Autonomous Landing for a Multirotor UAV Using Vision

Sven Lange, Niko Sünderhauf, and Peter Protzel

Chemnitz University of Technology, 09126 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. The paper describes the design of our landing pad and its advantages. We explain how the landing pad detection algorithm works and how the 3D-position of the UAV relative to the landing pad is calculated. Practical experiments prove the quality of these estimations.

1 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 [5], [9]. 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, multi-rotor 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) with the Hummingbird and its AscTec-Auto-Pilot system 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.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 482-491

1.1 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, München, 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 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. More technical details on the UAV itself and the controllers can be found in [4].

Extending the standard configuration, we equipped the quadrocopter with an Axis 207MW wifi camera. This CMOS camera streams Motion-JPEG or MPEG4 live video with a resolution of up to 1280×1024 pixel using its IEEE 802.11g or Ethernet interface. In its original configuration the camera weights 94 g. However, this weight can be reduced to 68 g by removing a dispensable metal plate on its back. Its resulting low weight allows the camera to be carried along with other payload, like a servo for tilting the camera, and a radio module on the UAV.

An IEEE 802.15.4 radio module (XBeePro) is used to transmit status messages and motion commands from and to the UAV. These small, yet powerful modules act as a transparent serial interface.

1.2 Autonomous Landing and 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. [6])

The problem of vision guided autonomous UAV landing has been addressed by several groups. [7] presents a real-time algorithm that identifies an "H"-shaped landing target using invariant moments. Another black and wh ite pattern consisting of 6 squares of different sizes is used by [8] 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 [3] and [2].

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

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 482-491

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.

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.

2 Landing by Vision

2.1 Landing Pad

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.

2.2 Algorithm Pad Detection

Our pad detection algorithm (Fig. 2) is programmed in C/C++ and based on OpenCV [1] 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.

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

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 482-491

(a) UAV

(b) Landing target

Fig. 1. Landing target

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 2.1 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

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 482-491

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. We need the current orientation information from the quadrocopter to get accurate results. Therefore we use the IEEE 802.15.4 radio module to communicate with the onboard sensor fusion software to get the current nick and roll angles. To verify our approach, we conducted a set of experiments. The results of these experiments can be seen in section 2.4.

2.3 Software and System Architecture

In the current configuration the UAV needs a ground station, e.g. a laptop, that receives and processes the images from the onboard camera, runs the PID control loop and generates the necessary motion commands. Figure 3 shows the general system and software architecture and the flow of information.

The onboard wifi-camera streams live video footage via IEEE 802.11g. The OpenCV-based software on the ground receives this stream and processes it as described above.

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.

A second process (written in Python) on the ground station laptop communicates with the UAV using the XBeePro radio modules (IEEE 802.15.4) as transparent serial interfaces. The communication protocol allows the retrieval of sensor data and internal status messages from the Hummingbird quadrocopter. The same process receives the position estimation results (x, y, z) over a named pipe from the OpenCV process and corrects these estimates using the current nick and roll angles Θ and Φ.

The 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. These motion commands are again communicated to the UAV using the same XBeePro radio module that was used to retrieve the sensor data from the UAV.

