Fully Self-Contained Vision-Aided Navigation and Landing of a Micro Air Vehicle Independent from External Sensor Inputs

Roland Brockers, Sara Susca, David Zhu, Larry Matthies

Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA

ABSTRACT

Direct-lift micro air vehicles have important applications in reconnaissance. In order to conduct persistent surveillance in urban environments, it is essential that these systems can perform autonomous landing maneuvers on elevated surfaces that provide high vantage points without the help of any external sensor and with a fully contained on-board software solution. In this paper, we present a micro air vehicle that uses vision feedback from a single down looking camera to navigate autonomously and detect an elevated landing platform as a surrogate for a roof top. Our method requires no special preparation (labels or markers) of the landing location. Rather, leveraging the planar character of urban structure, the landing platform detection system uses a planar homography decomposition to detect landing targets and produce approach waypoints for autonomous landing. The vehicle control algorithm uses a Kalman filter based approach for pose estimation to fuse visual SLAM (PTAM) position estimates with IMU data to correct for high latency SLAM inputs and to increase the position estimate update rate in order to improve control stability. Scale recovery is achieved using inputs from a sonar altimeter. In experimental runs, we demonstrate a real-time implementation running on-board a micro aerial vehicle that is fully self-contained and independent from any external sensor information. With this method, the vehicle is able to search autonomously for a landing location and perform precision landing maneuvers on the detected targets.

1. INTRODUCTION

Unmanned micro air vehicles (MAVs) will play an important role in future reconnaissance, exploration, and search and rescue applications. While missions in these scenarios most likely involve human supervision, the deployability of MAVs will greatly depend on their ability to perform simple navigation tasks autonomously to reduce human work load and increase safety. Additionally, operations in cluttered environments like urban canyons, close to buildings and other man made structure, or under tree canopy, make it much more challenging to control these systems manually as there is usually no external position information (GPS) available in these environments to navigate the system. As a consequence, future systems will need the ability to execute limited navigation tasks like obstacle avoidance for fast traverse, detecting a possible landing site, flying to a nearby navigation target, or entering buildings fully autonomously.

However, because of size weight and power (SWaP) constraints, it is challenging to deploy heavy power- and CPU intensive sensor suites on MAVs, and small passive vision sensors have seen increasing use for navigation tasks, as a single light weight, passive sensor can be employed simultaneously for detection and 3D reconstruction. Using only a vision sensor nevertheless creates new challenges, as a structure from motion approach with a single moving camera can reconstruct 3D information only up to scale, unless the exact motion is known. Systems that are deployed outdoors at higher altitudes can overcome this issue by using GPS data for pose recovery, but this is not an option for systems operating in GPS-denied environments.To cope with this issue, we developed a vision based navigation system on a small UAV with a minimal sensor suite - a single camera, sonar altimeter, and IMU - that operates with on-board resources only. Our system uses a monocular simultaneous localization and mapping (SLAM) approach that processes images from a down looking camera for vehicle localization in a constructed global map. SLAM position measurements are fused with IMU data to generate high frame rate low latency position updates as an input to the vehicle control algorithm. Because of its good performance and scalability, we adapted *parallel tracking and mapping* (PTAM) on our platform, a visual SLAM algorithm originally developed by Klein and Murray,[1] and integrated map scaling with a sonar altimeter into the algorithm to regain scale.

With this implementation, our vehicle can navigate without any external sensors and resources, which we demonstrate in an autonomous landing experiment, a maneuver that is of particular importance to many persistent surveillance tasks,

Author contact information: {brockers, sara.susca, david.q.zhu, lhm}@jpl.nasa.gov

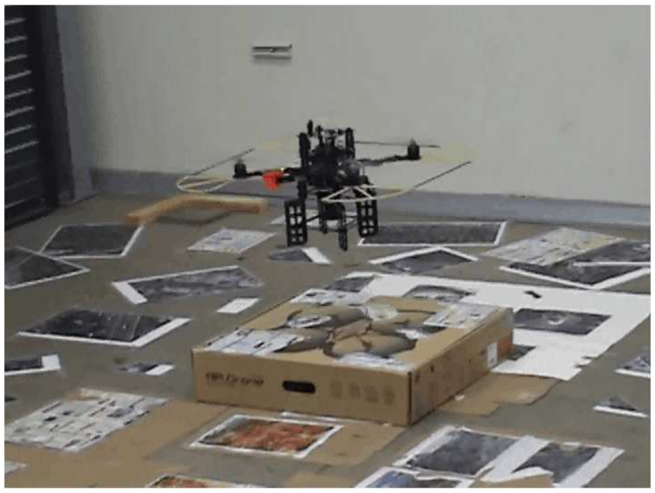

Figure 1. Quadrotor landing on the landing platform.

as the capability to land autonomously on elevated vantage points like flat roof tops or the top of poles and other man made structure is essential for perch and stare missions. To simplify this task to a laboratory set up, we implemented our autonomous landing software[2] on-board the system to detect an elevated box shaped landing surface as a surrogate roof-top and execute a fully autonomous landing maneuver onto the detected platform. The landing platform detector follows a homography based approach, which fits homographies to visual feature points to detect planar surfaces in view. It runs independently of the SLAM software, except that feature points in the input images together with frame pose are provided as a by-product of the SLAM algorithm. In our final implementation, we run the complete software package - visual SLAM, sensor fusion filter, landing spot detection, and the navigation software - on-board an AscTec Pelican quadrotor[*].

