Towards Flight Autonomy: Vision-Based Horizon Detection for Micro Air Vehicles

Scott M. Ettinger[1,2] sme@aemes.aero.ufl.edu
Michael C. Nechyba[1] nechyba@mil.ufl.edu
Peter G. Ifju[2] pgi@aero.ufl.edu
Martin Waszak[3] m.r.waszak@larc.nasa.gov

[1]Department of Electrical and Computer Engineering
University of Florida, Gainesville, FL 32611-6200

[2]Department of Aerospace Engineering, Mechanics and Engineering
University of Florida, Gainesville, FL 32611-6250

[3]Dynamics and Control Branch, NASA Langley Research Center
MS 132, Hampton, VA 23681-2199

Abstract

Recently, substantial progress has been made towards designing, building and test-flying remotely piloted Micro Air Vehicles (MAVs). This progress in overcoming the aerodynamic obstacles to flight at very small scales has, unfortunately, not been matched by similar progress in ***autonomous*** *MAV flight. Thus, we propose a robust, vision-based horizon detection algorithm as the first step towards autonomous MAVs. In this paper, we first motivate the use of computer vision for the horizon detection task by examining the flight of birds (biological MAVs) and considering other practical factors. We then describe our vision-based horizon detection algorithm, which has been demonstrated at 30Hz with over 99.9% correct horizon identification, over terrain that includes roads, buildings large and small, meadows, wooded areas, and a lake. We conclude with some sample horizon detection results and preview a companion paper [4], where the work discussed here forms the core of a complete autonomous flight stability system.*

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 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 *MAV*s.

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 will play 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 [6-8,13,14]. For example, Figure 1 shows one of our recently developed MAVs (six-inch maximum dimension) as well as a small UAV design (24 inch maximum dimension). The airvehicles in Figure 1 are both examples of *flexible wing* micro air vehicles that have distinct advantages over conventional lifting body designs [8,13,14].

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 array of missions. In order for MAVs to be usable in real-world deployments, however, *MAVs must be able to execute a slate of behaviors with a large degree of autonomy that does not, as of yet, exist.*

The first step in achieving such MAV autonomy is basic stability and control, although this goal 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 flight path of these vehicles.

To deal with these challenges, we propose a *vision-based horizon detection* algorithm as the first step of a complete MAV flight stability and control system. A companion paper [4] extends

Fig. 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.

this basic algorithm by addressing real-time control issues, and discusses our initial implementation of a complete control system, with self-stabilized MAV flight results. In this paper, we first motivate the use of computer vision for the horizon detection task by examining the flight of birds (biological MAVs) and considering other practical factors. We then, describe our vision-based horizon detection algorithm. Finally, we show sample results and offer some concluding thoughts.

2. Background and motivation

2.1 Biological inspiration

While it is certainly true that man-made or mechanical MAVs do not, as of yet, exhibit autonomy, their biological counterparts — namely, birds — do. For aerospace and robotic researchers, the extraordinary capabilities of birds are a source of wonderment and frustration at the same time — wonderment, because birds exhibit a remarkably complex and rich set of behaviors, frustration, because the duplication of those behaviors in man-made systems has thus far been elusive. Given this sad state of affairs, it is natural for engineers and researchers to want to learn from and emulate these biological systems. Here, we do not intend to seek *structural* equivalence between artificial MAV systems and biological neural systems; rather, we seek to learn important *functional* lessons from biology.

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* [15]. Through evolution over time, bird anatomy has adapted to streamline and lighten body weight in order to be able to fly. This has been achieved through the elimination of some bones, the "pneumatization" or hollowing of the remaining ones, and even the total elimination of other unnecessary body parts like the urinary bladder [1]. Yet, when it comes to their nervous system, and especially their eyes, similar reductions in weight have not occurred.

Bird brains are proportionally much larger than those of lizards and are comparable in size to those of rodents; yet, the most striking feature of avian nervous systems are the eyes [1]. Eyes in birds tend to be large in relation to the size of their head; in fact, for some birds, such as the European starling, eyes make up fully 15% of the mass of their head, compared to 1% for humans [12]. Not only is the relative size of avian eyes impressive, but so is their color perception and sharpness of vision. Photoreceptor (cone) densities in the foveae can be as high as 65,000 per square millimeter in bird eyes, compared to 38,000 per square millimeter for humans [15]. And some birds exhibit visual acuity three times that of humans; for example, the American Kestrel can recognize an insect two millimeters in length from a distance as far away as 18 meters [5]. Given the excellent sight of birds, a substantial portion of their brains is devoted to processing visual stimuli.

