Rapid Development of Vision-Based Control for MAVs through a Virtual Flight Testbed

Jason W. Grzywna, Ashish Jain, Jason Plew, and M. C. Nechyba
Machine Intelligence Lab
Dept. of Electrical and Computer Engineering
University of Florida, Gainesville, Florida 32611
Email: {grzywna, ashishj, jason, nechyba}@mil.ufl.edu

Abstract— **We seek to develop vision-based autonomy for small-scale aircraft known as Micro Air Vehicles (MAVs). Development of such autonomy presents significant challenges, in no small measure because of the inherent instability of these flight vehicles. Therefore, we propose a virtual flight testbed that seeks to mitigate these challenges by facilitating the rapid development of new vision-based control algorithms that would have been, in its absence, substantially more difficult to transition to successful flight testing. The proposed virtual testbed is a precursor to a more complex Hardware-In-the-Loop (HILS) facility currently being constructed at the University of Florida. These systems allow us to experiment with vision-based algorithms in controlled laboratory settings, thereby minimizing loss-of-vehicle risks associated with actual flight testing. In this paper, we first discuss our testbed system, both virtual and real. Second, we present our vision-based approaches to MAV stabilization, object tracking and autonomous landing. Finally, report experimental flight results for both the virtual testbed as well as for flight tests in the field, and discuss how algorithms developed in the virtual testbed were seamlessly transitioned to real flight testing.**

I. INTRODUCTION

Over the past several years, Unmanned Air Vehicles (UAVs) have begun to take on missions that had previously been reserved exclusively for manned aircraft, as evidenced in part by the much publicized deployment of the Global Hawk and Predator UAVs in the recent Afghan and Iraqi conflicts. While these vehicles demonstrate remarkable advances in UAV technology, their deployment is largely limited to high-altitude surveillance and munitions deployment, due to their size and limited autonomous capabilities. Moreover, while such UAV missions can prevent unnecessary loss of human life, at costs of $70 million and $4.5 million for the Global Hawk and Predator, respectively [1], these UAVs cannot be considered expendable.

Consequently, interest has grown for a different class of small-scale UAVs, known as *Micro Air Vehicles (MAVs)*, that overcome the limitations of larger and more expensive UAVs. At the University of Florida, our on-going research efforts have led to the development of a large number of MAV platforms, ranging in maximum dimension from 5 to 24 inches [2], [3]. [1] Given their small size, weight and cost (approximately $1,000/vehicle), MAVs allow for missions that are not possible for larger UAVs. For example, such small-scale aircraft could safely be deployed at low altitudes in complex urban environments [4], and could be carried and deployed by individual soldiers for remote surveillance and reconnaissance of potentially hostile areas in their path.

While MAVs present great possibilities, they also present great challenges beyond those of larger UAVs. First, even basic flight stability and control present unique 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.

Second, MAVs, due to severe weight restrictions, cannot necessarily make use of the same sensor suite as larger UAVs. While some MAVs recently developed have seen the incorporation of miniature on-board INS and GPS [5], [6], such sensors may not be the best allocation of payload capacity. For many potential MAV missions, vision is the only practical sensor than can achieve required and/or desirable autonomous behaviors, as is the case, for example, for flight in urban environments below roof-top altitudes [7]. 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 can exploit already present sensors, rich in information content, to significantly extend the capabilities of MAVs, without increasing their required payload.

In this paper, we seek to build on our previous success in vision-based flight stability and control [8], [9] on the MAV scale to achieve more complex vision-based autonomous behaviors. Development of such behaviors does, however, present some difficult challenges. First, dedicated flight test areas typically do not exhibit the type of scene diversity likely to be encountered in deployment scenarios. Second, closed-loop, vision-based approaches must operate within a tight computational budget for real-time performance, and require extensive flight testing for robust performance in many different scenarios. Because of the complexity

---

[1] Recent development of bendable wings allows even larger MAVs to fit inside containers with diameters as small as 4 inches.
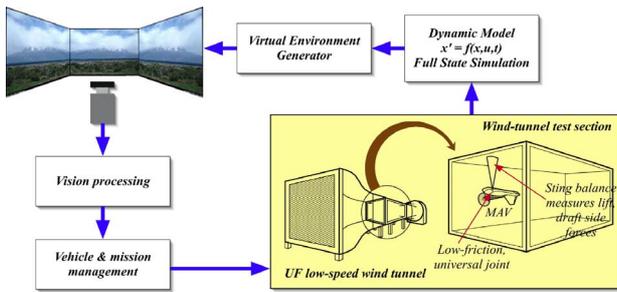
Fig. 1. UF HILS facility currently under construction: concept diagram.

involved, simple errors in software development can often lead to critical failures that result in crashes and loss of the MAV airframe. This in turn introduces substantial delays in the development cycle for intelligent, autonomous MAVs.

