2018 IEEE International Conference on Robotics and Automation (ICRA)
May 21-25, 2018, Brisbane, Australia

Flexible Stereo: Constrained, Non-Rigid, Wide-Baseline Stereo Vision for Fixed-Wing Aerial Platforms

Timo Hinzmann, Tim Taubner, and Roland Siegwart

Abstract— **This paper proposes a computationally efficient method to estimate the time-varying relative pose between two visual-inertial sensor rigs mounted on the flexible wings of a fixed-wing unmanned aerial vehicle (UAV). The estimated relative poses are used to generate highly accurate depth maps in real-time and can be employed for obstacle avoidance in low-altitude flights or landing maneuvers. The approach is structured as follows: Initially, a wing model is identified by fitting a probability density function to measured deviations from the nominal relative baseline transformation. At run-time, the prior knowledge about the wing model is fused in an Extended Kalman filter (EKF) together with relative pose measurements obtained from solving a relative perspective N-point problem (PNP), and the linear accelerations and angular velocities measured by the two inertial measurement units (IMU) which are rigidly attached to the cameras. Results obtained from extensive synthetic experiments demonstrate that our proposed framework is able to estimate highly accurate baseline transformations and depth maps.**

I. INTRODUCTION

Reliable and long-range obstacle detection is essential to enable low-altitude flights or landing maneuvers for fixed-wing unmanned aerial vehicles (UAVs). While full size drones often employ precise Lidar or Radar systems, small-scale UAVs usually cannot afford to carry this type of heavy payload with high power consumption, but need to rely on cheaper and more light-weight optical camera systems. To recover the absolute scale information from cameras, either a classical stereo setup with a fixed and calibrated baseline transformation is used, or virtual stereo pairs are computed from a monocular camera setup.

A. Monocular Camera Setup: Flying into the Epipole

Monocular visual-inertial sensor setups have proven to be well suited for applications such as simultaneous localization and mapping (SLAM) [1] or planar dense reconstruction [2]. The algorithms perform best when using a down-looking camera where the epipoles are outside of the camera field of view and fronto-parallel motion with respect to the ground is performed. However, within the application of obstacle detection and avoidance, the region of interest is in front of the aircraft and a forward facing or oblique camera needs to be employed. In this setup, the optical axis is closely aligned with the aircraft's direction of flight, resulting in a blind spot just in the region of interest. While this problem can be partially overcome by using, for instance, polar rectification [3], the procedure is inherently error prone since the pixels close to the image center have only small changes in image coordinates in the subsequent frame.

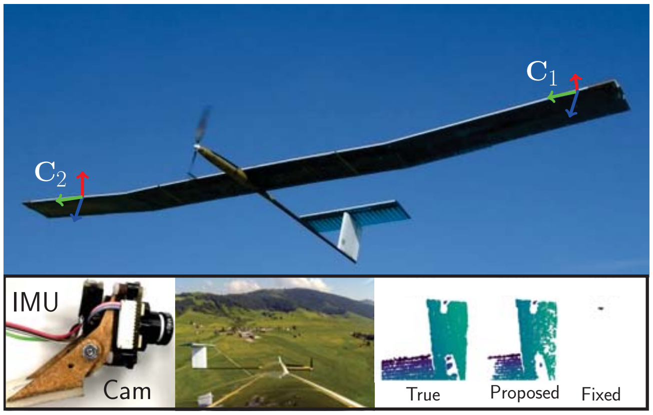

Fig. 1: Envisioned use-case for the proposed non-rigid wide-baseline stereo vision algorithm for fixed-wing UAVs.[1] Bottom left: One of the two light-weight, rigid camera-IMU rigs. Bottom right: Characteristic frame from synthetic dataset comparing depth map obtained from ground-truth poses, estimated poses, and poses obtained from a fixed baseline assumption.

B. Fixed-Baseline Stereo

Due to the shortcomings of the monocular sensor setup, rotary-wing UAVs, such as quadro- or hexacopters, are usually equipped with a fixed baseline stereo pair to perform onboard and real-time obstacle avoidance while providing a high level of reliability. However, the baseline of the stereo rigs, and hence the range where low depth uncertainties are obtained, are optimized for indoor scenarios with maximum speeds of few meters per second. The small baselines, ranging from $9\,\mathrm{mm}$ up to $0.5\,\mathrm{m}$ [4], leave not enough time for short-term path planning, control, and actuators to react in highly dynamic fixed-wing UAV scenarios. Upscaling the baseline to more suited depth ranges while still ensuring rigidity of the stereo rig, on the other hand, would impair aerodynamics, the required flexibility of the wing, and inevitably lead to a higher payload mass.

C. Aim of this Work: Non-Rigid Wide-Baseline Stereo

Consequently, for fixed-wing aircrafts it seems appealing to take advantage of the wing span and to mount the cameras on the outer wing region. However, especially solar-powered aircrafts [5] which are optimized for wing

All authors are with the Autonomous Systems Lab, ETH Zurich, Leonhardstrasse 21, LEE, CH-8092 Zurich, Switzerland. {firstname.lastname}@mavt.ethz.ch.

[1] A video illustrating *AtlantikSolar*'s flapping wing behavior is accessible under `https://youtu.be/8m76Mx9m2nM`.

978-1-5386-3081-5/18/$31.00 ©2018 IEEE

area and updraft, show flapping wing behavior leading to translational and rotational deformations from the nominal baseline in the range of several degrees and centimeters. These disturbances affect the depth estimation in two ways: Firstly, since the assumption of an accurately calibrated fixed relative transformation between the stereo pair is violated, the depth estimates become heavily distorted. For example, disturbances in yaw and pitch angle estimates lead to errors in the depth estimate which are quadratic to the depth [6]. Secondly, stereo vision algorithms usually rely on matching correspondences along epipolar lines [7]. Therefore, even small errors in the estimated rotation can already be fatal since the correspondences are searched along the wrong line. This leads to an almost empty depth map since no correspondences can be found. It could be overcome by a full 2D search over the whole image which, however, is not computationally feasible in real-time.

The aim of this work is therefore to estimate the time-varying relative pose between the two camera-IMU rigs with low delay and close to the image capture frequency as illustrated in Fig. 1. We achieve this by fusing the high-frequency, low-variance inertial sensors with the low-frequency, bias-free vision-based relative pose estimation in combination with the probabilistic wing model in an efficient Extended Kalman Filter (EKF) formulation [8], [9].

In summary, we present a light-weight formulation for wide-baseline non-rigid stereo that builds up on [8]. Compared to [8], we see the following contributions: Most importantly, we propose a wing model in form of a relative pose prior. While [8] makes no assumption about the relative pose of the IMUs, we take advantage of our wing deflection model to better constrain the EKF and to identify and reject visual outliers. While [8] treated the vision module as a black box, we extract and match features and incorporate a relative perspective n-point (PNP) solver as vision-based relative motion estimation module. The pipeline is validated extensively based on synthetic datasets, in particular, with respect to the quality of a) the estimated baseline transformation, and b) the resulting depth map.

