Southern Polytechnic State University Autonomous Remote Reconnaissance System

Walter L. Burleson
James G. Hudak
Kristopher Keller
John F. Sweigart

Southern Polytechnic State University

ABSTRACT

This paper describes the status of the development of the Southern Polytechnic Autonomous Remote Reconnaissance System (SPARRS-2002) to be entered into the 2001-2004 International Aerial Robotics Competition. Strategies for the entire four-year competition are described briefly, and the efforts to meet the goal of this year's competion—namely, auotonomous flight—are described in detail. Our planned mission this year will be to fly autonomously the ~3 km run (level 1 behavior), as well as gather vision system data for use in later competitons and vehicle behavior levels.

COMPETITIVE STRATEGY

LONG TERM STRATEGY

As described in the 2001 paper[2], for the final competition SPSU intends to use a system consisting of two air vehicles and a ground station. The main air vehicle, Collegiate Robotic Autonomous Surveillance Helicopter (CRASH) is based on the commercially available Bergen Intrepid helicopter. The second air vehicle, Bantam Untethered Reconnaissance Navigator (BURN), will be transported to the target building by CRASH and launched into the structure to perform the required surveillance. While BURN is surveying the interior of the structure, its low-weight, low-power transmitter will send images to CRASH, hovering outside. CRASH will use its higher-powered transmitter to relay the images to the base station, taking advantage of line of sight. This overall strategy has not changed, except that BURN will be a dual mode projectile/ground vehicle instead of a second aircraft.

CURRENT YEAR STRATEGY

The primary goal is to complete the level 1 behaviors, which includes flying autonomously through the 3km, 3-way-point course outlined in the 2001-2004 competition rules (1).

MAIN AIR VEHICLE CONTROLS AND NAVIGATION

CONTROLS SUMMARY

The SPSU helicopter control system includes a host of navigation, sensors which provides vehicle position and attitude. This data is distributed to all of the software processes and all computers on the network. A cascaded PID control algorithm ultimately provides the output to the control surfaces. Each control surface is fed with the output of a cascaded set of PID equations. The variable inputs of these equations come from corrected or converted data from the following software "driver" processes: AHRS (attitude heading reference system), GPS (global positioning system), Sonar (ultrasonic altitude sensor), and Linear Accelerometers (new for 2002). The output from these collective equations is sent to high resolution, high speed controllers for the helicopter's servos which are also new for 2002.

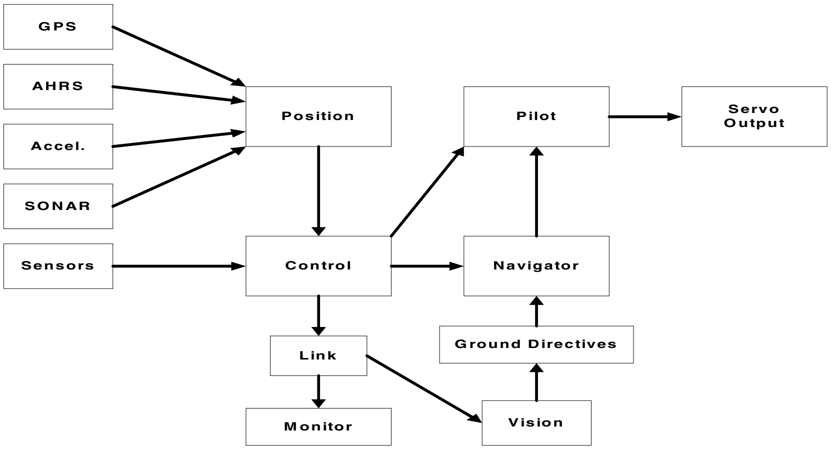

Figure 1: Revised Process Flow Diagram

(see 2001 paper[2] for more information on vehicle hardware and software)

NEW CONTROL AND NAVIGATION ITEMS FOR 2002

- Three-Axis Linear Accelerometer system installed
- Linear Position/Velocity Positional Filters added
- Control System Testing Methods changed
- On-The-Fly Gain Upload system installed
- High speed-high accuracy servo control output

THREE-AXIS LINEAR ACCELEROMETER SYSTEM ADDED

A notable improvement to the SPSU control system is the addition of a triple axis linear accelerometer. The accelerometer unit is small, lightweight and is mounted with vibration isolation on top of the AHRS. Much like the other sensors, the RS232 data is read and decoded in a separate "driver" process that passes the corrected information to the other software processes. The unit provides x, y, and z acceleration components of up to 4 G's per axis.

