Paper 080, INT 303

Centralized Vision-based Controller for Multi Unmanned Guided Vehicle Detection

Ravindra Thamma
Central Connecticut State University
thammarav@ccsu.edu

Leela Mohan Kesireddy
Central Connecticut State University
thammarav@ccsu.edu

Haoyu Wang
Central Connecticut State University
wanghao@ccsu.edu

Abstract

The traditional track-based unmanned ground vehicle (UGV) cannot deviate from the pre-recognized route due to track limitation in its navigation methods. Track limitation has to be overcome in order to render a UGV more flexible. It is desirable to have a UGV move without predetermined tracks and have the ability to deviate to and from routine routes, in order to have flexibility in tasks. This research proposes a navigation system to aid multiple UGVs in navigating to various locations without any physical tracks and without collision with each other. This research demonstrates the image-recognition-based trackless navigation system to enhance the flexibility of multiple UGVs.

This research implements an image recognition algorithm to identify the position and orientation of multiple UGVs using a Centralized Image-based Controller Unit (CIBCU). This CIBCU is connected to a vision system and radio frequency (RF) communicator. The CIBCU implements the image recognition algorithm, anti-collision, navigation algorithm, and centralized control center to track and navigate multiple UGVs without physical tracks. A prototype has been developed to demonstrate and test the Vision-based Navigation System. Statistical analyses have been carried out on this newly developed system to find behavior of positioning error.

Introduction

Conventionally, controlling an unmanned ground vehicle (UGV) and automatic guided vehicle (AGV) has been a challenge. Tracks serve as a key element for navigation systems for a mobile UGV. UGV tracking is a critical component for providing positions, directions, and travel information for motion along a trajectory with minimal deviation. Many researchers have proposed different tracking techniques, such as dead reckoning, navigation using active beacons, landmark- and map-based navigation, ultrasound, and the Global Positioning System (GPS).

Most of the navigation systems available in the present market involve a constant exchange of data between the controller and UGV that is costly in more ways than one. The exchange of data contributes to a slower system, resulting in lower UGV velocity. Furthermore, these systems are complicated by various parts and are often constrained to a predefined area. The post-implementation cost can also be a factor against its use. A newer system that will not carry these drawbacks will be beneficial to this area of study.

Many navigation techniques have been used over the last two decades for tracking a UGV. Dead-reckoning [1] is a process of estimating one's current position based upon a previously determined position of a UGV by advancing a previous position based upon known path and speed over a period of time. However, incremental motion often results in errors. Navigation using active beacons utilizes beacons [2] such as laser, sonar, or radio. This technique determines the position of a UGV by drawing a triangle through installed beacons and measuring the distance. The disadvantage of this technique is the inaccuracy of distance measurement caused by signal delay, as well as installation and maintenance costs. GPS systems are advanced and accurate at tracking the position of the UGV, but GPS does not work in an indoor environment where satellite signals are often blocked.

Presently, wireless techniques are extensively used to track the UGV using distance measurement techniques. Radio frequency (RF) and ultrasound [3, 4] are extensively used in the navigation techniques. In some cases, both ultrasound and RF are used together for greater precision. All of the previously mentioned navigation systems involve a constant exchange of data between the controller and the UGV, resulting in large amount of overhead, and the employment of more sensors and constraints to the predefined landmarks. This results in a higher power consumption and shorter battery life of the UGV. It also results in slower operation and response.

Vision-based navigation may use less data for tracking the UGV position and results include a faster operation and response. This is different from previous navigation systems like the landmark- and map-based navigation techniques, which rely on predefined landmarks and maps, or previously known information about the environment [5]. No assumptions about the knowledge of the location are made for the vision-based navigation system.