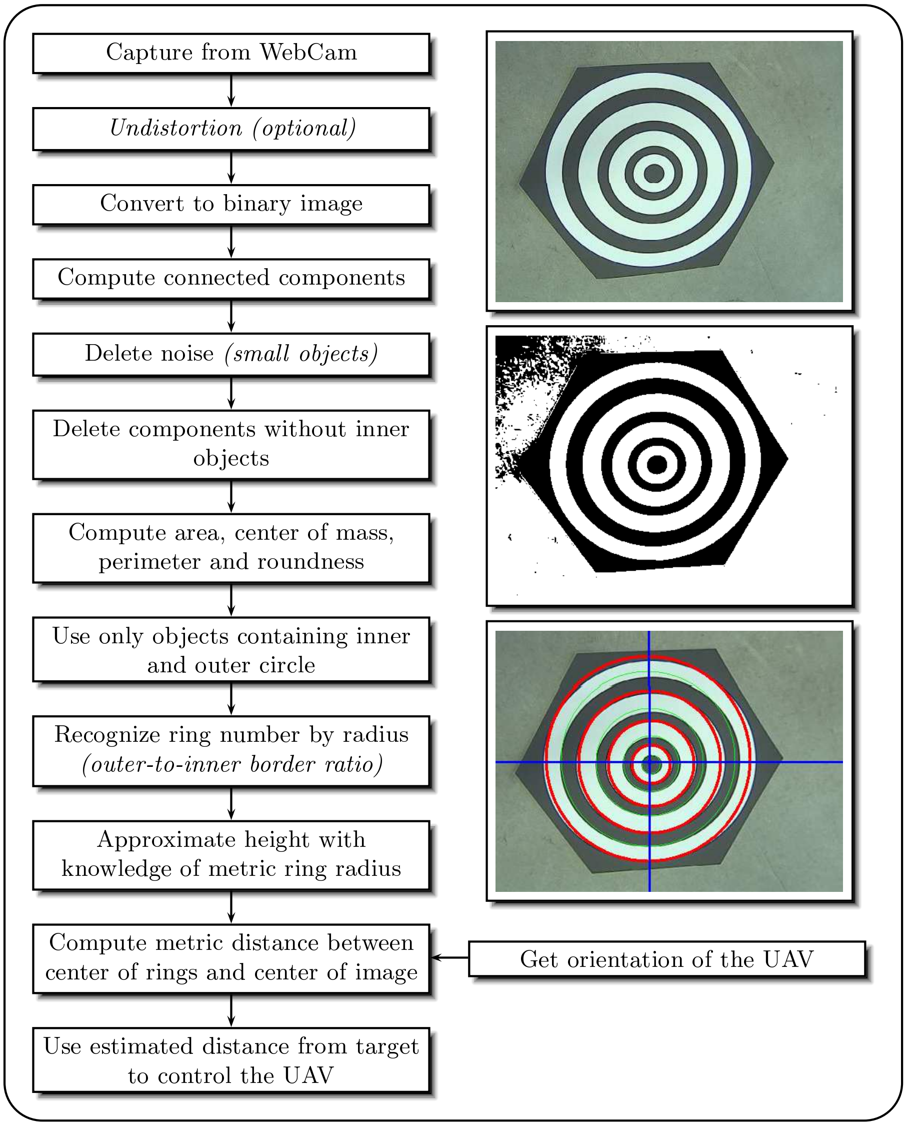


Fig. 2. Algorithm for landing pad detection

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 482-491

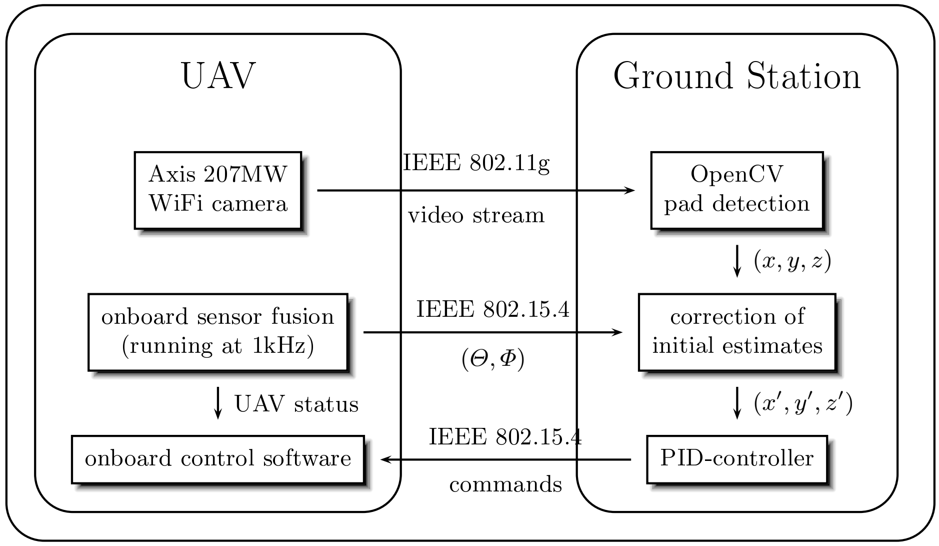


Fig. 3. System architecture and flow of information

2.4 Results

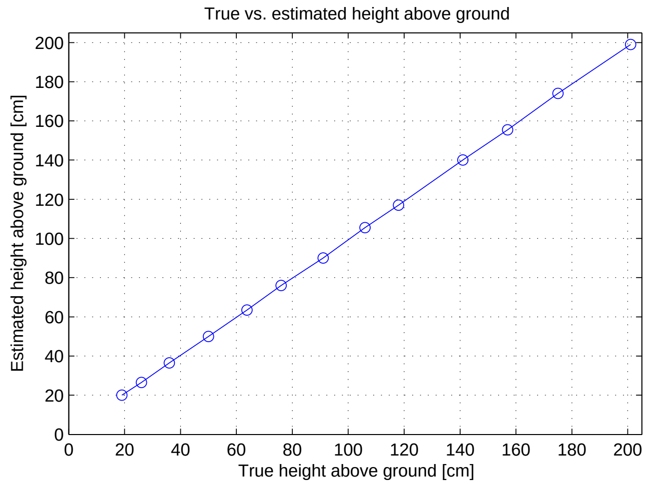
We conducted a number of experiments to prove the quality of our pad detection and position estimation algorithm and to show that our approach is applicable for autonomous landing.

The visual detection algorithm estimates both the height above the landing pad and the translation in the ground plane. This process was described in the previous section. The precision of both estimates was reviewed in a series of experiments.

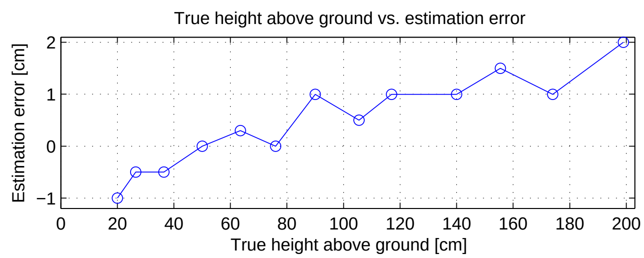
Figure 4(a) shows the real height above ground vs. the estimated one. In the ideal case where no errors occurred and the estimated height equals the true height exactly, a straight line with slope 1 would be shown in the diagram. The very small deviation from this ideal result clearly proves the very good quality of the height estimation process. Diagram 4(b) shows the deviation from the ideal case and a maximum error of 2 cm. Note that during the tests the UAV was fixed in a horizontal position. The error increases up to 5 cm with increasing nick and roll angles.

