2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Vision-based Pose Estimation for Autonomous Indoor Navigation of Micro-scale Unmanned Aircraft Systems

Piotr Rudol, Mariusz Wzorek and Patrick Doherty

Abstract— **We present a navigation system for autonomous indoor flight of micro-scale Unmanned Aircraft Systems (UAS) which is based on a method for accurate monocular vision pose estimation. The method makes use of low cost artificial landmarks placed in the environment and allows for fully autonomous flight with all computation done on-board a UAS on COTS hardware. We provide a detailed description of all system components along with an accuracy evaluation and a time profiling result for the pose estimation method. Additionally, we show how the system is integrated with an existing micro-scale UAS and provide results of experimental autonomous flight tests. To our knowledge, this system is one of the first to allow for complete closed-loop control and goal-driven navigation of a micro-scale UAS in an indoor setting without requiring connection to any external entities.**

I. Introduction

Micro-scale Unmanned Aircraft Systems operating indoors have many interesting applications both in military and civilian domains. For example, they can be used to perform surveillance missions or to provide emergency responders with an initial overview in catastrophe situations. To facilitate such tasks the ability to operate such systems autonomously is highly desired. Unfortunately, limited payload capabilities of smaller airframes demand miniaturization of sensors and computer systems. This results in degradation of sensor accuracy and restricts computational power. In addition to this, the lack of reliable positioning techniques (e.g. GPS) is a fundamental hinder towards accurate autonomous indoor flight.

Several solutions which enable autonomous indoor navigation have been suggested. However, most of them delegate computationally heavy tasks to external processors or employ off-board system components to implement certain functions (e.g. positioning). Such delegation relies on wireless communication links which have several disadvantages. First of all, a connection can be interfered with which would render a system inoperable thus leading to a crash. Secondly, wireless communication links add nondeterministic delays which make sensor fusion and closed-loop control more challenging problems.

The system proposed is capable of autonomous indoor navigation and is self-contained when it comes to computation and sensing. No communication with external entities is required. The system uses easily reproducible and low cost rectangular markers to obtain position and heading information. The placement of markers is to a large extent arbitrary (i.e. on floors, walls, furniture etc.). For certain relative poses of a camera and a marker, accuracy can be lower and this is analyzed in Section IV. The pose information delivered by the vision system is fused with inertial measurements by means of an Extended Kalman Filter (EKF) in a timely and accurate manner on-board the UAS. The ability of our system to map the environment and localize the UAS allows for performing goal-directed flight.

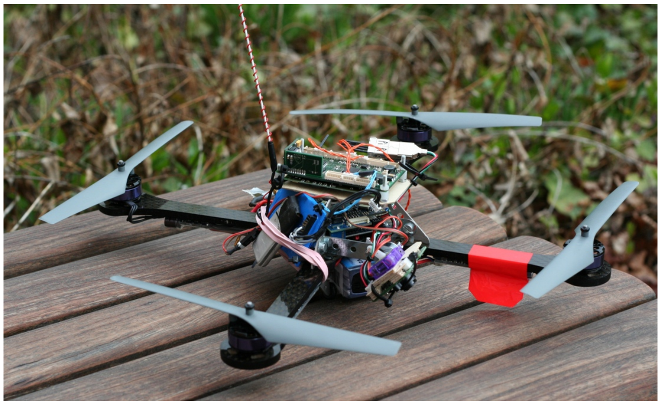

Fig. 1. The Hummingbird UAS platform with the LinkBoard flight control system used for experimental validation of the presented system.

The contributions of this research include a description of a complete and integrated system which allows for autonomous indoor flight. Analyses of the accuracy, robustness and timing requirements of marker detection, vision-based pose estimation and sensor fusion techniques are also provided. Additionally, the results of closed-loop control experiments performed with a quadrotor UAS are also presented.

Our solution provides indoor localization functionality and can jump-start many new research directions with UAS's in indoor settings. The system described can be considered as a testbed for development and experimentation with indoor flight control modes, path and motion planning algorithms, cooperative UAS operation in addition to many other applications.

The remainder of the paper is structured as follows. We begin with a description of related work in Section II and continue with an overview of the proposed system in Section III. Details of the vision-based pose estimation technique are presented in Section IV and followed by a description of the proposed sensor fusion method in Section V. Results

This work is partially supported by grants from the Swedish Foundation for Strategic Research (SSF) Strategic Research Center MOVIII, the Swedish Research Council (VR) Linnaeus Center CADICS and LinkLab and ELLIIT.

P. Rudol, M. Wzorek, and P. Doherty are with the Department of Computer and Information Science, Linköping University, SE-58183 Linköping, Sweden, {pioru|marwz|patdo}@ida.liu.se

978-1-4244-5040-4/10/$26.00 ©2010 IEEE

of experimental flight tests are presented in Section VII. The paper concludes with final remarks and a discussion of future work in Section VIII.

II. Related work

In recent years, autonomous indoor flight capabilities have been receiving increasing research interest. Solutions taking advantage of different sensor modalities in various configurations have been proposed.

A system based on the use of commercial motion capturing hardware by delivering position and attitude information to a UAS has been used to enable indoor navigation [1]. The main shortcomings of such a solution are the requirement for expensive infrastructure, off-board computation, and the need for constant communication with a Micro Air Vehicle (MAV) during flight.

