目标的坐标转换问题

汇报人：罗小兰

汇报时间：2021 年 6 月 11 日星期五

摘要：研究了目标的坐标转换问题，涉及到很多坐标系的相关知识。

1、 引言

（1） 上周存在的问题

① 在仿真的时候，与普通的 KF 跟踪器做了对比，差异很小，不知在实际的目标检测中滤波效果如何，有待进一步测试。

② 但目标检测部分只能直接提供目标在图像中的坐标，无法直接提供目标的世界坐标观测值，要在实现目标定位+坐标转换同时，加上滤波器加以测试。

③ 需要目标的世界坐标真实值与观测值，预测值做对比，可能还需要借助精确的定位设备来获取。

（2） 上周制定的工作思路及计划

查阅更多的文献，并研究目标定位的算法。

（3） 计划完成情况

研究了目标的坐标转换问题。

2、 理论

视觉中涉及到的四个坐标系：

① 像素坐标系

原点： 图像的左上角，也即图像数组中的 (0, 0)

坐标系： u - v

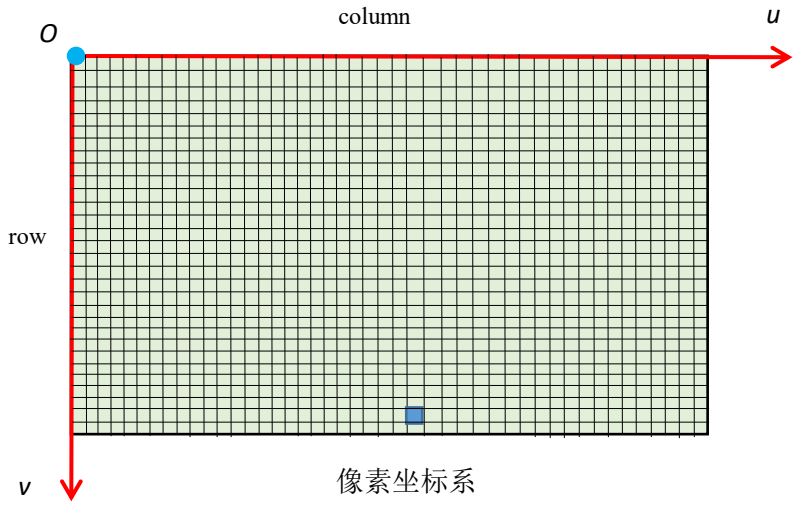
u 轴 — 平行于图像的宽向右

v 轴 — 平行于图像的高向下

坐标：(u, v) — 在图像数组中所在的列数和所在的行数

单位：像素（pixel）

（注意： 像素坐标系并非物理坐标系）



像素坐标系

② 图像坐标系

原点： 图像平面中心，该点在像素坐标系下的坐标表示为（u0，v0）

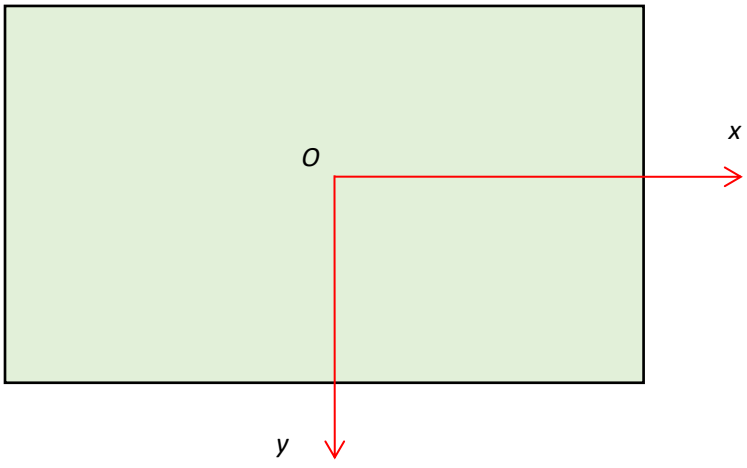
坐标系： x - y

x 轴 — 平行于图像的宽向右

y 轴 — 平行于图像的高向下

坐标：(x, y)

单位：毫米（mm）



图像坐标系

图像坐标系与像素坐标系的本质区别：它们的单位不同，图像坐标系是物理坐标系，像素坐标系不是物理坐标系。像素是一种虚拟化的数字，大小可以任意，没有实际的物理尺寸大小。

