Autonomous
Smart Car

Gerald Leung, Rui Han, James Tai, Hans Woo, Ying Teo

Supervisor: Kin Fun Li
**Introduction**

This report documents the design and implementation of an autonomous vehicle capable of tracking and following a person in an indoor environment. The motivation behind this idea is to help people with carriage transportation, may it be luggage, groceries, etc. This idea also proves to be very useful for handicap/disabled applications.

The autonomous vehicle used is an off-the-shelf RC (remote control) car. The autonomous vehicle must perform two main tasks: follow a user, and avoid obstacles in real time. In order for the autonomous vehicle to achieve these features, the system is implemented with a combination of computer vision, distance sensors and controls software algorithm. Additional software tracking algorithms are also implemented to recover the user of interest if the detection of the user is lost. This is accomplished by using a simple computer vision and IR distance sensors that will be discussed in detail later in the report.
# Table of Contents

I. Introduction .......................................................................................................................... 2

List of Figures.......................................................................................................................... Error! Bookmark not defined.

1. System Overview .................................................................................................................. 5
   1.1. RC Car ......................................................................................................................... 5

1.2. Distance sensors .............................................................................................................. 6

1.3. Camera ............................................................................................................................. 7

1.4. Beacon for tracking .......................................................................................................... 7

2. Hardware Design .................................................................................................................. 8
   2.1. Microcontroller ............................................................................................................. 8
       2.1.1. Steering Control ................................................................................................... 9
       2.1.2. Electronic speed Control ...................................................................................... 9
       2.1.3. Camera Tracking .................................................................................................. 10

2.2. Beacon ............................................................................................................................. 13

3. Method of Tracking ............................................................................................................. 14
   3.1. Range Detection .......................................................................................................... 14
   3.2. Direction Detection using PixArt Camera ................................................................... 15

4. Software Algorithm ............................................................................................................. 16
   4.1. PI Controller ............................................................................................................... 17
   4.2. Direction Prediction and Adjustment .......................................................................... 18
   4.3. Object Avoidance ....................................................................................................... 18

5. Conclusion ........................................................................................................................... 18

6. Limitations and Suggestions .............................................................................................. 19
   6.1. Appendix .................................................................................................................... 20
       a. Hardware Schematic ................................................................................................. 20
1. System Overview

In order to fulfill the design objectives, the autonomous vehicle is equipped with five distance sensors, and a camera tracking system. The system also comprises of a beacon that the user must wear in order for the car to track the user of interest.

1.1. RC Car

The RC model car used in this project is the Traxxas Slash. This particular model proves to be easy to interface with a digital controller, and is large enough to accommodate space for mounting a variety of sensors. The car is equipped with a servo control motor for steering, an electronic speed control, and a DC motor that drives the car, and a battery pack rated at 3000 mAh. The steering and electronic speed are controlled using a PWM (pulse width modulation) signal. The following figure shows the physical car used for this design.

![Figure 1 Traxxas Slash Model Car](image-url)
1.2. Distance sensors

The distance sensors must be mounted in such a way to cover as much range as possible for object avoidance. The following figures show how the sensors are physically mounted onto the autonomous car.

![Figure 2 Distance Mount Configuration](image)

All sensors are connected to one breakout board that is then connected to a microcontroller development board as shown in the following figure.

![Figure 3 sensor interface breakout](image)
1.3. Camera

The camera is mounted in such a way to optimize the viewing angle that will be later discussed in this report. The following figure shows the physical camera mount configuration in the back part of the car.

![Figure 4 Camera Mounting Configuration](image)

1.4. Beacon for tracking

The beacon that the user wears consists of two IR emitters that the camera will use to track the position of the user. The following figure is the physical construction of the beacon.

![Figure 5 Beacon Construction](image)
2. Hardware Design

2.1. Microcontroller

A suitable microcontroller needs to be selected to meet design specifications for this particular application. The following table lists the microcontroller design requirements:

- **Input Requirements**
  - 5x Analog to Digital Converts for IR sensors
  - 1x I2C Interface for PixArt camera

- **Output Requirements**
  - 2 PWM channels (RC servo steer and Electronic Speed Control Interface)

- **Microcontroller Tasks**
  - PID speed controller
  - Data conversion processing for sensors
  - Follower Task
  - Object Avoidance Task

- **Additional Microcontroller Features**
  - Low Power Operation and sleep modes
  - UART debugging
  - Standard C libraries and Software development tools

