

最详细、最完整的相机标定讲解

未经博主允许不得转载<https://blog.csdn.net/a083614/article/details/78579163>

最近做项目要用到标定，因为是小白，很多东西都不懂，于是查了一堆的博客，但没有一个博客能让我完全能看明白整个过程，绝大多数都讲的不全面，因此自己总结了一篇博客，给自己理一下思路，也能够帮助大家。(张正友标定的详细求解还未完全搞明白，后面再加)

参考博客：

相机标定 (Camera calibration) 原理、步骤

(<http://blog.csdn.net/lql0716/article/details/71973318?locationNum=8&fps=1>)

相机标定内容详解 (转载自祥的博客)

(<http://blog.csdn.net/humanking7/article/details/44756073>)

张正友标定介绍

(<http://blog.csdn.net/heroacool/article/details/50286677>)

[图像]摄像机标定(2) 张正友标定推导详解

(<http://blog.csdn.net/humanking7/article/details/44756235>)

MATLAB—相机标定教程

(<http://blog.csdn.net/heroacool/article/details/51023921>)

在图像测量过程以及机器视觉应用中，为确定空间物体表面某点的三维几何位置与其在图像中对应点之间的相互关系，必须建立相机成像的几何模型，这些几何模型参数就是相机参数。在大多数条件下这些参数必须通过实验与计算才能得到，这个求解参数（内参、外参、畸变参数）的过程就称之为 **相机标定**（或摄像机标定）。无论是在图像测量或者机器视觉应用中，相机参数的标定都是非常关键的环节，其标定结果的精度及算法的稳定性直接影响相机工作产生结果的准确性。因此，做好相机标定是做好后续工作的前提，提高标定精度是科研工作的重点所在。

畸变（distortion）是对直线投影（rectilinear projection）的一种偏移。简单来说直线投影是场景内的一条直线投影到图片上也保持为一条直线。畸变简单来说就是一条直线投影到图片上不能保持为一条直线了，这是一种光学畸变（optical aberration），可能由于摄像机镜头的原因。

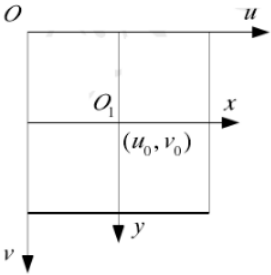
相机的畸变和内参是相机本身的固有特性，标定一次即可一直使用。但由于相机本身并非理想的小孔成像模型以及计算误差，采用不同的图片进行标定时得到的结果都有差异。一般重投影误差很小的话，标定结果均可用。

坐标转换基础

在视觉测量中，需要进行的一个重要预备工作是定义四个坐标系的含义，即 **摄像机坐标系**、**图像物理坐标系**、**图像像素坐标系** 和 **世界坐标系（参考坐标系）**。

一、图像坐标系(x,y)至像素坐标系(u,v)

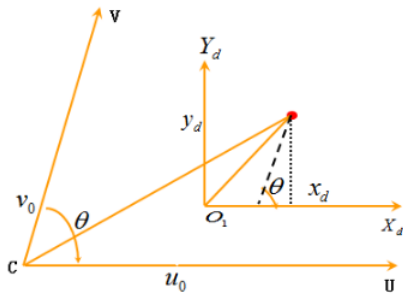
1.两坐标轴互相垂直



此时有

$$\begin{cases} u = \frac{x}{dx} + u_0 \\ v = \frac{y}{dy} + v_0 \end{cases}$$
$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

2.一般情况，两轴不互相垂直



此时有

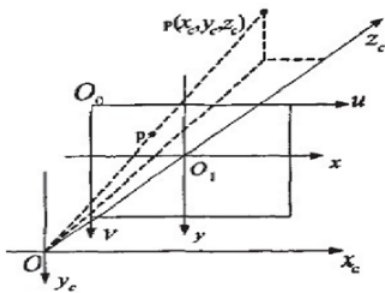
$$u = u_0 + \frac{x_d}{dx} - \frac{y_d \cot \theta}{dx}$$

$$v = v_0 + \frac{y_d}{dy \sin \theta}$$

写成矩阵形式为：

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & -f_u \cot \theta & u_0 \\ 0 & f_v / \sin \theta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} \quad \text{其中} \quad f_u = \frac{1}{dx}, f_v = \frac{1}{dy}$$

二、相机坐标系(Xc,Yc,Zc)至图像坐标系(x,y)

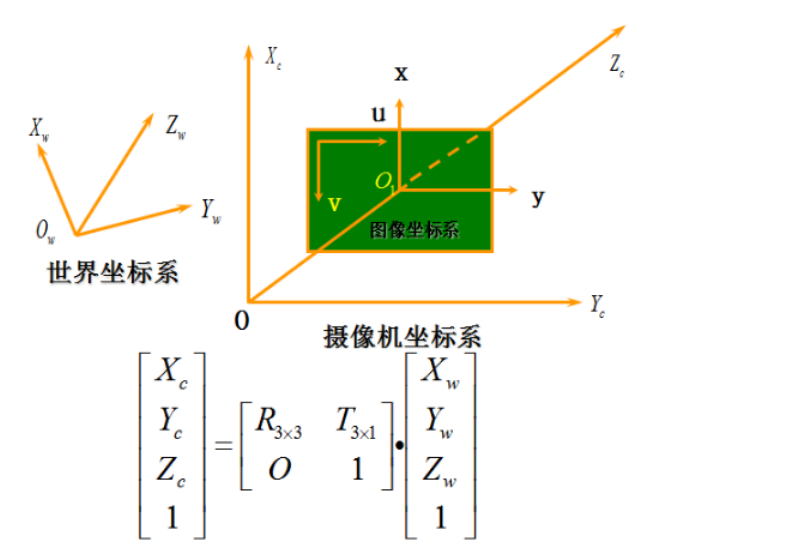


(根据小孔成像原理，图像坐标系应在相机坐标系的另一边，为倒立反向成像，但为方便理解和计算，故投影至同侧。)