The concept of vision navigation has been in development for the last 20 years [6] in the area of mobile robot navigation. Even though it was introduced to overcome the disadvantages in the previous techniques, it is implemented in accordance with the previous techniques. In many techniques, vision systems (e.g., cameras) are used to visualize the environment and to guide the robot. Vision systems are used to find and measure the location of 3D structures with respect to a CAD model [7]. The integration of a CAD model for visual measurement and the direct feedback of measurement results to the CAD model is a key aspect for this technique. In other techniques, vision systems are used to generate a three-dimensional (3D) environmental map from data taken with stereo vision [8]. Vision systems are used to develop more precise segmentation. From the obtained segmentation, a 3D environment is built using occupancy grid and floor height maps. In another vision-based technique, vehicle position and orientation are determined using panoramic images [9]. Omni directional sensors are used for obtaining a 360-degree field of view. Recognizing landmarks in a panoramic image from a prior model of distinct features in a given environment gives information about the robot's location.

Most of these techniques rely on assumptions of the prior knowledge of the scene. Some researchers have proposed using a vision-based system that functions without any prior knowledge of the scene. In this technique, a stereo-based vision system is built from feature correspondences and 3D information from image sequences of the scene [10]. This method uses two different cameras for capturing the image frames at a fixed point in time. One camera is used to capture interface images and a different camera is used to collect the stereo image. The relative position of the camera motion is then estimated by registering the 3D feature points from two consecutive image frames.

There are many different vision navigation techniques proposed by prominent researchers. Various techniques utilize different methodologies to track UGVs with a vision system. Some of the vision techniques use a prior model of the environment [11]. Some of the techniques draw imaginary horizontal and vertical lines to find the position of the vehicle [9]. Other techniques use information from gray scale images to find the path clearance to navigate the vehicles [12].

Some vision techniques use panoramic imagery. Omni-directional sensors are used in obtaining a 360-degree field of view, permitting the various objects near a robot to be imaged simultaneously. The robot's location is found by recognizing landmarks in a panoramic image from a prior model of distinct features in a given environment [11].Other vision techniques find the position of the vehicle using collective measurement data obtained directly from the raw data of gray level images. Such data is independent of the 3D surface texture, is measured in dimensional units, and requires no 3D reconstruction. The control schemes are based on a set of "if - then" fuzzy rules with almost no knowledge about the vehicle's dynamics, speed, and heading direction [12]. In some techniques, robot navigation is calibrated based on navigational lines. The position of a robot is found based on the extracted straight lines, supposing the robot moves on a plain ground. The effect in the image of camera rotation is computed from the homography of a line at infinity. The corresponding vertical lines in two uncalibrated images are then used to compute both the robot heading and a region in the image that corresponds to the free space ahead [11].

One more important feature to be considered in vision navigation is the nature of the vision system. The number and placement of vision system cameras play an important role in the function and performance of the navigation system. Some techniques utilize vision systems that are placed on the vehicle [13], while in others they are placed stationary in the navigation field [8]. Some techniques have only one vision system, while in others, multiple vision systems are placed at different positions in the navigation system.

Although vision-based navigation systems are implemented to get through all of the disadvantages of the traditional navigation systems, some of these systems still depend on traditional techniques like maps and the development of 3D environments from an image system. When such systems use more than one vision system, this further complicates the implementation of vision-based navigation. A vision-based system must be implemented in such a way as to overcome all of these disadvantages, while navigating on plain ground without using any tracks.

All of the previously mentioned systems need large infrastructure and software, resulting in complex and costly techniques. They are all dependent on the previous techniques and require some type of assumptions about the environment. Therefore, there is a need for the development of a navigation technique that uses less infrastructure and simpler algorithms. Such a system should navigate the multiple vehicles with a lower overhead and less software and hardware.

In this paper, a vision-based navigation system has been developed to navigate a UGV from a given position to a predefined final position. Automated software, developed as a Centralized Image-based Controller Unit (CIBCU), will run the algorithms for vision processing, orientation, anti-collision, and navigation. It has been tested by hardware prototype with the vision-based navigation system and UGVs, and data analysis is carried out on test data. The goal of this study is to develop a prototype vision-based navigation system to navigate multiple unmanned ground vehicles.

Methodology

The vision-based navigation system involves processing the image generated by the vision system and navigating multiple UGVs without colliding with other UGVs in accordance with a predefined priority. The basic layout of the proposed solution, as shown in Figure 1, helps to illustrate the methodology. The vision system generates the images and sends them to the vision processing algorithm. The vision processing algorithm then processes the image and generates the coordinates of each vehicle. The orientation algorithm then processes the orientation of each vehicle using the coordinates generated by the vision-processing algorithm. An anti-collision algorithm checks the probability of collision and stops the UGV according to its priority. The navigation algorithm navigates the vehicles according to the orientation of each vehicle. The data of the navigation is then transmitted to the vehicles using RF communication.

