CVPR 第三次作业-相机标定

强基数学 002

吴天阳 2204210460，马煜璇 2204220461，陈博 2203623001

1 实验目的

1、掌握针孔相机成像的原理与特点；

2、掌握针孔成像系统中的坐标系（世界坐标、摄像机坐标、成像平面坐标、图像坐标）及各个坐标系之间关系；

3、掌握针孔相机模型中内参数与外参数的物理意义；掌握相机的经向畸变与切向畸变的特点及校正方法；

4、掌握基于平面模式的相机标定原理（张氏标定法）；

5、统计分析标定误差形成原因，思考提升标定参数精度的方法。

2 实验原理

2.1 针孔成像原理及特点

针孔成像原理：将三维欧式空间中点 $P(X,Y,Z)$, $(Z\neq 0)$，通过中心 O（相机镜头透镜）的射线与成像平面的交点 p，称为 P 的中心投影像点. 其中成像平面是不过原点 O 的平面，我们假设成像平面为 $z=1$，则通过向量等比例缩放可得投影像点为 $p'(X/Z,Y/Z,1)$.

针孔成像特点：

1. 近大远小：假设三维空间中高为 h 的柱子，上端点位于 $(X,Y+h,Z)$，下端点位于 (X,Y,Z)，则成像结果分别为 $(X/Z,(Y+h)/Z,1)$, $(X/Z,Y/Z,1)$，则成像平面上两点高度差为 $h'=h/Z$，若 Z 越大则 h' 越小，Z 越小则 h' 越大，即近大远小.

2. 平行线汇聚一点：设 A,B,C 是三维空间中的三条向量，α_1, α_2 分别从 A,B 出发沿着方向 C 传播，则 $\alpha_1(\lambda):A+\lambda C$，$\alpha_2(\lambda):B+\lambda C$，则 α_1,α_2 在成像平面上的投影分别为

$$\begin{aligned}\beta_1(\lambda)=&\left(\frac{A_x+\lambda C_x}{A_z+\lambda C_z},\frac{A_y+\lambda C_y}{A_z+\lambda C_z},1\right)\to\left(\frac{C_x}{C_z},\frac{C_y}{C_z},1\right),\ (\lambda\to\infty),\\ \beta_1(\lambda)=&\left(\frac{B_x+\lambda C_x}{B_z+\lambda C_z},\frac{B_y+\lambda C_y}{B_z+\lambda C_z},1\right)\to\left(\frac{C_x}{C_z},\frac{C_y}{C_z},1\right),\ (\lambda\to\infty).\end{aligned}$$

这说明两条射线在相平面上投影的极限区域趋于一点，且起点分别为 $\left(\dfrac{A_x}{A_z},\dfrac{A_y}{A_z},1\right)$ 和 $\left(\dfrac{B_x}{B_z},\dfrac{B_y}{B_z},1\right)$，所以两条平行线投影到相平面上不再保持平行，称为**消失点**.

3. 无限延长平面汇聚成一条线：设三维平面为 $N_xX+N_yY+N_zZ=d$，其中 (N_x,N_y,N_z) 是三维平面的法向量，则投影到二维平面上为

$$N_xX/Z+N_yY/Z+N_z=N_xx+N_yy+N_z=d/Z\to 0,\quad(\lambda\to\infty)$$

则其收敛到投影平面上的直线，称为**消失线**.

2.2 针孔成像系统中的坐标系

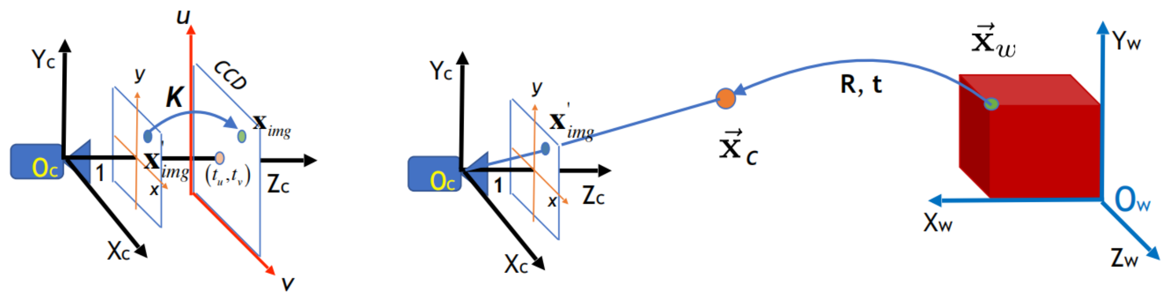


图 1: 四种坐标系转换

1. 世界坐标系：三维空间中建立的基准坐标系，用于标记空间点和摄像机位置，如图1中坐标系 O_w-$X_wY_wZ_w$ 所示.