The Microchip PIC24FGA002 microcontroller satisfied all the microcontroller design requirements. Since the car will be running on batteries, it is incentive to choose a low power microcontroller. The PIC24F controller offers a low power idle mode, which has the ability to shut off the CPU core when no program execution is running while keeping the hardware peripherals running. For example, in the firmware design, when there is no data available from the sensors, the CPU core is shut off, while leaving the PWM module running to keep the car in motion until data is available to wake up the CPU and read the new sensor data to control the car accordingly. From the PIC24F datasheet, an average of 24mA of current can be reduced when the processor is in idle mode. Microchip also provides a wealth of software support libraries and application notes that help speed up development and are also very competitively priced.
2.1.1. Steering Control

The RC car is equipped with a standard servo motor that is mechanically linked to the steering mechanism of the car. A servo motor is a small device that has an output shaft that can be positioned to a specific angular position by sending it a coded signal. As long as the control signal exists on the control line, the servo motor will maintain its position. The following figure illustrates the control signal required to control the servo motor.

![Figure 6 Control signal for steer control](image)

From an analysis from the figure above, a pulse is generated every 20 milliseconds. The duration of the pulse dictates the angular position of the servo. For the particular servo motor equipped on the RC car, a 1.0 millisecond pulse steers the front tires to hard left, while a 2.0 millisecond pulse steers the front tires to hard right. A 1.5 millisecond pulse keeps the front tires strait. This signal is generated using a PWM (pulse width modulation) channel on the PIC24F controller. With the PWM channel implemented, the microcontroller is now able to provide steer control of the autonomous car.

2.1.2. Electronic speed Control

The RC car is equipped with an electronic speed controller that takes in a logic level control signal and drives the DC motor accordingly. The electronic speed controller is actually a motor driver that allows logic level control signals to drive large loads in the range of Amps. The following figure is the electronic speed controller equipped to the RC car.

The control signals are very much similar to the servo steering control signals mentioned in the previous section. Again, a pulse is generated every 20 milliseconds, and the duration of the pulse dictates the speed of the motor. A 1.0 millisecond pulse sets the motor speed to maximum reverse speed. A 2.0 millisecond pulse sets the motor speed to maximum forward speed. A 1.5 millisecond pulse sets the motor speed to 0. Again, a separate PWM channel is used to generate the required control signal to control the electronic speed controller. With the PWM channel implemented, the microcontroller is able to provide speed control to the car.
2.1.3. Camera Tracking

A camera onboard the autonomous vehicle using computer vision is used to track the range and direction of the user of interest. The following table lists the design requirements for this task.

- Low Power
- High Refresh Rate
- Low Data Capacity to reduce the workload on the digital controller
- Low Cost

The PixArt camera taken apart from the Nintendo Wii system fulfilled the design requirements. The PixArt camera is equipped with an image processor inside that is capable of tracking up to four solid light sources (blobs) and sends the pixel coordinates of the detected blobs out through the I2C interface. This relaxes the workload on the digital controller by letting the camera perform all the image processing computation. As discussed in the previous section, the less computation the PIC24F processor has to process, the more time it can spend in idle mode to reduce power consumption.

To illustrate how the camera works, suppose the camera is directly pointing at a whiteboard and the camera resolution is 1024x768. If a laser pointer is used to project a solid dot onto the center of the whiteboard, the camera will detect the light source (blob) and it will send the pixel coordinates of the detected blob (512 x 384) out through the I2C communication interface.

The camera has a maximum polling rate of 200Hz, which is suitable based on the mechanical response of the vehicle.

The camera requires a dedicated 25 MHz clock source. A Pierce gate oscillator circuit was built to supply the camera with the required clock source. The following figures are the schematic and the physical hardware setup of the camera.

---

**Figure 7 Schematic Interface of the PixArt Camera**
2.1.4. Distance Detection

To achieve the object avoidance task, the autonomous vehicle needs the distance sensors to detect the objects around the vehicle. The following list shows the design requirements for this task.

- Short range detection
- High accuracy of detection
- Low power consumption
- Low Cost

IR (infrared) sensors were chosen for this project. There are debates around sonar sensors, touch sensors and infrared sensors. The sonar sensors have better accuracy, but suffer from “ghost echo” effect, which causes the transmitted sonar signal to deflect off walls that are not perpendicular to the sensor. Touch sensors are simple to implement, but does not provide variable distance detection.

