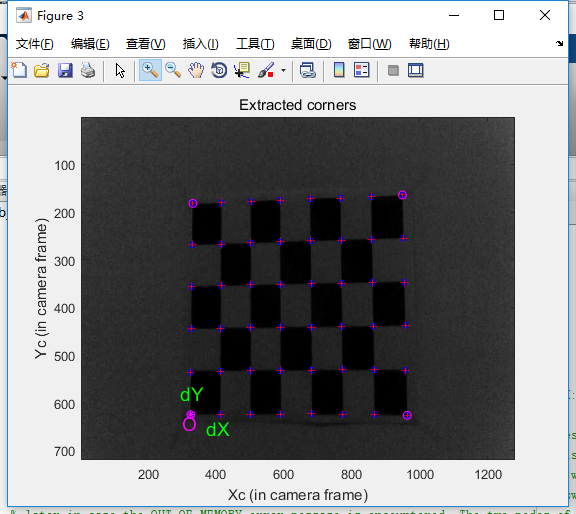
 

按照用户点选的原点产生了坐标系, 命令行窗口要求用户输入X方向和Y方向上的方块个数以及方块的实际尺寸. 然后输出一个估测的所有角点位置. 如图.

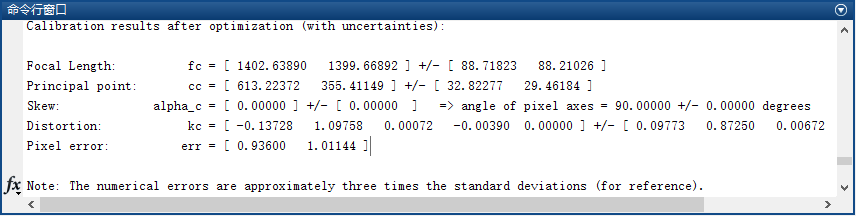




这时命令行会有一个询问, 如果用户认为估测点和图像有较大差异, 在此可以输入一个可能的径向畸变系数来帮助软件重新判断角点位置. 一般这个系数在-1到1之间. 在此没有必要. 略过. 这样就将角点提取出来了.



之后对其他图像进行相同的操作. 当所有图像处理完成后点击Calibration, 即可得到我们需要的参数. 点击save保存为.m文件, 其他功能感兴趣可以自行探索.



只关心Focal Length, Principal point, Distortion几组参数. 编写两个xml以在程序中导入它们.

设得到的参数结果为:

Focal Length: fc = [f1 f2]

Principal point: cc = [c1 c2]

Distortion: kc = [k1 k2 k3 k4 k5]

那么这两个xml应该是:

instrinsics\_cam.xml:

<?xml version="1.0"?>

[<opencv\_storage>](file:///C:\Users\Administrator\Documents\Visual%20Studio%202015\Projects\lasercenterTest\lasercenterTest\cali_result\instrinsics_cam.xml)

[<instrinsics\_cam type\_id="**opencv-matrix**">](file:///C:\Users\Administrator\Documents\Visual%20Studio%202015\Projects\lasercenterTest\lasercenterTest\cali_result\instrinsics_cam.xml)

<rows>3</rows>

<cols>3</cols>

<dt>d</dt>

<data>f1 0. c1 0. f2 c2 0. 0. 1.</data>

</instrinsics\_cam>

</opencv\_storage>

instrinsics\_distort.xml:

<?xml version="1.0"?>

[<opencv\_storage>](file:///C:\Users\Administrator\Documents\Visual%20Studio%202015\Projects\lasercenterTest\lasercenterTest\cali_result\instrinsics_cam.xml)

[<instrinsics\_distort type\_id="**opencv-matrix**">](file:///C:\Users\Administrator\Documents\Visual%20Studio%202015\Projects\lasercenterTest\lasercenterTest\cali_result\instrinsics_cam.xml)

<rows>1</rows>

<cols>5</cols>

<dt>d</dt>

<data>0.6\*k1 0.6\*k2 0.6\*k3 k4 k5</data>

</instrinsics\_distort>

</opencv\_storage>

这两个xml文件通过cameraundistortor.cpp中的loadCalibrationfromFile()导入.