The rest of 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, while section 4 explains the actual implementation of the algorithms on-board our test vehicle. The whole system is evaluated in Section 5 demonstrating its performance during flight experiments. Section 6 concludes this paper and discusses future work.

2. RELATED WORK

The problem of localizing a moving vehicle in its environment usually involves measuring the position of 3D world reference points and estimating pose with respect to the location of these points. Methods that use beacon type reference points, like radio aids or GPS emitters, can solve this problem by calculating body positions at each frame independently. In contrast, if arbitrary point observations are used to calculate egomotion, reference points have to be identified over time which introduces a tracking problem that usually also includes the creation of some kind of map to store previous observations.

Various methods have been proposed to solve this problem and the literature is vast. Examples range from visual odometry approaches (VO) that focus on egomotion estimation[3] to simultaneous localization and mapping (SLAM) algorithms that emphasize on the creation of a global map.[4–7] To track features in the environment, some approaches use active range sensors and match 3D point clouds over time (e.g. sonar,[8] lidar,[5,6,9] or kinect[10]). Other approaches track image features obtained from camera images and reconstruct 3D point locations with a structure from motion approach (single camera[1,7]) or with range from stereo (stereo vision based approaches[3,11]).

Most of the above algorithm are computationally very demanding and therefore not suitable for running on a small UAV platform with its limited resources. Although new real-time visual SLAM approaches have been proposed in recent years that reduce the computational load by introducing a key-frame based mapping approach,[1,12,13] these approaches usually make heavy use of multi-core CPU/GPU implementations to achieve real-time performance. Nevertheless, a visual key-frame based approach can be sufficiently down scaled to run even on a very limited platform. An example of such a method is the parallel tracking and mapping (PTAM) approach originally developed by Klein and Murray.[1] In this approach

[*]Ascending Technologies GmbH

tracking and mapping are split and distributed to two different tasks that run independently. With this architecture, the tracking task can run efficiently at a higher frame rate and provide localization, whereas the mapping task is only triggered when sufficiently new feature points are observed and a new key-frame needs to be added to the map. Klein and Murray demonstrated that such an algorithm can be executed on a camera cell phone (iPhone3G with an ARM11 processor)[14] and Weiss et al.[15] and Achtelik et al.[16] successfully implemented a down scaled version of this approach on a small UAV similar to the one that is used in our approach. An example of a simplified Lidar based SLAM approach which also runs on the Asctec Pelican platform and reduces the full 3D approach to a 2D occupancy grid-based incremental SLAM method was introduced by Shen et al.[6]

Despite the effort of implementing a 'lean' SLAM algorithm on a limited hardware platform, the real-time performance of such an algorithm is usually barely sufficient to run a vehicle controller directly with position estimates from the SLAM algorithm, as low frame rates and large latencies result in poor control performance. As a result, a standard approach for many platforms is the Kalman filter-based fusion of low-frequency global position data, which could be from SLAM, GPS, or VO, with high frequency inertial data to compensate for latencies and increase the position update frame rate.[6, 9, 16]

We follow a similar approach and demonstrate that such a system can be used for autonomous navigation of a MAV, which involves detection of a landing platform and performing an autonomous landing maneuver.

Detecting navigation targets has a long history in terms of detecting and localizing artificially labeled landing sites. A number of image-based methods have been proposed to detect fixed markers[17–19] or terrain features[20–22] for landing site identification. In this paper, we deploy an approach that uses multiple homography decomposition to detect an elevated planar surface without any artificial labeling[2] which relates more to other homography based methods for identification of a single planar surface as potential landing sites for various helicopter and aircraft applications.[23, 24]

3. APPROACH

Our approach consists of three main parts that are all implemented on-board an Asctec Pelican quadrotor: vehicle controller, position estimation, and the navigation system which includes landing platform detection.

3.1 Vehicle control

The vehicle is controlled with three different control loops.[2] The *inner loop*, implemented in the quadrotor's firmware, stabilizes attitude using high frame rate IMU inputs. The *outer loop* controls the vehicle position in a world coordinate frame with inputs from our pose estimation filter. It is implemented as three PID position controllers, one for each axis of the world north east down (NED) frame. The *autonomy loop* is responsible for the navigation of the vehicle. It receives inputs from the landing platform detector and triggers high level maneuvers including take-off, landing and trajectory following.

