A Vision Based Onboard Approach for Landing and Position Control of an Autonomous Multirotor UAV in GPS-Denied Environments

Sven Lange, Niko Sünderhauf, Peter Protzel
Department of Electrical Engineering and Information Technology
Chemnitz, University of Technology
09111 Chemnitz, Germany
{sven.lange, niko.suenderhauf, peter.protzel}@etit.tu-chemnitz.de

***Abstract*— We describe our work on multirotor UAVs and focus on our method for autonomous landing and position control. The paper describes the design of our landing pad and the vision based detection algorithm that estimates the 3D-position of the UAV relative to the landing pad. A cascaded controller structure stabilizes velocity and position in the absence of GPS signals by using a dedicated optical flow sensor. Practical experiments prove the quality of our approach.**

I. Introduction

Our research interests focus on enabling autonomous, mobile systems to be applicable in a variety of civil applications, mainly in the areas of emergency response, disaster control, and environmental monitoring. These scenarios require a high level of autonomy, reliability and general robustness from every robotic system, regardless of whether it operates on the ground or in the air.

In previous and parallel work with autonomous airships we gained experience with UAV control and autonomous navigation [1], [2]. Compared to airships, the multirotor UAVs we use in one of our current projects are of course much smaller and can carry much less payload. On the other hand, due to their compactness, they can be deployed a lot faster and do not require any preparation except for connecting the batteries. Their shorter flight time duration is compensated for by the quickly exchangeable batteries that allow a fast re-takeoff. Compared to helicopters, multirotor systems are cheaper, require less maintenance effort, are much more stable in flight and less dangerous due to their smaller and lighter rotors. E.g. the Hummingbird quadrotor we use, is equipped with flexible rotors. This way, the UAV is very safe and does not cause any injuries if a person accidently touches the rotor.

To reach the desired level of autonomy that is required by the mission scenarios, the UAV has to be able to take off, navigate, and land without the direct control of a human operator. While autonomous waypoint navigation is working well when GPS is available and autonomous take off is not a big challenge at all, autonomous landing remains a delicate process for all kind of UAVs. Several groups and authors have addressed the problem during the past years. However, a system that is robust and reliable enough for every day use by fire brigades, police, or emergency response teams has not been developed yet for small and lightweight multirotor UAVs with limited payload capabilities.

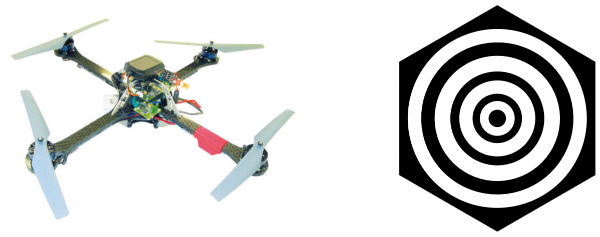
(a) UAV
(b) Landing target

Fig. 1. Our quadrotor UAV and the landing target.

In this paper, we describe several aspects of our work on quadrotor UAVs:

1) Design of a robustly recognizable landing target.
2) An efficient algorithm for the landing pad detection.
3) A sensor configuration suitable for velocity and position control when GPS signals are not available.
4) A cascaded controller structure for velocity and position stabilization.
5) We present results of several practical experiments that prove the capability of our system.

The first two points of the above list extend the preliminary results we reported earlier [3].

A. The Quadrocopter "Hummingbird"

The UAV we use in our project is a "Hummingbird" system (see Fig. 1(a)) that is manufactured by Ascending Technologies GmbH, Munich, Germany. These small four-rotor UAVs, or quadrocopters, can carry up to 200 g of payload for about 20 to 25 minutes. Measuring 53 cm in diameter, the Hummingbird's overall weight including LiPo batteries is 484 g.

