

# Optical Engineering

OpticalEngineering.SPIEDigitalLibrary.org

## **Prediction of microunmanned aerial vehicle flight behavior from two-dimensional intensity images**

Martin Rebert  
Stéphane Schertzer  
Martin Laurenzis

**SPIE.**

Martin Rebert, Stéphane Schertzer, Martin Laurenzis, "Prediction of microunmanned aerial vehicle flight behavior from two-dimensional intensity images," *Opt. Eng.* **58**(5), 053101 (2019), doi: 10.1117/1.OE.58.5.053101.

# Prediction of microunmanned aerial vehicle flight behavior from two-dimensional intensity images

Martin Rebert, Stéphane Schertzer, and Martin Laurenzis\*  
French-German Research Institute of Saint-Louis, Saint-Louis, France

**Abstract.** The increasing number of microunmanned aerial vehicles (MUAVs) is a rising risk for personal privacy and security of sensitive areas. Owing to the highly agile maneuverability and small cross section of the MUAV, effective countermeasures (CMs) are hard to deploy, especially when a certain temporal delay occurs between the localization and the CM effect. Here, a reliable prediction of the MUAV flight behavior can increase the effectiveness of CMs. We propose a pose estimation approach to derive the three-dimensional (3-D) flight path from a stream of two-dimensional intensity images. The pose estimation in a single image results in an estimation of the current position and orientation of the quadcopter in 3-D space. Combined with flight behavior model, this information is used to reconstruct the flight path and to predict the flight behavior of the MUAV. In our laboratory experiments, we obtained a standard deviation between 1 and 24 cm in a five-frame prediction of the 3-D position, depending in the actual flight behavior. © The Authors. Published by SPIE under a Creative Commons Attribution 4.0 Unported License. Distribution or reproduction of this work in whole or in part requires full attribution of the original publication, including its DOI. [DOI: 10.1117/1.OE.58.5.053101]

Keywords: unmanned aerial vehicles; computer vision; prediction; pose estimation.

Paper 190348 received Mar. 11, 2019; accepted for publication Apr. 23, 2019; published online May 20, 2019.

## 1 Introduction

In the recent past, the number of incidents with intentional and unintentional misuse of small or microunmanned aerial vehicles (MUAVs) has increased due to the increasing number of commercially available small quadcopters. In their annual reports,<sup>1</sup> the Federal Aviation Administration has pointed out an increasing number of sightings and close encounters of MUAVs reported in civilian air traffic. Further, the first serious incidents where a MUAV crashed into an air plane have occurred.<sup>2</sup> Beside this threat to civilian air traffic, misuse of MUAVs can also harm personal and public privacy and create security issues.

MUAVs can operate in highly agile flight patterns and are hard to detect due to having small cross sections.<sup>3–7</sup> Thus, in the past decade, several groups have been working on the problem of efficiently detecting, tracking, and classifying MUAVs in different scenarios. Detection and tracking of MUAVs have been studied using single millimeter-wave RADAR,<sup>8</sup> optical (VIS/SWIR)<sup>9</sup> sensors or within a multisensor approach using passive/active imaging, acoustics, and RADAR in various field trials.<sup>10–13</sup> Further, classification of MUAV has been investigated using image-based deep-learning and high-level data-fusion approaches.<sup>14–16</sup> Classification can also be used to distinguish MUAVs from, e.g., birds.

Different sensors (such as RADAR, acoustics, and optics) will be embedded in a counter-MUAV system where sensor information will be processed to reliably detect/track, classify, and localize MUAV. Based on this information, decisions are made and countermeasures (CMs) are engaged. Owing to the internal processing time of information generation and decision-making as well as external time constraints (e.g., propagation of a projectile), the effect on the MUAV

will occur after a certain inherent time delay  $\Delta t$ . Owing to the agile operation and small cross section, the probability of successful CMs is relatively low.<sup>17</sup> Here, the prediction of the MUAV position at the moment of effect could significantly increase the success rate.

First, works that predict the MUAV position in the image plane of the succeeding frame (frame-to-frame prediction) of a video stream<sup>18,19</sup> using a linear motion model have been presented. In our approach, we predict the position of the MUAV in three-dimensional (3-D) space and on a longer time scale  $t_P$  (with  $t_P \approx \Delta t$ ) or over  $n$ -frames. For a reliable result, we determine the 3-D position of the MUAV from pose estimation in a two-dimensional (2-D) intensity images and apply a navigation model to simulate the flight behavior.

## 2 Prediction of Microunmanned Aerial Vehicle Position in the Image Plane

Based on linear quadratic estimation, Kálmán<sup>20</sup> developed a filter to either reduce measurement noise effects and to estimate values (or determine the most probable value). In the computer vision community, for instance, the Kalman filter approach is widely used for tracking and navigating the MUAV.<sup>18,19</sup> The Kalman filter  $f(\cdot)$  is used to predict the MUAV state vector  $X_i$ , which contains a set of parameters, from a current measurement  $y_i$  and the previous state vector  $X_{i-1}$ , as defined in Eq. (1). Here,  $\omega$  describes noise characteristics

$$X_i = f(X_{i-1}, y_i, \omega). \quad (1)$$

Depending on the application (e.g., navigation or tracking), the state vector summarizes several physical parameters [such as position, velocity, angles (roll, yaw, pitch), and angular velocities] to describe the current flight state. In our approach, the state vector is defined as  $X_i = [x_i, \dot{x}_i, \ddot{x}_i]^T$  containing only the MUAV position  $x_i$ ,

\*Address all correspondence to Martin Laurenzis, E-mail: [martin.laurenzis@isl.eu](mailto:martin.laurenzis@isl.eu)

velocity  $\dot{x}_i$ , and acceleration  $\ddot{x}_i$  that are the first and second temporal derivatives of the position in the  $i$ 'th frame. Higher order of temporal derivatives [the jerk (third) and the snap (fourth)] could be of interest in future algorithms and are neglected in the current code.

We developed a tracking and prediction filter based on Murphy's "Kalman filter toolbox for Matlab"<sup>21</sup> and the tutorial of Welsh and Bishop.<sup>22</sup> Our filter was designed to predict a MUAV position in a  $n$ -frame future from a continuous data stream and has two stages: a Kalman filter to estimate the current state vector  $X_i$  and an  $n$ -frame prediction filter to estimate the future state  $\hat{X}_{i+n}$ .