2. 摄像机坐标系：以摄像机中心 O_c 作为原点，从 O_c 出发垂直于成像平面的射线作为主轴 Z_c，从 O_c 出发与像平面水平平行作为 X_c 轴，从 O_c 出发与像平面平行且铅垂直作为 Y_c 轴，如图1中坐标系 O_c-$X_cY_cZ_c$ 所示.

3. 成像平面坐标系（规范化坐标系）：假设焦距为 $f=1$，则成像平面为摄像机坐标系中 $Z_c=1$ 的平面，记成像平面与 Z_c 的交点为原点 p，以像平面水平线和铅垂线分别为 x 轴与 y 轴，如图1中坐标系 p-xy 所示.

4. 图像坐标系：原点 o 位于图像的左下角，平行于 x 轴作为 u 轴，平行于 y 轴作为 v 轴，如图1中坐标系 o-uv 所示.

这里需要引入**齐次坐标**概念，设三维坐标 $\boldsymbol{x}(X,Y,Z)$，则其对应的齐次坐标为四维坐标 $\boldsymbol{x}'(X,Y,Z,1)$，升维以后的齐次坐标能够通过矩阵完成平移操作，便于表示. 以下坐标定义均为齐次坐标.

我们记三维空间中物体的世界坐标为 $\boldsymbol{x}_w$，摄像机坐标为 $\boldsymbol{x}_c$（单位：m 或 mm），成像平面坐标为 $\boldsymbol{x}'_{img}$，图像坐标为 $\boldsymbol{x}_{img}$（单位：像素）.

则四个坐标系的转换关系如下图所示

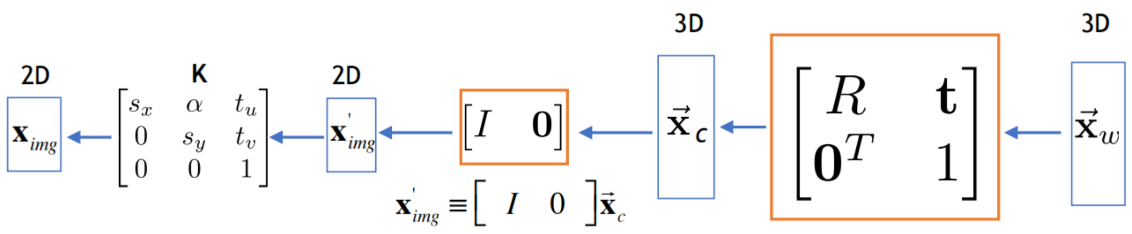


图 2: 四种坐标转换矩阵

上图变换主要包含三个矩阵：

1. 外参矩阵：4×4 矩阵，包括 3×3 的旋转矩阵 R 以及一个 3×1 的平移向量 $\boldsymbol{t}$. 该矩阵会随着物体或相机的移动而发生变化.

2. 投影矩阵：3×4 矩阵，取前三维分量，然后转化为齐次坐标，转化符号记为 $\equiv$. 注意，该过程会产生尺度因子 λ，即缩放比例. 如下式所示

$$[I \quad 0]\boldsymbol{x}_c = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} X_c & Y_c & Z_c \end{bmatrix} \equiv \begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \\ 1 \end{bmatrix} = \boldsymbol{x}'_{img}$$

其中 $[I \quad 0]\boldsymbol{x}_c = \lambda \boldsymbol{x}'_{img}$，$\lambda = 1/Z_c$.

3. 内参矩阵 K：3×3 矩阵，将像素从成像坐标系转化到图像坐标系中，由相机的自有特性决定，不随外部物体变化而变化. 其具体表示如下

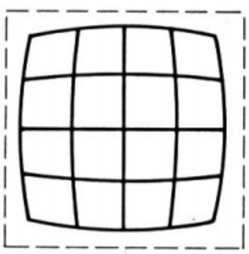
$$K = \begin{bmatrix} f_x & \alpha & t_u \\ 0 & f_y & t_v \\ 0 & 0 & 1 \end{bmatrix}$$

其中 (t_u, t_v) 表示图像坐标系原点在成像坐标系中所处的位置. 由于需要从单位 m 或 mm 转化为像素，所以还需要 f_x, f_y 分别表示沿 x 与 y 轴的缩放比例（相机焦距），由于 o-uv 不一定正交，所以还存在倾斜因子 α.