A number of solutions take advantage of different types of artificial features to enable pose estimation for MAVs. One of them employs two cameras, one mounted on a pan-tilt unit on the ground and one on-board a quadrotor MAV [2]. The two cameras track colored blobs attached both to the UAS and to the ground camera. Besides the need for off-board processing, the disadvantage of this solution is a rather limited flight envelope accessible to a MAV. This method allows for indoor flight, but preferably above the ground camera. This considerably limits the operational range of the MAV. A different method makes use of information obtained from a target which takes advantage of a moiré pattern [3]. The pose of a camera is calculated relative to a novel pattern which requires backlighting. The flight test results presented show the applicability of the method for controlling a quadrotor platform by calculating the position and the yaw angle of the UAS. The disadvantage of this system is a limited operational range because it is not easy to obtain multiple unique instances for this kind of marker.

A system based on intuitions similar to those described in this paper has been presented in [4]. With the exception of describing the components of the system, this work does not provide any accuracy evaluation of the complete method nor does it describe the results of using such an estimate for complete closed-loop control of a MAV.

Several attempts have been made to solve the indoor navigation problem by means of SLAM. For example, a monocular vision SLAM technique for UAS's has been proposed in [5]. It exploits the architectural features of manmade indoor environments, namely corners detectable in corridors. The contribution of this work is "a new absolute range and bearing measurement algorithm using monocular camera". Unfortunately the authors do not provide enough detail to judge the applicability of the method for on-board execution nor any results that use the technique in closed-loop experiments.

A single camera solution is a basis for a navigation system in [6]. The system is capable of 3D localization but it requires an a priori map of the environment. The proposed algorithm is computationally intensive and is executed off-board on a wirelessly transmitted video stream.

A number of solutions suggest incorporating biologically inspired methods in order to deal with limited computational capabilities of micro-scale UAS's. Utilization of optic flow techniques has received the widest attention. It can be shown that optic flow can constrain the error in velocity and attitude [7]. Unfortunately, the technique does not completely eliminate the positional drift and additional information (such as geo-referenced images) is required for an accurate position estimation as described in [8].

Recently, systems based on laser range finders have gained a considerable amount of attention. One such system uses a particle filter to globally localize a quadrotor platform in a pre-computed map [9]. Unfortunately, the need for off-board computation makes the system vulnerable to communication interruptions. Additionally, commercially available small-scale laser range finders are much heavier than miniature video cameras. From the platform endurance perspective imaging sensors are much more attractive for UAS navigation.

III. System overview

The main functional components of our system are shown in Figure 2. The vision-based pose estimation module makes use of a set of arbitrarily placed artificial markers which form a map of the operational environment. This module delivers a 6 degrees of freedom (6-DOF) pose of a UAS i.e. 3D position and 3 attitude angles. The pose information (excluding roll and pitch angles) is fused using a Kalman filter with the inertial data (i.e. three accelerations and three angular rates) to produce the full state of the system. It consists of the pose and 3 velocities (north, east, and up). It is used as input to the control module which implements the flight control modes (e.g. hovering, navigation to a given point). The details of the system components are presented in the remainder of the paper.

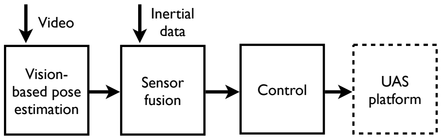

Fig. 2. Main functional components of the system.

A. Experimental UAS platform

The experimental system hardware components and their interconnections are schematically presented in Figure 3. The Hummingbird quadrotor UAS from Ascending Technologies GmbH has been used as the test platform [10]. It can carry up to 200 grams of payload and has 20 minutes of endurance. Its diameter is approximately 50 cm and the total weight is around 500 grams. The platform's own electronics (X-Base and ResearchPilot boards) implements the inner control loop (attitude stabilization) running at 1 kHz. The UAS can be flown by a human pilot via an RC transmitter but it also accepts control signals (i.e. roll, pitch, and yaw angles and thrust commands) via an RS232 connection.

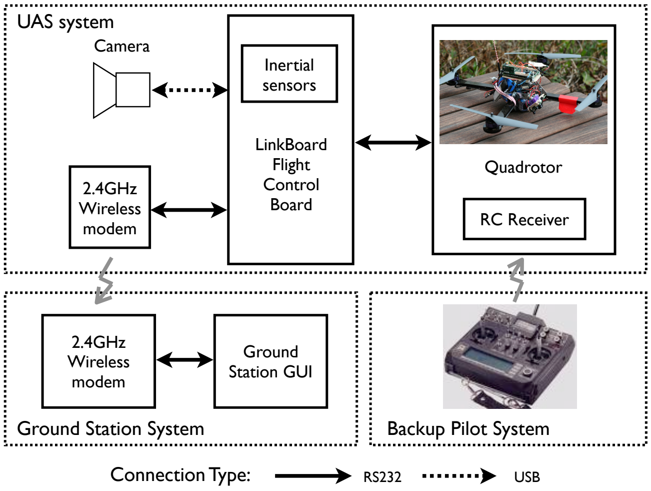

Fig. 3. The experimental system subcomponents and interconnections between them.