The accelerometer driver process has to do two important data calculation sets to provide the ideal resolved accelerations that the PID control loop inputs require. First, gravity (1 G) must be canceled. But, since the vehicle attitude angles are not always zero degrees, the attitude angle values from the AHRS must be used to properly compensate for gravity regardless of vehicle attitude. Secondly, the vehicle will almost never travel in an ideal straight line; therefore, calculations must be done to resolve the readings into the ideal north-south, east-west coordinate system (similar to what the GPS uses). Again, the AHRS attitude numbers are used to transform the vector quantities to a North, East and Up coordinate system.

The addition of the accelerometer data has many benefits over and above the AHRS/GPS-only system used in 2001. First, the GPS position and velocity data is at times inaccurate and of low bandwidth. The acceleration data is used to provide an approximate position and velocity between GPS readings. The data rate of the GPS is 5 Hertz maximum, while the linear accelerometer data is received at as much at 200 Hertz. Also, in the event that there is a temporary or permanent loss of GPS signal, the vehicle can safely navigate for a short time. In the event of total GPS loss, there will be sufficient time to switch to manual piloting mode before the accelerometer error accumulates to an unacceptable level.

LINEAR POSITIONAL FILTER ADDED

In previous years, this flight system relied on GPS only. The system also required that the GPS data be extremely reliable. Well, most days this just doesn't happen. GPS data will come in and out of 2cm lock, has a slow update rate and the Accelerometer data builds up too much error to be trusted after one second of calculations. The goal of the filter is to come up with an accurate position and velocity of the vehicle based on all of the collected data. This position filter has several parts.

The first part developed was a history buffer. This buffer stores 10 previous positions. Every time a new GPS value is received, the history buffer is updated with a new calculated position based on the GPS data. When the final position is calculated and sent to the flight systems, a **weighted average** is performed on the history buffer. The following table shows an example of the weighted average.

Position	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Average**
Meters	5	5	3	4	5	2	1	2	1	1	3.64
Weight	10	9	8	7	6	5	4	3	2	1	

Table 1: Weighted Average

Before the weighted average was incorporated, a simple mean average was used. This was proven to be useless since it produced a 2 second lag in the control surfaces.

Another part of the filter is a means of predicting what the next position point should be in case the GPS signal quality is poor. GPS signal quality is based upon the standard deviation provided with each log. The method of position prediction we decided to use was **Linear Regression** of the history buffer. When using predicted data, the change in position calculated from accelerometer data is added to prevent a constant slope from occurring. Below is a graph that shows the results of linear regression used for prediction.