如果要将图像坐标与像素坐标进行转换，就需要已知图像中每个像素的物理宽度 dx 和高度 dy（单位为 mm/pixel），这两个参数是联系两个坐标系的桥梁。x/dx 和 y/dy 的单位就

是像素（pixel）了。那么 dx 和 dy 如何得到呢？通过相机的标定可以间接得到。

相机标定可以得到相机的内参矩阵和外参矩阵。针孔相机模型的内参矩阵包括 6 个参数 *(f, k, fx, fy, u0, v0)*。其中，各个参数含义如下：

f 为相机的焦距，单位是毫米(mm)；

k 表示径向畸变量级，如果 *k* 为负值，畸变为桶型畸变，如果为正值，那畸变为枕型畸变；

fx 为 x 轴上的归一化焦距, $fx = f / dx$；

fy 为 y 轴上的归一化焦距，$fy = f / dy$；

（*dx* 和 *dy* 为前面提到的像元尺寸，表示一个像素在 *x* 方向和 *y* 方向的实际物理尺寸，单位为 mm/pixel）

u0 为图像中心与原点的像素坐标之间相差的横向像素数偏移量，*u0 = width / 2*。

v0 为图像中心与原点的像素坐标之间相差的纵向像素数偏移量，v0 = height / 2。

我们通过图像直接得到的目标坐标其实是目标在像素坐标系下的坐标 (u, v)，但是它不是成像物理坐标，需要将该坐标转换到图像坐标系下得到 (x, y)。

从图像坐标系到像素坐标系的转换公式如下：

$$u = \frac{x}{dx} + u_0$$

$$v = \frac{y}{dy} + v_0$$

表示为齐次坐标形式：

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

从像素坐标系到图像坐标系的转换公式如下：

$$x = (u - u_0)\,dx$$

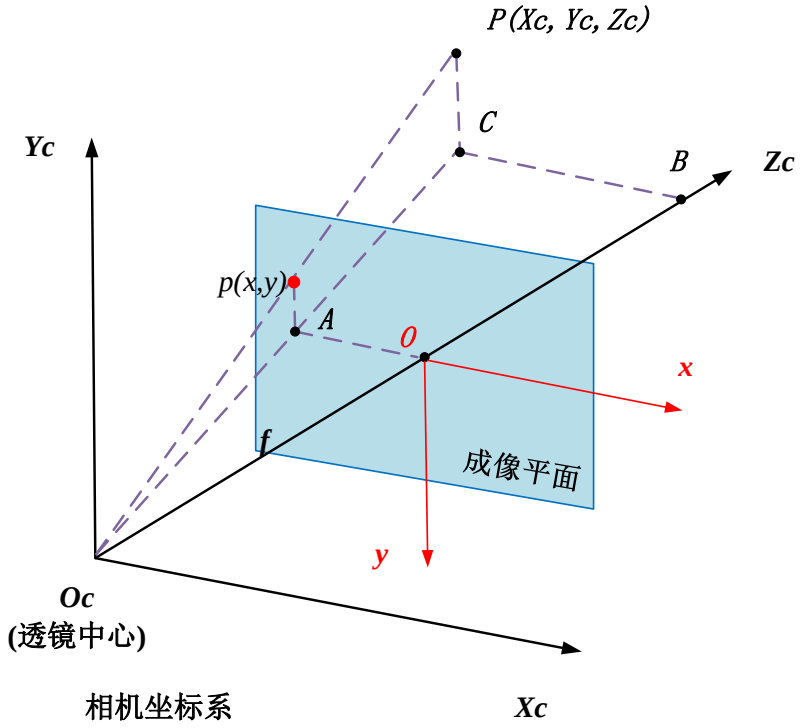
$$y = (v - v_0)\,dy$$

齐次坐标形式：

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} dx & 0 & -u_0 dx \\ 0 & dy & -v_0 dy \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$

③ 相机坐标系

在相机坐标系下位于 (Xc, Yc, Zc) 处的点 P，在相机的成像平面（底片）中成像为点 p，在图像坐标系下位于 (x, y)。如下图所示：



图中的线段与坐标的关系如下：

$$\begin{cases} OA = x \\ Ap = y \\ O_cO = f \end{cases}, \quad \begin{cases} BC = X_c \\ CP = Y_c \\ O_cB = Z_c \end{cases}$$

