AUTOMATED VISUAL DATABASE CREATION FOR A GROUND VEHICLE SIMULATOR

By

PEDRO J. CLAUDIO
B.S.Ch.E., University of Florida 1981
M.S.E., University of Central Florida 1986

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the School of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2006

Major Professor: Christian Bauer

ABSTRACT

This research focuses on extracting road models from stereo video sequences taken from a moving vehicle. The proposed method combines color histogram based segmentation, active contours (snakes) and morphological processing to extract road boundary coordinates for conversion into Matlab™ or Multigen OpenFlight™ compatible polygonal representations.

Color segmentation uses an initial truth frame to develop a color probability density function (PDF) of the road versus the terrain. Subsequent frames are segmented using a Maximum Apostiori Probability (MAP) criteria and the resulting templates are used to update the PDFs. Color segmentation worked well where there was minimal shadowing and occlusion by other cars.

A snake algorithm was used to find the road edges which were converted to 3D coordinates using stereo disparity and vehicle position information. The resulting 3D road models were accurate to within 1 meter.

Dedicated to Alexander, Katherine, Christina and Nathaniel Claudio

ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisor, Dr. Christian Bauer, for his support, patience, and encouragement throughout the long years of my graduate studies.

I also thank my son, Alexander and his friend Anthony Justiniano for assisting me in constructing the hardware and in data collection.

I would also like to thank my wife Vicki for her editorial assistance in assembling this dissertation and for her love and support.

Special thanks also go to my fellow students who supported me in my endeavors, Dahai Guo, Bobby Harris, Lijian Xu, and Vaibhav Joshi.

I would also like to express my appreciation to the Link Foundation for awarding me an Advanced Simulation and Training Fellowship in 2000-2001. This work also received financial support from Lockheed Martin with a Synthetic Environment Learning Laboratory (SELL) equipment grant.

Thanks to Dr. Alberto Broggi from the Artificial Vision and Intelligent Systems Lab (VisLab), University of Parma, Italy for permission to use images from the GOLD website. Thanks to Drs. Karl Kluge, Sridhar Lakshmanan and Christopher Kreucher for permission to reprint images from "Tracking Lane and Pavement Edges Using Deformable Templates" in Proceedings of SPIE Vol. 3364 "Enhanced and Synthetic Vision 1998"

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
4D	Four Dimensional
GB	Gigabyte
ALV	Autonomous Land Vehicle
ALVINN	Autonomous Land Vehicle in Neural Network
API	Application Program Interface
AVI	Audio Video Interleaved
AVS	AVI Synth Script
CCD	Charge Coupled Device
CCTV	Closed Circuit Television
CMU	Carnegie Mellon University
COC	Circle of Confusion
CP	Color Predicate
CRT	Cathode Ray Tube
DAS	Detector Angular Subtense
DGPS	Differential Global Positioning System
DLG	Digital Line Graph
DOD	Department of Defense
DOQ	Digitally Orthorectified Quadrangle
EMS	Expectation Based Multifocal Saccadic
FCS	Florida Coordinate System
FLIR	Forward Looking Infrared
FOV	Field of View
GB	Gigabyte
GOLD	Generic Obstacle Lane Detection
GPS	Global Positioning System
HDRC	High Dynamic Range CMOS
HMM	Hidden Markov Model
HMMWV	High Mobility Multipurpose Wheeled Vehicle
HSI	Hue Saturation Intensity
HW	Hardware
IEEE	Institute for Eletrical and Electronics Engineers
INS	Inertial Navigation System
IP	Internet Protocol
IPM	Inverse Perspective Match
ISA	Industry Standard Architecture
JPL	Jet Propulsion Laboratory
LOIS	Likelihood of Image Shape
MAP	Maximum Apostiori Probability
MB	Megabyte

MMX	Multimedia Extensions
MOBLAB	Mobile Laboratory
MPH	Miles Per Hour
MTF	Modulation Transfer Function
NEMA	National Electrical Manufacturers Association
NN	Neural Network
OTF	Optical Transfer Function
PAPRICA	PArallel PRocessor for Image Checking and Analysis
PC	Personal Computer
PCI	Peripheral Computer Interconnect
PCMCIA	Personal Computer Memory Card International Association
PDF	Probability Distribution Function
PSD	Power Spectral Density
RAID	Redundant Array of Independent Disks
RALPH	Rapidly Adaptive Lateral Position Handler
RF	Radio Frequency
RGB	Red, Green, Blue
RMS	Root Mean Square
RTCM	Radio Technical Committee, Marine
SA	Selective Availability
SCARF	Supervised Classification Applied to Road Following
SCSI	Small Computer Systems Interface
SSD	Sum of Square Differences
SW	Software
TCP	Transmission Control Protocol
TTL	Transistor Transistor Logic
UCF	University of Central Florida
UTM	Universal Transverse Mercator
VITS	Vision Task Sequencer
VSS	Vehicle Speed Sensor
WGS	World Geodetic System
XUV	Experimental Unmanned Vehicle
YARF	Yet Another Road Follower
YUV	Luminance, Blue Chrominance, Red Chrominance

1 INTRODUCTION

1.1 The UCF Driver Simulator

The University of Central Florida's (UCF) College of Engineering has been using driving simulators for research over the last 10 years. Their first driving simulator (1996) consisted of a full size station wagon shell, multi-channel audio system and a Silicon Graphics ONYX™ computer with a single Reality Engine II (RE2) graphics pipeline driving three Barco color video projectors and a 170° by 70° circular display screen (see Figure 1). Following the retirement of this system, a commercial motion platform based system was purchased. The new system allows for rapid interchange of standard vehicle cabs. It uses Image Generators based on a cluster of PC's with high power graphic generator cards.

- Side View -

- Top View-

Figure 1 – UCF Driving Simulator

1.2 Road Model Generation for the UCF Driving Simulator

The first database developed specifically for the UCF driving simulator represents a two and a half mile portion of the UCF campus road network. Student assistants using traditional manual methods with the Multigen™ database generation toolset developed it. Building models consist of simple polyhedral models with textures manually applied. Roads were developed using Multigen RoadTools™ . Approximately three man-years were required to complete the first generation database.

A great amount of effort was expended in correlating this database with the actual terrain. It was necessary to constantly compare the generated image from the driver's eye point with a video sequence recorded from a moving vehicle representing the same viewpoint. Properly placing buildings to appear in the same place and matching the road geometry required numerous iterations.

Given the level of effort required for generating a short segment of the UCF road network, it's apparent that using these traditional techniques for developing a large road network are impractical due to the large labor effort requirements.

1.3 Other Techniques for Generating Road Models

There are various commercial tools for creating models of buildings and other above ground structures. Photomodeler™ [1] allows the user to extract 3D textured models from photographs. Typically, sets of photographs with different perspective views of an object are taken. Corresponding feature points between the photos are manually selected by the user; the software then uses photogrammetric algorithms to extract the 3D geometry and texture. The resulting model can be exported to other programs in a variety of formats.

Using Photomodeler™, it would be possible to obtain several hundred stereo image pairs of a roadway and generate hundreds of 3D models of road segments. The 3D models could then spliced together with a tool such as Multigen Creator™ and textured. While this provides an accurate road path, it increases the manual modeling effort significantly.

Another approach would to be take aerial photographs and use a road extraction algorithm such as [2]. Unfortunately, aerial photography can cost hundreds of dollars per hour if the desired roads have not been photographed before or if the existing data is outdated.

Skipping image data altogether, [3] takes Global Positioning System (GPS) measurements of the road lane boundaries using dual GPS receivers mounted on a horizontal boom. Lacking any vision processing the road edges must be extrapolated from the vehicles trajectory.

1.4 Statement of the Problem

Our goal was to model the road using stereo video cameras. We planned to collect video and position data using a specially equipped station wagon, and post-process it in a semi supervised fashion to extract sections of the campus road network. The extracted 3D information can then be used to create an accurate georeferenced road with respect to buildings in the visual database.

1.5 Overview

Our literature review in Chapter 2 focuses on previous systems that have used vision for modeling the road environment. Detailed discussions of underlying algorithms incorporated into our proposed method are put off to Chapter 3.

Chapter 3 begins with a systems analysis done to estimate the expected 3D accuracy of the extracted road boundaries. Next we describe the hardware and software used to collect our sensor data and then we present our proposed method for extracting 3D road models from the collected sensor data.

Chapter 4 describes our results in terms of obtaining 3D models for selected portions of the UCF campus road network and comparing them against orthorectified aerial photographs of the campus. Finally, Chapter 5 contains our conclusions, recommendations for improvements in our methodology, and suggestions for further research.

2 LITERATURE REVIEW

Since our task was to model roads using stereo video, our literature review was focused in the area of autonomous vehicular navigation for techniques used to recognize roads in ground level video. Working in an outdoor environment also required overcoming additional problems such as:

- Lack of Controlled Lighting
- Mixture of Natural and man-made features
- Occlusions due to other vehicles
- Weather Effects

We will give a brief historical summary of different efforts over the years and then discuss various algorithms and platforms used for navigating over structured and unstructured environments.

2.1 Overview/History

2.1.1 ALV

In the mid 80's the U.S. Department of Defense (DOD) financed several Autonomous Land Vehicle (ALV) projects with the goal of cross-country driving. Two of these efforts were the Alvin [4] by Lockheed Martin and the Terregator [5] by Carnegie Mellon University (CMU).

The Terregator was a six wheeled autonomous vehicle containing a single color television camera, a laser range sensor and an odometer. Remote hosts connected via RF links performed the vision processing and navigation. The initial efforts were to navigate a sidewalk on the CMU Campus given a plan of the routes available. Three techniques

were developed to detect the sidewalk, Edge Detection in the Image Gradient, Oriented Windows and Color Segmentation.

Edge detection attempted to find the sidewalk by thresholding the image gradient, edge linking the resultant points and then searching for a pair of lines that met certain criteria (i.e. width, parallelism). This method had problems with low-contrast and jagged edges. This led to the Oriented Windows technique which would place a window in the detected image over the predicted road edge and rotate the window until the gradient of the column sums were maximized. This occurs when the window is oriented perpendicular to the sidewalk edge. Finally, a color segmentation technique was used to separate the image into sidewalk and terrain regions. Red, Green, Blue (RGB) color values normalized by the intensity were compared to typical values for sidewalk, grass, trees, etc. The sidewalk edges were found by fitting lines to the region boundaries.

The Terregator would navigate by using its estimated position to predict the sidewalk edges, passing the edges to the vision system for comparison to the extracted road edges and using the error update vehicle position estimate. Navigation software would then generate corrective commands in order follow the pre-planned path.

Alvin was an eight wheeled diesel powered all terrain vehicle with a fiberglass shell. Alvin's mission was to follow a road from point A to B while avoiding obstacles. Its sensors consisted of an odometer coupled to an Inertial Navigation System (INS), a color video camera mounted on a pan/tilt mechanism and a laser range scanner. Alvin's processing hardware was totally self contained and included an Image Processing subsystem capable of video rate operations such as 3x3 convolution and thresholding.

Alvin's Vision Processing System used a flat earth model to extract the 3D coordinates of the road boundaries from the detected 2D boundaries in the cameras image. It then passed the 3D road boundary coordinates to navigation system so it could point the camera towards the center of the road, this kept the road boundaries in view even around turns.

In 1987 [6] Alvin became the first robotic vehicle intentionally driven off road. It relied principally on the Laser Range Finder for detecting obstacles, its onboard hardware was not sufficient for stereo disparity processing in real time.

2.1.2 Demo I/II/III

After termination of the ALV program in 1989 the Army funded a series of efforts with progressively more ambitious aims known as Demo I/II/III [6]. Demo I (1990-92) used a stereo disparity algorithm developed by the Jet Propulsion Laboraty (JPL) which ran at 0.5 Hz over a 64x60 image, its main focus was teleoperation. The range data was used to emphasize obstacles in the operators image. Demo II (1992-98) vehicles were HMMWVs which used stereo black and white cameras for ranging and a panning color camera for road following. The Autonomous Land Vehicle in a Neural Network (ALVINN) neural based road follower [7] was used to pan the color camera when navigating on roads. Demo II was able to follow structured and unstructured roads in good weather.

The Demo III program (1998-2004) uses Experimental Unmanned Vehicles (XUVs) equipped with a sensor suite of fixed stereo color cameras, fixed stereo FLIR cameras and a gimbal mounted Laser Range Finder. At a demonstration in Ft. Indiantown Gap, Pennsylvania in 2002, the XUV's were able to navigate over dirt roads, trails and woods. Positive (rocks, trees) and negative (ditches) obstacles were detected and avoided. Obstacle detection relied upon the Laser Range Finder without the use of any stereo disparity information from the video cameras.

2.1.3 Navlab/CMU

CMU followed on its ALV[5] efforts by building the Navlab[8] , a modified Chevrolet Panel Van in 1986. The Navlab overcame the primary limitation of the Terregator, it had enough room to carry several racks of equipment This vehicle served as the platform for a multitude of efforts for following structured and unstructured roads. The algorithms developed, such as the Supervised Classification Applied to Road Following (SCARF) algorithm [9] for following unstructured roads and ALVINN, found application in the Demo II and III program. Another road following algorithm RALPH[10] was used to develop a lane tracking system for warning drivers of vehicle weave or drift.

2.1.4 Vamors/Vamp

Simultaneously with the DOD ALV (1980's) efforts the Europeans began an initiative to develop vehicles capable of autonomously navigating well structured roads like freeways. The Vamors[11], a commercial 5 ton van, was used for the initial development efforts using a single monochrome camera.

As opposed to US efforts the Europeans sought low level vision processing operations that operated in sub windows of the video stream. The sub windows were localized through a non-real time initialization process to lie on “interesting” areas of the road image, such the edges and horizon. They did this in order to implement the algorithms in hardware so as to process the image sensor data at video rates. This permitted them to approach the problem from a control engineering perspective (20 Hz video measurements .vs. 0.5 Hz seconds in ALV) .

Dickmanns [12] developed the 4D approach which would compare the 2D projection of the internal 3D road model to the detected edges and other features of the current road image and would use the deviations to modify the computed vehicle location. A Kalman filter fused the corrections with INS sensors.

The European efforts have continued to the present with extensions of the 4D approach to stereo cameras; the Expectation Based Multifocal Saccadic (EMS) method [13]. A Mercedes 500 SEL test vehicle known as the Vamp is used with a suite of sensors. These sensors include, stereo wide field of view (FOV) black and white cameras, a single narrow FOV color camera, a high resolution narrow FOV color camera all mounted on a pan/tilt platform and a bumper mounted radar. The Saccadic in EMS is a reference to how the human visual system is constantly scanning its environment in order to build an internal model of the surroundings. The wide FOV cameras are used to obtain road edge segments from sub-windows and the narrow FOV cameras focus on the road up ahead for vehicle detection and tracking.

2.1.5 ARGO/MOBLAB

Starting in 1993 the University of Parma developed the Mobile Laboratory (MOBLAB) [14] vehicle for the European PROMETHEUS project. The initial efforts were to develop an inverse perspective method [15] hosted on a real time massively parallel processor known as the PAPRICA, which stood for Parallel Processor for Image Checking and Analysis [16], that would convert the single black and white cameras road image to an overhead view . Road markings in the remapped image were detected by using a morphological operator to compare the local pixel values to its left and right neighbors. Thus, road markings could be detected even in the presence of shadows.

The PROMETHEUS project ended in late 1994, the university continued its work with the acquisition of a Lancia Therma Passenger car which was converted for fully automatic driving [17]. This vehicle was dubbed the Argo after the mythical many eyed creature. The Generic Obstacle Lane Detection (GOLD) [18] system which makes use of the Inverse Perspective Match (IPM) method on stereo video to perform lane and obstacle detection was developed for the Argo. Argo was demonstrated in 1998 in the MilleMiglia in Automatico Tour. During this test, ARGO drove itself autonomously along the Italian highway network, passing through flat areas, mountains including high bridges and tunnels. The system was able to cover 2000km of road, 95% of the mileage was driven autonomously. The original PAPRICA hardware has been replaced with PC's based on the Pentium MMX.

For further historical information on autonomous vehicle development see [6, 19]; the latter reference in particular has descriptions of Japanese work in this field.

2.2 Road Modellers

Some of the algorithms used for navigating structured (lane markings), unstructured (dirt roads) and terrain will now be presented.

2.2.1 Unstructured Roads

2.2.1.1 VITS

The Vision Task Sequencer (VITS) [4] was implemented on the autonomous land vehicle Alvin. VITS builds symbolic descriptions of road and obstacle boundaries using both video and range sensors.

VITS generates a scene description consisting of left and right road edge points and locations of potential obstacles. The road edge points are obtained by color segmentation, obstacles are located with the laser range scanner.

The VITS color segmentation algorithm maps the RGB pixel values into the blue-red plane. It then finds a linear discriminant function that separates the road pixel clusters from the terrain. A coordinate transformation is done that rotates the discriminant function so that it is horizontal, thus becoming a fixed threshold that separates the remapped pixel colors into road and non-road.

This dynamic thresholding approach was not used in practice on the Alvin, typical values for the pixel remapping for the test track for were:

$$R' = \begin{bmatrix} -0.5 & 0 & -0.5 \end{bmatrix} \begin{bmatrix} R & G & B \end{bmatrix}^T \quad (1)$$

, it was convenient to simply subtract the red color component from the blue and use a fixed threshold. This became known as the “Blue minus Red” segmenter.

The single threshold approach had problems handling shaded regions of the road. A "shadow box" technique was developed in which the red-blue plane was separated into two regions for the sunlit and shadowed pixels respectively. Separate linear discriminant functions were computed for the each region.

2.2.1.2 SCARF

The Navlab served as the development platform for SCARF [9] (Supervised Classification Applied to Road Following). SCARF can adaptively segment the roadway from its surroundings using a multi-class color classifier. The method is seeded by manually selecting a road and off road sections. The pixel colors are then clustered using a nearest mean clustering method. This is done by first arbitrarily assigning pixels to one of a predetermined number of classes. The class means are computed and then each pixel is checked to see if they are closer to another class. If a pixel is closer to another class it is switched and the class means recomputed again. Eventually no pixels have to be switched and the process ends. It has been empirically determined that four color classes suffice to represent road pixels and an additional four for non-road pixels. The models are computed as:

$$\begin{aligned} \mathbf{m}_i &= \frac{1}{N_i}\sum_{x\in W_i}\mathbf{x}_i \\ C_i &= \frac{1}{N_i}\sum_{x\in W_i}\mathbf{x}_i\mathbf{x}_i^T - \mathbf{m}_i\mathbf{m}_i^T \end{aligned} \tag{2}$$

Where m_i is the mean, C_i is the covariance, and the color is defined as the vector $x=[R\ G\ B]^T$. The probability of a pixel being in the road class using Bayes rule is:

$$P(road \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid road)P(road)}{P(\mathbf{x})} \tag{3}$$

where $P(\mathrm{x})$ is calculated by:

$$P(\mathbf{x}) = P(\mathbf{x} \mid road)P(road) + P(\mathbf{x} \mid offroad)P(offroad) \tag{4}$$

and where $P(\mathrm{road})$ and $P(\mathrm{offroad})$ represent the expected area of the road and off-road region.

The likelihood $P(\mathrm{x}|\mathrm{road})$ is found as the maximum of each of the road's color class probabilities:

$$P(\mathbf{x} \mid road) = \max_{w_r} \left\{ P(w_r) P(\mathbf{x} \mid w_r) \right\} \tag{5}$$

where w_r represents each color class model.

Each of the road color probabilities is represented by a Gaussian function:

$$P(\mathbf{x} \mid road) = (2\pi)^{-3/2} |C_r|^{-1/2} \exp\left[-\frac{1}{2} (\mathbf{x} - \mathbf{m}_r)^T C_r^{-1} (\mathbf{x} - \mathbf{m}_r) \right] \tag{6}$$

The maximum of this function can be computed by finding the maximum of its natural logarithm:

$$\begin{aligned} \ln(P(\mathbf{x} \mid road)) &= \max_{w_r} \left\{ L_r - \frac{1}{2} (\mathbf{x} - \mathbf{m}_r)^T C_r^{-1} (\mathbf{x} - \mathbf{m}_r) \right\} \\ L_r &= -\frac{1}{2} \ln\left[(2\pi)^3 |C_r| \right] \end{aligned} \tag{7}$$

where L_r is computed once per frame.

The results obtained compare very well against edge based detection schemes especially for dirt roads or partially obscured road edges. SCARF was able to perform better than a human selected color threshold value (recall the ALV/Alvin) in segmenting out the road from the background.

SCARF also detects intersections. A parameterized model of a branch intersection is used to generate bit masks representing perspective views of various types of intersections or straight road segments. The measure of fitness is computed as the sum of absolute differences between the image area classified as being on the road and the generated bit masks. The road segment is classified as the corresponding class of the best fitting bit masks road model. Curved roads are approximated with a straight road at an angle.

2.2.1.3 XUV

Part of the Demo III efforts involved developing a neural network (NN) approach [20] that takes range and color information to segment ill-structured dirt, grave and asphalt roads. A tele-operated XUV was used to collect color video and laser range information.

The training and testing data consisted of 107 video frames randomly chosen from the recorded data and matched up with the laser range data that was closest in time. A sparse sampling grid was used to apply a variety of filters to obtain fourteen different features from the color and range data. Truth data was obtained by manually segmenting the color and range images into road and non-road regions. The types of features used as inputs to the neural net were:

- Joint RGB histogram and component histograms
- Color texture features obtained using Gabor Filters
- Mean and Covariance of the laser range about each sampling point.

The neural net contained one hidden layer of 20 units. Various input feature configurations were used: Color, Texture, Color + Texture, Color + Laser, etc. For each of the frames chosen, a separate NN was trained and its performance against the other 106 frames measured. The mean and standard deviation of the NN performance for each feature combination were then tabulated.

The best single feature to use was found to be Color; adding Texture and Laser Range features did not increase the ensemble accuracy of the 107 NN's but decreased the standard deviation.

A flat earth model can be used to convert the NN classified pixels into a 3D point cloud. [20] did not address extracting or fitting a road model to the 3D points.

2.2.2 Structured Roads

2.2.2.1 YARF

YARF (Yet Another Road Follower) [21] was developed to permit the Navlab to follow well structured roads at higher speeds than SCARF was capable of. YARF was designed to extract the middle yellow lane markers and the solid outside lines and fit a road model to the extracted coordinates.

YARF uses color hue information to segment out the yellow lane markers, detects them using blob processing and compute their centroids. The solid white line is searched for using a horizontal stripe template that is correlated against the blue image component. The stripe template is sized according to the expected stripe width in the image, which for a flat earth model varies with the image row.

The computed yellow stripe locations and solid bar centers are collected over several frames and converted to the same 3D world coordinate system. An M-estimator [22] is used to fit a road model comprised of a centerline and lane boundaries to the extracted feature locations.

2.2.2.2 LOIS

Likelihood of Image Shape (LOIS) [23] is a lane tracker that uses a deformable template approach. A single camera is used to capture the road image. The projection of the road edges onto the 2D image plane can be approximated by a parabolic equation of the form $x = k * y^2 + m * y + b$, where x and y are the pixel coordinates.

A likelihood function is defined whose value is proportional to how well a particular set of parameters matches the pixel data in a specified image. For each pixel in the image the distance to the closest parabola is found and the normal to the curve at that point computed. The projection of the image gradient to the normal is computed and weighted by its distance to the curve with a function of the form $f(\alpha, d) = \frac{1}{1 + \alpha * d^2}$, where d is the distance from the pixel to the curve. The likelihood function is then expressed as:

$$L(k, m, b) = \sum_x \sum_y \overrightarrow{Gradient(x, y)} \bullet \overrightarrow{normal(x, y)} * f(\alpha, d(x, y)) \qquad (8)$$

LOIS searches through parameter space using the Metropolis Algorithm [24]. For an initial set of values $\mathbf{v} = \begin{bmatrix} k & m & b \end{bmatrix}^T$ a random perturbation $\mathbf{dv}$ is selected and the likelihood function $L(\mathbf{v})$ and $L(\mathbf{v}+\mathbf{dv})$ are compared. If the new likelihood is greater the perturbation is accepted, it is less it is accepted with a probability between 1 and 0. The probability density function is an exponential function of the form:

$$p(\mathbf{v}, \mathbf{dv}) = e^{\left(\frac{L(\mathbf{v})-L(v+\mathbf{dv})}{\alpha T_i}\right)} \tag{9}$$

The "temperature" T_i decreases as the number of iterations increases. This makes the probability of accepting a lower likelihood decrease over time. What this method does is allow transitions to a less likely set of parameters in the early stages when the temperature is "hot", this prevents being trapped in a local minima. LOIS finds the most likely curve shape with maximum support for image gradients oriented normal to the curve. This reduces the effect of shadows since most of the vertical gradients are ignored.

Figure 2 shows LOIS results after 40 iterations for a road with shadows.

Figure 2 – LOIS Results on a Road with Shadows

(Reprinted with permission from "Tracking Lane and Pavement Edges Using Deformable Templates" in Proceedings of SPIE Vol. 3364 "Enhanced and Synthetic Vision 1998", pages 167-176, by K.C. Kluge, C.M. Kreucher and S. Lakshmanan)

2.2.2.3 ARGO/MOBLAB/GOLD.

Bertozzi and Broggi have done extensive work with autonomous vehicle navigation [11, 15, 17, 18]. They use inverse perspective mapping to generate a top view of the roadway for determining the road boundaries and for finding obstacles. As Figure 3 shows their technique has shown very good results in real road situations. However they do not require a lot of precision in the farthest end of the image for steering their vehicle. Since there are less pixels on the road as the range increases, the boundary location accuracy suffers.

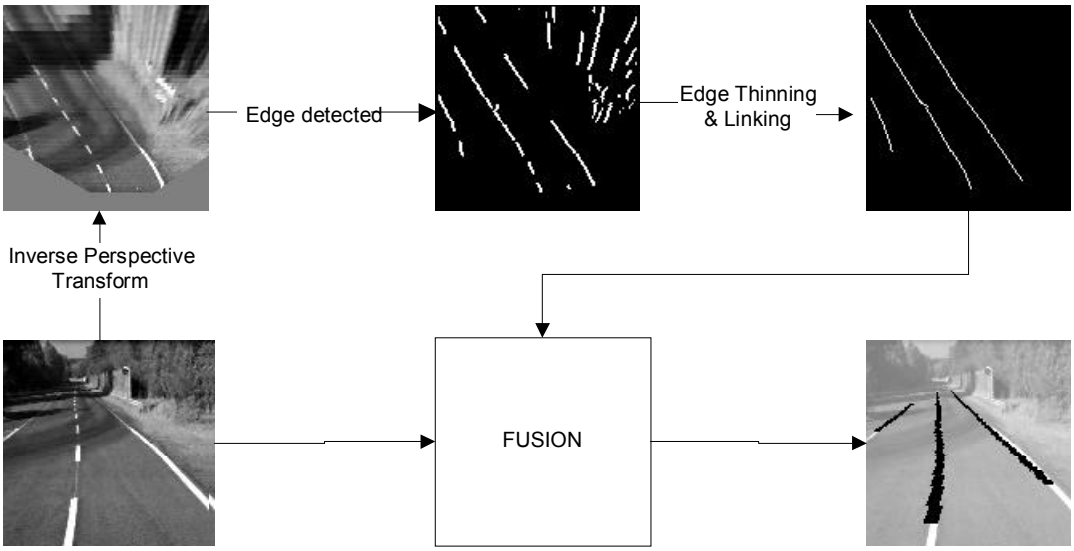

Figure 3 – Inverse Perspective Mapping for Road Localization

(Reprinted with permission from VisLab, Artificial Vision and Intelligent Systems Lab, Univ of Parma, Italy (http://vislab.unipr.it/GOLD/))

2.2.3 Road Followers

Road followers are designed to provide a steering command directly to a Navigation System. They do not attempt to provide road lane coordinates needed to construct a 3D road model.

2.2.3.1 ALVINN

ALVINN (ALV Neural Network) [7] is a neural network that is trained by observing the camera image and the drivers steering input. ALVINN uses a reduced resolution image fed directly to a 30x32 array of input units these are connected to a single hidden layer of 4 units and these in turn connect to 30 output units.

Once training is complete ALVINN will output a bell like curve with a narrow peak whose location corresponds to the road curvature.

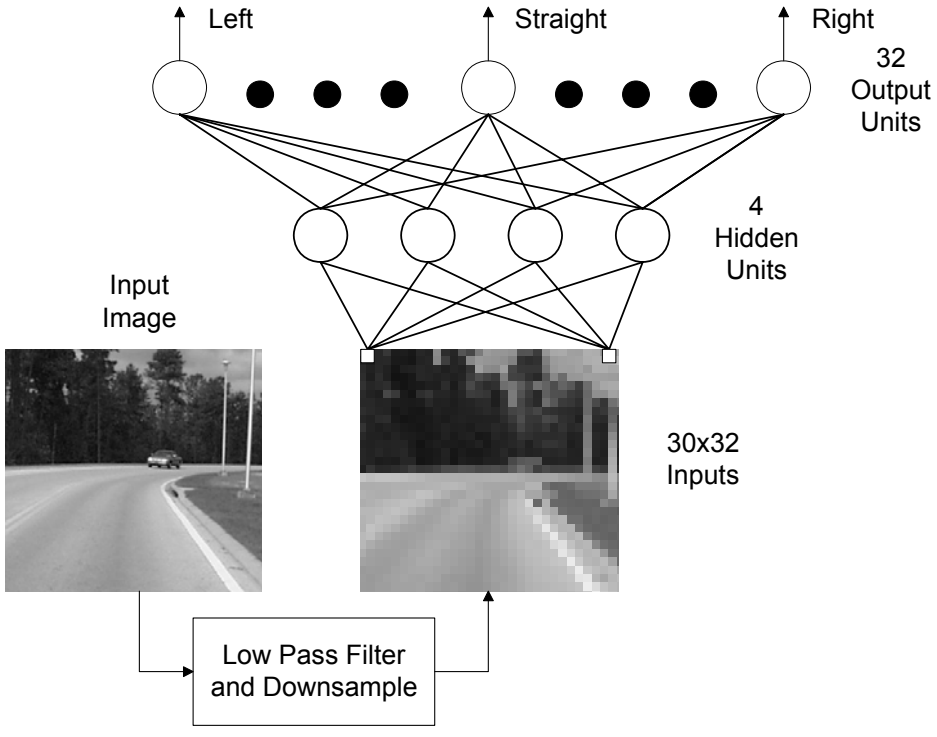

Figure 4 – ALVINN

2.2.3.2 RALPH

The Rapidly Adapting Lateral Position Handler (RALPH) [10] uses a single camera to determine a vehicles position in its lane and to measure the amount of road curvature up ahead.

RALPH uses a reduced resolution image of like ALVINN. RALPH distorts the image to remove the perspective effect and then computes the column averages. A gradient of the column average vector will show peaks corresponding to the road boundaries. The location of the peaks also shows where the vehicle is laterally positioned on the road.

If the road is angled or curved the gradient peaks are blurred in comparison to a straight ahead segment. RALPH can determine the degree of curvature by distorting the image and determining which of the transformations has the highest gradient contrast. RALPH is able to "lock" onto roads in conditions where the lane markers are not readily visible, it can follow discolorations in the middle of the road , it can even lock onto vehicles in front of it.

RALPH was demonstrated in 1995 on the Navlab 5 a 1990 Pontiac Trans Sport in the "No Hands Across America" [25] tour where it was able to autonomously drive 98.2% of 2797 mile course from Pittsburgh to San Diego.

2.2.3.3 Vanishing Point Based Method

A technique for determining road curvature is described in [26] that uses the vanishing points of the road edges. The road image in a single camera is divided into horizontal stripes which become shorter in height nearer to the horizon. The stripes represent road segments of equal length. A sobel edge detector is used to extract edge points which are linked together into line segments using a Hough transform. The vanishing point is defined as the intersection of the lines on which left and right road edge segments lie on.

For a straight road the vanishing points computed for each horizontal stripe would lie close to each other. A curved road is indicated by a shift in the vanishing point as one moves up through the image. The degree of shift corresponds to the road curvature and can be converted to a steering command.

2.3 Conclusion

Graefe and Kuhnert in the introductory chapter of the 1992 Vision-based Vehicle Guidance stated "It will certainly take a long time before autonomous vehicles will be an everyday appearance on public roads …" [27]. While a general purpose road follower is still years away, we see that for special purpose situations such as modeling portions of the UCF Campus road network we can use the techniques of color based pixel classification, inverse perspective matching, stereo disparity , oriented operators and sub-windowing as a starting point to develop a semi-supervised road modeller.

3 APPROACH

3.1 System Analysis

This System Analysis was performed to estimate the expected 3D accuracy of the extracted road boundaries. The System Analysis begins with a description of the nominal system architecture, the various coordinate transformations involved in going from 2D image plane measurements to 3D world coordinates and the throughput requirements.

The System Analysis decomposes the nominal architecture into the major components and estimates the error contributions of each component. Finally through an error block diagram we roll up the individual contributions to obtain the overall system accuracy.

3.1.1 System Description

The system is comprised of a pair of color CCD video cameras mounted on top of a 1984 Ford LTD Station Wagon along with a Solid State Compass, GPS receiver and an odometer. All data is collected by a Dual Pentium III PC and saved to a RAID disk array. The data is post-processed using Matlab software and an initial set of road boundaries input by the user.

The Matlab software finds the road boundaries in the video data then, using the other sensors, converts the 2D road boundaries to 3D world coordinates.

3.1.1.1 System Block Diagram

A block diagram of the system is shown in Figure 5. This diagram combines the capture and post-processing steps.

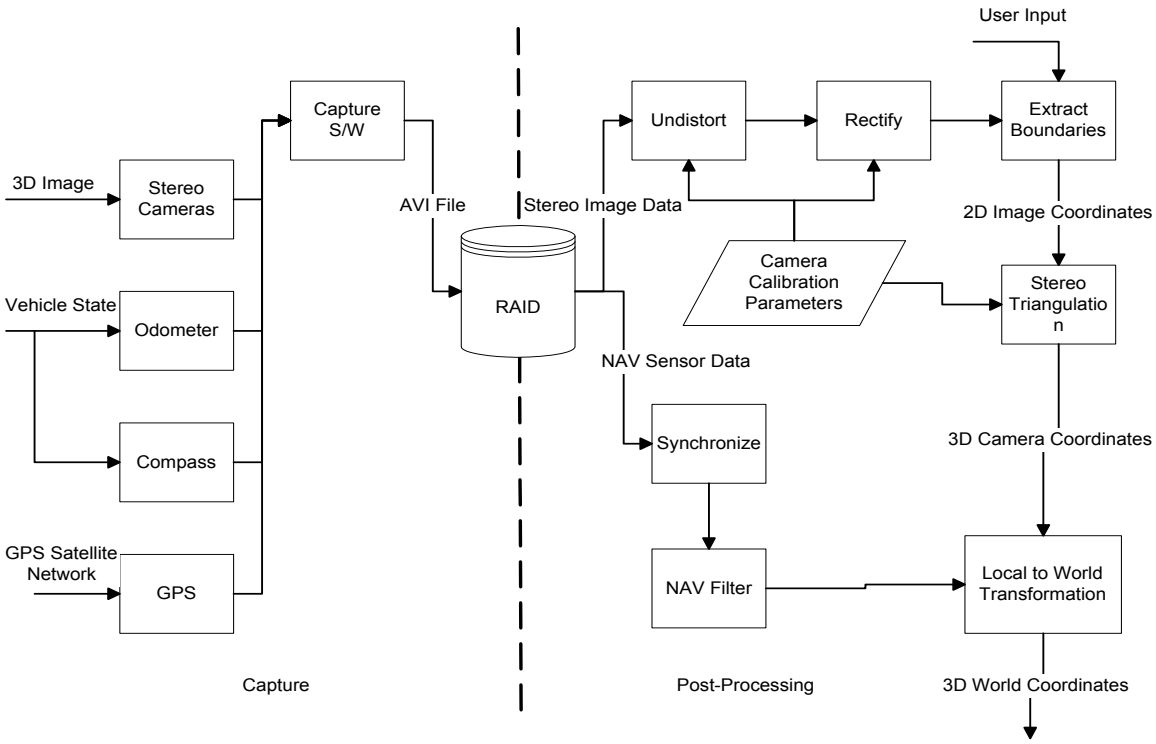

Figure 5 – System Block Diagram

Capture software combines the video and sensor data creating an AVI (Audio-Video Interleaved) file. The AVI file is processed by Matlab software routines which separate the video and sensor data and feed them to various software modules.

