Received April 14, 2015, accepted May 5, 2015, date of publication May 12, 2015, date of current version June 1, 2015.

Digital Object Identifier 10.1109/ACCESS.2015.2432455

Obstacle Detection and Collision Avoidance for a UAV With Complementary Low-Cost Sensors

NILS GAGEIK, PAUL BENZ, AND SERGIO MONTENEGRO
Chair Aerospace Information Technology, University of Würzburg, Würzburg 97074, Germany

Corresponding author: N. Gageik (nils.gageik@uni-wuerzburg.de)

This work was supported in part by the Würzburg-Schweinfurt Chamber of Industry and Commerce, in part by the Universitätsbund Würzburg, Würzburg, Germany, in part by the German Research Foundation, and in part by the University of Wuerzburg within the funding programme Open Access Publishing.

ABSTRACT This paper demonstrates an innovative and simple solution for obstacle detection and collision avoidance of unmanned aerial vehicles (UAVs) optimized for and evaluated with quadrotors. The sensors exploited in this paper are low-cost ultrasonic and infrared range finders, which are much cheaper though noisier than more expensive sensors such as laser scanners. This needs to be taken into consideration for the design, implementation, and parametrization of the signal processing and control algorithm for such a system, which is the topic of this paper. For improved data fusion, inertial and optical flow sensors are used as a distance derivative for reference. As a result, a UAV is capable of distance controlled collision avoidance, which is more complex and powerful than comparable simple solutions. At the same time, the solution remains simple with a low computational burden. Thus, memory and time-consuming simultaneous localization and mapping is not required for collision avoidance.

INDEX TERMS Collision avoidance, obstacle detection, ultrasonic, infrared, autonomous, UAV, quadrotor, quadrocopter.

I. INTRODUCTION

In the past decade, the interest in UAVs and autonomy has constantly increased. Collision avoidance is an important requirement for autonomous flights. Although multiple solutions for obstacle detection and collision avoidance of UAV's exist, these solutions suffer from different drawbacks.

In general, the existing solutions can be divided into two parts: The first part contains simple collision avoidance solutions which are based on avoiding collisions by steering the vehicle into opposite direction using different techniques [1]–[5]. The biggest drawback of such solutions is the intervention into the steering which may not be desirable for the mission of the UAV. These solutions do not allow a UAV to control its distance from an obstacle which may be necessary in a couple of situations. The second part can be described as SLAM-based solutions. Those solutions avoid collisions by mapping, positioning, and navigation within the map while the positioning and mapping is based on a complex SLAM algorithm [6]–[11], [13]. Compared to the first division, these solutions do not limit the mission, but the collision avoidance requires the SLAM-algorithm which needs considerable memory and computational power compared to the already mentioned simple solutions, because of its complexity. The proposed solution in this paper can be classified as being in between these two divisions, avoiding the drawbacks of each of them.

Besides the mentioned aspects, it has to be taken into consideration, that US and IR sensors are complementary technologies. IR sensors, like all optical sensors, fail under poor lighting conditions such as smoke or fog and cannot detect diaphanous obstacles; contrary to ultrasonic sensors [18]–[21], which do not pose such drawbacks. However ultrasonic sensors cannot detect sound absorbing surfaces like clothes or curtains properly. US sensors are therefore not reliable to detect people or the available distance from them, which is no challenge for IR sensors.

Compared to existing works, which do not have redundant sensors for obstacle detection, the presented solution combines two complementary technologies with acceptable costs, thereby increasing the reliability of the system. This combination is quite useful for fields of application such as search and rescue (SAR) missions, where the UAV is used to find injured persons in a smoky underground garage after a fire.

In this work the focus lies on quadrotors, as they are optimized to fly in low space environments, but the presented solution can also be applied to other classes of UAVs. This work is a part of the AQopterI8 project, which aims to develop

an autonomous flying quadrocopter for indoor application like supporting firefighters in a SAR mission.

II. RELATED WORKS

Hand in hand with an increased interest in UAVs, their applications, users, developers and manufacturers have grown constantly, particularly in the last decade. More and more features, functions, and solutions are being developed to empower the systems and to simplify handling for less experienced pilots. GPS-guided navigation and fully rotating camera systems for video recording are common approaches [14]–[17], [22]. These systems are being used for many applications like movies and television, documentation, 3D modelling of archaeological buildings and sites as well as for maintenance of power lines and industrial plants, just to mention a few [14], [31]. Adding a reliable system for collision avoidance, also referred to as an anti-collision system, is currently the next hurdle to take for more autonomy. This enables further applications and lessens the skill requirements of the pilots. That is why collision avoidance is a field of interest in research and development.

Scherer et al. [5] and Achtelika et al. [12] belong to the first division using laser scanners to detect obstacles. This approach has been extended by Gronzka [9], [10], who uses multilevel-SLAM with motion and altitude estimation for 3D mapping, positioning and navigation. The implementation is based on the Mikrokopter project and is capable of autonomous flight [14]. The Mikrokopter project started as open source platform but is now commercial. However, processing of the data is not performed on-board, rather on an external laptop because of its computational burden. Blösch [6] developed another SLAM algorithm which also requires external hardware. Shen [11] combined a laser scanner with a camera using the iterative closest point (ICP) algorithm for position estimation and an extended Kalman filter (EKF) for data fusion. His system is capable of autonomous flight only with on-board resources and requires a 1.6 GHz Atom for data processing. Weiss [23] uses the same system, a pelican quadrotor from AscTec [15] that costs approximately 7000€. He developed a SLAM-algorithm which requires only the on-board camera for positioning and autonomous flight. Engel [7], [8] presented an algorithm based on an EKF that exploits feature detection. His solution empowers the parrot quadrocopter [24] to fly autonomously through 3D figures. However, it also requires an external laptop for image data processing. Celik et al. [13] presented a SLAM-based system using a monocular camera and US sensors.

Becker and Bouabdallah [1] and Bouabdallah [2], [3] employed four US sensors for obstacle detection and a camera system based on over-ground optical flow computations for positioning. The system is capable to avoid collisions by controlling its position; however it can neither cover 360° nor perform distance control. In contrast to that, Roberts [4] uses four IR sensors and avoids collisions by steering towards the opposite direction. In opposite to these simple solutions, our approach combines IR and US sensors, fuses the data and performs distance control, while no sophisticated SLAM algorithm is used that would require high processing power. Hence the entire solution can be implemented on a 60MHz microcontroller.

III. HARDWARE ARCHITECTURE

Fig. 1 shows the hardware architecture of the stated quadrotor AQopterI8. Minimum required components for the discussed approach are marked as red. Fig. 2 shows corresponding equipped quadrotor. For height control two infrared sensors, one ultrasonic sensor and a pressure sensor are used [19]. The obstacle detection uses 16 infrared sensors and 12 ultrasonic sensors for redundant 360° coverage. Two kinds of infrared sensors are used to enhance the measurement range.

