

A Novel Marker Based Tracking Method for Position and Attitude Control of MAVs

A. Masselli and A. Zell

Abstract

In this paper we present a novel method for pose estimation for micro aerial vehicles (MAVs), which provides all six degrees of freedom (6DOF) and runs completely onboard and in real time at 60Hz. The approach uses a distinct pattern of orange table tennis balls as passive visual markers and a monocular color camera mounted onto the MAV as the only sensor. We show that our method can be used as input of position and attitude controllers for autonomous take-off, hovering and landing of MAVs. As a testing platform we choose an AscTec Hummingbird quadcopter, which we equipped with a Gumstix processor, running at 600MHz. The quadcopter is fully controlled by our method, which acts as input to four independent, classical proportional-integral-derivative (PID) controllers.

1. INTRODUCTION

During the last decade it became possible to shrink the size of unmanned aerial vehicles (UAV) further and further. Yet the problem is to provide these so called micro aerial vehicles (MAVs) with autonomy based on the limited hardware, which is itself constrained by the limited payload of such small UAVs. In many current research projects the MAVs still rely on external sensors or offboard processing, especially for estimating the MAV's pose. The major goal is to overcome this lack of autonomy and enable MAVs to localize themselves within the environment using only onboard hardware. In this paper we present a method for self localization in all six degrees of freedom based on passive visual markers which runs completely onboard. Marker based methods are still common, even for systems with external sensors [8]. We show the method's feasibility for autonomous flights of MAVs by having a quadcopter perform flights consisting of take-off, hovering and landing. This paper is structured as follows: We give an overview of work related to this paper in section 2 and describe the used hardware in section 3. In section 4 our new method is presented in detail. We describe the experiments and show their results in section 5 and discuss these results in section 6.

Andreas Masselli is a PhD student with the Chair of Cognitive Systems at the Faculty of Science, University of Tuebingen, Tuebingen, Germany
andreas.masselli@uni-tuebingen.de
Andreas Zell is full professor of Cognitive Systems at the Faculty of Science, University of Tuebingen, Tuebingen, Germany
andreas.zell@uni-tuebingen.de

2. RELATED WORK

Mellinger et al. [6] demonstrate a system which enables quadcopters to maneuver fast and accurate within a delimited indoor area, which is equipped with an external pose estimation system based on infrared (IR) markers on the MAVs. The evaluation of the pose data and calculation of the controller output is also done off-board. The output is subsequently sent to the quadcopter wirelessly.

Another marker based system is presented in Wenzel et al. [11]. Here the difference is that all sensing and calculation is done on-board of the MAV. However, the system relies on active IR-LED markers and is therefore limited to indoor or night use. The same method has shown to be feasible for starting and landing a quadcopter on a moving platform [12] as well as hovering above a person [13].

Since estimating the pose of an MAV is especially important during landing, much work is found which focuses on landing an MAV. Herisse et al. [4] demonstrate an MAV which used optical flow to hover above and land on a large textured platform which moves horizontally and vertically. Predefined targets have also been used in [14][10][7][5]:

A "T"-shaped marker object is also used in Xu et al. [14] to land an MAV on a ship deck, but only four degrees of freedom are determined. In Saripalli et al. [10] a landing pad with a typical "H"-Letter is used. This approach also relies on a differential GPS and is therefore bound to fail in an indoor environment. Merz et al. [7] use a landing pad with circle markers to estimate the pose of a helicopter during landing. In Lange et al. [5] the landing pad consists of several concentric circles. While being able to autonomously control the position of an MAV and land on the pad, the approach shows a systematic error, because it is assumed that the MAV is flying exactly parallel to the landing pad.

Other work focus on visually finding a suitable area for landing [3][1].

