手在眼上：进行手眼标定后获取到的  
手眼标定结果的转换矩阵：表示相机坐标系到机械臂末端坐标系的旋转变换。 通过它，可以将相机坐标系下的物体坐标转换为机械臂末端执行器的坐标系。

手眼标定结果的平移矩阵：表示相机坐标系到末端执行器坐标系的平移变换。 它帮助将物体位置从相机坐标系平移到末端执行器坐标系。

总体实现流程说明：  
**总体流程分析：**

1. **获取物体中心坐标（基于掩码）**：通过物体的掩码 mask，我们可以确定物体在图像中的轮廓信息。接下来，使用掩码计算物体的中心点坐标。
2. **计算物体深度**：使用深度图来估计物体的三维位置。该方法中获取的是物体的中心点深度值，也可以选取平均深度值。
3. **将相机坐标系下的物体位置转换为机械臂末端执行器坐标系**：使用 **相机内参**（color\_intr），将物体的像素坐标 (real\_x, real\_y) 和深度值 dis 转换为相机坐标系下的三维空间坐标 (x, y, z)。这个计算涉及相机的焦距和主点位置。
4. **坐标转换（从相机坐标系到末端执行器坐标系）**：使用手眼标定的结果，将物体在相机坐标系下的三维位置 (x, y, z) 转换为末端执行器坐标系下的位置。

这个转换通过齐次变换矩阵实现。

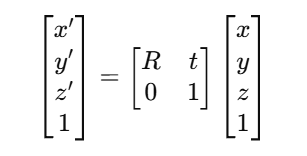
1. **物体位姿调整（考虑夹爪长度和垂直位姿）**：根据机械臂末端执行器的当前位置 current\_pose 和 **夹爪长度**（arm\_gripper\_length），计算物体在末端执行器上方的位置，确保夹爪能够准确地接触物体。夹爪的最终位姿是基于 **垂直抓取的旋转角度**（vertical\_rx\_ry\_rz）来调整的。
2. **旋转角度的计算和调整**：为了确保夹爪能够准确对准物体，计算物体的偏转角度（相对于末端执行器），并通过计算得到正确的抓取角度。这个角度计算基于物体的掩码信息和物体的轮廓。
3. **最终抓取位姿的计算**：最终，结合物体的深度信息、旋转角度、夹爪长度和预设位姿，计算出三个关键位姿：

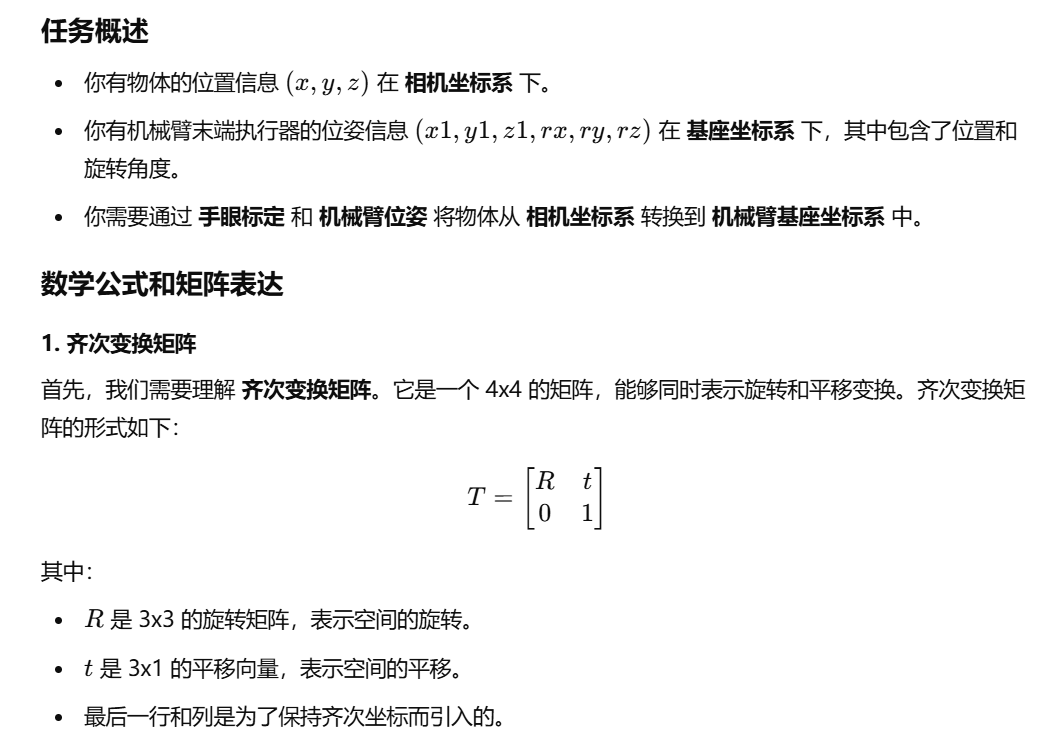
（1）above\_object\_pose：物体上方的位姿，用于将夹爪移动到物体上方。