II. RELATED WORK

The *auto-calibration* problem is the retrieval of the rigid, non-changing stereo baseline transformation in known [10] or unknown [11], [12] environments. In the latter, images are taken over time and keypoints are extracted to perform stereo bundle adjustment. This can be extended towards *re-calibration* of stereo rigs during operation, also known as *online-calibration*. By having a good initial estimate, e.g. obtained from auto-calibration, the baseline can be re-estimated assuming it only changed slightly. The techniques are then essentially the same as for auto-calibration in unknown environments. Perturbations from the nominal stereo baseline can be detected by an increase in reprojection errors, hence triggering a re-calibration process. Extending further on this idea, Warren et al. [11] continuously estimate slow changes (e.g. thermally induced) in the baseline. The key difference of the *continuous self-calibration* methods are that they do not assume that the baseline is rigid over the whole duration of the calibration process. However, in contrast to our setting, the baseline changes in [11] are still assumed to be only small and slow. Work on continuous calibration in presence of high-frequent noise exists. In [13], a down-looking stereo pair with a wide-baseline ($0.7\,\mathrm{m}$) is employed and the relative transform between the cameras, together with the poses of the stereo rig itself is estimated offline in a bundle adjustment problem. Similarly to our work, Warren et al. incorporate prior knowledge of the deformation in the stereo rig. This is achieved in form of a cost-based bound to tightly constrain the estimated transform. However, in [13], the anticipated changes in baseline are induced by vibration and hence relatively small (a few millimeters in translation and only tenth of a degree in rotation) making the prior a good assumption. In contrast, we are dealing with an increase of two order of magnitudes (decimeters and more than ten degrees).

We are aware of two different approaches to real-time baseline estimation in which the baseline deviations closer resemble our scenario: The *first approach*, described in [14], [15], [16], is based on a modal-analysis of a wing which enables to estimate vibrational disturbances by only measuring with accelerometers (and no gyroscopes). In [14], this method is introduced and show-cased on an Euler-Bernoulli beam model employing two accelerometers. This is extended in [15] to wing-mounted stereo rigs and experimental results are obtained during periodic and random excitation of a real wing model. A total of 24 accelerometers are mounted over the wing to perform modal analysis. The camera mount-points are on the wing-tips resulting in a theoretical baseline of $240\,\mathrm{cm}$. However, only an analysis of the wing and the corresponding baseline estimation errors was performed. The results suggest significant improvements over a fixed-baseline assumption. Note that these results are obtained in experiments performed in small scale and then extrapolated. In contrast, we mount the IMUs (including a gyroscope) spatially close and rigidly to the cameras and do not rely on a physically motivated wing-model. Furthermore, their approach suggests taking images when the motion is minimal to minimize effects of motion blur. Instead we rely on global shutter cameras with short and synchronized shutter times.

The *second approach* is described in [8]: It employs an EKF to estimate the relative position between two micro air vehicles (MAV), each equipped with an IMU and a down-looking camera, assuming an overlapping field of view. The authors leave the visual part as a black box and suggest the use of a visual SLAM framework with scale propagation such as PTAM [1]. In contrast to our work, since MAVs can move independently of each other, no prior knowledge on the baseline transform is assumed. Our work is based to a large extent on the EKF described in [8], but extends the measurement model by fusing our prior knowledge of the baseline. Furthermore, the vision module is implemented in form of a relative PNP solver. Since the latter is based solely on sparse correspondence matching of the current stereo pair, no scale propagation can be performed which simplifies the

EKF. Furthermore, as the relative baseline transformation may change quickly during flight, no feature tracking or descriptor matching over time is performed.

III. METHODOLOGY

An overview of our proposed framework is illustrated in Fig. 2. The incoming image stream is used both, for visual estimation of the relative pose using a relative PNP solver, and for generating depth maps. The visual estimates are fused with the baseline prior and fed into an EKF together with high-frequent (≥ 100 Hz) IMU measurements. The obtained filtered estimate of the baseline transformation $\hat{\boldsymbol{T}}$ is then used to rectify the images so that dense matching can be done reliably and fast via correspondence search along epipolar lines [17]. The resulting depth maps can be fed into external applications, e.g. a flight controller to avoid obstacles, or to reconstruct the environment. We adopt the EKF as described in [8]. There is one fundamental modification: As our visual pose estimator is based solely on the most recent two frames and does not have scale propagation we do not include λ in our state vector. Correspondingly, our measurement model for obtaining relative pose estimates is adopted. For completeness and to clarify the modifications with respect to [8], we summarize all elements of the Kalman filter that are required for understanding in the following.

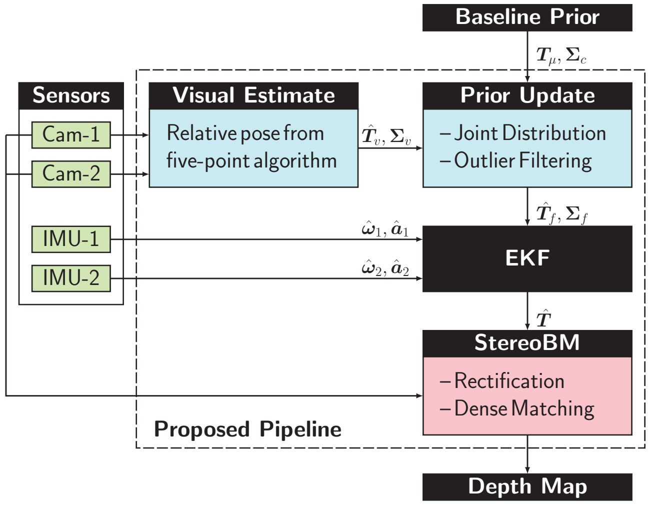
Baseline Prior
$\boldsymbol{T}_\mu, \boldsymbol{\Sigma}_c$
Sensors
Cam-1
Cam-2
IMU-1
IMU-2
Visual Estimate
Relative pose from five-point algorithm
$\hat{\boldsymbol{T}}_v, \boldsymbol{\Sigma}_v$
Prior Update
– Joint Distribution
– Outlier Filtering
$\hat{\boldsymbol{T}}_f, \boldsymbol{\Sigma}_f$
$\hat{\boldsymbol{\omega}}_1, \hat{\boldsymbol{a}}_1$
$\hat{\boldsymbol{\omega}}_2, \hat{\boldsymbol{a}}_2$
EKF
$\hat{\boldsymbol{T}}$
StereoBM
– Rectification
– Dense Matching
Proposed Pipeline
Depth Map

Fig. 2: Proposed framework to estimate the time-varying relative transformation between two cameras. The efficient EKF formulation fuses a wing deformation prior with the information from two IMUs and two cameras.

