SKUBA 2013 Team Description

Kanjanapan Sukvichai, Krit Chaiso, Thanakorn Panyapiang, Jaktip Yordsri, Suppawit Inhorm, Tachin, and Sutinai Tanalerkchai

Faculty of Engineering, Kasetsart University
50 Ngamwongwan Rd, Ladyao, Bangkok, Thailand
skuba2002@gmail.com
http://iml.cpe.ku.ac.th/skuba

Abstract. This paper is used to describe the SKUBA @home League robot team from Thailand. SKUBA@home is designed under the World Robocup 2012 rules. Based on the last participation, this year we're focusing on the new base platform which designed by using mecanum wheels and concentrating on ways to improve the performance of object recognition. The overview describes both the robot hardware and the overall software architecture of our team.

1 Introduction

SKUBA @home was established in 2011. In 2012, SKUBA @home made the first participation in Robocup Japan Open 2012 and made the way through finalist. Furthermore, SKUBA @home joined the World Robocup 2012 in Mexico and managed to pass to the 2nd stage as we anticipated. From last year issued, we decide to improve our robot performance by using mecanum robot platform in order to solve the "Walking in Elevator" issue. Thus, this year we hope to complete the "Follow Me" task. Furthermore, Team has developed a new object recognition technique.

The next section will explain about our robot that is designed for further research in the future. In the 3rd section is about software architecture and also includes object recognition algorithm. Section 4 will explained is about low level design and model of the robot including robot motion and robot odometry estimation. Finally in the last section, we will present the conclusion.

2 Robot Hardware

The new SKUBA@Home platform is designed as a three layer platform. The first layer is the driving mechanism layer. Second layer is a robot body which can be moved in vertical and the last layer is robot head and arm. Robot driving mechanism base consists of four 8" mecanum wheels. Each mecanum wheel is driven by Maxon EC 45 flat (brushless motor, 70 watt) BLDC motor combined with planetary gearhead of 1:36 gear ratio. The Hokuyo laser length finder is attached to the lower layer in order to obtain the environment information which

will be used in SLAM algorithm. This mecanum wheel has more mobility than the regular fixed wheel since it provides side movement. Now the robot can easily avoid the obstacles and has more flexibility to maneuver to the messy environment. The robot mecanum base is shown in Fig.1 below.

Fig. 1. The SKUBA@Home robot base

The second layer is the robot body. The robot body can be moved along vertical axis by using sliding bars which are driven by 70 watt DC windshield wiper motor. The robot head and arm are attached to this robot body. The Hokuyo laser length finder is also attached to the center of robot body in order to use in human tracking algorithm. The final layer is the robot head and arm. Robot head has two degrees of freedom neck which is duplicated from the human neck behavior. Kinect sensor is fixed to the top of the neck which can be used as the human eyes. High torque Dynamixel MX-106R smart servo, Dynamixel RX-28 smart servo and BLDC Maxon motor are used to construct the robot arm. Robot arm has 6 degrees of freedom which can perform more complex tasks. The robot arm is shown in Fig.2 .

Fig. 2. Robot arm

3 Software Architecture

Software system of our robot is divided into many modules with specific functionality, for example, object recognition module and planning module. The communication between modules is implemented by using Robot Operating System (ROS). The modules can be organized into three different layers:*perception layer*, *control layer* and *task layer*.

Perception layer consist of modules about environmental understanding. Speech recognition, Object recognition and Localization, for instance. Modules in this layer collect data from sensors, i.e. laser range finder, microphone and Kinect, to perform higher-level algorithms in order to identify state of the environment. Output of this layer is intermediate data for *task layer*

Task layer control robot's behaviour to solve complicated task.

When *Task layer* send command to robot, *Control layer* will interpret those commands into lower-level actions by using path planning algorithms.

Fig. 3. High level software architecture

3.1 Voice System

Speech recognition is implemented by using *pocketsphinx* base on *CMUSphinx* speech recognition toolkit. The incoming audio will be split into utterances and converted into sequence of words. The system require language model and dictionary which can create from words and sentences corpus,that we generate specifically for each task, by using Online Sphinx Base Tool.

The e-speak is selected as the robot speech synthesis software which uses a "formant synthesis" method. This allows many languages to be provided in a small size. The speech is clear, and can be used at high speeds, but it is not as natural or smooth as larger synthesizers which are based on human speech recordings.

3.2 Object Recognition

In field of recognition, the most essential are localization and recognition, because they are deeply associated to each other. Recognition requires localization; and reversely, localization requires recognition.

Improving the precision of recognition module, we enhance localization approach by using both RGB image and depth image together. For localization process, we apply depth information from OpenNI Library for extracting objects from the background scene using Euclidean Clustering Extraction (ECE). Additionally, we use some explicit constrains, e.g., table plane boundary and manipulator work space limitation, to distinguish the cloud that is definitely not the interested object such as, the objects that are beneath the table. After computing the centroid of extracted point cloud, we transform the object's border using the focal length equation from 3D space to RGB domain. From object's border, the segmented image is obtained. In image domain, descriptors of the segmented image are extracted using Speeded Up Robust Features (SURF), and can be classified by recognition process.

About recognition process, initially, we cluster the extracted descriptors using K-means clustering. Next, the histogram is created by counting the number of descriptors in each cluster. Finally, the histogram used as the parameter is passed into SVM model which is supervised learning for classification the object category.

