Depth Enhanced Visual-Inertial Odometry Based on Multi-State Constraint Kalman Filter

Fumin Pang, Zichong Chen, Li Pu, and Tianmiao Wang, *M*ember, *IEEE*

***Abstract*— There have been increasing demands for developing robotic system combining camera and inertial measurement unit in navigation task, due to their low-cost, lightweight and complementary properties. In this paper, we present a Visual Inertial Odometry (VIO) system which can utilize sparse depth to estimate 6D pose in GPS-denied and unstructured environments. The system is based on Multi-State Constraint Kalman Filter (MSCKF), which benefits from low computation load when compared to optimization-based method, especially on resource-constrained platform. In this paper, we enhance the features with depth information to form 3D landmark position measurements in space, which reduces uncertainty of position estimate. And we derivate measurement model to access compatibility with both 2D and 3D measurements. In experiments, we evaluate the performance of the system in different in-flight scenarios, both cluttered room and industry environment. The results suggest that the estimator is consistent, substantially improves the accuracy compared with original monocular-based MSKCF and achieves competitive accuracy with other research.**

I. INTRODUCTION

Accurate 6D pose estimate in unknown environment from a set of sensor measurements is one of the critical problems in robotics navigation task. Combination of complementary information from visual and inertial sensors is ubiquitous in mobile robot application with the requirement of high dynamic operation. Inertial sensors readings can be used to compute relatively accurate 6D pose in short period and give a real-metric estimate for absolute scale[1]. While visual sensor with the ability to provide rich texture of environment and bearing measurements for salient landmarks make it an efficient exteroceptive sensor to correct the motion and structure prediction. Each visual feature can always be tracked by a camera from a sequence of consecutive poses, which provide multiple constraint of camera motion and landmarks. VIO can be considered a subproblem of Simultaneous Localization and Mapping (SLAM) ,while it pays more focus on efficient 6D pose estimate. This method has achieved success in navigating Micro Aerial Vehicles(MAV) and cars.

To date, the majority of algorithms proposed for real-time VIO can be classified into two categories: extended Kalman filter-based methods and methods utilizing optimization approach. Filter-based approaches are the earlier ones used to solve SLAM and VIO problems. Davison et al. proposed one of the first real-time 3D monocular SLAM framework based on EKF in computer vision[2]. Based on this work, Roussillon et al. used IMU as a interoceptive to propagate position, orientation and velocity of platform in high dynamic environment[3]. Jones and Soatto[4] built a EKF using Lie derivatives and studied observability in visual-inertial system. Their results showed that the IMU biases, 3D velocity, absolute roll and pitch angles and IMU-camera transformation are observable. Some Similar conclusions were also drawn by Kelly et al.[5].

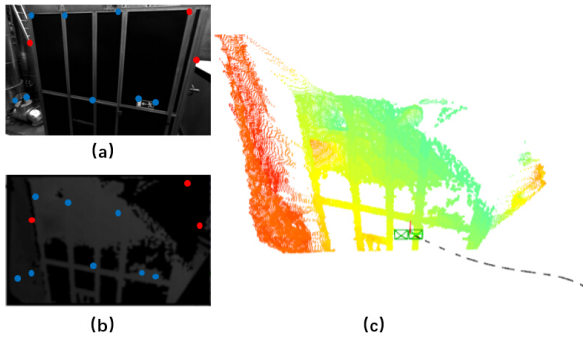

Fig. 1: (a) Features tracked at an image frame. The blue dots represent features whose depth comes from stereo matches. and the red dots represent features without depth. The proposed method uses both types of features in estimating 6D pose. (b) A disparity map from stereo a corresponding to (a), which shows depth information is only available in the vicinity of the camera. (c) A local point cloud built from (b), colors encode distance. This local map shows the sparse nature for space landmarks.