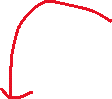
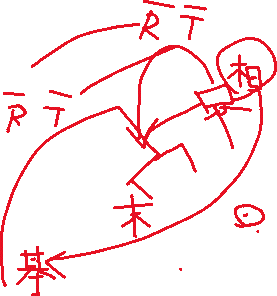
（2）correct\_angle\_pose：夹爪旋转到正确角度的位姿，用于确保夹爪对准物体。

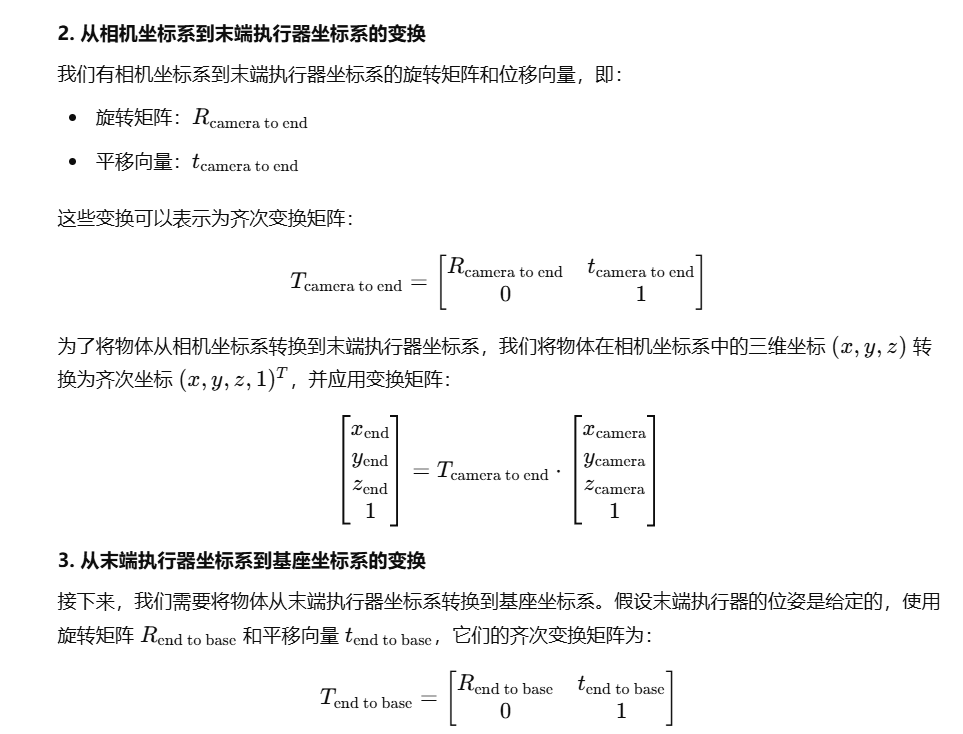
（3）finally\_pose：物体下方抓取的位姿，用于最终执行抓取。

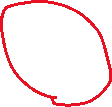
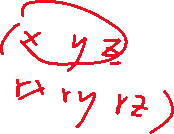
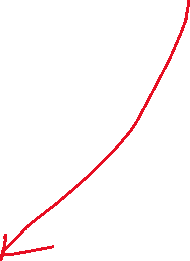
关键数学原理：