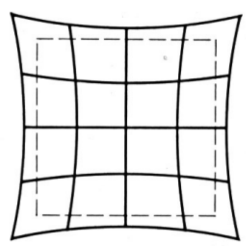
2.3 相机径向畸变与切向畸变的特点及矫正方法

在实际应用中，由于光学透镜不是完美的，通过透镜边缘的光线会发生偏转，导致偏离成像位置，发生扭曲，称为光学畸变. 光学畸变主要分为两种：

1. 径向畸变：主要由透镜问题产生，是一种非线性扭曲，畸变程度取决于与主点的距离，光线距离透镜中心越远，畸变效果更为严重. 有枕形畸变和桶形畸变两种.



(a) 桶形畸变



(b) 枕形畸变

2. 在摄像机安装过程中，光轴与成像平面无法完成平行，导致切向畸变.

两种畸变可用如下模型表示：

$$\begin{cases} x_d - x_c = \overbrace{(x_u - x_c)(1 + k_1 r^2 + k_2 r^4 + \cdots)}^{\text{径向畸变}} + \overbrace{\left[p_1(r^2 + 2(x_u - x_c)^2) + 2p_2(x_u - x_c)(y_u - y_c)\right](1 + p_3 r^2 + \cdots)}^{\text{切向畸变}} \\ y_d - y_c = (y_u - y_c)(1 + k_1 r^2 + k_2 r^4 + \cdots) + \left[p_2(r^2 + 2(y_u - y_c)^2) + 2p_1(x_u - x_c)(y_u - y_c)\right](1 + p_3 r^2 + \cdots) \end{cases}$$

（下述坐标均在成像坐标系中）其中 (x_u, y_u) 表示理想像素点坐标，(x_d, y_d) 表示畸变图像点坐标，(x_c, y_c) 表示畸变中心，k_n 为镜像畸变系数，p_n 为切向畸变系数，$r^2 = (x_u - x_c)^2 + (y_u - y_c)^2$.

令 $(x_c, y_c) = (0, 0)$，略去部分高阶项，可得简化版畸变模型

$$\begin{cases} x_d = x_u(1 + k_1 r^2 + k_2 r^4) + p_1(r^2 + 2x_u^2) + 2p_2 x_u y_u \\ y_d = y_u(1 + k_1 r^2 + k_2 r^4) + p_2(r^2 + 2y_u^2) + 2p_1 x_u y_u \end{cases} \tag{2.1}$$

通过对畸变参数 k_1, k_2, p_1, p_2 进行求解后，反解求得理想像素点坐标 (x_u, y_u).

2.4 张氏标定法

由于世界坐标系可以自由设定，我们不妨取三维平面中的黑白棋盘作为 xOy 平面，则棋盘上的点均有 $Z_w = 0$，由相机成像原理可知

$$x_{img} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \equiv K[R \quad \boldsymbol{t}]x_w = K[\boldsymbol{r}_1 \quad \boldsymbol{r}_2 \quad \boldsymbol{r}_3 \quad \boldsymbol{t}] \begin{bmatrix} X_w \\ Y_w \\ 0 \\ 1 \end{bmatrix} = K[\boldsymbol{r}_1 \quad \boldsymbol{r}_2 \quad \boldsymbol{t}] \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix} = H \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix}$$

其中 H 矩阵称为棋盘平面与图像平面之间的**单应性** (Homography) 变换矩阵，由于尺度因子的省略，H 矩阵对应了相机平面到**全体同一偏转角度棋盘平面**的一个同态映射.

通过大量的像素坐标以及对应的棋盘格坐标，可建立多个关于 H 的方程组，H 为 3×3 矩阵，但由于存在一个尺度因子，所以共有 8 个待定参数. 一组对应坐标可以得到两个方程，所以至少 4 组对应坐标，更多的对应坐标可以通过最小二乘法估计得到 H 矩阵. 下面通过 H 矩阵分别求解内参矩阵 K，外参矩阵以及径向畸变估计.