The stereo image data is processed to extract 3D coordinates of the road boundaries which are converted to world coordinates using the navigation filter output.

3.1.1.2 Coordinate System Definition

This section describes the different coordinate systems involved in converting pixel coordinate to world coordinates.

3.1.1.3 Camera Coordinates

The format of the saved video data is 1280x480 (640x320 also supported) with the first video line used to store sensor data. Each frame is made up of a left and right 640x480 images concatenated together, see Figure 6.

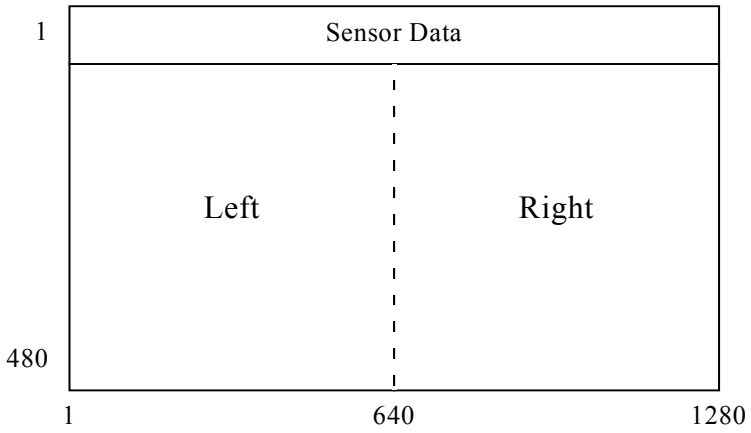

Figure 6 – Video Frame Format

We will assume our cameras are located on the X axis of a R.H.S. which has positive X towards the right, positive Y pointing down and positive Z pointing forward with respect to the vehicle. The left camera is located at the origin with the right camera a distance B away. This camera coordinate system fixed to the left camera's optical center and rolls and pitches with the vehicle (i.e. cameras are assumed fixed with respect to the vehicle).

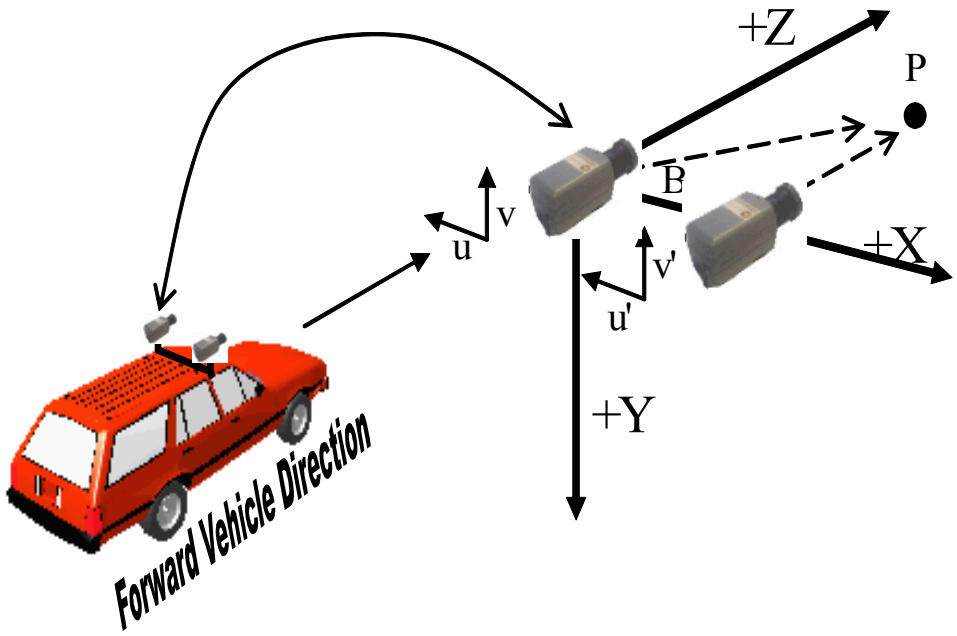

Figure 7 – Camera Coordinate System

Assuming a locally flat earth, a point P visible to both cameras would have Universal Transverse Mercator (UTM) [28] coordinates Px, Py and elevation Pz (relative to sea level) and is projected onto the cameras 2D sensors at coordinates u,v and u',v'.

A top-down view would look as follows,

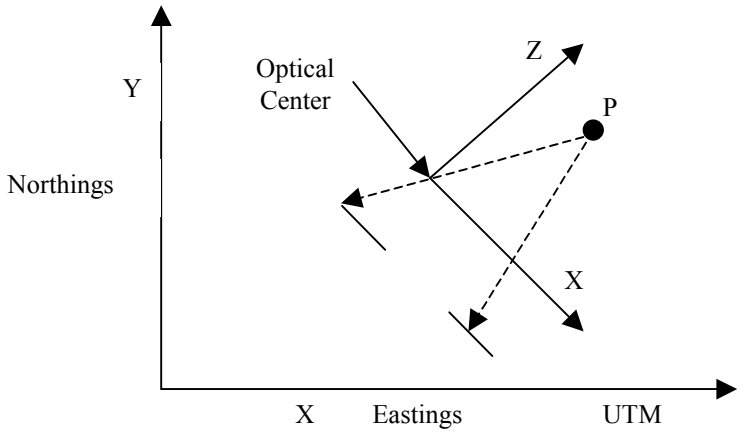

Figure 8 – Top Down View of Camera Coordinate System Relative to UTM

The Z axis of the camera coordinate system always points in the direction of the vehicles forward orientation, note that when traveling around a curve the vehicles forward orientation and the vehicles direction of travel are not aligned.

The image analysis S/W will find a set of points in the left and right images and determine that they correspond to projections of the same point P in 3D space. Assuming negligible roll we have the following.

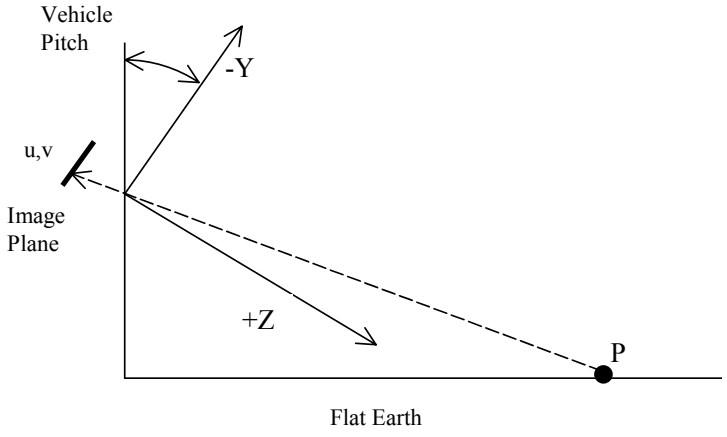

Figure 9 – Side View of Projection of 3D World Point P into Camera Coordinate System

A pair of optical rays can be constructed that begins at each camera's u,v coordinate, passes through the optical center and converge at P. Therefore P's location in the cameras coordinate system is x,y,z.

3.1.1.4 World Coordinates

The world coordinate system shall be Universal Transverse Mercator Projection (UTM) [28] .UTM uses a projection of 6^o wide strips of the earth onto a transverse cylindrical surface wrapped around[1] the earth at one of the 60 central meridians.

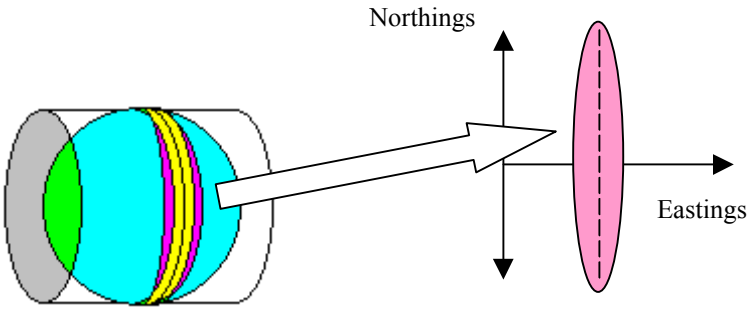

Figure 10 – Universal Transverse Mercator Projection

Latitude and Longitude measurements taken with the GPS shall be converted to UTM using the appropriate central meridian for Orlando, Florida.

UTM x and y coordinates are in meters and by convention the origin is placed 3^o left of the central meridian at the equator so x (eastings) is always positive.

The following figure relates the coordinate systems from measured image plane coordinates to world coordinates.

---

[1] Actually the imaginary cylinder is partially below the surface near the equator in order to minimize the error at the +/-3^o points from the meridian

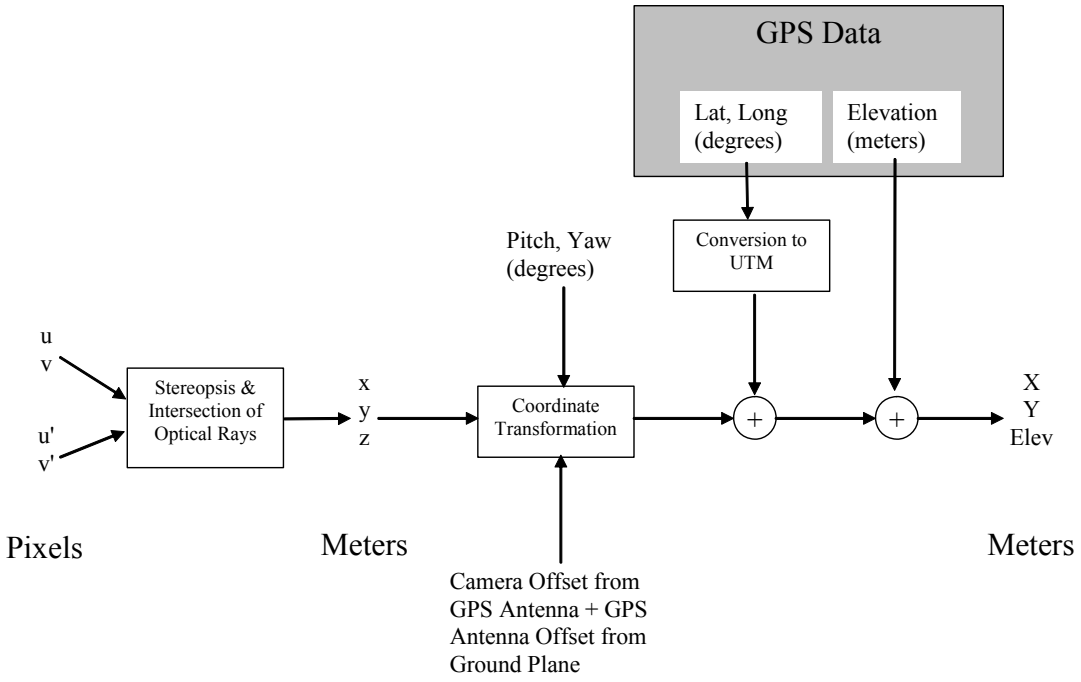

Figure 11 – Coordinate Systems Relationship

3.1.2 Throughput Requirements

3.1.2.1 Video Capture

The cameras used are the Sony DFW 500 models that use a Firewire™ (IEEE 1394 [29]) interface to the PC. The maximum 1394 bandwidth is 400 Mbit/s of which a maximum of 85%[30] can be allocated to isochronous (constant data rate) transfer. The DFW 500 is capable of capturing 4:2:2 YUV (Luminance, Blue Chrominance, Red Chrominance) [31] at resolutions of 640x480 and 320x240 and at frame rates of 30 Hz, 15 Hz, 7.5 Hz, etc or synced to an external trigger which can have a maximum rate of 15 Hz [32].

A single firewire adapter can connect to two or more cameras. Two cameras operating at 15 Hz, 640x480 resolution would require:

2 * 15 fps * 640 * 480 * 16bits/pixel = 147 Mb/s < 400 Mb/s x 0.85

For a standard PC the PCI bus will burst transfer in 64 byte (16 Dword) chunks before the bus is rearbitrated. Assuming a 2.4 microsecond latency time gives us a max sustained data rate of 100 MB/s [33]. This must be cut in half since the video data is transferred to memory first and then out of memory to the hard drive.

The hard drives used are a striped set of 9 GB SCSI drives which have been verified to support 28 MB/sec data transfer rate.

Experiments conducted on a dual Pentium III (450 MHz) with 256 MB of memory with 2 Sony DFW 500 cameras and a single firewire adapter have shown that capturing at 10 Hz (external sync) can be done with no missed frames.

3.1.2.2 Other Sensors

Compared to the video requirements, the requirements for the other sensors are trivial.

- GPS Sensor at 1 Hz Raw Measurement Rate : 256 bytes/second
- Solid State Compass 10 Hz Angles- Accelerations : 130 bytes/second
- Odometer 10 Hz Odometer : 40 bytes/second

3.1.3 Error Analysis

3.1.3.1 Intrinsic Camera Errors

Intrinsic camera errors cause a point/edge in real world space to project to a different position in the cameras CCD sensor than would be expected if ideal pinhole camera optics were used.

3.1.3.2 Optical Errors

Optical Distortion in lenses causes “barrel” or “pincushion” distortion in images (see Figure **12**), wide FOV lenses in particular have a lot of distortion.

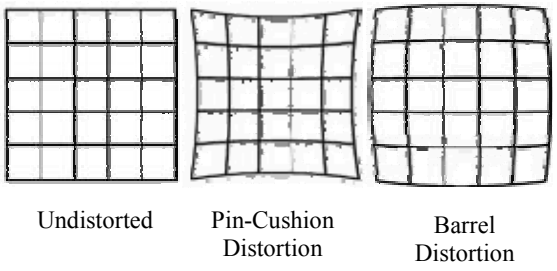

Figure 12 – Types of Lens Distortion

We will assume that lens distortion can be modeled with the radial distortion model used in calibration technique described in[34] which has been implemented in Matlab by[35] . This yields a maximum pixel error of less than 1 pixel due to errors in the estimation of radial distortion coefficients.

The Optical Transfer Function (OTF) of a lens determines the maximum resolution possible. Since there is no means to obtain direct CCD readings of the Sony camera due to the use of a Bayer Filter [36] in its design we will use the lens manufacturer’s design parameters for the Circle of Confusion (COC). The Circle of Confusion describes how much a point source that is not at the focused distance (i.e. 50 ft, infinity) is blurred as shown in Figure 13. Points B and C in Figure 13 will remain in focus as long as their respective COCs remain small enough do that it is approximately 70% covered by an image element as shown in Figure 14.

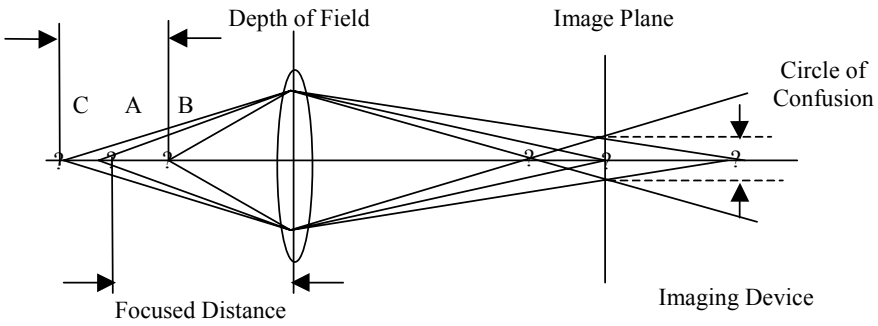

Figure 13 – Circle of Confusion

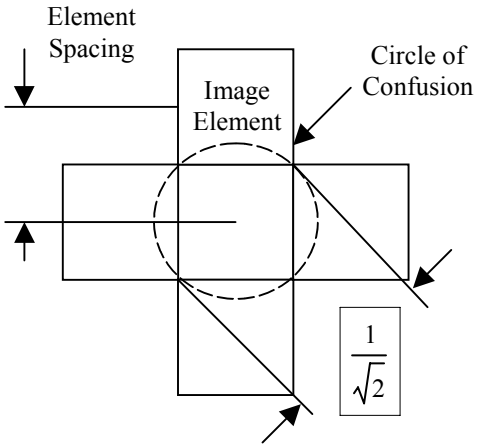

Figure 14 – COC Coverage by Pixel

A COC with a diameter equal to the diagonal element size has coverage of:

$$\frac{\text{Circle of Confusion Area}}{\text{Unit Element Size}} = \frac{\pi r^2}{d^2} = \frac{2}{\pi} = 0.637 \;\; or \; 63.7\% \qquad (10)$$

The manufacturer of the lens used has listed acceptable COC's for various standard CCD sizes. This table is reproduced below.

Table 1 – Permissible COC by CCD Type

CCD Type	Permissible Circle of Confusion
2/3"	0.021mm
1/2”	0.016mm
1/3”	0.011mm
1/4”	0.008mm

Source: From Rainbow CCTV (http://www.rainbowcctv.com/tech/depth.html)

The lens is a Rainbow S48WI and can be used with 2/3, ½ and 1/3 type CCD’s. The CCD used by the Sony DFW-V500 camera is a Wfine™ CCD Type 1/3, which has an element size of 7.4 by 7.4 micrometers (10.46 micrometers diagonal). From the manufacturer’s table we see that the COC is adequate for this CCD and therefore we can use a pixel error of ½ as the error contribution of the lens. (OTF)

Blurring of the image can also be caused by linear and random motion. The linear motion of the camera over the road results in an optical flow where distant parts of the image are still while the close up portions blur as they speed by. Random motion is due to vehicle vibration being transferred to the camera and causing jitter.

The Modulation Transfer Function (MTF) degradation due to linear motion is [37]

$$MTF_{linear} = \frac{\sin(\pi a_l f_x)}{\pi a_l f_x} \tag{11}$$

where:

a_l = angular distance moved in radians = $v_r t_{int}$

v_r = relative angular velocity in radians per second

t_{int} = integration time (shutter speed) (typical CCD value is 1 to 10 milliseconds)

f_x = spatial frequency in cycles per radian

Equation 11 can be rewritten to express the angular distance motion in fractions of the detector angular subtense and the spatial frequency in terms of fractions of the detector cutoff frequency. The detector angular subtense (DAS) is related to the cutoff frequency by the following equation:

$$\text{detector angular subtense} = \alpha = \frac{1}{f_{dco}} = \frac{\text{detector size}}{\text{effective focal length}} \tag{12}$$

Using this definition Equation 11 is rewritten as follows:

$$MTF_{linear} = \frac{\sin(\pi a_l f_{dco} \frac{f_x}{f_{dco}})}{\pi a_l f_{dco} \frac{f_x}{f_{dco}}} = \frac{\sin(\pi \frac{a_l}{\alpha} \frac{f_x}{f_{dco}})}{\pi \frac{a_l}{\alpha} \frac{f_x}{f_{dco}}} \tag{13}$$

For a point at a distance d lying on the road and assuming the cameras are at a height h and the vehicle has forward velocity dot-d the angular motion is:

$$\begin{aligned} w &= \frac{h}{d} \text{ (using small angle approximation)} \\ \dot{w} &= \frac{-h}{d} * d^2 \end{aligned} \tag{14}$$

For typical values of *d* equal to 15 m, speed at 50 km/hr and h equal to 2 m we get an angular velocity of 0.106 radians per second which for an integration time of 1 millisecond yields an angular movement of 0.106 milliradians. For a detector size of 7.4 micrometers and focal length of 4.8 millimeters, our DAS is equal to 1.54 milliradians and we see the motion of the point expressed as a fraction of a DAS is 0.068. Holst [37] gives a rule of thumb that for motion comprising less than 0.2 DAS there is a negligible degradation of the MTF.

In Figure 15 we see the linear MTF for distances of 16.8, 12.2 and 7.6 meters (55, 40 and 25 ft) given a speed of 50 km/hr, and integration time of 1 millisecond. If we compare this to Figure 16 where the distance is kept to 7.62 meters but the shutter speed is varied from 1 to 10 milliseconds we see that for a low shutter speed the MTF degradation is slight but for slow shutter speeds (such as those needed in low light conditions) we see a significant MTF degradation.

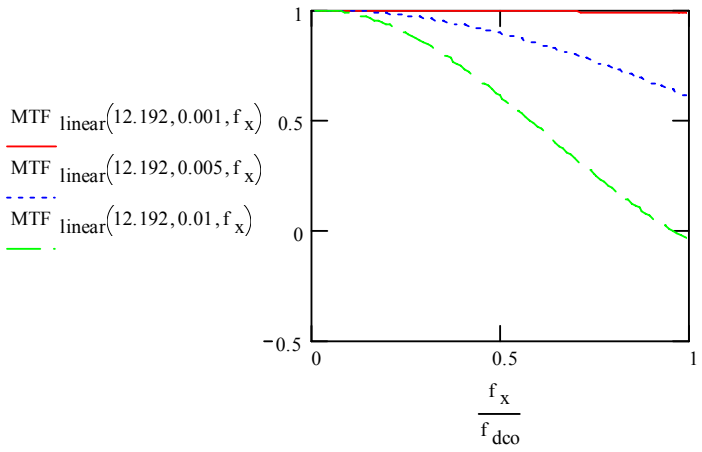

Figure 15 – Linear MTF as Function of Normalized Spatial Frequency

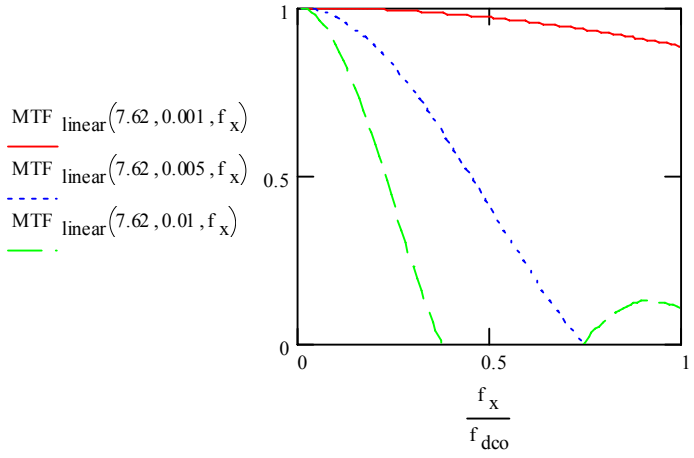

Figure 16 – Linear MTF as a Function of Shutter Speed

MTF degradation due to sinusoidal and random motion are given by the following equations expressed in fractional DAS motion and normalized spatial frequency.

$$MTF_{\text{sinusoidal}} = J_0(2\frac{a_{\sin}}{\alpha}\frac{f_x}{f_{dco}}) \tag{15}$$

where J_0 is the zero order Bessel function and $a_{\sin}$ is the amplitude the sinusoidal motion and:

$$MTF_{\text{jitter}} = \exp\left[-2\left(\pi\frac{\sigma_{rms}}{\alpha}\frac{f_x}{f_{dco}}\right)^2\right] \tag{16}$$

where σ_{rms} is the rms random displacement in milliradians

Equations 15 and 16 however require that there be many sinusoidal or random motions occurring during the integration time which would require frequencies in the kilohertz range. According to[38] such frequencies will not be passed by the vehicle suspension. Typically frequencies on the order of 1.5, 8 and 13 Hz will affect the vehicle pitch at a speed of 50 km/hr.

Experimental data using a solid state compass collected for the vehicle while stationary and moving shows that the power spectral density (PSD) is significantly attenuated beyond 1 Hz for smooth roads. Comparing the PSD of the vehicle pitch while in motion to the PSD of the sensor being shaken by hand shows how much the suspension is attenuating high vibration frequencies.

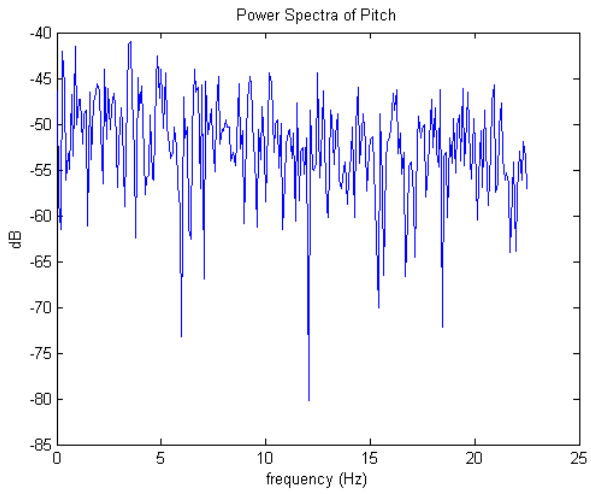

Figure 17 – Pitch Power Spectral Density for Stationary Vehicle

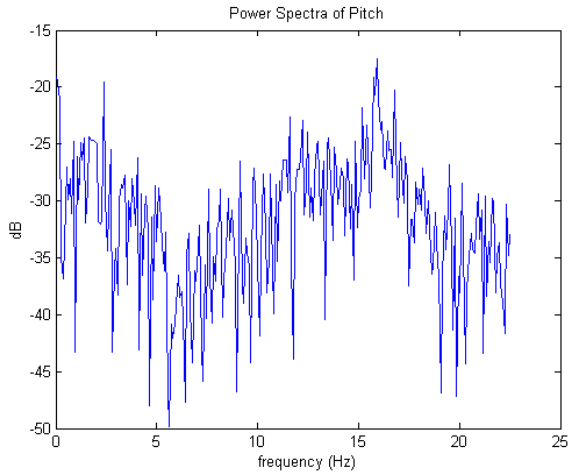

Figure 18 – Pitch Power Spectral Density for Vehicle at 30 MPH

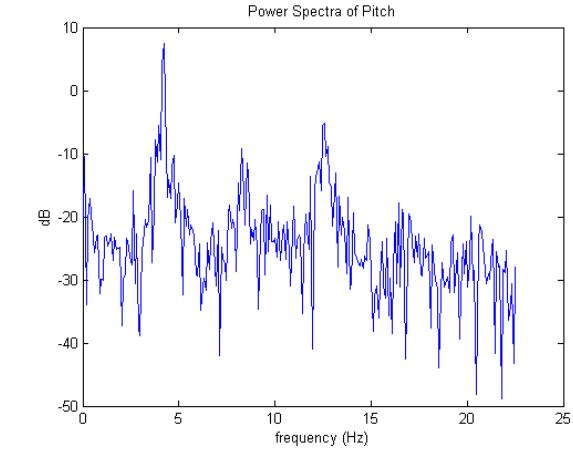

Figure 19 – Pitch Power Spectral Density for Sensor Being Shaken by Hand

Since the resonant frequencies are low for the pitch we can use Equation 15 and compute the fractional DAS motion for the 15 Hz component shown in Figure 18. The 15 hz motion is described by:

$$a_l = a_0 \sin(2\pi w t)\ t_{int} \tag{17}$$

which has a maximum value of:

$$a_l(\max) = a_0\, 2\pi w\, t_{int}$$
$$a_0 = -18dB = 0.0158 \quad t_{int} = 0.001\,s \quad w = 15 \tag{18}$$

and yields $a_l \approx 1.5$ milliradians or a fractional DAS coverage of 1. The MTF for this is shown in Figure 20 along with the MTF of the RMS value for the fractional DAS coverage. The lower frequency peaks at 12.5 and 2.5 Hz will yield RMS values for the fractional DAS coverage of 0.24 and 0.08 respectively which according to the Holsts rule of thumb results in negligible MTF degradation.

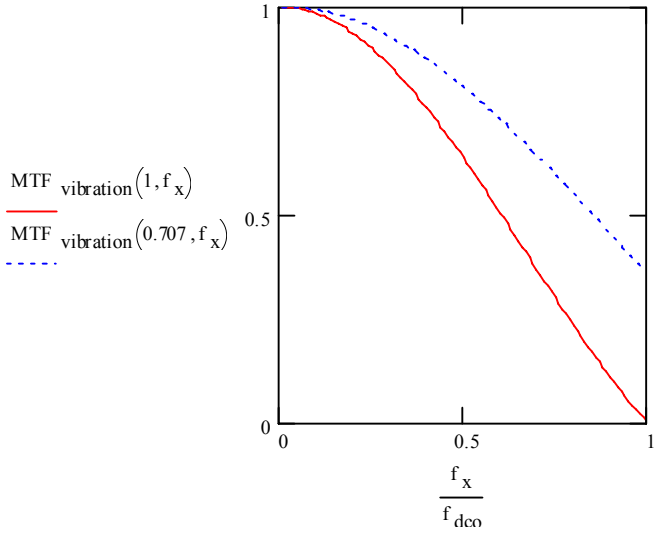

Figure 20 – Linear MTF for Fractional DAS Coverage of 1

3.1.3.2.1 Bayer Filter

Low cost digital cameras such as the Sony DFW-V500 use a single CCD with a Bayer filter[36] to obtain color information. As shown in Figure 21, the Bayer filter is made of individual color filters on the CCD elements, where 50% of the pixels have green filters, 25% red filters and the remaining 25% blue.

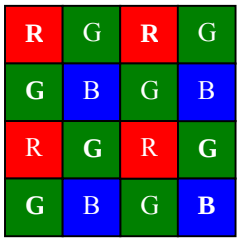

Figure 21 – Bayer Filter

In order to obtain a full RGB at each pixel it is necessary to interpolate the missing colors. The exact algorithms used by the varying camera manufacturers to demosaic the image are proprietary[39]. We will assume a simple linear interpolation is equivalent to convolving the following Kernels with the image:

$$H_B = H_R = \begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & 1 & 1/2 \\ 1/4 & 1/2 & 1/4 \end{bmatrix} \qquad H_G = \begin{bmatrix} 0 & 1/4 & 0 \\ 1/4 & 1 & 1/4 \\ 0 & 1/4 & 0 \end{bmatrix} \tag{19}$$

where the image is processed as follows:

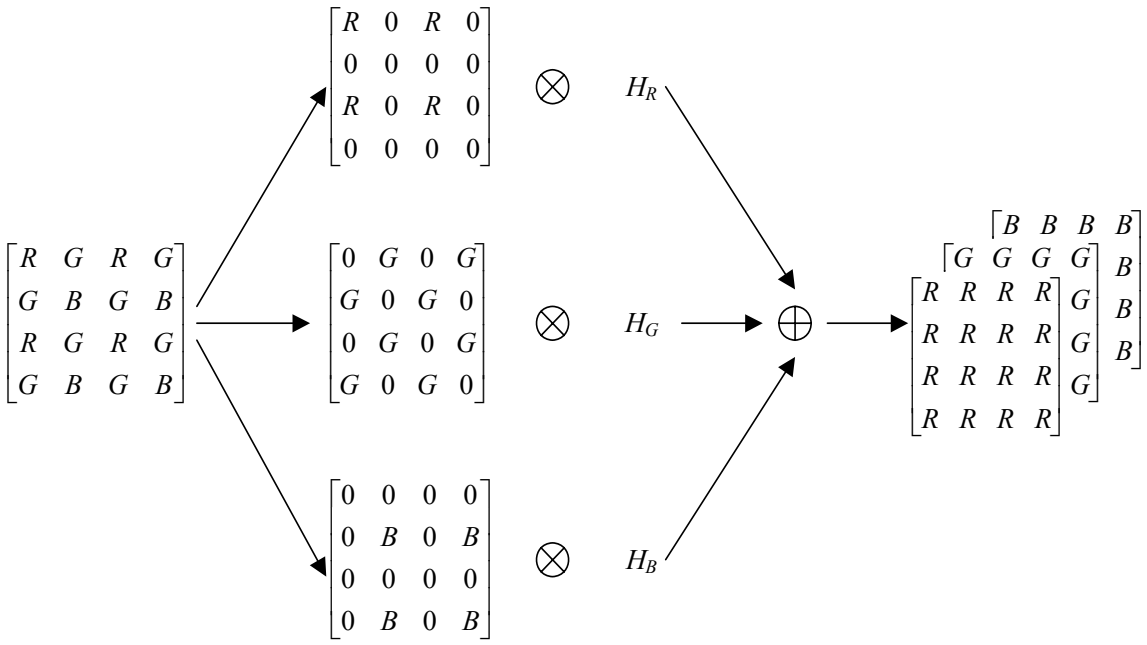

Figure 22 – Color Interpolation

Figure 16 shows the Fourier transform of the impulse response for H_B/H_R and H_G

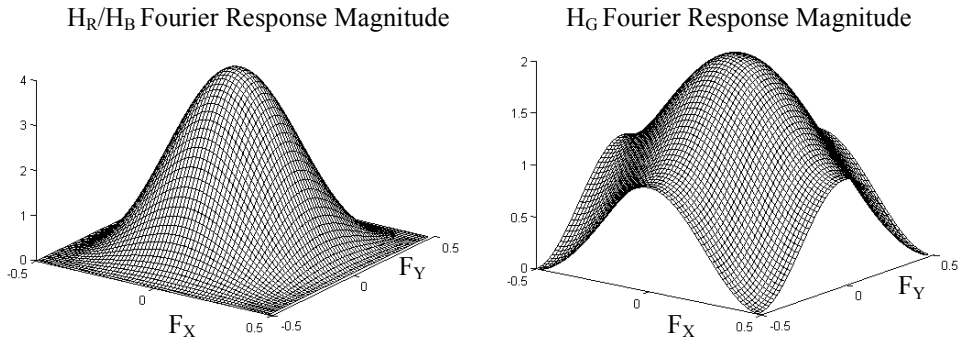

Figure 23 – Frequency Response of Demosaicing Filters

The interpolation filters act like low pass filters, with the Red and Blue filters showing more frequency rolloff than the Green filter.Using the Green filters response, since natural scenery has a preponderance of shades of green, we see an additional error contribution of 2 pixels rms by the demosaicing process.

3.1.3.3 Calibration Errors (Extrinsic)

Extrinsic calibration of the stereo cameras also includes rectification of the cameras so that corresponding points in the image pair lie on the same row. A set of images of a calibration pattern shown in Figure 24 were used to obtain the stereo camera parameters used to rectify one of the images (see Figure 25). Comparison of the left and right images shows a pixel error of 0.2181[2] in the vertical positions of the corresponding points.

The stereo calibration software also contains routines for computing the 3D camera coordinates of corresponding points. Table 2 has a comparison for the estimated 3D range vs. the measured range for the points shown in Figure 27.

[2] The Camera Calibration Toolbox developed by Jean-Yves Bouguet uses a Harris Corner Finder [40] to detect the pattern corners. A sub-pixel precision of 0.1 pixel is claimed by the developer.