The Kalman filter, as described in Sec. 5, returns a state vector  $X_i$  at time  $t_i$ , containing position, velocity, and acceleration. Then, the MUAV state vector  $\hat{X}_{i+n}$  at time  $t_{i+n}$  can be predicted using the motion equation approach, as defined in Eq. (2). The predicted position  $\hat{x}_{i+n}$  can be extracted by Eq. (3). We define the motion prediction equations as

$$\hat{X}_{i+n} = \begin{bmatrix} 1 & n & \frac{1}{2}n^2 \\ 0 & 1 & n \\ 0 & 0 & 1 \end{bmatrix} X_i \quad (2)$$

and

$$\hat{x}_{i+n} = [1 \quad 0 \quad 0] X_{i+n}. \quad (3)$$

The complete  $n$ -frame prediction procedure is illustrated in Algorithm 1 as a pseudocode. The computation process returns a predicted position  $\hat{z}_{i+n}$  and needs a set of input data consisting of the current image frame  $F_i$ , the previous MUAV state vector  $X_{i-1}$ , and the number of frames  $n$  as the prediction range. First, the MUAV position  $x_i$  in the current image has to be determined by, for instance, a tracking

**Algorithm 1** Pseudocode for the  $n$ -frame prediction of 2-D MUAV position in the image plane.

**Input:** Current frame  $F_i$ , previous state vector  $X_{i-1}$ , coefficient  $n$

**Output:**  $n$ -frame prediction of 2-D MUAV position  $\hat{z}_{i+n}$

- 1: Determine the MUAV position in frame:  $F_i \rightarrow x_i$
- 2: Apply the Kalman filter  $f(X)$
- 3: Predict the state vector in the  $i + n$ 'th frame:  $X_{i+n}$ . [Eq. (2)]
- 4: Extract the position in the  $i + n$ 'th frame:  $\hat{x}_{i+n}$ . [Eq. (3)]
- 5: **return**  $\hat{x}_{i+n}$

algorithm. This step will be discussed in the next paragraph. From these position measurements, the current state vector is calculated using the Kalman filter and the MUAV state  $\hat{X}_{i+n}$  at frame  $i + n$  is predicted. Finally, the estimated future position  $\hat{x}_{i+n}$  is extracted [see Eqs. (2) and (3)].

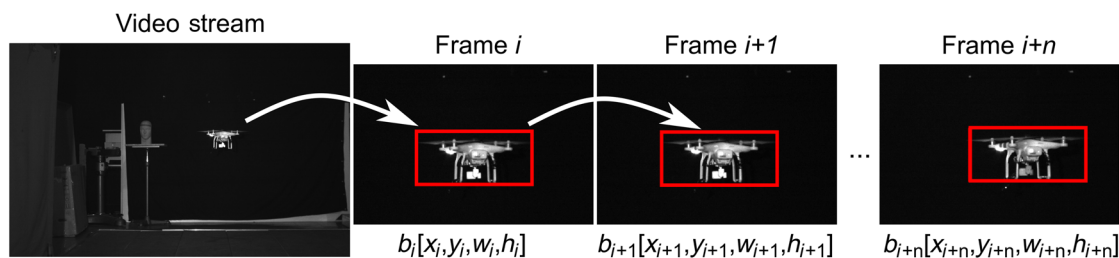
In Fig. 1, some frames of a video stream are presented. This video was recorded inside our laboratory, showing a small quadcopter MUAV (type DJI Phantom 3) flying around in the camera field of view. The MUAV is tracked with a state-of-the-art tracker based on the consensus-based matching and tracking of key points (CMT).<sup>23,24</sup> In this algorithm, the image is transformed to an alternative base using feature detection from accelerated segment test (FAST)<sup>25,26</sup> and binary robust invariant scalable key points of features<sup>27</sup> as feature detector and descriptor subroutines.

The tracker employs clustering of correspondences as the central idea to distinguish inlier and outlier key points. This approach improves the tracking results, even when correspondences are located on deformed parts of the object. As the flying MUAV shows very rapid aspect changes due to the angle between the vision system and the target, this tracker is well adapted to this particular situation. The tracking algorithm returns a vector containing the MUAV position ( $x_i$  and  $y_i$  are the column and row positions of the MUAV, respectively) and the size of the bounding box (width and height:  $w_i$ ,  $h_i$ ). In Fig. 1, some frames of the video stream illustrate the CMT tracking results.

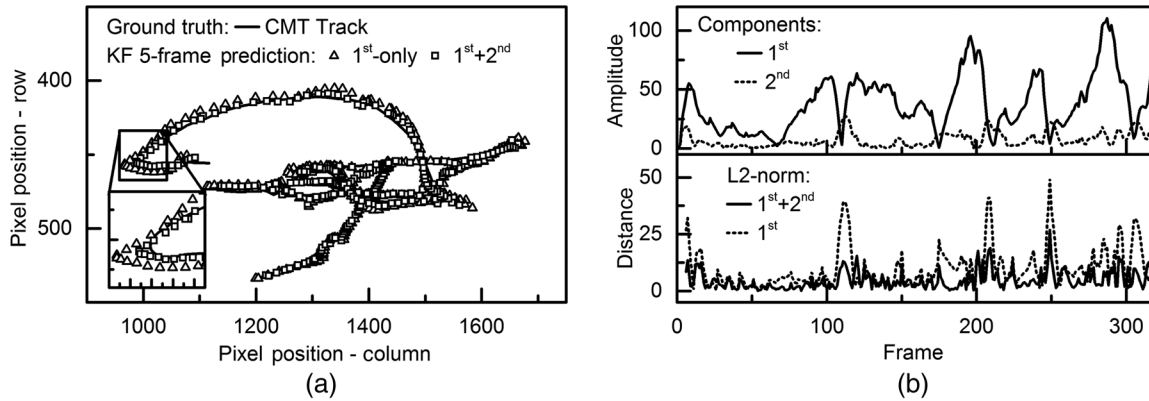
An exemplary result of our tracking and prediction algorithm (Algorithm 1) is presented in Fig. 2. In Fig. 2(a), a five-frame prediction result from our approach using first and second derivatives ( $\square$ ) is compared to a simple prediction approach using the first derivative only ( $\Delta$ ). Both prediction paths are compared to the ground-truth CMT tracking data ( $—$ ). As long as the MUAV flight operation is slow and smooth, only small differences between ground truth and predicted paths are observed. But at turning points and during high-agile flight operation, the predictions are less accurate, especially for the simple prediction approach.

A deeper discussion of the prediction results is presented in Fig. 2(b). In the top diagram, the amount of single flight behavior components are plotted for each frame as the amplitude of the first and second derivatives [ $n|\dot{x}|$  ( $\cdots$ ) and  $\frac{1}{2}n^2|\ddot{x}|$  ( $—$ )]. It is obvious that rapid changes in the first derivative lead to peaks in the amplitude of the second derivative.

Furthermore, in the bottom diagram of Fig. 2(b), the prediction error is illustrated as the mismatch distance between ground truth and predicted position given by the L2-norm distance:  $d_{L2} = \|x_i - \hat{x}_i\|_2$ . For the simple prediction approach ( $\cdots$ ), high errors can be observed in frames that are related to high agile or dynamic MUAV flight operations (see frames 110 to 125, 165 to 195, 205 to 220, etc.). In this



**Fig. 1** CMT of the key points in a MUAV video sequence.



**Fig. 2** Prediction of MUAV flight behavior in the 2-D image plane using the CMT tracker and the first and second derivatives ( $\dot{x}$  and  $\ddot{x}$ ): (a) ground-truth CMT track (—) and five-frame track prediction using  $\dot{x}$ -only ( $\Delta$ ) and  $\dot{x}-\ddot{x}$  ( $\square$ ) algorithms, (b) the amount of the  $n|\dot{x}|$  (---) and  $n^2|\ddot{x}|$  (—) components and the prediction errors that are the L2-norm distances from the original track.