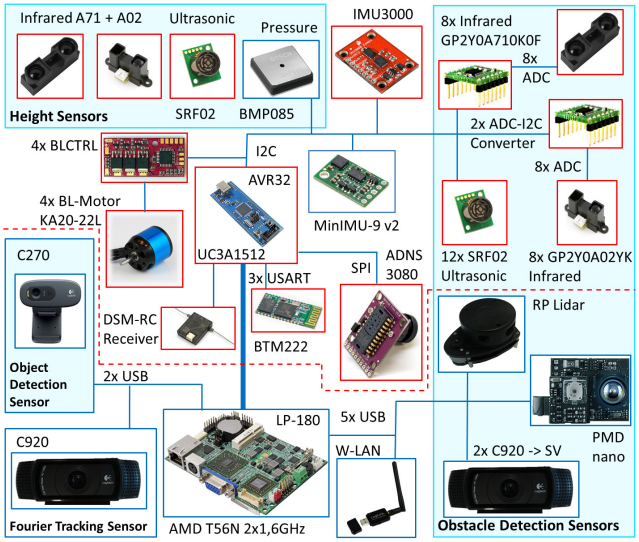

FIGURE 1. Hardware architecture of AQopterI8: The minimum required components are marked red.

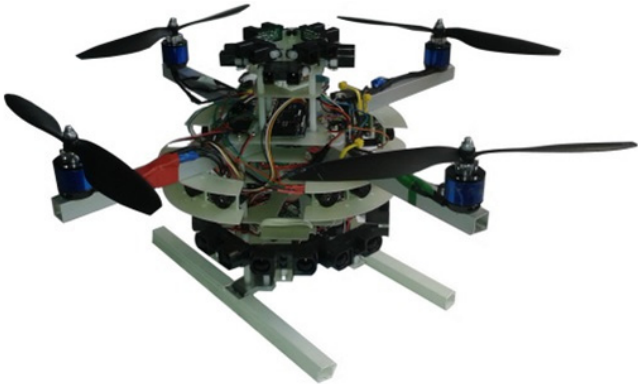

FIGURE 2. AQopterI8 equipped with 12 US sensors and 16 (2×8) IR sensors.

The entire system is manufactured and programmed at the chair Aerospace Information Technology (University of Würzburg). The advantage of this system can clearly be concluded from Fig. 1, as the system is quite modular and diverse components as well as support parts in the mechanical

design can easily be adapted to the requirements.

The overall system design supports many components like an AMD 1.6GHz processor and cameras for image processing e.g., object detection, stereo vision and Fourier tracking [27]–[30].

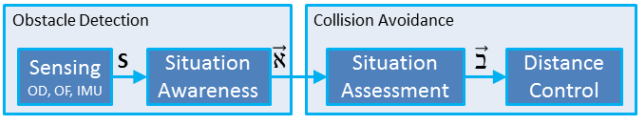

FIGURE 3. Concept overview.

IV. CONCEPT

A. OVERVIEW

Fig. 3 depicts the concept of the system from a high level view divided into obstacle detection (OD) and collision avoidance (CA). The obstacle detection consists of the sensing and the situation awareness (SA). The sensing is the reading, conditioning and filtering of the OD (IR + US), OF (optical flow) and IMU (inertial measurement unit) sensors, to receive distances d, position changes Δp and orientations α (roll) and β (pitch). These data $\mathbf{S}$ (1) are then the input to the situation awareness, whose task is to process the multiple information from the sensing to determine the vector $\vec{\aleph}$ (2), which contains the minimum distance available to the front, right, rear and left as well as the longitudinal (x) and lateral (y) velocity v.

$$\mathbf{S} = \begin{bmatrix} d_0^{US} & d_0^{IRL} & d_0^{IRS} & & & & \\ d_1^{US} & d_1^{IRL} & d_1^{IRS} & & & & \\ d_2^{US} & d_1^{IRL} & d_1^{IRS} & & & & \\ d_3^{US} & d_2^{IRL} & d_2^{IRS} & & & & \\ d_4^{US} & d_3^{IRL} & d_3^{IRS} & \Delta p_x^{OF} & \Delta p_y^{OF} & \alpha & \beta \\ d_5^{US} & d_3^{IRL} & d_3^{IRS} & \vdots & \vdots & \vdots & \vdots \\ d_6^{US} & d_4^{IRL} & d_4^{IRS} & \Delta p_x^{OF} & \Delta p_y^{OF} & \alpha & \beta \\ d_7^{US} & d_5^{IRL} & d_5^{IRS} & & & & \\ d_8^{US} & d_5^{IRL} & d_5^{IRS} & & & & \\ d_9^{US} & d_6^{IRL} & d_6^{IRS} & & & & \\ d_{10}^{US} & d_7^{IRL} & d_7^{IRS} & & & & \\ d_{11}^{US} & d_7^{IRL} & d_7^{IRS} & & & & \end{bmatrix} \tag{1}$$

$$\vec{\aleph} = \begin{pmatrix} d_{Front} \\ d_{Right} \\ d_{Rear} \\ d_{Left} \\ v_x \\ v_y \end{pmatrix} = \begin{pmatrix} \vec{d}_f \\ \vec{v} \end{pmatrix} \tag{2}$$

This information, vector $\vec{\aleph}$, is required for situation assessment, whose task is first the determination of the situation, i.e. if and where a collision may arise and second the configuration of the distance control with the matrix $\vec{\beth}$ (3). $\vec{\beth}$ contains the error e, distance d, speed v, speed in direction towards an obstacle w and control parameters $\vec{C}$ to react properly on the threat of a collision. With this information the distance control (DC) regulates the distance. Thus collisions are avoided.

$$\vec{\beth} = \begin{pmatrix} e_x & e_y \\ d_x & d_y \\ v_x & v_y \\ w_x & w_y \\ \vec{C}_x & \vec{C}_y \end{pmatrix} \tag{3}$$

B. WEIGHTED FILTER (WF)

The weighted filter (WF) is a concept for data fusion or data selection to obtain the best possible result of multiple sensors. It computes a weight for every sensor, which represents the sensors reliability. The WF is easy to design and implement and has a low computational burden compared to a Kalman filter (KF). Another advantage is the fact that the WF can adapt very rapidly to measurement jumps such as a distance discontinuity of sudden upcoming obstacles, while a KF minimizes the noise or variance. The WF can be used together with a Kalman filter to update its variances or without the filter, as it is done here to reduce the computational burden and delay. The usage of this approach is one special feature of this work.

The main idea of the data fusion is to use different sources of information in order to pick the best sensor. Information constitutes the history of every sensor, the total history and the bearing of every sensor as well as the current measurements of each sensor. Sensors with high noise can be considered as less accurate compared to those with low noise. Invariant (constant) measurements over a long period of time and the corresponding sensors are discarded by a preprocessing step based on the consideration that the distance from a flying UAV to surrounding obstacles cannot be constant in long terms.

All relevant sensors are divided into main data and reference data sensors. Main data are absolute distance measurements like the distance measurements from the ultrasonic and infrared sensors. Reference data are the sensors which cannot stand alone but are used to evaluate the reliability of the main sensor data and help to decide between those sensors. Reference data are the position changes computed from the inertial and optical flow sensor.

By comparing all possible sensor combinations of main data and reference data using a norm, in our case the differential norm, a weight for every sensor is calculated. This weight is used to decide which sensor measurement is to be taken as the final result. If the computed weights have a low variation, the sensor values are fused according to their weights, otherwise the sensor with the best weight is used and the bad sensor values are rejected based on the idea, that in this case they are corrupted.

