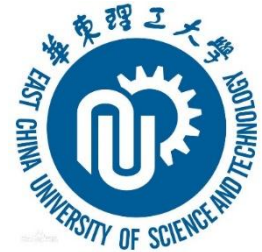


【机械臂视觉抓取教程】

第5讲 相机内参-像素坐标和实际坐标的转换

小五

日期 2022/11/9



目录

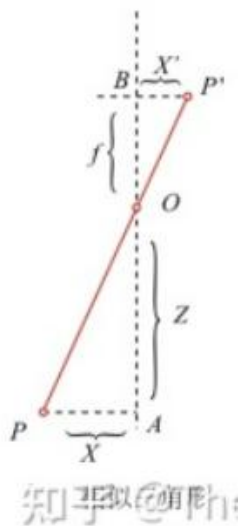
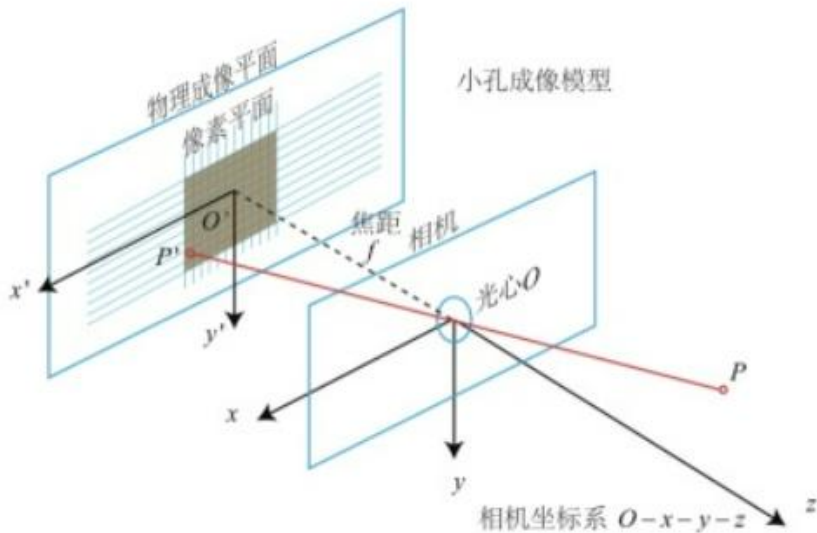
「₁」 相机内参原理

「₂」 代码讲解

「₃」 实战

1 相机内参简介

预备知识：相机坐标系



在对称后，有：

$$\frac{Z}{f} = \frac{X}{X'} = \frac{Y}{Y'}$$

整理解出 P' 的坐标：

$$X' = f \frac{X}{Z}$$

$$Y' = f \frac{Y}{Z}$$



真实世界点中的P的坐标为[X,Y,Z]

成像的点P'的坐标为[X',Y',Z']

物理成像平面和光心的距离为焦距f

成像后的X'与焦距和距离Z有关

1 相机内参简介

像素坐标系通常定义方式是：原点位于图像的左上角，u轴向右与x轴平行，v轴向下与y轴平行。

设P'在像素平面坐标系上的坐标是 $[u, v]^T$ ，我们设像素坐标在u轴上**缩放** α 倍，在v轴上**缩放**了 β 倍。同时，原点**平移**了 $[c_x, c_y]^T$ 。

因此可以得到P'与像素坐标的关系：

$$u = \alpha X' + c_x$$

$$v = \beta Y' + c_y$$

带入刚才得到的公式

$$X' = f \frac{X}{Z}$$

$$Y' = f \frac{Y}{Z}$$

得：

$$u = \alpha f \frac{X}{Z} + c_x = f_x \frac{X}{Z} + c_x$$

$$v = \beta f \frac{Y}{Z} + c_y = f_y \frac{Y}{Z} + c_y$$

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \frac{1}{Z} \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

其中， f_x 、 f_y 、 c_x 、 c_y 均可以从相机获得（相机厂家回提供函数给你调用）

- 代码位置：GRCNN项目中real文件夹下的touch.py文件

$$u = \alpha f \frac{X}{Z} + c_x = f_x \frac{X}{Z} + c_x$$

$$v = \beta f \frac{Y}{Z} + c_y = f_y \frac{Y}{Z} + c_y$$

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \frac{1}{Z} \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

➤ 像素坐标到实际坐标的转化

`click_point_pix = (u,v)` #算法得到的像素坐标 (u, v)