The performance of this approach gains us the higher efficiency of both localization and recognition. First of all, localization using depth information has the acceptable accuracy for scoping the object boundary from the background scene. Lastly, recognition process has a significance of accuracy in classification comparing to our old method. Finally, result of this module is shown in Fig.4.

Fig. 4. Result from object recognition module

3.3 Gesture Recognition

NITE algorithm is the algorithm built in OpenNI framework can retrieve motion gesture such as wave, circle, swipe, push and steady. This algorithm return with hand position of all gesture detects. Moreover, it can be detect more than one gesture at a time. Therefore, we select NITE algorithm as our gesture recognition algorithm.

3.4 Face Recognition

Face recognition is basic ability for service robot to learn and classify human. For example, Serving meal and beverage. To complete this task, the robot has to memorize who is owner of the order. To perform face recognition, Elastic Bunch Graph Matching (EBGM) algorithm is chosen because of robustness to oriented variation. This algorithm consist of

3.5 People Detection and Tracking

For following person, robot need two important ability which is detection and tracking the interested person. Our development in people detection use the RGB-D, point cloud, data as the input for performing the clustering of people. Avoiding the problem of detection such as merging cloud of more than one person or occlusion, sub-clustering of people's head is considered to use as the representation of each person. Another component of this process, tracking is implemented based on the Likelihoods computation, Joint likelihood data association, and HOG-based initializer.

4 Motion Control and Planning

4.1 Motor Control Module

Motor control module is composed of a collection of embedded controllers which executes the low level motor control loop, communication and debugging. First controller is base motor controller. The motor controller, increment quadrature decoder, PWM generation and onboard serial interfaces are implemented using FPGA. The PI controller is employed to achieve each wheel velocity. And the communication between controller and computer is established by using RS232. Furthermore, there is different controller to control arm motor. To interact with it, RS485 standard is selected and build in to the controller. For convenient usage, second controller also have PID-Torque control, torque limitation and orientation adjustment[4].

4.2 Localization and Path Planning

Adaptive Monte Carlo Localization system (AMCL), provided by *navigation* stack, is used for robot localization. For satisfy localization algorithm, Hokuyo laser length finder which acquires environmental data is attached to the robot. Additionally, the robot odometry information is also considered as one input information to localization algorithm. Another input is the known map which is predefine or dynamical construct. From previously describe input, AMCL which is a particle filter localization system accurately estimates position and orientation of the robot.

Predefine and dynamical map is acquired by using *hector_mapping* package within *hector_slam* stack. This package requires only length finder data. To construct an occupancy grid map, it performs laser beam matching using Gauss-Newton approach[5].

About path planning module, ??

Odometry Estimation Module This module helps to minimize the mean square error of the non-perfect sensor measurement originated from wheel slippage, dynamic surrounded environment and IMU drift over time. Using the laser scanner to estimate the change of position by laser scan matching technique can solve the wheel slip problem while facing with high sensitivity to dynamic environment. Point-to-line distance Iterative Closed Point (ICP)[6][7] method is used to be based algorithm for laser scan matching.

To perform estimation module, Kalman filter which consist of two steps is used. First, the Kalman's observation step calculates different between output from last prediction and actual position which combine laser scan matching and wheels' speed together. Next, the Kalman's prediction step predicts new output by compensate error from previous step and IMU data.

Collision Avoidance Module Object Avoidance

4.3 Manipulation Module

2 Grasping Inverse kinematric Grasping action Action Inverse kinematric

In manipulation part, our development of grasping action is classified to 2 method, Grasping and Inverse kinematic, for different kind of situation.

In manipulation part we develop and improve grasping action method For the manipulation system we develop in two main methods by situation there are Grasping and Inverse kinematic, Grasping action method is the method that classify object by shape of object then computer will select gesture action from database and send command to robot arm. In generally this method use with object that have a complex shape or implement with another task. Inverse kinematic is method that find the center point of object and send direct velocity to command robot's arm so this method use with simple object eg. can box bottle.

5   Conclusion

In this year, we mainly focus on developing more stable robot and also it's AI software. We'll improve performance of manipulator as future work. We have experiences from RoboCup@Home competition 2012 and we learn from our mistakes. Our robot and AI are improved in order to join RoboCup2013 event in Netherlands. We hope that our robot team will perform better in RoboCup than the previous year, and we are looking forward to sharing experiences with other guest teams around the world.

References

1. Serken Tuerker, Euclidean Cluster Extraction,
   `http://pointclouds.org/documentation/tutorials/cluster_extraction.php`.
2. Radu Bogdan Rusu, Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments, Germany (2009).
3. M. Munaro, F. Basso and E. Menegatti. Tracking people within groups with RGB-D data. In Proceedings of the International Conference on Intelligent Robots and Systems (IROS) 2012, Vilamoura (Portugal), 2012.
4. T. Ariyachartphadungkit and K. Sukvichai, Development of an Embedded BLDC motor controller using RS485 standard, 2013.
5. S. Kohlbrecher and J. Meyer and O. von Stryk and U. Klingauf. A Flexible and Scalable SLAM System with Full 3D Motion Estimation, in Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), November, 2011.
6. A. Censi, An ICP variant using point-to-line metric, In Proceedings of the 2008 IEEE International Conference on Robotics and Automation (2008).
7. A. Censi, An accurate closed-form estimate of ICP's covariance, In Proceedings of the 2007 IEEE International Conference on Robotics and Automation (2007).