

相机标定

1 相机标定基本原理

1.1 相机成像模型

目前大多数相机模型都是基于针孔成像原理建立的，因为针孔成像原理简单，并且能满足建模的要求。除此之外还有基于应用歪斜光线追踪法和近轴光线追踪法的成像模型^[1]。针孔成像虽然已经展示出了相机的成像原理，但是由于针孔成像是理想的物理模型，没有考虑相机本身的尺寸、镜头与相机轴心的偏斜等因素的影响，因此精度很低，不能满足工业机器视觉的要求。为了使相机模型能高精度的反应相机的实际成像过程，需要再针孔成像模型的基础上考虑镜头畸变等的因素。

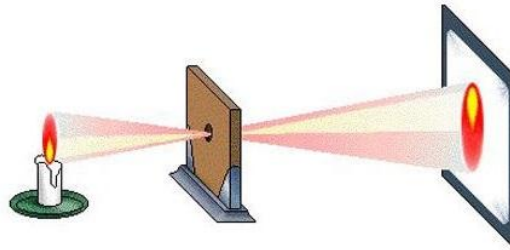


图 1 针孔成像

基于针孔成像原理建立的相机的成像模型，如下图所示。在相机的成像模型中，包含有几个坐标系分别是世界坐标系、相机坐标系、图像坐标系，相机的成像过程的数学模型就是目标点在这几个坐标系中的转化过程。

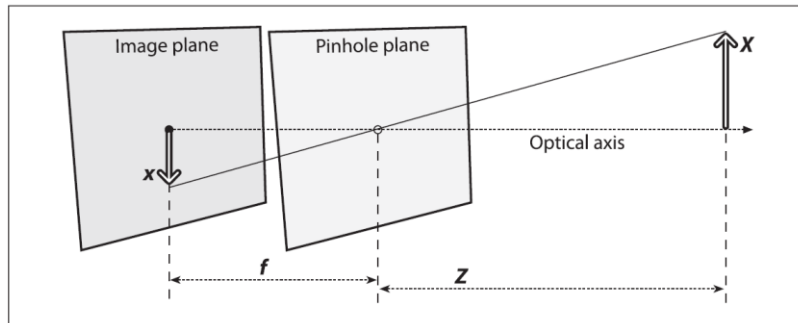


图 2 针孔成像模型

- (1) 世界坐标系(X_w, Y_w, Z_w)，就是现实坐标系，是实际物体在现实世界中的数学描述，是一个三维的坐标空间。
 - (2) 摄像机坐标系 (X_c, Y_c)，以针孔相机模型的聚焦中心为原点，以摄像机光学轴线为 Z_c 轴
 - (3) 图像坐标系：分为图像像素坐标系和图像物理坐标系
- 为了便于数学描述将图像平面移动到针孔与世界坐标系之间。如下图所示。

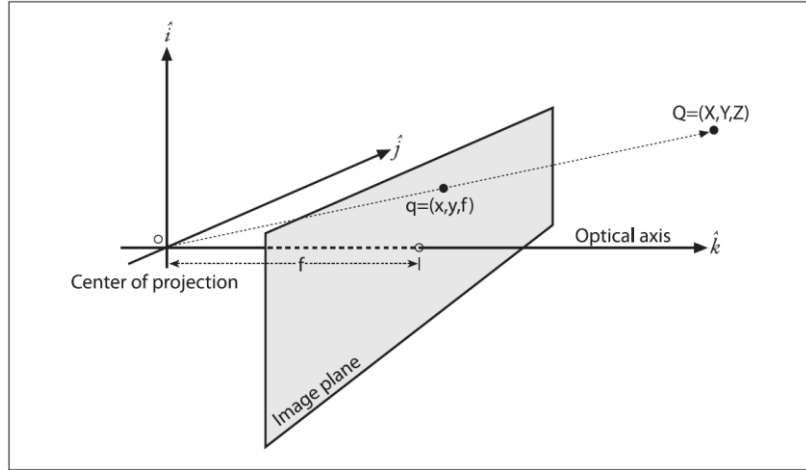


图 3 将相机平面移至针孔与目标物体之间的模型

1.2 坐标系间转换

从世界坐标系到相机坐标系：

$$P(X_c, Y_c, Z_c) = R(\alpha, \beta, \gamma) * P(X_w, Y_w, Z_w) + T$$

每一个世界坐标的对象都可以通过旋转和平移转移到相机坐标系上。将目标点旋转 θ 角度，等价于将坐标系方向旋转 θ 。如下图所示，是二维坐标的旋转变换，对于三维坐标而言，旋转中绕某一个轴旋转，原理实际与二维坐标旋转相同。如果，世界坐标分别绕 X, Y 和 Z 轴旋转 α, β, γ ，那么旋转矩阵分别为 $R(\alpha), R(\beta), R(\gamma)$

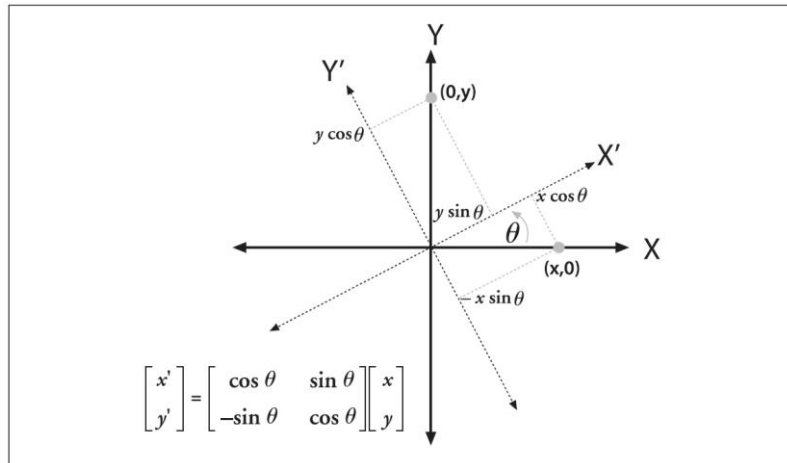


图 4 坐标旋转原理

$$R(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix} \quad (1-1)$$

$$R(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \quad (1-2)$$

$$R(\gamma) = \begin{bmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1-3)$$

