Karachi Chotu
Team Description Paper 2014
RoboCup @Home League

Imran Hemani, Nabeel Zahid Rajput, Safeer E Hussain, Sajjad Haider

Artificial Intelligence Lab, Faculty of Computer Science
Institute of Business Administration, Karachi
Karachi, Pakistan.
http://ailab.iba.edu.pk/TeamKC/

Abstract. This paper describes Team Karachi Chotu and its robot TurtleBot. The underlying architecture of the TurtleBot is based on the Robot Operating System (ROS) which is an open-source meta-operating system for the new generation Robots. The paper also describes different packages associated with ROS that are related to perception, depth, and gesture recognition. We present important modules of our robot control software which allows us to perform reliable service/social robotics applications in the @Home league.

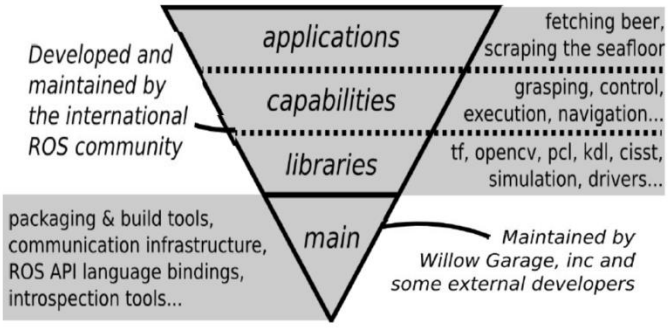
1. Introduction

Karachi Chotu was formed in the mid of 2013 within the context of a research project "The Helper Bot". The team consists of three undergraduate students and is being supervised by Dr. Sajjad Haider. IBA being one of the premier higher education schools in Pakistan is also committed to robotics related education and has taken several pioneering steps to introduce robotics at high school, undergraduate, and graduate levels. It has been participating in the World RoboCup Soccer Simulation 3D league since 2011 under the team name Karachi Koalas. The team was ranked 5th in the last year World RoboCup that was held in Eindhoven, Netherlands.

Being a new team in RoboCup@Home League, we had to develop everything from scratch. The rest of the paper describes the development, environment and code architecture of Karachi Chotu. In addition, high priority tasks that we aim to finish before the competition are also discussed.

2. Development Environment

We have used C++ as the programming language in addition to the libraries provided by ROS. We are using ROS Groovy and Catkin - *A CMake-based build system that is used to build all packages in ROS.* We have configured Eclipse IDE to work with the ROS libraries.



3. Robot Description

TurtleBot is a low-cost robot that works with an open-source software, ROS. TurtleBot can move around with the help of wheels that are fitted in the base of the TurtleBot which acts as motor. It can see in three dimension with the help of Microsoft Kinect or Asus Xtion Pro; both of which uses the same library of OpenNI which makes TurtleBot independent of the device used. It has enough horsepower to create exciting social applications that can be shown at the RoboCup @Home League

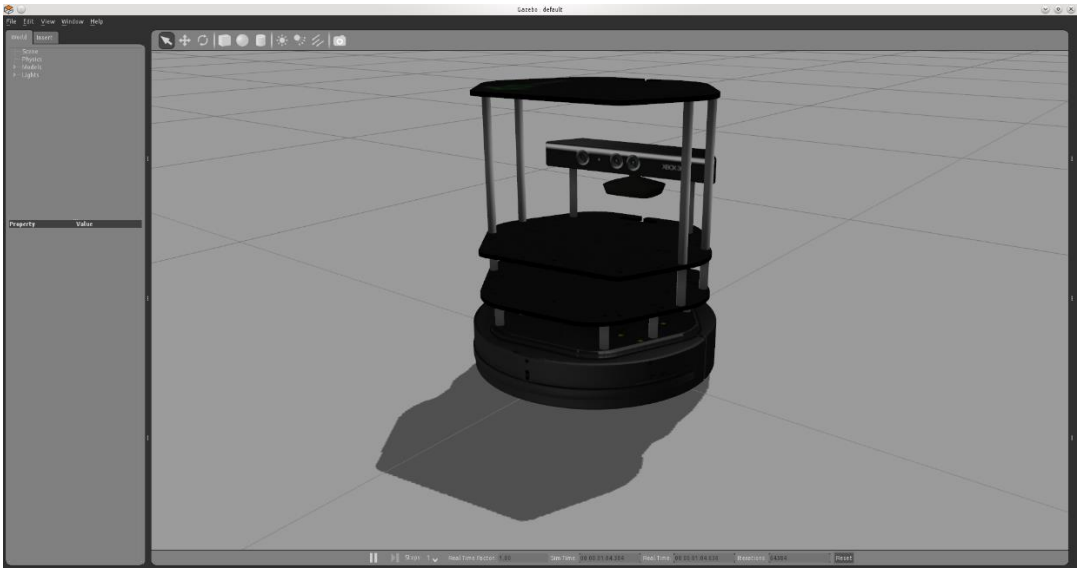
3.1 Hardware

The main hardware includes:

- Kobuki Base
- Asus Xtion Pro
- Asus Xtion Pro Mounting Hardware
- TurtleBot Structure
- TurtleBot Module Plate with 1 inch Spacing Hole Pattern

4. Software Architecture

We have used ROS C++ libraries for development. ROS is an operating system for robots which operates in a client/server mode. ROS is composed of different packages that include nodes, topics, services and parameters. Nodes, called also rosnodes, are the executables programs. Topics and services are ways of communication between nodes. Services rely on a query made by a given node or from a terminal to get a response from the node offering the service. Topics require a subscription to a node that will be broadcasting some particular info.