The reference sensors can help to select a main sensor based on the rationale that the changes in distance correspond to the position changes of the UAV. This holds good if the obstacles are not moving, because in this case the change

of distances from the obstacles and the position changes of the UAV are equal. If the obstacles are moving, this information is corrupted by their movement, but still it is unlikely that this movement corresponds better to randomly wrong measurements. However, this problem can be handled by parametrization of the WF.

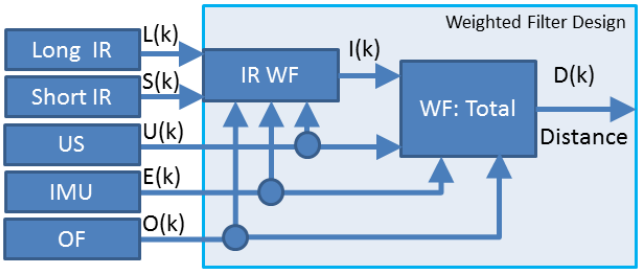

FIGURE 4. Weighted filter design.

Since the two infrared sensors exclude each other in most parts of their measurement range, the WF consists of two iterations (Fig. 4). In the first step, the better IR sensor is selected and in the second step it is fused with the ultrasonic sensor. Summarized by the vector **R**, there are five inputs to the weighted filter to determine the fused distance D(k). (4) is equivalent to one row of (1).

$$\mathbf{R} = \begin{bmatrix} U(k) & L(k) & S(k) & O(k) & E(k) \end{bmatrix} \tag{4}$$

The three distance measurements L(k), S(k) and U(k) correspond to the distance measurements of a long infrared, a short infrared and an ultrasonic sensors at time k. O(k) and E(k) are the measurements of position changes computed from the optical flow sensor and the IMU using (9) and (10). So O(k) and E(k) correspond to O_i and E_i, respectively. The weight $W_S(k)$ of the short IR sensor is computed with (5). The history is incorporated by the exponential moving average (EWMA) filter part with its smoothing factor e. The parameters a, b, c and d determine the influence of the different sources and q_O is a dynamic calculated parameter determined by the current quality of the optical flow sensor. The influence of the ultrasonic sensor d is dynamically updated according to its distance, since the ultrasonic sensor is more reliable for small distances. All other parameters are constant.

$$\begin{aligned} N_1 &= |S(k-1) + E(k) - S(k)| \\ N_2 &= |S(k-1) + O(k) - S(k)| \\ N_3 &= |D(k-1) + E(k) - S(k)| \\ N_4 &= |D(k-1) + O(k) - S(k)| \\ N_5 &= |U(k) - S(k)| \\ W_S'(k) &= a \cdot (N_1 + b \cdot q_O \cdot N_2) \\ &\quad + c \cdot (N_3 + b \cdot q_O \cdot N_4) + d \cdot N_5 \\ W_S(k) &= e \cdot W_S'(k) + (1-e) \cdot W_S(k-1) \end{aligned} \tag{5}$$

$W_L(k)$, the weight of the long IR sensor, is computed analogous to $W_S(k)$. The fused infrared distance I(k) is determined by these weights and is the result of the IR WF. If the difference between $W_S(k)$ and $W_L(k)$ is below a certain threshold, I(k) is computed by (6):

$$I(k) = \frac{W_S(k) \cdot L(k) + W_L(k) \cdot S(k)}{W_S(k) + W_L(k)} \tag{6}$$

Otherwise the fused infrared distance I(k) is equal to the sensor with the better weight (lower value). The overall fusion is analogue and (7) is given here for completeness. $W_I(k)$ is the weight of the fused infrared distance I(k) and $W_U(k)$, the analogue weight of the ultrasonic sensor, computed by replacing I(k) with U(k) in (7). If the difference between both weights is less than 10%, the overall distance D(k) is computed by (8).

$$\begin{aligned} N_6 &= |I(k-1) + E(k) - I(k)| \\ N_7 &= |I(k-1) + O(k) - I(k)| \\ N_8 &= |D(k-1) + E(k) - I(k)| \\ N_9 &= |D(k-1) + O(k) - I(k)| \\ W_I'(k) &= a \cdot (N_6 + b \cdot q_O \cdot N_7) \\ &\quad + c \cdot (N_8 + b \cdot q_O \cdot N_9) \\ W_I(k) &= e \cdot W_I'(k) + (1-e) \cdot W_I(k-1) \end{aligned} \tag{7}$$

$$D(k) = \frac{W_U(k) \cdot I(k) + W_I(k) \cdot U(k)}{W_I(k) + W_U(k)} \tag{8}$$

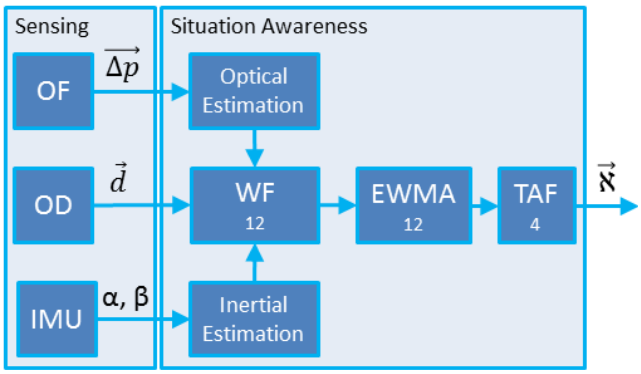

FIGURE 5. Obstacle detection with sensing and situation awareness.

C. SITUATION AWARENESS (SA)

Fig. 5 describes the principle of obstacle detection with the situation awareness. The situation awareness consists of 12 weighted filters (WF) [19] and 12 EWMA filters. Each row of **S**, corresponding to **R**, is the input of one WF which needs to be transformed for the last four columns by the optical and inertial estimation block properly. (9) and (10) express the computation of the optical estimation O_i and inertial estimation E_i of sector i with $g = 9.81 \frac{m}{s^2}$ and T = 0.01s.

$$O_i = \Delta p_x^{OF} \cos\left(i\frac{\pi}{2}\right) + \Delta p_y^{OF} \sin\left(i\frac{\pi}{2}\right) \tag{9}$$

$$E_i = g \cdot T(\sin(\alpha) \cdot \sin\left(i\frac{\pi}{2}\right) + \sin(\beta) \cdot \cos\left(i\frac{\pi}{2}\right)) \tag{10}$$

Algorithm 1 TAF

Input: distance main neighbor left: m_l
Input: distance direct neighbor left: n_l
Input: distance main: m
Input: distance direct neighbor right: n_r
Input: distance main neighbor right: m_r
Input: tolerance: τ
Savings: last weights: w_0, w_1, w_2
Output: fused minimum distance: d

if $(m_l < n_l \cdot \tau)$ **then** $n_l{=}\infty$ **end if**
if $(m_r < n_r \cdot \tau)$ **then** $n_r = \infty$ **end if**
if $(m < n_l)$ **then**
  if $(m < n_r)$ **then** $w_0 = 0$
  else $w_0 = 1$
  end if
else
  if $(n_l < n_r)$ **then** $w_0 = -1$
  else $w_0 = 1$
  end if
end if
$S{=}\sum_i w_i$
$w_2 = w_1$
$w_1 = w_0$
$d = m$
if $(S < -1)$ **then** $d = n_l$ **end if**
if $(S > 1)$ **then** $d = n_r$ **end if**
return d

The outputs of the WF or EWMA are 12 filtered distances of available place around the quadrotor, which can be described by 12 sectors (Fig. 5). The EWMA is used to reduce noise.