2.4.1 估计相机内参矩阵

$$H = [\boldsymbol{h}_1 \quad \boldsymbol{h}_2 \quad \boldsymbol{h}_3] = \lambda K[\boldsymbol{r}_1 \quad \boldsymbol{r}_2 \quad \boldsymbol{t}] \Rightarrow \begin{cases} \boldsymbol{r}_1 = \dfrac{1}{\lambda} K^{-1}\boldsymbol{h}_1, \\ \boldsymbol{r}_2 = \dfrac{1}{\lambda} K^{-1}\boldsymbol{h}_2. \end{cases} \tag{2.2}$$

由于 $\boldsymbol{r}_1 \perp \boldsymbol{r}_2$, $||\boldsymbol{r}_1|| = ||\boldsymbol{r}_2||$，于是

$$\begin{cases} \boldsymbol{h}_1^T (K^{-1})^T K^{-1} \boldsymbol{h}_2 = 0, \\ \boldsymbol{h}_1^T (K^{-1})^T K^{-1} \boldsymbol{h}_1 = \boldsymbol{h}_2^T (K^{-1})^T K^{-1} \boldsymbol{h}_2, \end{cases} \tag{2.3}$$

令

$$B = (K^{-1})^T K^{-1} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix}$$

由于 B 矩阵是对称阵，所以只需确定下三角部分 6 个参数，令

$$\boldsymbol{b} = [B_{11} \quad B_{12} \quad B_{22} \quad B_{13} \quad B_{23} \quad B_{33}]^T$$

则上述方程2.3可表示为方程 $\begin{cases} \boldsymbol{v}_{12}^T\boldsymbol{b} = 0, \\ [\boldsymbol{v}_{11}^T - \boldsymbol{v}_{22}^T]\boldsymbol{b} = 0. \end{cases}$ ，其中 $\boldsymbol{v}_{ij}$ 由 H 矩阵确定，具体形式请见https://zhuanlan.zhihu.com/p/136827980. 由于每种不同角度的棋盘图像可以确定不同的 H 矩阵，从而确定 $\boldsymbol{v}_{ij}$，于是每张不同角度的棋盘图像可以确定两个关于 $\boldsymbol{b}$ 的方程，总共有 6 个待定参数，所以需要至少 3 张不同角度的棋盘才能唯一确定内参矩阵.

2.4.2 估计相机外参矩阵

有了内参矩阵 K 以后，我们通过式2.2可知

$$\lambda[\boldsymbol{r}_1 \quad \boldsymbol{r}_2 \quad \boldsymbol{t}] = K^{-1}[\boldsymbol{h}_1 \quad \boldsymbol{h}_2 \quad \boldsymbol{h}_3]$$

则

$$\begin{cases} \lambda = ||K^{-1}\boldsymbol{h}_1||_2 = ||K^{-1}\boldsymbol{h}_2||_2, \\ \boldsymbol{t} = K^{-1}\boldsymbol{h}_3/\lambda, \\ \boldsymbol{r}_1 = K^{-1}\boldsymbol{h}_1/\lambda, \\ \boldsymbol{r}_2 = K^{-1}\boldsymbol{h}_2/\lambda, \\ \boldsymbol{r}_3 = \boldsymbol{r}_1 \times \boldsymbol{r}_2. \end{cases}$$

其中 $||\cdot||_2$ 为 2-范数（欧式范数），于是外参矩阵为 $[R \quad \boldsymbol{t}] = [\boldsymbol{r}_1 \quad \boldsymbol{r}_2 \quad \boldsymbol{r}_3 \quad \boldsymbol{t}]$.

在实际应用中，由于数据中存在噪音，R 矩阵不一定完全满足旋转矩阵的性质，所以通常使用奇异值分解求解 R.

