Vision-Guided Flight Stability and Control for Micro Air Vehicles[*]

Scott M. Ettinger
scott.m.ettinger@intel.com
Microprocessor Research Lab
Intel Corporation

Michael C. Nechyba
nechyba@mil.ufl.edu
Electrical and Computer Engineering
University of Florida
Gainesville, FL 32611-6200

Peter G. Ifju
pgi@mae.ufl.edu
Mechanical and Aerospace Engineering
University of Florida
Gainesville, FL 32611-6250

Martin Waszak
m.r.waszak@larc.nasa.gov
Dynamics and Control Branch
NASA Langley Research Center
MS 132, Hampton, VA 23681-2199

March 3, 2004

Abstract

*Substantial progress has been made recently towards designing, building and test-flying remotely piloted Micro Air Vehicles (MAVs) and small UAVs. We seek to complement this progress in overcoming the aerodynamic obstacles to flight at very small scales with a **vision-guided flight stability and autonomy system**, based on a robust horizon detection algorithm. In this paper, we first motivate the use of computer vision for MAV autonomy, arguing that given current sensor technology, vision may be the only practical approach to the problem. We then describe our statistical vision-based horizon detection algorithm, which has been demonstrated at 30Hz with over 99.9% correct horizon identification. Next, we develop robust schemes for the detection of extreme MAV attitudes, where no horizon is visible, and for the detection of horizon estimation errors, due to external factors such as video transmission noise. Finally, we discuss our feedback controller for self-stabilized flight, and report results on vision-based autonomous flights of duration exceeding ten minutes. We conclude with an overview of our on-going and future MAV-related research.*

Keywords: vision-based control, aerial robotics, real-time control.

1 Introduction

Ever since humankind's first powered flight, research efforts have continually pushed the envelope to create flying machines that are faster and/or larger than ever before. Now, however, there is an effort to design aircraft at the other, largely unexplored end of the spectrum, where the desire for portable, low-altitude

---

[*]An earlier version of this paper was presented at IROS2002 [5].

Figure 1: (a) six-inch UF MAV, (b) six-inch UF MAV in flight with view through its on-board camera, and (c) 24-inch small UAV.

aerial surveillance has driven the development and testing of aircraft that are as small and slow as the laws of aerodynamics will permit – in other words, on the scale and in the operational range of small birds. Vehicles in this class of small-scale aircraft are known as *Micro Air Vehicles* or *MAVs*.

Equipped with small video cameras and transmitters, MAVs have great potential for surveillance and monitoring tasks in areas either too remote or too dangerous to send human scouts. Operational MAVs will enable a number of important missions, including chemical/radiation spill monitoring, forest-fire reconnaissance, visual monitoring of volcanic activity, surveys of natural disaster areas, and even inexpensive traffic and accident monitoring. Additional on-board sensors can further augment MAV mission profiles to include, for example, airborne chemical analysis.

In the military, one of the primary roles for MAVs will be as small-unit battlefield surveillance agents, where MAVs can act as an extended set of eyes in the sky for military units in the field. This use of MAV technology is intended to reduce the risk to military personnel and has, perhaps, taken on increased importance in light of the U.S.'s new war on terrorism, where special operations forces are playing a crucial role. Virtually undetectable from the ground, MAVs could penetrate potential terrorist camps and other targets prior to any action against those targets, significantly raising the chance for overall mission success.

Researchers in the Aerospace Engineering Department at the University of Florida have established a long track record in designing, building and test-flying (remotely human-piloted) practical MAVs [7, 8, 12, 13]. For example, Figure 1 shows one of our recently developed MAVs as well as a small UAV design. While much progress has been made in the design of ever smaller MAVs by researchers at UF and others in the past five years, significantly less progress has been made towards equipping these MAVs with autonomous capabilities that could significantly enhance the utility of MAVs for a wide range of missions.

The first step in achieving such MAV autonomy is basic stability and control. Here, we present such a *flight stability and control system*, based on vision processing of video from a camera on-board our MAVs. In this paper, we first motivate the use of computer vision for such a control system, and describe our *vision-based horizon detection algorithm*, which forms the basis of the flight stability system presented here. Next, we address real-time control issues in the flight stability system, including extreme attitude detection (i.e.

2

no horizon in the image), confidence measures for the detected horizon estimates, and filtering of horizon estimates over time. Finally we report some results of self-stabilized MAV flights over the campus of the University of Florida and over Fort Campbell, Kentucky. We conclude with an overview of our on-going and future MAV-related research.

2 Horizon detection

MAV flight stability and control presents some difficult challenges. The low moments of inertia of MAVs make them vulnerable to rapid angular accelerations, a problem further complicated by the fact that aerodynamic damping of angular rates decreases with a reduction in wingspan. Another potential source of instability for MAVs is the relative magnitudes of wind gusts, which are much higher at the MAV scale than for larger aircraft. In fact, wind gusts can typically be equal to or greater than the forward airspeed of the MAV itself. Thus, an average wind gust can immediately affect a dramatic change in the vehicle's flight path.

Birds, the biological counterpart of mechanical MAVs, can offer some important insights into how one may best be able to overcome these problems. In studying the nervous system of birds, one basic observation holds true for virtually all of the thousands of different bird species: *Birds rely heavily on sharp eyes and vision to guide almost every aspect of their behavior* [2, 3, 6, 10, 11, 17]. Biological systems, while forceful evidence of the importance of vision in flight, do not, however, in and of themselves warrant a computer-vision based approach to MAV autonomy. Other equally important factors guide this decision as well. Perhaps most critical, the technologies used in rate and acceleration sensors on larger aircraft are not currently available at the MAV scale. It has proven very difficult, if not impossible, to scale these technologies down to meet the very low payload requirements of MAVs. While a number of sensor technologies do currently exist in small enough packages to be used in MAV systems, these small sensors have sacrificed accuracy for reduced size and weight. Take, for example, MEMs (Micro Electro-Mechanical Systems) rate gyros and accelerometers. MEMs piezoelectric gyros, while only weighing approximately one gram, have drift rates on the order of $100°$ per minute and are highly sensitive to changes in temperature. While elaborate temperature calibration procedures can improve their accuracy somewhat, their use in inertial navigation is problematic at best.

Even if sufficient rate and acceleration sensors did exist, however, their use on MAVs may still not be the best allocation of payload capacity. For many potential MAV missions, vision may be the only practical sensor than can achieve required and/or desirable autonomous behaviors. Furthermore, given that surveillance has been identified as one their primary missions, MAVs must necessarily be equipped with on-board imaging sensors, such as cameras or infrared arrays. Thus, computer-vision techniques exploit already present sensors, rich in information content, to significantly extend the capabilities of MAVs, without increasing the MAV's required payload.

2.1   Horizon-detection algorithm

Fundamentally, flight stability and control requires measurement of the MAV's angular orientation. While for larger aircraft this is typically estimated through the integration of the aircraft's angular rates or accelerations, a vision-based system can directly measure the aircraft's orientation with respect to the ground. The two degrees of freedom critical for stability – the *bank angle* ϕ and the *pitch angle* θ [1] – can be derived from a line corresponding to the horizon as seen from a forward facing camera on the aircraft. Therefore, we have developed a vision-based horizon-detection algorithm that lies at the core of our flight stability system, and which rests on two basic assumptions: (1) the horizon line will appear as approximately a straight line in the image; and (2) the horizon line will separate the image into two regions that have different appearance; in other words, sky pixels will look more like other sky pixels and less like ground pixels, and vice versa. The question now is how to transform these basic assumptions into a workable algorithm.