A. The State Vector

The state consists of the relative angular rates $\boldsymbol{\omega}_1^1$, $\boldsymbol{\omega}_2^2$, the linear accelerations expressed in the world frame ${}_w\boldsymbol{a}_1^1$, ${}_w\boldsymbol{a}_2^2$, the relative orientation expressed as a quaternion $\bar{q}_1^2$ and the (metric) relative position $\boldsymbol{p}_1^2$ and finally the relative velocity ${}_w\boldsymbol{v}_1^2$. Stacked together these values form the 22-element state vector

$$\boldsymbol{x} = \left[\bar{q}_1^{2\top} \; \boldsymbol{\omega}_1^{1\top} \; \boldsymbol{\omega}_2^{2\top} \; \boldsymbol{p}_1^{2\top} \; {}_w\boldsymbol{v}_1^{2\top} \; {}_w\boldsymbol{a}_1^{1\top} \; {}_w\boldsymbol{a}_2^{2\top}\right]^\top . \tag{1}$$

To simplify notation, super- and subscripts are dropped in the following and the states are simply denoted by

$$\boldsymbol{x} = \left[\bar{q}^\top \; \boldsymbol{\omega}_1^\top \; \boldsymbol{\omega}_2^\top \; \boldsymbol{p}^\top \; \boldsymbol{v}^\top \; \boldsymbol{a}_1^\top \; \boldsymbol{a}_2^\top\right]^\top . \tag{2}$$

B. State Equations

Angular velocities $\boldsymbol{\omega}_1, \boldsymbol{\omega}_2$ and linear accelerations $\boldsymbol{a}_1$, $\boldsymbol{a}_2$ are modeled as independent random zero-mean Gaussian walks with (diagonal) covariances $\boldsymbol{\sigma}_{\boldsymbol{\omega}_1}, \boldsymbol{\sigma}_{\boldsymbol{\omega}_2}, \boldsymbol{\sigma}_{\boldsymbol{a}_1}$ and $\boldsymbol{\sigma}_{\boldsymbol{a}_2}$. The corresponding random vectors are denoted by $\boldsymbol{n}_{\boldsymbol{\omega}_1}$, $\boldsymbol{n}_{\boldsymbol{\omega}_2}$, $\boldsymbol{n}_{\boldsymbol{a}_1}$ respectively $\boldsymbol{n}_{\boldsymbol{a}_2}$. For quaternion multiplication the angular velocities are represented as quaternions $\bar{\omega}_i = [0 \; \boldsymbol{\omega}_i^\top]^\top$ (for $i = 1, 2$), $\boldsymbol{C}$ denotes the rotation matrix corresponding to the quaternion $\bar{q}$ and $\lfloor \boldsymbol{\omega}_1 \times \rfloor$ denotes the skew-symmetric matrix. The final state equations are as follows:

$$\dot{\bar{q}} = 0.5 \cdot (\bar{q} \otimes \bar{\omega}_2 - \bar{\omega}_1 \otimes \bar{q}) \tag{3}$$

$$\dot{\boldsymbol{\omega}}_1 = \boldsymbol{n}_{\boldsymbol{\omega}_1} \tag{4}$$

$$\dot{\boldsymbol{\omega}}_2 = \boldsymbol{n}_{\boldsymbol{\omega}_2} \tag{5}$$

$$\dot{\boldsymbol{p}} = \boldsymbol{v} - \lfloor \boldsymbol{\omega}_1 \times \rfloor \cdot \boldsymbol{p} \tag{6}$$

$$\dot{\boldsymbol{v}} = \boldsymbol{C} \cdot \boldsymbol{a}_2 - \boldsymbol{a}_1 - \lfloor \boldsymbol{\omega}_1 \times \rfloor \cdot \boldsymbol{v} \tag{7}$$

$$\dot{\boldsymbol{a}}_1 = \boldsymbol{n}_{\boldsymbol{a}_1} \tag{8}$$

$$\dot{\boldsymbol{a}}_2 = \boldsymbol{n}_{\boldsymbol{a}_2} \tag{9}$$

Detailed derivations can be found in [8].

C. Error State Representation

From here on, an estimated state is denoted by $\hat{.}$ and a preceding Δ denotes the error state for an additive error. For the quaternion $\bar{q}$ a multiplicative error model is used. The error is denoted by $\delta\boldsymbol{\theta}$. The error state vector is then

$$\tilde{\boldsymbol{x}} = \left[\delta\boldsymbol{\theta}^\top \; \Delta\boldsymbol{\omega}_1^\top \; \Delta\boldsymbol{\omega}_2^\top \; \Delta\boldsymbol{p}^\top \; \Delta\boldsymbol{v}^\top \; \Delta\boldsymbol{a}_1^\top \; \Delta\boldsymbol{a}_2^\top\right]^\top . \tag{10}$$

Again, we refer to [8] for derivations of the error state update equations. Noting that $\hat{\boldsymbol{C}}$ is the corresponding rotation matrix of the quaternion $\hat{\boldsymbol{q}}$, the final error state equations are

$$\dot{\delta\boldsymbol{\theta}} = -\lfloor \hat{\omega}_2 \times \rfloor \cdot \delta\boldsymbol{\theta} - \hat{\boldsymbol{C}}^\top \cdot \Delta\boldsymbol{\omega}_1 + \Delta\boldsymbol{\omega}_2 \tag{11}$$

$$\dot{\Delta\boldsymbol{\omega}}_1 = \boldsymbol{n}_{\boldsymbol{\omega}_1} \tag{12}$$

$$\dot{\Delta\boldsymbol{\omega}}_2 = \boldsymbol{n}_{\boldsymbol{\omega}_2} \tag{13}$$

$$\dot{\Delta\boldsymbol{p}} = \lfloor \hat{\boldsymbol{p}} \times \rfloor \cdot \Delta\boldsymbol{\omega}_1 - \lfloor \hat{\omega}_1 \times \rfloor \cdot \Delta\boldsymbol{p} + \Delta\boldsymbol{v} \tag{14}$$

$$\begin{aligned} \dot{\Delta\boldsymbol{v}} = & -\hat{\boldsymbol{C}} \cdot \lfloor \hat{\boldsymbol{a}}_2 \times \rfloor \cdot \delta\boldsymbol{\theta} + \lfloor \hat{\boldsymbol{v}} \times \rfloor \cdot \Delta\boldsymbol{\omega}_1 \\ & - \lfloor \hat{\omega}_1 \times \rfloor \cdot \Delta\boldsymbol{v} - \Delta\boldsymbol{a}_1 + \hat{\boldsymbol{C}} \cdot \Delta\boldsymbol{a}_2 \end{aligned} \tag{15}$$