Birds use their eyes in a number of different ways. During flight, bird eyes assist in obstacle avoidance and flight stability. Predatory birds also rely on their sight to precisely track, target and capture prey. When migrating geese fly in a V-formation, it is believed that they use their vision to stay together as a group and to avoid flying into one another [2]. And while the mechanism for long-distance migration of birds is still not fully understood, various experiments that have been conducted suggest that at least some species of birds appear to use their vision to navigate based on landmarks on the ground, the sun's position in the sky and even the arrangement of stars at night [10]. It is certainly true that some bird species do rely on other sensing, such as smell, hearing (e.g. echolocation) and inner-ear balance as well; however, the relative importance of these senses varies between bird species [15], and, as such, cannot be the unifying principle behind complex bird behaviors, including flight, tracking and navigation.

2.2 Other considerations

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.

2.3 Horizon detection for stability

So what vision-based information do we hope to extract from the on-board camera? At a minimum, a measurement of the MAV's angular orientation is required for basic stability and control. 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) — can be derived from a line corresponding to the horizon as seen from a forward facing camera on the aircraft. The bank angle is given by the inverse tangent of the slope m of the horizon line,

$$\phi = \tan^{-1}(m) . \quad (1)$$

While the pitch angle cannot be exactly calculated from an arbitrary horizon line, it will be closely proportional to the percentage of the image above or below that line. In our development below, the percentage above or below the horizon line will be denoted by the symbols σ_a and σ_b, respectively, where of course,

$$\sigma_a = 1 - \sigma_b . \quad (2)$$

In a rectangular image, the relationship between σ_b and θ is non-linear, may be lightly coupled with ϕ, and can vary as a function of camera distortions. While a calibration table can be used to model the exact relationship between σ_b and θ, our flight tests have shown adequate performance modeling that relationship as linear.

3. Vision-guided horizon detection

3.1 Challenges

Thus, at the heart of our MAV stability and control system is a *vision-based horizon detection algorithm*, that finds the line (i.e. the horizon) with the highest likelihood of separating ground from sky in a given image. Identifying this line between ground and sky is a very straightforward task for humans; autonomous horizon detection, on the other hand presents significant challenges. Viewed from a statistical perspective, color and texture distributions of the ground and sky can vary dramatically based on the time of day, the current weather and the specific terrain over which the MAV is flying at any given moment. Therefore, any robust approach to the problem should not make too many or too restrictive *a priori* assumptions about the appearance of either the sky or the ground. The sky isn't always blue; it can vary in appearance from light blue, to textured patterns of blue, yellow and white (from clouds), to dark gray and anything in between. Ground terrain can be even more variable in appearance. Roads, buildings, meadows, forests, cars, rivers, lakes, sand and snow all introduce substantial variations in ground appearance, and, unlike the sky, ground images can sharply differ from one video frame to the next.

3.2 Initial attempt

Given these realities, it is difficult to envision a robust vision-based horizon fitting algorithm that relies heavily on *a priori* statistical assumptions about the sky and ground. Rather, a robust algorithm will look not at absolute appearance, but instead focus on relative differences in appearance between the ground and the sky. Our first attempt to capture this principle in an algorithm proceeds as follows. First, sharp differences in appearance are detected along the vertical columns of the image array, and y-values that maximally separate the aggregate distribution of pixels on either side (above and below) are identified.[1] Ideally, the identified (x, y) pixels should locate a rough outline of the border between sky and ground in the image, and a line that is fitted to those points should identify the two degrees of freedom of the horizon line. While this approach proves to be relatively successful for uniform ground terrain, the algorithm tends to break down in the presence of significant large objects on the ground, such as buildings or roads. Yet in MAV missions, man-made structures and vehicles are precisely those things in which we are likely to be most interested. Consider, for example, Figure 2, which identifies points of sharpest distinction in green, and the consequent (wrong) estimate of the horizon. Modifications to this initial algorithm, such as outlier rejection and other heuristic methods, do little to alleviate these problems.

Fig. 2: An early attempt at horizon detection. Pixels marked in green indicate the sharpest distinction in color values down vertical columns; the yellow line indicates the consequent incorrect horizon detection.

