## Title
Vision-Based Following of Structures Using an Unmanned Aerial Vehicle (UAV)

Authors
Rathinam, Sivakumar
Kim, ZuWhan
Sengupta, Raja

Publication Date
2006-03-21

Institute of Transportation Studies
University of California at Berkeley

# Vision-Based Following of Structures Using an Unmanned Aerial Vehicle (UAV)

**Sivakumar Rathinam, ZuWhan Kim and Raja Sengupta**

RESEARCH REPORT
UCB-ITS-RR-2006-1

March 2006
ISSN 0192 4095

# Vision Based Following of Structures using an UAV

Sivakumar Rathinam*, ZuWhan Kim and Raja Sengupta
Center for Collaborative Control of Unmanned Aerial Vehicles†, University of California, Berkeley.

*Abstract*— **Inspecting and monitoring oil-gas pipelines, roads, bridges, canals are very important in ensuring the reliability and life expectancy of these civil systems. An autonomous Unmanned Aerial Vehicle (UAV) can decrease the operational costs, expedite the monitoring process and be used in situations where a manned inspection is not possible. This paper addresses the problem of monitoring these systems using an autonomous UAV based on visual feedback. A single structure detection algorithm that can identify and localize various structures including highways, roads, and canals is presented in the paper. A fast learning algorithm that requires minimal supervision is applied to obtain detection parameters. The real time detection algorithm runs at 5 Hz or more with the onboard video collected by the UAV. Both hard ware in the loop and flight results of the vision based control algorithm are presented in this paper. An UAV equipped with a camera onboard was able to track a 700 meter canal based on vision several times with a average cross track error of around 10 meters.**

## I. Introduction

This paper describes a control and image processing system to enable an Unmanned Aerial Vehicle (UAV) to track structures like oil-gas pipelines, roads, bridges, canals etc. In particular the system has been tested with a road and an aqueduct. We are motivated by the vast infrastructure inspection industry using helicopter pilots. Infrastructures such as oil pipelines, or roads, are imaged by helicopter pilots to monitor their performance, detect faults, etc. UAV's could be a cheap way of executing these inspection functions, potentially revolutionizing the economics of this industry. Subsection I-A develops this point in some depth in the context of freeway and pipeline inspection. Our ultimate goal is to equip UAV's with the autonomy to track a specified structure and deliver images of the entire structure having the required quality.

This paper describes an important component of such autonomy. A UAV carrying out inspection activities should be able to localize itself relative to the structure it is inspecting and control itself to stay on top of the structure. For example, when flying over the California aqueduct, we are able to keep the UAV at an altitude of 100 meters and within 10 meters of the centerline of the canal. If the UAV stays on top of the structure with this kind of precision, it should be possible to control the imaging or inspecting sensor to produce images with the desired coverage and precision. We are currently working on the UAV control system for the inspecting sensor that will deliver images good enough for the detection of structural damage, leaks, etc.

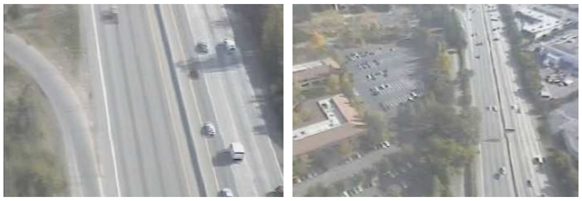

Fig. 1. Example image frames from traffic monitoring video clips. Video courtesy of MLB Company's Bat UAV; http://www.spyplanes.com.

The structure tracking control system in this paper is based on machine vision. We use vision because cameras are cheap, light, and consume little power. All of these are factors when using UAV's. We explore vision based control rather than GPS waypoint control, because the machine vision gives us the tracking error directly relative to the structure rather than via a coordinate system fixed to the earth. This will make the system robust to GPS inaccuracies. In the long run, a better tracking system may be obtained by fusing both vision and GPS.

We use GPS to guide the UAV to the vicinity of the structure and line it up. We then learn the structure from a sample image the UAV obtains. Once features of the structure are learnt from this sample image, the system is designed to keep recognizing the structure in subsequent images and track it. The following are the main contributions of this work:

1) A single algorithm can identify and localize various structures including highways, roads, and canals. A fast learning algorithm that requires minimal supervision is applied to obtain detection parameters. This algorithm has been tested with the onboard video collected by the UAV flying over highways[1] and canals[2]. The algorithm runs at 5 Hz or more.
2) An UAV equipped with a camera onboard was able to track a 700 meter canal based on vision several times with a average cross track error of around 10 meters.

### A. Motivation

- **Traffic Surveillance:** Unmanned Aerial Vehicles equipped with cameras are seen as potential low cost

*Corresponding author: rsiva@berkeley.edu

Preliminary version of the paper is accepted in the $44^{th}$ IEEE CDC-ECC 2005, Seville, Spain.

†Research supported by ONR AINS Program - grant # N00014-03-C-0187,SPO #016671-004

[1] Highway video courtesy of MLB Company's Bat video. Can be viewed at www.spyplanes.com

[2] video obtained by flying a UAV over a canal at Crows landing Naval Auxiliary Landing Field, Patterson, CA

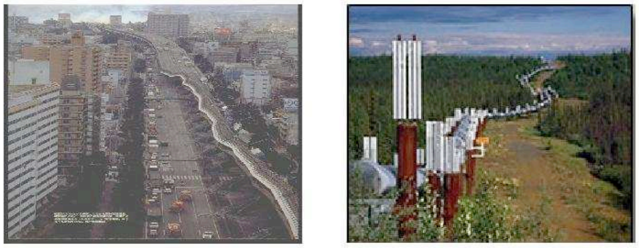

Fig. 2. Possible target applications: a) surveillance (Hanshin Expressway, Japan), and b) oil pipeline inspection (Alaska oil pipeline).

platforms that can be used for traffic surveillance (figure 1). The transportation departments in states such as Ohio [1], Florida [2] and Georgia [3] have already initiated research in this area. The information that would be useful to collect in this application are lane changes of the vehicles, average inter-vehicle distances, heavy vehicle counts, accidents, vehicle trajectories and type of vehicles. Even though loop detectors can be used, they provide only local information and cannot provide details like lane changes of vehicles. On the other hand, a UAV equipped with a camera can provide a global view of freeways with the relevant information thus enhancing the real time monitoring aspect. Also, the UAVs have an advantage over manned vehicles as they can fly at lower heights and can be used in conditions when weather conditions are not suitable for flying with pilots [1]. Collecting useful information with UAVs in this application has two problems to it: The first problem is to keep the camera in view of the road and the second is to process the image to collect relevant data. This paper attempts to address the first problem where the vehicle is trying to track the road in the presence of Global Positioning system (GPS) inaccuracies. Compensating for GPS inaccuracies would be useful in applications where the roads have been shifted from their original position as shown in figure 2. Figure 2 shows the effect of an earthquake (1995) on the Hanshin Expressway in Japan. The road has been completely displaced by several meters. It might be difficult to obtain instant feedback of the road conditions after such disasters. Also, the difficulty of relying on GPS and Geographic Information System (GIS) is that the GIS information should be retrieved beforehand and the UAVs path should be roughly pre-planned. The GIS information should be available and current, which may not always be the case. In addition, in certain cases, such as in a combat situations, GPS may not be available also.