In another experiment we investigated the accuracy of the position estimation. The quadrocopter was fixed in different heights above the ground and the landing pad was moved relative to the UAV. This way, the true position of the UAV with respect to the center of the landing pad in the ground plane was exactly known. Figures 5 and 6 show the results of this experiment for different heights and different distances. Again, the diagrams plot the true vs. the estimated distance of the UAV from the center of the pad and the corresponding

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 482-491



(a)



(b)

Fig. 4. True vs. estimated height above ground and corresponding errors

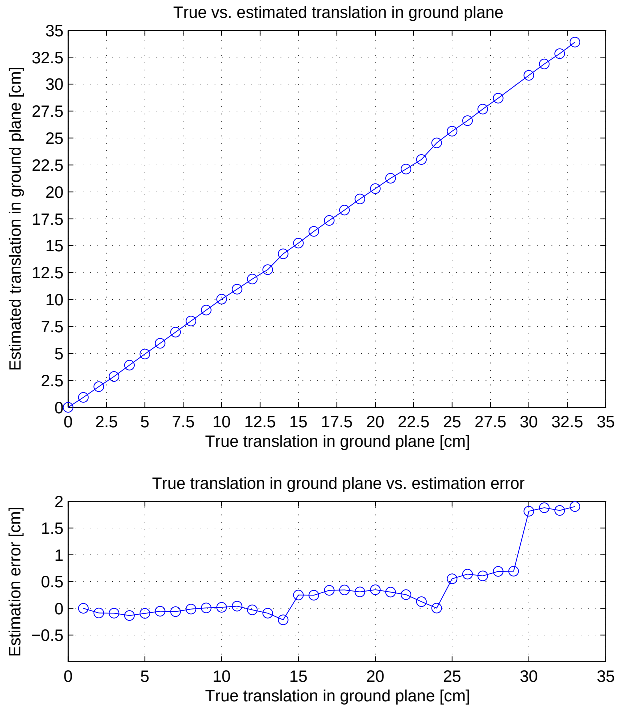
errors. In the ideal case a straight line with a slope equal of 1 would be seen. The real data show only little deviation from this ideal case which proves the very high accuracy. The maximum position error we found in our experiments was well below 4 cm or 3% of the height above ground.

