

Using Roads for Autonomous Air Vehicle Guidance

Ralph Hartley, Behrooz Kamgar-Parsi¹, *Senior Member, IEEE*, and Cody Narber

Abstract—This is a study on the development of a road following vision-based guidance system for unmanned air vehicles (UAV) in real world applications. Currently, autonomous navigation requires the use of GPS. In many applications, however, dependence on GPS is undesirable. GPS signals are weak and can be jammed easily. Also, GPS waypoints may not be up-to-date. In recent years, vision-based navigation has been gaining popularity. Vision-based guidance requires existence of visible paths or extended landmarks for air vehicles to detect and follow. Roads are the most commonly available path to follow. Moreover, the abundance of events that happen along roads make them appealing subjects of surveillance. Many road detection (single images) and road tracking (videos) algorithms have been proposed in the literature. Fast detection and tracking has been the emphasis of those intended for UAVs. Due to the complexity of road detection, we not only need advanced software but also the most effective sensors. In this paper, we propose a road following algorithm that uses both RGB camera and hyperspectral sensor and report the results of actual test flights conducted in different locations and different seasons.

Index Terms—Road detection, road following, autonomous air vehicles, hyperspectral sensor, unmanned air vehicle (UAV).

I. INTRODUCTION

AIR vehicle miniaturization and autonomy has long been a goal in certain applications [1]. To achieve guidance autonomy, vision based navigation has been steadily gaining popularity. A vision based guidance requires the existence of a visible path or extended landmark. Reference [2] suggests using shorelines for autonomous UAV guidance; [3] proposes a method to search and track rivers; [4] proposes an approach to follow “boundary” between distinct regions such as coastlines or edge of vegetation. However, most UAV vision-based systems have experimented with road following. This is not surprising because both in terms of availability and the wealth of events happening along them, roads constitute the most significant paths for UAVs to follow.

Monitoring activities along roads is of interest both to military and civilian institutions. Air vehicles are ideal platforms for collecting information for this purpose. Although the vehicle can be guided over the road by a human operator, increasingly it is becoming preferable to have autonomous road following air vehicles. Currently, UAVs that navigate

under the preferred autonomous mode must use GPS updates. Thus, they are vulnerable to GPS jamming, which is occurring with increasing frequency, especially in unfriendly environments. Furthermore, navigating by GPS requires GPS way points which may not be available, or may not be up-to-date.

“Road” literature is quite extensive as it concerns a variety of different problems. Some of the factors contributing to this variety are the diversity of road types and their surroundings, the viewing platforms, the intent of the study, etc. Some roads are well structured with clear lane markings and boundaries, while others are unstructured such as rural and unpaved roads. The contrast between a road and its background is different in different domains and seasons requiring various degrees of effort to delineate the road. The viewing platform may be a ground vehicle [5], [6] or an air vehicle [7], [8]. Not only the latter may fly at different altitudes, but it may also view the ground from different angles including straight down [4].

Studies that involve aerial road imagery address road detection, road tracking or road following. Generally speaking, road detection implies detection in a single image [6], [9], road tracking implies road detection/tracking in a video [7], road following implies developing control strategies to follow roads and possibly actual autonomous flight over a road [10]. Of course, road tracking starts with road detection, and road following requires road tracking. In road tracking studies, experiments are performed on public or private videos that are collected by human guided (road following) UAVs. Autonomous road following studies are few and are mostly concerned with control issues so that the UAV could actually follow the (detected) road in a stable manner. Typically, in such studies, the road (or the extended landmark, e.g. shoreline [2]) that they follow are not difficult to detect. However, in real world applications road detection/tracking, especially in rural areas, can be quite challenging.

As was said before, road following requires road tracking. Nevertheless, road tracking in videos and road tracking in actual flights are not the same. In the latter, images are taken while plane is guiding itself. Therefore, inaccuracies in guidance can affect tracking and vice versa. Indeed, errors in tracking and guidance can add up. A gust of wind, for example, can complicate autonomous navigation and impact captured views affecting road tracking if not road detection. Our objective is to develop a vision-based guidance system so that the air vehicle will be able to follow typical rural roads (and under a variety of weather conditions).

Here, we will briefly discuss a number of road tracking studies in the literature, followed by studies on road following.

Road Tracking: [11] initially learns a cross section of the linear structure to be followed, e.g. road, from a sample image.

Manuscript received November 19, 2016; revised July 25, 2017 and December 11, 2017; accepted December 21, 2017. Date of publication May 22, 2018; date of current version November 27, 2018. This was supported in part by the Naval Research Laboratory and in part by the Office of Naval Research under Grant N00014-15-WX-01729. The Associate Editor for this paper was Z. Mao. (*Corresponding author: Behrooz Kamgar-Parsi.*)

The authors are with the Naval Research Laboratory, Information Technology Division, Washington, DC 20375 USA (e-mail: ralph.hartley@nrl.navy.mil; behrooz.kamgarparisi@nrl.navy.mil; cody.narber@nrl.navy.mil).

Digital Object Identifier 10.1109/TITS.2018.2799485

The 1-D cross section, called structure, is somewhat wider than the road (see also [8]). Vanishing point is obtained by applying a variation of RANSAC algorithm to line segments extracted from edges. It then rectifies the image to detect (and track) the road by matching the structure to new frame. [7] suggests a road detection and tracking for aerial scenes by using a graph-cut approach based on modeling image color distribution with Gaussian mixture for detection, and a homography-alignment-based approach for road tracking in video images. Comprehensive descriptions of graph-cut and homography may be found in [12] and [13], respectively.

Our approach is to model the road with a family of piecewise splines, where the middle (piecewise) spline represents the middle of the road and side (piecewise) splines sample the road's RGB values. This road model will be used for tracking. Because our model is compact, it can be transmitted to the ground over the same radio link used by the autopilot for commands and telemetry. This simplifies interactions with users which will be needed both during testing and actual missions.