The Hummingbird is propelled by four brushless DC motors and is equipped with a variety of sensors: Besides the usual accelerometers, gyros and a magnetic field sensor, a pressure sensor and a GPS module provide input for a sophisticated sensor fusion algorithm and the control loop

This work has been partially sponsored by the InnoProfile project GPSV (ID 03IP505) and supported by the chair for Circuit and System Design and the Chemnitz University Communication Testbed TUC-CTB.

running at 1 kHz. Especially outdoors where the control loop can make use of GPS signals to enter the GPS Position Hold Mode, the Hummingbird is absolutely self-stable and requires no human pilot to operate. The deviation from the commanded hover position is in most cases below 1 m. However, when GPS signals are not available, the UAV tends to begin drifting very quickly. Therefore, when the UAV should be operated in GPS-denied environments (for instance close to buildings or indoors) a position stabilization that is independed from GPS signals is required. More technical details on the UAV itself and the controllers can be found in [4].

We extended the standard configuration using the available payload and equipped the quadrocopter with additional hardware. This includes a Gumstix Verdex embedded computer system running at 600 MHz *(55 g)*, a Logitech QuickCam Pro 4000 USB camera *(25 g)*, a SRF10 sonar sensor *(10 g)*, a microcontroller board based on an ATmega644P *(25 g)*, an ADNS-3080 optical flow sensor board based on an ATmega644P *(20 g)* and an IEEE 802.15.4 radio module *(10 g)*. Completely equipped the quadrocopter has an overall weight of 679 g including LiPo batteries, frame and cables. For more details see section III.

B. Related Work

Different research groups around the world have been working on UAVs for the past years, most of them starting with helicopters or aircrafts. Recently however, multi-rotor UAVs became more and more popular both among researchers and hobbyists. (e.g. [5])

The problem of vision guided autonomous UAV landing has been addressed by several groups. [6] presents a real-time algorithm that identifies an "H"-shaped landing target using invariant moments. Another black and white pattern consisting of 6 squares of different sizes is used by [7] to land a helicopter. The system is described to be accurate to within 5 cm translation.

Two different approaches for safe landing site identification without explicit markers or landing pads are described in [8] and [9].

Moiré patterns are used by [10] to control a quadrotor UAV. This approach is technically appealing but does not seem to be robust enough to be feasible for outdoor use.

Most approaches for autonomous landing that use known patterns have one disadvantage that we tried to overcome in our work: The patterns have to be completely visible in order to be identified successfully. This is indeed a problem. If the targets are too small, they can not be identified from greater heights. If they are too big, they do not fit onto the camera image anymore, if the UAV is coming closer during its descend.

A similar sensor concept for UAV velocity and position control using optical flow sensors in scenarios where GPS is not available has previously described by [11] and [12].

The next section describes the layout of our landing pattern and the algorithm we used for its identification. A description of the overall system architecture and experimental results follow.

II. Landing by Vision

A. Landing Pad Design

Our goal was to overcome the disadvantages of the commonly used patterns and to create a target pattern that scales in a way that it can be identified both from great and small heights, that can be identified when parts of the target are not visible, and which is very unique. Due to its uniqueness it should be identified with high reliability in natural and man-made environments without risking misidentification (false positives). Besides that, the pattern was required to be simple enough to be easily identified by a vision algorithm running at a high frame rate.

Our pattern consists of several concentric white rings on a black background. Each of the white rings has a unique ratio of its inner to outer border radius. Therefore the rings can be uniquely identified.

Because the detection and identification of *one* ring is independent from the identification of all the other rings in the target, the overall target (which is treated as a composition of the individual rings) can be identified even when not all rings can be found or seen. This can occur for example, if the landing pad is viewed from a near distance or parts of the target are outside the camera image.

For our experiments we used a target design with four white rings as can be seen in Fig. 1(b). The outer ring has an outer diameter of 45 cm and its ratio of inner-to-outer radius is 85%. Continuing to the center the ratios are 75%, 65% and 50%. Depending on the later application these ratios and the number of rings can be adjusted. If, for instance, the target is required to be identified from a greater height, additional larger rings can be added.