The velocity vector $\vec{v}$, one part of vector $\vec{\aleph}$, is computed by the derivation of the corresponding lowest distance from the vector $\vec{d_f}$. If the derivative is under a certain threshold, it is set to zero to dismiss jumps from noisy measurements or obstacles which do not correspond to a movement. The derivatives are then filtered with a EWMA filter. The accuracy of the velocity vector $\vec{v}$ can be improved by the usage of the optical flow sensor. Since the system should also be operable in case of bad lighting conditions, when the OF sensor fails, the vector $\vec{v}$ does not rely on the OF data. An integration of the IMU with a complementary filter for velocity estimation did not lead to better results and therefore has not been used. The proposed approach showed good results for velocity estimation.

$$\mathbf{T} = \begin{pmatrix} E_9 & E_0 & E_3 & E_6 \\ E_{11} & E_2 & E_5 & E_8 \\ E_0 & E_3 & E_6 & E_9 \\ E_1 & E_4 & E_7 & E_{10} \\ E_3 & E_6 & E_9 & E_0 \end{pmatrix} \tag{11}$$

The task of the triple awareness fusion (TAF) is to determine the four distances $\vec{d_f}$ (2) as one fundamental base for the situation assessment. The idea is to reduce the 12 outputs of the EWMA E_i to 4 inputs, which can be processed by the collision avoidance. It is called "triple", because it always fuses three sectors to one common sector (Fig. 8). Each column of **T** is the input to a single TAF filter (11).

The TAF algorithm picks the minimum relevant distance as the available space into the respective direction. There are two reasons for why a simple min() function is not sufficient. The first reason is the noise. The minimum of two noisy signals is significantly lower than each single signal or their average. The example data of Fig. 7 show that the average of D_1 is 2.25 and the average of D_2 is 3.6, but the average of their minimum is -2.25. This effect, if not considered, leads to large errors.

That is why the TAF algorithm not only considers the current measurement, but also the last two measurements, without the drawback of a delay. This information is stored in the last weights w_1, w_2. These weights together with the current weight w_0 determine the current result.

The second effect, which needs to be taken into consideration, can be explained as data dependency. An obstacle located on one axis may move through the data fusion unintentionally into another axis. This is the case, because transversal sensors measure two axes. For example a wall on the right is measured by US sensor 3 (d_3^{US}) (Fig. 6), but also by number 1 (d_1^{US}). If we would compute the front distance only with $\min(d_{11}^{US}, d_0^{US}, d_1^{US})$, the wall on the right would move to the front leading to erroneous results. To solve this problem, the TAF algorithm discards those measurements, when the corresponding neighbor is smaller because in that case the obstacle belongs to the other axis.

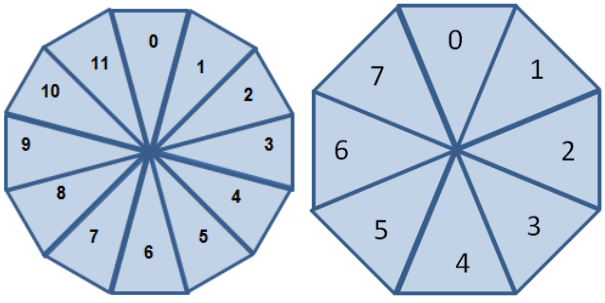

FIGURE 6. Left: 12 sectors around the quadrotor (WF + EWMA output): Correspondents to the position of the 12 US sensors $d_0^{US} \cdots d_{11}^{US}$ Right: 8 sectors around the quadrotor representing the 2×8 IR sensors.

The concept of the TAF can be visualized best in Fig. 8. The five inputs are the straight main distance (red), the direct neighbor distances left and right (yellow) and the main neighbor distances left and right (blue). Now, a direct neighbor is discarded (set to infinity), if the corresponding main neighbor is smaller. In the next step of the TAF algorithm the minimum value is picked between the sensor measurement of the main and direct neighbors. This is not only determined by the

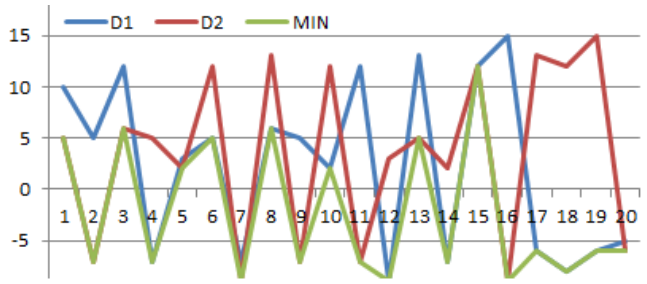

FIGURE 7. Minimum of two noisy signals D1 and D2.

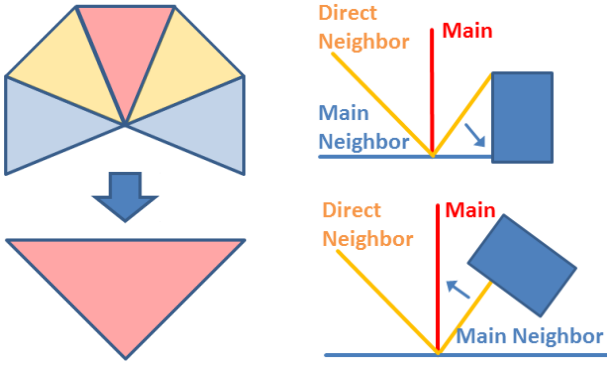

FIGURE 8. TAF concept.

current measurements but also by the last three measurements and their added weights. Hence, an obstacle on the side is ignored by the controller on the front (Fig. 8 right top) and a diagonal obstacle is controlled via both axes as intended (Fig. 8 right bottom).

D. SITUATION ASSESSMENT (SAs)

Fig. 9 shows the concept of the CA in more detail. Each axis can be controlled independently with one SAs and one DC block. The DC block is explained in the next subchapter, while this subchapter explains the situation assessment containing two identical SAs blocks.

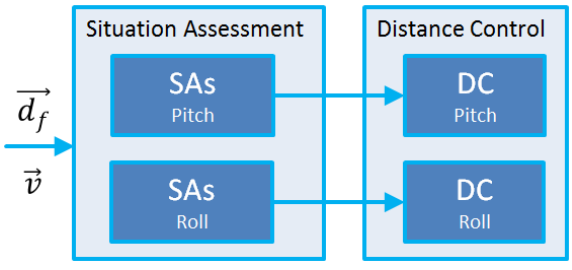

FIGURE 9. CA concept.

