Following a quadrotor with another quadrotor using onboard vision

Jacobo Jiménez Lugo and Andreas Masselli and Andreas Zell
Cognitive Systems, University of Tübingen
Tübingen, Germany
Email: jacobo.jimenez, andreas.masselli, andreas.zell@uni-tuebingen.de

***Abstract*—We consider a leader-following problem with two flying robots with different sensor configurations. The follower is equipped with a camera to detect the leader robot. The platforms are based on the low-cost Parrot AR.Drone quadrotor modified for on-board sensing and computing. Our approach relies on detecting artificial passive markers on the leader quadrotor to compute the relative distance between the vehicles. We solve the perspective-n-point problem using an algorithm based on the inscribed angle theorem that runs efficiently on resource-limited platforms. We validate the position estimation algorithm and leader-following controllers with autonomously flown figures.**

I. Introduction

During the last years, the interest in research on unmanned aerial vehicles (UAV) has increased. In particular the quadrotor has become a popular platform due to its size, agility and maneuverability. These abilities have been demonstrated through execution of triple flips [1] and aggressive flying [2]. Quadrotors have been used in applications like surveillance [3], mapping [4], inspection [5] and the aid of human teams during a search and rescue mission [6]. The variety of quadrotor platforms ranges from expensive high quality platforms to toys.

The AR.Drone quadrotor appeared in 2010 as a toy with sensors such as two cameras and a gyroscope and basic autonomous behaviors such as assisted take off, hovering and landing. Due to its relatively high performance and low price, it immediately captured the interest of the research community. Its practical applications usually concern tasks where the images from the AR.Drone cameras are used for navigation [7], [8]. It has been used for cooperative tasks with ground robots, where it was used to have a bird's-eye view of the team and helped them to coordinate to overcome obstacles [9], [10], [11]. Multiple AR.Drones can be acquired for multi-robot research due to its low price of around $300.

In a team of robots, it is necessary to estimate the position of a robot relative to other team members. The relative position can be estimated using sensors in the environment, such as motion capture systems or cameras covering the operation range of the robots. Such systems are often used to develop control algorithms for multi-quadrotor teams [12]. They are fast and precise but they limit the range of application of the robots to a controlled environment. A decentralized approach is to equip the robots with sensors that can detect other team members, for example cameras. This approach resembles flocks of birds that rely on vision to estimate the position and velocity of other members of the group. Vision based approaches have been used to estimate relative bearings to other vehicles of a team of ground robots [13], [14]. This requires more computing power to process the sensor data and extract the positions of the other robots which makes it slower and less accurate than motion capture systems but allow the robots to go outside the laboratory.

With the fast dynamics of a quadrotor it is difficult for a vehicle to pause and plan its next move as it is usually done in ground robot teams. In order to have an application outside of a laboratory, a quadrotor requires real time relative position estimation that allows for fast control of the vehicle. Systems computing the position of a fixed known pattern have been used on a computationally limited quadrotor [15], [16]. In a previous work [17] the leader was tracked using an infrared camera from a Nintendo Wii remote controller, limiting the application range to indoor environments because of the infra red interference of sunlight. We overcome these problems using a vision system that is not affected by sunlight and can be used outdoors.

In this work we consider the problem of following an autonomous quadrotor using a single camera for relative position estimation of the leader with on-board processing only. We use two AR.Drones, which were modified to be able to autonomously fly predefined trajectories and to perform computer vision on board. The relative position to the leader quadrotor is estimated from three passive visual markers placed on the leader in a known pattern. The pose estimation is fed to our controllers to track the leader. The algorithms are experimentally verified by letting the leader quadrotor fly autonomously different trajectories followed by another quadrotor.

The remainder of this paper is structured as follows: in Section II we describe the autonomous robot platform based on the AR.Drone. In Section III we present the relative position estimation algorithm. Section IV describes the control strategy. In Section V we describe our experiments, present and discuss the results obtained. Finally we draw the conclusions in Section VI.