1. *This approach was originally inspired by the work in [11] on cut/uncut crop-line detection for autonomous harvesting and appears similar in nature to the preliminary autonomous flying work described in [9].*

3.3 Robust algorithm

Given the result in Figure 2 (and other similar failures), it is clear that a different approach is required. We build the development of a second, much more robust algorithm on two basic assumptions: (1) the horizon line will appear as approximately a straight line in the image, something the initial algorithm fails to do; 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. Below, we discuss each of these in turn.

3.4 Optimization criterion

So far, we have talked about the "appearance" of the sky and ground in an image, but have not made explicit what exactly is meant by that. For our purposes, we view color, as defined in RGB space, as the most important measure of appearance. While other measures, such as texture, may also play an important role in distinguishing sky from ground, real-time constraints force a relatively simple representation of appearance.

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,

$$x_i^s = \left[r_i^s \; g_i^s \; b_i^s \right], \; i \in \{1, \ldots, n_s\}, \tag{3}$$

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 i th sky pixel, and let us denote all hypothesized ground pixels as,

$$x_i^g = \left[r_i^g \; g_i^g \; b_i^g \right], \; i \in \{1, \ldots, n_g\}, \tag{4}$$

where r_i^g denotes the red channel value, g_i^g denotes the green channel value and b_i^g denotes the blue channel value of the i th ground pixel.

Now, given this pixel grouping, 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 of variance exhibited by each distribution. Therefore, we propose the following optimization criterion:

$$J_1 = \frac{1}{|\Sigma_s| + |\Sigma_g|} \tag{5}$$

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

$$\Sigma_s = \frac{1}{(n_s - 1)} \sum_{i=1}^{n_s} (x_i^s - \mu_s)(x_i^s - \mu_s)^T \tag{6}$$

$$\Sigma_g = \frac{1}{(n_g - 1)} \sum_{i=1}^{n_g} (x_i^g - \mu_g)(x_i^g - \mu_g)^T \tag{7}$$

where,

$$\mu_s = \frac{1}{n_s} \sum_{i=1}^{n_s} x_i^s, \; \mu_g = \frac{1}{n_g} \sum_{i=1}^{n_g} x_i^g \tag{8}$$

denote the mean vectors for the sky and ground distributions respectively (note the implicit assumption that the sky and ground distributions are roughly Gaussian distributed). In equation (5), $|\cdot|$ denotes the determinant, which measures the volume or variance of each distribution; maximizing J_1 therefore minimizes the intra-class variance of the ground and sky distributions.

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.

For images with sufficient color information, the optimization criterion J_1 works well. In practice, however, the video signal from the MAV can easily lose color information[1] and become nearly black and white. This causes the covariance matrices Σ_s and Σ_g to become ill-conditioned or singular:

$$|\Sigma_s| \approx 0, \; |\Sigma_g| \approx 0. \tag{9}$$

In other words, one or more of the eigenvalues,

$$\lambda_i^s, \; \lambda_i^g, \; i \in \{1, 2, 3\}, \tag{10}$$

for each covariance matrix degenerates to zero, so that the determinant, which is equivalent to the product of the eigenvalues, no longer represents a useful measure of variance, and J_1 becomes ill-defined. For such cases, the augmented optimization criterion J_2 better captures the desired quality of the hypothesized horizon line:

---

1. On-board cameras are required to be very small due to the limited payload capacity of MAVs. As such, cameras used in our flight testing are single-chip CMOS models exhibiting poor color characteristics. Moreover, color information can occasionally be lost through video transmission as well, due to limited transmission range and/or outside electronic interference.

$$J_2 = \frac{1}{|\Sigma_s| + |\Sigma_g| + (\lambda_1^s + \lambda_2^s + \lambda_3^s)^2 + (\lambda_1^g + \lambda_2^g + \lambda_3^g)^2} \quad (11)$$

Note that in J_2, the determinant terms will dominate as long as enough color information is available; however, when such color information is not present, the largest eigenvalues of each covariance matrix will become controlling instead.

3.5 Horizon detection

Given the J_2 optimization criterion in equation (11), which allows us to evaluate any given hypothesized horizon line, we must now find that horizon line which maximizes J_2. As we have stated previously, this boils down to a search in two-dimensional line parameter space.

