2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Madrid, Spain, October 1-5, 2018

Robust Visual-Inertial State Estimation with Multiple Odometries and Efficient Mapping on an MAV with Ultra-Wide FOV Stereo Vision

M. G. Müller[1*], F. Steidle[1], M. J. Schuster[1], P. Lutz[1], M. Maier[1], S. Stoneman[1], T. Tomic[1,2], W. Stürzl[1]

***Abstract*— The here presented flying system uses two pairs of wide-angle stereo cameras and maps a large area of interest in a short amount of time. We present a multicopter system equipped with two pairs of wide-angle stereo cameras and an inertial measurement unit (IMU) for robust visual-inertial navigation and time-efficient omni-directional 3D mapping. The four cameras cover a 240 degree stereo field of view (FOV) vertically, which makes the system also suitable for cramped and confined environments like caves. In our approach, we synthesize eight virtual pinhole cameras from four wide-angle cameras. Each of the resulting four synthesized pinhole stereo systems provides input to an independent visual odometry (VO). Subsequently, the four individual motion estimates are fused with data from an IMU, based on their consistency with the state estimation. We describe the configuration and image processing of the vision system as well as the sensor fusion and mapping pipeline on board the MAV. We demonstrate the robustness of our multi-VO approach for visual-inertial navigation and present results of a 3D-mapping experiment.**

I. INTRODUCTION

Micro Aerial Vehicles (MAVs) have been used in a vast variety of applications in recent years. Their ability to quickly reach points of interest or to obtain perspectives which were previously difficult or impossible to reach not only makes them interesting for tasks like exploration, inspection, and search and rescue, but also for general consumer applications, e.g. generating virtual reality content by mapping environments. Since most MAVs do not have a very long flight time due to battery limitations, mapping large scenes has to be done in a very efficient manner. Therefore, it is helpful if the MAV's mapping sensors have a wide field of view (FOV). A wide FOV is also beneficial for obstacle detection and path planning. Furthermore, in semi-dynamic environments or situations where parts of the scene are texture-less, it is more likely to find enough reliable features for ego-motion estimation with a wide FOV than with a narrow one. As cameras are light-weight, and provide a huge amount of information, they are ideally suited for mobile robots that have limited payload. Stereo cameras have been employed successfully on MAVs to obtain images as well as depth information in both indoor and outdoor environments [1], [2], [3], [4].

These facts motivated us to build an MAV, Ardea, that supports wide-angle stereo vision, shown in Fig. 1. Ardea is equipped with four wide-angle cameras arranged in two stereo configurations. The total vertical FOV of 240° enables Ardea to efficiently and effectively map the environment. A fully spherical view of a scene including depth information can easily be obtained by rotation about the vertical axis (yaw-rotation). Each wide-angle camera image is remapped to two images with pinhole projection, resulting in four synthesized pinhole stereo systems that provide input to independent visual odometries (VOs). In [5], the visual features from multiple cameras are tracked in a joint optimization, leading to a tight coupling. In contrast, we process our stereo pairs decoupled form each other (similar to [6]) and fuse their visual odometry results in a real-time capable filter for local state estimation. Thereby, the complexity of our global graph-based estimation is not affected by the number of high-frequency sensors, allowing for fast online optimization steps [7]. The approach described in [8] uses two visual odometries where only one is selected to be fused with an inertial measurement unit (IMU). In our work, all four VO pose estimates are fused with the data of an IMU with a single filter. Running four independent visual pose estimations provides additional redundancy to the system, which can be critical in the case of complex scenes, where it is likely that one of the VOs will return a poor result or even entirely fail. Our main contributions are therefore:

Fig. 1: In-house built Y6-hexacopter Ardea on Mt. Etna. The field of view is illustrated by the blue overlay. (size: 68 cm × 68 cm × 30 cm, weight: 2.65 kg including battery).

- sensor fusion of multiple VOs with independent keyframe selection for improved navigation with respect to accuracy, robustness and redundancy
- description of a wide-angle multi-camera setup for efficient environment mapping on an MAV
- detailed description of a multi-fisheye camera calibration and pinhole remapping for computational efficiency

[1]Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Germany
[*]Corresponding author: marcus.mueller@dlr.de
[2]Current affiliation: Skydio Inc., United States

The paper is organized as follows: We begin by briefly introducing our MAV and its hardware in the next section. Then, in section III, we present the camera setup and image processing of the MAV vision system in detail. In section IV, we describe our multi-VO approach and the sensor fusion with the IMU. After summarizing our mapping framework in section V, we present results demonstrating the effectiveness of our approach (section VI).

II. General Hardware Setup

We chose a trigonal Y6 frame construction for our MAV to be able to place the cameras in a way that they have a high coverage without having any parts from the MAV in their FOV. This setup also enables the MAV to fly in more confined areas such as indoors or in caves. The MAV is propelled by three 10" coaxial rotor pairs providing a maximum thrust of 3.6 kg. It consists of two main, separable parts:

- Frame: includes all engines, speed-controllers and the outer carbon frame tubes.
- Stack: consists of all navigation sensors and on-board computers.

The stack can also be operated by itself, which makes it easier to develop and test new hardware and software before incorporating it with the frame. The on-board computing hardware consists of an Intel NUC with a dual core i7 (3 GHz), an FPGA (Xilinx Spartan6) for SGM-based stereo processing [9], [10] and a BeagleBoneBlack embedded computer which runs an attitude and position controller at 500 Hz. A wide-angle multi-camera system, described in detail in section III, and an IMU (Analog Devices ADIS16407) are used as on-board sensors.

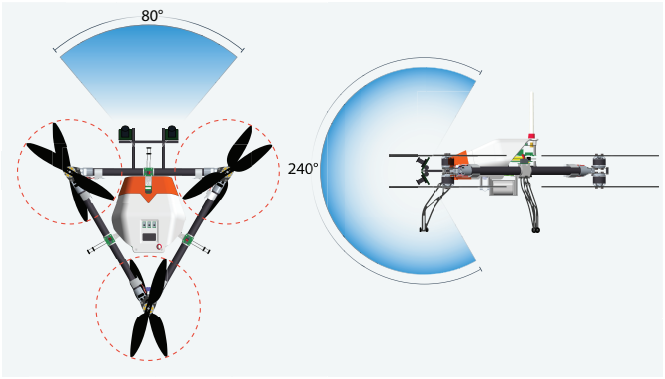

Fig. 2: Top and side view illustration of the Ardea multicopter. Bluish areas indicate the horizontal and vertical FOV of the multi-camera system.