II. Hardware

In our setup, the leader and the follower vehicles have different hardware configurations. The leader is capable of flying a predetermined trajectory autonomously. It carries

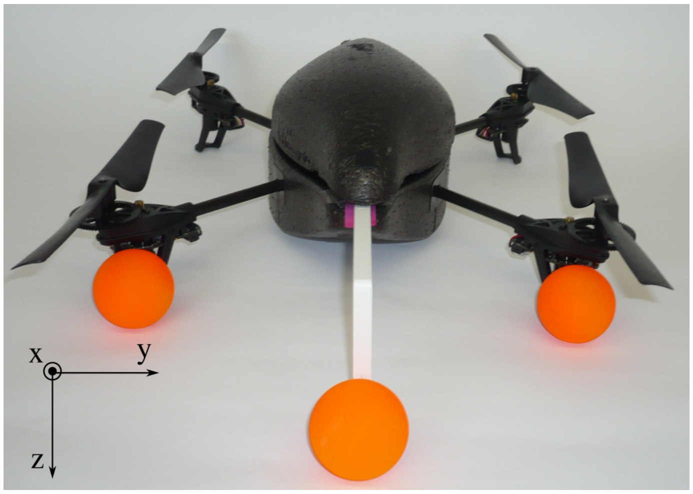

Fig. 1: The leader vehicle setup consisting of a modified AR.Drone version 1.0. The leader carries a microcontroller that uses the AR.Drones visual odometry to fly a predefined trajectory. It carries three orange table tennis balls as passive markers. In the coordinate frame of the pattern the x-axis points away (to the back of the AR.Drone), the y-axis points to the right and the z-axis points down.

Fig. 2: The follower vehicle setup consisting of a modified AR.Drone version 2.0, a Gumstix micro computer and a Point Grey firefly camera. The Gumstix microcomputer processes the images to find the markers and compute the relative distance to the leader. It computes the control commands and sends them to the AR.Drone.

a pattern formed with orange table tennis balls as passive markers. The follower is equipped with a color camera and a mini computer to detect the markers on the leader and compute its relative position with respect to the markers. The follower then tracks the leader's movements using the estimated relative position. Both leader and follower are completely autonomous and perform sensing and computations on board.

A. *Leader*

The leader quadrotor is a modified AR.Drone version 1.0. The AR.Drone is a low-cost toy quadrotor available for $300. Its sensor suite includes an internal measurement unit (IMU), an ultrasonic sensor and two cameras. One camera is at the front of the vehicle. It delivers images at 15 frames per second (fps) at a resolution of 640×480 pixels. The second camera points down; it has a resolution of 320×240 pixels and delivers images at 60 fps. This camera is used to extract information of the horizontal position and velocity of the AR.Drone. All the sensor processing and control of the vehicle is performed by an ARM processor running Linux.

To enable autonomous flight on the AR.Drone we use the framework proposed in [18]. We added an 8-bit microcontroller running at 14.745 MHz. The microcontroller runs navigation algorithms to generate a trajectory for the leader and according control commands to track it. The trajectory is defined as a series of intermediary waypoints computed using a formula describing the shape of the trajectory. The formula depends on time and is parametrized by the desired velocity of the vehicle and the length of the trajectory.

The microcontroller receives position information from the AR.Drone's visual odometry navigation data for position control.

The microcontroller communicates with the AR.Drone via the debugging port located on the main board. We uploaded to the AR.Drone's own microprocessor an application that retrieves the navigation data and sends it to the microcontroller and redirects the control commands from the microcontroller to the AR.Drone's control program.

The leader has three orange table tennis balls as passive markers. The markers are arranged in a 22 cm $\times$ 10 cm triangle pattern parallel to the ground. This configuration allows for a better estimate of the relative height of the follower.

The follower uses the markers to locate the leader and estimate its relative position (See Figure 1). In the coordinate frame of the pattern the x-axis points to the back of the drone; the y-axis points to the right and the z-axis points down.

B. *Follower*

The follower quadrotor is a modified AR.Drone version 2.0. This version of the AR.Drone has similar characteristics to the version 1.0 previously described. Its main difference is the front camera, which delivers images in 1280×720 pixels resolution at 30 fps. The newer AR.Drone version extends its sensor suite with an air pressure sensor that is combined with the ultrasonic sensor to estimate its altitude. We extended this AR.Drone to perform computer vision on board. The front camera of the AR.Drone is not suitable for our application since the images are too big to process in real time. Therefore we equipped the AR.Drone with an additional camera. The camera is a Point Grey Firefly that delivers images at 60 fps with a resolution of 640×480 pixels. We mounted a single-board computer to process the images and extract the location of the markers to estimate its position relative to the leader. The computer is a *Gumstix Overo Fire* with an ARM Cortex-A8 processor running at 700 MHz and 256 MB of RAM. The setup is shown in Figure 2.