B. An Efficient Pad Detection Algorithm

Our pad detection algorithm (Fig. 2) is programmed in C/C++ and based on OpenCV [13] to ensure fast and efficient image processing. Depending on the scene complexity, the algorithm runs with 70 to 100 Hz on a P4 at 2.4 GHz with an image resolution of 640×480. A Gumstix Verdex embedded computer board is still able to execute the algorithm based on low resolution images (320×240) with approximately 10 Hz, including image capture via USB 1.1.

The first step after capturing the image is the conversion into a greyscale and then into a binary image. Experiments showed that it is sufficient enough to use a fixed threshold during the binarization instead of using adaptive thresholding algorithms which would only induce more processing time. In order to find connected components, a segmentation and contour detection step follows and all objects smaller than a specific area are discarded to reduce computational costs. Furthermore the remaining components are rejected if they do not have exactly one hole inside, which is the most basic requirement for being a "ring". Afterwards, all surviving objects are candidate rings. We perform a roundness check as stated below:

$$o = \frac{4\pi A}{u^2} \tag{1}$$

where A stands for the area and u defines the contour length. The resulting roundness o is a value between 0 and 1 where 1 is the roundness of an ideal circle. All objects whose inner or outer contour are not found to be circles are discarded. The roundness threshold we use in this test was determined empirically and set to 0.82. Additionally we can check that the center of mass of the inner and outer contour are close together and within a small area. We can now be sure that all objects which passed these tests are rings from the target. As already mentioned in section II-A all rings can be identified uniquely. The ring number inside the target is determined with the help of outer-to-inner radius ratio c_r as given below:

$$c_r = \sqrt{\frac{A_{inner}}{A_{outer}}} \tag{2}$$

where A_{inner} and A_{outer} are the inner and outer areas of the ring. Here, the inner area is the area contained by the inner countour of the ring. The outer area is the inner area plus the area of the ring itself. Our experiments showed that even from extreme perspectives and viewpoints the ring number was continuously detected correctly.

Because we have knowledge of the intrinsic camera parameters, the size of the rings in the camera image, and their metric real size, we can approximate the height of the camera above the target. Because the UAV will always have a near parallel orientation relative to the landing target if the target lies on flat ground, we can assume that the UAV is flying *exactly* parallel to the landing pad and the ground plane.

We can then calculate the height from every visible and identified ring i as:

$$h_i = \frac{1}{2}\left(\frac{r_{i,outer}\,[cm]}{r_{i,outer}\,[pix]} + \frac{r_{i,inner}\,[cm]}{r_{i,inner}\,[pix]}\right)\alpha_x\,[pix] \tag{3}$$

where $r_{i,outer}$ ($r_{i,inner}$) is the outer (inner) radius of the i-th ring in cm (which is known a priori) or respectively in pixel (which is calculated from the ring area: $r\,[pix] = \sqrt{A/\pi}$).

The simplifying assumption that the UAV is always level proved to be sufficiently accurate in our experiments regarding the height estimation. However, regarding the relative position (x, y) of the UAV projected to the ground plane, this simplification is insufficient. Therefore, the current nick (pitch) and roll angles from the quadrocopter's internal data fusion are used to correct the position estimate.

We conducted a set of experiments to prove the high accuracy of the landing pad position estimation process. These experiments showed the error between estimated and true position to be below 5 cm in all tested cases. The detailed results were reported in an earlier paper [3].

III. System Architecture

The vision algorithm described in the last section is only one part of the overall system. This section describes the complete system architecture and how the different parts work together.

A. Additional Hardware and Sensors

We equipped the quadrocopter with additional hardware and sensors (see Fig. 3). A Gumstix Verdex embedded computer system (running at 600 MHz with Linux as operating system) interfaces a Logitech QuickCam Pro 4000 USB camera and executes the vision algorithm. The camera was chosen because it is lightweight, has an M12 lens mount and is equipped with a global shutter. Tests run with different cameras revealed that rolling shutter cameras are not suitable for this kind of UAV. Due to the high dynamics and the possible rapid rotary motions of the flying quadrocopter, cameras with a rolling shutter suffer from large motion-induced image distortions and image blur. For additional improvement of the landing pad detection from low height, we replaced the lens supplied with the Logitech QuickCam Pro 4000 by a $f = 2.1mm$ lens.