3. HARDWARE PLATFORM

All experiments were made with a *Hummingbird Scientific* quadcopter made by Ascending Technologies GmbH (*AscTec*) from Munich, Germany (see Fig. 1). Additionally we equipped the quadcopter with a color camera and a single-board computer, which processes the images, calculates the pose and controls the quadcopter via a serial interface in order to maintain a constant

position and yaw. The control loop comprises four independent, classical proportional-integral-derivative (PID) controllers for roll, pitch, yaw and thrust. Low level controlling of the brushless motors is done on the internal board of the *AscTec* quadcopter.

The single-board computer is a *Gumstix Overo Fire*, hosting an ARM Cortex-A8 CPU with 600MHz and 256 MB RAM. The camera is a *PointGrey FireFly USB 2.0* color camera with global shutter, which captures images with VGA resolution (640×480) at 60 fps. With our lens setup it has an opening angle of 48 degrees. It is mounted vertically to maximize the vertical opening angle during flight. This way we can ensure the marker pattern remains visible for the camera. We need to have a large vertical opening angle because we cannot influence the pitch angle of the quadcopter, since pitching always has an effect on the quadcopter’s horizontal acceleration, therefore being reserved for controlling the quadcopter in order to maintain the desired position. In contrast we accept a narrower horizontal opening angle, since the quadcopter is able to maintain a desired yaw angle in order to point the camera towards the pattern.

Including all components the quadcopter has a total weight of 570 g.



Figure 1: Tests were done on an AscTec Hummingbird quadcopter.

4. POSE ESTIMATION

The position estimation method presented in this paper uses four orange table tennis balls as passive markers, which are mounted on a black piece of styrofoam in a distinct constellation (see Fig. 2). All markers lie in one plane to avoid cases where the pose estimate becomes ambiguous. The method consists of two steps: First the markers are detected within the camera image. Second the pose is calculated from the pixel positions of the detected markers. The algorithm has been optimized for low performance hardware, which is usually found on MAVs due to weight limits. We arranged to process all of the 60 frames coming from the camera in real-time on the ARM processor.

4.1. Marker Detection

Detecting the orange ball markers is done by first applying color segmentation. To achieve fast processing, a

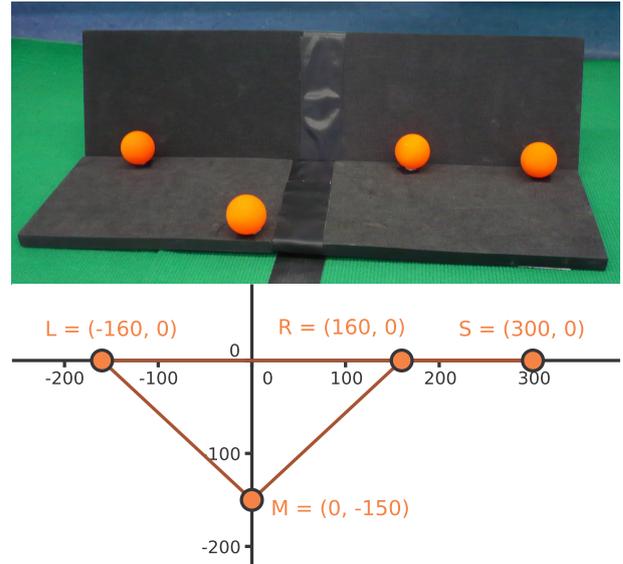


Figure 2: The passive marker pattern seen by the camera. Coordinates in mm.

lookup table is used. Furthermore the algorithm operates on the raw image rather than debayering the whole image. Instead, the pixel’s color is only evaluated when being tested as a candidate for a ball pixel. This test is not applied to all pixels in the image, since we assume each ball marker to be at least 3 pixels big in radius in the image. Therefore the image is scanned in a 4×4 grid. For pixels with a color matching with the lookup table a floodfill algorithm is applied, filling each orange pixel within the surrounding region and incrementally determining the region’s bounding box. This strategy speeds up the color segmentation by almost 16 compared to scanning the whole image, since the markers appear rarely in the image. The next step excludes those regions having bounding boxes either too small or with impossible aspect ratios. The contour of the remaining regions is determined and tested for being similar to a circle with the *Randomized Hough Transform (RHT)*. Finally the four largest regions are considered to be the ball markers of the pattern. The centers of their bounding boxes are passed to the pose estimation. Their size is too small to provide usable distance information.