2.4.3 最大似然估计

在实际标定过程中，一般存在大量标定图片同时对参数进行估计，这时使用最大似然估计对上述算法进行优化. 假设有 n 张标定图片，每张图片中都有 m 个棋盘格角点，且棋盘格大小一致，棋盘中角点相对位置相同. 利用平方损失函数，构造最小化风险函数如下

$$\min_{K,\boldsymbol{k},R,\boldsymbol{t},\lambda} \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}||x_{ij} - x'(K,\boldsymbol{k},R_i,\boldsymbol{t}_i,\lambda_i,X_j)||^2$$

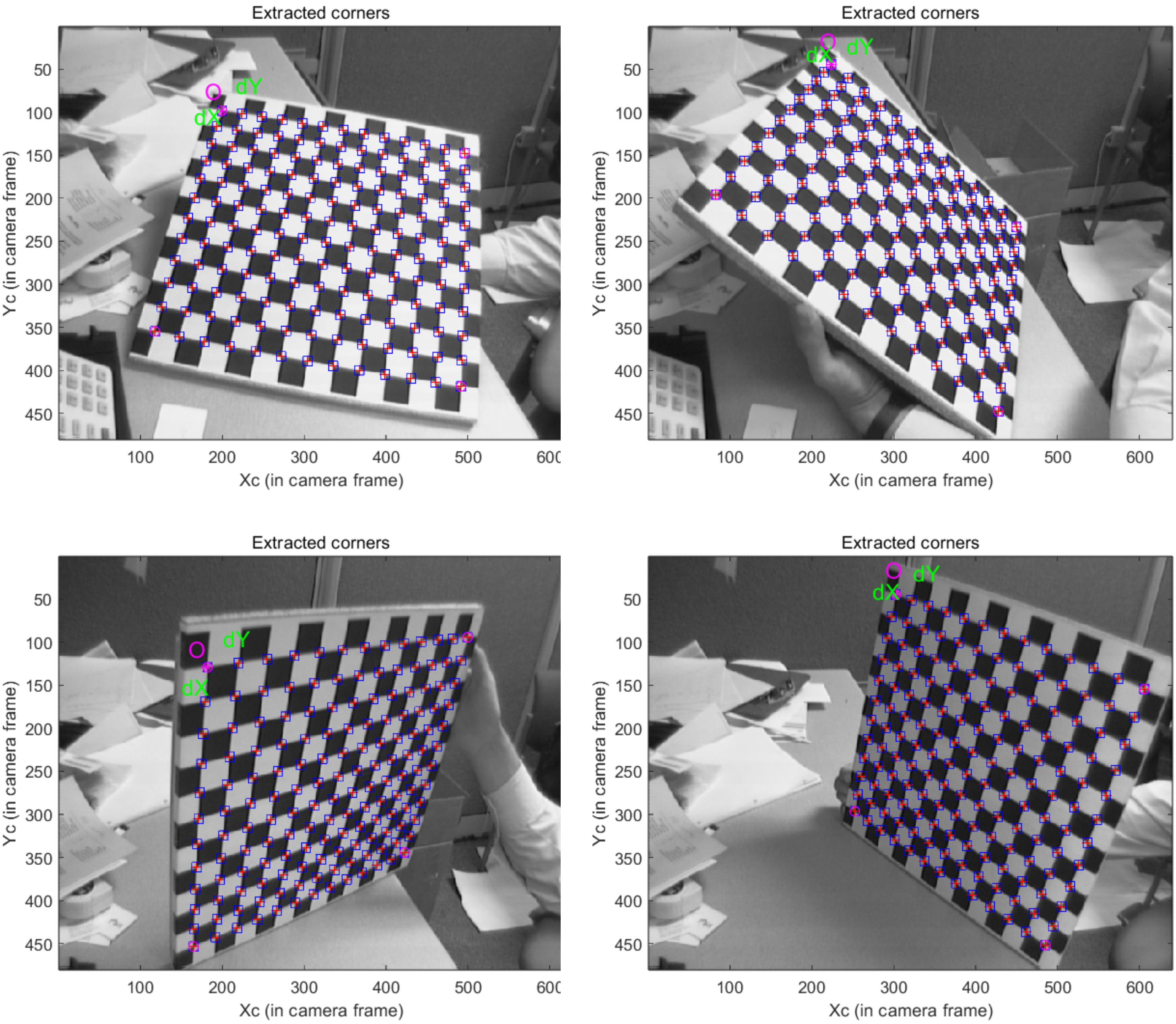
其中 K 为内参矩阵，$\boldsymbol{k}$ 为径向畸变参数，$[R_i \quad \boldsymbol{t}_i]$ 为第 i 张图像对应的外参矩阵，λ_i 为第 i 张图像对应的尺度因子，第 i 张图像中与棋盘中第 j 个角点坐标 X_j 对应的图像坐标为 x_{ij}，$x'(\cdot)$ 为根据棋盘中角点坐标 X_j 以及由该图片对应的外参数矩阵 $[R_i \quad \boldsymbol{t}_i]$、尺度因子 λ_i 及内参矩阵和径向畸变，预测得到的图像像素坐标.