Road Following: Most studies on following roads (or curvilinear structures) either suggest flying on a path tangent to the road such as [2] and [14], or flying toward a target point on the road some distance ahead such as [10]. Reference [15] on the other hand suggests an approach based on proportional navigation where the target being chased/tracked is the edge of the camera frame at which the road flows into the image.

Our objective is not to always fly directly over the road, rather to keep the road well inside the field of view. If the plane tried to fly directly over the road with too much crosswind, the road would be outside of the camera's field of view. In our approach, a target point for steering is chosen on the (modeled) road some distance ahead of the plane. In addition, simple proportional control is used to point the plane towards the target point, sending roll angle commands to the autopilot, proportional to the angle between the plane's heading and the target.

A. Choice of Sensors

A rural road may pass through different types of terrain. For a UAV to be able to follow long stretches of a road, detection has to remain successful throughout the flight. Furthermore, different seasons and weather conditions can also affect detection, in part because of changes in the background from which the road must be delineated. The complexity of detection task not only requires robust software, but also the use of the most effective sensors.

In fact, detection of other extended landmarks can also become more reliable through the use of suitable sensors. For example, the remote sensing community has long been using near IR (NIR) sensors to discriminate water from land. NIR detects water regardless of its color or texture [16], [17]. Note that water may appear in a variety of colors from blue to brown depending on turbidity, time of day, etc, in particular in rivers, and its texture is affected by wind. To distinguish the boundary between water and land in RGB images, [4] uses a prior supervised learning process based on hue. Whereas using a

NIR sensor, the vision-based system developed in [2] easily detects (and follows) shorelines without a need for learning.

A color camera alone may not be the best sensor option for rural roads. On the other hand, the use of active sensors is not desirable as they are not energy efficient; they can also be detected in hostile environments. In this study, we use both an RGB camera and a hyperspectral sensor (HSS). Not only they are both passive sensors, but they can also complement each other well.

A hyperspectral sensor (HSS) is useful for recognizing the (surface) material from which an object is made. While a color camera has only three channels (red, green, blue), a hyperspectral sensor typically has several hundred channels. It usually covers a wider range of electromagnetic wavelength than visible light, but more significantly it can resolve the electromagnetic spectrum into hundreds of bands, thus discriminating materials through their spectral "signature".

The HSS can detect the road based on its material. However, an HSS image is only 1-dimensional. Furthermore, the rate at which HSS images are acquired is comparable to those of the RGB camera. At the speed with which a typical fixed wing UAV flies, there will be large spatial gaps between consecutive 1-D HSS images. Nevertheless, the RGB camera produces 2-dimensional views of the scene, while the HSS provides us with cross sections of the road. By calibrating the two sensors we will know the location of the road cross sections detected by the HSS in the RGB collected color images as well as the road color.

II. APPROACH

We recognize roads based on several properties common to most roads: they are generally long and thin following a piecewise smooth curve, they are mostly uniform in the long direction, but with contrast in the crosswise direction, and they often have at least some degree of bilateral symmetry.

We attempt to fit the image data with a simple spline based model that will fail if these properties are not present. The fitting is performed by a form of Stochastic Gradient Descent (SGD), a widely used method which randomly applies operators to the model to improve its fit to the data (for our problem to make it more road-like).

Gradient descent requires initialization to a model not too different from the actual road. We get initial models by extrapolating from a point that the hyperspectral sensor indicates could be part of a road. Once we have found a road, only small updates to the model are needed to track it from frame to frame, which gradient descent can do quickly.

Because our model is compact, it can be transmitted to the ground station over a low bandwidth link, eliminating the need to transmit video. We send it over the same radio link used by the autopilot for commands and telemetry. This is important because if the user wants the plane to follow a road, he will need to verify that the right road is being followed from the start, and will want some indication that the correct road continues to be followed afterwards. (Without displaying the road models during flights, testing and debugging road following would have been very difficult.)

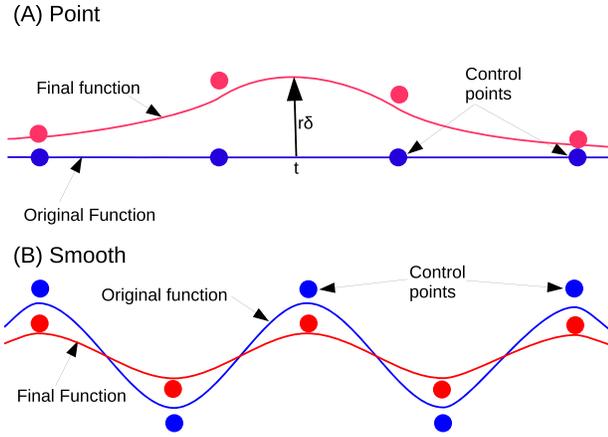


Fig. 1. Spline adjustment operators. Changes exaggerated for clarity. (A) Point - Function adjusted to better match selected point. (B) Smooth - Function smoothed.

A. Splines

Cubic B-splines are well-suited to modeling smooth functions of a parameter t (which in our case is position along the road).

B-splines model a function $f(t)$ as a weighted sum of a vector of control points X

$$f(t) = X^T B(t) = \sum_{i=0}^3 \sum_{j=0}^3 X_{[t]+i-1} \mathbf{M}_{ij}(t - [t])^j \quad (1)$$

Where

$$B_i(t) = \sum_{j=0}^3 \mathbf{M}_{(i-[t]+1)j}(t - [t])^j \quad (2)$$

is the cubic B-spline basis with coefficients

$$\mathbf{M}_{ij} = \begin{bmatrix} 1/6 & -1/2 & 1/2 & -1/6 \\ 2/3 & 0 & -1 & 1/2 \\ 1/6 & 1/2 & 1/2 & -1/2 \\ 0 & 0 & 0 & 1/6 \end{bmatrix} \quad (3)$$

for $0 \leq i \leq 3, 0 \leq j \leq 3$ and $\mathbf{M}_{ij} = 0$ otherwise.