Figure 3 shows the different steps for detecting the table tennis ball markers:

4.2. Retrieving 6DOF Pose

Using the image coordinates from the ball detection described above, there is enough information to calculate the full pose of the camera relative to the pattern. The problem that has to be solved in this stage is well-known in literature as the *Perspective-4-Point Problem (P4P)*, in general *PnP* for n points: Suppose n points are given in world coordinates and their corresponding projections onto the image plane of a camera are given in pixel coordinates. The set of camera poses meeting the given constraints is to be computed. If $n = 4$, as it is in our case with four

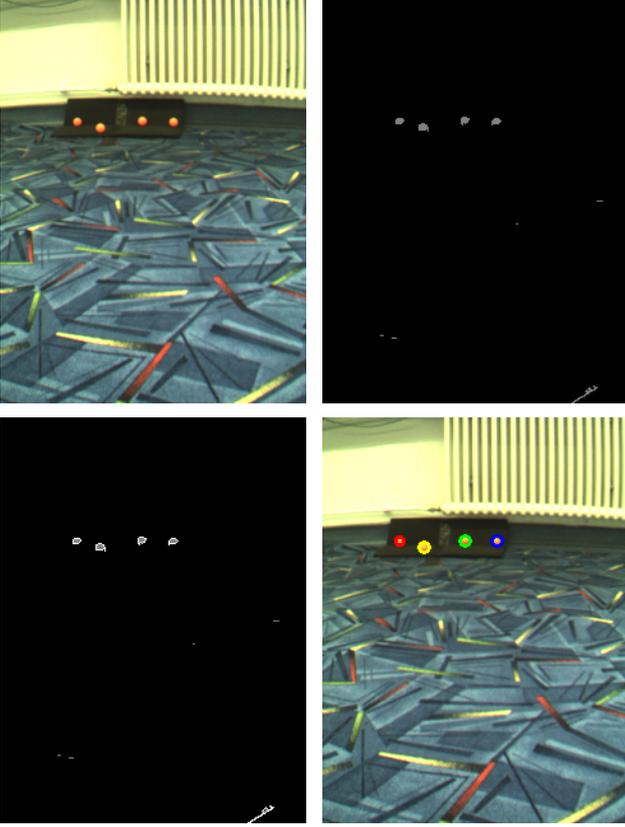


Figure 3: Detection of the table tennis ball pattern: Upper left: Input image from the camera Upper right: Segmented regions. Lower left: Output of the contour following algorithm. Lower right: Successful detection.

markers, then in general there is only one unambiguous solution [2]. Before solving the $P4P$, we have to find the correspondences between world and image points. This can easily be done by assuming the quadcopter is neither flying behind the marker pattern nor upside-down. Then we can sort the image points along their y -coordinate (recall the camera is vertically mounted) and identify them as L_i , M_i , R_i and S_i . We then concentrate on L , M , R , solve the $P3P$ and add S to verify the found solution. In general, $P3P$ yields up to four solutions [2]. Considering the limited operating range of the quadcopter a second time, we can exclude three of these solutions and therefore estimate the full pose with only three markers. However, we later consider the fourth marker to reject false detections that come from the former segmentation stage and attach a confidence to our estimate.

We solve the $P3P$ with the following, self-developed method: Having the three world points L , M and R and their corresponding image points L_i , M_i and R_i , we will determine the position P of the camera in the world. If needed, we can later infer its orientation.