III. Multi-Camera Setup

In the following we describe the wide-angle multi-stereo camera system of our multicopter Ardea that provides approx. 240° vertical field of view as illustrated in the right side of Fig. 2. In addition to the large FOV, the arrangement of cameras is well suited for the high dynamic range situation in outdoor scenes with often much higher brightness above the horizon than below. As separate cameras cover the FOV below and above the horizon, longer exposure and higher gains can be used for the lower FOV to cope with the different intensities. The camera system consists of four synchronized cameras, each providing 1280×860 px. For achieving a large FOV and reasonable resolution in the vertical axis, the smaller image dimension (860 px) of the camera sensors are horizontally aligned. The cameras are arranged in two stereo systems with the optical axes of the lower cameras at a $-60°$ angle with respect to the horizon, and those of the upper cameras at $+60°$ (see Fig. 3). The FOV of each camera is about $80°$ horizontally and about $125°$ vertically.

A. Calibration of Wide-Angle Cameras with Modified Kannala-Brandt Model

For cameras with very large FOVs, the standard pinhole camera model is not suitable, since, for instance, the z coordinate of an object point can approach zero or can even become negative for FOV $> 180°$. Therefore, we used a modified version of the Kannala-Brandt model [11] with 10 parameters $(k_1, k_2, k_3, k_4, p_1, p_2, m_u, u_c, m_v, v_c)$. The radial projection of the lens is modeled by a polynomial

$$\tilde{\theta} \;=\; f(\theta) \;=\; \theta + k_1\theta^3 + k_2\theta^5 + k_3\theta^7 + k_4\theta^9 \;, \quad (1)$$

where $\theta = \arccos(\frac{z}{\sqrt{x^2+y^2+z^2}})$ is the angle between the optical axis and a 3D point with coordinates (x, y, z) in the camera frame. The mapping onto the camera plane $(u(x, y, z), v(x, y, z))$ is given by

$$\tilde{u} \;=\; \tilde{\theta}\cos\phi \;=\; \tilde{\theta}\frac{x}{\sqrt{x^2+y^2}} \;, \quad (2)$$

$$\tilde{v} \;=\; \tilde{\theta}\sin\phi \;=\; \tilde{\theta}\frac{y}{\sqrt{x^2+y^2}} \;, \quad (3)$$

$$\hat{u} \;=\; \tilde{u} \;+\; 2p_1\tilde{u}\tilde{v} + p_2(3\tilde{u}^2+\tilde{v}^2) \;, \quad (4)$$

$$\hat{v} \;=\; \tilde{v} \;+\; 2p_2\tilde{v}\tilde{u} + p_1(3\tilde{v}^2+\tilde{u}^2) \;, \quad (5)$$

$$u \;=\; m_u\hat{u} + u_c \;, \quad v \;=\; m_v\hat{v} + v_c \;. \quad (6)$$

In contrast to the pinhole model, the Kannala-Brandt model does not discriminate between radial projection and (symmetric) radial distortion. Eqs. (4) and (5) describe the tangential distortion model due to lens misalignment (decentering distortion) motivated by the tangential distortion model of pinhole lenses. As in the pinhole projection model, the tangential distortion is a function of the undistorted projection coordinates $\tilde{u}$, $\tilde{v}$ and depends on just 2 additional parameters, p_1 and p_2. In contrast, the generic distortion model proposed in [11] introduces 14 additional parameters.

B. Epipolar Geometry of the Multi-Camera System

As the four wide-angle cameras are arranged in two stereo configurations, the epipolar geometry of each stereo pair has to be computed [12]. From extrinsic camera calibration we obtain the transformation ${}^i_1\hat{\mathcal{T}}$ from the reference camera 1 to all other camera frames $i = 2, 3, 4$, i.e.

$${}^i\mathbf{x} \;=\; {}^i_1\hat{\mathcal{T}}\,{}^1\mathbf{x} \;=\; {}^i_1\hat{\mathbf{R}}\,{}^1\mathbf{x} + {}^i\mathbf{t}_1 \;. \quad (7)$$

${}^i_1\hat{\mathbf{R}}$ is the rotation of the reference frame 1 with respect to camera frame i and ${}^i\mathbf{t}_1$ is the vector to the origin of 1 in

coordinates of frame i. All epipolar planes intersect the line connecting the nodal points of both cameras. If, as it is the case for the lower stereo system on our multicopter, reference camera 1 is one of the cameras of the stereo systems, then the vector pointing from the reference camera to the other camera ($i = 2$ on our multicopter) of the stereo pair is simply

$$ {}^1\mathbf{t}_2 \quad = \quad -{}^2_1\hat{\mathbf{R}}^\top {}^2\mathbf{t}_1 \ . \tag{8} $$

${}^1\mathbf{t}_2$ defines the first axis (the x-axis) of the stereo coordinate system in the coordinate system of reference camera 1,

$$ {}^1\mathbf{e}_x \quad = \quad \frac{{}^1\mathbf{t}_2}{\|{}^1\mathbf{t}_2\|} \quad = \quad -{}^2_1\hat{\mathbf{R}}^\top \frac{{}^2\mathbf{t}_1}{\|{}^2\mathbf{t}_1\|} \ . \tag{9} $$

The length of this vector is the stereo baseline, $b_{\text{st1}} = \|{}^1\mathbf{t}_2\|$. As stereo z-axis, i.e. as direction of the optic axes, we use the vector

$$ {}^1\mathbf{e}_z \quad = \quad \frac{{}^1\mathbf{m}_z - ({}^1\mathbf{e}_x^\top {}^1\mathbf{m}_z){}^1\mathbf{e}_x}{\|{}^1\mathbf{m}_z - ({}^1\mathbf{e}_x^\top {}^1\mathbf{m}_z){}^1\mathbf{e}_x\|} \ , \tag{10} $$

where ${}^1\mathbf{m}_z = {}^1\mathbf{e}_{z,1} + {}^1\mathbf{e}_{z,2} = {}^1\mathbf{e}_{z,1} + {}^2_1\hat{\mathbf{R}}^\top {}^2\mathbf{e}_{z,2}$ is the sum of the z-axis vectors of both camera frames in reference frame 1, i.e. ${}^1\mathbf{e}_{z,1} = {}^2\mathbf{e}_{z,2} = (0,0,1)^\top$. The stereo y-axis can be calculated using the cross product of ${}^1\mathbf{e}_x$ and ${}^2\mathbf{e}_z$,

$$ {}^1\mathbf{e}_y \quad = \quad {}^1\mathbf{e}_z \times {}^1\mathbf{e}_x \ . \tag{11} $$

