Trilateration From Sensor Readings

Christian McArthur

CS 5306 – Adv. Operating Systems

Abstract

It can be important to be able to determine the location of an object or an event based upon its physical relation to known points in space. In wireless sensor networks this could be needed to determine where a sensor exists among a network of other sensors with known, fixed locations. It could also be used to determine the location of a fire in a forest or an animal's movement. The ability to determine this location is solved through a process called trilateration. For this term project, I will use trilateration to be able to determine where a light is shining on a grid of sensors with fixed positions. This paper describes the steps involved in completing this project as well as some of the results collected while implementing the steps.

Introduction

The end goal of the term project is to create an application that receives communications from a grid of wireless sensors such that the path of a light can be tracked. This requires the completion of a number of parts including: accurately estimating the distance of a light from a single sensor based upon a light reading, using distances from multiple sensors to discover the light's location through trilateration, and displaying this location on a computer screen. This report cover the following areas: related work in the areas of localization and trilateration for wireless sensor networks, the hardware used to complete the project, results on the relationship between light intensity readings and distance, how to calculate the light's distance based upon that light intensity reading, how to perform trilateration, and the final integration of all the parts into a working project.

Related Work

This project is closely related to localization issues with wireless sensor networks (WSN). Assume we have a network of sensors deployed in some environment. If a group of sensors detects some event how can you determine where the event occurred. For instance, WSNs are used to detect forest fires.[1, 2]. If a spike in heat associated with a fire is detected or a lightening strike is detected, where do you send firefighters? If an animal, tagged with a wireless transmitter or RFID chip[3], travels through a WSN that picks up the animal, how do you know the exact location of the animal and the path it takes? These questions are solved through localization.

How to perform this localization is an area of research. Trilateration is a simple method of determining the location of an object based upon its distance from three known points in space. [4] provides a multidimensional scaling algorithm for determining the location of sensors among a grid of sensors with known locations. [5] provides two localization algorithms that do not depend on knowing the signal strength or "angles of arrival" of communications from a sensor.

MICAz Mote & Sensor Board

For this project, MICAz MPR2400 modules (motes) were used along with MDA100 sensor boards from Crossbow Technology. The motes communicate with one another wirelessly on the 2.4GHz radio frequency. The sensor boards have sensors to report the intensity of light, temperature, and the voltage coming from the mote's battery.[6] The motes run programs created with nesC on an operating system called TinyOS. nesC is an event-driven, component based extension to the C programming language.[7]

Since the project's goal is to track a light, the photoresistor was activated on the sensor boards. The reading reported by the mote is the resistance rating of the photoresistor. A higher resistance reading means the light hitting the photoresistor is stronger. It is interesting to note that experimentation shows that the maximum reading provided by the photoresistor is 1000, leading me to assume a 1k resistor is used. The mote periodically broadcasts to the world a packet containing a set of readings. A second mote, connected to a desktop computer, receives these packets and transfers them to the computer. Applications running on the computer can receive these packets and process them to complete some task.

Sensor Reading vs Distance

The first task to completing the project is to discover the relationship between light intensity as seen by the sensor and the distance a light source is from the sensor. To accomplish this programs were written for the wireless sensor modules and the desktop computer. The motes with sensors simply recorded readings from the photoresistor and broadcasted them to the world. The receiving mote transmitted this information to the computer where an application was running to process the packets. The application for this phase of the project would collect readings over a period of time and output the average of these readings. This reading and reporting process by the application was repeated a number of times with a light source at varying distances from the sensing mote.

The experiment was performed in a mostly dark room. A single set of florescent light bulbs could not be turned off for security and safety reasons. The sensing mote was placed at the end of a grid with markings every five inches from it to a total of 40 inches away. A small flashlight would be turned on and held over the sensing mote at a height of approximately one foot. Once the software application on the desktop computer recorded the average reading for a period of ten seconds, the flashlight would be moved one inch away from the sensing mote. With the flashlight at this new distance, readings from the sensing mote were recorded, averaged and displayed. This process continued with the flashlight moving one inch away on each iteration until the flashlight was 40 inches away.

