Project Report

Ian Goldsmith – ijgo605

Making a robot follow people in an office environment.

My project aims to identify people within the surroundings of a robot and follow them. This was done primarily using a laser rangefinder to identify shapes that look like a persons legs. Two factors can be used to improve accuracy – looking for shapes that look like legs, and secondly looking for movement within the environment which may well indicate the presence of a person

Literature Summary:

Initially I set about researching what work had been done in the field – I didn't wish to reinvent the wheel and in retrospect it should've been quite intuitive to expect many people had done similar things before. Bellotto[1] used a very simple approach of looking within the scan data for two local minimum points, expecting this to be legs standing out from the background. This wouldn't be expected to be a very accurate approach and so in that paper accuracy was augmented by also using face recognition on likely points to identify people.

I soon found that very similar work to what I planned to do has been done before; some of it even using Player (Xavier[2]), the platform for robot device abstraction that we used in the course. That work was slightly different in that its scope involved attempting to do Simultaneous Localization And Mapping(SLAM) with the aim of being able to navigate more accurately when it can identify anomalies like people and remove them from recorded mapping.

In order to identify legs using laser scan data there were two clear categories of approaches, identifying legs by their geometry, and statistical approaches based on identifying movement within the environment and checking whether the magnitudes are correct for it to be a person, along with whether a person has previously been in that area.

Xavier[2] found a high degree of success by using Internal Angle Variance (IAV), which was a system to identify whether point readings are likely to be part of an arc that's likely to be a leg, and identify them as such. This is clearly a geometric approach to trying to find people. It's a very efficient way of doing things, but though the algorithm is described, code is not made available so I had to do it myself.

Though IAV may be reasonably good at identifying legs it will also identify other similarly shaped objects, so I am to augment that based on what statistical information is available. It's unlikely there will be many one legged people in the Robotics lab so it can be expected that where one leg is found there should be two, though at times one will be obscured by the other.

Similarly it's unlikely anyone can teleport in or out of the lab, it can be expected that people can only move a finite amount between laser scans, so based on whether things that appeared to be people were previously in the same area the accuracy in predicting whether an anomaly is a person can be improved. Statistical approaches like this I found seemed to be surprisingly robust in identifying people; Mendes[3] even found that people passing within very close proximity of each other could still accurately be individually identified.

---

[1] Nicola Bellotto and Huosheng Hu, *"Multisensor Integration for Human-Robot Interaction"*, http://privatewww.essex.ac.uk/~nbello/doc/Bellotto2005.pdf

[2] João Xavier, Marco Pacheco, Daniel Castro and António Ruano, *"Fast line, arc/circle and leg detection from laser scan data in a Player driver"* http://w3.ualg.pt/~dcastro/a1738.pdf

[3] A. Mendes, L. Conde and U. Nunes, "*Multi-target detection and tracking with a laserscanner*," Proc. IEEE Intelligent Vehicles Symposium, Parma, June 2004.

Methods used:

Based on what I found in previous research it seemed most prudent to try and combine a number of approaches in order to improve the accuracy of what I was doing. I would rely on the laser rangefinder due to its excellent accuracy, range and resolution.

The laser scan data is recorded into an array. This is analyzed for areas within acceptable ranges to be likely to be a leg that are distanced away from their surroundings. The scan data is also compared with a previous scan that's been transformed by the changes in odometry, in order to try and find movement.

Because odometry isn't very accurate objects at greater distances that appear to have moved are almost certainly errors and can't be relied on. Without the robot having better localization than odometry this is to be expected but the approach is excellent at spotting movement when the robot isn't moving as error is reduced.

The regions these two approaches find to be possible legs then get run through a method to check their Internal Angle Variance. If the region falls within acceptable ranges it is believed to be a leg and stored.

Legs found during each scan are compared with locations legs were previously found in order to improve the likelihood that targets are correct. Initially the strategy used was to always follow the closest leg but this was found to be a short-sighted approach; as soon as a target walked too far away the robot was likely to stop following the target and go for something closer.

In order to make the robot follow people more successfully the code was changed to give scores to all the known legs based on how long they were known about and how frequently they got identified as likely legs. Based on this the accuracy should improve and a person should be able to get the robot to follow them simply by standing in front of the robot long enough for it to change focus to the person, as it's current targets score would reduce whilst it can't be seen.

Implementation:

Having gotten an idea of the problems involved I set about starting my project, coding a client in C for 'Player'[4] that would identify people. By starting in small steps the aim was to keep progress reasonably steady and not do a great deal of work only to find something wrong with the approach afterwards.

