VISION BASED AUTONOMOUS SECURITY ROBOT

PROJECT PROPOSAL

By:

Kevin Farney

Project Advisor:

Dr. Joel Schipper

EE 451 Senior Capstone Project

December 10, 2009

PROJECT SUMMARY

Computer vision is defined as making useful decisions about real physical objects and scenes based on sensed images [1]. Humans generally take the details of eyesight and the amount of information naturally derived from sight for granted. Machine vision incorporates artificially intelligent systems with image processing systems. The design of this project is to develop an autonomous vehicle with computer vision as its primary sensor for gaining information about its environment.

For most intelligent machines the capability of understanding the visual world is a prerequisite. Typically autonomous vehicles have many different types of sensors all operating simultaneously. This project will use a camera for navigation and obstacle avoidance as well as other tasks the robot must perform.

Applications for this type of technology are numerous. This particular design will be used as a security robot. At the most complex level this autonomous vehicle will receive a signal indicating that a disturbance in a particular room has been discovered. The Vision Based Autonomous Security Robot (VBASR) will then travel to that room and investigate, taking pictures of any culprits that it locates there. After the task has been completed it will return to its charging station.

The VBASR will follow the Sense-Plan-Act procedure. A camera will do all of the sensing. Planning will be done by a processor. Computer vision techniques will be used to make decisions. Based on what the images dictate, the iRobot Create (chassis) will act appropriately. See Figure 1 below for the overall system block diagram.

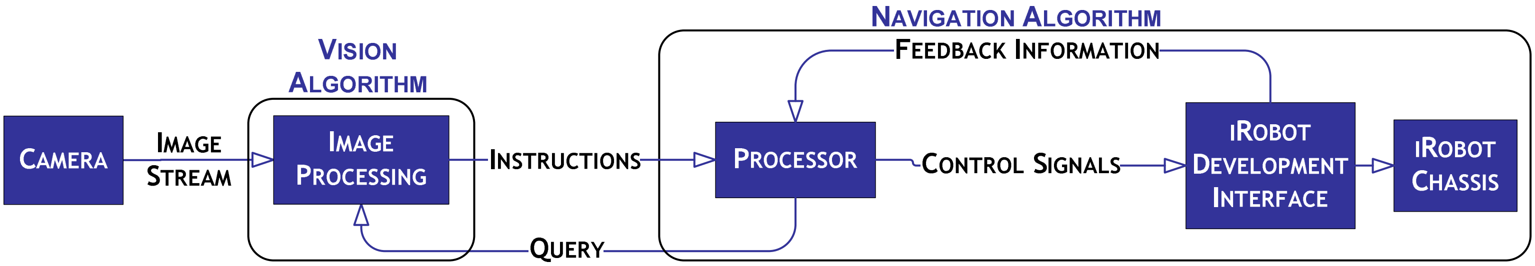

FIGURE 1 – COMPLETE SYSTEM BLOCK DIAGRAM

FIGURE 2 – IROBOT CREATE AND ACCESSORIES

EQUIPMENT LIST

This project is meant to focus on learning and applying computer vision. As a result the hardware is minimal. That is why the iRobot Create was purchased. In this manner low level hardware implementation is avoided. Software implementation will be an issue but that is in the scope of the project.

- Hardware
  - Camera
  - iRobot Create (see Figure 2)
  - Processor
- Software
  - OpenCV
  - Microsoft Robotics Developer Studio
  - Visual Studio 2008

FUNCTIONAL DESCRIPTION

CAMERA – As shown in Figure 1, the camera is the primary sensor for the robot. A simple Logitech webcam has been selected. The camera continually supplies images for the microcontroller to analyze. High-quality cameras, with integrated processors, were researched but deemed unnecessary.

IROBOT CREATE – In Figure 2, the iRobot chassis is pictured along with several of its accessories. The actual chassis of the VBASR (white circular platform) contains many built-in safety sensors such as bump and cliff sensors. The camera will be mounted atop the iRobot Create. The chassis is controlled by the iRobot Open Interface via a serial cable. As shown in Figure 2, the iRobot Create Premium also includes the following: two virtual walls, a self-charging home base with AC/DC converter, standard remote, and a rechargeable battery [2].

PROCESSOR – Currently, a laptop is being used to develop the computer vision software. Upon completion of this proof-of-concept, a small netbook computer (around 2.2 pounds) will be purchased to mount on the robot's cargo bay. The netbook will run Windows, enabling the VBASR to be programmed using Microsoft Robotics Developer Studio (MRDS), and OpenCV for implementing the vision algorithm. Off-board computing and other hardware processors (individual boards) were researched but not chosen. Drawbacks include implementing wireless communications (for the webcam and the iRobot) and powering issues among others. In addition, both of these solutions introduce significant hardware challenges, but the focus of this project is computer vision, not system integration.