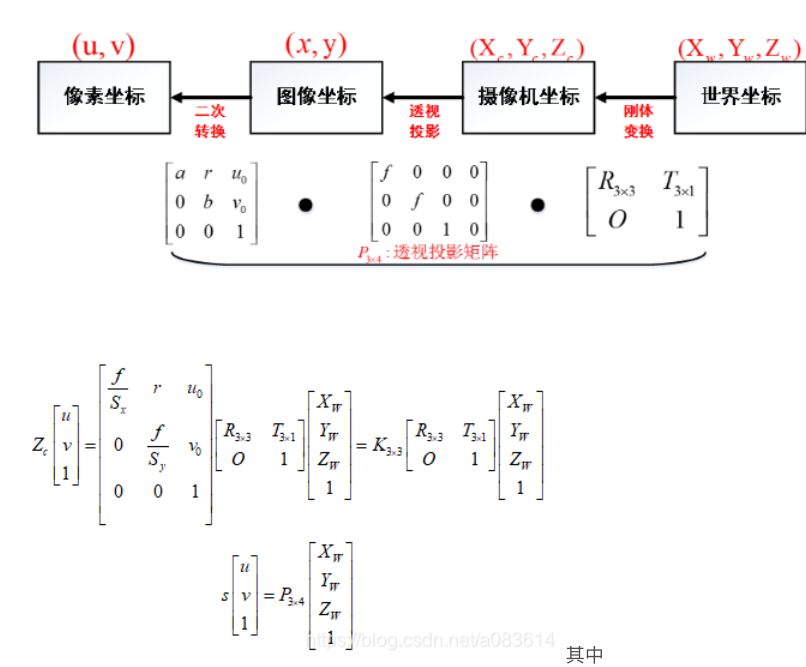
根据三角形相似性原理得：

$$Z_c \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

三、世界坐标系(Xw,Yw,Zw)至相机坐标系(Xc,Yc,Zc)



###四、合并公式



$K_{3,4}$: 摄像机内参矩阵⁴⁾

$P_{3,4}$: 透视摄影矩阵⁴⁾

$s = Z_c$: 尺度因子⁴⁾

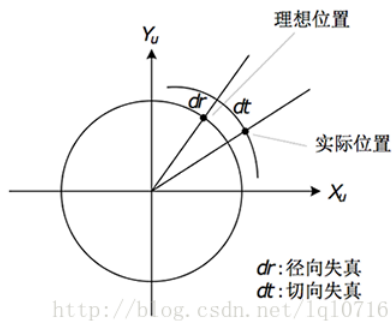
畸变

畸变一般可以分为：径向畸变、切向畸变

ps:畸变是相机本身的固有特性，和相机内参相同，标定一次之后即可。

径向畸变来自于透镜形状。
切向畸变来自于整个摄像机的组装过程。

畸变还有其他类型的畸变，但是没有径向畸变、切向畸变显著，所以忽略不计。



径向畸变（桶形畸变和枕形畸变）：

实际摄像机的透镜总是在成像仪的边缘产生显著的畸变，这种现象来源于“筒形”或“鱼眼”的影响。

如下图，光线在原理透镜中心的地方比靠近中心的地方更加弯曲。对于常用的普通透镜来说，这种现象更加严重。筒形畸变在便宜的网络摄像机中非常厉害，但在高端摄像机中不明显，因为这些透镜系统做了很多消除径向畸变的工作。

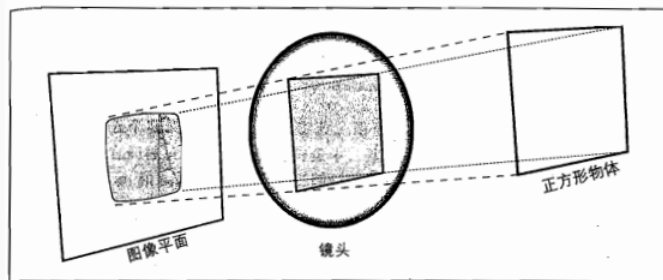


图 11-3：径向畸变。远离透镜中心的光线弯曲比靠近中心的严重。因此正方形的边在图像平面上为弯曲(即筒形畸变)

对于径向畸变，成像仪中心（光学中心）的畸变为0，随着向边缘移动，畸变越来越严重。

对于径向畸变可以通过下面的泰勒级数展开式进行校正：

$$x_{\text{corrected}} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

$$y_{\text{corrected}} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

这里的x、y是图像中的畸变后的位置坐标，通过校正得到真实坐标。r是该点距成像中心的距离。

切向畸变（薄透镜畸变和离心畸变）：

切向畸变是由于透镜制造上的缺陷使得透镜本身与图像平面不平行而产生的。

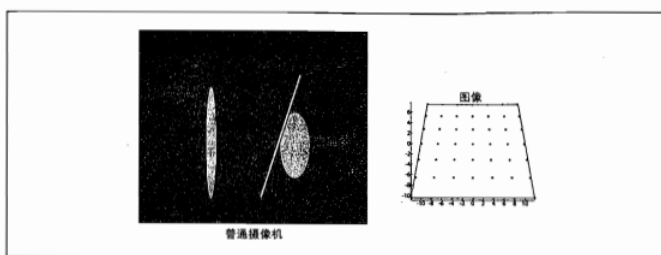


图 11-5：当透镜不完全平行于图像平面的时候产生切向畸变，这种现象发生于成像仪被粘贴在摄像机的时候(经 Sebastian Thrun 授权)

