

# Robotica Autonoma Homework 1

## *Perception*

Elena Camuffo, Emilio Olivastri, Edoardo Trombin

October 23, 2021

### 1 Problem Setup

The initial setup given includes a robotic arm (marrtino), placed on a table and a set of 8 cubes, 4 blue and 4 red, each with an *apriltag* attached. A kinect is placed above the scene to detect the cubes poses, by means of the apriltags recognition. The task of the homework is to first place the cubes in the area detectable by the kinect and then compute the pose of each cube, both with respect to the camera coordinates (kinect) and the world coordinates.

### 2 Development

The pose (i.e. position + rotation) of the cubes with respect to the **camera** frame, are retrieved from the *tag-detection* message. That pose's values are then computed, as explained in the following, to finally retrieve the pose with respect to the **world** frame. A message containing both the poses is returned to the user. That message contains also the **ID** of the cube associated to the apriltag detected.

The transformation of coordinates is shown in figure 1.

For each cube, the initial **position** is represented by  $\mathbf{x}_c$ , while the final coordinates are represented by  $\mathbf{x}_w$  and  $\mathbf{r}$  represents the position of the camera in the world frame.

The following equation gives the position of the cube in the world frame:

$$\mathbf{x}_w = \mathbf{r} + \mathbf{R}\mathbf{x}_c$$

Where  $\mathbf{R}$  is the **rotation matrix**, whose values are extracted from the information about the orientation of the camera relative to the world, expressed in *quaternions*.

That information is exploited also to compute the **orientation** of the cube with respect to the world, and thanks to the quaternions representation we are able to compute it as:

$$\boldsymbol{\theta}_w = \boldsymbol{\theta}_r \cdot \boldsymbol{\theta}_c$$

where  $\boldsymbol{\theta}_w$  is the orientation of the cube with respect to the world,  $\boldsymbol{\theta}_c$  is the orientation of the cube with respect to the camera and  $\boldsymbol{\theta}_r$  is the orientation of the camera with respect to the world.

The implemented method follows the theoretical guidelines proposed for this homework, performing a matrix multiplication to obtain one frame from the other. However, also a lighter implementation in terms of computational requirements is inserted in this project, in order to evaluate the relative performances and check the results.

This second procedure makes use of a library function, the *doTransform* method, an optimization with respect to the matrices operations previously used. This procedure is selected by default when the node is started, and it is named **Standard Procedure**, while the previously implemented one is named **Custom Procedure** and needs to be manually selected by the user.

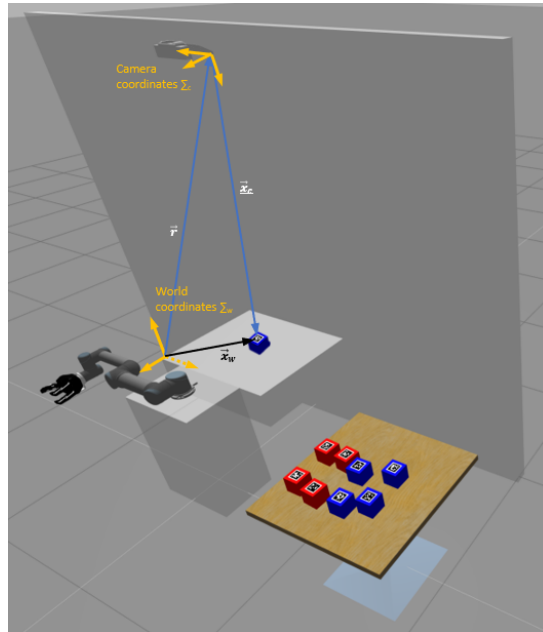


Figure 1: Representation of the coordinate systems and the change of coordinates.

### 3 Conclusions

Comparing the outputs of our procedures, their results are proven to be almost equal to the output obtained with the command `roslaunch tf_echo /world /tag_i1` which automatically retrieves the poses of the cubes in the world coordinates. There is a slight difference in the custom solution (an error of order  $0.01m$ ) that is due to the manual computation of the values, in fact the procedure using the *doTransform* function gives results that are a little more accurate.

The standard and custom procedures differs also in terms of **time required** to produce the output: the former gives on average  $2.1\mu s$  while the latter  $3.7\mu s$ , confirming that the library function is more convenient to be used and a manual computation is more time requiring and prone to errors.

<sup>1</sup>where *i* is the index of the cube