The first assumption – namely, that the horizon line will appear as a straight line in the image – reduces the space of all possible horizons to a two-dimensional search in line-parameter space. For each possible line in that two-dimensional space, we must be able to tell how well that particular line agrees with the second assumption – namely that the correct horizon line will separate the image into two regions that have different appearance. Thus our algorithm can be divided into two functional parts: (1) for any given hypothesized horizon line, the definition of an optimization criterion that measures agreement with the second assumption, and (2) the means for conducting an efficient search through all possible horizons in two-dimensional parameter space to maximize that optimization criterion.

2.2   Optimization criterion

For our current algorithm we choose color, as defined in RGB space, as our measure of appearance. In making this choice, we do not discount the potential benefit of other appearance measures, such as texture; however, in exploring possible feature extraction methods, we believe that simple appearance models ought to precede pursuit of more advanced feature extraction methods.

For any given hypothesized horizon line, we label pixels above the line as sky, and pixels below the line as ground. Let us denote all hypothesized sky pixels as,

$$x_i^s = \begin{bmatrix} r_i^s & g_i^s & b_i^s \end{bmatrix}, \ \ i \in \{1, \cdots, n_s\}, \tag{1}$$

where r_i^s denotes the red channel value, g_i^s denotes the green channel value and b_i^s denotes the blue channel value of the ith sky pixel, and let us denote all hypothesized ground pixels as,

$$x_i^g = \begin{bmatrix} r_i^g & g_i^g & b_i^g \end{bmatrix}, \ \ i \in \{1, \cdots, n_g\}. \tag{2}$$

Given these pixel groupings, we want to quantify the assumption that sky pixels will look similar to other sky pixels, and that ground pixels will look similar to other ground pixels. One measure of this is the degree

---

[1]Instead of the pitch angle θ, we actually recover the closely related pitch percentage σ, which measures the percentage of the image below the horizon line.

of variance exhibited by each distribution. Therefore, we propose the following optimization criterion:

$$J = \frac{1}{|\Sigma_s| + |\Sigma_g| + (\lambda_1^s + \lambda_2^s + \lambda_3^s)^2 + (\lambda_1^s + \lambda_2^s + \lambda_3^s)^2} \tag{3}$$

based on the covariance matrices Σ_s and Σ_g of the two pixel distributions,

$$\Sigma_s = \frac{1}{(n_s - 1)} \sum_{i=1}^{n_s} (x_i^s - \mu_s)(x_i^s - \mu_s)' \tag{4}$$

$$\Sigma_g = \frac{1}{(n_g - 1)} \sum_{i=1}^{n_g} (x_i^g - \mu_g)(x_i^g - \mu_g)' \tag{5}$$

where,

$$\mu_s = \frac{1}{n_s} \sum_{i=1}^{n_s} x_i^s, \quad \mu_g = \frac{1}{n_g} \sum_{i=1}^{n_g} x_i^g \tag{6}$$

and λ_i^s and λ_i^g, $i \in \{1, 2, 3\}$, denote the eigenvalues of Σ_s and Σ_g, respectively.

For video frames with sufficient color information, the determinant terms in (3) will dominate, since the determinant is a product of the eigenvalues; however, for cameras with poor color characteristics or video frames exhibiting loss of color information due to video transmission noise, the covariance matrices may become ill-conditioned or singular. When this is the case, the sum-of-eigenvalues terms will become controlling instead, since the determinants will evaluate to zero for all possible horizon lines.

Assuming that the means of the actual sky and ground distributions are distinct (a requirement for a detectable horizon, even for people), the line that best separates the two regions should exhibit the lowest variance from the mean. If the hypothesized horizon line is incorrect, some ground pixels will be mistakenly grouped with sky pixels and vice versa. The incorrectly grouped pixels will lie farther from each mean, consequently increasing the variance of the two distributions. Moreover, the incorrectly grouped pixels will skew each mean vector slightly, contributing further to increased variance in the distributions.

2.3 Maximizing the optimization criterion

Given the J optimization criterion in equation (3), which allows us to evaluate any given hypothesized horizon line, we must now find that horizon line which maximizes J. As we have stated previously, this boils down to a search in two-dimensional line parameter space, where our choice of parameters are the bank angle ϕ and pitch percentage σ with ranges,

$$\phi \in [-\pi/2, \pi/2] \ \text{ and } \ \sigma \in [0\%, 100\%]. \tag{7}$$

To meet real-time processing constraints [2], we adopt a two step approach in our search through line-parameter space. We first evaluate J at discretized parameter values in the ranges specified by (7) on down-sampled images with resolution $X_L \times Y_L$. Then, we fine-tune the coarse parameter estimate from the previous step

---

[2]See [4] for details on additional algorithmic considerations.

through a bisection-like search about the initial guess on a higher resolution image ($X_H \times Y_H$, $X_L \ll X_H$, $Y_L \ll Y_H$). Further details on the search part of the algorithm may be found in [4].

Thus, we can summarize the horizon-detection algorithm as follows. Given a video frame at $X_H \times Y_H$ resolution:

1. Down-sample the image to $X_L \times Y_L$, where $X_L \ll X_H$, $Y_L \ll Y_H$.

2. Evaluate J on the down-sampled image for line parameters (ϕ_i, σ_j), where,

$$(\phi_i, \sigma_j) = \left(\frac{i\pi}{n} - \frac{\pi}{2}, \frac{100j}{n} \right), \;\; 0 \le i \le n, \;\; 0 \le j \le n \tag{8}$$

3. Select (ϕ^*, σ^*) such that,

$$J|_{\phi=\phi^*, \sigma=\sigma^*} \ge J|_{\phi=\phi_i, \sigma=\sigma_j}, \;\; \forall i, j. \tag{9}$$

4. Use bisection search on the high-resolution image to fine-tune the values of (ϕ^*, σ^*).

At this point, the reader might be wondering whether a full search of the line-parameter space (even at coarse resolution) is really required once flying, since the horizon at the current time step should be very close to the horizon at the previous time step; perhaps speed improvements could be made by limiting this initial search. There is, however, at least one important reason for conducting a full search with every frame, rather than just a partial search in the neighborhood of the previous frame's horizon estimate. Assume, for example, that the algorithm makes an error in the horizon estimate at time t; then, at time $t + 1$, a limited search could permanently lock us into the initial incorrect horizon estimate, with potentially catastrophic results.

We illustrate this idea in Figure 2, where we show four frames from a flight sequence over the University of Florida campus. Note that the first frame is severely corrupted by video transmission noise, so that the horizon estimate for the first frame in the sequence is far from the true horizon. In Figure 2(a), a full search is conducted for every frame, so that the noise-corrupted estimate of the first frame does not influence subsequent horizon estimates. In Figure 2(b), however, we restrict the search for the horizon line to a limited range around the previous frame's horizon estimate; note that with the limited search, subsequent horizon estimates, continue to exhibit large error. Thus, a full, coarse search of line parameter space guards against this type of cascading failure due to single-frame errors.