切向畸变可以通过如下公式来矫正：

$$x_{\text{corrected}} = x + [2p_1 y + p_2 (r^2 + 2x^2)]$$

$$y_{\text{corrected}} = y + [2p_2 x + p_1 (r^2 + 2y^2)]$$

这里的x、y是图像中的畸变后的位置坐标，通过校正得到真实坐标。r是该点距成像中心的距离。

opencv 畸变模型

径向畸变模型：

$$x_c = x_d (1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots)$$

$$y_c = y_d (1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots)$$

切向畸变模型：

$$\begin{aligned}x_c &= 2p_1xdy + p_2(r^2 + 2xd^2) \\y_c &= 2p_2xdy + p_1(r^2 + 2yd^2)\end{aligned}$$

可得：

$$\begin{aligned}x_c &= x_d(1 + k_1r^2 + k_2r^4) + 2p_1xdy + p_2(r^2 + 2xd^2) \\y_c &= y_d(1 + k_1r^2 + k_2r^4) + 2p_2xdy + p_1(r^2 + 2yd^2)\end{aligned}$$

x_d, y_d 为图像中的点（存在畸变）， x_c, y_c 是畸变校正后的点。

由公式可得畸变有 k_1, k_2, k_3, p_1, p_2 五个参数，对于质量比较好的相机来说，切向畸变很小，可忽略，径向畸变系数 k_3 也可忽略，只计算 k_1, k_2 两个参数。张正友标定中就默认为 p_1, p_2 为0。

标定方法

1、传统相机标定

最简单的相机标定为线性标定，即不考虑相机的畸变而只考虑空间坐标转换。

每个坐标点有X,Y两个变量，可列两个方程，相机内参有5个未知数，外参平移和旋转各3个，共有11个变量，因此至少需要6个特征点来求解。

2、非线性标定

当镜头畸变明显时必须考虑畸变，一般较为便宜的网络摄像头畸变特别大，而价格较贵的工业摄像头则畸变很小，因为其中已经嵌入了许多消除畸变的程序。这时线性模型转化为非线性模型，需要通过非线性标定方法求解。有最速下降法，遗传算法，高斯牛顿法和神经网络算法等。

3、张正友标定介绍

（张正友标定只考虑了径向畸变，没有考虑切向畸变）

由上面的坐标转换可得：

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ 1 \end{bmatrix}$$

H即为单应性矩阵

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = H \begin{bmatrix} X_W \\ Y_W \\ 1 \end{bmatrix}$$

H为3x3矩阵，并且有一个元素作为齐次坐标，则有8个未知元素，一组坐标对应两个方程，则至少需要四组对应的点即可算出单应性矩阵H。

$$H = A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}^T$$

因为R旋转矩阵为正交矩阵，存在：

$$\begin{aligned}r_1^T r_2 &= 0 \\ r_1^T r_1 &= r_2^T r_2 = 1\end{aligned}$$

代入H可得出两组A和H的公式：

$$\begin{aligned}h_1^T A^{-T} A^{-1} h_2 &= 0 \\ h_1^T A^{-T} A^{-1} h_1 &= h_2^T A^{-T} A^{-1} h_2\end{aligned}$$

矩阵A包含5个元素，需要3组H方可解出A的唯一封闭解，因此在标定时需拍摄3组以上的图片。由A可计算出相应的外参矩阵。在上述计算中，忽略了相机畸变的影响。对内外参应用最小二乘法估计实际存在的径向畸变的畸变系数（忽略切向畸变），最后通过极大似然法进行优化，得到精度较高的解。

标定过程

相机标定步骤：

- 1、打印一张棋盘格，把它贴在一个平面上，作为标定物。
 - 2、通过调整标定物或摄像机的方向，为标定物拍摄一些不同方向的照片。
 - 3、从照片中提取棋盘格角点。
 - 4、估算理想无畸变的情况下，五个内参和六个外参。
 - 5、应用最小二乘法估算实际存在径向畸变下的畸变系数。
 - 6、极大似然法，优化估计，提升估计精度。