Figure 24 – Left and Right Images of Calibration Pattern

Figure 25 – Rectified Image Pair

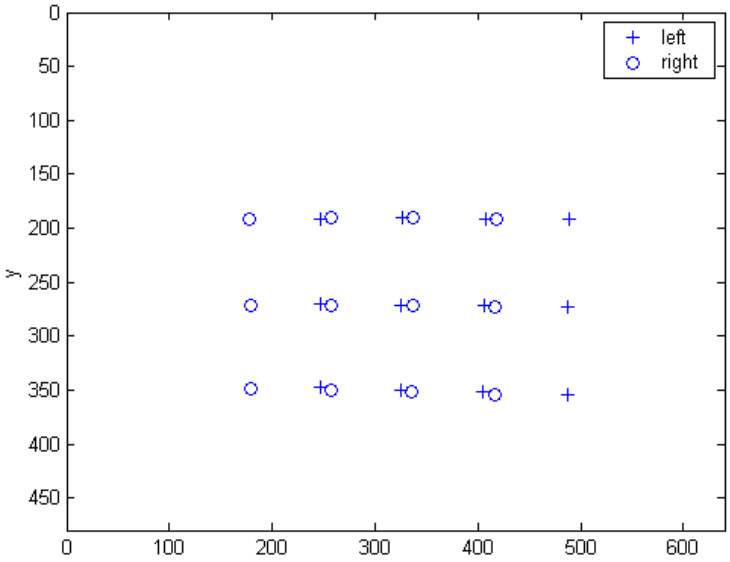

Figure 26 – Extracted Grid Points for Left and Right Rectified Images of Test Pattern

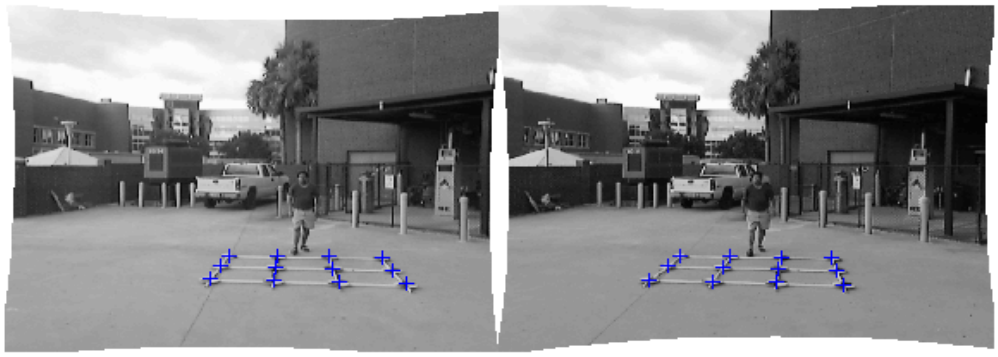

Figure 27 - Corresponding Points

Table 2 – Estimated .vs. Measured Distances for Test Pattern in Camera Coordinate System

Values in Meters							
Estimated				Measured			
8.98	9.09	9.08	8.98	9.04	9.06	9.08	9.09
8.34	8.32	8.17	8.22	8.12	8.14	8.17	8.19
7.22	7.13	7.11	7.21	7.19	7.22	7.26	7.29
Std Deviation of Error = 0.11 (4.2 inches) Std Deviation of % Error = 1.4 %							

3.1.3.4 Inverse Perspective Errors

Inverse Perspective Mapping transforms u,v camera coordinates to 3D coordinates. For a single camera system the world is assumed to lie on a flat plane and the 3D coordinates lie on the intersection of the projected ray from u,v with the plane.

The computed intersection point is sensitive to pitch errors so it is necessary to use an attitude measuring system such as a solid state compass.

Figure 28 shows the range error resulting from a ΔP pitch error.

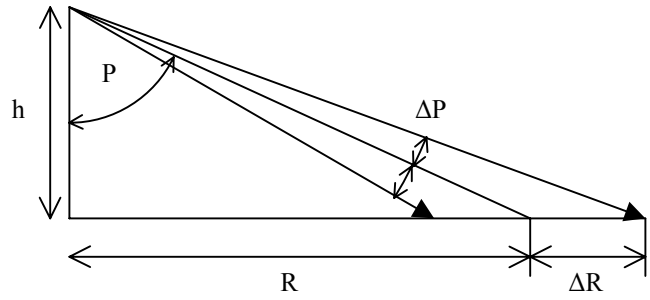

Figure 28 – Range Error from Pitch Error

For R=15 m, h=2 m and ΔP=±1deg we have:

$$p = \arctan\left(\frac{R}{h}\right), \quad R = h \cdot \tan\left(p \pm 1\right) \tag{20}$$

which yields R= ±h tan(82.4) or ΔR = +1.7, -2.2 m.

For working ranges from 7.6 m to 30.5 m (25 ft to 100 ft) we see the bounding errors in figure Figure 29.

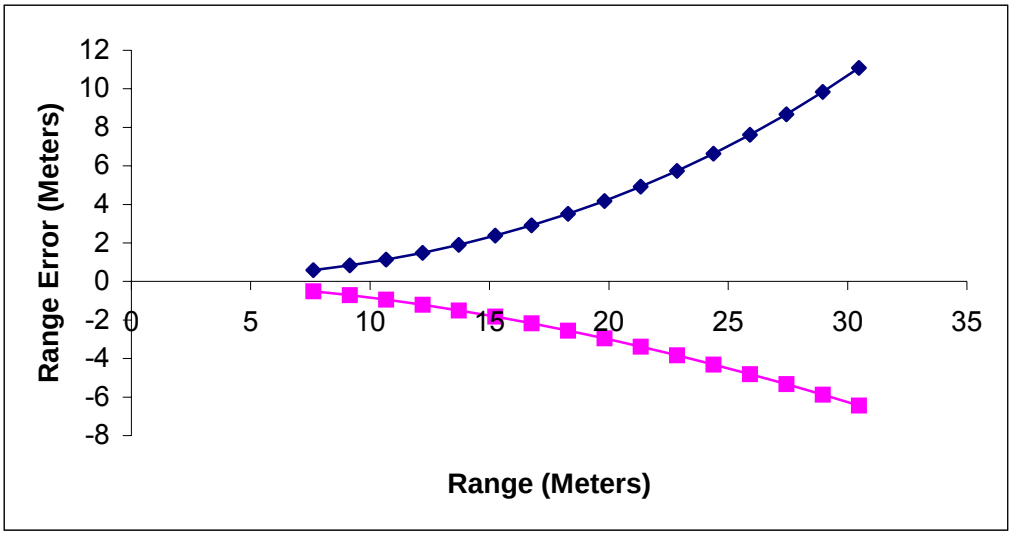

Figure 29 – Range Error for +/- 1 degree Pitch Error

3.1.3.4.1 Stereo System

A stereo system bounds the range error with the disparity measurement error. If we assume a ± 1 pixel error[3] in the disparity we see the Figure 29 above becomes:

We can express this range error in the 3D camera coordinates as a bounding box,

$$\mathbf{X}_c = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} + \frac{\Delta R}{2} \begin{bmatrix} \pm n_x \\ \pm n_x \\ \pm n_x \\ 0 \end{bmatrix} \tag{21}$$

where n_x, n_y and n_z are the direction cosines of the unit vector from the camera origin to x,y,z.

The world coordinates $\mathbf{X}_w$ are then affected by the range error:

---

[3] Sub pixel accuracy has been demonstrated on various systems [41] [42] but will be ignored for this analysis.

$$\Delta \mathbf{X}_w = \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T\mathbf{T} \\ 0 & 1 \end{bmatrix} \frac{\Delta \mathbf{R}}{2} \begin{bmatrix} \pm 1 \\ \pm 1 \\ \pm 1 \\ 0 \end{bmatrix} \tag{22}$$

where **R** and **T** are the rotation matrix and translation vector to go from camera coordinates to world coordinates.

3.1.3.4.2 Boundary Segmentation Error

The boundary segmentation error is the error in determining the correct pixel boundary between the road and terrain.

A segmentation performance of 0.8 is assumed which is consistent with reported results for road color segmentation[2], neural network[20] and Hidden Markov Model (HMM)[43] classifiers.

The segmentation performance was used to generate a series of class membership images as shown in Figure 30.

A fast snake algorithm [44] was then used to find the edges of the Gaussian filtered gradient of the class membership and the final boundary compared to the known boundary for a series of images in which the boundary was being shifted according to a sinusoidal pattern. As shown in Figure 31, after the snake has "captured" the boundary it is able to follow the moving boundary over time with a RMS pixel error of less than 1.5 pixels.

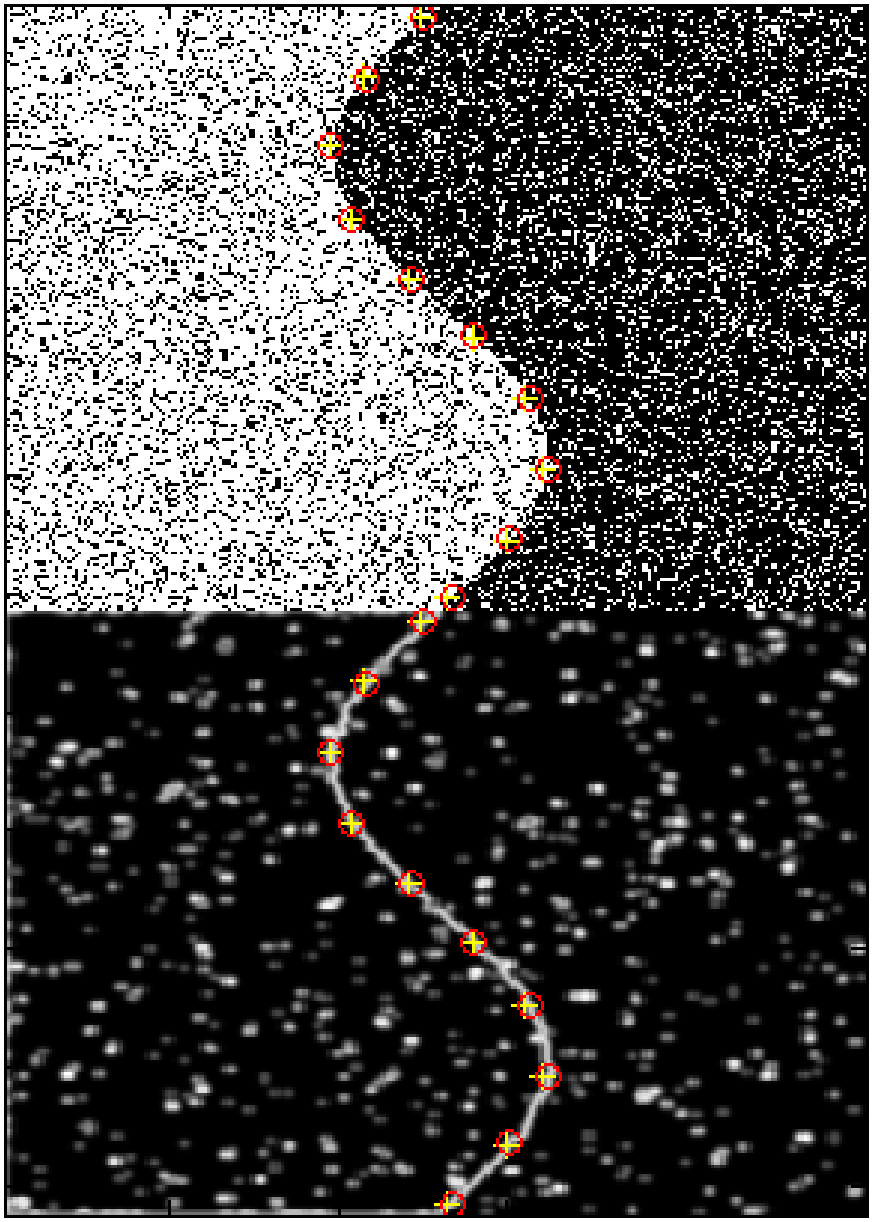

Figure 30 – Synthetic Segmentation Image (top) and Gradient (bottom)

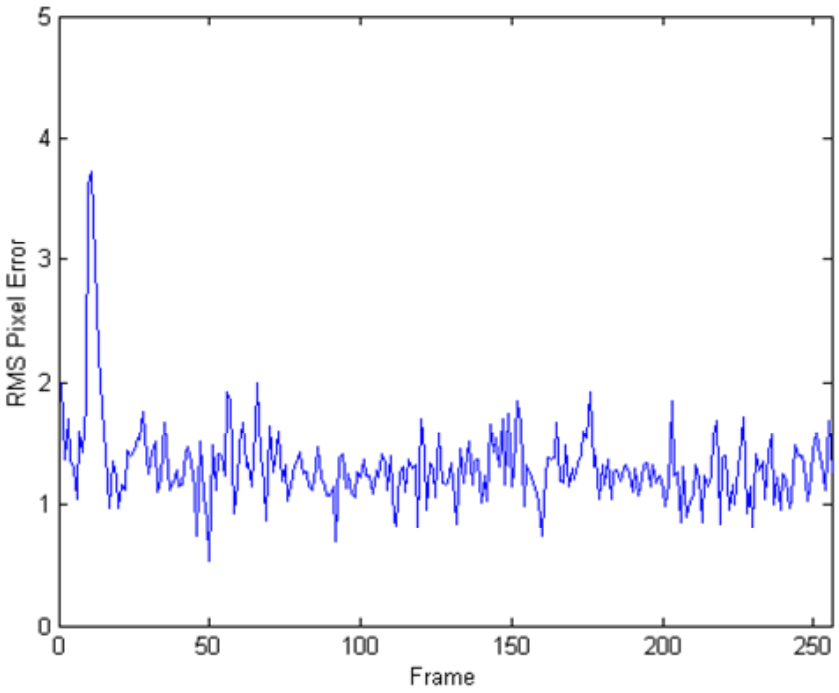

Figure 31 – RMS Pixel Error Over Time

3.1.3.5 Navigation Errors

The navigation errors are represented by the errors in the rotation matrix and translation vector used in converting from camera centered coordinates to world coordinates.

3.1.3.5.1 Odometer

The odometer is a Hall effect sensor attached to the speedometer output shaft of the transmission. A 50% duty cycle square wave is produced whose frequency is a function of speed as shown in Table 3. Measurements were made of the accuracy of the odometer sensor by constructing a digital counter and determining the counts given over a 37.0 m (121.5 ft) distance. The results are shown in Table 4 which shows each odometer count represents approximately 0.27 m (10.5 in) of forward motion.

Table 3 – Frequency Measurement of Odometer at Various Vehicle Speeds

Speed km/hr (mph)	Frequency (Hz)	
	Run 1	Run 2
8 (5)	13.99	14.02
16 (10)	20.53	20.53
24.1 (15)	25.45	26.96
40.2 (25)	43.86	42.01
56.3 (35)	58.47	56.18

Table 4 – Odometer Counts for Fixed Distance

Odometer Count for a Distance of 37.0 m (121.5 ft)	
Run	Count
1	139
2	139
3	138
4	139

Odometer Count for a Distance of 37.0 m (121.5 ft)	
Run	**Count**
5	139
6	138

3.1.3.5.2 Compass

A solid state compass is used which consists of three accelerometers and three magnetometers in an orthogonal arrangement. Experimental measurements show a stationary measurement noise of ±0.5 degrees RMS with a frequency response as shown in Figure 17. From these results we will assume a measurement noise of ±1 degrees in roll pitch and yaw in determining the vehicle attitude with respect to the world coordinates.

3.1.3.5.3 GPS

The GPS is used to determine the vehicle position in the world coordinate system. GPS measures the lat, long and altitude using satellite signals from a constellation of satellites that are currently in view[45] . Since the configuration of viewable satellites is constantly changing the measurement accuracy varies over time. Planning software [46] can be used to determine the best times to take measurements.

GPS data was collected for several roads on the UCF campus to compare raw measurements versus differential global positioning system (DGPS) measurements before and after the Selective Availability (SA) was turned off[4].

Since DGPS requires a base station in addition to a remote unit, the base unit antenna was placed on top of the Engineering I building.

Two DGPS procedures were used:

- A post-processed version where special firmware on the GPS devices was used to collect additional measurements which were processed by proprietary Novatel software.
- A real time version where correction measurements were sent from the base station to the remote unit over a wireless network link.

As shown in Figure 32 and Figure 33 the performance of the real time DGPS with SA off was as good as or better than the post-processed DGPS with SA on. For the real-time DGPS with SA off; the results were ±0.3 meters error in horizontal position and ±0.6 meters error in altitude.

[4]Selective Availability is intentional degradation of the Global Positioning System (GPS) signals available to the public and was turned off by Order of the President of the United States on Midnight May 1, 2000.

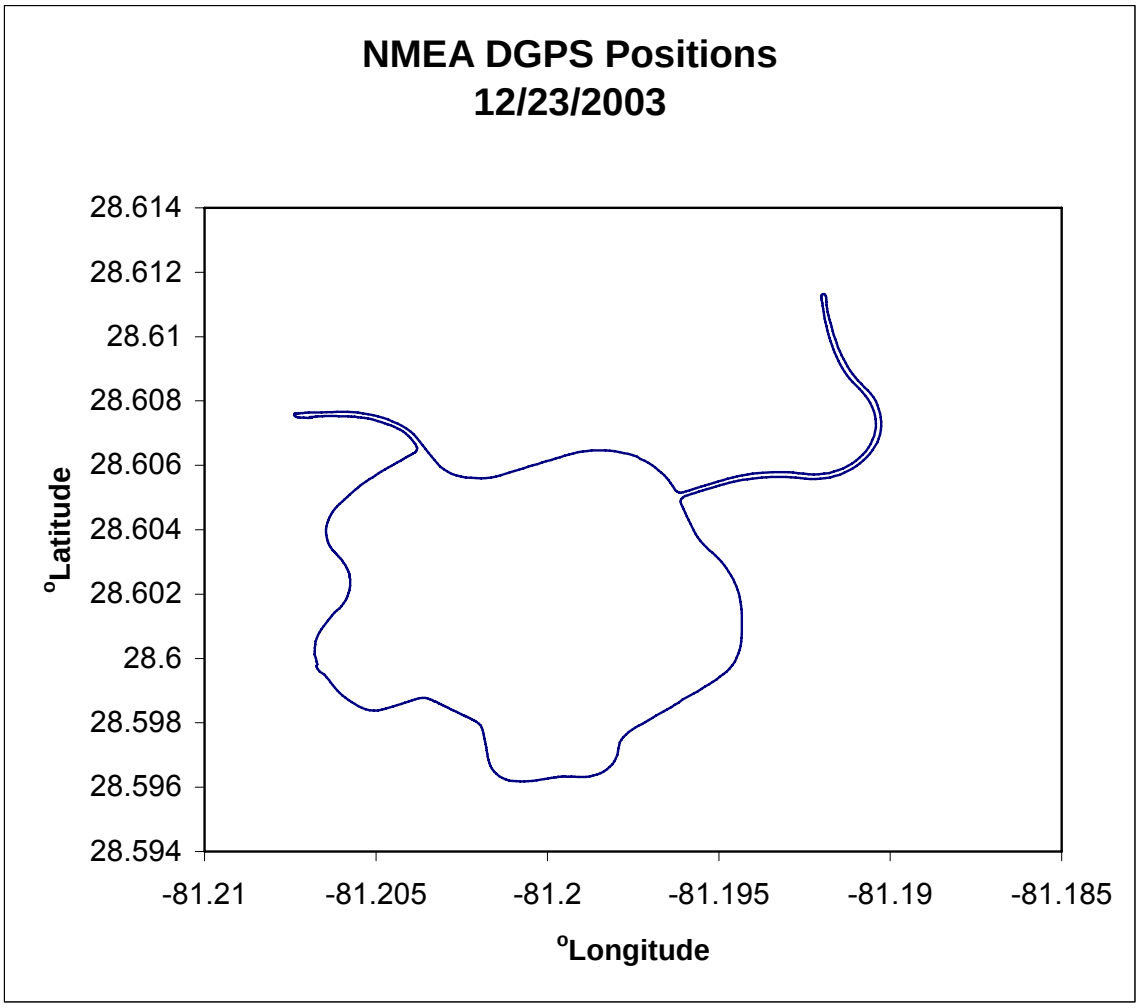

Figure 32 – Real Time DGPS with SA Off

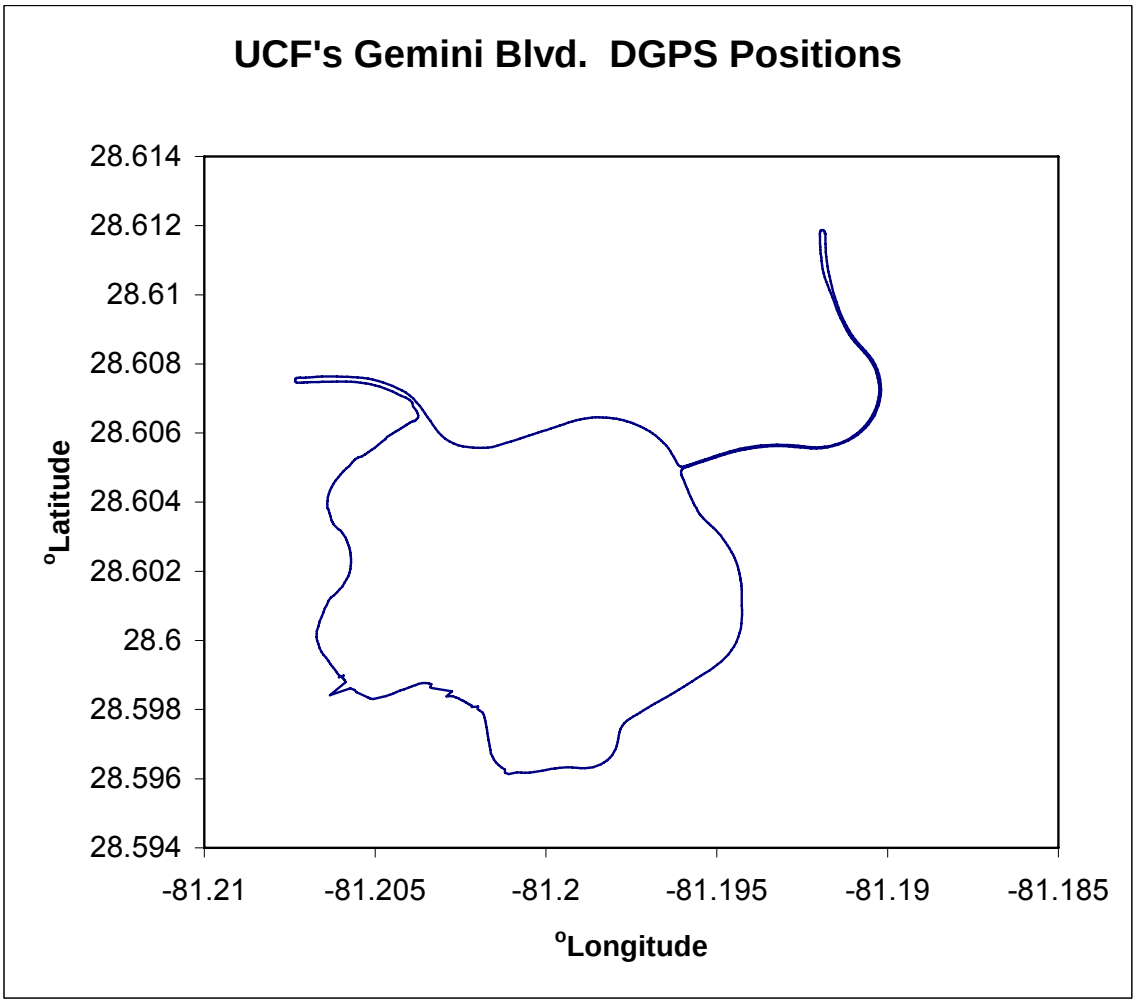

Figure 33 – Post Processed DGPS with SA On

3.1.3.6 System Accuracy

The system accuracy can be obtained by determining the effect of the previous errors on the 3D accuracy of points in the camera coordinate system and then analyzing the effect of errors in the camera to world coordinate transformation.

For a set of matching points with coordinates $[U_1, V_1]^T$ and $[U_2, V_2]^T$ in the left and right images respectively the 3D location is obtained by finding the closest point of approach of the respective 3D vectors formed by the ray from the U,V point to the focal point. Noise in the U,V points will affect the computed closest point of approach.

Going through all our error sources we summarize their contribution/effect on ΔV and ΔU in Table 5.

Table 5 – Pixel Error Sources

Source	**Pixel Error**	**Note**
Optics	0.5	Section 3.1.3.1.1
Vibration	2	Section 3.1.3.1.1
Bayer Filter	2	Section 3.1.3.1.2
Rectification	0.2	Section 3.1.2
Segmentation	1.5	Section 3.1.3.3.3
Stereo Disparity	1	Section 3.1.3.3.2 (Does not affect the pixel position, only the 3D triangulation)

The 3D error ΔP is determined by performing a Monte-Carlo computation using Gaussian distributed U,V. Assuming a grid of points as shown in Figure 34 is projected as shown in Figure 35 and the 3D position is triangulated with a pixel position error of 3.2 pixels and with a disparity error of 1 pixel we have an error in the camera coordinate system as a function of distance as shown in Figure 36.

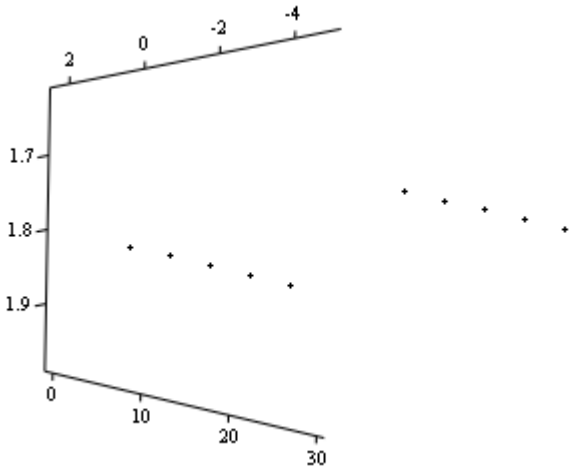

Figure 34 – Grid of Road Boundary Points

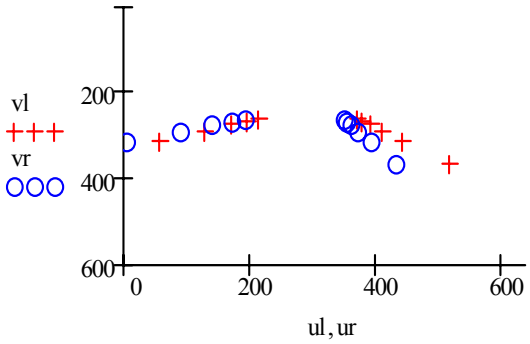

Figure 35 – Projection of Road Boundary Points into Left and Right Camera Image Planes

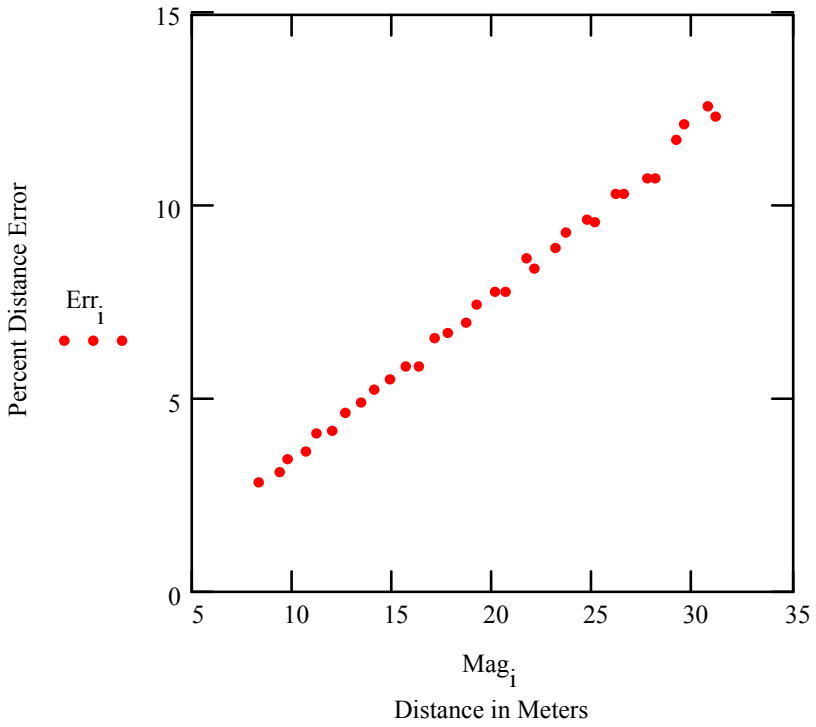

Figure 36 – Inverse Projection Error of Road Grid Points as a Percent of the Grid Point Distance

The transformation matrix from camera coordinates to world coordinates is:

$$\mathbf{R}_C^W = \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T\mathbf{T} \end{bmatrix} \tag{23}$$

where,

$$\mathbf{R} = \text{Roll}(\phi)\,\text{Pitch}(\theta)\,\text{Yaw}(\psi)$$
$$\mathbf{T} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \tag{24}$$

and the Roll, Pitch and Yaw matrices are computed using the Rodrigues[47] formula for rotation about a vector,

$$\begin{aligned}
\mathbf{Roll}(\phi) &= \mathbf{I} + \tilde{\mathrm{w}}(\mathbf{n}_y)\sin(\phi) + \tilde{\mathrm{w}}(\mathbf{n}_y)^2(1-\cos(\phi)) \\
\mathbf{Pitch}(\theta) &= \mathbf{I} + \tilde{\mathrm{w}}(\mathbf{n}_x)\sin(\theta) + \tilde{\mathrm{w}}(\mathbf{n}_x)^2(1-\cos(\theta)) \\
\mathbf{Yaw}(\psi) &= \mathbf{I} + \tilde{\mathrm{w}}(\mathbf{n}_z)\sin(\psi) + \tilde{\mathrm{w}}(\mathbf{n}_z)^2(1-\cos(\psi))
\end{aligned} \tag{25}$$

and n_x, n_y, n_z are the axis unit vectors and $\tilde{w}(\mathbf{x})$ is the anti-symmetric matrix function.

$$\tilde{w}(\mathbf{x}) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}, \ \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{26}$$

Assuming we are located at 0 degrees latitude and 0 degrees longitude while driving east at sea level with no pitch or roll in the vehicle; we have, using small angle approximations:

$$\mathbf{R}' = \begin{bmatrix} 1 & -\phi & \psi \\ \phi & 1 & -\theta \\ -\psi & \theta & 1 \end{bmatrix} \tag{27}$$

The translation vector T becomes the vector from the left camera to the ground point directly below the GPS sensor, expressed in the camera coordinate system; and corrupted by the GPS measurement noise.

$$\mathbf{T} = \begin{bmatrix} 0.46 + \sigma GPS_{HOR} \\ 1.8 + \sigma GPS_{ALT} \\ 0 + \sigma GPS_{VERT} \end{bmatrix} \tag{28}$$

The overall system error can be expressed as the expected value of the absolute difference between world coordinate position of the road grid points and the inverse projected road grid points corrupted by pixel noise, attitude measurement noise and GPS measurement noise.

$$
\begin{aligned}
&E\left\{\left|\begin{bmatrix}\mathbf{R}'^{T} & -\mathbf{R}'^{T}\mathbf{T}'\end{bmatrix}\mathbf{X}' - \begin{bmatrix}\mathbf{R}^{T} & -\mathbf{R}^{T}\mathbf{T}\end{bmatrix}\mathbf{X}\right|\right\} \\
&\mathbf{X}' = \mathbf{X} + \Delta\mathbf{X} \\
&\mathbf{R}' = \mathbf{R} + \Delta\mathbf{R} \\
&\mathbf{T}' = \mathbf{T} + \Delta\mathbf{T}
\end{aligned}
\tag{29}
$$

Performing a Monte Carlo run using Gaussian distributions for φ, θ, ψ with a mean of zero and a standard deviation of 1 degree and Gaussian distributed GPS standard deviation errors of 0.3 meters horizontal, 0.6 meters in altitude with zero mean gives us the results shown in Figure 37, where $\sigma\mathbf{W}$ is the error in meters for the UTM X,Y ,altitude distance computed using an inverse projection and $\sigma\mathbf{W}_f$ is error in meters for the inverse projection which assumes a flat world (zero altitude).

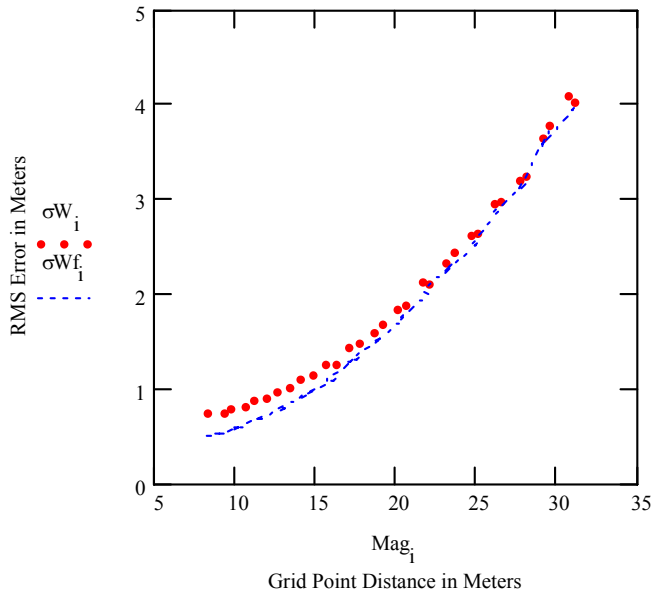

Figure 37 – Error in UTM Distance .vs. Grid Point Distance in Meters

The preceding discussions are summarized in the error block diagram of Figure 38. Note that the world coordinate position error is directly affected by the position of the points in the camera coordinate system. Creating a first order approximation of Equation 29 we get:

$$E\left\{\left\|\left[\Delta\mathbf{R}^T \quad -\left(\Delta\mathbf{R}^T\mathbf{T}+\mathbf{R}^T\Delta\mathbf{T}\right)\right]\mathbf{X}+\left[\mathbf{R}^T \quad -\mathbf{R}^T\mathbf{T}\right]\Delta\mathbf{X}\right\|\right\} \tag{30}$$

which shows for a fixed R and T the position error is directly affected by the true position of the points in the camera coordinate system and the triangulation error obtained when inverse projecting the 2D camera point projections which have been corrupted pixel noise.

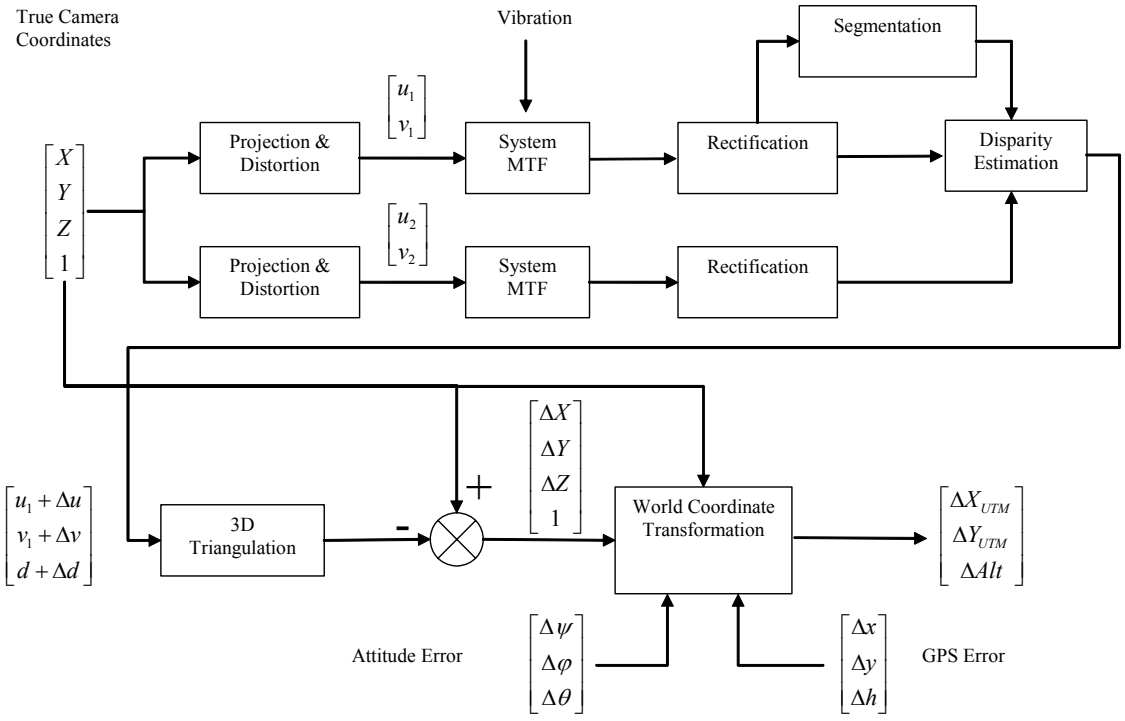

Figure 38 – Error Block Diagram

Revisiting Figure 37 we note that if we assume a flat world by ignoring the GPS altitude measurements (trace $\mathbf{W}_f$) we still don't get appreciably better results. However, if we decompose the position errors and look at the individual coordinates as shown by Figure 39 we see that the primary source of error is due to the Z world coordinate (this translates to Eastings in the UTM system). We see that the Z component begins diverging significantly from the other components beyond 10 meters. If our goal were to achieve just a couple of feet of error then we know that we need to perform our road modeling giving greater weight to road boundary points that are closer .