B. Flight control hardware

The LinkBoard (revision I) flight control board from UAS Technologies Sweden AB[1] has been used as the host for the state estimation algorithm and control. Thanks to the board's modular design it is capable of performing a wide range of tasks. In this application, the 520 MHz XScale processor, running a Windows CE 6.0 operating system, hosts the vision-based pose estimation algorithm. A Logitech QuickCam Pro 5000 provides the video stream via a USB connection. The result of the pose estimation is delivered via RS232 link to a second processor (60 MHz ARM7) which implements the sensor fusion algorithm. A second processor of the same type implements four proportional-integral-derivative (PID) loops commanding all the control channels of the UAS.

The flight control board interfaces with the ground control station software using a 2.4GHz wireless modem from Aerocomm[2]. The ground station software is used to monitor the on-board software execution by displaying the telemetry data. The modem can also be used to send commands up to the UAS. For example, the hovering position, altitude, or heading can be changed during an autonomous flight.

All the additional components, i.e. the LinkBoard flight control board, the camera, the modem, and a 330mAh battery, have a total weight of approximately 50 grams.

C. Software

The vision-based pose estimation algorithm is implemented using the ARToolkitPlus open source library [11]. It is a popular solution used in many augmented reality applications. The applicability of the ARToolkitPlus library for UAS navigation is evaluated in the following subsections. This includes an analysis of the usable range i.e. the attitudes and distances for which the accuracy of the pose estimation is sufficient for a controlled flight. It is important to note that the system architecture is generic in nature and allows for the use of other pose estimation algorithms.

IV. Vision-based Pose Estimation

The process of calculating a pose is divided into two major steps, *marker detection* and *pose estimation*. Marker detection is accomplished in the following manner. First, rectangular regions in a binarized image are found as hypotheses of legitimate markers (see Marker boundary in Figure 4B). Second, the interior of each hypothesis is analyzed in order to decode the identifying number (see Figure 4B ID boundary). The goal of the marker detection phase is to generate the ID and a sequence of the marker corners' coordinates measured in pixels. The pose estimation algorithm calculates the relation (rotational and translational) between the camera and the marker coordinate systems. The calculation is based on the coordinates of detected marker corners (i.e. projection of the marker model on the image plane, cf. Figure 4A).

Both steps of the vision-based pose estimation process perform differently depending on the relation of the marker and the camera in terms of translations and rotations. For example, for certain distances and camera attitudes the estimation of the pose is not possible at all or the result has a considerable error. The following subsections describe the process and results of evaluating the algorithm from the perspective of UAS navigation.

A. Marker detection

A marker ID can be coded in three different ways in the ARToolkitPlus library. The first one uses a template matching technique. This type is most computationally and memory expensive since the interior of a detected rectangle is compared with an image loaded at startup. For multiple markers all images have to be stored in memory for comparison. The second type called *Simple ID* supports 512 predefined markers and relies on decoding the ID from an array of bits (e.g. 6×6) located inside a rectangle boundary of a marker. The third type called *BCH ID* (depicted in Figure 4B) encodes marker IDs using a forward correcting code and supports up to 4096 combinations.

Both Simple- and BCH ID coding methods were evaluated to assess the optimal camera placement relative to a marker which assures reliable detection. A set of images was produced by projecting a marker (with linear interpolation) of 100×100 mm size onto the image plane of a camera with image size of 320×240 pixels, focal lengths $f_x = f_y = 386$ and the principal $c_x = c_y = (160, 120)$.

The camera was placed as schematically depicted by black dots in Figure 4C i.e. in 50 mm increments in X_M and Z_M directions up to 2 m. Three representative rotations around the Z_M axis were used: 0, 22.5, and 45 degrees (respectively, red, green, and blue marked regions). The rotation around the X_{CAM} axis was chosen so that the camera was always pointing to the center of the marker which was located at the origin of the coordinate system. Marker detection was performed on the set of generated images.

[1] Homepage: www.uastech.com

[2] Homepage: www.aerocomm.com

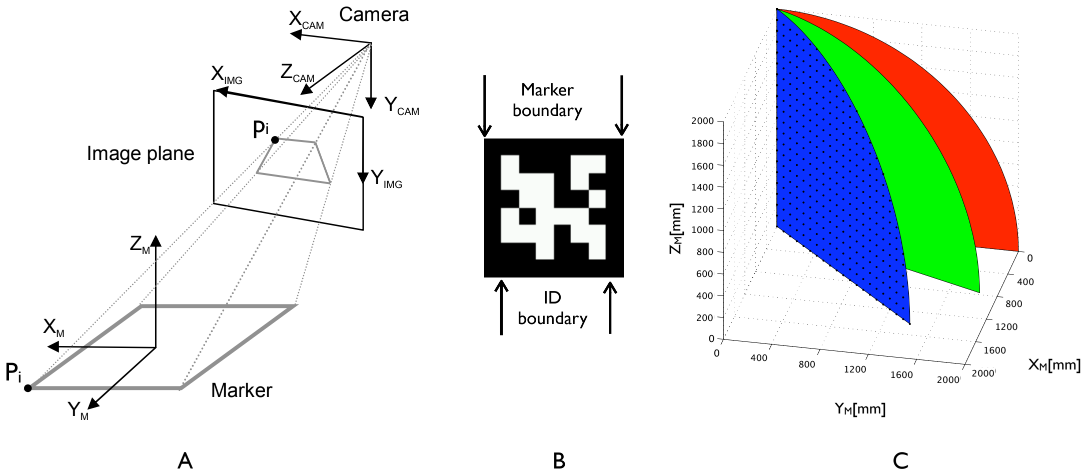

Fig. 4. A. Vision-based pose estimation coordinate systems B. Example marker C. Camera positions (black dots) for accuracy estimation.