The CIBCU Control Panel is the user interface for operating the vision-based navigation system and works with all five main parts of the system. This control panel provides manual control levers and overriding capabilities to the human user and has a user-friendly interface that provides user controls for all of the UGVs.

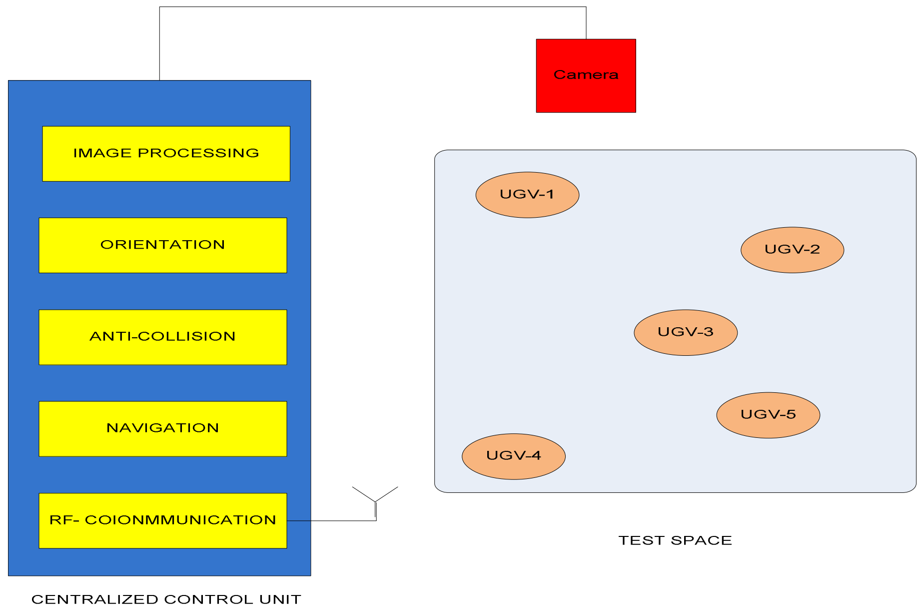

Figure 1. Layout of proposed Vision-based Navigation System

The positions of the different UGVs are captured through the vision system camera, and these captured images are transferred to the vision-processing algorithm. The vision-processing system uses the data from the images to provide *x* and *y* coordinates for the UGVs. The orientation system calculates the orientation angle of each UGV. The position and orientation information is used by the anti-collision algorithm to find the possibility of any collisions. Finally, the information about the *x* and *y* coordinates is fed to the navigation-control algorithm.

The navigation algorithm forms the heart of the entire set up as this system acts as the brain. Navigating all UGVs from their present positions to their final positions is the basic responsibility of the navigation algorithm. The RF communication control system handles the transmission of control data to each UGV. A conventional multiplexed radio transmission serves as the communication medium.

All of the previous discussion is developed using a VB.net program. The main function of this program is to navigate the UGVs according to the positions and orientations derived from the latest acquired images. The flow of the program is to acquire the latest image from the vision system, resize and compress the image for faster processing, generate the coordinates of the vehicles by comparing the acquired image with reference images, calculate the orientation of the vehicles from the generated coordinates, and navigate the vehicles to the final positions.

This flow is implemented by using the previously described algorithms. The program is mainly divided into five different parts, based on these algorithms (see Figure 2):

- Vision processing
- Orientation
- Anti-collision
- Navigation
- RF communication