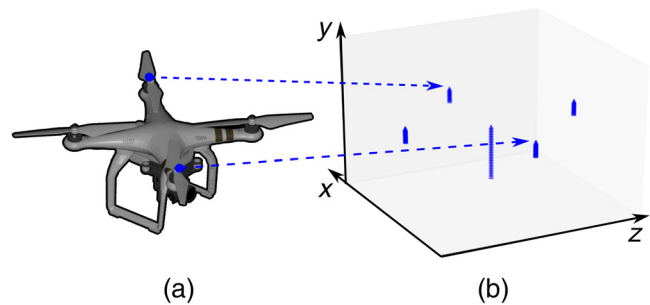
specific sequence, we obtain a mean prediction error of 10.1 pixels. This mismatch distance can be reduced significantly by additionally employing the second derivative (—) in our prediction approach. With this algorithm it is possible to reduce all previously occurring peaks significantly and a mean prediction mismatch distance of 4.7 pixels is reached.

### 3 Prediction of Microunmanned Aerial Vehicles Position in Three-Dimensional Space

In three-dimensional (3-D) space, the flight path prediction has to take into account the position and motion in all spatial dimensions to reflect the complex MUAV flight behavior. Here, we want to introduce a pose estimation algorithm to derive the 3-D position and orientation of the MUAV from the 2-D intensity images. Further, we use the obtained data in an aerodynamic flight behavior model to calculate the acceleration as an additional measurement value. Then, both the determined 3-D position and the acceleration values are fed into the aforementioned Kalman filter and prediction algorithm to return the probable future MUAV position in the 3-D space.

Flight behavior of a quadcopter MUAV, for instance, is modeled in the algorithm used for navigation and control.<sup>28–33</sup> These algorithms analyze the dynamic behavior of the spatial and the angular motion to control the power consumption of the four rotors individually. Typically, these models combine a global or inertial and a MUAV or body coordinate system. In our approach, we rely on the global coordinate system of the optical sensing device. In the coordinate system the  $x$ - and  $y$ -directions are the width and the height corresponding to the previous pixel row and column directions, respectively. Then, the  $z$ -direction is the range.

Further, we developed a simplified flight behavior model to describe the MUAV motion from the thrust  $T$ , the gravitational force  $G$ , and the aerodynamic drag force  $D$ , which can be derived from some fundamental parameters such as position and orientation in 3-D space, velocity, acceleration, and few boundary conditions (aerodynamic constants and parameters). In principle, we assume that the MUAV is a floating object (the vertical component of the thrust compensates for earth gravitation). Details of this flight behavior model can be found in Secs. 6 and 7.



**Fig. 3** Representation of a drone as (a) a computer graphics model and (b) a skeleton, as used in the pose estimation of the quadcopter.

Position and orientation of the MUAV in 3-D space can be derived from pose estimation. For this task, the quadcopter used in our experiment can be modeled by a skeleton model consisting of four small vertical segments equally distributed around a bigger vertical segment, as shown in Fig. 3. The four small segments represent the rotation axes of each propeller and the bigger one represents the body of the quadcopter.

The use of a skeleton model has the benefit of not being too specific to the investigated MUAV. Later adaption to another MUAV can easily be done by changing the model. The only values that have to be known are the number of propellers and their distance to the center.

For the investigated vehicle, the DJI Phantom 3,<sup>34</sup> there are four propellers located 175 mm from the center. The number of points per segment can also vary, but it does not have a great influence; we usually use 20 points per segment. The skeleton points are designated by  $S = \{s_j\}$  when placed at the origin. The pose of the skeleton corresponding to the image  $i$  is represented by a vector containing six values  $\phi_i = [\theta_x \theta_y \theta_z x y z]$ , where  $x$  is the  $x$ -coordinate and  $\theta_x$  is the angle of rotation around this axis. Analogous definitions are made for the  $y$ -axis and the  $z$ -axis. Here,  $S_i(\phi_i) = \{s_{i,j}\}$  is the transformed skeleton (rotation and translation) and stands for the points at pose  $\phi_i$

$$S_i(\phi_i) = R_i S + t_i, \quad (4)$$

$$R_i = R_z(\theta_z) R_y(\theta_y) R_x(\theta_x), \quad (5)$$



**Algorithm 2** Pseudocode for the pose estimation of the MUAV from two successive views.

**Input:** Current frame  $F_i$ , previous pose  $\phi_{i-1} \in \mathbb{R}^6$ , previous detected corners  $C_{i-1}$ ,  $b_i$  from the CMT, threshold  $\epsilon$ , skeleton  $S$

**Output:**  $\phi_i$ ,  $C_i$

- 1: Detect corners  $C_i$  within  $b_i$  using the FAST algorithm
- 2: Match  $C_i$  with  $C_{i-1}$  using BRIEF descriptors and Hamming distance  $\rightarrow M_{i,i-1}$
- 3: Filter out propellers and background motion from  $M_{i,i-1}$  by estimating the essential matrix
- 4: Use  $\phi_{i-1}$  to project  $S$  into  $F_i \rightarrow S_{i-1,proj}$
- 5: Associate  $C_{i-1}$  with  $S_{i-1,proj}$ ,  $(c_{i-1,j}, s_{i-1,proj,j}) \in C_{i-1} \times S_{i-1,proj}$  such that  $d(c_{i-1}, s_{i-1,proj}) \leq \epsilon \rightarrow C_{i-1,opti}, C_{i,opti}$
- 6: Align  $S$  on rays issued from  $C_{i,opti}$  using  $\phi_{i-1}$  as initial conditions  $\rightarrow \phi_i$
- 7: **return**  $\phi_i$ ,  $C_i$

$$t_i = [x \ y \ z]^T, \quad (6)$$

where  $R_x(\theta_x)$  is the rotation of angle  $\theta_x$  around the  $x$  axis.

The pose estimation is based on the use of two successive views. The pseudocode is given in Algorithm 2. First, corners are detected in the region  $b_i$  of the current frame  $F_i$  using the FAST corner detector<sup>26</sup> with a nonmaximum suppression to obtain a better distribution of the corners  $C_i = \{c_{i,j}\}$  over the MUAV, as illustrated in Fig. 4(a). The detected corners are described using the BRIEF<sup>35</sup> algorithm and matched with the corners from the previous frame,  $C_{i-1}$ , to obtain the matches  $M_{i,i-1}$ . At this point, the set of matches  $M_{i,i-1}$  contains corners located on the drone but also mismatches, corners on the background and corners on the propellers. In the next step, the unwanted corners (mismatch, background, and propellers) are filtered out by estimating the essential matrix<sup>36</sup> [see Fig. 4(b)]. Hereafter  $M_{i,i-1}$ ,  $C_{i-1}$ , and  $C_i$  correspond to the filtered data set.

