RoboCupRescue 2013 - Robot League Team Jacobs Robotics (Germany)

Max Pfingsthorn, Sören Schwertfeger, Natasha Danailovska, Mjellma Berisha, Jan Brenstein, Ahmed Reman Ghazi, Remus Dumitru, Vaibhav Mehta, Andrei Militaru and Andreas Birk

Jacobs University Bremen
Campus Ring 1
28759 Bremen, Germany
`a.birk@jacobs-university.de`
`http://robotics.jacobs-university.de/`

Abstract. The 2013 Jacobs University Bremen Rescue Robot team is described in this paper. The team has been active in RoboCup Rescue since 2001. The team takes a radically new approach to the traditionally land-bound rescue scenario. Instead of threaded land robots, low-cost quad-rotors are used. During the German Open competition in 2012, the team already showed a working system using teleoperation. As a major improvement over last year's prototype system, the team implements dense 2D overhead image SLAM, sparse 3D SLAM, closed-loop control, robust communication using the 868Mhz band, automatic detection of objects of interest (victims, hazmat labels, QR codes, etc), as well as a combined 3D user interface for multiple robots. Additionally, the team aims to field multiple quad-rotors at once, some running an autonomous mapping algorithm.

Introduction

The Jacobs University robotics group has been working in the domain of rescue robots since 2001. The team has already participated in the real robot rescue league at RoboCup 2002 in Fukuoka (4th place), RoboCup 2003 in Padua (4th place), the US Open 2004 in New Orleans (2nd place), RoboCup 2004 in Lisbon, RoboCup 2005 in Osaka, the Dutch Open 2006 in Eindhoven (1st place), the US Open 2006 in Atlanta (1st place), the RoboCup world championship 2006 in Bremen (best European), the German Open 2007 (1st place), RoboCup 2007 in Atlanta (best European), RoboCup German Open 2008 (1st place), the RoboCup 2008 in Suzhou, China, as well as RoboCup German Open 2009 and RoboCup 2009 in Graz, Austria [31, 30, 7, 6, 3, 2, 4, 5].

Since 2009, the Jacobs Robotics team has reoriented the focus of its RoboCup development efforts towards undergraduate and graduate guided research. The first product was a team of 3 undergraduate students that participated in the RoboCup German Open in 2012. Additionally, the team takes a radically new approach to the traditionally land-robot focused RoboCup Rescue competition.

Instead of land robots, the team focuses on deploying quad-rotor UAVs exclusively.

The quad-rotor platform used by the team is the Parrot AR.Drone, versions 1 and 2 (please see table 1 for a list of features for these platforms). This low cost quad-rotor is very robust as well as easy and cheap to repair. Such a platform allows very high-risk and aggressive uses, especially required in the close-quarters situation of the RoboCup Rescue arena.

Fig. 1. Used quadrotor platforms: Parrot AR.Drone 1.0 (left) and 2.0 (right)

Feature	**AR.Drone 1.0**	**AR.Drone 2.0**
Front Video	640x480 (VGA)	1280x720 (720p)
Front FOV	93°	92°
Bottom Video	176x144 (CIF)	640x480 (VGA)
Bottom FOV	64°	62°
Video Modes	320x240 or 176x144	1280x720 or 640x360
Video Framerates	fixed (18 fps)	variable (15-30 fps)
Video Bitrate	auto or variable (720kb/s-3.6Mb/s)	auto or variable (250kb/s-4Mb/s)
Video Compression	P264 (H264 variant)	H264 (HW accelerated)
Gyroscope	2+1 axis	3 axis
Accelerometer	3 axis	3 axis
Magnetometer	×	3 axis
Ultrasonic Altimeter	✓	✓
Barometric Altimeter	×	✓
USB storage	×	✓
CPU	ARM9 468 MHz	ARM Cortex A8 1GHz
RAM	128MB	1GB
OS	Linux 2.6.27	Linux 2.6.32
Flight Time (one charge)	≈ 12 min	≈ 11 min
802.11 Connectivity	b/g	b/g/n

Table 1. Features of the Parrot AR.Drone 1.0 and 2.0

After showing a prototype tele operated system and winning 4th place at the RoboCup German Open 2012 (see figure 2), several improvements have been made. Most notably, the team will use two quad-rotor UAVs at the same time. One shall autonomously map the arena from a relatively high altitude of 3 to 4 meters. The other is used in a tele operated fashion at a very low altitude, between 0.2 to 2 meters. The low-altitude drone is used to find and map objects of interest such as victims, QR codes, and other fiducials in the arena.