To mitigate these problems, we are currently constructing a Hardware-In-the-Loop Simulation (HILS) facility, expected to be completed by the summer of 2004, that will enable testing and debugging of complex vision-based behaviors without risking destruction of the MAV flight vehicles. As conceived and depicted in Figure 1, the HILS facility will simulate the flight of a single MAV through diverse photo-realistic virtual worlds (e.g. urban environments), by measuring and modeling aerodynamic flight characteristics in a wind tunnel in real time. The virtual display will render the correct perspective of the virtual world as the MAV's trajectory is computed from its dynamic model.

Herein, we present an early prototype version of this HILS facility that implements a subset of the capabilities of the full-scale facility, as illustrated in Figure 2. We show how even this simplified virtual testbed facilitates rapid development of new vision-based control algorithms, that would have been, in its absence, substantially more challenging to move to successful flight testing. In this paper, we first discuss our testbed system (Figure 2). Second, we present our vision-based approaches to MAV stabilization, object tracking and autonomous landing. Finally, report experimental flight results for both the virtual testbed as well as for flight tests in the field, and discuss how algorithms developed in the virtual testbed were seamlessly transitioned to real flight testing.

II. VIRTUAL AND FLIGHT TESTBED

Below, we first describe our experimental MAV-based flight testbed, and then discuss its extension to a virtual testbed for rapid development and deployment of vision-based algorithms.

A. Flight Testbed

Figure 2 (top right) illustrates the flight testbed system that has previously been developed for vision-based and more traditional autopilot experiments [5], [10]. Fully equipped with all electronics, GPS, inertial sensors, servos, motors, transmitters and batteries, the electrically powered 24" flight vehicle weighs less than 250g; the pusher-propeller configuration facilitates central-axis housing of

the on-board camera. With current light-weight battery technology, the pictured MAV can fly for as long as 45 minutes on a single charge, at speeds ranging from 20mph to 40mph.

The ground station (bottom center in Figure 2) consists of (1) a 2.4 GHz video patch antenna (not pictured), (2) a video capture device from the Imaging Source (formerly a Sony Video Walkman) for NTSC-to-firewire video conversion, (3) a 12" G4 laptop (1GB/1GHz), (4) a custom-designed Futaba signal generator for converting computer-generated control commands to PWM Futaba-readable signals, (5) and a standard Futaba RC controller. Video is input to the computer in uncompressed YUV format, and then converted to RGB for subsequent processing. The Futaba transmitter, the traditional remote-control mechanism for piloting RC aircraft, is interfaced to the laptop computer through a Keyspan serial-to-USB adapter, and has a pass-through trainer switch that allows commands from another transmitter to be selectively relayed to the aircraft. Our custom-designed Futaba signal generator lets the laptop emulate that other transmitter, and, therefore, allows for instantaneous switching between computer control and human-piloted remote control of the flight vehicle during testing.

If needed, the on-board GPS an inertial sensors are relayed to the ground station via a 115kbs transceiver, as has been done in some of our previous autopilot work [5], [10]. In this paper, however, all control is based on vision (i.e. the camera) exclusively; therefore, this data link is not pictured in Figure 2. As we have previously pointed out, a MAV system that relies only on vision requires significantly less on-board hardware (and weight), and, as such, is much more easily scaled to smaller-sized MAV platforms.

B. Virtual Testbed

In the prototype virtual testbed that we have so far developed as a precursor to the more sophisticated UF HILS facility (Figure 1), the ground station and interface hardware to the "flight vehicle" does not change. That is, the ground station is completely interchangeable between
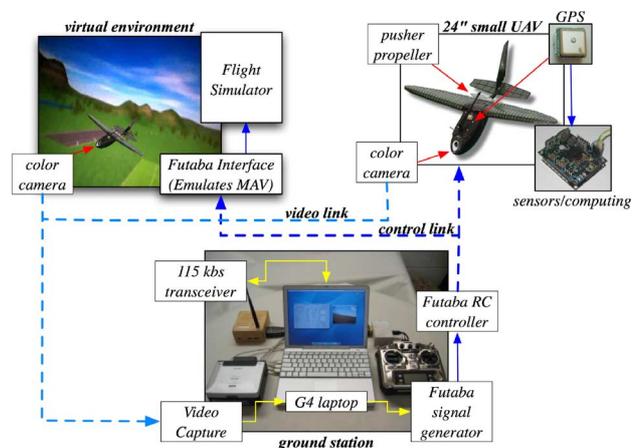

Fig. 2. Prototype system: virtual and flight testbed.