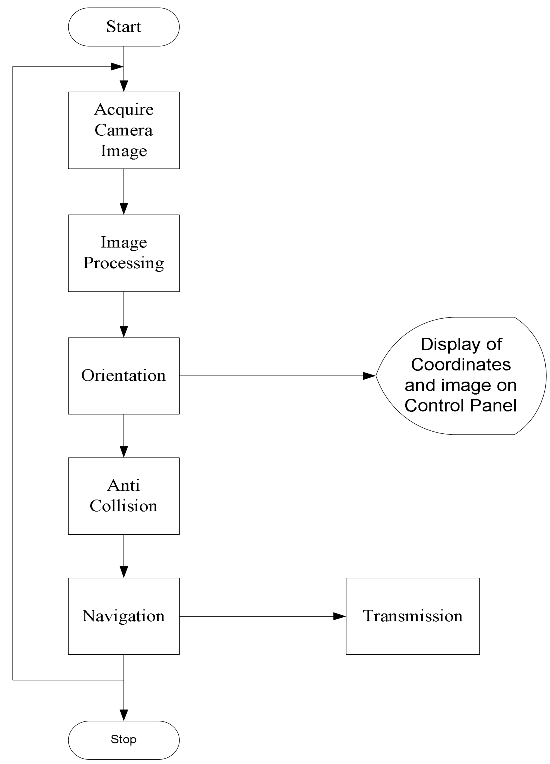

Figure 2. Vision Navigation Program flow chart

A. RF Communication

The signals from the CIBCU are received by the transmission hardware to send navigation signals to the UGVs for navigation. A Parallax *Basic Stamp* (BS2SX) microcontroller in the transmission hardware is programmed to receive a signal from the CIBCU and transmit corresponding navigation signals. Transmission hardware is connected to the controller serial port to receive serial commands, and it is connected to the RF transmitter to transmit radio signals. The BS2SX microcontroller is capable of receiving and sending serial data.

The BS2SX in transmission hardware is programmed to receive a unique command and transmit a corresponding unique set of navigation signals. As the BS2SX receives a command from the serial port, it analyzes the command using an “if else” loop. Each UGV has a predefined set of signals for the navigation of each UGV and some universal signals, which have a specific navigation control for every UGV. Universal signals are range 68–99, and 66 is the universal stop command. Table 1 gives the range of signals for each UGV.

The BS2SX program receives a command through a serial input (SERIN), in the form of numbers. The number received is assigned to a variable, which is first checked to see if it falls in the range of universal signals.

UGV No.	Signal Range
1	11 – 15
2	21 – 25
3	31 – 35
4	41 – 45
5	51 – 55

Table 1. Signal Ranges for Each UGV

If the signal falls in the range of universal signals, then the corresponding signal is transmitted through RF transmitter (Figure 9). If the variable is not in that range of universal signals, then it is verified to which UGV it belongs. Once it falls under a specific UGV range, it is then checked for corresponding signals and sent it out. The transmit signal is a combination of three numbers.

BS2SX receives the commands through the serial port at 9600 baud rates on pin 16, which is connected to the serial port. A PULSOUT signal is sent to the transmitter to place it in a wake-up state before sending the signal. Then, the actual signal is sent out to the transmitter from pin 7. The basic stamp is programmed to receive the signal continuously from the serial port by using a loop.

RF receivers are placed on every UGV to receive the signals. The BS2SX receives the signals through another RF receiver and navigates the UGVs. The receiver on each UGV is connected to a BS2SX, which controls the UGV. All of the UGVs receive all of the signals but only respond to signals assigned to them.

UGV -1			UGV -2			UGV -3	
Signal	**Movement**		**Signal**	**Movement**		**Signal**	**Movement**
11	Forward		21	Forward		31	Forward
12	Backward		22	Backward		32	Backward
13	Clockwise		23	Clockwise		33	Clockwise
14	Counterclockwise		24	Counterclockwise		34	Counterclockwise
15	Stop		25	Stop		35	Stop

UGV -4			UGV -5	
Signal	**Movement**		**Signal**	**Movement**
41	Forward		51	Forward
42	Backward		52	Backward
43	Clockwise		53	Clockwise
44	Counterclockwise		54	Counterclockwise
45	Stop		55	Stop

Table 2. Signals specific to each UGV

Every UGV has a unique signal for every movement (Table 2), all of which are preprogrammed on the UGV. The RF receiver on each UGV is always in the wake-up state to receive the signals. It receives every signal at its frequency and sends it to the BS2SX for processing the signal.