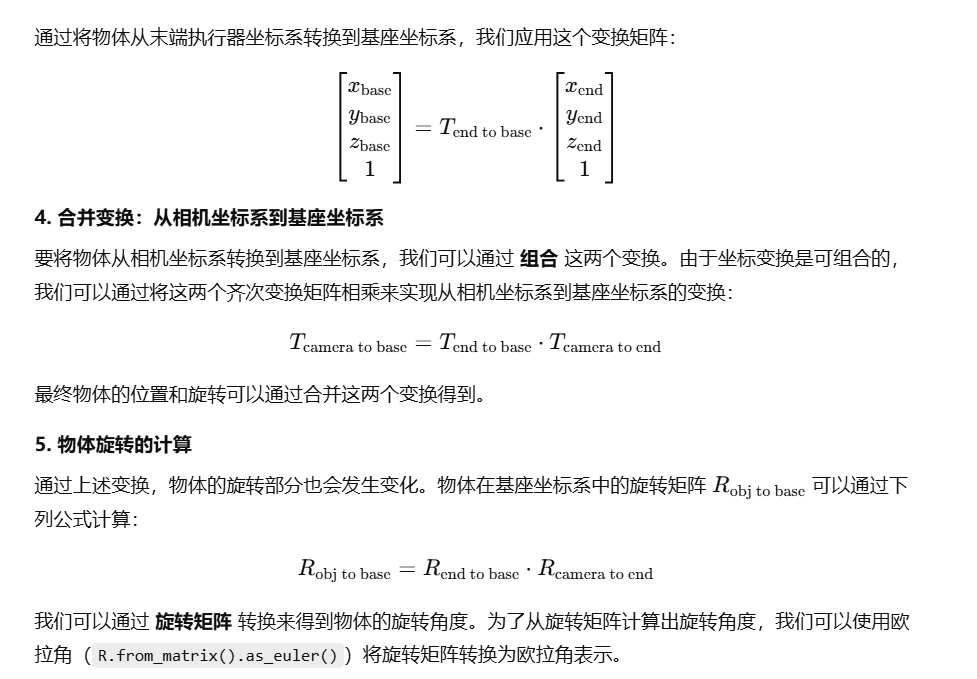
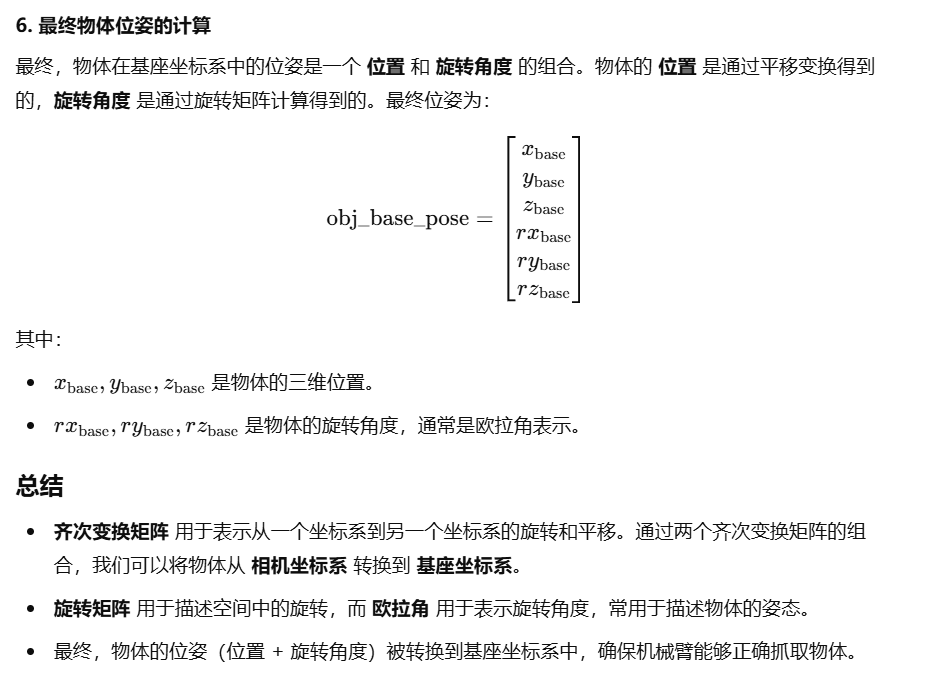
1. 坐标变换：  
   核心：  
   