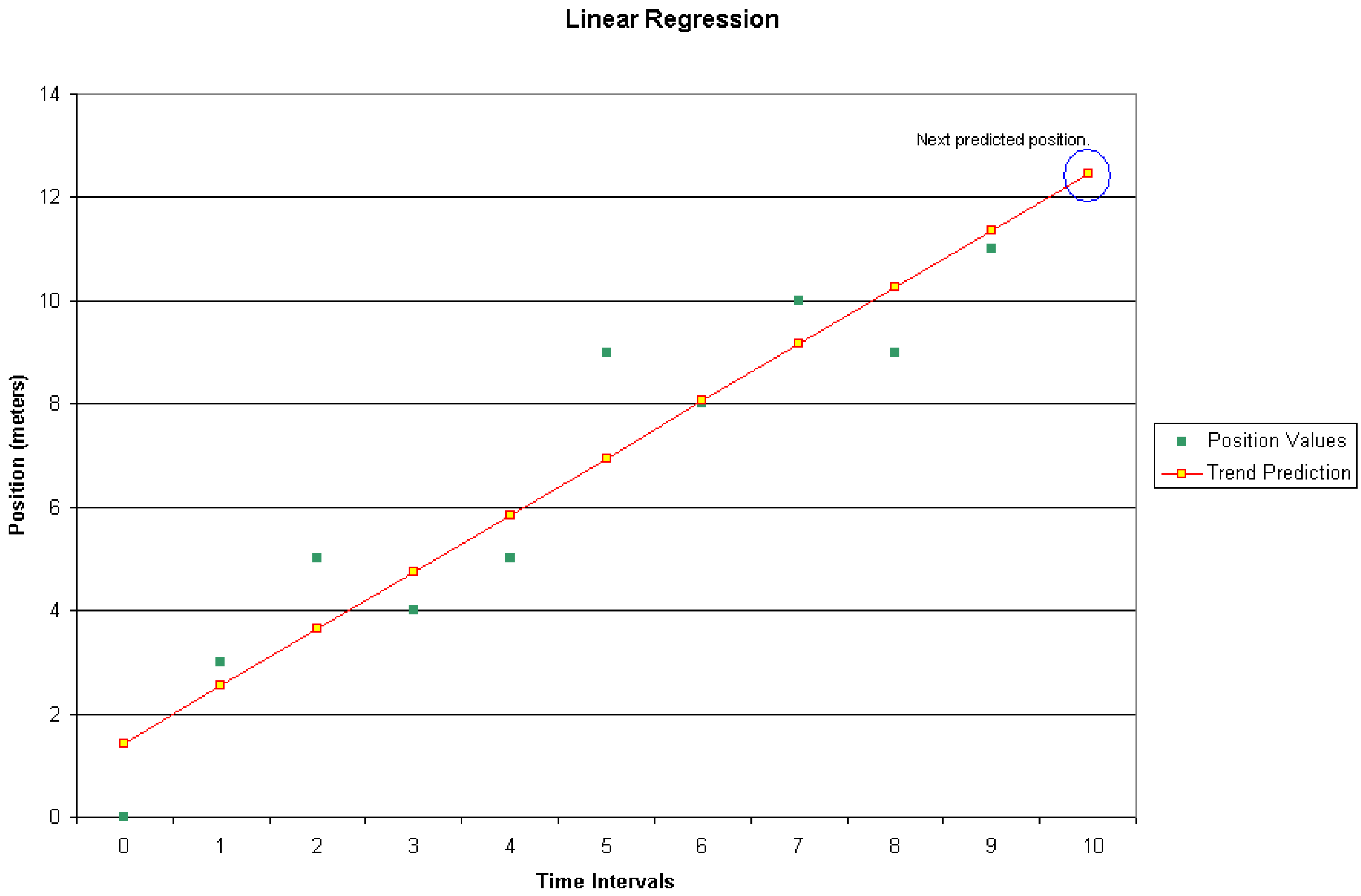

Figure 2: Example of Linear Regression

A third part of the filter is to determine whether or not the GPS data can be trusted and at what level. Again GPS signal quality is based upon the standard deviation that is provided from each log received. If the GPS data is completely trusted it is placed in the history buffer. If the GPS data is completely not trusted then the predicted value from the linear regression is placed into the history buffer. And if the GPS data is somewhat trusted then a portion of the GPS data is added to a portion of the predicted data and the summation is placed into the history buffer. The following equation shows how a position is calculated when the GPS data is in the gray area.

Position = GPS*%trust +PREDICTION*(1-%trust)

Standard Deviation	**%Trusted**
Less than 0.1 meters	100%
Between 0.1 and 0.2 meters	75%
Between 0.2 and 1 meters	50%
Between 1 and 3 meters	25%
Greater than 3 meters	0%

Table 2: GPS Weights

The final part of the filter is calculating position and velocity data, based upon accelerations. Each accelerometer message received is integrated to determine the vehicle velocity. The calculated velocity is integrated to determine the vehicle position. These values are added to the calculated values based upon GPS data. One thing to consider is that the position and velocity calculations will eventually build up error, and in order to prevent this from happening, each time a GPS message is received the accelerometer calculations are reset to zero.