总的旋转矩阵就是三者的乘积： $R(\alpha, \beta, \gamma) = R(\alpha) * R(\beta) * R(\gamma)$

平移矩阵 $T = (t_x, t_y, t_z)$, t_x, t_y, t_z 是世界坐标系原点与摄像机坐标系原点之间的差值。

相机坐标系到图像坐标系

$$X_d = f \frac{X_c}{Z_c}, \quad Y_d = f \frac{Y_c}{Z_c} \quad (1-4)$$

$$\begin{bmatrix} X_d \\ Y_d \end{bmatrix} = \frac{f}{Z_c} \begin{bmatrix} X_c \\ Y_c \end{bmatrix} \quad (1-5)$$

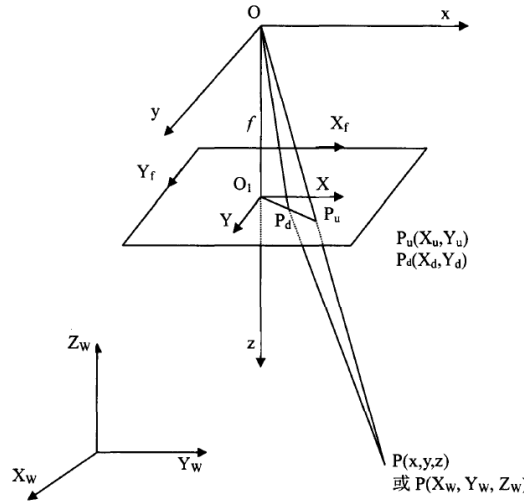


图 5 相机成像坐标变换原理图

图像物理坐标到图像像素坐标系

图形坐标系是一个二维坐标系，又分为图像像素坐标系和图像物理坐标系。图像像素坐标系 X_f - Y_f 是以图像左上角为原点，以图像互为直角的两个边缘为坐标轴，满足右手准则而建立的。图像是由一个个小的像素点组成的，图像像素坐标系的横纵坐标正是以像素点为单位，用来描述图像中每一个像素点在图像中的位置。图像物理坐标系以光轴与像平面交点为圆心建立的，图像物理坐标系的两个坐标轴分别与图像像素坐标系的坐标轴平行，并且方向相同，如图 3 所示。图像物理坐标系是以毫米为单位的直角坐标系 X - Y 。用 (X_f, Y_f) 来描述图像像素坐标系中的点，用 (X_d, Y_d) 来描述图像物理坐标系中的点。图像物理坐标系的原点 O 在图像像素坐标系中的坐标为 (C_x, C_y) ，用 dx, dy 来表示相邻像素点中心在 X 轴方向和 Y 轴方向的实际物理距离，则与图像像素坐标系的转化关系为：

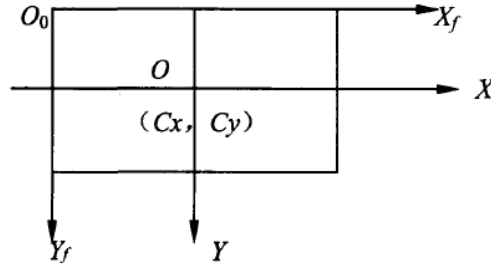


图 6 图像像素坐标系和图像物理坐标系

$$\begin{cases} X_f = \frac{X_d}{dx} + C_x \\ Y_f = \frac{Y_d}{dy} + C_y \end{cases} \quad (1-6)$$

1.4 相机畸变模型

相机畸变主要分为径向畸变和切向畸变，其中径向畸变是由透镜造成的，切向畸变是由成像仪与相机透镜的不平行造成的。

针孔模型是理想透镜的成像模型，但是实际中相机的透镜不可能是理想的模型，透镜形状的非理想特征造成像点会沿径向发生畸变。一个像点沿径向内缩叫负畸变,或桶形畸变沿径向外延叫正畸变,或枕形畸变。这种畸变相对于光轴严格对称的,也是畸变的主要分量^[2]。

(X_u, Y_u) 为矫正后的坐标, (X_d, Y_d) 是受到镜头失真影响而偏移的像平面坐标, 径向畸变模型:

$$X_u = X_d + X_d(k_1 r^2 + k_1 r^4 + \dots) \quad (1-7)$$

$$Y_u = Y_d + Y_d(k_1 r^2 + k_1 r^4 + \dots) \quad (1-8)$$

其中 k_1, k_2, \dots 为径向畸变系数, $r = \sqrt{X_d^2 + Y_d^2}$, 一般只需要考虑一阶或二阶就可以满足要求。

切向畸变模型:

$$X_u = X_d + P_1 X_d(3X_d^2 + Y_d^2) + 2P_2 X_d Y_d + O[(X_d, Y_d)^4] \quad (1-9)$$

$$Y_u = Y_d + P_2 X_d(3X_d^2 + Y_d^2) + 2P_1 X_d Y_d + O[(X_d, Y_d)^4] \quad (1-10)$$