Let us first define our search space of horizon lines more precisely. Two-dimensional lines can be parameterized in at least two ways that are relevant to our application. The first is the familiar slope-intercept parameterization:

$$y = mx + b \quad (12)$$

where m denotes the slope of the line, and b denotes the y-intercept. Alternatively, we can parameterize a line using the bank angle ϕ and pitch value σ_b previously defined. On the one hand, the (ϕ, σ_b) parameterization is more convenient for clearly bounding the search space,

$$\phi \in [-\pi/2, \pi/2], \ \sigma_b \in [0\%, 100\%], \quad (13)$$

and for stability and control during flight; on the other hand, the (m, b) parameterization is required for rectangular image processing. Therefore, we have developed an efficient transformation between the two parameter spaces (as detailed in [3]); for the remainder of the paper, however, we will refer exclusively to the (ϕ, σ_b) parameterization.

Now, given sufficient computing resources, we could determine the horizon line by simply evaluating J_2 at the following values of ϕ and σ_b:

$$(\phi_i, \sigma_{b,j}) = \left(\frac{i\pi}{n_1} - \frac{\pi}{2}, 100\frac{j}{n_2}\right), \ 0 \le i \le n_1, \ 0 \le j \le n_2 \quad (14)$$

and choosing the maximum value of J_2. As long as n_1 and n_2 are sufficiently large — that is, we are evaluating J_2 at sufficiently fine resolution over ϕ and σ_b — this procedure will generate a good approximation of the actual horizon line. Note, however, that the required $n_1 \times n_2$ evaluations of J_2 may be prohibitively expensive at full video resolution, especially if we wish to run our algorithm at 30 Hz. Gradient-based optimization techniques also do not offer a viable alternative, since (1) the gradient of J_2 over (ϕ, σ_b) is not expressible in closed form, and (2) gradient-based optimization techniques are guaranteed to converge only to local maxima.

We therefore adopt the following two-step approach. First, we evaluate J_2 on coarse, down-sampled images $(X_L \times Y_L)$ with $n_1 = n_2 < 40$ [see equation (14)]. Second, we fine-tune this coarse estimate through a bisection-like search about the initial guess on a higher resolution image $(X_H \times Y_H, \ X_L \ll X_H, \ Y_L \ll Y_H)$. Exact values for n_1, n_2, (X_L, Y_L) and (X_H, Y_H) depend on the available processing power of the current system; sample values are given in Section 4 below. Further details on the search part of the algorithm may be found in [3].

The reader might be wondering at this stage 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 very important reason for not limiting the initial search — namely robustness to single frame errors in horizon estimation. 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. A full, coarse search of line parameter space, on the other hand, guards against cascading failures due to single-frame errors.

3.6 Summary

Here, we summarize the horizon-detection algorithm. Given an image 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_2 on the down-sampled image for line parameters $(\phi_i, \sigma_{b,j})$, where,

$$(\phi_i, \sigma_{b,j}) = \left(\frac{i\pi}{n} - \frac{\pi}{2}, 100\frac{j}{n}\right), \ 0 \le i \le n, \ 0 \le j \le n \quad (15)$$

3. Select (ϕ^*, σ_b^*) such that,

$$J_2\big|_{\phi=\phi^*, \sigma_b=\sigma_b^*} \ge J_2\big|_{\phi=\phi_i, \sigma_b=\sigma_{b,j}}, \ \forall i, j. \quad (16)$$

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

For space reasons, our presentation thus far has omitted some computational efficiency details, that allow the algorithm to be run at full frame rate (30 Hz). For example, as we perform the search in line-parameter space, we do not have to recompute J_2 from scratch for every new set of values (ϕ, σ_b). The statistics of the distributions on each side of the hypothesized horizon line change only incrementally. Computations can likewise be streamlined to be incremental for each new J_2 evaluation when combined with on-line computations of μ_s, μ_g, Σ_s and Σ_g. These and other algorithmic optimizations can be found in [3].

4. Results

Figure 3 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. Normally, the plane is controlled during flight by a remote human pilot through a standard radio link. The goal of this work, is of course, to automate flight control and stability of MAVs.

Figures 4 and 5 illustrate several examples of our current algorithm at work. Additional examples and videos can be found at http://mil.ufl.edu/~nechyba/mav. In each image, the yellow line indicates the algorithm's estimated location of the horizon. Note that the images in Figure 4(a) and Figure 2 are identical; thus, where the initial algorithm failed, the newer algorithm succeeds.

