IDENTIFICATION OF LIGHT SOURCES FOR LANDMARK-BASED ROBOT LOCALIZATION

Carlos García-Saura

Grupo de Neurocomputación Biológica (GNB), Universidad Autónoma de Madrid (SPAIN)
carlos.garciasaura@estudiante.uam.es

Abstract

In this paper we explore the possibilities of using a low-cost light sensor array placed in a robot as a method to identify light-based landmarks in a controlled environment. We present the GNBot, the open-source 3D-printable robot used, altogether with the light sensor array developed for the project.

The scenarios, data generation and techniques used for data processing are explained. We then show the progress in the application of different artificial intelligence algorithms (such as K-means and polynomial regression) to solve the light source identification problem.

Finally, a model for the light sources is proposed as well as the next steps towards robot localization.

1 INTRODUCTION

One of the projects at our research group at *Universidad Autónoma de Madrid* is the development of a low-cost robot swarm where each of the robots will be equipped with an artificial nose. The goal is to evaluate various algorithms for odor localization. These algorithms are going to require some way to estimate the location of each robot, and this project shows the very first steps towards this goal.

Robot localization is a widely-studied field, and most of the approaches for indoor environments are webcam-based. Some decide to place cameras in the robots [1,2,3] while others have managed to lower the cost by placing the cameras in the ceiling instead [4,5,6].

We particularly like the second approach, but it has a big downside: Since we place a webcam outside the robot, we have a centralized “master” that tells each robot its position. The algorithms we will be testing are bio-inspired so it makes more sense to have the robots sensing its environment rather than having an external centralized element sensing each robot and telling them their position.

Vision is great for spatial localization in the natural environment. But as cameras are complex, we want to experiment with the possibilities of more simple sensors.

We have found [7] to be a very inspiring work in this sense: an array of IR sensors has been used to locate moving reflective cylinders. But there are some key differences with our project: For instance, instead of having a distributed sensor array, we decided to place a small light sensor array in each robot.

This way our approach is to have various stationary light sources around the testing environment, and use artificial intelligence algorithms to use them as landmarks (as shown in the remarkable work by Sebastian Thrun [8]) to infere the relative position of each robot.

2 METHODOLOGY

Given that we are using such a simple sensor, it is key to generate the adequate data. And generating this data requires a well-described goal.

In first place we are going to describe the scenario and its constrains.

2.1 Experiment description and goal

To simplify these initial experiments, we decided to restrict the motion of the robot to a rotation in the XY plane. This decision was made because the robots for the swarm will be operating in a 2-D environment. By restricting the motion to a rotation, we can evaluate the performance more easily.

N light sources are placed around the robot's operating environment, and the robot will then make M turns spining at a constant speed while logging the following data:

- Intensity of light measured by each sensor of the array

- Angle of rotation of the robot

At the end, our goal is to determine the direction (angle) of each of the light sources.

2.2 Robot description

For this project it was needed to design and 3D-print the chassis and wheels of the GNBot.

The robot uses an Arduino Mega board and a custom shield (GNBoard) that integrates all the components and connectors.

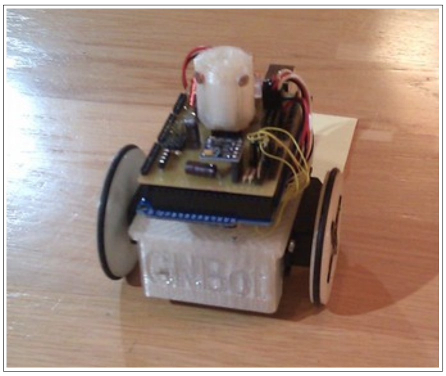

Fig 2.1 – Assembled GNbot

The key features used for this project are:

- *Servo-motor connectors:* Continuous rotation servo-motors provide the main motion.
- *Electronic compass:* An I^2C compass module provides accurate rotation measurement.
- *Bluetooth link:* A serial bluetooth module was used to interface with the robot remotely. This was done to avoid any possible light interference caused by tether wires.
- *Light sensor array:* The light sensor array is composed of four light dependent resistors arranged in a ring shape (Fig 2.2.1). These are held in position using a 3D-printed shape designed for the purpose. The measurement is an analog voltage value [0,1023] proportional to the resistance.