$$\dot{\Delta\boldsymbol{a}}_1 = \boldsymbol{n}_{\boldsymbol{a}_1} \tag{16}$$

$$\dot{\Delta\boldsymbol{a}}_2 = \boldsymbol{n}_{\boldsymbol{a}_2} \tag{17}$$

D. State Covariance Prediction

These equations enable the computation of the continuous system matrix $\boldsymbol{F}_c = \frac{\partial \dot{\tilde{\boldsymbol{x}}}}{\partial \tilde{\boldsymbol{x}}}$ and noise matrix $\boldsymbol{G}_c = \frac{\partial \dot{\tilde{\boldsymbol{x}}}}{\partial \boldsymbol{n}}$ with $\boldsymbol{n} = \left[\boldsymbol{n}_{\boldsymbol{\omega}_1}^\top \; \boldsymbol{n}_{\boldsymbol{\omega}_2}^\top \; \boldsymbol{n}_{\boldsymbol{a}_1}^\top \; \boldsymbol{n}_{\boldsymbol{a}_2}^\top\right]^\top$. The computed Jacobi matrices are given in [9, (4.21)]. Assuming $\boldsymbol{F}_c$ to be constant over the integration period yields $\boldsymbol{F}_d = \exp(\Delta t \cdot \boldsymbol{F}_c)$ for a given time-step Δt. This is approximated to the zeroth order term by expanding the exponential series, resulting in $\boldsymbol{F}_d \approx \boldsymbol{I} + \boldsymbol{F}_c \Delta t$. With the continuous time noise covariance matrix $\boldsymbol{Q}_c = \operatorname{diag}\left(\left[\boldsymbol{\sigma}^2_{\boldsymbol{\omega}_1}\ \boldsymbol{\sigma}^2_{\boldsymbol{\omega}_2}\ \boldsymbol{\sigma}^2_{\boldsymbol{a}_1}\ \boldsymbol{\sigma}^2_{\boldsymbol{a}_2}\right]\right)$, the discrete time noise covariance matrix $\boldsymbol{Q}_d$ is computed according to [18]. Assuming $\boldsymbol{F}_d$ to stay constant during the integration period yields

$$\boldsymbol{Q}_d = \int_0^{\Delta t} \boldsymbol{F}_d(\tau)\cdot\boldsymbol{G}_c\cdot\boldsymbol{Q}_c\cdot\boldsymbol{G}_c^\top\cdot\boldsymbol{F}_d(\tau)^\top\cdot d\tau \tag{18}$$

$$= \Delta t\cdot\boldsymbol{F}_d\cdot\boldsymbol{G}_c\cdot\boldsymbol{Q}_c\cdot\boldsymbol{G}_c^\top\cdot\boldsymbol{F}_d^\top. \tag{19}$$

The updated state covariance matrix for the EKF is then computed as

$$\boldsymbol{P}_{k+1|k} = \boldsymbol{F}_d\cdot\boldsymbol{P}_{k|k}\cdot\boldsymbol{F}_d^\top + \boldsymbol{Q}_d. \tag{20}$$

E. State Prediction

The states are predicted by zeroth order integration according to (3), (6), respectively (7) given above:

$$\hat{\bar{q}}_{k+1} = \hat{\bar{q}}_k + 0.5\cdot\Delta t\cdot\left(\hat{\bar{q}}_k\otimes\hat{\bar{\omega}}_{2,k} - \hat{\bar{\omega}}_{1,k}\otimes\hat{\bar{q}}_k\right) \tag{21}$$

$$\hat{\boldsymbol{p}}_{k+1} = \hat{\boldsymbol{p}}_k + \left(\hat{\boldsymbol{v}}_k - \lfloor\hat{\boldsymbol{\omega}}_{1,k}\times\rfloor\cdot\hat{\boldsymbol{p}}_k\right)\cdot\Delta t \tag{22}$$

$$\hat{\boldsymbol{v}}_{k+1} = \hat{\boldsymbol{v}}_k + \left(\hat{\boldsymbol{C}}\cdot\hat{\boldsymbol{a}}_{2,k} - \hat{\boldsymbol{a}}_{1,k} - \lfloor\hat{\boldsymbol{\omega}}_{1,k}\times\rfloor\cdot\hat{\boldsymbol{v}}_k\right)\cdot\Delta t \tag{23}$$

F. Vision-Based Relative Pose Measurements

The vision-based relative pose estimates are obtained in three steps. Classical feature descriptor, matcher, and relative PNP types are selected for a proof of concept: Firstly, SURF keypoints [19] are detected and their descriptors are extracted in both images. Secondly, feature correspondences are established using a FLANN-based matcher [20]. Finally, the bearing vectors are computed for both frames and every matched feature. Based on the two sets of bearing vectors, a relative pose up to a scale is estimated by means of finding the fundamental matrix $\boldsymbol{F}$ [7]. The estimated fundamental matrix $\boldsymbol{F}$ is then converted to a unit vector $\hat{\boldsymbol{p}}_v$ representing the direction of translation and the rotation quaternion $\hat{\bar{q}}_v$. To solve the PNP problem, we employ the 5-point Nister algorithm [21]. Since PNP solvers are sensitive to wrong associations, Random Sample Consensus (RANSAC) [22] is employed [23]. To fuse the visual estimates, the output of the algorithm is interpreted probabilistically: The measurements are modeled as

$$\hat{\boldsymbol{p}}_v = \frac{\boldsymbol{p}+\Delta\boldsymbol{p}_v}{\|\boldsymbol{p}+\Delta\boldsymbol{p}_v\|_2} \tag{24}$$

$$\hat{\bar{q}}_v = \bar{q}\otimes\delta\bar{q}_v \tag{25}$$

where $\hat{\boldsymbol{p}}_v$ and $\hat{\bar{q}}_v$ are the direction of translation respectively orientation estimated by the vision-based relative pose estimation module (i.e. in our case the PNP solver). The corresponding measurement error is denoted by $\Delta\boldsymbol{p}_v$, $\delta\bar{q}_v$. Approximating the error quaternion $\delta\bar{q}_v$ by small angles $\delta\boldsymbol{\theta}_v$ allows expressing the error transformation as a 6-dimensional vector $\delta\boldsymbol{T}_v = \left[\delta\boldsymbol{\theta}_v^\top\ \Delta\boldsymbol{p}_v^\top\right]^\top$. This error is then approximated by a zero-mean Gaussian with covariance $\boldsymbol{\Sigma}_v$.

G. Establishing a Wing Model

In this section, the notion of baseline calibration is extended beyond rigidity. Instead of finding the fixed relative transform between the two cameras of a stereo rig as the result of an (usually over-constrained) optimization problem flexibility is embraced. The wing model is captured by a nominal baseline transformation $\boldsymbol{T}_\mu$ and a probabilistic error model captured in a random vector $\delta\boldsymbol{T}$. For notational clearance the baseline transform $\boldsymbol{T}_{\mathbf{I_1}}^{\mathbf{I_2}}$ is simply denoted by $\boldsymbol{T}$ or in its parts $\bar{q}$ and $\boldsymbol{p}$. The mean transform $\boldsymbol{T}_\mu$ is expressed in parts by the mean quaternion $\bar{q}_\mu$ and the mean relative position $\boldsymbol{p}_\mu$. This allows to express the disturbance in rotation via a multiplicative error model $\bar{q} = \bar{q}_\mu\otimes\delta\bar{q}$ and in position as a simple additive error $\boldsymbol{p} = \boldsymbol{p}_\mu + \Delta\boldsymbol{p}$. The (small) quaternion $\delta\bar{q}$ is then approximated via the small angle approximation $\delta\boldsymbol{\theta}$. With that, $\delta\boldsymbol{T} = \left[\delta\boldsymbol{\theta}^\top\ \delta\boldsymbol{p}^\top\right]^\top$ is defined. Note that the random vector $\delta\boldsymbol{T}$ has zero mean. In the spirit of the EKF, the error of the *calibration* is modeled as a Gaussian: $\delta\boldsymbol{T}\sim\mathcal{N}(\mathbf{0},\boldsymbol{\Sigma}_c)$. The six dimensions of $\delta\boldsymbol{T}$ are correlated (i.e. when there is a roll-disturbance, there is also a disturbance in the z direction), however not linearly. Therefore, the distribution is approximated as independent Gaussians and thus the covariance matrix is diagonal: $\boldsymbol{\Sigma}_c = \operatorname{diag}(\boldsymbol{\sigma}^2_{\delta\boldsymbol{T}})$.