UGV -1	
Signal	**Movement**
11	Forward
12	Backward
13	Clockwise
14	Counterclockwise
15	Stop

UGV -2	
Signal	**Movement**
21	Forward
22	Backward
23	Clockwise
24	Counterclockwise
25	Stop

UGV -3	
Signal	**Movement**
31	Forward
32	Backward
33	Clockwise
34	Counterclockwise
35	Stop

UGV -4	
Signal	**Movement**
41	Forward
42	Backward
43	Clockwise
44	Counterclockwise
45	Stop

UGV -5	
Signal	**Movement**
51	Forward
52	Backward
53	Clockwise
54	Counterclockwise
55	Stop

Table 2. Signals specific to each UGV

Each *Boe-Bot* (Figure 3) UGV is programmed for five important movement functions: forward, backward, clockwise, counter-clockwise, and stop. The BS2SX sends out the pulses to the servo motors according to the signal received by it. Every UGV is programmed to make uniform movements. The movement of the UGV is controlled by the PULSOUT signals to the servos. The servos on every UGV are connected to pins 12 and 13 of the BS2SX.

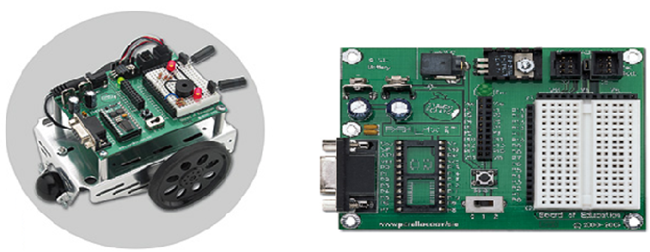

Figure 3. Parallax *Boe-Bot* and *Board of Education*

All of the UGVs receive the entire signals transmitted from the transmitter. Once the signal is received by the UGV, the BS2SX on each UGV analyzes the signal. If the signal received belongs to that UGV, then it responds with specific movement.

B. Centralized Image Based Controller Unit (CIBCU)

The main window of the vision-based navigation system is shown in Figure 4. This screen shows the image of the bots, which are moving along with the *x* and *y* coordinates and the orientation angles for all of the UGVs.

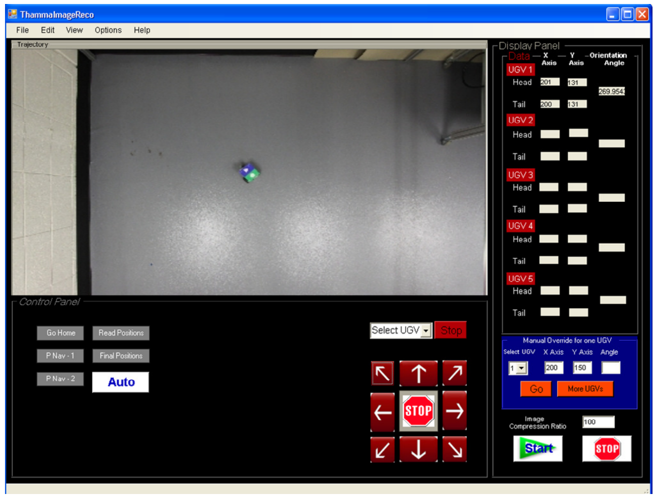

Figure 4. Snap shot of main screen

C. Test Environment

Verification of this vision-based navigation system is accomplished with a test environment, set up to navigate the bots according to the images acquired from the vision system. The setup needs a predetermined space in which the bots are going to be navigated, a frame to hold the camera at a top center of the predetermined space, a server to run the program to process the images and send the navigation signals, five UGVs with RF receivers, and transmission hardware.

A test environment was set up with all of the requirements for full verification. The space for the UGV navigation was determined and a frame was built covering the predetermined area and a vision system was hung from the top of the frame. A server with high processing power was used to handle the overhead caused by the program. The vision system and transmission hardware are hardwired to the server. All of the UGVs were labeled with unique labels, which had the symbols of head and tail of each vehicle.