Fig. 3. Some sample virtual scenes: (a) field, trees and mountains, (b) simple urban, (c) urban with features, and (d) complex urban.

the real flight vehicle and the virtual testbed, so that code, controllers, and hardware developed in one environment are immediately transferable to the other.

Our current virtual testbed is based on an off-the-shelf remote control airplane simulation package. The advantages of this software are that (1) it contains a diverse set of scenery as well as vehicle models, including a realistic-physics engine; (2) additional scenery and vehicle models can be defined externally; (3) it supports full collision detection and simulation of partial vehicle damage (e.g. loss of a wing); and, finally, (4) environmental factors such as wind or radio noise, for example, can also be incorporated. Figure 3 illustrates a few examples of the type of scenery supported by the software package; note that the types of scenery available are significantly more diverse than what is easily accessible for real test flights of our MAVs.

The only additional hardware required for the virtual testbed (as opposed to the real flight vehicle) is a small interface board that converts control outputs from the ground station into simulator-specific syntax. As such, the ground station does not distinguish between virtual and real-flight experiments, since the inputs and outputs to it remain the same in both environments.

Given the virtual testbed, virtual flight experiments proceed as follows. First, the flight simulator displays a high-resolution image which reflects the current field-of-view of the simulated aircraft at a particular position, orientation and altitude. Then, a video camera, which is identical to the one mounted on the actual MAV, is fixed in front of the display to record that image. The resulting signal from this video camera is then processed on the ground-station laptop. Next, the extracted information from the vision algorithms being tested is passed to the controller, which generates control commands to maintain flight vehicle stability and user-desired heading (depending, for example, on ground-object tracking). These control commands are digitized and fed into the flight simulator. Finally, the simulator updates the current position, orientation and altitude of the aircraft, and a new image is displayed for image capture and processing.

Note that this system allows us to experiment with vision algorithms in a stable laboratory environment prior to actual flight testing. This means that we can not only develop and debug algorithms without risking loss of the flight vehicle, but we can also experiment with complex 3D environments well before risking collisions of MAVs

with real buildings in field testing. While the scenes in our current prototype system are not as photo-realistic as desirable, even with this limitation, we were able to develop significant vision-based autonomous capabilities in real flight tests without a single crash (Section IV). Moreover, our larger-scale HILS facility will have substantially more computing power for rendering photo-realistic views of complex natural and urban settings.

III. VISION-BASED CONTROL

Below, we develop a purely vision-based approach to flight stability and ground-object tracking for MAVs. We then apply this framework towards vision-based autonomous landing. Results for both virtual and real-flight experiments of these algorithms are then demonstrated in Section IV.

A. Flight Stability

Fundamentally, flight stability and control requires measurement of the MAV's angular orientation. The two degrees of freedom critical for stability – the *bank angle* ϕ and the *pitch angle* θ [2] – can be derived from a line corresponding to the horizon as seen from a forward facing camera on the aircraft. Below, we briefly summarize the horizon-detection algorithm used in our experiments; further details can be found in [8], [9].

For a given hypothesized horizon line dividing the current flight image into a *sky* and a *ground* region, we define the following optimization criterion J:

$$J = (\mu_s - \mu_g)'(\Sigma_s + \Sigma_g)^{-1}(\mu_s - \mu_g) \qquad (1)$$

where μ_s and μ_g denote the mean vectors, and Σ_s and Σ_g denote the covariance matrices in RGB color space of all the pixels in the sky and ground regions, respectively. Since J represents the Mahalanobis distance between the color distributions of the two regions, the true horizon should yield the maximum value of J, as is illustrated for a sample flight image in Figure 4.

Given J, horizon detection proceeds as follows for 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]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.
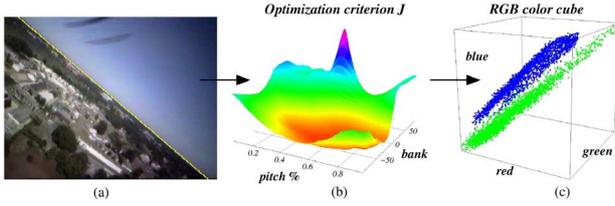
Fig. 4. (a) original image; (b) optimization criterion J as a function of bank angle and pitch percentage; (c) resulting classification of sky and ground pixels in RGB space.