H. Fusing the Baseline Prior with Visual Estimates

The visual measurements of the baseline $\hat{\bar{q}}_v$ and $\hat{\boldsymbol{p}}_v$ are then combined with the baseline prior to obtain the maximum a posteriori estimates $\hat{\bar{q}}_f$ and $\hat{\boldsymbol{p}}_f$. Since both, the baseline prior and the measurement error, are modeled as zero-mean Gaussians, the a posteriori distribution is again a Gaussian.

The visual estimates are expressed as the deviations $\delta\hat{\bar{q}}_v$ from the calibrated mean such that $\hat{\bar{q}}_v = \bar{q}_\mu\otimes\delta\hat{\bar{q}}_v$ and $\Delta\hat{\boldsymbol{p}}_v = \hat{\boldsymbol{p}}_v - \boldsymbol{p}_\mu$. The error quaternion $\delta\hat{\bar{q}}_v$ is approximated by small angles $\delta\hat{\boldsymbol{\theta}}_v$.

The new estimates of the baseline deviation $\delta\hat{\bar{q}}_f$ and $\Delta\hat{\boldsymbol{p}}_f$ are computed such that the estimated baseline is $\hat{\bar{q}}_f = \bar{q}_\mu\otimes\delta\hat{\bar{q}}_f$ and $\hat{\boldsymbol{p}}_f = \boldsymbol{p}_\mu + \Delta\hat{\boldsymbol{p}}_f$. Again, $\delta\hat{\bar{q}}_f$ is approximated by small angles $\delta\hat{\boldsymbol{\theta}}_f$. By interpreting the fusion of the measurement with the prior as a KF update step, the new estimates and the a posteriori covariance matrix is given by:

$$\begin{bmatrix}\delta\hat{\boldsymbol{\theta}}_f\\ \Delta\hat{\boldsymbol{p}}_f\end{bmatrix} = \boldsymbol{\Sigma}_c(\boldsymbol{\Sigma}_c+\boldsymbol{\Sigma}_v)^{-1}\begin{bmatrix}\delta\hat{\boldsymbol{\theta}}_v\\ \Delta\hat{\boldsymbol{p}}_v\end{bmatrix} \tag{26}$$

$$\boldsymbol{\Sigma}_f = \boldsymbol{\Sigma}_c - \boldsymbol{\Sigma}_c(\boldsymbol{\Sigma}_c+\boldsymbol{\Sigma}_v)^{-1}\boldsymbol{\Sigma}_c \tag{27}$$

I. Visual Outlier Rejection

The baseline prior is used to filter outlier estimates from the relative PNP pipeline. Depending on the scene, the visual pose estimation may give severe outliers. This happens, for instance, if the cameras see mostly uniformly colored sky or landscape without salient keypoints for detection and matching. In these cases, the visual pose estimate is ignored altogether and, instead, an artificial measurement update corresponding to the baseline prior is fed into the EKF. Filtering of outliers is achieved by limiting the maximum distance of the visual estimate $\hat{\boldsymbol{T}}_v$ to the mean baseline $\boldsymbol{T}_\mu$ in any dimension by a factor k (in our case 2) of the standard deviation. That is, we set $\hat{\bar{q}}_f = \hat{\bar{q}}_\mu$ and $\hat{\boldsymbol{p}}_f = \hat{\boldsymbol{p}}_\mu$ if $\left\|\boldsymbol{\Sigma}_v^{-1}\left[\delta\hat{\boldsymbol{\theta}}_v^2\ \Delta\hat{\boldsymbol{p}}_v^2\right]^\top\right\|_\infty > k^2$.

IV. Simulation Environment

Gazebo-based *RotorS* [24] is used to simulate the forces acting on the flexible wings, and to collect precise ground truth poses and IMU measurements. The camera poses and a publicly available mesh are imported into *Blender* and the photo-realistic images are rendered.

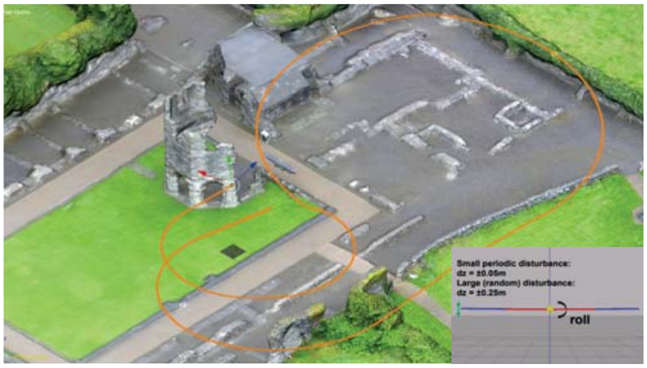

Fig. 3: Synthetic dataset: Overview mesh and aircraft trajectory visualized in *Blender*. The image in the bottom right corner shows the aircraft with two flexible wings modeled in *Rotors*. The mesh was downloaded from `https://skfb.ly/Sq7J`.

a) Aircraft with flexible wings: The aircraft is modeled by two wings, each connected to the airframe by a joint. The joints permit only roll motion, i.e. rotational movement around the aircraft's body's $\boldsymbol{x}$-axis. Furthermore, as shown in Fig. 4, each wing in itself is modeled by two rigid bodies connected by another joint permitting only movement around the wing's center line, i.e. pitching movements. Both joint angles are controlled by a PD-controller which emulates a spring-damper system. The total mass of the aircraft is 2.8 kg, each wing has a mass of 0.4 kg. A disturbance force $\boldsymbol{F}_{\text{dist}} = \boldsymbol{F}_p + \boldsymbol{F}_r$ is applied on the wing tips. The disturbance force is composed of a periodic force $\boldsymbol{F}_p = a_p \sin(f_p t)$ with magnitude a_p and frequency f_p and a random force $\boldsymbol{F}_r$ of random magnitude $a_r \sim \mathcal{N}(a_r, \sigma_{a_r})$ applied at fixed frequency f_r for a duration of t_r. The random forces simulate the effect of wind gusts and excite the wing into brief non-uniform oscillations. The sinusoidal force is applied in same phase with a magnitude of 0.25 N and frequency of $f_p = 1.5\,\text{Hz}$ on both wing tips. The random magnitude is sampled from $\mathcal{N}(1.0, 0.1)$ every 8 s (i.e. $f_r = 0.125\,\text{Hz}$), however, independently for each side and applied over a duration of 0.4 s. The aircraft follows the predefined trajectory as depicted in Fig. 3. A 6-DoF PID-controller is used which applies forces along and torques around all three body axes based on the desired velocity and position.

