Autonomous landing and ingress of micro-air-vehicles in urban environments based on monocular vision.

Roland Brockers [a], Patrick Bouffard [b], Jeremy Ma [a], Larry Matthies [a], Claire Tomlin [b]

[a] Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA
[b] Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, USA

ABSTRACT

Unmanned micro air vehicles (MAVs) will play an important role in future reconnaissance and search and rescue applications. In order to conduct persistent surveillance and to conserve energy, MAVs need the ability to land, and they need the ability to enter (ingress) buildings and other structures to conduct reconnaissance. To be safe and practical under a wide range of environmental conditions, landing and ingress maneuvers must be autonomous, using real-time, onboard sensor feedback. To address these key behaviors, we present a novel method for vision-based autonomous MAV landing and ingress using a single camera for two urban scenarios: landing on an elevated surface, representative of a rooftop, and ingress through a rectangular opening, representative of a door or window. Real-world scenarios will not include special navigation markers, so we rely on tracking arbitrary scene features; however, we do currently exploit planarity of the scene. Our vision system uses a planar homography decomposition to detect navigation targets and to produce approach waypoints as inputs to the vehicle control algorithm. Scene perception, planning, and control run onboard in real-time; at present we obtain aircraft position knowledge from an external motion capture system, but we expect to replace this in the near future with a fully self-contained, onboard, vision-aided state estimation algorithm. We demonstrate autonomous vision-based landing and ingress target detection with two different quadrotor MAV platforms. To our knowledge, this is the first demonstration of onboard, vision-based autonomous landing and ingress algorithms that do not use special purpose scene markers to identify the destination.

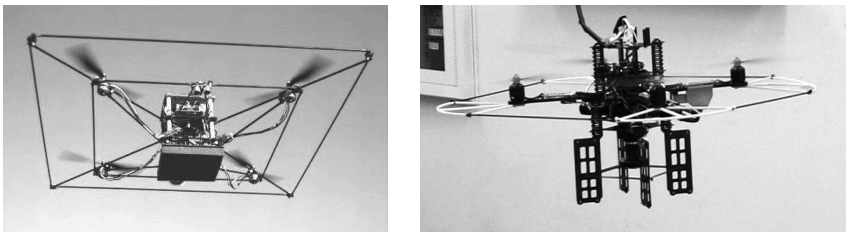
1. INTRODUCTION

New applications in surveillance, rescue and aerial observation demand more and more autonomous behavior to support human operators and guarantee safe operation. Accordingly, research in autonomous navigation for unmanned aerial vehicles (UAVs) has been an intense field of study in the past few years. Research activity has also been boosted by the ready availability of small, relatively inexpensive small quadrotor platforms, some of which are easily adaptable for use as research vehicles or even expressly designed as such. Nevertheless, payload capacity limitations are inescapable, and in turn drive constraints on available energy and computational resources. Power- and CPU-intensive sensor suites are, consequently, difficult to deploy on such systems. For this reason, small passive vision sensors (i.e. cameras) have seen increasing use for navigation, as a single passive sensor can be employed simultaneously for detection and 3D reconstruction. Using only a vision sensor creates new challenges, as a structure from motion approach with a single moving camera can reconstruct 3D information only up to an unknown scale factor, additional information is necessary to remove the extra degree of freedom. Systems that are deployed outdoors can overcome this issue by using GPS data for pose recovery, but this is not an option for systems operating in deep space or indoors.

To cope with this issue, we are developing a system on a small UAV platform using a minimal sensor suite that can operate using only on-board resources to autonomously achieve basic navigation tasks. Here we present, as a first step towards this goal, a navigation approach that visually detects and reconstructs the position of navigation targets, but depends on an external motion capture system* to regain scale and for closed loop control. In a future approach we plan on replacing the motion capture system with an on-board state estimation that fuses IMU and visual information to estimate pose and scale.

To demonstrate the capability of our approach, we chose two important example scenarios for a small UAV platform: autonomous landing on an elevated landing platform (representative of a rooftop), and autonomous building ingress through

Author contact information: {brockers,jeremy.c.ma,lhm}@jpl.nasa.gov, {bouffard,tomlin}@eecs.berkeley.edu
*VICON, http://www.vicon.com



(a) (b)

Figure 1. We evaluated our approach on two different quadrotor platforms: STARMAC (a) and AscTec Pelican (b).

a rectangular opening (representative of windows and doors). Both scenarios are important for applications such as rescue or surveillance missions, and allow us to deploy a multiple homography based approach that uses planar surfaces like walls and roof-tops to reconstruct structure from motion.

This paper is organized as follows: Section 2 discusses related work and how our approach differs. In section 3 we introduce our approach in detail, highlighting the planar homography estimation framework for waypoint generation, our control algorithm for autonomous navigation, and the vision and navigation system interaction. Experimental results are given in Section 4. Section 5 concludes this paper and discusses future work.

2. RELATED WORK

Vision-based methods for aerial vehicles to detect navigation targets have been used in the past to identify e.g. landing sites with fixed markers,[1–3] or terrain features such as flat surface areas.[4–6] To reconstruct the target position with a single moving camera, different structure from motion approaches are deployed. While most methods use external pose information like GPS to calculate accurate target positions,[1,4,7] other applications use the imagery itself to regain the camera motion up to scale before a structure from motion approach is applied.[5] These approaches usually need a separate mechanism to retrieve the unknown scale factor in order to accurately scale the reconstructed 3D structure of the environment with information from an additional sensor (e.g. an IMU or altimeter). A different approach is the use of optical flow to perform closed loop navigation. This is interesting, as the flow field, calculated from the camera images, can be used for egomotion stabilization without additional pose sensors and, at the same time, for basic navigation tasks like obstacle avoidance or landing without an explicit 3D reconstruction.[8,9]

In the work that is described in this paper, we use a multiple homography decomposition for motion estimation and target detection. Homography based approaches have been used for landing site detection in various helicopter and aircraft applications.[7,10] These methods usually perform a RANSAC based homography calculation to identify a planar surface in the image. We extend this approach to a multiple homography estimation method to detect planar elevated landing surfaces above flat ground, and to significantly improve the accuracy of motion estimates from homography decompositions by aligning the detected homographies.