Since the team's development effort has only started last year, other contributions this year are manyfold:

- Visual SLAM in 2D using the bottom camera of the high-altitude UAV
- Sparse Visual SLAM in 3D using the front camera of the low-altitude UAV
- Closed-loop control using estimates from Visual SLAM
- Automatic detection of objects of interest for both
  a) known objects, e.g. hazmat labels, QR codes, fiducials
  b) unknown objects, e.g. victims
- Map merging of 2D overhead map and 3D low-altitude map
- Improved reliability and range of the control channel by switching to the 868MHz band
- Improved signal quality of the 802.11 connection to the UAV by utilizing directional antennas

Fig. 2. Tele operated flight with the AR.Drone 1.0 at the RoboCup German Open 2012.

1 Team Members and Their Contribution

Max Pfingsthorn	Team Leader, SLAM
Sören Schwertfeger	Team Leader, Human Robot Interface
Mjellma Berisha	Structure from Motion
Natasha Danailovska	Object Detection, Multi-Robot Cooperation
Ahmed Reman Ghazi	2D Image Registration
Jan Brenstein	Image Mosaicking
Remus Dumitru	2D Perspective Image SLAM
Vaibhav Mehta	Control and Exploration
Robert Savu	SLAM
George Mandresi	Place Recognition
Krishna Raj Sapkota	QR Code Localization
Ernesto Gonzales Huaman	Communications
Cornel Amariei	Communications
Andrei Militaru	Image Processing
Kaustubh Pathak	Advisor
Andreas Birk	Advisor

2 Operator Station Set-up and Break-Down (10 minutes)

The operator station consists of one or more laptop computers connected by ethernet to a wireless access point (802.11b/g/n). The two AR.Drone UAVs weigh less than 500g each, so both the UAVs and the laptop(s) as well as the wireless access point can easily be carried by a single person. Because the UAVs require a level surface to calibrate their gyroscopes before takeoff, two adjustable starting pads to even out the ramps at the starting positions in the arena are needed as well.

To start a mission, the following steps are necessary:

- Place laptops on the table and connect to ethernet wired network
- Connect wireless access point to wired network and power
- Place external antennas towards the operational area
- Connect AR.Drone battery for both UAVs
- Launch main ROS system across the laptop computers
- Test connection to AR.Drones via main laptop's wireless interface
- Configure AR.Drones and connect to main access point
- Test connection to AR.Drones via access point
- Launch AR.Drone ROS interface and test communications
- Take off with first AR.Drone, start autonomous mapping
- Take off with second AR.Drone

3 Communications

Figure 3 shows the communications architecture for the multi-UAV control system. The main communications link to the UAVs consists of an 802.11n

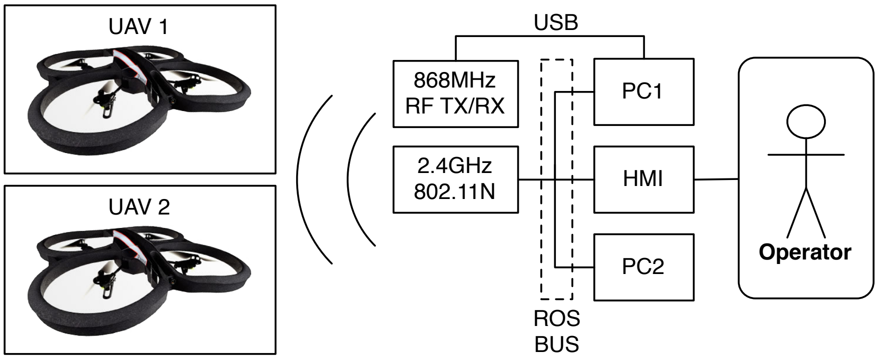


Fig. 3. Communications architecture for multi-UAV control using a single operator.