To estimate the pose of the drone, we need to associate the detected corners with points on the skeleton. A pinhole camera model is used to project the skeleton in the image plane. The camera is placed in the origin of the coordinate system with the standard axes, i.e., the depth of the scene is along the positive  $z$ -axis and the  $y$ -axis points to the ground. Using homogeneous coordinates, the camera projects the skeleton  $S_i$  to  $S_{i,proj}$  according to

$$S_{i,proj} = PS_i, \quad (7)$$

where  $P$  is the  $4 \times 4$  homogeneous projection matrix.<sup>37</sup>

Further, in Fig. 4(c), we couple the detected corners and the projected skeleton points that are close together,  $d(c_{i-1}, s_{i-1,proj}) \leq \epsilon$ ,  $c_{i-1} \in C_{i-1}$  and  $s_{i-1,proj} \in S_{i-1,proj}$ . If a skeleton point satisfies the condition with several corners, we keep the closest couple. From this we obtain subsets of  $C_{i-1}$  and  $S_{i-1,proj}$  called  $C_{i-1,opti}$  and  $S_{i-1,opti}$ . It is then straightforward to obtain  $C_{i,opti}$ ,  $S_{i,opti}$ , and couples  $(c_{i,opti}, s_{i,opti})$  with  $c_{i,opti} \in C_{i,opti}$  and  $s_{i,opti} \in S_{i,opti}$ .

For each corner in  $C_{i,opti}$ , a ray  $r$  passing through the camera center (the origin) and the corner is computed. The pose of the drone in image  $i$ ,  $\phi_i$ , is then updated by minimizing the distance between each element of  $S_{i,opti}$  and their corresponding rays (see Fig. 5). The minimization is performed using  $\phi_{i-1}$  as a starting point

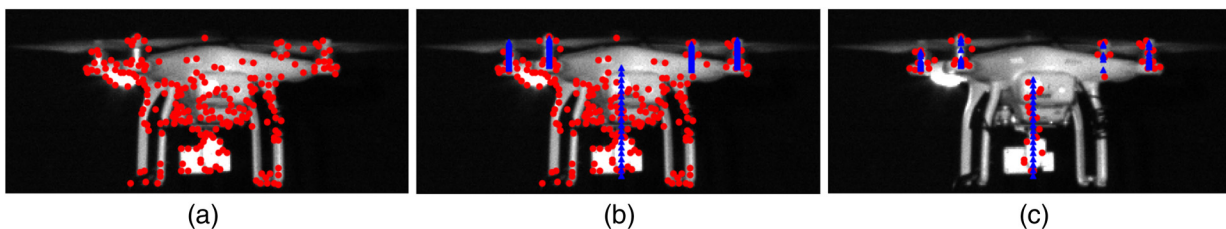
$$\phi_i = \arg_{\phi_i} \min \sum_j d(r_j, s_j)^2, \quad s_j \in S_{i,opti}, \quad (8)$$

$$r_j = P^{-1}c_{i,opti,j} = P^{-1}[x_c \ y_c \ 1 \ 1]^T, \quad (9)$$

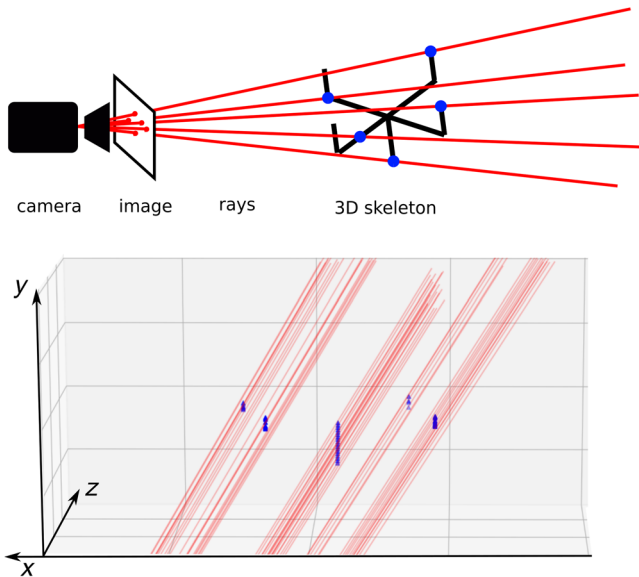
where  $d(r_j, s_j)$  is the orthogonal distance from the point  $s_j$  to the line  $r_j$ . Here,  $x_c$  and  $y_c$  are the corner coordinates in pixel. The minimization is performed by the Trust Region Reflective<sup>38</sup> algorithm, which is implemented in SciPy.<sup>39</sup> Bounds are used on the angular components of the pose to obtain meaningful results according to the MUAV specifications (see Table 2 in Sec. 6). The whole pose estimation algorithm is summarized as a pseudocode in Algorithm 2.

The resulting pose vector  $\phi_i$  of the current frame and the state vector  $X_{i-1}$  of the previous frame are used to estimate the current state vector  $X_i$  using the MUAV flight behavior model in Sec. 6 and the Kalman filter in Sec. 5. Orientation  $(\theta_x, \theta_y, \text{ and } \theta_z)$  and velocity  $\dot{x}$  are needed to estimate the current acceleration  $\ddot{x}_i$  from the flight model [Eq. (15)], taking into account the gravitation, thrust, and aerodynamic drag force. The determined acceleration vector and the  $\phi_i$  positions are then fed into the Kalman filter as measurement values ( $z_i$ ) to update the MUAV state vector  $X_i$ . Finally, the  $n$ -frame prediction is calculated as mentioned previously.

Figure 6 shows the results from the analysis of the same image sequence as used in Sec. 2. Owing to the missing reliable data for ground truth for the MUAV position, a reference for the 3-D flight path (—) is calculated from the state vectors ( $X_i$ ). In Fig. 6(a), this track is compared to the five-frame prediction (gray bullets). A good agreement between flight path and prediction can be observed in area of constant or



**Fig. 4** Link between the detected corners on the image and the drone skeleton model. (a) The detected corners using FAST, (b) the corners after the estimation of the essential matrix and the projected skeleton from the previous frame, and (c) skeleton points and corners used for the pose estimation of the drone.



