Camera Tracking Positional System

Owen Bartolf, Terra Linda Innovation Lab

Table of Contents

Introduction

The Terra Linda Innovation Lab has continued to expand into the realm of robotics, and with great success. The first iteration of the robot model has already been physically built and tested with placeholder movement logic, and more advanced navigation logic is already in development. With autonomous programs becoming increasingly advanced, more infrastructure is required to aid in the positioning and navigational purposes. A lack of an accurate method to determine the positioning of vehicles is one of the primary limiting factors of the lab's robotics programming. Thus, this proposal to create such a positioning system has been submitted.

Background Research

Problems with Existing Solutions

Many large-scale solutions to the positioning problem have already been created. The two most common solutions to the problem of determining an accurate system come in the form of radio triangulation and GPS.

Triangulation has existed for hundreds of years (Laertius). The technique involves using trigonometry to derive the position of an unknown object when given only three distances to known points. In the modern world, radio transmissions are often used in triangulation. The main problem with this technique, however, is a lack of accuracy on small scales. There are very few methods to measure the distance that a radio wave has travelled. The most common way to determine this is to compare the signal strength to the inverse square law. The inverse square law states that the intensity, or strength, of a radio wave is inversely proportional to the distance (Nave). Though this can be calculated with ease, radio waves don't lose strength in a stable pattern on small scales. Thus, this method is not accurate enough for the scales in use for the project.

As GPS also uses triangulation via radio waves, it holds the same accuracy issues as the radio technique. GPS is not inaccurate by any means; the system can determine positional coordinates with an average deviation of 4.6 meters (GPS Accuracy). However, as the robots in the Innovation Lab are meant to navigate areas that do not exceed 100 square feet, this system is not useful in this design context.

Possible Applications of Technology

The possibilities of a system that can accurately derive positions of objects at small scales has limitless potential for use in robotics, analytics, and augmented reality. Some of the many applications have been listed below in bullet form.

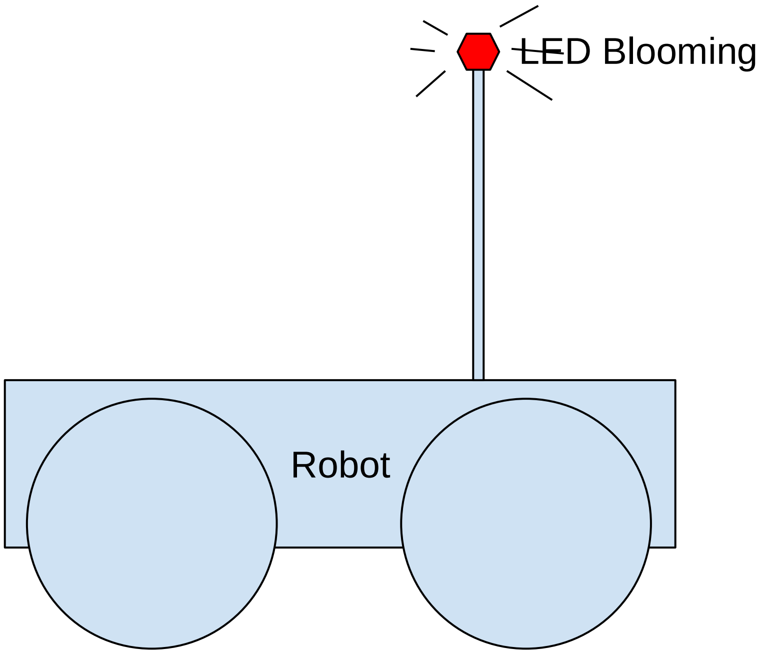
- Tracking foot traffic and building use to improve architectural design
- Tracking positions of objects for better integration with augmented reality systems like Microsoft HoloLens
- Investigating the interactions of organisms in small ecosystems, like creeks or rivers.
- Traffic research