The overall experiment was performed six times. Two sets of results were discarded due to their extreme difference from the results of the other trials. The readings from the four valid trials were placed into an Excel spreadsheet and graphed. An average of the the readings were also plotted as well as a “best fitting” line to this average (a trendline to use Excel terminology). Illustration 1 shows this graph. The best fitting curve to the average of the readings resulted in a fifth-degree polynomial:

$$y = -0.000094x^5 + 0.010890x^4 - 0.477611x^3 + 9.802377x^2 - 93.204854x + 950.523720 \quad (1)$$

Given the equation (1) and a known light reading, a person can substitute 'y' for the light reading and solve for x, the distance the light is from the sensor.

Solving polynomials larger than three or four degrees is complex and is generally only done by computers. The Jenkins-Traub algorithm is commonly used to perform this operation and was implemented in

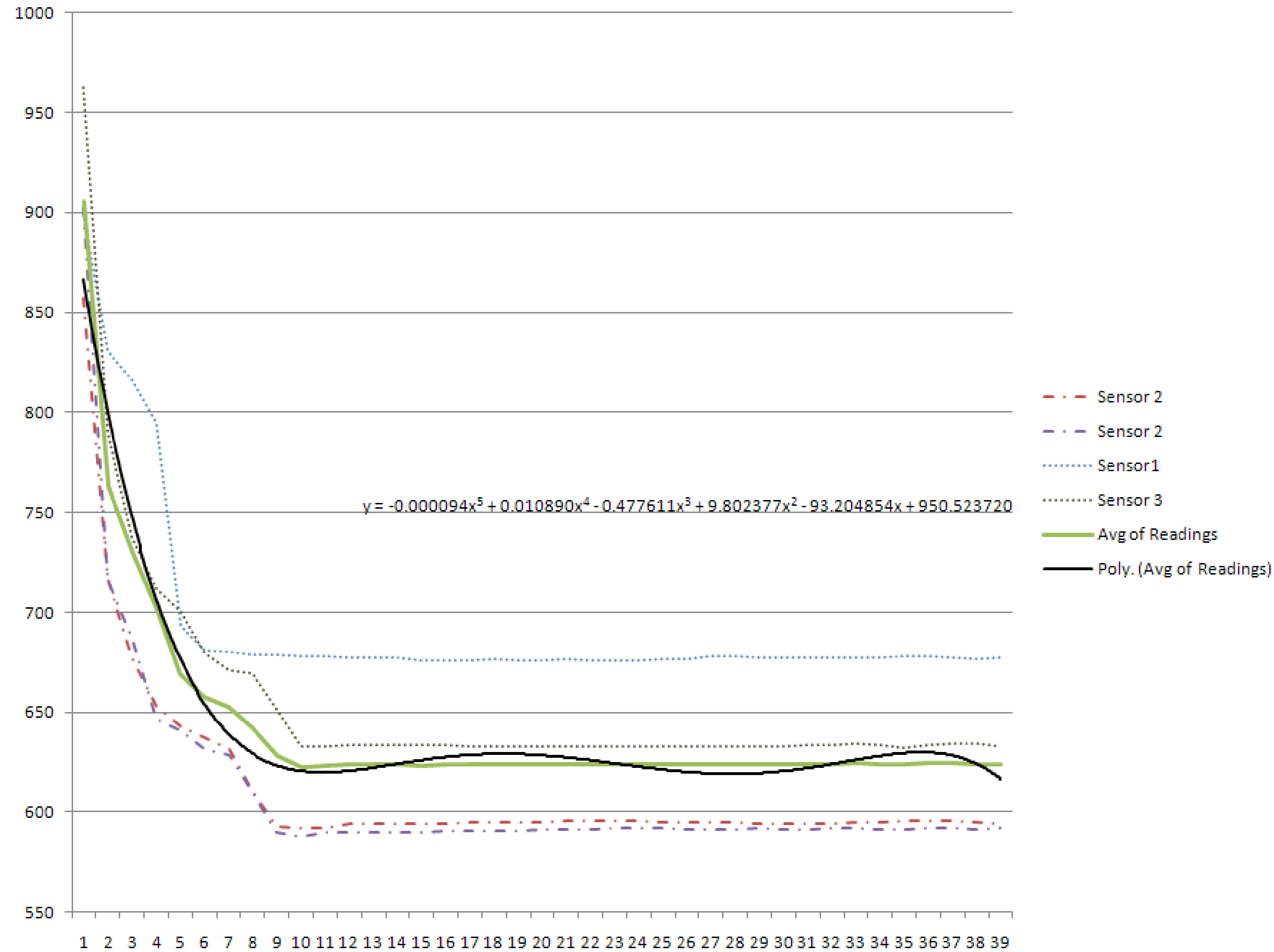

Illustration 1: Graph of light intensity readings vs distance

Fortran by Jenkins and Traub in 1972.[8] However, I was unable to find a Java implementation of the algorithm or be able to program the algorithm myself in Java.