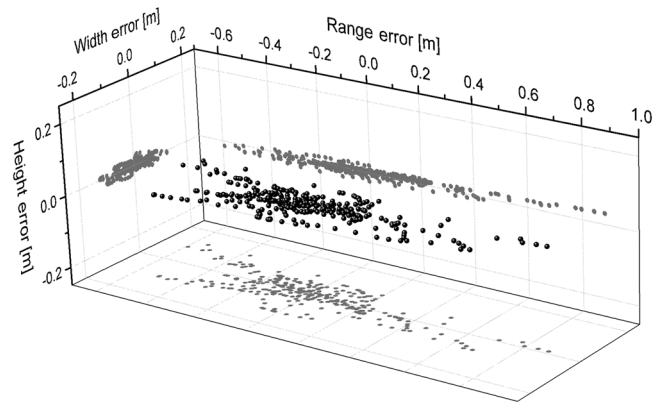
**Fig. 5** Principle for aligning the drone skeleton on the rays issued from the detected corners. Aligning the skeleton  $S$  starting at the pose  $\phi_{i-1}$  on rays issued from  $C_{i,opti}$  shown in Fig. 4.

linear movement (no rapid change of flight direction), while at turning points, where the flight direction changes rapidly, the algorithm still tends to overestimate the flight motion.

In Fig. 6(b), a detailed analysis is presented. In the upper diagram, the amplitudes of the motion components, i.e., absolute velocity and absolute acceleration, are depicted. These values are directly derived from the algorithm. The velocity and acceleration are measured in standard units (m/s and m/s<sup>2</sup>).

In the lower diagram, the prediction error, that is L2-norm distance between predicted and ground-truth positions, is plotted for each frame. In most frames, this prediction error is < 20 cm. Only in a few sections of the sequence does this error exceed a value of 50 cm. These are the parts of the track where rapid changes of the flight direction occur. Further, significant prediction errors occur in the same frames as in the 2-D case [compare Fig. 2(b)], only.

A more detailed discussion of the prediction error can be derived from Fig. 7. Here, the prediction error is plotted for each coordinate axis as the difference between the MUAV

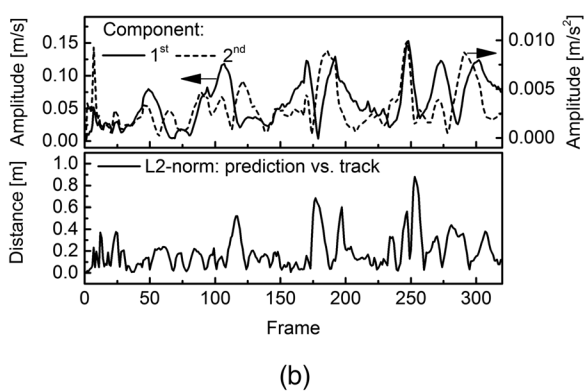
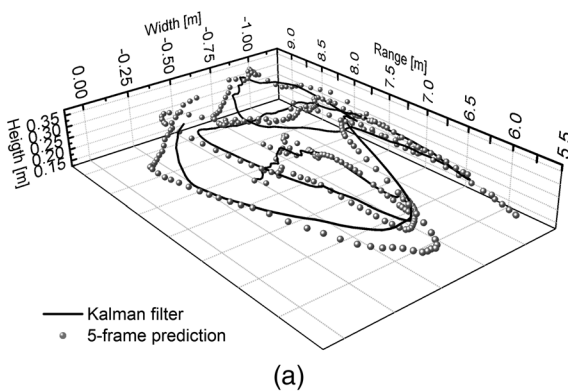


**Fig. 7** Illustration of the prediction error as a 3-D scatter plot. The difference between track and prediction is plotted for each coordinate axis: height, width, and range.

flight path (or track) and the predicted positions. Along the three coordinate axes, the prediction error is scattered around the zero position, giving evidence that no systematic error is present. In height ( $\Delta y$ ) and width ( $\Delta x$ ), the error is scattered with marginal standard deviation  $\sigma$  of  $\pm 1$  and  $\pm 3$  cm, respectively. Owing to a more dynamic flight behavior, the standard variation along the range axis ( $\Delta z$ ) is  $\sigma = 24.4$  cm. Here, we want to point out that the retrieval of 3-D information from a single view in a 2-D intensity image is a challenging task, especially for range estimation. The statistical analysis of the whole sequence of 319 frames is summarized in Table 1.

**Table 1** Summary of statistics on the prediction error in a sequence of 319 frames.

Data	Mean (m)	Std. deviation $\sigma$ (m)	Variance $\sigma^2$ (m <sup>2</sup> )	Min. (m)	Max. (m)
$\Delta x$	-0.002	0.03	0.001	-0.13	0.13
$\Delta y$	0.0002	0.0109	0.0001	-0.03	0.03
$\Delta z$	-0.002	0.244	0.06	-0.59	0.88



**Fig. 6** Results of the 3-D MUAV tracking and  $n$ -frame prediction algorithm applied to a video sequence recorded under laboratory conditions: (a) 3-D tracks from the Kalman filter (—) and the  $n$ -frame prediction (gray bullets); (b) amplitude of the components (first and second derivatives) and the prediction mismatch (L2-norm).

## 4 Conclusions

We want to point out again that the 3-D information was carried out from pose estimation in 2-D intensity images. Further, in the analyzed sequences, the MUAV was observed under a very gradual observation angle. Here, a more slanted observation angle could help to increase the accuracy of the spatial position estimation.

We successfully demonstrated the capability to extract 3-D information from 2-D intensity images by applying pose estimation with a simple skeleton model. From this pose information (i.e., position and orientation) and a simplified flight behavior model (with a floating body approximation), we could calculate the state vector that describes the current MUAV flight operation. Further, using a motion model that includes velocity and acceleration, we could predict the position of the MUAV in the near future.

The overall performance gives evidence that our approach is capable of resulting in a good first estimation of the future MUAV flight behavior. Unfortunately, in dynamic operation that includes rapid changes in velocity and direction, the underlying model tends to an overshooting effect. In future investigation, this error could be diminished by the use of a higher order of temporal derivative.

In our experiments, we used a single view from a single camera to retrieve a 3-D pose estimation by mainly analyzing the position of the four rotors. Further, the experiments were carried out at a very narrow viewing angle close to the MUAV principal plane. In these conditions, sometimes the line of sight on one or two rotors is blocked by the MUAV main body. Then, the pose estimation uses only the remaining visible rotors (three visible rotors are optimal). This situation is sufficient for a reliable pose estimation and has only a slight impact on the overall estimation of the 3-D position. In the later application in outdoor scenarios, we assume to have a steeper viewing angle, which should allow for an even better pose estimation.

To date, we have investigated our approach only in laboratory conditions as a proof-of-principle study. But, image data recorded in outdoor scenarios under operational conditions will contain a lot of additional disruptive factors such as signal-to-noise ratio, resolution, lighting conditions, and background. Further, MUAV will fly at larger ranges. Here, an active imaging device with a very narrow field of view could counter these challenges and could be used to gate out the background, increase resolution and signal-to-noise ratio, and be independent from lighting conditions. Our team has access to a shortwave-infrared laser gated-viewing system as presented by Christnacher et al.<sup>10</sup> In the near future, we will perform tests to investigate the pose estimation approach in operational conditions. Lighting conditions and observing MUAV flight behavior at a larger range can reduce the ranging accuracy of the proposed pose estimation approach due to an expected smaller scaling effect of the MUAV with the range at narrower observation angles. This effect could be compensated for by ranging capabilities of laser imaging devices. Notwithstanding this, the determination of the tilt angles ( $\theta_x$  and  $\theta_y$ ) and, thus, the application of the flight model should not be affected as long as the target is imaged with a high resolution.