Five Sharp GP2Y0A21YK infrared distance sensors fulfill our design requirements. The IR sensors have a distance range from 10 to 80 cm and consume an average current consumption of 30 mA and provide reasonable accuracy for our application.

The sensors are supplied with 5 volts and the output range is from 0 to 3.2 volts. Fortunately, our PIC24F microcontroller operates on 3.3 Volts; therefore, no signal conditioning is necessary. The five Sharp GP2Y0A21YK sensors are interfaced to the PIC24F analog to digital converters. The analog to digital converters are 10 bits.

The following figure is the analog output voltage characteristics of the Sharp IR distance sensors.

![Figure 8 Sharp IR sensor output characteristics](image)

Notice that the output is non linear with respect to distance. This issue was rectified by plotting the inverse of the distance instead to obtain a linear relationship between the analog output voltage and the reciprocal of the distance. The following figure illustrates this effect.
We now have a linear relationship between the output voltage and the inverse distance between the range of 10 to 80 cm. The slope of the graph is approximately 27 Volts.cm. Therefore, the following formula illustrates how the physical distance is computed in the firmware from using the analog to digital converter in real time.

\[
d = \frac{1024 \text{ steps}}{3.3 \text{ V}} \times \frac{27 \text{ V cm}}{ADC \text{ sample}}
\]
2.2. **Beacon**

The beacon uses two wide view angle infrared emitters to ensure the maximum detection performance. The infrared emitter has light wavelength 950nm which matched the infrared filter that attached in front of the tracking camera. Then the beacons are attached to a belt powered by two AA batteries. By some calculation, the two emitters can work continuously for approximately 5 to 6 hours.

The following figure is the physical construction of the IR Beacon.

![Figure 10 IR Beacon used for Tracking](image-url)
3. Method of Tracking

3.1. Range Detection

In short, the distance between two infrared dots that PixArt camera detected will vary depending on the physical distance between camera and the beacon. The closer distance between the robot and the target, the further distance between two dots, and vice versa. The approximately distance data are experimented and calculated using MATLAB and plotted to the figure below. Therefore, the approximately distance between robot and the beacon can be determined.

Through several experiments we could plot the physical distance vs. Dot Distance for distance characteristic. The further the distance between the car and the IR emitters, the less dot distance is.

From this experiment, we were able to make the car recognize how far a person is.

![Distance Characteristics of camera/beacon configuration](image)
3.2. **Direction Detection using PixArt Camera**

The two infrared light sources detected by the IR camera (PixArt camera) in plotted onto a 2D plane. The resolution of the plane is 1024 x 768. In order to obtain the movement direction data of the target, we sampled and calculate the midpoint of two dots. Also the plane is separated into three parts, the left limit is 430 and right limit is 562 (15% of the plane). Once the midpoint is on the center of 2D plane, the car recognizes the person is walking straight. Therefore, depending on the location of the midpoint, the car can recognize the direction of the target.

![Direction Prediction Diagram](image)

In addition to the direction prediction, we integrate a buffer that kept tracking the midpoint position and recorded the data history into a table. Then refer to the table, the average midpoint position is calculated to ensure the most accurate direction that the robot is following. The reason is because there will be overshoot error data that camera misread from the external light sources. The buffer will eliminate the possible errors and keep the robot drives in the expected direction.
4. Software Algorithm

The autonomous vehicle must perform two main tasks: following the user of interest and avoiding obstacles in its way. There are various methods to coordinate the two tasks. One method is using a real time system to provide the multi thread feature and interrupts driving algorithm. The other method is using polling instead of interrupt and an algorithm to switch from one task to the other.

Building a real time operation system on this PIC24F is feasible but it takes time and the operation system adds overhead on the performance of the system as well as it will consume more power. From the programming experience of the team and time line of this project, the second approach has been taken and implemented successfully.

Following is the overview of the whole software structure.

![Figure 13 Flow Chart of Software Algorithm](image)

Above is one cycle of the main task, and the system will start from “process camera data” when it reaches the end of “tree”. Each individual function will be discussed in following sections.
4.1. PI Controller

If the beacon is detected by the camera, the digital controller can compute its range and direction with respect to the vehicle. Distance control is needed for this system since the car needs to be able to speed up, slow down, and stopped. In addition, due to the limited viewing angle of the camera (30 degree up and 20 degree down), there will be blind spots if the target is too close to the camera. Thus the car needs maintain an optimal distance away from the target. The optimal distance following distance between the target and the car was set to 80 cm. A PI controller is used in the software algorithm to maintain the optimal following distance.