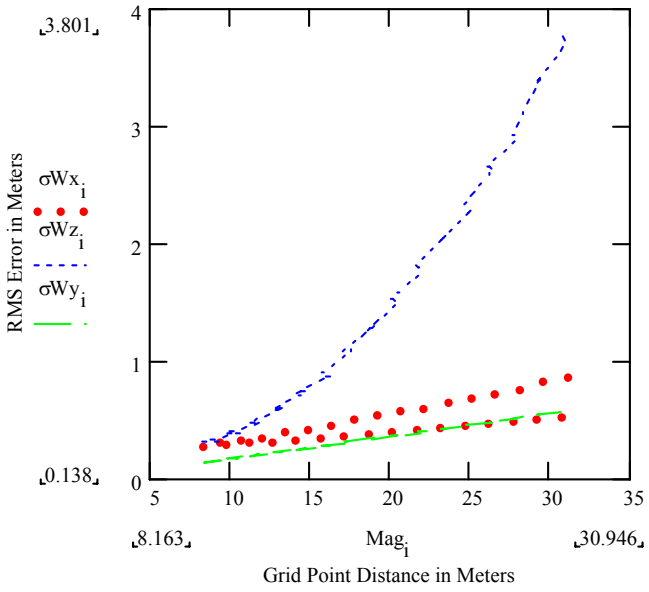

Figure 39 – Position Error .vs. Camera in Meters for the Individual UTM Coordinates

(X=Northings, Z=Eastings, Y=Altitude)

3.2 Hardware Description

The system hardware consists of a mobile platform and a base station. The mobile platform is a station wagon carrying various sensors and recording equipment. The base station is fixed and transmits RTCM [48] correction messages via the internet which is picked up by a cellular modem on the station wagon. This communications link is only active when operating in DGPS mode. See Figure **40** for a system level view of the mobile platform and base station.

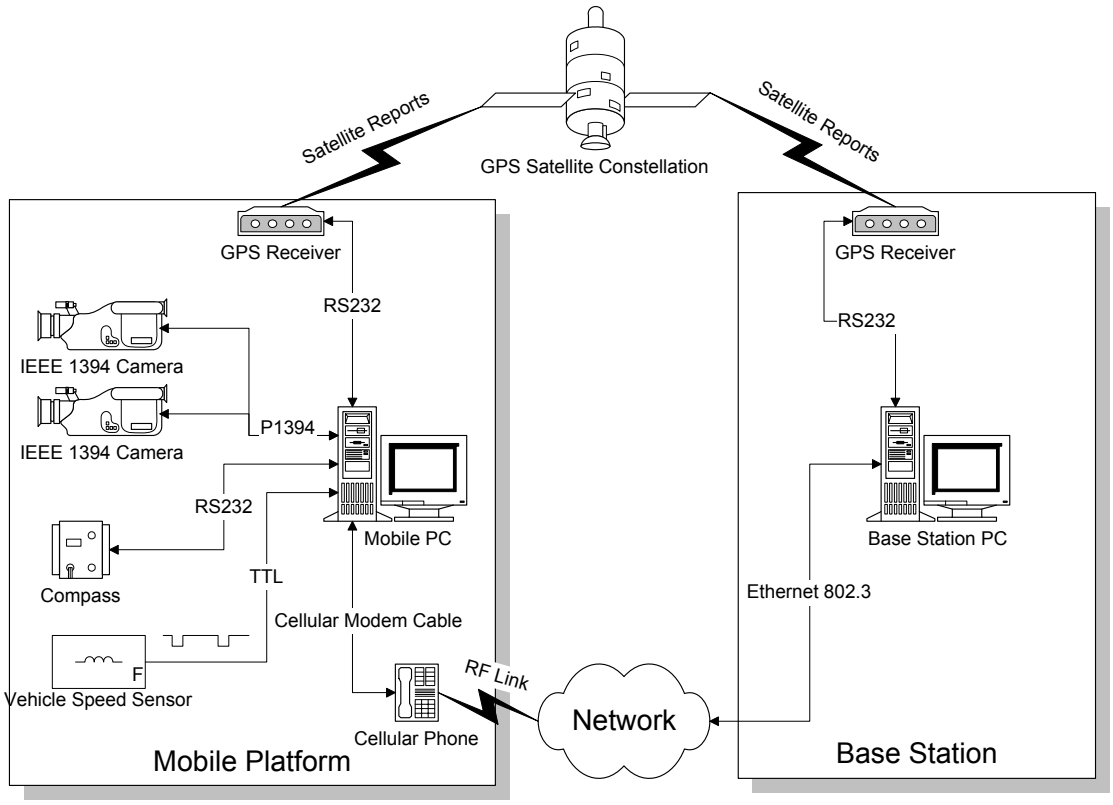

Figure 40 – System Level view of Sensor Capture Hardware

3.2.1 Mobile Platform

A 1984 Ford LTD station wagon is used to carry the video and sensor capture hardware. The station wagon has been previously used by the UCF Driver Simulation Laboratory to collect field data for simulator validation studies and comes equipped with a Hall Effect Vehicle Speed Sensor (VSS) mounted on the transmissions speedometer output shaft. See Figure 41.

Figure 41 – 1984 FORD LTD Station Wagon Used for Mobile Platform, Vehicle Speed Sensor Attached to Transmission Shown in Inset.

Two video cameras, a GPS antenna and a solid state compass are mounted on a specially constructed carrier as shown in Figure 42. The carrier attaches to the roof of the vehicle thru the use of four Quick-N-Easy gutter clamps and contains a stiff aluminum U-beam which provides a rigid surface for attaching cameras and sensors. The U-beam attaches to the carrier using four rigid rubber shock mounts.

The rest of the sensor capture hardware is located in the cargo section of the station wagon with the exception of the operator station in the rear passenger seat which contains the monitor, mouse and keyboard. See Figure 43.

Figure 42 – Roof Mounted Carrier for Cameras, GPS Antenna and Compass

Figure 43 – Rear shot of Station Wagon Cargo Area Showing Mobile PC and Other Equipment.

A dual Pentium-III PC runs the capture software. Power is provided by a 300 Watt power inverter attached to a 12 Volt Marine Battery.

The PC contains an internally mounted disk array consisting of two 9 GB SCSI drives striped together as one logical drive using the Windows Disk Management utility. This disk capacity allows 90 minutes of data collection at the sensor sampling rates used.

3.2.2 Cameras

The video cameras are equipped with 4.8mm wide field of view lenses. The combination of the camera's type 1/3 CCD and the lens results in a field of view of 53° horizontal by 41° vertical. A lens hood with an ultraviolet filter are used to reduce glare from the sun and prevent ultraviolet scatter from the sky casting a bluish tinge to the imagery.

The cameras are mounted on pant/tilt tripod heads attached to the aluminum U-beam. The pan/tilt heads have full-three way action with each axis movement controlled by separate locking handles. Quick release plates permit the cameras to be removed from the tripod heads without disturbing their adjustments. Earlier attempts to use low cost CCTV camera mounts demonstrated the need for being able to precisely adjust the cameras in all three axis and to be able to remove and replace the cameras without affecting the camera orientation.

3.2.3 Sensor Capture Operation

During sensor capture operations, the video cameras are synchronized by driving their external sync inputs with a TTL clock signal from the signal generator. The cameras are controlled via their 1394 interface to the PC. Streaming video is captured, time stamped and saved to the RAID.

The GPS receives satellite measurements via its antenna and computes a position which it reports via an RS232 serial interface to the PC. If the GPS receives RTCM correction messages via its second serial port it reports DGPS corrected position messages. The RTCM messages are forwarded from the internet to the GPS by the PC. The PC connects to the internet via a cellular modem installed in a PCMCIA adaptor card and connected to the cellular phone which uses an external roof mounted antenna.

The electronic compass provides a continuous stream of angle measurements and raw acceleremoter and magnetometer measurements to the PC via an RS232 serial port. The VSS pulse signals are passed through an optocoupler, filtered and sent to the clock counter card's screw terminal connector.

Measurements from the GPS, compass and VSS are timestamped and saved to the system drive. See Figure 44 for the system interconnect diagram of the Mobile Platform.

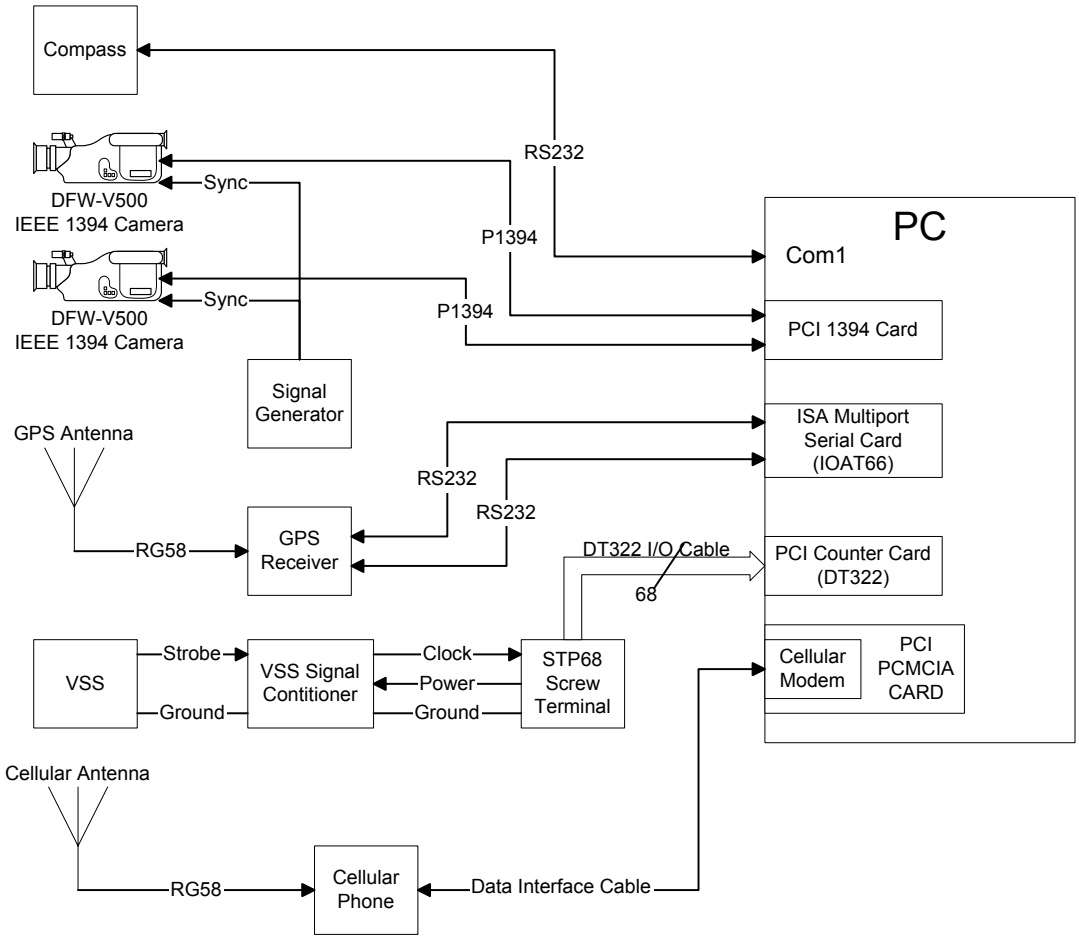

Figure 44 – System Interconnect Diagram for Mobile Platform

3.2.3.1 Other Hardware Considerations

3.2.3.1.1 Power

System power is provided by a 12 Volt marine battery with a reserve capacity of 140 minutes for a 25 amp load. The battery supplied 12 VDC is converted to 120 VAC by a dual outlet power inverter rated at 300W continuous output.

One of the inverter's outlets provides power directly to the PC; the other outlet is connected to a power strip that distributes power to the monitor, signal generator, and

various AC/DC power supplies. The AC/DC power supplies provide power to the GPS, compass and the cooling fan for the Power Inverter's heat sink.

Although laboratory measurements of current draw indicated an expected battery life of 150 minutes, field results showed a battery life of only 45 minutes. The laboratory equipment configuration tested consisted of the mobile platform's PC with an external 4 drive RAID and a 17 inch monitor. The RAID was stressed during the measurement using Adaptec's SCSIBench32 drive benchmarking utility. Power usage by the other equipment (i.e. GPS, compass, etc) was negligible. Replacing the 15" CRT used in the field with a 14" flat panel display extended the battery life from 45mins to over 2 hours.

3.2.3.1.2 AC Line Filter

The signal generator proved to be sensitive to noise on its power line. A Constant voltage transformer was used to condition the Power Inverter's pseudo-sine wave AC output prior to routing it to the signal generator.

3.2.3.1.3 Cooling

Several hardware changes were implemented to overcome cooling problems encountered in the operation of the hardware.

- Mounting the Power inverter on a fan cooled aluminum heat sink.
- Adding hard drive cooling fans to Internal RAID drives.
- Adding additional cooling fans to the PC case.

3.2.3.1.4 Base Station

The base station consists of a GPS receiver, antenna and PC. The GPS receiver is mounted on the roof of UCF's Engineering I building inside a NEMA enclosure located next to an antenna mast (see Figure 45). The GPS antenna is mounted at the top of the mast. The GPS receivers serial channels and remote power on signal for the GPS Power Supply are routed to the Senior Design Lab in the floor below via a 30 meter (~100 foot) ruggedized cable. The controlling PC and remote power on switch are contained within a metal storage cabinet.

One of the serial channels is used for controlling the GPS receiver and the other is used for receiving a continuous stream of RTCM correction messages that are forwarded to the campus network. The remote power on signal controls a solid state relay mounted in the NEMA enclosure on the roof. When switched on the relay connects line power to the GPS's power supply.

Figure 45 – GPS Antenna on Mast

(Top right inset shows closeup of GPS Antenna with its choke ring, bottom right inset shows GPS receiver in NEMA enclosure)

3.2.3.1.5 Equipment List

Table 6 and Table 7 contain parts list of the equipment used in the Mobile Platform and Base Station respectively.

Table 6 – Mobile Platform Equipment List

Item	Description	Notes
1	VSS	Magnetic Sensor Corporation 413031-10 (2 pulses/rev)
2	VSS Signal Conditioner	Custom Built Optocoupler w/ Noise Filter
3	Solid State Compass	MicroStrain 3DM
4	GPS Card	Novatel OEM 3151RM
5	Power Conditioner	SOLA Constant Voltage Transformer 23-13-030-2
6	Signal Generator	Feedback
7	PC	Asus P2BDS Dual Pentium III Motherboard
8	PCI Counter Timer	Data Translation DT322
9	1394 PCI Adaptor	Generic TI Chipset
10	ISA Multiport Serial Card	Boca IOAT66
11	Data Interface Cable	3COM ERIC-2
12	PCMCIA Adaptor	Generic
13	Cellular Modem	Megahertz 3CXM556
14	Cell Phone	Ericsson KH668
15	RAID	Striped 9GB SCSI Drives
16	4.8 mm lens C-Mount	Rainbow S48WI
17	Stereo Cameras	Sony DFW-V500
18	Camera Mounts	Bogen/Manfrotto 3030
19	Mobile GPS Antenna	Novatel Antenna 511
20	800Mhz Cellular Antenna	Radio Shack 17-318
21	300 Watt DC Inverter	Wang Tech
22	Marine Battery	Interstate Battery SRM-24

Table 7 – Base Station Equipment List

Item	Description	Notes
1	PC	Generic PC with dual serial ports, ethernet card, PS/2 mouse & keyboard.
2	NEMA Enclosure w/ 120VAC solid state relay and power supply.	Standard 12x10x6 Size, 25W +5V/+12V/-12V power supply.
3	GPS Card	Novatel OEM 3151RM
4	Relay Control Box	Custom Built
5	GPS Antenna	Novatel GPS-701 with choke ring

3.3 Software Description

This section describes the software used on the mobile platform to capture the sensor data and the software used on the base station to provide DGPS corrections over IP to the mobile platform. Windows 2000 is used as the operating system on both the mobile platform PC and the base station PC.

3.3.1 Sensor Capture

3.3.1.1 Video Capture

Video capture is performed using a modified version of the CMU Video Capture Code [49] to generate time stamped stereo AVI files. The CMU code is written in Microsoft Visual C++ and was modified to work with two cameras simultaneously and to capture both camera frames and write them side by side into an AVI file as shown in Figure 46.

During video capture, each frame is time stamped by bit encoding the time in milliseconds since system startup into the first 32 pixels (1 bit per pixel) of the first line of video from each camera. The video is captured in its native YUV422 [31] format and saved uncompressed into the AVI file. Two types of video sequences are captured, calibration video sequences of test patterns for use by the stereo calibration software, and road video sequences for use by the road extraction software.

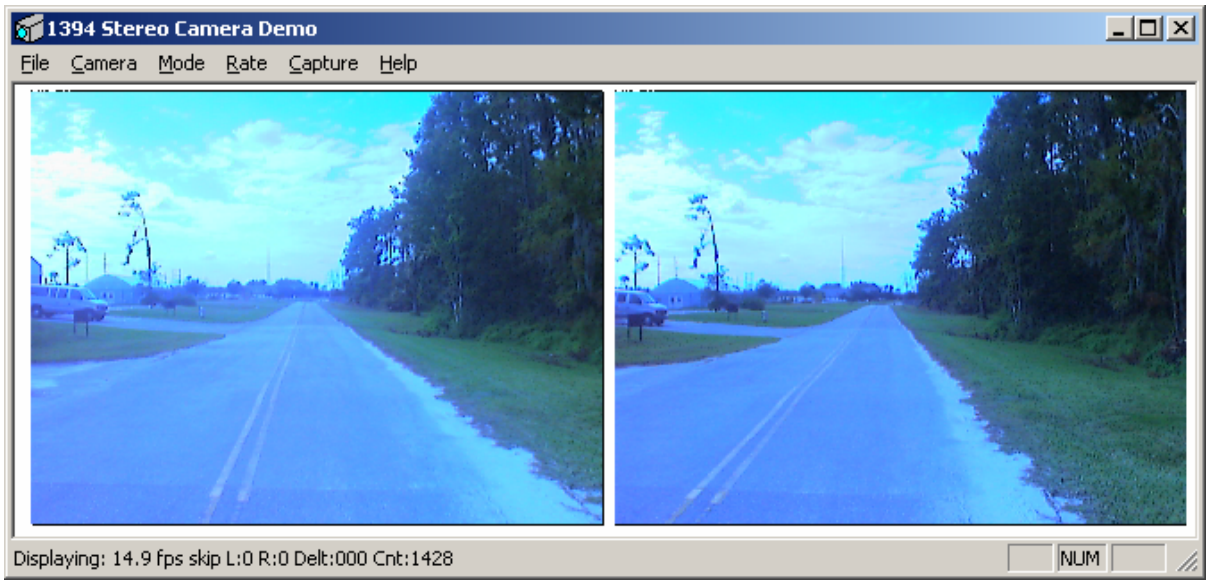

Figure 46 – Screenshot of Stereo Video Capture Software

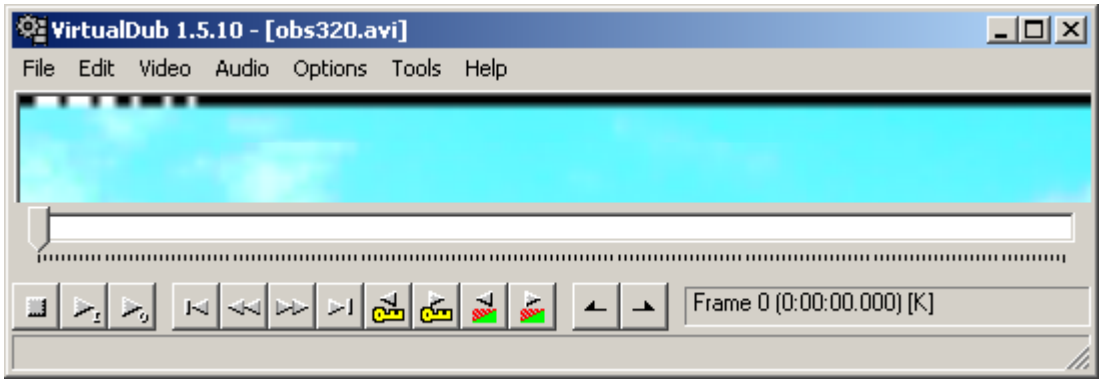

Figure 47 – Closeup of First Video Line Showing Timestamp Bit Encoded Into Pixels

3.3.1.1.1 Stereo Calibration Video

Stereo image pairs for the camera calibration software are generated using VirtualDub 1.5.10 [50] to select frames from the calibration video which show the calibration patterns in several orientations and Microsoft Photo Editor™ 3.0 to extract the left and right parts of the frame.

3.3.1.1.2 Stereo Road Video

The road video sequences captured in YUV422 format must be edited and converted to an uncompressed RGB format for use by the road extraction software. This is performed by using VirtualDub together with AviSynth 2.5 [51]. AviSynth works as a frame server, it uses an AviSynth Script (AVS) text file (see example below) which functions as a script to instruct the frame server what conversions need to be performed. For example, if Windows Media Player™ opens the file output12.avs shown below, AviSynth will convert the video from YUV422 to an RGB format on a frame by frame basis and pass it to the media player; the player sees an RGB formatted file. If VirtualDub is used to open the AVS file (see Figure 48 for a screenshot), it will also see an RGB formatted file. Any editing performed on the file and saved to an output file will result in an RGB format output file. The conversion from YUV422 to RGB performed by AviSynth also converts the timestamp bit encoded into the pixels from their sixteen bit values of 0xFF80 or 0x0080 for bit value “1” or “0” into 24 bit RGB triplets of 0xFFFFFF or 0x000000.

Example AVS file: Output12.avs

```
AVISource("I:\nov13run\output12.avi")
ConvertToRGB24()
```

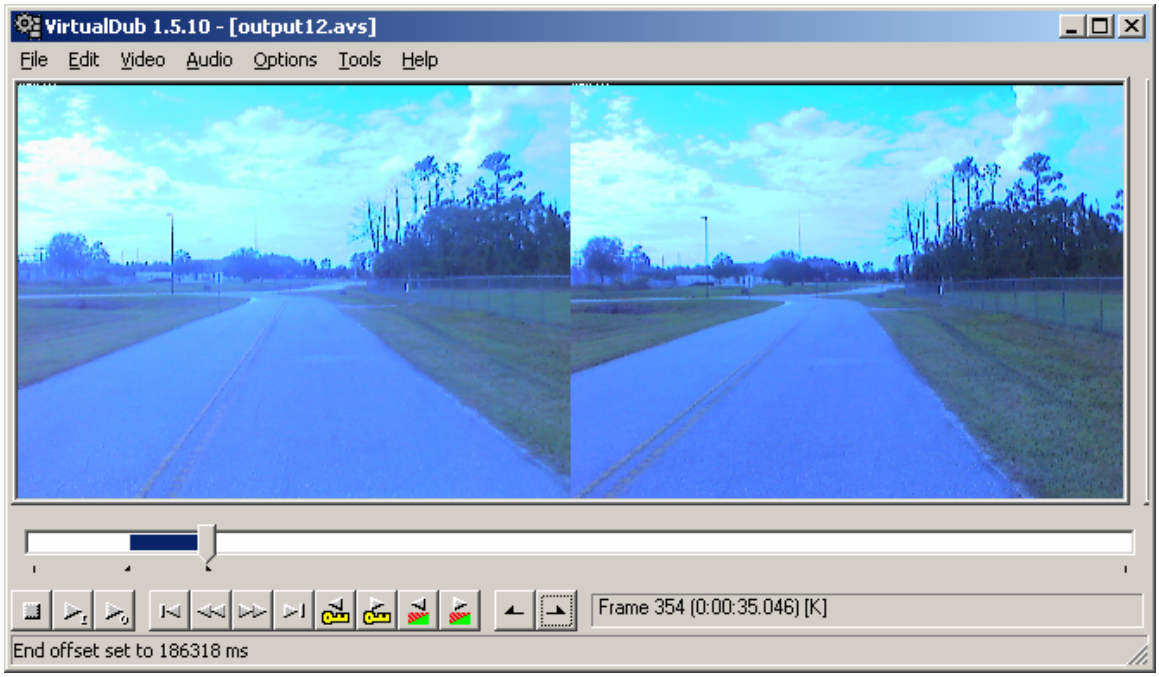

Figure 48 – Screenshot of VirtualDub

3.3.2 Instrumentation Capture

SensorCap (see Figure 50) is a Visual C++ application that was developed to perform instrumentation capture. SensorCap is run concurrently with the video capture software to capture the GPS, odometer and compass measurements.

SensorCap uses asynchronous threads to read the GPS and compass measurements. During capture, each thread use the windows messaging interface to send messages to a file writing thread. For the odometer, SensorCap creates a named pipe connection to an odometer server program and creates a callback routine that handles any messages received from the server. The callback routine uses the windows messaging interface to forward odometer messages to the file writing thread during capture. All messages received from the sensors are time stamped before being written to the capture file.

The odometer server is a small Visual C++ server application that was written to interface with the Data Translation Timer/Counter Card. It continually polls the DT330 card for the count of odometer pulses received, timestamps it using the Windows Win32 Library precision timer function QueryPerformanceCounter [52] and forwards the data to any clients that may have connected its named pipe. The odometer server can also be compiled to function as an odometer emulator on PCs without the DT330 that are being used for software development.

Figure 49 shows an example of a sensor capture file. Sensor data is captured in a text file with one line per sensor reading. Each line is composed of comma separated fields, with the first field representing the timestamp. The second filed contains a token identifying the sensor data. The format of the rest of the data fields is shown in Table 8.

```
2338095,$COMPR,6.230393,0.003931,3.349185
2338110,$ODOM,0,1405721893710
2338106,$COMPG,2436,1714,690,1938,1961,3241
2338111,$GPGGA,190743.0,2835.4933367,N,08111.4150776,W,2,08,1.0,17.34,M,,,2,0000*20
```

Figure 49 – Example of Sensor Capture File

Table 8 – Format of Sensor Data

Token	Data Format
$COMPG	16 bit counts for Raw Magnetometer x,y,z and Raw Accelerometer x,y,z
$COMPG	Roll, Pitch, Yaw in Radians
$GPGGA	Global positioning system fixed data:
	UTC Time hhmmss.sss
	Latitude ddmm.mmmm
	N/S Indicator N = North, S = South
	Longitude dddmm.mmmm
	E/W Indicator E = East, W = West
	Position Fix 0 = Invalid, 1 = Valid GPS, 2 = Valid DGPS
	Satellites being used (0-12)
	Horizontal dilution of precision
	Altitude in meters according to WGS-84 ellipsoid
	Altitude Units M = Meters
	Geoid seperation according to WGS-84 ellipsoid (Not Used)
	Geoid seperation units M = Meters (Not Used)
	Age of DGPS data in seconds (Only Valid when Position Fix = 2)
	DGPS Station ID (Only Valid when Position Fix = 2)
	Checksum
$ODOM	Odometer Reading, System Clock

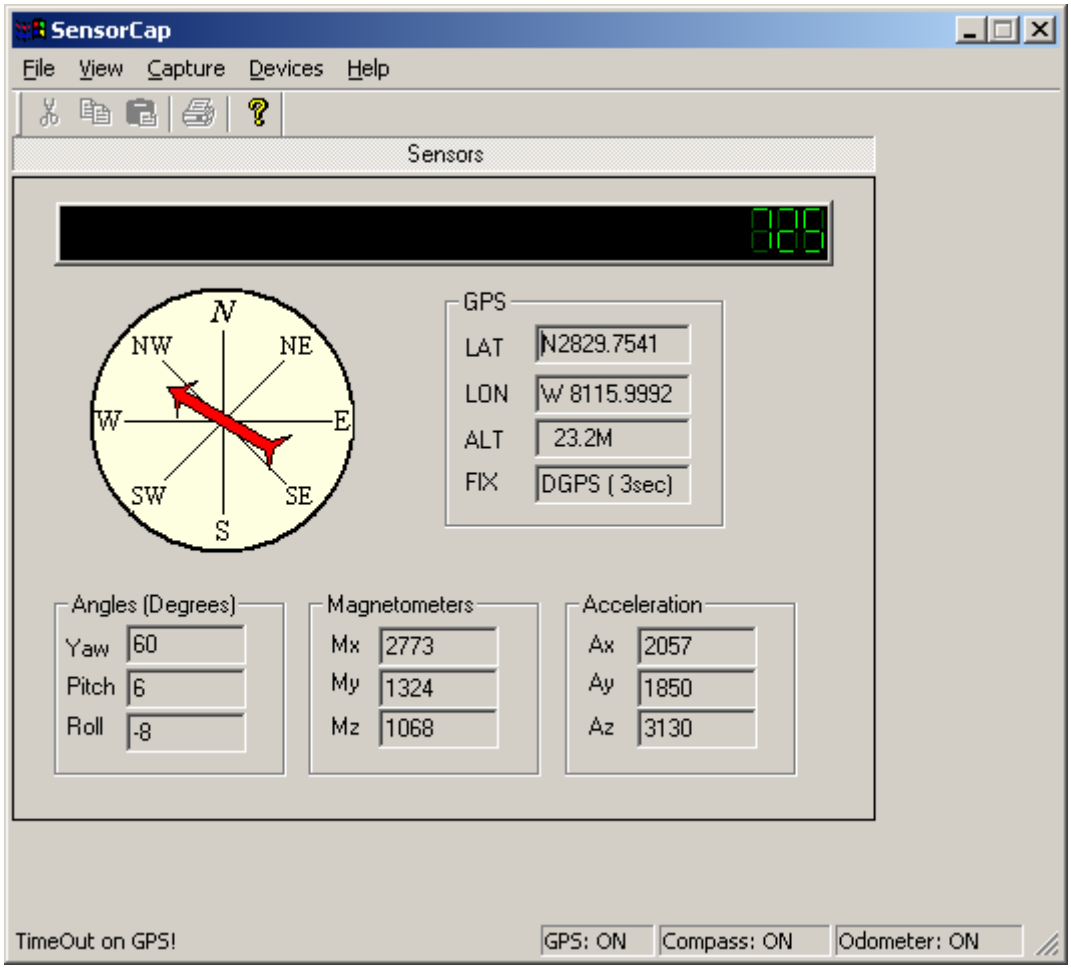

Figure 50 – Screenshot of SensorCap

3.3.2.1 Remote GPS Section

The GPS card is configured using the manufacturer's software, GPSolution [53]. The GPS retains its configuration even if its power is cycled. The configuration report for the remote GPS is shown in the following figure:

```
$RCCA,COM1,38400,N,8,1,N,OFF*36
$RCCA,COM2,38400,N,8,1,N,OFF*35
$RCCA,COM1_DTR,HIGH*70
$RCCA,COM2_DTR,HIGH*73
$RCCA,COM1_RTS,HIGH*67
$RCCA,COM2_RTS,HIGH*64
$RCCA,UNDULATION,TABLE*56
$RCCA,DATUM,WGS84*15
$RCCA,USERDATUM,6378137.000,298.257223563,0.000,0.000,0.000,0.000,0.000,0.000,0.000*6A
$RCCA,SETNAV,DISABLE*5C
$RCCA,MAGVAR,0.000*33
$RCCA,DYNAMICS,HIGH*1B
$RCCA,UNASSIGNALL*64
$RCCA,ACCEPT,COM1,COMMANDS*5B
$RCCA,ACCEPT,COM2,RTCM*44
$RCCA,UNLOCKOUTALL*20
$RCCA,RESETHEALTHALL*37
$RCCA,UNFIX*73
$RCCA,RTCMRULE,6CR*32
$RCCA,RTCM16T,*48
$RCCA,CSMOOTH,20.00*7E
$RCCA,ECUTOFF,0.00*45
$RCCA,FREQUENCY_OUT,DISABLE*12
$RCCA,CLOCKADJUST,ENABLE*47
$RCCA,MESSAGES,ALL,ON*67
$RCCA,SETCHAN,12*56
$RCCA,DGPSTIMEOUT,60,120*51
$RCCA,SETDGPSID,ALL*1D
$RCCA,LOG,COM2,GPGGA,ONTIME,1.00,0.00*67
$RCCA,LOG,COM2,GPGSV,ONCHANGED,0.00,0.00*32
$RCCA,LOG,COM2,GPGSA,ONCHANGED,0.00,0.00*25
$RCCA,LOG,COM1,GPGGA,ONTIME,1.00,0.00*64
```

Figure 51 – Configuration Report for Remote GPS Card

As Figure 51 shows the GPS card powers up with its serial ports set to a 38,400 baudrate, it will accept RTCM correction messages on its second serial port and will send out global positioning fixed data messages on both its ports once a second. Information on the other configuration setting can be obtained from [48].

A modified version of WinDGPS [54] is used to relay any RTCM messages received over the internet from the DGPS server to the second serial port of the GPS card (see Figure 52). WinDGPS is written in Delphi was modified (using Borland Delphi 7 Personal Edition) to use up to eight serial ports and to support baudrates up to 38400. Windows 2000 Virtual Private Networking is used to set up the internet connection over a cellular modem. The default modem configuration string for the PCMCIA cellular modem card is modified to force a connection at 4800 baud instead of the default 14400 baud.

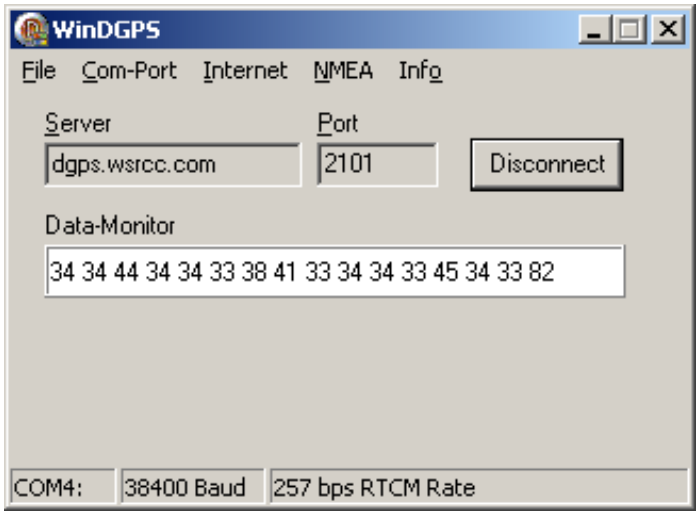

Figure 52 – Screenshot of WinDGPS

3.3.2.2 Base Station

The base station sends out RTCM104 messages in binary format every 2 seconds over the internet. GPSSolution (see Figure **54**) was used to configure the GPS card as shown in the following figure:

$RCCA,COM1,38400,N,8,1,CTS,OFF*3C
$RCCA,COM2,38400,N,8,1,CTS,OFF*3F
$RCCA,COM1_DTR,HIGH*70
$RCCA,COM2_DTR,HIGH*73
$RCCA,COM1_RTS,HIGH*67
$RCCA,COM2_RTS,HIGH*64
$RCCA,UNDULATION,TABLE*56
$RCCA,DATUM,WGS84*15
$RCCA,USERDATUM,6378137.000,298.257223563,0.000,0.000,0.000,0.000,0.000,0.000,0.000*6A
$RCCA,SETNAV,DISABLE*5C
$RCCA,MAGVAR,0.000*33
$RCCA,DYNAMICS,HIGH*1B
$RCCA,UNASSIGNALL*64
$RCCA,ACCEPT,COM1,COMMANDS*5B
$RCCA,ACCEPT,COM2,COMMANDS*58
$RCCA,UNLOCKOUTALL*20
$RCCA,RESETHEALTHALL*37
$RCCA,FIX,POSITION,28.60151555,-81.19857042,41.398,0*58
$RCCA,RTCMRULE,6CR*32
$RCCA,RTCM16T,*48
$RCCA,CSMOOTH,20.00*7E
$RCCA,ECUTOFF,0.00*45
$RCCA,FREQUENCY_OUT,DISABLE*12
$RCCA,CLOCKADJUST,ENABLE*47
$RCCA,MESSAGES,ALL,ON*67
$RCCA,SETCHAN,12*56
$RCCA,DGPSTIMEOUT,60,120*51
$RCCA,SETDGPSID,ALL*1D
$RCCA,LOG,COM2,RTCM1,ONTIME,2.00,0.00*0B
$RCCA,LOG,COM1,CTSB,ONTIME,2.00,0.50*32
$RCCA,LOG,COM1,TM1B,ONTIME,30.00,0.00*6A
$RCCA,LOG,COM1,PRTKB,ONTIME,10.00,0.00*5D
$RCCA,LOG,COM1,POSB,ONTIME,1.00,0.00*3C
$RCCA,LOG,COM1,MKPB,ONNEW,0.00,0.00*6E

Figure 53 – Configuration Report of Base Station GPS card

The fixed position was initially obtained by running the GPSCard in a position averaging mode for 48 hours.