Finally, we point out that since we perform our full search through line-parameter space for a single frame such that consecutive horizon hypotheses are very close to one another [4], we need only compute the full statistics in the J optimization criterion for the first horizon hypothesis. Subsequent computations are incremental, and typically involve only a few pixel values that have to be added to one class and subtracted from the other class. As such, when implemented efficiently, the full search through line parameter space is significantly less computationally burdensome than might be expected.
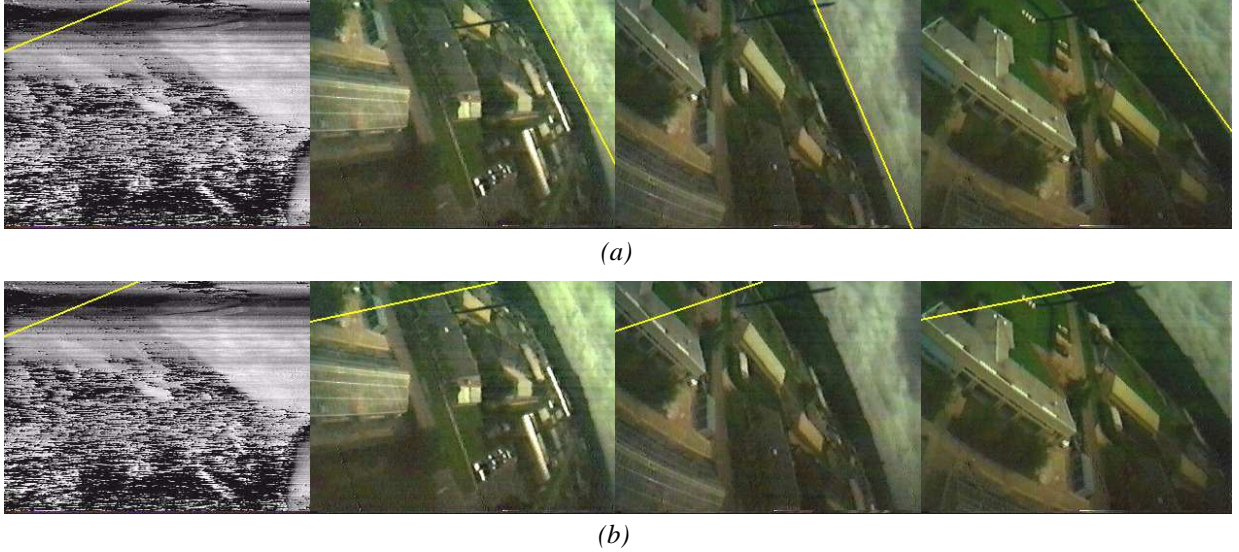
(a)

(b)

Figure 2: (a) Video noise in the first frame does not influence subsequent horizon estimation when performing a full search of the line-parameter space for every frame; (b) when consecutive searches are restricted to a limited range around the previous frame's horizon estimate, however, the horizon detection algorithm is not able to recover from the first noise-corrupted estimate.

2.4   Horizon-detection examples

Figure 3 illustrates several examples of the horizon-detection algorithm at work, while Figure 4 illustrates a more detailed example plotting J as a function of the bank angle and pitch percentage, and the consequent classification of pixels as sky and ground in RGB space. Additional examples and videos can be found at `http://mil.ufl.edu/~nechyba/mav`. Our horizon-detection algorithm has been demonstrated to run at 30Hz on a 900MHz x86 processor with a down-sampled image of $X_L \times Y_L = 80 \times 60$ resolution, a search resolution of $n = 36$, and a final image of $X_H \times Y_H = 320 \times 240$ resolution. If such computing power is not available, we have shown only slightly reduced performance at values as low as $X_L \times Y_L = 40 \times 30$, $n = 12$ and $X_H \times Y_H = 160 \times 120$.

At different times of the day, and under both fair and cloudy conditions, we have gathered hours of video on-board our MAV, flying under manual control over terrain that includes roads, buildings large and small, meadows, wooded areas, and a lake. For these data, our horizon-detection algorithm correctly identifies the horizon in over 99.9% of cases.

3   Flight stability and control

In this section, we extend the basic horizon-detection algorithm developed in the previous section to *real-time horizon tracking*. Below, we consider the following important issues: (1) extreme attitude detection, (2) error detection in horizon estimation, (3) filtering of the horizon estimate over time, and (4) basic feedback control and stabilization of the MAV.

Figure 3: Various horizon-detection examples under different lighting conditions (sunny and cloudy), and with varying degrees of video transmission noise. For each example, the yellow line indicates the algorithm's horizon estimate.
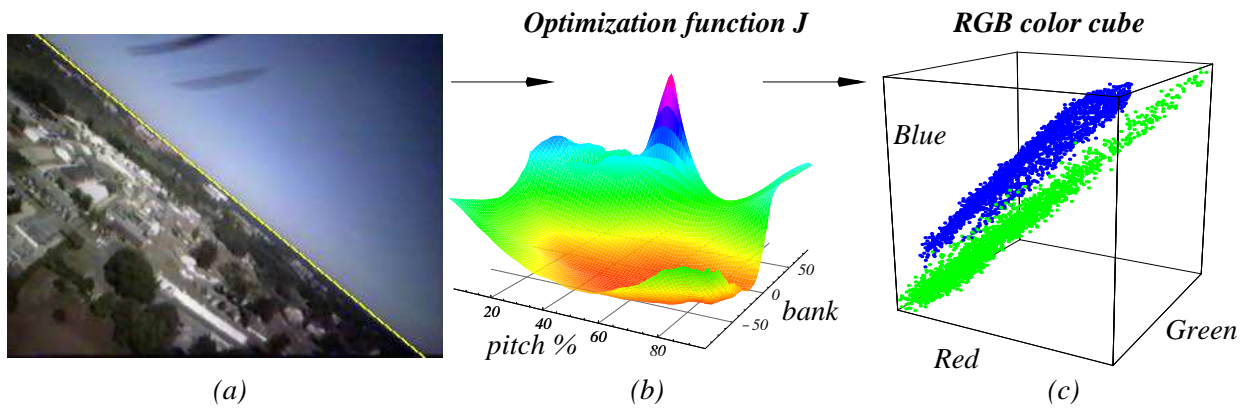

Figure 4: (a) original image, (b) optimization criterion as a function of bank angle and pitch percentage and (c) resulting classification of sky and ground pixels in RGB space.

3.1 Extreme attitude detection

One of the implicit assumptions of the horizon detection algorithm is that there will always be a horizon in the images from the forward looking camera on board the MAV. In real-time control of the MAV, the MAV may, however, encounter times when no visible horizon appears in the image, if, for example, a gust of wind forces the nose of the aircraft too far up or down. Such cases cannot simply be ignored; if the aircraft is heading straight towards the ground, no horizon will be visible in the camera image, yet the control system will certainly be required to take action to save the MAV from certain and possibly catastrophic crashing.