3 Conclusions and Further Work

We described our algorithm for robust and reliable landing pad detection to be used for autonomous UAV landing. The algorithm estimates the 3D-position of the UAV relative to the landing pad using the known real world dimensions of the marker and the intrinsic camera parameters.

Our experiments proved both the high efficiency and accuracy of the detection and position estimation algorithm and that the landing target can be identified from different heights in real time. Based on our vision system we are currently parameterizing the controller structure for first autonomous landing tests with our quadrocopter.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 482-491



(b)

Fig. 5. True vs. estimated translation in ground plane and corresponding errors

In future work, we want to port the algorithm to an embedded computer running at 600 MHz (Gumstix Verdex). This tiny board can operate directly onboard the UAV which would supersede the dedicated ground station laptop and the IEEE 802.15.4 radio transmission.

Acknowledgement 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 TU Chemnitz Communication Testbed TUC-CTB.

References

1. The OpenCV Library. `http://opencvlibrary.sf.net`.
2. Bosch, S., Lacroix, S., and Caballero, F. 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).
3. Garcia-Pardo, P. J., Sukhatme, G. S., and Montgomery, J. F. Towards vision-based safe landing for an autonomous helicopter. *Robotics and Autonomous Systems 38*, 1 (2001), 19–29.
4. Gurdan, D., Stumpf, J., Achtelik, M., Doth, K.-M., Hirzinger, G., and Rus, D. Energy-efficient autonomous four-rotor flying robot controlled at 1 khz.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 482-491

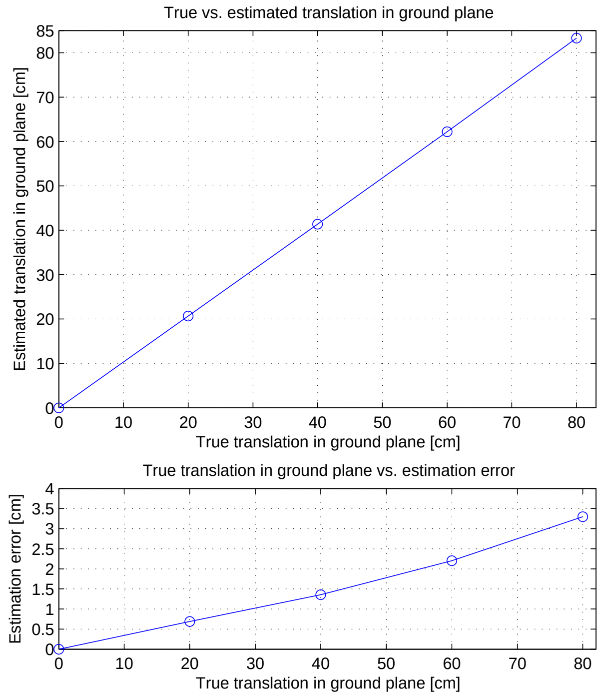


Fig. 6. True vs. estimated translation in ground plane and related error

In *Proc. of IEEE International Conference on Robotics and Automation, ICRA07* (Rome, Italy, April 2007).
5. Krause, T., and Protzel, P. Verteiltes, dynamisches Antriebssystem zur Steuerung eines Luftschiffes. In *Tagungsband 52. Internationales Wissenschaftliches Kolloquium* (Ilmenau, Germany, September 2007).
6. Pounds, P., Maloy, R., Hynes, P., and Roberts, J. Design of a four-rotor aerial robot. In *Proc. of Australasian Conference on Robotics and Automation* (Auckland, New Zealand, November 2002).
7. Saripalli, S., Montgomery, J. F., and Sukhatme, G. S. Vision-based autonomous landing of an unmanned aerial vehicle. In *IEEE International Conference on Robotics and Automation (ICRA* (2002), pp. 2799–2804.
8. Sharp, C. S., Shakernia, O., and Sastry, S. S. A vision system for landing an unmanned aerial vehicle. In *IEEE International Conference on Robotics and Automation (ICRA), Seoul, Korea* (2001), pp. 1720–1727.
9. Sünderhauf, N., Lange, S., and Protzel, P. 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).
10. Tournier, G. P., Valenti, M., How, J. P., and Feron, E. Estimation and control of a quadrotor vehicle using monocular vision and moir patterns. In *In AIAA Guidance, Navigation and Control Conference* (2006), AIAA, pp. 2006–6711.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 482-491