其中 P_1 和 P_2 为切向畸变影响系数。

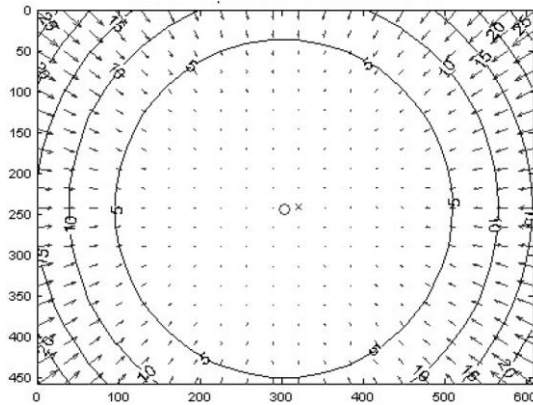


图 7 径向畸变

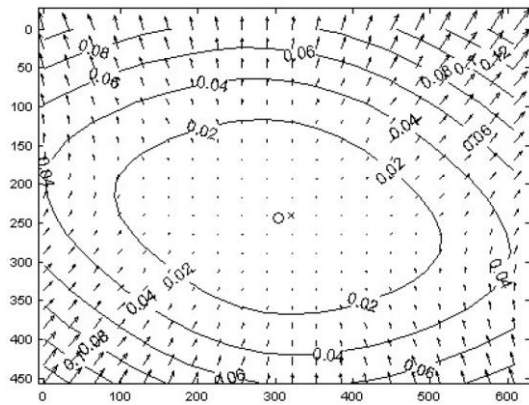


图 8 切向畸变

1.5 需要标定的参数

外部参数

由前面可知,摄像机的外部参数是用来描述摄像机坐标系与世界坐标系的关系,它表明摄像机在世界坐标系中的位置和方位,可用的旋转矩阵和平移向量来表示,如式一所示。其中由于为单位正交矩阵,必须满足个正交约束,故实质上旋转矩阵只有三个独立参数,加上平移向量的三个参数,故一共有个独立的外部参数。

内部参数

内部参数只与摄像机内部结构有关,而与摄像机位置参数无关,主要包括图像主点坐标 (c_x, c_y) , 单个像元的高宽 s_x, s_y , 摄像机的有效焦距 f 和透镜的畸变失真系数 k 等。

摄像机的内部参数有时也可以从制造商提供的说明书中查到,但是其精确性不能满足要求,仅可作为参考。实际应用中需要对它们进行标定。主点坐标 (c_x, c_y) ,理论上主点坐标一般位于图像中心处,但实际上由于摄像机制作和使用摄像机的镜头可转动和拆卸等原因,使得面阵安装并不能保证以透镜光轴为中心,且图像采集数字化窗口的中心不一定与光学中心重合,这就使得主点不一定在图像的帧存中心,故需要标定,,的值。

单个像元的高宽 s_x, s_y , 该数据可以在制造商提供的技术文档中查到, 但是该数据不是完全准确的。单个像元的高宽理论上应该是相等的,但是由于制造的误差,两者不可能完全相等,因此需要根据实际情况对其进行修正。

透镜的畸变失真系数前面讲到理想的透镜成像才满足线性关系,实际上透镜存在多种非线性畸变,需要根据实际情况对它们进行修正。

1.6 计算方法

张正友标定法、Tsai 两步标定法。

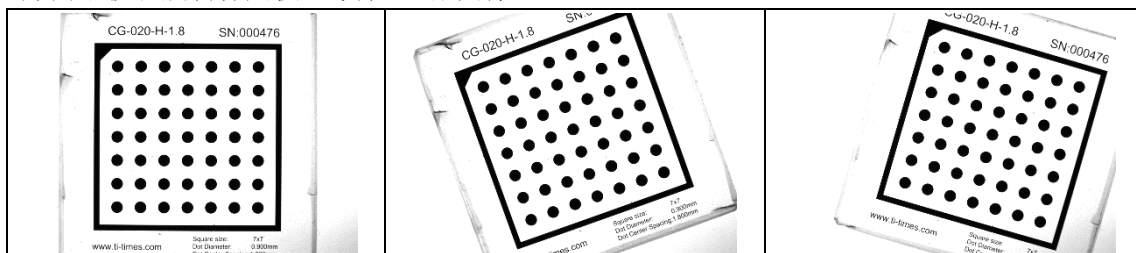
参考文献:

- 1 宋旗桂. 应用歪斜光线追踪法及近轴光线追踪法于照相机校正[D]. 台湾: 国立成功大学, 2007.
- 2 邹凤娇. 摄像机标定及相关技术研究[D]. 成都: 四川大学, 硕士学位论文, 2005.

1.7 目前设备中采用过的标定方法

1、目前在 RFID 设备、LCR 设备中标定都只采用一张, 忽略相机畸变。通过图像处理, 提取标定板圆心的间距除以像素值得到。该方法得到像素 X 方向和 Y 方向采用同一距离, 忽略其误差。

2、Halcon 的三维标定, 是 halcon 默认的标定方法, 消除径向畸变和视角畸变, 在 XYZ 方向平移和旋转标定板, 采集 18 张图像



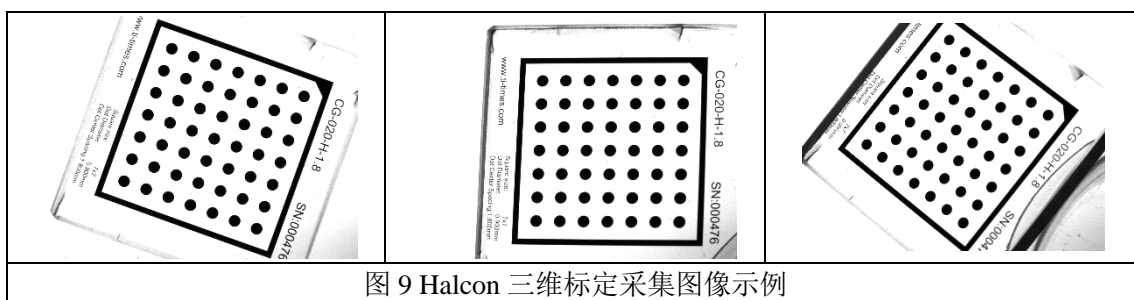


图 9 Halcon 三维标定采集图像示例

3、Halcon 的二维标定，只在 XY 平面内平移和旋转标定板，采集 18 张图像，减小 Z 的影响提高标定精度

实验数据：

	图像坐标 X 方向像素物理距离	图像坐标 Y 方向像素物理距离
方法一	15.5618um	15.5618um
方法二	15.7575um	15.7308um
方法三		