Further, in the presented work, we have tested our approach on a single known MUAV, but we are convinced that our approach has a general nature and it could be easily

adapted to other multicopter MUAV. Finally, the tracking and prediction of the flight behavior of unknown MUAV is possible if a classification of the target is done previously. As long as we have a proper skeleton model, this approach should be able to track and predict flight behavior of any multicopter MUAV. For the prediction of other MUAV types, such as fixed-wing or vertical takeoff and landing, the flight model has to be adapted carefully and in detail.

## 5 Appendix A: Brief Description of the Kalman Filter

The Kalman filter is an iterative forward recursion process that consists of two steps: the “time update” and the “measurement update.”<sup>22</sup> Within the “time update,” for the  $i$ 'th measurement, the state vector  $X_i^-$  and the error covariance  $P_i^-$  are projected from previous values using the linear stochastic differential equation

$$X_i^- = AX_{i-1} + Bu_k, \quad (10)$$

$$P_i^- = AP_{i-1}A^T + Q. \quad (11)$$

Here, Eq. (10) reflects the process model as a system of motion equations with state-transition matrix  $A$ , the previous state vector  $X_{i-1}$ , and the control-input model  $Bu_k$  (in our case,  $Bu_k = 0$ ). Analogously, the projected error covariance matrix  $P_i^-$  is calculated in Eq. (11) using the error covariance  $P_i$  and the process noise covariance matrix  $Q$ .

Then in “measurement update,” the state vector  $X_i$  is updated by weighting the measurement  $z_i$  and the projected state vector  $X_i^-$  with the Kalman gain  $K_i$ . Finally, the error covariance  $P_i$  is updated

$$K_i = P_i^- H^T (HP_i^- H^T + R)^{-1}, \quad (12)$$

$$X_i = X_i^- + K_i(z_i - HX_i^-), \quad (13)$$

$$P_i = P_i^- - K_i H P_i^-. \quad (14)$$

Here, the matrices  $H$  and  $R$  describe the measurement (or observation) process and covariance.

## 6 Appendix B: The Microunmanned Aerial Vehicles Flight Behavior Model

The MUAV flight behavior can be described by the motion in Eq. (15). Here,  $m$  is the mass of the MUAV and  $m\ddot{x}$  is the resulting force that accelerates the MUAV in any direction. This force is the sum of the gravitational force  $G = [0 \ -mg \ 0]^T$ , the total thrust  $T$  of all propellers, and the drag force  $D$ . Owing to the lack of information about the single motor thrusts, rotational motion is neglected in this model

$$m\ddot{x} = G + T + D. \quad (15)$$

The overall thrust  $T$  is assumed to be perpendicular to the MUAV main plane, that is, the plane in which all motors lie. Here,  $T$  is oriented in the direction of the normal vector  $n = [n_x, n_y, n_z]^T$ . We define the thrust amplitude as  $|T| = \sqrt{T_x^2 + T_y^2 + T_z^2}$  and the thrust vector as  $T = |T|n = [T_x, T_y, T_z]^T$ . The thrust amplitude has positive values  $|T| \in [0, T_{\max}]$ . Further, in a first approximation, we

assume a floating body that means the vertical thrust ( $T_y$ ) compensates for the gravitational force

$$T_y + G_y = 0 \Rightarrow T_y = mg. \quad (16)$$

From this assumption we can determine the forces induced by the thrust in the other direction

$$T_x = \frac{n_x}{n_y} T_y = \frac{n_x}{n_y} mg, \quad (17)$$

$$T_z = \frac{n_z}{n_y} T_y = \frac{n_z}{n_y} mg. \quad (18)$$

Finally, we use a simplified model based on Stokes's law to determine the drag force  $D = -k_d \dot{x}$  with  $k_d$  as an aerodynamic constant. This force describes the aerodynamic drag on the MUAV that is proportional to the velocity. The aerodynamic constant  $k_d$  can be determined experimentally.<sup>40,41</sup> In this paper, we assume an isotropic aerodynamic coefficient of  $k_d = 0.55 \frac{\text{kg}}{\text{s}}$  and a maximum thrust of  $T_{\max} = 15.3 \frac{\text{kgm}}{\text{s}^2}$ , which can be calculated from Eq. (15) and the technical specification of the investigated MUAV.<sup>34</sup> Details of this calculation are summarized in Sec. 7.

From Eqs. (15)–(18), we can derive a calculus to determine the accelerations in the  $x$ -,  $y$ - and  $z$ -directions from the MUAV tilt angles ( $\theta_x$  and  $\theta_z$ ), which is analogous to the 3-D pose of the MUAV

$$\ddot{x} = \frac{n_x}{n_y} g - \frac{k_d}{m} \dot{x}, \quad \text{with } \frac{n_x}{n_y} = \tan \theta_x, \quad (19)$$

$$\ddot{y} = -\frac{k_d}{m} \dot{y}, \quad (20)$$

$$\ddot{z} = \frac{n_z}{n_y} g - \frac{k_d}{m} \dot{z}, \quad \text{with } \frac{n_z}{n_y} = \tan \theta_z. \quad (21)$$

Further, stable flight operation has to comply with the condition that the thrust amplitude  $|T|$  is limited by the maximum thrust  $T_{\max}$  by  $|T| < T_{\max}$ . This condition has a direct impact on the expected values of the normal vector [see Eq. (22)] and is analogous to the definition of a maximum tilt angle  $\theta_{\max}$

$$\sqrt{\frac{n_x^2 + n_y^2 + n_z^2}{n_y^2}} < \frac{T_{\max}}{mg}, \quad (22)$$

$$\sqrt{\frac{n_x^2 + n_z^2}{n_y^2}} < \sqrt{\left(\frac{T_{\max}}{mg}\right)^2 - 1} \equiv \tan \theta_{\max}. \quad (23)$$

## 7 Appendix C: Estimation of Aerodynamic Coefficient $k_d$ and Maximum Thrust $T_{\max}$

With the definitions of the gravitational force  $G$ , the thrust  $T$ , and the drag force  $D$  as

$$G = \begin{bmatrix} 0 \\ -mg \\ 0 \end{bmatrix}, \quad T = \begin{bmatrix} \tan \theta_x T_y \\ T_y \\ \tan \theta_z T_y \end{bmatrix} \quad \text{and} \quad D = \begin{bmatrix} -k\dot{x} \\ -k\dot{y} \\ -k\dot{z} \end{bmatrix}, \quad (24)$$

**Table 2** Physical flight parameter  $T_{\max}$  and  $k_d$  for the investigated MUAV calculated from the manufacturer specifications.<sup>34</sup>

