Home Work 2

Robotics 2017 - ShanghaiTech University

1 Select paper to present (5 %)

Till Monday, October 9rd, 22:00 select one paper from the ICRA 2017 conference. See the list here: http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7960754. It has to be a regular paper (6 pages or more). Send an email to houjw@shanghaitech.edu.cn with the name and the link to the paper. The subject of the e-mail has to have "[Robotics]" in it.
Suggestion: Select a paper which is easy to present. Some papers have videos - but you cannot use the video in your presentation! Your presentation will be about 8 minutes (plus 2 minutes questions). If a paper consists mainly of math it might be hard to present.
More instructions for preparing the presentation will follow - now just select the paper, and start reading and understanding it. Two students cannot present the same paper - first come first serve!. We will present the papers in 4 extra sessions (times tbd.) (everybody will have to attend at least 2 sessions).

2 Robots. (10 %)

1. Make a list: Name the five coolest/ best/ most interesting robots (or types of robots) for **you**. Those have to be robots that actually exist(ed) and work(ed). (As long as you name five robots this answer is correct - it is up to you what robots you find interesting!)
2. For one of those robots, write at least three sentences **why** this is a cool robot.
3. Imagine a humanoid household robot. This robot should do lot's of things in the house, for example wash the dishes and put them back in the shelf, cook dinner, play with your kids, walk the dog, go shopping, etc. What do you think - what are the greatest challenges towards humanoid household robots? Name three to five challenges, max two sentences each.

Save your answers in a text file called "Robots.txt" in the root folder of your HW2 repo (see below).

3 ROS Tutorials (5%)

Work through the tutorials on http://wiki.ros.org/ROS/Tutorials/ till number 13. You do not need to do number 12 (Python).

4 Academic Honesty

Please everybody do the homework completely by yourself. You are doing this to learn about robotics and ROS. Do not ask other students for help. Ask in piazza! If the problem persists go to Xiangyang's office hour (see on piazza) and/ or ask Prof. Schwertfeger - we will help you!

5 Preparation

Make a new ROS workspace, e.g. hw2_ws:

```
mkdir -p ~/hw2_ws/src
cd ~/hw2_ws/src
```

Clone your (empty) HW2 repo (replace soerensch with your user id) (your HW2 repo will become available around Sep. 24):
`git clone git-teaching@robotics.shanghaitech.edu.cn:soerensch_HW2`
In the repository add a remote repo that contains the provided HW2 framework files:
`cd soerensch_HW2`
`git remote add framework git-teaching@robotics.shanghaitech.edu.cn:soerensch_HW2`
Go and fetch the files:
`git fetch framework`
Now merge those files with your master branch:
`git merge framework/master`
Now you should have the files in your master repo. If you do ls you should see the following:
`mobile_robotics_hw_2 pose2d_to_3d random_odom2d`
The rest of the git commands work as usual.

Now go and make the workspace:
`cd ~/hw2_ws`
`catkin_make`

Do not forget to always source "setup.bash" in the "devel" folder of your workspace for every new shell that you open!

```
source devel/setup.bash
```

If everything went well up to here (i.e. no errors during compilation) you are ready to proceed with the actual homework. If there was a problem, try your best to solve it - maybe ask on piazza.

6 Calculate Global Pose Estimate (40%)

Your task is to calculate the global pose estimate of a robot in 2D. Input are 2D odometry estimates published on topic "/odometry2D". The odometry as well as the pose are of message type "mobile_robotics_hw_2/Pose2DStamped", that can be found in the downloaded code. Looking at Pose2DStamped.msg you will see that it consists of a "Header" and a "geometry_msgs/Pose2D". You can find out more about the messages by executing:
`rosmsg show mobile_robotics_hw_2/Pose2DStamped`

Your task is to use 2D geometry to calculate an estimate of the global position of the robot based on the odometry input. Your global reference frame should have the name "global". The first odometry estimate received should be relative to this global frame.

Make sure to publish the pose with the correct frame_id and the timestmap of the received odometry message.

File "mobile_robotics_hw_2/src/odometry.cpp" has everything you need already prepared - you just need to do the calculation and fill the corresponding message! You can add this around line 19.
`gedit mobile_robotics_hw_2/src/odometry.cpp &`

For every line of code that you write, add one line of comments (starting with "//"), describing what the line of code does.

Compile by going into the root of your workspace and execute "catkin_make".

Test by running:
`rosrun mobile_robotics_hw_2 mobile_robotics_hw_2_node`

See the "readme.txt" on how to send a test odometry message and how to replay and record the bag files.

Create a bagfile named "poses.bag" from the content of the **whole** "odometry.bag". So you have to make sure that you start your node first, then start the rosbag record, and only then start the rosbag play!

6.1 Purely Optional

If you are bored and want to play around with more random odometry values, you can start the node "random_odom2d". This will publish random messages of type "mobile_robotics_hw_2/Pose2DStamped" on the topic "/odometry2D". You can take a look at "random_odom2d/src/random_odom2D.cpp" for more details.

7 Calculate and Publish Speed (20%)

Additionally to the pose, calculate the speed of the robot in units per second (a unit in ROS is by default meter). Publish the speed of type "std_msgs/Float64" under the topic name "/speed". Add all the needed code to "odometry.cpp". Record a bagfile called "speed.bag".

8 Display Pose and Path in Rviz (20%)

Rviz is a ROS program to display robot data. It does not handle our 2D pose message, so a converter to 3D pose and path message is provided. Run: `rosrun pose2d_to_3d pose2D_to_3D_node`
This will publish a "geometry_msgs/PoseStamped" and a "nav_msgs/Path" on the topics "pose3D" and "path3D", respectively. It listens to the output of your program on "pose2D".

You can run your program live together with the "pose2D_to_3D_node" or you can play your recorded "poses.bag".

Start rviz with:
`rosrun rviz rviz`
Add a path display and a pose display and set the according topics in those displays. You also need to set the Fixed Frame in Global Options on the left to "global". Also try what happens when you set Fixed Frame to "pose2D_to_3D_node".

Show the complete path and the last pose of "odometry.bag" in rviz, using "global" as fixed frame, and make screen shot. Make sure to also show (parts of) the relevant terminals in that screen shot. Save the screen shot in "mobile_robotics_hw_2/" under the file name "rviz.png".

9 Submission

Your submission consists of the framework with your changes, the bagfiles and the image, all commited to git and pushed to the robotics server. The following files are important:

- "Robots.txt" from the first task.
- Everything that was in the original "mobile_robotics_hw_2/" folder - The code must compile and be the code you used to create the .bag file!
- Especially "odometry.cpp", including your code and the comments.
- "poses.bag" created by you!
- "speed.bag" created by you!
- "rviz.png": the screen-shot from rviz.