- **Oil Pipeline Inspection:** The Trans-Atlantic pipeline system (figure 2) transports oil from the north slope of Alaska to Valdez-the most northern most ice free port in Alaska. The pipeline system [4] is 800 miles long and supplies approximately 17 percent of the crude oil produced in the United States. Majority of this pipeline system lies within a 7.00 Richter zone with the zone near the Valdez port (maximum earthquake prone zone) at 8.5 Richter magnitude [5] [6]. Based on the crude oil price estimates in 2002, the shutdown of this pipeline system represents an economic impact of 1 million dollars per hour [5]. Monitoring and maintenance of both pre and post earthquake conditions plays a important role in protecting the structural integrity and functional reliability of this system. This was best illustrated by the satisfactory performance of the system during the Denali fault earthquake (7.9 Richter scale) in 2002 due to a comprehensive and focused inspection effort by Alyeska [4]. There was no oil spilled and the entire pipeline operation was resumed after 66 hours [5]. This system which is on this massive scale requires sustained monitoring efforts to increase its life expectancy. Currently, Alyeska conducts routine aerial surveillance at least twice per month. In addition to the current efforts, it would be of immense value if the same monitoring operations could be automated by an autonomous unmanned aerial vehicle. The pipeline system is designed to be zigzag to allow for sideways movements that are a result of temperature changes. Even though the exact GPS location of this zigzag pattern might be known before an occurrence of a earthquake, exact location of these pipelines may not known after its occurrence. For example, the pipeline systems around the Denali fault are designed so that they can move 20 feet horizontally and 5 feet vertically. Hence, an autonomous unmanned aerial vehicle that navigates based both on visual information would be very useful for this application also.

The underlying problem in these applications is that of requiring an unmanned aerial vehicle to follow structures[3] based on visual feedback. This is the problem that is addressed in the paper. Structures could be pipelines, roads, canals, power grids etc.

## II. Related Literature

Vision based following has engaged researchers for nearly two decades. Almost all the work done in this area relevant to this work is on *ground vehicles following roads or lanes*. This section is organized as follows: First, most of the techniques that are used for road detection using ground vehicles is discussed; then the road following strategies that have worked well in speeds closer to that of an UAV is presented next; and finally the previous work related to UAVs and the current approach is discussed.

The road detections algorithms that exist in the literature can be categorized based on the kind of the roads they can handle. The roads can be mainly divided as follows:

- Roads with explicit lane marks or boundaries

[3] Structure is a entity made up of a number of parts that are held or put together in a particular way. Depending on the application, these parts could be pipelines, road segments, power grids etc.

- Roads with differences in color or texture compared to the non-road sections

Algorithms in the literature dealing with roads having **explicit lane marks or boundaries** generally use vanishing points to help detect lane boundaries. The idea is that if the boundaries of the road are assumed to be parallel and lie on the ground plane, they meet in the image at a pixel location called the vanishing point. Vanishing points can be found by first finding all the lines in the image and then using a Hough transform. The lines in the image can themselves be found again using a Hough transform. Apart from accounting for the perspective effect of the cameras, the vanishing points help identify roads even when parts of the road are covered by shadows. For example, even if there are shadows covering portions of road, as long as there are line segments (road boundaries) that have a dominant orientation towards the vanishing point, the information in a vanishing point can used to cover the shadow regions. This idea can also be extended to curved roads as given in [7]. In [7], the roads are split into horizontal strips and vanishing points are found for each strip. If all the road boundaries in the different sections are assumed to be in the ground plane, then the vanishing points from all the sections lie on a line (called the vanishing line). Again this vanishing line acts as a global constraint when finding curved roads.

In [8],[9] & [10] the image is first rectified using the position of the camera and the internal camera parameters. The pixels that have relative superiority (in brightness) over neighboring pixels are identified. Then the best concatenation of pixels that have road markings are identified. These algorithms do not assume any road model and found the road just by collecting low level features. There are also algorithms that use a specific model like lines, parabolas or splines to describe a road. Essentially, the problem is then to identify the set of parameters that describe the different road models. In [11], the road are modeled as lines. The algorithm in [11] uses the fact that each image should have a maximum number of line segments along a common orientation, the line segments are clustered based on the orientation of the segments and total length of the segments. Finally road boundaries are picked from the remaining set. In [12], the road is represented by a set of piecewise line segments called line-snakes. The lines are picked initially by dynamic programming and then a energy function is minimized to perturb the lines to fit the road edges in the successive frames. In [13] & [14] roads are modeled as a parabola. A cubic B-spline is used to represent the road in [7]. This way of representing the curve uses fewer parameters (control points) but the curve can take arbitrary shapes. Vanishing points act as global constraints to initialize this B-spline. Then an iterative algorithm is used to update the control points of the curve for successive frames.

There are also region based methods that primarily use **color or texture** as the main source of difference between the road and the non-road sections. These methods use the three color components to differentiate between road and non-road. In [15], clusters of pixels with approximately similar RGB values are identified. Then using the mean RGB value of each cluster, neural networks is used to classify them as road or non-road. Then, using the road segments, the center line of the road is found. In [16],[17] bayesian methods are used to mainly identify the road and non-road sections. The algorithm given in [17] is a bit more general in the sense that they can detect road intersections. Also [18] presents a road detection system primarily based on color. In both [17] and [18], motion cues are used across frames to help classification. In [18], only two road classes (sunny and shaded) are used where as [17] accommodates more road appearances. But the road following system was fast enough to follow roads at 20 km/h. All the algorithms using neural nets and bayesian classifiers need to be trained using a sample image in the beginning.

The aerial image analysis literature also has road detection algorithms [19]. However, aerial image analysis applications do not usually require realtime computation. Most of the proposed algorithms are non-realtime, and focused on rural roads with low-resolution.

Ozugner [20] provides a **control strategy** based on simple proportional control and showed that the closed loop stability can be always be guaranteed by choosing an appropriate look ahead distance. Dickmans et. al [21] provide a road detection algorithm based on extended kalman filter and employed a control strategy based on full state feedback. The vehicle was tested at speeds up to 100 m/s in the race track(Autobahn). [22] provide a simple edge detection algorithm and compare different control strategies (lead-lag control, full state feedback and input-output linearization) for the road following problem. The vehicle moved at a speed of around 70 km/hr.

There have been many approaches to vision based landing [23] [24] and navigation [25] [26] for UAVs. Structure following by small autonomous vehicles is a relatively new area. All the known previous work uses artificial markings such as special landing patterns to enable UAV localization. Our previous vision based road following work on a short runway (refer [27]) was the first contribution in this area without the use of any artificial markings. In [27], Bayesian RGB pixel classifiers were applied to detect the lane markings in the middle of the road (figure 3). Then the road was localized by fitting a line to the detected lane markings. Although the same approach may be applied to some other types of roads by re-learning the road and/or lane colors, it is still not adequate to apply such algorithms to highways or other local streets unless the UAV flies at a low altitude so that the lane markings are visible.

### A. *Our Approach*

Our road detection approach is clearly distinguishable from the previous work available in the literature. While most of the previous work use model-based detection algorithms, we introduce a semi-supervised learning algorithm (which can also be done on-line with simple manual verification) to detect many kinds of structure including highways, roads and canals. No lane markings are required, but the assumption is

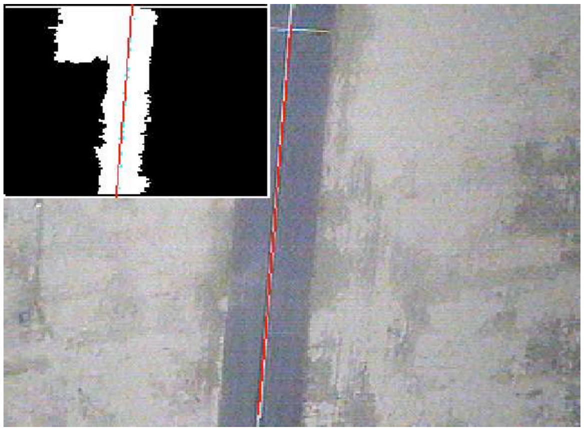

Fig. 3. Our previous experiment [27]. First vision-based UAV navigation on outdoor environment without artificial markings. The experiment was done on a short runway. The road detection was based on off-line learning (supervised) of lane marking and road pixel colors.

that the target structure is locally linear (consists of lines and curves) and has distinguishable boundary to the background. In addition, the user intervention in the learning is minimal. All we need is to pick an sample image for learning and verify the result.

