CMPUT 412: Experimental Mobile Robotics
Winter 2018
Demo #6
Date: March 22, 2018
Pose Estimation with QR-Code, AR-Code and Texture Object Model

Objectives

- Learn how to estimate the pose of a planar object with respect to a camera

Part I (Individual): QR-Code tracking with "vision_visp" (individual) 30%

Download and install the ROS package "visp_auto_tracker" on your computer. Run the example described in Section 6.1 of the wiki page to make sure that you have installed the package correctly. Print the example QR code on the wiki page, and use the tracklive_usb.launch file to duplicate the pose estimation exercise described in the Youtube video in Overview. Note that for this step to work, you need to set up your own camera (USB or Xtion Pro) and provide its intrinsic parameters to the tracker. Parameters within the launch file need to be set properly. **TA Demo**

Part II: AR-code tracking with "ar_track_alvar" (individual) 30%

Download and install the ROS package "ar_track_alvar" on your computer. Note that the wiki page describes how the package works for the PR2 robot. To get the package to work for your Xtion Pro camera, you need to

1. Download a sample AR code from this link or this.
2. Launch your Xtion Pro camera.
3. Launch ar_track_alvar with the launch file provided.
4. Move the camera in front of the printed AR code and echo the ROS topic /ar_pose_marker that describes the camera pose with respect to the marker.

Once this works, **write a ROS node** that projects the marker's pose into the camera image in a way similar to the Youtube video on the wiki page. **Demo & email your ROS node code to TA**

Part III: Pose estimation with a texture object (group) 40%

Study the OpenCV tutorial on Feature Matching + homography to find Objects.

1. Modify the code so that it works with ORB features rather than SIFT as in the tutorial, and it is able to determine the object in live camera view.
2. Create a model of the UA emblem in terms of its ORB feature locations.
3. **Write a Python/C++ program** to display the pose of the camera relative to the U of A emblem by using the solvePnPRansac() function explained on this page.
4. Display the pose estimation result by projecting the axes of emblem to the image as shown on the above page, using the live camera view. **Demo & email your ROS node code to TA (per group)**

Marking

If you are able to complete the demos before the end of the lecture, you will receive: 60% for Part I & II(independent demonstration) and 40% for Part III (group demonstration). *If a student or group is not*

able to complete any parts of the demo within the lecture session, you will get a 20% penalty of that marking component, and an additional 20% for each day of delayed demo.