Efficient People-Searching Robot

Anh Pham, Mayang Parahita, Andy Tsang, Mathias Chaouche, James Rees, B.Sc.

School of Computer Science, University of Birmingham

Introduction

Mobile robots are often required to interact with people, and one such interaction is the ability to efficiently and robustly search for specific people. To be efficient, the robot should search using strong prior knowledge over the likely locations of a particular person. To be robust, person recognition must be reliable in the face of varying lighting conditions, dynamic obstacles and unreliable modes of interaction (speech recognition, face recognition).

In this project we present a system that has been engineered to be robust to these challenges, using strong prior knowledge over person locations and multiple face recognition methods to improve performance. The system shows promising results, particularly in navigation and face recognition.

Design

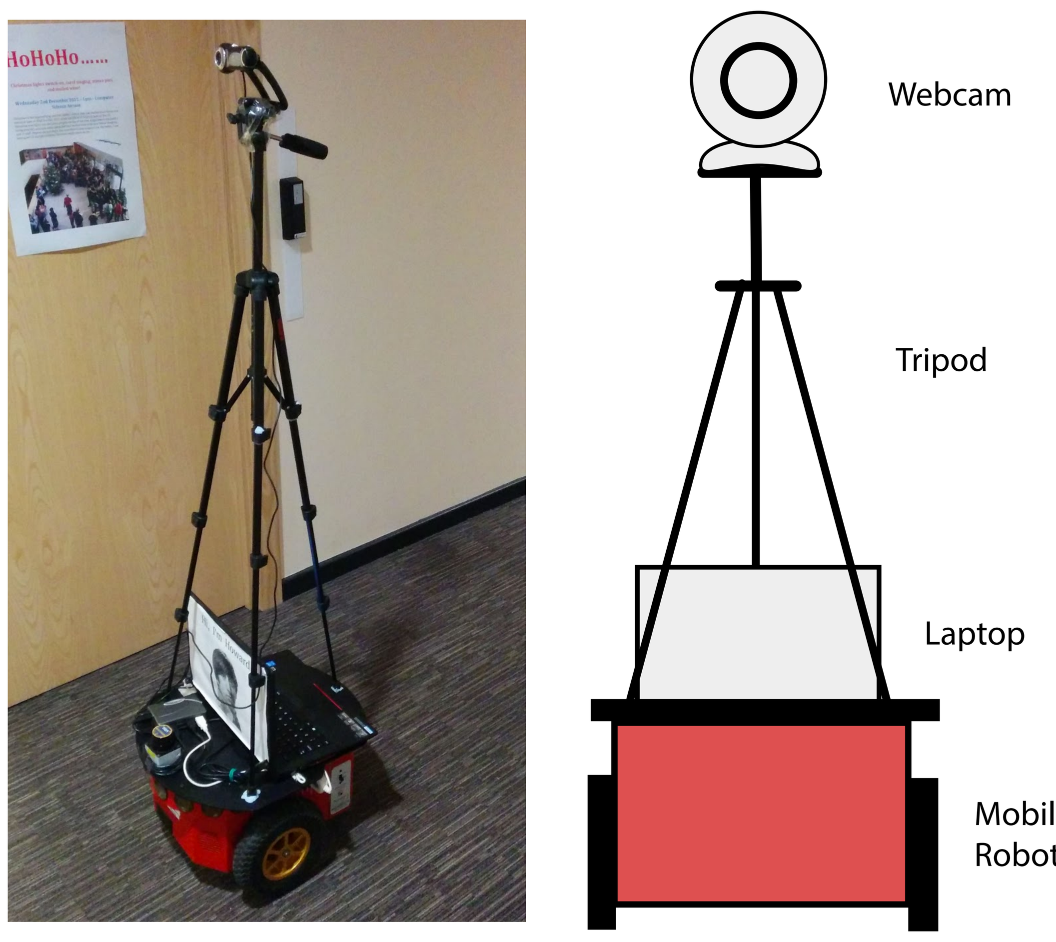

Figure 1. The physical set up of the system

The physical set up consists of a P3-DX mobile robot connected to a laptop with an external camera and microphone mounted on a tripod (height of 150 cm). A laser range scanner is included with the mobile robot. The system is made up of separate constituent components. The software is written in Python using Rospy, a popular library providing interface to mobile robots.