The algorithm proposed in this work has two main components: 1) learning the target structure from a single sample image and 2) realtime detection of the structure. The learning algorithm automatically generates a *cross-section profile* of the target structure with the boundaries marked in it. The detection algorithm uses this cross-section profile to fit a cubic-spline curve to the target structure.

Once the equation of the curve to be followed is obtained, following it by a non holonomic vehicle has been addressed in the literature [28] [29]. In [28], the problem of following a curve is formulated for a unicycle kinematic model in the moving frame. They primarily present local stability and *simulation* results of the control law. This control law was primarily used for the vision based control experiments in this work. This will be discussed in more details in the later sections.

## III. Problem Setup

In this paper, the fixed wing airplane is modeled as a unicycle model traveling at a linear speed $v$ with a bounded angular rate $\omega$. A point $(x, y)$ that is stationary in the inertial frame (north,east directions), as seen from the moving airplane, evolves according to the following equations:

$$\begin{aligned} \dot{x}(t) &= \omega y(t) - v, \\ \dot{y}(t) &= -\omega x(t), \\ \omega(t) &\in [-a, +a], \end{aligned} \tag{1}$$

where $a$ is the bound on the yaw rate command. The structure (road, canal, pipeline etc) is assumed to be a simple curve $\nu(x, t)$ as shown in the figure 4 lying on the ground plane. Moreover, the curve is assumed not to self intersect or wind up. As the vehicle moves the curve $\nu(x, t)$ evolves according to $\frac{\partial \nu}{\partial t} = -\omega x - \frac{\partial \nu}{\partial x}(\omega \nu - v)$. A vehicle tracking a curve perfectly implies that at all times $\nu(0, t) = 0$ and $\frac{d\nu(0,t)}{dx} = 0$. The avionics package on the airplane ensures that the vehicle is flying level at a fixed commanded altitude. Hence, the relative $(x, y)$ coordinates of the curve (structure) as seen from the moving frame is all that is required to control the vehicle. The detection algorithm explained in the next section identifies the structure and outputs the image coordinates of points on the structure. These image coordinates are then converted to $(x, y)$ coordinates of the curve as seen from the moving airplane by applying the appropriate transformation. The goal of the controller is to choose the yaw rate command $\omega$ based on the measurements from the vision sensor and control the evolution of the curve $\nu(x, t)$ in order to drive the vehicle along it.

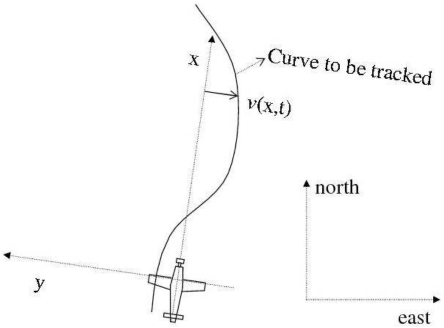

Fig. 4. Problem setup

## IV. Learning and Real-time Detection of the Structure

The algorithm consists of three steps. First is the learning phase where a sample image is used to generate a *cross-section profile* of the structure. This step is implemented **off line**. The second step matches this *cross-section profile* with the horizontal samples of the target image to find the boundaries and the position of the structure in the target image. A curve fitting algorithm in the third step is then applied to determine the equation of the structure being followed in the image plane. The second and the third steps are implemented **real time** for each target image.

Basically, the approach consists of the following steps:

- Learning the structure from a sample image
  1) Image Rectification
  2) Structure identification and localization
- One dimensional signal matching
- Curve fitting

Before we present the details of each of the steps, the following subsection discusses the geometry of a linear structure and is basic to the algorithm design in the learning and detection steps.

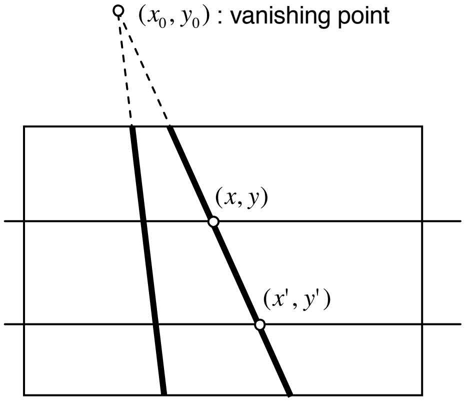

Fig. 5. A geometry of parallel lines.

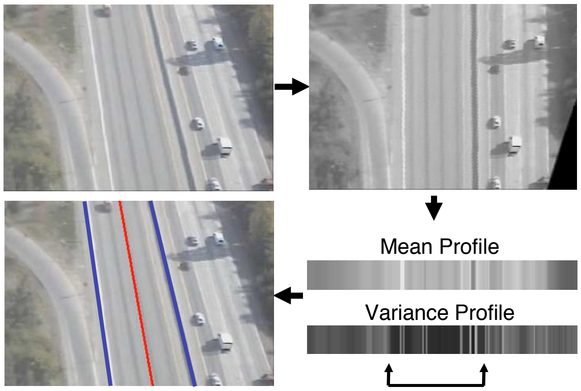

Fig. 6. The suggested learning approach. First, the image is rectified. The statistics of the rectified image (the means and the variances) suggests a rough position of the road.

### A. Geometry of a Linear Structure

Lines along a linear structure (for example, road boundaries and lanes) are roughly parallel to each other. Figure 5 illustrates a geometry of two parallel lines. All parallel lines meet at a single vanishing point, $(x_0, y_0)$, in the image coordinates. Consider two points, $(x, y)$ and $(x', y')$, on any line as shown in figure 5. Since all these points lie on a line, we have,

$$\frac{x - x_0}{y - y_0} = \frac{x' - x_0}{y' - y_0}$$
$$\Leftrightarrow x' = \frac{y_0 - y'}{y_0 - y}x + \frac{y - y'}{y - y_0}x_0 = \alpha x + \beta, \tag{2}$$

where $\alpha$ and $\beta$ are constants given $y$, $y'$ and the vanishing point $(x_0, y_0)$. If $I(.)$ and $I'(.)$ indicate the horizontal intensity profiles given $y$ and $y'$ respectively, then the above equation implies that the intensity profiles of two horizontal cuts of an image can be modelled by a following linear relationship:

$$I(x) = I'(\alpha x + \beta) + \eta_I, \tag{3}$$

where $\eta_I$ is a noise term.

This leads to our definition of a *cross-section profile*. Given a sample image, we define a cross-section profile of a structure by $I(x)$ which minimizes the sum of $\eta_I$ for all scan lines of an image. The main part of our learning procedure is to automatically find a cross-section profile given an image, and use it to localize the structure.

Note that images taken from different perspectives may have incompatible cross-section profiles(even if there is no illumination variation). However, when the airplane is looking roughly straight down at the structure, the variance among the cross-section profiles will be small, and our detection algorithm works well, as shown in the experiments.

### B. Learning the structure from a sample image

The objective of this off line phase is to generate a cross-section profile that represents the desired structure. The boundaries of the structure are assumed to be roughly parallel to each other. Figure 5 illustrates the perspective projection of two parallel lines (i.e. boundaries) in an image. The learning phase consists of two steps: removing this perspective effect in the image (also called image rectification) and choosing an appropriate representation of the structure from this rectified image. Figure 6 illustrates the learning algorithm. The learning algorithm is based on the assumption that the example image contains a linear part of the structure. It is usually not difficult to find a linear part of the structure because the learning requires only a single sample image. Also, the learning procedure does not need to work in realtime. However, since the current implementation of the algorithm takes approximately 200 ms, it can also be used in applications where it is required to *learn the structure on the fly*.