3.2 Visual SLAM based localization

To localize the vehicle in the world frame, we use a monocular visual SLAM approach to generate pose estimates and fuse this global position measurement with IMU data for latency reduction and to generate a high frame rate control input to the outer loop controller. We implemented a tailored version of PTAM (parallel tracking and mapping)[1] on the on-board embedded computer of the Asctec Pelican which runs on images that are acquired from a downward looking camera. We streamlined PTAM's tracking task to accommodate the limited computational resource on a MAV platform, and included map scaling by inputs from a sonar altimeter which we explain in more detail in section 4.

3.3 Sensor fusion with IMU

Since the quadrotor is a fairly agile vehicle, the achievable control performance is dominated by the rate and latency of the 6DOF pose estimation loop in addition to accuracy of pose estimates. While the accuracy of SLAM algorithms is usually acceptable for vehicle control, computational requirements especially on low power platforms prevent these algorithms from running at reasonable speeds to serve directly as a position input to the outer loop controller without compromising performance. To overcome this issue, we fuse SLAM position measurements with data from an on-board IMU via an Extended Kalman Filter (EKF)[25] to obtain accurate and decreased latency 6DOF pose. Our EKF sensor fusion filter models 9 states: the error in position (δp), velocity (δv), and accelerometer bias (δb). As mentioned above, attitude

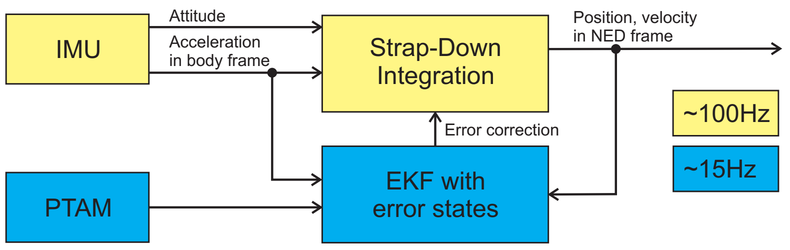

Figure 2. Structure of the EKF based pose estimation filter.

is provided by the low-level firmware filter, and we do not model attitude and gyro biases within the filter to decrease computational cost. The architecture of the sensor fusion filter is shown in figure 2.

The quadrotor dynamics for position (p) and velocity (v) are shown in eq. 1-3.

$$\dot{p}^n = v^n \tag{1}$$

$$\dot{v}^n = C_b^n(a^b - b_a^b) - g^n \tag{2}$$

$$\dot{b}_a^b = 0 \tag{3}$$

The superscript n stands for the NED frame, the superscript b indicates the IMU (body) frame, and C_b^n is the cosine direction matrix that rotates a vector from the body to the NED frame. a^b is the measured acceleration in body frame, b_a^b the accelerometer bias and g^n gravity in the NED frame.

The filter equations are shown in eq. 4-6 while the measurement equation is illustrated in eq. 7.

$$\delta\dot{p}^n = \delta v^n \tag{4}$$

$$\delta\dot{v}^n = C_b^n \delta b_a^b + \eta_a \tag{5}$$

$$\delta\dot{b}_a^b = 0 + \eta_{ba} \tag{6}$$

$$z = y_{PTAM} - y_{strapdown},\ H = [I_{3x3}, 0_{3x3}, 0_{3x3}] \tag{7}$$

y_{PTAM} and $y_{strapdown} \in \mathbb{R}^3$ are the quadrotor positions as computed by PTAM and by integrating the accelerometers.

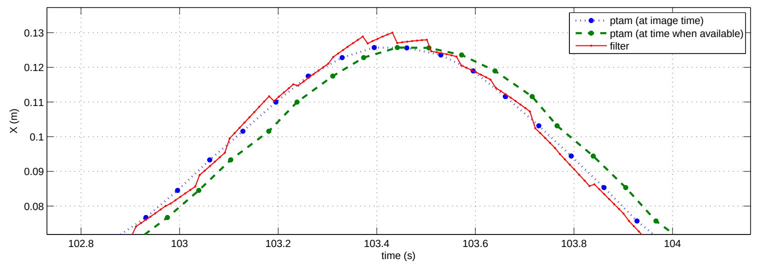

Figure 3. Latency reduction with the sensor fusion filter: Filter output and PTAM position measurement at time of availability compared to quasi ground truth (PTAM measurements at time of image acquisition).

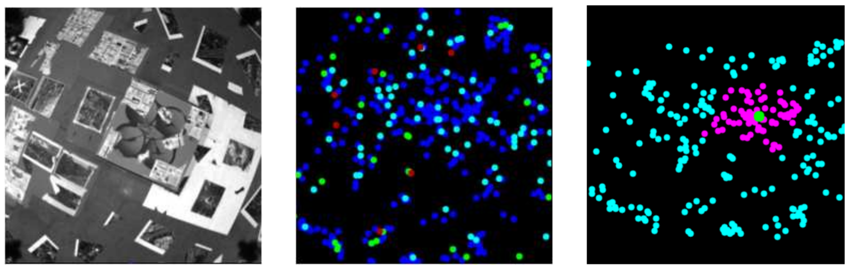
(a) (b) (c)

Figure 4. Inflight image processing steps: (a) original input image; (b) detected feature points that are used by PTAM to localize; (c) the landing detector separates those features as located on the ground plane (blue) and on the elevated surface (purple); the final target waypoint is marked in green.