其中， f 为焦距，是透镜中心与成像平面中心的距离， (X_c, Y_c, Z_c) 为相机坐标系下的坐标， (x, y) 为图像坐标系下的坐标。

由相似三角形原理可以得到：

$$\frac{OA}{BC} = \frac{Ap}{CP} = \frac{O_cO}{O_cB} \quad \Rightarrow \quad \frac{x}{X_c} = \frac{y}{Y_c} = \frac{f}{Z_c}$$

进一步得到：

$$\begin{cases} X_c = x\dfrac{Z_c}{f} \\ Y_c = y\dfrac{Z_c}{f} \end{cases}$$

将像素坐标到图像坐标的转换关系代入上式：

$$x = (u - u_0)\,dx$$
$$y = (v - v_0)\,dy$$

可得到像素坐标到相机坐标的转换关系：

$$\begin{cases} X_c = (u - u_0)\dfrac{Z_c}{fx} \\ Y_c = (v - v_0)\dfrac{Z_c}{fy} \\ Z_c = d \end{cases}$$

其中，d 为深度相机测出的深度值，也即是真实点到相机平面的垂直距离。

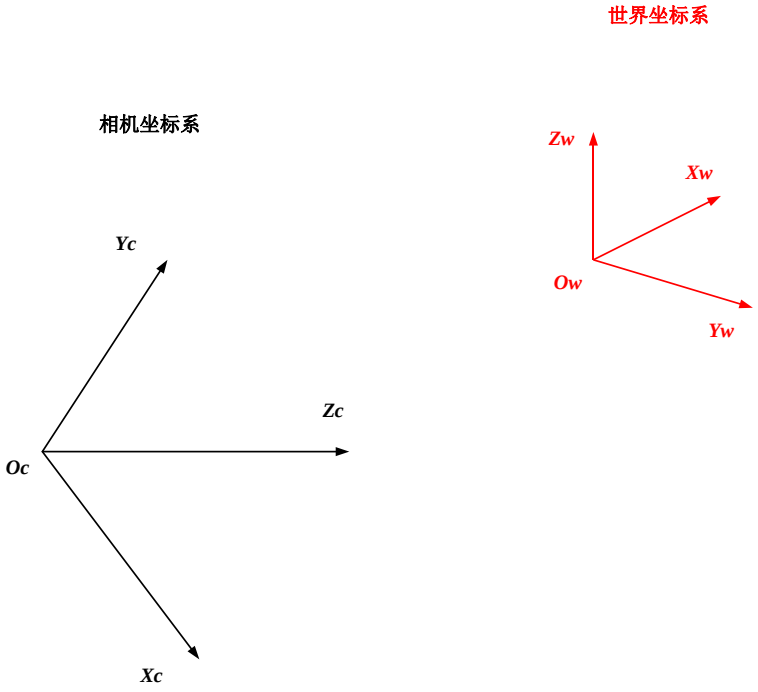
像素坐标到相机坐标的转换关系表示为齐次坐标形式：

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \begin{bmatrix} \dfrac{1}{fx} & 0 & -\dfrac{u_0}{fx} \\ 0 & \dfrac{1}{fy} & -\dfrac{v_0}{fx} \\ 0 & 0 & 1 \end{bmatrix}$$

相机坐标到像素坐标的转换关系表示为齐次坐标形式：

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{Z_c} \begin{bmatrix} fx & 0 & u_0 \\ 0 & fy & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

④ 世界坐标系



世界坐标到相机坐标的转换关系：

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = R \begin{bmatrix} X_\omega \\ Y_\omega \\ Z_\omega \end{bmatrix} + t$$

或者

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = T \begin{bmatrix} X_\omega \\ Y_\omega \\ Z_\omega \end{bmatrix}$$

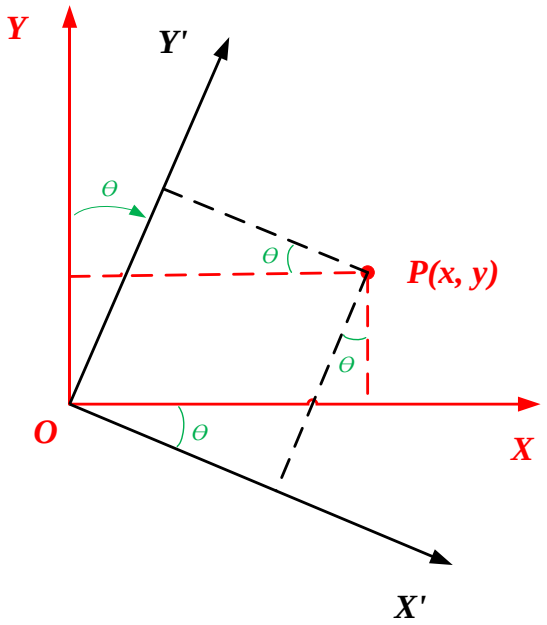
其中，$(X_\omega, Y_\omega, Z_\omega)$ 为世界坐标系下的坐标，R 为 3x3 的旋转矩阵，t 为三维的平移向量，T 为齐次变换矩阵。