The rotation matrix ${}^1_{\text{st1}}\hat{\mathbf{R}} = ({}^1\mathbf{e}_x, {}^1\mathbf{e}_y, {}^1\mathbf{e}_z)$ describes the rotation of the stereo system with respect to the reference camera frame 1. For other stereo systems (that do not contain the reference camera 1), we first have to compute the transformation between cameras. In our setup, the second stereo system consists of camera 3 and 4. The transformation from 4 to camera 3, the reference camera of this stereo system, is the transformation from 4 to 1 followed by transformation from 1 to 3,

$$ {}^3\mathbf{x} \quad = \quad {}^3_4\hat{\mathcal{T}}\, {}^4\mathbf{x} \quad = \quad {}^3_1\hat{\mathcal{T}}\, ({}^4_1\hat{\mathcal{T}})^{-1}\, {}^4\mathbf{x} \ . \tag{12} $$

As for the first stereo system (with camera 1 and 2), the direction of vector ${}^3\mathbf{t}_4$ defines the x-axis of the second stereo system (with camera 3 and 4) in coordinates of camera 3,

$$ {}^3\mathbf{e}_x \quad = \quad \frac{{}^3\mathbf{t}_4}{\|{}^3\mathbf{t}_4\|} \ , \tag{13} $$

and the length of vector ${}^3\mathbf{t}_4$ gives the stereo baseline. Similarily, using

$$ {}^3\mathbf{e}_z \quad = \quad \frac{{}^3\mathbf{m}_z - ({}^3\mathbf{e}_x^\top {}^3\mathbf{m}_z){}^3\mathbf{e}_x}{\|{}^3\mathbf{m}_z - ({}^3\mathbf{e}_x^\top {}^1\mathbf{m}_z){}^3\mathbf{e}_x\|} \ , \tag{14} $$

where ${}^3\mathbf{m}_z = {}^3\mathbf{e}_{z,3} + {}^3\mathbf{e}_{z,4} = {}^3\mathbf{e}_{z,1} + {}^3_4\hat{\mathbf{R}}\, {}^4\mathbf{e}_{z,4}$ is the sum of the z-axis vectors in both camera frames, i.e. ${}^3\mathbf{e}_{z,3} = {}^4\mathbf{e}_{z,4} = (0,0,1)^\top$, and

$$ {}^3\mathbf{e}_y \quad = \quad {}^3\mathbf{e}_z \times {}^3\mathbf{e}_x \tag{15} $$

we can define the rotation of the second stereo system with respect to camera 3, ${}^3_{\text{st2}}\hat{\mathbf{R}} = ({}^3\mathbf{e}_x, {}^3\mathbf{e}_y, {}^3\mathbf{e}_z)$.

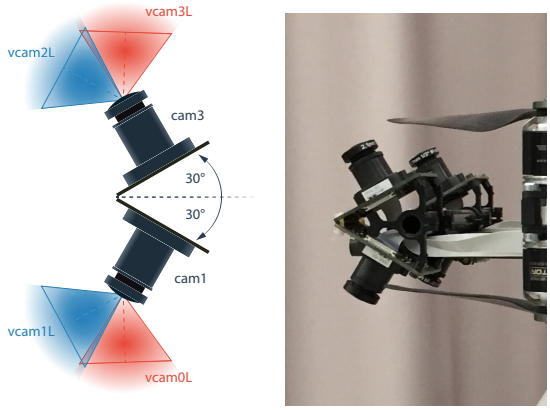

Fig. 3: Left: Lateral view of the left side of the multi-camera system. From each physical wide-angle camera ('cam1', 'cam3'), two "virtual" pinhole cameras are synthesized that are rotated by $\pm 30°$ around the stereo axes ('vcam0L' and 'vcam1L belong to 'cam1', 'vcam2L' and 'vcam3L belong to 'cam3'). On the mirror symmetric right side, the physical cameras 'cam2' and 'cam4' are located and remapped to 'vcam0R'/'vcam1R' and 'vcam2R'/'vcam3R', respectively. Each virtual pinhole camera has a FOV of $80° \times 65°$; the total vertical FOV of the stereo camera system is approx. $240°$. Right: Photo of the cameras.

C. Remapping to Pinhole Cameras

Although alternative projection models, like the Kannala-Brandt model presented in the previous section, are well established, many computer vision algorithms still require images projected according to the pinhole camera model. Depending on the application, the pinhole projection model can also simplify image processing. For instance, planar patches viewed frontally do not change their apparent size for translations perpedicular to the camera axis. Also, matching in motion stereo is easier as corresponding points lie on lines and not on more complex curves [13].

Although it is possible in principle to remap any camera image with FOV $< 180°$ to a single pinhole image, it is not recommended. Since the pinhole projection is usually a poor description of the actual mapping of wide-angle cameras[1], it would either artificially magnify the angular resolution in the outer part of the image and lead to image blur, or, if matched to the resolution in the outer part of the camera image, would reduce resolution in the center significantly. A reasonable workaround is the remapping to several pinhole images, which allows closer approximation of the original image resolution.In the following, we describe the remapping of wide-angle images to multiple pinhole images, which allowed us to run several instances of our efficient stereo odometry and integrate their estimations into our filter framework, as described in section IV. As depicted in Fig. 3, each wide-angle camera is split into two "virtual" pinhole cameras. The virtual cameras share the same viewpoint (the nodal point /center of projection of the wide-angle lens) but are rotated by $\pm 30°$ around the stereo axis to which their horizontal axes (the "u-axes") are aligned. For

[1]Wide-angle or fisheye cameras are often close to having a constant angular resolution radially, i.e. camera angle $\theta \propto \rho$, where $\rho = \sqrt{(u-u_c)^2 + (v-v_c)^2}$ is the distance from projection center. In the pinhole projection with $\rho \propto \tan\theta$, however, the resolution increases with camera angle according to $\frac{\partial\rho}{\partial\theta} \propto \frac{1}{\cos^2\theta}$ with $\lim_{\theta\to 90°} \frac{\partial\rho}{\partial\theta} = \infty$.