Fig. 5. In object tracking, the search region for the next frame is a function of the object location in the current frame.

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

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

3) Select $(\phi*, \sigma*)$ such that,

$$J\mid_{\phi=\phi*,\sigma=\sigma*} \ge J\mid_{\phi=\phi_i,\sigma=\sigma_j}, \forall_{i,j}$$

4) Perform a bisection search on the high-resolution image to fine-tune the values of $(\phi*, \sigma*)$.

For experiments reported in this paper we use the following parameters: $X_H \times Y_H = 320 \times 240$, $X_L \times Y_L = 20 \times 15$, and $n = 60$. Also, the precise value of the pitch percentage σ that results in level flight (i.e. no change in altitude) is dependent on the trim settings for a particular aircraft. For our experiments, we assume a perfectly aligned forward looking camera (see Figure 2, such that a σ value of 0.5 corresponds to level flight.

B. Object Tracking

Object tracking is a well studied problem in computer vision [11], [12]; our intent here is to use object tracking to allow a user to easily control the flight vehicle's heading (instead of, for example, GPS). We specifically do not perform autonomous target recognition, since we want to be able to dynamically change what ground region the MAV tracks. As such, a user can select which ground region (i.e. object) to track by clicking on the live video with a mouse. This action selects an $M \times M$ region, centered at the (x, y) coordinates of the mouse click, to track. For the experiments reported in Section IV, we set $M = 15$, for video resolutions of $X_H \times Y_H$.

We employ template matching in RGB color space for our object tracking over successive video frames. Our criterion in the sum of square differences (SSD), a widely used correlation technique in stereo vision, structure from motion and egomotion estimation. Our approach differs from some of that work in that we compute the SSD for RGB instead of intensity, since tracking results are much better with full color information than intensity alone. To deal with varying image intensities as environmental factors (e.g. clouds) or the MAV's attitude with respect to the sun changes, we also update the $M \times M$ template to be the matched region for the current frame prior to searching for a new match in subsequent video frames. Furthermore, since ground objects move relatively slowly in the image plane from one frame to the next, due to
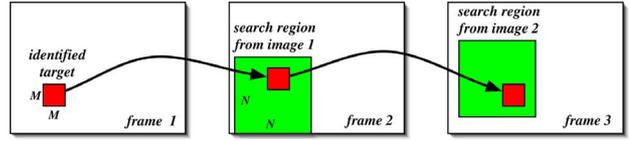
the MAV's altitude above the ground, we constrain the search region for subsequent frames to be in an $N \times N$ neighborhood ($N = 25 \ll X_H, Y_H$) centered around the current ground object location (x, y), as illustrated in Figure 5. This reduces the computational complexity from $O(M^2 X_H X_L)$ to $O(M^2 N^2)$, and allows us to perform both horizon tracking for stabilization and object tracking for heading control in real time (30 frames/sec). In fact, with the G4 Altivec Unit, we are able to dramatically reduce CPU loads to as little as 35% with both vision-processing algorithms running simultaneously.

Below, we briefly summarize the object-tracking algorithm:

1) User selects the image location (x, y) to be tracked for frame t.
2) The template T is set to correspond to the $M \times M$ square centered at (x, y) for frame t.
3) The search region R for frame $t + 1$ is set to the $N \times N$ square centered at (x, y).
4) The location (x, y) of the object for frame $t + 1$ is computed as the minimum SSD between T and the image frame within search region R.
5) Go to step 2.

C. Controller

Here, we describe the controller architecture that takes the information extracted in horizon and object tracking and converts it to control surface commands. Figure 6 shows the overall architecture.

There are two possible inputs to the system from a ground-station user: (1) a joystick that commands a desired bank angle ϕ and pitch percentage σ, and (2) the desired location x_{des} of the ground object to be tracked. In the absence of object tracking, the joystick serves as the primary heading control; with object tracking, the joystick is typically not engaged, such that the trim settings $(\phi, \sigma)_{des} = (0, 0.5)$ are active. The two outputs of the controller are δ_1 and δ_2 corresponding to the differential elevator surfaces controlled by two independent servos.
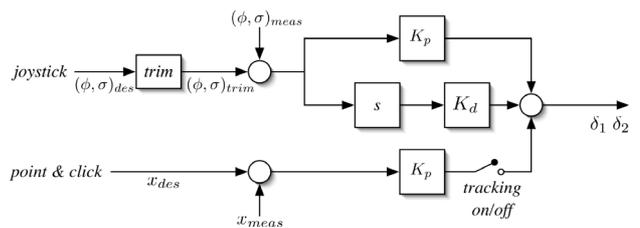