The EKF update step is triggered anytime a PTAM measurement becomes available. To compensate for PTAM latency, the accelerometer, position, and velocity data is saved in a buffer and after every EKF update the current position is computed by reintegrating the acceleration data from PTAM time of validity (image acquisition time) to current time.

Figure 3 shows the effect of the latency reduction during a real experiment. The filter is able to largely compensate the latency introduced by the PTAM algorithm. In this figure and during the experimental evaluation we use PTAM position measurements at image acquisition time as a quasi ground truth, as PTAM accuracy at flight altitudes of 1m was better than 1cm.

3.4 Autonomous landing

To detect an elevated landing platform as a surrogate roof top we use a multiple homography approach to separate image feature points that are located on the ground and on the elevated platform. The landing platform detection is similar to previous work[2] with the difference that visual image features and frame poses are directly provided from the visual SLAM algorithm. As a result, in this implementation the landing platform detector uses PTAM's FAST corner features as inputs. A RANSAC based homography estimation first detects all feature points that are located on the ground surface (e.g. floor). In a second step, the algorithm fits a second homography to all remaining feature points to detect an elevated surface. If a second plane is detected, the plane parameters of both planes are refined by homography alignment.[2,26] Once sufficient feature points are detected on the elevated surface and the landing platform is completely in view (fig. 4c) a 3D waypoint is generated on top of the landing surface at its estimated center. If the platform height is above a minimum threshold, the 3D waypoint is than added to a sample pool that contains all waypoints that were generated previously in different camera frames. If the pool surpasses a minimum number of samples and the mean variation drops below a threshold, the landing platform is assumed to be detected stably and the mean 3D way point is passed to the autonomy loop as the new target waypoint. Once detected, the autonomy loop generates a hovering point directly over the target waypoint at the current height and commands the vehicle to this new hovering point. After the vehicle hovers stably at the hovering point, it slowly descends towards the target waypoint until it reaches a minimum altitude above the landing platform, where the motors are cut off to let the vehicle settle on the landing platform.

4. IMPLEMENTATION

4.1 Platform description

We implemented our algorithms on an AscTec Pelican quadrotor (figure 1), which is a 750g, 50cm diameter quadrotor that can carry up to 500g payload. In addition to the original sensor suite that comes with the vehicle (IMU, magnetometer, barometric pressure sensor) we installed a downward looking camera to track features on the ground (a PointGrey Chameleon USB camera with a Fujinon fisheye lens capturing 140° FOV, 640x480 images) and a MaxSonar EZ1 sonar altimeter (2.5cm resolution, 20Hz, 0.12-6.45m range) on the vehicle. Two ARM7 processors on a flight electronics board and an additional Intel Atom 1.6GHz CPU on an embedded computing board are available for on-board computation. Additionally, the embedded computing board is linked to a base station via WiFi for telemetry monitoring and for safety intervention.

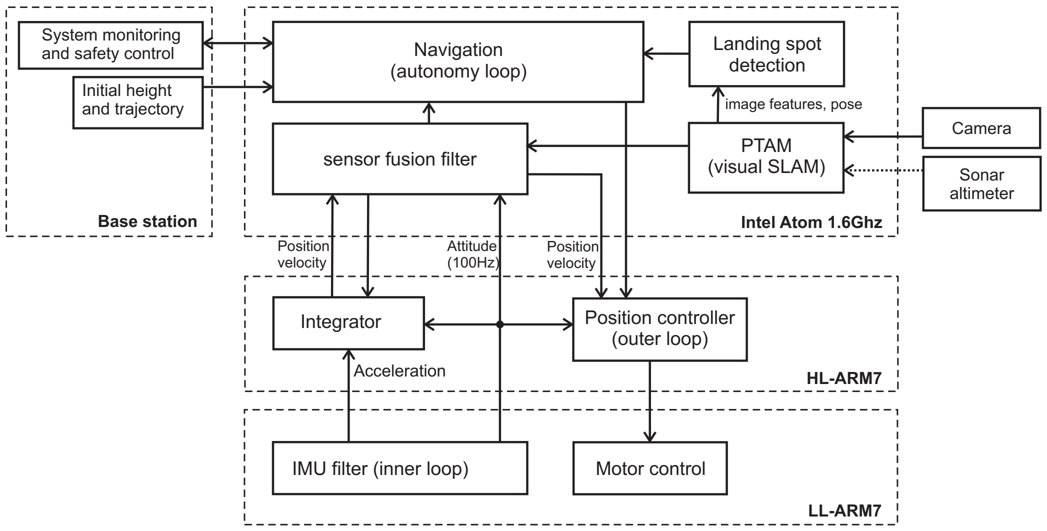

Figure 5. Software distribution on the quadrotor hardware.