First, we define a right-handed intermediate coordinate system as follows (see Fig. 4):

Let the origin be the middle of L and R , let the y -axis go through L and R pointing towards R , and let the x, y -plane go through L , R and P with the x -axis being directed such

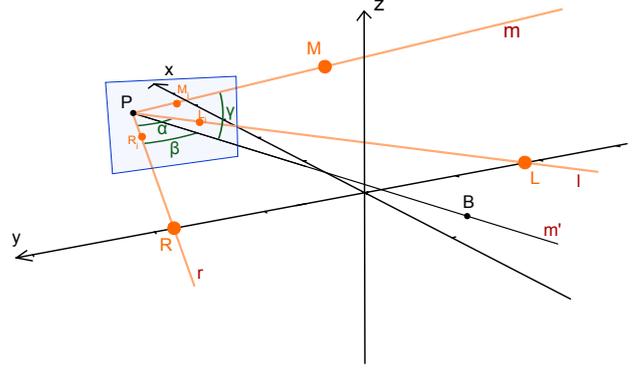


Figure 4: Scheme of the setup of the $P3P$ problem.

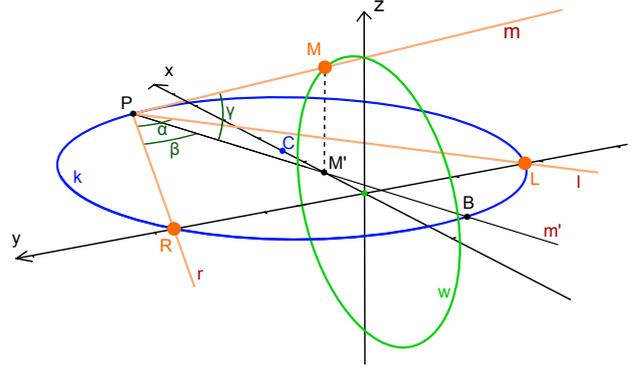


Figure 5: Solving the $P3P$ problem.

that $P_x \geq 0$.

We then define the light rays l , m , r going through L_i , M_i and R_i and meeting in P w.r.t. camera coordinates, using a standard pinhole camera model. Note that these rays also go through the respective world points L , M and R . Further, we define the ray m' , being the projection of m onto the x, y -plane. We calculate the angles $\alpha = \angle(l, r)$, $\beta = \angle(m', r)$ and $\gamma = \angle(m', m)$, which only depend on the coordinates of L_i , M_i and R_i .

Using the inscribed angle theorem, we know that P must lie on a circle k in the x, y -plane, with center $C(\frac{y_R}{\tan \alpha}, 0, 0)$ and radius $d(C, R)$ (see Fig. 5). The term $d(X, Y)$ denotes the euclidean distance between two points X and Y .

Let B the intersection of m' with k (opposed to P). By using the inscribed angle theorem again, we can determine the coordinates of B directly by rotating R around C by 2β . Note that B is independent of P .

We know from the definition of our intermediate coordinate system and from the symmetry of the pattern, that M lies on a circle w in the x, z -plane with the center $W = O(0, 0, 0)$ and radius $r_w = d(O, M)$, which is the fixed offset of M in world coordinates (150 mm).

We therefore know that

$$x_M^2 + z_M^2 = r_w^2. \quad (1)$$

M also lies on m , from which we can derive that

$$z_M = d(P, M') \tan \gamma, \quad (2)$$

with $M'(x_M, 0, 0)$ being the foot of M in the x, y -plane.

We can now intersect m and w by merging Eq. (1) and (2) to get

$$x_M^2 + d(P, M')^2 \tan^2 \gamma = r_w^2. \quad (3)$$

This equation still contains the yet unknown P . However, since m' goes through B and M' , we can describe P dependent on x_M by intersecting m' with k (and picking the solution unequal to B). Therefore, Eq. (3) only contains a single unknown variable x_M , which can be found by solving Eq. (3): Further expanding of (3) yields the quartic polynomial

$$c_4 x^4 + c_3 x^3 + c_2 x^2 + c_1 x + c_0 = 0, \quad (4)$$

with $x = x_M - x_C$ and coefficients

$$\begin{aligned} c_4 &= m^2 + 1 \\ c_3 &= 2(x_C - x_B) \\ c_2 &= v + (1 - 2x_C^2)B^2 - 4x_C x_B \\ c_1 &= 2(x_C B^2 - v x_B) \\ c_0 &= v B^2 + m^2 (B^2)^2, \end{aligned}$$

with $v = x_C^2 - r^2$ and $B^2 = x_B^2 + y_B^2$.