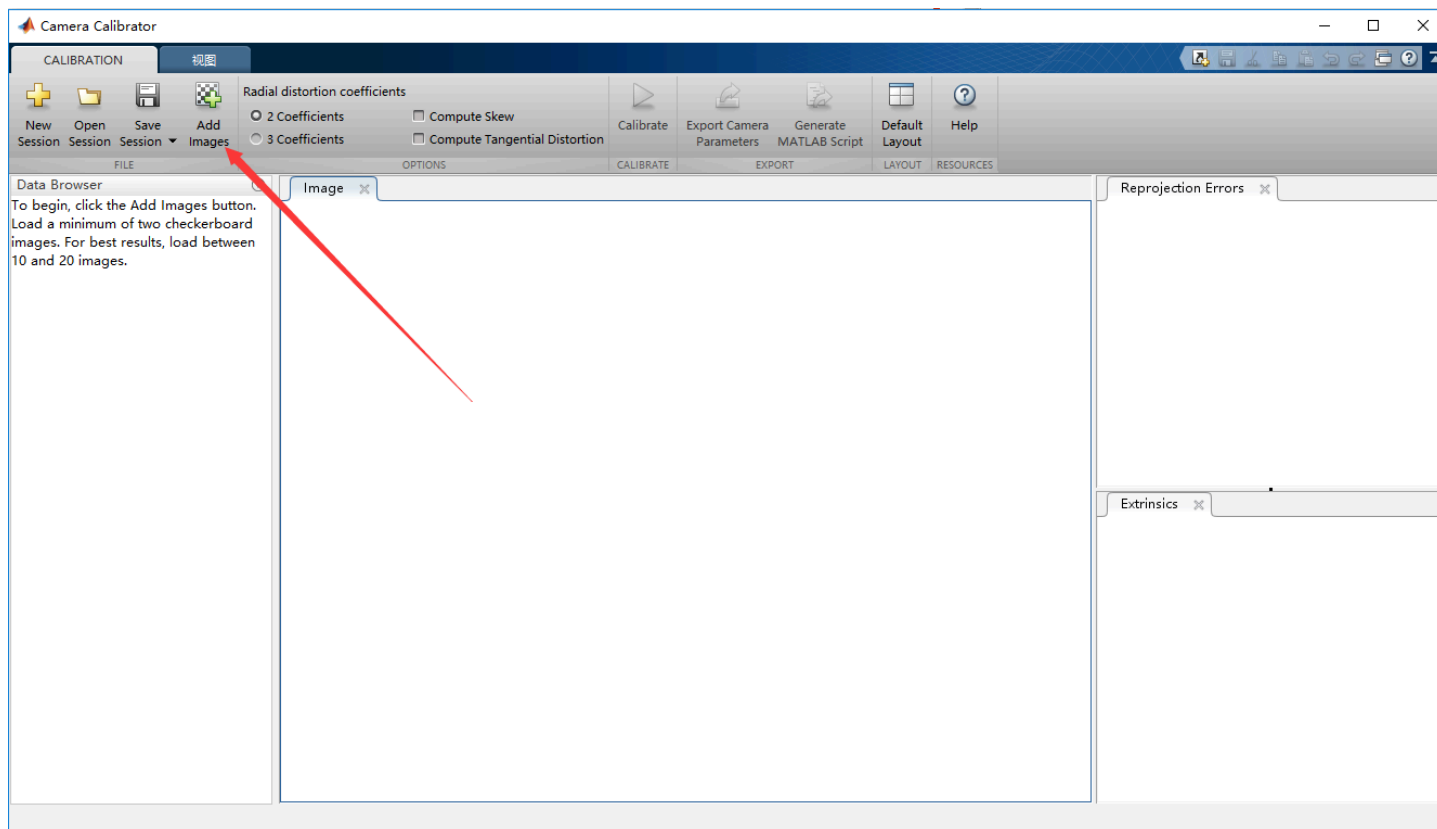
Matlab 工具箱标定：

- 1、应用程序中找到Camera Calibration

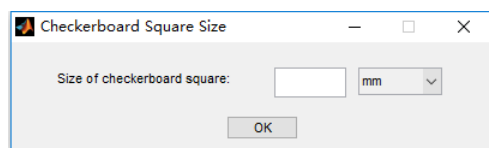
