AMAZON PICKING CHALLENGE FRAME CONVENTIONS FOR GRASPING

JEFF MAHLER

This document describes the conventions used to generate and visualize grasps for the PR2 in the Amazon Picking Challenge.

1. Background and Notation

Let $T \in SE(3)$ denote a pose matrix:

$$T = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}$$
$$R \in \mathbb{R}^{3\times 3}, R^T R = I_{3\times 3}$$
$$\mathbf{t} \in \mathbb{R}^3, \mathbf{t} = [t_x, t_y, t_z]^T$$

Let A and B be coordinate frames. Let T_A^B denote the pose of coordinate frame B with respect to coordinate frame A. This means that for a vector $\mathbf{v}_A = [x, y, z, 1]^T$ in coordinate frame A, $\mathbf{v}_B = T_A^B \mathbf{v}_A$. Note that $\left(T_A^B\right)^{-1} = T_B^A$ and for a third coordinate frame C, $T_A^C = T_A^B * T_B^C$ where $*$ denotes matrix multiplication.

2. Grasp Convention

Let g be the coordinate frame of the gripper and o be the coordinate frame of the object. As of April 25, 2015, these coordinate frames are as follows:

(1) **Frame** o: Centered at the object centroid $\mathbf{c} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{p}_i$ where $\mathbf{p}_i$ are the object mesh vertices and N is the number of vertices. Oriented along the object principal axes, meaning that if you do PCA on the object vertices then the first principal component is the z-axis, the second component is the y axis, and the x axis is orthogonal to that basis.
(2) **Frame** g: The gripper frame according to TF, which I believe is called 'l_gripper_tool_frame.' This is the frame used by Rviz in the grasp_viewer.launch file.

Grasps are stored as the pose of the gripper with respect to the object frame of reference T_o^g. However, the gripper frame of reference in OpenRAVE r appears to be different. We manually estimated the relative transform

$$T_g^r = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & -0.0375 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Current Implementation: In existing scripts T_g^r is referred to as T_{fix} and the coordinate frame r is the frame of the 'leftarm_torso' manipulator.

Finally let w denote the world reference frame of Rviz and o' denote the object reference frame in a different basis (e.g. with respect to the center of the object bounding box, as computed from Meshlab). We have the pose of the alternate object frame with respect to world frame $T_w^{o'}$ from our setting of its position in OpenRAVE. We also have the pose of the centered object frame with respect to the alternate object frame $T_{o'}^{o}$ from the known location of the object centroid. Then the pose of the OpenRAVE gripper in the OpenRAVE world frame is

$$T_w^r = T_w^{o'} * T_{o'}^o * T_o^g * T_g^r$$

If our models are stored in the coordinate frame o instead, we have the more compact relative transform

$$T_w^r = T_w^o * T_o^g * T_g^r.$$

Current Implementation: In the existing script *src/pr2_grasp_checker.py* the object vertices are assumed to be stored in frame o and the world frame w is assumed to be the frame of the PR2. For ease of viewing, the robot is moved instead of the object, which means that the original PR2 frame is set to be relative to the object center.

Date: April 25, 2015.

Therefore the computation you see on lines 104 and 107 of *src* *pr2_grasp_checker.py* does the following

$$T_w^o = T_w^r * T_r^g * (T_o^g)^{-1} \qquad \text{Line 104}$$

$$T_o^w = (T_w^o)^{-1} \qquad \text{Line 107}$$

and then sets the robot to have this transform.