Similarly to the leader, the Gumstix computer is connected to the debugging port of the AR.Drone. The proxy application runs on the AR.Drone, redirecting control commands to the main control program.

III. Pose Estimation

In this section we describe how to retrieve the relative pose of the follower from the table tennis balls visible in the camera image.

The method has been formerly used in [16] and is split in two stages: First we detect the orange ball markers within the camera image. Second we calculate the pose based on the pixel coordinates of the detected markers. The algorithm has been optimized for low performance hardware, which is usually found on MAVs due to weight and power limits. We managed the ARM processor of the Gumstix to process all of the 60 frames from the camera.

A. Marker Detection

To ensure fast evaluation, we skip debayering the whole image and operate on the raw image itself. We apply color segmentation to extract all orange regions from the image, using a lookup table for faster processing. 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 implausible aspect ratios. The contour of the remaining regions is determined and tested for being similar to a circle with the *Randomized Hough Transform* [19]. Finally the three largest round 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 from their size itself.

B. Retrieving the Pose

In the next step we calculate the position of the camera relative to the pattern using the pixel position from the ball detection described above. The problem that has to be solved in this stage is well-known in literature as the *Perspective-3-Point Problem* (*P3P*), in general *PnP* problem for n points. Fischler and Bolles [20] introduce the PnP problem as being equivalent to the "Location Determination Problem" (LDP), which they define as follows:

Given a set of m control points, whose 3-dimensional coordinates are known in some coordinate frame, and given an image in which some subset of the m control points is visible, determine the location (relative to the coordinate system of the control points) from which the image was obtained.

Before solving the *P3P*, the correspondences between world and image points have to be known. This can easily be done by regarding the operating range of the quadrotor. First it is always following the pattern, second the quadrotor does not fly upside down. Therefore we can sort the image points along their y-coordinate (recall the camera is vertically mounted) and identify them as L_i, M_i and R_i. In general, *P3P* yields up to four solutions [20]. Considering the limits of the operating range of the quadrotor again, we can exclude three of these solutions and therefore estimate the full pose with only three markers.

Solving the *P3P* is described in detail in[16].

By applying a fixed transformation from the camera to the quadrotor frame, we get the position of the quadrotor from the camera position relative to the leader.

IV. Control

To maintain a desired distance to the leader, the follower uses three independent proportional-integral-derivative (PID) controllers for roll, pitch angles and its height. A fast proportional-derivative (PD) controller is used for the yaw angle in order to maintain the pattern in the center of the image. The controllers use the relative pose estimate of our algorithm that has been previously smoothed by a low pass filter. The controller gains were experimentally determined to ensure stability of the vehicle.

V. Experiments and results

To analyze the performance of our system we carried out a series of flights in different shapes with the quadrotor setup described in Section II.

We first evaluated the accuracy of the position estimation algorithm. To obtain ground truth measurements, we performed the first experiment in a room equipped with a motion capture system. The motion capture system covers an approximate volume of $3 \times 2 \times 2\, m^3$ and is able to track objects with a precision of $1\, mm$. The operation range of our system is from 0.5 m to 3 m approximately due to the size and placement of the markers and the total width of the pattern. The pattern of the leader is at a fixed position inside the tracked volume at an approximate height of $1.2\, m$. We command the follower quadrotor to hover at a fixed distance from the pattern of $160\, cm$ on the x-axis, and $0\, cm$ on y- and z-axes. We track the quadrotors position over a period of 10 minutes over multiple flights. Then the relative distance is computed from the difference in position from the quadrotor and the pattern. The position estimated by the follower is then compared to the relative position computed from the motion capture system data to get the accuracy of the measurements made by the follower. Overall our method showed a position estimation error of less than $2\, cm$ and a standard deviation over $1\, cm$ (See Table I). This accuracy is sufficient for tracking another vehicle for our following experiments. Figure 3 shows the position estimate from our algorithm and ground truth data of the relative position between the fixed pattern and the hovering quadrotor. The overlap of the two plots show that the accuracy

TABLE I: Error analysis of position estimation algorithm.

