Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Vision Based Following of Locally Linear Structures using an Unmanned Aerial Vehicle

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

Abstract— **Inspecting and monitoring oil-gas pipelines, roads, bridges, power generation grids is very important in ensuring the reliability and life expectancy of these civil systems. An autonomous UAV can decrease the operational costs, expedite the monitoring process and be used in situations where manned inspection is not possible. This paper addresses the problem of monitoring these systems using an autonomous unmanned aerial vehicle (UAV) which follows the locally linear structures using visual feedback.**

I. INTRODUCTION

This paper addresses the problem of using an unmanned aerial vehicle to follow locally linear structures like roads, highways and canals based on visual feedback. The problem of following a structure using a camera on the vehicle has two main subparts to it:

1) Vision based structure detection
2) Controlling the vehicle to follow the structure.

One of the key challenges is to have a real-time detection algorithm of the structure. The fixed wing aerial vehicles used in this work, travel approximately at 20 m/s and hence faster detection algorithms are required to have minimal tracking errors of the structure. Structures could be pipelines, roads, canals, power grids etc. Clearly, one expects the detection algorithms to also vary with the application. This paper presents a structure detection algorithm that is fairly general and has been tested for highways, roads and canal videos. The structure as seen in the image is assumed to have a strong vertical correlation. This is a reasonable assumption provided the vehicle is close to the structure and the camera mounted on the vehicle is roughly oriented along the structure. It is also assumed that the width of the structure in the image varies slowly. The boundaries, lanes, and other lines along the structure as seen in the image are assumed to be parallel. Hence, these structures are called locally linear as, even though they may be curved over a long stretch, they are linear as seen in one image.

Road detection has been an important research topic in aerial image analysis [3]. However, aerial image analysis applications do not usually require realtime computation. Most of the proposed algorithms are non-realtime, and are focused on rural roads with low-resolution. Previous work on realtime road detection used Bayesian lane marking classification algorithms[2]. It uses Bayesian RGB pixel classifiers to detect lane markings, and localizes the road center by fitting a line to the detected lane markings. Although this algorithm may be applied to some other types of roads by re-learning the road or lane colors, it is still not adequate for highways or other local streets unless the UAV flies low enough so that the lane markings are visible. This paper presents a novel detection approach that can detect a variety of structures (currently tested with roads and canals) in realtime with a single hint from a user. All it requires is a single image of the target structure without any specification (such as the position or the width of the road). Initial results on straight highways are shown in [6].

Transforming the image coordinates of the structure to the inertial coordinates (or relative to the airplane) is relatively easy since the yaw, pitch and the roll of the camera relative to the ground is known. Following a given curve by a non holonomic vehicle has been addressed in the literature. Autonomous tracking of curves using vision has been studied in [10] [11] for ground vehicles. The control strategy adopted in this paper is the one proposed in [10]. The following are the main results presented in the paper:

1) A detection algorithm that can identify and localize structures like highways, roads and canals. 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) The vehicle with the camera onboard was able to track a 700 meter canal based on vision with a cross track error of around 10 meters.

II. MOTIVATION

The visual feedback compensates GPS inaccuracies and tracks the structure even it is shifted from the assumed location. The following application presents a motivation as to why visual feedback would be useful even if the exact GPS location of the structure is known. The Trans-Atlantic pipeline system [7] transports oil from the north slope of Alaska to Valdez-the most northern most ice free port in Alaska. The pipeline system is 800 miles long and supplies approximately 17 percent of the crude oil produced in the United States. More than half of this length, nearly 420 miles, is above ground to protect the delicate permafrost. Majority of this pipeline system lies within a 7.00 Richter zone with the zone near the Valdez port (maximum earthquake

*Corresponding author: rsiva@berkeley.edu

†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 got by flying a UAV over a canal at Crowslanding Naval Auxillary Landing Field, Patterson, CA

0-7803-9568-9/05/$20.00 ©2005 IEEE

prone zone) at 8.5 Richter magnitude [8] [9]. 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 [8]. 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 focussed inspection effort by Alyeska [7]. There was no oil spilled and the entire pipeline operation was resumed after 66 hours [8]. This system which is on this massive scale requires sustained monitoring efforts to increase its life expectancy. Currently, Alyeska conducts routine aerial surveillance atleast 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 and GPS information would be very useful for this application.