In the following, we give a brief introduction of different vision-aided navigation approaches that are closely related to our work. An example of an approach which visually detects a fixed landing target is the method of Saripalli et al.[1] Their approach to autonomous landing of a UAV uses GPS and vision to navigate to a known landing site marked with a predetermined 'H' pattern. Various control strategies for landing are considered, but the approach depends on the detection of the known pattern by thresholding and segmentation. Inspired by the way flies navigate, Ruffier et al. proposed an optical-flow based method for autonomous landing and navigation,[8] controlling the aircraft by adjusting thrust and pitch to maintain a reference optical flow value. The experimental results are limited to a highly controlled environment with tethered rotorcraft and arbitrarily textured terrain. Landing is considered from the control perspective but no identification of landing sites is performed. Goncalves et al. presented a method for vision-aided navigation for aircraft approach and landing on a planar runway[10] which is comparable to the method presented in this paper. The authors similarly utilize a homography estimation schema to determine aircraft pose relative to the runway. However, in their approach, assumptions are made (e.g. the slowly varying depth to feature) which are valid only for large aircraft at high altitude, preventing extension to micro UAVs.

In terms of autonomous control, Templeton et al.[4] demonstrated a model-predictive flight control system using a monocular camera on an autonomous helicopter. They tracked features in geo-referenced images to estimate 3D structure

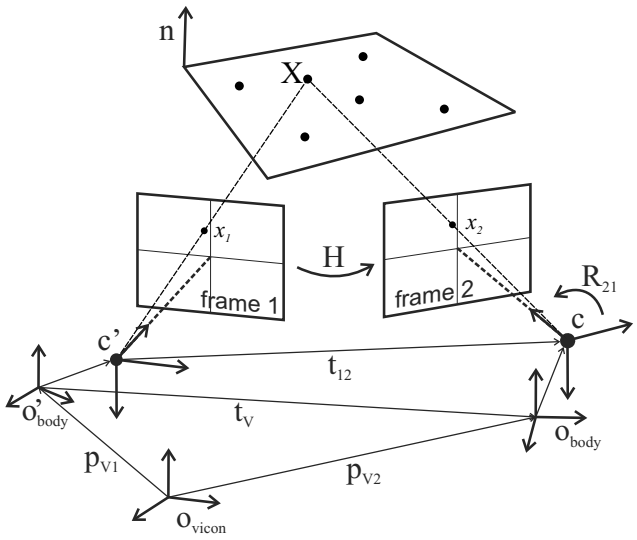


Figure 2. Coordinate frames and homography induced by a planar surface. t_{12} and R_{21} are the camera motion parameters reconstructed from the homography H.

and generate modular elevation and appearance maps, but their approach assumes the availability of GPS information. Herisse et al.[9] addressed the control problem associated with hovering flight and vertical landing control, using an optical flow technique, but assume only a single flat plane and do not consider landing site selection. Recent work by Blosch et al.[11] demonstrates the use of a monocular vision-based SLAM algorithm for accurate determination of the pose of a UAV in GPS denied environments. Using a technique first introduced by Klein and Murray[12] for augmented reality applications, Blosch et al. are able to demonstrate the use of a SLAM algorithm for UAV applications in real-time with sustained tracking made possible via image keyframes. The authors point to future work with an onboard IMU to determine scale, which is an issue not yet resolved in their work. This work, while not exploring landing techniques per se, is a significant step towards the capability of operating UAV systems in all types of environments.

3. APPROACH

Our approach employs a single camera as a vision sensor to detect a navigation target and generate 3D waypoints to the target for autonomous navigation of a quadrotor platform (figure 1). Autonomous flight maneuvers are triggered based on vision system inputs following a three step approach. In a first phase, the vehicle flies a search pattern to detect the navigation target and to estimate an initial waypoint to the target position. Once detected, additional waypoints are gathered by flying the system in a random pattern in the vicinity of the target. These new waypoints are used by the control algorithm to improve the accuracy of the target estimate. After the target estimate converges, the vehicle executes a fixed fly-to-waypoint maneuver to the target. The following sections introduce the vision algorithm to generate target waypoints from point correspondences for our two different application scenarios: autonomous landing on an elevated landing platform, and approaching a rectangular opening for automated ingress, followed by a description of the vehicle control algorithm that uses the vision inputs for navigation.

3.1 Vision based target detection and waypoint generation

Structure from motion with a single camera can be solved, in theory, by estimating the essential or fundamental matrix[13,14] to recover camera motion up to scale, followed by triangulation for 3D reconstruction. However, because of the well-known stability issues of estimating the transfer matrices from unconstrained point correspondences, we limit our approach by assuming a planar surface in view – a ground plane or wall surface. In this case, a homography decomposition can be used for ego motion estimation (figure 2), which is in general more stable than using essential or fundamental matrix estimates. In our approach the homography is derived similar to[15] from matched feature points.

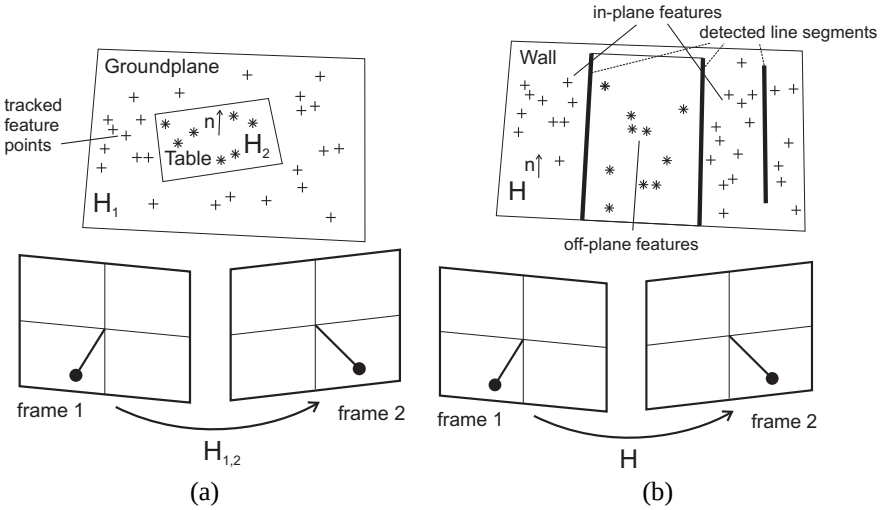


Figure 3. Homography based target detection: (a) Multiple homography detection for landing. (b) Opening detection for ingress scenario.