代码实现原理：  


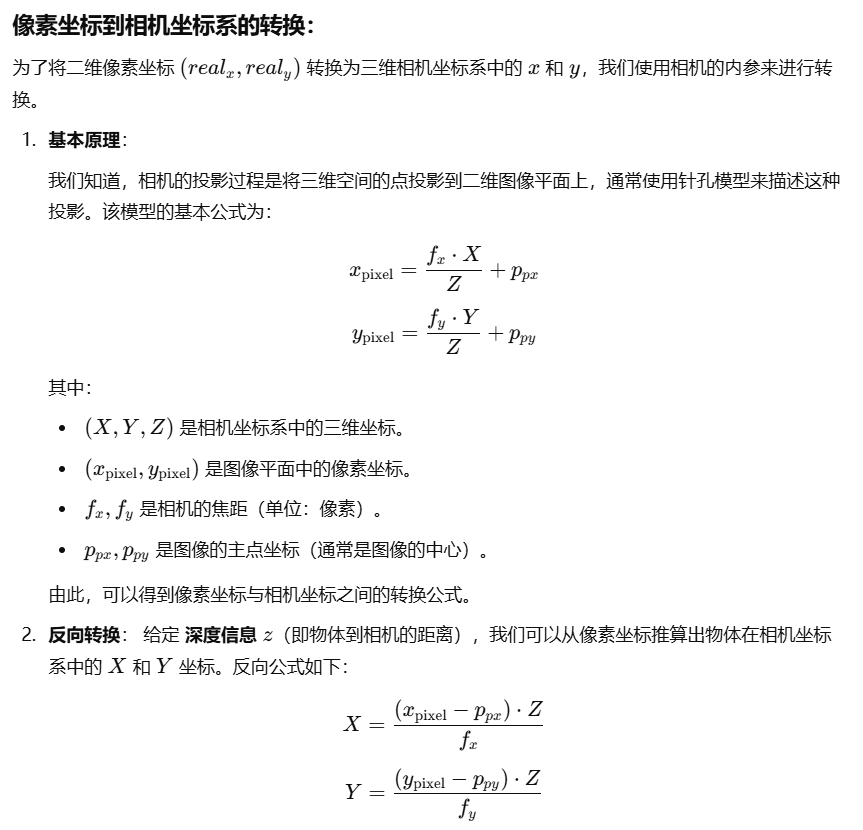






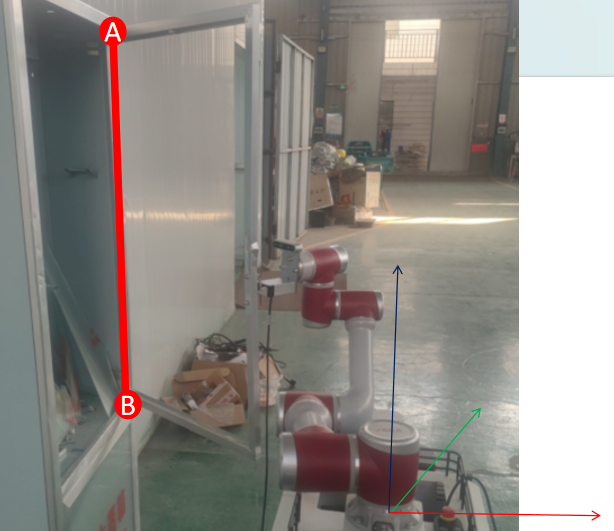
2.