From the limited operating range of the quadcopter we can infer that the largest root will yield the correct solution, which can easily be found using Newton's method. Back substitution of x will retrieve M and P within the intermediate coordinate system. We can finally transform P into world coordinates by simply rotating it around the y -axis by $\delta = \text{atan2}(z_M, x_M)$.

To verify the solution, we calculate a confidence from the reprojection error: Let X_r be the virtual projection of X , for all four points $X = \{L, M, R, S\}$, using the current pose estimate. Then the total error ϵ is

$$\epsilon = \sum_X d(X_r, X_i). \quad (5)$$

Assuming Gaussian noise, the confidence c is then:

$$c = e^{-\frac{\epsilon^2}{\sigma^2}}, \quad (6)$$

with a previously chosen deviation $\sigma = 14$.

Having the camera's pose, we can finally get the pose of the quadcopter by applying a fixed transformation from the camera to the robot frame.

5. EXPERIMENTS

We tested the method's use for autonomous MAVs with two scenarios: In the first scenario we flew the *Hummingbird* several times by manual remote control. In the second scenario we let the quadcopter perform ten autonomous flights, each consisting of take-off, hovering and landing. We evaluate the method's accuracy for all 6DOF using ground truth data, which is generated by a Natural Point OptiTrack¹ tracking system which runs at 100 fps.

1. <http://www.naturalpoint.com/optitrack>

Table 1: Error statistics during manual flights. X, Y and Z in cm, roll, pitch and yaw in degrees

	X	Y	Z	roll	pitch	yaw
mean	1.21	2.34	1.84	1.20	1.65	1.40
std. deviation	0.87	1.71	1.35	1.07	2.68	1.37

5.1. Manual Flight

In order to evaluate the operating range and the accuracy of the proposed pose estimation, we flew the quadcopter remotely controlled. This way we could easily cover the whole operating range of the system, while in the autonomous flight the quadcopter successfully maintains the desired position, only varying slightly. In Fig. 6 the trajectory of a representative flight is depicted. One can see that the curves align well with the ground truth data across the whole flight range. The error statistics are shown in table 1. Hereby the error is defined as the absolute difference between the pose estimate of the tracking system and of our method.

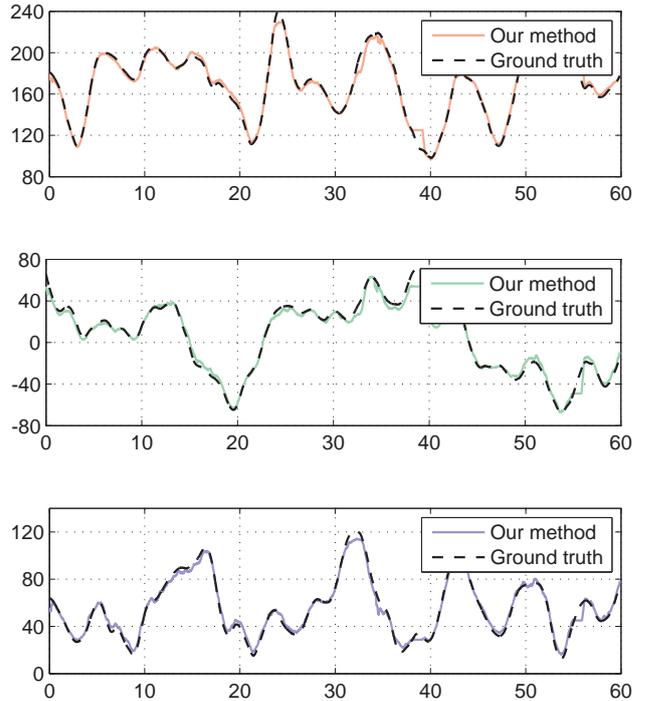


Figure 6: Position estimates during a manual flight compared to ground truth, in cm. Top: X. Middle: Y. Bottom: Z. x -axis: Time in seconds.