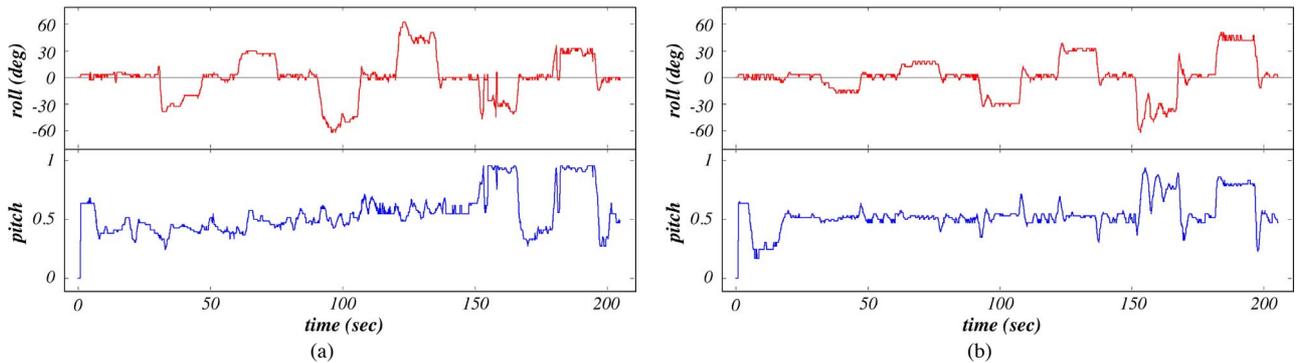Fig. 6. Vision-based control architecture.

Fig. 7. (a) Direct RC-piloted flight, and (b) horizon-stabilized (joystick-controlled) flight. Maneuvers for flight trajectory (b) were executed to mimic flight trajectory (a) as closely as possible.

The bank angle ϕ and pitch percentage σ are treated as independent from one another, and for both parameters we implement a PD (proportional-derivative) controller. The gains K_p and K_d were determined experimentally in the virtual testbed. Because of the differential elevator configuration, the control signals δ_1 and δ_2 will obviously be coupled. For tracking, a P (proportional) controller is used. When engaged (on activation of object tracking), the controller adjusts the bank angle ϕ proportional to the distance between the center of the tracked target and from the center of the current field-of-view. As before, the gain K_p is also determined experimentally in the virtual testbed.

Thus, there are two possible modes of supervised control: (1) direct heading control through the joystick or (2) indirect heading control through object tracking. The first case allows users who are not experienced in flying RC aircraft to stably command the trajectory of the flight vehicle. This is especially critical for MAVs, because it is substantially more difficult to learn direct RC control of MAVs than larger, more stable RC model airplanes. In the second case, commanding trajectories for the MAV is even simpler and reduces to point-and-click targeting on the flight video ground display. Either way, the controller will not permit "unsafe" flight trajectories that may lead to a crash.

IV. Experiments and Results

Below, we describe several experiments for both the virtual and real-flight test beds. First, we contrast direct RC control with horizon-stabilized joystick control, and illustrate object tracking on some sample image sequences. Then, we apply the object tracking framework to develop autonomous landing capabilities, first in the virtual testbed and then in field testing. The principal difference in testing procedures between the virtual and real-flight testbeds occurs at take-off. In the virtual testbed, the aircraft takes off from a simulated runway, while in field testing, our MAVs are hand-launched. After take-off, however, testing is essentially the same for both environments. Initially, the aircraft is under direct RC control from a human pilot until a safe altitude is reached. Once the desired altitude has been attained, the controller is enabled. Throughout our test flights, both virtual and real, throttle control is typically set to a constant level of 80%.

A. Simple stabilization experiment

Here we illustrate simple horizon-based stabilization and contrast it to direct RC control in the virtual testbed; similar experiments have previously been carried out in field testing [8], [9]. Figure 7 illustrates some simple rolling and pitch trials for (a) direct RC-piloted and (b) horizon-stabilized (joystick-controlled) flight trajectories. As can be observed from Figure 7, horizon-stabilized control tends to do a better job of maintaining steady roll and pitch than direct RC flight; this phenomenon has also been observed previously in field testing. Not only does horizon stabilization lead to smoother flights, but no special training is required to command the flight vehicle with horizon stabilization engaged, as is the case for direct RC control of any model aircraft, but especially MAVs.

