Project Scopes of Work (SOW): Cocktail waiter robot

1. Group Members

1. Hsuan Ting Kao ASUID: 1213386848
2. Chao Wei Kao ASUID: 1213382480
3. Chia-Yu Hsu ASUID: 1211265754
4. Chi Ping Hsiung ASUID: 1213433271 (outside-course members)

2. List of Abbreviations

1. ROS: Robot Operating System

3. Background

In recent years, significant advances have been made in computer vision. It increases the machine ability on human face detection and human pose estimation. With this development combined with planning and control of autonomous robot, there have been lot of applications and products about human robot interactions being applied to real world situations. In the service industry, robots serve as waiters in restaurants in China, Japan and Korea [1]. However, with limited technology, the existing robotic waiters can only follow and move along magnetic strips. They are novel and fun but not efficiency and lack of interactions between human and robots.

In this study, we aim to design a serving robot that is able to recognize the targeted people, plan a path visiting all of them, avoid obstacles and deliver food at front side of targeted people in an indoor scene. By using the depth information from depth camera and by adopting machine learning, the design principles of proximity and motion pattern in human robot interaction field, the optimized path to all the targeted people can be made.

The novelty of our solution is that our robot has more degrees of freedom. This brings the flexibility to work in different scenarios. The design of our robot includes ability to detect human face and principles of proximity so, compared to the traditional robotic waiters, ours can show more human-like activities, which are important in the service industry.

4. Scope

4.1. Basic Assumptions and Constraints

In our scenario, we only consider indoor situations. The people in the room do not surpass five people, and will not be walking or moving around. There will be some obstacles that our robot needs to avoid. Our robot also needs to move to the front side of people in order to act as a waiter. We build the system upon deep learning applications.

4.2. Functionalities

There are four main functions we want to implement:

1. Map the room and detect objects.

In order to find a path visiting all people, we need to understand the 3D geometry of the indoor scene. We will use the RGB and depth sensor on our robot to capture images and depth information. The robot we will use is TurtleBot2 [2] and the sensor is Asus Xion Pro Live, which the support distance of use is only 3.5 meter. Therefore, we propose using deep learning methods to help detect depth such like [3][4].

2. Detect human figures.

After generating the 3D geometry, we then distinguish people from obstacles. We will use OpenPose [5] to detect human body, hand and facial keypoints on the images. This not only helps us detect people but also their face direction so that our robot could move to their front side.

3. Calculate distances and optimize best routes to deliver cocktails.

The motion planning of the robot in our project is under two main constraints. The first one is that the robot needs to travel to all people while the second one is the robot needs to avoid all obstacles. Since we don't have a precise depth map at first, the locations of all objects might keep changing when the sensor takes new images and depth information. Therefore, we need a learning-based approach to motion planning of robotic systems.

4. Move to the front side of a person

When it approaches a person, it stops in front of the person as soon as it detects a face, otherwise it rotates around the person until it is facing the person. To get more accurate human face orientation, we plan to use Mobile Vision API [6] to perform face detection. After that, the robot stays for a while and it goes onto its next route after the person takes the drinks.

5. Other Technical Requirements

5.1. Usability

We analyze our usability from 5 aspects:

1. Efficient: The user just needs to place food and push the start button in under 1 minute. Then, our robot will automatically run the whole delivery process.
2. Effective: The robot will deliver food to all people. Since the robot scans all the people in party at first and people won't move around, it is unlikely that the robot will miss someone in the party.

3. Engaging: By using robots as waiters in the party, it not only saves the manpower, but makes the people in the party feel fun and enjoy this technology.
4. Error tolerant: Once there is error during delivery, the robot will try to return to the original or specified position so it won't stuck in the party.
5. Easy to learn: The robot just needs one button to activate. It will try to detect the natural behavior of people in the party so they don't have to learn any special gestures to interact with the robot.

5.2. Requirement on Ease of Use

The user interface is easy so the user just needs to place the food and push the start button. The user doesn't have to learn much to control the robot. And the people in the party don't have to learn any special gestures, either. They just need to act as their usual way.

5.3. Safety Requirement

During the delivery, the motion of the robot is not very stable so that the food should be placed steadily. And in case our robot bumps into some objects, the robot will move slowly and keep updating the surrounding views.

5.4. Scalability

Our robot has several scalabilities:

1. The number of robots: we can increase the number of robots and let them cooperate with each other. They can share the location information and complete the delivery task together.
2. The number of people: we can easily add more people in the party once we complete the task of three people.
3. The size of of the space: the problem of larger space is that we cannot have the whole room geometry at first. However, if we set up multiple cameras around the room, we can build up the room geometry even the room size is larger than the use of distance of our robot sensor.
4. The motion of people: once we have no problem with the scenario of still people, we can further apply the robot on moving people.

6. Project Plan

Table 1: Proposed Work Plan

Weeks	Milestones
0 - 4 (1/8 - 1/29)	We will finish project proposal and scopes of work.
5 - 6 (2/5 - 2/12)	Be familiar with ROS, connect the turtlebot and load room

	mapping information.
7 - 8 (2/19 - 2/26)	Detect objects and build up 3D room geometry.
9 - 10 (3/5 - 3/12) Midterm Presentation	Use OpenPose to locate human figures.
11 - 12 (3/19 - 3/26)	Robot motion planning
13 - 14 (4/2 - 4/9)	Robot motion planning with moving to front side of people
15 - 16 (4/16 - 4/23)	Extra functions, like moving people or adaptive motion planning.
17 - 18 (4/30 - 5/7)	Finish the project report and presentation.

7. Reference

[1] Nguyen, Clinton. "Restaurants in China are replacing waiters with robots." *Business Insider*, Business Insider, 26 July 2016, www.businessinsider.com/chinese-restaurant-robot-waiters-2016-7.
[2] *TurtleBot2*, www.turtlebot.com/turtlebot2/.
[3] Eigen, David, Christian Puhrsch, and Rob Fergus. "Depth map prediction from a single image using a multi-scale deep network." *Advances in neural information processing systems*. 2014.
[4] Godard, Clément, Oisin Mac Aodha, and Gabriel J. Brostow. "Unsupervised monocular depth estimation with left-right consistency." *CVPR*. Vol. 2. No. 6. 2017.
[5] CMU-Perceptual-Computing-Lab. "CMU-Perceptual-Computing-Lab/Openpose." *GitHub*, 1 Feb. 2018, github.com/CMU-Perceptual-Computing-Lab/openpose.
[6] "Mobile Vision | Google Developers." *Google*, Google, developers.google.com/vision/.