Under the constraint of a fixed given height, the quadrotor has only two independent translational axes which it has to manipulate for collision avoidance: x (forth / longitudinal) and y (strafe / lateral). Two independent axes mean there are four independent minimal distances (available spaces) around the quadrotor which are sufficient to describe the problem. Diagonal distances or measurements into two axes are a vectorial composition of the two orthogonal axes. Those diagonals are then expressed by the two orthogonal axes via trigonometric equations. Each two distances and a velocity correspond to one axis, one SAs and one DC: One for pitch (x) and one for roll (y). For simplification roll and pitch are considered to be independent, but the attitude can also be expressed via a quaternion without changes in the concept. Any further or other distance necessarily is a composition out of the mentioned four distances, since only the minimum distance is relevant. That is why the problem of control can be sufficiently handled by these distances $\vec{d}_f$ and their respective velocity vector $\vec{v}$.

Hence the problem of collision control can be solved with two fixed-axis controllers, one for longitudinal control and one for lateral control. Our experiments with distance control have shown that a different concept using more controllers does not assure stability. The reason is that the controllers hobble each other and continuity is not given due to switching between sensors and directions as an effect of the sensor noise and dynamic of the system. Therefore two fixed-axis controllers are supposed.

The problem of situation assessment can be generalized with respect to its characteristics and symmetry. The reaction of the system on obstacles at the front and on those at the rear is analogous. This information can be incorporated in the sign of the error. An obstacle at the front and an error of 10cm means that the quadrotor is further away than intended. In this case it should fly 10cm forward. An obstacle at the rear with an error of 10cm means that the quadrotor is too close. Again it needs to fly 10cm forward to avoid the obstacle at the rear. Although both situations are different, a similar reaction is intended. That's why the same error and therefore the same controller can be used. So the SA for each axis can be further simplified, if we describe it with a positive distance d_+ aligned with the corresponding axis and a negative distance d_- opposite to that axis. Now the controllers for both directions of both axes can be implemented analogous. For the x-axis (12) applies:

$$\begin{pmatrix} d_{Front} \\ d_{Rear} \end{pmatrix} = \begin{pmatrix} d_+ \\ d_- \end{pmatrix} \tag{12}$$

Hereby no switching between different controllers is necessary and continuity of the control is guaranteed, if an obstacle appears and disappears at the front or/and at the rear. This design increases the stability of the system significantly.

With two thresholds d_{th} (*threshold distance*) and d_{min} (*minimal distance*) every distance is classified in one of three areas: *Danger Area*, *Near Area* and *Free Area* (Fig. 10). In *Free Area*, if the distance is higher than d_{th}, the CA algorithm is inactive. In *Near Area*, if the distance is between d_{min} and d_{th}, the velocity is controlled. In *Danger Area* the distance to the obstacle is controlled to keep d_{min}. The distance is also controlled, if it is in *Near Area*, but was previously below d_{min}. That enables the quadrotor to keep its distance to obstacles.

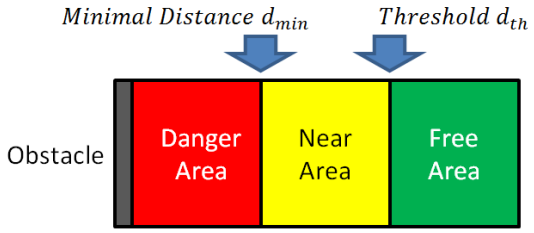

FIGURE 10. Area Classification.

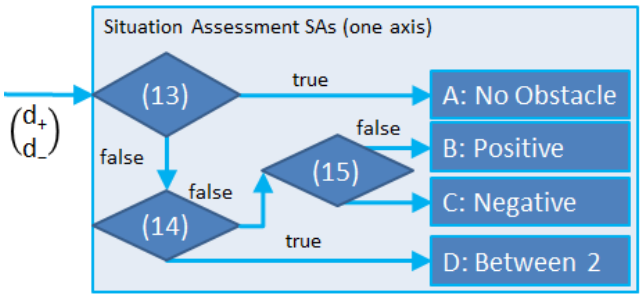

FIGURE 11. Situation assessment state machine.

The situation assessment distinguishes four different states for every axis (Fig. 11). The current state is determined by (13), (14) and (15):

$$d_{th} < min(d_+, d_-) \tag{13}$$

$$(d_+ < d_{th}) \,\&\, (d_- < d_{th}) \,\&\, (d_+ + d_- < 2 \cdot d_{min}) \tag{14}$$

$$d_+ < d_- \tag{15}$$

If both distances correspond to *Free Area*, the SAs is in state "A". This means there is no obstacle and CA is off. Otherwise and if (14) is true, the current state is "D" and there is an obstacle on both sides. The quadrotor has to control its position in the center of them. If (13) and (14) are false, the minimum of the distances d_+ and d_- determine the current state. In this case the quadrotor controls only the distance to the closer obstacles to avoid probable collision. This is sufficient, because there is enough space in the opposite direction. Furthermore, the distance measurements are more reliable, because shorter distances can be measured with lower noise and errors than long distances. This can also be seen in the evaluation data (Fig. 15-17), where long distances have a much higher variance and high errors than short distances.

Fig. 12 summarizes the system in a control loop. The set point depends on the state of the situation assessment.

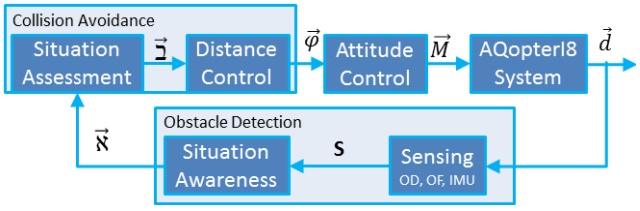

FIGURE 12. Control Loop.

Matrix $\vec{\Sigma}$ includes all information needed for control. $\vec{\varphi}$ is a two dimensional vector with the set points roll and pitch for the attitude control. The output of the attitude control are the set points for the motors $\vec{M}$, which manipulate the system and its distances $\vec{d}$ towards obstacles. These distances are measured by the sensors. Hence, the obstacle detection is the feedback of the control loop.

E. DISTANCE CONTROL

Fig. 13 illustrates the main principles of the distance control for one axis. Its main task is to control the distance from an obstacle to keep the *minimal distance* d_{Min}, whenever the measured distance d falls below. It is deactivated, if the distance is greater than the *threashold distance* d_{th}. The output of the DC is the angle φ. Two identical DC blocks compute the set points of the attitude control for roll α and pitch β (Fig. 9).

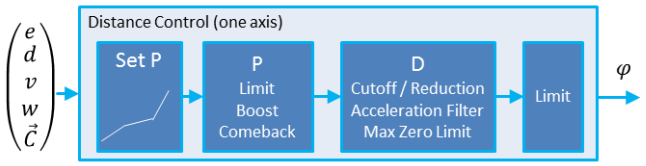

FIGURE 13. Principle Distance Control: DC block for one axis.

For control purpose, a modified PID or state space controller is used. It is extended by an error depended growth change of the proportional part. This modification divides the proportional part into three sections and improves the stability of the system. In the first section the growth of the proportional part is balanced, so the controller corrects a small error, but the reaction is not too abrupt to prevent overshooting. Next the growth is reduced, to improve the stability. In the third section the growth is raised, to prevent collisions.