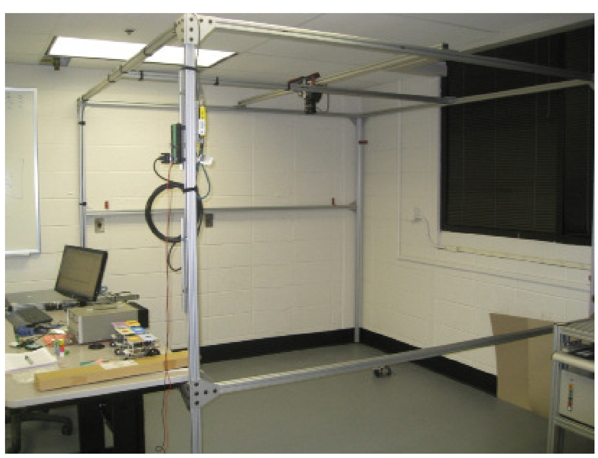

Figure 5. Test space

The UGVs' test space is the predetermined space in which the bots are going to be navigated. The test space is recognized by the frame built surrounding the test area. The frame is built using aluminum bars. It is eight feet in length, eight feet in breath and seven feet in height. The frame holds the vision system at the top of the mid-center of the predetermined area.

A high-quality vision system is needed to provide robust images of the environment. The vision system should be capable of capturing images at regular intervals and should be capable of operating remotely. The vision system should cover the total test space and should provide high quality images. A Canon *50D SLR* camera is used, as it fulfills all of these requirements. It is fitted with a wide-angle lens to cover the test area.

The vision system is connected to the server using the USB cable and is operated by the EOS utility provided by Canon. It can be programmed to capture continuous images with predefined delay between the images and to save in a specific location on the server. It can also be operated manually.

The transmission hardware is built using a *Board of Education* (BOE) component carrier board, a BS2SX, and an RF transmitter. The BOE provides the interface for the BS2SX to connect to the serial port and hold the transmitter module. The BS2SX is programmed to receive the signals from the CIBCU and transmit signals using an RF transmitter. The RF transmitter used in this project is a Parallax 433.92 MHz RF transmitter module. This module comes with a transmitter chip, an antenna, and four connection pins. It operates at a 12,000 – 19.2 K baud rate and transmits up to 500 feet.

Testing and data analysis

All of the previously discussed algorithms and programming are validated by testing in the test environment. At first, synchronization between the different parts of the setup is tested. Then, different modes of operation of the program are tested.

The vision system is tested for remote shooting by connecting it to the controller and taking test shots. Lens zoom and focus are adjusted to cover the whole test area. The vision system can be programmed for different photo formats and orientation of image. The vision system is set to save images in jpeg format, with the date and time stamp in the specified folder by the user. The vision system is set to take images in continuous mode, with an image capture interval. The number of images is controlled from the remote control panel.

RF communications of both the transmitter and receivers are tested for functionality. RF transmission is tested by checking which signal RF hardware it is receiving and which signal it is sending out. The RF receiver is tested by placing a DEBUG code in the receiver program and checking which signals it is receiving. RF communication is tested by sending signals to the vehicles for moving them forward, backward, right, and left. It is tested with all vehicles for continuous signal transmission and transmission range.

The synchronization test is conducted to test the synchronization among the controller, vision system, and RF transmission. It is tested by running the program to navigate the vehicles to verify proper performance. It is tested for all different modes of operation and at different speeds of the UGV and camera intervals.

The vision navigation is first tested by the navigation of one vehicle from the present position to a given position. This is tested by running the program in the individual UGV mode. Only one vehicle is placed in the test field and navigated to a final position specified by the coordinates. In this mode, an image is taken and processed and the UGV navigates toward the final position. After every movement in its trajectory, another image is taken and processed to check if the vehicle has deviated from its path. If the vehicle has deviated from its original path, then it is rotated to get it back to its final position. If the vehicle is near to the final position in accepted tolerance, then the vehicle navigation will be stopped. The data for every position is recorded automatically in a .csv file.

The vision navigation test is run continuously for ten different times by navigating the same UGV between the same start and final positions. All of this data is recorded for analysis using the save data option, as seen in Table 3. The *x* and *y* coordinates of head and tail and orientation of the vehicle for every image is recorded. It gives ten different sets of data. The error in the data will be different every time the bots are navigated. Therefore, mean is calculated from this data to get the mean values. This mean data can be plotted in graphed to illustrate performance of this system.