*1) Image Rectification:* The basic idea here is to find the vanishing point of the lines in the image and use the information in that to rectify each horizontal scan of the image. The Vanishing point is found by using a RANSAC style algorithm. The RANSAC (RANdom SAmple Consensus, [30]) algorithm is a robust fitting algorithm that has been successfully applied to computer vision problems. In this algorithm, long line segments are first extracted by using the edges in the sample image. This is followed by line grouping and line fitting as shown in the figure 7. The group of lines that are closer to the vertical axis of the image is selected (This is under the assumption that the sample image was taken with the structure roughly parallel to the vertical sides of the image). Since any two line segments suggest a vanishing point, vanishing point hypotheses can be generated from any pair of line segments. A vanishing point hypothesis created by line segments a and b is evaluated by the calculating its alignment error with all the other line segments (figure 8). The alignment error of a line segment with respect to a hypothesis is defined as the pixel distance, $d$, between an end-point of the line segment and a virtual line, $l$, which connects the hypothesis and the center of the line segment. The best scoring hypothesis is chosen as the vanishing point.

Once the vanishing point is known, the target image can be rectified using equation 3. For each horizontal intensity

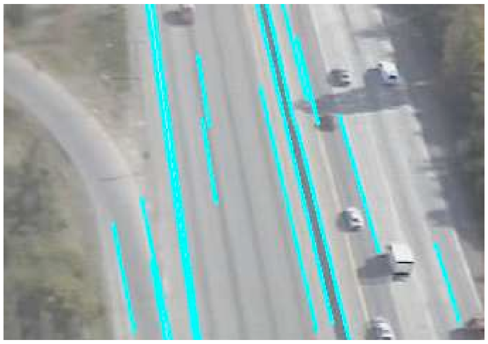

Fig. 7. Vertical line segments are detected to find the vanishing point.

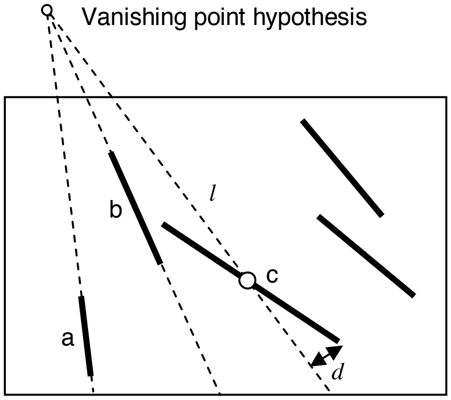

Fig. 8. Evaluation of a vanishing point hypothesis.

profile $I'$ in the sample image, the corresponding profile $I$ in the rectified image can be obtained by a following linear relationship: $I(x) = I'(\alpha x + \beta)$. An example of the rectified image is shown in figure 6.

*2) Structure Identification and Localization:* Given the rectified image, the objective of this part of the learning phase is to output a cross-section profile of the sample image with the boundaries of the structure marked in it. The cross-section profile is obtained by taking the mean of the horizontal scan lines in the rectified image (figure 6). The boundaries of the structure in the cross-section profile is marked based on a couple of observations. The first observation is that the there are sharp intensity changes in the cross-section profile that indicate the possible positions of the boundaries. Also, note that there is a strong vertical correlation of the structure that results in less variance for the corresponding pixels of the cross-section profile. But the background on the other hand has more noise with uncorrelated pixels and hence has more variance. The boundaries are therefore marked by the following procedure:

1) A set of boundary candidates are chosen based on the observation that they have large intensity changes in the cross-section profile. Detected boundary candidates for the highway image are shown in figure 9.
2) For each pair of boundary candidates, the variance profiles between them is examined. The pair that has low variance for the majority of its variance profile is chosen as the left and the right boundaries of the target structure.

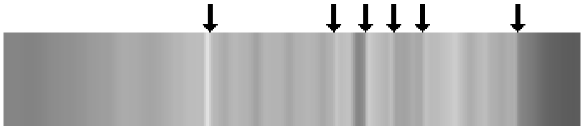

Fig. 9. Structure boundary candidates based on the cross-section profile.

### C. One-dimensional Scan Line Matching

The objective of this algorithm is to find the boundaries and the location of the structure in any given target image. The detection is performed by matching the cross-section profile attained in the learning step with the horizontal scan lines of the given target image. Essentially from equation 3, finding the position and the width of the structure is equivalent to finding the parameters $\alpha$ and $\beta$. The difficulty of estimating $\alpha$ and $\beta$ is that the two-dimensional search space is too large to meet the real time constraint.

For the matching process, a feature matching technique is applied. Let $\mathbf{I}_h$ be the one dimensional intensity profile of a given horizontal scan $h$ in the target image. Let $\mathbf{L}$ be the learnt cross-section profile of the structure. First, two distinct features that are far apart (greater than a chosen threshold) is detected in $\mathbf{L}$. A distinct feature is defined as a feature that has large local intensity changes. For each distinct feature, match candidates are found in $\mathbf{I}_h$ by performing correlation-based template matching. For each and every possible pair of the match candidates, say $\mathbf{p}$, $\mathbf{I}_h$ is scaled and shifted (by finding the appropriate parameters $\alpha$ and $\beta$ in equation 3) such that the two match candidates are aligned at the same positions with those of $\mathbf{L}$'s features. We denote this new profile as $\mathbf{I'}_h(\mathbf{p})$. The correlation[4], denoted by *Corr(*$\mathbf{I'}_h(\mathbf{p})$ of the two vectors, $\mathbf{I'}_h(\mathbf{p})$ and $\mathbf{L}$, indicate the matching quality. Three best hypotheses of $\alpha$ and $\beta$ that correspond to the three highest correlation scores are chosen per scan line for curve fitting.

### D. Curve Fitting

The objective of the curve fitting algorithm is to find the equation of the centerline of the structure being followed on the image plane. For each horizontal scan line of the target image, from the previous step, center hypothesis candidates can be obtained using parameters $\alpha$ and $\beta$. To reduce computation, matches were found for every four scan lines. As shown in the figure 10, it still gives enough matches for curve fitting.

A cubic-spline is used to represent the curve. In a cubic-spline representation, a point $(x_i(t), y_i(t))$ on a curve between $i$-th and $(i+1)$-th control points is represented as:

$$\begin{aligned} x_i(t) &= a_i + b_i t + c_i t^2 + d_i t^3, \\ y_i(t) &= e_i + f_i t + g_i t^2 + h_i t^3, \end{aligned} \tag{4}$$

[4] Only the parts of the profiles $\mathbf{I'}_h(\mathbf{p})$ and $\mathbf{L}$ in between the positions of their respective features are used in the correlation

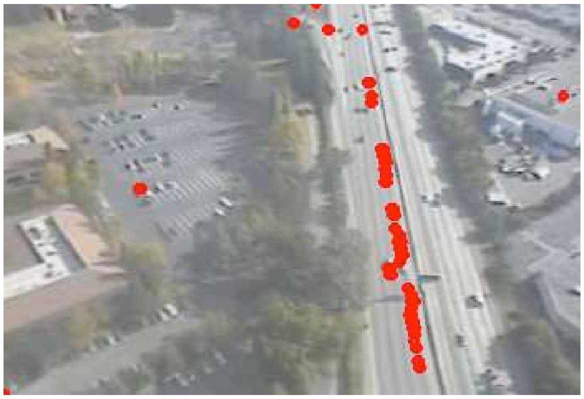

Fig. 10. Center hypotheses of an example image.

