

摄像机标定工具箱

1.1 Matlab 摄像机标定工具箱

工具箱下载: http://www.vision.caltech.edu/bouguetj/calib_doc/download/index.html

说明文档: http://www.vision.caltech.edu/bouguetj/calib_doc/

安装: 将下载的工具箱文件 toolbox_calib.zip 解压缩, 将目录 toolbox_calib 拷贝到 Matlab 的目录下。

采集图像: 采集的图像统一命名后, 拷贝到 toolbox_calib 目录中。命名规则为基本名和编号, 基本名在前, 后面直接跟着数字编号。编号最多为 3 位十进制数字。

1.1.1 标定模型

内参数标定采用的模型如式(1-1)所示, Brown 畸变模型式(1-2)所示。

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} k_x & k_s & u_0 \\ 0 & k_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c / z_c \\ y_c / z_c \\ 1 \end{bmatrix} = M_{in} \begin{bmatrix} x_{c1} \\ y_{c1} \\ 1 \end{bmatrix} \quad (1-1)$$

式中: (u, v) 是特征点的图像坐标, (x_c, y_c, z_c) 是特征点在摄像机坐标系的坐标, k_x, k_y 是焦距归一化成像平面上的成像点坐标到图像坐标的放大系数, k_s 是对应于图像坐标 u, v 的摄像机的 x, y 轴之间不垂直带来的耦合放大系数, (u_0, v_0) 是光轴中心点的图像坐标即主点坐标, (x_{c1}, y_{c1}) 是焦距归一化成像平面上的成像点坐标。 $k_s = \alpha_c k_x$, α_c 是摄像机的实际 y 轴与理想 y 轴之间的夹角, 单位为弧度。

$$\begin{cases} x_{c1d} = x_{c1}(1 + k_{c1}r^2 + k_{c2}r^4 + k_{c5}r^6) + 2k_{c3}x_{c1}y_{c1} + k_{c4}(r^2 + 2x_{c1}^2) \\ y_{c1d} = y_{c1}(1 + k_{c1}r^2 + k_{c2}r^4 + k_{c5}r^6) + k_{c3}(r^2 + 2y_{c1}^2) + 2k_{c4}x_{c1}y_{c1} \end{cases} \quad (1-2)$$

式中: (x_{c1d}, y_{c1d}) 是焦距归一化成像平面上的成像点畸变后的坐标, k_{c1} 是 2 阶径向畸变系数, k_{c2} 是 4 阶径向畸变系数, k_{c5} 是 6 阶径向畸变系数, k_{c3}, k_{c4} 是切向畸变系数, r 为成像点到摄像机坐标系原点的距离, $r^2 = x_{c1}^2 + y_{c1}^2$ 。

1.1.2 操作界面

将 Matlab 的当前目录设定为含有标定工具箱的目录, 即 toolbox_calib 目录。在 Matlab 命令窗口运行 calib_gui 指令, 弹出图 1 所示选择窗口。



图 1 内存使用方式窗口

图 1 窗口中, 具有两个选项, 分别是“Standard”和“Memory efficient”。如果点击选择“Standard”, 则将目录中的所有图像读入内存中, 所需内存较大。如果点击选择“Memory efficient”, 则将目录中的图像按照需要每次一幅图像读入内存中, 所需内存较小。在选择了

内存使用方式后，弹出标定工具箱操作面板。图 2 是选择“Standard”后弹出的标定工具箱操作面板。



图 2 标定工具箱操作面板

图 2 所示的标定工具箱操作面板具有 16 个操作命令键，其功能如下：

- (1) “Image names” 键：指定图像的基本名(Basename)和图像格式，并将相应的图像读入内存。
- (2) “Read names” 键：将指定基本名和格式的图像读入内存。
- (3) “Extract grid corners” 键：提取网格角点。
- (4) “Calibration” 键：内参数标定。
- (5) “Show Extrinsic” 键：以图形方式显示摄像机与标定靶标之间的关系。
- (6) “Project on images” 键：按照摄像机的内参数以及摄像机的外参数(即靶标坐标系相对于摄像机坐标系的变换关系)，根据网格点的笛卡尔空间坐标，将网格角点反投影到图像空间。
- (7) “Analyse error” 键：图像空间的误差分析
- (8) “Recomp. corners” 键：重新提取网格角点。
- (9) “Add/Suppress images” 键：增加/删除图像。
- (10) “Save” 键：保存标定结果。将内参数标定结果以及摄像机与靶标之间的外参数保存为 m 文件 Calib_results.m，存放于 toolbox_calib 目录中。
- (11) “Load” 键：读入标定结果。从存放于 toolbox_calib 目录中的标定结果文件 Calib_results.mat 读入。
- (12) “Exit” 键：退出标定。
- (13) “Comp. Extrinsic” 键：计算外参数。
- (14) “Undistort image” 键：生成消除畸变后的图像并保存。
- (15) “Export calib data” 键：输出标定数据。分别以靶标坐标系中的平面坐标和图像中的图像坐标，将每一幅靶标图像的角点保存为两个 tex 文件。
- (16) “Show calib results” 键：显示标定结果。

1.1.3 内参数标定

预先将命名为 Image1~Image20 的 tif 格式的 20 幅靶标图像保存在 toolbox_calib 目录中。当然，采集的靶标图像也可以采用不同的格式，如 bmp 格式、jpg 格式等。但应注意，用于标定的靶标图像需要采用相同的图像格式。摄像机的内参数标定过程，如下所述。

(1) 指定图像基本名与图像格式

在图 2 所示的标定工具箱操作面板点击“Image names”键，在 Matlab 命令窗口分别输入基本名 Image 和图像格式 t，出现下述对话内容：

```

Baseline camera calibration images (without number nor suffix): Image
Image format: (['r']='ras', 'b']='bmp', 't']='tif', 'p']='pgm', 'j']='jpg', 'm']='ppm') t
Loading image 1...2...3...4...5...6...7...8...9...10...11...12...13...14...15...16...17...18...19...20...
done