Figure 4(b) plots the optimization criterion J_2 for the image in Figure 4(a) as a function of the bank angle ϕ and pitch percentage σ_b. Note the definitive peak in J_2 at the appropriate horizon estimate. Figure 4(c) plots the distribution of pixels on each side of the estimated horizon; blue crosses correspond to sky pixels in RGB color space, while green circles correspond to ground pixels in RGB space. In Figure 4(d) we show that the current algorithm correctly identifies the horizon, despite the fact that in this image, the uneven horizon violates the straight-line assumption implicit in our current approach. In Figure 4(e) we show that our algorithm is very robust to video interference or noise. This is so, despite drastic differences in color distributions, as depicted in Figure 4(c) and 4(f), between images 4(a) and 4(d), respectively.

Our horizon-detection algorithm has been demonstrated to run at 30 Hz on a 900 MHz 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 $X_H \times Y_H = 160 \times 120$.

5. Conclusions

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. This horizon-detection algorithm lies at the core of a flight-stability system for MAVs that we describe in a companion paper [4], where we address real-time control issues, including extreme attitude detection (i.e. no horizon in the image), confidence measures in the detected horizon estimate, filtering of horizon estimates, and actual self-stabilized flight data.

Horizon detection and flight stability are, of course, only the first step in full vision-based autonomy of Micro Air Vehicles. We are currently exploring additional vision processing of the on-board video to perform detection and recognition of targets of interest on the ground, vision-based navigation through landmark detection, and tracking of other MAVs, as a precursor to MAV formation flying (i.e. swarming). We also believe that in-flight horizon detection and tracking may well allow partial system identification of notoriously difficult-to-characterize micro air vehicles.

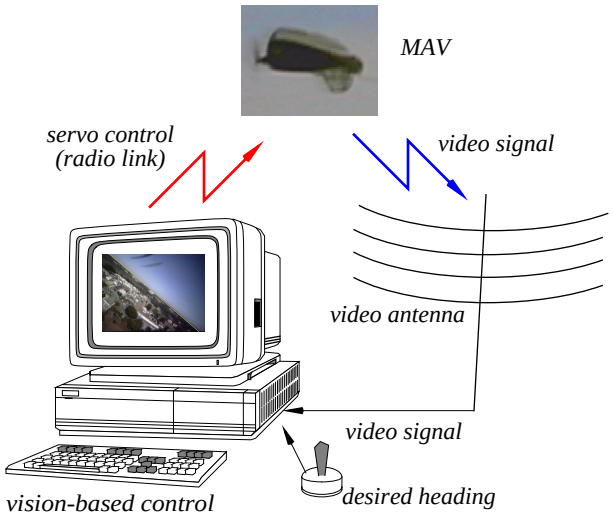

Fig. 3: Experimental setup.

6. References

[1] P. R. Ehrlich, D. S. Dobkin and D. Wheye, "Adaptions for Flight," http://www.stanfordalumni.org/birdsite/text/essays/Adaptions.html, June, 2001.

[2] P. R. Ehrlich, D. S. Dobkin and D. Wheye, "Flying in Vee Formation," http://www.stanfordalumni.org/birdsite/text/essays/Flying_in_Vee.html, June, 2001.

[3] S. M. Ettinger, *Design And Implementation Of Autonomous Vision-Guided Micro Air Vehicles*, M.S. Thesis, Electrical and Computer Engineering, University of Florida, August, 2001.

[4] S. M. Ettinger, M. C. Nechyba, P. G. Ifju and M. Waszak, "Vision-Guided Flight Stability and Autonomy for Micro Air Vehicles," submitted to *Proc. IEEE Int. Conf. on Robotics and Automation*, Washington, D.C., 2002.

[5] R. Fox, S. W. Lehmkule and D. H. Westendorf, "Falcon Visual Acuity,", *Science*, vol. 192, pp. 263-5, 1976.

[6] P. G. Ifju, S. Ettinger, D. A. Jenkins and L. Martinez, "Composite Materials for Micro Air Vehicles," *Proc. of the SAMPE Annual Conf.*, Long Beach CA, May 6-10, 2001.

[7] P. G. Ifju, S. Ettinger, D. A. Jenkins, and L. Martinez, "Composite Materials for Micro Air Vehicles," to appear in *SAMPE Journal*, July 2001.

[8] D. A. Jenkins, P. Ifju, M. Abdulrahim and S. Olipra, "Assessment of Controllability of Micro Air Vehicles," *Proc. Sixteenth Int. Conf. On Unmanned Air Vehicle Systems*, Bristol, United Kingdom, April 2001.