CONTROL TESTING METHODS REVISED

The testing and tuning methods have undergone vast improvement for 2002. The fact that the center of gravity (c.g.) of the vehicle is un-obstructed has proven useful in running angular-only experiments. A "test stand" (as it is called) was constructed in a way that allows both the pitch (x) and roll (y) axes to be tested on the helicopter in complete safety. The rotation axis, which in this case is a steel tube, passes directly through the c.g. of the vehicle. In fact the vehicle with spinning rotors can do full 360 degree rotations on the stand without any interference. This test stand method reduces the chances of damage to the vehicle as well as enhancing safety during the quite sensitive part of control tuning. See Figure 3 below:

Armed with starting point gain numbers provided by software simulation [2], the vehicle was subjected to test sets with all linear controls set to null. Tests such as zero holding, offset recovery, and collective load changes were executed. The gain numbers for the axis under testing were adjusted "on-the-fly" to provide the optimal control stability under various expected conditions. A full 90 degree error could be corrected in less than 2 seconds without visible overshoot or oscillation. Before the "test stand" was developed, the angular AND linear had to be tested and tuned simultaneously, which proved to be a daunting task. The test stand not only greatly improved speed in tuning, and made robust control an attainable goal. Using safety poles to minimize the chances of tipping over ,the helicopter over was removed from the test stand and allowed to angularly stabilize itself while the pilot controlled the other axis. As predicted, the attitude was in full control, meaning that the x and y attitude angles were held at near zero even when un-tethered. Note that, with all linear control disabled, the helicopter was free to accelerate to the left due to the thrust of the tail rotor to the right.

After both pitch (x) and roll (y) were successfully tested and tuned, then the yaw (w) axis was tested on yet another test fixture. The vehicle was strapped to a ball raced table with the rotation axis in line with the center of mass of the yaw axis. This setup allowed the complete testing of the yaw axis controls without risk of damage or injury. On the fly tuning allowed a steadily controlled tail rotor which allowed 180 degree + corrections without overshoot or oscillation even while compensating for main rotor torque. After satisfactory performance was attained, the yaw axis was left to computer PID control and the other axes under human pilot control. A set of dynamic flight tests including forwards, backwards and sideways flight were successfully executed. The yaw control axis continues to perform beyond expectations.

ON-THE-FLY GAIN UPLOAD CAPABILITY ADDED

As stated in the above paragraphs, on-the-fly gain value adjustments are possible. Using a new scheme for data-uplink, a system for uploading new control loop values is installed in the system software. After the desired PID gains are found, they are saved for future use. Note: For most of the cases in the SPSU control system, the gains derived from simulation have proven to be reasonable compared to physical tests, if not in magnitude, then in proportion.

IMPROVED SERVO CONTROL OUTPUT/INPUT

Also, new for the 2002 control system is a new high-resolution and high-speed servo output/input system. Prior to 2002, the update rate for each control servo was limited to 10 Hertz, severely reducing theoretical control performance. In addition to the bandwidth limitation, the maximum resolution was limited to 255 positions from full positive to full negative (or, full left to full right).

U-Nav Incorporated, produces a high-performance servo control circuit. Two of these controllers are installed in the 2002 vehicle. These units allow for 50 Hertz update rates (the maximum for an RC-type servo), and have a maximum of one thousand steps over the full travel. The improvement in update alone greatly increased the control of the SPSU helicopter. The improvement in resolution increased smoothness of control, reduced workload on each servo, and eliminated oscillations seen in the low-resolution system.

In addition to the high-performance output capability, these boards also allow for real time input of the commands read from the safety pilot’s transmitter. This capability has many uses including: reading actual trim positions after test flights, more accurate readings of maximum and minimum values for each servo axis, and allowing logging of pilot control during test flights (for control output comparison). For complete accuracy in mechanical/software settings it is important to match the software’s high, center and low limits for each servo. Inaccurate settings for any servo causes a mismatched response compared to the mechanical setup of the helicopter.

VISION SYSTEM DETAILS

TARGET IDENTIFICATION TECHNIQUES

Presently, our Target Identification Software has three modes of operation as were briefly described in our previous report. Two modes are fairly well developed and are described here: **Identify Symbol** and **Locate Window**. These modes are designed to allow the helicopter software to control the activity of the Vision Software. **Identify Symbol** locates dark object blobs, preprocess the blob and sends it's outline pattern through a neural net for identification. **Locate Window** searches for dark blobs that comprise four corners and sides that are parallel and perpendicular to the horizon. Both these modes use the same preprocessing and blob detection; however, they diverge in methodology on how to process the blob.