As a result of the difficulties in calculating the distance of a light source based upon equation (1), a simpler method involving a table look-up was implemented. Using the data gathered in the first part of this experiment, the distance can be estimated on the light reading. For example, if the light reading is between 907 and 740, we will assume the light source is one inch from the sensor.

There some issues with the light intensity reading that need to be addressed. It was discovered that different sensors will return different light intensity readings for the same distance. As shown in Illustration 1, two lines for “sensor 2” show almost exact readings for each distance. The other two sensors shown in the graph provide different sets of readings. While the overall trend of the lines are similar, it is obvious that sensors provide differing readings. In addition, once this data has been collected and is intended to be used in future parts of the project, the same general lighting situation must be used. If there is more or less ambient light in future trials versus the first set of experiments, then results cannot be relied upon. In addition to the ambient light, the source of the light to be tracked must also be consistent with the data collection trials performed. A light source that is brighter and/or disperses more will result in different readings by the sensors. All of these factors must be taken into consideration in the final part of the project.

Determining Light Position via Trilateration

Trilateration is a method of determining the position of an object based upon its distance from three other objects with known positions. For the purpose of this project the light is the object we want to determine the position of. A grid of motes with specific locations are used to calculate the light's location. As the light shines down on the grid of motes, each mote will report some light intensity reading. A desktop computer will receive messages from the motes announcing the readings from their photoresistors. The three motes with the highest reading are chosen; it is assumed that the light is closest to these motes due to the higher readings and will lead to a more accurate determination of the light's location. The readings from these three motes are converted into distances as described in the previous section.

$$r_1^2 = x^2 + y^2 + z^2$$
$$r_2^2 = (x-d)^2 + y^2 + z^2$$
$$r_3^2 = (x-i)^2 + (y-j)^2 = z^2$$

$$x = \frac{r_1^2 - r_2^2 + d^2}{2\mathrm{d}}$$

$$y = \frac{r_1^2 - r_3^2 + (x-i)^2 + j^2}{2\mathrm{j}}$$

Illustration 2: Trilateration Equations

A series of calculations are performed based upon the distances of the light from the three motes and the known locations of the motes. The equations used to perform the calculations are shown in Illustration 2.[9] In the first three equations, r_1, r_2, and r_3 represent the distance the light is from each sensor. The sensor for r_1 is assumed to be at the origin of the grid for purposes of the calculations. In the second equation, 'd' is the distance from two motes that lie on the same y axis. In the third equation, 'i' and 'j' are the distance on the x and y axis, respectively, the third mote is from the first. Using these six pieces of information (r_1, r_2, r_3, d, i, and j) you can compute the x and y coordinates of the light using the equations in Illustration 2. The resulting coordinate pair is based upon an origin of sensor 1. By adding the (x,y) coordinate of sensor 1 on the actual grid to the calculated coordinates, you can determine where on the grid the light is shining.

A new application was written to run on the desktop computer. It would select the three motes with the highest reading and use that information to calculate the position of the light source as described. The application for this part would then display the calculated coordinates to the console. This was compared to where the light was actually shining. I found that while the resulting coordinates were close to the actual location of the light, it was not the exact location. I can try to explain this discrepancy based on two things.

First, as mentioned in the last section, not all sensors will provide the same reading for light at the same distance. This means since there are multiple sensors being used, three or four for this particular part, there are potentially different sets of readings being reported for a single distance. In addition, the light source used in this project, a small mag-lite flashlight did not provide a purely circular light onto the grid. The light at times would be an oval shape or have a kind of spike on one side. This means that two sensors on different sides of the light source could get very different readings even if they were the same distance from the light.

This differences in readings due to individual sensors and due to the light source used will results in discrepancies in the calculations of the light's location. However, even with these discrepancies, the calculated coordinates were still in the general area of the actual location.

"Follow The Light"

"Follow the Light" is the final part of the term project and the name of the final application. The wireless sensors run the same programs as in the previous parts. The desktop computer runs the application from the previous part but with an added piece. This piece is a graphical interface that is displayed to the user. This display shows the locations of the wireless sensors on the grid with a "X". The display also shows a mark, an "O", representing where on the grid the light is shining.