OPENCV – For computer vision functions, OpenCV will be used. OpenCV is a large, open-source, compilation of image processing functions. The libraries are downloadable and programmed in C/C++. The Vision Algorithm will pass simple instructions to the Navigation Algorithm (speed up, turn left, etc.) using MRDS. Also see the Vision Algorithm section of the Functional Requirements below.

MICROSOFT ROBOTICS DEVELOPER STUDIO (MRDS) – Microsoft Robotics Developer Studio will communicate and control the iRobot chassis via the iRobot Development Interface (see Figure 1). Details can be found in the Functional Requirements section under the Navigation Algorithm. The iRobot Development Interface is simply a way of communicating directly to the iRobot chassis. MRDS utilizes the bi-directional serial cable to send and receive instructions. Feedback information is received via the safety sensors built into the iRobot create (primarily bump and cliff sensors).

VISUAL STUDIO 2008 – Visual Studio 2008 (VS 2008) is used to program the functions within MRDS. MRDS uses C#, and C# can be coded and compiled using VS 2008. The OpenCV functions and libraries (written in C/C++) can also be used in C# programs by utilizing software wrappers that enable communication between various languages.

PROJECT GOALS

The main goal of this project is to learn and apply computer vision techniques. This will be done to complete the two primary goals described below. The secondary goals are not necessary for the project to be a success.

- Primary – Using Computer Vision
  - Navigate down the center of a corridor.
  - Avoid obstacles in real time.
- Secondary
  - Map the hallway using images.
  - Recognize locations (rooms/offices) using pre-defined images of those locations, enabling point-to-point navigation into rooms.
  - Security Application: Locate and take a picture of an intruder sensed by motion detectors in a particular room. The environment for this goal is the second floor of Jobst Hall in the ECE wing.

FUNCTIONAL REQUIREMENTS

Overall, the specifications for the VBASR are behavioral rather than numeric. The specifications for the primary goals of this project include autonomous navigation, avoiding obstacles, and interacting with the environment in real time. Since humans can interact effectively in real time and the robot is required to interact in a human environment, human reaction time was chosen as the upper limit for the software loop. This requires the main software loop to run in less than 190ms which is the average human reaction time for light stimuli [3].

To accomplish the secondary goals of VBASR, a database of images for each doorway and section of hallway must be created. During navigation, VBASR will compare its images to the image database to determine its current location. Moveable obstacles that may be left in a hallway must be filtered to accurately compare the images.

To accomplish intruder detection, a network of low-cost motion detectors will be monitored via an input signal from a networked computer. When motion is detected after hours in a particular room, the computer will send a signal to dispatch the VBASR to investigate.

VISION ALGORITHM – The vision algorithm provides the primary source of information to the microcontroller (see Figure 1). This is the main focus of the project: learning to use computer vision techniques to gather the information needed to control the robot. Images will be taken via a camera. These images will be still shots, not streaming video. After the analysis of an image is completed, the next image is then requested while the rest of the system responds to the information gathered from the first image. The gathered information will result in instructions that will be sent to the processor. The Navigation Algorithm will utilize MRDS to control the navigation of the VBASR.

NAVIGATION ALGORITHM – As shown in Figure 1, the navigation algorithm and the vision algorithm combine to form a complete system. The navigation algorithm controls VBASR by responding according to the conclusions of the vision algorithm.

To operate effectively in its environment, the VBASR should have a shorter reaction time than humans and must not drive at a rate that exceeds its ability to process images and make navigational decisions. The VBASR will be sampling images as fast as the processor can analyze them. Thus, the robot will have the fastest possible reaction time, ideally far greater than any human. In such a manner, the VBASR will quickly identify and avoid any sudden obstacles and also quickly identify any movement to photograph it.

PERFORMANCE SPECIFICATIONS SUMMARY

Listed below is the summary of specifications for the VBASR.

- Vision Algorithm – Sampling and processing images in real time.
- Navigation Algorithm – Main loop code must execute in 190ms max.
  - Locate and navigate down the center of a corridor.
  - Avoiding obstacles.
  - Cannot outdrive the vision algorithm.
- Secondary
  - Map the hallway using images.

- Recognize locations (rooms/offices) using pre-defined images of those locations, enabling point-to-point navigation into rooms.
- Security Application: Locate and take a picture of an intruder sensed by motion detectors in a particular room.