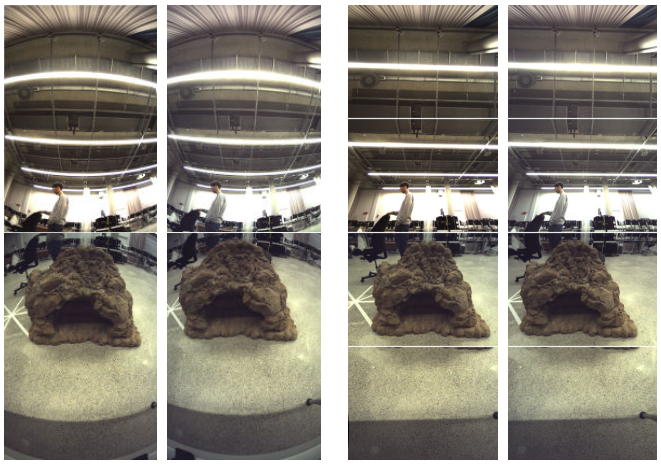

Fig. 4: Raw and pinhole camera images. From left to right: raw images of camera 1 (bottom) and camera 3 (top); raw images of camera 2 (bottom) and camera 4 (top); remapped pinhole images of virtual cameras 0L,1L,2L,3L (from bottom to top); remapped pinhole images of virtual cameras 0R,1R,2R,3R. Note that "straight lines are straight" after remapping to pinhole images, but appear bent in the original camera images.

each of the 4 virtual cameras ($i = 0, 1,\ D = L, R$) of the first fisheye stereo system with pixel coordinates $({}^{iD}u, {}^{iD}v)$ the non-normalized direction vector (in coordinates of stereo system 1) is

$$\begin{aligned} {}^{\mathrm{st1}}\mathbf{v}({}^{iD}u, {}^{iD}v) &= {}^{\mathrm{st1}}_{iD}\hat{\mathbf{R}}\ ({}^{iD}u - {}^{iD}u_c, {}^{iD}v - {}^{iD}v_c, f_i)^\top \\ &= {}^{\mathrm{st1}}_{iL}\hat{\mathbf{R}}\ ({}^{iD}u - {}^{iD}u_c, {}^{iD}v - {}^{iD}v_c, f_i)^\top \ , \end{aligned} \tag{16}$$

where we used the fact that ${}^{\mathrm{st1}}_{iR}\hat{\mathbf{R}} = {}^{\mathrm{st1}}_{iL}\hat{\mathbf{R}}$, i.e the $30°$ rotations for the left and right virtual stereo camera are the same since their coordinate systems have the same orientation with respect to stereo frame "st1". Similarly, for the 4 virtual cameras ($i = 2, 3,\ D = R, L$) of the second stereo system

$${}^{\mathrm{st2}}\mathbf{v}({}^{iD}u, {}^{iD}v) = {}^{\mathrm{st2}}_{iL}\hat{\mathbf{R}}\ ({}^{iD}u - {}^{iD}u_c, {}^{iD}v - {}^{iD}v_c, f_i)^\top \ . \tag{17}$$

Using Eqs. (16) and (17), the rotation matrices ${}^{1}_{\mathrm{st1}}\hat{\mathbf{R}}$, ${}^{3}_{\mathrm{st2}}\hat{\mathbf{R}}$ derived in the previous sub-section, the transformation between camera frames estimated by extrinsic calibration, as well as the instrinsic camera calibration parameter, the remapping tables for each virtual camera can be calculated. Note that x-axes of stereo systems are defined by lines connecting the nodal points of the cameras. Therefore, while the x-axes of the virtual stereo cameras that belong to the same physical cameras are perfectly aligned, this is not true for physically different stereo cameras systems. For our multi-camera system, this means that the rotation of virtual stereo frames $0L$ with respect to $1L$, and of $2L$ with respect to $3L$, is a rotation of $-60°$ around the x-axis. However, for instance, the rotation of $2L$ with respect to $1L$ is only approximately described by a rotation of $60°$ around the x-axis of $1L$. As shown in Fig. 4, each of the four wide-angle images of size 860×1280 px are debayered and then remapped to two RGB pinhole images of size 666×506 px using bilinear interpolation. For FPGA stereo processing the left and right pinhole image are scaled by factor 0.5 and

Fig. 5: FPGA Stereo processing: left input image consisting of the pinhole images of all left virtual cameras, i.e. 0L,1L,2L,3L, arranged in a 2×2 grid. The resulting depth map is shown on the right.

arranged in a 2×2 layout, see Fig. 5. Currently, the depth resolution is limited by the maximum image height of 508 px in the FPGA based stereo implementation. An overview of the image processing pipeline is shown in Fig. 6.

IV. Visual Odometry and Fusion

The following two paragraphs describe the visual odometry and the state estimation used on the MAV.

A. Visual Odometry

The task of a visual odometry is to give an estimate of the camera motion based on the perceived images. Our visual odometry estimates the relative transformation from one camera frame to another taken at different timestamps. The algorithm is based on [14], [15], where the reader is referred to for more details. We assume that the scene is mainly static and that the camera motion can be arbitrary. AGAST features [16] are detected in each of the left remapped virtual camera images. For each corner feature, the 3D information is provided by the resulting depth maps from dense stereo matching. Therefore, we obtain three-dimensional features, which can be used for motion estimation. Features which have no valid depth value, because of occlusions or other reasons, are discarded. Also, features that have too large depth values are ignored, as uncertainty increases quadratically with distance in stereo vision. In theory, just three non-colinear 3D feature points are sufficient to calculate

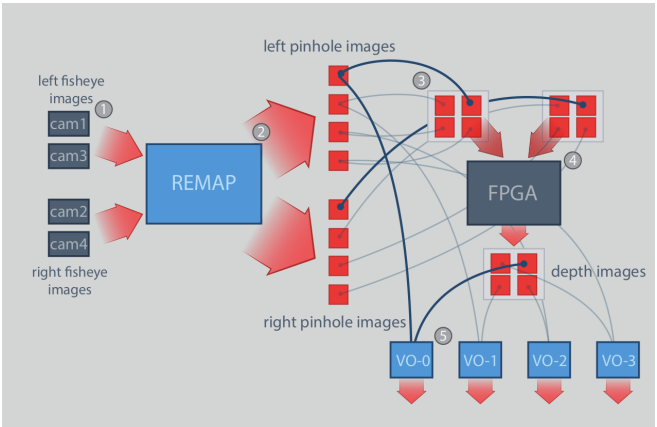

Fig. 6: Basic camera and visual odometry setup. ① Fisheye images are captured. ② The fisheye images are remapped into eight pinhole images. ③ All left and right images are grouped into one combined left and one combined right image. ④ Left and right images are sent to FPGA for stereo processing, resulting in a depth map. ⑤ Each VO instance receives a pinhole image and the corresponding depth map.