Mean Values				
Tail X	**Tail y**	**Head X**	**Head Y**	**Orientation Angle**
170	108	170	105	261
127	58	122	58	266
136	61	130	60	260
144	66	141	65	253
154	74	150	72	244
160	80	152	80	240
187	105	164	85	242
171	93	170	92	245
178	101	178	101	248
203	127	182	107	252
213	136	188	112	258
191	114	192	115	305
229	154	191	122	245
194	130	194	130	258

Table 3. Mean Values Table of Test Data Recorded

Tail *y* coordinates vs. Tail *x* coordinates graph

Figure 6 is plotted from the mean data to show the relation between the tail *x* and tail *y* coordinates. There is not much deviation from the expected path.

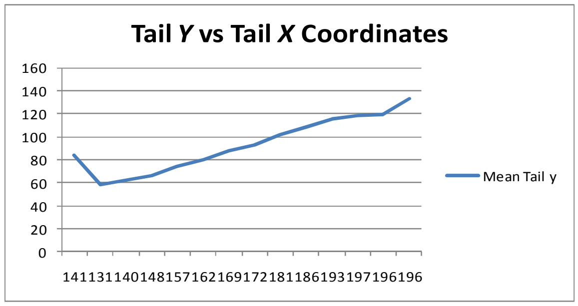

Figure 6. Tail *y* vs. Tail *x* coordinates

Head *y* coordinates vs. Head *x* coordinates graph

Figure 7 is plotted from the mean data to show the relation between the head *x* and head *y* coordinates. There is not much deviation from the expected path.

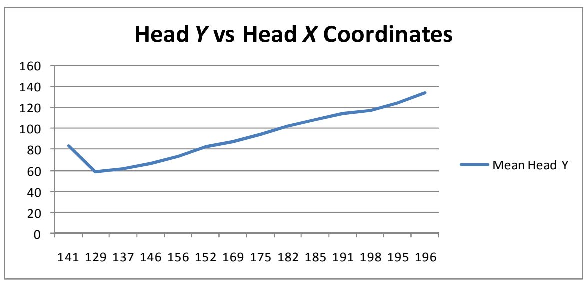

Figure 7. Head *y* vs. Head *x* coordinates

Change in Orientation angle graph:

Figure 8 is plotted from the mean data to notice the deviation from expected values of orientation angle. The orientation angle graph also should be a straight line. However, there is more deviation from UGV's path. The deviation is the effect of deviation of coordinates on heads and tails of the vehicles. We can notice the deviation is only in the area where there is a deviation in the graphs in Figures 6 and 7. This is because much deviation in the orientation angle is dependent on both head and tail coordinates.

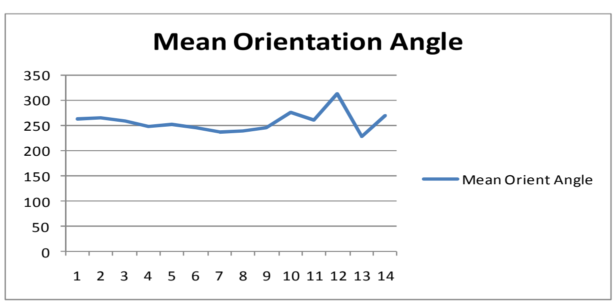

Figure 8. Deviation in Orientation Angle

Bot Path deviation graph:

Figure 9 is plotted from the mean data calculated from the midpoints of mean data to illustrate the deviation from expected values. This graph gives the actual path of the bot. There is not much deviation as the UGV moving from the present position to final position.

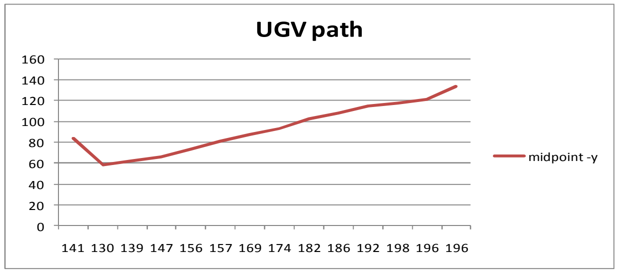

Figure 9. Deviation in UGV Path

Conclusions

The implementation and testing of this vision-based navigation system has allowed some conclusions to be drawn for better operation of the system.