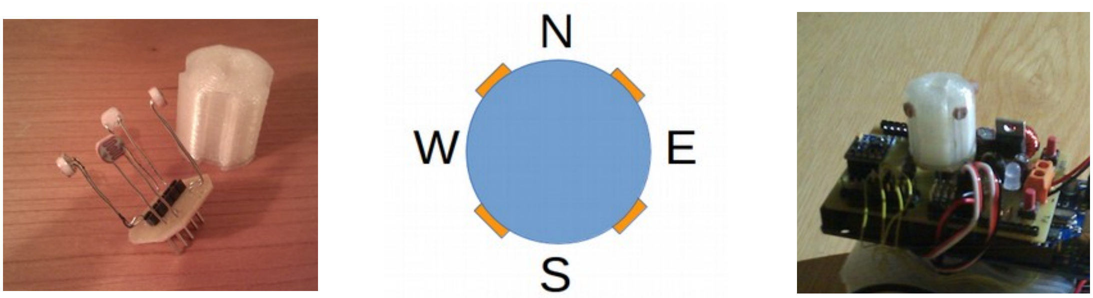

Fig 2.2 – Light sensor array (left), Diagram (middle), Actual sensor mounted in the robot (right)

2.2.1 Firmware and software

For this project it was also necessary to create the firmware for the robot's electronics, as well as the host software to communicate and process the data from a computer.

The firmware that goes in the electronics has been developed using the Arduino IDE[1]. It interfaces with the bluetooth module via serial, with the electronic compass via I^2C, controls the servo-motors using PWM, and reads the analog value from each light sensor. It is basically a layer that gives full remote control over the robot from the host software.

[1] *http://www.arduino.cc/*

The host software has been written in Python[2]. A python module was created to abstract the communication with the robot, and PyBluez[3] was used to manage the bluetooth connection.

2.3 Data generation

After some tests, we decided to do the following experiments. Each test was repeated both with ambient light and in the dark, with M = 3 full rotations for each experiment.

1. No light focuses
2. One light focus (bright)
3. Two light focuses (both bright) – shown in Fig 2.3
4. Three light focuses (two bright, one dim)
5. Four light focuses (two bright, two dim)

A light focus was added at a time, and all the parameters of the experiment were carefully recorded with the python script:

- Angle/distance/intensity of each light
- Ambient light intensity
- Textual description of the scenario
- Number of turns

These experiments resulted in 10 data files saved using the Pickle module from Python. Some of the experiments were recorded in video.[4]

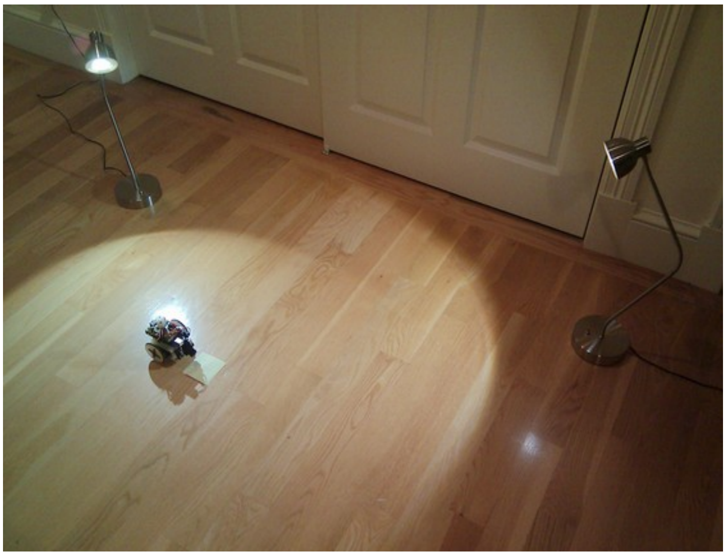

Fig 2.3 – Experiment with two light focuses and no ambient light

2.4 Data visualization

We used Matplotlib[5] to visualize the data. Fig 2.4 shows the raw data obtained from the experiments.

The data did not have any significative noise, so no pre-filtering was required.

[2] *http://www.python.org/*

[3] *https://code.google.com/p/pybluez/*

[4] Videos: *http://www.youtube.com/watch?v=Pbjn0CGMeAM* and *http://www.youtube.com/watch?v=Rh3S7srPF7w*

[5] *http://matplotlib.org/*

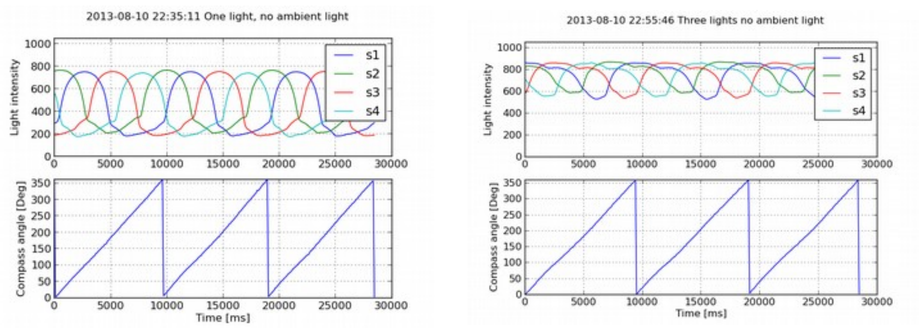

Fig 2.4 – Raw data from two of the experiments

As the intensity values are recording toguether with an orientation measure (in degrees), the datapoints are periodic and require special polar treatment.