(AR.Drone 2.0) or 802.11b/g (AR.Drone 1.0) wireless LAN network. While the AR.Drone 2.0 does support 802.11n, it does not support the 5GHz band, so both networks run on the 2.4GHz band. The access point in use (LevelOne WAP-6012) allows direct selection of the channel as well as the modulation range within that channel in order to minimize interference with other leagues. Directional patch antennas are used in a MIMO configuration to ensure good reception over the whole arena.

A separate control connection is established using a serial-based RF module using frequency modulation (FM) on the 868MHz band. Using the Abacom ATRT100 modules, it is possible to transmit serial data at up to 100kb/s in a half-duplex channel over up to 180 meters. This separate control channel is used because of less signal interference on this band, as well as for range and robustness of the RF module used. A corresponding receiver unit is placed on the AR.Drone UAV, connected via a serial interface to the AR.Drone. A custom program on the AR.Drone then forwards the control commands read via the serial connection to the main AR.Drone firmware via a local UDP socket.

Rescue Robot League			
Jacobs Robotics (Germany)			
Frequency	**Channel/Band**	**TX Power (mW)**	**Bandwidth**
2.4 GHz - 802.11b/g	selectable (1-13)	70	54 MBit/s
2.4 GHz - 802.11n	selectable (1-13)	70	300 MBit/s
868 MHz - FM	HW select	10	max. 100kBit/s

Table 2. Communication Frequencies used in the Jacobs Robotics team.

4 Control Method and Human-Robot Interface

Using freely available software within the ROS system, it is possible to control the AR.Drone very accurately [15]. The state estimation method of the `tum_ardrone` package uses most onboard sensors (gyroscope, accelerometers), their onboard processing, as well as the video stream to estimate the pose of the AR.Drone in an absolute reference frame. Parallel Tracking and Mapping (PTAM) [18] is used to provide a sparse 3D visual map and a position estimate within it. An Extended Kalman Filter is used for data fusion and state estimation. A good wireless link with low interference is however required. Video frames as well as telemetry data (e.g. linear speed and orientation estimates from the AR.Drone itself) will arrive with a significant delay or not at all if the wireless signal quality is decreased. Usually, the effect is that the Kalman filter diverges far from the actual pose, with disastrous results of any controller that may use this estimate.

The same ROS package also provides a simple position controller on top of the state estimation and AR.Drone ROS interface. It is a very simplistic approach, controlling x, y, z, and yaw separately with PID controllers. The controller parameters are tuned rather aggressively, resulting in large accelerations and thus large *roll* and *pitch* angles. However, our image registration method (described in section 5) assumes very small *roll* and *pitch* angles. In order to keep the *roll/pitch* and linear speeds within the limits imposed by the mapping system, a smoothing method of the speed commands is being developed to allow the use of (slightly differently) tuned PID controllers as well as the ability to produce detailed maps. The focus of this extension will be to allow the high-altitude UAV to smoothly follow a predetermined trajectory in order to generate the best possible map of the entire RoboCup Rescue arena within the tightly limited flight time available with one battery charge.

Another focus is to provide an easy operator interface for interacting with the low-altitude UAV. Direct teleoperation is difficult if the UAV is not at least semi-autonomous: Any reach for the keyboard or mouse, e.g. to confirm a victim identification dialog, almost certainly leads to a crash since reacting to any disturbance in time is impossible when the pilot is busy otherwise. Either the pilot, or a software program, needs to be in control of the UAV at all times. In order to reduce the operators workload, the above control mechanisms are employed in a semi-autonomous setting. Joystick or joypad commands are not directly forwarded to the UAV but used to alter the controller set points. Thus, when the hand is taken off the joystick or joypad, it returns to the zero position, which is interpreted as a command to keep the current position with active control. External disturbances, e.g. through AC vents, are thus automatically corrected and the operator can focus on interacting with the system as a whole.

Since ROS is used to integrate the various software modules, *rviz* is used to visualize most data processed by the system. A 3D view of the map and the UAV poses within it are shown in *rviz*. A heads-up view for a separate video stream display will be implemented, designed to look much like flight simulators, including an artificial horizon and overlays indicating found objects of interest and the like. Objects of interest that are localized in 3D are also shown in the 3D display of *rviz*.