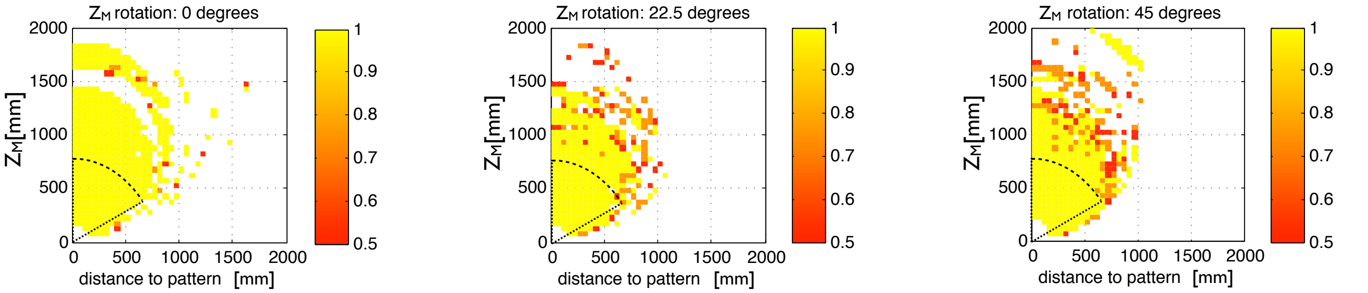

Fig. 5. BCH Marker detection confidence larger than 0.5 for three rotations around the Z_M axis.

The confidence of detecting a marker for three rotations around the Z_M axis is shown in Figure 5. The confidence for detecting BCH IDs is decreased from 1 by 0.25 for every detected and corrected error bit. Results with a confidence factor lower than 0.5 are not included in the plot. The result of detecting Simple IDs was on average 13 percent worse i.e. detection was successful for fewer camera poses. The use of BCH markers adds almost no computational penalty and is preferred for UAS navigation because it allows for the coverage of large operational environments due to a larger number of available IDs.

In order to maximize the marker detection rate, the distance of the camera to a marker should not be larger than 75 cm for the given marker size and camera. Additionally, the image plane should not be tilted more than approximately 60 degrees from the marker. The area with high detection rate is depicted by the dashed line in Figure 5.

The evaluation did not include the effects of lens distortion. However, with accurate camera calibration this is a minimal error contributor. Additionally, the experimental setup did not include effects of blurring when the camera moves relative to a marker. Depending on the imaging sensor used the effect of blooming or streaking can also influence the detection performance. However, both effects can be minimized by appropriately adjusting the camera parameters as described in Section VII.

B. Pose Estimation

After detecting a marker and extracting its four corners $\mathrm{p}_{i=1\ldots4} = \left[p_{i_x}, p_{i_y}\right]^T$ and given four coplanar model points $\mathrm{P}_{i=1\ldots4} = \left[P_{i_x}, P_{i_y}, 0\right]^T$ a pose estimation algorithm calculates the rotation expressed in Euler angles $R = f\left(\alpha, \beta, \gamma\right)$ and translation $\mathrm{t} = \left[t_x, t_y, t_z\right]^T$ of a camera such that:

$$\mathrm{p}_i \propto R\mathrm{P}_i + \mathrm{t} \tag{1}$$

Since p_i is expressed in camera coordinates, it is in practice perturbed by noise and measured in an image as $\hat{\mathrm{p}}_i$.

In general, calculating a pose i.e. finding R and t can be achieved by minimizing one of the commonly used error functions. For example, an *image space* error function in case of bundle-adjustment algorithms or an *object space* error function (E_{os}) used in the *Robust Pose Estimation Algorithm for Planar Targets* (RPP) [12] can be used. In the latter case, a special parameterization allows one to deal with pose ambiguities with up to a 50 percent better success rate over a standard definition. The algorithm uses an initial pose estimate and finds a second minimum of the E_{os} error function. If it exists, the correct pose should yield the lower value of the error function. Due to this property, the algorithm exhibits a considerably lower ”jerkiness” in the calculated pose. While in augmented reality applications this property gives a better visual result, for UAS navigation it gives more stability and robustness.

