Target Tracking Remote Controlled Robotic Vehicle

University of St Andrews

sicsa* The Scottish Informatics & Computer Science Alliance

By Adebayo I. Osipitan

Supervised by: Michael Weir Second Marker: Saleem Bhatti

Abstract and Project Aim

Abstract

A robot is defined as "*a machine system that can carry out a series of complex actions automatically, especially one programmable by a computer*", this notion is coined from the word "*Robota*", which means forced labour (Oxford Dictionary 2014).

Each action performed by a robotic system is facilitated by simultaneous use of a variety of sensors and actuators. The coordinated interaction and feedback between sensors, allow the system to traverse its immediate environment and accomplish given tasks.

In this specific case, the PINESBOT uses of both a Camera and sonar sensors achieve goal tracking and autonomous movement.

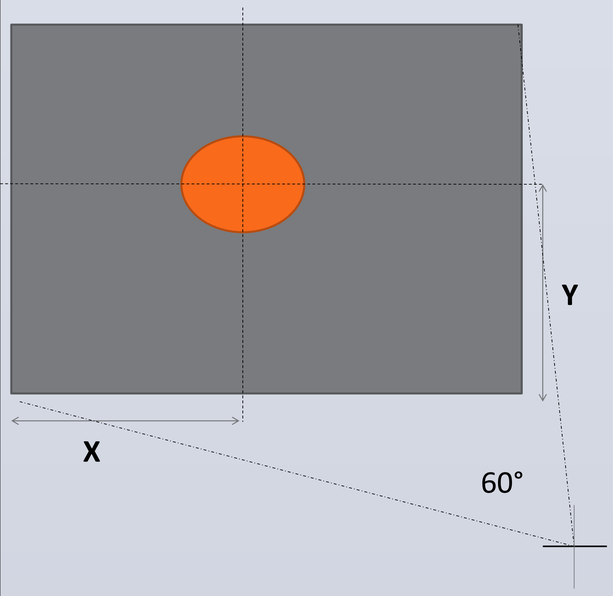
Aim

The aim of this project is to design and implement a robotic vehicle equipped with various kind of sensors to facilitate autonomy of visual object tracking in a 2D space, wireless control and facility to allow for autonomous movement.

Achieved Objectives

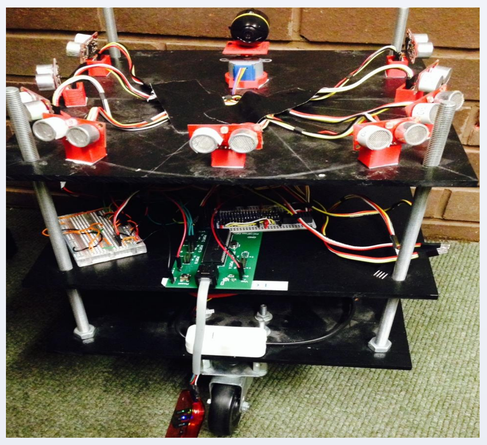
- **Goal Tracking:** Goal tracking was achieved using a wireless mini webcam attached to a stepper motor for 2-D motion control. That is real time images were processed to retrieve x,y coordinates which were then translated to stepper control during tracking.
- **Private wireless network:** A wireless network interface was created for communication between the robot and the User-Control interface(Controller).
- **Interface for image viewing:** The user interface was a rich GUI that allowed the user to visualize the target in real time.
- **Interface for remote robot control:** The user interface also allowed the user to remotely control the motion of the robot as well as speed within a given radius.

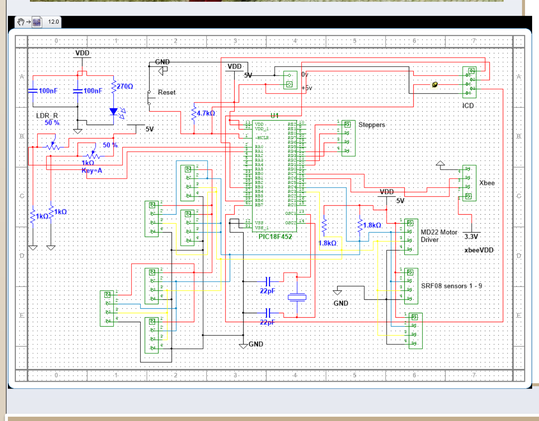
2-D Target Tracking



- Image processing made use of C++ OpenCV libraries.
- For this project, tracking along the X-coordinate was most important, as tilt motion was not available.
- Limitation in hardware allow only for tracking targets in pixel space which does not translate directly to real space.
- As a solution for future works the use of Triangulation methods which use trigonometric methods for locating an object in real space.