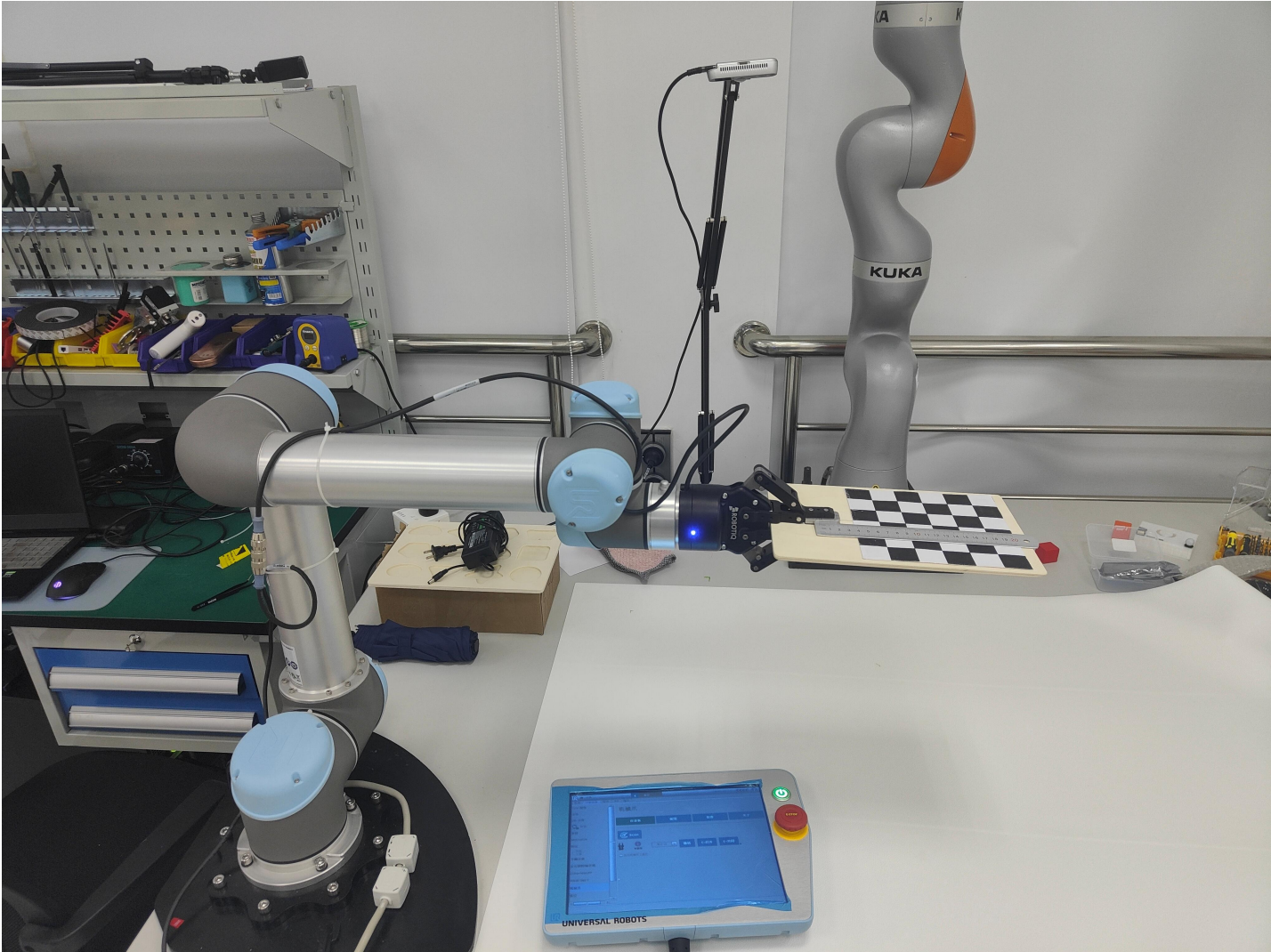
`Z = camera_depth_img[v][u] * robot.cam_depth_scale` #得到实际深度