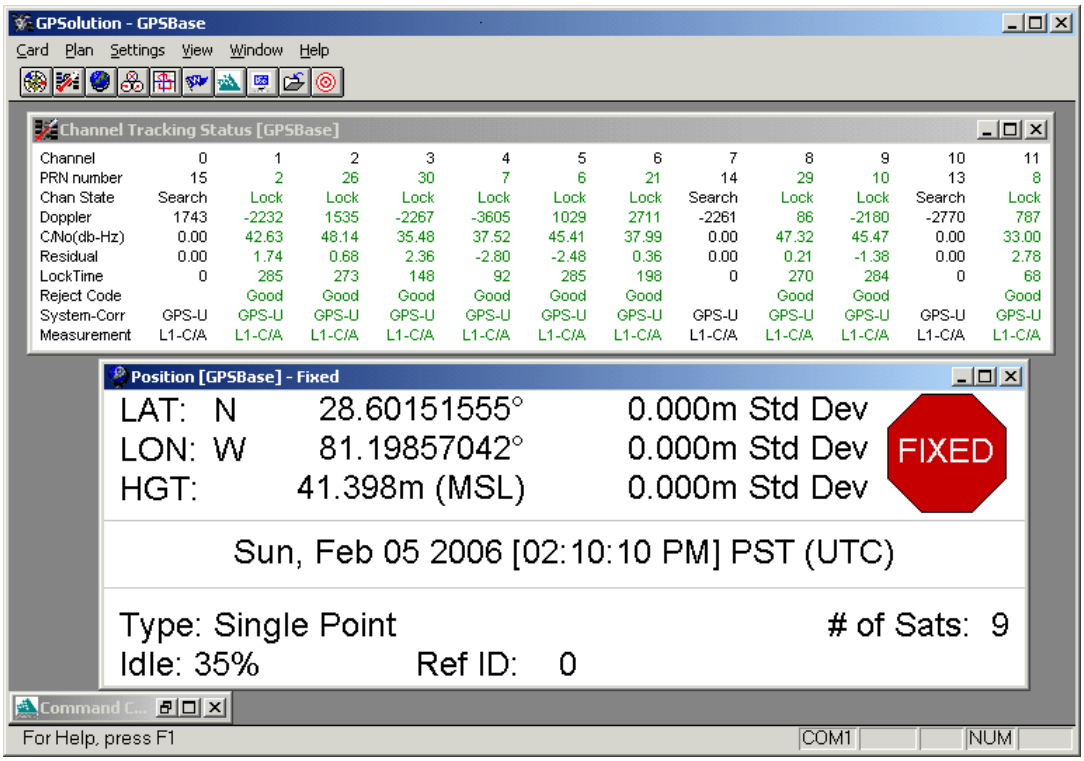

Figure 54- Screenshot of GPSolution

The RTCM messages sent out over the GPS card's second serial port are received by IP->Com [55] (see Figure 55) and then sent out over TCP/IP port 2101.

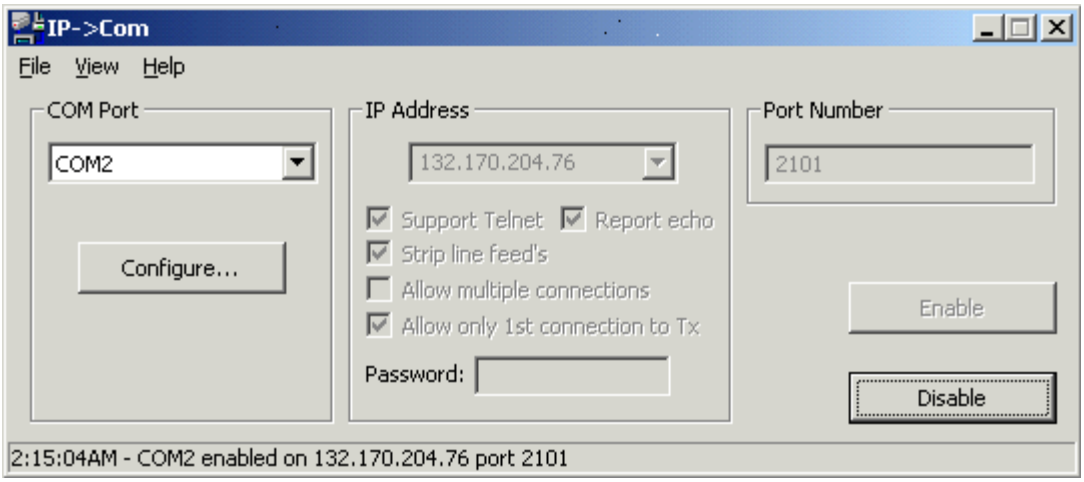

Figure 55 – Screenshot of IP->Com

3.4 Proposed Method

3.4.1 Distortion Correction & Rectification

As discussed in section 3.1.3.1, stereo video captured using wide field of view lenses must be distortion corrected and rectified as a necessary first step. Calibration is performed manually using the Camera Calibration Toolbox developed by Jean-Yves Bouguet [35].

Bouguet's method was largely inspired by the camera calibration techniques developed by Zhengyou Zhang [56]. The method uses a checkerboard pattern with a known spacing and size. Given an ideal pinhole camera and m (at least three) images of the pattern for different angular orientations with respect to the camera; the projection of the pattern's corner points into image space can be expressed as:

$$s\,\tilde{\mathbf{o}} = \mathbf{A}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\tilde{\mathbf{O}}, \quad with\ \mathbf{A} = \begin{bmatrix}\alpha & \lambda & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1\end{bmatrix} \tag{31}$$

where $\tilde{\mathbf{O}}$ represents the homogenous 3D coordinates $\begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T$ of a corner point, $\tilde{\mathbf{o}}$ represents the homogenous 2D projection of the point into the image space $\begin{bmatrix} u & v & 1 \end{bmatrix}^T$, **R** and **t** are the rotation matrix and translation vector, which relate the world coordinate system to the camera coordinate system; and s is an arbitrary scale factor. The expression $\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}$ is also known as the extrinsic parameter matrix.

A represents the intrinsic parameters of the camera, with $\begin{pmatrix} u_0 & v_0 \end{pmatrix}$ the coordinates of the center point of the imaging plane, α and β are scaling factors for the *u* and *v* axes and λ represents the skew between the image axis.

The model plane is assumed to be lying in the XY plane at Z=0, this leads to rewriting Equation 31 as:

$$s\,\tilde{\mathbf{o}} = \mathbf{H}\,\tilde{\mathbf{O}}, \quad with\ \mathbf{H} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix} \tag{32}$$

$\tilde{\mathbf{O}}$ now represents the reduced homogenous vectors $\begin{bmatrix} X & Y & 1 \end{bmatrix}^T$ and $\mathbf{r}_1$ and $\mathbf{r}_2$ are the first two columns of the rotation matrix **R**.

For each image the homography **H** can be estimated by minimizing the following functional:

$$\min_H \sum_i \left\| \mathbf{o}_i - \hat{\mathbf{o}}_i \right\|^2 \quad where\ \hat{\mathbf{o}}_i = \frac{1}{\mathbf{h}_3^T \tilde{\mathbf{O}}_i} \begin{bmatrix} \mathbf{h}_1^T \tilde{\mathbf{O}}_i \\ \mathbf{h}_2^T \tilde{\mathbf{O}}_i \end{bmatrix} \quad and \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{bmatrix} = \mathbf{H} \tag{33}$$

The measured corner points in image space are represented by $\mathbf{o}_i$. $\tilde{\mathbf{O}}_i$ represents the known 3D world coordinates of the corner points.

The non-linear minimization of Equation 33 is performed using the Levenberg-Marquardt Algorithm [57]. An initial guess for **H** is obtained by rewriting Equation 32 as:

$$\begin{bmatrix} \tilde{\mathbf{O}}_i^T & \mathbf{0}^T & -u\tilde{\mathbf{O}}_i^T \\ \mathbf{0}^T & \tilde{\mathbf{O}}_i^T & -v\tilde{\mathbf{O}}_i^T \end{bmatrix} \mathbf{x} = 0 \quad where\ \mathbf{x} = \begin{bmatrix} \mathbf{h}_1^T \\ \mathbf{h}_2^T \\ \mathbf{h}_3^T \end{bmatrix} \tag{34}$$

then stacking Equation 4 n times (corresponding to all the corner points) so that it is in the form $\mathbf{L}x=\mathbf{0}$, **L** will be a 2n by 9 matrix, and solving for ***x***. The solution [58] of ***x*** is obtained by obtaining the singular value decomposition of $\mathbf{L}^T\mathbf{L} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ and taking **x** as the last column of $\mathbf{V}^T$. Since ***H*** has 6 degrees of freedom, n should be greater than 3, which is satisfied by the simplest 2x2 checkerboard with 9 corners.

Once the **H**'s correponding to the m images have been found, the intrinsic parameters can be found by minimizing the following functional using Levenberg-Marquardt over all m images containing n points each.

$$\sum_{i=1}^{m}\sum_{j=1}^{n}\left\| o_{ij} - \hat{o}\left(\mathbf{A}, \mathbf{R}_i, \mathbf{t}_i, \mathbf{O}_j\right)\right\|^2 \tag{35}$$

The function $\hat{\mathbf{o}}(\mathbf{A},\mathbf{R_j},\mathbf{t_j},\mathbf{O_j})$ is the projection of point $\mathbf{O}_j$ in image i, as given by Equation 32.

An initial guess for A is obtained by using the orthogonality of vanishing points constraint [59]. For $\mathbf{R}_j = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 \end{bmatrix}_j$, estimates for $\mathbf{r}_1$ and $\mathbf{r}_2$ have already been obtained; $\mathbf{r}_3$ can be found by using the orthogonality property of rotation matrices, $\mathbf{R}^T\mathbf{R} = \mathbf{I}$.

Since real cameras have lens distortion, Bouguet initially computes their intrinsic parameters as shown above in Equation 35 and uses this as the initial guess for minimizing the following functional:

$$\sum_{i=1}^{m}\sum_{j=1}^{n}\left\| o_{ij} - \hat{o}\left(\mathbf{A}, k, \mathbf{R}_i, \mathbf{t}_i, \mathbf{O}_j\right)\right\|^2 \tag{36}$$

Where ***k*** represents the distortion coefficients described by the following:

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = \left(1 + k_1 r^2 + k_2 r^4 + k_5 r^6\right)\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 2k_3 xy + k_4\left(r^2 + 2x^2\right) \\ k_3\left(r^2 + 2y^2\right) + 2k_4 xy \end{bmatrix} \tag{37}$$

where x and y are the ideal image plane coordinates, x_d and y_d are the distorted coordinates, $k = \begin{bmatrix} k_1 & k_2 & k_3 & k_4 & k_6 \end{bmatrix}$ and $r = \sqrt{x^2 + y^2}$.

The homogenous pixel coordinates now become:

$$s\,\tilde{\mathbf{o}} = \mathbf{A}\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} \tag{38}$$

Figure 56 shows a mosaic of a typical set of calibration images taken from the viewpoint of the left camera. Since the calibration video sequence captures both cameras there will also exist a corresponding set of calibration images for the right camera. The calibration pattern was positioned vertically at five different angles with respect to the camera image plane and at one tilted back position.

A two step process is performed. First, the calibration for each camera is performed using their corresponding images. This step gives us the intrinsic parameters for each camera. Next a stereo calibration is performed to obtain a stereo calibration file which will enable us to rectify the captured stereo video road sequences and to obtain the 3D position of corresponding image points in these road sequences.

Figure 56 – Mosaic of Calibration Images

Bouguet uses the Gauss-Newton method to perform stereo calibration. An initial estimate of the homography between the left and right camera is made by taking the median of the m left to right homographies.

$$\begin{aligned} \mathbf{R}_{left \to right} &= median(\mathbf{R}_{right,i}\mathbf{R}_{left,i}^{T}; i = 1 \ldots m) \\ \mathbf{t}_{left \to right} &= median(\mathbf{t}_{right} - \mathbf{R}_{right \to left,i}\mathbf{t}_{left}; i = 1 \ldots m) \end{aligned} \tag{39}$$

The intrinsic and extrinsic coordinates of each camera computed previously along with the initial $\mathbf{R}_{\text{left}\rightarrow\text{right}}$ and $\mathbf{t}_{\text{left}\rightarrow\text{right}}$ are formed into a column vector $\mathbf{p}$ as follows:

$$\mathbf{p} = \begin{bmatrix} [\alpha \quad \beta]^T_{left} \\ [u_0 \quad v_0]^T_{left} \\ (\lambda/\beta)_{left} \\ \mathbf{k}^T_{left} \\ [\alpha \quad \beta]^T_{right} \\ [u_0 \quad v_0]^T_{right} \\ (\lambda/\beta)_{right} \\ \mathbf{k}^T_{right} \\ rodrigues(\mathbf{R}_{left\rightarrow right}) \\ \mathbf{t}_{left\rightarrow right} \end{bmatrix} \tag{40}$$

The Rodrigues formula [60] is used to parametize the rotation matrix into a three parameter column vector. The function f($\mathbf{p}$) takes the model points and projects them into each camera. The errors between the projected model points and the measured points for m images with n points each are stacked into an column vector $\mathbf{e}$ that is 4*n*m elements high (2*n*m for the left camera and 2*n*m for the right). The Jacobian of f($\mathbf{p}$) is computed, this represents the sensitivity of each of the 4*n*m elements of the error vector to each variable in the parameter vector. Finally the parameter vector is updated as follows until the maximum iteration limit has been reached or the maximum element of $|\mathbf{e}|$ is below a threshold.

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \left(\mathbf{J}_k^T \mathbf{J}_k\right)^{-1} \mathbf{J}_k^T \mathbf{e} \tag{41}$$

3.4.1.1 Obtaining the Intrinsic Coordinates for Each Camera

Once a suitable set of calibration images have been collected Matlab is used to run the calib_gui script from the Camera Calibration Toolbox. Default parameters are used with the exception of the following:

The default mechanism for attempting to automatically count the grid pattern size is disabled, this normally works well for solid checkerboard patterns but is not suited to our pattern. As a result the grid pattern size must be manually entered.

The toolbox automatically extracts the corner points using a corner detector, these point locations are then manually refined by running the manual_corner_extraction script from the command line.

The aspect ratio, center point and skew are assumed fixed by manually setting the following matlab variables at the command line prior to performing the calibration step using the gui:

est_aspect_ratio = 0;

center_optim=0;

est_alpha=0;

In addition a second order radial distortion model [61] is chosen by setting the following variable at the command line

est_dist = $\begin{bmatrix}1 & 0 & 0 & 0 & 0\end{bmatrix}^T$;

Typical output for the calibration step is shown in Figure 57**.**

```
Aspect ratio not optimized (est_aspect_ratio = 0) -> fc(1)=fc(2). Set
est_aspect_ratio to 1 for estimating aspect ratio.
Principal point not optimized (center_optim=0). It is kept at the center of the
image.
Skew not optimized (est_alpha=0) - (DEFAULT)
Distortion not fully estimated (defined by the variable est_dist):
    Fourth order distortion not estimated (est_dist(2)=0).
    Sixth order distortion not estimated (est_dist(5)=0) - (DEFAULT) .
    Tangential distortion not estimated (est_dist(3:4)~=[1;1]).

Focal Length:          fc = [ 335.26191   335.26191 ] ± [ 10.99124   10.99124 ]
Principal point:       cc = [ 159.50000   119.50000 ] ± [ 0.00000   0.00000 ]
Skew:             alpha_c = [ 0.00000 ] ± [ 0.00000  ]   => angle of pixel axes =
90.00000 ± 0.00000 degrees
Distortion:            kc = [ -0.21969   0.00000   0.00000   0.00000  0.00000 ] ± [
0.04823   0.00000   0.00000   0.00000  0.00000 ]
Pixel error:          err = [ 0.24219   0.23909 ]

Note: The numerical errors are approximately three times the standard
deviations (for reference).
```

Figure 57 – Typical Results for Single Camera Calibration of 320x240 Images

After performing the calibration for each camera it is necessary to save the results for use by the stereo calibration routines.

3.4.1.2 Stereo Calibration

The stereo_gui script from the Camera Calibration Toolbox uses the saved camera calibration files for the left and right camera to compute the relative rotation matrix and translation vector between the left and right cameras using the method previously described. This information along with the jointly optimized intrinsic camera coordinates is saved into a stereo calibration file.

3.4.1.3 Rectification of the Stereo Video Road Sequences

The stereo calibration file is used to rectify the left right image pairs of the road video. Rectification of a stereo image pair results in horizontal epipolar lines so that any feature in one image will lie on the same horizontal line in the other image. A matlab script based on the rectify_stereo_pair toolbox script is used to rectify the captured avi files.

Figure 58 shows a before and after comparison of a stereo pair. The rectified stereo pair are shown in the lower half of the figure. Note how corresponding features line vertically in the rectified imagery.

Figure 58 – Example of Rectification of Stereo Pair

3.4.2 Initialization

The proposed method requires an initialization frame containing the road boundaries in the left and right image and selection of a horizon line that will limit the area for performing segmentation. Setting the horizon line at the true horizon yields poor results, therefore a horizon line lying about 10 pixels (for a 320x240 resolution frame) below the true horizon is chosen.

Figure 59 shows a typical starting frame after initialization of the boundaries and selection of a horizon.

Figure 59 – Typical First Frame Showing Initial Boundary Selections.

3.4.2.1 Color Histogram Initialization

The initial boundaries are used to define road and terrain regions. The color distribution of pixels lying on the road and terrain below the user selected horizon are used to initialize the color histograms. See 3.4.3 for a discussion of the color segmentation algorithm used. Figure 60 shows a typical set of road and terrain regions.

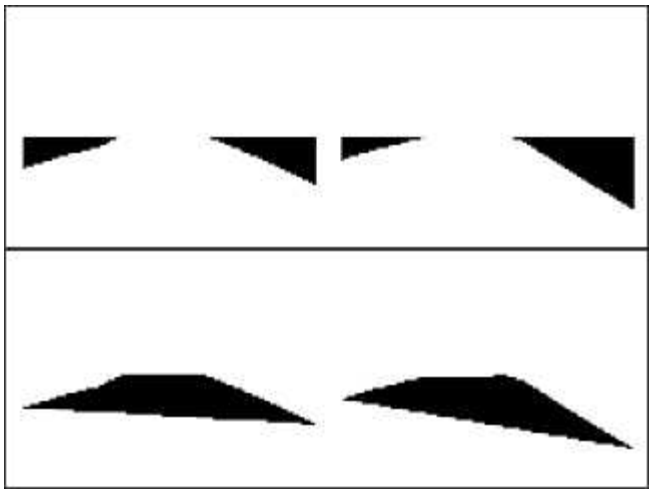

Figure 60 – Terrain and Road Templates

3.4.2.2 Snake Initialization

The left and right boundaries are used to initialize active contours (Snakes) that will follow the road boundaries. The boundaries are subdivided into an equal number of segments

3.4.3 Color Histogram Based Segmentation

Our color segmentation method is based on a simplified version of the Color Predicate (CP) method used for skin segmentation [62]. The Color Predicate method uses the Hue-Saturation-Intensity (HSI) color space which is related to the RGB color space by the following equations [63] [page 27]:

$$
\begin{aligned}
H &= \cos^{-1}\left[\frac{\frac{1}{2}\left[(R-G)+(R-B)\right]}{\left[(R-G)^2+(R-G)(G-B)\right]^{1/2}}\right] \\
S &= 1-\frac{3}{(R+G+B)}\left[\min(R,G,B)\right] \\
I &= \frac{1}{3}(R+G+B)
\end{aligned}
\tag{42}
$$

The CP method uses a training image which has been roughly pre-segmented into a target and background. Pixels with intensities that are too large or small are ignored. The rest of the pixels are subdivided by intensity regions, typically the four most significant bits of a 8 bit intensity scale or 16 levels. For each level the pixel hue and saturation value are used to increment the corresponding index in a histogram map for pixels inside the target area. Typical index ranges for a hue and saturation histogram are to use the 6 most significant bits for 8 bit values, i.e. a 64 by 64 index array. Pixels outside the target area decrement the corresponding index count. The neighboring index locations are also incremented or decremented according to a Gaussian weighting function. When all the pixels have been processed the histograms are thresholded to a binary 1 or 0 dependent on the index count being positive or negative.

New images are segmented by classifying the pixels according to their HSI values as target, background or unknown in accordance with the binary histograms.

3.4.3.1 Color Model

Our color segmentation method uses the YUV color space and ignores intensity variations. The RGB pixel values are converted into the YUV color space using the following equation [64]:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} .299 & .587 & .114 \\ -.147 & -.289 & .436 \\ .615 & -.515 & -.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 0.5 \\ 0.5 \end{bmatrix} \tag{43}$$

Equation 43 assumes that the RGB pixel values have been normalized into a 0 to 1 range. The YUV output values will be in the range 0 to 1. A full white normalized RGB pixel [1 1 1] will be represented by a YUV value of [1 0.5 0.5]. *Y* represents the pixel intensity, *U* and *V* are the blue and red chrominance respectively.

3.4.3.2 Color Histogram

Our color segmentation method uses the difference in color distribution between the road and terrain to classify the image pixels lying below the horizon line. From Bayes rule we have:

$$P(Road \mid \mathbf{c}_i) = \frac{P(\mathbf{c}_i \mid Road)P(Road)}{P(\mathbf{c}_i)}$$
$$P(Terrain \mid \mathbf{c}_i) = \frac{P(\mathbf{c}_i \mid Terrain)P(Terrain)}{P(\mathbf{c}_i)} \quad (44)$$

where $\mathbf{c}_i$ represents the pixel color vector $\begin{bmatrix} u_i & v_i \end{bmatrix}^T$. Equation 44 gives us the conditional probability of a pixel being in the road or terrain given a specific color. Taking the ratio of the conditional probabilities gives us:

$$\frac{P(Road \mid \mathbf{c}_i)}{P(Terrain \mid \mathbf{c}_i)} = \frac{P(\mathbf{c}_i \mid Road)P(Road)}{P(\mathbf{c}_i \mid Terrain)P(Terrain)} \quad (45)$$

For typical two lane roads, given the wide field of view of the cameras, it is evident from looking at the template sizes in Figure 60 that *P(Road)>P(Terrain)* ,therefore we can replace the ratio $\frac{P(Road)}{P(Terrain)}$ with $\alpha > 1$ to give us:

$$\frac{P(Road \mid \mathbf{c}_i)}{P(Terrain \mid \mathbf{c}_i)} = \frac{P(\mathbf{c}_i \mid Road)\alpha}{P(\mathbf{c}_i \mid Terrain)}$$

$$Pixel\ i\ Class = \begin{cases} \dfrac{P(\mathbf{c}_i \mid Road)}{P(\mathbf{c}_i \mid Terrain)} > 1 > \dfrac{1}{\alpha}\ ;\ Pixel\ i \in Road \\ \dfrac{P(\mathbf{c}_i \mid Terrain)}{P(\mathbf{c}_i \mid Road)} < 1 < \alpha\ ;\ Pixel\ i \in Terrain \end{cases} \quad (46)$$

$P(\mathbf{c}_i|Road)$ and $P(\mathbf{c}_i|Terrain)$ are the color histograms of the Road and Terrain regions respectively. For a specific pixel color, we see that if the ratio is greater than 1 we can classify the pixel as road.

The 2D color histograms for the road and terrain regions are determined in the UV coordinate system. The 0 to 1 range of U and V is divided into 100 equally spaced bins.

3.4.3.3 Histogram Spatial Filtering

The color histograms are smoothed using a 5x5 averaging filter and then normalized to form pdf's. These pdf's are used to classify incoming frames using Equation 46 into road and non-road pixels as shown in Figure 61.

Figure 61 – Classification into Road Pixels

3.4.3.4 Filtering and Morphological Operations

To remove isolated points and smooth the binary road image prior to performing gradient operations the image is filtered with a 3x3 majority operator that operates like a median filter by convolving the image with a 3x3 summing kernel and thresholding as one any value greater than 4. Then a 3x3 averaging filter smooths the median filtered road image to give the results shown in Figure 62.

Figure 62 – Smoothed median filtered Bayesian

The horizontal and vertical gradients are obtained using the Sobel filter and the gradient magnitude is computed. All three gradient images are low pass filtered using a rotationally symmetric Gaussian lowpass filter of size 5 with a standard deviation 1 (see Figure 63).

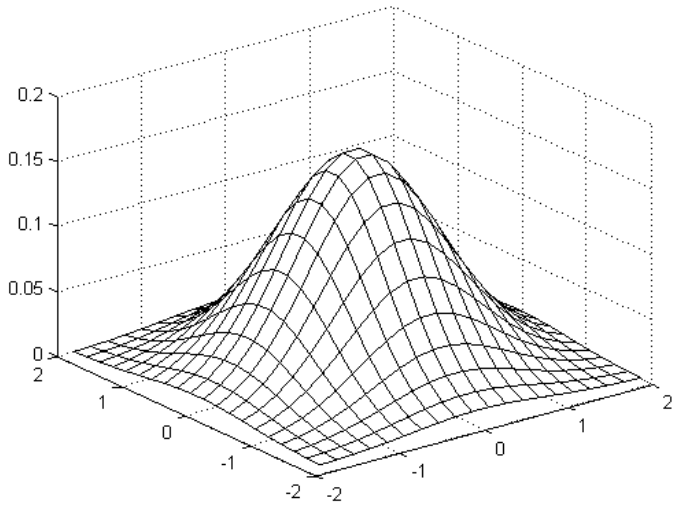

Figure 63 – Gaussian Low Pass Filter Kernel of Size 5, $\sigma = 1$

Taking the gradient of the median and low pass filtered binary road image yields artifacts, due to the rectification process, horizon removal and classification errors; that do not lie in the vicinity of the true road boundaries. To prevent the active contours from attaching to such false boundaries we have developed two masks that are used to attenuate the gradient images.

The first mask is created by taking the union of the image area above the horizon and the white fill created by the rectification process at the image boundaries. The union is morphologically dilated [65] twice using a 3x3 structuring element to obtain the static mask shown in Figure 64.

Figure 64 – Horizon and Rectification Mask

The second mask is dynamic and is created by dilating the road template as defined by the left and right road boundaries found in the previous frame. This road mask is applied to the second frame onwards of the video sequence. The road mask prevents the active contour's ability to wander away from the true road boundaries in a single frame while still permitting it to follow the changes in the roadway curvature over time. A typical road mask is shown in Figure 65.

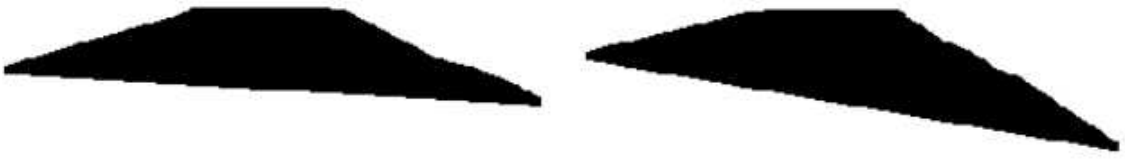

Figure 65 – Road Mask

The preceding operations give us images of the left and right road boundaries as shown in Figure 66

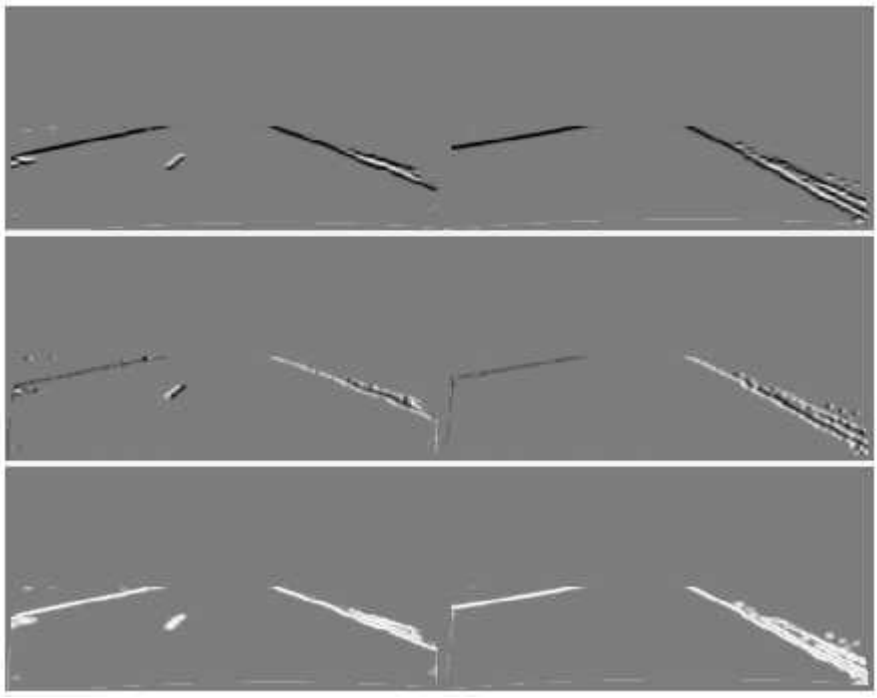

Figure 66 – Horizontal (Top) and Vertical (Middle) Components and the Magnitude After Applying the Masking Operations

3.4.4 Snake Algorithm for finding Road Boundaries

Having obtained the left and right road boundaries, we use a variant of the active contour algorithm (snake) introduced by Kass [66] and modified by Williams and Shah [44].

The traditional snake model as developed by Kass seeks to minimize the following energy functional:

$$E = \int_0^1 [\frac{1}{2}(\alpha|\mathbf{x}'(s)|^2 + \beta|\mathbf{x}''(s)|^2) + E_{ext}(\mathbf{x}(s))]ds \quad (47)$$

Where $\boldsymbol{x}$(s)=[*x(s), y(s)*] is a parametric representation of a spline and α and β are weighting parameters controlling the snakes resistance to bending and changes in the radius of curvature. E_{ext} represents external forces and is typically set to the negative of the image gradient

$$E_{ext} = -\left|\nabla I(x(s), y(s))\right|^2 \tag{48}$$

Williams and Shah proposed minimizing the following energy functional using a discrete greedy algorithm:

$$E = \int (\alpha E_{cont} + \beta E_{curv} + \gamma E_{image}) ds \tag{49}$$

Assuming discrete points on a closed contour are labeled 0 to n, the greedy algorithm selects the new location of a point from the local 3x3 neighborhood.

E_{image} is the image force which can be defined as a gradient or the image intensity. E_{cont} is the absolute difference between the distance from point i to i-1 and the average distance:

$$\begin{aligned} E_{cont} &= \left|\bar{d} - \left|\mathbf{v}_i - \mathbf{v}_{i-1}\right|\right| \\ \bar{d} &= \frac{\sum_{i=1}^{n} \left|\mathbf{v}_i - \mathbf{v}_{i-1}\right|}{n} \end{aligned} \tag{50}$$

E_{cont} attempts to keep the points evenly spaced to counteract a tendency of the Kass technique to bunch up points near areas of strong external forces. The second term E_{curv} is a curvature term defined as:

$$E_{curv} = \left| \frac{\mathbf{u}_i}{|\mathbf{u}_i|} - \frac{\mathbf{u}_{i+1}}{|\mathbf{u}_{i+1}|} \right|^2 \tag{51}$$

Where $\mathbf{u_i}$ is the vector from point i-1 to i. This definition for E_{curv} has the advantage of being rotationally invariant for discrete digital curves.

Our snakes technique minimizes the following energy functional

$$E = E_{first} + \sum_{i=2}^{n-1} (E_{length} + E_{curv} + E_{image} + E_{contour}) + E_{last} \tag{52}$$

The E_{length} and E_{curv} terms are identical to the E_{cont} and E_{curv} terms respectively of Equation 49. E_{image} is set to the negative of the gradient magnitude as computed in section 3.4.3.4.

The E_{contour} term is used to force points that are far from the image gradient toward the gradient by using the normal projection of the image gradient on the neighboring points as an additional force moment.

Using a greedy algorithm technique we search the neighborhood about each point for a minimum local energy. Since our contours are not closed the first and last points are treated differently than the interior points. The first and last points are only permitted to move horizontally. Once the minimum local energy location is found for each point in the contour we adjust the point coordinates. This process is repeated until the stopping criteria is met. Our stopping criteria is when two or less points have been moved in a pass or the maximum allowed iterations have been reached. Additional constraints restrict the permissible movement of the snake points so as to stay in a valid image area.

Holding the first and last points fixed vertically prevents our snake from growing or shrinking. We also resample the interior points to equalize the segment lengths after each boundary has been found. This is necessary because we have found that even with the E_{cont} term in the energy functional of Equation 52, the snake points tend to become unevenly distributed.

3.4.5 Matching Boundary Points Between Stereo Pairs

Since the road boundaries found by the snake can differ in vertical extent between the left and right frames it is necessary to find which portion of the boundary exists in both frames. Since the stereo frames have been rectified, we know that corresponding points lie on the same horizontal line, therefore all that is necessary to find the common sub segment is to find the intersection of the vertical extents.

3.4.5.1 Stereo Disparity Refinement

The common subsegments for the left and right road boundaries from the left frame are used to find corresponding points between the left and right frames that lie close to a road boundary. A set of evenly set points are extracted from the subsegments by interpolation. For each point, the corresponding match in the right frame is found by finding the minimum correlation distance between corresponding 7 by 3 pixel patches. The correlation distance is determined by performimg a Sum of Square Differences (SSD) correlation using the grey level image and the grey level gradient magnitude.

The grey level image is computed by taking a simple average of the RGB components, $grey = (R + G + B)/3$ and then filtering with a rotationally symmetric Gaussian lowpass filter of size 5 with a standard deviation 0.75. The grey gradient is computed by applying the vertical and horizontal sobel filter to the grey level image and obtaining the magnitude.

$$\nabla I = \left| \mathbf{h} * I + \left(\mathbf{h}^T * I \right) j \right|, \mathbf{h} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{53}$$

$$dist(d) = \sum_{i=-3}^{3} \sum_{j=-1}^{1} \left[\begin{array}{l} \dfrac{\left(I_{left}(x+i, y+j) - I_{right}(x+i+d, y+j) \right)^2}{\sigma_I^2} + \\ \dfrac{\left(\nabla I_{left}(x+i, y+j) - \nabla I_{right}(x+i+d, y+j) \right)^2}{\sigma_{\nabla I}^2} \end{array} \right] \tag{54}$$

where $\sigma_I, \sigma_{\nabla I}$ represent the standard deviation of the squared difference between the left and right frames 7 by 3 pixel patches.

$$\sigma_I^2 = \left[\begin{array}{l} \dfrac{1}{n \cdot m} \sum_{i=-m}^{m} \sum_{j=-n}^{n} \left(I_{left}(x+i, y+j) - I_{right}(x+i+d, y+j) \right)^2 \\ - \left(\dfrac{1}{n \cdot m} \sum_{i=-m}^{m} \sum_{j=-n}^{n} \left(I_{left}(x+i, y+j) - I_{right}(x+i+d, y+j) \right) \right)^2 \end{array} \right] \tag{55}$$

The disparity search space for *d* shown in Equation 56 is limited to a range of 0 to 50 pixels for 320 by 240 pixel frame sizes which corresponds to a range of infinity to approximately six meters.Range as a function stereo disparity in pixels is given by the following:

$$Range(d) = \frac{b}{2}\cot\left(\frac{d \cdot fov \cdot \frac{\pi}{180}}{2 \cdot hres}\right) \tag{56}$$

Where b is the baseline in meters, d is the disparity in pixels fov is the field of view in degrees and $hres$ is the camera resolution in pixels.

An example of the stereo matching procedure is shown in figure Figure **67** for typical road stereo pairs.

Figure 67 – Matching Points Between Left and Right Road Images

3.4.6 Extraction of 3D Coordinates

The 3D world coordinates of the road boundaries are found by triangulation and coordinate transformation of the left and right point matches. Please refer to section 3.1.1.2 for a description of the coordinate systems used.