The Robotic System





PINESBOT

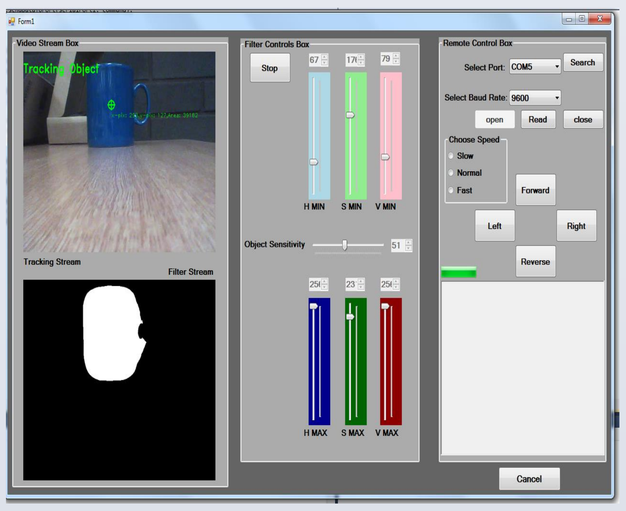
The robot pictured on the right is currently just a remote controlled robot where, the system movement and speed control is mostly controlled by the user using the UI inputs. But target tracking is.

The hardware overview includes:

- A multipurpose PIC microcontroller board
- Radio frequency router module for wireless communication
- Independent motors
- A wireless webcam
- Stepper motor based Platform for rotating the webcam

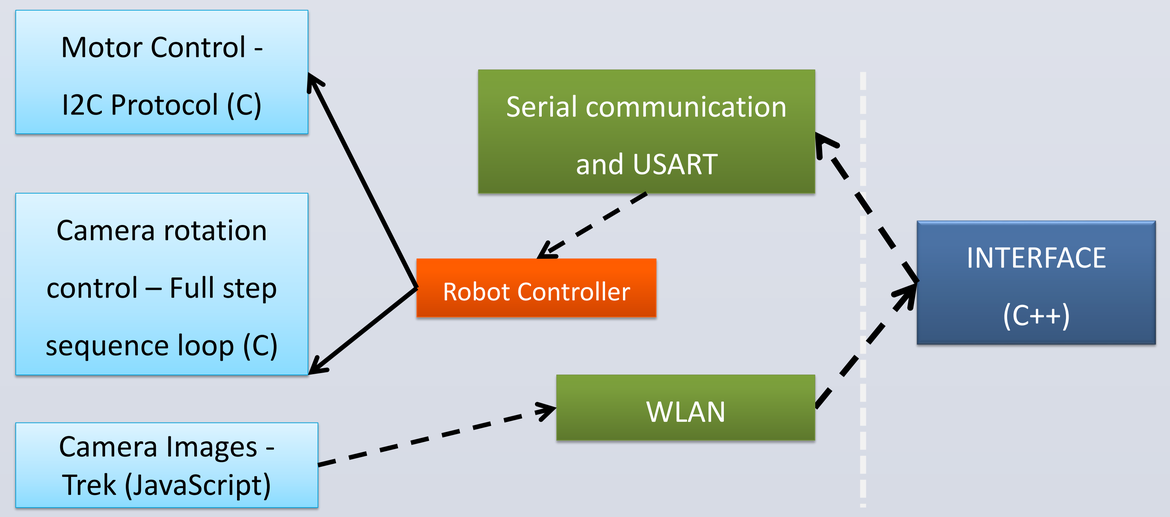
The system software was implemented using both C and C++ for low and high level applications respectively, with OpenCV 2.4.8 libraries for image processing from the attached webcam.

Interface and Controller Architecture



The interface was created as a windows form, using Visual C++, it was developed to be a rich UI for high flexibility, such as;

- Serial COM port selection.
- Serial COM baud rate selection.
- Three distinct modes for movement speed.
- Image filtering for target selection.



Future Works

- Follow up for future improvements include; creating an autonomous mode of operation by using the ultrasonic sensors to facilitate obstacle avoidance within the robot's immediate environment. Integrating additional sensors, such as an accelerometer and gyroscope (to measure robot speed and relative position) as well as depth perceptive robot vision using triangulation.
- In implementing an autonomous use of both navigation algorithms and potential fields methodologies can be employed, as these work best for the two degrees of freedom (DOF_{frame} and DOF_{system}) constraints imposed by the current mechanical design of the system.
- Also, further improvements on the remote control operations, the sensors could be used to warn the user via the UI about obstacles within close proximity,. Movement towards that direction can then be restricted in the UI.