IMAGE PRE-PROCESSING

Once the image-capture card receives the 320 by 240 image frame, the image is passed through a color histogram that displays light level peaks for the entire frame. This aids in setting brightness, contrast, and threshold levels. Next, the color image is converted to gray scale image. Next, all image filtering and threshold contrasting are used to remove image noise and highlight dark objects.

Blob Detection creates two histograms for locating blobs. Blobs are large dark objects that may be the symbol that we are looking for. An X histogram is created which plots the total number of pixels over the set blob Threshold for each pixel column. Likewise, a Y histogram is created which plots the total of pixels over a set blob Threshold for each pixel row. The peaks of the two histogram are use as x and y coordinate witch are checked to confirm blob coordinate of target blobs. When the look back loop confirms a grouping of dark pixels, blob pixel location is placed into a list for all blobs that were found in the current image. The software passes the list to the mode-dependent software for symbol identification or window location.

SYMBOL IDENTIFICATION MODE

When the helicopter commands Symbol Identification Mode, the vision software would perform the following operations on each blob in the blob list. Eight fast edge detection operations are performed to determine approximate center of the object. One detection for every 45 Degrees will provide a better estimate of the blob centroid, and provide height and width information. All further analysis is done on a region of the image which is just large enough to hold the blob. Each blob detected has this region call the Testing Window. This saves computer operations and reduces interference from other objects in the full image.

Next, the Test Window image is scanned with a Sobel edge detection scheme [6]. All dark pixels have their edge angles recorded to create a histogram of angles from 0 to 359 degrees vs. the number of occurrences of the angle. Peaks in this histogram identify straight lines and their angles of the Target blob. Since the neural net operates only angular information, the Symbol Size will have minimal effect on Symbol Identification.

To reduce neural net training the first step in analyzing the histogram is to shift it (left-right) so the highest peak is at 0 degrees. This removes the effect of orientation of the target.

This Four-Layer Neural Net takes 360 inputs from the Edge Angle Histogram. Our neural net has two intermediate layers of reduction with 10 nodes each. The final Output represents up to six objects that it is trained for. These outputs generate matching real numbers that range 0 to 1, where one represents perfects match for that symbol. After the neural net output numbers are retrieved, the highest output value over .6 identifies the target shape. If no output is greater than .6 the blob is not identified.

The Neural Net is designed with Qnet 2000 neural net development software by Vesta Services Inc. Qnet allows the development of a custom fit neural net base, the input of external training files, the training, the testing of the neural net, and the export of a trained neural net to a DLL file.

WINDOWS SEARCH AND TARGET MODE

When the helicopter control process commands **Window Search Mode**, The vision software would perform the following operation on each blob in the blob list.

As in Symbol Identification, fast edge searches are performed along eight rays radiating out from the center of the blob. The edges detected by Sobel Edge Detection determine the approximate dimension of the blob.

Next, another Sobel edge detection is performed to determine the slope of the edge of the blob at each point where a ray crosses it. The slope of the edge found at the intersection is plotted in memory against the ray angle. MV-3 searches the plot for slope changes that indicate corners of the blob. If the object has 4 corners, the angles of the four sides are checked within the plot to see if they are parallel and perpendicular to the horizon. If so, the object is treated as a window or door of a building.

Once an object is identified as a window or door, Sonar readings are used to measure the distance to the wall containing the window or door. Since pre-flight calibration creates a linear mathematical model of object distance vs. pixel size, the sonar measurements allow real height, width, and position to be calculated. This information will be sent to the helicopter to adjust targeting for launching a small robot through the window or door.

Pre-flight calibration uses a dark square calibration target for sonar to range against for distance. Likewise, vision software reads the number of pixels horizontal and vertical size (in pixels) of the square calibration target. Both a short range and long range readings are taken to create the linear model of target distance vs. pixel width and height in meters. Lens focal length can not be change at this point without corrupting the calibration.

SUBVEHICLE

OVERVIEW

The sub vehicle and its launching system have been designed around these general guidelines:

1) The sub-vehicle is to be less than three (3) inches in diameter, less than twelve (12) inches in length and weigh less than one pound.
2) The vehicle must be launched from the main helicopter (CRASH) through a window no less than twenty feet away. The window is to be one meter square and hopefully open.
3) The launching system is not to use any explosives or flammable propellants and is to weigh less than one pound.
4) Upon vehicle entry into the building the sub vehicle is to become active and acquire as many pictures of the objects in the room as possible. The pictures must be relayed back to CRASH to be sent to the ground station.