We switched to the polar representation shown in Fig 2.5, and time measure was discarded as the sistem of the experiment is time invariant.

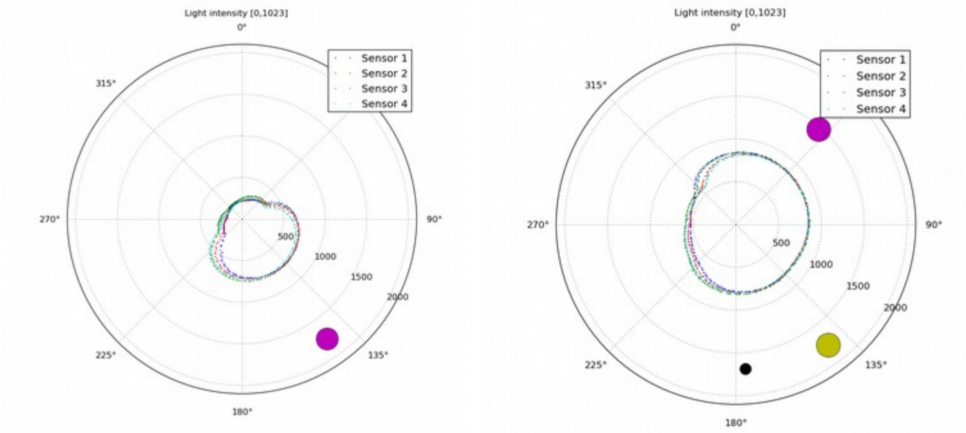

Fig 2.5 – Polar representation of the correspondent data in Fig 2.4. Light sources shown as circles.

All the graphs were examined closely and we evaluated all the possible algorithms that could be used to solve the problem.

3 RESULTS

Our initial idea was to use K-Means to “cluster” the light intensities and thus find the position of the light sources. But since the first visualization we realized this was not a possibility since the datapoints don't form clusters at all.

But it could be very possible to use K-Means by extracting different features from this data.

3.1 Data processing using polynomial regression

The second approach was to use some kind of regression to obtain a curve from the datapoints, and then use it to determine the maximum and thus the light position.

We used existing polynomial regression functions from Numpy, but found that these were not prepared to handle data with periodicity so we got a discontinuity (Fig 3.1). The polynomial was chosen with a degree of 6, to allow it to fit the curves accurately (in this case overfitting is not that much of an issue).

After calculating the polynomial coefficients, the derivate was computed and finally its roots. This way we figured the maximums and minimums of the data point cloud. Our belief was that this method would perform well but only with single light sources.

The error measure used is the sum of the absolute difference in angle of each light source detected, and we did specify the number of light sources N to the algorithm to return the N maximums.

The mean error for five test using a single light was of 5.32^{o} (12.54^{o} with ambient light), while for more than one light the minimum error was of 112.67^{o} (133.85^{o} with ambient light).

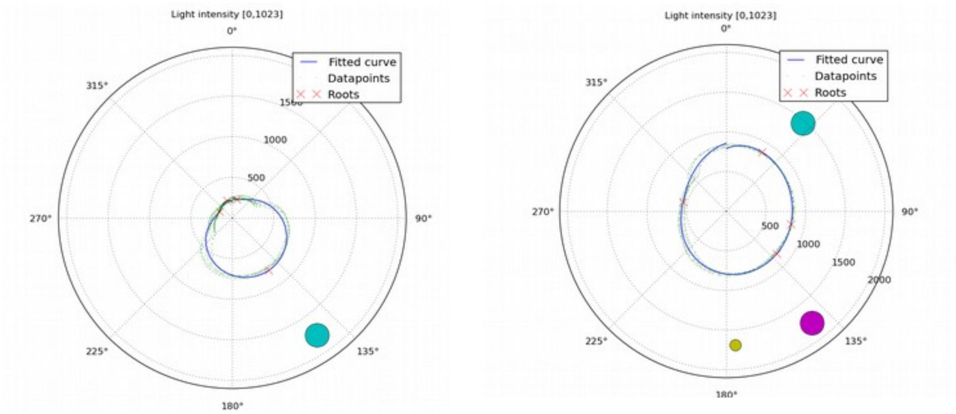

Fig 3.1 – Polynomial regression results for the data in Fig 2.4 and Fig 2.5

In the end, polynomial regression performed surprisingly well with single light sources, but doesn't scale to multiple light sources.