```

同时，在 Matlab 的图形窗口显示出 20 幅靶标图像，如图 3 所示。

Calibration images

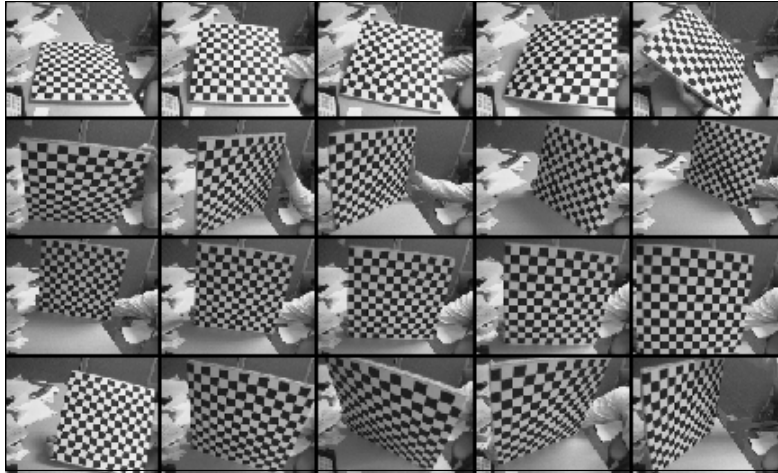


图3 靶标图像

(2) 提取角点

在图2所示的标定工具箱操作面板点击“Extract grid corners”键。

- 在 Matlab 命令窗口出现“Number(s) of image(s) to process ([] = all images) = ”时，输入要进行角点提取的靶标图像的编号并回车。直接回车表示选用缺省值。选择缺省值式，对读入的所有的靶标图像进行角点提取。
- 在 Matlab 命令窗口出现“Window size for corner finder (wintx and winty): ”时，分别在“wintx ([] = 5) = ”和“winty ([] = 5) = ”输入行中输入角点提取区域的窗口半宽 m 和半高 n 。 m 和 n 为正整数，单位为像素，缺省值为 5 个像素。选定 m 和 n 后，命令窗口显示角点提取区域的窗口尺寸 $(2n+1) \times (2m+1)$ 。例如，选择缺省角点提取区域的窗口尺寸为 11×11 像素。
- 在 Matlab 命令窗口出现“Do you want to use the automatic square counting mechanism (0=[]=default) or do you always want to enter the number of squares manually (1,other)? ”时，选择缺省值 0 表示自动计算棋盘格靶标选定区域内的方格行数和列数，选择值 1 表示人工计算并输入棋盘格靶标选定区域内的方格行数和列数。
- 到显示所选择靶标图像的图形窗口，利用鼠标点击设定棋盘格靶标的选定区域。点击的第一个角点作为靶标坐标系的原点，顺序点击 4 个角点形成四边形。注意，所形成的四边形的边应与棋盘格靶标的网格线基本平行。否则，影响角点提取精度，甚至导致角点提取错误。
- 在 Matlab 命令窗口出现“Size dX of each square along the X direction ([] = 100mm) = ”和“Size dY of each square along the Y direction ([] = 100mm) = ”时，分别输入方格长度和宽度，单位为 mm。方格长度和宽度的缺省值均为 100mm。
- 在 Matlab 命令窗口出现“Need of an initial guess for distortion? ([] = no, other = yes) ”时，如果选择 no 则不输入畸变初始值，如果选择 yes 则输入畸变初始值。输入的畸变初始值，将同时赋值给需要估计的 5 个畸变系数，即径向畸变系数 $kc(1)$ 、 $kc(2)$ 、 $kc(5)$ 和切向畸变系数 $kc(3)$ 、 $kc(4)$ 。如果不估计 6 阶径向畸变系数 $kc(5)$ ，则 $kc(5)$ 被赋值为 0。

按照上述步骤，对用于标定的每一幅靶标图像进行角点提取。例如， $m=5$ ， $n=5$ 时，角点提取区域的窗口尺寸为 11×11 像素，未输入畸变初始值，此时图像 Image6 的角点提取结果如图 4 所示。图 4(a) 只标出了待提取角点的位置，图 4(b) 标出了角点提取区域窗口和提取出的角点。从图 4 中可以发现，图 4(a) 中的十字标记位置与角点具有明显偏差，但在角点附

近：图 4(b)中的每个角点提取区域窗口包含了角点，表示角点提取结果的十字标记位置与角点位置具有很好的吻合度。同样在 $m=5, n=5$ 时，未输入畸变初始值，但通过鼠标点击设定棋盘格靶标的选定区域时，所形成的四边形的边与棋盘格靶标的网格线成较大夹角，此时图像 Image1 的角点提取结果如图 5 所示。从图 5 中可以发现，图 5(a)中的十字标记位置与角点具有明显偏差，部分十字标记远离角点；图 5(b)中的很多角点提取区域窗口没有包含角点，表示角点提取结果的十字标记位置并不在角点位置，说明角点提取存在错误。

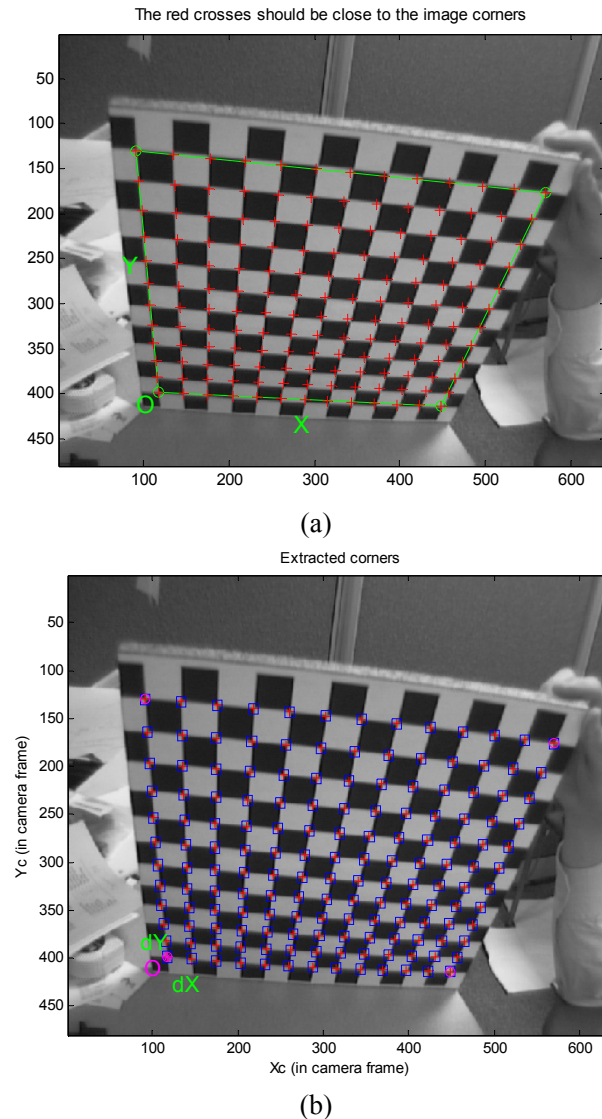


图 4 合适的靶标选定区域与角点提取结果，(a) 靶标选定区域，(b) 角点提取结果

(3) 内参数标定

对用于标定的每一幅靶标图像进行角点提取后，在图 2 所示的标定工具箱操作面板点击“Calibration”键，即可完成摄像机的内参数标定。

内参数标定时，Matlab 工具箱首先进行初始化，即将图像中心点坐标作为主点坐标的初始值，采用平面靶标网格的消失点估计出摄像机的内参数作为内参数的初始值，畸变初始值设为 0。镜头畸变采用包括径向畸变和切向畸变的 Brown 畸变模型，并假设 6 阶径向畸变系数 $kc(5)=0$ 。假设摄像机的 x 轴与 y 轴严格垂直，即图像坐标 (u, v) 与归一化成像平面内的成像点坐标 (x_{c1}, y_{c1}) 解耦， $k_s=0$ ，内参数采用 4 参数模型。数组 `est_dist(1:5)` 是畸变系数 $kc(1:5)$ 是否标定的标志，只对标志取值为 1 的畸变系数标定，标志取值为 0 的畸变系数不标定。