4.2 Software architecture

In our implementation, we use the firmware attitude stabilization that ships with the vehicle, which includes a filter to estimate attitude angles in NED frame. This filter is running on one of the ARM7s (LL processor) at 1kHz. The other ARM7 (HL processor) runs our vehicle outer loop controller which communicates with the autonomy loop that is executed on the Atom board.

Figure 5 illustrates the different software components and how they are implemented on our system. The inner loop controller runs at 1kHz and the outer loop controller at 50Hz on the low level ARM processors. PTAM is running at approximately 15Hz and feeding into our pose estimation filter which is executed at 100Hz on the Atom. The autonomy loop runs at 5 Hz, with inputs from pose estimation and the landing platform detector. It also communicates with the base station computer that is linked via WiFi for data monitoring and for safety commands (engine cut-off). All inter-process communication on the Atom and the base station computer is implemented using ROS.[27]

4.3 PTAM adaption

We amended PTAM's map making thread to scale the generated map with measurements from a sonar altimeter during the initial map generation phase. In this phase, height measurements from a downward pointing sonar sensor are collected whenever a new key frame is added, and an average scale factor is calculated to scale PTAM's map to a metric map. This initial phase is terminated after a fixed number of key frames were added to the global map, which fixes the map scale for the rest of the map making process. All key frame generation and tracking parameters were adapted for fast tracking performance. Additionally, the tracking part of PTAM directly passes the rectified 2D feature point positions of detected image features to the landing spot detector. As a result, no additional image pre-processing is needed by the landing detection subsystem.

5. EXPERIMENTAL RESULTS

We conducted two experiments to evaluate the control accuracy of our system. In the first experiment the quadrotor is commanded to hover over a fixed position in the world frame. In the second experiment we evaluate the autonomous landing performance of our system.

For both experiments, we used pre-built PTAM maps to compare the performance of several experimental runs which all use the same global map. Additionally, this approach guarantees that delays introduced by PTAM's map making task - PTAM's bundle-adjustment usually generates a substantial delay when incorporating a new key-frame into its map - do not corrupt flight performances evaluations. Computationally, running only the localization part of PTAM results in a small speed up of the on-board implementation. For flight configurations that include on-board mapping, a good strategy to limit delays from the map making thread to a non-critical value is to reduce the number of key-frames in the map to a very small number in a sliding window approach.[16] Nevertheless, while generating a local map on the fly and ‘forgetting’ old key frames the position will drift over time.

To generate a map for our experiments the quadrotor was attached to a boom via a tether and moved manually with PTAM’s localization and mapping thread running on-board the quadrotor. Once the flight area was mapped sufficiently, the map making process was terminated and the collected map was stored to be used during the actual experiment.

5.1 Hovering in place

To evaluate the performance of the outer loop position controller independently, we conducted a hover in place experiment were we command the vehicle to hover over a fixed place in a pre-generated PTAM map and recorded position estimates from PTAM as a quasi ground truth to monitor the vehicle location (figures 6 & 7). To quantify the effect of the sensor fusion filter, we first feed the outer loop controller with position measurements from PTAM-only (figure 6). In this case