5 Map generation/printing

Two separate mapping idioms are used in conjunction: 2D photo mosaicking as well as sparse 3D visual mapping.

As mentioned above, PTAM is used to estimate the 3D pose of the UAV in an absolute reference frame. The mapping process in PTAM is initialized using a stereo technique and continuously updated via new keyframes throughout the tracking process. Tracking is made more agile and real-time capable via adding intensity edge features represented as short, straight segments due to their robustness to blur, and inter-frame rotation estimation for relocation in case of tracking failure. The result of building map with PTAM is a sparse 3D point cloud representing the estimated positions of tracked features.

For the 2D photo mosaic, a spectral image registration technique called iFMI [11, 12] is used. iFMI uses a polar-logarithmic re-sampling of image information to turn rotation and scaling into a corresponding phase shift and allows for image registration in one step. The basis for the 2D signal registration in this approach is a Phase-Only Matched Filter (POMF). This correlation approach makes use of the fact that two shifted signals having the same spectrum magnitude are carrying the shift information within its phase. The advantages of iFMI are that it detects translation, rotation, and scale, without the use of images features. Instead, it takes the whole image content into account. It is fast (50Hz), has a constant computation time for a given image resolution, and is very robust to occlusions and some change in perspective. This registration method has been integrated into a graph-based SLAM framework [25] by providing an uncertainty analysis.

This method has recently been extended to relax the constraint that the camera has to be normal to the observed plane [24]. When the same physical plane is observed by two different camera poses, there exists a homography between the image coordinates of corresponding points.

$$p' = KHK^{-1}p$$

Here, p is the image point, K is the camera intrinsic matrix, and H is the homography matrix. Using an approach that registers sub-images with POMF after an initial iFMI registration step, point correspondences between the two registered images are generated. These correspondences are much more robust than feature points because a larger area can be taken into account. A combination of Random Sample Consensus (RANSAC) [19] and the four-point algorithm [22] can robustly compute the homography matrix using these point correspondences.

The homography matrix can be decomposed into an estimate of translation and rotation in 3D, as well as the parameters of the plane (normal vector and distance) that was observed between the two camera frames. The translation between the camera poses and the distance to the plane are only recoverable up to scale, thus even two consecutive homography decompositions are not necessarily compatible. Combining the observations of the underlying plane over two pairs in an optimization framework significantly increases the quality of the camera motion estimate and equalizes the scale across these pairs so their translation estimates are made compatible. Further images are added to the trajectory while keeping the plane parameters constant and only optimizing the relative translation and rotation of the new image relative to the previous image in the sequence.

Current research efforts try to utilize the point correspondences to gain more information about the environment than just the perspective transformation to the ground plane. After computing the relative camera pose, all found point correspondences, if they were inliers during the RANSAC process or not, could be triangulated to provide 3D point estimates. These may be used as a forth image channel in the generated photo mosaic denoting "height" above the ground plane, creating a 2.5D map. Additionally, it may allow to create a parallax-free orthoimage.

In the current system, the 2D (potentially 2.5D) overhead photomosaic is merged with the sparse 3D feature map of the low-altitude UAV by selecting control points by hand. This is easier than it sounds: Victims and objects of interest will be placed in the sparse 3D map by the system, and are usually easily identifiable in the overhead photomosaic as well.

6 Sensors for Navigation and Localization

As mentioned, the AR.Drone has a built-in gyroscope, accelerometer, and ultrasonic rangers to measure altitude [26]. Version 2.0 extends this set of navigation sensors with a magnetometer as well as a barometric altimeter. A down-looking camera is used onboard to provide speed estimates using optical flow (the exact method of which is not documented by Parrot). All these separate sensor readings are fused onboard into a pose and speed estimate that is communicated to the control computer via UDP packets.

As mentioned in the previous section, the video cameras are also used for localizing the UAVs in the map. The Visual SLAM method(s) used allow the robot to localize itself in the incrementally built map in an absolute coordinate frame.