To ensure realtime execution of the controllers, they are implemented on a custom made microcontroller board based on an ATmega644P. It is connected to the quadrocopter via USART for polling the quadrocopter's internal sensor readings and sending flight commands. The second USART of the microcontroller is connected to an IEEE 802.15.4 radio module (XBeePro) for transmitting and receiving status messages and commands from a ground station. These small, yet powerful modules act as a transparent serial interface and allow amongst others the adjustment of the controller parameters and the forwarding of messages to the Gumstix computer system or the UAV.

Several sensors used in the controller architecture are linked to the microcontroller via I^{2}C bus, acting as slave devices. A SRF10 sonar sensor measures the current altitude over ground with high precision. Furthermore the Avago ADNS-3080 optical flow sensor combined with a $f = 4.2mm$ lens is linked to determine the UAV's current velocity over ground. The sensor is not connected directly over the I^{2}C bus, but is interfaced via SPI to another ATmega644P which is acting as an I^{2}C slave device. In addition, the Gumstix computer system is connected to the main microcontroller via I^{2}C to provide vision processing results from the pad detection algorithm. In order to get the Gumstix computer working as a I^{2}C slave device it was necessary to add slave cabability to the linux *i2c-pxa* kernel driver.

B. Controller Structure

In order to achieve a stable flight behavior in GPS-denied areas, we used an altitude controller and a cascaded controller structure for position stabilization. Figure 4 shows the principal structure along with the main sensors used by the position and velocity controllers.

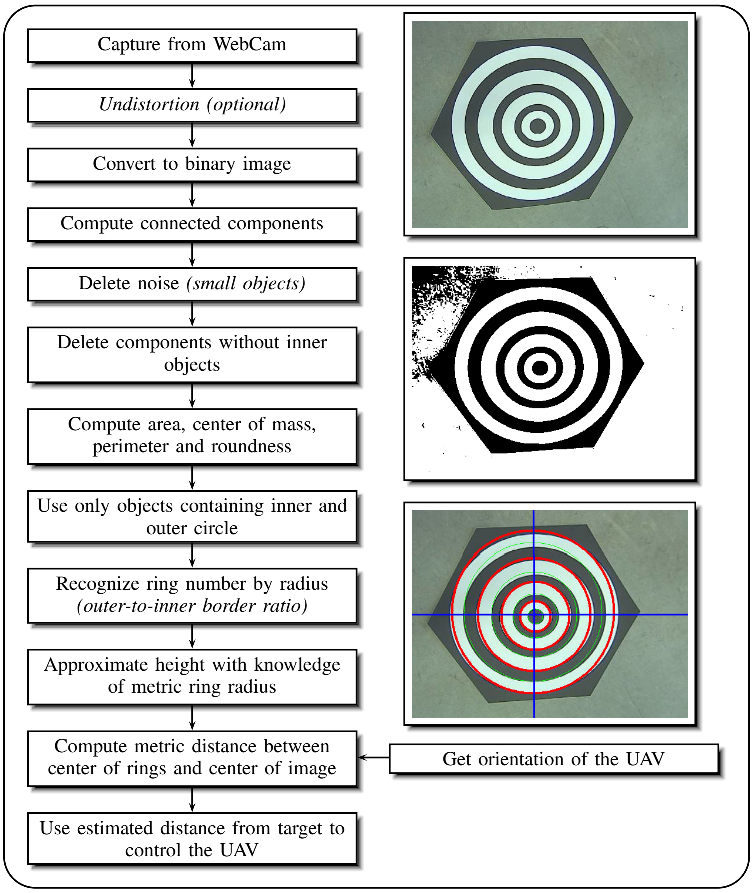
Capture from WebCam
Undistortion (optional)
Convert to binary image
Compute connected components
Delete noise (small objects)
Delete components without inner objects
Compute area, center of mass, perimeter and roundness
Use only objects containing inner and outer circle
Recognize ring number by radius (outer-to-inner border ratio)
Approximate height with knowledge of metric ring radius
Compute metric distance between center of rings and center of image
Get orientation of the UAV
Use estimated distance from target to control the UAV