A snippet of TurtleBot Simulator

4.1 OpenNI

OpenNI is an open-source framework for "natural interaction" - using your hands and body to interact with your digital devices. The robot must integrate the features of face recognition, people recognition, and object recognition in order to take orderly decisions. *Asus Xtion* is used for image and vision data. The features used in the project are:

Tracking: For person tracking, the tracker, a feature of the *OpenNI* driver, was employed. *Asus Xtion Pro* is used to get the data

Face Recognition**:** The face detector employs the *OpenCV* face detector to obtain an initial set of detections. It then prunes false positives using stereo depth information. The depth information is used to predict the real-world size of the detected face which is then preserved as a true face detection only if the size is realistic for a human face. This removes the majority of false positives given by the OpenCV detector. The ROS *face_detector* is used to obtain the results.

4.2 Point Cloud

A Point Cloud is a set of data points in any coordinate system. PCL is used in conjunction with ROS through Point Cloud Library Framework. PCL contains a number of state-of-the-art algorithms that can be embedded to attain added functionality. It includes algorithms for segmentation, filtering, feature extraction and many more. A robot follower is being implemented using the point cloud library via 3d Points.

4.3 Computer Vision (OpenCV)

OpenCV is a library of programming functions for real time computer vision. It makes image manipulations easier and also helps in developing complex vision based algorithms.

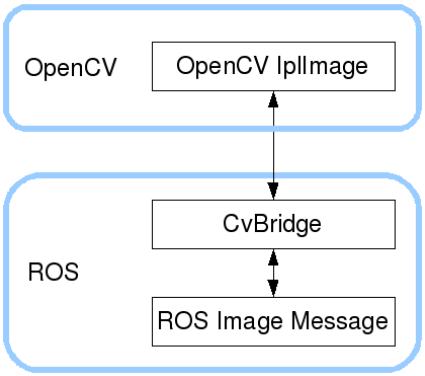
For the integration of OpenCV with ros, **vision_opencv** package is used which includes several packages:

cv_bridge: Bridge between ROS messages and OpenCV.

image_geometry: Collection of methods for dealing with image and pixel geometry

Cv_Bridge

ROS passes around images in its own sensor_msgs/Image message format but many users will want to use images in conjunction with OpenCV. CvBridge is a ROS library that provides an interface between ROS and OpenCV.



5. Functionalities

In this section, the functionalities that have been added to the TurtleBot are presented.

5.1 Object Recognition

It is necessary for service robots to recognize objects that are in front of them to perform an action accurately. For this purpose, TurtleBot is equipped with Asus Xtion Pro. The Xtion Pro works as “eyes” of the robots. It provides the knowledge of the surroundings and the objects around the robot. However, the robot has to recognize different objects using different algorithms. We have used PCL[1] (point cloud library). The robot is capable of recognizing a ball among different objects using the attributes such as shape and color. For the initial stages, we have used a specific color ball.

5.2 People Detection

Awareness about people in the surroundings helps the robot to operate safely while co-existing with humans and to react to their movements and actions. We have used Asus Xtion Pro for people detection in 3D. Early testing has shown that the robot is able to detect multiple people and can follow one of them accurately.

[1] For more information on Point Cloud Library, please refer to http://pointclouds.org/

5.3 Human Robot Interaction

Interaction between human and robots is an integral part of RoboCup @Home League. Most humans present their intentions in the form of speech, gestures, or facial expressions. The social robot should be aware of those intentions and should be able to understand them to react accordingly. We have used only one approach to carry out this task, that is, gesture recognition.

6. Conclusion

The paper presented Karachi Chotu RoboCup @Home team. We are using TurtleBot as our hardware for this project. We are working on ROS which serves as the meta-operating system. The paper described the underlying architecture of our team along with the various frameworks that we are using in addition to ROS.

References

- https://github.com/ros/catkin
- ROS: Robot Operating System for AGV KGP by Jit Ray Chowdhury
- http://www.turtlebot.com/
- http://wiki.ros.org/Robots/TurtleBot
- The b-it-bots Team Description Paper 2013
- AllemaniACs Team Description Paper 2009
- http://wiki.ros.org/vision_opencv
- http://wiki.ros.org/cv_bridge