![PI Controller diagram](image)

For the PI distance control in this system, it calculates the proportion of how much speed needs to increase to reach the target. The distance difference between the optimal and measured distance divided by the distance error constraint is the proportional part of the control. The result form the proportional calculation multiply the available speed factor will be the integration part. Finally, the integration value adds to the current speed will be the new driving rpm for the car. Current speed minus the integration factor will be the new speed to slow down the car.

Furthermore, when the car is reaching the blind spot of the camera, it needs to be stopped. Since there is no break for the car, reverse driving for short period of the time is used for stopping the car.

For detailed code, please see the attached code section.
4.2. Direction Prediction and Adjustment

These two important functions are used while tracking the target. With the position of the IR-emitter, the car will make the corresponding turn. Above section have already discussed the algorithm, here will present the software implementation.

When the target is out of the camera sight, the average of the position buffer will be used to predict the target moving direction. And the car will try to searching the target in that direction. In the case of IR-emitter is still in sight, the car will make tiny adjustment if the target is not in the centre of the car’s sight.

4.3. Object Avoidance

There are five sensors used in the system: four front sensors and one back sensor. All sensor data are polled every clock cycle, and used to control the steering of the car.

Here are the detailed four scenarios:

- Front left and from right sensor detect the min distance (25 cm), this indicates the car cannot make any turn to avoid the object but to reverse.
- Front left sensor detects the turning distance (35cm), right sensor doesn’t detect anything within 45cm as least, and front straight sensors detect enough distance (70cm ) for any turn; this indicates the car needs to do right turn.
- Front right sensor detects the turning distance (35cm), left sensor doesn’t detect anything within 45cm as least, and front straight sensors detect enough distance (70cm ) for any turn; this indicates the car needs to do right turn.
- Two front straight sensors detect the distance (45cm), this indicates the car needs to make either left turn or right turn to avoid the object. Based on the above detection algorithm, the car can drive with the correct steering.

5. Conclusion

An autonomous vehicle capable of tracking and following a target was design and implemented. In order to track a target, the range and direction with respect to the vehicle must be known. This was accomplished by using a combination of an IR beacon and a special camera that specifically detects and outputs blobs to a digital controller. The autonomous vehicle is also equipped with distance sensors for avoiding obstacles if any are in the way.

The software algorithm performs two prioritized task that follows the target of interest, and avoiding obstacles. A simple PI controller was implemented in software to maintain an optimal distance between the car and the target while in motion. If an obstacle is in the way of the vehicle during this process, the obstacle avoidance task is given a higher priority.

The design and implementation proved success. The car was able to follow and maintain an optimal distance between the target and the autonomous vehicle fairly smoothly. The autonomous vehicle is also able to avoid and turn away from obstacles if there are any in the way during the following task.
6. Limitations and Suggestions

The difficulty of controlling the car increases as the speed of the vehicle increases due to the mechanical response of the system. The car needs to be set at a walking pace speed in order for the system to function optimally. The tracking system is also unable to detect the beacon when making sharp turns. The detection range of the camera and the beacon is also limited to 2.5 meters.

The limited detection range is due to the limited intensity of the IR emitters on the beacon, in which the camera fails to detect. The IR emitters are supplied with a constant current supply of 100 mA. If the intensity of the IR emitters is increased, the camera can detect it from a further distance. The intensity of the IR emitters can be increased by pulsing the IR emitters up to 1 or 2 Amps for a short duration of time.

Reversing the tracking configuration by mounting the camera on the beacon and the IR emitters on the car could also possibly enhance the vehicle’s ability to detect sharp turns made by the target.
6.1. Appendix

a. Hardware Schematic

Figure 15 Microchip 28 Pin Starter Board Schematic Page 1
Figure 17 Sensor Interface Schematic

The following table lists the physical IR sensor connections on the car:

IR1 -> Front Right IR Sensor
IR2 -> Front Mid Right IR Sensor
IR3 -> Front Left IR Sensor
IR4 -> Front Mid Left IR Sensor
IR5 -> Rear IR Sensor

NOTE:
Refer to "16-BIT 28-PIN STARTER DEVELOPMENT BOARD SCHEMATIC" in the "Microchip 28 PIN STARTER BOARD USER GUIDE" Document

Note:
PicoJet Camera is interfaced to the DC2 module of the PIC24FJ34GA002