[9] J. Kellogg, et. al., "The NRL Mite Air Vehicle," *Proc. Sixteenth Int. Conf. On Unmanned Air Vehicle Systems*, pp. 25.1-25.14, Bristol, United Kingdom, April 2001.

[10] Northern Prairie Wildlife Research Center, "Migration of Birds: Orientation and Navigation," http://www.npwrc.usgs.gov/resource/othrdata/migration/ori.htm, June, 2001.

[11] M. Ollis and T. Stentz, "Vision-Based Perception for an Automated Harvester," *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, vol. 3, pp. 1838-44, 1997.

[12] G. Ritchison, "Ornithology: Nervous System: Brain and Special Senses II," http://www.biology.eku.edu/RITCHISO/birdbrain2.html, June 2001.

[13] W. Shyy, D. A. Jenkins and R. W. Smith, "Study of Adaptive Shape Airfoils at Low Reynolds Number in Oscillatory Flows," *AIAA Journal*, vol. 35, pp.1545-48, 1997.

[14] 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*, vol. 8, no. 12, 1996.

[15] G. C. Whittow, ed., *Sturkie's Avian Physiology, Fifth Ed.*, Academic Press, San Diego, 2000.

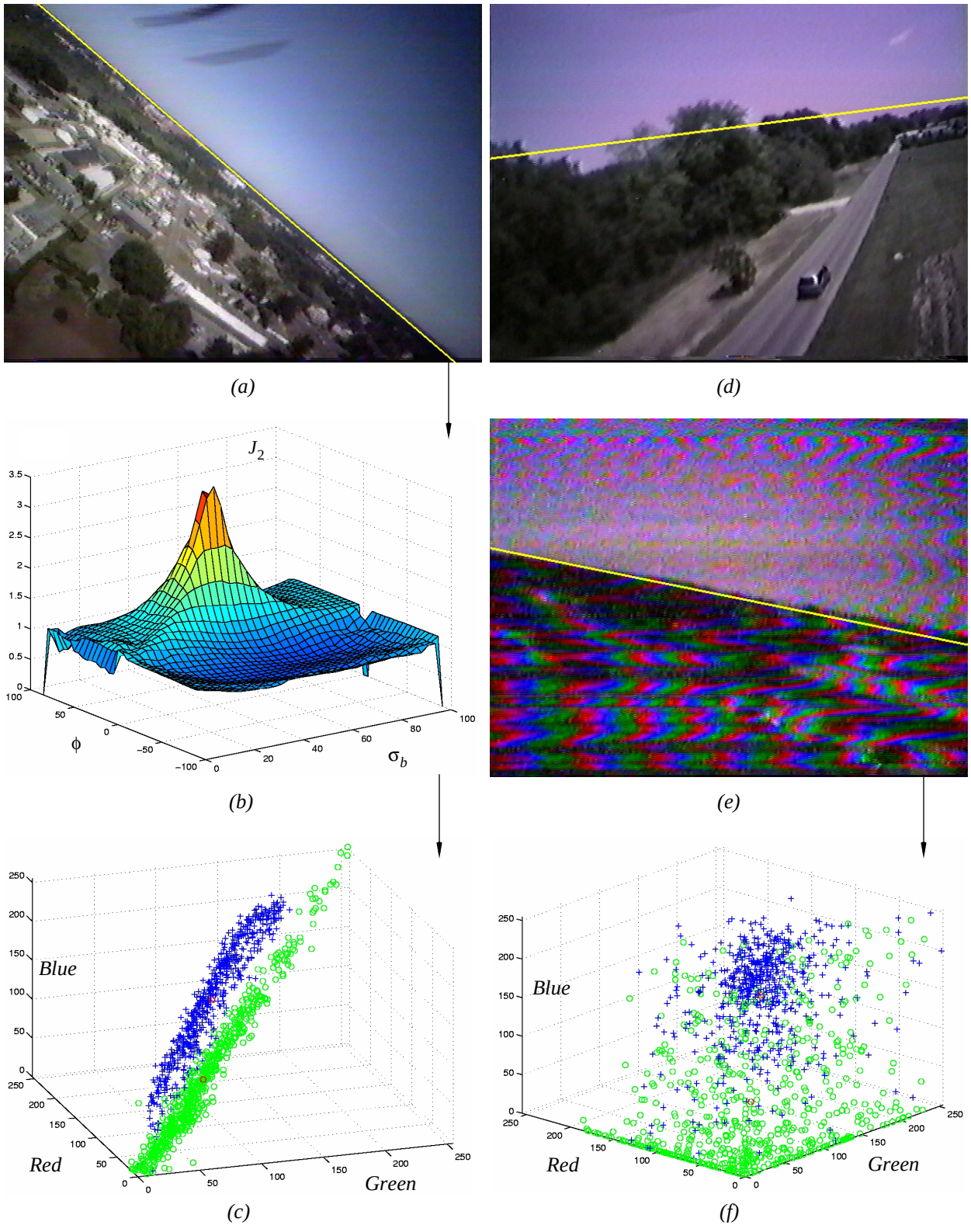

Fig. 4: (a) Horizon detection example; (b) optimization function in line-parameter space for image (a); (c) Distribution of sky (blue crosses) and ground pixels (green circles) in RGB space for image (a); (d) horizon detection with uneven horizon; (e) horizon detection with severe video interference; (f) distribution of sky and ground pixels for image (e).

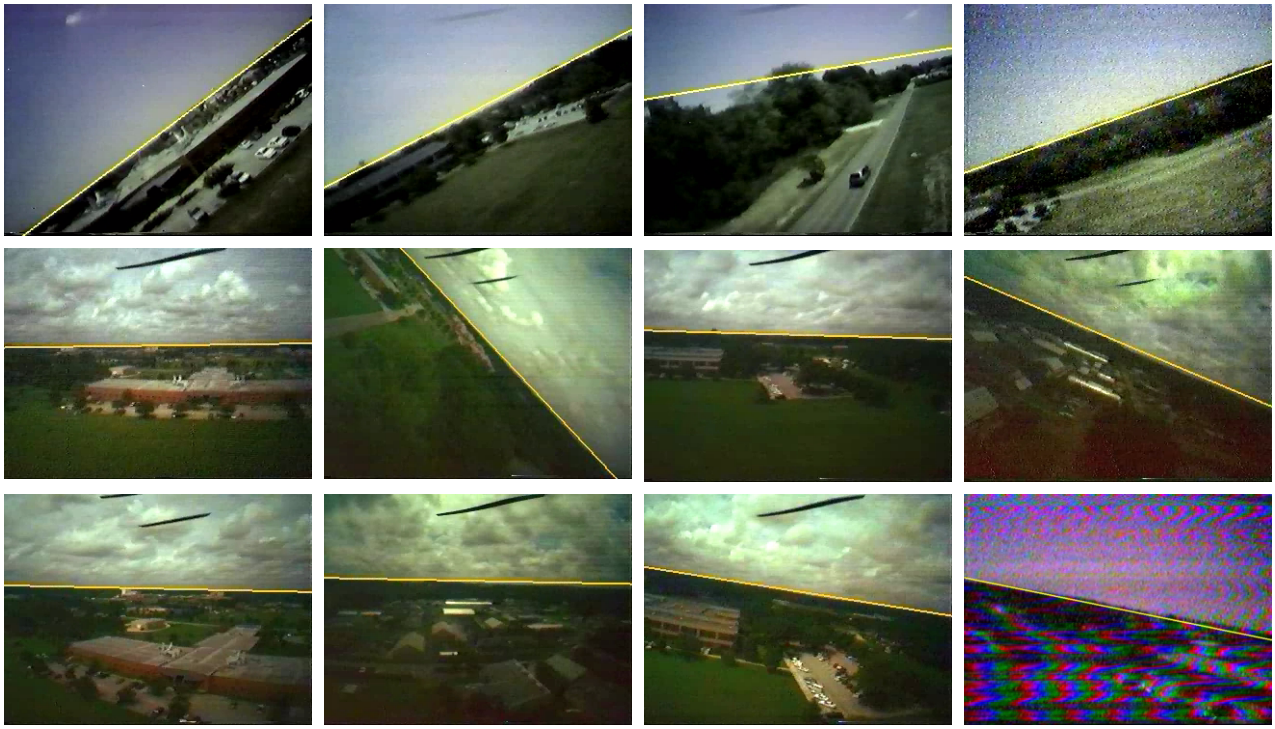

Fig. 5: Various horizon-detection examples under different lighting conditions (sunny and cloudy), and with varying degrees of video transmission noise. For each example, the yellow line indicates the algorithm's horizon estimate.