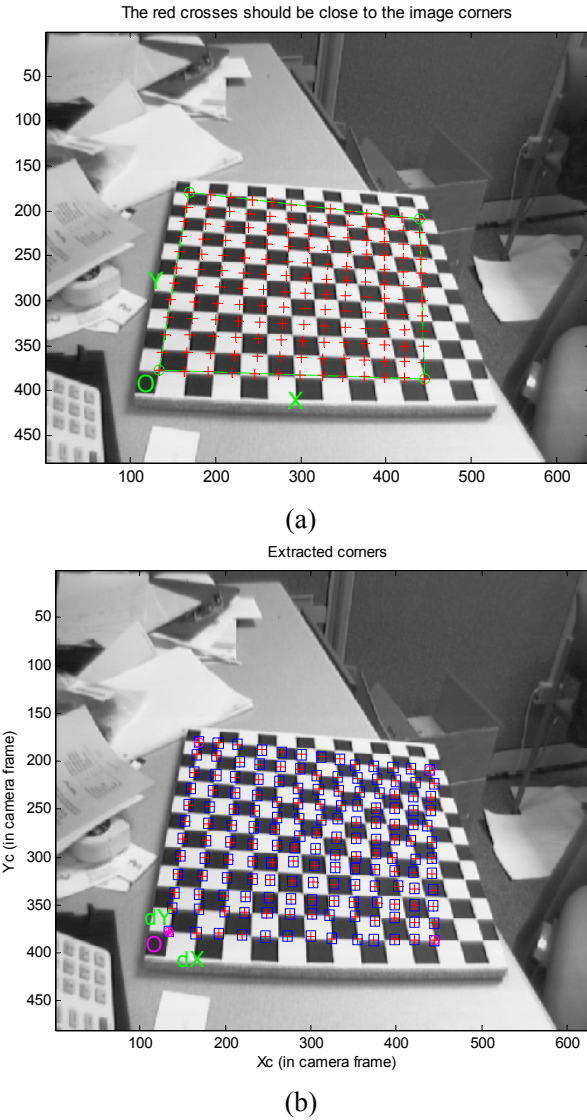


图 5 错误的靶标选定区域与角点提取结果, (a) 靶标选定区域, (b) 角点提取结果

内参数标定给出初始化后的标定结果和优化后的标定结果。其中, 对内参数的优化采用 L-M 梯度下降法。优化后的结果中给出的参数不确定性, 是 3 倍的标准方差。fc 中的两个数据分别是 k_x 和 k_y , 即焦距归一化成像平面上的成像点坐标到图像坐标的放大系数。cc 为光轴中心点的图像坐标(u_0, v_0), 又称为主点坐标, 单位为像素。alpha_c 是对应于图像坐标 v 的摄像机的实际 y 轴与理想 y 轴之间的夹角 α_c , 单位为弧度, 默认值为 0 弧度。后续给出的图像轴之间的夹角为对应于图像坐标 u 、 v 的摄像机的 x 、 y 轴之间的夹角, 默认值为 90° 。est_alpha 是 alpha_c 是否标定的标志位, 只有 est_alpha=1 时对 alpha_c 进行标定。kc 为畸变系数 $k_{c1} \sim k_{c5}$, kc(1)为二阶径向畸变系数 k_{c1} , kc(2)为 4 阶径向畸变系数 k_{c2} , kc(5)为 6 阶径向畸变系数 k_{c5} , kc(3)为图像坐标 u 对应于 xy 项的切向畸变系数 k_{c3} , kc(4)为图像坐标 v 对应于 xy 项的切向畸变系数 k_{c4} 。err 为将网格角点反投影到图像空间的误差的标准方差, 单位为像素。在优化后的结果中, 不确定性的数值越小, 说明标定的精度越高。如果不确定性项的数值与结果值相比所占比例较大, 则需要重新标定。

初始化后的标定结果:

Focal Length:	fc = [673.45516 673.45516]
Principal point:	cc = [319.50000 239.50000]

Skew: $\alpha_c = [0.00000] \Rightarrow$ angle of pixel = 90.00000 degrees
 Distortion: $kc = [0.00000 \quad 0.00000 \quad 0.00000 \quad 0.00000 \quad 0.00000]$

优化后的标定结果:

Focal Length: $fc = [657.80887 \quad 658.51372] \pm [1.86106 \quad 1.34683]$

Principal point: $cc = [302.95191 \quad 248.06759] \pm [1.88046 \quad 2.85817]$

Skew: $\alpha_c = [0.00000] \pm [0.00000] \Rightarrow$ angle of pixel axes = 90.00000 \pm 0.00000 degrees

Distortion: $kc = [-0.25853 \quad 0.14834 \quad 0.00074 \quad -0.00030 \quad 0.00000] \pm [0.00784 \quad 0.03727 \quad 0.00085 \quad 0.00042 \quad 0.00000]$

Pixel error: $err = [0.15205 \quad 0.12424]$

(4) 显示摄像机与标定靶标之间的关系

完成内参数标定后，在标定工具箱操作面板点击“Show Extrinsic”键，即可在新的图形窗口显示摄像机与标定靶标之间的关系，如图 6 所示。图 6(a)为假设摄像机固定时摄像机与靶标之间的关系，图 6(b)为假设靶标固定时摄像机与靶标之间的关系。

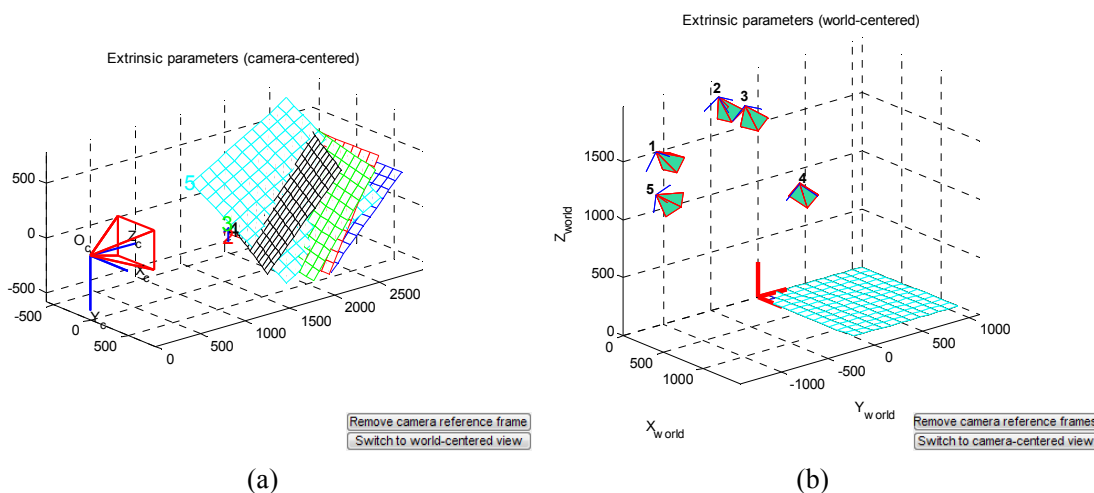


图 6 摄像机坐标系与靶标之间的关系，(a) 摄像机固定，(b) 靶标固定

(5) 误差分析

完成内参数标定后，在标定工具箱操作面板点击“Analyse error”键，即可在新的图形窗口显示出标定使用的所有角点反投影到图像空间的图像坐标误差，如图 7 所示。在图 7 所示的图形窗口，利用鼠标移动十字标尺可以选择角点，即可在命令窗口显示出该角点的信息，包括该角点所属图像、索引号、以方格为单位的坐标、图像坐标、反投影后的图像坐标误差、角点提取区域的窗口半宽 m 和半高 n 。

选择角点后在命令窗口显示的信息:

Selected image: 5

Selected point index: 51

Pattern coordinates (in units of (dX, dY)): (X,Y)=(11, 8)

Image coordinates (in pixel): (426.32, 261.58)

Pixel error = (-0.05908, 0.17309)

Window size: (wintx,winty) = (6, 6)

(6) 反投影到图像空间

完成内参数标定后，在标定工具箱操作面板点击“Project on image”键，在 Matlab 命令窗口出现“Number(s) of image(s) to process ([] = all images) =”时，输入要进行反投影的靶标图像的编号并回车。直接回车表示选用缺省值。选择缺省值式，对用于标定的所有靶标图