	Errors		
	x [cm]	y [cm]	z [cm]
RMSE	1.98	1.48	1.27
Std. Dev.	1.27	1.10	1.09
Max Error	7.30	4.46	4.52

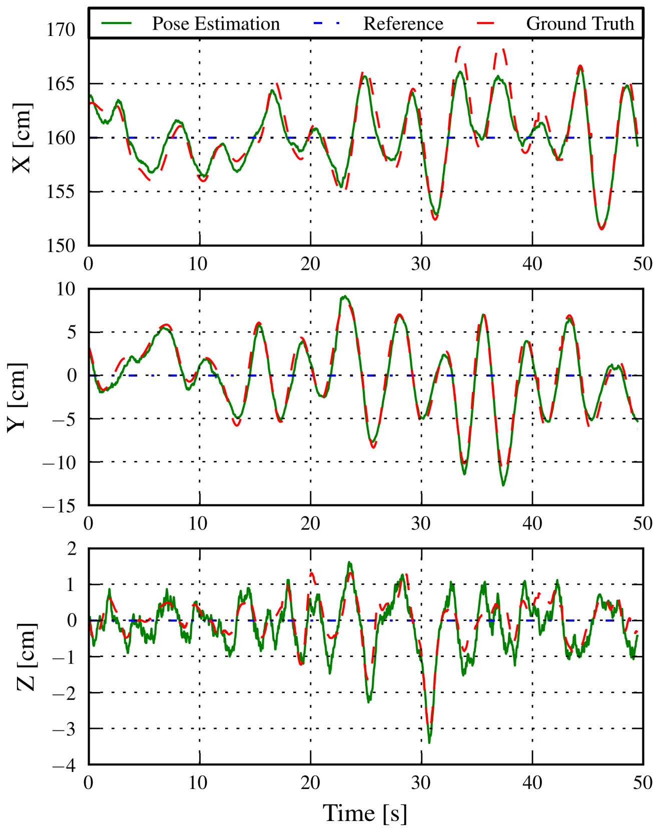

Fig. 3: Relative distance plots showing the estimates of our vision system (green), ground truth measurements obtained from a motion capture system (red dashed) and desired relative position for the follower quadrotor hovering (blue). The overlap of the vision estimate and ground truth plots show the accuracy of our method to estimate relative position.

of our method is sufficient for relative position measurements.

We tested the performance of our controllers in two different scenarios. One scenario tests the capabilities to hold a desired position at a fixed distance from the pattern. The other scenario regards tracking of another quadrotor carrying the pattern while it flies predefined trajectories autonomously.

We achieved an average error of 10 cm on the x-axis, 7 cm on the y-axis and a height difference of only 2.6 cm (See Table II). The results show that our method is able to hold its position based on the data received by the vision system within a small space. The position of the follower during fifty seconds of hovering is plotted in Figure 3.

The capture volume of our motion capture system is unsuitable for two quadrotors to maneuver while being tracked. In the previous experiment we demonstrated the accuracy of the estimates of our algorithm. Therefore, we consider the position estimates from our vision system as ground truth for our further experiments.

TABLE II: Error analysis of hovering control.

	Errors		
	x [cm]	y [cm]	z [cm]
RMSE	10.08	12.70	2.66
Std. Dev.	7.33	10.33	2.37
Max Error	25.2	30.90	11.10

TABLE III: Error analysis of line trajectory.

	Errors		
	x [cm]	y [cm]	z [cm]
RMSE	16.76	43.97	3.28
Std. Dev.	16.07	26.07	2.18
Max Error	64.22	93.01	14.87

The first flight with both quadrotors was a straight line of 1.5 m flown sideways six times. The leader flew in one direction until it reached the specified distance of 1.5 m and then reversed its direction. The resulting trajectory was a series of start-accelerate-break sequences along the leader's y-axis with a reversal in direction at the start and end of the trajectories length. The follower flew at the same height as the leader keeping a distance 1.6 m behind it.

The mean error along the x-axis is around 16 cm (See Table III). The error along the y-axis was larger since it was the direction in which the follower had to correct larger errors. Like in the hovering experiment the height error remained around 3 cm.

The large errors are caused by sudden change in direction of the leader at the end of the line trajectory. The follower reacts to this change in direction without losing track of the pattern.