3.4.6.1 Intersection of Coincident Rays (Camera Relative)

We use a routine from the Camera Calibration Toolbox which determines the 3D camera coordinates of the left and right point correspondences. This routine determines the point of closest approach of two vectors representing the inverse projection of the point correspondences. The first vector passes thru the left image plane coordinates and the left cameras focal point which is defined to be at the origin. The second vector passes thru the image plane coordinates and the focal point of the right camera after they have been translated into the left camera's coordinate system.

For rectified stereo images converting from the right camera coordinate system to the left is performed by adding the camera baseline found during the rectification process described in 3.4.1.3. The focal points of the cameras are:

$$\mathbf{f}_{left} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \quad \mathbf{f}_{right} = \begin{bmatrix} b & 0 & 0 \end{bmatrix}^T \tag{57}$$

where b is the rectified stereo baseline.

The image plane coordinates are obtained from the point correspondences using the following equation:

$$\mathbf{x}_{left} = \begin{bmatrix} f\left(u_{left} - cx\right) \\ f\left(v_{left} - cy\right) \\ -f \end{bmatrix}, \quad \mathbf{x}_{right} = \begin{bmatrix} f\left(u_{right} - cx\right) \\ f\left(v_{right} - cy\right) \\ -f \end{bmatrix} \tag{58}$$

where $\mathbf{x}$ is the image plane coordinates, f is the focal length , $\begin{bmatrix} u & v \end{bmatrix}^T$ are the pixel coordinates of the corresponding points and $\begin{bmatrix} cx & cy \end{bmatrix}^T$ are the pixel coordinates of the camera center.

The inverse projection unit vectors $\mathbf{n}_{left}$ and $\mathbf{n}_{right}$ are then:

$$\mathbf{n}_{left} = \frac{\overrightarrow{\mathbf{x}_{left}\mathbf{f}_{left}}}{\left\|\overrightarrow{\mathbf{x}_{left}\mathbf{f}_{left}}\right\|}, \quad \mathbf{n}_{right} = \frac{\overrightarrow{\mathbf{x}_{right}\mathbf{f}_{right}}}{\left\|\overrightarrow{\mathbf{x}_{right}\mathbf{f}_{right}}\right\|} \tag{59}$$

Assuming the two vectors are not parallel, the closest point of approach $\mathbf{X}$ is:

$$\begin{aligned}
&\mathbf{X} = \frac{\mathbf{I}_{left} + \mathbf{I}_{right}}{2} \\
&\mathbf{I}_{left} = \mathbf{f}_{left} + s \cdot \mathbf{n}_{left} \quad \mathbf{I}_{right} = \mathbf{f}_{right} + t \cdot \mathbf{n}_{right} \\
&\begin{bmatrix} s \\ t \end{bmatrix} = \frac{-1}{\left(a^2 - 1\right)} \begin{bmatrix} -1 & a \\ -a & 1 \end{bmatrix} \begin{bmatrix} \left(\mathbf{f}_{left} - \mathbf{f}_{right}\right) \cdot \mathbf{n}_{left} \\ \left(\mathbf{f}_{left} - \mathbf{f}_{right}\right) \cdot \mathbf{n}_{right} \end{bmatrix}, \quad for\ all\ \left(a = \mathbf{n}_{left} \cdot \mathbf{n}_{right}\right) \neq 1
\end{aligned} \tag{60}$$

3.4.6.2 Navigation Solution Using GPS

Converting from the camera relative coordinates found in the preceding section into world coordinates requires that the camera position be known.We developed a GPS only solution due to the poor performance of the solid state compass. The solid state compass performs poorly on a moving platform due its lack of a rate gyro which results in any acceleration being interpreted as a tilt by the compass's firmware.

The raw GPS latitude and longitude readings are converted into meters relative from the base station position. A 48 hr position average is used as the base station position. We have developed second order polynomial equations to convert the latitude and longitude deltas into distances which closely approximate land surveys taken of the UCF campus.

The classical equations to convert latitude/longitude coordinates into grid coordinates using a transverse mercator projection are given by Snyder [28]:

$$
\begin{aligned}
x &= x' + false\_easting \\
y &= y' + false\_northing \\
x' &= k_0 N\left[A + (1 - T + C) A^3/6 + \left(5 - 18T + T^2 + 72C - 58e'^2\right) A^5/120 \right] \\
y' &= k_0 N\left[M - M_0 + N\tan(\phi)\left\{ \begin{array}{l} A^2/2 + \left(5 - T + 9C + 4C^2\right) + \\ A^4/24 + \left(61 - 58T + T^2 + 600C - 330e'^2\right) A^6/720 \end{array} \right\} \right] \\
e^2 &= 2f - f^2 \\
e'^2 &= e^2/\left(1 - e^2\right) \\
N &= a/\sqrt{1 - e^2 \sin(\phi)} \\
T &= \tan^2(\phi) \\
C &= e'^2 \cos^2(\phi) \\
A &= (\lambda - \lambda_0)\cos^2(\phi) \\
M &= a\left[\begin{array}{l} \left(1 - e^4/4 - 3e^4/64 - 5e^6/256\right)\phi - \left(3e^2/8 - 3e^4/32 - 45e^6/1024\right)\sin(2\phi) + \\ \left(15e^4/256 - 45e^6/1024\right)\sin(4\phi) - \left(35e^6/3072\right)\sin(6\phi) \end{array} \right]
\end{aligned}
\qquad (61)
$$

Where x and y are in meters, a is the radius in meters and f is the ellipsoid flattening factor of the earth model, ϕ and λ are the latitude and longitude in radians, ϕ_0 and λ_0 are the latitude and longitude of the designated origin for the x, y coordinate grid; and M_0 is the true distance from the equator to ϕ_0.

Like the UTM; the Florida Coordinate System (FCS) also uses a transverse mercator projection for the eastern half of the state. The difference between the FCS and UTM projections for the Orlando/UCF area is summarized in the following table:

Table 9 – Transverse Mercator Projection Parameters for UTM and Florida Coordinate System

Parameter	**FCS**	**UTM**
a (meters)	6378137	6378137
f	$\frac{1}{298.25722210088}$	$\frac{1}{298.25722210088}$
False Easting (meters)	200000	500000
False Northing (meters)	0	0
Origin	24° 20’ 00”	0°
Central Meridian	−81°	−81°
k_0	0.9999	0.9996

We have implemented these equations in Matlab and validated the results against the freely available GeoPosCalc [67] and Corpscon [68] data conversion utilities using the WGS84 [69] datum.

Using the Snyder equations we generated the UTM coordinates for a 10 by 10 minutegrid with a 1 minute grid spacing (approximately 18 km by 16 km)centered on the “center” of the campus survey at 28.60211819°N and 81.20027213°W [70] and compared them against the Corpscon results for the FCS. The relative distances of the grid points to the center differed by the ratio of 0.9999/0.9996, i.e the ratio of the respective k_0 factors. A least squares fit of the following equation to the FCS over the grid area was then obtained:

$$
\begin{aligned}
x' &= \begin{bmatrix} dx^2 & dy \cdot dx & dy & dx \end{bmatrix} \begin{bmatrix} k_1 & k_2 & k_3 & k_4 \end{bmatrix}_{lat}^{T} \\
y' &= \begin{bmatrix} dx^2 & dy \cdot dx & dy & dx \end{bmatrix} \begin{bmatrix} k_1 & k_2 & k_3 & k_4 \end{bmatrix}_{lon}^{T} \\
dy &= r\left(\phi - \phi_0\right) \\
dx &= r_p\left(\lambda - \lambda_0\right); \\
r_p &= r\cos\left(\phi_0\right) \\
r &= \frac{ab}{\sqrt{a^2 \sin^2\left(\phi_0\right) + b^2 \cos^2\left(\phi_0\right)}} \\
b &= a\sqrt{1-e^2}
\end{aligned}
\tag{62}
$$

Where x', y' are the grid coordinates relative to the grid center at lat lon coordinates ϕ_0, λ_0, $e = \sqrt{2f - f^2}$ with the equatorial radius and flattening factos, a and f taken from Table 9. The fitting parameters, $\mathbf{k}_{lat}$ and $\mathbf{k}_{lon}$ are obtained by minimizing following error:

$$
\begin{aligned}
&\min_{\mathbf{k}_{lat} \in \mathbb{R}^4} \sum_{\lambda_{\min}}^{\lambda_{\max}} \sum_{\phi_{\min}}^{\phi_{\max}} \left(x'_{FCS}\left(\phi, \lambda\right) - x'\left(\phi, \lambda, \mathbf{k}_{lat}\right)\right)^2 \\
&\min_{\mathbf{k}_{lon} \in \mathbb{R}^4} \sum_{\lambda_{\min}}^{\lambda_{\max}} \sum_{\phi_{\min}}^{\phi_{\max}} \left(y'_{FCS}\left(\phi, \lambda\right) - y'\left(\phi, \lambda, \mathbf{k}_{lon}\right)\right)^2
\end{aligned}
\tag{63}
$$

For x'_{FCS} and y'_{FCS} representing the grid coordinates relative to the origin at ϕ_0, λ_0 obtained using Equation 61 with the FCS projection parameters.

Minimizing Equation 63 yields the following results:

Table 10 – Fitting Parameter Values

Coefficient	Latitude	Longitude
k_1	4.255394309135730e-008	3.903712494678972e-010
k_2	-.406854361615458e-010	-8.524189272125915e-008
k_3	9.963091255383771e-001	1.667173372644590e-003
k_4	-.675822473819229e-003	1.001484988434475e+000

Figure 68 shows the squared error in meters of the polynomial approximations over the grid area. The maximum error is on the order of 0.08 m (0.26 ft). This error is within the range of typical state land survey accuracy standards of one part in 5,000 to one part in 10,000 [71].

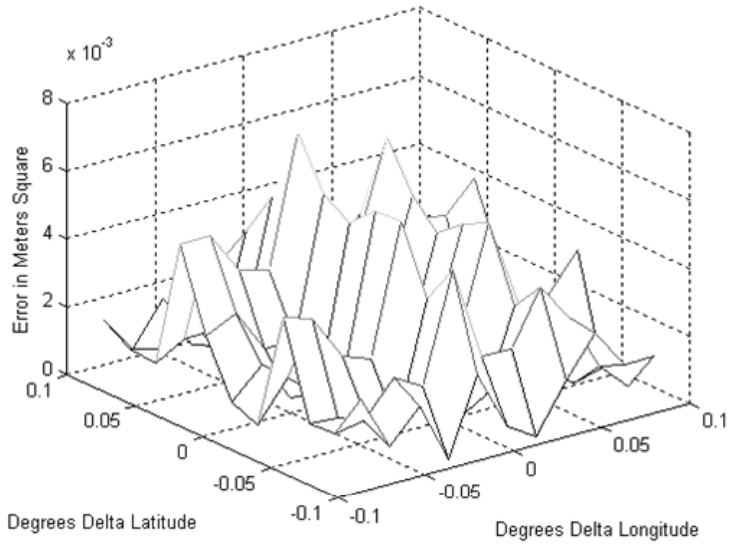

Figure 68 – Error Over Campus Area

The time stamped positions are then smoothed by separating the x and y coordinates into separate vectors and applying an overlapping piecewise second order polymomial fit to the parametric representation of the x and y coordinates as functions of time. The heading is obtained by differentiating the polynomial approximations and taking the arctangent of their ratio.

3.4.6.3 Conversion to World Coordinates

The video timestamps are used to obtain synchronized camera positions by linear interpolation. The corresponding camera relative coordinates found in 3.4.6.1 are then converted into world coordinates by applying the following transformation matrix.

$$
\begin{aligned}
&x = x' + x_1 \\
&y = y' + x_3 \\
&\mathbf{X}' = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T = \begin{bmatrix} \mathbf{R} & -\mathbf{RT} \end{bmatrix} \mathbf{X} \\
&\mathbf{X} = \begin{bmatrix} x & y & z & 1 \end{bmatrix}^T \\
&\mathbf{R} = \begin{bmatrix} \cos(hdg) & 0 & \sin(hdg) \\ 0 & 1 & 0 \\ -\sin(hdg) & 0 & \cos(hdg) \end{bmatrix} \\
&\mathbf{T} = \begin{bmatrix} 0.46 & 1.8 & 0 \end{bmatrix}^T
\end{aligned}
\tag{64}
$$

Where x and y are the grid coordinates of the road boundary point relative to the origin, x' and y' are the FCS coordinates from Equation 62, **X** is the road boundary point's camera relative coordinates, **R** and **T** are the rotational matrix and translation vector and *hdg* is the camera heading in radians. The translation vector was obtained by measuring the distance from the left cameras optical center to a point lying on the ground directly below the GPS antenna,

3.4.7 Polygonal Model Extraction

In order to create a suitable polygonal model given the extracted 3D world coordinates it is necessary to reject outliers thru robust filtering and define polygons to connect the smoothed data points.

3.4.7.1 Robust Parametric Fitting

Although the road boundary world coordinates consists of overlapping left right segments each consisting of multiple points, we only use the closest point in each segment. Since we have assumed a constant zero pitch while in reality the vehicle pitches due to road imperfections, acceleration and braking effects; some of the selected road boundary points have significant errors and need to be rejected. We accomplish this by performing a robust second order curve fit to reject these outliers.

The road boundary *x* and *y* coordinates are processed independently using a parametric representation. Median filtering is performed using a neighborhood size of three to reject outliers then a second order fit is performed using the following lorentzian distance measure.

$$\left\| x_i - f\left(t_i\right) \right\|_{\text{Lorentzian}} = \log\left(1 + \frac{\left(x_i - f\left(t_i\right)\right)^2}{2\sigma^2} \right) \tag{65}$$

where x_i and $f\left(t_i\right)$ represent the measurement and fit respectively and σ is the outlier threshold.

The outlier threshold is computed by taking the mean of the absolute x and y deltas from the parametric representation of the road boundaries:

$$\begin{aligned} \sigma_x &= \frac{\sum_{i=2}^{i=N} \left| x_i - x_{i-1} \right|}{N} \\ \sigma_y &= \frac{\sum_{i=2}^{i=N} \left| y_i - y_{i-1} \right|}{N} \\ x_i &= f_x\left(t_i\right); \quad y_i = f_y\left(t_i\right) \end{aligned} \tag{66}$$

Figure 69 shows a typical result, this figure represents a closeup view of a portion the the left road boundary. The raw input points are denoted by a +, the solid line is the robust fit and the dashed line represents the cars path; note how the outliers are bypassed:

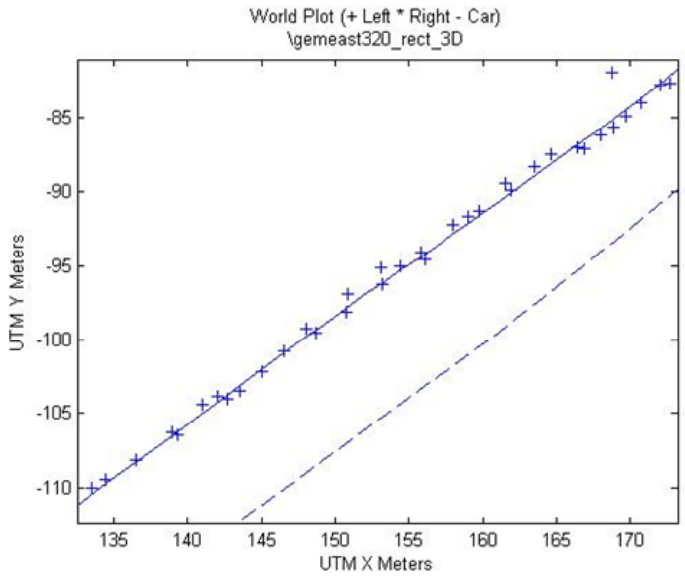

Figure 69 – Outlier Removal Detail

3.4.7.2 Polygonization

We have used two different polygonization methods:

The first method takes the left and right road boundaries and increases the point density by interpolation, it then forms left right point pairs that are spaced at minimum threshold distance and which lie on the closest perpendicular to the right boundary. These point pairs are extracted and assigned sequential vertex numbers; odd numbers for the left boundary points and even for the right boundary points as illustrated below:

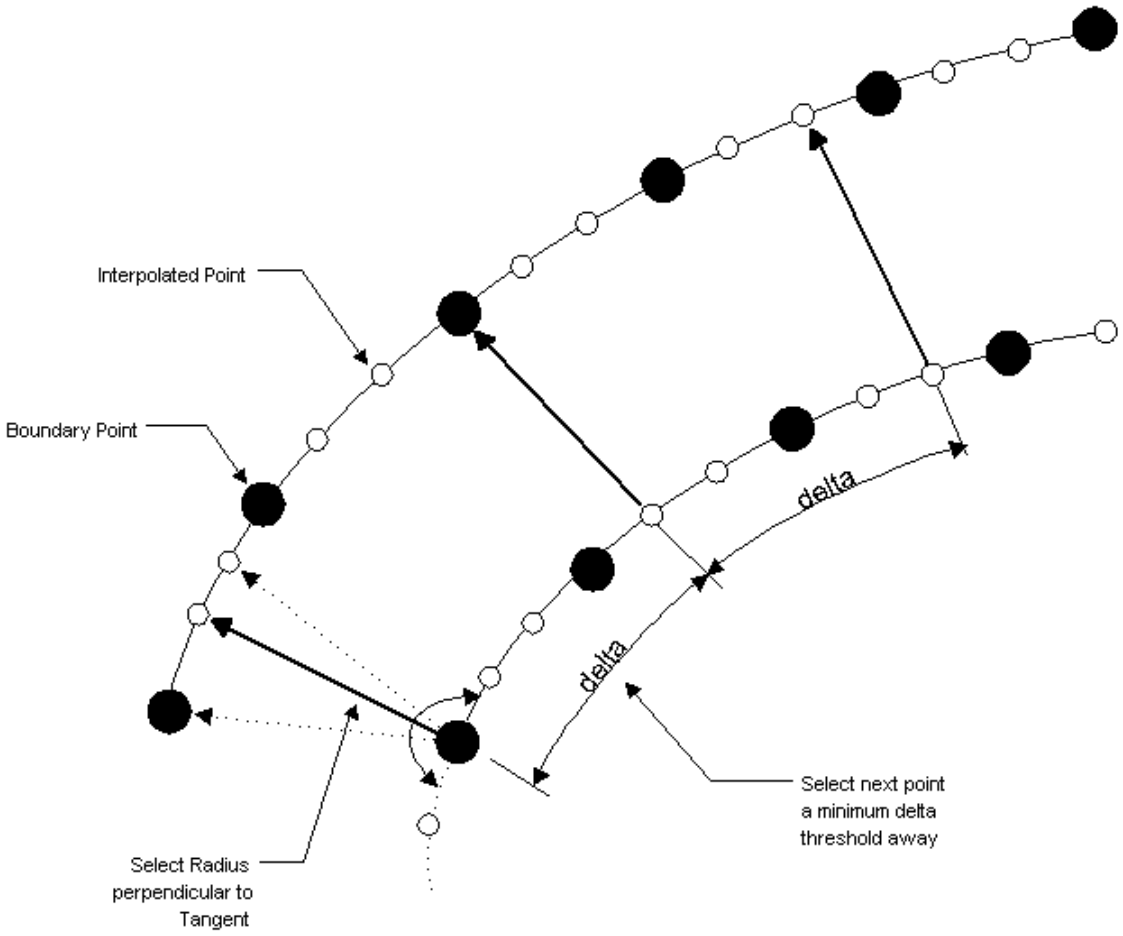

Figure 70 – Left and Right Boundary Point Matching

Polygons are then defined using the vertex sequence shown in the following figure:

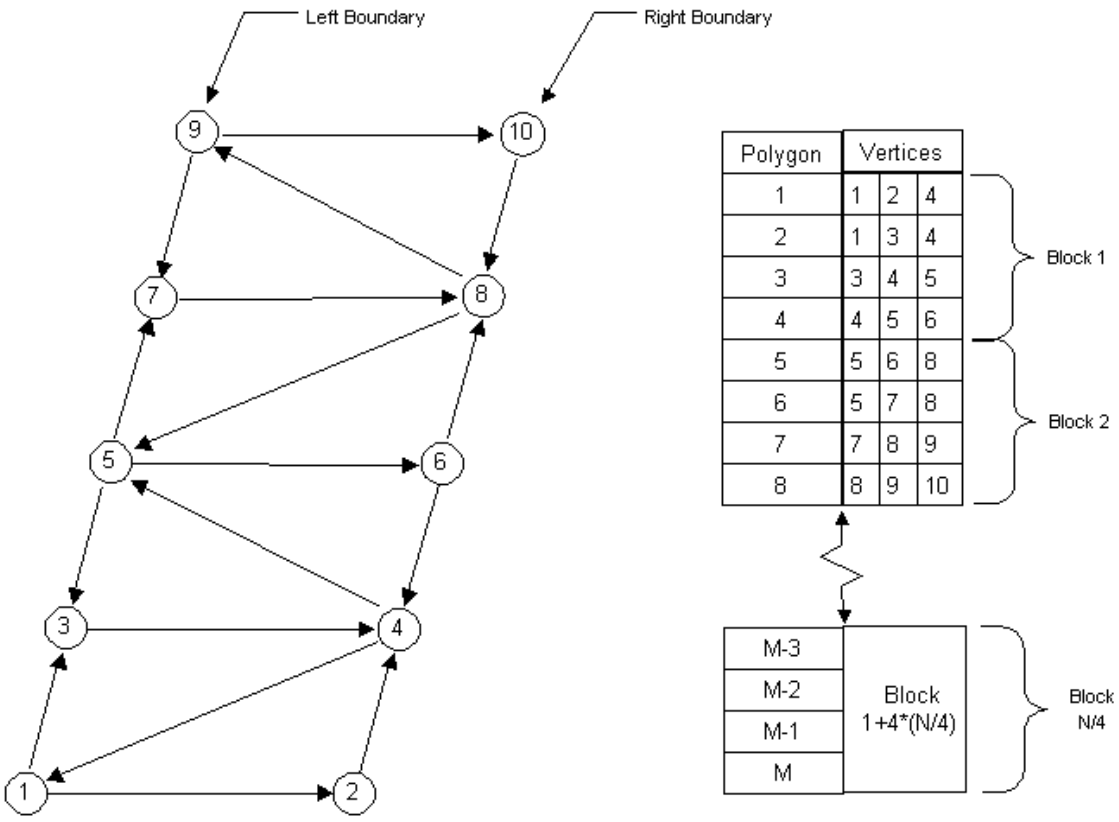

Figure 71 – Vertex Sequence

This method is suitable for defining a patch model which can be rendered in Matlab and overlayed onto a texture mapped surface as shown in the following figure:

Figure 72 – Demo of Matlab Rendering Road Model

(Rendered Sky Texture Map courtesy of Lopez-Fabrega Design, http://www.lfgrafix.com)

The second method uses a C++ routine adapted from [2] which uses the OpenFlight™ API to generate a Multigen Creator™ compatible flt format model. Left right point pairs are found by finding the intersection of right boundary normals spaced a fixed distance apart with the left boundary. The rectangle formed by consecutive point pairs is then subdivided into smaller rectangles representing thel left right and center lane markings and the left and right lanes. An example of a road model is shown in the following figure.

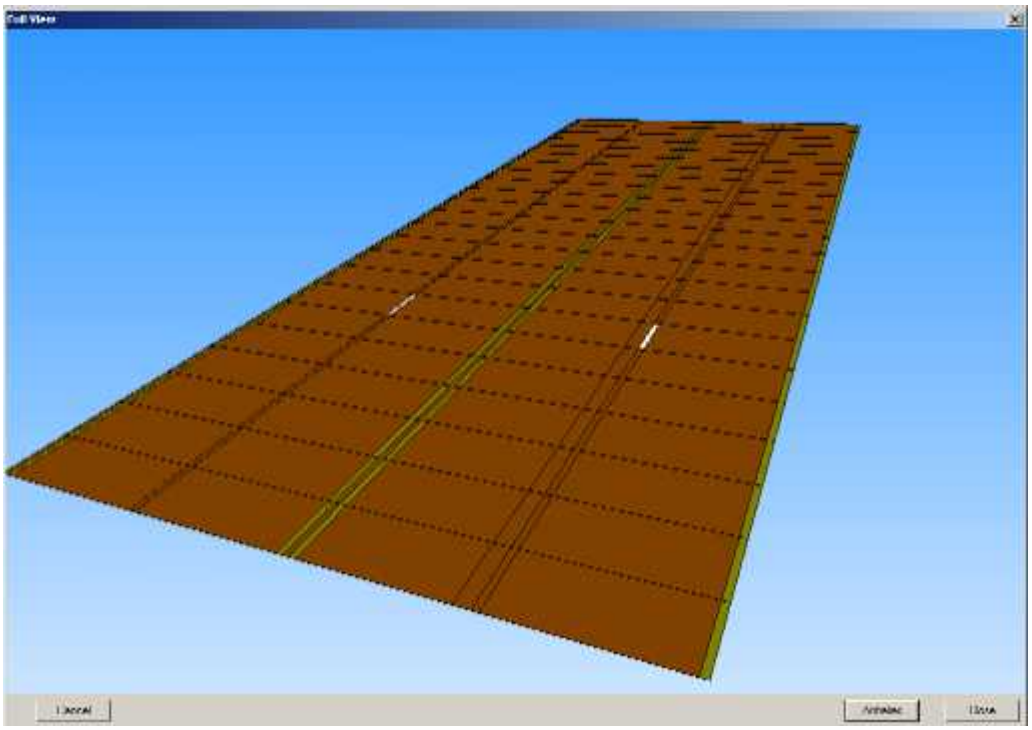

Figure 73 – Section of Road Model

3.4.8 Summary of Proposed Method

In summary our proposed method is composed of the following steps:

1. Collect time stamped road video, calibration video and time stamped GPS sensor data

2. Obtain the camera calibration data and use this to rectify the stereo road video.

3. Choose the initial road boundaries and use a color histogram classifier to identify the road pixels.

4. Extract the boundary coordinates of the road area using a snake algorithm.

5. Convert the 2D boundary coordinates into 3D by matching points between the stereo frames that are near the road boundaries and determining the closest point of approach of the vectors representing the inverse point projection.

6. Convert the GPS data into grid coordinates relative to the base station, smooth the navigation data and synchronize with the video sequence.

7. Use the heading and position information from the navigation data to convert the 3D boundary coordinates into world coordinates.

8. Perform a robust piecewise second order curve fit of the world coordinates.

9. Select left right boundary pairs lying on a perpendicular radial line and create a polygonal model.

4 RESULTS

Various data collection runs were performed on portions of the UCF road campus network at pixel resolutions of 320 by 240 and 640 by 480 . The following campus map is highlighted to show the areas selected for generation of 3D road models.

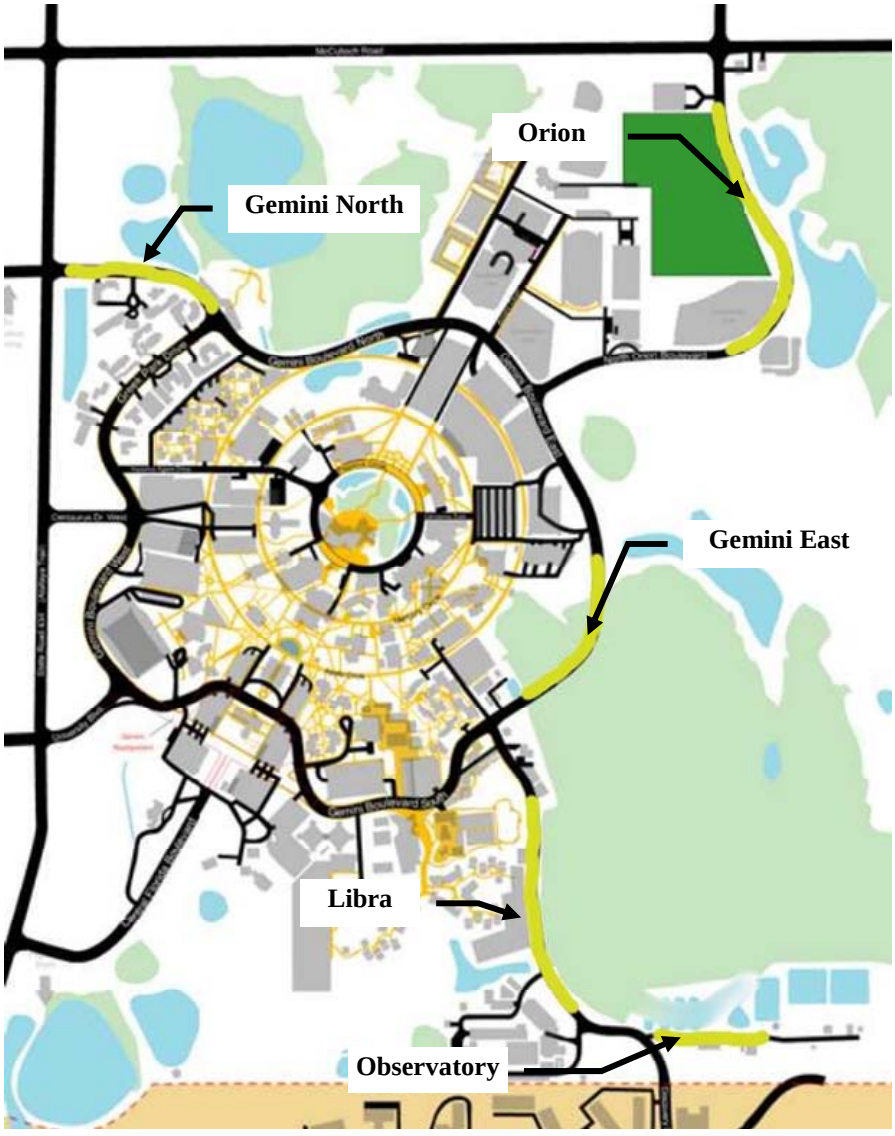

Figure 74 – UCF Campus Map

(Map courtesy of the University of Central Florida)

The following table describes the different road segments:

Table 11 – Campus Road Segments

Segment	Resolution	Description
Obs320	320x240	Observatory Road, two lane no side markers
Obs640	640x480	Same road section as Obs320
Libra320	320x240	Libra. Two lane road with bicycle path , sidewalk. And sidemarking. Road boundary obscured by passing car.
Gemeast320	320x240	Gemini East. Four Lane divided roadwith grass median, bicycle path and sidemarking
Orion320	320x240	Orion. Four Lane divided road with grass median, bicycle path and sidemarking. Road boundaries obscured by construction.
Gemnorth320	320x240	Gemini North. Four Lane divided road with grass median, bicycle path and sidemarking. Road surface pockmarked and median broken by large turnaround.

4.1 Segmentation Results

The following figures show the segmentation results for two of the road segments listed in Table 11. The original left and right color frames are on top. The bottom frames show the result of the majority operator to remove isolated pixels. Pixels classified as Road are in white.

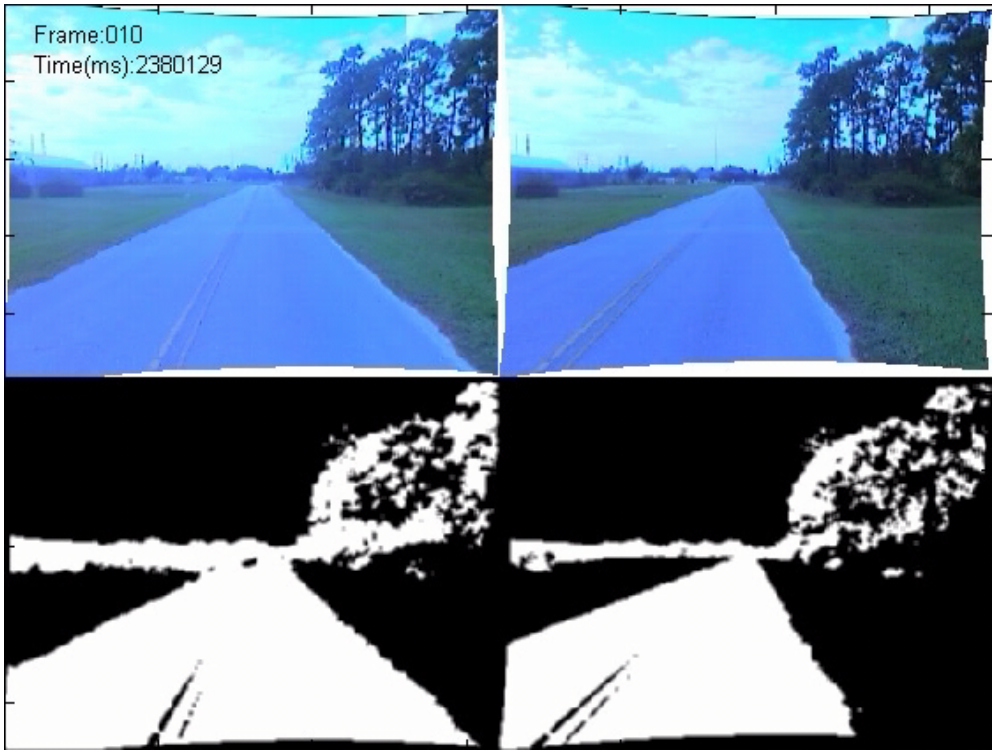

Figure 75 – Segmentation of Observatory Segment at Frame 10 of Sequence

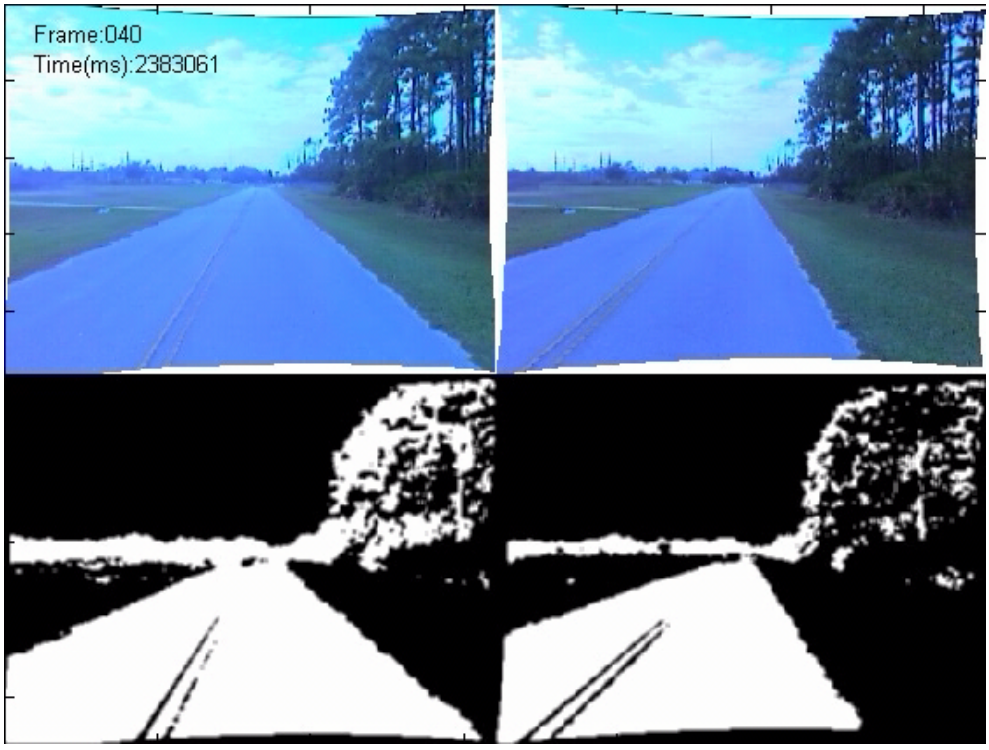

Figure 76 – Segmentation of Observatory Segment at Frame 40 of Sequence

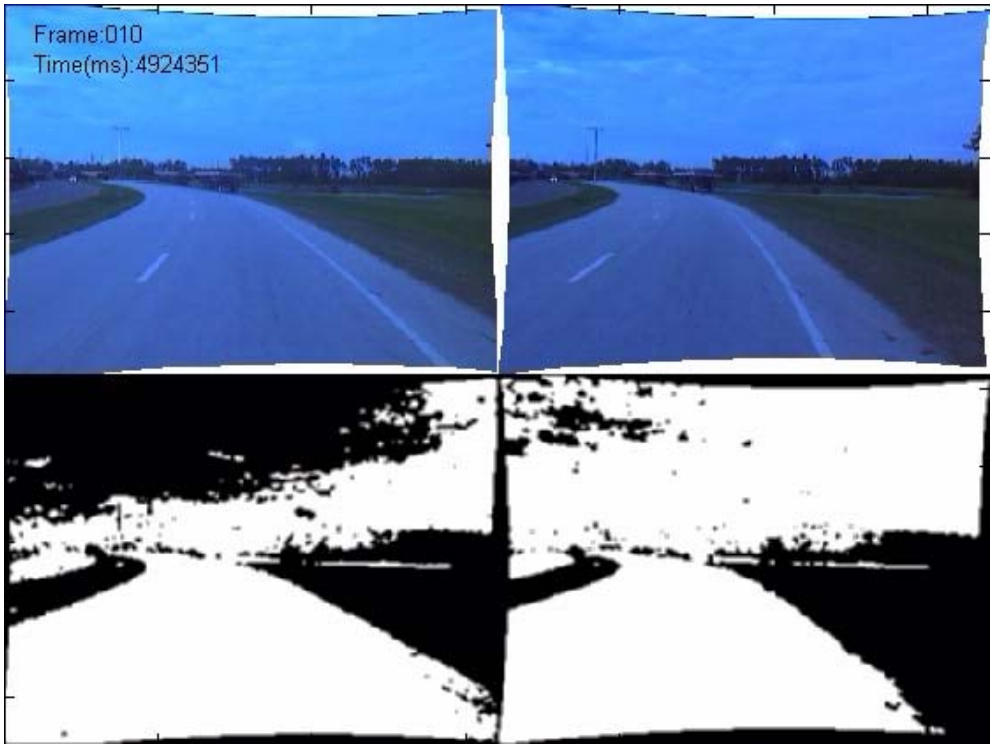

Figure 77 – Segmentation of Orion Segment at Frame 10 of Sequence

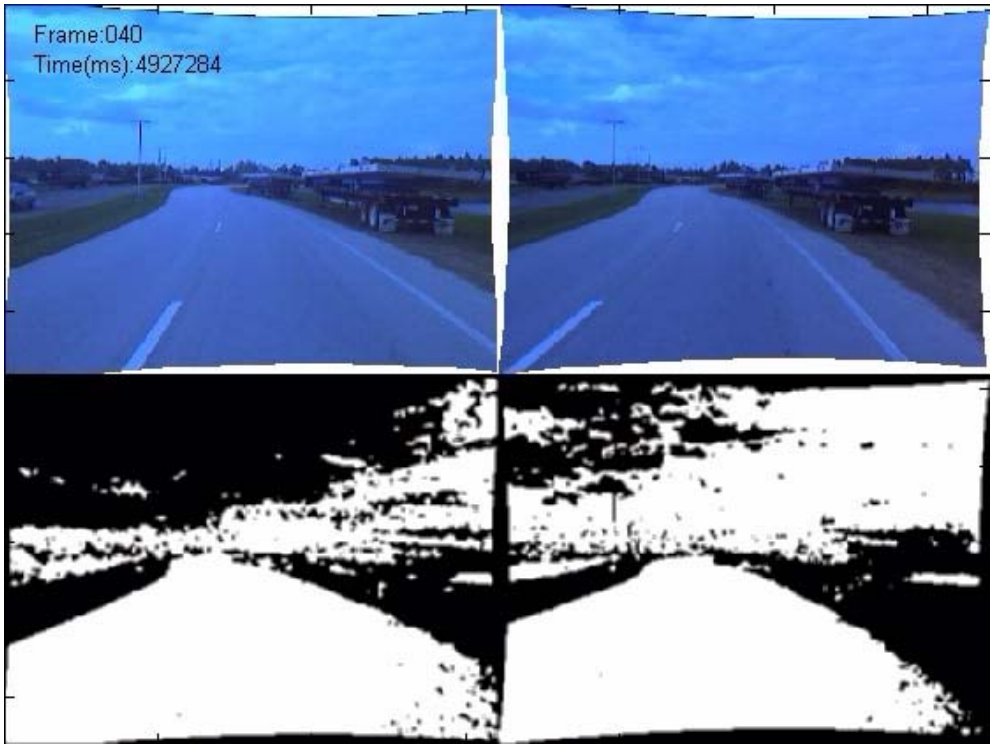

Figure 78 – Segmentation of Orion Segment at Frame 40 of Sequence

4.2 Road Boundary Extraction Results

The following figures show the road boundary extraction results of the snake routine for the Gemeast road segment. The first figure shows the left and right boundaries found by the snake algorithm overlayed onto the source image. The lower portion of the figure represents the boundary gradients that the snake algorithm uses. The crosses represent the snake node points and the circles are the common overlap area between the left and right frames. These overlapping points are passed to the stereo disparity routine for refinement prior to being triangulated.