translation and rotation of the relative movement. Nevertheless, it is advantageous to have more feature points to reduce the effect of noise which increases accuracy and to improve rejection of outliers. After feature extraction, we search for correspondences from the previous image and the current image. Before a feature is used for motion estimation, it has to pass two additional outlier rejection steps. Since we assume a mainly static scene, we can expect that the relative distance d_{PQ} of two points P and Q does not change from frame $i-1$ to current frame i. Therefore, the distance d_{PQ}^{i-1} and distance d_{PQ}^{i} in Eq. (18) should equal up to a measurement error.

$$\begin{aligned} d_{PQ}^{i-1} &= \|\mathbf{P}^{i-1} - \mathbf{Q}^{i-1}\| \\ d_{PQ}^{i} &= \|\mathbf{P}^{i} - \mathbf{Q}^{i}\| \end{aligned} \tag{18}$$

The result of this outlier rejection step is a set of consistent correspondences, which maintain a relative distance between each other. The next outlier rejection step is based on an upper limit for the rotation angle of the camera. After the outlier rejection, the remaining corresponding features $\mathbf{P}_k^i$ and $\mathbf{P}_k^{i-1}$, $k = 1, 2, ..., n$, can be used to estimate the camera motion. In principle, it is done by minimizing

$$E(\hat{\mathbf{R}}, \mathbf{t}) = \sum_k \frac{1}{\sigma_k^2} (\mathbf{P}_k^{i-1} - (\hat{\mathbf{R}}\,\mathbf{P}_k^i + \mathbf{t}))^2 \ . \tag{19}$$

In Eq. (19), a spherical error model is used as a rough approximation of an image-based error model. The advantage of this error model is that the transformation can be calculated in a closed form solution. After $\hat{\mathbf{R}}$ and $\mathbf{t}$ are estimated, Chauvenet's criterion [17] is applied to remove the likely false correspondences. Finally, the values for the translation and rotation with the initial guess from the previous spherical error model, and the reduced set of consistent correspondences are optimized using an ellipsoid error model as described by Matthies and Shafer [2].

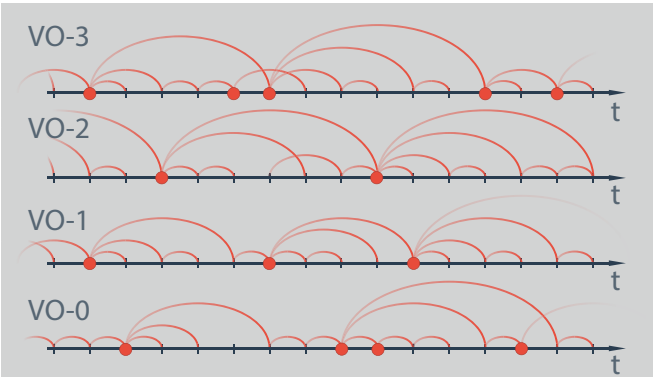

Fig. 7: Keyframe handling of multiple visual odometries. Red dots illustrate keyframes, arcs indicate relative to which reference frame the camera poses are estimated. Samples without arc indicate frames for which a pose estimation was not possible. The four VOs select different keyframes.

Similar to [18], [19], [20], [21] we are using keyframes for estimating not only the pose of the current camera frame relative to the previous one, but also relative to a set of selected camera frames in the past to increase the accuracy of the overall pose estimation. Every time an image is taken, its relative pose to all previous keyframes is calculated. The keyframe with the highest residual error is replaced by the new image. In our current setup, we use 5 keyframes. A key feature of running independent visual odometries with different FOV is the ability to select different keyframes for each VO, as shown in Fig. 7. Each displayed point illustrates one keyframe and each arc from which to which frame the motion is estimated. The four VOs use different keyframes. For instance, while flying close to ground, features in the lower field of view used by VO-0 are usually visible for much shorter intervals than features closer to the horizon, therefore VO-0 and VO-1 will select different keyframes, which will be demonstrated in section VI.

B. State Estimation with Delayed Multi-VO Measurements

We obtain robust and accurate state estimates by combining inertial measuerments from the IMU and the output of four independent visual odometries. The acceleration and angular rate readings from the inertial measurement unit are used to update the system state at high frequency. They are fused with lower frequent measurements from multiple visual odometries in an error state space Kalman filter as described in [22] and [23]. The direct form of the estimated state, also called main state, is defined by

$$\mathbf{x} = ({}^n_b\mathbf{p}, {}^n_b\mathbf{v}, {}^n_b\mathbf{q}, {}^b\mathbf{b}_\mathbf{a}, {}^b\mathbf{b}_\omega)^\top \ , \tag{20}$$

where ${}^n_b\mathbf{p} \in \mathbb{R}^3$ is the position of the body frame (b-frame) relative to an earth-fixed, inertial frame (n-frame), ${}^n_b\mathbf{v} \in \mathbb{R}^3$ is the velocity, ${}^n_b\mathbf{q}$ the orientation represented as a quaternion and ${}^b\mathbf{b}_\mathbf{a}$ and ${}^b\mathbf{b}_\omega$ are the acceleration and angular rate biases of the IMU. To fuse the VO measurements, it is important to take time delays due to sensor processing into account. Therefore hardware triggers are used to define the time stamp of an image with high accuracy. Each time an image is triggered a sub-state is added to the main state. The components of the main state, which have to be augmented are defined by the measurement equation. In case of VO measurements, which provide estimates of pose differences and their covariances, the equation is in the form of

$$\mathbf{h}_{t_1,t_2} = \mathbf{h}({}^n_b\mathbf{p}_{t_1}, {}^n_b\mathbf{q}_{t_1}, {}^n_b\mathbf{p}_{t_2}, {}^n_b\mathbf{q}_{t_2}) \ . \tag{21}$$

The times t_1 and t_2 refer to the start and end time of the visual odometry measurement, i.e. t_1 is the time stamp of the keyframe, relative to which the motion at time t_2 has been estimated. Measurements from different VOs with identical end times t_2 can have different start times t_1. Therefore, each time a hardware trigger arrives, the main state has to be augmented by the current pose ${}^n_b\mathbf{p}_t, {}^n_b\mathbf{q}_t$, and the covariance matrix by the sub-matrix representing the uncertainty of the current pose. In addition, the equation for system propagation has to be adapted. The result from a visual odometry is available to the filter with some delay. However due to the augmentation of the main state at the time of image capture, the measurement can refer to the relevant system sub-state when the measurement finally arrives and correct the current state including the augmentations. To reject outliers in the

measurements from a visual odometry, the Mahalanobis distance [24] is calculated. It compares the actual measurement from the visual odometry and the predicted measurement based on the filter estimate, and rejects measurements above a threshold depending on the fusion estimation uncertainty.