像进行反投影。选择图像后，在新的图形窗口显示反投影结果，并在命令窗口输出用于标定的所有靶标图像的角点反投影的图像误差的标准方差。Image6 反投影的结果如图 8 所示，其中“+”为角点的图像坐标位置，“o”为角点反投影的图像坐标位置。

点击“Project on image”键后，在命令窗口显示的信息如下：

Number(s) of image(s) to show ([] = all images) = 6

Pixel error: err = [0.66512 0.41428] (all active images)

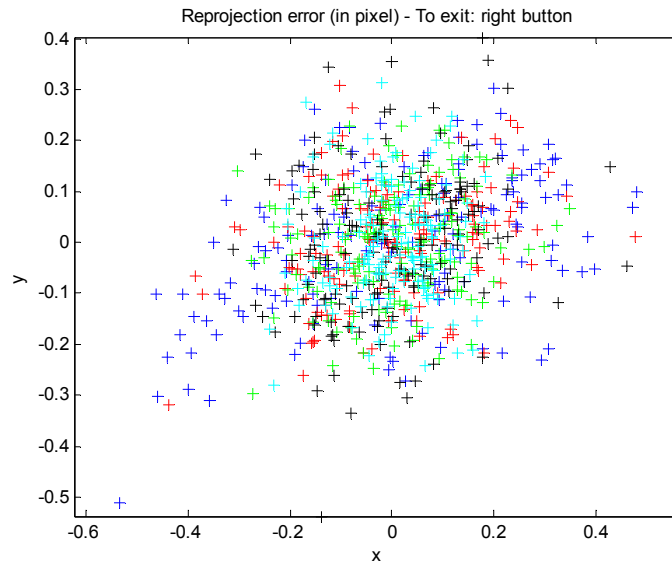


图 7 反投影到图像空间的角点图像坐标误差

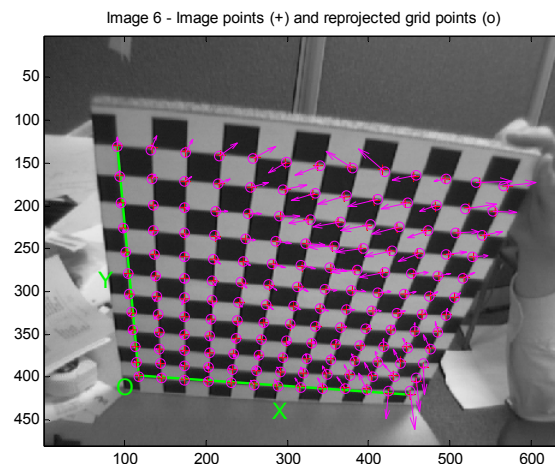


图 8 Image6 的角点反投影结果

(7) 图像畸变校正

完成内参数标定后，在标定工具箱操作面板点击“Undistort image”键，则按照畸变系数对读入的所有靶标图像进行处理，生成消除畸变后的图像并保存在 toolbox_calib 目录中。生成的消除畸变后的图像，以原图像的文件名在基本名和编号之间插入_rect 作为其文件名。图 9 所示为 Image6 的原始图像和消除畸变后的图像。

(8) 增加/删除图像

在标定工具箱操作面板点击“Add/Suppress image”键，在命令窗口输入代表增加图像、删除图像的数字，然后输入图像编号，可以增加/删除标定使用的图像。

(9) 重新提取网格角点

完成内参数标定后，在标定工具箱操作面板点击“Recomp. corners”键，在命令窗口输入角点提取区域的窗口半宽 m 和半高 n 、输入重新提取网格角点的图像编号、选择冲投影方式后，即可对所选图像重新进行角点提取。重新进行角点提取后，可以再次进行内参数标定。重新提取网格角点时采用了以前的标定结果，经过重新提取网格角点再次标定后，内参数标定精度会有所提高。如果初次角点提取的效果较好，则重新进行网格角点提取的作用不大。

点击“Recomp. corners”键后，在命令窗口显示的信息如下：

```
Re-extraction of the grid corners on the images (after first calibration)
Window size for corner finder (wintx and winty):
wintx ([]) = 5 =
winty ([]) = 5 =
Window size = 11x11
Number(s) of image(s) to process ([] = all images) =
Use the projection of 3D grid or manual click ([]=auto, other>manual):
Processing image 1...2...3...4...5...
done
```

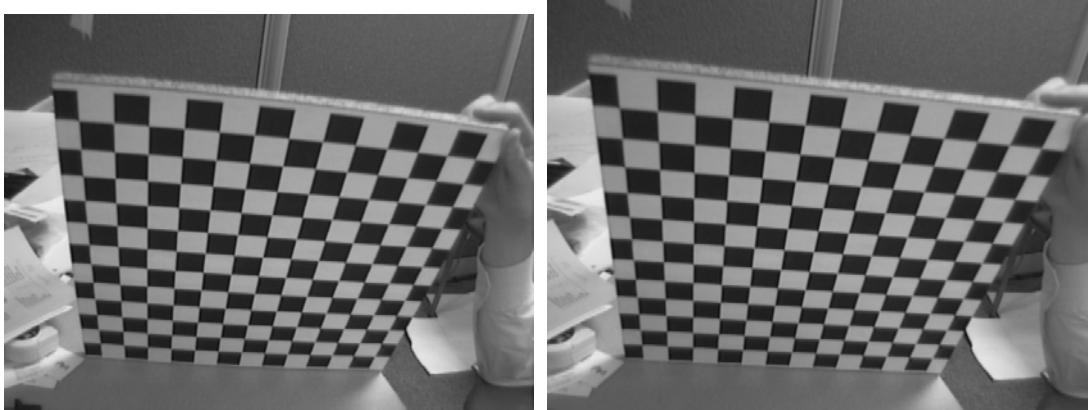


图 9 原始图像和校正后图像，(a) 原始图像 (b) 消除畸变后的图像

1.1.4 外参数标定

外参数标定是在内参数已知的前提下进行的。在完成摄像机的内参数标定后，或者在命令窗口输入摄像机的内参数后，可以对棋盘格靶标相对于摄像机的外参数进行标定。在标定工具箱操作面板点击“Comp. Extrinsic”键，在命令窗口输入靶标图像名称和图像格式，再按照 1.1.2 节(2)中的过程提取角点，可实现靶标相对于摄像机的外参数标定。

外参数标定时，在命令窗口显示的信息如下：

```
Computation of the extrinsic parameters from an image of a pattern
The intrinsic camera parameters are assumed to be known (previously computed)
Image name (full name without extension): Image1
Image format: ([]=r='ras', 'b'='bmp', 't'='tif', 'p'='pgm', 'j'='jpg', 'm'='ppm') t
Extraction of the grid corners on the image
Window size for corner finder (wintx and winty):
wintx ([]) = 5 =
winty ([]) = 5 =
Window size = 11x11
```


Click on the four extreme corners of the rectangular complete pattern (the first clicked corner is the origin)...

Size dX of each square along the X direction (l=30mm) = 100

Size dY of each square along the Y direction (l=30mm) = 100

Corner extraction...