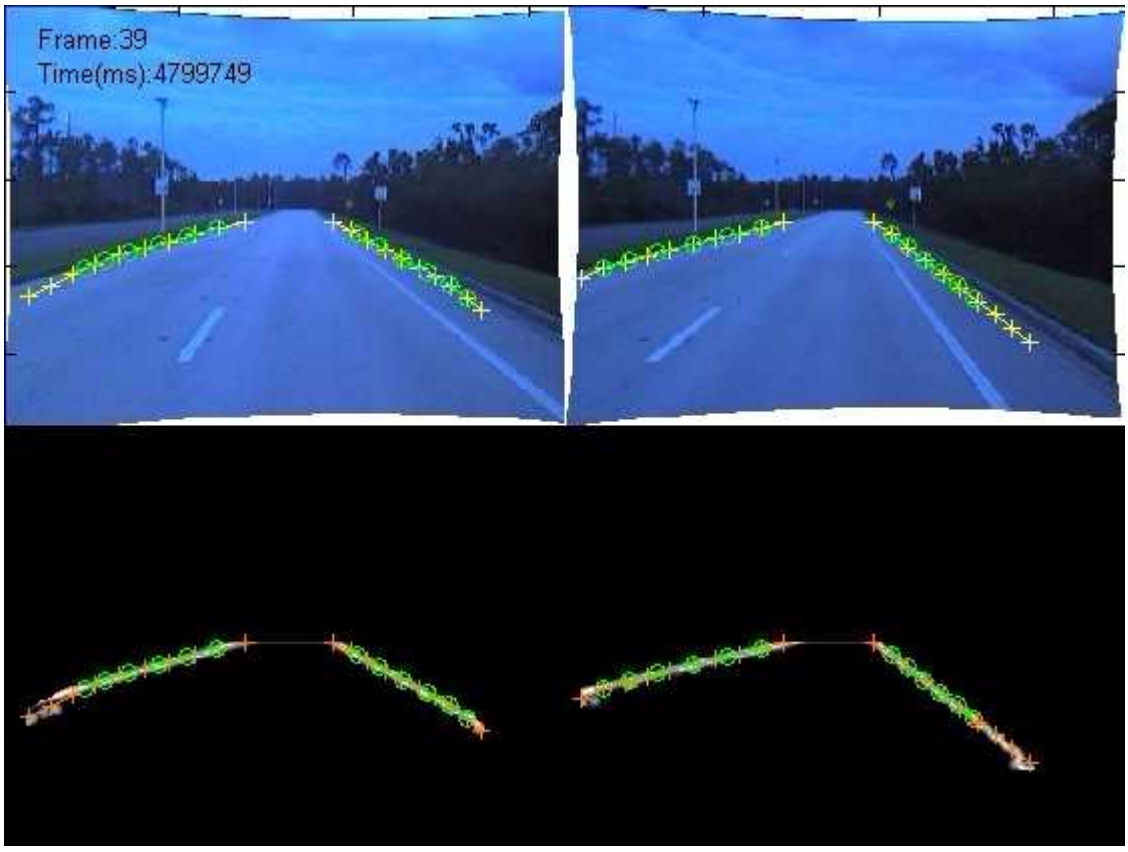

Figure 79 – Snake Results from Gemeast Road Segment

The following figure shows the camera relative 3D coordinates of a single frame of the Observatory segment. Since this road is very straight the left and right boundary points should be at constant 3D relative coordinates A scatter plot of the location of the left and right boundary points for the for the three closest points is shown in Figure 81 and Figure 82

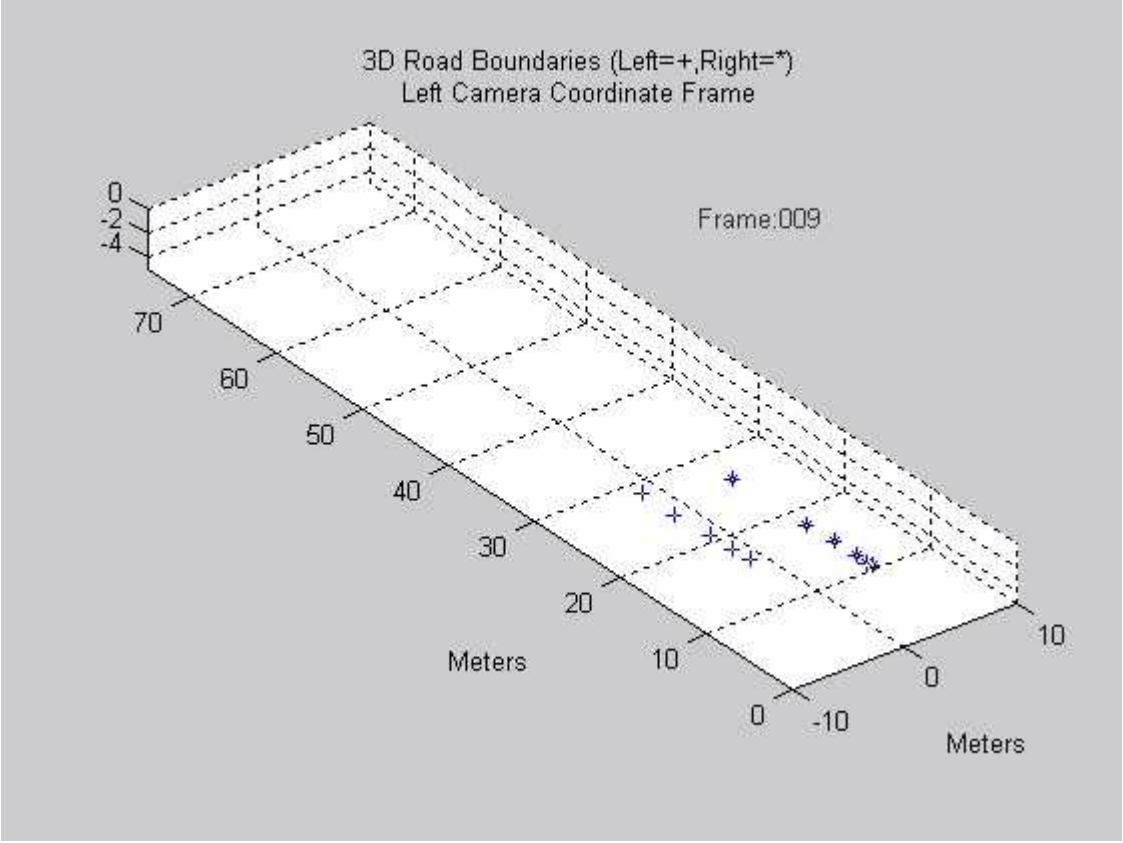

Figure 80 – 3D Camera Relative Road Boundaries

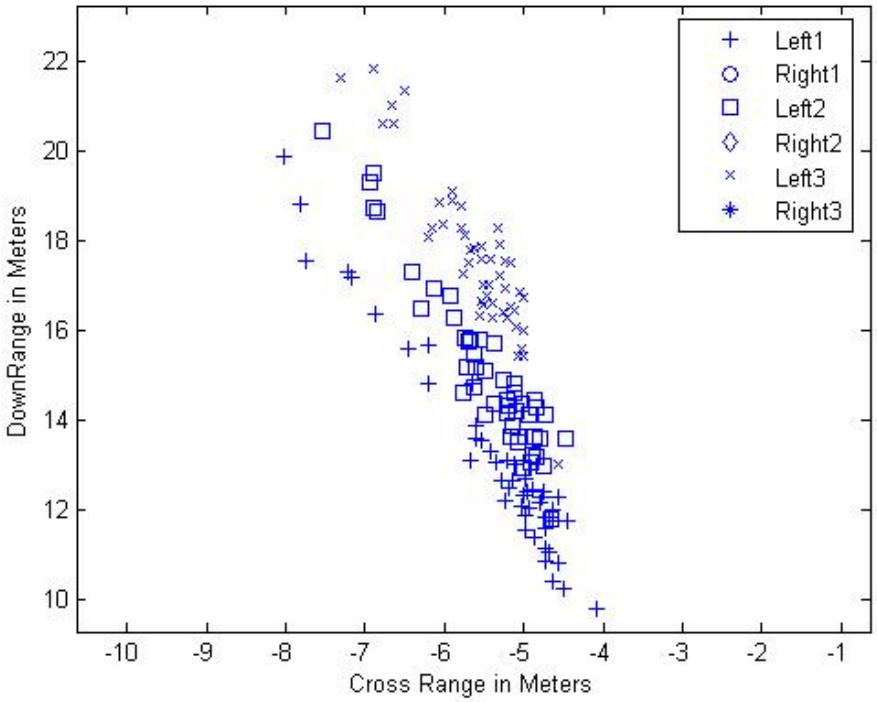

Figure 81 – Scatter Plot of Left Boundary Points

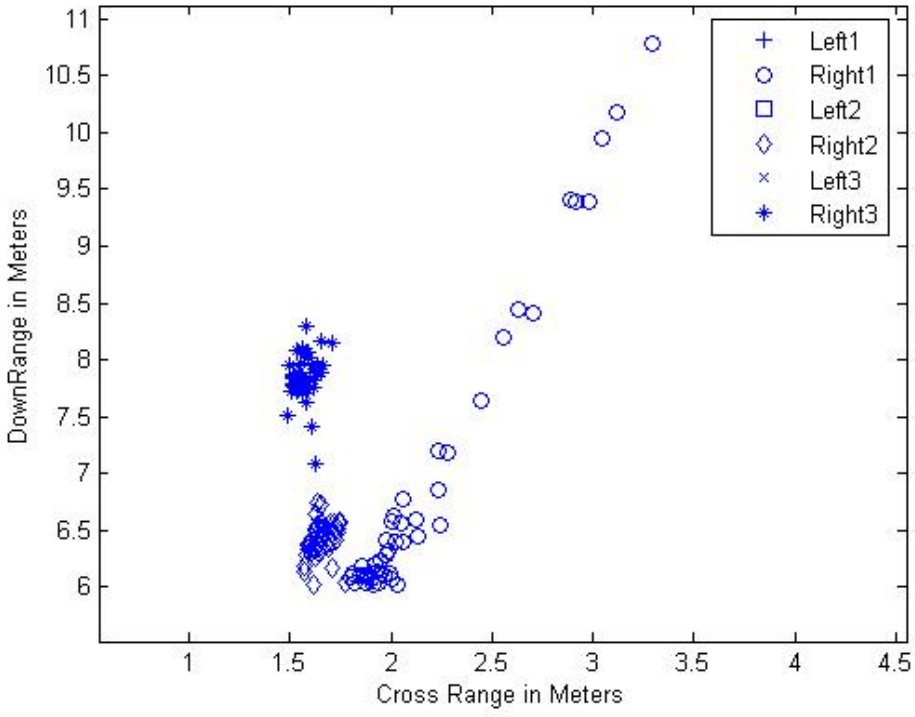

Figure 82 – Scatter Plot of Right Boundary Points

The left points exhibit more noise than the right points because they are farther away from the camera in addition to the effect of mismatches in the disparity refinement. The efficacy of the robust filtering of the 3D points after transformation to the world coordinate frame is seen in the following figure. Note how outliers are rejected, the raw input points are the + and *, the left and right boundary points respectively.

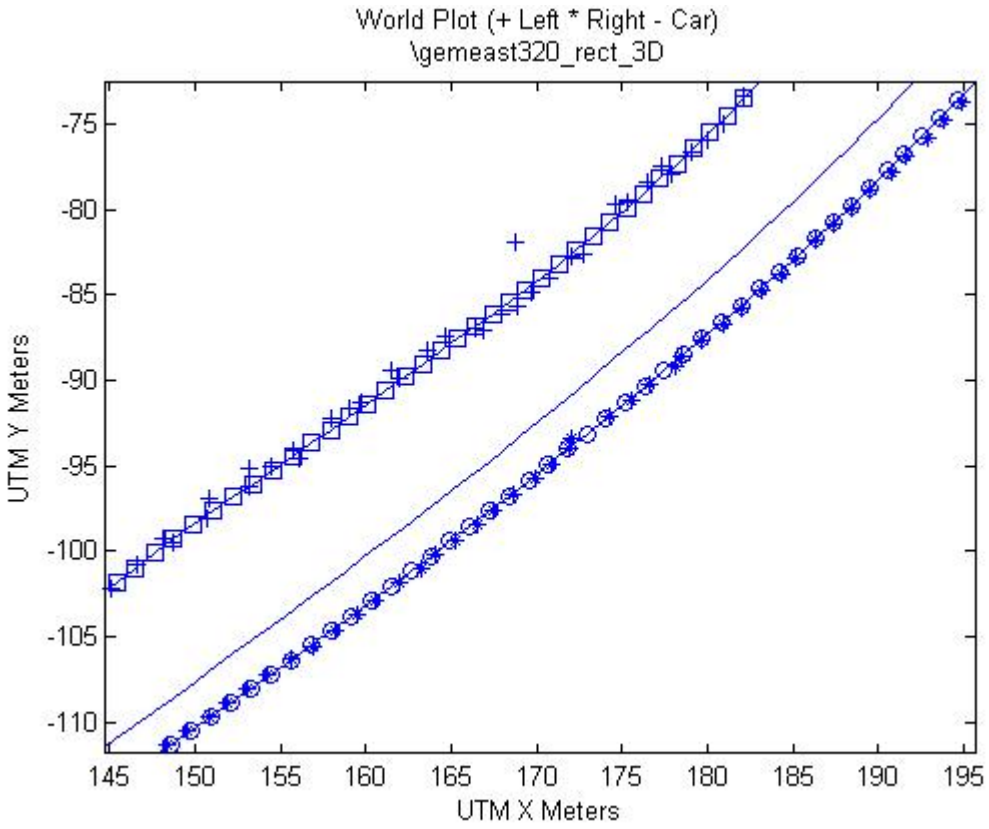

Figure 83 – Comparison of Gemeast Road Segment Boundaries with Path of Car

The following figure shows the road boundaries for the Gemini East segment overlayed on an orthorectified aerial picture of the road section obtained from the U.S. Geological survey. The boundary has a constant offset bias but follows the shape of the road. The + and * points represent the raw left and right boundary points. Similar results can be seen in the next figure using an orthorectified picture provided by UCF.

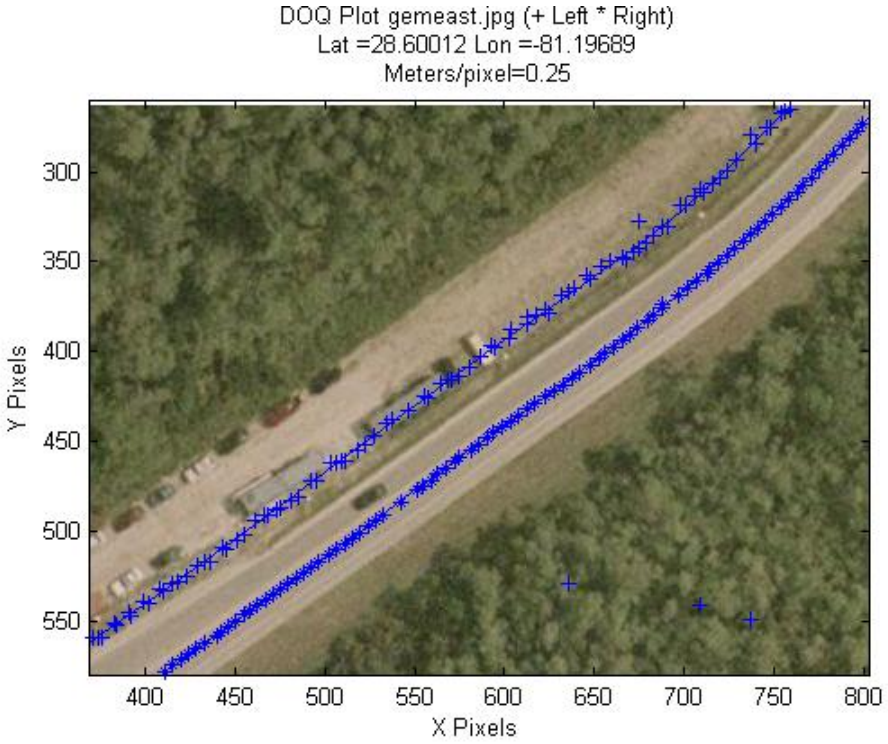

Figure 84 – Gemeast Road Segment Boundaries Overlayed on DOQ

(Image Courtesy of the U.S. Geological Survey)

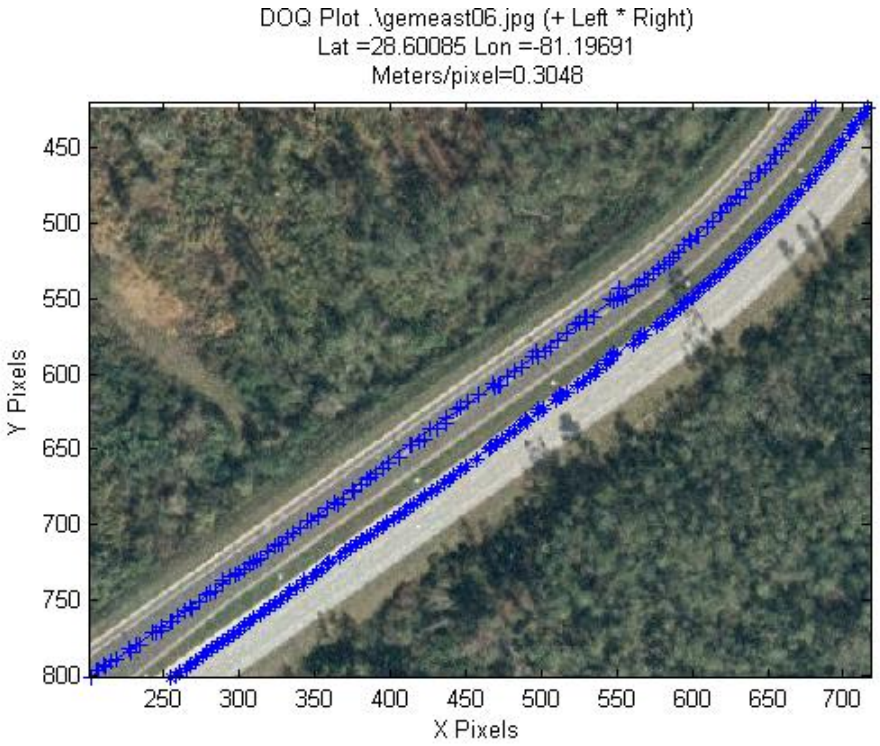

Figure 85 – Gemeast Road Segment Boundaries Overlayed on DOQ

(Image Courtesy of UCF)

Multiplying the scale of the orthorectified images by 0.97 to 0.98 results in a better match as shown in the following figures.

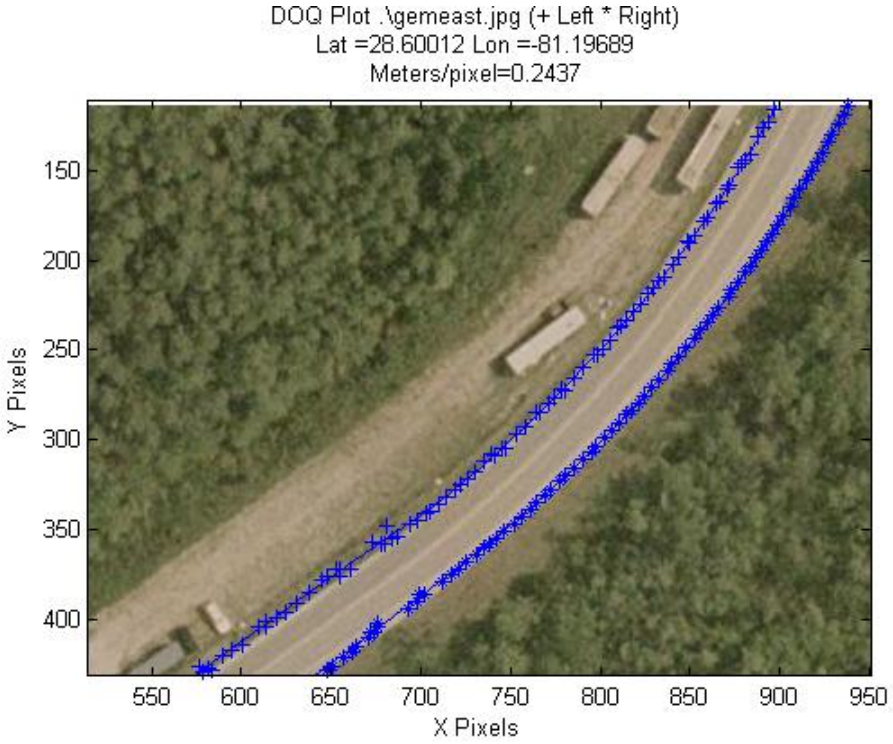

Figure 86 – Gemeast Road Segment Boundaries Overlayed on DOQ

(Image Courtesy of the U.S. Geological Survey)

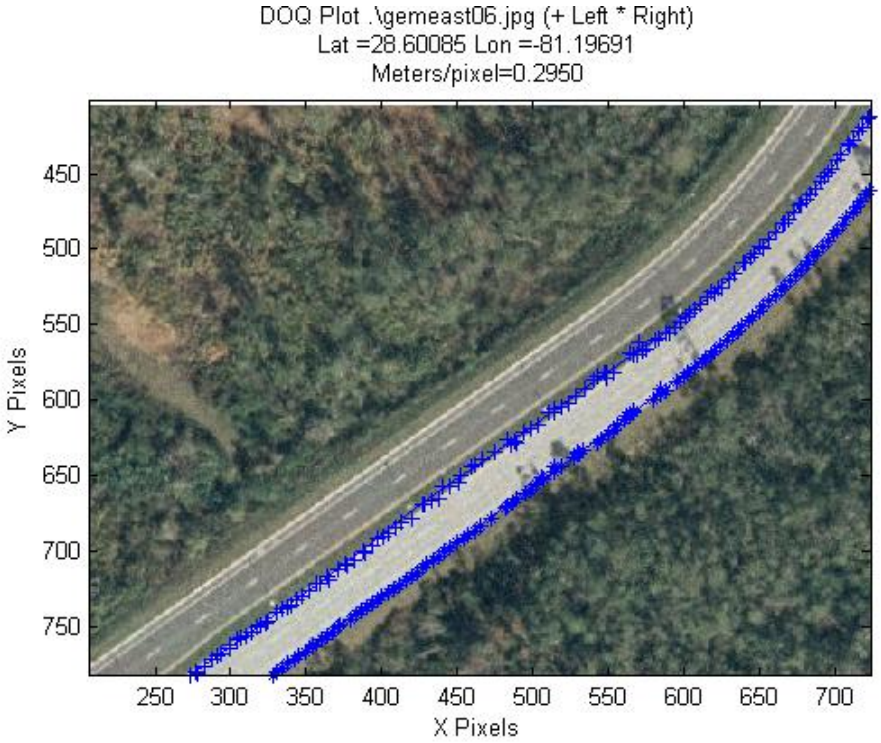

Figure 87 – Gemeast Road Segment Boundaries Overlayed on DOQ

(Image Courtesy of UCF)

This indicates that a projective transform [58] using GPS truth needs to be done to align the road boundary coordinates to the image boundaries. A projective transform requires knowing the true GPS coordinates of selected points in the image. This information is not available for the road portions of the aerial imagery; therefore, a heuristic approach was used to match manually selected road boundary in the image to the "closest" computed road boundary coordinates. The closest point was selected as the either the nearest computed boundary point to the manually selected point or the computed boundary point with the smallest distance to the vector formed from the origin to manually selected point. The method yielding the best match for each particular road segment was then used to obtain a least squares fit using the following projective transform.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{67}$$

where the x, y and x', y' represent the pixel coordinates of the corresponding truth and road boundary points.

The results of the projective transform alignment are shown in the following figures:

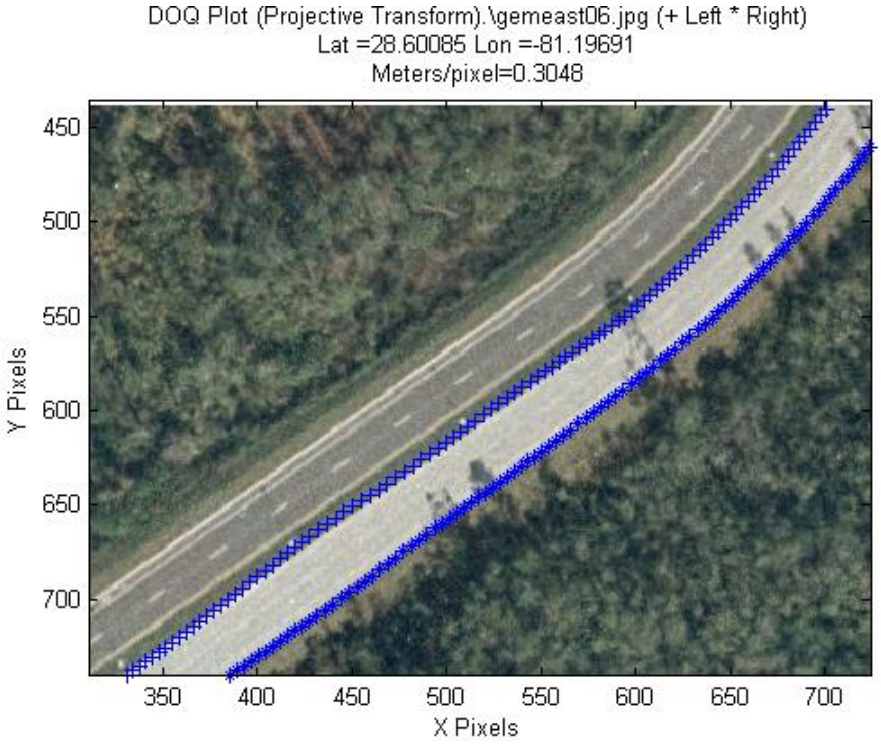

Figure 88 – Gemeast Road Segment Boundaries Overlayed on DOQ

(Image Courtesy of UCF)

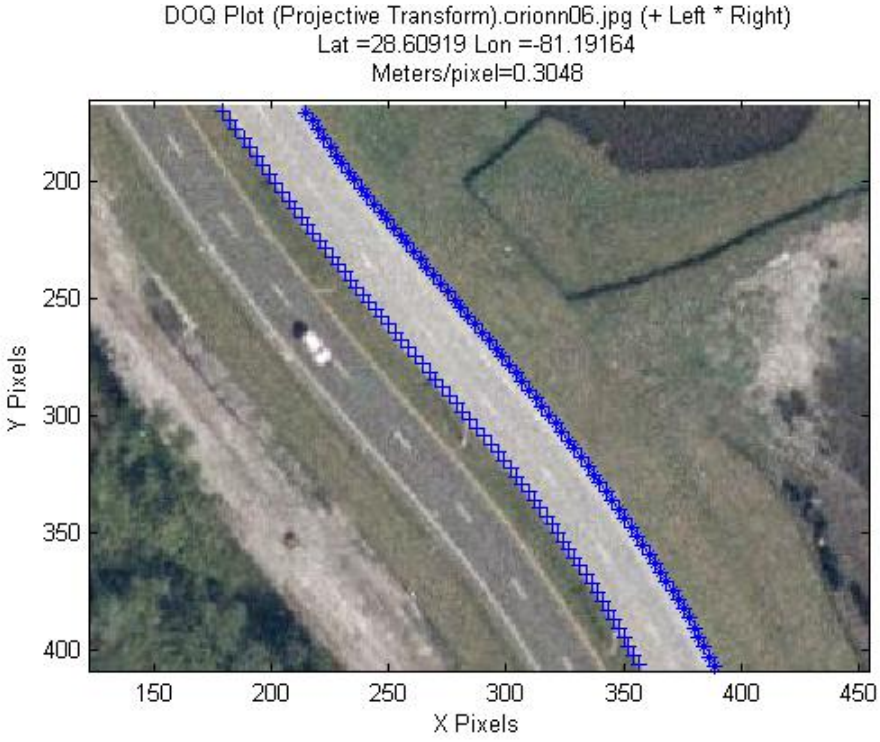

Figure 89 – Orion Road Segment Boundaries Overlayed on DOQ

(Image Courtesy of UCF)

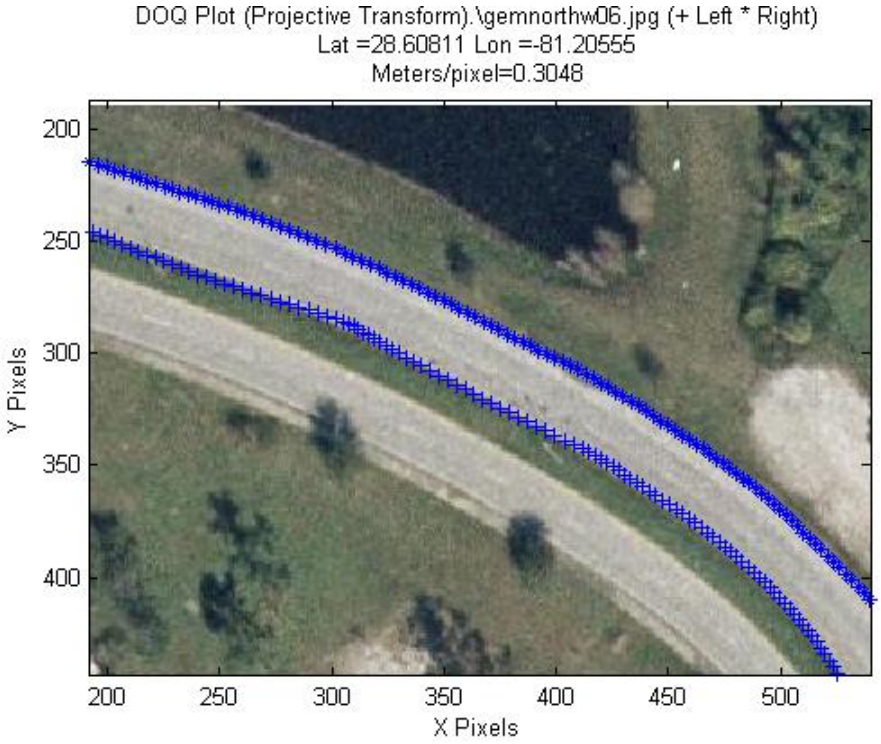

Figure 90 – Gemnorth Road Segment Boundaries Overlayed on DOQ

(Image Courtesy of UCF)

Plots of the boundary errors before and after applying a projective transform are shown in Figure 91, Figure 92 and Figure 93 and show an error less than 1 meter RMS after using the projective transform.