It is desired then, to be able to detect instances when the horizon is not in view of the camera, and if so to determine what action to take in order to bring the horizon back into view. There are two valuable sources of information which we can draw on to detect these types of extreme attitudes: (1) recent appearance of the sky and ground from previous time steps, and (2) recent location of the horizon line from previous time steps. For example, if the horizon line was recently estimated to lie near the top of the image, it is logical that a subsequent image without a horizon line is most likely a view of the ground. We can use these two pieces of information to quantitatively determine if the horizon line exists in the image and if not, to determine whether we are looking at the sky or the ground.

Using the statistics already computed as part of the horizon-detection algorithm, we can model the appearance of the sky and ground over a recent time history of the MAV's flight. Our general approach for detection of extreme attitudes keeps running statistical models for both the sky and ground from previous frames, where horizon lines were detected with a high degree of confidence. With each new frame, the result of the horizon detection algorithm can be checked by comparing the sky and ground models for the current frame with the computed, time-dependent statistical models for sky and ground. If the distributions on either side of the line in the current frame both appear to be more similar to the known ground distribution, then it would appear that the aircraft is pointing towards the ground. Conversely, if they both match the sky better, then it is advisable to nose downward. Interestingly, if the sky in the current frame matches the ground model while the ground in the current frame matches better with the sky model, we can detect situations where the plane is flying upside down.

One additional piece of information is required to implement the extreme attitude detection scheme, namely, a time history of the horizon line estimate. For the purposes of detecting extreme attitudes, we are most concerned with a recent history of the pitch percentage σ, the percentage of the image below the horizon line. One measure of that history is a running average σ_{avg} of the pitch percentage over the previous ten frames.

Upon startup of the system, the camera is assumed to be oriented such that the horizon is in its view. When the first frame of video is processed by the system, the means and covariance matrices of the ground and sky models are set equal to those found by the horizon detection algorithm. The system then begins to update the models using the results of the horizon detection algorithm for a set number of initialization frames. Our current implementation uses 100 initialization frames (3.3 seconds). Once boot-strapped, it is

necessary to continually update the sky and ground models as the aircraft flies to account for changes in lighting associated with changes in orientation and changes in landscape, etc. The running statistical models are updated as follows:

$$\Sigma_{s(t)} = \alpha \Sigma_{s(t)} + (1 - \alpha) \Sigma_s \tag{10}$$

$$\Sigma_{g(t)} = \alpha \Sigma_{g(t)} + (1 - \alpha) \Sigma_g \tag{11}$$

$$\mu_{s(t)} = \alpha \mu_{s(t)} + (1 - \alpha) \mu_s \tag{12}$$

$$\mu_{g(t)} = \alpha \mu_{g(t)} + (1 - \alpha) \mu_g \tag{13}$$

where $\Sigma_{s(t)}$, $\Sigma_{s(t)}$, $\mu_{s(t)}$ and $\mu_{g(t)}$ are the time-dependent model covariances and means, respectively, while Σ_s, Σ_g, μ_s and μ_g are the covariances and means for the current frame. Note that the constant α controls how rapidly the models change over time.

For a new image, we first compute the estimated horizon for that image. We then compare the resultant current statistics with the running statistical models from previous frames, using the following four distance measures:

$$D_1 = (\mu_s - \mu_{s(t)})' \Sigma_{s(t)}^{-1} (\mu_s - \mu_{s(t)}) + (\mu_s - \mu_{s(t)})' \Sigma_s^{-1} (\mu_s - \mu_{s(t)}) \tag{14}$$

$$D_2 = (\mu_s - \mu_{g(t)})' \Sigma_{g(t)}^{-1} (\mu_s - \mu_{g(t)}) + (\mu_s - \mu_{g(t)})' \Sigma_s^{-1} (\mu_s - \mu_{g(t)}) \tag{15}$$

$$D_3 = (\mu_g - \mu_{s(t)})' \Sigma_{s(t)}^{-1} (\mu_g - \mu_{s(t)}) + (\mu_g - \mu_{s(t)})' \Sigma_g^{-1} (\mu_g - \mu_{s(t)}) \tag{16}$$

$$D_4 = (\mu_g - \mu_{g(t)})' \Sigma_{g(t)}^{-1} (\mu_g - \mu_{g(t)}) + (\mu_g - \mu_{g(t)})' \Sigma_g^{-1} (\mu_g - \mu_{g(t)}) \tag{17}$$

The value of D_1 measures the similarity between the region selected as the sky by the horizon detection algorithm in the current frame and the sky model from recent frames. D_2 represents the similarity between the currently computed sky region and the ground model from recent frames. Likewise, the values of D_3 and D_4 are the similarity measures between the current ground region and the sky and ground models from recent frames, respectively. Table 1 now summarizes four possible cases and the conclusions we are able to draw for each case.

The determinations in the above table can be combined with the past history of the horizon line to decide what action to take. If the current frame is determined to be normal by the validity test (case 1), then the horizon estimate is assumed to be accurate, and commands sent to the MAV are determined by the normal

case	condition	conclusion
1	$D_1 < D_2$ and $D_3 > D_4$	valid horizon present
2	$D_1 > D_2$ and $D_3 > D_4$	all ground
3	$D_1 < D_2$ and $D_3 < D_4$	all sky
4	$D_1 > D_2$ and $D_3 < D_4$	upside down

Table 1: Extreme attitude detection

control system loop described in Section 3.4. Also, the statistics of the validated frame are used to update the sky and ground models per equations (14) through (17). If the validity test returns a higher likelihood of all ground (case 2), we verify that result with the recent history of the horizon line σ_{avg} to determine what action to take. When the value of σ_{avg} is above a set threshold, then the system goes into a "pull-up" mode that sends commands to the aircraft to rapidly increase its pitch angle. A value of 0.8 was used for this threshold. While the system is in pull-up mode, the time-dependent statistical models are not updated since the horizon estimate during this time will most likely be incorrect. Also during pull-up mode, σ_{avg} is only updated with the estimated value of σ if the validity test indicates the current frame has returned to a visible horizon line; otherwise, σ_{avg} is updated using a value of 1.01. The system will stay in pull up mode until a valid horizon is detected. Similarly, if the validity test returns a higher likelihood of all sky (case 3) and the value of σ_{avg} is below a given threshold (set at 0.2), the system goes into a "nose-down" mode. Updating of the time-dependent statistical models and σ_{avg} in nose-down mode is the same as in pull-up mode, except that the default update value for σ_{avg} is 0.01 instead of 1.01.

3.2   Error detection in horizon estimates

Extreme attitude detection can also help us to detect possible errors in the horizon estimation algorithm; such errors can occur when transient noise causes video degradation. Consider, for example, the following possibility: the validity test returns case 2 (all ground), but $\sigma_{avg} < 0.8$. In this situation, we must assume an error occurred in horizon detection, because the aerodynamic characteristics of the plane do not permit such sharp changes in pitch over 1/30th of a second. More generally, if the validity test returns any of the non-normal cases (2, 3 or 4) and the value of σ_{avg} does not conform to the appropriate threshold values, we consider the horizon detection for that frame to be in error. In this case, the horizon estimate from the previous frame is used to estimate the horizon parameters for the current frame.