MATLAB R2014a



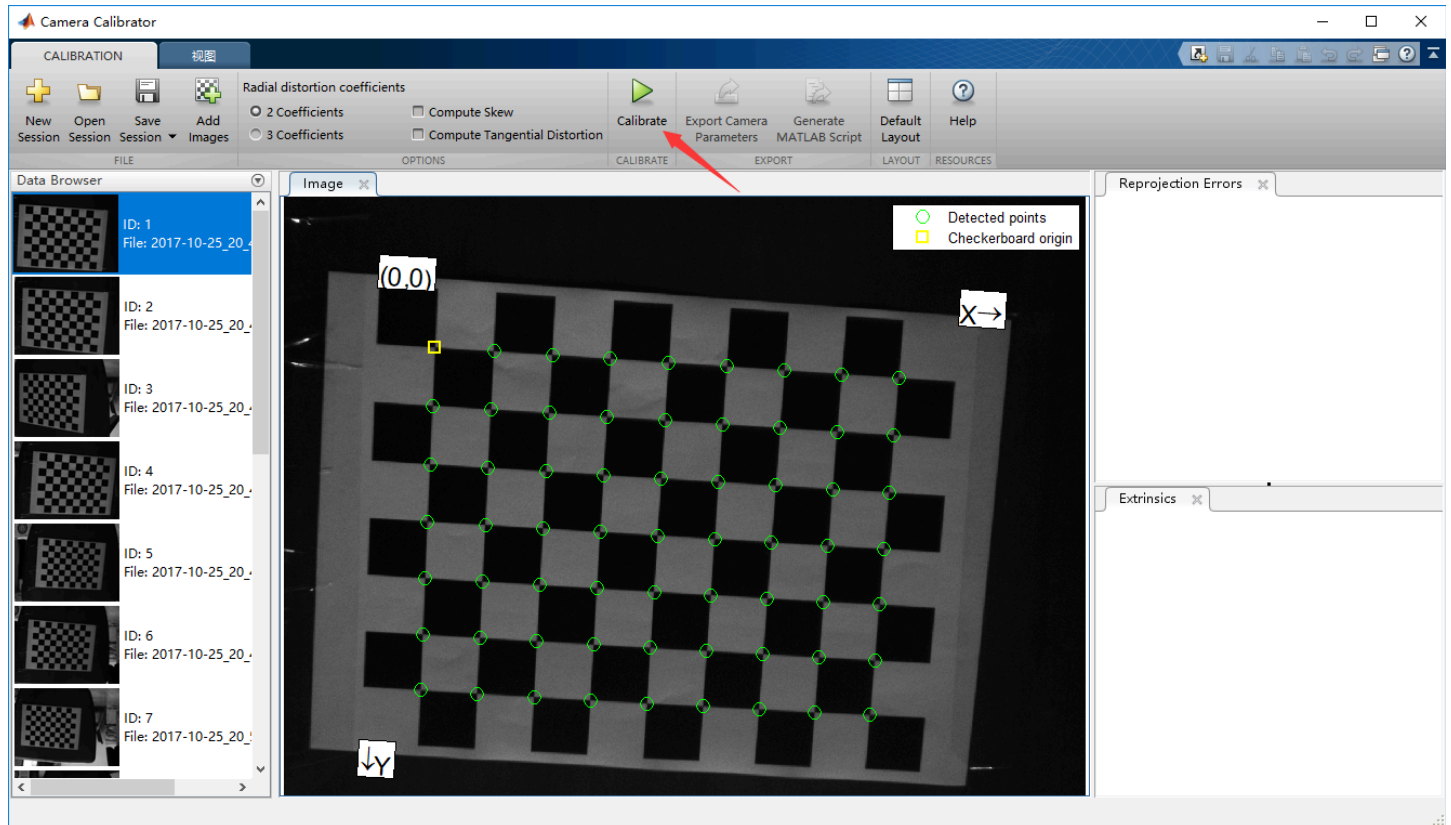
2、添加标定板拍摄图片（按Ctrl可一次添加多张）



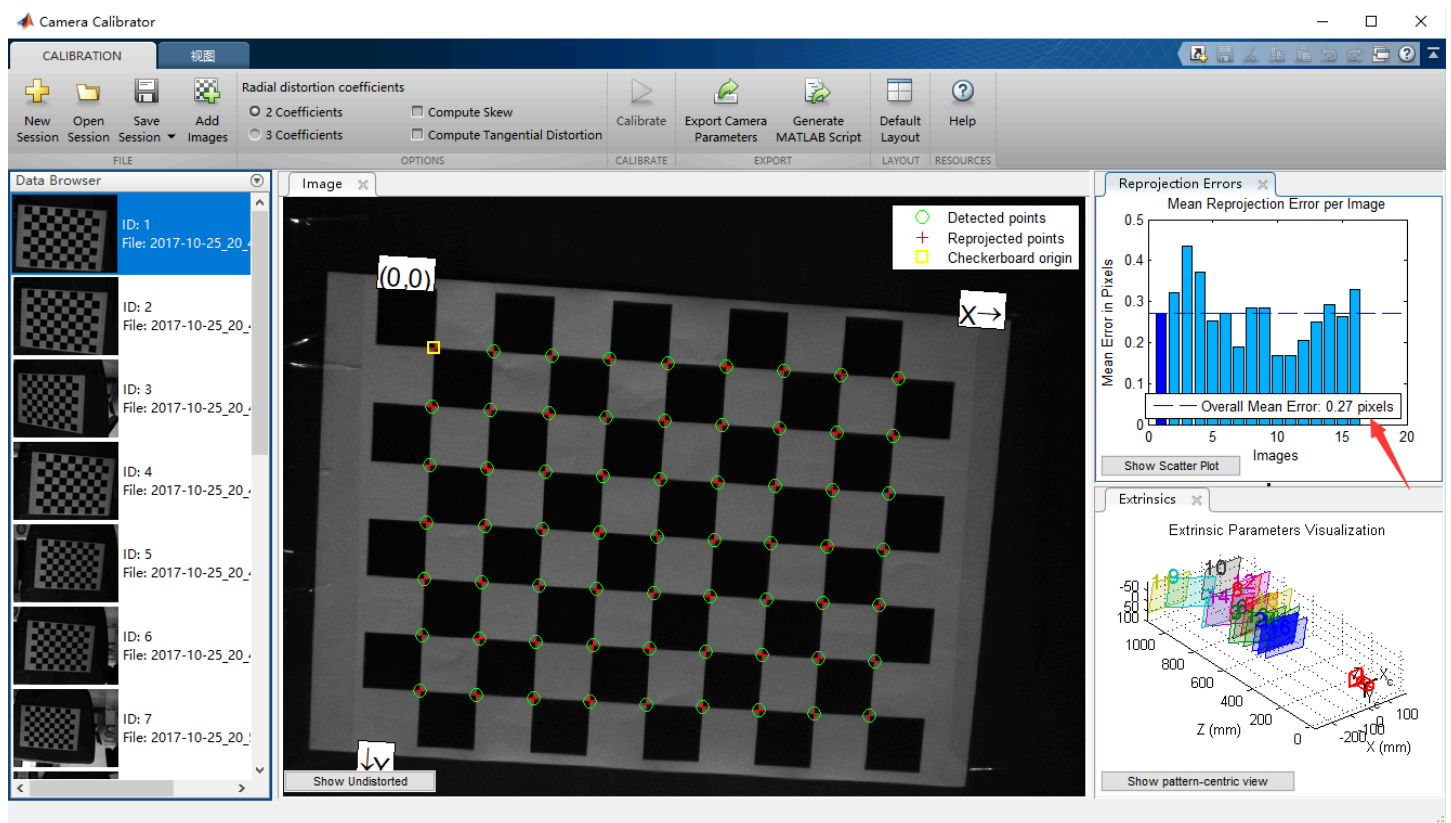