This method of using an autonomous system to follow a structure[3] (in the above example, it was a pipeline) can be used in other applications. Earthquakes could damage highways, bridges also. It might be difficult to obtain instant feedback of the road conditions after such disasters and hence an autonomous system that can follow roads and send back visual data will be helpful. Traffic monitoring can be done with low operational cost and in conditions that are not suitable for manned flying operations. This system can be also used to find failures or inspect electric power grids that run through the entire country. Monitoring many of these urban infrastructures such as bridges, roads, railways and power transmission corridors can be done without much attention and, hence, could be useful for homeland security also.

The underlying problem in all these applications is that of requiring an unmanned aerial vehicle to follow structures based on visual feedback. This is the problem that is addressed in the paper. Section III discusses the assumptions, the model of the airplane and other related issues. Section IV presents the structure detection algorithm. Section V discusses the controller that helps the vehicle track the structure. Results and the conclusions follow in the remaining sections.

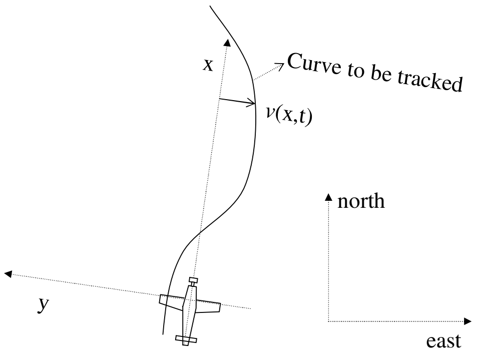

Fig. 1. Problem setup

III. Problem Setup

In this paper, the fixed wing airplane is modelled as a unicycle model travelling at a linear speed v with a bounded angular rate ω. 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:

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

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 1 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 for controlling the vehicle. The detection algorithm explained in the next section identifies the structure and outputs the image coordinates of the points on 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 ω 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.

IV. Realtime Detection of the structure

The basic approach to find the desired structure in a given image consists of the following steps:

1) *Learning the structure from a single example:*
   a) *Image rectification*

[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.

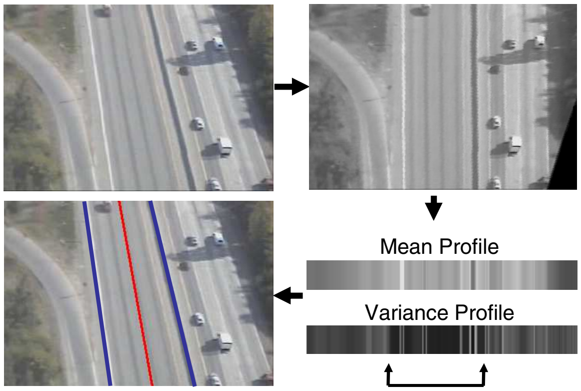

Fig. 2. The suggested learning approach. First, the image is rectified. The statistics of the rectified image (the means and the variances) suggests a rough position of the road.

b) *Structure identification and localization*

2) *One dimensional signal matching*

3) *Curve fitting*

A. Learning Structure from a Single Example

The desired structure is learnt first using a sample image (see figure 2). The learned information is then used to detect and localize structures from new images.

Image Rectification

Given a sample image, the algorithm first detects the orientation and the perspective angle of the structure and rectifies the image. The boundaries of the linear structure must be parallel to each other. All parallel lines meet at a single vanishing point,(x_0, y_0) , in the image coordinates, and any, two points, (x, y) and (x', y') on a line (figure 3) satisfy,

$$x' = \frac{y_0 - y'}{y_0 - y}x + \frac{y' - y}{y_0 - y}x_0 = ax + b \qquad (2)$$

where a and b are constants given y, y', and the vanishing point. Therefore, intensity profiles of horizontal cuts of a image can be modelled by a linear relationship:

$$I(x) = I'(ax + b) + \eta_I, \qquad (3)$$

where $I(.)$ indicates the intensity profile of the sample image at a given y, $I'(.)$ indicates the corresponding intensity profile of the rectified image, and η_I is a noise term. The idea is to find the vanishing point first, and then use the above equation to rectify each horizontal intensity profile of the sample image.

The vanishing point is found by applying a RANSAC-style (RANdom SAmple Concensus) algorithm. First, long line segments are extracted by applying a vertical edge detection algorithm followed by line grouping and fitting. Any two line segments suggest a vanishing point. Vanishing point hypotheses are generated from all possible line pairs, and the one which has the most support by the other lines is chosen. Finally, the image is rectified based on this vanishing point hypothesis. For more details on this procedure, see [6].