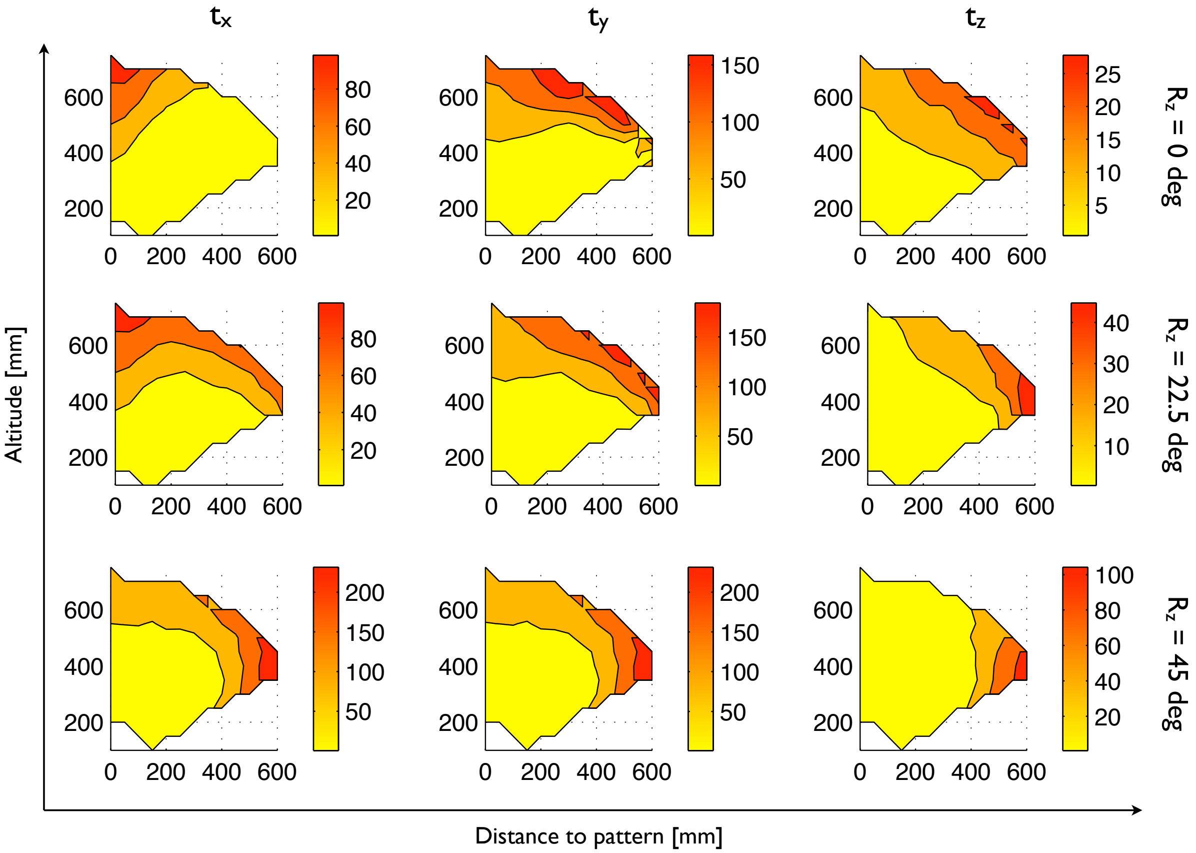

Fig. 6. Error standard deviation in mm of translation for 0, 22.5, and 45 degrees of camera rotation around Z_M axis of marker coordinate system. Noise of 2 pixels.

The accuracy and the noise of the RPP were assessed in a series of Monte Carlo simulations. Poses of the camera were the same as for marker detection experiments and are shown in Figure 4C. The model points P_i were projected onto the image plane for the given R and t and the camera model described earlier. The values of $\hat{\mathrm{p}}_i$ were calculated by perturbing p_i with a uniform noise of 2 pixels in 750 iterations. The error in attitude angle estimation was measured as a difference between the nominal values of (α, β, γ) and the average results of the algorithm in the presence of noise. Similarly, the error of position and altitude was measured as the difference between the nominal value of t and the average result in the presence of noise. Figure 6 shows the standard deviation of the error in three axes (columns t_x, t_y, and t_z) of the camera translation in the marker coordinate system with 2 pixel noise. The three rows correspond to the three rotations around the Z_M axis as described earlier. The range of altitude and distance to a marker were chosen to emphasize the range of marker detection described in the previous subsection.

The RPP algorithm exhibits mean position error and the error of standard deviation in a range of several centimeters for distances up to 50 cm for the given set of parameters (i.e. 10 x 10 cm pattern size, 320 x 240 pixels image size). The attitude angles are estimated with an average error below 3 degrees with a standard deviation below 6 degrees. In order to assess a usable range of the algorithm, the results can be scaled. For example, doubling the size of the pattern cuts the error in half or doubles the range. The same applies to doubling the resolution of the camera. For an image of size 640 x 480 pixels and a pattern of size 50 x 50 cm, the error measured in centimeters can be reached for up to 5 meters.

The usable range can additionally be extended by fusing the vision result with inertial data. In such cases, increased noise from the vision result can be handled by, for example, a Kalman filter.

The minimum distance to a marker depends on the field of view of the camera at hand. Without loss of generality, for a given field of view of α degrees and a rectangular marker width of w mm the minimum distance can be calculated as:

$$d_{min} = \frac{w}{2 \cdot \tan(\frac{\alpha}{2})} \tag{2}$$

In practice the distance should be larger to avoid cases when a marker occupies the full frame.

C. Mapping

Navigation relative to one marker allows only for hovering above it. In order to enlarge the operational range a set of markers is necessary. Thanks to a large number of available IDs (each identified by a BCH code) it is possible to build a map and effectively enlarge the area of navigation. Such a map is a set of marker poses relative to one marker chosen as the origin of the coordinate system. During navigation, when a marker is detected and the relative camera pose is obtained it can be recalculated into the absolute pose in the environment. A map can be built online during environment exploration. A straightforward way to build a map is to iteratively calculate relative poses of newly detected markers in relation to the one chosen as the origin. This requires two markers to be visible in the camera frame at once.