There are two basic operations we apply to splines to update our model: Point adjustment and smoothing. They both take an old function f_0 represented by control points X_0 and produce a new function f represented by X , depending on a rate parameter r , (usually small) which represents the step size in SGD.

1) *Point Adjustment*: Point adjustment updates the function to reduce the error δ , measured at a single point t , by modifying the control points that affect $f(t)$ by

$$X = X_0 + \delta r B(t) \quad (4)$$

Since this only changes four control points, and the computation of the four nonzero components of $B(t)$ is shared with the measurement of δ , the update is fast. We need it to be, because it is applied many times. (Fig. 1(A)).

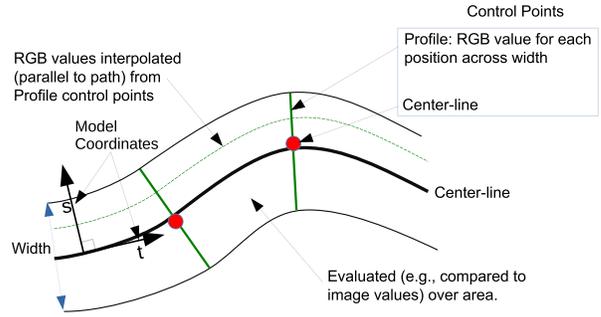


Fig. 2. The road model.

2) *Smoothing*: The smoothing operator makes a small change in the function to reduce its total second derivative. It finds the spline that minimizes the error:

$$E^2 = \int (r f''(t)^2 + (1-r)(f(t) - f_0(t))^2) dt \quad (5)$$

The first term represents curvature, and the second change relative to the old function, with r determining their relative importance. Since f is piecewise polynomial, the derivatives and integral can be evaluated symbolically, giving:

$$\int f''(t)^2 dt = X^T \mathbf{A} X \quad \int f(t)^2 dt = X^T \mathbf{P} X \quad (6)$$

Where \mathbf{A} and \mathbf{P} are the symmetric band matrices

$$\mathbf{A}_{uv} = \sum_m \sum_{i=2}^3 \sum_{j=2}^3 \mathbf{M}_{(u-m+1)i} \mathbf{M}_{(v-m+1)j} \frac{ij(i-1)(j-1)}{i+j-3}$$

$$\mathbf{P}_{uv} = \sum_m \sum_{i=0}^3 \sum_{j=0}^3 \frac{\mathbf{M}_{(u-m+1)i} \mathbf{M}_{(v-m+1)j}}{i+j+1} \quad (7)$$

In terms of \mathbf{A} and \mathbf{P} (5) is

$$E^2 = r X^T \mathbf{A} X + (1-r)(X - X_0)^T \mathbf{P} (X - X_0) \quad (8)$$

which is minimized by setting its derivatives to zero and solving for X , giving

$$X = \left(\frac{r}{1-r} \mathbf{A} + \mathbf{P} \right)^{-1} \mathbf{P} X_0 \quad (9)$$

\mathbf{A} is positive semi-definite and \mathbf{P} is positive definite, so for $0 \leq r < 1$ the matrix is always invertable. All matrix operations are limited to four non-zero diagonals, so splines can be smoothed in time proportional to their length. The cost of the inversion is reduced by pre-computing the LDL decomposition of the matrix to be inverted for each length and smoothing rate. (Fig. 1(B)).

B. Road Model

The Road model has two parts, the *center-line* a two-dimensional spline curve representing a path on the ground, and the *profile* comprising 21^1 three-dimensional splines representing RGB values along equally spaced paths parallel to the center-line (Fig. 2).

¹We chose this number because it was small enough for data transmission and large enough to represent the color image and for tracking. It worked well and was never modified.

The center-line is the primary output of the road-finding subsystem, and is used for road following. The profile is matched against image at the points determined by the center-line. To find and track the road, both parts of the model are adjusted to find the best match.

Points in the road model are parameterized by two coordinates t and s , where t , the spline parameter, is position along the center-line and s is perpendicular distance to the left or right (so $s = 0$ on the center-line). All splines share the continuous parameter t , with the same values at the equally spaced control points, while s is represented by 21 samples between -10 and 10 , and selects one of the profile splines.

The road model is expressed in ground coordinates. For each frame the mapping from ground coordinates to image coordinates is calculated based on the position and attitude reported by the autopilot. Because we only need the position of the plane relative to the road, absolute error in the reported pose is not important.

To counteract the effect of drift, we track the road from frame to frame. The projection of the actual road into the image changes as a function of the actual motion of the plane, while the projection of the road model changes according to the reported motion. Error in the reported motion causes a small misalignment between the road model and the image of the road. This is corrected each frame by the tracking process, so that the model and the image continue to agree. Because the correction is usually small, it can be computed very quickly.

C. Adjustment Operators

The basic algorithm for finding and tracking the road is stochastic gradient descent (SGD). Operators are applied at randomly chosen points in the model. Each application of an operator has a small effect, which is reduced for each iteration. Each of the six operators improves one desirable property of the model.

Two of the operators are smoothing using (9) on either the center-line or on a profile spline. There are also four operators that use point adjustment (4), one operating on profile and three on the center line. Each time one of these operators is used, it chooses a random position t along the spline to apply to.

1) *Profile Adjustment*: Profile adjustment changes the profile part of the model to more closely match the current image. In addition to choosing a random t , it also chooses a random lateral offset s . The color predicted by the spline at (t, s) is compared to the corresponding point in the image. The difference in color is used as the error for the point adjustment operation (4) on the profile spline indexed by s .

2) *Match Adjustment*: Match adjustment moves the center-line to improve the match between the model and the image. It picks random points t along the road model and finds the maximum of the correlation between the profile at that position and the line on the ground perpendicular to the center-line, as observed in the image. The center-line is then moved left or right using point adjustment (4) toward the offset that gives the best match. (Fig. 3(A)).