Design of Proposed System

Term Definitions

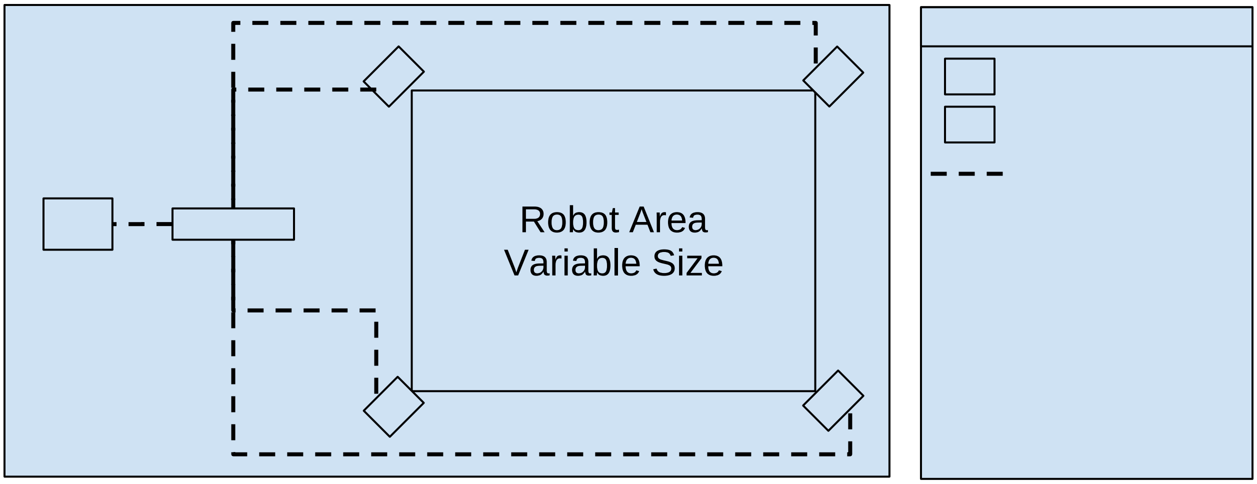
- **World Space**: Refers to the three-dimensional euclidean coordinate plane that is reflective of points in the real world. Y = 0 is the ground.
- **Screen Space**: Refers to the two-dimensional euclidean coordinate plane that is reflective of positions on the plane of a given camera's viewport.
- **LED Bloom**: Refers to the colored flare created by colored LEDs in a camera's viewport.

Robot LED Bloom



To begin, the robot must have a defining feature that can be detected easily by using a camera. Thus, each robot will have an LED blooming device attached to them. The LED blooming device is a small, elevated sphere of colored LEDs. When seen with a camera, these LEDs should leave a defining bloom that is easy to detect using even the simplest of color detection algorithms. The power consumption footprint of the LED blooming device is also extremely tiny and should not inhibit other functions from operating properly.

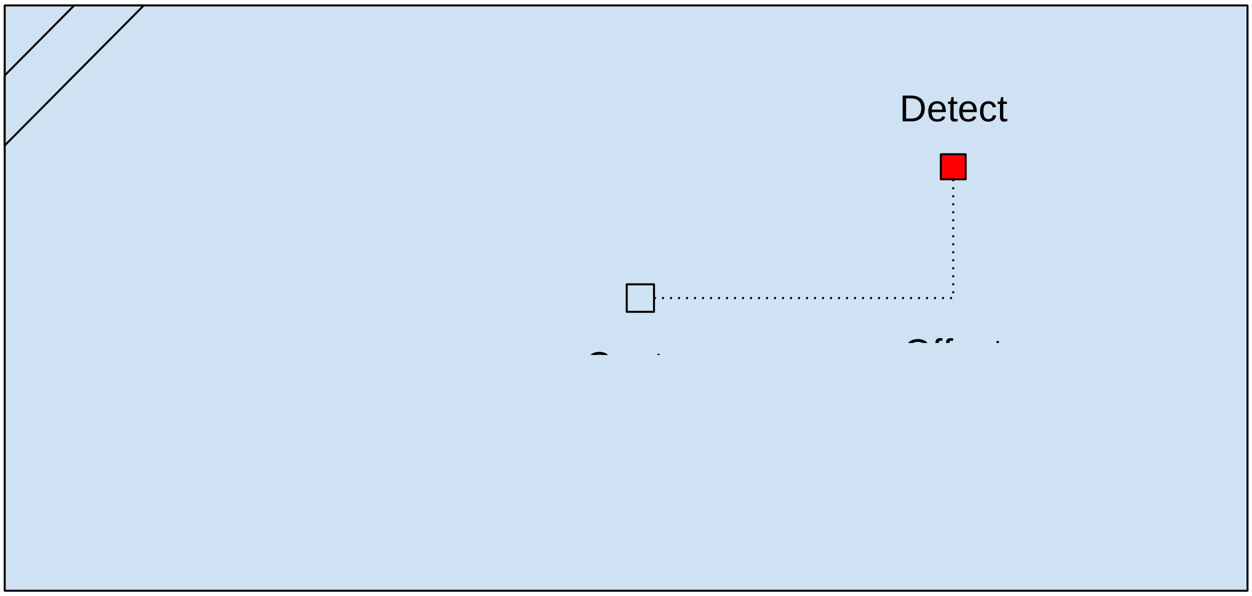
Hardware Setup



The above illustration elucidates the general hardware setup for the tracking system. In addition to the physical configuration, the software will need to be configured. In the central server, each individual camera's position and orientation must be recorded with exact precision. More than four cameras can be setup to increase accuracy and reduce tracking dead zones, but a larger ethernet switch may be required.