From extensive flight testing, we observer qualitatively that this extreme attitude and error detection system performs well. It is difficult to quantitatively assess the performance of the system on real-time data since there is no "correct" answer with which to compare it. Both the qualitative viewing of the output, however, along with successful flight tests indicate that the system performs adequately.

3.3   Kalman filtering

In order to make the horizon estimates usable for self-stabilization and control, the horizon estimates, after being processed by the extreme attitude and error detector, are passed through a Kalman filter [1]. The Kalman filter provides an optimal estimate of a system's current state, given a dynamic system model, a noise model, and a time series of measurements. While a dynamic model of the system is desirable, the formalism of the Kalman filter can be employed even without an accurate dynamic model. Since no dynamic model is readily available for our flexible-wing MAVs [3], we model the system state (the two parameters of the horizon estimate) as two simple first-order, constant-velocity systems. As such, the Kalman filter has the effect of removing high frequency noise from the system measurements and eliminating any radical single frame errors not first caught by the error detection system. The principal benefit of the Kalman filter for our application is that it effectively eliminates unnecessary small control surface deflections due to noise.

3.4   Feedback control

To date, we have employed a very simple controller to validate vision-based flight stability and control for MAVs. For simplicity, the bank angle ϕ and pitch percentage σ are treated as independent from one another, and for both parameters, we implement a simple PD (proportional/derivative) feedback control loop, with gains determined experimentally from flight tests; each control loop is updated at full frame rate (i.e. 30 Hz). In initial flight tests, the derivative gains were set to zero.

4   Self-stabilized flight

4.1   Experimental setup

Figure 5 illustrates our current experimental setup. The video signal from the MAV is transmitted from the plane through an antenna to a ground-based computer, where all vision processing is performed. In manual mode, the plane is controlled during flight by a remote human pilot through a standard radio link. In autonomous mode, the plane is controlled through the feedback controller which sends control surface commands to the MAV through a custom designed interface over the same radio link. Our interface allows the PC to control a standard Futaba radio transmitter through an RS-232 serial port.

The MAV used for test flights is the one depicted in Figure 1(c). While we have designed and flown MAVs with wing spans as small as six inches, we selected the somewhat larger platform both for its increased dynamic time constants and its ability to carry a high-powered video transmitter (i.e. increased payload). The on-board camera is a monolithic CMOS type camera with a 1/3 inch sensor area, and is connected to an 80 mW video transmitter. The MAV is powered by electric propulsion and has differential elevons for control, although the software is written to support both elevon and rudder-elevator control designs.

---

[3]As is explained later in this article, we are currently working on developing a dynamic model of our MAVs and small UAVs.
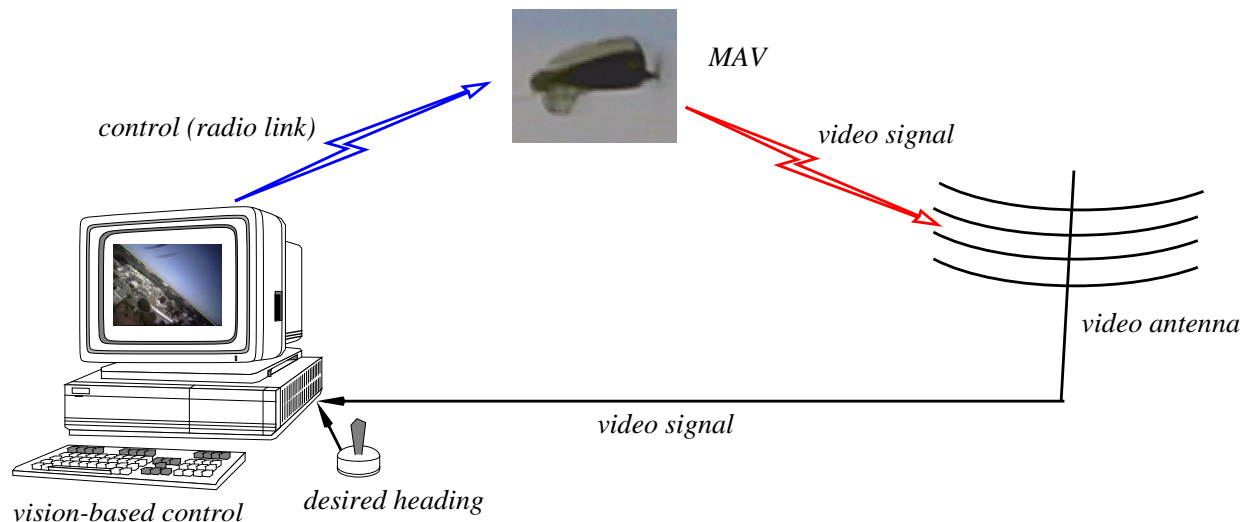
Figure 5: Experimental setup for video-based flight control.

The PC interface uses a PIC microcontroller to translate serial commands from the PC into the pulse-width modulated signals required for input to the transmitter. A carbon fiber housing was constructed to hold the circuit board and port connectors for the interface.

4.2 Flight testing

Flight testing proceeds as follows. Prior to launch, the aircraft is oriented such that the horizon is in the field-of-view of the camera. This allows the algorithm to build initial models of the sky and the ground; while these models are not used in the horizon-detection algorithm itself, they are used for extreme attitude and error detection.

Upon launch, flights are controlled by a human pilot until the MAV reaches sufficient altitude. At that point, control is transferred to the automated flight control and stability system; in case of catastrophic failure (loss of video signal, etc.), the radio transmitter is equipped with an override button to allow the human pilot to regain control at any time if necessary.

A joystick connected to the PC can be used to adjust the desired heading for the controller. The joystick input effectively commands a bank and pitch angle for the aircraft to follow. Later flights used a pre-programmed set of maneuvers for truly autonomous flight. To date, we have flown uninterrupted autonomous flights of over 10 minutes, flights that ended only due to video transmission interference, or low on-board battery power.

Figure 6(a) plots a 72-second run of actual flight data, where the flight vehicle was under vision-guided control above the University of Florida campus (the full length of the flight exceeded 10 minutes, and was primarily limited by low battery power). During this flight, the MAV was instructed to execute a trajectory that consisted of straight line segments, followed by left-bank turns (to keep the MAV within range of the
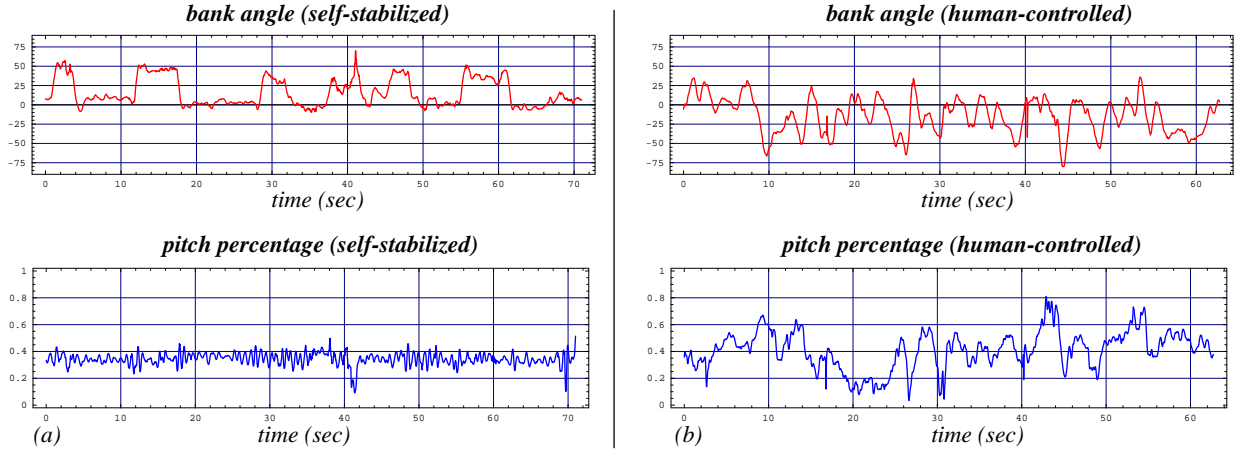
13