The robot is controlled by speech and keyboard input. A set of regular expressions is used to extract the command from the input. Speech input is implemented to provide a natural mean of communication, while keyboard is used as a fail-safe option, in case the speech recognition has difficulty getting the command.

The locations of people and rooms are stored in an SQL database. Each location-person pair is assigned a value corresponding to how likely it is to find that person at that particular location. The database is constantly updated after each task.

The robot uses Adaptive Monte-Carlo Localisation and Grid-based Navigation. Feed-forward control is applied to improve the driving performance. When failures in localisation are identified, the robot stops and tries to re-localise before continuing its task. The robot can also detect whether doors are open or closed to improve navigation and people recognition (figure 2).

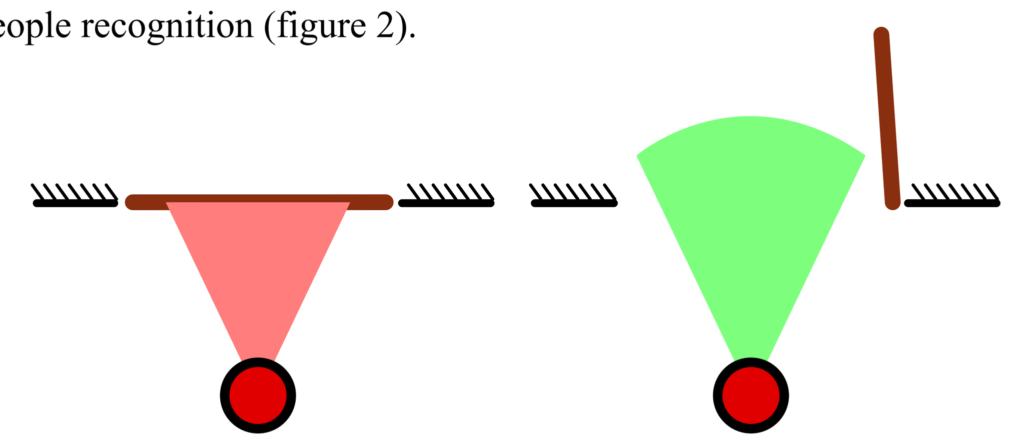

Figure 2. The door detection mechanism

The face recognition module is an ensemble of several state-of-the-art face recognisers: Fisher, Eigen and LBPH, combined using a plurality vote to improve the accuracy.

A speech synthesizer was implemented to provide user friendly feedback by vocalizing the current progress, as well as to request assistance from people.