5.2. Autonomous Flight

The quadcopter performed ten flying including take-off, one minute hovering and land. This time it was autonomously controlled using the current pose estimate from the proposed pose estimation method. As desired position during hovering we chose 160 cm in front of the pattern and 80 cm in height. See Fig. 6 for the trajectory of a representative flight. In the hovering phase

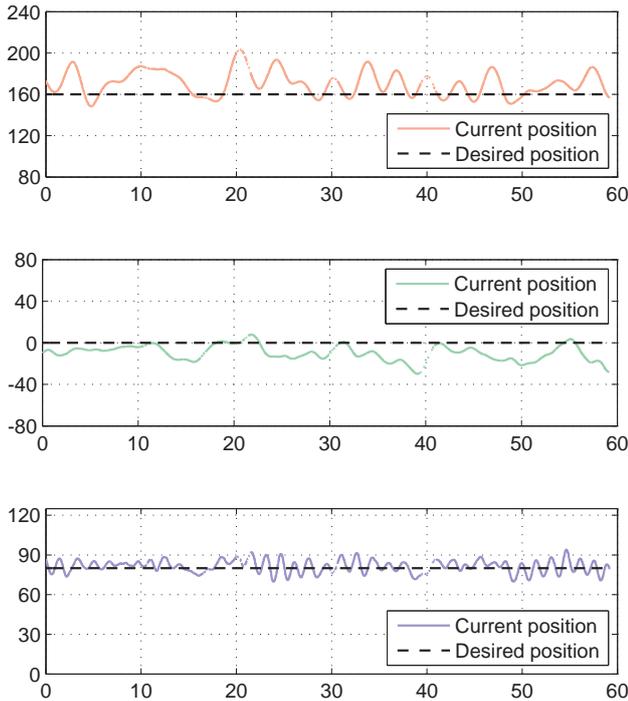


Figure 7: Position estimates during an autonomous flight compared to desired position, in cm. Top: X. Middle: Y. Bottom: Z. x-axis: Time in seconds. Current position measured with the external tracking system.

Table 2: Error statistics during the hovering phase of autonomous flights. X, Y and Z in cm, yaw in degrees. Note: roll and pitch are needed to maneuver the quadcopter and therefore cannot be set to a desired value.

	X	Y	Z	yaw
mean	12.39	10.27	4.02	4.87
std. deviation	9.45	6.64	2.75	3.96

we evaluated the error in x , y , z and yaw, defining the error as the absolute difference between the desired pose and the current pose measured by the tracking system. See table 2 for the statistical results. After the landing phase we evaluated the error a second time, which is shown in table 3.

5.3. Runtime Analysis

Detecting the orange markers took 9 ms on the Gumstix Overo Fire. Solving the P4P problem took 6 ms on the Gumstix, therefore yielding a total processing time of 15 ms per frame. Therefore the algorithm can process all frames coming at 60 fps using 90% CPU time.

Table 3: Error statistics after landing. X and Y in cm, yaw in degrees

	X	Y	yaw
mean	12.25	11.63	7.62
std. deviation	12.22	10.47	9.07

6. CONCLUSIONS

We developed and tested a method for pose estimation being able to run in real-time onboard of an MAV, even with limited hardware. The experimental results show that our method allows accurate and reliable control of MAVs, which in return enables autonomous flights of MAVs. The marker pattern is not lost during autonomous hovering, since the operating range covers a larger area, which was shown with the manual flights. The noise introduced by the proposed method is significantly below the noise from the PID controllers and wind turbulence during the autonomous flights.