外参数标定结果如下:

Translation vector: $Tc_ext = [-531.674715 \quad 400.142025 \quad 1999.135937]$

Rotation vector: $omc_ext = [2.258838 \quad -0.002845 \quad 0.157413]$

Rotation matrix: $Rc_ext = \begin{bmatrix} 0.992075 & -0.055515 & 0.112716 \\ & 0.051405 & -0.639246 & -0.767282 \\ & & 0.114649 & 0.766996 & -0.631327 \end{bmatrix}$

Pixel error: $err = [0.14406 \quad 0.10000]$

在外参数标定结果中, Tc_ext 为靶标坐标系原点在摄像机坐标系中的位移向量, 单位为 mm; omc_ext 为对应于姿态矩阵的 rodrigues 旋转向量; Rc_ext 为旋转矩阵; err 为将网格角点反投影到图像空间的误差的标准方差, 单位为像素。

1.1.5 立体视觉标定

在进行立体视觉标定之前, 按照 1.1.2 节方法分别标定立体视觉系统的左、右摄像机的内参数。左摄像机采集的图像与右摄像机采集的图像分别命名, 左摄像机采集的图像命名为 left1~left14, 右摄像机采集的图像命名为 right1~right14。左摄像机内参数的标定结果保存后的 Calib_results.mat 文件重命名为 Calib_Results_left.mat, 右摄像机的标定结果保存后的 Calib_results.mat 文件重命名为 Calib_Results_right.mat。

在 Matlab 命令窗口运行 stereo_gui 指令, 弹出图 10 所示立体视觉标定工具箱窗口。

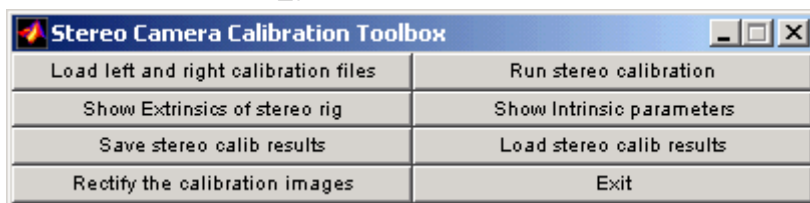


图 10 立体视觉标定工具箱操作面板

图 10 所示的标定工具箱操作面板具有 8 个操作命令键, 其功能如下:

(1) “Load left and right calibration files”键: 读入左、右摄像机的标定结果, 并对左摄像机相对于右摄像机的位姿进行初步标定。

在图 10 所示操作面板点击“Load left and right calibration files”键, 在 Matlab 的命令窗口输入左摄像机的标定结果文件名和右摄像机的标定结果文件名, 则在命令窗口显示下述内容:

Intrinsic parameters of left camera:

Focal Length: $fc_left = [533.00371 \quad 533.15260] \pm [1.07629 \quad 1.10913]$

Principal point: $cc_left = [341.58612 \quad 234.25940] \pm [1.24041 \quad 1.33065]$

Skew: $\alpha_c_left = [0.00000] \pm [0.00000] \Rightarrow$ angle of pixel axes = 90.00000 ± 0.00000 degrees

Distortion: $kc_left = [-0.28947 \quad 0.10326 \quad 0.00103 \quad -0.00029 \quad 0.00000] \pm [0.00596 \quad 0.02055 \quad 0.00030 \quad 0.00037 \quad 0.00000]$

Intrinsic parameters of right camera:

Focal Length: $fc_right = [536.98262 \quad 536.56938] \pm [1.19786 \quad 1.15677]$

Principal point: $cc_right = [326.47209 \quad 249.33257] \pm [1.36588 \quad 1.34252]$
 Skew: $alpha_c_right = [0.00000] \pm [0.00000] \Rightarrow$ angle of pixel axes = 90.00000 ± 0.00000 degrees
 Distortion: $kc_right = [-0.28936 \quad 0.10677 \quad -0.00078 \quad 0.00020 \quad 0.00000]$
 $\pm [0.00488 \quad 0.00866 \quad 0.00027 \quad 0.00062 \quad 0.00000]$
 Extrinsic parameters (position of right camera wrt left camera):
 Rotation vector: $om = [0.00611 \quad 0.00409 \quad -0.00359]$
 Translation vector: $T = [-99.84929 \quad 0.82221 \quad 0.43647]$

显示的结果中, fc_left 是左摄像机的放大系数, 即焦距归一化成像平面上的成像点坐标到图像坐标的放大系数。 cc_left 为左摄像机的主点坐标, 单位为像素。 $alpha_c_left$ 是对应于左摄像机的实际 y 轴与理想 y 轴之间的夹角, 单位为弧度, 默认值为 0 弧度。 kc_left 为左摄像机的畸变系数。 fc_right 是右摄像机的放大系数, 即焦距归一化成像平面上的成像点坐标到图像坐标的放大系数。 cc_right 为右摄像机的主点坐标, 单位为像素。 $alpha_c_right$ 是对应于右摄像机的实际 y 轴与理想 y 轴之间的夹角, 单位为弧度, 默认为 0 弧度。 kc_right 为右摄像机的畸变系数。 om 为左摄像机相对于右摄像机的姿态矩阵的 *rodrigues* 旋转向量, 利用函数 *rodrigues* 可以转换为姿态矩阵。 T 为左摄像机相对于右摄像机的位移向量, 即左摄像机坐标系原点在右摄像机坐标系中的位移向量, 单位 mm。

```
rodrigues(om) = [0.999983631582173    0.003519746065755    0.004510870798495
                -0.003489484865457    0.999971473651176   -0.006698908811843
                -0.004534320577757    0.006683058545923    0.999967387800907]
```

(2) “Run stereo calibration” 键: 计算优化后的外参数。

在图 10 所示操作面板点击“Run stereo calibration”键, 则在 Matlab 的命令窗口输出左、右摄像机的内参数和优化后的外参数。输出结果如下所示:

Intrinsic parameters of left camera:
 Focal Length: $fc_left = [533.52331 \quad 533.52700] \pm [0.83147 \quad 0.84055]$
 Principal point: $cc_left = [341.60377 \quad 235.19287] \pm [1.23937 \quad 1.20470]$
 Skew: $alpha_c_left = [0.00000] \pm [0.00000] \Rightarrow$ angle of pixel axes = 90.00000 ± 0.00000 degrees
 Distortion: $kc_left = [-0.28838 \quad 0.09714 \quad 0.00109 \quad -0.00030 \quad 0.00000]$
 $\pm [0.00621 \quad 0.02155 \quad 0.00028 \quad 0.00034 \quad 0.00000]$
 Intrinsic parameters of right camera:
 Focal Length: $fc_right = [536.81376 \quad 536.47649] \pm [0.87631 \quad 0.86541]$
 Principal point: $cc_right = [326.28655 \quad 250.10121] \pm [1.31444 \quad 1.16609]$
 Skew: $alpha_c_right = [0.00000] \pm [0.00000] \Rightarrow$ angle of pixel axes = 90.00000 ± 0.00000 degrees
 Distortion: $kc_right = [-0.28943 \quad 0.10690 \quad -0.00059 \quad 0.00014 \quad 0.00000]$
 $\pm [0.00486 \quad 0.00883 \quad 0.00022 \quad 0.00055 \quad 0.00000]$
 Extrinsic parameters (position of right camera wrt left camera):
 Rotation vector: $om = [0.00669 \quad 0.00452 \quad -0.00350] \pm [0.00270 \quad 0.00308 \quad 0.00029]$
 Translation vector: $T = [-99.80198 \quad 1.12443 \quad 0.05041] \pm [0.14200 \quad 0.11352 \quad 0.49773]$

(3) “Show Extrinsic of stereo rig” 键: 显示靶标相对于摄像机的位姿, 如图 11 所示。

(4) “Show Intrinsic parameters” 键: 在 Matlab 的命令窗口显示左、右摄像机的内参数和优化后的外参数。

(5) “Save stereo calib results” 键: 将标定结果保存为文件 *Calib_Results_stereo.mat*, 存放于 *toolbox_calib* 目录中。