where $0 \leq t \leq 1$, $(x_i(0), y_i(0))$ is the $i$-th control point, and $(x_i(1), y_i(1))$ is the $(i+1)$-th control point. The parameters $a_i, \dots, h_i$ are uniquely determined by the control points assuming that the curve is smooth. A cubic-spline curve has the useful property that its control points are actually on the curve. This property is used to apply a RANSAC type algorithm [30] for curve fitting. The curve fitting algorithm randomly chooses a large number of (500 in our implementation) center hypotheses with $4 \sim 6$ control points. Each center hypothesis determines a cubic-spline curve. For each cubic-spline curve hypothesis, *supporting scan line matches* are collected. A *supporting scan line match* of a curve hypothesis is a match where its center point is close to the curve and its width is compatible to other supporting scan line matches. Among the hypothesis candidates, the curve chosen to represent the structure is the one that minimizes a score function. The score function of a curve hypothesis $\mathbf{c}$ is defined as

$$\mathrm{S}(\mathbf{c}) = (1 - \lambda_{\mathbf{c}}) \sum_{p \in m(\mathbf{c})} \mathit{Corr}(\mathbf{I'}_h(\mathbf{p}), \mathbf{L}), \tag{5}$$

where $m(\mathbf{c})$ is the set of supporting scan line matches of $\mathbf{c}$, $\lambda_{\mathbf{c}}$ is a penalty for the minimum description length (MDL) criteria and *Corr(*$\mathbf{I'}_h(\mathbf{p})$,$\mathbf{L}$*)* is the correlation score calculated for the match $\mathbf{p}$ in the one-dimensional scan line matching process. $\lambda_{\mathbf{c}} = 0.05$ was chosen for the curve hypotheses with 5 control points and $\lambda_{\mathbf{c}} = 0.1$ for the curve hypotheses with 6 control points. The output of this algorithm is basically in terms of the control points (image coordinates) of the structure that resulted in the maximum score.

The processed images of the road detection algorithm for video are shown in figure 11. These image coordinates are converted to the ground coordinates using the roll, pitch angles and the height measurements from the sensors onboard of the plane. These ground coordinates determines the curve the UAV must follow. The yaw rate command required to steer the UAV along this curve is discussed in the next section.

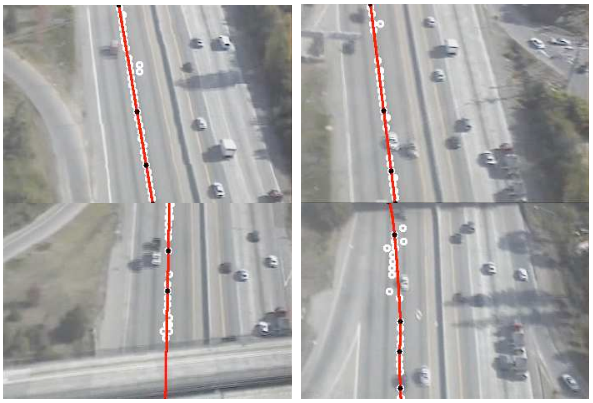

Fig. 11. Detection algorithm working with the video courtesy of MLB company (www.spyplanes.com).

## V. Controlling the vehicle to follow the detected structure

If the vehicle were exactly on the structure at time $t$ with no error (that is $\nu(0,t) = 0$ and $\frac{\nu(x,t)}{dt}\big|_{x=0} = 0$ ), then the yaw rate required to steer the vehicle along the curve is proportional to the curvature, i.e, $\omega(t) = v\frac{d^2\nu(x,t)}{dt^2}$, is sufficient to steer the vehicle along the curve[5]. This is off course an ideal scenario. In the practical situation, there is noise in the measurement process (vision) or in the estimation of the state of the vehicle relative to the curve (Not knowing the exact roll, pitch or yaw etc.). Hence, the control proposed above may not steer the vehicle along the curve for all practical purposes. A novel idea was proposed in [28] where the vehicle chooses a control based on a connecting contour. The connecting contour joins the current location of the vehicle to a point on the curve, thereby satisfying the geometric and non-holonomic constraints. Figure 12 illustrates a connecting contour. This connecting contour $\nu_c$ essentially must satisfy at least the following set of conditions:

$$\begin{aligned} \nu_c(0,t) &= 0 \\ \frac{d\nu_c(x,t)}{dt}\bigg|_{x=0} &= 0 \\ \nu_c(x_c,t) &= \nu(x_c,t) \\ \frac{d\nu_c(x,t)}{dt}\bigg|_{x=x_c} &= \frac{d\nu(x,t)}{dt}\bigg|_{x=x_c} \end{aligned} \tag{6}$$

where $x_c$ is the coordinate along the moving frame where the connecting contour joins the desired path to be followed. The simplest connecting contour that satisfies these four conditions is a polynomial of degree 3. This simple contour has the form:

$$\nu_c(x,t) = \alpha(t)x^3 + \beta(t)x^2 \tag{7}$$

[5] Assuming the maximum curvature is less than the maximum yaw rate of the vehicle.

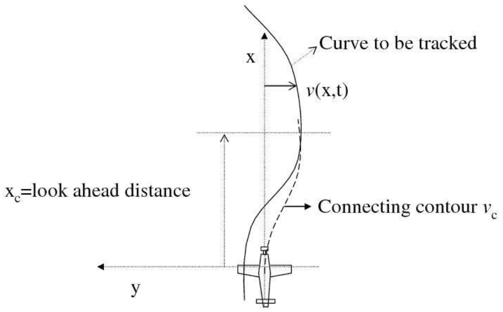

Fig. 12. Illustration of the curve to be tracked and the connecting contour.

where $\alpha(t)$, $\beta(t)$ are given by:

$$\begin{aligned}\alpha(t) &= -2\frac{\nu_c(x_c,t)}{x_c^3} + \frac{\nu_c'(x_c,t)}{x_c^2}\\ \beta(t) &= 3\frac{\nu_c(x_c,t)}{x_c^3} - \frac{\nu_c'(x_c,t)}{x_c^2}\end{aligned}$$

Now, the yaw rate command required to steer the vehicle is chosen to be proportional to the curvature of the connecting contour at the origin. That is,

$$\begin{aligned}\omega(t) &= v\frac{d^2\nu_c(0,t)}{dt^2}\\ &= 2v(3\frac{\nu_c(x_c,t)}{x_c^3} - \frac{\nu_c'(x_c,t)}{x_c^2})\end{aligned} \tag{8}$$

This control is computed at each sample period $k$ based the information of the state of the vehicle and the curve at sample $k$. The parameter $x_c$, which is the look ahead distance at which the connecting contour joins the desired curve, can be tweaked for performance. For example, a small look ahead distance implies a large yaw rate command and faster convergence. But this may not be possible because the yaw rates are bounded (maximum absolute value for the airplane used was 0.22 rad/sec). Large look ahead distances may not be possible because the camera on the airplane may be mounted such that only a specified forward distance of the desired path could be seen. A larger look ahead distance also implies that the vehicle is going to take a longer time to reach the curve. In this work, a reasonable $x_c$ was chosen based on several hardware in the loop simulations.

## VI. Experimental Results

The road detection algorithm and the control algorithm were tested closed loop in both hard ware in the loop and UAV flight results. The following subsections presents them.

### A. Hardware in the loop

A hardware-in-the-loop (HIL) simulation was developed in order to perform initial closed loop tests of the vision and the control algorithms. The aircraft dynamics are replaced by a high-dimensional nonlinear simulation provided with the Piccolo avionics package [31]. The camera outputs were simulated using the Vega software package [32]. This software [32] provides real time visualization of three dimensional models of the testing environment. A curved road, as shown in figure 13, was created in a virtual city using the Vega software. The radius of the curve R (figure 13) was chosen to be 200 meters to ensure that it is greater than the minimum turning radius of the UAV. Camera was mounted at an angle of $\theta = 30$ degrees with respect to the yaw axis of the aircraft (figure 14). Sample images of the final results of vision based control algorithm are shown in figure 15. Video of the closed loop results can be downloaded or seen at http://path.berkeley.edu/∼siva/videos. The cross track deviation error from the centerline of the canal is shown in figures 16. The UAV was able to track a curved road with an average error of 13.74 meters over 3200 meters of the road.