3.1.1 Feature detection and matching

To establish point correspondences between frames, we implemented a sub-pixel precise version of the STAR feature detector, a center-surround type detector, for detecting blob-like features.[16,17] Feature points are matched among frames using an upright version of SURF[18] for efficiency. To maximize the accuracy of motion estimation and the 3D reconstruction, the most current frame is matched against a reference frame, which is selected by tracking features in subsequent frames and picking the frame in the past for which a minimum number of features can be tracked continuously.

3.1.2 Homography Estimation

A homography H is defined by constraining the general rigid body motion equation by a 3D plane equation. Using homogeneous coordinates, the relationship between points that are located on a planar surface can be expressed as:

$$x_2 = R_{21}(x_1 - t_{12}) \;,\; \tilde{n}_1^T x_1 = 1 \;,\; \tilde{n}_1 = \frac{1}{d} n_1 \tag{1}$$

$$\Rightarrow x_2 = H x_1 \;\text{ with }\; H = R_{21}(1 - t_{12}\tilde{n}_1^T) \tag{2}$$

Starting from unconstraint point correspondences, we use a RANSAC approach[19] to calculate a homography for the maximum set of matched in-plane feature points for a given reprojection error threshold following the method of Longuet-Higgins[15] to parameterize the dominant plane in view. If multiple surfaces are visible (landing scenario) a refinement step further increases the accuracy of the estimated homography by detecting additional homographies and applying a multiple homography alignment algorithm (see section 3.2). The final homography is then decomposed to reconstruct the inverse surface normal $\tilde{n}_1$ and vehicle motion R_{21}, t_{12} up to scale (eq. 2) using the second part of Longuet-Higgins' algorithm.[15] Note, that the distance to the plane surface is coded in the inverse plane normal $\tilde{n}_1$ (eq. 1). In addition, the vision algorithm also uses the distinction between in-plane and off-plane features for detection. We will discuss further details on how this is done in sections 3.2 and 3.3.

3.1.3 Waypoint Generation

Once a navigation target is detected in the image, the reconstructed vehicle motion parameters and the plane normal of the detected surface are used to calculate a 3D waypoint which is located on the target plane. To recover scale, VICON pose information is used to scale the estimated camera translation t_{12} with the vehicle translation t_V measured by the VICON system:

$$s = \frac{|c - c'|}{|t_{12}|} = \frac{|t_V - R_{V1} t_c + R_{V2} t_c|}{1} \tag{3}$$

Since the vehicle body frame o_{body} (figure 2) is not centered on the camera projection center c, the vehicle orientation R_{V1}, R_{V2} in o_{vicon} and the vehicle-camera calibration vector t_c in o_{body} are needed to account for rotational motion.

To calculate the 3D position of a waypoint X_c located on the detected planar surface, the ray defined by the camera center and a specific normalized image coordinate is intersected with the detected surface (as seen from the new camera position) to obtain a 3D point in camera coordinates:

$$\vec{r} = (x_2, y_2, 1)^T, \;\; \tilde{n}_2 = R_{21} \frac{1}{1 - t_{12}^T \cdot \tilde{n}_1} \tilde{n}_1 \tag{4}$$

$$X_c = \beta \vec{r}, \;\; s^{-1} \tilde{n}_2^T X_c = 1 \Rightarrow \beta = s(\tilde{n}_2^T \vec{r})^{-1} \Rightarrow X_c = \frac{s \cdot \vec{r}}{n_{2x}x_2 + n_{2y}y_2 + n_{2z}} \tag{5}$$

This point is finally transferred into VICON coordinates that are used to control the vehicle, by a transformation with the vehicle-camera calibration parameters R_c and t_c and the vehicle position and orientation within the VICON frame P_{V1} and R_{V1}:

$$X_V = P_{V1} + R_{V1} t_c + R_{V1} R_c X_c \tag{6}$$

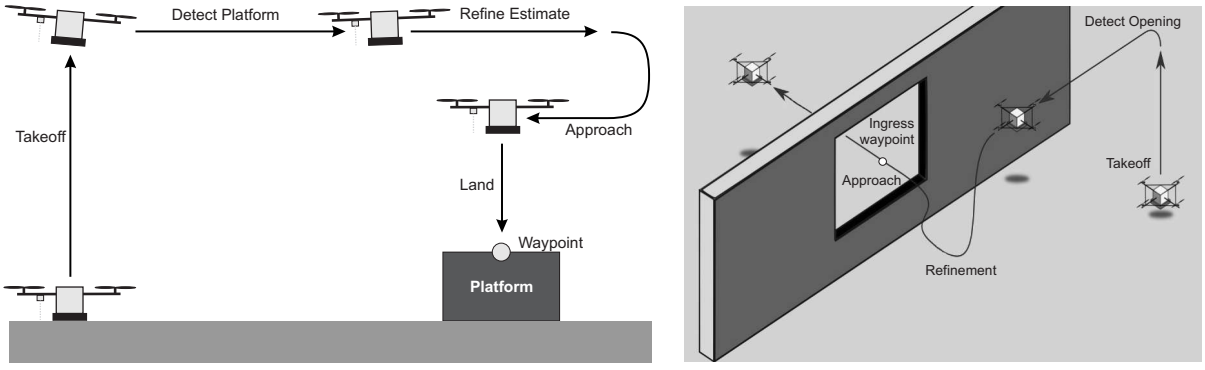
In the following, we apply the homography based reconstruction to two different application scenarios.