Fig. 2. Algorithm for landing pad detection

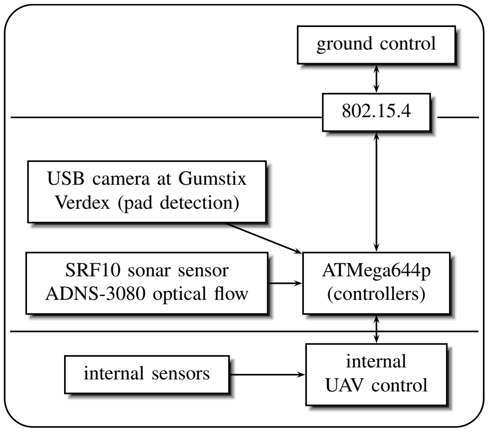

Fig. 3. General system architecture.

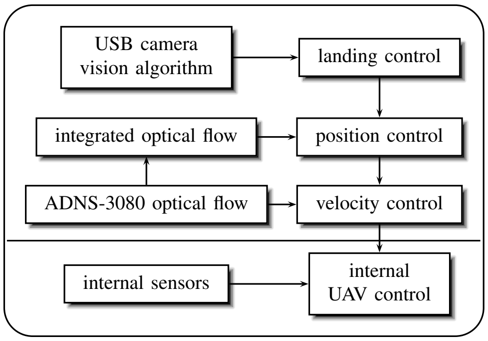

Fig. 4. Cascaded controller structure and main sensors used. For reasons of clarity the altitude controller is not shown here.

1) Altitude Controller: The altitude controller receives its inputs from the SRF10 sonar sensor. This off-the-shelf sensor commonly used in robotics projects provides accurate altitude information and offers considerable advantages because of its short minimum operation range of 3 cm. The controller itself is implemented on the ATmega644P microcontroller board as a standard PID-controller. It is executed with 25 Hz and able to stabilize the UAV's altitude with 3 cm accuracy (see section IV-A for experimental results).

The standard PID altitude controller is very sensitive to steps in its measurements or setpoint values, which are common when flying over obstacles. If they appear, an overshoot is the common reaction of the quadrocopter. For that reason we implemented a step detection combined with a ramp function for both cases.

2) Position and Velocity Controller: When no GPS signals are available, the only way to measure position and velocity is to integrate the information from the onboard acceleration sensors and gyros. However, due to the noisy input signals large errors accumulate quickly, rendering this procedure useless for any velocity or even position control.

In our approach, an optical flow sensor facing the ground provides information on the current velocity and position of the UAV. The Avago ADNS-3080 we use is commonly found in optical mice and calculates its own movements based on optical flow information with high accuracy and a framerate of up to 6400 fps. After exchanging the optics and attaching a M12-mount lens with $f = 4.2mm$ to the sensor, we are able to retrieve high quality position and velocity signals that accumulate only small errors during the flight. The sensor also provides a quality feedback which correlates with the number of texture features on the surface the sensor is facing. If the surface is not textured enough, the quality indicator drops and the velocity and shift signals become more noisy. Further investigations have to show which of the commonly encountered surface types in indoor and outdoor scenarios provide sufficient texture information. Especially a lens with a larger focal length may be useful over low-textured indoor surfaces.

Both the velocity and position controller are implemented as standard controllers (PD and PID respectively) and executed with 25 Hz. In combination, they are able to stabilize the UAV's position and prevent the drift that quickly occurs when the GPS-Position-Hold mode is inactive (see Fig. 7).