Meanwhile, optimization-based methods generally attain higher accuracy, as they re-linearize at each iteration to better deal with their nonlinear measurement models. Leutenegger et al. describe a tightly coupled approach in which the robot poses and sparse 3D landmarks are estimated through minimize a joint optimization problem using inertial error terms as well as the reprojection error of the landmarks in stereo image[6]. However, these methods always implement bundle adjustment in a sliding window of states, using multiple iterations to minimize cost function, which results in increased computational cost. Thus, filter-based methods might be a better choice for long-time running tasks, especially in resource-constrained systems, such as micro aerial vehicles and Augmented Reality (AR) devices[7].

Multi-State Constraint Kalman Filter improve efficiency of filter-based method further[8]. Instead of including the 3D feature position in the state vector of the EKF, MSCKF

Fumin Pang, Zichong Chen and Li Pu are with Segway Robotics Inc. Beijing, China. E-mail: {fumin.pang, zichong.chen, li.pu}@segwayrobotics.com

Tianmiao Wang is with the School of Mechanical Engineering and Automation, Beihang University, Beijing, China. E-mail: itm@buaa.edu.cn

maintains a sliding window of historical poses in state vector. The landmarks positions are triangulated from multiple observations form different camera poses when they slip away from field of view. Because the number of poses in sliding window is usually much less than the number of features, the computational complexity is reduced. Li et al. correct the observation properties of original MSCKF using First-Estimate Jacobian (FEJ)[9].

The sparsity of depth mainly lies on two aspects. First, with limitation of the sensors, scenarios with large depth variation leave large area in the images where depth is unavailable spatially. Second, as in MSCKF paradigm, there need to exist multiple measurements for one feature from a series of different robot poses before its 3D position is triangulated. However, only a subset of measurements for one feature has corresponding depth information temporally in general. The inevitable sparsity makes people unable to use ICP-like method in large scale environment to estimate poses.

In this paper, we propose a method that can utilize sparse depth information along with the imagery based on MSCKF, inspired by [10] and [11]. We utilize 2D visual feature and depth to form a 3D measurement of a landmark. Measurement model is adapted to be compatible with both 2D and 3D measurements . In the update step of this depth enhanced MSCKF, reprojection measurement errors for landmarks where depth is unavailable and 3D position measurement errors are jointed to correct the pose estimate. The proposed method has been tested on real-world data. The results demonstrate a lower drift than mono-MSCKF. The attained accuracy is competitive with other research in this field.

In this work, we use stereo camera to obtain depth information. But it can be extended and adapted to various sensors which can provide bearing and range infomation.

II. ESTIMATOR DESCRIPTION

Four different coordinate frames are used throughout the paper: we affix camera-IMU rig body frame $\{B\}$ to IMU, to track the 6D motion with respect to a global coordinate frame, $\{G\}$. The camera coordinate frame, $CAM0$, is $\{C^0\}$. In this paper, it is $CAM0$ where features are tracked. An additional camera , $CAM1$, is added to form a stereo rig to estimate depth by potentially matching with features in $CAM0$. Its coordinate frame is $\{C^1\}$.

A. Overall Filter Structure and State Parametrization

The full state representation can be partitioned into two parts according to MSCKF paradigm. The first is the evolving current body state. As we affix body frame $\{B\}$ to IMU, we use $\mathbf{x}_I$ to present body state. The body state at time *k* is a 17-dimensional vector as follow:

$$\mathbf{x}_{I,k} := \begin{bmatrix} {}^B_G\bar{\mathbf{q}}_k^T & {}^G\mathbf{p}_{B,k}^T & {}^G\mathbf{v}_{B,k}^T & \mathbf{b}_{g,k}^T & \mathbf{b}_{a,k}^T & t_{d,k} \end{bmatrix}^T \quad (1)$$

In this world-centric presentation , ${}^B_G\bar{\mathbf{q}}_k$ is the unit quaternion representing the rotation which rotate vectors from the global frame $\{G\}$ to the body frame $\{B\}$. In this paper, all quaternions follow JPL convention [12]. ${}^G\mathbf{p}_{B,k}$ is the vector from the origin of $\{G\}$ to the origin of $\{B\}$ expressed in $\{G\}$ (i.e., the position of body in the global frame). ${}^G\mathbf{v}_{B,k}$ is the vector representing original velocity of frame $\{B\}$ expressed in $\{G\}$. $\mathbf{b}_{g,k}$ is the bias on the gyro measurements $\boldsymbol{\omega}_m$, $\mathbf{b}_{a,k}$ is the bias on the velocity measurements $\mathbf{a}_m$. t_d is a scalar modelling the unknown $time\ offset$ between IMU and cameras ,which will help to improve accuracy [13]. In this paper, we assume CAM0 and the depth sensor are synchronized temporally.