B. Object tracking

Here, we report results on ground object tracking on some sample flight sequences for both virtual and real-flight videos. Figure 8 illustrates some sample frames that illustrate typical tracking results for (a) a virtual sequence and (b) a real-flight sequence; complete videos are available at `http://mil.ufl.edu/˜number9/ mav_visualization`.

Once we had satisfied ourselves that tracking was sufficiently robust for both virtual and real-flight videos, we proceeded to engage the tracking controller in the virtual
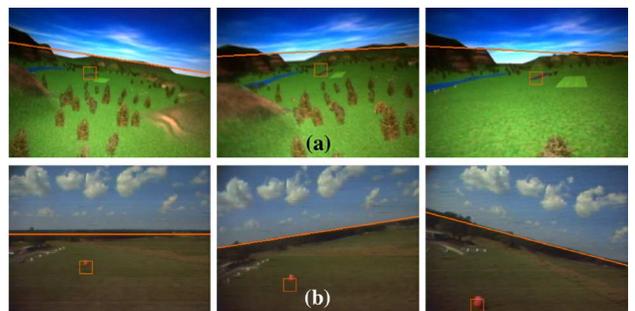

Fig. 8. Object tracking: (a) virtual testbed, and (b) real flight image sequence.
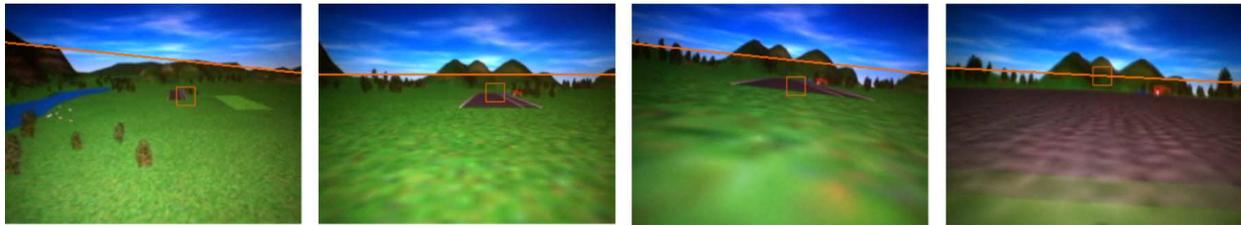
Fig. 9. Autonomous landing in a virtual environment: four sample frames.

testbed, and verified that the aircraft was correctly turning towards the user-selected targets. This led us to formulate *autonomous landing* as a ground object-tracking problem, where the "object" to be tracked is the area where we want the flight vehicle to land. We first developed and verified autonomous landing in the virtual testbed and then, *without any modifications* of the developed code, successfully executed several autonomous landings in real-flight field testing. It is noteworthy that our most experienced MAV pilot commented that these autonomous landings were smoother and suffered from less impact than any of his directly controlled landings. We describe our experiments in autonomous landing further below.

C. Autonomous landing: virtual testbed

An aircraft without a power source is basically a glider as long as roll and pitch stability are maintained. It will land somewhere, but without any heading control yaw drift can make the landing location very unpredictable. However, using our object tracking technique we are able to exercise heading control and execute a predictable landing. Landing at a specified location requires knowledge of the glide slope – that is, the altitude and distance to the landing location. Since we currently do not have access to this data in our virtual testbed, we assume that we can visually approximate these values. Although somewhat crude, this method works well in practice and is repeatable.

We proceed as follows. First, the horizon-stabilized aircraft is oriented so that the runway (or landing site) is within the field of view. The user then selects a location on the runway to be tracked, and the throttle is disengaged. Once tracking is activated, the plane glides downward, adjusting its heading while maintaining level flight. In our virtual testbed, mountains are visible, introducing some error in horizon estimates at low altitudes. As the plane nears ground level during its descent, these errors become increasingly pronounced, causing slight roll and pitch anomalies to occur. Nevertheless, the aircraft continues to glide forward, successfully landing on the runway in repeated trials. Sample frames from one autonomous landing are shown in Figure 9, while the roll, pitch and tracking command are plotted in Figure 10 for that landing. As before, complete videos are available at `http://mil.ufl.edu/~number9/mav_visualization`.