(6) “Load stereo calib results” 键：读入标定结果。从存放于 toolbox_calib 目录中的标定结果文件 Calib_Results_stereo.mat 读入。

(7) “Rectify the calibration images” 键：按照畸变系数对左、右摄像机采集的所有靶标图像进行处理，生成消除畸变后的图像并保存在 toolbox_calib 目录中。生成的消除畸变后的图像，以原图像的文件名在基本名和编号之间插入_rectified 作为其文件名。

(8) “Exit” 键：退出立体视觉标定。

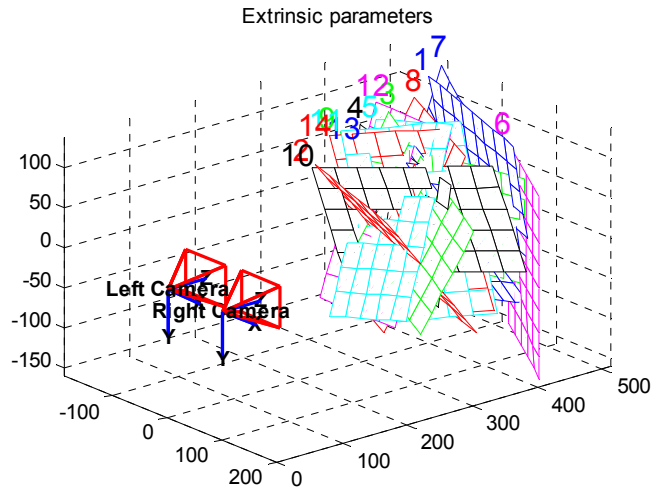


图 11 靶标相对于摄像机的位姿

1.1.6 标定注意事项

在采用 Matlab 工具箱的摄像机标定中，内参数标定是进行其它参数标定的基础。首先需要完成内参数标定，才能进行外参数标定或者立体视觉标定、手眼标定等。因此，内参数标定过程中需要注意的问题，在其它所有的标定中都应注意。

(1) 内参数标定需要注意的问题

- 制作棋盘格靶标时应特别注意，黑色方格与白色方格尺寸需要相同，而且所有方格的尺寸必须严格一致。靶标的方格数量不宜太小，行数和列数以大于 10 为宜。方格的尺寸不宜太大或太小，采集的整幅靶标图像中方格的边长尺寸不小于 20 像素。
- 采集靶标图像时应特别注意，需要在不同的角度不同的位置采集靶标的多幅图像。采集到的图像必须清晰，靶标图像尺寸以占整幅图像尺寸的 1/3~3/4 为宜。靶标图像最好在整幅图像的不同位置都有分布，不宜过于集中于同一区域。靶标放置位置与摄像机之间的距离最好为视觉系统的主要工作距离。靶标相对于摄像机的角度应有较大范围的变化，应包含绕三个轴较大角度的旋转，最好不小于 30 度。采集的靶标图像数量不应太少，建议以 10~20 幅靶标图像为宜。
- 采集图像过程中，摄像机的焦距不能调整。因为焦距属于摄像机的内参数，不同焦距下采集的图像隐含了不同的内参数，这些图像放在一起进行标定不能得到正确的结果。
- 采集的靶标图像统一命名，由基本名和编号构成，如 Image1~Image15。靶标图像的数据格式必须相同。
- 将靶标图像拷贝到 toolbox_calib 目录中。
- 提取角点时，在图形窗口利用鼠标点击设定棋盘格靶标的选定区域。点击的第一个角点作为靶标坐标系的原点，顺序点击 4 个角点形成四边形。相邻两次点击的角点

应在同一条网格线上，使得所形成的四边形的边应与棋盘格靶标的网格线基本平行。为提高点击的角点的精度，建议将显示靶标图像的图像窗口放大到最大，利用鼠标的十字标线尽可能准确的点击 4 个角点。

- 摄像机的实际 y 轴与理想 y 轴之间的夹角 α_c 是否标定，由 `est_alpha` 标志位设定。`est_alpha=1` 时对 `alpha_c` 进行标定，`est_alpha=0` 时不对 `alpha_c` 进行标定。
- 数组 `est_dist(1:5)` 是畸变系数 `kc(1:5)` 是否标定的标志，只对标志取值为 1 的畸变系数标定，标志取值为 0 的畸变系数不标定。默认值为 `est_dist(1:5)=[1 1 1 1 0]`，即对畸变系数 $k_{c1} \sim k_{c4}$ 进行标定，对 k_{c5} 不进行标定， $k_{c5}=0$ 。
- 运行 `calib_gui` 指令后，Matlab 处于 `busy` 状态，Matlab 命令窗口不再响应其它命令。只有在点击标定工具箱的“Exit”键退出标定后，Matlab 命令窗口才能恢复响应其它命令。

(2) 外参数标定需要注意的问题

- 方格尺寸必须输入实际尺寸。
- 提取角点时，在图形窗口利用鼠标点击的第一个角点作为靶标坐标系的原点，得到的外参数是靶标坐标系在摄像机坐标系中的位姿。
- `rodrigues` 旋转向量 `omc_ext` 与姿态矩阵 `Rc_ext` 可以利用 `rodrigues` 函数进行转换。`omc_ext=rodrigues(Rc_ext)`，`Rc_ext=rodrigues(omc_ext)`。

(3) 立体视觉标定需要注意的问题

- 提取角点时，在图形窗口利用鼠标点击的第一个角点作为靶标坐标系的原点，左右摄像机对应的靶标图像对需要选择相同的第一个角点作为原点。其他的 3 个角点在左右摄像机的靶标图像中也应相同。
- 左右摄像机采集的图像数量必须相同。相同的编号的左右摄像机采集的图像是靶标在同一位姿时左右摄像机采集的图像，构成一组立体视觉的靶标图像对。
- 得到的外参数是左摄像机相对于右摄像机的位姿，即左摄像机坐标系在右摄像机坐标系中的位姿。
- 运行 `stereo_gui` 指令后，Matlab 命令窗口可以响应其它命令。

(4) 手眼标定(Eye-in-Hand)需要注意的问题

- 首先进行摄像机的内参数标定。在摄像机内参数标定时，机器人可以不运动，通过改变靶标的位置和姿态采集 10~20 幅靶标图像。角点提取时，不需要具有相同的靶标坐标系原点。
- 然后进行外参数标定。靶标固定不动，较大幅度的改变机器人的位姿，采集 5~10 幅靶标图像。角点提取时，对采用的 5~10 幅靶标图像必须选择相同的靶标坐标系原点，并具有相同的靶标坐标轴方向。得到的外参数为靶标坐标系在摄像机坐标系中的位姿。
- 利用摄像机的内参数、机器人的末端位姿(或工具坐标系位姿)、对应于机器人位姿的靶标坐标系相对于摄像机坐标系的位姿，计算出摄像机坐标系相对于机器人末端坐标系(或工具坐标系)的位姿。具体计算方法可以利用最小二乘法。
- 机器人的位姿从机器人控制器中读取，一般为 6 维向量。其姿态部分一般为绕 X、Y、Z 轴的旋转角度，对应的 3 个旋转变换的相乘顺序可以查阅机器人说明书，或者通过特定的末端位姿进行验证。

1.2 OpenCV 摄像机标定工具箱

OpenCV 网站说明文档:

http://www.opencv.org.cn/opencvdoc/2.3.2/html/doc/tutorials/calib3d/camera_calibration/camera_calibration.html

http://docs.opencv.org/doc/tutorials/calib3d/camera_calibration/camera_calibration.html