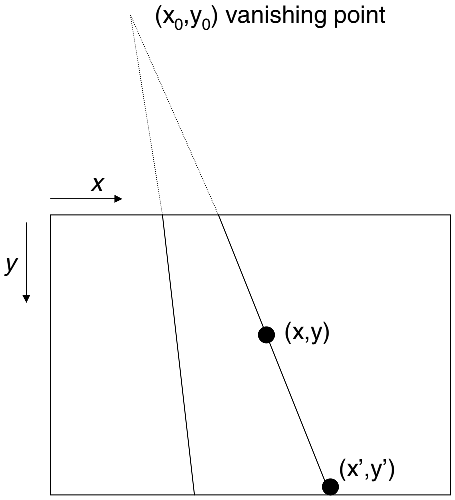

Fig. 3. Vanishing point and the line geometry.

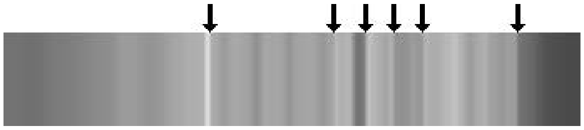

Fig. 4. Structure boundary candidates based on the mean profile.

Structure Identification and Localization

The boundaries of the structure are identified and this structure is represented as a one-dimensional intensity profile which corresponds to the cross sections of a structure. Once the image is rectified 2, statistics of the structure is gathered. That is, for each vertical line of the rectified image, the mean and the variance are calculated. The horizontal profiles of the mean and the variance (also shown in 2) indicate a rough position of the structure.

Since the background has less vertical correlation than the structure, the mean profile of the background is smoother (has smaller horizontal variation) than that of the structure. Unless the background is texture-less, the variance of the background is usually higher than that of the structure. The localization algorithm is based on such observation. A similar approach is used for digit recognition [4] also.

Structure boundary candidates are chosen based on their mean profile as they have large intensity changes in the mean profile. Detected boundary candidates for the highway is shown in figure 4. For each and every boundary candidate pair (which are atleast 50 pixels apart), the variance profile (between the two boundary candidates) is examined. The pair, of which the largest portion has low variances, is finally chosen as the left and the right structure boundaries.

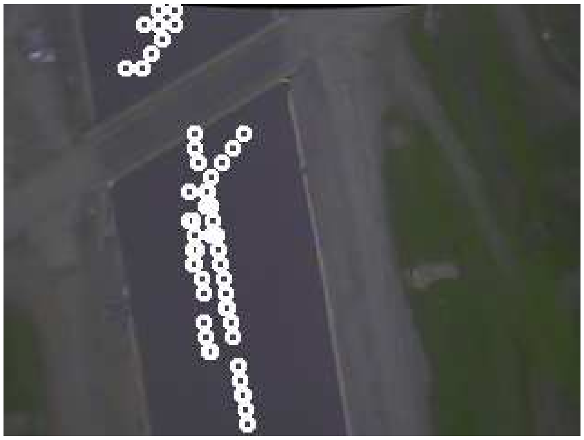

Fig. 5. Center hypotheses of an example image.

B. One-dimensional Signal Matching

The detection is performed by matching the one dimensional intensity profile (from the learned image) to the horizontal scan lines of a target image. The matching process needs to allow varying scales because the widths of cross-sections can vary due to a perspective view. The structure, obtained from the proposed learning procedure, is a 1-D intensity (or color) signal segment. A series of 1-D signal matching processes is applied to detect and localize it in the new image. Given a target image, the goal is to find a and b in equation 3 which minimize η_I.

The difficulty of estimating a and b is in that the two-dimensional search space is too large to meet the realtime constraint. Therefore, a feature matching technique is applied assuming that the scales (the parameter a) are not changing too much. First, two distinct features (where the changes in the intensity profile is high) for a given scan line in the target image is detected. A template-matching-based search is performed to find match candidates for each of these features in the learnt intensity profile. Three best hypotheses of a and b are chosen per scan line for curve fitting.

C. Curve Fitting

Once the position b and the scale a of the structure is found for each horizontal scan line, a curve fitting algorithm is applied to determine the equation of the structure being followed on the image plane. Given a match hypotheses, a curve is estimated for the center of the structure. This is the middle line between the two boundaries as indicated by all the horizontal scan lines. Center hypotheses of an example image are shown in figure 5. To reduce computation, matches are found for every four scan lines. As shown in the figure, it still gives enough matches for curve fitting. The challenges of curve fitting include:

- finding a good curve representation, and
- finding a robust curve fit in a short amount of time.