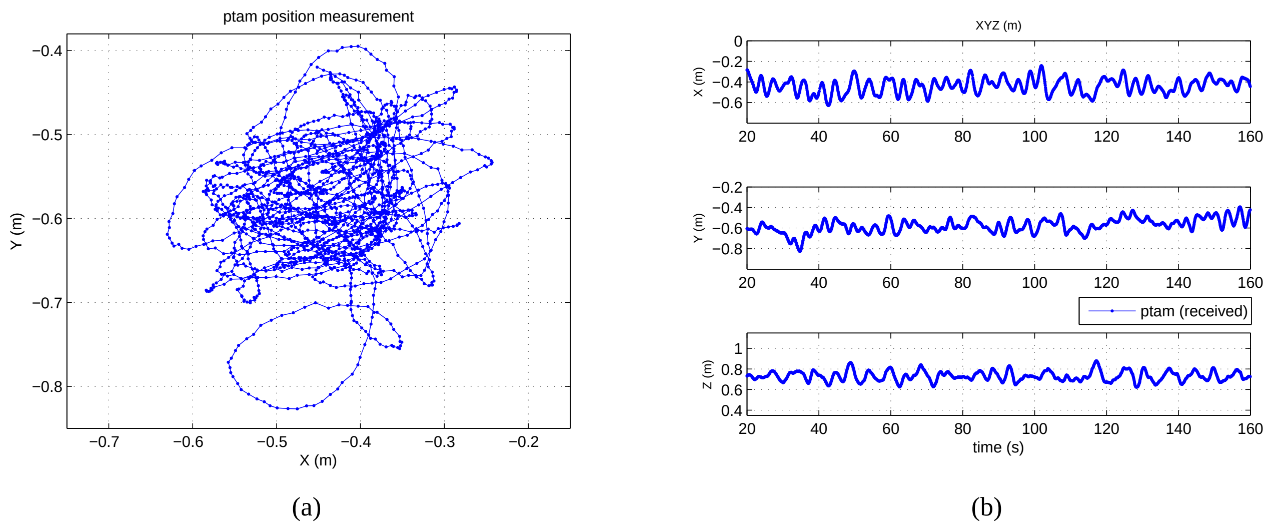

Figure 6. Hovering with PTAM-only position estimates: (a) top-down view of vehicle position during steady hover; (b) XYZ position estimates from PTAM.

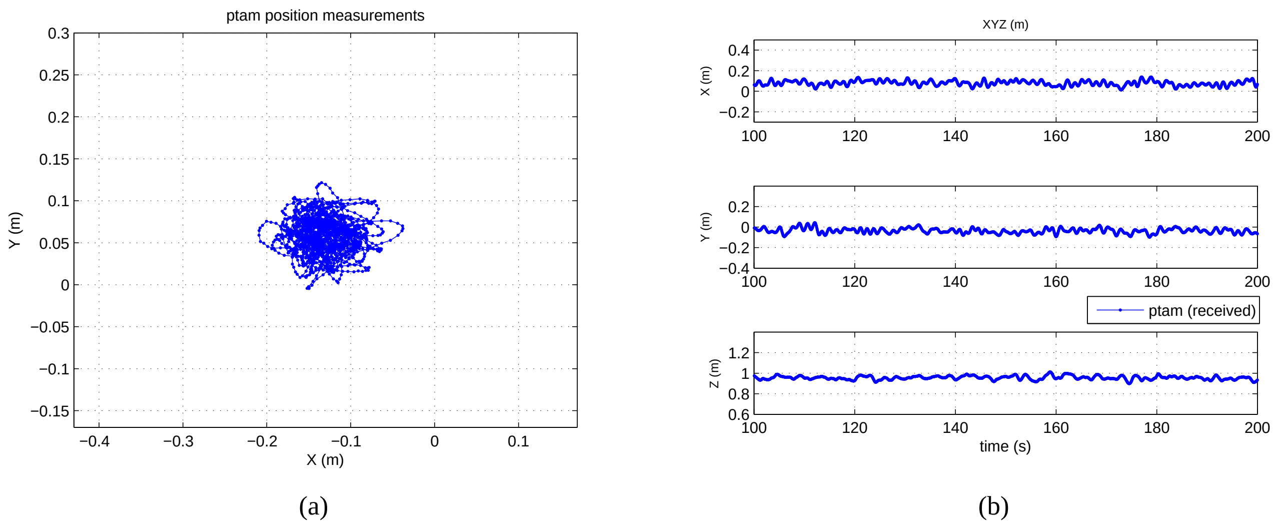

Figure 7. Hovering with sensor fusion filter position estimates: (a) top-down view of vehicle position during steady hover; (b) XYZ position estimates of the sensor fusion filter.

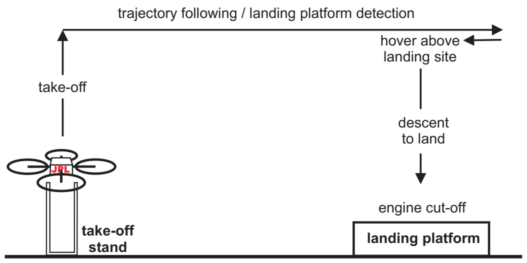

Figure 8. Overview of the landing experiment.

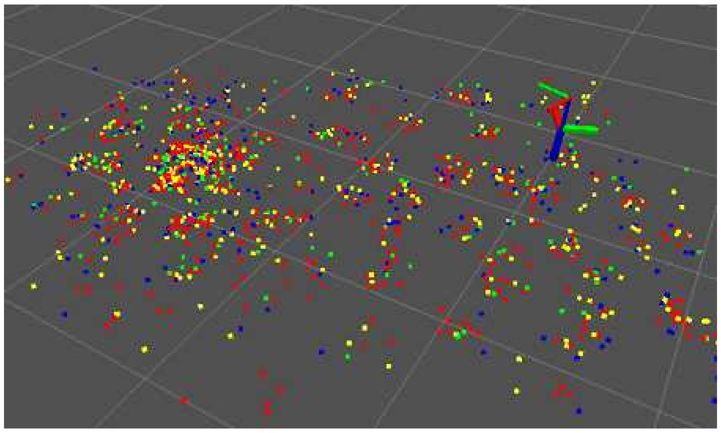

Figure 9. PTAM global map: the coordinate systems represent the position of the camera and the vehicle in the world frame. Colored dots are feature points in the global map with color coding feature scale.

velocity was calculated from PTAM position estimates, which came in at about 15Hz. The controller was running at a 20Hz rate, using pose from PTAM that was extrapolated with PTAM velocity to cope with occasional large PTAM delays.