7 Sensors for Victim Identification

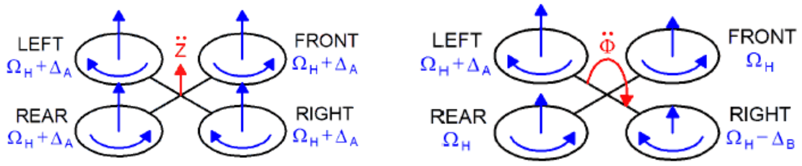
Since the sensor payload of the AR.Drone is very hard to extend, the development effort was directed towards getting the most out of the existing onboard sensors. This only leaves the color video cameras as sensors to be used for victim detection. Both front and bottom cameras can be useful for detecting victims.

In order to detect hazmat signs and simulated victims in the camera images, a template based object recognition method will be used. Dominant Orientation Template (DOT) [17] uses a representation of the target object based on image gradient information. This representation is related to Histograms of Gradients (HoG) [13] method, the difference being that DOT does not use histograms but rather takes the locally dominant orientations which are then transformed to binary representation of the templates. Thus DOT is very fast since efficient bit-wise operations are be used for similarity computations. The advantage of this is that DOT can detect texture-less objects in real time and obtain their 3D pose. Another advantage of DOT is online learning with an auto-generated training set, since the fast template creation uses just a few exemplars from different viewpoints in real-time. This makes DOT an excellent choice for RoboCup Rescue, as both objects known and trained a-priori (e.g. hazmat signs) should be detected as well as initially unknown or changing objects (e.g. victims).

Victim candidates are presumably human, so a face detection method [20] implemented in OpenCV [10] is used to initialize candidate victim templates online.

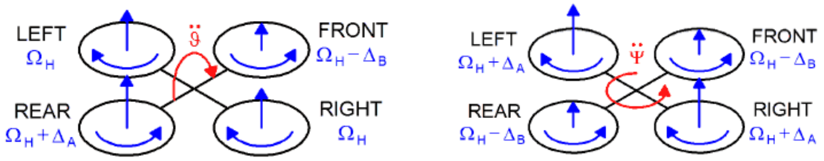
8 Robot Locomotion

As any quad-copter, the AR.Drone has four rotors, spinning in opposite directions to cancel the *yaw* moment. Figure 4 shows how the four possible movements in x, y and z directions as well as *yaw* rotations are achieved by varying each rotor's speed.



(a) Throttle

(b) Roll



(c) Pitch

(d) Yaw

Fig. 4. AR.Drone Movements, from [26].

The AR.Drone has an advanced onboard control software that takes over many of the piloting tasks. It supports automatic take off and landing, an altitude limit, linear and rotational speed limits, automatic fault detection, as well as optically controlled hover.

9 Other Mechanisms

The team's development effort has been focused on achieving a competitive state in order to participate in RoboCup. As such, there are no further developments beyond the ones described above.

However, the software to generate photomosaic maps is a standalone application, and works with a variety of video input sources. A ROS-based video source had been integrated last year in order to use it with the video stream of the AR.Drone. This software was used to generate the photo map shown in figure 5. Most importantly, this software was used by operators trained by Prof. Robin Murphy at the Fukushima Daiichi nuclear plant to generate overview maps of the area. Unfortunately, due to confidentiality issues, the maps generated there were not shared with us or Prof. Robin Murphy.

10 Team Training for Operation (Human Factors)

Flying quad-rotors is not easy, but the advanced control software onboard the AR.Drone as well as the extensive suite of supporting software offboard allows novice pilots to operate the multi-UAV system outlined above. Minimal training is necessary, borrowing mostly from computer game interfaces generally known. While a short period is needed to get to know the basic UAV controls, the system should work as expected and be robust enough to be effectively used by a relatively novice operator.