Parameter	Symbol	Unit	DJI phantom 3
Total weight	$m$	Kg	1.28
Max ascent speed	$v_{v,\max}$	m/s	5
Max horizontal speed	$v_{h,\max}$	m/s	16
Max tilt angle	$\theta_{\max}$	deg	35
Max thrust	$T_{\max}$	$\frac{\text{kgm}}{\text{s}^2}$	15.3
Drag const.	$k_d$	$\frac{\text{kg}}{\text{s}}$	0.55

the MUAV motion can be described as motion in the  $x$ -,  $y$ - and  $z$ -directions by the equation system in Eqs. (25)–(27) in analogy to Eq. (15). Now,  $\theta_i$  is the inclination of the MUAV or tilt angle in the  $i$ -direction and  $\tan \theta_i$  is equal to  $\frac{n_i}{n_y}$

$$m\ddot{x} = \tan \theta_x T_y - k_d \dot{x}, \quad (25)$$

$$m\ddot{y} = -mg + T_y - k_d \dot{y}, \quad (26)$$

$$m\ddot{z} = \tan \theta_z T_y - k_d \dot{z}. \quad (27)$$

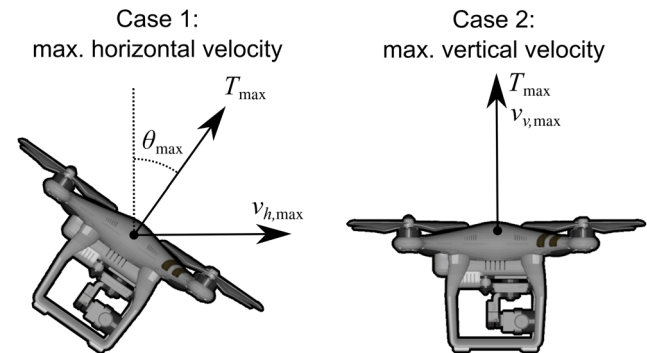
In the following, the aerodynamic parameters  $T_{\max}$  and  $k_d$  will be derived from this equation system using manufacturer specifications for weight, maximum horizontal and vertical velocity, and maximum tilt angle of the investigated MUAV.<sup>34</sup> The values are summarized in Table 2.

As illustrated in Fig. 8, we can distinguish two borderline flight operations: maximum horizontal velocity and maximum vertical velocity.

### 7.1 Maximum Horizontal Velocity

First, we assume a flight operation with maximal horizontal velocity in the  $x$ -direction. In this case, the MUAV will be tilted to maximum inclination toward the flight direction, which is given by the maximum tilt angle  $\theta = \theta_{\max}$ . Further, in the horizontal direction ( $x$ ), the velocity  $\dot{x}$  is equal to  $v_{h,\max}$ , the maximum horizontal velocity.

Further, due to borderline flight operation, the velocities in the other directions and all accelerations are zero,



**Fig. 8** The two different extreme flight operations, case 1: maximal horizontal velocity and case 2: maximal vertical velocity, are distinguished.



$\dot{y} = \dot{z} = 0$  and  $\ddot{x} = \ddot{y} = \ddot{z} = 0$ . The thrust is at maximum level,  $T = T_{\max}$ , but, due to the floating body assumption, its vertical component  $T_y$  has to compensate for the gravitational force, that is  $T_y = mg$ . Thus, the maximal thrust  $T_{\max}$  can be calculated as

$$T_{\max} = \sqrt{T_x^2 + T_y^2}, \quad (28)$$

$$= mg \sqrt{(\tan \theta_{\max})^2 + 1}, \quad (29)$$

$$T_{\max} = 15,3 \frac{\text{kg m}}{\text{s}^2}. \quad (30)$$

Then, the aerodynamic constant  $k_d$  can be calculated from Eq. (25) as

$$k_d = mg \frac{\tan \theta_{\max}}{v_{h,\max}} = 0,55 \frac{\text{kg}}{\text{s}}. \quad (31)$$

Although we have already determined the values for both  $T_{\max}$  and  $k_d$ , these results can be validated by the second case, that is, maximum vertical velocity.

## 7.2 Maximum Vertical Velocity

Here, we can assume a flight operation with maximum vertical velocity  $\dot{y} = v_{v,\max}$ . In this case, the inclination angle is zero,  $\theta = 0$  deg, and the maximum thrust pushes the MUAV upward in the  $y$ -direction,  $T_y = T_{\max}$ . The velocities in the horizontal directions are zero,  $\dot{x} = \dot{z} = 0$ , as well as any accelerations,  $\ddot{x} = \ddot{y} = \ddot{z} = 0$ . Then, from Eq. (26) and with the result for either  $k_d$  or  $T_{\max}$  of the previous case, we can again derive

$$k_d = \frac{T_{\max} - mg}{v_{v,\max}} = 0,55 \frac{\text{kg}}{\text{s}}, \quad (32)$$

$$T_{\max} = mg + k_d v_{v,\max} = 15,3 \frac{\text{kg m}}{\text{s}^2}. \quad (33)$$

Both results agree with the previously found values.

## Acknowledgments

The authors would like to thank Dr. Sebastien Hengy from ISL's acoustics group for carefully operating the MUAV during our indoor experiments.

## References

1. U.S. Department of Transportation, "Federal Aviation Administration—UAS sightings report," [https://www.faa.gov/uas/resources/uas\\_sightings\\_report/](https://www.faa.gov/uas/resources/uas_sightings_report/) (16 August 2018).
2. Transport Canada, "Statement by Minister of Transport about a drone incident with a passenger aircraft in Quebec City," 2017, [https://www.canada.ca/en/transport-canada/news/2017/10/statement\\_by\\_minister\\_of\\_transport\\_about\\_drone\\_incident\\_with\\_passenger.html](https://www.canada.ca/en/transport-canada/news/2017/10/statement_by_minister_of_transport_about_drone_incident_with_passenger.html) (15 January 2019).
3. L. To, A. Bati, and D. Hilliard, "Radar cross section measurements of small unmanned air vehicle systems in non-cooperative field environments," in *3rd Eur. Conf. Antennas and Propag.*, IEEE, pp. 3637–3641 (2009).
4. S. Harman, "Characteristics of the radar signature of multi-rotor UAVs," in *Eur. Radar Conf. (EuRAD)*, IEEE, pp. 93–96 (2016).
5. C.-C. Tsai, C.-T. Chiang, and W.-J. Liao, "Radar cross section measurement of unmanned aerial vehicles," in *IEEE Int. Workshop Electromagn.: Appl. and Stud. Innov. Competition (iWEM)*, IEEE, pp. 1–3 (2016).