OpenCV 中文网站摄像头标定: <http://wiki.opencv.org.cn/index.php/摄像头标定>

张正友标定算法: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=888718>

本文完整的程序代码(运行环境 VS2008+opencv2.3.1): <http://pan.baidu.com/s/1kTvCsb9>

1.2.1 标定模型

OpenCV 内参数标定采用的模型如式(1-3)所示, Brown 畸变模型如式(1-4)所示。

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} k_x & 0 & u_0 \\ 0 & k_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c / z_c \\ y_c / z_c \\ 1 \end{bmatrix} = M_{in} \begin{bmatrix} x_{c1} \\ y_{c1} \\ 1 \end{bmatrix} \quad (1-3)$$

式中: (u, v) 是特征点的图像坐标, (x_c, y_c, z_c) 是特征点在摄像机坐标系的坐标, k_x, k_y 是焦距归一化成像平面上的成像点坐标到图像坐标的放大系数, (u_0, v_0) 是光轴中心点的图像坐标即主点坐标, (x_{c1}, y_{c1}) 是焦距归一化成像平面上的成像点坐标。 $k_y = ak_x$, a 是纵横比系数。

$$\begin{cases} u'_d = u_d(1 + k_1r^2 + k_2r^4 + k_3r^6) + 2p_1u_dv_d + p_2(r^2 + 2u_d^2) \\ v'_d = v_d(1 + k_1r^2 + k_2r^4 + k_3r^6) + p_1(r^2 + 2v_d^2) + 2p_2u_dv_d \end{cases} \quad (1-4)$$

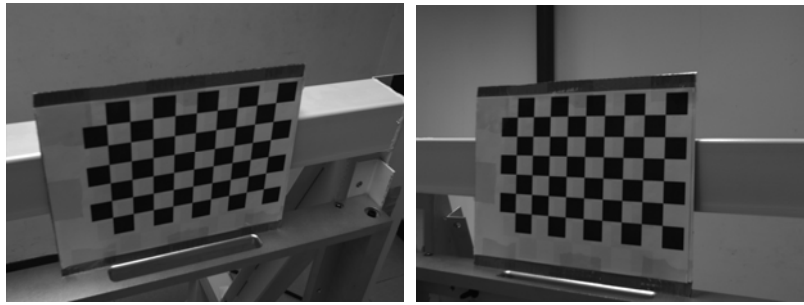
式中: (u_d, v_d) 是具有畸变相对于主点的图像坐标, $(u_d, v_d) = (u, v) - (u_0, v_0)$, (u'_d, v'_d) 是消除畸变后相对于主点的图像坐标, $(u'_d, v'_d) = (u', v') - (u_0, v_0)$, (u', v') 是消除畸变后的图像坐标, k_1 是 2 阶径向畸变系数, k_2 是 4 阶径向畸变系数, k_3 是 6 阶径向畸变系数, p_1, p_2 是切向畸变系数, r 为图像点到主点的距离, $r^2 = u_d^2 + v_d^2$ 。

标定结果给出的畸变系数为 $[k_1, k_2, p_1, p_2, k_3]$ 。由于 k_3 主要针对与畸变较大的镜头, 例如鱼眼镜头, 所以一般情况下该值取 0。

1.2.2 内参数标定

(1) 首先加载多幅棋盘格靶标图像并设置棋盘格靶标的内角点参数。标定过程中, 需要不同视角下拍摄的棋盘格靶标的图像, 如图 12 所示。其中, 内角点是指黑色方块相互联通位置的角点, 图 12 所示棋盘格靶标的内角点数量为 9×6 。

```
//设置棋盘格X方向上的角点个数
ChessBoardSize.width = 9;
//设置棋盘格Y方向上的角点个数
ChessBoardSize.hight = 6;
//设置棋盘格子的边长, 单位为mm
float SquareWidth = 30;
//利用cvLoadImage函数加载图像
chessBoard_Img=cvLoadImage( filename, 1 );
```



(a) (b)
图 12 不同视角下拍摄的棋盘格图像

(2) 对棋盘格靶标图像进行特征点提取。选用 `cvFindChessboardCorners` 函数提取一幅靶标图像的角点特征。当一幅靶标图像的角点正确提取后，再调用函数 `cvFindCornerSubPix` 可以进行亚像素图像特征的提取。依次对每一幅靶标图像进行角点特征提取。然后，调用 `cvDrawChessboardCorners` 函数，可以在棋盘格靶标图形上绘制出检测到的角点。绘制的图像如图 13 所示。

//获取棋盘格靶标中的角点，如果所有角点都被检测到且它们都被以一定顺序排布（一行一行地，每行从左到右），函数返回非零值，否则在函数不能发现或者记录所有角点的情况下，函数返回0。

`find_corners_result=cvFindChessboardCorners(current_frame_gray, //棋盘格的灰度图像，将chessBoard_Img进行转换得到`

`ChessBoardSize, //棋盘格每行角点的个数和每列角点的个数`
`&corners[captured_frames*NPoints], // 检测到的角点坐标`
`&corner_count[captured_frames], //检测到的角点的个数`
`CV_CALIB_CB_ADAPTIVE_THRESH); //使用自适应阈值`

//通过迭代方法求取具有子像素精度的角点位置

`cvFindCornerSubPix(current_frame_gray, //棋盘格靶标的灰度图像`
`&corners[captured_frames*NPoints], //检测到的角点坐标`
`NPoints, //单幅图像中角点的个数`
`cvSize(5,5),cvSize(-1,-1), //不忽略corner临近的像素进行精确估计，`
`cvTermCriteria(CV_TERMCRIT_ITER|CV_TERMCRIT_EPS,`
`30, //最大迭代次数（iteration）`
`0.01)); //最小精度（epsilon）`

//在棋盘格靶标图像上绘制检测到的角点

`cvDrawChessboardCorners(current_frame_rgb, //显示绘制结果的图像`
`ChessBoardSize, //棋盘格靶标中每行和每列角点的个数（9×6）`
`&corners[captured_frames*NPoints], //检测到的角点坐标`
`NPoints, //单幅图像角点的个数`
`find_corners_result); //角点检测成功标志`

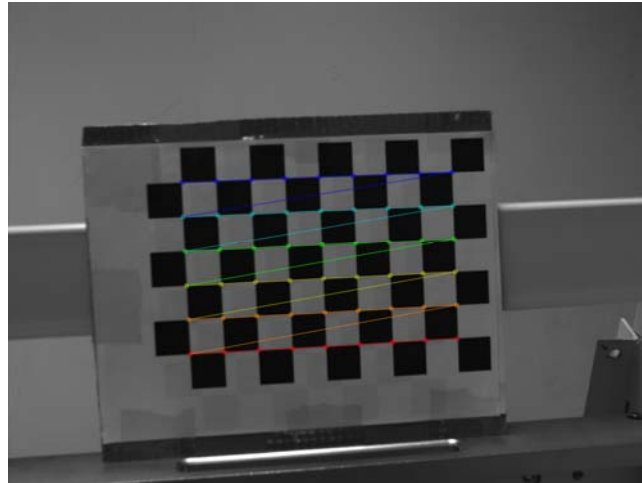


图 13 将角点绘制在棋盘格上的效果图

(3) 求取角点的三维世界坐标。每一幅图像检测的角点对应于相同的世界坐标系，世界坐标系以最左上方的角点为原点。例如图 13 中，最左上方的角点世界坐标为(0, 0, 0)，其右边相邻角点的世界坐标为(30, 0, 0)，其下方相邻角点的世界坐标为(0, 30, 0)。依次类推，可以得到所有角点的世界坐标。注意所有角点的 Z 坐标为 0，这是因为棋盘格靶标位于 Z=0 的平面上。