3 实验步骤与结果分析

利用 MATLAB 工具箱 Camera Calibration Toolbox for Matlab 进行相机标定.

3.1 提取棋盘格角点

提取 calib_example 中奇数编号的棋盘图片角点，角点矩阵大小为 $12 \times 13 = 156$，编号为 3,5,7,9 的图像角点检测结果如下图所示：



3.2 求解单应性矩阵

通过 compute_homography.m 文件计算每幅图像的单应性矩阵，下面通过上述角点检测计算出的 x, X 数组，分别为图像坐标和世界坐标，要求大小为 3×156，但 x 矩阵为 2×156，所以需要增加全为 1 的一行，X 矩阵为 3×156，但最下一行为 0，所以需要做前两行的切片，再增加全为 1 的一行位于最下. 代码如下

```matlab
function [H] = calc_homography(x, X)
    shape = size(x);
    x_camera = [x; ones(1, shape(2))];
    x_world = [X(1:2,:); ones(1, shape(2))];
    H = compute_homography(x_camera, x_world);
end
```

得到图像奇数图像的 H 矩阵如下所示

```
H_1 =                                   H_11 =
   -0.2204    0.8074  166.6303              0.0612    0.9032  166.2815
    0.2787    0.0759  178.1498              0.9289    0.1280   28.6891
   -0.0009    0.0001    1.0000              0.0006    0.0000    1.0000
H_3 =                                   H_13 =
   -0.3254    0.8275  197.7143              0.0877    1.2629  145.6157
    0.5622    0.1381   95.3857              1.2514    0.1500   71.6536
   -0.0006   -0.0000    1.0000              0.0008    0.0003    1.0000
H_5 =                                   H_15 =
   -0.5147    0.7639  222.2079              0.0331    1.3813   40.7023
    0.2629    0.5986   41.0748              1.4056    0.1315   64.2955
   -0.0010    0.0003    1.0000              0.0005    0.0003    1.0000
H_7 =                                   H_17 =
    0.1342    1.5613  180.1398              0.2169    1.1970  124.6471
    1.5318    0.0204  127.4220              1.3618    0.2535   58.2397
    0.0012    0.0013    1.0000              0.0012   -0.0002    1.0000
H_9 =                                   H_19 =
   -0.0230    0.4463  301.6882              0.2281    2.0098   99.4373
    0.9281    0.2101   40.7188              1.8122   -0.1245   90.3232
    0.0005   -0.0007    1.0000              0.0019    0.0015    1.0000
```

通过逆变换，加深理解. 这里以图像 1,3 的逆变换为例，其逆矩阵如下所示：

```
invH_1 =                                invH_3 =
   -0.1448    2.2840 -382.7619             -0.2553    1.5238  -94.8838
    1.2632    0.2221 -250.0522              1.1446    0.3676 -261.3736
   -0.0003    0.0019    0.7054             -0.0002    0.0010    0.9378
```