The result of the pose estimation of two markers is given by rotation matrices ${}^{c}_{m}R_i$ and translations ${}^{c}_{m}\mathrm{t}_i, i = 0, 1$. The results capture marker pose relative to the camera coordinate system. The camera pose relative to a marker can be calculated as ${}^{m}_{c}R = {}^{c}_{m}R^T$ and ${}^{m}_{c}\mathrm{t} = {}^{m}_{c}R\ {}^{c}_{m}\mathrm{t}$. The relative pose can be calculated as follows:

$$R_R = {}^{m}_{c}R_0 {}^{c}_{m}R_1 \tag{3}$$

$$\mathrm{t}_R = {}^{m}_{c}R_0 \left({}^{m}_{c}\mathrm{t}_1 - {}^{m}_{c}\mathrm{t}_0\right) \tag{4}$$

These two values are saved. During navigation, when the i-th marker is observed the absolute pose (relative to the marker designated as the origin M) can be calculated as follows:

$${}^{c}_{M}R = R_R\ {}^{c}_{m}R_i \tag{5}$$

$${}^{c}_{M}\mathrm{t} = R_R\ {}^{m}_{c}\mathrm{t}_i + \mathrm{t}_R \tag{6}$$

This method has a very small computational footprint but has a drawback. The error of the pose estimation accumulates in relative poses and grows with distance from the marker chosen as the origin. In other words, the farther away from the origin, the larger the error of the absolute pose. The error can be minimized by measuring the relative displacement of two markers several times and using an average. This will make the error smaller but will not eliminate it.

One solution to this problem could be to employ a method for explicit loop closure which in this case would be a relatively easy task due to the fact that loop detection is solved thanks to using unique IDs.

Another way to solve the problem could be to calculate a globally consistent map given all measurements as in [13]. This algorithm operates on a graph where markers are represented as vertices and relative poses of two markers as edges. It is more computationally expensive than the simple solution and is more suitable for offline calculation.

D. Timing Analysis

In order to assess the feasibility of using the proposed vision-based pose estimation on-board a micro-scale UAS, a timing analysis of the algorithm was performed. It was made by measuring execution times of marker detection and pose calculation implemented in the ARToolkitPlus library on the LinkBoard flight control board. The marker detection phase of the vision-based pose estimation takes approximately 10 percent of the total time of processing of a single frame. Large numbers of markers visible at once in a frame do not add much computational time. Additionally, the detection time is independent of the camera pose.

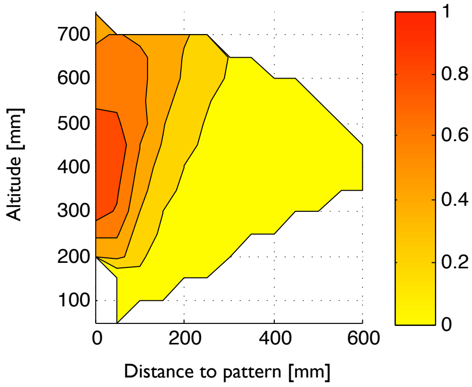

Fig. 7. Normalized average number of iterations for vision-based pose estimation algorithm depending on the relative camera and marker pose.

The pose calculation phase takes the majority of a frame processing time. The amount of time depends on the pose of the camera relative to the marker. Figure 7 presents normalized average number of iterations performed by the RPP algorithm. It is approximately the same for all experiments described in Section IV-A. The number of iterations grows when the image plane and the marker plane become parallel. From a computational efficiency perspective, and to achieve the highest frame processing rate, such a relation between the camera and the marker should be avoided during navigation. When the angle between the marker and the image plane is more than approximately 30 degrees the pose calculation takes between 100-200 ms for a single marker (note that the PXA270 microprocessor does not have a floating-point unit). Calculating poses of multiple markers in one frame should be avoided in order to maintain a high processing rate. In fact, it is only required during mapping of the environment. If a map is available and several markers are detected only one of them can be selected to calculate the absolute pose. Selecting the largest (measured in pixels) detected marker is a very good strategy. This allows for calculating the pose only for this marker, maintaining a high processing rate.

V. Sensor Fusion

In order to provide a robust navigation solution, the position and heading information delivered by the vision-based pose estimation is fused with the inertial data (from 3-axis accelerometers and 3 rate gyroscopes) by means of a 9 state Extended Kalman Filter (EKF). The use of pitch and roll angles from vision could improve the accuracy of the state estimation but would increase the computational load and was empirically proved not to be necessary to achieve good performance. The state of the system is represented by position, horizontal and vertical velocities, and attitude angles. The schematics of the sensor fusion technique is presented in Figure 8. The EKF uses a linear model for the measurement and process equations. The filter is implemented in the error dynamics formulation and the state represents the errors of the inertial navigation system (INS). The INS mechanization process is responsible for the time

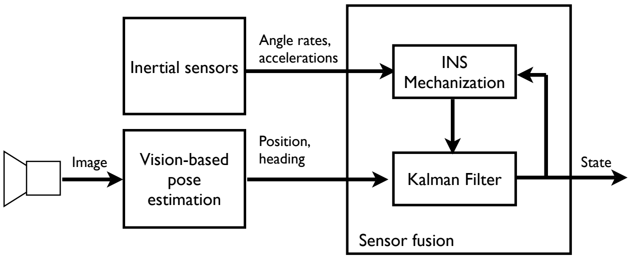

Fig. 8. The state estimation schematics.

integration of the inertial sensor data (i.e. dead reckoning). It provides the full navigation solution but it suffers from a position drift which grows in time. In order to bound this error the output of the EKF (i.e. errors of the position, velocities, and attitude) which is aided by the position update is fed to the INS mechanization process. Details of such a formulation of the sensor fusion technique can be found in [14].