V. Mapping

In this section, we give a quick overview of our global localization and mapping framework, which is based on the architecture presented in [25], [7]. We perform an online global 3D mapping of the environment based on our filter estimates and dense depth data from our fisheye camera system. We thereby aggregate the merged depth data from the four virtual pinhole stereo cameras along the trajectory estimated by the filter. As the filter estimates are locally stable but globally subject to drift, we split the aggregated data into so-called submaps of limited uncertainty and size. Our navigation filter is a *local reference filter* [23], we thus can always switch its frame of reference into the origin of the current submap. This allows us to maintain long-term consistency and numerical stability within the filter as well as a more accurate integration of the filter's estimate into the overlying SLAM system [26]. We add the submap origins as nodes to a SLAM graph and connect them via the filter estimates as edges weighted by their Gaussian uncertainty. Loop closure constraints from landmark detections or map matches, can easily be integrated into the graph for online global pose and map optimization, as described in [7]. We construct the SLAM graph at a high level of abstraction, i.e., on top of the local reference filter estimates and can thereby keep its size small and incremental online optimization steps fast. This is in particular beneficial in a setup with multiple high-frequency data sources and filter-internal states, like our setup on Ardea with four key frame-based visual odometries. As the information of all visual odometries is fused in the local reference filter, the SLAM graph does neither increase in size nor in complexity by adding more high-frequency measurements or estimates like, in this case, the additional visual odometries.

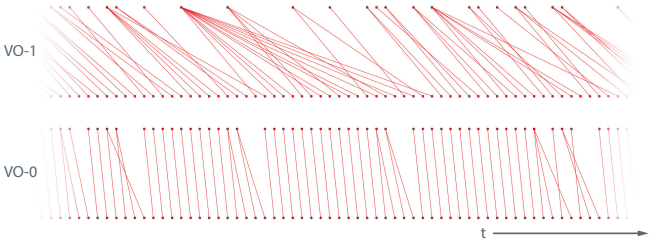

Fig. 8: Keyframes selected by VO-0 and VO-1 during a forward flight. The upper row of dots in each sub-figure illustrates the keyframes. The lower row of dots represents the current image. Lines indicate which keyframe was picked to perform the pose estimation. Note that both VOs choose quite different keyframes.

VI. Experiments and Results

To show the benefits of our system, we present results of four experiments. The first experiment demonstrates the independent selection of keyframes by each visual odometry. In the second experiment, data from an MAV flight was used to simulate a camera failure. The third experiment shows a robust flight over a poorly-textured scene. The final experiment illustrates the mapping capabilities of our MAV.

A. Independent Keyframe Selection

The motion estimates of the different visual odometries are based on different regions of a scene depending on the field of views of the corresponding cameras. Depending on the current movement and structure of the scene, each VO will experience different optic flow. Therefore, depending on the field of view, the optimal set of keyframes will be different. Our system is taking this into account since each visual odometry is choosing their own individual keyframes as described in section IV. This is an advantage over a system running just a single VO in a joint-optimization over the entire field of view. Here the system is just picking a single set of keyframes, which is likely to be sub-optimal. In our case each VO chooses a set of keyframes based on its individual FOV, which will result in a better selection.

To demonstrate that behavior, we moved the MAV in a straight line and recorded the keyframes selected for each VO which will be used for pose estimation. Figure 8 illustrates the keyframes of VO-0 and VO-1. The VO-0 is looking straight down on the floor, whereas the VO-1 is looking in an 30° angle (see Fig. 3). One can see that the upper VO is referencing to images much further back in time than the lower one. This is an expected behavior since the flow field is stronger in the lower image than the upper one. This is because the movement is perpendicular to the z-axis of the lower virtual camera, whereas the z-axis of the upper camera is closer to the direction of translation. Therefore, features cannot be tracked for a very long time in the lower image, since they do not occur for many frames.

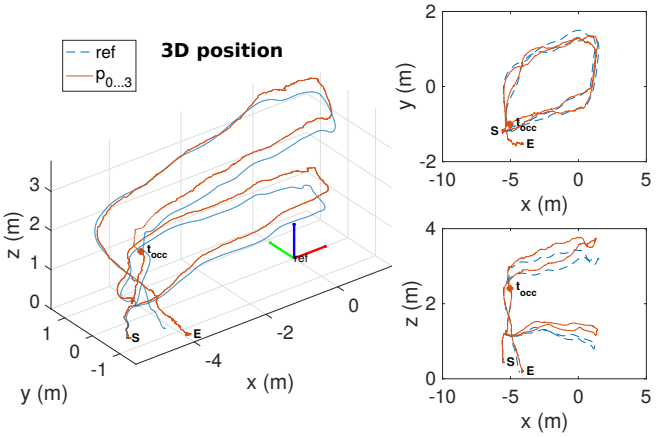

Fig. 9: Left: Estimated and reference trajectory of a flight starting at point **S** and ending at point **E**. Top-right: Top-down view of the trajectories. Bottom-right: Lateral view of the trajectories.

B. Robust State Estimation despite Camera Failures

During this experiment, the MAV was flying along a trajectory as shown in Fig. 9. It took off at location **S** and landed at location **E**. Between its starting and landing location it flew two rounds at different heights. During

the flight, all four visual odometries were used for state estimation until time t_{occ}. At time t_{occ}, the lower left camera 1 that provides input to V0-0 and VO-1 was switched off in order to show the robustness of the approach to camera or visual odometry failures. In Fig. 10, the translational error $\Delta p_{0\ldots3}$ between ground truth provided by a Vicon tracking system and the estimated position is shown. The subscript $\mathbf{0\ldots3}$ indicates that all four visual odometries were used if available. In addition to the error $\Delta p_{0\ldots3}$, the positional errors Δp_i for state estimation using a single odometry $i = 0, 1, 2, 3$ Δp_0, Δp_1, Δp_2, Δp_3 are shown. They result from several replays of the filter with only a single visual odometry activated at a time. Due to multiple, unrecognized outliers and poor features caused by direct illumination from lights suspended from poorly lighted ceiling, which are used for the visual odometry VO-2, the error Δp_2 increases very fast. As the visual odometries VO-0 and VO-1 are switched off at time t_{occ}, the corresponding errors increase afterwards. Nevertheless, the error $\Delta p_{0\ldots3}$ stays small after t_{occ}, and is below the smallest error based on a single odometry, Δp_3.