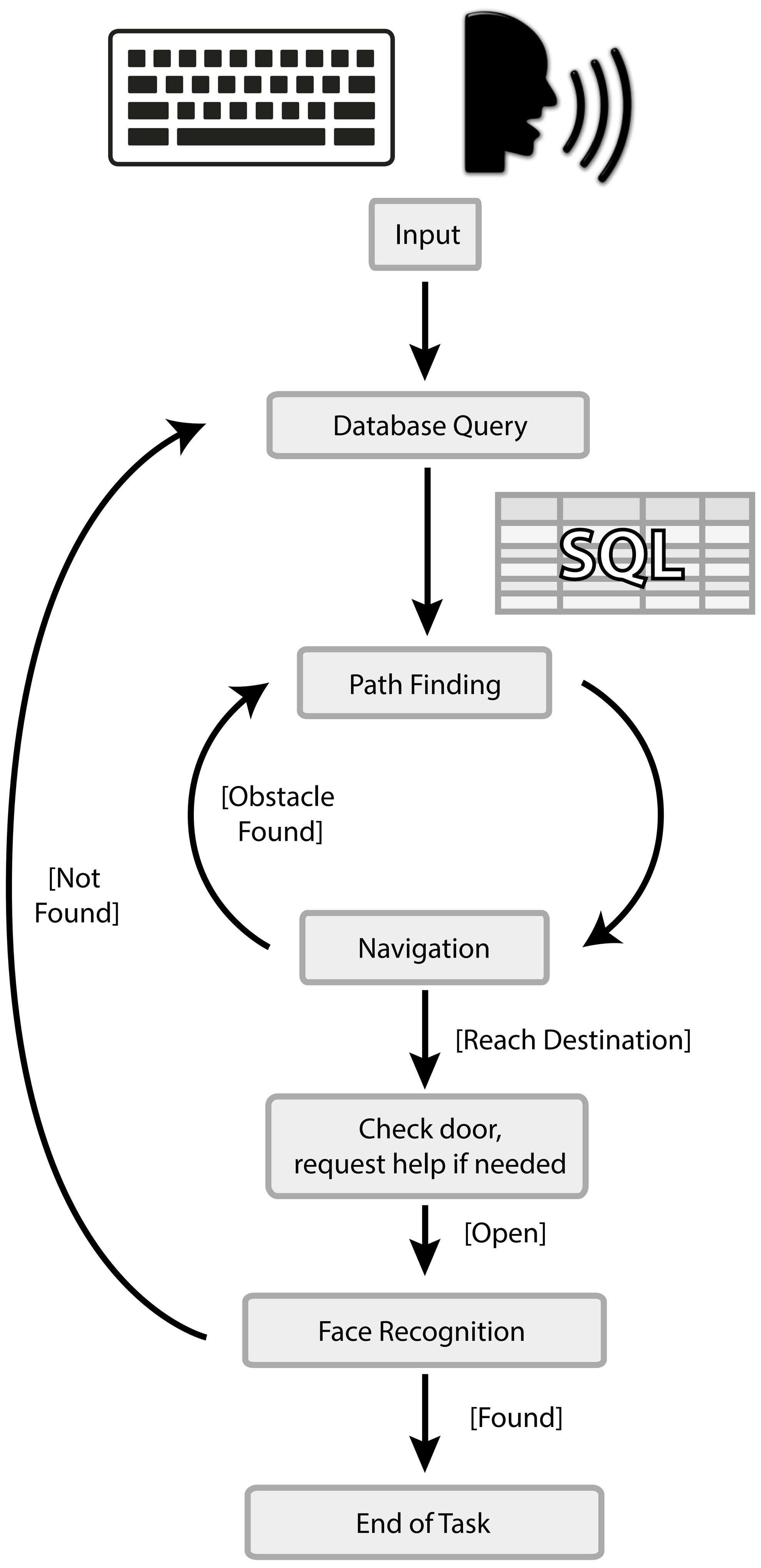

Figure 3. The steps involved in a task

The robot is meant to help visitors and new students find staff and rooms inside the school building. From the input, the command is extracted using regular expressions matching. Then, the robot guides the user through the most likely sequence of locations for the target person. The robot drives at a suitable pace and avoids dynamic obstacles. The robot can request help with doors and obstacles.

At destination, the target person is found using the described face recognition ensemble. If the target person is not found, the robot will search other possible rooms in descending order of probability. The location database is constantly updated to keep an accurate record of people's location.