SOFTWARE FLOWCHART

Figure 3 shows the preliminary software flowchart. Each of these tasks will be performed utilizing computer vision. The flowchart shows the decisions that are required of the vision algorithm. It also shows the priority levels associated with each task. The flowchart also includes the secondary goals completed.

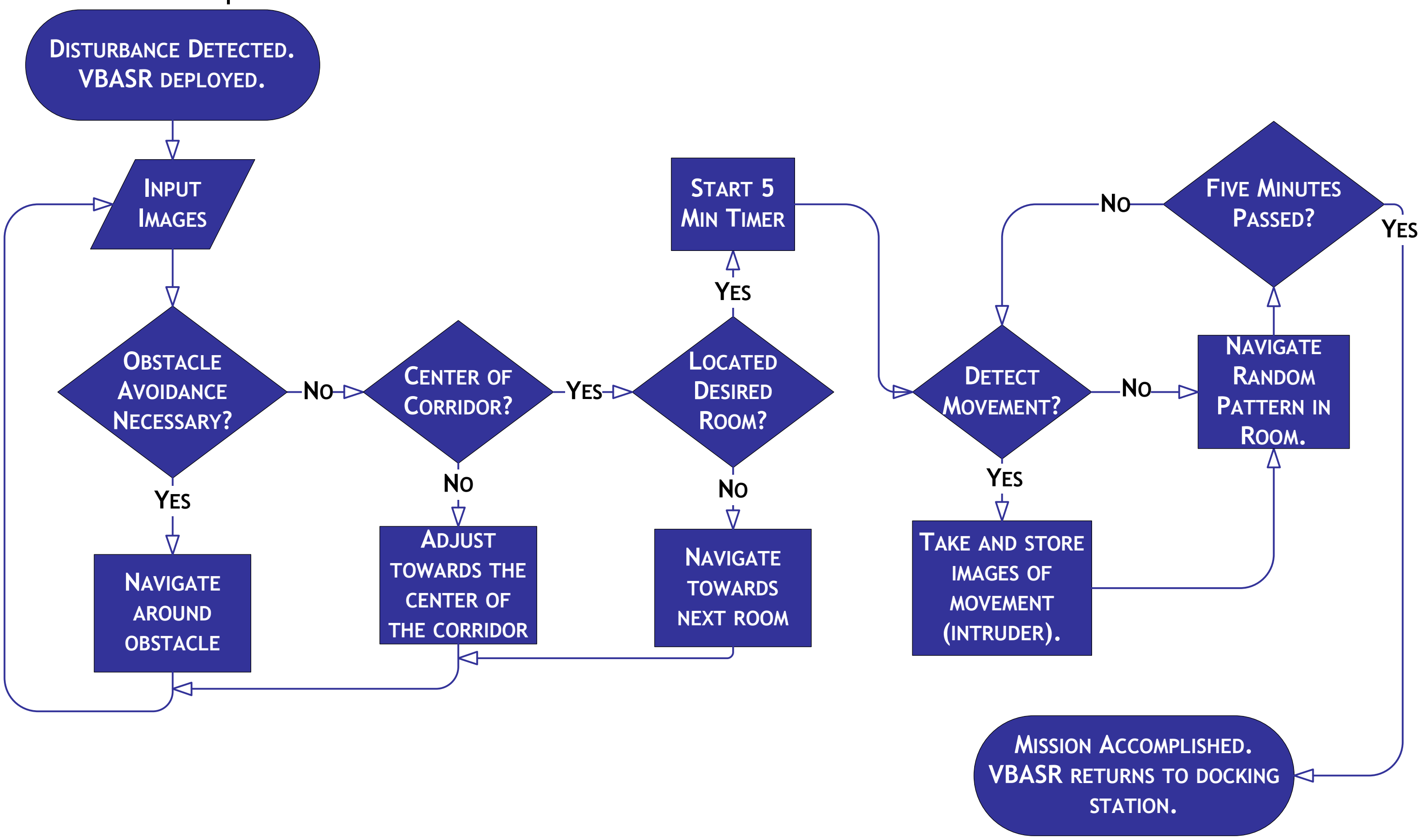

FIGURE 3 - SOFTWARE FLOWCHART

Logically, the software is simple. Using the information gathered from the images three main questions are continually asked:

- Does the VBASR need to avoid an obstacle?
- Is the VBASR in the center of the corridor?
- Has the VBASR located the correct room?

Adjustments will be made based on the answers of those questions.