Figure 6: (a) Bank angle and pitch percentage for a self-stabilized flight (sequence of level-flight and left-turn segments), and (b) bank angle and pitch percentage for typical human-controlled flight.

receiving video antenna). For comparison, we also plot a 65-second segment of manual (human-controlled) flight in Figure 6(b). (Videos corresponding to these and other flight segments can be viewed at `http://mil.ufl.edu/~nechyba/mav`.) More recently, the same vision-based control system successfully flew over substantially different terrain at a Special Ops demo over Fort Campbell, Kentucky, where audience members, who had never previously controlled any type of aircraft (e.g. model airplane, MAV, etc.) successfully kept the MAV in the air for extended flight times.

From Figure 6, we observe that the human-controlled flight is much more erratic than the self-stabilized flight with respect to both the bank angle and pitch percentage. It is important to note that the erratic nature of the human-piloted flight is not caused by inexperience on the part of the pilot, who had flown our MAVs and small UAVs for many hours over a number of years; in, fact, he was one of our most experienced pilots. Rather, the seemingly erratic nature of the human-piloted flight indicates that controlling MAVs and small UAVs is significantly more difficult than controlling larger, more traditional model airplanes, which tend to exhibit much more passive stability.

Thus, qualitatively, even our simple PD control system provides much more stable control than that of our best human pilots, both in terms of steady, level flight, and in coordinated turns. As illustrated by Figure 6(b), human pilots can typically not hold the plane on a steady, level heading for more than a few fractions of a second; under vision-guided control, however, we were able to fly long straight segments that were limited only by the range of the video transmitter (see Figure 7, for example, for a 7.5 second stretch of self-stabilized straight and level flight). Prior to the development of the horizon-tracking control system, only pilots with extensive training could learn to fly our micro air vehicles; with the automated control system, however, people who have never piloted any aircraft before are able to easily guide the MAV above the flying arena. It is this fact alone that speaks the most to the potential value of this work. Ideally, one wants MAVs to be deployable by a wide range of people, not only expert RC pilots; while much remains

14

Figure 7: Image sequence of self-stabilized level flight (7.5 seconds, images are 20 frames apart). Note that the black lines at the top of the image are the propeller.

to be done, including automating landings and take-offs, the work in this paper is a big step towards the development and deployment of usable and practical MAVs.

4.3   Discussion

While the J optimization criterion defined in equation (3) performs well over a wide range of weather conditions and landscapes, it occasionally fails in difficult flight images. Consider, for example, the sample image and estimated horizon in Figure 8(a), which was taken from a flight video captured at the 6th International MAV Competition in Provo, Utah (for the complete video, see `http://mil.ufl.edu/~nechyba/mav`). For this example, the false horizon estimate is due to the sharp contrast between the deep blue sky and the bright, white clouds. To deal with these types of errors, we have begun to experiment with a second optimization criterion J_2 as defined below:

$$J_2 = (\mu_s - \mu_g)'(\Sigma_s + \Sigma_g)^{-1}(\mu_s - \mu_g) \tag{18}$$

Note that the J_2 optimization criterion, rather than minimizing the *intra-class* variance of the sky and ground distributions, instead maximizes the *inter-class* differences (i.e. the Mahalanobis distance) between the two distributions. It not only incorporates the variance information from the two distributions, but the two means as well; as such, the J_2 optimization criterion can succeed where the J optimization criterion fails, as illustrated in Figure 8(b). Note how the J_2 optimization criterion exhibits three peaks in the optimization surface – the smaller two peaks correspond to the sky/cloud line (as identified with the J criterion) and the

RGB color cube   Optimization function J

Blue
Red   Green

(a)

RGB color cube   Optimization function J$_2$
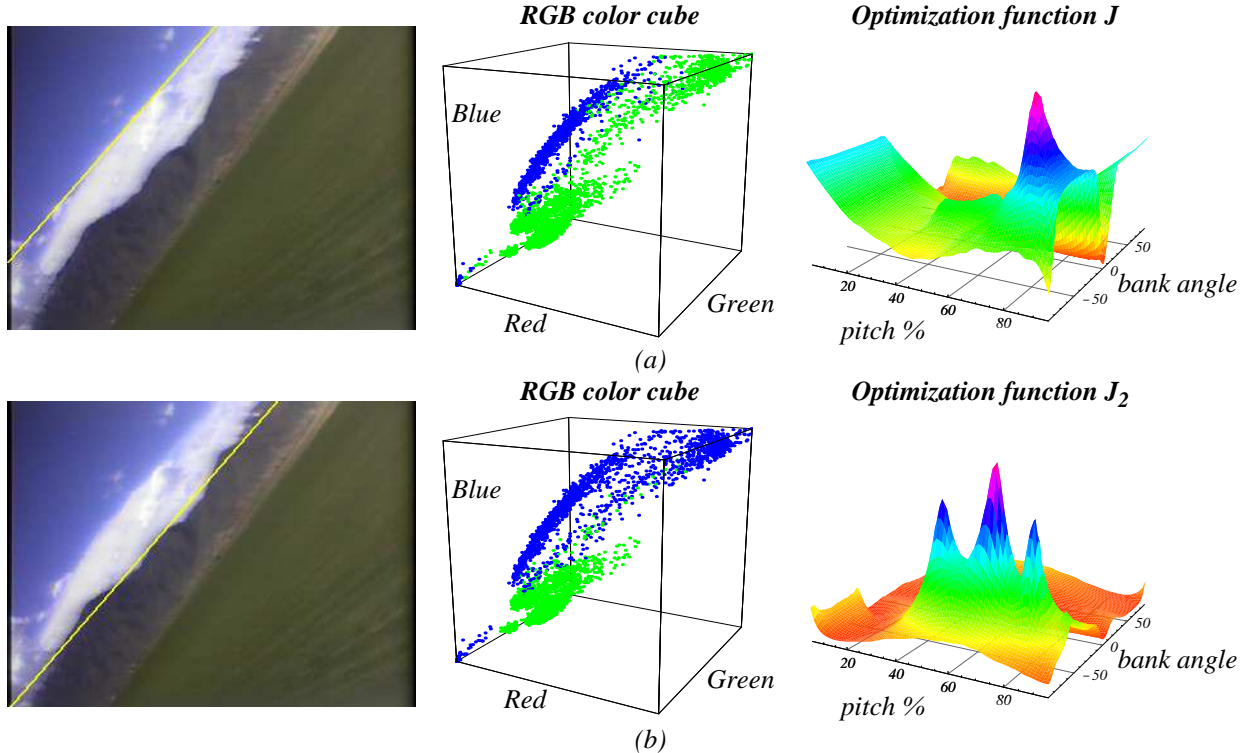
Blue
Red   Green