Figure 7 depicts the hovering performance of the quadrotor when running our sensor fusion filter. The size of the hovering ellipse when flying with PTAM-only is about $\pm$ 15cm (RMS=9.91cm). The sensor fusion approach reduces the ellipse by a factor of 2 to $\pm$ 7.5cm and reduces the RMS by a factor of 3 to 3.44cm. This is largely due to the reduction of latency and the increase of control frequency.

5.2 Autonomous landing experiment

The setup for the autonomous landing experiment is shown in figure 8. The quadrotor is initially positioned on a stand to guarantee PTAM localization from the beginning of the experiment. To start the experiment, an operator initiates take-off to a predefined height. Once the vehicle hovers at this height, it is commanded to fly in a straight line towards the experimental landing platform - a card box with the dimensions 57cm x 57cm x 14cm (WxHxD). After this command is issued, the vehicle flies fully autonomously, detects the landing platform during overflight, and performs a landing maneuver onto the landing target.

The pre-built map for the landing experiment contained 117 key frames and is shown in figure 9. Figure 10 shows the trajectory (measured with PTAM) of the vehicle performing the autonomous landing experiment. The quadrotor was able to land on the platform for both control input configurations: when running with PTAM measurements as the sole inputs

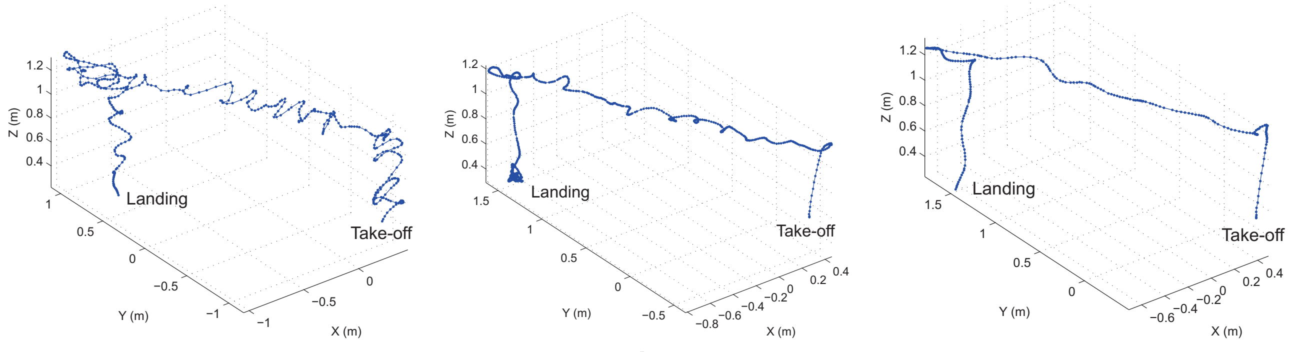

Figure 10. Trajectory of quadrotor landing on the landing platform: (a) PTAM-only (flight time 57s); (b) sensor fusion filter (flight time 47s); (c) fast flight with sensor fusion filter (flight time 18s).

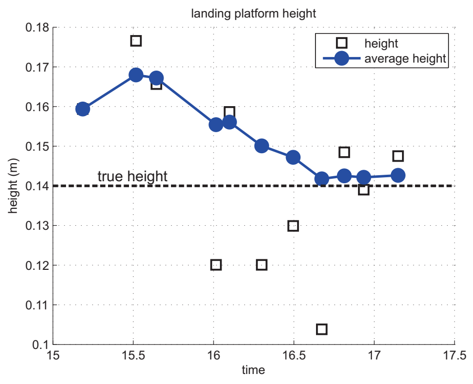

Figure 11. Estimated landing platform height.

to the outer loop controller, and when using pose estimates from the sensor fusion filter. The better performance of the sensor fusion approach is directly visible in the smaller trajectory fluctuations when controlling the vehicle with position estimates from our sensor fusion filter during flights of similar duration (fig. 10a & b). Additionally, the improved control stability permitted us to increase the vehicle speed by a factor of 3 with the same flight performance (fig. 10c).

To validate the accuracy of the landing platform detection software, the calculated platform height during detection is shown in figure 11 until convergence. The true platform height in our experiment was 14cm, which was approximated with sufficient accuracy by the landing platform detection algorithm.