The second part of the full state is a sliding window of *N* past body states, in which active feature tracks were visible. The *i-th* body state, $i = 0...N-1$, including 6D pose and velocity, is a 10-dimensional vector. Velocity state involves in t_d estimate.

$$\mathbf{x}_{B_i,k} := \begin{bmatrix} {}^{B_i}_G\bar{\mathbf{q}}_k^T & {}^B\mathbf{p}_{B_i,k}^T & {}^G\mathbf{v}_{B_i,k}^T \end{bmatrix}^T \quad (2)$$

Combining this two parts, the full state of MSCKF is a (17 + 10* *N*) vector consisting of current body state estimate and *N* last body states at time *k*.

In MSCKF, $Error\ State\ Kalman\ Filter$(ESKF)[12] is used to avoiding singularity brought by quaternions which use 4 dimensions to describe 3 degrees of freedom. While the error $(\tilde{x})$ between the true value (x) and the estimated value $(\hat{x})$ for position, velocity , bias and t_d can be defined as $\tilde{x} = x - \bar{x}$ trivially, the error for quaternion is defined as:

$$\boldsymbol{\delta\bar{q}} := \hat{\bar{\boldsymbol{q}}}^{-1} \otimes \bar{\boldsymbol{q}} \approx \begin{bmatrix} \frac{1}{2}\boldsymbol{\delta\theta}^T & 1 \end{bmatrix}^T \quad (3)$$

Using $\boldsymbol{\delta\theta}$ to represent orientations in the Kalman filter reduces their dimensionality to 3. As a result, the *error state* of the estimator at time *k*:

$$\tilde{\mathbf{x}}_k := \begin{bmatrix} \tilde{\mathbf{x}}_{I,k}^T & \tilde{\mathbf{x}}_{B_0,k}^T & \dots & \tilde{\mathbf{x}}_{B_{N-1},k}^T \end{bmatrix}^T \quad (4)$$

where

$$\tilde{\mathbf{x}}_{I,k} := \begin{bmatrix} {}^G_B\boldsymbol{\delta\theta}_I^T & {}^G\tilde{\mathbf{p}}_{B,k}^T & {}^G\tilde{\mathbf{v}}_{B,k}^T & \tilde{\mathbf{b}}_{g,k}^T & \tilde{\mathbf{b}}_{a,k}^T & \tilde{b}_{a,k} \end{bmatrix}^T \quad (5)$$

$$\tilde{\mathbf{x}}_{B_i,k} := \begin{bmatrix} {}^G_{B_i}\boldsymbol{\delta\theta}_I^T & {}^G\tilde{\mathbf{p}}_{B_i,k}^T & {}^G\tilde{\mathbf{v}}_{B_i,k}^T \end{bmatrix}^T \quad (6)$$

The full error state has (16 + 9 *N*) dimensions. Accordingly, the MSCKF error state covariance $\boldsymbol{P}$ is a (16 + 9 *N*) × (16 + 9 *N*) matrix.

B. Filter Propagation and Augment

Every time an IMU measurement is received, it is used to propagate the IMU state. As mentioned before, IMU measurement provide rotational velocity $\boldsymbol{\omega}_m$ and $\boldsymbol{a}_m$, described as below equations:

$$\boldsymbol{\omega}_m = {}^G\boldsymbol{\omega} + \boldsymbol{b}_g + \boldsymbol{n}_g \quad (7)$$

$$\boldsymbol{a}_m = {}^B_G\mathbf{R}\left({}^G\boldsymbol{a} - {}^G\boldsymbol{g}\right) + \boldsymbol{b}_a + \boldsymbol{n}_a \quad (8)$$

where ${}^{B}_{G}\mathbf{R}$ is the rotation matrix corresponding to ${}^{I}_{G}\mathbf{q}$. ${}^{G}\boldsymbol{g}$ is the gravitational acceleration. $\boldsymbol{n}_g$ and $\boldsymbol{n}_a$ are zero-mean white Gaussian noise vectors. We follow the approach described in [9] and [14] to propagate filter state in discrete time.