11 Possibility for Practical Application to Real Disaster Site

Unmanned Aerial Vehicles (UAV) are obviously well suited to give a bird's eye view over an incident site. They can create overview maps or, in the form of Micro Aerial Vehicles (MAV), search for victims or hazardous material in more confined spaces. They are used in a wide range of SSRR applications including search and rescue, reconnaissance, and surveillance [34, 8, 14, 21, 32, 16, 1, 23] and have already been used in practice [27]. The Jacobs Robotics Group has already demonstrated the use of UAVs during the 2009 European Land Robot Trials (ELROB-2009) and the 2010 Response Robot Evaluation Exercises (RREE-2010) in Disaster City in College Station, Texas [9]. RREE is an annual event organized by the Intelligent Systems Division (ISD) of the National Institute of Standards and Technology (NIST)[33]. At this event, the task for the aerial vehicle was to gather overview of a rubble pile, i.e., the mission is more related to rescue than safety and security (see Figure 5).

Also in search and rescue missions, it is of interest to detect motion in the scene, e.g., to detect hand waving of a trapped victim when a UAV is autonomously surveying a rubble pile. The challenge is that the UAV itself is always moving - even during station-keeping it is never perfectly stable - and

Fig. 5. An aerial map of a rubble pile at Disaster City made out of 630 video frames. Loop closing edges from a SLAM algorithm are shown (from [9]).

hence this ego-motion has to be compensated. A solution for that problem was presented by our group at the IEEE International Symposium on Safety, Security, and Rescue Robotics 2011 [28]. Properties of the iFMI algorithm that is used for mapping and motion detection have been investigated in [29].

The low cost systems used by the Jacobs Robotics team in this competition provide an ideal starting point to develop the control-, autonomy- and mapping-software needed for navigating confined environments. Hardened and more reliable hardware that was designed towards search and rescue missions while still being relatively low cost will enable the use of such Micro Aerial Vehicles in real rescue mission, making a loss of a robot permissible.

12 System Cost

Item	**Number**	**Cost in EUR**
AR.Drone 2.0	2	600
Access point	1	110
868MHz RF Module	2	100
Directional antennas	3	40
Accessories (e.g. Joystick)	-	50
Control Laptop	1	-
Sum		900

Table 3. System cost.

As already mentioned, the whole system comes at a very low cost. Each complete AR.Drone 2.0 platform costs only 300 EUR. The 300 Mbps access point (LevelOne WAP-6012) is available at around 110 EUR plus 40 EUR for directional antennas. One of the Abacom ATRT100 modules costs around 50 EUR. The total system cost is thus about 900 EUR (see Table 3).

Instead of costly well-equipped, possibly rugged, control laptops which typically cost more than 2000 EUR, regular consumer laptops are used that are readily available in our group.

It should also be noted that the replacement parts for the AR.Drone system are very inexpensive. For example, a new set of rotors comes for just under 7 EUR, while the price would be several hundred euro for professional platforms. This is quite important, because especially in aerial robotics, the chance of braking parts during experiments and trials is very high.

References

1. G. Belloni, M. Feroli, A. Ficola, S. Pagnottelli, and P. Valigi. An autonomous aerial vehicle for unmanned security and surveillance operations: design and test. In *Safety, Security and Rescue Robotics, 2007. SSRR 2007. IEEE International Workshop on*, pages 1–4, 2007.
2. Andreas Birk. The IUB 2004 rescue robot team. In Daniele Nardi, Martin Riedmiller, and Claude Sammut, editors, *RoboCup 2004: Robot Soccer World Cup VIII*, volume 3276 of *Lecture Notes in Artificial Intelligence (LNAI)*. Springer, 2005.
3. Andreas Birk, Stefano Carpin, Winai Chonnaparamutt, Viktoras Jucikas, Hamed Bastani, Ivan Delchev, Ivan Krivulev, Seongchu Lee, Stefan Markov, and Andreas Pfeil. The IUB 2005 rescue robot team. In Itsuki Noda, Adam Jacoff, Ansgar Bredenfeld, and Yasutake Takahashi, editors, *RoboCup 2005: Robot Soccer World Cup IX*, Lecture Notes in Artificial Intelligence (LNAI). Springer, 2006.
4. Andreas Birk, Stefano Carpin, and Holger Kenn. The IUB 2003 rescue robot team. In D. Polani, B. Browning, A. Bonarini, and K. Yoshida, editors, *RoboCup 2003: Robot Soccer World Cup VII*, volume 3020 of *Lecture Notes in Artificial Intelligence (LNAI)*. Springer, 2004.