公式中，相机坐标 (X_c, Y_c, Z_c) 中的 Z_c 是深度相机直接可以获取的，X_c, Y_c 可以通过像素坐标，内参和深度 Z_c 计算间接来得到。再根据相机的齐次变换矩阵 T 或者旋转矩阵和平移向量 R, t 就可以求出世界坐标系下的坐标 $(X_\omega, Y_\omega, Z_\omega)$。

如何得到的？

先不看三维坐标的旋转，为了简化问题，先看二维坐标的旋转。坐标系 $X-Y$ 旋转到 $X'-Y'$，坐标系 $X-Y$ 下的坐标 $P(x, y)$ 映射到坐标系 $X'-Y'$ 下坐标为 $P'(x', y')$。（旋转角度是有符号的，假设坐标逆时针旋转时旋转角度 Θ 为正，顺时针旋转时旋转角度 Θ 为负）

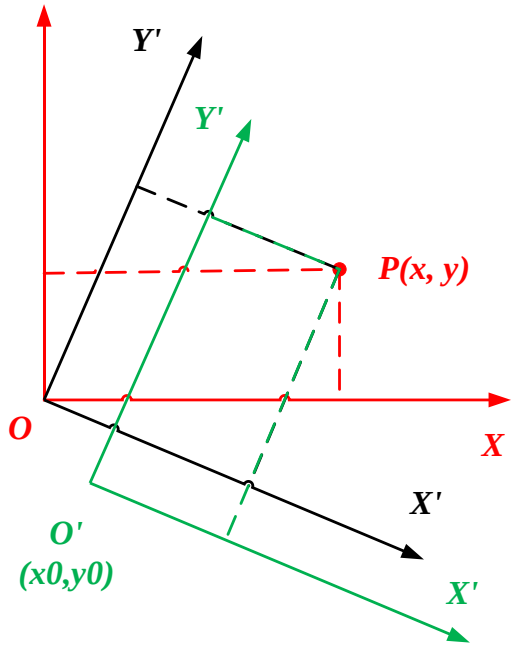
下面是只考虑绕原点坐标轴的旋转，不考虑平移的情况。



如果已知坐标系 $X-Y$ 下的坐标 $P(x, y)$ 和旋转角度 Θ，可以得到坐标系 $X'-Y'$ 下的坐标 $P'(x', y')$：

$$\begin{cases} x' = x\cos\theta + y\sin\theta \\ y' = y\cos\theta - x\sin\theta \end{cases} \quad (\text{逆时针旋转时}\theta > 0)$$

除了坐标轴的旋转之外，也考虑原点的平移：



如果已知坐标系 $X-Y$ 下的坐标 $P(x,y)$，旋转角度 Θ 以及平移后得到的原点 O' 在原坐标系 $X-Y$ 下的坐标 $(x0,y0)$，可以得到坐标系 $X'-Y'$ 下的坐标 $P'(x',y')$：

$$\begin{cases} x' = x\cos\theta + y\sin\theta - x0 \\ y' = y\cos\theta - x\sin\theta - y0 \end{cases}$$

表示为齐次坐标形式：

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta & -x0 \\ -\sin\theta & \cos\theta & -y0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