Depending on the direction of movement towards the obstacle and the available distance the proportional part is further limited or boosted.

For the differential part of the controller the previously explained velocity is used which leads to a state space controller. A distance dependent cutoff is used to prevent the differential part from steering the quadrocopter towards the obstacle. It is also used, as the sensor measurements become noisier with higher distance. Otherwise this could lead to unwanted jumps in the differential part and instability. Similarly in case of an erroneous high velocity due to jumps in the distance measurement, the differential part is set to zero.

V. IMPLEMENTATION

All the functions were implemented on the 32bit AVR UC3A0512 microcontroller [32] and programmed in C. The implementation follows the mentioned concept with modules for every building block. For control, debugging and evaluation a ground station has been designed using a PC with Qt SDK (software development kit) running in C++ (Fig. 14).

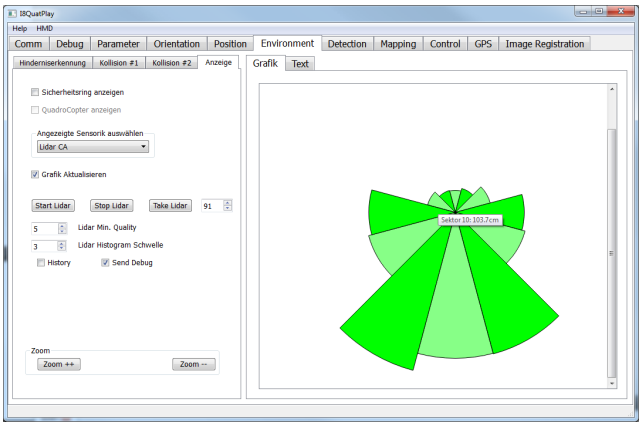

FIGURE 14. Qt-based ground station.

The *Minimal Distance* d_{min} was set to 1.4m, so that every tip of propellers keeps a distance of approximately 1m to any obstacle to ensure collision avoidance. For the *Threshold Distance*, $d_{th} = 2.2$m has been chosen. This is about the maximum reliable distance measurement of the ultrasonic sensors.

VI. EVALUATION

The system has been evaluated for more than 100 experimental tests. In the first empiric experiments the parameters of the system have been optimized. Then, for evaluation of the system, its performance during three different test cases has been analyzed and the results are presented.

The test cases are a stationary flight in a corner, a stationary flight between two walls and a flight in a room with a person moving towards the quadrotor. Every experiment has been repeated 10 times, lasting about 3 minutes, but only 60s has been display. These short time periods were chosen both to have many repetitions with comparable data under the same conditions and to avoid corruption of the data by the voltage drop of the batteries.

During all experiments the quadrotor operated fully autonomously without manual input or external devices. An external computer operating as ground station was used to receive telemetry data such as the evaluated values.

It is not claimed, that all parameters are optimized in research prototype. That is why the performance of the system is analyzed by its data fusion output and control input, which can be seen as almost true, as otherwise a failure would be clearly seen in the data as well (compare Fig. 15 and Fig. 17).

A. STATIONARY: CORNER

The idea of the first test case was to investigate the obstacle detection, collision avoidance and autonomous flight behavior in the simplest scenario. Since the system was designed for distance controlled flight, stationary obstacles like one wall in the front and on the left side are sufficient for autonomous stationary flight. In this experiment the quadrotor has been

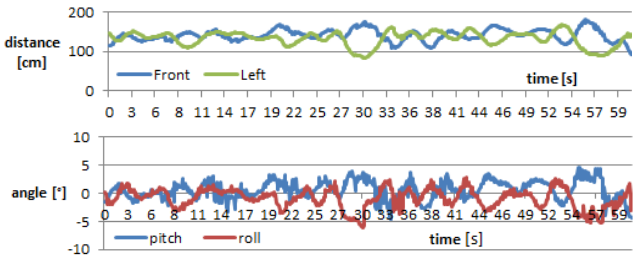

FIGURE 15. Test Case A (corner, run 1). Top: Data fused distances to the wall, input to the DC. Bottom: Measured angle from IMU, manipulated by the DC.

pushed into a corner till the front and left distance was below d_{min}. Consequently the collision avoidance automatically activated, controlled the flight of the quadrotor and kept its position in the corner by controlling the distance to the front and to the left. Table 1 shows the results. The mean distances measured from the center of the quadrotor towards the walls were 144cm and 135cm. Accordingly the mean distance between the tip of the propellers and the wall was about 1m as desired.

TABLE 1. Results: Test case A (stationary in corner).

B	Mean Distance [cm]		Standard Deviation [cm]	
Run	Front	Left	Front	Left
1	141.8	133.0	14.6	18.2
2	150.1	138.9	17.2	19.3
3	144.9	140.9	16.6	15.4
4	141.8	137.2	22.5	22.1
5	143.7	134.6	11.4	14.2
6	140.1	135.3	19.7	19.5
7	139.0	135.4	16.3	19.2
8	145.5	139.5	18.7	14.6
9	137,7	128.7	15.3	22.1
10	150.0	131.7	22.3	17.7
Mean	144.1	135.5	17.5	18.2

The standard deviation of the measured distances was between 11cm and 23cm with its mean at about 18cm. For run 1 (one typical case) Fig. 15 displays the measured distances received from the data fusion as the input to the controller and the measured resulting angles as its output. It seemed to be sufficient to record only these results to estimate the true distance, as a failure in the data fusion would result in a significant jump. However, this jump is not seen from the data.

Fig. 15 shows minor oscillating of the distance. This means that the quadrotor is moving within a maximum range of about 1m. This emphasizes the low damping of the system, as the control input is only in the range of about $+/-5°$. That means the quadrotor performs only small corrections. More dynamic would be possible, but it would lead to a higher oscillation and did not show better results, due to the noisy characteristic of the used sensors.

The low friction of a free flying system in combination with the propeller inducted wind reflection from the wall intensifies the difficulty of distance control. The minimum measured distance was 85cm for this run and the peak to peak amplitude of the oscillation is in most cases far below the maximum of ca. 1m.

B. STATIONARY: BETWEEN TWO WALLS

In test case B the quadrotor has been flown to the end of a dead-end tunnel. This means there was a wall in front and a wall on both sides left and right of the quadrotor. This experiment investigates the distance control for the case of two obstacles and the quadrotor in between them. The width of the tunnel was about 1.8m.

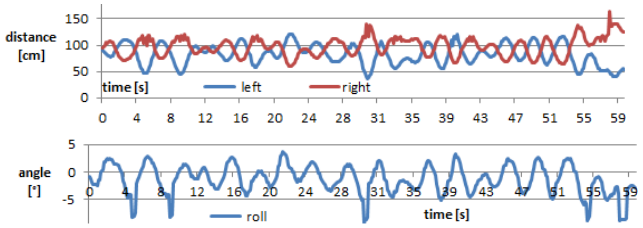

FIGURE 16. Test Case B (between two walls, run 1). Top: Data fused distances to the wall, input to the DC. Bottom: Set point angle, output of the DC.