Efficient operation of the vision-processing results in the better operation of the vision-based navigation system. Vision processing is the most important part of the project. Efficiency of the vision-based navigation system depends on effective vision processing. If the coordinates generated by the vision processing are incorrect, all other parts of the system result in error, as all other parts of the system depend on the values of the coordinates generated by the vision processing. Therefore, the vision processing acts as the key feature of this system.

The vision processing depends on the image generated by the vision system and the reference images. Both the image and reference image directly depend on the image labels on the UGVs, therefore the image labels on the bots are the key element for the success of the project. This means more work and time are spent coming up with better image labels that can help in generating exact coordinates of the vehicles. After working extensively with the reference images, we came up with some recommendations to be followed for making image labels:

1. All of the images should have a square background with a different shape on them.
2. All of the shapes on the images should be as sharp as possible.
3. Both the background and shape of a label should be of different colors.
4. Care should be taken to have different colors on the labels for backgrounds and shapes.

References

[1] Bowditch, N. (1995). Dead Reckoning. In The American Practical Navigator (pp. 113-118). Bethesda, MD: National Imagery and Mapping Agency.. Retrieved from http://www.irbs.com/bowditch/pdf/chapt07.pdf.
[2] Rathbone, R. R. (2000). Beacon-referenced dead reackoning: a versatile guidance system. Robotics Engineers, 11-16.
[3] Thamma, R. (2008). Navigational aids to predict the position of automated guided vehicles with ultrasound and radio frequency sensing. Proceedings of the IAJC-IJME International Conference, 154-165.
[4] Thamma, R. (2009). Estimating Position and Orientation of an Unmanned Guided Vehicle with Ultrasound and Radio Frequency Sensing. Journal of Technology Interface. Special Edition, Winter 2009.
[5] Parvaneh Saeedi, P. D. (2006). Vision-Based 3D trajectory tracking for unknown environments. IEEE Transactions on Robotics, 22 (1), 119-136.
[6] Kak, G. N. (2002, February). Vision for mobile robot navigation: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 237-267.
[7] Antonios Gasteratos, C. B. (2002). PRONTO: a system for mobile robot navigation via CAD-model guidance. Microprocessors and Microsystems, 17-26.
[8] Jens-Steffen Gutmann, M. F. (2008). 3D perception and environment map generation for humanoid robot navigation. The International Journal of Robotics Research, 27 (10), 1117-1134.
[9] Fiala, M., & Basu, A. (2004). Robot navigation using panoramic tracking. Pattern Recognition, 34, 2195-2215.
[10] Huei-Yung, J.-H. L. (2006). A Visual Positioning system for vehicle or mobile robot navigation. IEEE Transactions on Information & Systems, 89, 2109-2116.
[11] Guerrero, J. J., & Sagues, C. (2001). Uncalibrated vision based on lines for robot navigation. Mechatronics, 11, 759-777.
[12] Kundur, S. R., & Raviv, D (2000). Active vision-based schemes for autonomous navigation tasks. Pattern Recognition, 33, 295-308.
[13] Wu, C.-J., & Tsai, W.-H. (2009). Location estimation for indoor autonomous vehicle navigation by omni-directional vision using circular landmarks on ceilings. Robotics and Autonomus system, 57, 546-555.

Biographies

RAVINDRA THAMMA is with the Department of Manufacturing and Construction Management, Central Connecticut State University, New Britain, CT 06053 USA (Tel: 860-832-3516; Fax: 860-832-1806; E-mail: thammarav@ccsu.edu).

LEELA MOHAN KESIREDDY is a graduate student at Central Connecticut State University, New Britain, CT 06053 USA (Tel: 860-329-2373; E-mail: kesireddylem@ccsu.edu).

HAOYU WANG is currently an assistant professor of Manufacturing and Construction Management Department at Central Connecticut State University. He received his PhD in mechanical engineering from Syracuse University. Dr. Wang's teaching and research interests include GD&T, CAD/CAM, manufacturing systems, and injury biomechanics.