Spotting movement was decided to be an easy step to start on. Initially I would work with the robot stationary to limit likely problems. By recording previous data in to an array and comparing it with the current data I was able to spot movement very easily in the room, but sometimes in regions that weren't a moving person. I believe these were places where the laser would sometimes hit an object and sometimes just miss it, instead shooting past it and hitting a distant object. In order to counter these anomaly readings I made my code ignore single points that feature movement.

Once the client skipped single instances of ranges changing it worked extremely well, only identifying people that were moving in the room. As was to be expected it sometimes didn't detect people that weren't moving, but it was surprising what little movement by people would be recognized.

Whilst these results were good they were of limited usefulness in my project. I needed to implement the rest of the code. The approach I decided on for looking for finding targets and following them works thusly:

- Look for legs by spotting movement or by being distanced from what's around them and of a roughly appropriate size.
- Once a possible leg region is established assess its shape is feasibly that of a leg using the IAV algorithm.

This successfully resulted in the Pioneer being able to follow me when there weren't possible confusing objects around, but there was its weakness; it often easily got confused, and problems were rapidly becoming very difficult to understand the nature of without being able to see what it was that the robot was stumbling on.

In order to better understand what was happening I used the game programming library Allegro to with relative ease draw out what it was the robot was seeing and classifying as a leg, by having it label in different colors what it was classifying as what. In retrospect I should've taken that approach from the beginning, it made things vastly easier for me. I also had it draw circles for distances it tried to keep objects away at, and a line to show which target it was attempting to follow.

Two different colored circles can be seen in Figure B. The smaller red circle is the 'safety distance' – when objects come within that distance the robot tries to steer or reverse away from them. The larger blue circle is the distance it tries to keep its target at.

It was at this point that I introduced a variable for each target which for want of a better name I called 'faith'. Each time a robot identifies an object as a leg it increases its faith. If it doesn't identify it as a leg it loses faith in it. The robot follows the object with the highest faith, thus meaning it should also keep following a target it already was.

When an object moves as long as it's within a specified distance of its previous distance it's regarded as the same object.