3.2 Landing

In the autonomous landing scenario (figure 4a), the quadrotor is equipped with a downward looking camera (cp. figure 3a) to detect an elevated, planar landing site. In the detection phase, the vehicle initially flies a predefined search pattern, until the landing area comes into view and is detected by the vision system. For each frame in which the vision algorithm detects the landing surface, a waypoint is generated for the center of the landing spot. These individual waypoints are collected by the vehicle control algorithm, which initiates an automated landing once the mean of the waypoint coordinates stabilizes. In order to detect a landing platform it is assumed that the landing surface is surrounded by a dominant plane – the floor. A first homography estimation is used to reconstruct the plane parameters and to extract in-plane features that are located on the ground. For all off-plane features, a second homography estimation is applied to find the elevated landing surface. If enough features on a second surface are found, the landing platform is detected (figure 5). Theoretically both homographies could be used to calculate camera motion up to scale. However, in our approach we first align the two homographies using the epipolar constraint as described in[20] before applying a homography decomposition algorithm to increase the accuracy of the reconstruction results for ego motion and plane normals. Finally, an aiming point in the image is calculated which is defined by the center of all detected feature points on the landing target, and a 3D waypoint is calculated as described in section 3.1.3.

3.3 Ingress

The homography estimation approach can also be applied to the problem of ingress (figure 4b) to detect wall openings. In this application, the vehicle is equipped with a forward looking camera (figure 3b) to perform autonomous building ingress. Similar to the landing scenario, we assume a dominant plane in view – the wall surface – and track outliers to detect an



(a) (b)

Figure 4. After takeoff, the autonomous landing (a) and automated ingress (b) algorithm proceed in three phases: *Detection*, *Refinement*, and *Approach*.

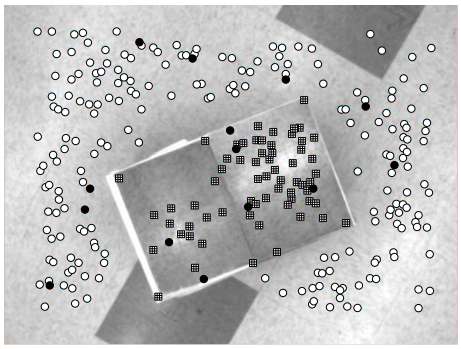
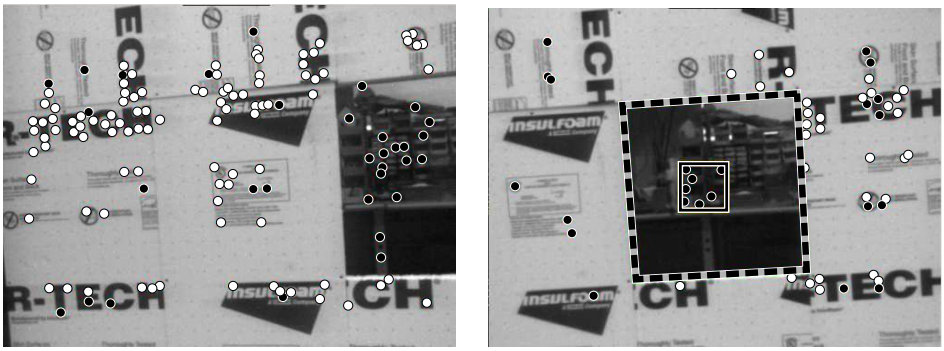


Figure 5. Landing surface detection: The platform is detected if sufficient matched features points are observed on the elevated platform (squares). White dots are features located on the ground plane, black dots are outliers.

opening. Once detected, waypoints for the ingress target are generated as explained before, now using the wall surface to position the waypoint in the center of the opening.

While off-plane features can serve as a good indicator of a wall opening, the detection of the opening target is in particular more challenging due to (1) false positives for outliers that are not always isolated to opening areas and may appear on the planar wall surface (figure 6a), and (2) clusters of outliers that do not clearly define the true size of the wall opening (figure 6b). To remedy these issues, we adopt an approach to ingress that detects square or rectangular openings since these types of openings are typical for urban/indoor environments. To detect a rectangular opening area, we search for the bounding box that contains the most number of outliers without including any inliers. This is done in a three step process. First, line segments are extracted from the image by a Hough transform, filtering on predominantly vertical and horizontal line segments (figure 7a and b). Second, a 2D histogram of in- and outliers is generated and partitioned according to vertical and horizontal segments detected in the previous step (figure 7c). And third, the partition which contains the largest number of outliers above a given threshold and without including any inliers is selected as an opening (figure 7d).

Following the identification of the largest bounding box of outliers (with a prior cue seeded by line segments from the Hough Transform), the ingress point is identified as the center point of the box opening on the wall surface and a 3D waypoint is generated and passed to the vehicle controller.



(a) (b)

Figure 6. (a) Detected in-plane (white) and off-plane features (black). Note that some false positives are located on the planar surface, rather than being completely isolated in the opening. (b) Outliers are only detected on out-of-plane regions that are descriptive in features. Featureless off-plane areas of the image may not always be detected as such due to the lack of features. This limits the detectable size of the actual opening (dashed box) to something much smaller (solid box).

