Ros Nao Tutorial

Caitlin Lagrand, Michiel van der Meer & Arnoud Visser

Future of Rescue Robot Simulation workshop,
Leiden, March 4, 2016
Simulation & Real robots
Humabot Challenge

www.irs.uji.es/humabot/
Humabot Challenge

Standard Environment

DUKTIG
Play kitchen

HUMABOT

RobotProgramming.Net

Home ➔ My courses ➔ Competitions ➔ HUMABOT ➔ Kitchen environment ➔ Side Cameras
Bio-inspired perception

Some Nonaccidental Differences Between a Brick and a Cylinder

Brick

- Three parallel edges
- Three outer arrow vertices
- Inner Y vertex

Cylinder

- Two parallel edges
- Curved edges
- Two tangent Y vertices
- (Occluding edge tangent at vertex to discontinuous edge)

Bio-inspired perception

Bio-inspired perception

Bio-inspired perception

Colour based + Contours
Circle based + Colour
Colour invariant blob detection
**TABLE 1. RESULTS OF THE DIFFERENT DETECTING ALGORITHMS**

1 MEANS DETECTED TOMATO, 0 MEANS DETECTED NOTHING

<table>
<thead>
<tr>
<th></th>
<th>color based</th>
<th>Circle based</th>
<th>Blobs</th>
</tr>
</thead>
<tbody>
<tr>
<td>tomato</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>carrot</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>cucumber</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>garlic</td>
<td>0</td>
<td>background</td>
<td>0</td>
</tr>
<tr>
<td>lettuce</td>
<td>0</td>
<td>0</td>
<td>lettuce</td>
</tr>
<tr>
<td>all1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>all2</td>
<td>1</td>
<td>0</td>
<td>garlic</td>
</tr>
<tr>
<td>all3</td>
<td>1</td>
<td>1</td>
<td>1 &amp; background</td>
</tr>
<tr>
<td>without1</td>
<td>0</td>
<td>0</td>
<td>garlic</td>
</tr>
<tr>
<td>without2</td>
<td>0</td>
<td>0</td>
<td>carrot</td>
</tr>
</tbody>
</table>
Localization

\[
X_{\text{robot}} = \frac{\text{RADIUS\_TO\_METERS}}{\text{radius}}
\]
\[
Y_{\text{robot}} = \left(\frac{\text{REAL\_WIDTH}}{\text{radius}}\right) \times \left(\frac{\text{img\_width}/2 - x_{\text{img}}}{2}\right)
\]
\[
Z_{\text{robot}} = 0.35
\]
### TABLE 2. RESULTS OF THE LOCALIZATION ALGORITHMS, IN METERS

<table>
<thead>
<tr>
<th>Real x</th>
<th>Real y</th>
<th>Estim. x</th>
<th>Estim. y</th>
<th>Diff. x</th>
<th>Diff. y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.22</td>
<td>0</td>
<td>0.23</td>
<td>-0.006</td>
<td>0.001</td>
<td>-0.006</td>
</tr>
<tr>
<td>0.22</td>
<td>0.076</td>
<td>0.27</td>
<td>0.077</td>
<td>0.046</td>
<td>0.001</td>
</tr>
<tr>
<td>0.22</td>
<td>-0.076</td>
<td>0.27</td>
<td>-0.076</td>
<td>0.046</td>
<td>0.000</td>
</tr>
<tr>
<td>0.30</td>
<td>0</td>
<td>0.31</td>
<td>0</td>
<td>0.009</td>
<td>0</td>
</tr>
<tr>
<td>0.30</td>
<td>0.076</td>
<td>0.31</td>
<td>0.062</td>
<td>0.009</td>
<td>0.014</td>
</tr>
<tr>
<td>0.30</td>
<td>-0.076</td>
<td>0.39</td>
<td>-0.071</td>
<td>0.086</td>
<td>0.005</td>
</tr>
<tr>
<td>0.38</td>
<td>0</td>
<td>0.37</td>
<td>0.001</td>
<td>0.004</td>
<td>0.001</td>
</tr>
<tr>
<td>0.38</td>
<td>0.076</td>
<td>0.37</td>
<td>0.061</td>
<td>0.004</td>
<td>0.015</td>
</tr>
<tr>
<td>0.38</td>
<td>-0.076</td>
<td>0.54</td>
<td>-0.076</td>
<td>0.164</td>
<td>0.000</td>
</tr>
</tbody>
</table>
Motion Planning
Humabots environment

Environment ported from Webots to Rviz / MoveIt.
ROS pipeline

Camera (Image) -> Detecting/localization (Point) -> Movelt

Pose2D

Walker
Result
Tutorial

Software for day 5
To have multiple robots in their own namespace, you could follow the instructions of the pioneer3at_demo provided by Stefan Kohlbrecher and Nate Koenig.

What is needed for to control a real robot is a working version of the simulation of a Nao robot.

- Follow the steps of the ros Nao installation
- mkdir ~/naoqi
- Download pynaoqi-python2.7-2.1.4.13-linux64.tar.gz
- cp ~/Downloads/pynaoqi-python2.7-2.1.4.13-linux64.tar.gz ~/naoqi
- cd ~/naoqi
- tar xzf pynaoqi-python2.7-2.1.4.13-linux64.tar.gz
- echo 'export PYTHONPATH=~/naoqi/pynaoqi-python2.7-2.1.4.13-linux64:$PYTHONPATH' >> ~/.bashrc
- python
  inside python shell
  >>> from naoqi import ALProxy
  >>> quit()

If this works, the NaoQi Python bindings are correctly installed. Continue with the ROS packages for the Nao robot.
- sudo apt-get install ros-indigo-nao-robot
- sudo apt-get install ros-indigo-nao-bringup
- sudo apt-get install ros-indigo-naoqi-bridge
- sudo apt-get install ros-indigo-naoqi-extras
Conclusion

In this workshop:

• we have a simulated arena from two virtual rescue competitions
• we have used two protocols
• we have experimented with multiple robot settings
• we have seen that the ros-code could be directly applied to real robots

• **Robots, Games, and Research: Success stories in USARSim**
  A [full day workshop](#) held at IROS 2009
  Steve Balakirsky, Stefano Carpin and Mike Lewis

• **USARSim/MOAST: Highly Realistic Simulation and Control for Multi Robot**
  A [full day workshop](#) held at ICRA 2006
  Stefano Carpin, Mike Lewis, Adam Jacoff, and Stephen Balakirsky

• **Urban search and rescue: from Robocup to real world applications**
  A [full day workshop](#) held at IROS 2004
  Stefano Carpin, Andreas Birk, Daniele Nardi, Adam Jacoff and Satoshi Tadokoro