3、输入棋盘格每格的尺寸大小



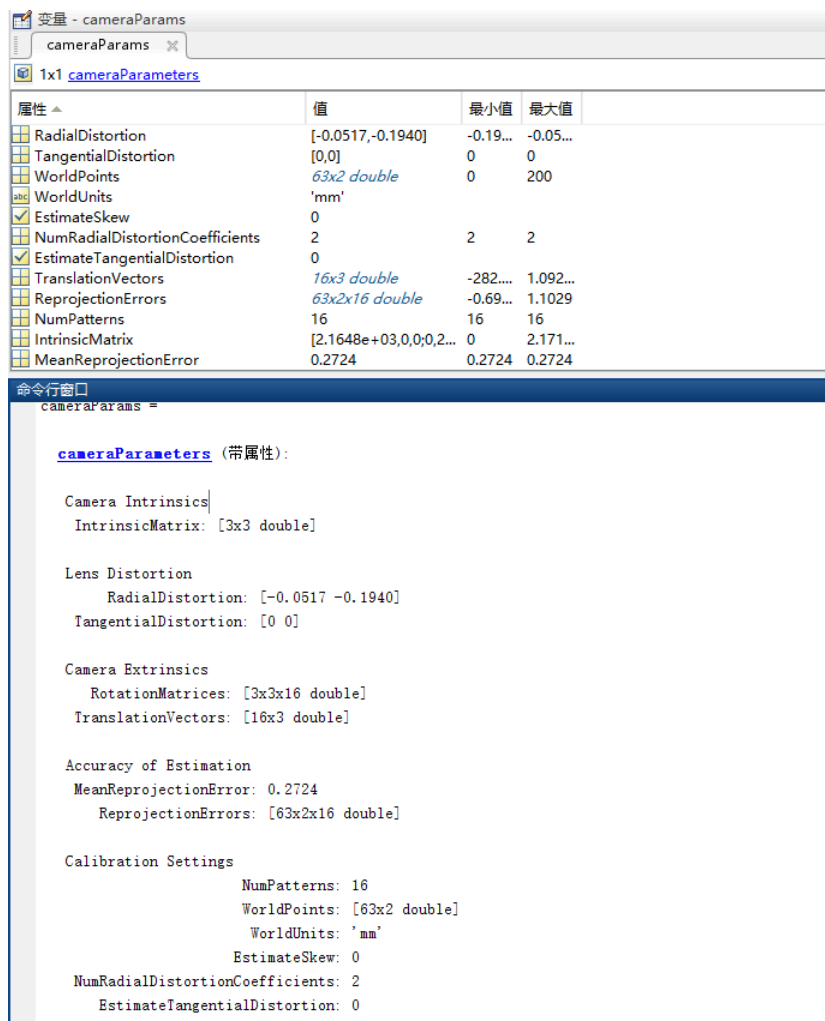
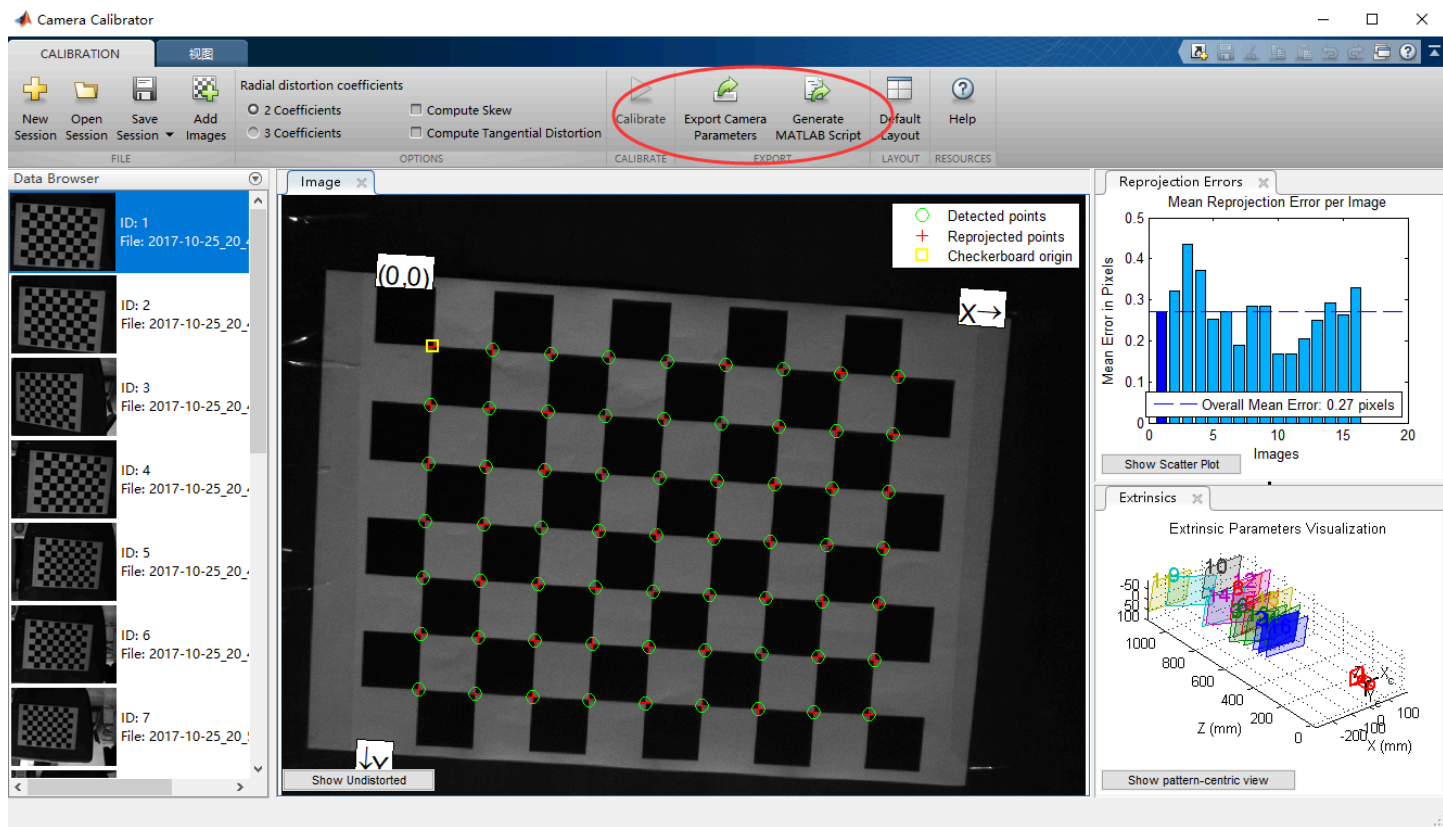
4、显示已检测出的棋盘格，点击Calibration，开始标定。