逆变换效果，得到棋盘平面的正投影，利用第二次作业双线性插值完成像素获取，代码及效果如下图所示：

Image1

逆变换

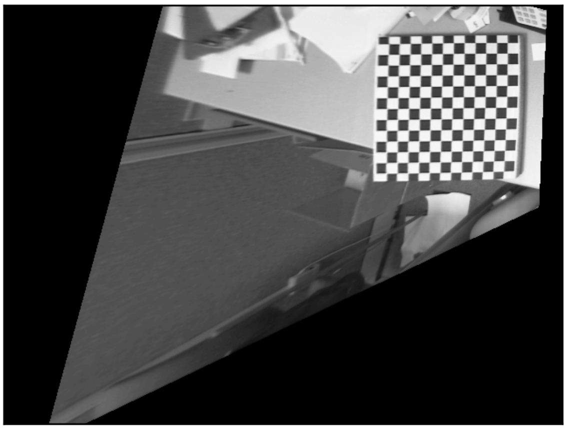
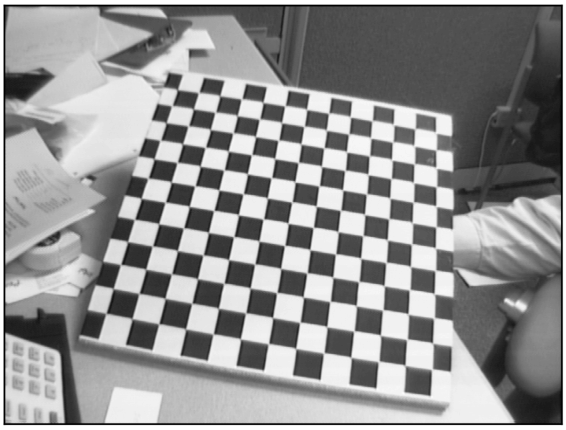
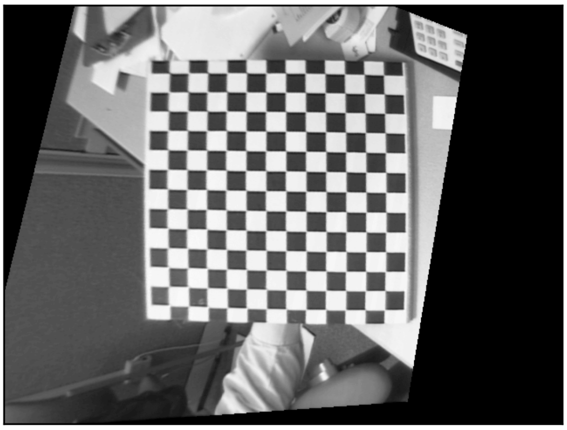


Image3



逆变换



```python
def transform(img, H):
H = H.T
n, m, c = img.shape
mid = np.array([n/2, m/2, 0])
H[2, :] += mid @ H - mid  # 中心化
output = np.zeros(img.shape)
for i in tqdm(range(n)):
    for j in range(m):
        x, y, z = (np.array([j, i, 1]) @ H)
        x /= z  # 正则化
        y /= z
        output[i, j, :] = bilinear_interpolate(img, y, x)  # 利用双线性插值
return output
```

3.3 估计相机内参与外参

3.3.1 初始化内参

利用 init_intrinsic_param.m 计算初始化内参矩阵，得到以下输出结果：

```
Focal Length:          fc = [ 668.00353   668.00353 ]
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 ]
```

于是内参矩阵为

$$K = \begin{bmatrix} 668 & 0 & 319.5 \\ 0 & 668 & 239.5 \\ 0 & 0 & 1 \end{bmatrix}$$

由上述推导可知，约束条件为 $\boldsymbol{r}_1 \perp \boldsymbol{r}_2$, $||\boldsymbol{r}_1|| = ||\boldsymbol{r}_2||$.

由于没有相机规格书，无法得知其物理含义.

3.3.2 计算外参

利用 compute_extrinsic_init 计算外参矩阵，返回值中的 Tckk 为平移向量，Rckk 为旋转矩阵：

```
function [out] = calc_external_par(x, X, fc, cc)
    [omckk,Tckk,Rckk] = compute_extrinsic_init(x,X,fc,cc,0,0);
    out = [[Rckk, Tckk]; zeros(1, 4)];
    out(4, 4) = 1;
end
```

于是外参矩阵为：

```
T_1 =
   0.0742    0.9897    0.1226 -199.2004
   0.6416    0.0467   -0.7656  -79.9566
  -0.7634    0.1355   -0.6315  875.4722
        0         0         0    1.0000
T_3 =
  -0.1463    0.9861    0.0788 -144.6158
   0.8508    0.1660   -0.4986 -171.5600
  -0.5047   -0.0059   -0.8633  794.8232
        0         0         0    1.0000
T_5 =
  -0.2073    0.7679    0.6061 -111.9479
```

```
  0.5806    0.5952   -0.5555 -224.5798
 -0.7874    0.2367   -0.5692  758.0974
       0         0         0    1.0000
T_7 =
 -0.1607    0.7761    0.6098  -95.8754
  0.8366   -0.2207    0.5014  -70.8197
  0.5237    0.5908   -0.6138  452.0687
       0         0         0    1.0000
T_9 =
 -0.2121    0.7567   -0.6184  -22.9072
  0.8961    0.4031    0.1858 -220.2822
  0.3899   -0.5148   -0.7635  757.5272
       0         0         0    1.0000
```