It is critical that the cameras' positions and orientations remain static throughout the use of the tracking system. If a tracking camera is moved, the mathematical functions will not return the correct positions and the system will have to be recalibrated.

Cameras



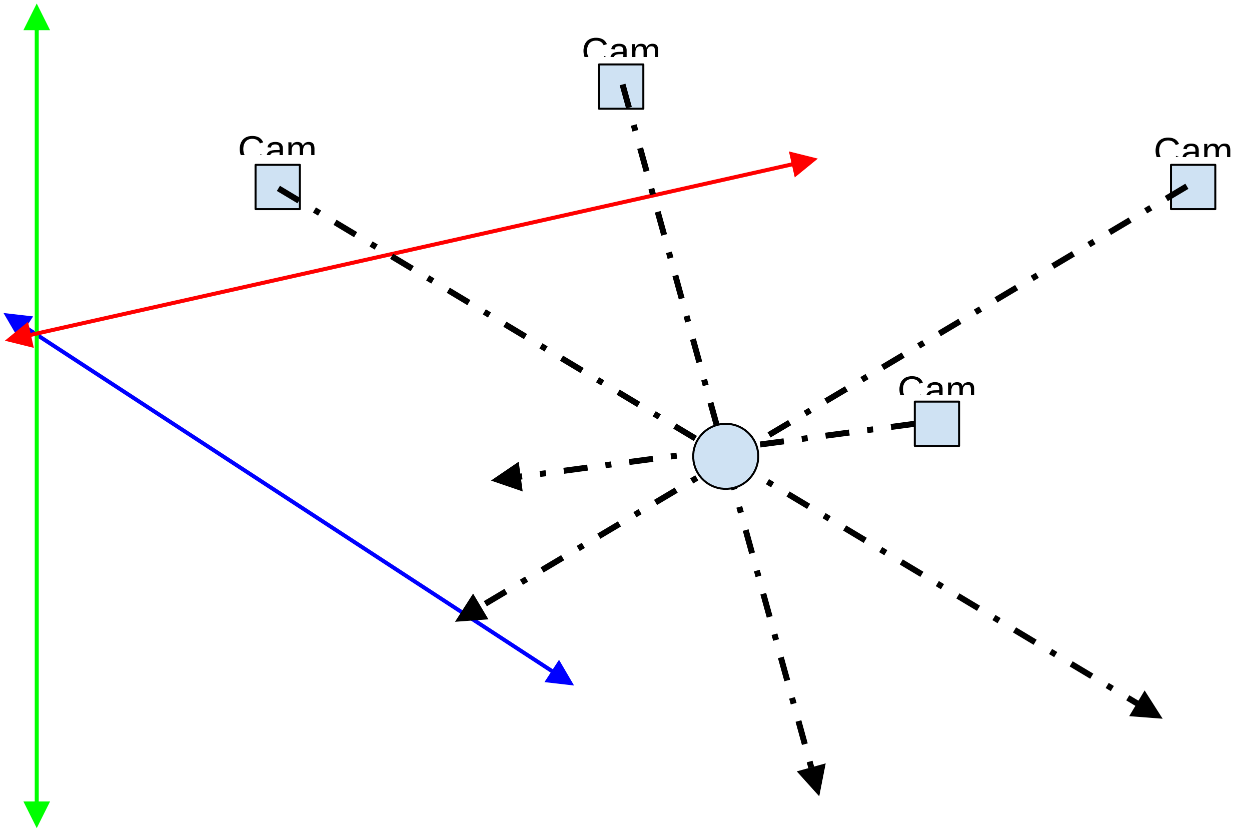
The camera units have two components: the camera and its attached microcontroller. The microcontroller will receive the input from the cameras. From the input, a simple color detection routine will scan the frame for high blooms of light. Once a bloom is detected, the screen space positional coordinates of the bloom will be sent to the central server.

Detection of the bloom will be conducted using a special light-detection camera developed by Charmed Labs.

Central Server

The central server is responsible for determining the actual positional coordinates of the bloom. As the position of each camera is known, and the two-dimensional position of the bloom is known for each camera, virtual lines and vector math can be used to derive the position of each coordinate.

Each camera’s two-dimensional position will be used to define a three-dimensional vector in world space. The two-dimensional offset and camera’s position and rotation in world space will be used to create the camera’s vector-line equation. For each vector line equation, the nearest point intercept will be calculated. The final position of the object in world space is the average of every recorded intercept.



After this process, the position is successfully retrieved and can be used for a variety of purposes. The end vision of the central server component is to provide an interactive digital visualization that can assist the robot in pathfinding a basic obstacle course.