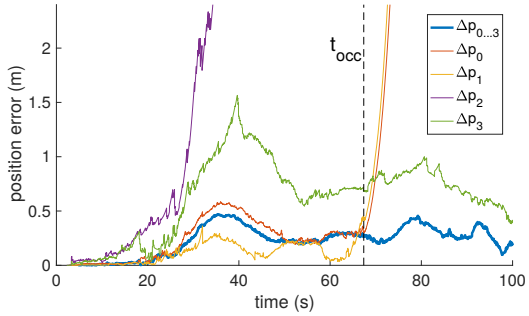

Fig. 10: Position error between estimated and reference trajectory for five different configurations with a failure of two visual odometries (VO-0, VO-1) beginning at time t_{occ} (indicated by dashed vertical line).

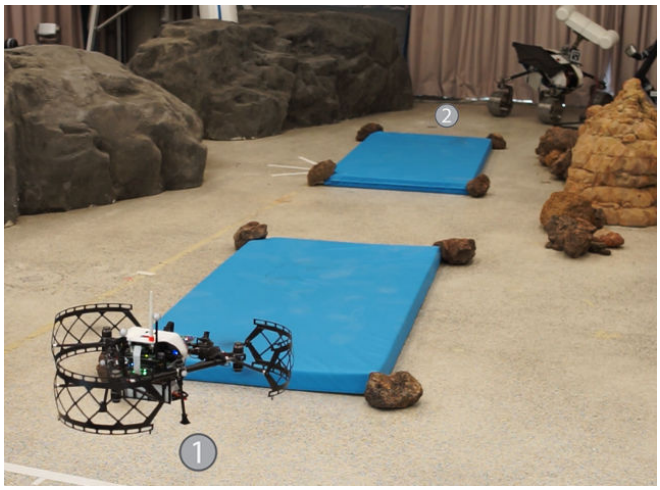

Fig. 11: Flight of Ardea from the starting and landing area (1) over the two blue floor mats to point (2). While flying over the poorly-textured floor mats, not all four visual odometries generate valid output.

C. Robust Flight over Texture-Less Areas

In practice, a flight over or next to areas with poor or indiscernible texture, like small bodies of water, texture-less

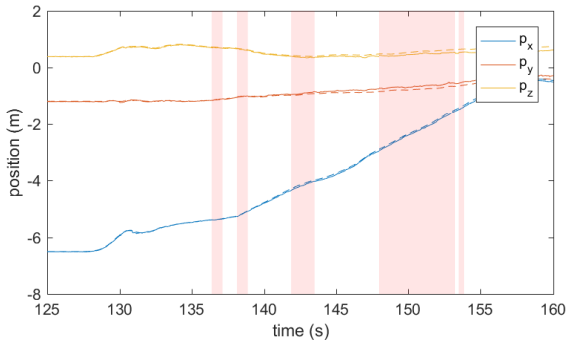

Fig. 12: Estimated (solid lines) and reference (dashed lines) position during flight over two blue floor mats. The colored areas indicate periods where VO-0 failed.

floors, walls and ceilings, is not uncommon. If the visual odometry of a robot perceives mainly such an area, it cannot track enough features to calculate a reliable pose estimation. As a result the filter begins to drift and in cases where the area is large, the MAV might even crash due to the accumulation of drift errors. To study such a scenario and to evaluate the performance of our approach, we conducted an experiment in which mattresses were laid out in the laboratory as shown in figure 11. In this experiment, Ardea flew from position 1 to 2 and back again. The trajectory leads over two mattresses, which have almost no visual texture. 4 VOs are fused online and on-board with the IMU to control the position of the MAV. Figure 12 shows the results of the experiment. The illustration displays the estimated trajectory with the actual flown trajectory obtained by a visual tracking system. When Ardea flew over the mattresses, VO-0 failed several times to perform a reliable pose estimation. This occasions are illustrated in the figure with shaded red regions. Since the fusion filter is not just fusing the output of one VO the failing of this particular VO can be compensated and a potential crash of the MAV could be avoided. Therefore, this experiment shows the strength of fusing multiple visual odometries to get a more robust pose estimation.

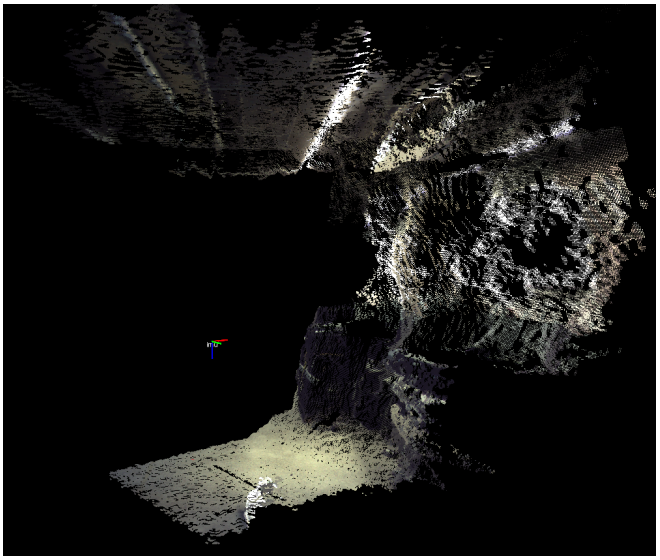

Fig. 13: Single shot pointcloud computed from the eight pinhole images.

D. Mapping of Scene with MAV

Fig. 13 shows the point cloud created by the eight virtual pinhole cameras from a single time sample. The coordinate frame indicates the position of Ardea. The ultra wide field of view of the point clouds provides valuable information above, in front of, and below the MAV. Fig. 14 shows the flight trajectory of Ardea and the resulting accumulated 3D point cloud map computed by our SLAM system, described in section V. It consists of a series of nine submaps. To keep the accumulated error in the local reference filter low, new submaps were started whenever the estimated positional or rotational covariance reached a threshold of $0.1\,\mathrm{m}$ or 5° respectively. In this experiment, the MAV was manually controlled to two waypoints in our laboratory. At these points, it rotated around its yaw-axis. Due to Ardea's large vertical field of view, the floor and ceiling can be mapped simultaneously, resulting in a dense 3D point cloud.