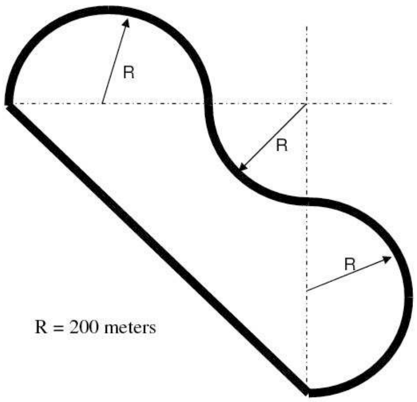

Fig. 13. Shape of the road created using the Vega visualization software.

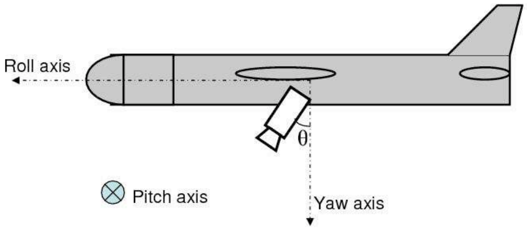

Fig. 14. Camera is oriented at an specified angle $\theta$ with respect to the yaw axis. Hence, if the aircraft is flying parallel to the ground plane, then $\theta = 0$ is the case where the camera is looking straight down.

### B. Flight results

The vision-based control system was tested using a Sig Rascal model aircraft (Figure 17). Low level aircraft control and stabilization is performed by a commercial Cloud Cap Piccolo avionics package (Figure 18) [31]. The vision and the control algorithms run on the onboard PC104 that communicates directly with the Piccolo avionics box through a serial port. The flight tests were conducted at Crows landing Naval

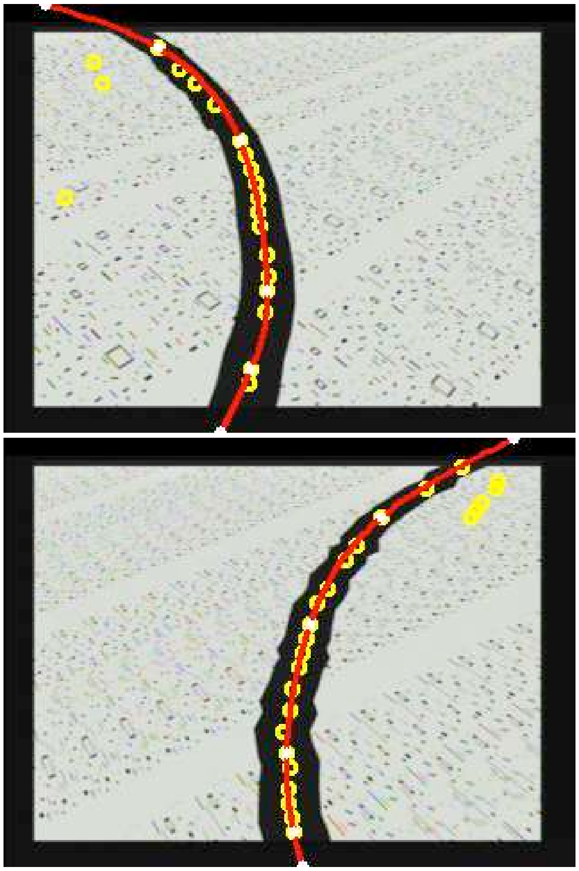

Fig. 15. Sample images from the processed video for the HIL simulation.

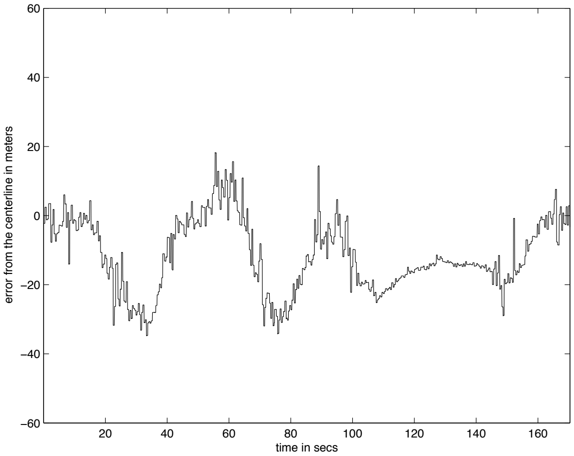

Fig. 16. Cross track error from the centerline of the road for a look ahead distance of 250 meters and camera at $\theta = 30$ degrees with respect to the yaw axis of the plane.

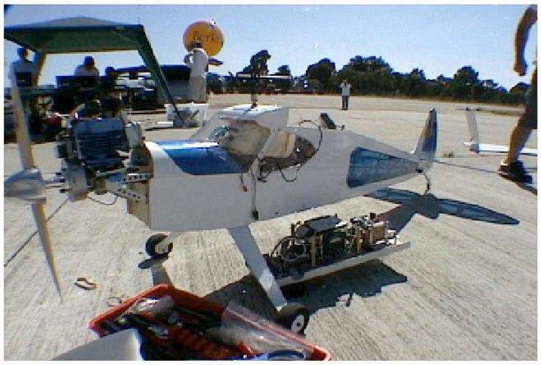

Fig. 17. Sig Rascal Model Aircraft used for flight tests.

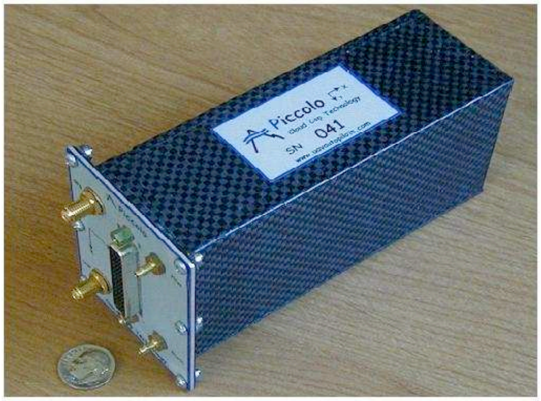

Fig. 18. Piccolo avionics package that performs low level flight control and stabilization.

Auxillary Landing Field, Patterson, California. A picture of the canal over which the UAV was flown is shown in figure 19. The length of the canal as shown in the picture is around 700 meters. Test video of the canal was collected first by flying the vehicle by waypoint navigation. The airplane was held at a constant altitude of 100 meters. This height was chosen in order to have a good resolution of the image of the canal. The recorded onboard video was used as an input to the learning phase of the road detection algorithm.

The final results[6] of vision based control algorithm for the UAV are shown in the figure 20. Two sets of experiments were conducted:

1) The first with a look ahead distance of 175 meters and the camera tilted at $\theta$ =10 degrees with respect to the yaw axis of the plane and,
2) The second with a look ahead distance of 150 meters and the camera at $\theta$ =15 degrees with respect to the yaw axis of the plane.

The road detection algorithm runs at 5 Hz (takes $\leq$ 200 ms) or faster on the onboard PC104 (700 MHz, Intel Pentium III). The cross track deviation error from the centerline of the

[6] Processed video of all the experimental results can be downloaded or seen at http://path.berkeley.edu/∼siva/videos

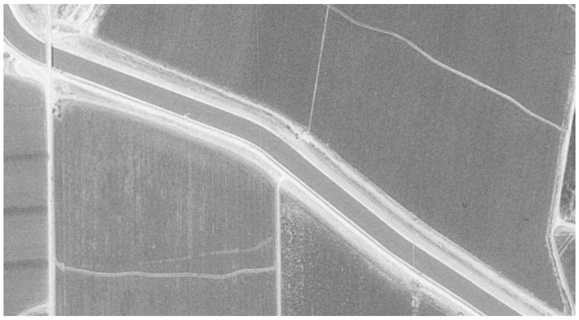

Fig. 19. Canal at the Crowslanding facility.

canal is shown in figures 21 and 22. On average, the cross track deviation error was around 10 meters over a stretch of 700 meters of the canal.