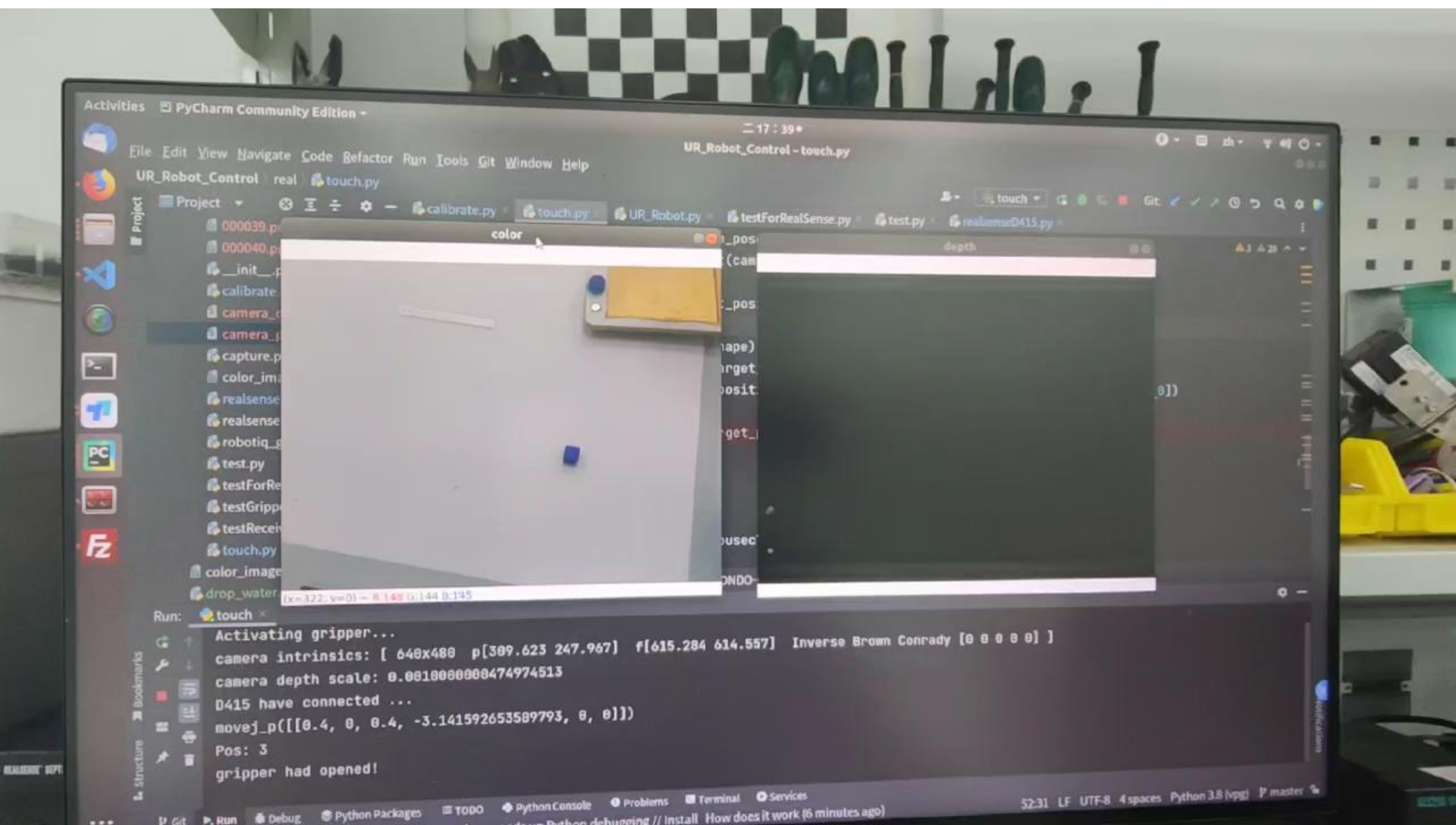
`X = np.multiply(u-robot.cam_intrinsics[0][2],Z/robot.cam_intrinsics[0][0])`

`Y = np.multiply(v-robot.cam_intrinsics[1][2],Z/robot.cam_intrinsics[1][1])`

`click_point = np.asarray([X,Y,Z])` #得到实际坐标

- 注意事项：更改像素大小后，上述内参会改变





视觉抓取教程目录

➤ 算法部分：平面抓取姿态估计

教程1：概述

教程2：项目环境搭建与模型训练

教程3：GRCNN代码讲解

➤ 视觉部分

教程4：手眼标定--眼在手外

教程5：像素位置到实际坐标的转换--相机内参解释

➤ 控制部分

教程6：上位机与机械臂通讯--以优傲机器人为例

教程7：机械臂编程实现对机械臂的控制

教程8：GRCNN项目部署讲解--代码开源

➤ 机器人方向学习路线

补充教程：本人学习路线分享



机器人方向学习交流群



该二维码7天内(11月16日前)有效，重新进入将更新

特点：偏工程、偏基础

**本人水平有限，如有讲错，
请在评论区批评指正！！**