Linearized continuous-time model of current state error state, $\tilde{\mathbf{x}}_I$, can be drawn from (9):

$$\dot{\tilde{\mathbf{x}}}_I = \boldsymbol{F}\tilde{\mathbf{x}}_I + \boldsymbol{G}\boldsymbol{n}_I \tag{9}$$

where $\boldsymbol{F}$ is the continuous-time error-state transition matrix. $\boldsymbol{G}$ is the Jacobian of current body error state with respect to the noise vector $\boldsymbol{n}_I = [\boldsymbol{n}_g^T \ \boldsymbol{n}_a^T \ \boldsymbol{n}_{wg}^T \ \boldsymbol{n}_{wa}^T]^T$.

State covariance matrix alse propagates. And upon recording a new image, a copy of current body state is appended to the state vector, and the covariance matrix of the MSCKF is augmented accordingly [8].

C. Mean and Covariance of 3D measurement

In this paper, visual measures from stereo camera form a 3D measurement of a landmark when the depth is available. Stereo measurement can efficiently reduce the uncertainty (covariance) of 3D position estimate . We consider one feature,$\boldsymbol{f}$, which is observed by stereo rig at timestep $\boldsymbol{i}$. We will derivate its mean and covariance of 3D position measurement.

First, in frame $\{C^0\}$, landmark position ${}^{C_i^0}\boldsymbol{p}_f = [{}^{C_i^0}X \ {}^{C_i^0}Y \ {}^{C_i^0}Z]^T$ is calculated by stereo triangulation. We present its 3D measurment using a pixel bearing vector $[\alpha \ \beta]^T$ and a inverse depth scalar ρ :

$${}^{C_i^0}\boldsymbol{z}_3 = \begin{bmatrix} {}^{C_i^0}\alpha \\ {}^{C_i^0}\beta \\ {}^{C_i^0}\rho \end{bmatrix} = \boldsymbol{\pi}_3 \begin{pmatrix} {}^{C_i^0}X \\ {}^{C_i^0}Y \\ {}^{C_i^0}Z \end{pmatrix} = \begin{bmatrix} \frac{{}^{C_i^0}X}{{}^{C_i^0}Z} \\ \frac{{}^{C_i^0}Y}{{}^{C_i^0}Z} \\ \frac{1}{{}^{C_i^0}Z} \end{bmatrix} \tag{10}$$

Similarly, in $\{C_1\}$, the 3D measurement ${}^{C_i^1}\boldsymbol{z}_3$ of ${}^{C^0}\boldsymbol{p}_f$ is

$${}^{C_i^1}\boldsymbol{z}_3 = \begin{bmatrix} {}^{C_i^1}\alpha \\ {}^{C_i^1}\beta \\ {}^{C_i^1}\rho \end{bmatrix} = \boldsymbol{\pi}_3 \left({}^{C_1}_{C^0}\boldsymbol{R} \begin{pmatrix} {}^{C_i^0}X \\ {}^{C_i^0}Y \\ {}^{C_i^0}Z \end{pmatrix} + {}^{C^1}\boldsymbol{p}_{C^0} \right) \tag{11}$$

As above, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ are given by pixel coordinates in image with measurement noise $\boldsymbol{diag}(\sigma_\alpha^2, \sigma_\beta^2)$. ρ can be calculated by inverse depth. Following paper, we assume $\rho \sim \mathcal{N}\left(\rho, \sigma_\rho^2\right)$, where σ_ρ is set manually.