## VII. Conclusions and Future work

This paper addresses the problem of monitoring structures such as pipelines, roads, bridges etc using an autonomous unmanned aerial vehicle (UAV) based on visual feedback. A realtime detection algorithm for various structures such as highways, roads and canals has been introduced. A fast learning algorithm with minimal supervision is also introduced. The algorithm has been tested with video of highways and canals. Experimental results indicate a reasonable performance of the closed loop vision based control with a average error of around 10 meters over the stretch of 700 meters of the canal. Some of the future directions of the current work include the following:

- Filtering the incoming vision data over a batch of frames and updating the equation of the curve.
- Processing the collected video by following a structure and inferring useful information from it.
- The airplane was modelled as a simple unicycle model without taking into account the dynamics of the vehicle.
- Wind disturbances seems to be common while flying the vehicle and hence they need to be included in modelling.

## References

[1] Coifman, B., McCord, M., Mishalani, M., Redmill, K., ”Surface Transportation Surveillance from Unmanned Aerial Vehicles” Proc. of the 83rd Annual Meeting of the Transportation Research Board, 2004.

[2] S. Srinivasan et. al, ”Airborne Traffic Surveillance Systems - Video Surveillance of Highway Traffic” , ACM VSSN 2004 conference.

[3] http://avdil.gtri.gatech.edu/RCM/RCM/DroneProject.html

[4] http://www.alyeska-pipe.com/

[5] Douglas J. Nyman, Elden R.Johnson and Christopher H.Roach, Trans-Alaska pipeline emergency response and recovery following the november 3, 2002 denali fault earthquake. *Sixth U.S. Conference and Workshop on Lifeline Earthquake Engineering, ASCE Technical Council on Lifeline Earthquake Engineering*, Long Beach, CA, August 2003.

[6] William J. Hall, Douglas J. Nyman, Elden R.Johnson and J. David Norton, Performance of the Trans-Alaska pipeline in the november 3, 2002 denali fault earthquake. *Sixth U.S. Conference and Workshop on Lifeline Earthquake Engineering, ASCE Technical Council on Lifeline Earthquake Engineering*, Long Beach, CA, August 2003.

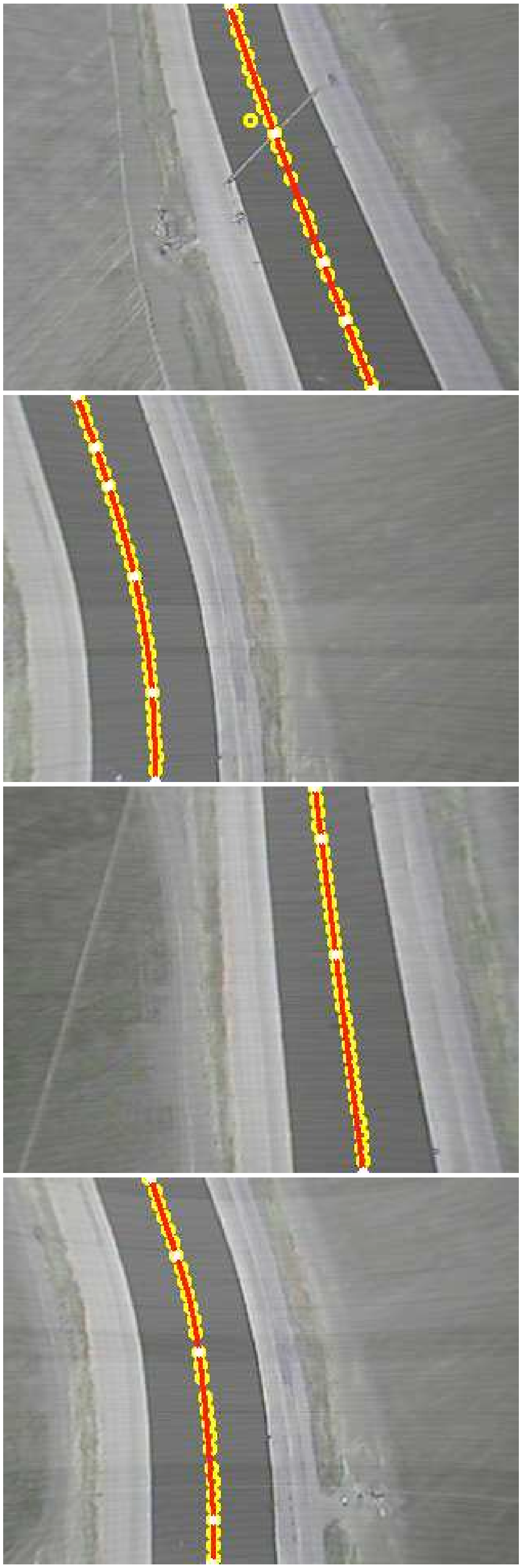

Fig. 20. Sample images from the onboard processed video.

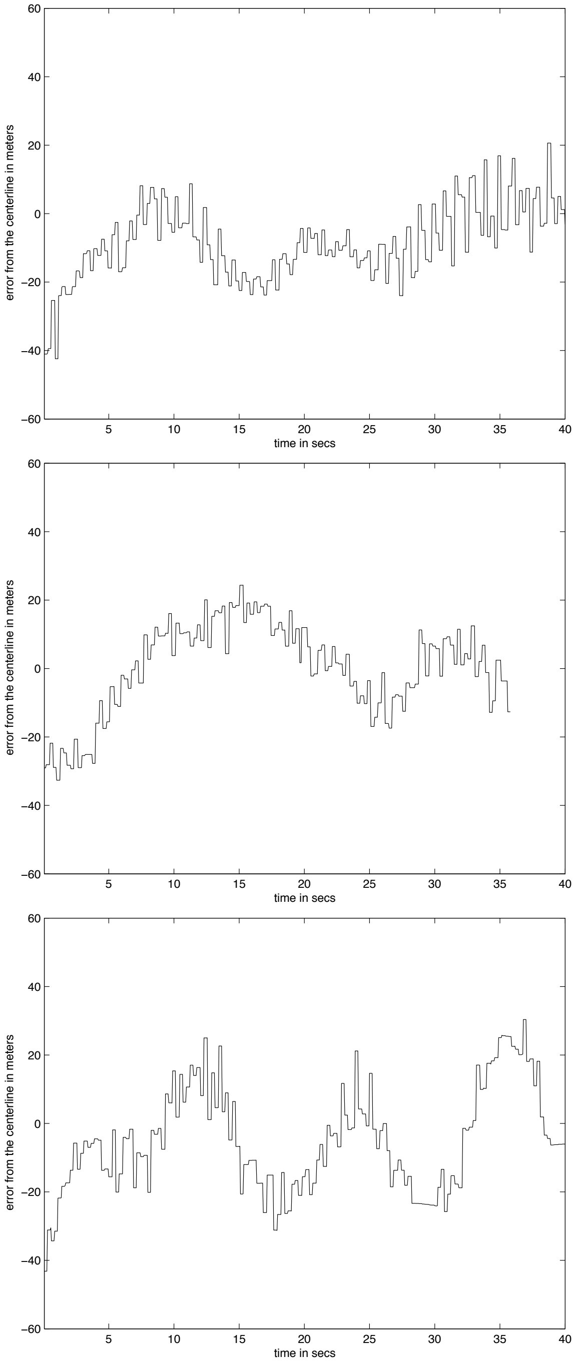

Fig. 21. Cross track error from the centerline of the road for a look ahead distance of 175 meters and camera at $\theta = 10$ degrees with respect to the yaw axis of the plane.