The second figure was a circle with a diameter of 2 m. The leader trajectory is planned such that the x-axis of the vehicle is always perpendicular to the circumference of the circle. This keeps the pattern pointing always to the center of the circle. With the follower quadrotor at a distance of 1.6 m, the space needed to fly the figure is reduced and the target position of the follower remains always inside the circle. We calculated statistics of the errors achieved during 5 rounds of a 2 m diameter circle (See Table IV). The errors on the x-axis of 23 cm and y-axis of 18 cm are about half the length of the quadrotor, indicating a good track of the trajectory. The errors on the height difference remained around 3 cm as in the previous tests. The height difference is kept since the leader

TABLE IV: Error analysis of circle trajectory.

	Errors		
	x [cm]	y [cm]	z [cm]
RMSE	23.83	18.85	3.11
Std. Dev.	21.92	15.19	2.39
Max Error	59.03	64.00	15.04

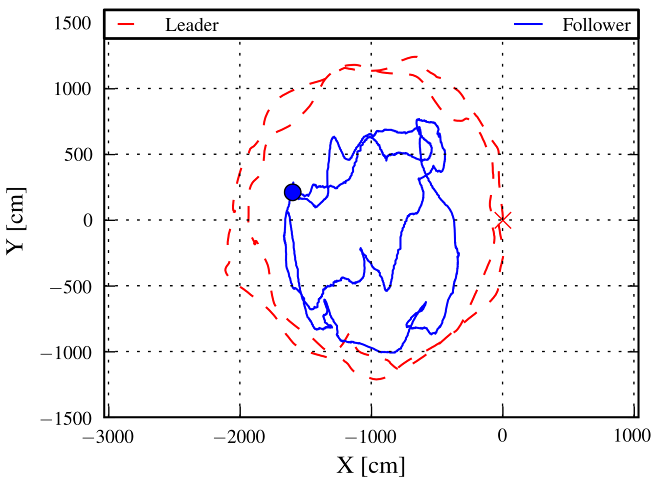

Fig. 4: Two rounds of a $2\,m$ circle path flown. The red dashed line is the path flown by the leader as reported by its odometry. The blue line is the path of the follower computed from the relative distance estimated by our method.

flew at a constant height with little variance on its height. This trajectory involves fast changes in the leaders yaw angle. That induces a large error on the follower's x- and y-axes since it flies behinds the leader. The controller of the follower reacts quickly to avoid losing the markers in a disadvantageous perspective where they will be behind each other or merged into one blob.

An example of two rounds of the circular trajectory is shown in Figure 4. The trajectory on the outside is computed from the leader's visual odometry. The follower's trajectory is computed from the relative position reported by the vision system. The follower trajectory always remains inside the diameter of the circle, showing the robustness of the controllers against changes in yaw angle of the leader.

VI. Conclusion

We have presented a system capable of following an autonomous leader using vision to estimate the relative distance between the vehicles. The leader and follower are based on the low cost quadrotor AR.Drone enhanced with a microcontroller and a microcomputer-camera setup for autonomous flight with on-board sensing and computing. The vision system estimates the relative distance to the leader using passive markers on the leader that are detected and identified by the follower. We solve the P3P problem with an algorithm based on the inscribed angle theorem and a unique solution is found using constraints inherent to the task. Furthermore the algorithm is able to run on a microcomputer at 60 Hz. The precision of this algorithm was found to be around $2\,cm$. The experiments showed that the follower quadrotor is able to track the position of a leader with a mean error of around $23\,cm$ on the x-axis and $43\,cm$ on the y-axis under sudden direction and heading changes of the leader. With these errors being less than one length of the quadrotor the system is capable to fly in space constrained environments.

References

[1] S. Lupashin, A. Schöllig, M. Sherback, and R. D'Andrea, "A simple learning strategy for high-speed quadrocopter multi-flips," *IEEE International Conference on Robotics and Automation*, pp. 1642–1648, 2010.

[2] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *International Journal of Robotics Research*, 2012.

[3] E. Semsch, M. Jakob, D. Pavlicek, and M. Pechoucek, "Autonomous uav surveillance in complex urban environments," in *Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT '09. IEEE/WIC/ACM International Joint Conferences on*, vol. 2, 2009, pp. 82–85.