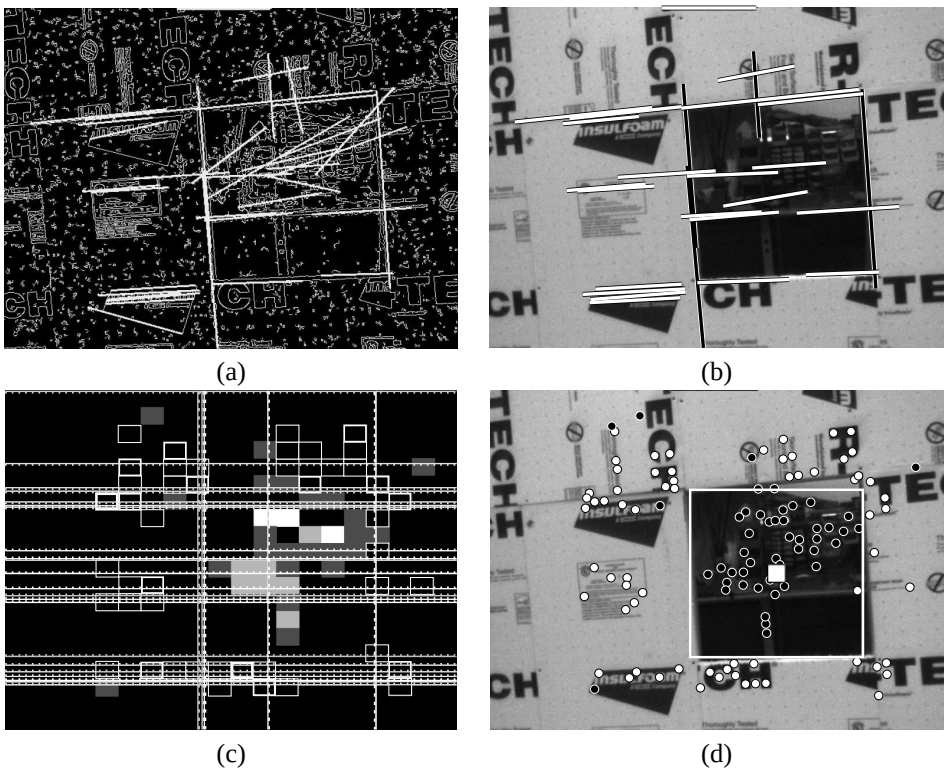


Figure 7. (a) Raw output image from Hough Transform. (b) Filtered output with vertical and horizontal edges. (c) 2D histogram image of inliers (outlined boxes) and outliers (solid boxes), partitioned using the detected vertical and horizontal line segments. (d) The detected opening (white rectangle, centroid marked by white square), and features (inliers white, outliers black), superimposed on the captured image.

3.4 Navigation and Control

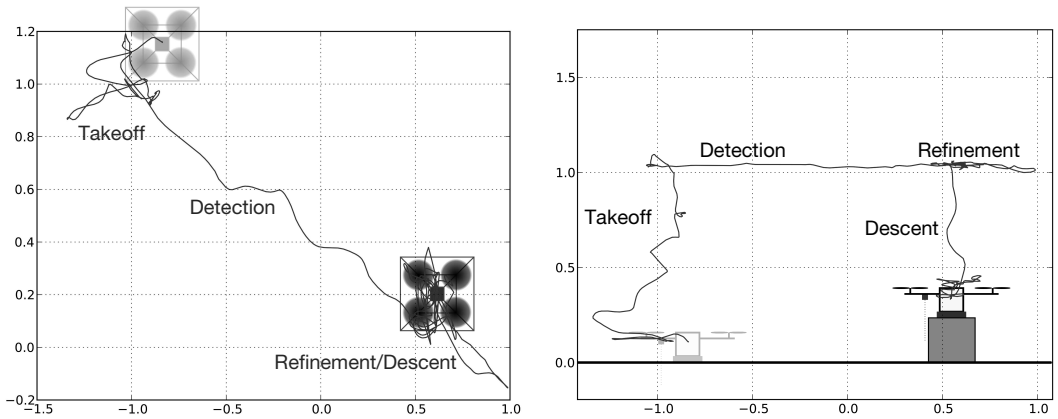
Our control scheme is designed to operate in a hierarchical framework. There are three distinct levels to the hierarchy which are termed the *autonomy loop*, the *outer loop* and the *inner loop*. The autonomy loop is responsible for the highest-level behaviors and is the most abstract, whereas the outer and inner loop are responsible for low-level behaviors like direct flight to waypoints and vehicle stabilization. In the proposed scheme, the vision system becomes an input to the high-level autonomy loop. Note, that this implies that this approach is adaptable to any vehicle for which the level abstractions hold and not only to the specific vehicle type – a quadrotor UAV – considered in our experiments.

From the perspective of the autonomy loop, the vision subsystem is a sensory input, providing "measurements" of the true position of a navigation target. The navigation system operates in three sequential phases with three distinct behaviors at the autonomy level, in order to first detect the target (*detection* phase), then determine an accurate estimate of the target location (*refinement* phase), and finally to perform an autonomous maneuver based on the final target estimate (*approach* phase) (cp. figure 4).

3.4.1 Autonomous Landing Control

Figure 8 illustrates the phases of the autonomous landing scenario. The objective is to detect and localize an unknown, unmarked elevated landing platform, and subsequently land upon it.

Platform Detection In the *detection* phase, the vehicle follows a predefined path, while the vision subsystem gathers imagery for an initial detection of the landing platform. Once the initial detection occurs and a first landing spot waypoint $X_{V,1}$ is generated, the control system transitions to the *refinement* phase.



(a) (b)

Figure 8. (a) Top-down view of an actual vehicle trajectory through the phases. (b) Side view of an actual vehicle trajectory through the phases.

Refinement of Estimate The waypoint measurements are subject to noise and outliers, which makes it necessary to gather additional measurements for a refined estimate. This happens in the *refinement* phase, where additional 3D waypoints are collected into a sequence of n waypoints $\left\{\mathrm{X_{V,i}}\right\}_{i=1}^{n}$. To produce additional waypoints, the vehicle has to move to collect imagery from different vantage points in the vicinity of the current estimate. This is accomplished by directing the vehicle outer loop control to position the vehicle at points $\tilde{\mathrm{X}}_{\mathrm{V,i}}$ in the neighborhood above the first waypoint $\mathrm{X_{V,1}}$. To address the uncertainty in the estimate of the platform location at any given time, these positions are drawn from a bivariate normal distribution $\mathcal{N}(\mu,\Sigma)$, where mean μ and covariance Σ are determined from the sample mean and covariance of the waypoints collected up to that point. Positions that would take the vehicle more than some arbitrary cutoff distance away from the estimate are discarded.