D. Autonomous landing: real-flight experiments

In real-flight testing of autonomous landing, we did not have access to the same ground feature (i.e. a runway) as

in the virtual environment. Our MAVs do not have landing gear and do not typically land on a runway. Instead, they are typically landed in large grass fields. As such, we sought to first identify ground features in our test field that would be robustly trackable. We settled on a gate area near a fence where the ground consisted mostly of sandy dirt, which provided a good contrast to the surrounding field, and, as such, good features for tracking.

Flight testing proceeded as follows. The horizon-stabilized MAV is oriented such that the sandy area is within the field of view. The user then selects a location at the edge of the sandy area to be tracked, and the throttle is disengaged. As in the virtual environment, the MAV glides downward toward the target, adjusting its heading to keep relatively level while maintaining the target centered in the image. When the aircraft approaches ground level, the target being tracked may fall out of view. However, if the target is lost at this point, the plane will still land successfully, since even the maximum allowable turn command generated by the object tracking controller at that point will not cause the plane to roll significantly. Once on the ground, the MAV skids to a halt on its smooth underbelly. In several repeated trials, we landed the MAV within 10 meters of the target location. Our expert MAV pilot, who has logged many hours of remote RC MAV flight, commented that these autonomous landings were smoother and more precise than he could himself achieve. Sample frames from one of those autonomous landings are shown in Figure 11 (along with ground views of the
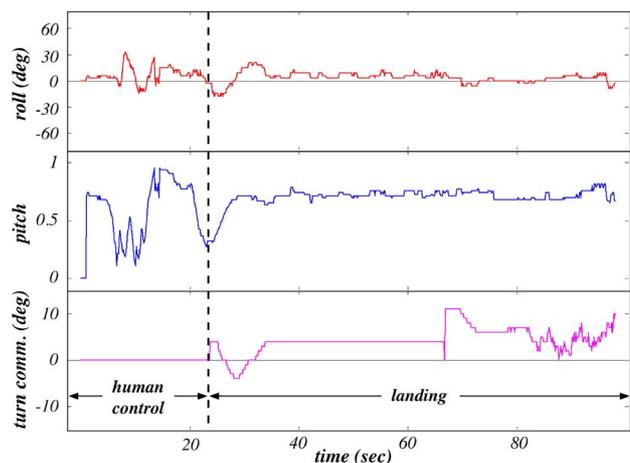

Fig. 10. Roll, pitch and tracking command for virtual autonomous landing in Figure 9.
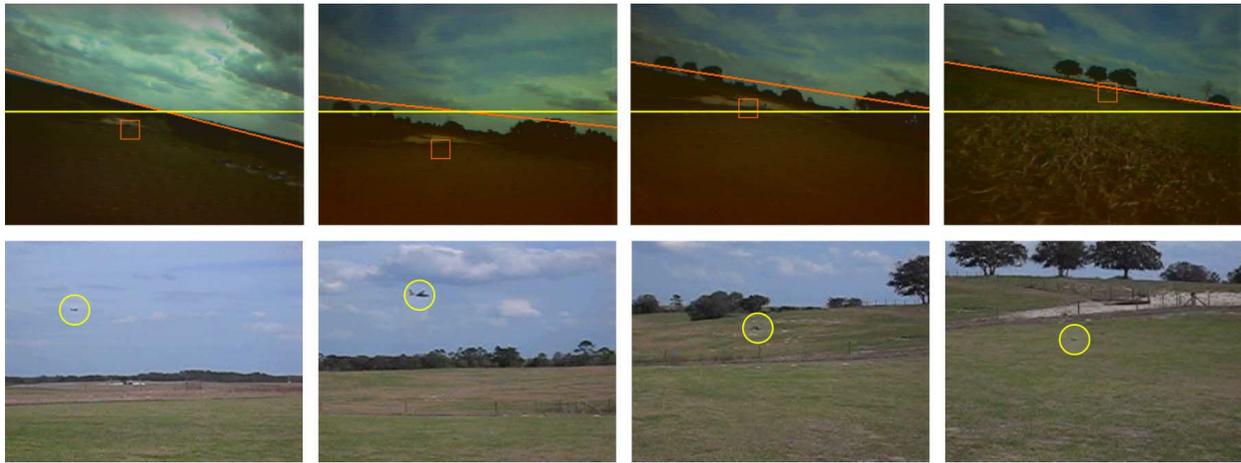
Fig. 11. Real-flight autonomous landing in field testing: four sample frames.

MAV during landing), while the roll, pitch and tracking command are plotted in Figure 12 for that landing. As before, complete videos are available at `http://mil.ufl.edu/~number9/mav_visualization`.