As the "Follow the Light" application runs the display is updated every half second. When an update occurs, as with the previous part, the three sensors with the highest readings will be used. These three readings will be used to calculate where on the grid the light is shining. This location is then updated on the graphical display.

As noted in the previous section, the light's location on the graphical interface is not always the exact location of the real light shining on the grid of motes. Often the location is in the general area and sometimes no where close to the real location. The reason for this is the same as the previous section where not all sensors report similar readings for a given distance and a light source that was non-circular.

Future Work

A calibration mechanism should be added to attempt to resolve the issue of differing readings from the motes. This could possibly be accomplished by taking a series of base readings in which the light source travels a specified path. The computer can examine the readings from each sensor as the light travels this pre-specified path. Knowing the distance the light is at a particular point in time, and seeing the readings from the mote over a period of time, the computer application can create a lookup table of sensor readings to distance for each individual mote. This would require the light source to travel the exact specified path in the exact specified time for this calibration to be successful. If the light does not perform as specified during this calibration then the "Follow The Light" application may have just as many accuracy issues as the current version does. This calibration technique would still not take into consideration the actual light source. For more accurate results, a light source that was more circular.

Conclusion

For this term project, an application was to be written to track the path of light source as it shines down on a grid of wireless sensors. This was accomplished by combining various experiments and applications. I determined what readings a sensor reports based on the light source's distance from the sensor. In the course of this experiment I discovered that sensors will report different readings for the same distance from a light source.

Using an average of readings from several sensors, application code was written to determine the distance a light is based upon the sensor reading. From there, multiple sensor readings were used to calculate the light's position via trilateration. Then a graphical interface was added to display on the computer screen the light's location in respect to the grid of motes.

As a result, a reasonable implementation of the required application was created. The “Follow The Light “ application does not perfect track a light source's path over a grid of motes. This lack of perfection is due to the non-similar readings provided by the motes used in the project as well as a light source that is not provide a completely circular dispersion of light.. However, the resulting application does come close to showing the true location of the light source most of the time.

References

[1] Liyang Yu; Neng Wang; Xiaoqiao Meng; , "Real-time forest fire detection with wireless sensor networks," *Wireless Communications, Networking and Mobile Computing, 2005. Proceedings. 2005 International Conference on* , vol.2, no., pp. 1214- 1217, 23-26 Sept. 2005

[2] Hefeeda, M.; Bagheri, M.; , "Wireless Sensor Networks for Early Detection of Forest Fires," *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on* , vol., no., pp.1-6, 8-11 Oct. 2007

[3] Pereira, D.P.; Dias, W.; Braga, M.; Barreto, R.; Figueiredo, C.M.S.; Brilhante, V.; , "Model to integration of RFID into Wireless Sensor Network for Tracking and Monitoring Animals," *Computational Science and Engineering, 2008. CSE '08. 11th IEEE International Conference on* , vol., no., pp.125-131, 16-18 July 2008

[4] Shang, Y., Ruml, W., Zhang, Y., and Fromherz, M. P. 2003. Localization from mere connectivity. In *Proceedings of the 4th ACM international Symposium on Mobile Ad Hoc Networking &Amp; Computing* (Annapolis, Maryland, USA, June 01 - 03, 2003). MobiHoc '03. ACM, New York, NY, 201-212.

[5] Rudafshani, M. and Datta, S. 2007. Localization in wireless sensor networks. In *Proceedings of the 6th international Conference on information Processing in Sensor Networks* (Cambridge, Massachusetts, USA, April 25 - 27, 2007). IPSN '07. ACM, New York, NY, 51-60.

[6] “Product Reference Guide,” Crossbow Technology, Inc. San Jose, CA. [Online]. Available: http://www.xbow.com/Support/Support_pd_files/Product_Feature_Reference_Chart.pdf. [Accessed: April 29, 2010].

[7] TinyOS. [Online] Available: http://www.tinyos.net. [Accessed: April 29, 2010].

[8] Jenkins, M. A. and Traub, J. F. 1972. Algorithm 419: zeros of a complex polynomial [C2]. Commun. ACM 15, 2 (Feb. 1972), 97-99.

[9] “Trilateration”, Wikipedia. [Online] Available: http://en.wikipedia.org/wiki/Trilateration. [Accessed: May 2, 2010].