As each new measurement is obtained, sample statistics of the set of measurements are computed. In particular, the waypoint estimate after the n-th measurement has been obtained is the sample mean of the estimates $\hat{\mathrm{X}}_{\mathrm{i}} = \frac{1}{n}\sum_{i=1}^{n}\mathrm{X_{V,i}}$,

When the stopping criteria for the refinement phase,

$$n \geq n_{\min}, \tag{7}$$

$$\left|\begin{bmatrix} 1 & 1 & 0 \end{bmatrix}(\hat{\mathrm{X}}_{\mathrm{n}} - \hat{\mathrm{X}}_{\mathrm{n-1}})\right| \leq \varepsilon_{\mathrm{horz}}, \tag{8}$$

$$\left|\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}(\hat{\mathrm{X}}_{\mathrm{n}} - \hat{\mathrm{X}}_{\mathrm{n-1}})\right| \leq \varepsilon_{\mathrm{vert}}, \tag{9}$$

have been met, the estimate is considered to be 'stable', and $\hat{\mathrm{X}}_{\mathrm{n}}$ is taken as the final target waypoint. The controller then transitions to the *approach* phase.

Descent and Landing In the *approach* phase, the vehicle is first sent to the point at the current altitude immediately above $\hat{\mathrm{X}}_{\mathrm{n}}$. Then the vehicle descends at a fixed rate v_{des}, as long as the radial offset $e_r(t)$ between the current position and $\hat{\mathrm{X}}_{\mathrm{n}}$ is within a tolerance. The altitude setpoint is then updated accordingly, $x_3(t+1) = x_3(t) + v_{\mathrm{des}}dt$ and the updated setpoint is passed to the low-level altitude control loop. Thus, the vehicle only descends when it is within a prescribed volume. Finally, once the vertical offset reaches a threshold altitude, the altitude setpoint is rapidly decreased before cutting off power to the motors to complete the landing. During this final phase the onboard accelerometer readings are employed to detect touch down and cut off the motor power.

3.4.2 Autonomous Ingress Control

The autonomous landing scheme described above can be easily adapted for autonomous ingress – ingress basically follows the same high-level control phases. Due to the forward looking camera, flight maneuvers during *detection* and *refinement* are modified to reflect the change in viewing direction.

At the present time we have performed limited approaches to an opening for ingress, performing maneuvers for *detection* and *refinement*, but due to limitations in the available flying space in the current laboratory setup, we are not able to perform the actual fly-through. We plan to remedy this in the near future, with a larger flying area suitable for a full ingress.

Several control options for maneuvering through the opening are feasible: A heuristic, similar to that used for landing could prove adequate, depending on the control performance achievable under the aerodynamic disturbances present during the ingress. However provably-safe schemes could be considered as well, wherein the controller design would be supported by a reachability-based analysis.[21]

4. EXPERIMENTAL RESULTS

4.1 Hardware Testbed

Our approach has been implemented on quadrotor platforms flying indoors in laboratory environments, using a VICON motion capture system as the primary means of state estimation. We have employed two different vehicle architectures (figure 1), porting the approach from one (the STARMAC[22] quadrotor) vehicle to the next (the AscTec Pelican UAV*). While the two vehicles differ in implementation details, they are similar in that they are both equipped with high-level computing (in the case of STARMAC, a dual-core 2.16 GHz Pentium PC-104 board; for the Pelican, a 1.6 GHz Intel Atom board). The high-level computer executes a multiprocess and multi threaded system, using Robot Operating System (ROS)[23] for system management, interprocess communication, and communication with the operator ground station. Modules were written variously in Python and C++. Note, that our formulation does not require a minimum frame rate for good results – the accuracy depends solely on the spatial separation of the camera location between image acquisition, not on the rate at which images are captured.

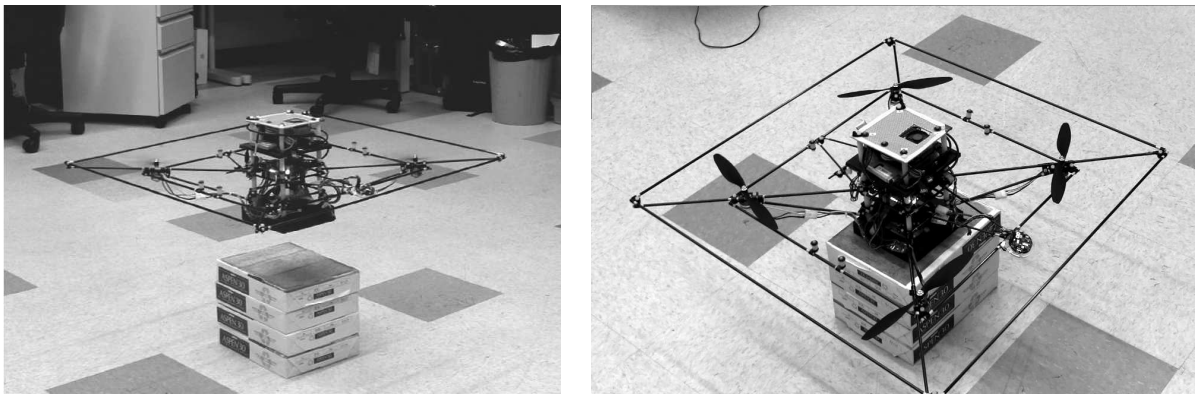
Operator control of the vehicle is performed via a 'ground station' laptop equipped with a joystick, which has bidirectional communication with the onboard high level computer via a WiFi link. Live raw video from the onboard camera as well as a diagnostic image output from the vision algorithm (as shown in figs. 5 and 7) can be transmitted to the ground station to allow for rapid debugging. Data may be logged to on-board mass storage for later transfer to a desktop PC where the algorithms can be tested and refined on the 'played back' data.