From (16)-(17), the covariance of Gaussian distribution in Euclidean space of ${}^{C^0}\boldsymbol{p}_f$ can be fused by the two measurements uncertainties:

$$\boldsymbol{\Sigma}_{p^j} = \boldsymbol{J}_{p^j}\boldsymbol{diag}\left(\sigma_{\alpha^j}^2, \sigma_{\beta^j}^2, \sigma_{\rho^j}^2\right){\boldsymbol{J}_{p^j}}^T \tag{12}$$

where $j = 0, 1$ and $\boldsymbol{J}_{p^j} = \frac{\partial^{C^0}\boldsymbol{p}_f}{\partial^{C_i^j}\boldsymbol{z}_3}$.

Given two 3D measurement covariances $\boldsymbol{\Sigma}_{p^0}$ and $\boldsymbol{\Sigma}_{p^1}$, we can fuse the position covariance using a Bayesian update step [15]:

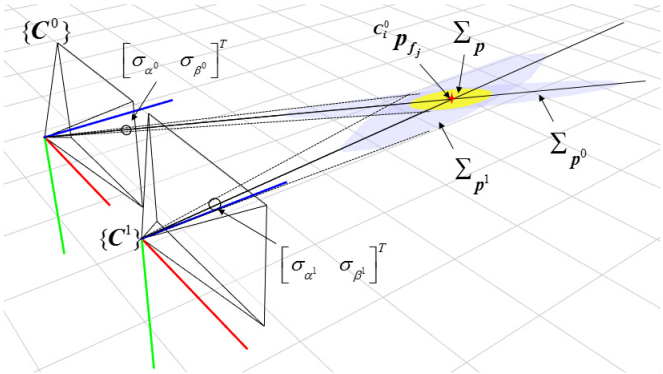

Fig. 2: 3D measurement can be achieved by fusing two visual measurements from stereo camera. Mean of 3D measurement can be calculate from triangulation, and the covariance Σ_p comes from merging two 3D Gaussian distributions, Σ_{p_0} and Σ_{p_1}. This Bayesian fusion results in smaller 3D uncertainty than either Σ_{p_0} or Σ_{p_1}, visualized as a smaller ellipsoid. The pixel bearing vector uncertainty involve visual feature uncertainty and scale.

$${}^{C_i}\boldsymbol{\Sigma}_p = \boldsymbol{\Sigma}_{p^0}\left(\boldsymbol{\Sigma}_{p^0} + \boldsymbol{\Sigma}_{p^1}\right)^{-1}\boldsymbol{\Sigma}_{p^1} \tag{13}$$

Fusing two 3-dimensional Gaussian distributions reduces the uncertainty of the distribution (see Fig.2). Finally, the 3D measurement mean and covariance can be derivated:

$${}^{C_i}\boldsymbol{z}_3 = \begin{bmatrix} {}^{C_i}\alpha \\ {}^{C_i}\beta \\ {}^{C_i}\rho \end{bmatrix} \qquad {}^{C_i}\boldsymbol{\Sigma}_3 = \boldsymbol{J}_{\boldsymbol{\pi}_3}{}^{C_i}\boldsymbol{\Sigma}_p{\boldsymbol{J}_{\boldsymbol{\pi}_3}}^T \tag{14}$$

where $\boldsymbol{J}_{\boldsymbol{\pi}_3} = \frac{\partial^{C_{0,i}}\boldsymbol{z}_3}{\partial^{C_0}\boldsymbol{p}_f}$. The 3D covariance in Euclidean space is transformed to bearing and inverse depth parameterization space. Here, for simplicity, C_i is used to present camera frame C_i^0 at timestep *i*. Thus, fusing the visual measures in stereo provides a more precise estimate on landmark position and reduces uncertainty by fusing the two Gaussian distributions, which will potentially improve performance of visual odometry accuracy. In addition, these 3D measurements make filter update more efficient compared to two separate updates in left and right image due to a lower-dimensional measurement jacobain.

D. Depth Enhanced Feature Position Estimation

In this work, once a tracked feature is lost in current frame, *CAM0*, the corresponding landmark position need to be estimated. Compared to original MSCKF which uses triangultion from a sliding window of 2D imagery measeurements, we utilize both 2D and 3D measures.