(b)

Figure 8: Comparison of optimization criteria for a difficult image: (a) J criterion and (b) J_2 criterion. In each sub-figure, we plot the estimated horizon line, the consequent classification of pixels in RGB space, and the optimization criterion over the line parameter space.

meadow/mountain line, while the global peak estimates the true horizon line very closely. We are continuing to experiment with this and other criteria to further improve the robustness of the overall horizon-tracking algorithm. Moreover, we have begun work in incorporating prior statistical models of sky and ground in horizon detection. These prior models incorporate both color and texture cues, and explicitly characterize dependencies of neighboring pixels at multiple resolutions using the Hidden Markov Tree (HMT) modeling framework [14, 16].

5   Conclusion

This paper describes our initial work towards flight autonomy for MAVs and small UAVs. We presented a vision-based flight stability and control system for MAVs and small UAVs that relies on no sensors except for a forward looking camera; furthermore, this system appears to produce more stable flights than those remotely controlled by a human pilot.

For our research team, these are, however, only the first steps towards our eventual goal of fully autonomous, mission-capable MAV systems. Work within our research team is on-going, and we have made significant progress in recent months. First, we have developed on-board GPS capability that provides heading information, not presently available through the vision system. Our compact GPS subsystem is pictured
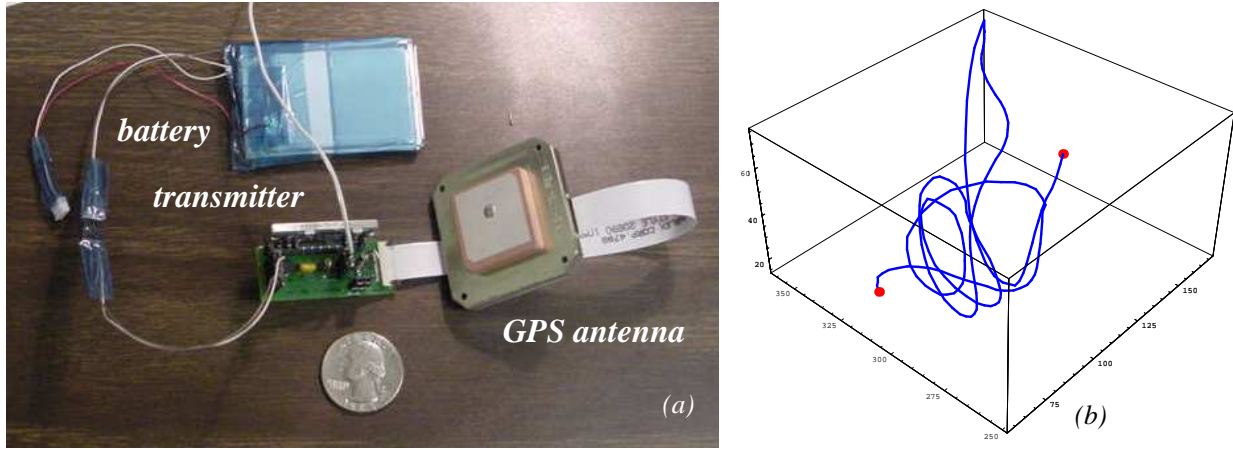
Figure 9: (a) Small, on-board GPS system, and (b) data from a recent, remotely controlled test flight.

in Figure 9, along with the GPS trajectory of a recent test flight of the system on-board a 24-inch UAV. With the antenna and related electronics and transmitter, the GPS subsystem weighs approximately 30 grams, which we believe will allow us to incorporate this GPS on platforms with wingspans of approximately 12 inches or less.

Second, we are in the final stages of incorporating the Microstrain 3DM-G, three-axis sensor [9], which has been developed recently and weighs less than 20 grams. While not suitable for stability control of smaller MAVs, we will use this sensor to measure the state of our small UAVs in flight along with servo positions to develop dynamic models of our UAVs. Once we characterize our flight vehicles, we will then be in a much better position to select appropriate control gains, without extensive experimental tuning, as well as to analyze the stability of the overall control system. To date, such stability analysis has not been possible, since we have had no good dynamic models of our flight vehicles.

We expect to be able to conduct fully autonomous, GPS way-point navigated flights by the summer of 2003, combining either the GPS sensor with our vision stabilization system, or the GPS sensor with the 3DM-G sensor (for the large vehicles in our small UAV fleet). Once we have achieved basic guidance, control and navigation, we will have enabled one of the most important missions for these systems, namely, remote surveillance (see Figure 10).

On the computer vision front, horizon tracking, while very useful, does not make full use of this important sensor. Currently, we are developing multi-resolutional methods for detecting man-made and other artificial structures on the ground [15]. We are also investigating structure-from-motion (SFM) algorithms for recovering the 3d scene structure in the vicinity of the UAV during flight; we expect that this will eventually enable more advanced mission profiles that include flying in complex 3d environments such as urban settings. Finally, we are exploring other optimization criteria and features, such as texture, to extend our current approach to monochrome imaging sensors (such as low-light cameras).
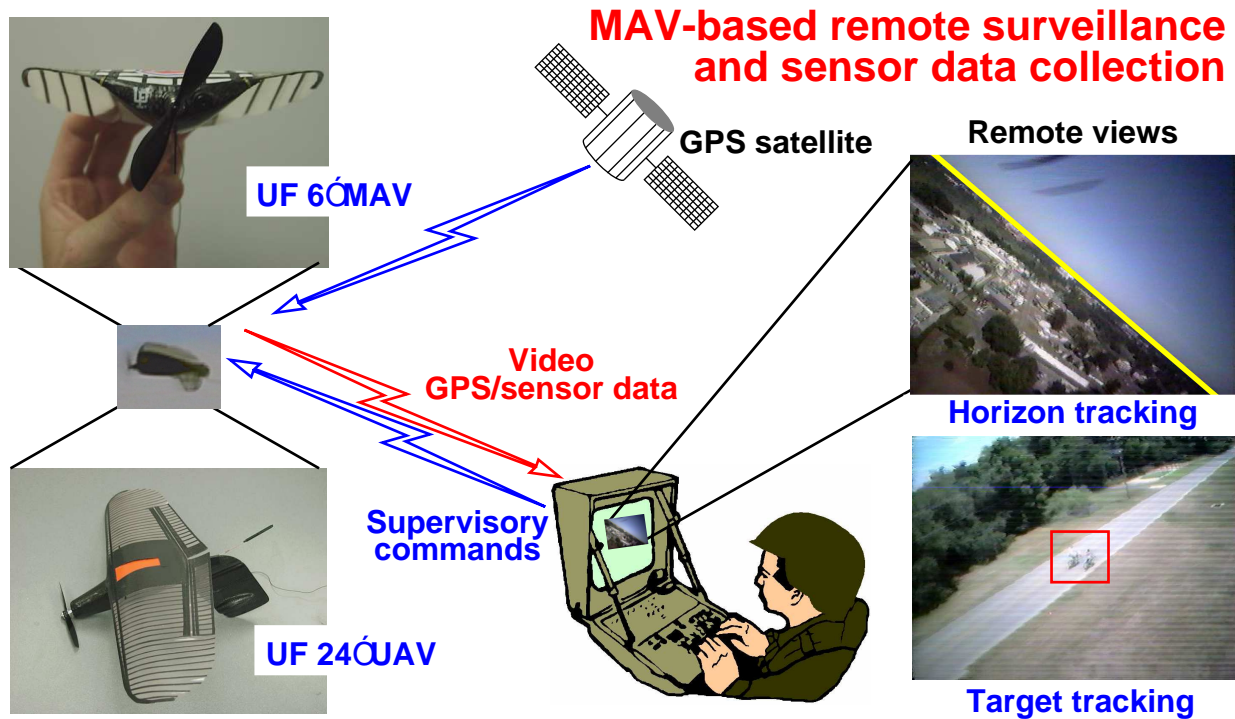
17