It is important to note that the position update bounds the drift of the INS "locally" . The global position can suffer from an error which in turn can be bounded depending on the method used to create a map as described in Section IV-C.

Due to the sensor fusion technique, the UAS is able to navigate for a limited amount of time even without the position information delivered by the vision system. The inertial sensors provide the complete state estimate with a drift, which is corrected as soon as a vision-based position update is available. Experimental results of the presented sensor fusion technique are presented in Section VII-B.

VI. Control

All computations in the system are executed on-board the UAS. The inner control loop (i.e. attitude stabilization) is performed by the Hummingbird quadrotor platform electronics. It accepts input through a serial connection in the form of angles in case of roll and pitch, angular velocity in case of yaw, and the thrust value of the altitude channel.

These signals are produced by the outer control loop (i.e. position stabilization) computed on the LinkBoard autopilot. Four PID control loops are responsible for calculating the control signals. During initialization of the outer loop four initial control values are taken from the UAS avionics. They correspond to the values of the sticks of the RC transmitter when the UAS is operated manually. The initialization step is performed when the autonomous flight is engaged and results in a smooth transition from the manual flight.

VII. Experimental Evaluation

The system has been evaluated in real flight tests with all the described components operational. No tethers or other aids were used to validate the performance of the system in fully autonomous flight.

A. Experimental Setup

The flights were performed in an office environment. A set of 10 randomly placed artificial markers of size 17×17 cm printed on white paper (see Figure 10) were used to compute a map of the environment. The map was constructed on-line during an exploration flight using the procedure described in Section IV-C and then saved for successive flights.

In order to minimize the influence of the "rolling shutter" of the CMOS sensor the camera parameters were adjusted to use the shortest possible exposure time. To compensate for darker image the contrast and brightness parameters were appropriately adjusted. Due to this, the typical skew and wobble effects were to a large extent removed from the images.

The camera was pointed 45 degrees down to achieve minimal pose estimation error and a high processing rate (cf. Section IV). The relation between the camera and the UAS coordinate systems was measured with centimeter accuracy. This proved to be sufficient but for the best precision the relation could be fully calibrated using, for example, the method described in [15].

The camera was calibrated to find intrinsic and lens distortion parameters using the Camera Calibration Toolbox for Matlab [16].

B. Experimental results

Several hours of autonomous flights were performed during development and evaluation of the system. The marker size used allowed for stable hovering up to approximately 1.5 m altitude at different positions and headings in a 4×4 meters region. The results achieved confirm the simulation results presented in Section IV. Since only 10 out of 4096 markers were used, the operational range can be substantially extended in the future. The vision-based pose estimation algorithm achieves a rate of 5-10 Hz. The Kalman filter based state estimation delivers results at a rate of 50 Hz. Figure 9

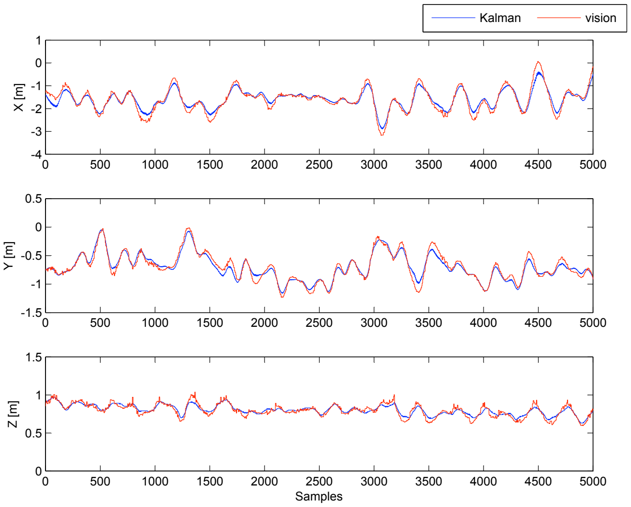

Fig. 9. Pose estimation results. Kalman filter output compared to raw vision data.

shows example Kalman filter estimated position and altitude plots in comparison with raw vision results. The system is capable of maintaining stable flight without position update for up to approximately 5 seconds. This time can be extended by additionally estimating biases of the inertial sensors. This will decrease the drift rate of the INS mechanization process.

In order to achieve stable hovering, it was necessary to include the estimated velocities especially for the horizontal control and to a lesser degree for altitude. The yaw channel did not require velocity in the outer control loop to achieve good performance. Due to battery power depletion over time, it was necessary to include the integral term of the PID controller, especially for the altitude channel. Autonomous flights within the 20 minutes of the platform's endurance were repeatedly achieved.