SYSTEM DESIGN

A sub-vehicle consisting of two main systems has been adopted, the sub (BURN) is being jointly designed by the ART and SPSU'S GEAR team (General Engineering of Autonomous Robots). The GEAR team is highly experienced in building the complex miniaturized systems that are required to successfully build a micro autonomous vehicle. The second part of the system is the launching system. The launcher is designed exclusively by the SPSU ART.

SUB-VEHICLE LAUNCHER

In order to fulfill the requirements put forth in the current rules our launcher has been designed to run on compressed air. The compressed air is held in a steel tank pressurized by a standard tire valve. In order to nearly instantaneously release the compressed air SPSU students have modified the internal workings of a standard lawn sprinkler valve. The sprinkler valve being designed for water originally was found to be too slow when used with high pressure gas. Using the original valve our test object didn't travel more than a few feet and it was apparent that the valve was not staying open through the duration of the launch. After the valve had been redesigned it was capable of staying open for the entire launch resulting in an estimated power increase of almost 66%. The sprinkler valve is triggered by the standard electromagnetic valve and there is a redesign of the valve that will eliminate the electromagnetic trigger and replace it with a smaller and lighter mechanical valve.

BURN (The sub-vehicle)

While designing BURN many designs were prototyped for testing and it was established that the lightest most durable systems perform best. A design limit of three inches in diameter was established but in order to tackle more challenging terrain the wheels needed to be larger than three inches. After discussing this issue with the GEAR team we came up with a design for wheels that would expand.

While in the closed position the wheels will be slightly less than three inches in diameter and when the drive motors are activated the wheels will expand to be much larger. The wheels are made from machined aluminum and have been designed to offer some shock absorption while isolating the drive servos from the shock of a high speed impact. In order to minimize points of failure the wheels have been designed to only expand after the drive motors have been activated.

All on board computing is provided by a basic stamp and when coupled with a series of optical sensors is capable of providing navigation and obstacle avoidance. In addition the stamp will handle all motor control. . In order to capture images of the room a color CMOS camera has been chosen. They offer low current operation further reducing weight and are very durable. All of the pictures will be sent on the 900Mhz band to a receiver on CRASH where a more powerful transmitter will relay them to the base station monitor. BURN is designed to operate for a short time (less than fifteen minutes) allowing it a lower weight and by shutting itself down allows us to launch other vehicles using the same radio frequency

BURN is a low cost and low weight vehicle, which is designed to be expendable. The low weight of both BURN and its launcher provide the opportunity to carry multiple vehicles to further increase the chances of obtaining useful visual information.

ACKNOWLEDGMENTS

The Southern Polytechnic Aerial Robotics Team extends its appreciation to SPSU's Alumni Association, Foundation, and Student Government Association for financial support. We thank Watson Industries and Novatel for AHRS and GPS equipment loans, respectively. Many thanks also to Flash Corliss, President of Flash Automation, for his donation of computer equipment and technical expertise. Flash is an SPSU alumnus as well as a team member from earlier IARC competitions. Finally, we also appreciate the support of our team's advisor, Professor John Sweigart.

REFERENCES

1. AUVS, *Rules for the AD2002 International Aerial Robotics Competition*
    a. *Qualifier*, http://avdil.gtri.gatech.edu/AUVSI/IARCLaunchPoint.html

2. Burleson, Walter L., D. denHaring, J.G. Hudak, J.F Sweigart, "Southern Polytechnic State University Autonomous Remote Recconasaince System", Proceedings, AUVS, Orlando, FL, 2001

3. Kannan, *et al, Control Algorithm and Flight Simulation Intergration Using the Open Control Platform for Unmanned Aerial Vehicles*.

4. Johnson, Curtis D., *Process Control Instrumentation Technology*, Prentice Hall, 1997.

5. Wheeler, Anthony J and Ganji, Ahmad R, Introduction to Engineering Experimentation, Prentice Hall, 1996.

6. Umbaugh, Scott E., *Computer Vision and Image Processing*, First Edition, 1997

7. E. R. Davies *Machine Vision Theory Algorithms Practicalities, Second Edition, 1997*