[4] "The Player Project", http://playerstage.sourceforge.net/

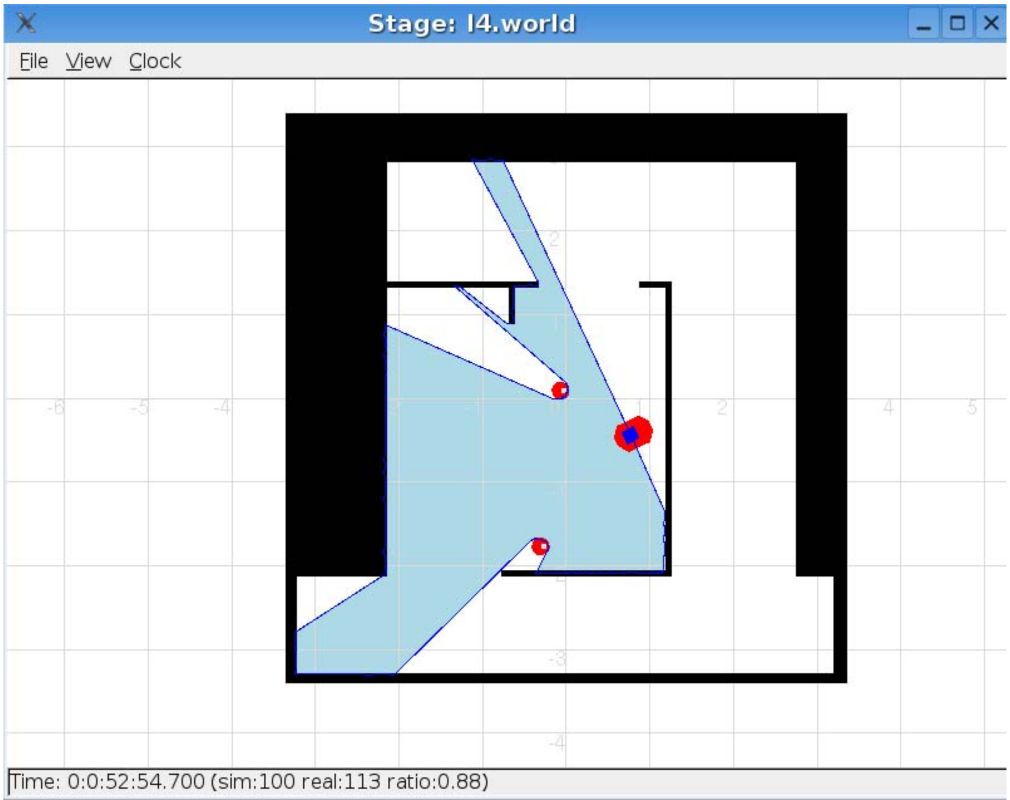
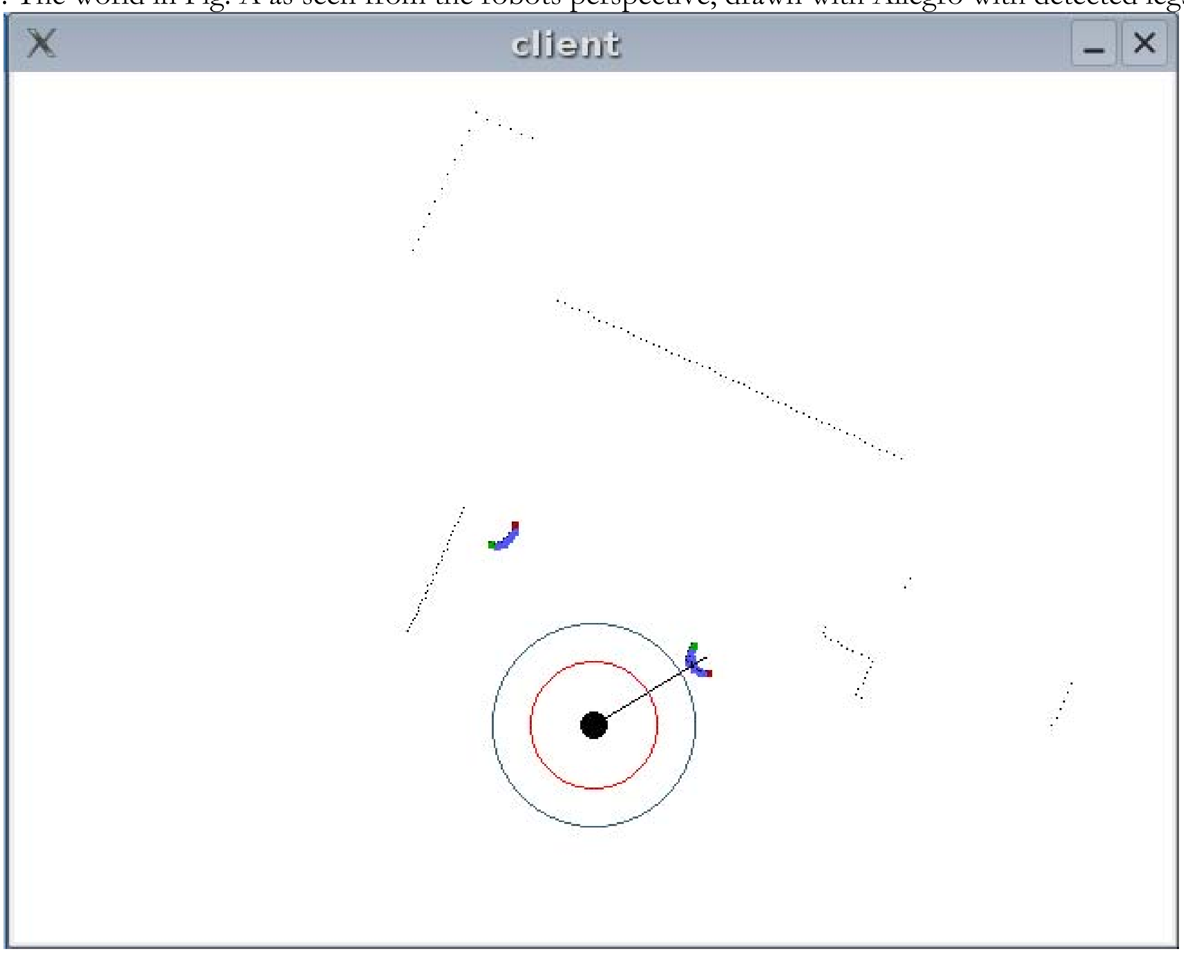


Fig. A: The world simulated in stage

Fig. B: The world in Fig. A as seen from the robots perspective, drawn with Allegro with detected legs.



Evaluation

The robot's success was mixed at following people in the real world. With variables set to be quite lax on what's regarded as a leg it would follow people quite well until other objects with similar features (like the thick bases of some chairs) were detected as legs.

Provided an actual persons legs were the most attractive target in the vicinity however it would follow the person, including being able to follow a person through a small maze, and though it wasn't programmed to explore the maze after losing a target around a corner it would stay stopped until the target came back and which point it would continue following them and could be led through the maze.