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:

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

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 $a_i, \ldots, h_i$ are uniquely determined by the control points requiring that the curve is smooth. A cubic-spline curve has a useful property that the control points are actually on the curve. This property can be used to apply a RANSAC algorithm [1] for curve fitting. A RANSAC algorithm is a robust fitting algorithm that has been successfully used in many computer vision problems.

The curve fitting algorithm randomly chooses a large number of (500 in our implementation) set of two (for a line), three (for a circle), or four (for a complex curve) structure center hypotheses[4]. For each set of hypotheses (of 4∼6 control points), a cubic-spline curve is uniquely determined. For robustness, it is required that the $y_i(t)$ coordinates on the image be monotonic for all i^s. 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. A curve hypothesis is evaluated based on the following equation:

$$\text{CurveScore} = (1 - \lambda) \sum_m \text{CorrelationScore}(m), \tag{5}$$

where m is a supporting scan line match, and λ is a penalty for the minimum description length (MDL) criteria. The output of this algorithm is basically in terms of the image coordinates (control points) of the structure that resulted in the maximum CurveScore. The processed images of the road detection algorithm for video (courtesy MLB company's bat video) are shown in figure 6. These image coordinates can be converted to the ground coordinates using the roll, pitch angles and the height measurements from the sensors onboard of the plane. These ground coordinates are the input to the control discussed in the next section.

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 $\frac{\nu(x,t)}{dt}_{x=0} = 0$), then the yaw rate required to steer the vehicle along the curve 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

[4] Since the first and the last points are usually not the bottom- and the top-most points, the algorithm extrapolates the first and the last points such that the curvature is maintained

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

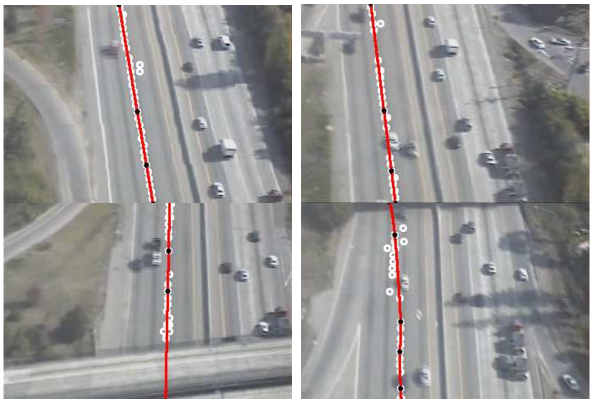

Fig. 6. Detection algorithm working with the video courtesy of MLB company (www.spyplanes.com).

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 [10] 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 7 illustrates a connecting contour. This connecting contour ν_c essentially must satisfy atleast the following set of minimal conditions:

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

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:

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

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,

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

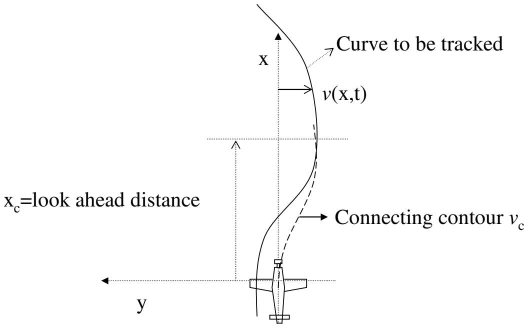

Fig. 7. Illustration of the curve to be tracked and the connecting contour.

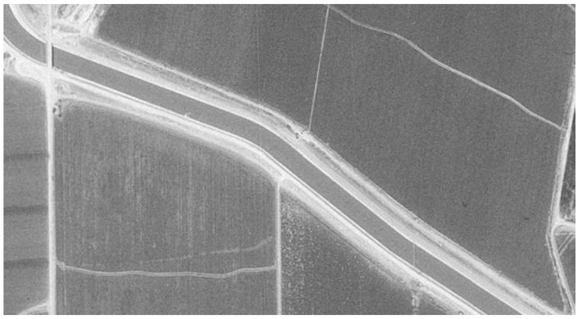

Fig. 8. Canal at the Crowslanding facility.

This control is computed at each sample period k based the information of the state of the vehicle and the curve at sample k.

VI. Experimental Results

The road detection algorithm and the control algorithm were tested separately with the UAV following a canal at Crowslanding Naval Auxillary Landing Field, Patterson, California. A picture of the canal over which the UAV was flown is shown in figure 8. The length of the canal as shown in the picture is around 700 meters. Test video of the canal was collected 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 a input to the learning phase of the road detection algorithm. The learnt structure is checked for correctness by visual inspection.

