Cognitive Interaction with Robots

Safety & Control Improvement in Human-Robot Interaction

Authors:
Alex Pardo
Marc Bolaños
Pablo Pardo

Professors:
Cecilio Angulo
Marta Díaz

January 5, 2015

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Contents

1 Introduction

This project is comprehended within the teleoperated robots area. A broad topic recently being pulled further given the increasing need of elderly attention.

The robot we were working with was MASHI, which is a teleoperated robot with the purpose of guiding in "La Bóbila" social centre in L'Hospitalet del Llobregat. The root consists in an Arduino board embedded to a wheeled structure that resembles the human figure.

The environment the robot is going to work in the main entrance of the social centre, where some temporary expositions are hold. MASHI is supposed to welcome the visitors and guide them through the exposition being able to communicate with the visitors.

Our project is focused on two different but necessary aspects:

1. **Collisions Avoidance**: We developed a simple but efficient way to avoid collisions using an RGB-D camera attached in the neck of MASHI. Our method tackles the so called *Freezing Robot Problem*.

2. **Gesture Control**: Facilitate the teleoperator control of the robot by allowing him/her to use the hands to move the robot. This feature will allow the teleoperator to manage more complex movements such as control head, arms, hands and so on. We use a Microsoft Kinect to capture the depth image of the teleoperator and be able to recognise the gestures.

The Section 2 is a brief compilation of related works and common problems in the area. In the Section 3 we explain with all the detail how was carried out the implementation part. The Section 4 contains the experiments with users done in order to test the acceptance and efficiency of out work. In the last section (Section 5) we expose our conclusions of this project and the possible future work.

2 Related Work

2.1 The Freezing Robot Problem

The *Freezing Robot Problem* is a classic problem in robotics and consists of finding a balance between being conservative in terms of movement and safety and reach efficiently the goal. If the robot is too conservative and the environment too complex could end up freezing because none of the possible movements would be considered to be safe enough.

A classical approach to this problem is to use a planner and try to predict the movement of the surrounding in order to decide the next move, however these approaches usually fail to find a path when the environment is too complex. There are other approaches that tackle the problem from different perspectives. Peter Trautman and Andreas Krause [3] studied and proposed a non-parametric statistical model based on dependent output Gaussian processes that can estimate crowd interaction from data. Chung-Che Yu and Chieh-Chih Wang [4] proposed a Learning from Demonstration (LfD) approach proving that these kind of methods are efficient to both avoid collisions and freezing free navigation.

2.2 Gesture Recognition

Gesture Recognition is an important topic in computer vision and an active line of research for lots of companies. The reason behind this interest in the area is because of the multiple and revolutionary applications to the world of technology, it changes completely the conventional machine input mechanisms (mouse, keyboard and even touch-screens).

This technology has being successfully applied in the game industry, changing the gaming experience to a whole new level. Its also has being applied to device control such as tv or intelligent home systems improving the user experience.

One of the greatest advances in the topic was the improvement and large scale production of RGB-D cameras which started with the Kinect camera developed by *Microsoft* for the gaming industry. This busted the research in the area. There are countless examples of papers studying the area, a good example of them was presented by Lin Song1, Ruimin Hu, Yulian Xiao and Liyu Gong [2] in 2013 proposing a method for hand segmentation and gesture recognition in real time. Another interesting example was given by Yi Li [1] with a method capable of recognizing 9 different gestures with an accuracy between 84% and 99%.

3 Implementation

3.1 Collision Avoidance

We use the depth map given by the RGB-D camera from the robot to analyse whether there is collision risk. We use a sliding window to compute the minimum average distance among the patches (Eq. 1) and get the approximate distance to the closest visible object to the robot.

$$\min \frac{\sum_{w,h}^{i,j} depth(p_{i,j})}{w * h} < threshold \;\; \forall p \in Img \tag{1}$$

By setting two thresholds the robot sends a warning to the teleoperator, when there is an object close enough, or stops, when there is an object too close. Once the robot is freeze because of an obstacle the teleoperator can still move backwards (at its own risk) or turn around.

In order to tackle the *Freezing Robot Problem* we use a time window to store the closest distance and check if it disappears in time, i.e. if the sum of the distances in that window frame is larger that certain threshold then we consider that the object is a danger, otherwise the obstacle is just passing by and so not a real danger.

Since stop the robot abruptly might scare or disturb the people around we decided to reduce the speed of the robot once it enters to a warning zone inverse proportionally to the closest distance until the stop zone.

3.2 Gesture Robot Control

For the *Gesture Robot Control* we use a Kinect camera setted in the teleoperator side. It has two main phases, *depth image processing* and *gesture recognition*.

- **Depth Image Processing**: We use OpenNI sdk to extract the skeleton of the teleoperator. OpenNI skeleton tracking algorithm is not easy to find, however the Microsoft approach is to use a Random Forest classifier trained using a labelled dataset of depth images.

  The result is an efficient algorithm that works in real time.

- **Gesture Recognition**: We use a gesture recognition system based on right hand positions in the space (3D) and its variation over time, i.e. three axis and its two possible directions, which do 6 different gestures. The movement is controlled by an empirically tested threshold.

  Moving the hand in the depth axis will command to the robot to move forward or backward depending on the direction of the hand's movement, moving the hand in the horizontal axis will send to the robot the order of turning right or left and finally the vertical axis will send an stop signal to the robot.

3.3 Communication

The **Anti-collision system** code runs on the server side and is able to stop the MASHI when an obstacle is close to the robot autonomously. Also blocking the user to move the MASHI forward if the obstacle still in the path.

The **Gestural Control** code runs in the user's machine, able to track the skeleton and to send the commands to the robot server via a websocket connection.

4 Test with Users

4.1 Experiment Design

We perform two different experiments, one for each peace of the project:

Collision Avoidance: Test the system against different possible environments with occlusions, tricky cases, moving obstacles, etc. With the feedback of the test we adjusted the different thresholds of the system, improving then the performance.

Gesture Control: To test this part the tester has to teleoperate MASHI using our gesture control system and perform some simple tasks. After the test the tester has to answer a short survey about the user friendliness and the efficiency of our system (Table 1).

Question	**Answer**
Did you find the system easy to use?	
Did you find it intuitive?	
Did the system respond accordingly to your expectations?	

Table 1: Survey questions

4.2 Experiment Results

5 Conclusions

References

[1] Yi Li. Hand gesture recognition using kinect. In *Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference on*, pages 196–199, June 2012.

[2] Yulian Xiao Liyu Gong Lin Song, Ruimin Hu. Real-time 3d hand gesture recognition from depth image. ICSEM, 2013.

[3] Peter Trautman and Andreas Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *IROS*, pages 797–803. IEEE, 2010.

[4] Chung-Che Yu and Chieh-Chih Wang. Collision- and freezing-free navigation in dynamic environments using learning to search. In *Technologies and Applications of Artificial Intelligence (TAAI), 2012 Conference on*, pages 151–156, Nov 2012.