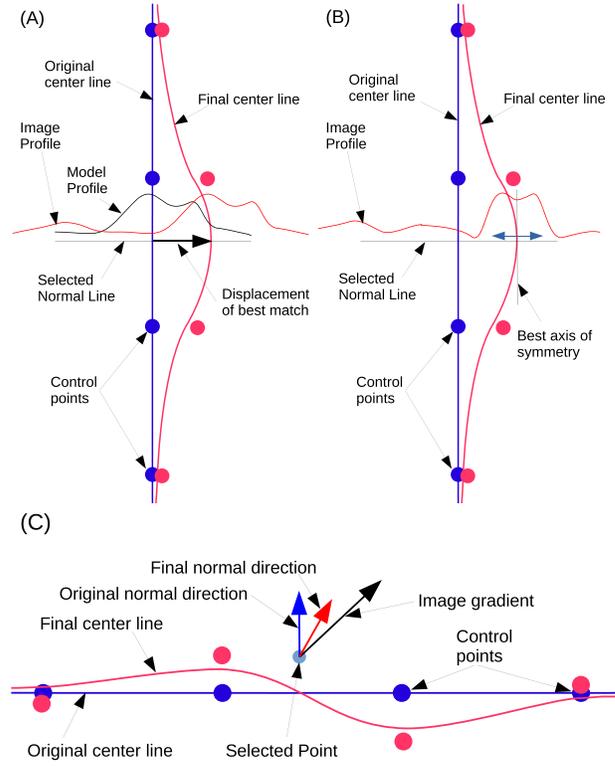


Fig. 3. Center Line adjustment operators. Changes exaggerated for clarity. (A) Match - Center line moved towards best match. (B) Bilateral - Center line moved towards best symmetry axis (C) Bend - Center line bent to move normal direction towards image gradient.

3) *Bilateral Adjustment*: Bilateral adjustment moves the center-line to try to make the road more bilaterally symmetric. Looks at the line on the ground perpendicular to the center line at t , finds the offset about which it is most symmetric, and moves the center-line in that direction. It ignores the profile. (Fig. 3(B)).

4) *Bend Adjustment*: Bend adjustment applies a small local rotation to the center-line to reduce the variation in the profile as a function of t . The direction of rotation is chosen according to the sign of $\frac{\partial f_{model}}{\partial t} \frac{\partial f_{image}}{\partial s}$, so that, for example, if f increases in the forward direction and to the left, then the center line is rotated to the right, reducing the increase along the curve. (Fig. 3(C)).

D. Road Finding

Finding the road initially is more computationally expensive than tracking it from frame to frame once it has been found. We employed a three step process to generate, refine, and select candidate road detections. First, seed points are found with the hyperspectral sensor, they are extended to linear features in a color image, which is fit by our spline based road model. If any of the steps fail to find something it can interpret as a road, the search starts over in the next frame. Because false alarms in each step are largely independent of each other, this results in few false alarms overall.

The hyperspectral sensor (HSS) is used to find “seed” points that can be extended into road detections, the process



Fig. 4. Top: The initial spline seeded from the outcome of least cost graph approach (dynamic programming). Bottom: Spline model is refined to fit image. This stage and preceding ones are not done every frame.

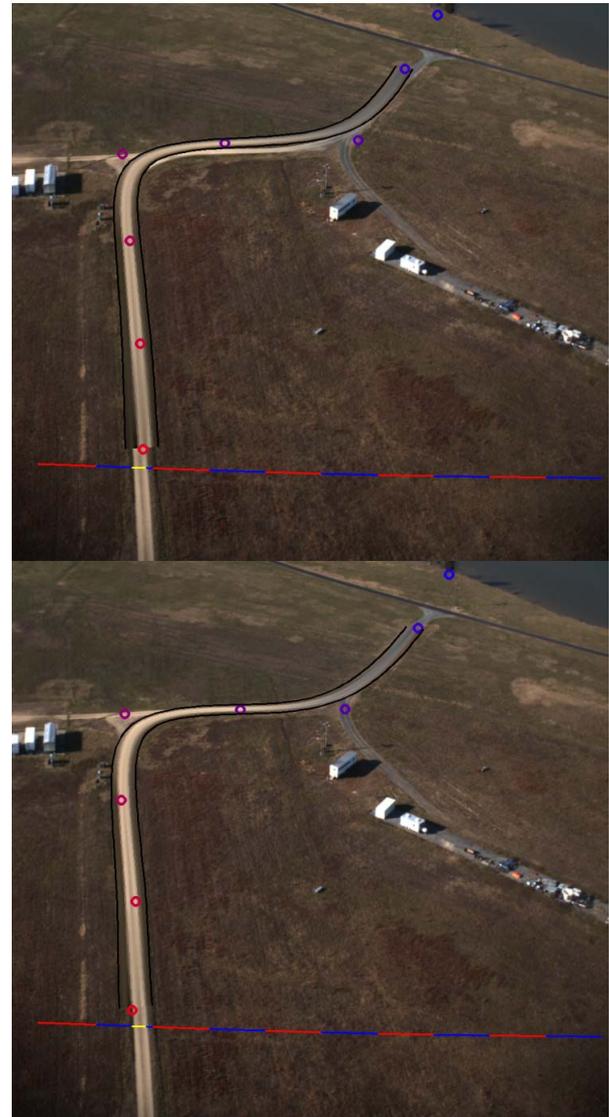


Fig. 5. Top: Spline model extrapolated to new frame. It is done every frame; cost is negligible (10 frame difference shown to illustrate mismatch). Bottom: Spline adjusted to fit new frame.

is very simple, but robust. Pixels of the hyperspectral sensor are first classified as “possible-road” or “non-road”, based on relative brightness in just two bands, $\approx 580\text{\AA}$ in the visible and $\approx 750\text{\AA}$ in the near IR. A pixel is possible-road if $0.2 < \frac{Vis-IR}{Vis+IR} < 0.4$. This reliably eliminated all pixels containing vegetation or water.² A seed point is the center of a contiguous set of possible-road pixels of a rather reasonable width projected onto the ground. (Of course, the user may supply an estimate of the width for greater confidence.) This detects most roads, but a special case was added to handle dirt roads with grass growing between the tire tracks.