采用方法一每个像素距离与方法二有 0.2um 的差距，当距离较小时，这个误差可能表现的并不明显，在目前的设备中基本能满足要求。

根据方法二知道 X 方向与 Y 方向像素物理差值是 0.027um。

存在的问题：

- 1、目前标定板相对视野范围显得有点大，会降低标定的精度。
考虑自己制作标定板，打印出来，但是遇到问题是，打印出来标定板大小与我设计的标定板大小有些许差距，可能是由于文件格式转换过程中导致像素改变使打印的大小出现偏差。
- 2、因尽量减小光圈，提高景深，提升标定精度。
后期采图时，尽量减小光圈。

2 使用 Halcon 标定助手进行标定

2.1 标定描述

机器视觉本质就是通过图像来获取三维世界的信息，然后基于该信息进行相应的图像处理手段，从中获取我们想要的信息。标定的过程实际是建立图像世界与三维世界位姿的关系，只有准确地建立了该关系，才能从图像准确得知三维世界的真实状态。但是镜头在拍摄物体时存在成像的畸变，镜头成像的畸变导致图像不能完全地反映真实的世界，对图像的后续处理会产生干扰。尤其对于定位和检测，需要更高精度的矫正。镜头成像的畸变分为径向畸变和切向畸变，相对切向畸变而言，径向畸变对图像影响较大，而径向畸变分为枕形畸变（Pincushion Distortion）和桶形畸变（Barrel Distortion），如图 3 所示。因此，相机标定要矫

正镜头的畸变，然后获取相机的内部参数和外部参数，建立起图像与三维世界的联系。

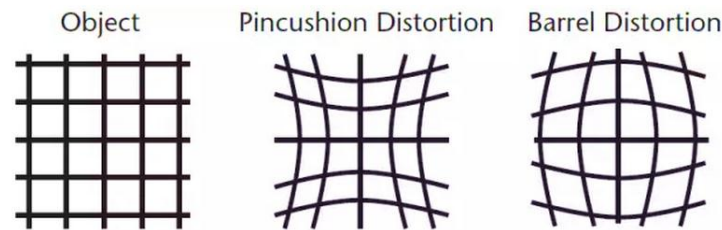


图 3 枕形畸变与桶形畸变

一个成功的标定，至少需要一个知道其精确尺寸的标定板。在标定前需要采集不同姿态的标定板图像，确保每张图像都完整的包含了标定板。标定图像的质量，很大程度上决定了标定是否能成功

2.2 标定板

标定的方法从原理上主要分为三大类：线性标定、非线性标定和二次标定。使用的标定板有棋盘格也有原点阵列，如图 4 所示。

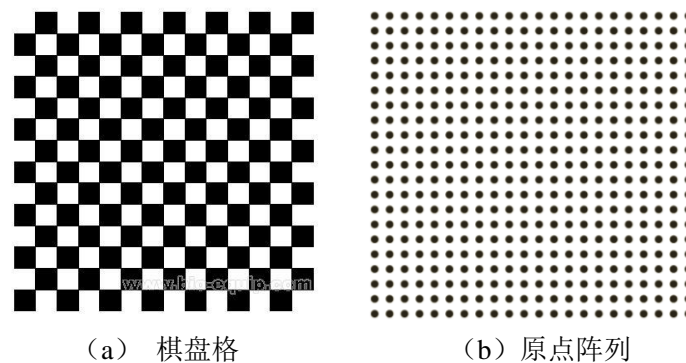


图 4 标定板

但是，halcon 并非必须使用其特制的标定板才能进行相机标定，但是如果要使用 Halcon 的标定算子进行标定，则必须使用 Halcon 特定的标定板，否则就需要自己去写标定板和特征点提取的算法了。Halcon 的标定板如图 5 所示，黑色的边框中有阵列分布的原点，在边框的左上角有个小三角用来确定其旋转的位置。

以 30*30mm 的标定板为例：
黑色原点：7*7
边框长度：30*30mm
内边框长度：28.125mm*28.125mm
黑色原点半径：0.9375mm
原点中心间距：3.75mm
裁剪宽度：30.75*30.75mm

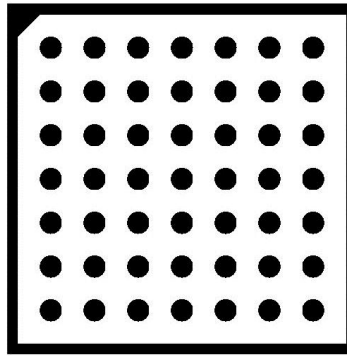


图 5 halcon 标定板

标定板制作：

Halcon 的安装后的文件中含有部分型号标定板的描述文件，但是由于实际应用中的标定板尺寸与系统描述文件不一致，因此需要定制相应的描述文件。

Halcon 中用 `gen_caltab(,XNum,YNum,MarkDist,DiameterRatio,CalTabDescrFile,CalTabPSFile,)` 算子来制作一个标定板

XNum	每行黑色标志圆点的数量。
YNum	每列黑色标志圆点的数量。
MarkDist	两个就近黑色圆点中心之间的距离。单位是 meter
DiameterRatio	黑色圆点直径与两圆点中心距离的比值。
CalTabDescrFile	标定板描述文件的文件路径（.descr）
CalTabPSFile	标定板图像文件的文件路径（.ps），

其中，.descr 文件为标定板描述文件，.ps 文件为标定板图形文件，可以用 photoshop(PS) 打开。在实际应用中使用的标定板都是采用光刻工艺制作而成，精度能达到 1um，自己制作的标定板则精度很低。