5、得到标定结果（平均误差小于0.5即可认为结果可靠，这个地方我不是很清楚）



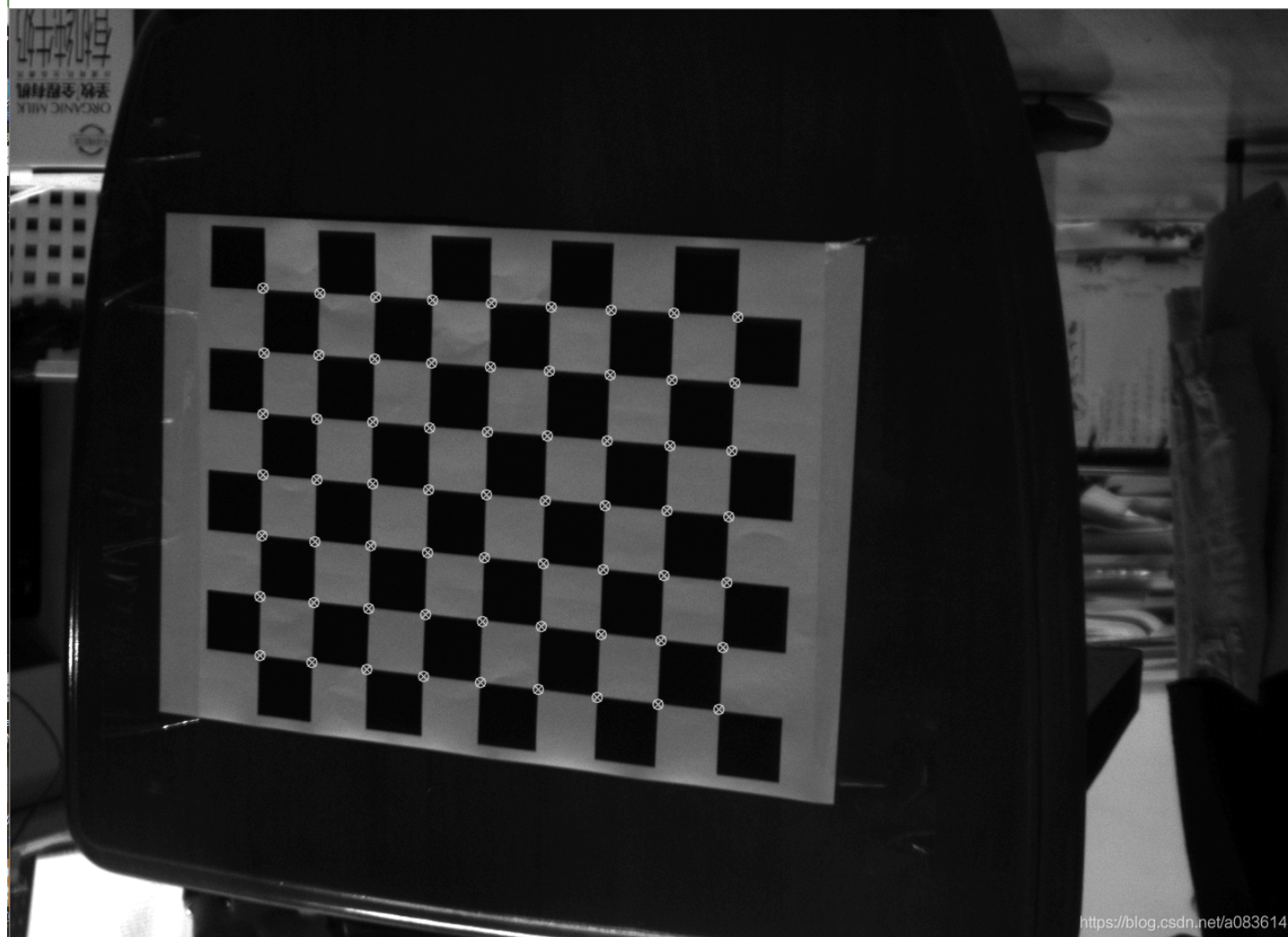
6、可查看标定结果和程序




```
编辑器 - Untitled*
Untitled* x +
14 'D:\VS\Calibration\calibrationTargetPicture\2017-10-25_20_50_45_097.bmp',...
15 'D:\VS\Calibration\calibrationTargetPicture\2017-10-25_20_50_57_065.bmp',...
16 'D:\VS\Calibration\calibrationTargetPicture\2017-10-25_20_51_13_327.bmp',...
17 'D:\VS\Calibration\calibrationTargetPicture\2017-10-25_20_51_23_125.bmp',...
18 'D:\VS\Calibration\calibrationTargetPicture\2017-10-25_20_51_30_276.bmp',...
19 'D:\VS\Calibration\calibrationTargetPicture\2017-10-25_20_51_37_751.bmp',...
20 'D:\VS\Calibration\calibrationTargetPicture\2017-10-25_20_51_46_927.bmp',...
21 'D:\VS\Calibration\calibrationTargetPicture\2017-10-25_20_52_09_122.bmp',...
22 };
23
24 % Detect checkerboards in images
25 [imagePoints, boardSize, imagesUsed] = detectCheckerboardPoints(imageFileNames);
26 imageFileNames = imageFileNames(imagesUsed);
27
28 % Generate world coordinates of the corners of the squares
29 squareSize = 25; % in units of 'mm'
30 worldPoints = generateCheckerboardPoints(boardSize, squareSize);
31
32 % Calibrate the camera
33 cameraParams = estimateCameraParameters(imagePoints, worldPoints, ...
34 'EstimateSkew', false, 'EstimateTangentialDistortion', false, ...
35 'NumRadialDistortionCoefficients', 2, 'WorldUnits', 'mm');
36
37 % View reprojection errors
38 h1=figure; showReprojectionErrors(cameraParams, 'BarGraph');
39
40 % Visualize pattern locations
41 h2=figure; showExtrinsics(cameraParams, 'CameraCentric');
42
43 % For example, you can use the calibration data to remove effects of lens distortion.
44 originalImage = imread(imageFileNames{1});
45 undistortedImage = undistortImage(originalImage, cameraParams);
46
47 % See additional examples of how to use the calibration data. At the prompt type:
48 % showdemo('MeasuringPlanarObjectsExample')
49 % showdemo('SparseReconstructionExample')
50
```

