摄像头标定原理

目的

$$\begin{cases} \text{摄像头标定} \\ \\ \text{目标物体测距} \end{cases}$$

原理展示

- 摄像头标定
  - 目的：测量并求出摄像头的内参矩阵 K
  - 公式推导：

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K[r_1, r_2, r_3, t]\begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = K[r_1, r_2, t]\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

其中，$\begin{bmatrix} U \\ V \\ 1 \end{bmatrix}$ 是图像像素坐标系下的坐标，$\begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$ 是虚拟的世界坐标系，即假定世界坐标系是在 $z = 0$ 上，K 是内参矩阵，其大小只与摄像机本身有关，是“固定”的，而$[r_1, r_2, t]$则是外参矩阵 $[r_1, r_2, r_3]$ 和 t 分别是摄像机坐标系相对于世界坐标系的旋转矩阵和平移向量。下面是求解步骤：

a. 令 $H = K[r_1, r_2, t]$，则欲通过

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = H\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

求解出 H（大小为3×3，最后一个元素为1），需要四组数据，即每幅标定图上至少需要有四个角点。

b. 我们的最终目的是求解出 K，所以我们要通过上一步所求解出的 $H = [h_1, h_2, h_3]$ 来解出 K。
在上面中，我们假定虚拟世界坐标系 $z = 0$，即其 Oxy 平面与图像坐标系所在的平面是水平的，z完全不起作用。
故从虚拟世界坐标系 $\rightarrow$ 图像坐标系平面，其实就是虚拟世界坐标系（可视为平面）进行**旋转+平移**后转换到图像坐标平面，即两个平面之间的转换。
根据二维平面的转移矩阵

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

可知，旋转向量 r_1 和 r_2 具备如下约束条件：

$$\begin{cases} r_1 r_2 = 0 \\ r_1^2 = r_2^2 \end{cases} \tag{1}$$

而

$$\begin{cases} h_1 = K r_1 \\ h_2 = K r_2 \end{cases} \tag{2}$$

由（1）和（2）可得

$$\begin{cases} (K^{-1}h_1)^T(K^{-1}h_2) = 0 \\ (K^{-1}h_1)^T(K^{-1}h_1) = (K^{-1}h_2)^T(K^{-1}h_2) \end{cases} \tag{3}$$

所以要求得 K，而 $K = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix}$，有四个未知参数，故根据公式（3），则至少需要两个矩阵 H 才能解出 K。这也是为何我们在实验中需要拍摄很多幅标定图片的原因之一——一幅图片可得到一个 H。

- 目标测距
  - 目的：根据图像中给出的目标点，求出该目标物体在飞行器坐标系下的3D坐标

- 公式：

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = RK^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

其中

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos roll & \sin roll \\ 0 & -\sin roll & cosroll \end{bmatrix} \begin{bmatrix} \cos pitch & 0 & -\sin pitch \\ 0 & 1 & 0 \\ \sin pitch & 0 & \cos pitch \end{bmatrix} \begin{bmatrix} \cos yaw & \sin yaw & 0 \\ -\sin yaw & \cos yaw & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

其中 $roll$（滚转角——x），$pitch$（俯仰角——y）、yaw（偏航角——Z）， $\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$ 是图像物理坐标系下的像素点在飞行器坐标系下的坐标，而 $\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$ 是图像坐标系下像素点的坐标。

最后根据公式

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \frac{h}{z} = \begin{bmatrix} X_t \\ Y_t \\ Z_t \end{bmatrix}$$

求出目标物体相对于飞行器坐标系下的真实坐标 $\begin{bmatrix} X_t \\ Y_t \\ Z_t \end{bmatrix}$

实验问题记录

1. 刚开始无法确定所计算出来的内参矩阵 K 是否正确，后来通过 Matlab 工具箱自带的函数所计算出的结果和我们自己代码运算出的结果进行对比后才确定 K 值正确；
2. 在进行摄像机标定时发现，根本无法找到拍摄的图像中的棋盘格角点。在百度了很久，外加自己的一些实验后，才发现，在进行摄像机标定时，对所拍摄的图片质量是有要求的：
   i. 棋盘格不能有任何遮挡
   ii. 棋盘格表明必须要光滑，不能有任何褶皱
3. 对各个坐标系之间的转换晕头转向，后来在网上查资料和TA大大的帮助下顺利解决：

$$\text{图像坐标系} \times \text{内参矩阵} K^{-1} \rightarrow \text{摄像机坐标系}$$
$$\text{摄像机坐标系} \times \text{旋转矩阵} R \rightarrow \text{飞行器坐标系}$$

4. 刚开始我们以为，我们组的输出是飞行器到目标物体的距离，后来在和控制组沟通后，才知道我们的输出实际实际上是目标物体在飞行器坐标系下的坐标
5. 刚开始标定时，我们采用的图像大小是4000*2250，在该大小下计算出的 K，在后续运算中发现误差在可接受范围内。但后来在和目标识别组沟通后，才知道他们是降采样后的图像。于是我们在原图像上，手动降采样，获得与之一样大小的图像后再重新计算内参矩阵 K，但在实验中却发现，测量出的目标位置，在 x 轴上的误差非常大，达到了 $10cm$ 甚至 $20cm$ 以上，显然不在可接受范围内，但我们找不出误差的原因，只能推断问题出在了降采样那一步。后来TA告诉我们，原因就在降采样那一步，而解决方法是直接从摄像机获取到目标大小的图像，不要自己降采样。后来在TA的帮助下，我们成功获取到图像，重新标定，最后实际测量发现误差几乎在 $1 \sim 2cm$ ，最大的不超过 $5cm$（除了一个异常值），于是实验初步成功。