5. Andreas Birk, Holger Kenn, Martijn Rooker, Agrawal Akhil, Balan Horia Vlad, Burger Nina, Burger-Scheidlin Christoph, Devanathan Vinod, Erhan Dumitru, Hepes Ioan, Jain Aakash, Jain Premvir, Liebald Benjamin, and Luksys Ge. The IUB 2002 rescue robot team. In Gal Kaminka, Pedro U. Lima, and Raul Rojas, editors, *RoboCup-02: Robot Soccer World Cup VI*, LNAI. Springer, 2002.
6. Andreas Birk, Kausthub Pathak, Winai Chonnaparamutt, Soeren Schwertfeger, Ivan Delchev, Stefan Markov, and Ravi Rathnam. The IUB 2006 rescue robot team. In G. Lakemeyer, E. Sklar, D. Sorrenti, and T. Takahashi, editors, *RoboCup 2006: Robot Soccer World Cup X*, volume 4434 of *Lecture Notes in Artificial Intelligence (LNAI)*. Springer, 2007.
7. Andreas Birk, Kaustubh Pathak, Sören Schwertfeger, Jann Poppinga, Heiko Bülow, Winai Chonnaparamutt, Ivan Delchev, Stefan Markov, and Ravi Rathnam. Robocuprescue - robot rescue league team jacobs university bremen, germany. In U. Visser, F. Ribeiro, T. Ohashi, and F. Dellaert, editors, *RoboCup 2007: Robot Soccer World Cup XI*, volume 5001 of *Lecture Notes in Computer Science (LNCS)*. Springer, 2008.
8. Andreas Birk, Burkhard Wiggerich, Heiko Blow, Max Pfingsthorn, and Soeren Schwertfeger. Reconnaissance and camp security missions with an unmanned aerial vehicle (uav) at the 2009 european land robots trials (elrob). In *IEEE International Workshop on Safety, Security, and Rescue Robotics (SSRR)*. IEEE Press, 2009.
9. Andreas Birk, Burkhard Wiggerich, Heiko Blow, Max Pfingsthorn, and Soeren Schwertfeger. Safety, security, and rescue missions with an unmanned aerial vehicle (uav): Aerial mosaicking and autonomous flight at the 2009 european land robots trials (elrob) and the 2010 response robot evaluation exercises (rree). *Journal of Intelligent and Robotic Systems*, 64(1):57–76, 2011.
10. G. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, 2008.
11. H. Buelow and A. Birk. Fast and robust photomapping with an unmanned aerial vehicle (uav). In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 3368–3373, Oct. 2009.
12. H. Buelow, A. Birk, and V. Unnithan. Online generation of an underwater photo map with improved Fourier Mellin based registration. In *OCEANS 2009-EUROPE, 2009. OCEANS '09.*, pages 1–6, May 2009.
13. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886 –893 vol. 1, june 2005.
14. S.S. Dhaliwal and A. Ramirez-Serrano. Control of an unconventional vtol uav for search and rescue operations within confined spaces based on the marc control architecture. In *Safety, Security and Rescue Robotics (SSRR), 2009 IEEE International Workshop on*, pages 1–6, 2009.
15. J. Engel, J. Sturm, and D. Cremers. Camera-Based Navigation of a Low-Cost Quadrocopter. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.
16. M.A. Goodrich, J.L. Cooper, J.A. Adams, C. Humphrey, R. Zeeman, and B.G. Buss. Using a mini-uav to support wilderness search and rescue: Practices for human-robot teaming. In *Safety, Security and Rescue Robotics, 2007. SSRR 2007. IEEE International Workshop on*, pages 1–6, 2007.
17. S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab. Dominant orientation templates for real-time detection of texture-less objects. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2257 –2264, june 2010.