标定板选取：

标定板大小一般选取为相机视野大小的 1/3 左右是合适的。

2.3 相机标定流程

标定流程：

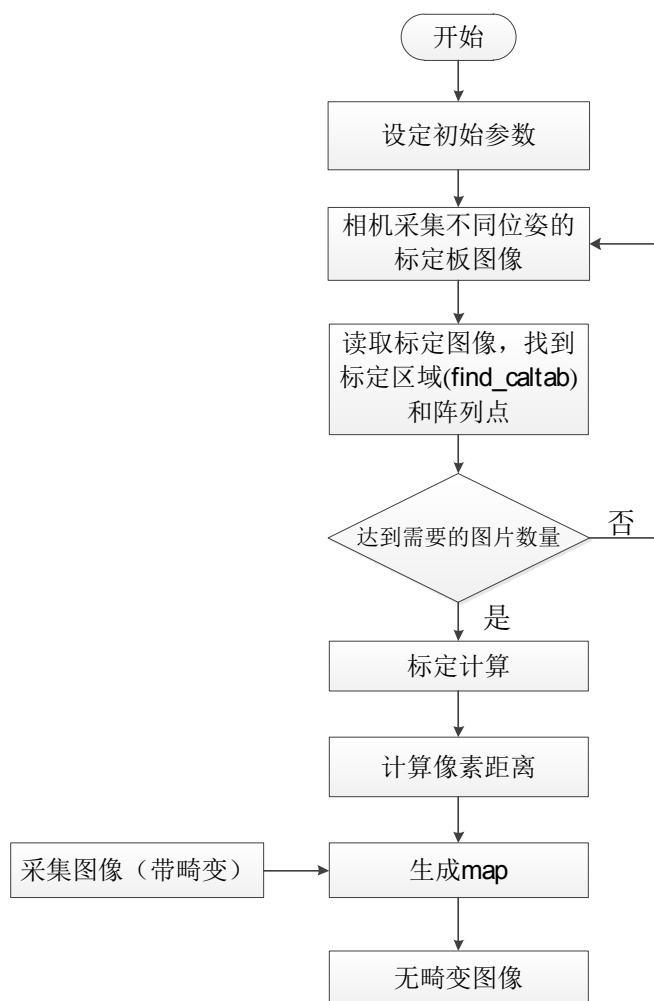
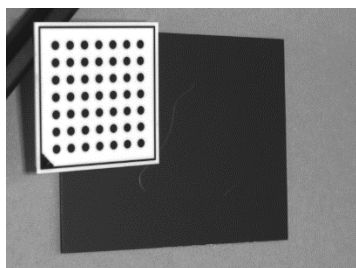


图 6 标定流程图

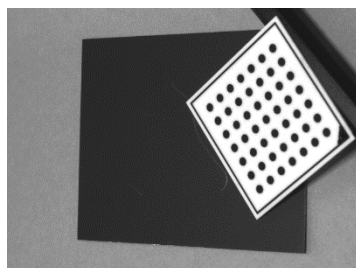
2.4 标定注意事项

标定图像采集注意点：

1. 标定板材质：光源在标定板前，可选陶瓷标定板；光源在后可选玻璃标定板，避免出现标定板反光的情况；
2. 标定板图像最好 10 幅以上，但是标定图像也并非越多越好，一般不超过 20 幅；
3. 标定图像采集过程中，相机的光圈不能发生改变，否则相机会被重新标定；
4. 图像覆盖标定板四角，标定板四个角要完全在图像内，标定板应该被放置在视野中间、边缘、四个角点处，还应使标定板适当的被旋转和倾斜；



(a)



(b)

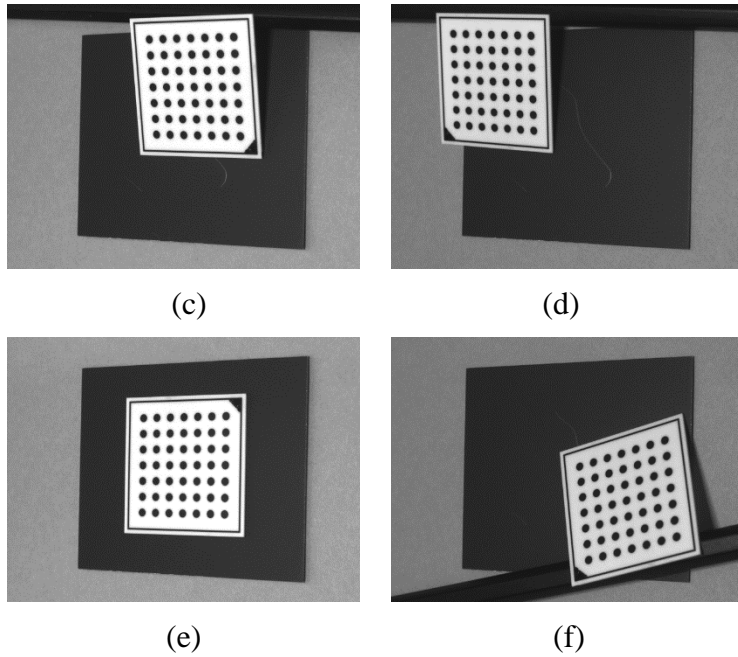
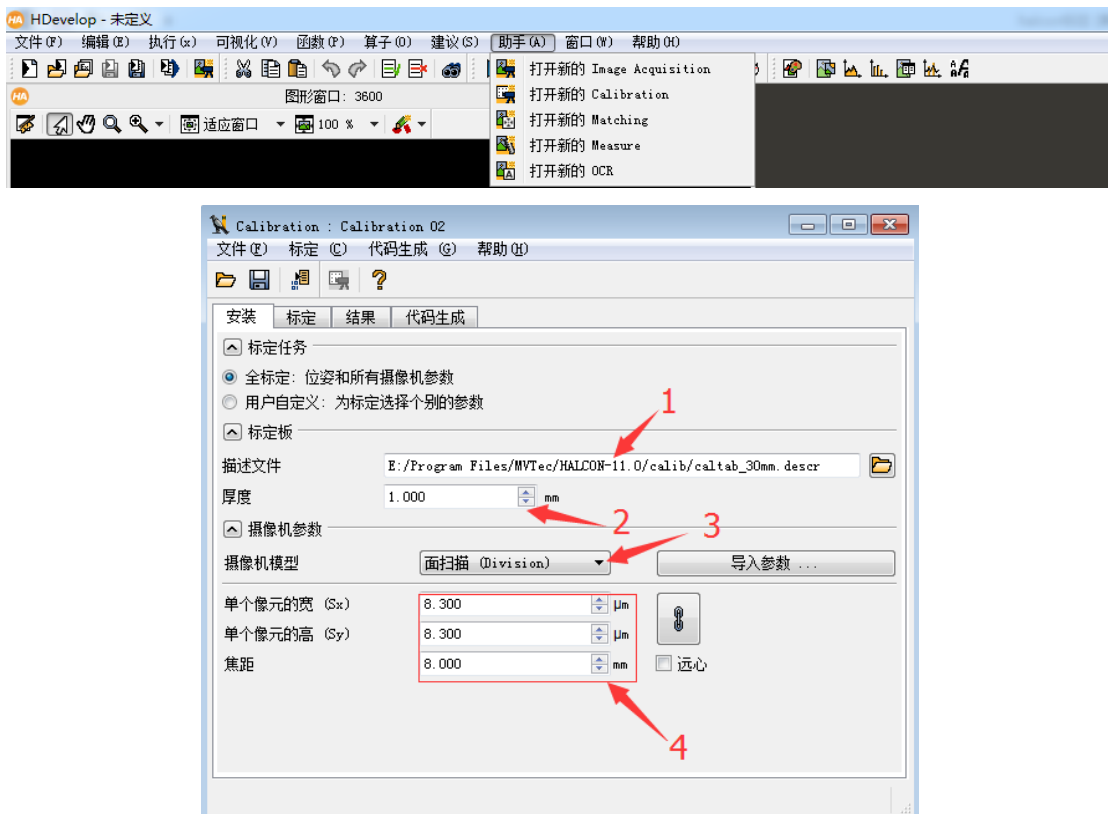