6. CONCLUSIONS AND FUTURE WORK

Being able to navigate autonomously in unknown environments without any external input is crucial for future MAV reconnaissance applications. This paper presents a vision based position estimation approach that enables small UAVs to navigate autonomously when no external position information is available. Our approach fuses position measurements from a low-frequency, high latency visual SLAM approach with IMU data to estimate a high frame rate, low latency pose that can be used to navigate the vehicle within a global map. The algorithm was implemented on-board an AscTec Pelican quadrotor using only on-board sensors and resources. After demonstrating accurate performance of the system when hovering in place, we fused our autonomous landing detection software with the new system and demonstrated fully autonomous detection and landing on an elevated landing platform that does not need artificial labeling. Future work will include the transition of the landing detection approach to more general landing surfaces (sloped surfaces, poles, tree branches) and the addition of a safety layer in between the inner loop attitude stabilization and the high level map localization, which takes over control when the high level SLAM system fails to provide localization data (lost tracking). In this case, a fast optical flow based approach can provide egomotion estimation until the high level SLAM relocalizes.

Acknowledgments

This work was funded by the Army Research Laboratory under the Micro Autonomous Systems & Technology Collaborative Technology Alliance program (MAST-CTA). JPL contributions were carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

REFERENCES

[1] G. Klein and D. Murray, “Parallel tracking and mapping for small AR workspaces,” in *Proc. Int. Symposium on Mixed and Augmented Reality*, pp. 225–234, 2007.

[2] R. Brockers, P. Bouffard, J. Ma, L. Matthies, and C. Tomlin, “Autonomous landing and ingress of micro-air-vehicles in urban environments based on monocular vision,” in *Proc. SPIE 8031, 803111*, 2011.

[3] A. Howard, “Real-time stereo visual odometry for autonomous ground vehicles,” in *Proc. Int. Conf. on Intelligent Robots and Systems (IROS),* pp. 3946 –3952, 2008.

[4] E. Eade and T. Drummond, “Monocular SLAM as a graph of coalesced observations,” in *Proc. Int. Conv. on Computer Vision (ICCV)*, pp. 1 –8, 2007.

[5] D. Cole and P. Newman, “Using laser range data for 3D SLAM in outdoor environments,” in *Proc. Int. Conf. on Robotics and Automation (ICRA)*, pp. 1556 –1563, 2006.

[6] S. Shen, N. Michael, and V. Kumar, “Autonomous multi-floor indoor navigation with a computationally constrained MAV,” in *Proc. Int. Conf. on Robotics and Automation*, pp. 20–25, 2011.

[7] A. Davison, I. Reid, N. Molton, and O. Stasse, “MonoSLAM: Real-time single camera SLAM,” *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)* (6), pp. 1052 –1067, 2007.

[8] D. Ribas, P. Ridao, J. Neira, and J. Tardos, “SLAM using an imaging sonar for partially structured underwater environments,” in *Proc. Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 5040 –5045, 2006.

[9] A. Bachrach, A. de Winter, R. He, G. Hemann, S. Prentice, and N. Roy, “RANGE - robust autonomous navigation in GPS-denied environments,” in *Proc. Int. Conf. on Robotics and Automation (ICRA)*, pp. 1096 –1097, 2010.

[10] R. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, “KinectFusion: Real-time dense surface mapping and tracking,” in *Proc. 10th IEEE Int. Symp. on Mixed and Augmented Reality*, pp. 127–136, 2011.

[11] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid, “A constant time efficient stereo SLAM system,” in *BMVC*, 2009.

[12] B. Clipp, J. Lim, J.-M. Frahm, and M. Pollefeys, “Parallel, real-time visual SLAM,” in *Proc. Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 3961 –3968, 2010.

[13] K. Konolige and M. Agrawal, “FrameSLAM: From bundle adjustment to real-time visual mapping,” *Trans. on Robotics* **24**(5), pp. 1066 –1077, 2008.

[14] G. Klein and D. Murray, “Parallel tracking and mapping on a camera phone,” in *Proc. Int. Symp. on Mixed and Augmented Reality*, pp. 83 –86, 2009.

[15] S. Weiss, D. Scaramuzza, and R. Siegwart, “Monocular-SLAM-based navigation for autonomous micro helicopters in GPS-denied environments,” *J. Field Robot.* **28**, pp. 854–874, Nov. 2011.

[16] M. Achtelik, M. Achtelik, S. Weiss, and R. Siegwart, “Onboard IMU and monocular vision based control for MAVs in unknown in- and outdoor environments,” in *Proc. Int. Conf. on Robotics and Automation*, pp. 3056–3063, 2011.

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

[18] S. Lange, N. Suenderhauf, and P. Protzel, “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*, (Munich, Germany), 2009.

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

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

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

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

[23] S. Bosch, S. Lacroix, and F. Caballero, “Autonomous detection of safe landing areas for an UAV from monocular images,” in *Proc. Int. Conf. on Intelligent Robots and Systems*, pp. 5522 –5527, 2006.

[24] T. F. Goncalves and J. R. Azinheira, “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*, pp. 94–101, 2009.

[25] R. M. Rogers, *Applied mathematics in integrated navigation systems*, AIAA, 2nd ed., 2003.

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

[27] “ROS (robot operating system) website.” http://www.ros.org.