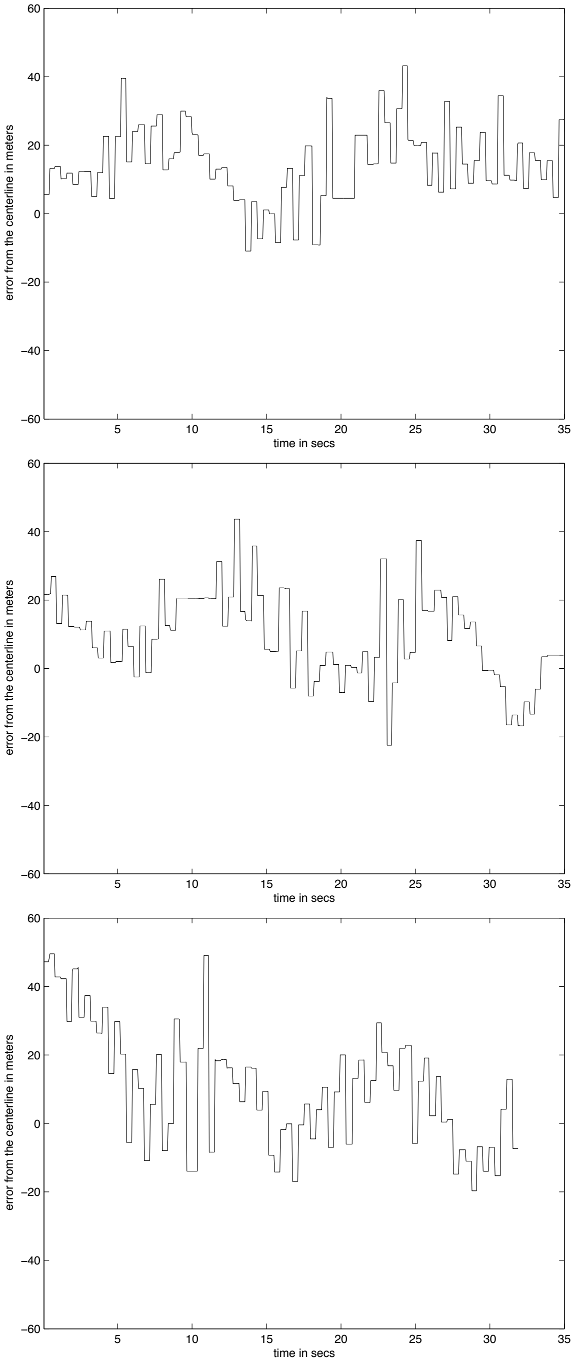

Fig. 22. Cross track error from the centerline of the road for a look ahead distance of 150 meters and camera at $\theta = 15$ degrees with respect to the yaw axis of the plane.

[7] Yue Wang, Eam Khwang Teoh, Dinggang Shen, Lane detection and tracking using B-Snakes, Image and Vision Computing 22 (2004) 269-280.

[8] A. Broggi, Robust real-time lane and road detection in critical shadow conditions, Proceedings IEEE International Symposium on Computer Vision, Coral Gables, Florida, November 19-21 (1995).

[9] M. Bertozzi, A. Broggi, GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection, IEEE Transactions of Image Processing (1998) 62-81.

[10] A. Broggi, A massively parallel approach to real-time vision-based road markings detection, in: I. Masaky (Ed.), Proceeding IEEE Intelligent Vehicles '95, 1995, pp. 84-89.

[11] H. Andrew, S. Lai, H. Nelson, C. Yung, Lane detection by orientation and length discrimination, IEEE Transactions On Systems, Man and Cybernetics, Part B 30 (4) (2000) 539-548.

[12] D. Jung Kang, J. Won Choi, I.S. Kweon, Finding and tracking road lanes using line-snakes, Proceedings of Conference on Intelligent Vehicle, 1996, pp. 189-194, Japan.

[13] A. Kaske, D. Wolf, R. Husson, Lane boundary detection using statistical criteria, International Conference on Quality by Artificial Vision, QCAV9 (1997) 28-30. Le Creusot, France.

[14] A. Kaske, R. Husson, D. Wolf, Chi-square fitting of deformable templates for lane boundary detection, IAR Annual Meeting '95, November 1995 (1995) Grenoble France.

[15] Todd Jochem and Shumeet Baluja, Massively Parallel, Adaptive, Color Image Processing for Autonomous Road Following, Research Report CMU-RI-TR-93-10, The Robotics Institute and Department of Computer Science Carnegie Mellon University, 1993.

[16] J. Crisman and C. Thorpe. UNSCARF, a color vision system for the detection of unstructured roads. In Proc. Int. Conf. Robotics and Automation, pp. 2496-2501, 1991.

[17] Chrisman and Thorpe, SCARF: A color vision system that tracks roads and intersections,IEEE Transactions on Robotics and Automation, Vol. 9, No. 1, February 1993.

[18] M. Turk, D. Morgenthalar, K. Gremban and M. Marra. VITS-A Vision System for Autonomous Land Vehicle Navigation, IEEE Transactions on Pattern Analysis and Machine Intelligence, May 1988.

[19] A. Gruen, E. Baltsavias, and O. Henricsson, editors. *Extraction of Man-Made Objects from Aerial and Space Images*, II. Birkhauser Verlag, 1997. Proceedings of meeting held in Ascona, Switzeland.

[20] U. Ozguner, K. A. Unyelioglu, and C. Hatipoglu. An analytical study of vehicle steering control. In Proc. 4th IEEE Conference on Control Applications, pages 125-130, 1995.

[21] Dickmanns, Ernst D., and Mysliwetz, Birger D. 1992. Recursive 3-D road and relative ego-state recognition. IEEE Trans. Patt. Analysis Machine Intell.14(2):199213.

[22] C. J. Taylor, J. Kosecka, R. Blasi, and J. Malik. A comparative study of vision-based lateral control strategies for autonomous highway driving. Int. J. Robotics Research, 18(5), May 1999.

[23] S. Saripalli, J.F. Montgomery and G.S. Sukhatme, "Vision-based autonomous landing of an unmanned aerial vehicle," in Proceedings of the 2002 International Conference on Robotics and Automation, 2002.

[24] O. Shakernia, R. Vidal, C.S. Sharp, Y. Ma and S. Sastry, "Multiple View Motion Estimation and Control for Landing an Unmanned Aerial Vehicle," in Proceedings of the 2002 International Conference on Robotics and Automation, 2002.

[25] S. Furst and E.D. Dickmanns, "A vision based navigation system for autonomous aircraft," in Intelligent Autonomous Systems. IAS-5, pp. 765-774, 1998.

[26] B. Sinopoli, M. Micheli, G. Donato and T.J. Koo, "Vision Based Navigation for an Unmanned Air Vehicle," in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA2001), pp. 1757-1765, 2001.

[27] E. Frew, T. McGee, Z. Kim, X. Xiao, S. Jackson, M. Morimoto, S. Rathinam, J. Padial, and R. Sengupta. Vision-based road following using a small automonous aircraft. *In Proc. IEEE Aerospace Conference*, 2004.

[28] R. Frezza, G. Picci, and S. Soatto. "A Lagrangian formulation of nonholonomic path following". *The Confluence of Vision and Control, A. S. Morse et al. (eds.)*, Springer Verlag, 1998.

[29] Yi Ma, Jana Kosecka and Shankar S. Sastry, Vision Guided Navigation for A Nonholonomic Mobile Robot. *IEEE Transactions on Robotics and Automation*, vol. 15, no. 3, page 521-37, June 1999.

[30] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24:381-395, 1981.

[31] Cloud Cap Technology, Inc, "Piccolo User's Guide".

[32] http://www.multigen.com/