其实三维坐标轴的旋转可以分解为分别饶三个坐标轴的旋转。

（1）假设相机坐标系的 Xc 轴与世界坐标系的 $X\omega$ 轴一致，只有 $Yc-Zc$ 轴所构成的平面绕着 $X\omega$ 轴旋转了角度 θ，不考虑平移。同理可以得到世界坐标到相机坐标的转换关系：

$$\begin{pmatrix} Xc \\ Yc \\ Zc \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} X\omega \\ Y\omega \\ Z\omega \end{pmatrix}$$

$$r1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix}$$

（2）假设相机坐标系的Yc轴与世界坐标系的$Y\omega$轴一致，只有$Xc-Zc$轴所构成的平面绕着$Y\omega$轴旋转了角度θ，不考虑平移。同理可以得到世界坐标到相机坐标的转换关系：

$$\begin{pmatrix} Xc \\ Yc \\ Zc \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} X\omega \\ Y\omega \\ Z\omega \end{pmatrix}$$

$$r2 = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix}$$

（3）假设相机坐标系的Zc轴与世界坐标系的$Z\omega$轴一致，只有$Xc-Yc$轴所构成的平面绕着$Z\omega$轴旋转了角度θ，不考虑平移。可以得到世界坐标到相机坐标的转换关系：

$$\begin{pmatrix} Xc \\ Yc \\ Zc \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X\omega \\ Y\omega \\ Z\omega \end{pmatrix}$$

$$r3 = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

因此，世界坐标到相机坐标的转换关系可以表示为：

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = R \begin{bmatrix} X_\omega \\ Y_\omega \\ Z_\omega \end{bmatrix} + t$$

其中，旋转矩阵和平移向量R,t为相机的外部参数，反映了相机的位姿变化。

旋转矩阵R：

$$R = r1 \cdot r2 \cdot r3 = \begin{pmatrix} r_{00} & r_{01} & r_{02} \\ r_{10} & r_{11} & r_{12} \\ r_{20} & r_{21} & r_{22} \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

平移向量t：

$$t = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$$

代入世界坐标到相机坐标的转换关系中得到：

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{pmatrix} r_{00} & r_{01} & r_{02} \\ r_{10} & r_{11} & r_{12} \\ r_{20} & r_{21} & r_{22} \end{pmatrix} \begin{bmatrix} X_\omega \\ Y_\omega \\ Z_\omega \end{bmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$$

那么，这个旋转矩阵和平移向量怎么得到呢？

（1）欧拉角

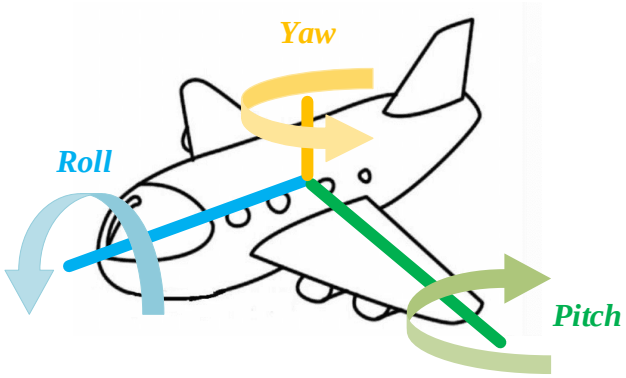
欧拉角是绕三个坐标轴旋转的角度（物体坐标系 W -> 惯性坐标系 A），无平移数据。

① 翻滚角（roll）：绕 Z 轴的旋转角度 α

② 偏航角（yaw）：绕 Y 轴的旋转角度 β

③ 俯仰角（pitch）：绕 X 轴的旋转角度 γ

（注意这些角度表示从 W 坐标系到 A 坐标系经历了上述旋转，顺序是 Z-Y-X）



绕三个轴旋转的旋转矩阵为：

$$\begin{cases} R_x(\gamma) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{pmatrix} \\ R_y(\beta) = \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix} \\ R_z(\alpha) = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{cases}$$

如果坐标系 W 经过旋转变换到坐标系 A（旋转用旋转矩阵 R_{AW} 来描述），目标在惯性坐标系 A 下的坐标为 P_A，在坐标系 W 下的坐标为 P_W，两个坐标的关系为：

$$R_{AW} \cdot P_W = P_A \quad \Leftrightarrow \quad P_W = R_{AW}^T \cdot P_A = R_{WA} \cdot P_A$$

当 W 坐标系依次绕 $Z-Y-X$ 轴进行旋转，旋转角度为右手系（逆时针旋转时角度为正，否则为负），从 W 坐标系变换到 A 坐标系的旋转矩阵 R_{AW} 为：

$$R_{AW} = R_x^T \cdot R_y^T \cdot R_z^T$$

则从 A 坐标系变换到 W 坐标系的旋转矩阵 R_{WA} 为：

$$R_{WA} = R_{AW}^T = (R_x^T \cdot R_y^T \cdot R_z^T)^T = R_z \cdot R_y \cdot R_x$$