Fig. 16 displays the measured distances and the set points (angles) of the DC. The figure shows, that the system is oscillating between the two walls and keeps the distance to the walls on both sides. The control output is almost in the range of $+/-$ 5°. However the quadrotor approaches several (five) times a distance of 50cm to the left, which activates a boost and puts it back. All boosts happened on the left side, though theoretically the behavior should be symmetric. The reason for that might be a little drift in the orientation. Another explanation could be the hardware (motors, propellers, design), which might be a bit asymmetric.

TABLE 2. Results: Test case B (stationary between two walls).

B	Mean Distance [cm]		Standard Deviation [cm]	
Run	Right	Left	Right	Left
1	97.4	82.5	17.4	19.5
2	100.5	78.7	15.0	16.8
3	96.3	81.7	17.1	18.4
4	93.7	84.7	15.5	17.1
5	95.7	82.9	13.5	15.2
6	96.0	83.8	13.4	15.4
7	91.1	88.2	17.3	18.6
8	97.8	81.9	15.2	17.1
9	94.9	84.8	10.9	12.2
10	95.1	84.8	13.0	13.7
Mean	95.9	83.4	14.8	16.4

Table 2 summarizes the results of this experiment. In comparison to Table 1 it can be seen, that the mean distance as well as the standard deviation are lower. The explanation for this is quite simple. Because there was less space available, only about 90cm to each side, the set point for distance control had to be lower, leading to a lower mean distance.

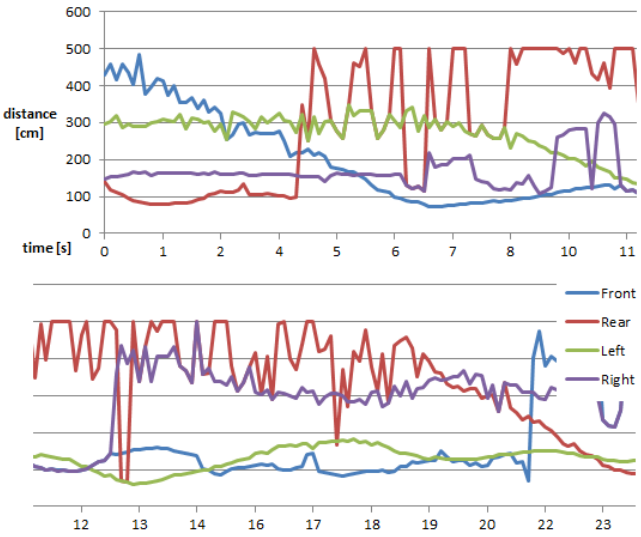

FIGURE 17. Test Case C: Dynamic collision Avoidance.

At first glance the lower standard deviation seems to be a bit unreasonable, as in the dead-end tunnel there are much higher turbulences because of propeller wind reflected from the walls. The explanation for this is the switching of the controller into a state with smoother control parameters $\vec{C}$. Since there is less space, the quadrotor cannot and should not react as aggressively as it would otherwise.

In case where there is no obstacle, or only one obstacle on one side, the quadrotor can possess a much higher velocity. The control parameters need to be more aggressive to slow down the quadrotor as necessary. This behavior can clearly be seen in the data.

C. DYNAMIC: PERSON & QUADROTOR

In this scenario a person and the quadrotor are together in a room. Initially the person stands behind the quadrotor and pushes it by approaching forward towards the front wall (first phase), then pushes it from the right side to the left wall (second phase) and finally from the front backwards to the rear wall (third phase.).

Fig. 17 depicts the results of run 5 of the dynamic collision avoidance test case. From the graph it can be seen, that in the first phase (seconds 0 – 7) the frontal distance decreases, while the right and the rear distance are almost constant. The quadrotor is pushed from behind towards the wall. During the second phase (seconds 7 – 14) the left distance decreases while the front distance is constant. This means that the distance controller keeps the front distance to the wall while the person pushes the quadrotor from the right to the left side. During the third phase (seconds 14 – 24) the rear distance decreases. Furthermore it can be clearly seen, that the data becomes noisy above a distance of about 2m but shows reliable results below this distance.

This experiment demonstrates that the quadrotor is capable of autonomous flight with people in its environment and can control its distance simultaneously to multiple obstacles from multiple changing directions. The speed of the moving obstacle (person) in this experiment was about 1m/s.

VII. CONCLUSION AND FUTURE WORK

The evaluation shows that the demonstrated solution is capable of autonomous flight and collision avoidance in multiple scenarios. A main contribution of this work is to disprove the myth that only expensive sensors like laser scanner are suitable for such a task and low-cost range finders are not. The evaluation showed that the system is capable to avoid collisions with obstacles such as walls and people while it can control its distance towards them.

In the previous work about distance controlled collision avoidance and fully autonomous flight using ultrasonic sensors only [18], [20], it was not possible to avoid collisions with persons, because the ultrasonic sensors failed to detect them reliably. The present paper demonstrated a solution to that limitation by fusing IR and US sensors.

Because the presented solution merges multiple low-cost sensors of different technologies (infrared, ultrasonic), it is more reliable and cheaper than state of the art solutions. Furthermore the presented approach is much easier to implement in case of mathematical complexity compared to existing approaches, while keeping the computational burden low, reducing development and maintenance costs. To the best of the authors' knowledge, no fully autonomous system exists, which is comparable in power.

The infrared sensors may be replaced by a low-cost laser scanner such as the RP Lidar. Such a solution would be slightly more reliable in case of better sensor coverage, however more expensive and heavier. A more powerful processor like an Intel Atom or the AMD T56N might be required in this case, which also increases costs and weight. Also the sample rate of the infrared sensors can be set higher compared to the frequency of a scanner. Nevertheless the RP Lidar could be a good improvement of the mentioned solution, which needs to be investigated in the future.

Another option for future work is the evaluation of the obstacle detection sensors by all sensors of the AQopterI8. This means the quadrotor would use besides the RP Lidar also the SV (stereo vision) system and the PMD nano (photonic mixing device). This can improve the performance and reliability of the system further, but it would also increase the weight and the costs of the system.

ACKNOWLEDGMENT

The authors would like to thank Thilo Müller, David Courtney and Qasim Ali for their contributions to this work.

REFERENCES

[1] M. Becker, R. C. B. Sampaio, S. Bouabdallah, V. de Perrot, and R. Siegwart, "In-flight collision avoidance controller based only on OS4 embedded sensors," *J. Brazilian Soc. Mech. Sci. Eng.*, vol. 34, no. 3, pp. 294–307, Jul./Sep. 2012.

[2] S. Bouabdallah, "Design and control of quadrotors with application to autonomous flying," Ph.D. dissertation, Faculte Sci. Techn. l'ingenieur, École Polytech. Fédérale Lausanne, Switzerland, 2007. [Online]. Available: http://biblion.epfl.ch/EPFL/theses/2007/3727/EPFL_TH3727.pdf

[3] S. Bouabdallah, M. Becker, V. de Perrot, and R. Siegwart, "Toward obstacle avoidance on quadrotors," in *Proc. 12th Int. Symp. Dyn. Problems Mech.*, Ilhabela, Brazil, 2007, pp. 1–10.