18. G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225 –234, nov. 2007.
19. Peter Kovesi. MATLAB and Octave functions for computer vision and image processing. "Centre for Exploration Targeting, School of Earth and Environment, The University of Western Australia", 2000.
20. Shengcai Liao, Xiangxin Zhu, Zhen Lei, Lun Zhang, and Stan Z. Li. Learning multi-scale block local binary patterns for face recognition. In *Proceedings of the 2007 international conference on Advances in Biometrics*, ICB'07, pages 828–837, Berlin, Heidelberg, 2007. Springer-Verlag.
21. D. Longo, D. Melita, G. Muscato, and S. Sessa. A mixed terrestrial aerial robotic platform for volcanic and industrial surveillance. In *Safety, Security and Rescue Robotics, 2007. SSRR 2007. IEEE International Workshop on*, pages 1–6, 2007.
22. Yi Ma, Stefano Soatto, Jana Kosecka, and S. Shankar Sastry. *An Invitation to 3-D Vision*. Springer, 2003.
23. A. Ollero, M. Bernard, M. La Civita, L. van Hoesel, P.J. Marron, J. Lepley, and E. de Andres. Aware: Platform for autonomous self-deploying and operation of wireless sensor-actuator networks cooperating with unmanned aerial vehicles. In *Safety, Security and Rescue Robotics, 2007. SSRR 2007. IEEE International Workshop on*, pages 1–6, 2007.
24. K. Pathak, M. Pfingsthorn, H. Bülow, and A. Birk. Robust Estimation of Camera-Tilt for iFMI based Underwater Photo-Mapping using a Calibrated Monocular Camera. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013. Accepted for Publication.
25. Max Pfingsthorn, Andreas Birk, Sören Schwertfeger, Heiko Bülow, and Kaustubh Pathak. Maximum likelihood mapping with spectral image registration. In *Robotics and Automation, 2010. ICRA 2010. Proceedings of the 2010 IEEE International Conference on*, 2010.
26. Stephane Piskorski, Nicolas Brulez, Pierre Eline, and Frederic DHaeyer. Ar.drone developer guide sdk 2.0. http://projects.ardrone.org/, December 2012.
27. K.S. Pratt, R.R. Murphy, J.L. Burke, J. Craighead, C. Griffin, and S. Stover. Use of tethered small unmanned aerial system at berkman plaza ii collapse. In *Safety, Security and Rescue Robotics, 2008. SSRR 2008. IEEE International Workshop on*, pages 134–139, 2008.
28. Soeren Schwertfeger, Andreas Birk, and Heiko Blow. Using ifmi spectral registration for video stabilization and motion detection by an unmanned aerial vehicle (uav). In *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 1–6. IEEE Press, 2011.
29. Soeren Schwertfeger, Heiko Blow, and Andreas Birk. On the effects of sampling resolution in improved fourier mellin based registration for underwater mapping. In *7th Symposium on Intelligent Autonomous Vehicles (IAV), IFAC*. IFAC, 2010.
30. Sören Schwertfeger, Jann Poppinga, Kaustubh Pathak, Hamed Bastani, Heiko Bülow, Max Pfingsthorn, and Andreas Birk. Robocuprescue - robot rescue league team jacobs university bremen, germany. In Luca Iocchi, Hitoshi Matsubara, Alfredo Weitzenfeld, and Changjiu Zhou, editors, *RoboCup 2008: Robot Soccer World Cup XII*, volume 5399 of *Lecture Notes in Artificial Intelligence (LNAI)*. Springer, 2009.
31. Sören Schwertfeger, Jann Poppinga, Kaustubh Pathak, Heiko Bülow, Narunas Vaskevicius, and Andreas Birk. Robocuprescue - robot rescue league team jacobs university bremen, germany. In J. Baltes, M.G. Lagoudakis, T. Naruse, and

S Shiry, editors, *RoboCup 2009: Robot Soccer World Cup XIII*, volume 5949 of *Lecture Notes in Computer Science (LNCS)*. Springer, 2010.

32. N. Sunderhauf, S. Lange, and P. Protzel. Using the unscented kalman filter in mono-slam with inverse depth parametrization for autonomous airship control. In *Safety, Security and Rescue Robotics, 2007. SSRR 2007. IEEE International Workshop on*, pages 1–6, 2007.
33. TEEX. Disaster city. http://www.teex.com/teex.cfm?pageid=USARprog&area=USAR &templateid=1117, 2008.
34. T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I.L. 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*, 19(3):46 –56, sept. 2012.