3) Landing Controller: The OpenCV-based software on the Gumstix embedded computer receives the images from

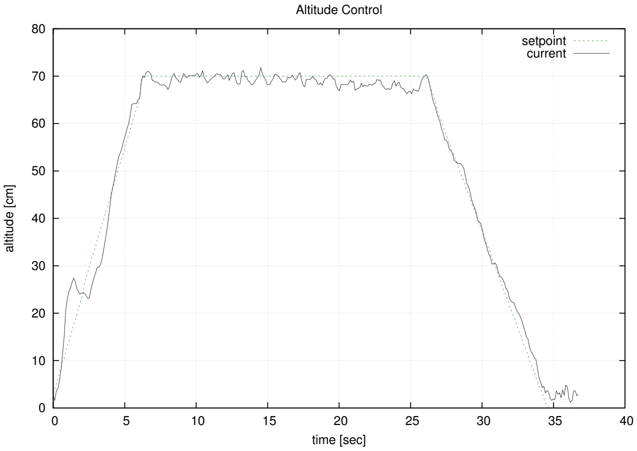

Fig. 5. Altitude plot showing a take-off, altitude-hold, and landing maneuver. The altitude setpoint is automatically increased and decreased in a ramp. The PID-controller is able to follow the setpoint very closely and stabilizes the UAV at the commanded setpoint with 3 cm accuracy.

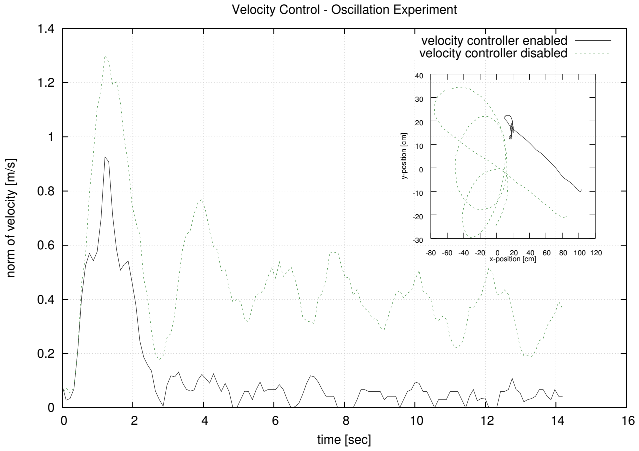

Fig. 6. Pendulum experiment to show the performance of the underlying velocity controller. The diagram shows the norm of the velocity and the integrated position measurements based upon the optical flow sensor.

the onboard USB camera and processes it as was described in II-B. The complete algorithm including image capture via the slow USB 1.1 interface runs at approximately 10 Hz. The results of this processing step are the estimated height z above ground and the position (x, y) of the UAV relative to the landing pad, projected to the ground plane. These estimates (especially the translation (x, y)) need to be corrected for the current nick and roll angles of the UAV. This is necessary because the camera is fixed on the frame of the UAV and is not tilt-compensated in any way. The initial position estimates (x, y, z) are corrected using the current nick and roll angles Θ and Φ. These corrected position estimates are then used as inputs for a PID-controller that generates the necessary motion commands to keep the UAV steady above the center of the landing pad. This controller is again executed on the ATmega644P microcontroller.

IV. Experiments and Results

A. Altitude Controller

In this section we present the results of our altitude controller in form of a simple experiment. Standing at ground level, a setpoint of 70 cm was given. After reaching this height, the setpoint remained unchanged for about 20 seconds followed by a setpoint of 0 cm to perform a landing. The resulting sonar measurements are shown in Figure 5. A setpoint ramp is implemented to avoid considerable overshoot when a step function is used as controller input. As a result the control deviation is marginal over the whole experiment and has an error of about 3 cm. All control parameters are based upon empirical trials and can not be assumed to be perfect.

B. Velocity Controller

To control the horizontal position of the quadrocopter, we used the cascaded control structure we described in section III-B. The quality of the underlying velocity controller will be demonstrated through a simple pendulum experiment. For this experiment the quadrocopter was attached to a rope hanging downwards from the ceiling. The UAV was then manually deflected horizontally by about 1 m. The resulting motion without using the velocity control is a slightly damped oscillation with change to rotary motion (dashed line in Fig. 6). We repeated the same experiment with activated velocity control with the result of a dramatically damped oscillation. The results are illustrated in Fig. 6.