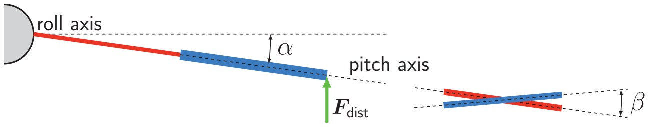

Fig. 4: View from back and right on the flexible wing modeled by two rigid bodies. There are two degrees of freedom: the roll angle α and the pitch angle β.

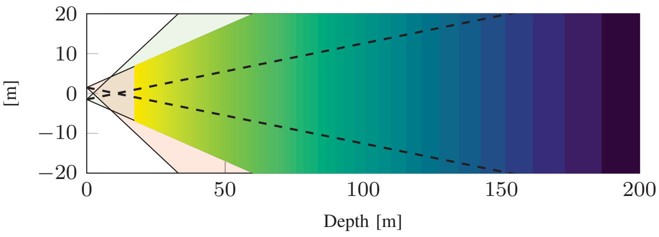

Fig. 5: The FOV in the x-z-plane of the left (red) and right (green) camera with a baseline of 3 m. In the overlapping FOV the area corresponding to each disparity value (12 to 140) is colorized uniformly.

b) Visual-Inertial Camera Rig: The geometry of the visual-inertial stereo rig with a nominal baseline of 3 m is depicted in Fig. 5. The cameras[2] are rotated towards each other by 8° to increase the overlapping field of view (FOV). The IMU coordinate frames coincide with the camera frames, their axes are kept parallel. The IMU *ADIS 16448* [25] is simulated with white noise variances of 1.225e−7 rad/s for angular velocities and 1.6e−5 m^2/s for linear accelerations.

c) Initialization of EKF: The baseline prior is obtained by empirically estimating the mean and variances of the baseline transform as described in Section III-G. To simulate an imperfect baseline-prior calibration the variances are increased by 10 %. The mean and standard errors of the real relative transform are shown in Table I. The mean rotation is parametrized as roll-pitch-yaw angles $\boldsymbol{\mu_\theta}$.

	$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{z}$	unit
$\boldsymbol{\mu_\theta}$	-0.51	-1.8e-6	-3.3e-4	deg
$\boldsymbol{\sigma_{\delta\theta}}$	1.9	7.1e-3	1.3e-2	deg
$\boldsymbol{\mu_p}$	-0.0087	-3000.0	13.4	mm
$\boldsymbol{\sigma_{\Delta p}}$	0.27	3.0	50.5	mm

TABLE I: True nominal baseline transform used in the synthetic dataset.

V. Simulation Results

A. Evaluation of Estimated Camera Poses

In this section, we compare the camera pose estimates obtained from a) using the fixed calibration only, b) using joint prior-visual estimates only (with five-point Nister algorithm), c) using the prior calibration and IMU measurements only and finally e) using the full pipeline (prior, vision, and IMU). Note that the EKF fused with IMU measurements only (without baseline prior) diverges quickly and is not shown in the following. Fig. 6 visualizes the individual normalized root mean squared errors (RMSE) with respect to the ground truth poses. There are two main observations: Firstly, incorporating the IMU decreases the errors

[2] Perfect pinhole cameras without distortion and a resolution of 720 × 480 pixels (0.36 MP).

significantly. Secondly, since the RMSE of the relative PNP solution is relatively high, fusing them into the EKF reduces the overall RMSE only slightly. As can be seen in Table II, the rotations around the $\boldsymbol{y}$- (pitch) and $\boldsymbol{z}$-axis (yaw) as well as the motions along $\boldsymbol{x}$- and $\boldsymbol{y}$-axis have similarly small deviations as our error estimate and hence do not significantly affect correspondence matching or depth estimation. For the rotation around $\boldsymbol{x}$-axis we see an improvement of over one order of magnitude (from around $2°$ to $0.08°$) and in movement along $\boldsymbol{z}$-axis a three-fold improvement. Rotational disturbances affect the matching process much stronger than translational disturbances, especially since we deal with objects far away. Given our large baseline and long distances, the translational deviations on $\boldsymbol{y}$- and $\boldsymbol{z}$-axis do not have a significant impact. For instance, the disturbance of around 15 mm on $\boldsymbol{z}$-axis, which corresponds to the optical $\boldsymbol{y}$ axis, does not result in any pixel disturbance for points that are 13 m or further away. Comparing to the fixed baseline, which suffers both from the periodic, regular disturbances, and the low-frequent random disturbances, our proposed approach performs significantly better as further shown in Fig. 7. The first peak at 4 s in positional error is due to the initialization of the EKF. Relative velocity, angular velocities, and linear accelerations are all initialized to zero. Position and orientation are initialized using the baseline prior. However since our trajectory starts mid-air, our initialization values for the EKF are incorrect, especially for the linear accelerations. This leads to relatively bad early estimates in position of up to about 5 s, until the acceleration values and thus velocity estimates converge. This behavior is less pronounced in orientation since the angular velocities are directly measured. The small peak at around 17 s can be explained by an exceptionally large random disturbance at this time step. Since our IMUs operate at only 100 Hz not all acceleration spikes can be captured correctly. This leads to wrong acceleration values similar to the wrong initialization case.

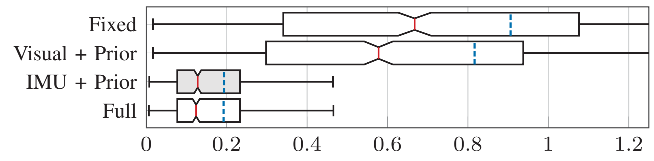

Fig. 6: Normalized RMSE of a fixed baseline assumption, fused estimate of visual pose estimation with the baseline prior, using IMU and baseline prior, and the full pipeline (visual, IMU, and baseline prior). The dashed line shows the mean.

	$\delta\tilde{\boldsymbol{\theta}}_x$	$\delta\tilde{\boldsymbol{\theta}}_y$	$\delta\tilde{\boldsymbol{\theta}}_z$	$\Delta\tilde{\boldsymbol{p}}_x$	$\Delta\tilde{\boldsymbol{p}}_y$	$\Delta\tilde{\boldsymbol{p}}_z$
Fixed	1.96	0.0071	0.0102	0.269	3.06	51.2
IMU + Prior	0.083	0.0070	0.0095	0.375	2.83	14.7

TABLE II: Rotational RMSE in deg and translational RMSE in mm.

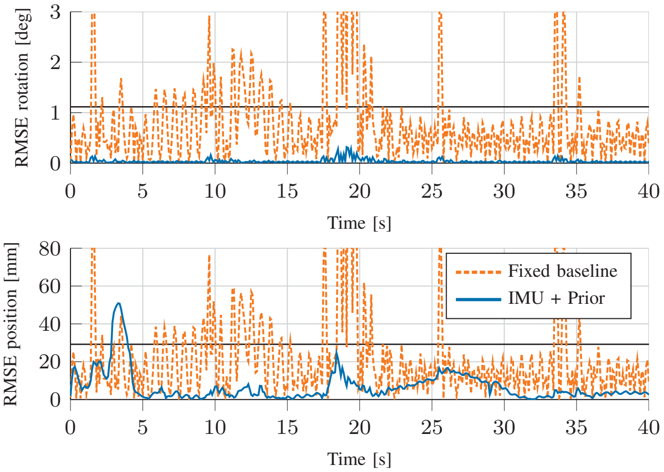

Fig. 7: RMS errors plotted over the first 40 s in orientation and position of estimated deviations based on the fixed baseline assumption or using the proposed pipeline. The random disturbance force results in high peaks every 8 s while the periodic disturbance results in high-frequent errors. The horizontal black line represents the standard error of the disturbance.