(4) 计算摄像机内参数。运行 cvCalibrateCamera2 函数实现摄像机内参数的标定。实际标定时，采集 10~20 幅棋盘格靶标图像可以很好地实现摄像机的参数标定。应当注意，多幅靶标图像必须在不同距离不同视角下进行采集。

```
//摄像机标定
cvCalibrateCamera2( object_points,//角点的三维世界坐标
    image_points,//角点的图像坐标，由corners[]转换得到
    point_counts,//角点的数目，由corner_count[]转换得到
    cvSize(image_width,image_height),//图像尺寸
    intrinsics,//输出的摄像机内参数
    distortion_coeff,//输出的摄像机畸变系数
    rotation_vectors,//摄像机外参数中的旋转向量
    translation_vectors,// 摄像机外参数中的平移向量
    0); //额外选项
```

利用图 12 所示的 10 幅不同角度和位置的靶标图像进行了内参数标定，结果为：

$$\begin{bmatrix} 1887.1410 & 0 & 756.0219 \\ 0 & 1896.0780 & 618.7198 \\ 0 & 0 & 1 \end{bmatrix}$$

畸变系数为：

$$[-0.1318 \quad 0.4704 \quad -0.0017 \quad -0.0027]$$

(5) 图像重投影。获得摄像机的内参数、畸变系数和变换矩阵后，利用函数 cvProjectPoints2 可以实现笛卡尔空间点到图像的重投影，从而在图像空间分析误差。

```

//投影三维点到图像平面
cvProjectPoints2(object_matrix,// 物体点的坐标, 为3xN或者Nx3的矩阵, 这里N
是单幅图中的所有角点的数目。
    rotation_matrix,// 旋转向量, 1x3或者3x1
    translation_matrix,// 平移向量, 1x3或者3x1。
    intrinsics,// 摄像机的内参数矩阵
    distortion_coeff,//畸变系数向量, 4x1或者1x4, 为[k1, k2, p1, p2]。如果是NULL,
所有畸变系数都设为0。
    image_matrix,// 输出重投影后的图像坐标数组
    0,0,0,0,0);

//计算误差
err=cvNorm(image_matrix,//重投影后图像坐标数组
    project_image_matrix, //角点的图像坐标数组
    CV_L2,// 欧几里德距离
    0);

```

(6) 获取消除畸变后的图像。经过摄像机标定, 获得了摄像机的内参数和畸变参数。利用摄像机的内参数和畸变参数, 运行 `cvUndistort2` 函数, 可以获得消除畸变后的图像。棋盘格模板的原始图像与消除畸变后的图像如图 14 所示。

```

//消除畸变
cvUndistort2(grayimage,//棋盘格靶标的灰度图像
    result_image,//消除畸变后的图像
    intrinsics,//摄像机内参数矩阵
    distortion_coeff);// 畸变系数向量

```

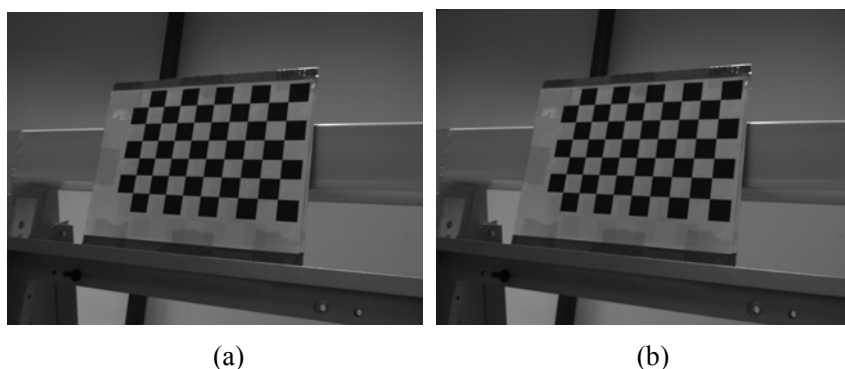


图 14 棋盘格模板图像, (a) 原始图像, (b) 消除畸变后图像

1.2.3 外参数标定

在内参数标定中, 通过 `cvCalibrateCamera2` 能够得到旋转向量和平移向量。其中, 平移向量的单位与输入的角度间的距离单位相同。利用罗德里格斯(Rodrigues)变换进行转换, 可以将旋转向量转换为旋转矩阵, OpenCV 提供了 Rodrigues 变换函数为 `cvRodrigues2`。

```

//罗德里格斯(Rodrigues)变换

```



```

cvRodrigues2(&pr_vec, //输入的旋转向量
&pR_matrix, //输出的旋转矩阵
0);

```

上例中得到相对于其中一幅靶标图像的旋转向量为[-1.7121 1.8199 0.1880]，平移向量为[93.3122 44.8024 599.3773]，平移向量的单位为 mm。经过罗德里格斯(Rodrigues)变换后得到对应于该图像的靶标坐标系相对于摄像机坐标系的变换矩阵，即摄像机的外参数矩阵，如下所示：

$$\begin{bmatrix} 0.0240 & 0.0005 & -0.9997 & 93.3122 \\ -0.0005 & 1.0000 & 0.0005 & 44.8024 \\ 0.9997 & 0.0005 & 0.0240 & 599.3773 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

此外，若已知摄像机内参数和靶标上的角点坐标，利用 `cvFindExtrinsicCameraParams2` 函数可以求取摄像机相对于靶标的外参数。函数 `cvFindExtrinsicCameraParams2` 说明如下：

```

void cvFindExtrinsicCameraParams2( const CvMat* object_points, // 靶标角点在笛卡尔空间的坐标，为3xN或者Nx3的矩阵，N是视图中的角点个数
const CvMat* image_points, // 靶标角点的图像坐标，为2xN或者Nx2的矩阵
const CvMat* intrinsic_matrix, //摄像机的内参数
const CvMat* distortion_coeffs, //摄像机的畸变系数[k1,k2,p1,p2]
CvMat* rotation_vector, //输出的旋转向量
CvMat* translation_vector ); //输出的位移向量

```

1.2.4 标定注意事项

在采用 OpenCV 方法的摄像机标定中，需要注意以下几个问题。

- 在设置棋盘格角点数目时，应按照棋盘格的内角点数目进行设置。
- 基于 OpenCV 摄像机标定算法采用张正友的标定方法，该算法是基于 2D 模型的，如果棋盘摆放的不平整，会造成很大的影响。
- 采集靶标图像时应特别注意，需要在不同的角度不同的位置采集靶标的多幅图像。
- 采集到的靶标图像必须清晰，靶标图像尺寸以占整幅图像尺寸的 1/3~3/4 为宜。采集的靶标图像数量不应太少，建议以 10~20 幅靶标图像为宜。
- 靶标相对于摄像机的角度在 45 度左右较好，太大的角度对角点提取的精度影响比较大，太小的角度不利于摄像机参数的精确标定。
- 在进行外参数标定时，方格尺寸必须是实际尺寸。
- 靶标坐标系的原点默认在左上角的内角点。在手眼标定时，应保证机器人处于不同姿态采集到的靶标图像中的左上角的内角点是同一个角点。即求取靶标坐标系相对摄像机坐标系的外参数时，必须保证靶标坐标系原点相同，且靶标坐标轴方向相同。