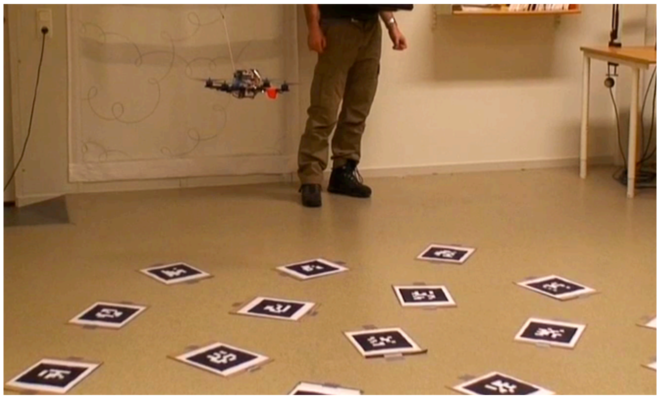

Fig. 10. Autonomous hovering of the presented system.

VIII. Conclusion and future work

An implemented and empirically validated system for autonomous indoor navigation has been presented. The system employs a monocular vision-based pose estimation which uses uniquely coded low-cost rectangular markers. The use of multiple markers allows for mapping large areas where a UAS can operate. The range and accuracy of the pose estimation algorithm has been evaluated and a sensor fusion technique based on a Kalman filter has been presented. Compared to other solutions the system is computationally self contained. No data nor video transmission to other entities is required to achieve autonomous flight. All computations are performed on-board the UAS. This removes the need for continuous wireless communication and increases the robustness of the system.

The system has been tested during several hours of flight integrated with a commercially available quadrotor UAS platform. The successful experiments with the existing system show that it is a very promising step towards true autonomous indoor navigation. In fact, further development of the system will focus on moving away from the use of artificial landmarks. One possible approach could take advantage of the results presented in [17] and making appropriate adjustments for applicability of that work to indoor environments.

Additionally, future work will include an evaluation of the absolute accuracy of the solution presented, using a high accuracy reference system as well as integrating path planning and execution techniques and additional control modes.

Acknowledgements

Authors would like to thank Gianpaolo Conte for fruitful discussions and help during testing of the system.

References

[1] J.P. How, B. Bethke, A. Frank, D. Dale, and J. Vian. Real-time indoor autonomous vehicle test environment. *Control Systems Magazine, IEEE*, 28(2):51–64, April 2008.

[2] E. Altuğ, J. P. Ostrowski, and C. J. Taylor. Control of a quadrotor helicopter using dual camera visual feedback. *International Journal of Robotics Research*, 24(5):329–341, 2005.

[3] G. Tournier, M. Valenti, J. P. How, and E. Feron. Estimation and control of a quadrotor vehicle using monocular vision and moire patterns. In *Proc. of the AIAA Guidance, Navigation and Control Conference*, 2006.

[4] W. Bath and J. Paxman. UAV localisation & control through computer vision. In *Proc. of the Australasian Conference on Robotics and Automation*, 2005.

[5] K. Celik, Soon Jo Chung, and A. Somani. Mono-vision corner slam for indoor navigation. In *Proc. of the IEEE International Conference on Electro/Information Technology*, pages 343–348, May 2008.

[6] S. P. Soundararaj, A. Sujeeth, and A. Saxena. Autonomous indoor helicopter flight using a single onboard camera. In *Proc. of the International Conference on Intelligent Robots and Systems (IROS)*, 2009.

[7] J. Kim and G. Brambley. Dual optic-flow integrated navigation for small-scale flying robots. In *Proc. of the Australasian Conference on Robotics and Automation*, 2007.

[8] G. Conte and P. Doherty. Vision-based unmanned aerial vehicle navigation using geo-referenced information. *EURASIP J. Adv. Signal Process*, 2009:1–18, 2009.

[9] S. Grzonka, G. Grisetti, and W. Burgard. Towards a navigation system for autonomous indoor flying. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, 2009.

[10] D. Gurdan, J. Stumpf, M. Achtelik, K.-M. Doth, G. Hirzinger, and D. Rus. Energy-efficient autonomous four-rotor flying robot controlled at 1 khz. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 361–366, April 2007.

[11] D. Wagner and D. Schmalstieg. Artoolkitplus for pose tracking on mobile devices. In *Proc. of the Computer Vision Winter Workshop*, 2007.

[12] G. Schweighofer and A. Pinz. Robust pose estimation from a planar target. In *Proc. of the IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 28, pages 2024–2030, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[13] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.

[14] G. Conte. *Vision-Based Localization and Guidance for Unmanned Aerial Vehicles*. PhD thesis, Linköping University, Department of Computer and Information Science, The Institute of Technology, 2009.

[15] J. Hol, T. Schön, and F. Gustafsson. A new algorithm for calibrating a combined camera and imu sensor unit. In *Proc. of the 10th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Hanoi, Vietnam, December 2008.

[16] J.-Y. Bouguet. Camera calibration toolbox for matlab. Accessed February 2010.

[17] K. Konolige, M. Agrawal, and Joan Solà. Large scale visual odometry for rough terrain. In *Proc. of the International Symposium on Research in Robotics (ISRR)*, November 2007.