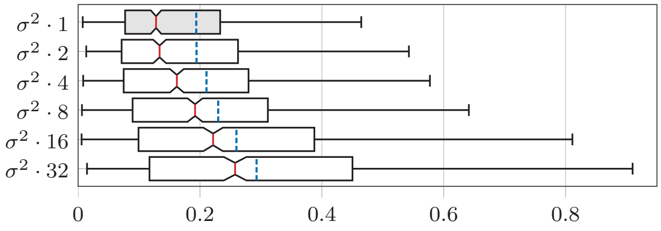

Fig. 8: Normalized RMSE versus noise-level expressed as multiples of the noise variances of the *ADIS 16448* IMU. Note the logarithmic y-axis.

In a next step the sensitivity of our approach with respect to IMU noise is analyzed. Fig. 8 plots the obtained normalized RMS errors for the proposed pipeline. We express the noise levels as multiples of the noise of the *ADIS 16448* IMU. Clearly, increasing noise affects the estimates, but for up to about an eight-fold increase in noise the estimates still stay relatively close. From this experiment, we conclude that even in real-world scenarios our approach is expected to yield good results.

B. Evaluation of Depth-Maps

Finally, we wish to support our claim that the resulting relative pose estimates are of such a high quality that accurate depth maps can be obtained. For comparison, three depth maps are generated at each timestep: one based on the ground-truth transform $\boldsymbol{T}$, one based on our estimate $\hat{\boldsymbol{T}}$, and one based on a fixed baseline transformation, denoted by $\boldsymbol{T}_\mu$. The disparities of dense correspondences in the rectified stereo images are computed using Stereo Block Matching (BM) [26]. Given the disparity d of a pixel in the rectified images, the depth of a pixel is given by $z = f \cdot b/d$ where f is the focal length and b the baseline of the rectified stereo pair. The obtained depth map image from transform $\boldsymbol{T}$ is

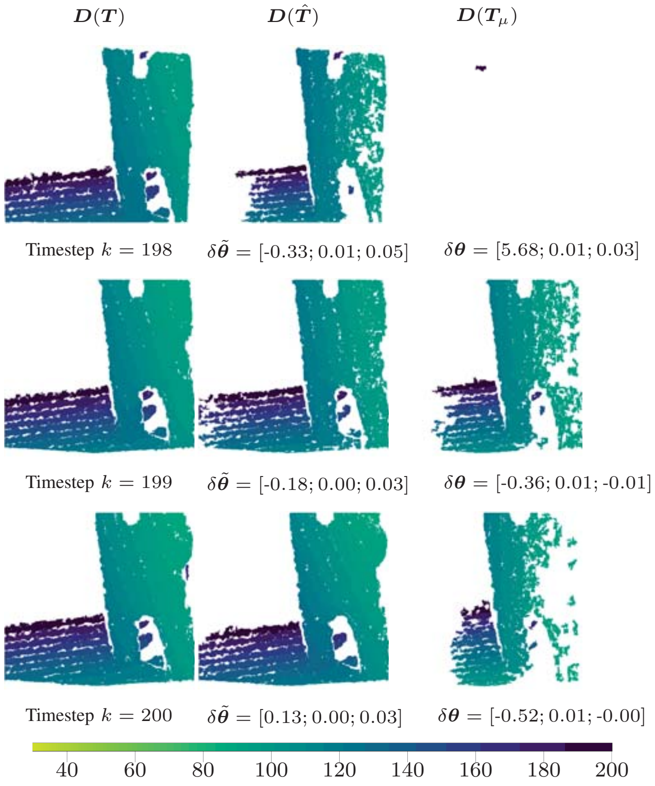

Fig. 9: Timesteps 198–200: Depth maps obtained with T, $\hat{T}$ and T_μ. The given colormap is truncated from the full range (cf. Fig. 5). Note that the colormap is intentionally not continuous but all depths that are mapped to one disparity value are colored uniformly.

denoted by $D(T)$. First, the performance of our approach is qualitatively demonstrated based on three characteristic frames in Fig. 9. For each frame color-coded depth maps are shown and denoted by $D(T)$, $D(\hat{T})$ and $D(T_\mu)$, i.e. depth maps obtained from the real, the estimated respectively the mean baseline transform. Let us first focus on what happens with the depth map based on a fixed baseline $D(T_\mu)$: In frame 198, it is evident that the stereo correspondence search failed almost completely. This demonstrates exemplary how an error in roll angle θ_x affects the matching process in such a way that no correspondences are found. Due to an erroneous rectification, the epipolar lines, along which the correspondences are searched, are flawed. In timestep 199, however, one can see that $D(T_\mu)$ resembles the ground truth depth map. This is due to the fact that the relative transform periodically "traverses through the origin", i.e. corresponds to the mean baseline. In these cases, T_μ is close to T and thus errors are small. In the last frame, at timestep 200, this effect already is seen to be diminished. The depth maps obtained using the poses estimated by our proposed framework show only small reconstruction errors compared to the ground-truth (at timestep 200). In frame 198, the disturbance is not estimated perfectly, evident by a remaining error in roll of almost 0.4°. This affects mostly pixels lying towards the left and right of the image due to the correspondence search along epipolar lines. This can also be observed in depth map $D(T_\mu)$ in timestep 199. In contrast, for pixels close to the center line a slight misalignment does not result in a matching failure.

There are two different errors present in the generated depth maps. We consider pixels that are valid in the ground-truth depth map. For each valid pixel in $\Delta(T)$, two error cases can occur: First, this pixel might be invalid in the estimated depth map. Second, the pixel might be valid, but off by a certain value. In order to quantify these errors, the error functions $\Delta_\#(D, \hat{D})$, $\Delta_z(D, \hat{D})$ are defined. $\Delta_\#(D, \hat{D})$ is defined to be the fraction of additionally invalid pixels in the estimate $\hat{D}$ compared to the total number of valid pixels (in D), i.e. capturing the *completeness* of the depth map estimate. $\Delta_z(D, \hat{D})$ is defined to be the RMS error in depth for all valid pixels thus a measure of *accuracy* for the estimates. Note that pixels in $\hat{D}$ are ignored even if they are valid in $\hat{D}$ but not valid in D. Defining the indicator function $v(D_{i,j}) = [D_{i,j}\text{is valid}]$ and $\delta(D_{i,j}, \hat{D}_{i,j}) = D_{i,j} - \hat{D}_{i,j}$ if both pixels are valid and else 0, the formula to calculate these two errors is given by:

$$\Delta_\#(D, \hat{D}) = \left(\frac{\sum_{i,j} [v(D_{i,j}) \wedge \neg v(\hat{D}_{i,j})]}{\sum_{i,j} v(D_{i,j})} \right)^{1/2} \tag{28}$$

$$\Delta_z(D, \hat{D}) = \left(\sum_{i,j} \delta^2(D_{i,j} - \hat{D}_{i,j}) \right)^{1/2} . \tag{29}$$

These two error functions are evaluated at each timestep and the result is aggregated in form of box plots as shown in Fig. 10. The improvement over the fixed baseline assumption is drastic. From a mean of over 16 m average depth error, the proposed framework reduces the depth error to a mere 2.2 m. For comparison, the average scene depth over all valid ground-truth pixels is 122 m. In completeness, a similar improvement is demonstrated: On average, almost 74 % of all pixels that are valid in the ground-truth depth map D are invalid in D_μ. For depth maps obtained using the proposed framework, however, this number is reduced to an average of 11.5 %. As stated above and observable in Fig. 9, these invalid pixels occur mostly towards the edge of the images.