Because the color camera and the HSS are not synchronized, the HSS frames in which seeds are found are usually not simultaneous with any color frame, and movement during the

intervening time must be taken into account. We seed the road in a color frame only if compatible seed points are found in both the preceding and the following HSS frames. The position on the seed is linearly interpolated to the time of the color image. This also eliminates any seed points that fail to persist for at least two HSS frames (small patches of pavement or bare ground).

Seeding by the HSS is easily fast enough to do in real time on every frame and has a low enough false alarm rate (though not zero) that when it indicates a possible road there is a reasonable probability that a road is actually present, and that it will be found in the color image. Combined with fast and reliable road tracking (see II-E), this means that road finding in the color image (which is the most expensive step) does not need to run very often, since most frames either have no seed point, or have a road that can be tracked from the previous frame.

²This ratio was recommended by the Remote Sensing Division at NRL. The effectiveness and the simplicity of this method ruled out the need for a more elaborate technique.

A seed point is extended to a path in the color image by a dynamic programming based search so that the total change along the path in HSV space is minimized. Control points are chosen at regular intervals along the path and used to initialize the spline based road model.

Once the path part of the road model has been initialized, the profile part is filled in by SGD using 10000 iterations of profile adjustment (Fig. 4 (Top)).

The complete road model is then refined by another round of SGD, using all the operators but match adjustment (100000 iterations, about 97% profile, 1 % bilateral, 1 % bend, 0.2% profile smoothing, 0.2% center-line smoothing). This is the most computationally expensive stage of the processing, but it doesn't need to be done very often, and it could be reduced by careful tuning. The high percentage of profile adjustments is dictated by the need to correct the profile each time the center-line moves (Fig. 4 (Bottom)).

E. Road Tracking

Once the road has been found, tracking it from frame to frame is much easier.

Tracking first adjusts the center-line, to correct misalignment between the road model and the road in the image (due to e.g inaccurate pose updates), and then the profile to correct for changes in illumination, and to improve resolution where the plane has moved closer to part of the road.

Center-line adjustment is done by SGD with just 500 iterations, mostly match adjustment interspersed with about 2% of center-line smoothing steps to prevent kinks from forming. Then the profile is corrected by 500 iterations of profile adjustment (Fig. 5).

After the road is tracked to a new frame, it is trimmed or extended. Trimming removes control points from the ends of the spline that no longer match the image (as measured during the tracking stage), or that are too far outside the field of view. If too little of the road model remains, the road is considered lost. The road is extended by searching the image beyond the end of the model for a matching profile, and adding a new control point if one is found.

F. Road Following

Once a road has been found and is being tracked, it has to be followed in a stable manner. There have been several studies in following curvilinear structures such as [2], [10], and [11]. In [14] cubic trajectories are constructed to control the aircraft so that it intercepts the target path on a tangent.

Road following could be done one of several ways, but note that our objective is not to always fly directly over the road, or even as close as possible. We want to fly a smooth stable path, while keeping the road well inside our field of view.

To do this, a target point for steering is chosen on the (modeled) road some distance ahead of the plane. This is a common path following method [10]. We used the point on the road where the horizontal distance from the plane is twice the plane's altitude, or if the road does not go that far, the end of the road. Choosing a point well ahead of the plane's current

position has the effect of smoothing and stabilizing the flight path, ignoring small wiggles in the road.

There is another reason to choose a target point well in front of the plane, it makes it easier to keep the target in sight while making the required turns. In order to turn to the right, the plane must roll to the right, and rolling to the right moves the field of view of the camera to the left. Suppose we chose a target too close to the plane and it is near the right edge of our field of view. If we turned right towards the target, the field of view would move to the left and the target would be lost.

This lead distance resulted in a smooth approach to the road, while keeping the road in the camera's field of view. It also compensates for crosswind. To fly parallel to the road in a crosswind, the plane must point upwind of the road direction. At the airspeed we fly ($\approx 13\text{m/sec}$), the angle is sometimes significant. As a result, if the plane tried to fly directly over the road with too much crosswind, the road would be outside of the camera's field of view. Turning the plane so that its *heading*, not its velocity vector, points toward the target point results in the plane's track being downwind of the road by the required distance. This method corrects for crosswind without even measuring it.

Simple proportional control is used to point the plane towards the target point, sending roll angle commands to the autopilot, proportional to the angle between the plane's heading and the target. The autopilot continues to control all other degrees of freedom (e.g. altitude, airspeed, rudder coordination).

III. AIRCRAFT

For this project an Ion Tiger aircraft is used with the fuel cell removed and replaced with batteries, our sensors and the road following computer. Ion Tiger is a custom-built electric aircraft that NRL Vehicle Research Section used primarily to test long-endurance hydrogen fuel cell propulsion technologies. Ion Tiger has a wing-span of 20ft and flight speeds of 25kt at cruise, it can cruise on approximately 250W of propulsion power after a ground-launch using a small winch motor.

The aircraft is controlled by a Cloud Cap Technologies Piccolo-II autopilot. The autopilot has an inertial navigation system, and communicates with a ground station over a radio link.

While not following roads, the aircraft is controlled by the autopilot in waypoint mode, except for launch and landing, which are controlled manually from the ground station.

The road following computer comprises an Intel Atom D525 1.8GHz processor, 4GB of memory, firewire interface, and a 160GB flash drive, in a mini-itx case.

All data received from the sensors or the autopilot is time-stamped and recorded.