3.2 Modeling the light sources

We decided to try to model each light source as a combination of both ambient light and directional light following the -simplified- equation: ***intensity(phi,dist) = ambient + cos(phi)*directional/(dist^2)***

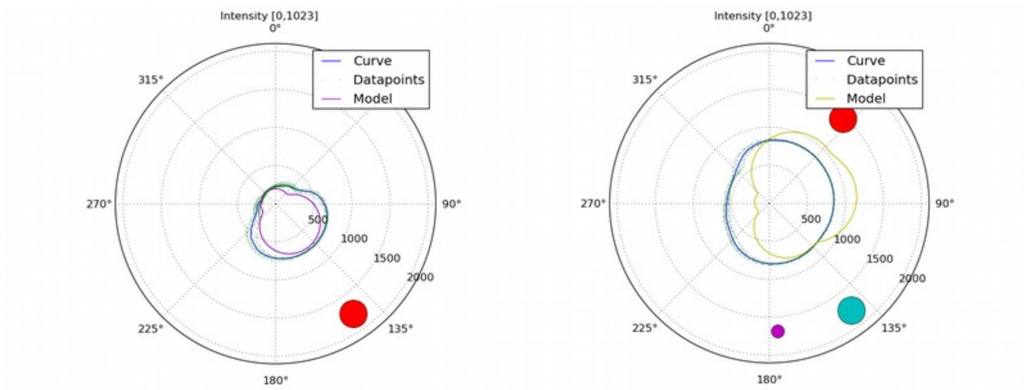

Fig 3.2 – Polynomial regression results for the data in Fig 2.4 and Fig 2.5

As shown in Fig 3.2a, the model is good enough for one light source but fails with more due to the non-linear nature of the system. We missed having made more light measurements for a single light at different distances, and of each light in the experiments with the rest turned off.

4 CONCLUSION AND FUTURE WORK

A new robotic platform, the GNBot, has been developed and published open-source.

For future work, we will try to improve our light source model to use it in a Mixture Model applied to polar data. Once we have enough data measurements, we will also try to apply feature learning to come up with a better light model.

A Hidden Markov Model together with a particle filter will be used to estimate the robot localization using a single light source and the intensity and direction of the light determined with the polynomial regression technique developed for this project. This will later be extended to multiple light sources.

Updates on this research, as well as all the source files, data and visualizations generated can be found in **https://github.com/carlosgs/GNBot** under the ***Attribution - Share Alike - Creative Commons License (http://creativecommons.org/licenses/by-sa/3.0/)***

5 ACKNOWLEDGEMENTS

I would like to express my deep gratitude to Professor Chris Piech for his patient guidance, enthusiastic encouragement and useful critiques of this research work.

REFERENCES

[1] Hongbo Wang; Hongnian Yu; Lingfu Kong, "Ceiling Light Landmarks Based Localization and Motion Control for a Mobile Robot," Networking, Sensing and Control, 2007 IEEE International Conference on , vol., no., pp.285,290, 15-17 April 2007

[2] C.M. Gifford, "Low-Cost Mobile Robot Localization Using Only a Downward-Facing Webcam," Tech. Report, Dec. 2009.

[3] A. De La Escalera; L. Moreno; M. A. Salichs; J. M. Armingol, "Continuous mobile robot localization by using structured light and a geometric map", International Journal of Systems Science, Vol. 27, Iss. 8, 1996

[4] REINA, Andreagiovanni, and Marco DORIGO. "A network of ceiling cameras performs distributed path planning to guide a ground robot."

[5] Haoyao Chen; Dong Sun; Jie Yang; Jian Chen, "Localization for Multirobot Formations in Indoor Environment," Mechatronics, IEEE/ASME Transactions on , vol.15, no.4, pp.561,574, Aug. 2010

[6] Haoyao Chen, "Towards multi-robot formations : study on vision-based localization system", Thesis (Ph.D.), City University of Hong Kong, 2009

[7] Pavlov, V.; Ruser, H.; Horn, M., "Feature extraction from an infrared sensor array for localization and surface recognition of moving cylindrical objects", Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007. IEEE , vol., no., pp.1,6, 1-3 May 2007

[8] Sebastian Thrun. 1998. Bayesian Landmark Learning for Mobile Robot Localization. Mach. Learn. 33, 1 (October 1998), 41-76