Once the VBASR reaches and enters the specified room, it will run a random pattern looking for movement (intruders). If any movement is discovered within five minutes, incriminating photos will be taken and stored.

After five minutes the VBASR will return to its docking station and await a new disturbance. Five minutes is an arbitrary time limit selected to ensure that the VBASR has enough time to monitor the entire room.

SCHEDULE

Figure 4 details the projected schedule for the upcoming spring semester. From the beginning of the semester through spring break the schedule focuses on the primary goals of the project. This schedule leaves additional time for integration and debugging if necessary. If all goes smoothly, the time after spring break may also be used to implement the VBASR's secondary goals. Both sets of goals are described above in the Project Goals section.

Schedule of Milestones Spring Semseter		
Milestone	**Description**	**Expected**
9	Edge Detection	26-Jan-10
10	Blob Tracking	2-Feb-10
11	Obstacle Detection	9-Feb-10
12	Navigation Vectors	16-Feb-10
13	Navigation Algorithms	22-Feb-10
14	Real-Time	2-Mar-10
15	Real-Time	9-Mar-10
	Spring Break	16-Mar-10
16	Motion Detection	23-Mar-10
17	Motion Detection and Capture	30-Mar-10
18	Comparing and Recognizing Images	6-Apr-10
19	Mapping Hallway	13-Apr-10
20	Room Recognition	20-Apr-10
21	Complete Integration	27-Apr-10
	Wrap up - Conference papers	4-May-10
	Finals	11-May-10

FIGURE 4 – PROJECTED SCHEDULE SPRING SEMESTER - 2010

Figure 5 details the schedule for the fall semester. Currently all of the milestones have been completed except number 6. This will be completed over winter break. Most of the fall semester has been spent in preparation for the computer vision aspect of the VBASR. The focus of the fall semester was on implemented the various software and hardware together to ensure that the VBASR had a workable platform.

Schedule of Milestones Fall Semseter		
Milestone	Description	Expected
1	Order iRobot Create	15-Oct-09
2	Test iRobot Create Functionality	22-Oct-09
3	iRobot programming via Command Module	29-Oct-09
4	iRobot communication via MRDS	5-Nov-09
5	Webcam controls via MRDS	12-Nov-09
6	Webcam controls via OpenCV	19-Nov-09
	Thanksgiving	26-Nov-09
7	Project Presentations	3-Dec-09
8	Finals - Camera on order (if necessary)	10-Dec-09

FIGURE 5 – SCHEDULE FALL SEMESTER - 2009

LITERARATURE REVIEW

Computer vision has been explored in many different projects and papers, yet there are few standards for this area of research. Kingsly and Young discuss a stationary security camera. To this end, Kingsly and Young argue that image processing is traditionally concerned with pre-processing operations (Fourier filtering, edge detection, etc) [4]. To summarize their conclusions, computer vision is an extension of these ideas including understanding scene content, tracking and object classification. A good example of a security application for a fixed camera is identifying license plates. Another example presented in their paper is the use of fixed security cameras to detect intruders. A method of subtracting an undisturbed background image from a second image including an intruder enables identifying the intruder [4].

Andrea Cavallaro attempted to create a succinct computer vision project for undergraduate students [5]. Because the learning curve is steep for beginners in this field, Cavallaro attempted to create several simple yet insightful projects for students to design. OpenCV was used for each of these projects. Since OpenCV will be used for the VBASR, these projects will serve as a useful reference in the learning and implementing image the processing techniques necessary for this project’s success [5].

BIBLIOGRAPHY

[1] Shapiro, Linda G., Linda G. Shapiro, and George Stockman. Computer Vision. Upper Saddle River: Prentice Hall, 2001. Print

[2] "iRobot Create Premium Development Package." IRobot. IRobot Corporation, 2009. Web. 10 Nov. 2009. <http://store.irobot.com/product/index.jsp?productId=2591901&cp=2591511&parentPage=family>.

[3] Kosinski, Robert J. "Literature Review on Reaction Time." Clemson University, Aug. 2009. Web. 10 Nov. 2009. <http://biae.clemson.edu/bpc/bp/Lab/110/reaction.htm>.

[4] Sage, Kingsly, and Stewart Young. Security Applications of Computer Vision. Tech. IEEE AES Systems Magazine, 1999. IEEEXPLORE. Web. 20 Oct. 2009. <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=756080&isnumber=16393>

[5] Cavallaro, Andrea. Image Analysis and Computer Vision for Undergraduates. Tech. ICASSP, 2005. IEEEXPLORE. Web. 20 Oct. 2009. <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1416369&isnumber=30654>