图 7 标定图像采集示例

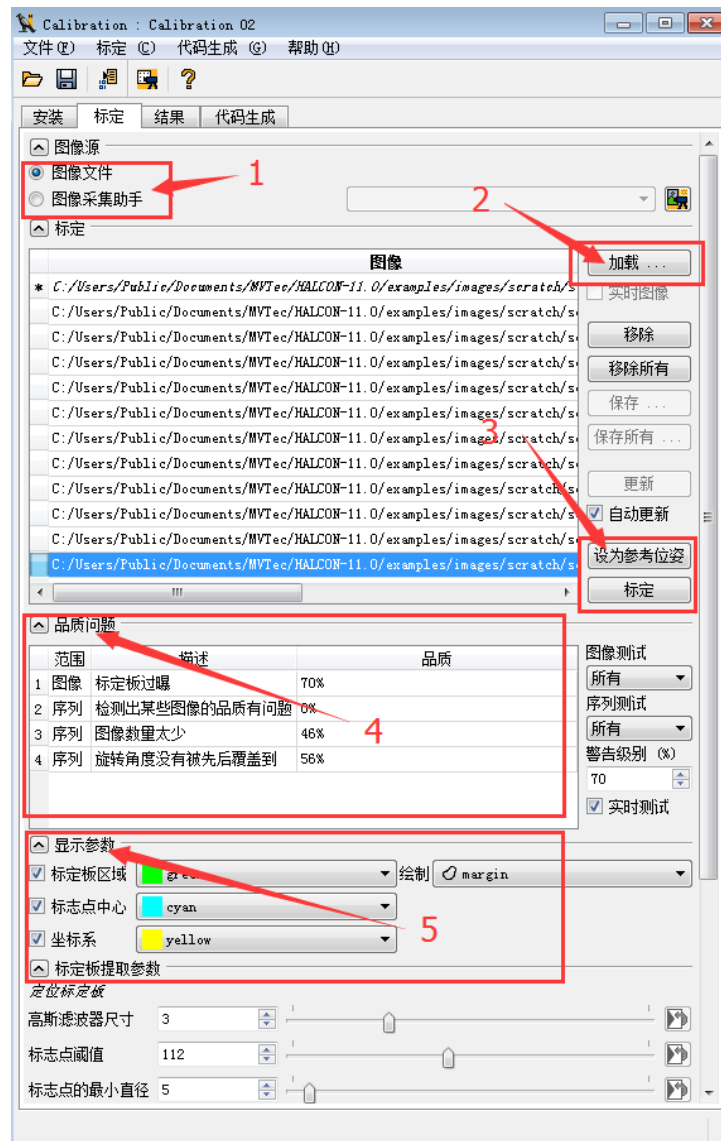
5. 图像灰度值差 100 以上;