6. M. Pieraccini, L. Miccinesi, and N. Rohhani, "RCS measurements and ISAR images of small UAVs," *IEEE Aerosp. Electron. Syst. Mag.* **32**, 28–32 (2017).
7. M. Laurenzis, E. Bacher, and F. Christnacher, "Experimental and rendering-based investigation of laser radar cross sections of small unmanned aerial vehicles," *Opt. Eng.* **56**(12), 124106 (2017).
8. D. Noetel et al., "Detection of MAVs (micro aerial vehicles) based on millimeter wave radar," *Proc. SPIE* **9993**, 999308 (2016).
9. T. Müller, "Robust drone detection for day/night counter-UAV with static VIS and SWIR cameras," *Proc. SPIE* **10190**, 1019018 (2017).
10. F. Christnacher et al., "Optical and acoustical UAV detection," *Proc. SPIE* **9988**, 99880B (2016).
11. A. Hommes et al., "Detection of acoustic, electro-optical and RADAR signatures of small unmanned aerial vehicles," *Proc. SPIE* **9997**, 999701 (2016).
12. M. Laurenzis et al., "Multi-sensor field trials for detection and tracking of multiple small unmanned aerial vehicles flying at low altitude," *Proc. SPIE* **10200**, 102001A (2017).
13. S. Hengy et al., "Multimodal UAV detection: study of various intrusion scenarios," *Proc. SPIE* **10434**, 104340P (2017).
14. A. Schumann et al., "Deep cross-domain flying object classification for robust UAV detection," in *14th IEEE Int. Conf. Adv. Video and Signal Based Surveill. (AVSS)*, IEEE, pp. 1–6 (2017).
15. A. Coluccia et al., "Drone-vs-bird detection challenge at IEEE AVSS2017," in *14th IEEE Int. Conf. Adv. Video and Signal Based Surveill. (AVSS)*, IEEE, pp. 1–6 (2017).
16. J. Sander et al., "High-level data fusion component for drone classification and decision support in counter UAV," *Proc. SPIE* **10651**, 106510F (2018).
17. F. Racek et al., "Tracking, aiming, and hitting the UAV with ordinary assault rifle," *Proc. SPIE* **10441**, 104410C (2017).
18. T. Kojima and T. Namerikawa, "Image-based position estimation of UAV using Kalman filter," in *IEEE Conf. Control Appl. (CCA)*, IEEE, pp. 406–411 (2015).
19. S. H. Tang et al., "Unscented Kalman filter for position estimation of UAV by using image information," in *54th Annu. Conf. Soc. Instrum. and Control Eng. Jpn. (SICE)*, IEEE, pp. 695–700 (2015).
20. R. E. Kálmán, "A new approach to linear filtering and prediction problems," *J. Basic Eng.* **82**(1), 35–45 (1960).
21. K. P. Murphy, "Kalman filter toolbox for Matlab," 1999, [https://www.cs.ubc.ca/murphyk/Software/Kalman/kalman\\_download.html](https://www.cs.ubc.ca/murphyk/Software/Kalman/kalman_download.html) (11 November 2018).
22. G. Bishop and G. Welch, "An introduction to the Kalman filter," in *Proc. SIGGRAPH*, Course, 8(27599-3175), p. 59 (2001).
23. G. Nebehay and R. Pflugfelder, "Consensus-based matching and tracking of keypoints for object tracking," in *IEEE Winter Conf. Appl. Comput. Vision (WACV)*, IEEE, pp. 862–869 (2014).
24. G. Nebehay and R. Pflugfelder, "Clustering of static-adaptive correspondences for deformable object tracking," in *IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 2784–2791 (2015).
25. E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," *Lect. Notes Comput. Sci.* **3951**, 430–443 (2006).
26. E. Rosten, R. Porter, and T. Drummond, "Faster and better: a machine learning approach to corner detection," *IEEE Trans. Pattern Anal. Mach. Intell.* **32**, 105–119 (2010).
27. S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: binary robust invariant scalable keypoints," in *IEEE Int. Conf. Comput. Vision*, IEEE, pp. 2548–2555 (2011).
28. K. D. Sebesta and N. Boizot, "A real-time adaptive high-gain EKF, applied to a quadcopter inertial navigation system," *IEEE Ind. Electron. Mag.* **6**(1), 495–503 (2014).
29. P. Castillo, R. Lozano, and A. Dzul, "Stabilization of a mini rotorcraft with four rotors," *IEEE Control Syst. Mag.* **25**(6), 45–55 (2005).
30. S. Bouabdallah and R. Y. Siegwart, "Full control of a quadrotor," in *IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, IEEE, San Diego, California, pp. 153–158 (2007).
31. R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: modeling, estimation, and control of quadrotor," *IEEE Rob. Autom. Mag.* **19**, 20–32 (2012).
32. C. G. Mayhew, R. G. Sanfelice, and A. R. Teel, "On path-lifting mechanisms and unwinding in quaternion-based attitude control," *IEEE Trans. Autom. Control* **58**(5), 1179–1191 (2013).
33. G. Hoffmann et al., "Quadrotor helicopter flight dynamics and control: theory and experiment," in *AIAA Guid., Navig. and Control Conf. and Exhibit*, p. 6461 (2007).
34. DJI, "Phantom 3 professional specs," <https://www.dji.com/> (19 September 2018).
35. M. Calonder et al., "BRIEF: binary robust independent elementary features," *Lect. Notes Comput. Sci.* **6314**, 778–792 (2010).
36. D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Trans. Pattern Anal. Mach. Intell.* **26**, 756–770 (2004).
37. R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd edn., Cambridge University Press, New York (2004).
38. D. C. Sorensen, "Newton's method with a model trust region modification," *SIAM J. Numer. Anal.* **19**(2), 409–426 (1982).

39. SciPy Community, "SciPy v0.18.1, optimization and root finding," <https://www.scipy.org/> (15 January 2019).
40. C. R. Russell et al., "Wind tunnel and hover performance test results for multicopter UAS vehicles," in *72nd Am. Helicopter Soc. (AHS) Int. Annu. Forum* (2016).
41. C. R. Russell, C. R. Theodore, and M. K. Sekula, "Incorporating test data for small UAS at the conceptual design level," in *AHS Int. Tech. Meeting Aeromech. Des. Transformative Vert. Flight* (2018).

**Martin Rebert** received his master's degree in engineering from Télécom Bretagne (France) in 2015. He is now preparing for his PhD with the collaboration of the IRIMAS laboratory of the University of Haute Alsace in Mulhouse (France) and the French-German Research Institute of Saint-Louis in Saint-Louis (France). His work is focused on computer vision and autonomous navigation.

**Stéphane Schertzer** is a research engineer at the French-German Research Institute of Saint-Louis (ISL) in Saint-Louis (France).

He received his MSc degree in electrical engineering and computer sciences from the University of Haute-Alsace in Mulhouse (France) in 2009. He specializes in computer vision, active imaging and software development (C++, CUDA etc.) of algorithms and man-machine interfaces. He is currently developing new algorithms for active imaging systems.

**Martin Laurenzis** received his MS degree in physics from the University of Dortmund (Germany) in 1999 and his PhD in electrical engineering and information technology from the University of Aachen (Germany) in 2005. He has been with the French-German Research Institute of Saint-Louis (France) since 2004. His research interests are focused on microunmanned aerial vehicle detection using heterogeneous sensor networks, applied computer vision, and computational imaging.