As mentioned above, the autonomy layer implements the 3-phase behavior described above by commanding autosequence destinations to the outer loop of the control architecture. In the outer loop, a linear controller stably regulates the vehicle's lateral (horizontal) position of the vehicle towards a designated setpoint. Vehicle yaw angle is maintained constant, and altitude is also regulated with a linear controller based on state measurements from the motion capture system.

The camera used is a Point Grey Research Firefly MV USB monochrome camera equipped, with a 4 mm micro lens. Capture was set in streaming mode at full (640×480) resolution. For the landing scenario, various platforms were tested. Note that the vision algorithm is not dependent on special features or artificial texturing of the elevated platform. The landing platform is relatively small in comparison to the 'footprint' of the quadrotor's landing gear (approx. 15×15cm), which requires precise waypoint estimation. The ingress scenario employed a mockup wall built from Styrofoam elements with one square opening of 60x60cm (cp. figure 6 and 7).

We tested our algorithm intensively running on-board the two different quadrotor platforms. Autonomous landing was performed successfully on both, the STARMAC and the Asctec Pelican quadrotor. For the ingress experiment we used the STARMAC platform to gather imagery for the waypoint calculation during the *detection* and *refinement* phase.

*Ascending Technologies GmbH, http://www.asctec.de/



(a) (b)

Figure 9. View of the STARMAC quadrotor (a) just prior to landing and (b) after landing in one of the experiments.

4.2 Landing

We tested the landing algorithm on the above described setup with the landing platform placed at an arbitrary location. The sequence of events during an experimental run was as follows. The human operator first commands the vehicle to enter a position-holding mode, and commands a take-off maneuver, bringing the vehicle to a hover above the initial location at some suitable altitude. Upon command of the operator's 'go' signal, a predetermined flight path is executed (platform *detection* phase). After *detection* and *refinement* of the platform position, the vehicle lands autonomously on the platform. After the automatic shut-down of the motors, the experimental run is considered complete. All operations after the operators initiates the *detection* phase are performed completely autonomous.

In our experiments, we used three different sizes for the landing platform, assembled out of printer paper reams: a) double wide, 2 reams high (43.2×27.9×10.9cm); b) double wide, 4 reams high (43.2×27.9×21.9cm); and c) single wide, 4 reams high (21.6×27.9×21.9cm). In all three scenarios, both quadrotor systems were able to successfully land (see figure 9). The vision algorithm was very effective in detecting and localizing the landing platform. As expected, the platform location was estimated most accurately in the lateral direction, while the vertical component of the estimate was poorest. This behavior is acceptable, as a highly accurate lateral position is necessary to safely land the vehicle on the platform, whereas a more uncertain height estimate is compensated within the final descent step. Figure 10 shows the measured raw lateral waypoint coordinates that are generated from the vision system, and the refined waypoint estimate during *refinement*. Figure 11 compares the estimated height with the true height of the landing spot. The refined estimate and the estimated platform height converge over time, as new measurements arrive from the vision algorithm. Target waypoints where generated at a maximum rate of approximately 5 fps on the STARMAC quadrotor and at approximately 2 fps on the Pelican system.

4.3 Ingress

The initial start up of the system is similar to the landing experiment. The vehicle is brought up to altitude and the *detection* phase is initiated. To detect an opening, the vehicle flies a trajectory along a wall surface, with the camera facing directly to the wall. Once detected, the vehicle performs a predefined horizontal flight maneuver parallel to the wall surface in front of the opening to execute the *refinement* phase. For the reasons described above, we did not perform an *approach* phase.

To confirm the quality of our reconstruction results, we added one measurement for the ingress experiment. In addition to the calculation of a navigation waypoint in the center of the ingress target, we also calculate the lateral position of the left and right border of the window opening to measure the opening width. Figure 12 shows the estimated opening width (mean width) during *refinement* phase for a typical test run. Note that larger deviations are outliers caused by false detections of the window borders. The estimated opening width at convergence is very close to the true opening width of 60cm. This confirms the accuracy of the vision output for the ingress scenario.

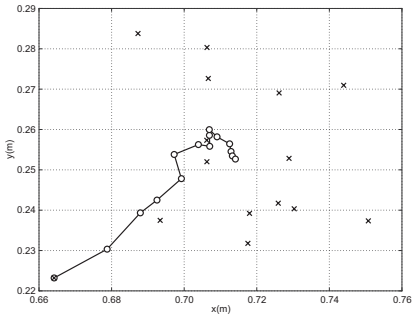


Figure 10. Example of landing point estimate refinement. Crosses are raw measurements, Circles show the evolution of the waypoint estimate as each new measurement is incorporated.

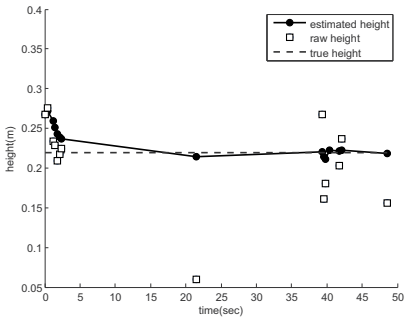


Figure 11. Landing spot height estimation.

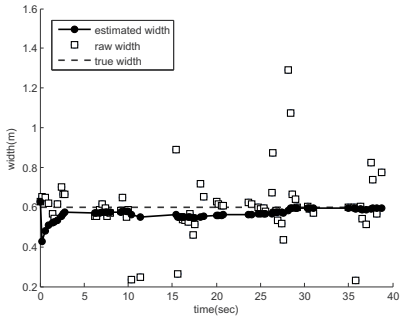


Figure 12. Opening width estimation for the ingress target.

5. CONCLUSIONS AND FUTURE WORK

This paper demonstrates a visual navigation approach for autonomous landing and ingress of a micro air vehicle, which uses images from a monocular camera to detect navigation targets and generate 3D navigation waypoints based on a multiple homography approach with scale recovery from a motion capture system. The system is able to generate highly accurate target waypoints as shown in our experiments and illustrated by the very accurate reconstruction of the landing platform height and the ingress opening width. Using a three stage control scheme, we are able to autonomously detect, approach and land on an elevated landing surface that is only slightly larger than the footprint of our aerial vehicles, and gather navigation target waypoints for building ingress. All algorithms run on-board the vehicles.