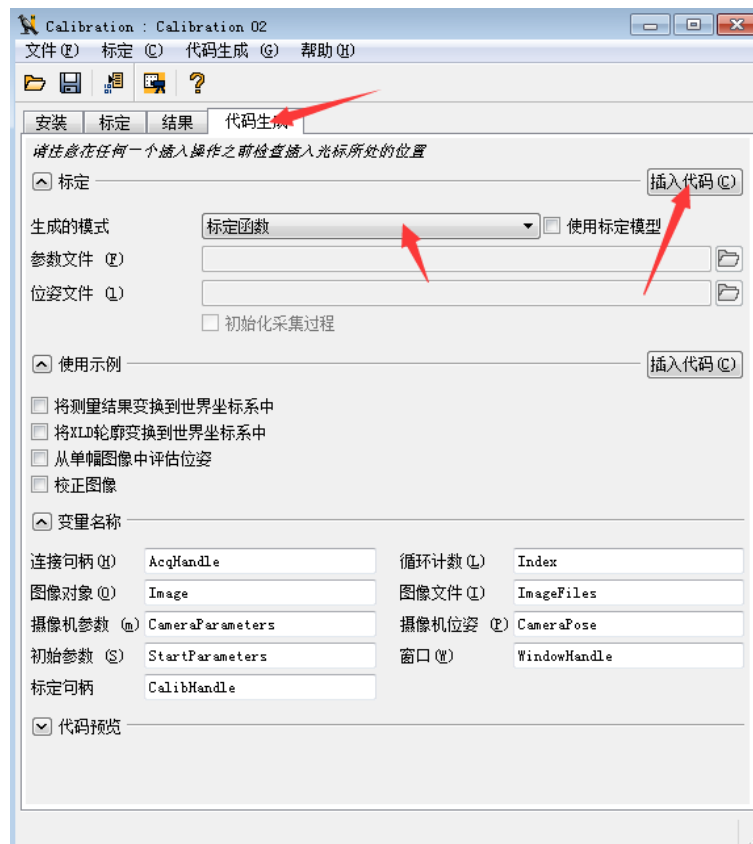
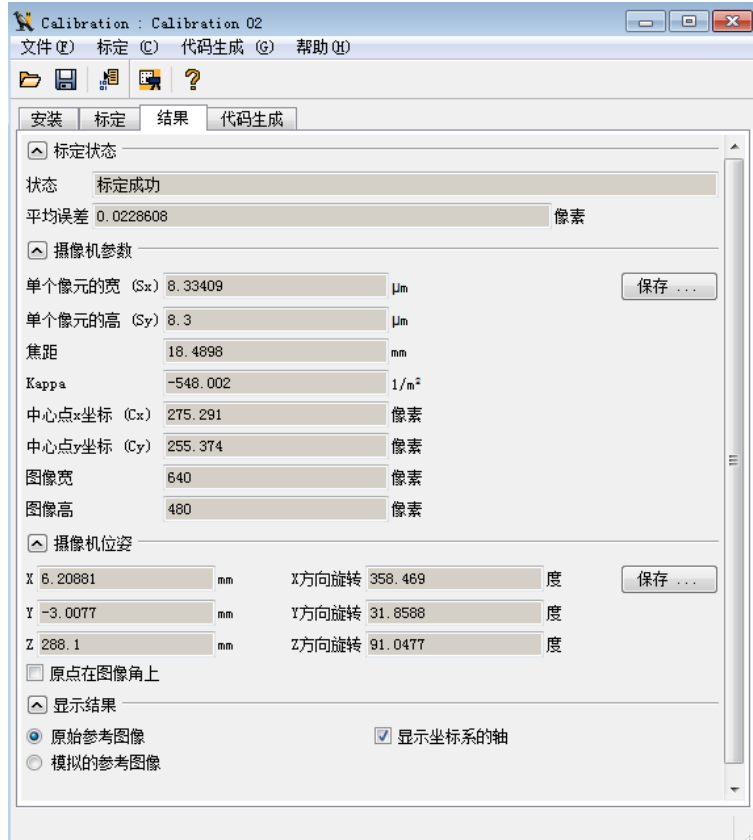
2.5 使用 halcon 标定助手标定过程:

Step1: 打开标定助手, 设定描述文件, 标定板厚度, 相机类型、焦距等参数。



Step2: 加载图像，可以实时采集，也可以采集好后再一起标定，建立先采图后标定。





2.5 写标定

采用 Halcon 标定助手生成的代码只能获取相机的内外参数，对于实际应用还需要进一步处理，完成的标定代码是更加复杂的，包含内外参数的获取、将图像坐标转换到世界坐标、像素实际物理距离、生成用于矫正的 Map 图像、将畸变图像矫正。

Halcon 相机标定代码：

*相机标定程序，包括相机内参与外参的获取、单个像素物理距离=实际距离/物理距离

*****设定相机参数*****

dev_close_window()

read_image (Image, 'scratch/scratch_calib_01.png')

get_image_size(Image, Width, Height)

dev_open_window(0, 0, Width, Height, 'black', WindowHandle)

dev_set_draw ('margin')

dev_set_line_width (1)

*创建标定数据（标定类型，相机数量，标定板数量，标定板模型句柄）

create_calib_data('calibration_object', 1, 1, CalibDataID)

*相机初始参数

StartCameraParam:=[0.012,0,0.0000055,0.0000055,Width/2, Height/2,Width, Height]

*设定相机参数

set_calib_data_cam_param (CalibDataID, 0, 'area_scan_division', StartCameraParam)

*指定标定描述文件

set_calib_data_calib_object(CalibDataID, 0, 'caltab_30mm.descr')

*****读取含标定板的图像*****

for i:=1 to 12 by 1

 read_image(Image,'scratch/scratch_calib_'+i\$'02d'+'.png') //i\$'02d'整数 i 转化为字符串

 *找到标定区域

 find_caltab(Image, CaltabRagion,'caltab_30mm.descr', 3, 112, 5)

 *find_calib_object()

 dev_set_color('green')

 dev_display (CaltabRagion)

 *参数解释（包含标定板的图像，标定板描述文件，相机初始参数，初始提取轮廓阈值 128，轮廓阈值减小步长 10，提取黑点轮廓滤波器参数 0.9，黑圆点最小轮廓

 *长度 15，黑圆点最大直径 100，圆心坐标，预估外参）

 find_marks_and_pose(Image, CaltabRagion, 'caltab_30mm.descr', StartCameraParam, 128, 10, 18, 0.9, 15, 100, RCoord, CCoord, StartPose)

 *StartPose 为粗略得到的外参（即位姿）

 dev_set_color('red')

 disp_cross (WindowHandle, RCoord, CCoord, 6, 0)