[4] J. F. Roberts, T. S. Stirling, J.-C. Zufferey, and D. Floreano, "Quadrotor using minimal sensing for autonomous indoor flight," in *Proc. MAV*, 2007, pp. 1–8.

[5] S. Scherer, S. Singh, L. Chamberlain, and M. Elgersma, "Flying fast and low among obstacles: Methodology and experiments," *Int. J. Robot. Res.*, vol. 27, no. 5, pp. 549–574, 2008.

[6] M. Blösch, S. Weiss, D. Scaramuzza, and R. Siegwart, "Vision based MAV navigation in unknown and unstructured environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Anchorage, AK, USA, May 2010, pp. 21–28.

[7] J. Engel, J. Sturm, and D. Cremers, "Accurate figure flying with a quadrocopter using onboard visual and inertial sensing," in *Proc. IEEE Int. Conf. Intell. Robot Syst.*, Oct. 2012.

[8] J. Engel, J. Sturm, and D. Cremers, "Camera-based navigation of a low-cost quadrocopter," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 2815–2821.

[9] S. Gronzka, "Mapping, state estimation, and navigation for quadrotors and human-worn sensor systems," Ph.D. dissertation, Technische Fakultät, Albert-Ludwigs-Univ. Freiburg, Breisgau, Germany, 2011.

[10] S. Grzonka, G. Grisetti, and W. Burgard, "A fully autonomous indoor quadrotor," *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 90–100, Feb. 2012.

[11] S. Shen, N. Michael, and V. Kumar, "Autonomous multi-floor indoor navigation with a computationally constrained MAV," in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, China, May 2011, pp. 20–25.

[12] M. Achtelik, A. Bachrach, R. He, S. Prentice, and N. Roy, "Autonomous navigation and exploration of a quadrotor helicopter in GPS-denied indoor environments," in *Proc. 1st Symp. Indoor Flight*, 2008. [Online]. Available: http://robot-chopper.googlecode.com/svn-history/r479/trunk/References/2009MIT.pdf

[13] K. Celik, S.-J. Chung, M. Clausman, and A. K. Somani, "Monocular vision SLAM for indoor aerial vehicles," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 1566–1573.

[14] MikroKopter. (Jan. 1, 2015). [Online]. Available: http://www.mikrokopter.de/

[15] AscTec. (Jan. 15, 2015). [Online]. Available: http://www.asctec.de/

[16] Aibotix. (Jan. 15, 2015). [Online]. Available: http://www.aibotix.de/

[17] Microdrones. (Jan. 15, 2015). [Online]. Available: http://www.microdrones.de/

[18] N. Gageik, T. Müller, and S. Montenegro, "Obstacle detection and collision avoidance using ultrasonic distance sensors for an autonomous quadrocopter," in *Proc. UAVveek Workshop Contrib.*, 2012.

[19] N. Gageik, J. Rothe, and S. Montenegro, "Data fusion principles for height control and autonomous landing of a quadrocopter," in *Proc. UAVveek Workshop Contrib.*, 2012.

[20] T. Müller, "Implementierung und evaluierung eines systems zur hinderniserkennung und kollisionsvermeidung für indoor-quadrokopter," diploma thesis, Aerosp. Inf. Technol., Univ. Würzburg, Germany, 2011.

[21] J. Rothe, "Implementierung und evaluierung einer höhenregelung für einen quadrokopter," bachelor thesis, Aerosp. Inf. Technol., Univ. Würzburg, Germany, 2012.

[22] G. Hoffmann, D. G. Rajnarayan, S. L. Waslander, D. Dostal, J. S. Jang, and C. J. Tomlin, "The Stanford testbed of autonomous rotorcraft for multi agent control (STARMAC)," in *Proc. 23rd Digit. Avionics Syst. Conf.*, Oct. 2004, pp. 12.E.4-1–12.E.4-10.

[23] S. Weiss, D. Scaramuzza, and R. Siegwart, "Monocular-SLAM–based navigation for autonomous micro helicopters in GPS-denied environments," *J. Field Robot.*, vol. 28, no. 6, pp. 854–874, 2011.

[24] Parrot. (Jan. 15, 2015). [Online]. Available: http://www.parrot.com

[25] K. Nonami, F. Kendoul, S. Suzuki, W. Wang, and D. Nakazawa, *Autonomous Flying Robots*. Tokyo, Japan: Springer-Verlag, 2010.

[26] R. Brockers, M. Hummenberger, S. Weiss, and L. Matthies, "Towards autonomous navigation of miniature UAV," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2014, pp. 645–651.

[27] H. A. Lauterbach and D.-I. N. Gageik, "Stereo-optische abstandsmessung für einen autonomen quadrocopter," bachelor thesis, Aerosp. Inf. Technol., Univ. Würzburg, Germany, 2013.

[28] E. Reinthal and D.-I. N. Gageik, "Positionsbestimmung eines autonomen quadrokopters durch bildverarbeitung," bachelor thesis, Aerosp. Inf. Technol., Univ. Würzburg, Germany, 2013.
[29] N. Gageik, C. Reul, and S. Montenegro, "Autonomous quadrocopter for search, count and localization of objects," in *Proc. UAV World Conf.*, 2013.
[30] N. Gageik, E. Reinthal, P. Benz, and S. Montenegro, "Complementary vision based data fusion for robust positioning and directed flight of an autonomous quadrocopter," *Int. J. Artif. Intell. Appl.*, vol. 5, no. 5, 2014.
[31] G. Verhoeven, "Using the MD4-1000 in archaeological research," in *Proc. UAVveek Workshop Contrib.* 2012.
[32] Atmel. (Mar. 2015). [Online]. Available: http://www.atmel.com

NILS GAGEIK was born in Aachen, Germany, in 1982. He received the Diploma (Dipl.-Ing., old German master) degree in computer engineering from RWTH Aachen University, in 2010. He is currently pursuing the Ph.D. degree in computer science with the University of Würzburg, Germany.

He has been a Research Assistant with the Department of Aerospace Information Technology, University of Würzburg, since 2010. Besides teaching, he is involved in research on autonomous quadrotors. He leads the AQopterI8 Project, which aims to research and develop autonomous quadrocopters for indoor applications. He won the University Promotion Prize for his work in 2014.

PAUL BENZ was born in Aschaffenburg, Germany, in 1990. He received the B.Sc. degree in aerospace computer science from the University of Würzburg, in 2013. He is currently pursuing the M.Sc. degree in computer science.

He participated as a member of the AQopterI8 Project in the field of obstacle detection and collision, during his bachelor's thesis and an internship. Since 2014, he has been a Student Assistant with the Department of Aerospace Information Technology.

SERGIO MONTENEGRO was born in Guatemala in 1959. He received the B.S. degree from the Universidad del Valle, in 1982, and the M.S. degree in computer science and the Ph.D. degree from the Technical University of Berlin, in 1985 and 1989, respectively.

He was a Research Assistant in Distributed Computing with the Fraunhofer-Gesellschaft Institut, Berlin, from 1985 to 2007. From 2007 to 2010, he was the Leader with the Department of Central Avionic Systems, DLR, Bremen. Since 2010, he has been a Professor and the Chair of the Department of Aerospace Information Technology with the University of Würzbug, Germany.