**------------------------------------分割线-----------------------------------------------**

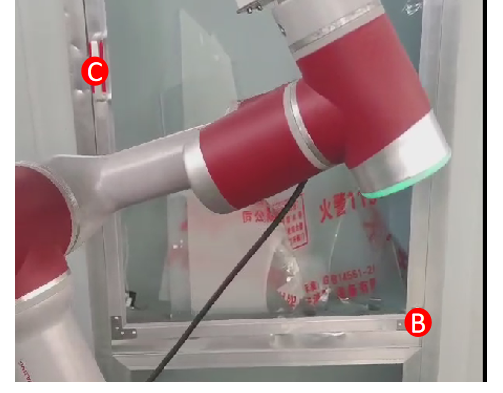
#### 标定门转轴位置

首先，要标定如图所示AB转轴的位置以确定旋转中心，可以机器人示教到B点并记录B点坐标（Bx,By,Bz）。由于你的末端装了夹具，为了精确一点你需要用B点的X坐标减去夹爪长度。



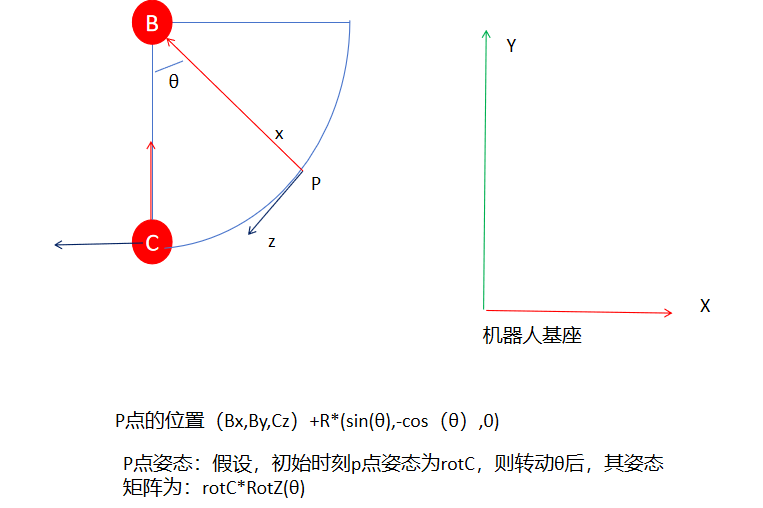
#### 标定开门半径

C点为门把手位置，机器人示教到C点记录坐标（Cx,Cy,Cz）。则开门半径大小为：R=sqrt((Cx-Bx)^2+(Cy-By)^2)。

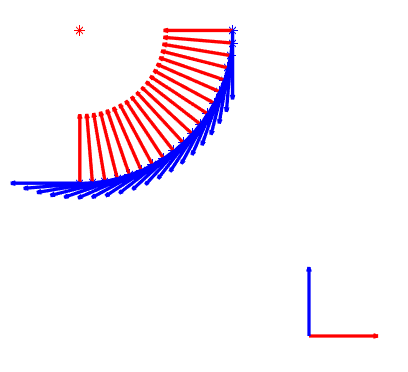


#### 标定完成开始建模

我们开门，是绕着轴让门把手沿着切线方向去开门的。下面画一个开门的俯视图。



已验证如下：



#### 4.轨迹规划

已经得到了开门过程中的P点的位置和姿态，轨迹规划只需要规划θ这个变量即可，采用最简单的五次多项式；