进一步得到：

$$R_{WA} = \begin{pmatrix} \cos(\alpha)\cos(\beta) & \cos(\alpha)\sin(\beta)\sin(\gamma) - \sin(\alpha)\cos(\gamma) & \cos(\alpha)\sin(\beta)\cos(\gamma) + \sin(\alpha)\sin(\gamma) \\ \sin(\alpha)\cos(\beta) & \sin(\alpha)\sin(\beta)\sin(\gamma) + \cos(\alpha)\cos(\gamma) & \sin(\alpha)\sin(\beta)\cos(\gamma) - \cos(\alpha)\sin(\gamma) \\ -\sin(\beta) & \cos(\beta)\sin(\gamma) & \cos(\beta)\cos(\gamma) \end{pmatrix}$$

由欧拉角计算旋转矩阵的程序：

```
Eigen::Matrix3d zyxToRotationMatrix(Eigen::Vector3d zyx)
{
    // 计算旋转矩阵的x分量
    Eigen::Matrix3d R_x;
    R_x << 1,           0,            0,
           0,  cos(zyx[2]),  -sin(zyx[2]),
           0,  sin(zyx[2]),   cos(zyx[2]);

    // 计算旋转矩阵的y分量
    Eigen::Matrix3d R_y;
    R_y << cos(zyx[1]),  0,  sin(zyx[1]),
                    0,  1,           0,
          -sin(zyx[1]),  0,  cos(zyx[1]);

    // 计算旋转矩阵的z分量
    Eigen::Matrix3d R_z;
    R_z << cos(zyx[0]),  -sin(zyx[0]),  0,
           sin(zyx[0]),   cos(zyx[0]),  0,
                    0,            0,  1;

    // 依次左乘，合并
    Eigen::Matrix3d R = R_z*R_y*R_x;
    return R;
}
```

如果已知目标在 A 坐标系下的坐标 P_A 和 A 系 -> W 系的旋转矩阵 R_{WA}，就能计算出 W 坐标系下的坐标 P_W。

欧拉角的问题：万向锁。

（2）四元数

除了欧拉角外，还可以用四元数来描述三维空间的旋转，可以避免万向锁问题，也能能加方便第给出旋转的转轴和旋转角度。

$$q = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k} + w \quad (x^2 + y^2 + z^2 + w^2 = 1)$$

假设 A 坐标系下的坐标 P_A，通过四元数 q 转换到 W 坐标系下得到坐标 P_W。

$$P_W = q \cdot P_A \cdot q^{-1}$$

其中，q^{-1}为q的共轭。

$$q^{-1} = \begin{bmatrix} -x \\ -y \\ -z \\ w \end{bmatrix}$$

如果已知绕 Z 轴，Y 轴和 X 轴旋转的欧拉角α, β, γ，可以计算出四元数：

$$x = \sin\left(\frac{\gamma}{2}\right)\cos\left(\frac{\beta}{2}\right)\cos\left(\frac{\alpha}{2}\right) - \cos\left(\frac{\gamma}{2}\right)\sin\left(\frac{\beta}{2}\right)\sin\left(\frac{\alpha}{2}\right)$$

$$y = \cos\left(\frac{\gamma}{2}\right)\sin\left(\frac{\beta}{2}\right)\cos\left(\frac{\alpha}{2}\right) + \sin\left(\frac{\gamma}{2}\right)\cos\left(\frac{\beta}{2}\right)\sin\left(\frac{\alpha}{2}\right)$$

$$z = \cos\left(\frac{\gamma}{2}\right)\cos\left(\frac{\beta}{2}\right)\sin\left(\frac{\alpha}{2}\right) - \sin\left(\frac{\gamma}{2}\right)\sin\left(\frac{\beta}{2}\right)\cos\left(\frac{\alpha}{2}\right)$$

$$w = \cos\left(\frac{\gamma}{2}\right)\cos\left(\frac{\beta}{2}\right)\cos\left(\frac{\alpha}{2}\right) + \sin\left(\frac{\gamma}{2}\right)\sin\left(\frac{\beta}{2}\right)\sin\left(\frac{\alpha}{2}\right)$$

如果已知四元数，也可以计算出欧拉角：

$$\gamma = atan2\left(2\left(wx + yz\right), 1 - 2\left(x^2 + y^2\right)\right)$$

$$\beta = \arcsin 2\left(2\left(wy - zx\right)\right)$$

$$\alpha = atan2\left(2\left(wz + xy\right), 1 - 2\left(y^2 + z^2\right)\right)$$