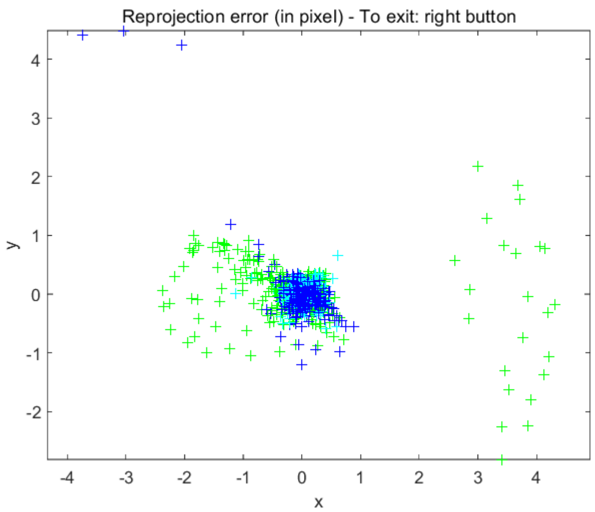
优化参数部分不理解原理无法实现.

3.4 分析标定误差

通过 Calibration 功能得到标定结果及误差（全部奇数张图片，共 10 张图片）:

```
Focal Length: fc = [ 661.07479   665.06267 ] +/- [ 1.91695   2.10219 ]
Principal point: cc = [ 303.25670   241.93662 ] +/- [ 3.53331   3.43451 ]
Skew: alpha_c = [ 0.00000 ] +/- [ 0.00000  ]   => angle of pixel axes = 90.00000 +/- 0.00000 degrees
Distortion: kc = [ -0.29392   0.38862   0.00060   -0.00101   0.00000 ] +/- [ 0.01548   0.06653   0.00084   0.00087   0.00000 ]
Pixel error: err = [ 0.57177   0.33143 ]
```

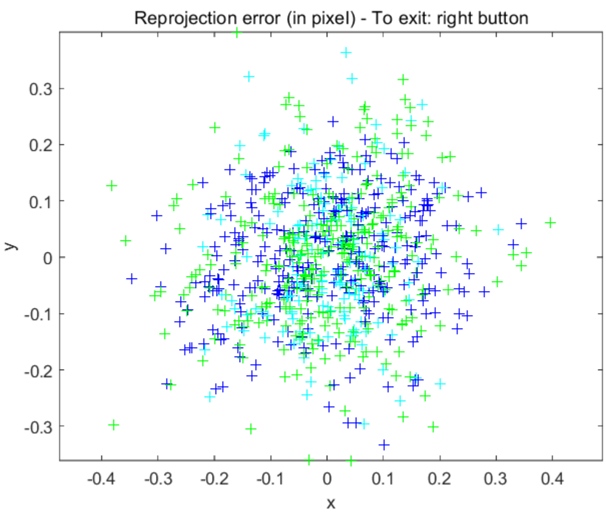
误差分布



通过 Calibration 功能得到标定结果及误差（全部奇数张图片，共 5 张图片）:

```
Focal Length: fc = [ 658.13644   658.74730 ] +/- [ 0.75708   0.81297 ]
Principal point: cc = [ 301.46396   246.52052 ] +/- [ 1.60017   1.55244 ]
Skew: alpha_c = [ 0.00000 ] +/- [ 0.00000  ]   => angle of pixel axes =
↪  90.00000 +/- 0.00000 degrees
Distortion: kc = [ -0.26042   0.16194   0.00028   -0.00042  0.00000 ] +/- [
↪  0.00631   0.03001   0.00033   0.00036  0.00000 ]
Pixel error: err = [ 0.12323   0.11696 ]
```

误差分布



可以发现，虽然图像使用更多，但是可能导致噪声增多，误差分布中会出现较多的离散点，使得估计误差增大，而使用少量图像进行标定噪声数目较少，不会出现这种现象，说明每张标定图片降噪十分重要. 同时误差可能来源于由相机晶状体引起的图像畸变.

4 结论与讨论

通过本次实验，基本理解相机成像原理，以及张氏标定法具体计算方法. 利用 MATLAB 工具箱进行相机标定，通过角点检测，将图像坐标与三维网格坐标对应求解单应性矩阵，进而求解相机内参与外参，但由于没有具体相机参数，无法得知其物理含义. 在标定误差分析中，主要通过逆变换重构三维坐标得到误差大小. 对参数优化部分和具体误差分析部分仍有待进一步理解.