V. Conclusion

Flight testing of MAVs is difficult in general because of the inherent instability of these flight vehicles, but even more so when implementing complex vision-based behaviors. Over the years, we have crashed many planes due to relatively simple errors in coding or algorithmic weaknesses. The prototype virtual testbed facility described in this paper was developed in large measure to deal with these problems, and to investigate potential uses of the full-scale UF HILS facility currently under construction. It is virtually inconceivable that we could have developed object tracking and autonomous landing without any crashes in the absence of the virtual testbed. In the coming months, we plan to extend the use of the virtual testbed facility to more complex vision problems, such as, for example, 3D scene estimation within complex urban environments, a problem which we are now actively investigating.
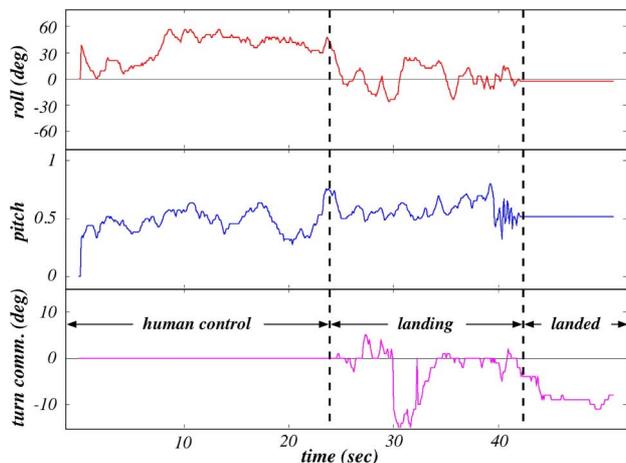

Fig. 12. Roll, pitch and tracking command for real-flight autonomous landing in Figure 11.

References

[1] Global Security.org, "RQ-4A Global Hawk (Tier II+ HAE UAV)," World Wide Web, `http://www.globalsecurity.org/intell/systems/global_hawk.htm`, March 2004.
[2] P. G. Ifju, S. Ettinger, D. A. Jenkins, Y. Lian, W. Shyy and M. R. Waszak, "Flexible-wing-based Micro Air Vehicles," *40th AIAA Aerospace Sciences Meeting*, Reno, NV, AIAA 2002-0705, January 2002.
[3] P. G. Ifju, S. Ettinger, D. A. Jenkins and L. Martinez, "Compostie materials for Micro Air Vehicles," *SAMPE Journal*, vol. 37, No. 4, pp. 7-13, July/August 2001.
[4] J. M. McMichael and Col. M. S. Francis, "Micro Air Vehicles - toward a new dimension in flight," World Wide Web, http://www.darpa.mil/tto/mav/mav_auvsi.html, Dec. 1997.
[5] S. Jung, P. Barnswell, K. Lee, P. G. Ifju, J. W. Grzywna, J. Plew, A. Jain and M. C. Nechyba, "Vision-based Control for a Micro Air Vehicle: Part 1: Testbed," submitted to *AIAA Conf. for Guidance, Navigation, and Control*, August 2004.
[6] J. M. Grasmeyer and M. T. Keennon, "Development of the Black Widow Micro Air Vehicle," *AIAA APATC*, AIAA Paper 2001-0127, Jan 2001.
[7] A. Kurdila, M. C. Nechyba, R. Lind, P. Ifju, W. Dahmen, R. DeVore and R. Sharpley, "Vision-based control of Micro Air Vehicles: progress and problems in estimation," submitted to *IEEE Int. Conf. on Decision and Control*, 2004.
[8] S. Ettinger, M. C. Nechyba, P. G. Ifju and M. Waszak, "Vision-guided flight stability and control for Micro Air Vehicles," *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, vol. 3, pp. 2134-40, 2002.
[9] S. Ettinger, M. C. Nechyba, P. G. Ifju and M. Waszak, "Vision-guided flight stability and control for Micro Air Vehicles," *Advanced Robotics*, vol. 17, no. 7, pp. 617-40, 2003.
[10] R. Causey, J. Kehoe, K. Fitzpatrick, M. Abdulrahim and R. Lind, "Vision-based control for a Micro Air Vehicle: part 2: autopilot," submitted to *AIAA Conf. for Guidance, Navigation, and Control*, August 2004.
[11] Jianbo. Shi and Carlo. Tomasi, "Good features to track," *IEEE Conf. on Computer Vision and Pattern Recognition*, pp.593-600, June 1994.
[12] L. G. Brown, "A survey of image registration techniques," *ACM Computing Surveys*, Vol.24, No.4, pp.325-376, December 1992.