C. Position Controller

In another experiment we tested the performance of the position controller: The UAV was commanded to hold its position for about 5 minutes. As we laid out before, the position controller works on data obtained by the optical flow sensor and sends commands to the velocity controller in order to keep the commanded position. Without this position-hold control loop, the UAV would immediately begin to rapidly drift away.

In order to evaluate the performance of the controller, we measured the ground truth position of the UAV with the help of a calibrated wide-angular camera that looked down from the ceiling of our lab. The UAV could be automatically identified in the image and together with the known altitude, the true 2D position above the ground plane could be retrieved.

The results of the experiment are presented in Fig. 7. The plot shows the ground truth position of the UAV and the position estimated by the onboard optical flow sensor during the flight over 5 minutes. As can be seen, the UAV's ground truth position starts to drift slightly during the flight, while the position estimated by the onboard optical flow sensor remains centered at the origin. Due to this drift over 5 minutes, the maximum deviation from the origin was 28.7 cm. We consider this value to be very good, given that the controller works solely on integrated optical flow information.

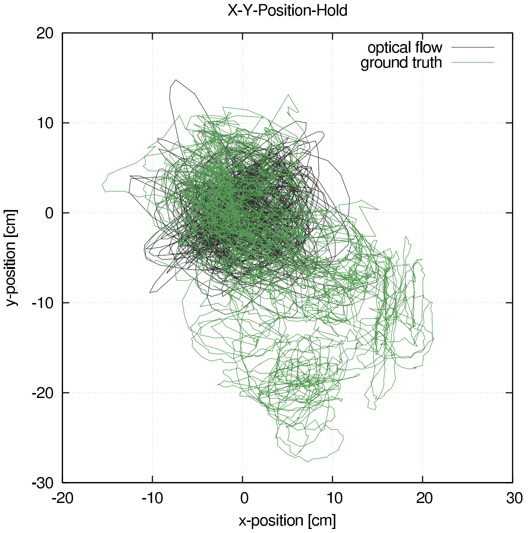

Fig. 7. Performance of the optical flow based position controller: Position estimated by the internal optical flow sensor and ground truth position of the UAV during 5 minutes of flight. The UAV tried to stabilize its position at the origin using the position and velocity controller. The landing pad detection was not used in this experiment.

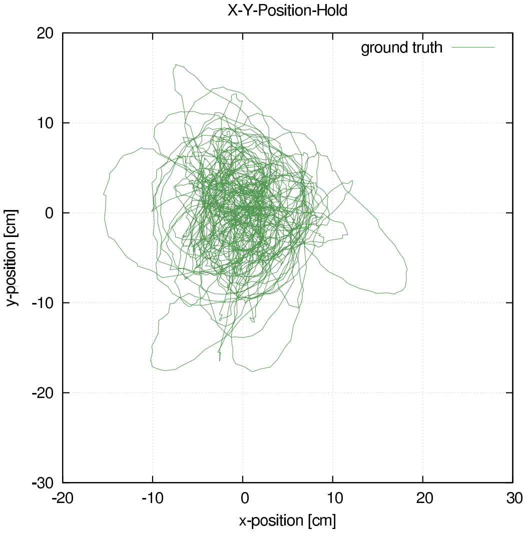

Fig. 8. Performance of the position controller combined with the landing pad detection control loop: The plot shows the 2D ground truth position of the UAV during 5 minutes of flight. The UAV tried to stabilize its position directly over the landing pad center located at the origin of the plot.

D. Landing Controller

To show the overall system performance, we tested the velocity and position controller together with the landing pad detection and control loop. As the landing pad detection provides a position measurement in the global reference frame, one would expect the slow drift that was visible in the experiment described above to be eliminated. Indeed, with the landing pad detection measuring the UAV's position relative to the pad center, the position controller was now able to keep the quadrocopter centered above the landing pad with a standard deviation of 3.8 cm and a maximum deviation of 23 cm over 5 minutes. Fig. 8 shows the ground truth position of the UAV during this experiment.