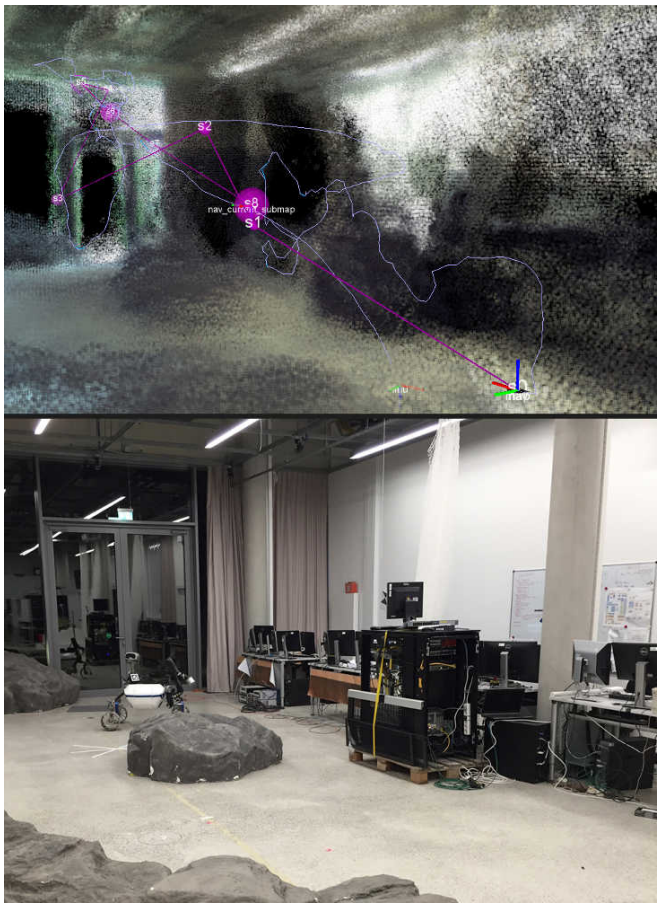

Fig. 14: Top: Resulting map of an MAV flight. The covariance ellipsoids (in magenta) indicate the estimated positional uncertainty with respect to the start frame. Bottom: Photo of the lab for comparison.

VII. Conclusion

In this paper an MAV equipped with four wide-angle cameras was introduced. The large vertical stereo FOV of 240° enables the MAV to perceive objects below, above and in front of the MAV, which is relevant for obstacle avoidance, path planning and efficient mapping. State estimation also benefits from the large FOV due to robust motion estimation provided by four stereo odometries with independent keyframes, which was shown in experiments.

References

[1] T. Tomic, K. Schmid, P. Lutz, A. Dömel, M. Kassecker, E. Mair, I. L. Grixa, F. Ruess, M. Suppa, and D. Burschka, "Toward a fully autonomous UAV," *IEEE Robotics & Automation Magazine*, 2012.

[2] L. Matthies, R. Brockers, Y. Kuwata, and S. Weiss, "Stereo vision-based obstacle avoidance for micro air vehicles using disparity space." in *ICRA*, 2014.

[3] P. Gohl, D. Honegger, S. Omari, M. Achtelik, and R. Siegwart, "Omnidirectional visual obstacle detection using embedded FPGA." in *IROS*, 2015.

[4] A. J. Barry and R. Tedrake, "Pushbroom stereo for high-speed navigation in cluttered environments." in *ICRA*, 2015.

[5] S. Houben, J. Quenzel, N. Krombach, and S. Behnke, "Efficient multi-camera visual-inertial SLAM for micro aerial vehicles," in *IROS*, 2016.

[6] M. Beul, N. Krombach, Y. Zhong, D. Droeschel, M. Nieuwenhuisen, and S. Behnke, "A high-performance MAV for autonomous navigation in complex 3D environments," in *ICUAS*, 2015.

[7] M. J. Schuster, K. Schmid, C. Brand, and M. Beetz, "Distributed Stereo Vision-Based 6D Localization and Mapping for Multi-Robot Teams," *Journal of Field Robotics*, 2018.

[8] T. Oskiper, Z. Zhu, S. Samarasekera, and R. Kumar, "Visual odometry system using multiple stereo cameras and inertial measurement unit." in *CVPR*, 2007.

[9] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2008.

[10] K. Schmid and H. Hirschmüller, "Stereo vision and IMU based real-time ego-motion and depth image computation on a handheld device." in *ICRA*, 2013.

[11] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2006.

[12] S. Abrahama and W. Förstner, "Fish-eye-stereo calibration and epipolar rectification," *ISPRS Journal of Photogrammetry and Remote Sensing*, 2005.

[13] D. Caruso, J. Engel, and D. Cremers, "Large-scale direct SLAM for omnidirectional cameras," in *IROS*, 2015.

[14] H. Hirschmüller, P. R. Innocent, and J. M. Garibaldi, "Fast, unconstrained camera motion estimation from stereo without tracking and robust statistics," in *ICARCV*, 2002.

[15] A. Stelzer, H. Hirschmüller, and M. Görner, "Stereo-vision-based navigation of a six-legged walking robot in unknown rough terrain," *The International Journal of Robotics Research*, 2012.

[16] E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger, "Adaptive and generic corner detection based on the accelerated segment test," in *ECCV*, 2010.

[17] J. R. Taylor, *An Introduction to Error Analysis*. University Science Books, 1982.

[18] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect visual odometry for monocular and multicamera systems," *IEEE Transactions on Robotics*, 2017.

[19] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras," *IEEE Transactions on Robotics*, 2017.

[20] R. Wang, M. Schwörer, and D. Cremers, "Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras," in *ICCV*, 2017.

[21] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2018.

[22] K. Schmid, F. Ruess, M. Suppa, and D. Burschka, "State estimation for highly dynamic flying systems using key frame odometry with varying time delays," in *IROS*, 2012.

[23] K. Schmid, F. Ruess, and D. Burschka, "Local reference filter for life-long vision aided inertial navigation," in *FUSION*, 2014.

[24] H. Wu, S. Chen, B. Yang, and K. Chen, "Feedback robust cubature Kalman filter for target tracking using an angle sensor," *Sensors*, 2016.

[25] M. J. Schuster et al., "Towards autonomous planetary exploration," *Journal of Intelligent & Robotic Systems*, 2017.

[26] M. J. Schuster, C. Brand, H. Hirschmüller, M. Suppa, and M. Beetz, "Multi-robot 6D graph SLAM connecting decoupled local reference filters," in *IROS*, 2015.