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

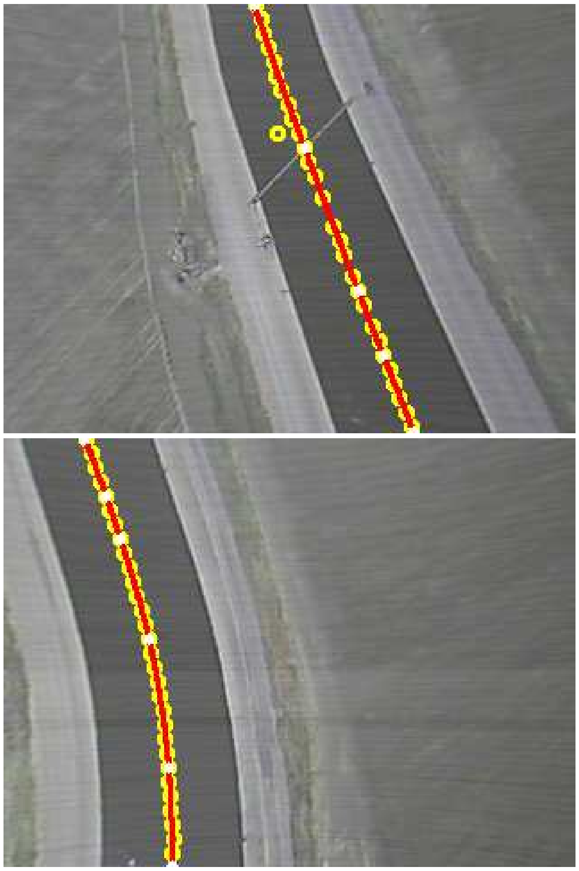

Fig. 9. Sample images from the onboard processed video.

the curve. A reasonable look ahead distance (175 meters) was chosen based on several hardware in the loop simulations before it was tested in the closed loop experiment.

The final results of detection algorithm when this vehicle was flying closed loop control are shown in the figure 9. The road detection algorithm runs at 5 Hz (takes $\leq$ 200 ms) or faster on the PC104 (700 MHz, Intel Pentium III). For the curve fitting part, $\lambda = 0.05$ was chosen for the hypotheses with 5 control points (circles) and $\lambda = 0.1$ for the hypotheses with 6 control points (complex curves). The cross track deviation error from the centerline of the canal is shown in figures 10. On an average the deviation 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, power generation grids etc using an autonomous unmanned aerial vehicle (UAV) based on visual feedback. A real time detection algorithm of locally linear structures such as highways, roads and canals has been presented. 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:

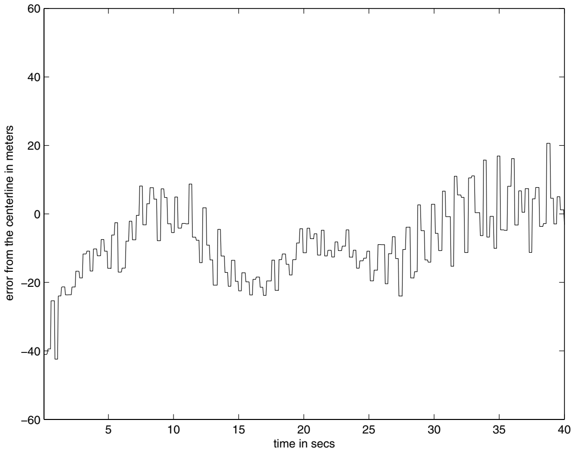

Fig. 10. Cross track error from the centerline of the road for a look ahead distance of 175 meters and camera at 10 with respect to the roll axis of the plane.

- Filtering the incoming vision data over a batch of frames and updating the equation of the curve.
- 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] 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.

[2] 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.

[3] 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.

[4] E. G. Miller. Learning from One Example in Machine Vision by Sharing Probability Densities. PhD thesis, Massachusetts Institute of Technology, 2002.

[5] K. Price. Urban street grid description and verification. In *Proc. IEEE Workshop on Application of Computer Vision*, pages 148-154, 2000.

[6] Z. Kim. Realtime Road Detection by Learning from One Example. *In Proc. IEEE Workshop on Application of Computer Vision*, pages 455-460, 2005.

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

[8] 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.

[9] 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.

[10] 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.

[11] 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.