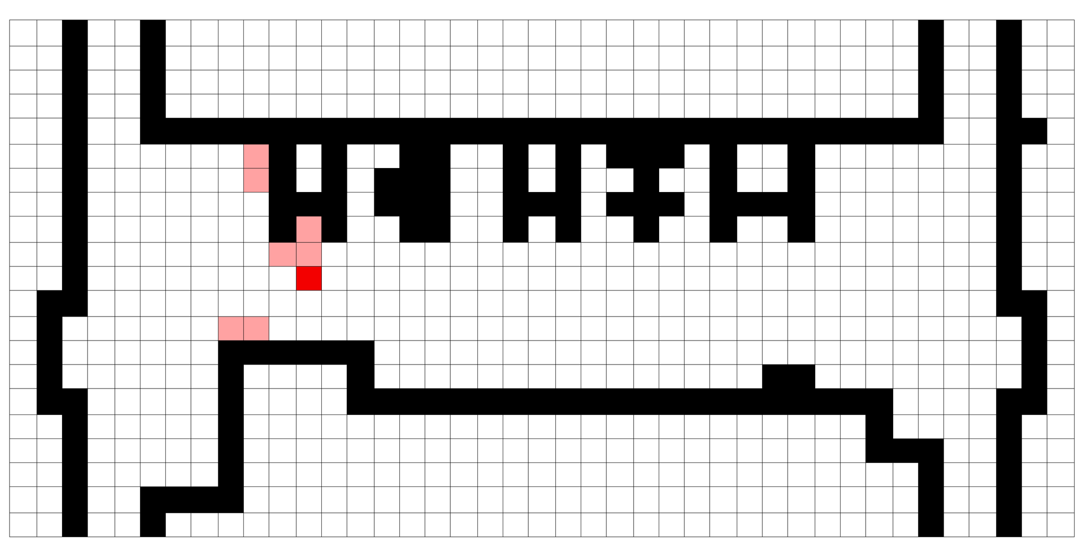

Figure 4. The occupancy grid used in navigation, visualised with colors

At initialisation, an overlay grid is generated from the floor plan, with cells marked as either a static obstacle or a free space. To account for noise in the map data, the intensity of a cell needs to exceed a certain threshold for it to be a identified as a static obstacle.

To navigate, an A* search is run on grid to find the shortest path to goal. To detect and avoid dynamic obstacles, each free cell is given an occupancy probability, which is constantly adjusted based on the laser scan. Cells exceeding 50% are marked as temporary occupied. The values decay overtime as dynamic obstacles move or disappear from the environment. If the current path is blocked by a dynamic obstacle, a new path will be computed and followed, allowing the robot to avoid unexpected obstacles such as bags and people.