[4] F. Fraundorfer, L. Heng, D. Honegger, G. H. Lee, L. Meier, P. Tanskanen, and M. Pollefeys, "Vision-based autonomous mapping and exploration using a quadrotor mav," in *IROS*, 2012, pp. 4557–4564.

[5] C.-M. K. C.-H. B. C. Eschmann, Christian; Kuo, "Unmanned aircraft systems for remote building inspection and monitoring," in *Proceedings of the Sixth European Workshop on Structural Health Monitoring Dresden, Germany, July 3-6, 2012*, 2012.

[6] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. Grixa, F. Ruess, M. Suppa, and D. Burschka, "Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue," *Robotics Automation Magazine, IEEE*, vol. 19, no. 3, pp. 46–56, 2012.

[7] J. Engel, J. Sturm, and D. Cremers, "Camera-based navigation of a low-cost quadrocopter," *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[8] C. Bills, J. Chen, and A. Saxena, "Autonomous mav flight in indoor environments using single image perspective cues," *IEEE International Conference on Robotics and Automation*, pp. 5776 –5783, may 2011.

[9] N. Mathews, A. Christensen, R. O'Grady, and M. Dorigo, "Spatially targeted communication and self-assembly," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 2678–2679.

[10] P. Twu, R. Chipalkatty, J. de la Croix, T. Ramachandran, J. Shiveley, M. Egerstedt, A. Rahmani, and R. Young, "A hardware testbed for multi-uav collaborative ground convoy protection in dynamic environments." in *AIAA Modeling and Simulation Technologies Conference.*, August 2011.

[11] M. Saska, V. Vonásek, T. Krajník, and L. Preucil, "Coordination and navigation of heterogeneous uavs-ugvs teams localized by a hawk-eye approach," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2012)*, 2012, pp. 2166–2171.

[12] M. Turpin, N. Michael, and V. Kumar, "Trajectory

design and control for aggressive formation flight with quadrotors," *Autonomous Robots*, vol. 33, no. 1-2, pp. 143–156, 2012. [Online]. Available: http://dx.doi.org/10.1007/s10514-012-9279-y

[13] G. Mariottini, F. Morbidi, D. Prattichizzo, N. Vander Valk, N. Michael, G. Pappas, and K. Daniilidis, "Vision-based localization for leader-follower formation control," *Robotics, IEEE Transactions on*, vol. 25, no. 6, pp. 1431–1438, 2009.

[14] A. Das, R. Fierro, V. Kumar, J. Ostrowski, J. Spletzer, and C. Taylor, "A vision-based formation control framework," *Robotics and Automation, IEEE Transactions on*, vol. 18, no. 5, pp. 813–825, 2002.

[15] S. Yang, S. Scherer, and A. Zell, "An onboard monocular vision system for autonomous takeoff, hovering and landing of a micro aerial vehicle," *Journal of Intelligent & Robotic Systems*, vol. 69, no. 1-4, pp. 499–515, 2013. [Online]. Available: http://dx.doi.org/10.1007/s10846-012-9749-7

[16] A. Masselli and A. Zell, "A Novel Marker Based Tracking Method for Position and Attitude Control of MAVs," in *Proceedings of International Micro Air Vehicle Conference and Flight Competition*. Braunschweig, Germany: DGON, 2012, pp. 1–6. [Online]. Available: http://www.cogsys.cs.uni-tuebingen.de/publikationen/2012/Masselli2012IMAV.pdf

[17] K. E. Wenzel, A. Masselli, and A. Zell, "Visual tracking and following of a quadrocopter by another quadrocopter," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2012)*, pp. 1–6, October 7-12 2012, accepted for publication.

[18] J. Jimenez and A. Zell, "Framework for autonomous onboard navigation with the a.r.drone," in *Proceedings of the International Conference on Unmanned Aerial Systems (ICUAS)*, 2013.

[19] P. Kultanen, L. Xu, and E. Oja, "Randomized hough transform (rht)," in *Pattern Recognition, 1990. Proceedings., 10th International Conference on*, vol. i, 1990, pp. 631–635 vol.1.

[20] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," in *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms*, M. A. Fischler and O. Firschein, Eds. Los Altos, CA.: Kaufmann, 1987, pp. 726–740.