IV. SENSORS

The visual sensor is a Stingray F145C color firewire camera with a 12mm lens producing 1388×1038 pixel images with a $40^\circ \times 30^\circ$ field of view. It is mounted on the bottom of the

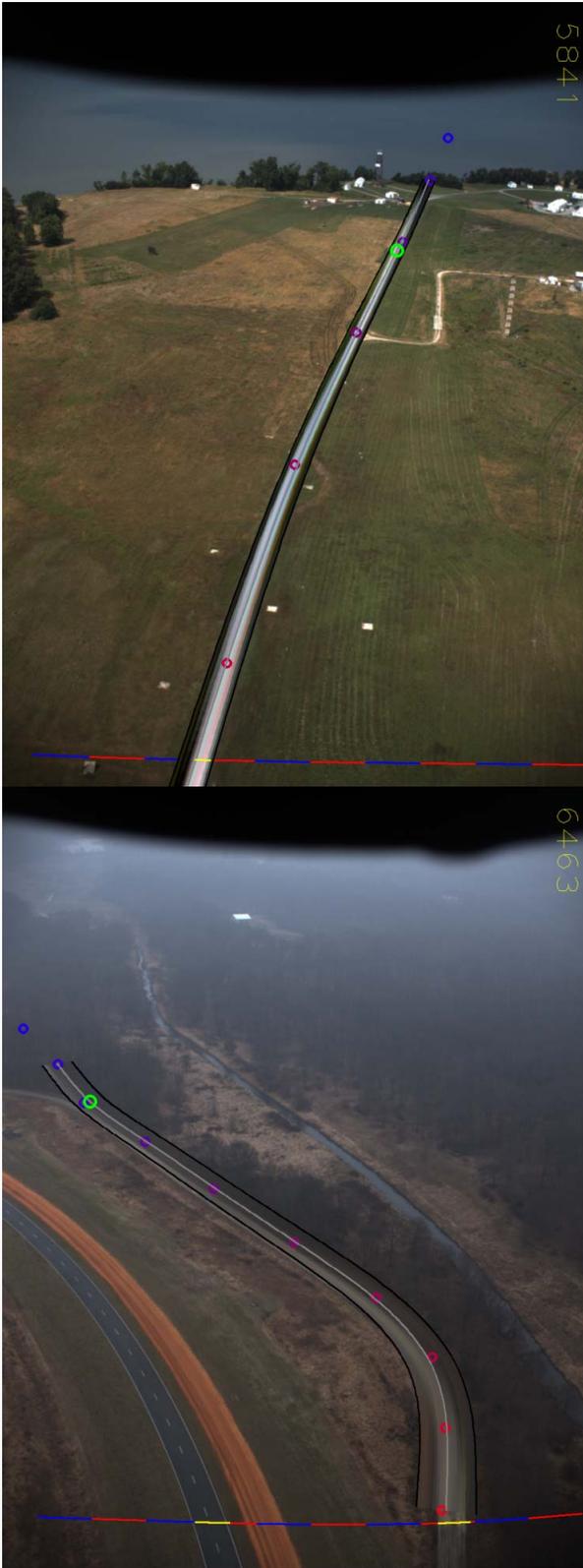


Fig. 6. Two test flight images (different seasons and test ranges) with road model superimposed. Green circle - target point. Blue and purple circles - Spline control points. Red and blue line - line seen by the HSS.

plane looking forward at about a 45° angle, oriented so that the wider field of view is vertical. Its internal calibration was determined with a checkerboard target [18], [19].

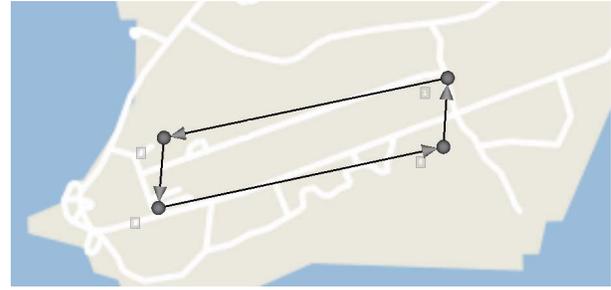


Fig. 7. This figure shows the Picolo map and waypoints for one of our test sites. (Roads are white, waypoints and arrows indicating clockwise direction are black.) The map is old and inaccurate. The two target roads are the two longest ones. The plane would fly toward a waypoint while looking for a road. Once a road was found it would follow it until that road ended. It would then fly toward the next waypoint. If the plane followed a (target) road for too long and was about to go out of sight, the command “Stop Following Road” would be issued, in which case the plane would turn back to fly toward the next waypoint.

The hyperspectral sensor (HSS) is a Specim V10E spectrograph with a 16mm objective lens and a Stingray F145B monochrome firewire camera. It measures brightness values for 518 wavelength bands from 360nm to 1000nm (but is only sensitive above 400nm) at each of 692 pixels along a line. The line observed by the HSS is horizontal, lying near the bottom of the field of view of the color camera.

The mapping from the HSS pixels to points in the color image is measured before each flight by finding a third order polynomial that minimizes differences over a set of calibration pairs each comprising one image of a distant target taken at the same time by each camera.

Gain and Exposure time for both sensors is continuously adjusted to optimize dynamic range. The maximum frame rate for the HSS and color cameras is set (by the requirement to save all images to disk) to 15Hz and 7.5Hz respectively. In some flights, cloudy weather sometimes required longer exposures for the HSS, reducing its frame rate to about 10Hz. The cameras are not synchronized, but each frame has an accurate timestamp.

V. TESTS

A. Simulation

Two types of simulation were used to test different parts of the road following software: replay and full simulation.

In a replay, the program is given data recorded from a previous flight. Images and autopilot data are delivered to the program at the same times as in the actual flight, either at real time rates, sped up, or slowed down. The advantage of a replay is that the data is exactly what the program would see in a real flight, so it is a realistic test, especially of the sensor processing. A common problem in testing in simulation is that assumptions made about the data may hold in the simulation, but not in the real world. That is not possible in a replay. However, a replay does not include any interaction with world, so it can test road detection, modeling, and tracking, but not the interaction with the autopilot required for actually following the road. Replay tests were used during program



Fig. 8. This a sequence of images showing the view seen by the aircraft during a short period of an actual test flight. The road model generated by the aircraft, which is the detected road, is overlaid on the images. The images show the impact of a tree canopy occluding the road at various degrees. The flight is not impacted significantly as the aircraft rediscovers the road.

development (using data from flights using only waypoint control), and for analysis of road following results.