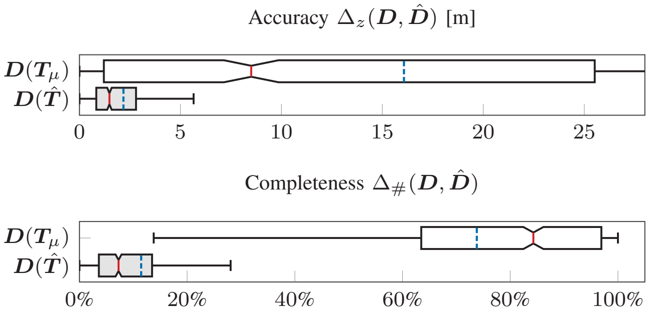

Fig. 10: Comparing the errors in accuracy (top) and completeness (bottom).

However, as our goal is to detect and avoid obstacles, the important region of interest lies in the middle of the image.

VI. CONCLUSIONS

In this paper, we present the theory to accurately estimate the time-varying baseline transformation of a flexible wide-baseline stereo pair in order to generate high-quality depth maps. The light-weight nature of the proposed EKF, as well as the extensive analysis of the relative pose estimates and depth map led to promising results. In particular, including the measurements of the two rigidly attached IMUs resulted in a significant reduction of the baseline transformation error. While the two IMUs fused with an EKF precisely estimate the deformation of the relative baseline transformation in the short-term, the Gaussian prior for the wing model ensures a constrained estimation problem in the long-term. The incorporation of more sophisticated wing models could further improve the result of the relative baseline estimate. For instance, a long short-term memory (LSTM) network can learn the periodic wing deflection and predict the future relative poses as a time series. Extensive experiments need to show if a similar level of robustness, precision, and accuracy is also achievable in the real-world by handling camera and IMU time synchronization and IMU bias estimation.

ACKNOWLEDGMENT

The research leading to these results has received funding from the *Federal office armasuisse Science and Technology* under project n°050-45. Furthermore, we wish to thank Lucas Teixeira (Vision for Robotics Lab, ETH Zurich) for providing scripts that bridge the gap between Blender and Gazebo.

REFERENCES

[1] G. Klein and D. Murray, “Parallel Tracking and Mapping for Small AR Workspaces,” in *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2007.

[2] A. Fusiello, E. Trucco, and A. Verri, “A compact algorithm for rectification of stereo pairs,” *Machine Vision and Applications*, vol. 12, pp. 16–22, Jul 2000.

[3] M. Pollefeys, R. Koch, and L. V. Gool, “A simple and efficient rectification method for general motion,” in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 1, pp. 496–501 vol.1, 1999.

[4] M. Warren, P. Corke, and B. Upcroft, “Long-range stereo visual odometry for extended altitude flight of unmanned aerial vehicles,” *The International Journal of Robotics Research*, vol. 35, no. 4, pp. 381–403, 2016.

[5] P. Oettershagen, A. Melzer, T. Mantel, K. Rudin, T. Stastny, B. Wawrzacz, T. Hinzmann, S. Leutenegger, K. Alexis, and R. Siegwart, “Design of small hand-launched solar-powered UAVs: From concept study to a multi-day world endurance record flight,” *Journal of Field Robotics*, 2017.

[6] T. Dang, C. Hoffmann, and C. Stiller, “Continuous stereo self-calibration by camera parameter tracking,” *IEEE Trans. Image Processing*, vol. 18, no. 7, pp. 1536–1550, 2009.

[7] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, second ed., 2004.

[8] M. Achtelik, S. Weiss, M. Chli, F. Dellaert, and R. Siegwart, “Collaborative Stereo,” in *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems*, 2011.

[9] M. Achtelik, *Advanced closed loop visual navigation for micro aerial vehicles*. PhD thesis, ETH Zurich, 2014.

[10] P. Furgale, J. Rehder, and R. Siegwart, “Unified temporal and spatial calibration for multi-sensor systems,” in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 1280–1286, IEEE, 2013.

[11] M. Warren, D. McKinnon, and B. Upcroft, “Online calibration of stereo rigs for long-term autonomy,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 3692–3698, IEEE, 2013.

[12] P. Hansen, H. Alismail, P. Rander, and B. Browning, “Online continuous stereo extrinsic parameter estimation,” in *Computer Vision and Pattern Recognition, 2012 IEEE Conference on*, pp. 1059–1066, IEEE, 2012.

[13] M. Warren, P. Corke, and B. Upcroft, “Long-range stereo visual odometry for extended altitude flight of unmanned aerial vehicles,” *The International Journal of Robotics Research*, vol. 35, no. 4, pp. 381–403, 2016.

[14] P. Lanier, N. Short, K. Kochersberger, and L. Abbott, “Modal-based camera correction for large pitch stereo imaging,” in *Structural Dynamics, Volume 3*, pp. 1225–1238, Springer, 2011.

[15] P. Lanier, “Stereovision Correction Using Modal Analysis,” Master’s thesis, Virginia Polytechnic Institute and State University, 2010.

[16] N. J. Short, “3-D Point Cloud Generation from Rigid and Flexible Stereo Vision Systems,” Master’s thesis, Virginia Tech, 2009.

[17] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[18] P. S. Maybeck, *Stochastic models, estimation, and control*, vol. 1. Academic Press, 1979.

[19] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” *Computer vision–ECCV 2006*, pp. 404–417, 2006.

[20] M. Muja and D. G. Lowe, “Fast approximate nearest neighbors with automatic algorithm configuration.,” *VISAPP (1)*, vol. 2, no. 331-340, p. 2, 2009.

[21] D. Nistér, “An efficient solution to the five-point relative pose problem,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 6, pp. 756–770, 2004.

[22] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[23] L. Kneip and P. Furgale, “OpenGV: A unified and generalized approach to real-time calibrated geometric vision,” in *Robotics and Automation, 2014 IEEE International Conference on*, pp. 1–8, IEEE, 2014.

[24] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, “Rotors – A modular Gazebo MAV simulator framework,” in *Robot Operating System (ROS)*, pp. 595–625, Springer, 2016.

[25] I. Analog Devices, “ADIS16448 datasheet,” 2017.

[26] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal: Software Tools for the Professional Programmer*, vol. 25, no. 11, pp. 120–123, 2000.