Results

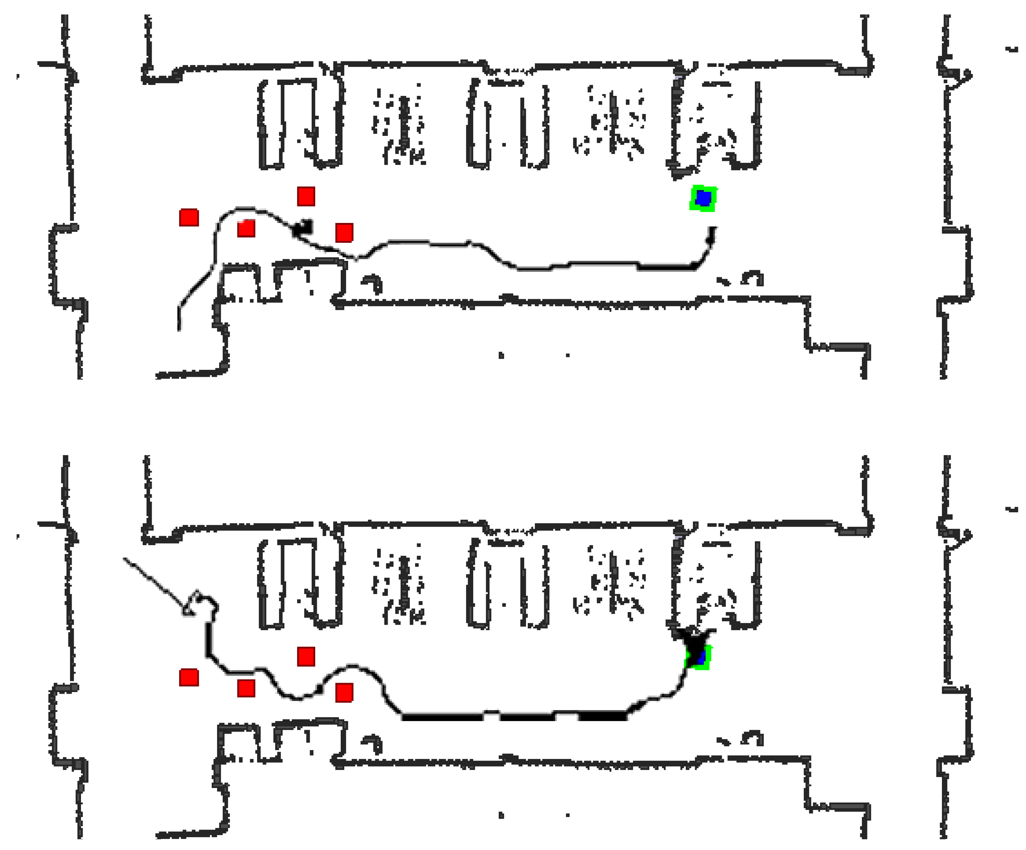

Figure 5. The paths generated by the robots in the presense of dynamic obstacles

Figure 5. above recorded the paths generated with dynamic obstacles. Using coordinates provided by the robot's odometry in simulation (Rviz) and python's turtle package, the travelled paths were plotted to visualise the performance of the path planning module. The red boxes are new obstacles which were not in the original map. This shows how the robot can robustly avoid obstacles using very narrow pathways in between. The performance in real life is identical to that in the simulation, even with moving obstacles (e.g. humans).

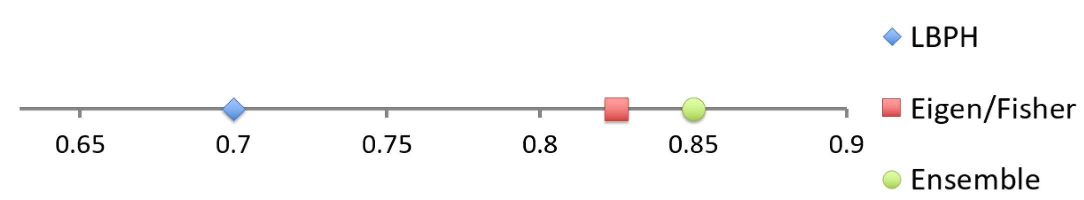

Figure 6. Accuracy of the ensemble compared to single recognisers

Figure 6 shows that the ensemble of three recognizers performs better than single ones, producing an accuracy of 0.85. Moreover, it has a False Positive Rate of 0 which is extremely desirable in our system since with a constant video feed, it will never incorrectly recognize the target person.

Conclusion

In conclusion, although it is still work in progress, the project has successfully combined various techniques, including both standard and novel methods. In future work, components will be improved in various ways: a denser grid will be used to smoothen the robot's movement and detect obstacles more precisely, while the face recognizers will be optimized further to give better results. Moreover, the tracking of people will be fully autonomous. Once refined, it can be widely deployed, especially in an office environment such as a university building.

Reference

1. Thrun, S.: Robotic mapping: a survey. In: Lakemeyer, G., Nebel, B. (eds.) Exploring Artificial Intelligence in the New Millennium, pp. 1–35. Morgan Kaufmann Publishers Inc., San Francisco (2003)

2. Jaiswal, S., Bhadauria, S., Jadon, R.: Comparison between face recognition algorithms - eigenfaces, fisherfaces, and elastic bunch graph mapping. JGRCS 2(7), 187–193 (2011)

3. Ahonen, T., Hadid, A., Pietikainen, M.: Face description with local binary patterns: application to face recognition. IEEE Trans. Pattern Anal. Mach. Intell. 28(12), 2037–2041 (2006). doi:10.1109/TPAMI.2006.244

4. Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, A.: Face recognition: a literature survey. ACM Comput. Surv. 35(4), 399–458 (2003). doi:10.1145/954339.954342

5. Database of Faces. AT&T Laboratories, Cambridge. http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html (2002). Accessed 11 Jan 2016

Acknowledgement

Undertaken as a part of Intelligent Robotics (taught by Prof. Jeremy Wyatt), this project has been funded by STRANDS and the School of Computer Science, University of Birmingham. We would like to express our appreciation to the professor and the school for their support.