如果已知四元数，计算旋转矩阵：

$$R(q) = \begin{bmatrix} 1-2y^2-2z^2 & 2xy-2zw & 2xz+2yw \\ 2xy+2zw & 1-2x^2-2z^2 & 2yz-2xw \\ 2xz-2yw & 2yz+2xw & 1-2x^2-2y^2 \end{bmatrix}$$

ROS 中为四元数提供了两种数据类型 msg geometry_msgs::Quaternion 和 tf tf2::Quaternion，一般都是将 w 放在最后，但在 Eigen 库中 w 是放在最前面的。其中 tf 中的 setRPY()将绕轴旋转的角度转化为四元数，为了防止四元数的值大于 1，用 normalize()进行归一化。Tf 中的 toMsg()可以将 Eigen::Quaternond 转化为 geometry_msgs::Quaternion。

无人机的位姿信息由 MAVROS 的/mavros/local_position/pose 话题提供，但所有的位置和姿态信息都要根据坐标系来定义。

marvros/imu.cpp：

```
/**
 * @warning A rotation from the aircraft-frame to the base_link frame is applied.
 * Additionally, it is reported the orientation of the vehicle to describe the
 * transformation from the ENU frame to the base_link frame (ENU <-> base_link).
 * THIS ORIENTATION IS NOT THE SAME AS THAT REPORTED BY THE FCU (NED <-> aircraft).
 */
```

```
/**
 * @brief Fill and publish IMU data message.
 * @param time_boot_ms     Message timestamp (not syncronized)
 * @param orientation_enu  Orientation in the base_link ENU frame
 * @param orientation_ned  Orientation in the aircraft NED frame
 * @param gyro_flu         Angular velocity/rate in the base_link Forward-Left-Up frame
 * @param gyro_frd         Angular velocity/rate in the aircraft Forward-Right-Down frame
 */

        /** MAVLink quaternion exactly matches Eigen convention.
         *  The RPY describes the rotation: aircraft->NED.
         *  It is required to change this to aircraft->base_link:
         *  @snippet src/plugins/imu.cpp ned->baselink->enu
         */


                /** Publish only base_link ENU message
                 *  @snippet src/plugins/imu.cpp pub_enu
                 */
                // [pub_enu]
                imu_pub->publish(imu_enu_msg);
                // [pub_enu]
```

MAVROS 中 IMU 发布的四元数描述的是从 ENU 坐标系 -> Baselink 坐标系（X-机头朝前方向，Y-向左，Z-向上）的旋转。

```
odom.pose.pose.position = tf2::toMsg(enu_position);
odom.pose.pose.orientation = enu_orientation_msg;
tf2::toMsg(baselink_linear, odom.twist.twist.linear);
odom.twist.twist.angular = baselink_angular_msg;
```

MAVROS 的/mavros/local_position/pose 中发布的位置信息是 ENU 坐标系下的坐标，四元数是从 ENU 坐标系旋转到 Base-link 坐标系的四元数。已知 ENU 坐标系下的坐标和 ENU 坐标系旋转到 Base-link 坐标系的四元数可以计算出 Base-link 坐标系下的坐标。

综上所述，目标的坐标转换过程大致为：
（1）已知目标在像素坐标系下的坐标
（2）计算出目标在相机坐标系下的坐标
（3）如果云台中内置 IMU，能够提供 ENU 坐标系到相机坐标系的旋转关系，那么相机坐标系到 ENU 坐标系的旋转关系就是前者的转置，再加上无人机能够获取到平移信息，就能计算得到目标在 ENU 坐标系下的坐标。

如果没有内置 IMU，就需要先将目标在相机坐标系下的坐标转换到在无人机机体坐标系下的坐标，然后借助无人机姿态信息，将目标在机体坐标系下的坐标转换到 ENU 坐标系下。
（无人机需要在同一点上电，这样 ENU 的坐标原点才能保持一致）

3、 结论

（1） 本周工作总结

研究目标坐标转换的问题。

（2） 存在的问题及原因

涉及到很多种坐标系，且有些坐标系的定义不统一，绕轴旋转顺序不同，旋转矩阵的公式也不一样，还没有试验测试，不知道转换的思路和公式是否有误。

（3） 下一步工作思路及计划

编写程序进行试验测试，验证转换的思路是否有问题。