In full simulation, the entire system is simulated using simulations of the autopilot, vehicle dynamics, and sensors. This tests the interaction between the program and the world, but the data (especially image data) is not as representative of real conditions. Our simulation uses aerial images to generate images as they would be seen by the color camera, ignoring the 3 dimensional structure of the scene (which is sometimes significant). The simulation of the HSS is much cruder, since

hyperspectral data was not available. Hyperspectral pixels were classified by thresholding the RGB values at the corresponding point of the color image. This works for some roads (that were significantly brighter than the background) but not for others that the real HSS does detect.

B. Flight

The system as a whole can only be tested in an actual flight. Fig. 6 shows two still images from a test flight.

From gathering test data to tests of the final system, we flew a total of 14 flights each lasting up to an hour. Some were cut short due to hardware or software problems or weather conditions. Flights were done over a number of roads at two separate test ranges and in all four seasons. Flights in different seasons provided us with information regarding the system's performance across a variety of color changes, which especially impacts background from which the road has to be delineated, as well as different weather conditions.

During the final successful test flight, which lasted about an hour, the plane flew a circuit of four waypoints, two sides of which were roughly aligned with the two target roads. When the road follower found a road, it took control and steered the plane until the road was lost, usually at the road's end. It then returned the autopilot to waypoint control, until the next road was found. The roads could only be followed for about a minute at a time, due to the safety requirement that the aircraft remain in sight of the ground station at all times. It flew ten cycles around the circuit at three different altitudes 100m, 150m, and 200m. In each cycle both roads were found and followed, but at the lower altitudes one of roads was not always found immediately because the placement of the waypoints and the crosswind conditions resulted in the road being outside of the camera's field of view. In general, road following is harder at lower altitude because time limits are shorter and the area inside the field of view is smaller (Fig. 7).

The simple 2-band method of road detection with HSS worked remarkably well in all seasons as brown vegetation remained very bright in near IR wavelength, and that the system as a whole was not affected by the season.

Some "false" detections occurred, many of which were actually short segments of road. None were followed for more than a few seconds.

C. On the Accuracy of Actual Flights

Considering the variety of roads, seasons and weather conditions in which we conducted actual test flights, the accuracy of the flight path of the aircraft cannot provide a meaningful metric representing the success rate. In fact, as mentioned earlier, our objective was not to always fly directly over the road; rather to fly a smooth stable path, while keeping the road well inside the field of view. If the plane tried to fly directly over the road with too much crosswind, the road would be outside of the camera's field of view. Therefore, a more relevant question is the accuracy with which the road has been detected, though even here quantification will not be very meaningful. We are thus content with making only certain qualitative observations in regard to road detection accuracy. In the section "Conclusions and Future Work," we discuss a number of situations where a vision system alone may not be adequate for road following. If a rural road goes through a wooded area, a common example of such complications is tree canopy. In Fig. 8 a sequence of images show the view from the aircraft during a short period of an actual test flight. The road model generated by the aircraft, i.e. the detected road, is overlaid on the image. The images show the impact of a tree canopy occluding the road to various degrees. The flight

is not impacted significantly as the aircraft rediscovers the road. However, if the canopy was too large/long the outcome could be different. In the absence of complexities such as crossing roads, forks, occlusion (caused mostly by trees), our vision system always detected the road (as far as needed for guidance) with virtually perfect accuracy while flying at 100, 150 or 200 meters altitude. (We also experimented at 50 meter altitude, but the flight was not always stable as the view of the road ahead could be limited.) Furthermore, if the road was visible too far ahead, the distant segment was not always detected correctly, but that had no impact, especially because it was typically beyond the length needed for guidance, i.e. twice the plane altitude.

VI. CONCLUSIONS AND FUTURE WORK

We have shown that following roads is possible under a variety of conditions when using a color camera and a hyperspectral sensor in a complementary manner. That is not to say that the road following problem is solved. Roads are harder to follow than many other extended landmarks, such as coastlines. In particular, roads are very diverse structures. Moreover, they are often ambiguous, e.g. slowly morphing into a path, and then to nothing, or splitting into multiple roads.

Some roads are difficult or even impossible to see from the air. Trees (rural), buildings (urban), etc. can partially or totally obscure a road (especially a small road), or make it visible only from certain directions. Some roads can be seen only from directly overhead. Even if the road is straight, this can be difficult when the plane has not yet reached the road or if due to cross wind, the plane does not fly directly over the road, as explained in section II-F.

Single isolated roads are rare. Where there is one road, there are almost always others, all of which are connected to form a network. It makes sense to say "follow the coastline," there is often only one coastline around, but it is usually ambiguous to say "follow the road." Which road? When the road forks, which way should I go? To follow a road in a useful sense is really to follow a specified route in a road network.

In some ways, finding and navigating a road network may be *easier* than doing the same with a single road. Our assumptions about the appearance of roads (section II) naturally break down at the end of a road. That is not a problem because it is correct to stop following them there. However, the assumptions also break down at intersections. Intersections are much more common than road ends (most road ends are also intersections), but the road often needs to be followed beyond the intersection. Our method usually succeeded in doing this but often with some difficulty. If the intersecting road was also found and tracked, the vicinity of the intersection could be excluded from the model, resulting in a better and more reliable fit. Intersections with other roads could also be used as evidence confirming the existence of a road, especially at its end. Roads with sharp turns, which are not modeled well by a smooth curve, could be represented as two roads that both end at their intersection.

Most roads appear on some sort of map. Finding a road using a map should be much easier than finding one from

scratch, even if the map is not perfectly accurate, or the aircraft's position is not known exactly. Dirt roads have some properties in common with interstate highways, but knowing what type of road to look for can only help. That is especially true when looking for one particular road, not just any road.