Possible Challenges, and Proposed Solutions

Attitude Derivation

In its current form, the rotation of the tracked objects is unable to be derived. However, there is a simple solution. The robots can mount an onboard compass, and the compass data can be either sent to the central server or the robot can use the compass data locally for calculating correct turning angles.

Accidental Movement of Camera Positions

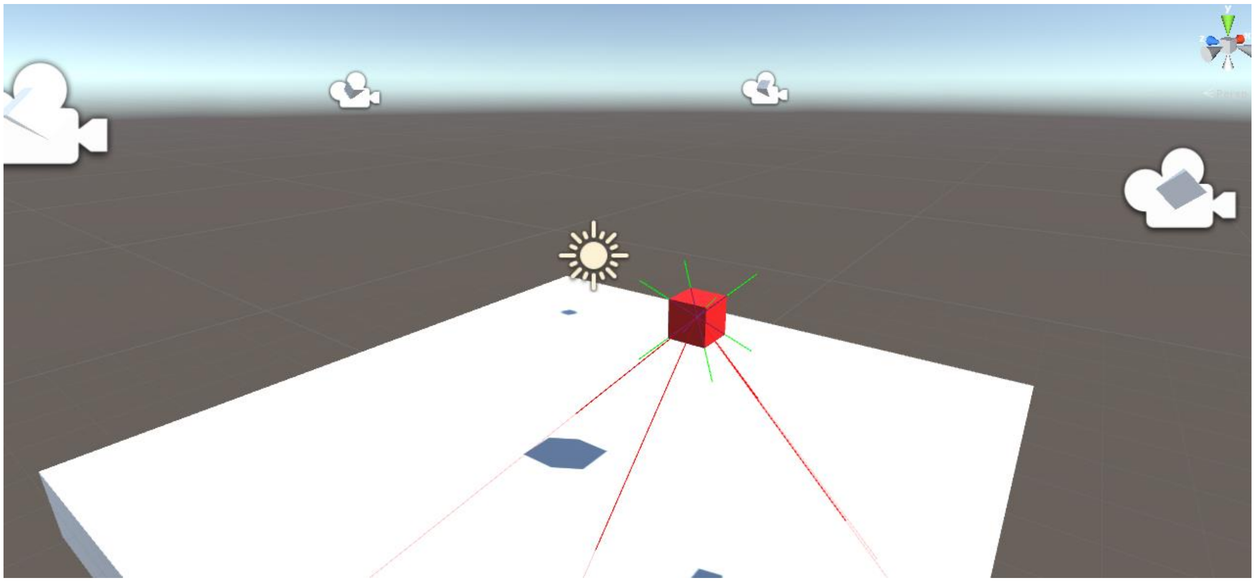
The whole system depends on the camera positions remaining static. Thus, if a camera’s position or orientation changes unexpectedly, the system will cease to function properly. Though no simple solution exists for re-calibrating the system in the event of an unexpected reorientation, simple checks can be introduced to shutdown the system. This would prevent incorrect navigation that could potentially damage the robots.

Algorithm Edge-Case Scenarios

The line-drawing algorithm present on the central server has a rare edge-case scenario. This edge-case was discovered in the technical simulation. When two cameras are pointing directly at each other, floating point errors can create two lines that are identical. As identical lines have an infinite amount of intercepts, this introduces errors into the positioning calculation. Luckily, the check for this edge-case is very simple: just check if the vectors and their inverses are identical.

Technical Simulation of Concepts

A technical simulation of the tracking system was built in the Unity Game Engine. The simulation was programmed to only use the known variables that would be available in a real world scenario. The scenario also proved the concept is feasible with a high degree of accuracy. For verification purposes, the source code of the simulation will be uploaded at a later date.



Estimated Budget Requirements

Raspberry Pi (5x)	$188.85
100 LED + 50 IR LED	$14.43
Mini Tripod	$27.96
Pixy Color Tracking Camera (4x)	$300.90
8-Port Ethernet Switch (1x)	$22.85
Ethernet Cables (1x Short, 4x Long)	$27.19
Total Cost (with Tax)	**$557.28**

Contacts

Owen Bartolf, Project Manager
Email: owen.bartolf@Gmail.com
Mobile: 1-415-577-1440

Bibliography

"GPS Accuracy." GPS.Gov. United States Air Force, 10 Feb. 2017. Web. 22 Mar. 2017.

Laertius, Diogenes. Peithô's Web. Trans. C. D. Yonge. N.p., n.d. Web. 22 Mar. 2017.

Nave, R. "Inverse Square Law, General." Hyperphysics. Georgia State University, n.d. Web. 22 Mar. 2017.