 *将标定数据保存在标定模型中

```

        set_calib_data_observ_points(CalibDataID, 0, 0, i, RCoord, CCoord, 'all', StartPose)
    *stop()
endfor

*****标定相机*****
*标定相机，注意 calibrate_cameras 与 camera_calibration 的区别
    calibrate_cameras(CalibDataID, Error)
**取相机内外参数
    get_calib_data(CalibDataID, 'camera', 0, 'params', CamParam)
    get_calib_data (CalibDataID, 'calib_obj_pose', [0,1], 'pose', PoseCalib)

*****保存相机内参，且该函数只能保存内参*****
    write_cam_par (CamParam, 'camparam.dat')

*****矫正图像径向畸变*****
read_image (SnapImage, 'scratch/scratch_calib_01.png')
get_image_size(SnapImage, Width1, Height1)
*赋值相机内参（带畸变），标定得到的数据
CamParIn := [0.012354, -1184.63, 5.53477e-006, 5.5e-006, 244.859, 254.739, 640, 480]
CarParamVirtualFixed:=CamParIn
CarParamVirtualFixed[1]:=0
*转化为无畸变内参,采用 adaptive 模式
change_radial_distortion_cam_par ('adaptive', CamParIn, 0, CarParamVirtualFixed)
*生成 map
gen_radial_distortion_map (Map, CamParIn, CarParamVirtualFixed, 'bilinear')
*write_image (Map, 'tiff', 0, MapFile)
map_image(SnapImage, Map, ImageMapped)

*****计算像素距离*****
read_image (DistImage, 'scratch/scratch_calib_01.png')
get_image_size (DistImage, Width2, Height2)
*赋值相机内参（带畸变），标定得到的数据
* CamParIn := [0.012354, -1184.63, 5.53477e-006, 5.5e-006, 244.859, 254.739, 640, 480]
* CarParamVirtualFixed:=CamParIn
* CarParamVirtualFixed[1]:=0
*转化为无畸变内参,采用 adaptive 模式
* change_radial_distortion_cam_par ('adaptive', CamParIn, 0, CarParamVirtualFixed)
*生成 map
* gen_radial_distortion_map (DistMap, CamParIn, CarParamVirtualFixed, 'bilinear')
* map_image(DistImage, DistMap, ImageMapped)

```

*矫正后再测量

*选取两个像素点

Image_X1:=100

Image_Y1:=100

Distance_XY:=500

Image_X2:=Image_X1 + Distance_XY

Image_Y2:=Image_Y1 + Distance_XY

*考虑标定板厚度的影响

set_origin_pose(PoseCalib, 0, 0, 0.001, PoseNewOrigin)

*将像素坐标转化为世界坐标（单位 m）

image_points_to_world_plane(CamParam, PoseNewOrigin, Image_Y1, Image_X1, 'm',
World_X1, World_Y1)

image_points_to_world_plane(CamParam, PoseNewOrigin, Image_Y2, Image_X2, 'm',
World_X2, World_Y2)

*计算世界坐标距离

distance_pp(World_Y1, World_X1, World_Y2, World_X2, DistanceWorld)

*计算像素坐标

distance_pp(Image_X1, Image_Y1, Image_X2, Image_Y2, DistanceImage)

*每毫米世界坐标距=距离对应的像素点个数

DistanceOneMilimeter:=DistanceImage/DistanceWorld

*每个像素对应的世界坐标距离

DistanceOnePixel:=DistanceWorld/DistanceImage

*调整世界坐标至中心偏移量

OffSetX:=(Width/2)*DistanceOnePixel

OffSetY:=(Height/2)*DistanceOnePixel

*****消除采集图像畸变*****

*真实图片测距

*读取图像，实际过程应该改为采集图像

read_image(ImagePerspective, 'scratch/scratch_perspective.png')

map_image(ImagePerspective, Map, ImageMapped1)

dev_display(ImageMapped1)

*调整相机外参

tuple_replace (PoseNewOrigin, 5, PoseCalib[5], PoseCalibRot)

*设定世界坐标

set_origin_pose(PoseCalibRot, -OffSetX, -OffSetY, 0, PoseNewOriginFinal)

```
*image_to_world_plane(ImagePerspective, ImageWorld, CamParam, PoseNewOriginFinal, Width, Height, DistanceOnePixel, 'bilinear')
```

*生成 map，消除径向畸变与视角畸变

```
gen_image_to_world_plane_map (Map1, CamParam, PoseNewOriginFinal, Width, Height, Width, Height, DistanceOnePixel, 'bilinear')
```

*矫正图像

```
map_image (ImagePerspective, Map1, ImageMapped2)
```

```
*****计算距离*****
```

```
dev_set_draw ('fill')
```

```
fast_threshold(ImageMapped2, Region, 20,80, 20)
```

```
fill_up(Region, RegionFillUp)
```

```
erosion_rectangle1(RegionFillUp, RegionErosion,20, 20)
```

```
reduce_domain(ImageMapped2, RegionErosion, ImageReduced)
```

```
var_threshold (ImageReduced, Region1, 5,5, 0.2, 5, 'light')
```

```
dilation_circle(Region1, RegionDilation, 3.5)
```

```
connection(RegionDilation, ConnectedRegions)
```

```
select_shape(ConnectedRegions, SelectedRegions, 'area', 'and', 1400,2000)
```

```
skeleton(SelectedRegions, Skeleton)
```

```
gen_contours_skeleton_xld(Skeleton, Contours, 1, 'filter')
```

```
length_xld(Contours, Length)
```

```
Legth:=Length*DistanceOnePixel
```

```
write_string(WindowHandle, '细线真实长度为：
```

相机标定不仅需要获取获取相机内参，而且需要将像素距离转化为世界坐标距离，得到单个像素的实际物理距离。

遇到的问题总结：

- 1) 使用 `image_points_to_world_plane` 时，由于未将单位设定为‘m’，导致使用算子 `gen_image_to_world_plane_map` 时，不能正确得生成映射图像，因为该算子默认的单位是‘m’。最后转换得到的实际物理距离的单位也是‘m’。
- 2) `gen_radial_distortion_map` 该算子只能矫正径向畸变，而 `gen_image_to_world_plane_map` 不仅能矫正径向畸变而且可以矫正视角的畸变，使图像绕着坐标轴旋转。
- 3) `calibrate_cameras` 与 `camera_calibrate` 的区别，应该尽量使用 `calibrate_cameras`，使用该算子可以很方便的修改相机的等的参数，而且 `halcon` 中所有的例程都使用的该算子。`camera_calibrate` 一次标定完成，不需要设定相机等的参数。

刘腾

2016年7月14日星期四

liutengpop@163.com