opencv标定

- 1、循环读取图片
- 2、使用findChessboardCorners函数检测角点（需提前输入角点数）。

<https://blog.csdn.net/a083614>

- 3、使用find4QuadCornerSubpix函数对角点进行亚像素精确化
- 4、可用drawChessboardCorners将角点显示。
- 5、根据角点数和尺寸创建一个理想的棋盘格（用point向量存储所有理论上的角点坐标）。
- 6、通过calibrateCamera函数由理想坐标和实际图像坐标进行标定，可得到标定结果。
- 7、由projectPoints函数计算反向投影误差。

```
开始标定.....标定完成!
开始评价标定结果.....
每幅图像的标定误差:
第1幅图像的误差: 4.041像素 第1幅图像的单角点平均误差: 0.064像素
第2幅图像的误差: 4.109像素 第2幅图像的单角点平均误差: 0.065像素
第3幅图像的误差: 4.930像素 第3幅图像的单角点平均误差: 0.078像素
第4幅图像的误差: 4.821像素 第4幅图像的单角点平均误差: 0.077像素
第5幅图像的误差: 4.403像素 第5幅图像的单角点平均误差: 0.070像素
第6幅图像的误差: 4.390像素 第6幅图像的单角点平均误差: 0.070像素
第7幅图像的误差: 4.956像素 第7幅图像的单角点平均误差: 0.079像素
第8幅图像的误差: 5.388像素 第8幅图像的单角点平均误差: 0.086像素
第9幅图像的误差: 5.523像素 第9幅图像的单角点平均误差: 0.088像素
第10幅图像的误差: 4.135像素 第10幅图像的单角点平均误差: 0.066像素
第11幅图像的误差: 4.192像素 第11幅图像的单角点平均误差: 0.067像素
第12幅图像的误差: 3.875像素 第12幅图像的单角点平均误差: 0.062像素
第13幅图像的误差: 4.783像素 第13幅图像的单角点平均误差: 0.076像素
第14幅图像的误差: 5.591像素 第14幅图像的单角点平均误差: 0.089像素
第15幅图像的误差: 4.355像素 第15幅图像的单角点平均误差: 0.069像素
第16幅图像的误差: 6.962像素 第16幅图像的单角点平均误差: 0.111像素
总平均误差: 4.778像素 总体单角点平均误差: 0.076像素
评价完成!
```

```
开始保存标定结果.....
相机内参数矩阵:
[2168. 2070837522692000, 0.0000000000000000, 622.0269815288818300;
 0.0000000000000000, 2174.0999387391389000, 487.0448693139729800;
 0.0000000000000000, 0.0000000000000000, 1.0000000000000000]

畸变系数:
[-0.0160153759520404, -1.8080310799364416, 0.0004931896335570, -0.0045382327818885, 15.5888625467851970]
https://blog.csdn.net/a083614
```

标定结果:

matlab标定的内参经过转置与VS标定的内参基本一致。

每一幅图像都有一个外参矩阵，它的作用是：通过将棋盘格上坐标系上的角点坐标 $(x,y,0,1)$ （计算过程中设定参考坐标系在棋盘格上左上方角点，Z轴垂直棋盘格平面，因此z为0,1表示齐次坐标）左乘外参矩阵再左乘内参矩阵可以得到该外参矩阵对应图像中的相应角点坐标。VS的标定结果经过代入点进行上述计算得到的坐标与图像基本一致，但matlab的结果不同直接代入并不相同。原因在于两种方法标定的结果表示形式不同，两种方法结果中外参矩阵的平移向量相同，但旋转矩阵结果不一致，估计是内部实现不同。ps:平时事情比较多，更新比较慢。

ps：如上图opencv标定结果中内参矩阵里的2168和2174并不是焦距，而是焦距/每个像素大小，否则数量级不会这么大，焦距一般只有几毫米或几十毫米。刚看到国内的硕士论文里有人写错了。