Our results show that our method is feasible to act as a reliable pose estimate, which allows autonomous flights of MAVs. In the near future we will filter the pose estimate with a Kalman Filter. This would also allow a restriction of the image search space during the marker detection stage and therefore a further acceleration of the overall method.

To get further towards true autonomy, we plan to replace the passive marker detection with markerless features like FAST descriptors[9] in the future.

7. ACKNOWLEDGMENTS

The authors would like to thank Prof. Andreas Schilling from the computer graphics group in our faculty for providing us with the OptiTrack tracking system.

References

- [1] A. Cesetti, E. Frontoni, A. Mancini, P. Zingaretti, S. Longhi, "A Vision-Based Guidance System for UAV Navigation and Safe Landing using Natural Landmarks", *Journal of Intelligent & Robotic Systems*, Vol. 57, No. 1-4, 2010, pp. 233-257.
- [2] M. A. Fischler and R. C. Bolles: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381-395, 1981
- [3] P.J. GarciaPardo, G.S. Sukhatme, J.F. Montgomery, "Towards vision-based safe landing for an autonomous helicopter", *Robotics and Autonomous Systems*, Vol. 38(1), 2002, pp. 19-29.
- [4] B. Herisse, T. Hamel, R. Mahony, F.-X. Russotto, "Landing a VTOL Unmanned Aerial Vehicle on a Moving Platform Using Optical Flow", *IEEE Transactions on Robotics*, Vol. 28(1), Feb. 2012, pp. 77-89.
- [5] S. Lange, N. Suenderhauf, P. Protzel, "A Vision Based Onboard Approach for Landing and Position Control of an Autonomous Multirotor UAV in GPS-Denied Environments", *Proceedings 2009 International Conference on Advanced Robotics*, Munich, June 2009, pp. 1-6.
- [6] D. Mellinger, N. Michael, V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors", *The International Journal of Robotics Research*, Jan. 2012 online first, doi:10.1177/0278364911434236.

- [7] T. Merz, S. Duranti, G. Conte, "Autonomous landing of an unmanned helicopter based on vision and inertial sensing", *Experimental Robotics IX*, Vol. 21, 2006, pp. 343-352.
- [8] N. Michael, D. Mellinger, Q. Lindsey, V. Kumar, "The GRASP Multiple Micro UAV Testbed", *Robotics and Automation Magazine*, Vol. 17(3), 2010, pp. 56-65.
- [9] E. Rosten, R. Porter, T. Drummond, "FASTER and better: A machine learning approach to corner detection", in *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 32, 2010, pp. 105-119.
- [10] S. Saripalli, J.F. Montgomery, G.S. Sukhatme, "Visually guided landing of an unmanned aerial vehicle", *IEEE Transactions on Robotics and Automation*, Vol. 19(3), 2003, pp. 371-380.
- [11] K. E. Wenzel, P. Rosset, A. Zell., "Low-Cost Visual Tracking of a Landing Place and Hovering Flight Control with a Microcontroller", *Journal of Intelligent and Robotic Systems*, 2009, Vol. 57(1-4), pp. 297-311.
- [12] K. E. Wenzel, A. Masselli, A. Zell, "Automatic Take Off, Tracking and Landing of a Miniature UAV on a Moving Carrier Vehicle", *Journal of Intelligent & Robotic Systems*, Vol. 61, 2011, pp. 221-238.
- [13] K. E. Wenzel, A. Masselli, A. Zell, "A Quadcopter Hovering above a Person Wearing a Modified Cap", *Proceedings of International Micro Air Vehicle Conference and Flight Competition*, Braunschweig, Germany, 2010, pp. 1-7.
- [14] G. Xu, Y. Zhang, S. Ji, Y. Cheng, Y. Tian, "Research on computer vision-based for UAV autonomous landing on a ship", *Pattern Recognition Letters*, Vol. 30(6), 2009, pp. 600-605.