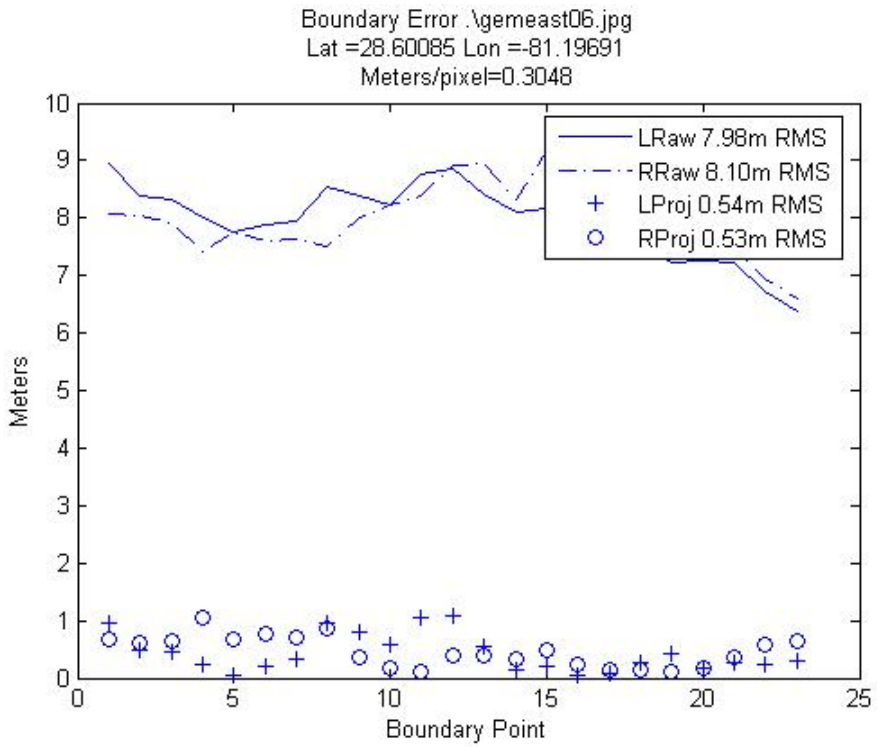

Figure 91 – Gemeast Road Segment Boundary Error

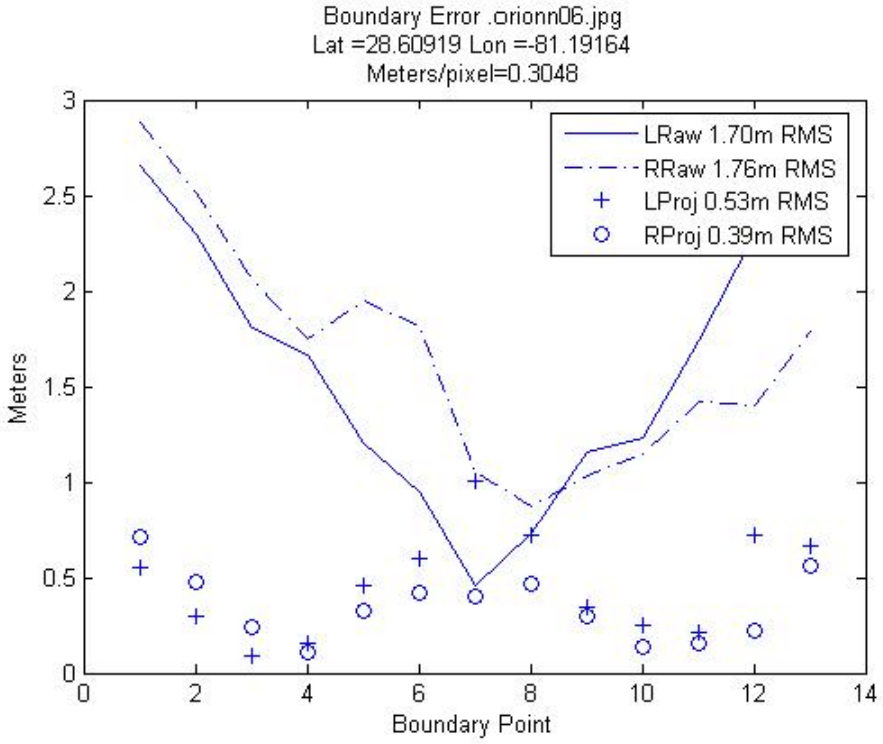

Figure 92 – Orion Road Segment Boundary Error

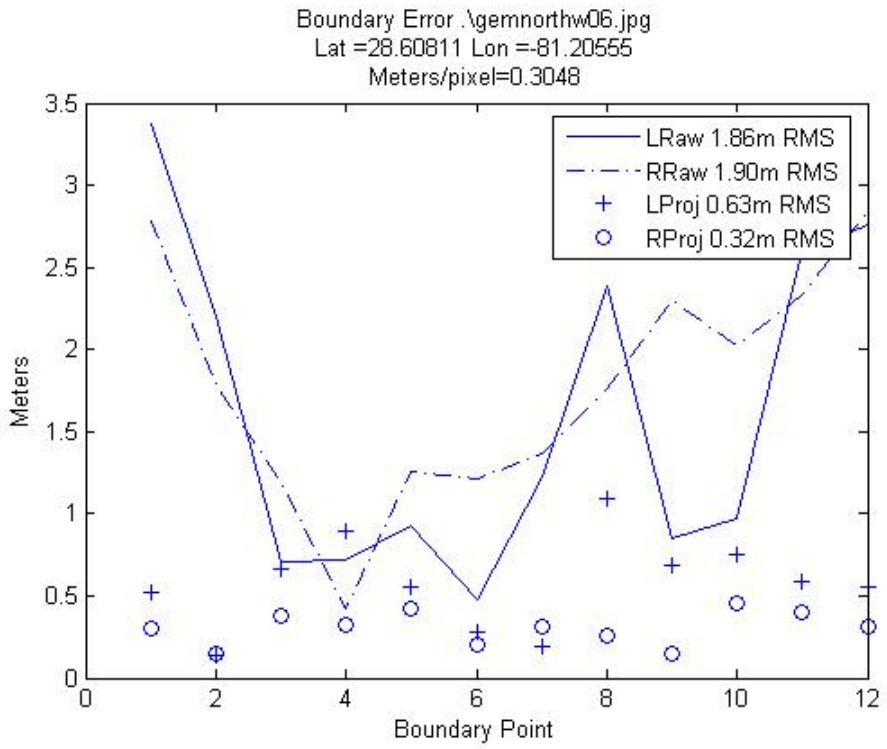

Figure 93 – Gemnorth Road Segment Boundary Error

4.3 3D Model Extraction Results

The following figures are Multigen Creator™ screenshots of the Openflight models of the Gemeast, Orion and Gemnorth road segments. These models were created using the program discussed in section 3.4.7.

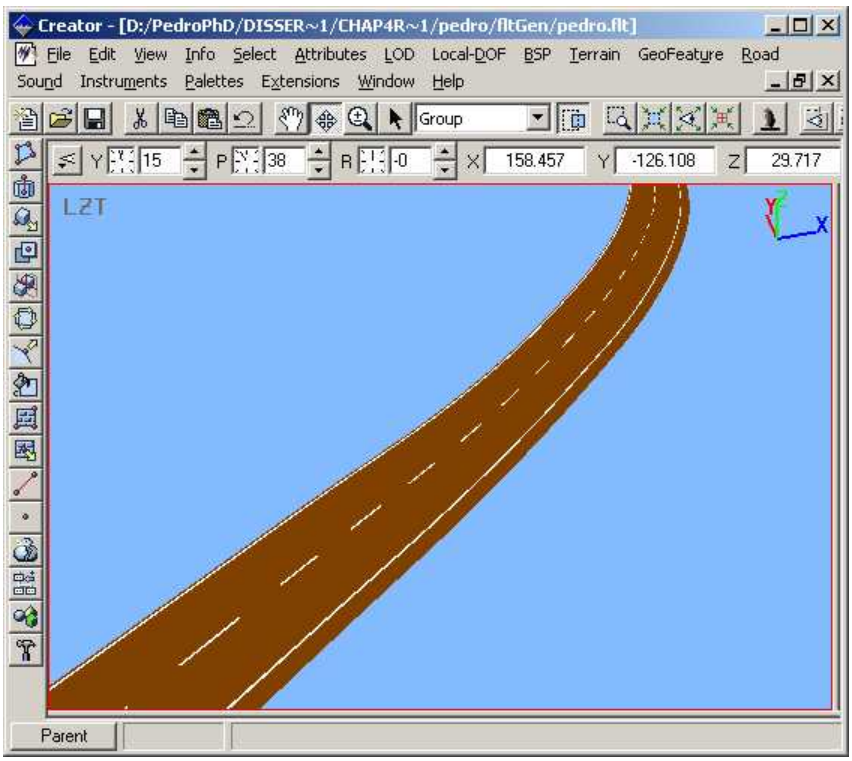

Figure 94 – OpenFlight Model of Gemeast Road Segment

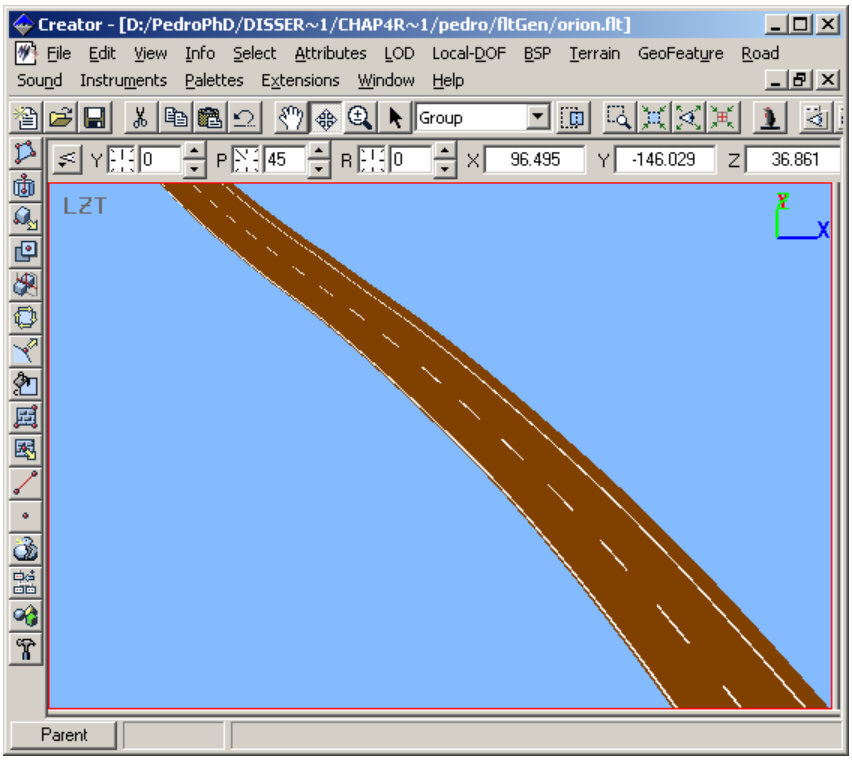

Figure 95 – OpenFlight Model of Orion Road Segment

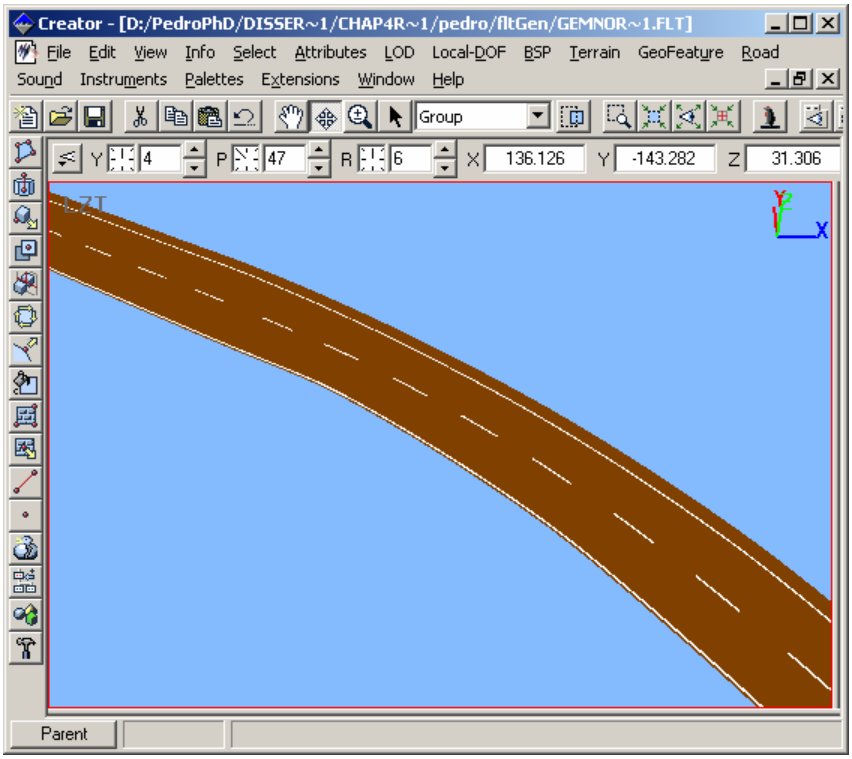

Figure 96 – OpenFlight Model of Gemnorth Road Segment

5 CONCLUSION & SUGGESTIONS FOR FUTURE WORK

5.1 Conclusion

In this dissertation the problem on how to create a georeferenced 3D road model from stereo video was addressed. A method based on using a modified version of the CP algorithm using YUV color and a novel extension of the fast snakes algorithm was proposed to produce stereo class images. The stereo class images were processed using stereo disparity and robust filtering to obtain 3D road boundary information

We have developed a model of our system and used it to predict the system performance. Our boundary accuracy was found to be in agreement with our model prediction once they were properly aligned to georeferenced aerial images of the roads. Our error model can also be used as a design aid for mobile stereo vision processing tasks.

We have successfully extended the road modeling techniques of Bossard [3] to include vision techniques to measure the road width. A comparison of our method with Bossard's techniques is given in the following table:

Table 12 – Comparison of Proposed Method with Bossard's Techniques

	Proposed Method	**Bossard**
Data Collection	Stereo Video and DGPS	Dual DGPS
Approach to finding road boundaries	Color based segmentation with Snakes	Direct measurement of single lane boundaries
Segmentation	Bayesian Classification using UV color histograms	N/A
Automation	After selection of initial boundaries, process is fully automated	Manual editing of boundary points necessary to compensate for GPS dropouts
Model Output	OpenFlight	OpenFlight

	Proposed Method	Bossard
Features	Adapts to changing road widths and color	
Limitations	Works on segments. Susceptible to GPS dropouts Does not handle intersections. Cars and sidewalks cause boundary to wander.	Susceptible to GPS dropouts. Cannot adapt to changing road width.

Matlab software implementing the proposed method along with sample data is available on the UCF Driver Simulation Lab public FTP site at ftp://storage.cecs.ucf.edu/Groups/DSimPublic/. Contact the Driver Simulation Lab administrator at dsim@mail.ucf.edu for downloading instructions.

5.2 Suggestions for Future Work

Future research should be done to address the limitations of the proposed method.

Methods for detecting the loss of a road boundary due to poor segmentation would enable the development of contingencies that would use the opposite boundary to estimate the boundary location. One approach would be to use a lane following technique to bound the extents of the road template, an excursion from this bounded template would indicate a poor boundary.

Hybrid techniques incorporating stereo disparity into the snake algorithms such as those proposed by [72] should be considered. Another area for investigation is to develop a means of selecting optimal snake coefficients given a class of boundaries.

Other improvements to the hardware configuration should be done to improve the quality of the collected road and sensor data.

The data collection platform should be upgraded with faster processors and increased memory to improve the performance of the system at high resolutions where the system was limited to a 5 Hz frame rate.

To improve the DGPS performance the analog cellular modem link should be replaced with a digital wireless cellular card to improve dropout performance. The GPS cards should be upgraded to newer models supporting more satellite channels and providing centimeter level accuracy. This will permit the use of altitude data in developing road models for hilly terrain.

The manually adjusted lenses should be replaced with an auto iris lens system controlled by software via an analog i/o card. Gradient lens filters will also reduce excessive light from the sky thereby improving the color contrast of the collected road data. Alternatively, one could consider replacing the Sony cameras with High Dynamic Range Camera (HDRC©) technology FireWire cameras which have a dynamic range comparable to the human visual system.

Finally, the solid state compass needs to be upgraded to a unit that incorporates gyros into its tilt compensation mechanism. This will improve the accuracy of the navigation solution by providing a backup system that can function during short disruptions of the satellite signals due to terrain or building interference.

APPENDIX: PERMISSIONS

----- Forwarded message from Alberto Broggi <broggi@ce.unipr.it> -----
Date: Wed, 15 Mar 2006 00:37:09 +0100
From: Alberto Broggi <broggi@ce.unipr.it>
Reply-To: broggi@ce.unipr.it
Subject: Re: Permission to reprint images in dissertation
To: pe685631@pegasus.cc.ucf.edu
I see no problem in using the images you mentioned.
Please credit them as coming from
VisLab, Artificial Vision and Intelligent Systems Lab, Univ of
Parma, Italy (vislab.unipr.it)
Thanks,
ALberto Broggi

On Tue, Mar 14, 2006 at 11:19:34AM -0500, pe685631@pegasus.cc.ucf.edu
wrote:
Dr. Broggi,
I am a doctoral student at the University of Central Florida in Orlando, Florida; United States. I would like to reprint some of your webpage images(http://vislab.unipr.it/GOLD/)(see attachments) in my doctoral dissertation (with full accreditation) "Automatic Generation of a Road Database for a Driver Simulator" which will be made available thru UMI, a company that provides copies of doctoral dissertations for a fee. The requested permission extends to any future revisions and editions of my dissertation, including non-exclusive world rights in all languages. These rights will in no way restrict republication of the material in any other form by you or by others authorized by you.

If you are not the copyright holder of these images please inform me who is The copyright holder or if there is no copyright claimed.

Sincerely,
Pedro Claudio

----- Forwarded message from "Kreucher, Chris M." <christopher.kreucher@gd-ais.com> -----
Date: Fri, 24 Mar 2006 10:02:40 -0500
From: "Kreucher, Chris M." <christopher.kreucher@gd-ais.com>
Reply-To: "Kreucher, Chris M." <christopher.kreucher@gd-ais.com>
Subject: RE: Request to reprint figure from article
To: pe685631@pegasus.cc.ucf.edu

Pedro,

You have my permission to use the image in your dissertation, provided you give proper reference to its source.

You may also be interested in results of a related algorithm, called LANA, published in C. Kreucher and S. Lakshmanan, "LANA: A Lane Extraction Algorithm that uses Frequency Domain Features", IEEE Transactions on Robotics and Automation, 15(2):343 - 350, April 1999.

-- Chris

-----Original Message-----
From: pe685631@pegasus.cc.ucf.edu [mailto:pe685631@pegasus.cc.ucf.edu]
Sent: Wednesday, March 22, 2006 2:15 PM
To: Kreucher, Chris M.
Subject: Request to reprint figure from article

Dr. Kreucher,

Thank you for sending me a copy of your Master's Thesis "A Tool for Query and Analysis of MPEG Encoded Video".

I would like to reprint portions of the following figure(see attachments) in my doctoral dissertation "Automatic Generation of a Road Database for a Driver Simulator" which will be Made available thru UMI, a company that provides copies of doctoral Dissertations for a fee.

I have received permission from the publisher, SPIE, as shown below in Their email.

This image is contained in the article "Tracking Lane and Pavement Edges Using Deformable Templates" by K.C. Kluge, C.M. Kreucher and S. Lakshmanan which appeared in the Proceedings of SPIE Vol. 3364 "Enhanced and Synthetic Vision 1998", pages 167-176. The image appears on page 170, Figure 1.

The requested permission extends to any future revisions and editions Of my dissertation, including non-exclusive world rights in all languages. These rights will in no way restrict republication of the material in any other form by you or by others authorized by you.

Sincerely,
Pedro Claudio

----- Forwarded message from "Kluge, Karl C." <KARL.C.KLUGE@saic.com> -----
Date: Wed, 22 Mar 2006 15:54:09 -0500
From: "Kluge, Karl C." <KARL.C.KLUGE@saic.com>
Reply-To: "Kluge, Karl C." <KARL.C.KLUGE@saic.com>
Subject: RE: Request to reprint figure
To: pe685631@pegasus.cc.ucf.edu

Pedro,

You are welcome to use the illustration. If you need explicit permission from the coauthors, let me know -- I have Sridhar's email address, but am uncertain about Chris's address.

Karl

-----Original Message-----
From: pe685631@pegasus.cc.ucf.edu [mailto:pe685631@pegasus.cc.ucf.edu]
Sent: Wed 3/22/2006 3:16 PM
To: kck@cs.cmu.edu
Subject: Request to reprint figure

Dr. Kluge,

I am a doctoral student at the University of Central Florida in Orlando, Florida. I would like to reprint portions of the following figure(see attachments) in my doctoral dissertation "Automatic Generation of a Road Database for a Driver Simulator" which will be made available thru UMI, a company that provides copies of doctoral dissertations for a fee.

I have received permission from the publisher, SPIE, as shown below in their email.

This image is contained in the article "Tracking Lane and Pavement Edges Using Deformable Templates" by K.C. Kluge, C.M. Kreucher and S. Lakshmanan which appeared in the Proceedings of SPIE Vol. 3364 "Enhanced and Synthetic Vision 1998", pages 167-176. The image appears on page 170, Figure 1.

The requested permission extends to any future revisions and editions of my dissertation, including non-exclusive world rights in all languages. These rights will in no way restrict republication of the material in any other form by you or by others authorized by you.

Sincerely,
Pedro Claudio

----- Forwarded message from Sridhar Lakshmanan
<lakshman@engin.umd.umich.edu>

Date: Wed, 22 Mar 2006 22:49:59 -0500 (EST)
From: Sridhar Lakshmanan <lakshman@engin.umd.umich.edu>
Reply-To: Sridhar Lakshmanan <lakshman@engin.umd.umich.edu>
Subject: Re: Request to reprint figure
To: pe685631@pegasus.cc.ucf.edu
Pedro: Fine with me. Go for it!
Sridhar Lakshmanan
On Wed, 22 Mar 2006 pe685631@pegasus.cc.ucf.edu wrote:
Dr's. Kluge, Lakshman
I am a doctoral student at the University of Central Florida in Orlando, Florida. I would like to reprint portions of the following figure(see attachments) in my doctoral dissertation "Automatic Generation of a Road Database for a Driver Simulator" which will be made available thru UMI, a company that provides copies of doctoral dissertations for a fee. I have received permission from the publisher, SPIE, as shown below in their email.
This image is contained in the article "Tracking Lane and Pavement Edges Using Deformable Templates" by K.C. Kluge, C.M. Kreucher and S. Lakshmanan which appeared in the Proceedings of SPIE Vol. 3364 "Enhanced and Synthetic Vision 1998", pages 167-176. The image appears on page 170, Figure 1.
The requested permission extends to any future revisions and editions of my dissertation, including non-exclusive world rights in all languages. These rights will in no way restrict republication of the material in any other form by you or by others authorized by you.
Sincerely,
Pedro Claudio

----- Forwarded message from Beth Huetter <bethh@SPIE.org> -----
Date: Tue, 21 Mar 2006 10:56:54 -0800
From: Beth Huetter <bethh@SPIE.org>
Reply-To: Beth Huetter <bethh@SPIE.org>
Subject: RE:
To: pe685631@pegasus.cc.ucf.edu
Dear Pedro Claudio,
Thank you for your request to reprint Fig. 1 of K.C. Kluge, C.M. Kreucher and S. Lakshmanan "Tracking Lane and Pavement Edges Using Deformable Templates" in Proceedings of SPIE Vol. 3364 "Enhanced and Synthetic Vision 1998", pages 167-176.
Publisher's permission is hereby granted under the following conditions: (1) you obtain permission of the author(s); (2) the material to be used has appeared in our publication without credit or acknowledgment to another source; and (3) you credit the original SPIE publication. Include the authors' names, title of paper, volume title, SPIE volume number, and year of publication in your credit statement.
Please write back if you have any other questions or concerns.
Sincerely,
Beth Huetter for
Eric Pepper, Director of Publications
International Society for Optical Engineering (SPIE)
P.O. Box 10, Bellingham WA 98227-0010 USA
360/676-3290 (Pacific Time) eric@spie.org
-----Original Message-----
From: pe685631@pegasus.cc.ucf.edu [mailto:pe685631@pegasus.cc.ucf.edu]
Sent: Tuesday, March 21, 2006 10:00 AM
To: reprint_permission
Subject:
I am a doctoral student at the University of Central Florida in Orlando, Florida. I would like to reprint portions of the following figure(see attachments) in my doctoral dissertation "Automatic Generation of a Road Database for a Driver Simulator" which will be made available thru UMI, a company that provides copies of doctoral dissertations for a fee.
This image is contained in the article "Tracking Lane and Pavement Edges Using Deformable Templates" by K.C. Kluge, C.M. Kreucher and S. Lakshmanan Which appeared in the Proceedings of SPIE Vol. 3364 "Enhanced and Synthetic Vision 1998", pages 167-176. The image appears on page 170, Figure 1.
The requested permission extends to any future revisions and editions of My dissertation, including non-exclusive world rights in all languages. These rights will in no way restrict republication of the material in any other form
Sincerely,
Pedro Claudio

REFERENCES

[1] Eos Systems Inc., "PhotoModeler Pro [computer program], 2006, http://www.photomodeler.com

[2] D. Guo, "Creating geo-specific road databases from aerial photos for driving simulation," Ph.D. dissertation, University of Central Florida, Orlando, Florida, U.S.A, 2005

[3] B. A. Bossard, "Generation of real-time synthetic environment using a mobile sensor platform," M.S. thesis, University of Iowa, Iowa City, IA, U.S.A, 2003

[4] M. A. Turk, D. G. Morgenthaler, K. D. Gremban, and M. Marra, "VITS-a vision system for autonomous land vehicle navigation," presented at Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1988.

[5] T. Kanade, C. Thorpe, and W. R. L. Whittaker, "Autonomous Land Vehicle Project at CMU," presented at Proc. 1986 ACM Computer Conference, 1986.

[6] Board on Army Science and Technology, *Technology development for Army unmanned ground vehicles*. Washington, D.C.: National Academies Press, 2002.

[7] D. Pomerleau, "Neural Networks For Intelligent Vehicles," *Intelligent Vehicles' 93 Symposium*, pp. 19-24, 1993.

[8] C. Thorpe, M. H. Hebert, T. Kanade, and S. A. Shafer, "Vision and navigation for the Carnegie-Mellon Navlab," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 10, pp. 362-373, 1988.

[9] J. D. Crisman and C. E. Thorpe, "SCARF: a color vision system that tracks roads and intersections," presented at Robotics and Automation, IEEE Transactions on, 1993.

[10] D. Pomerleau, "RALPH: Rapidly Adapting Lateral Position Handler," presented at Intelligent Vehicles '95 Symposium, Detroit, USA, 1995.

[11] M. Bertozzi and A. Broggi, "Vision-based Vehicle Guidance," vol. 30, pp. 55-55, 1997.

[12] E. D. Dickmanns, R. Behringer, D. Dickmanns, T. Hidebrant, M. Maurer, F. Thomanek, and J. Schielen, "The seeing passenger car VaMors-P," *IEEE Symposium on Intelligent Vehicles*, pp. 68, 1994.

[13] R. Gregor, M. Lutzeler, M. Pellkofer, K. H. Siedersberger, and E. D. Dickmanns, "EMS-Vision: a perceptual system for autonomous vehicles," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 3, pp. 48-59, 2002.

[14] S. Denasi, C. Lanzone, P. Martinese, G. Pettiti, G. Quaglia, and L. Viglione, "Real-time system for road following and obstacle detection," *Proceedings of SPIE*, vol. 2347, pp. 70, 2003.

[15] M. Bertozzi, A. Broggi, and A. Fascioli, "Stereo inverse perspective mapping: Theory and applications," *Image and Vision Computing*, vol. 16, pp. 585-590, 1998.

[16] A. Broggi, G. Conte, F. Gregoretti, C. Sansoe, and L. M. Reyneri, "The Paprica massively parallel processor," *Massively Parallel Computing Systems, 1994., Proceedings of the First International Conference on*, pp. 16-30, 1994.

[17] M. Bertozzi, A. Broggi, G. Conte, and A. Fascioli, "Obstacle and lane detection on ARGO," *Intelligent Transportation System, 1997.ITSC 97.IEEE Conference on*, pp. 1010-1015, 1997.
[18] M. Bertozzi and A. Broggi, "GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection," *Image Processing, IEEE Transactions on*, vol. 7, pp. 62-81, 1998.
[19] I. Masaki, *Vision-based vehicle guidance*. New York: Springer-Verlag, 1992.
[20] C. Rasmussen, "Combining laser range, color, and texture cues for autonomous road following," 2002.
[21] K. Kluge and C. Thorpe, "The YARF System for Vision-Based Road Following," *Mathematical and Computer Modelling*, vol. 22, pp. 213-233, 1995.
[22] P. J. Huber, "Robust Estimation of a Location Parameter," *The Annals of Mathematical Statistics*, vol. 35, pp. 73-101, 1964.
[23] K. Kluge and S. Lakshmanan, "A deformable-template approach to lane detection," presented at Intelligent Vehicles '95 Symposium., Proceedings of the, 1995.
[24] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. Teller, and E. Teller, "Equations of state calculations by fast computing machines," *Journal of Chemical Physics*, vol. 21, pp. 1087-1092, 1953.
[25] D. Pomerleau and T. Jochem, "Rapidly adapting machine vision for automated vehicle steering," *Expert, IEEE [see also IEEE Intelligent Systems]*, vol. 11, pp. 19, 1996.
[26] A. Polk and R. Jain, "A parallel architecture for curvature-based road scene classification," *Springer Series In Perception Engineering*, pp. 284-299, 1992.
[27] V. Graefe and K. D. Kuhnert, "Vision-based autonomous road vehicles," *Springer Series In Perception Engineering*, pp. 1-29, 1992.
[28] J. P. Snyder, *Map Projections - A Working Manual*. Washington: United States Government Printing Office, 1987.
[29] Microprocessors and Microcomputers Standards Committee of the IEEE Computer Society, "IEEE Std 1394-1995, IEEE standard for a high performance serial bus." New York, N.Y., USA: Institute of Electrical and Electronics Engineers, 2000, pp. 196.
[30] Steinberg and Birk, "An empirical analysis of the IEEE-1394 serial bus protocol," *Micro, IEEE*, vol. 20, pp. 58-65, 2000.
[31] D. Wilson, "Video codec and pixel format definitions: YUV Formats," November 19, 2005, http://www.fourcc.org/indexyuv.htm
[32] Sony Corporation, "DFW-V500/DFW-VL500 Technical Manual Version 1.0," 2000.
[33] S. Schönberg, "Using PCI-Bus Systems in Real-Time Environments," Ph.D. dissertation, Technische Universitat Dresden, Dresden, Germany, 2002
[34] Z. Zhang, "A flexible new technique for camera calibration," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, pp. 1330-1334, 2000.
[35] J.-Y. Bouguet, "Camera Calibration Toolbox for Matlab [computer program], 2005, http://www.vision.caltech.edu/bouguetj/calib_doc/
[36] B. E. Bayer, "Color Imaging Array," U. S. Patent 3,971,065, July 20, 1976

[37] G. C. Holst, *Electro-optical imaging system performance*. Winter Park, FL; Bellingham, Wash.: JCD Pub.; SPIE Optical Engineering Press, 1995.
[38] A. Butkunas, "Random Vibration Analysis and Vehicle Shake," *Institution of Mechanical Engineers*, vol. C111, pp. 161, 1971.
[39] B. Tao, "Demosaicing for Digital Imaging Device Using Perceptually Uniform Color Space," U. S. Patent 6,809,765, Oct 26, 2004
[40] C. Harris and M. Stephens, "A combined corner and edge detector," *Alvey Vision Conference*, pp. 147–151, 1988.
[41] L. Matthies, "Stereo vision for planetary rovers: stochastic modeling to near real-time implementation," *Int.J.Comput.Vision*, vol. 8, pp. 71-91, 1992.
[42] K. A. Mühlmann, D. A. Maier, J. A. Hesser, and R. A. Männer, "Calculating Dense Disparity Maps from Color Stereo Images, an Efficient Implementation," *International Journal of Computer Vision*, vol. 47, pp. 79-88, 2002.
[43] S. Klein, A. M. Bazen, and R. N. J. Veldhuis, "Fingerprint Image Segmentation Based on Hidden Markov Models," presented at Proc. ProRISC 2002, 13th Annual Workshop on Circuits, Systems, and Signal Processing, Veldhoven, The Netherlands, 2002.
[44] D. J. Williams and M. Shah, "A fast algorithm for active contours and curvature estimation," *CVGIP: Image Understanding*, vol. 55, pp. 14-26, 1992.
[45] D. J. Biezad, *Integrated navigation and guidance systems*. Reston, Va.: American Institute of Aeronautics and Astronautics, 1999.
[46] Trimble Navigation Limited, "Trimble Planning Software [computer program]," ver. 2.7, 2005, http://www.trimble.com/planningsoftware_ts.asp
[47] S. Belongie, "Rodrigues' Rotation Formula," From MathWorld--A Wolfram Web Resource, created by Eric W. Weisstein, June 15, 2003, http://mathworld.wolfram.com/RodriguesRotationFormula.html
[48] Novatel Inc., *GPSCard Command Description Manual, Rev 3*, 1998.
[49] I. Ulrich, "CMU 1394 Digital Camera Driver [computer program], The Robotics Institute, Carnegie Mellon University, 2004, http://www.cs.cmu.edu/~iwan/1394/index.html
[50] A. Lee, "VirtualDub [computer program]," ver. 1.5.10, 2006, http://www.virtualdub.org/
[51] B. Rudiak-Gold, "AviSynth [computer program]," ver. 2.5, 2004, http://www.avisynth.org/
[52] Microsoft Corporation, "QueryPerformanceCounter Function," http://msdn.microsoft.com/library/en-us/winui/winui/windowsuserinterface/windowing/timers/timerreference/timerfunctions/queryperformancecounter.asp
[53] Novatel Inc., "GPSolution [computer program]," ver. 3.1, 2000, http://www.novatel.com/support/fwswupdates.htm
[54] W. S. Rupprecht, "WinDGPS [computer program]," ver. 2.0A, 1999, http://www.wsrcc.com/wolfgang/gps/dgps-ip.html
[55] S. Hill, "IP->Com, Version 2.5.21 [computer program], 2005, http://members.lycos.co.uk/ipcom/
[56] Z. Zhang, "A Flexible New Technique For Camera Calibration, Technical Report MSR-TR-98-71," Microsoft Research, Report 1998.

[57] D. W. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, pp. 431-441, 1963.
[58] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*: Cambridge University Press, 2004.
[59] B. W. Caprile and V. W. Torre, "Using vanishing points for camera calibration," *International Journal of Computer Vision*, vol. 4, pp. 127-139, 1990.
[60] E. W. Weisstein and etc, "Rodrigues' Rotation Formula," personal interview,
[61] D. Brown, "Lens Distortion for Close-Range Photogrammetry," *Photometric Engineering*, vol. 37, pp. 855–866, 1971.
[62] R. Kjeldsen and J. Kender, "Finding skin in color images," *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pp. 312â€“317, 1996.
[63] K. N. Plataniotis and A. N. Venetsanopoulos, *Color Image Processing and Applications*: Springer, 2000.
[64] A. Ford and A. Roberts, "Colour Space Conversions," July 7, 2001, http://www.poynton.com/PDFs/coloureq.pdf
[65] R. C. Gonzalez and R. E. Woods, *Digital image processing*. Reading, Mass.: Addison-Wesley, 1992.
[66] M. I. Kass, A. I. Witkin, and D. I. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, pp. 321-331, 1988.
[67] American Outland, "GeoPosCalc [computer program]," ver. 1.0.0.3, 2004, http://www.american-outland.com
[68] U. S. Army Corps of Engineers, "CorpsCon [computer program]," ver. 6.0.1, U. S. Army Corps of Engineers, 2005, http://crunch.tec.army.mil/software/corpscon/corpscon.html
[69] National Imagery and Mapping Agency, *Department of Defense World Geodetic System 1984: Its Definition and Relationships with Local Geodetic Systems, NIMA TR8350.2*. Bethesda, MD: National Imagery and Mapping Agency, 1997.
[70] P. Fletcher, "UCF Mapping Info and Orthorectified Aerial Map," P. Claudio, personal interview, March 24, 2006
[71] U. S. Army Corps of Engineers, "Topographic Surveying; EM 1110-1-1005," http://www.usace.army.mil/usace-docs/eng-manuals/em1110-1-1005/toc.htm
[72] R. Ghaffari, "Snake Contours in Three-Dimensions from Colour Stereo Image Pairs," M.S. thesis, Simon Fraser University, Vancouver, B.C. Canada, 2005