Figure 10: Remote surveillance application for GPS-guided, autonomous MAVs and small UAVs.

Acknowledgments

Biographical sketches

Scott M. Ettinger: Scott M. Ettinger received his B.S. and M.S. degrees from the University of Florida in Electrical Engineering with an emphasis in computer architecture, machine learning, and aerospace applications. In 2001, Scott joined Intel Research at Intel Corporation. His current areas of research include computer vision, nano-technology applications, and robotics. Scott has previously worked at the NASA Langley Research Center on design and control of Micro Air Vehicles.

Michael C. Nechyba: Michael C. Nechyba received his B.S. degree in Electrical Engineering at the University of Florida in 1992, and his Ph.D. in Robotics at Carnegie Mellon University in 1998. Upon completion of his thesis, Michael joined the Department of Electrical and Computer Engineering at the University of Florida as assistant professor in August 1998. There he serves as associate director

of the Machine Intelligence Laboratory. His research interests include machine learning, computer vision and robotics. Currently, Michael's research is focused on two primary areas: (1) vision-based and sensor-based autonomy for Micro Air Vehicles (MAVs), and (2) direct brain-machine interfaces (BMIs). He has published over 30 journal and refereed conference papers.

Peter G. Ifju: Peter G. Ifju received his B.S. degree in Civil Engineering in 1986, his M.S. degree in Engineering Mechanics in 1989, and his Ph.D. in Material Engineering Science in 1992, all from Virginia Polytechnic Institute. He then went on to a post-doc position at the NASA Langely Research Center from 1992 to 1993. In 1993 he joined the Department of Aerospace Engineering, Mechanics and Engineering Science at the University of Florida as assistant professor in November 1993, and was promoted to associate professor in 2000. His research interests include experimental mechanics, Moire interferometry, composite materials, and micro air vehicles (MAVs). He has published dozens of journal and refereed conference papers.

Martin Waszak: Martin Waszak is a Senior Researcher in the Dynamics and Control Branch at the NASA Langley Research Center. Marty received his Bachelor and Master degrees from the School of Aeronautics and Astronautics at Purdue University in 1981 and 1984, respectively. He also pursued post graduate studies at Purdue. Since joining NASA in 1987, Marty has been involved in research addressing mathematical modeling and control system design for aeroelastic vehicles. This research has been applied to flight control, flutter suppression, and multidisciplinary design studies. Most recently, Marty has been involved in research involving biologically inspired flight systems and micro aerial vehicles.

References

[1] B. D. O. Anderson and J. B. Moore. *Optimal Filtering.* Prentice-Hall, Englewood Cliffs, 1979.

[2] P. R. Ehrlich, D. S. Dobkin, and D. Wheye. Adaptions for flight. World Wide Web, `http://www.stanfordalumni.org/birdsite/text/essays/Adaptions.html`, June 2001.

[3] P. R. Ehrlich, S. Dobkin, and D. Wheye. Flying in vee formation. World Wide Web, `http://www.stanfordalumni.org/birdsite/text/essays/Flying_in_Vee.html`, June 2001.

[4] S. M. Ettinger. Design and implementation of autonomous vision-guided Micro Air Vehicles. M.S. Thesis, University of Florida, Electrical and Computer Engineering, August 2001.

[5] S. M. Ettinger, M. C. Nechyba, P. G. Ifju, and M. Waszak. Vision-guided flight stability and control for Micro Air Vehicles. In *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, volume 3, pages 2134–40, Laussane, Switzerland, 2002.

[6] R. Fox, S. W. Lehmkule, and D. H. Westendorf. Falcon visual acuity. *Science*, 192:263–5, 1976.

[7] P. G. Ifju, S. M. Ettinger, D. A. Jenkins, and L. Martinez. Composite materials for Micro Air Vehicles. In *Proc. SAMPE Annual Conference*, Long Beach, May 2001.

[8] D. A. Jenkins, P. G. Ifju, M. Abdulrahim, and S. Olipra. Assessment of controllability of Micro Air Vehicles. In *Proc. 16th Int. Conf. on Unmanned Air Vehicle Systems*, Bristol, United Kingdom, April 2001.

[9] Inc. Microstrain. Inclinometers and orientation sensors. World Wide Web, `http://www.microstrain.com/3DM-G.htm`, June 2001.

[10] Northern Prairie Wildlife Research Center. Migration of birds: Orientation and navigation. World Wide Web, `http://www.npwrc.usgs.gov/resource/othrdata/migration/ori.htm`, June 2001.

[11] G. Ritchison. Ornithology: Nervous system: Brain and special senses ii. World Wide Web, `http://www.biology.eku.edu/RITCHISO/birdbrain2.html`, June 2001.

[12] W. Shyy, D. A. Jenkins, and R. W. Smith. Study of adaptive shape airfoils at low Reynolds number in oscillatory flows. *AIAA Journal*, 35:1545–48, 1997.

[13] R. W. Smith and W. Shyy. Computation of aerodynamics coefficients for a flexible membrane airfoil in turbulent flow: A comparison with classical theory. *Phys. Fluids*, 8(12), 1996.

[14] S. Todorovic. Image modeling using the tree-structured Hidden Markov Model. M.S. Thesis, University of Florida, Electrical and Computer Engineering, December 2002.

[15] S. Todorovic and M. C. Nechyba. Multiresolution linear discriminant analysis: efficient extraction of geometrical structures in images. In *Proc. IEEE Int. Conference on Image Processing*, volume 1, pages 1029–32, September 2003.

[16] S. Todorovic, M. C. Nechyba, and P. Ifju. Sky/ground modeling for autonomous MAVs. In *Proc. IEEE Int. Conference on Robotics and Automation*, volume 1, pages 1422–7, Taipei, Taiwan, September 2003.

[17] G. C. Whittow, editor. *Sturkie's Avian Physiology*. Academic Press, San Diego, 5th edition, 2000.