V. Conclusions and Futher Work

We successfully extended our previous work [3] and implemented the proposed vision algorithm on board the UAV. Our platform is now able to stabilize its altitude, position and velocity in the absence of GPS signals (e.g. in indoor scenarios, close to buildings) and can detect and approach the landing pad marker. The feasibility of the landing pad design and the vision algorithm has been proved in practical experiments, as well as the overall system architecture and the accuracy and performance of the cascaded controller structure.

In future work we will have to examine how good the optical flow sensor copes with different surface textures (e.g. grass, concrete, indoor floor covering) and how the quality of motion estimation is influenced by different optic setups. Furthermore, a sensor fusion between optical flow, GPS and IMU information can help to improve the overall stability of the system and increase the robustness against otherwise critical sensor failures.

References

[1] T. Krause and P. Protzel, "Verteiltes, dynamisches Antriebssystem zur Steuerung eines Luftschiffes," in *Tagungsband 52. Internationales Wissenschaftliches Kolloquium*, Ilmenau, Germany, September 2007.

[2] N. Sünderhauf, S. Lange, and P. Protzel, "Using the Unscented Kalman Filter in Mono-SLAM with Inverse Depth Parametrization for Autonomous Airship Control," in *Proc. of IEEE International Workshop on Safety Security and Rescue Robotics, SSRR 2007*, Rome, Italy, 2007.

[3] S. Lange, N. Sünderhauf, and P. Protzel, "Autonomous Landing for a Multirotor UAV Using Vision," in *Workshop Proc. of SIMPAR 2008 Intl. Conf. on Simulation, Modeling and Programming for Autonomous Robots*, Venice, Italy, November 2008, pp. 482–491.

[4] 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 IEEE International Conference on Robotics and Automation, ICRA07*, Rome, Italy, April 2007.

[5] P. Pounds, R. Maloy, P. Hynes, and J. Roberts, "Design of a Four-Rotor Aerial Robot," in *Proc. of Australasian Conference on Robotics and Automation*, Auckland, New Zealand, November 2002.

[6] S. Saripalli, J. F. Montgomery, and G. S. Sukhatme, "Vision-Based Autonomous Landing of an Unmanned Aerial Vehicle," in *IEEE International Conference on Robotics and Automation (ICRA*, 2002, pp. 2799–2804.

[7] C. S. Sharp, O. Shakernia, and S. S. Sastry, "A Vision System for Landing an Unmanned Aerial Vehicle," in *IEEE International Conference on Robotics and Automation (ICRA), Seoul, Korea*, 2001, pp. 1720–1727.

[8] P. J. Garcia-Pardo, G. S. Sukhatme, and J. F. Montgomery, "Towards vision-based safe landing for an autonomous helicopter," *Robotics and Autonomous Systems*, vol. 38, no. 1, pp. 19–29, 2001.

[9] S. Bosch, S. Lacroix, and F. Caballero, "Autonomous Detection of Safe Landing Areas for an UAV from Monocular Images," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing (China)*, 2006.

[10] G. P. Tournier, M. Valenti, J. P. How, and E. Feron, "Estimation and Control of a Quadrotor Vehicle Using Monocular Vision and Moiré Patterns," in *In AIAA Guidance, Navigation and Control Conference*. AIAA, 2006, pp. 2006–6711.

[11] N. G. Johnson, "Vision-Assisted Control of a Hovering Air Vehicle in an Indoor Setting," Master's thesis, Brigham Young University, 2008.

[12] J. Kim and G. Brambley, "Dual Optic-flow Integrated Navigation for Small-scale Flying Robots," in *Proc. of Australasian Conference on Robotics and Automation*, Brisbane, Australia, December 2007.

[13] "The OpenCV Library," `http://opencvlibrary.sf.net`.