This is, to our knowledge, the first vision-based, experimentally tested algorithm for small UAVs, using a single monocular camera and requiring no artificial labeling or external computation, enabling autonomous landing on an elevated landing platform and ingress into a building.

The current approach uses pose estimates based on VICON motion capture data to regain scale for our structure from motion approach, and to control the vehicle. We plan on substituting the VICON input with vision-based on-board state estimation in a future extension. Ultimately, we aim to navigate a small aerial vehicle with on-board processing using only a minimal sensor suite of a single camera with an additional sensor to regain scale (e.g. an on-board IMU), in a GPS denied environment. A challenge for this type of vehicle when maneuvering close to structure or in wind gusts is the incorporation of a dynamic controller to consider changes in vehicle dynamics caused by aerial disturbances. In particular, we want to explore this aspect of the problem for performing autonomous ingress maneuvers through small openings. With the complexity of the unsteady airflow in this flight regime, effective models will likely require additional detail beyond the simple double integrator type models which are quite effective when clear of such aerodynamic disturbances.

Acknowledgments

This work was funded by the Army Research Laboratory under the Micro Autonomous Systems & Technology Collaborative Technology Alliance program (MAST-CTA).

REFERENCES

[1] Saripalli, S., Montgomery, J. F., and Sukhatme, G. S., "Vision-based autonomous landing of an unmanned aerial vehicle," in [*Proc. Int. Conf. on Robotics and Automation*], 2799–2804 (2002).

[2] Lange, S., Suenderhauf, N., and Protzel, P., "A vision based onboard approach for landing and position control of an autonomous multirotor uav in gps-denied environments," in [*Proc. of Int. Conf. on Adv. Robotics*], (2009).

[3] Wenzel, K. E., Rosset, P., and Zell, A., "Low-cost visual tracking of a landing place and hovering flight control with a microcontroller," *Intelligent and Robotic Systems* **57**, 297–311 (2009).

[4] Templeton, T., Shim, D. H., Geyer, C., and Sastry, S. S., "Autonomous vision-based landing and terrain mapping using an MPC-controlled unmanned rotorcraft," in [*Proc. Int. Conf. on Robotics and Automation*], 1349–1356 (2007).

[5] Johnson, A., Montgomery, J., and Matthies, L., "Vision guided landing of an autonomous helicopter in hazardous terrain," in [*Proc. Int. Conf. on Robotics and Automation*], 3966 – 3971 (2005).

[6] Mejias, L., Usher, K., Roberts, J., and Corke, P., "Two seconds to touchdown – vision-based controlled forced landing," in [*Proc. Int. Conf. on Intelligent Robots and Systems*], 3527–3532 (2006).

[7] Bosch, S., Lacroix, S., and Caballero, F., "Autonomous detection of safe landing areas for an uav from monocular images," in [*Proc. Int. Conf. on Intelligent Robots and Systems*], 5522 –5527 (2006).

[8] Ruffier, F. and Franceschini, N., "Visually guided micro-aerial vehicle: automatic take off, terrain following, landing and wind reaction," in [*Proc. Int. Conf. on Robotics and Automation*], 2339–2346 (2004).

[9] Herisse, B., Russotto, F.-X., Hamel, T., and Mahony, R., "Hovering flight and vertical landing control of a vtol unmanned aerial vehicle using optical flow," in [*Proc. Int. Conf. on Intelligent Robots and Systems*], 801–806 (2008).

[10] Goncalves, T. F. and Azinheira, J. R., "Vision-based autonomous approach and landing for an aircraft using direct visual tracking method," in [*Proc. Int. Conf. on Informatics in Control, Automation and Robotics*], 94–101 (2009).

[11] Blosch, M., Weiss, S., Scaramuzza, D., and Siegwart, R., "Vision based mav navigation in unknown and unstructured environments," in [*Proc. Int. Conf. on Robotics and Automation*], 21–28 (2010).

[12] Klein, G. and Murray, D., "Parallel tracking and mapping for small ar workspaces," in [*Proc. Int. Symposium on Mixed and Augmented Reality*], 225–234 (2007).

[13] Longuet-Higgins, H., "A computer algorithm for reconstructing a scenen from two projections," *Nature (293)* , 133–135 (1981).

[14] Luong, Q.-T. and Faugeras, O., "The fundamental matrix: Theory, algorithm, and stability analysis," *Comp. Vis. (17)* , 43–75 (1996).

[15] Longuet-Higgins, H., "The reconstruction of a plane surface from two perspective projections," *Proc. Roy. Soc. London Series B 227 (1249)* , 399–410 (1986).

[16] "Star detector wiki." http://pr.willowgarage.com/wiki/Star_Detector.

[17] Agrawal, M., Konolige, K., and Blas, M. R., "Censure: Center surround extremas for realtime feature detection and matching.," in [*Proc. Europ. Conf. on Comp. Vis.*], *LNCS* **5305**, 102–115 (2008).

[18] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L., "Speeded-up robust features (surf)," *Comp. Vis. Image Underst.* **110**(3), 346–359 (2008).

[19] Fischler, M. A. and Bolles, R. C., "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography.," *Commun. ACM* , 381–395 (1981).

[20] Cheng, Y., "Real-time surface slope estimation by homography alignment for spacecraft safe landing," in [*Proc. Int. Conf. on Robotics and Automation*], 2280–2286 (2010).

[21] Mitchell, I. M., Bayen, A. M., and Tomlin, C. J., "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games," *Trans. on Autom. Control* **50**(7), 947–957 (2005).

[22] Hoffmann, G. M., Huang, H., Waslander, S. L., and Tomlin, C. J., "Quadrotor helicopter flight dynamics and control: Theory and experiment," in [*Proc. of the AIAA Guidance, Navigation, and Control Conference and Exhibit*], (2007).

[23] "Ros (robot operating system) website." http://www.ros.org.