It could also help to use any imagery available, even if out of date or out of season. An image of a particular road, even if only approximate, could supply a template for finding it.

We plan to develop an approach that uses our proposed road following method in conjunction with a pre-existing map containing the route of interest.

ACKNOWLEDGMENT

We are grateful to Chris Bovais, Steve Carruthers and Dan Edwards for maintaining and piloting the plane for actual flights, and Jeffrey Bowles for his valuable help with the hyperspectral sensor.

REFERENCES

- [1] J. Kellogg *et al.*, "The NRL micro tactical expendable (MITE) air vehicle," *Aeronaut. J.*, vol. 106, no. 1062, pp. 431–442, 2002.
- [2] P. Baker and B. Kamgar-Parsi, "Using shorelines for autonomous air vehicle guidance," *Comput. Vis. Image Understanding*, vol. 114, no. 6, pp. 723–729, 2010.
- [3] S. Rathinam *et al.*, "Autonomous searching and tracking of a river using an UAV," in *Proc. Amer. Control Conf.*, 2007, pp. 359–364.
- [4] A. Xu and G. Dudek, "A vision-based boundary following framework for aerial vehicles," in *Proc. Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 81–86.
- [5] K.-H. Chen and W.-H. Tsai, "Vision-based autonomous land vehicle guidance in outdoor road environments using combined line and road following techniques," *J. Robot. Syst.*, vol. 14, no. 10, pp. 711–728, 1997.
- [6] J. Alvarez and A. Lopez, "Road detection based on illumination invariance," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 1, pp. 184–193, Mar. 2011.
- [7] H. Zhou, H. Kong, L. Wei, D. Creighton, and S. Nahavandi, "Efficient road detection and tracking for unmanned aerial vehicle," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 297–309, Feb. 2015.
- [8] Z. W. Kim, "Realtime road detection by learning from one example," in *Proc. IEEE Workshop Appl. Comput. Vis.*, Jan. 2005, pp. 455–460.
- [9] H. Kong, Y.-J. Audibert, and J. Ponce, "General road detection from a single image," *IEEE Trans. Image Process.*, vol. 19, no. 8, pp. 211–220, Aug. 2010.
- [10] E. Frew *et al.*, "Vision-based road-following using a small autonomous aircraft," in *Proc. IEEE Aerosp. Conf.*, vol. 5, Mar. 2004, pp. 3006–3015.
- [11] S. Rathinam, Z. W. Kim, and R. Sengupta, "Vision-based monitoring of locally linear structures using an unmanned aerial vehicle," *J. Infrastruct. Syst.*, vol. 14, no. 1, pp. 52–63, 2008.
- [12] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.
- [13] E. Dubrofsky, "Homography Estimation," Ph.D. dissertation, Dept. Comput. Sci., Univ. British Columbia, Vancouver, BC, USA, 2009.
- [14] R. Frezza, G. Picci, and S. Soatt, "A Lagrangian formulation of nonholonomic path following," in *The Confluence of Vision and Control*. Springer, 1998, pp. 118–133.
- [15] R. Holt and R. Beard, "Vision-based road-following using proportional navigation," *J. Intell. Robot. Syst.*, vol. 57, nos. 1–4, pp. 193–216, 2010.
- [16] W. R. Phillipson and W. R. Hafker, "Manual versus digital landsat analysis for delineating river flooding," *Photogram. Eng. Remote Sens.*, vol. 47, pp. 1351–1356, Sep. 1981.
- [17] H. Liu and K. Jezek, "Automated extraction of coastline from satellite imagery by integrating canny edge detection and locally adaptive thresholding methods," *Int. J. Remote Sens.*, vol. 25, no. 5, pp. 937–958, 2004.
- [18] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, vol. 1, Sep. 1999, pp. 666–673.
- [19] J. Bouguet. (2013). *Camera Calibration Toolbox for MATLAB*. [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/



Ralph Hartley received the B.S. degree in mathematics from University of Rochester in 1980 and the Ph.D. in computer science from University of Maryland in 1985. Since 1985, he has been a Research Scientist with the Naval Research Laboratory, Washington, DC, USA. He has performed a wide variety of research in computer vision, robotics, autonomous vehicles, image processing (both 2-D and 3-D), and machine learning. His research has ranged from purely theoretical work to development and construction of hardware. He designed and built a new range sensor including both the hardware and image processing, for which he holds a patent. He has published results on robotic architectures, path planning, image processing, computer vision, and neural networks. He has extensive experience with machine learning methods including genetic algorithms with stochastic evaluation functions.



Behrooz Kamgar-Parsi (SM'18) received the B.Sc. degree in physics from Tehran University and the Ph.D. degree in physics from Catholic University of America. From 1985 to 1988, was a Research Scientist with the Computer Vision Laboratory, University of Maryland, College Park. From 1988 to 1989, he was an Assistant Professor with the Computer Science Department, George Mason University, Fairfax, VA, USA. Since 1989, he has been a Senior Research Scientist with the Naval Research Laboratory, Information Technology Division, Washington, DC, USA. He conducted research on medical research at the NIMH, National Institutes of Health, from 1978 to 1981, and numerical analysis while at the Center for Information Technology, National Institutes of Health, Bethesda, MD, from 1981 to 1984. He has authored many scientific publications and holds several US patents. His current research interests include pattern recognition, neural networks, biometrics, and vision system for autonomous air vehicles. He has been an Associate Editor of *Journal Pattern Recognition Letters*.



Cody Narber received the B.S. degree from Bloomsburg University, majoring in both mathematics and computer science, in 2006 and the M.S. degree in computer science from Kansas State University in 2008, with his thesis focusing on affective speech detection. He is currently pursuing the Ph.D. degree with George Mason University. His doctoral work is focused on haptic technology and how it can be used to identify features (strengths, weaknesses, and strategies) about the user, primarily for use by medical clinicians. He has contributed through both teaching and research, and was awarded the outstanding graduate teaching assistant award in 2009. He has published several articles on machine learning and human computer/systems interaction.