Quantitative analysis of success I initially didn't think was feasible because it would require a person to identify what was in fact a leg and what wasn't. I later decided that this was certainly not true, I should've just written code to simulate legs moving within entirely simulated laser range data. I didn't find if arbitrary readings can be fed to a laser device in player/stage though so quite a lot of code might need to be written extending player/stage itself.

Identifying legs whilst the robot wasn't moving could be done with a high degree of accuracy, however when the robot was moving this was very difficult; when everything is moving relative to the robot find people moving relative to the world is possible, but relies on you knowing where everything in the observed world is meant to be. Trying to transform previous laser scan data by the movement of the robot was difficult but with a lot more work might be possible.

Conclusion

As is found in most problems, things aren't as easy as they first appear. Finding legs in laser range finder data is quite possible, but eliminating false-positives can be tedious. In this case the robot was in an environment with many non-moving objects that look like legs.

The obvious solution to this problem is to be able to deduce which found anomalies are real legs by whether they've moved at some observed time. This could be done easily and with very high accuracy when a robot is stationary, but when the robot a robot is moving it becomes a very difficult problem. Objects that are stationary but small enough to be missed in some scans will appear to be moving. A recorded map of the area with good localization may mitigate this; previous laser scan data can be transformed by the recorded movement of the robot since the scan, but this is very difficult and had very limited success in this case.

The project proved a robot can successfully be made to follow a person in an office environment, using only the information learned from a laser rangefinder. The accuracy and success of this left something to be desired in this instance, but it was made quite clear which directions further work could go to achieve these improvements.

Possible future work

I'm continuing to work on the project because I'm very interested in seeing if the robot is able to follow people safely and accurately at its full speed. Though the project aim was fulfilled of having the robot follow people there are many improvements that could've been made, some are a small but some could amount to a large body of work.

Small features that probably should've been put in from the start include having the robot learn the shape of legs it's following, so that it learns to recognize the difference between a child and heavyset grownup's legs as it's following them and will stay with a target despite crossing paths by knowing the width it should recognize for instance.

Also, I intended to have it learn to build vectors for the magnitude and direction of movement people seem to have so that when people cross paths it can be estimated where the target would move to, as it could be expect to be unlikely that people crossing paths abruptly both change direction and go the opposite direction with the same speed. This might even raise other benefits; it may be able to differentiate a target from a slower person they're overtaking based on the fact that they move faster, or at least were moving faster up to the point of confusion.

References:

João Xavier, Marco Pacheco, Daniel Castro and António Ruano,
"Fast line, arc/circle and leg detection from laser scan data in a Player driver"
http://w3.ualg.pt/~dcastro/a1738.pdf

A. H. A. Fod and M. J. Mataric, *"A laser-based people tracker,"* in
Proc. IEEE Int. Conf. on Robotics and Automation, (Washington D.C),
PP. 3024-3029, May 2002.

A. Mendes, L. Conde and U. Nunes, "*Multi-target detection and tracking with a laserscanner,*"
Proc. IEEE Intelligent Vehicles Symposium, Parma, June 2004.

Nicola Bellotto and Huosheng Hu*, "Multisensor Integration for Human-Robot Interaction",*
http://privatewww.essex.ac.uk/~nbello/doc/Bellotto2005.pdf

M. A. Sen Zhang, Luhua Xie and T. Fan, *"Gemoetric feature extraction using 2d range scanners,"* in
International Conference on Control & Automation, (Montreal, Canada), 2003.

S. Press and D. Austin, *"Approaches to pole detection using Ranged Laser Data",*
http://www.araa.asn.au/acra/acra2004/papers/press.pdf

J. Bobruk and D. Austin, *"Laser Motion Detection and Hypothesis Tracking from a Mobile Platform",*
http://www.araa.asn.au/acra/acra2004/papers/bobruk.pdf

Lu and Milios, *"Robot pose estimation in unknown environments by matching 2d range scans",* in
CVPR94 (Computer Vision and Pattern Recognition 1994), pages 935-938.

R. Peterson, "*On Mobile Robot Navigation using Range Discontinuities in Laser Scans",*
http://epubl.ltu.se/1402-1617/2005/216/LTU-EX-05216-SE.pdf

A. Morate, "People detecting and tracking using laser and vision",
http://www.ee.kth.se/php/modules/publications/reports/2005/IR-RT-EX-0512.pdf

H. Gonzalez-Banos, C. Lee and J. Latombe, *"Real-time Combinatorial Tracking of a Target Moving Unpredictably Among obstacles"*
http://ai.stanford.edu/~latombe/cs326/2004/class16/tracking.pdf