To describe the position estimate process and update model, we consider one landmark,${}^{G}\boldsymbol{p}_{f_j}$, which is observed at a set of poses, $\left\{\boldsymbol{T}_i = \left\{{}^{C_i}_{G}\boldsymbol{R}, {}^{G}\boldsymbol{p}_{C_i}\right\} \in SE(3)\right\}$ with $i = 1, 2, 3, 4$ (see Fig.3). In C_1 and C_3, depth is available.We initialize the optimization by estimating the position of feature f_j in camera frame C_1 using a linear least-squares

method [16]. We can express the feature position in camera frame C_i in terms of its position in camera frame C_1 as

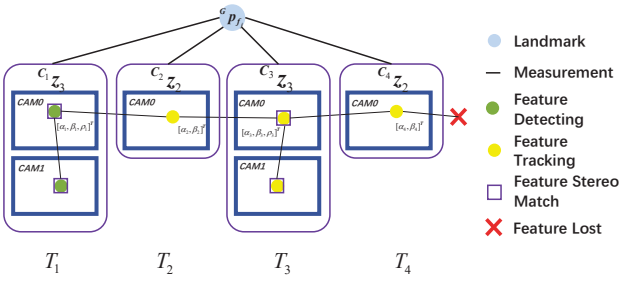

Fig. 3: An example indicates one feature f_j observed by camera at 4 pose. At T_1 and T_3, depth is available which generates 3D measurements, while at T_2 and T_4, no matches can be found in $CAM1$ resulting in two 2D measurements. Landmark position corresponding to f_j is estimated by minimizing either 2d or 3D measurement errors through least-square method.

$$ {}^{C_i}\hat{\boldsymbol{p}}_{f_j} = {}^{C_i}_{C_1}\hat{\boldsymbol{R}}\,{}^{C_1}\hat{\boldsymbol{p}}_{f_j} + {}^{C_i}\hat{\boldsymbol{p}}_{C_1} \tag{15} $$

we can write the 2D imagery measurement error as

$$ \boldsymbol{e}_{2,i}({}^{C_1}\hat{\boldsymbol{p}}_{f_j}) = {}^{C_i}\boldsymbol{z}_{2,i} - \boldsymbol{\pi}_2\left({}^{C_i}_{C_1}\hat{\boldsymbol{R}}\,{}^{C_1}\hat{\boldsymbol{p}}_{f_j} + {}^{C_i}\hat{\boldsymbol{p}}_{C_1}\right) \tag{16} $$

where ${}^{C_i}\boldsymbol{z}_{2,i}$ is the pixel bearing vector given by feature coordinate and

$$ \boldsymbol{\pi}_2(\boldsymbol{h}) = \begin{bmatrix} \frac{h(1)}{h(3)} \\ \frac{h(2)}{h(3)} \end{bmatrix} \tag{17} $$

and 3D measurement error as

$$ \boldsymbol{e}_{3,i}({}^{C_1}\hat{\boldsymbol{p}}_{f_j}) = {}^{C_i}\boldsymbol{z}_{3,i} - \boldsymbol{\pi}_3\left({}^{C_i}_{C_1}\hat{\boldsymbol{R}}\,{}^{C_1}\hat{\boldsymbol{p}}_{f_j} + {}^{C_i}\hat{\boldsymbol{p}}_{C_1}\right) \tag{18} $$

By stacking the measurements together, the least-squares system then becomes

$$ \left(\boldsymbol{J}^T\boldsymbol{W}^{-1}\boldsymbol{J}\right)\boldsymbol{\delta}^{C_1}\boldsymbol{p}^{\star}_{f_j} = -\boldsymbol{J}^T\boldsymbol{W}^{-1}\boldsymbol{e}\left({}^{C_1}\hat{\boldsymbol{p}}_{f_j}\right) \tag{19} $$

where $\boldsymbol{e}\left({}^{C_1}\hat{\boldsymbol{p}}_{f_j}\right)$ is stacked vector of four measurement errors and $\boldsymbol{J}$ is stacked matrix of four Jacobians of measurement error with respect to ${}^{C_1}\hat{\boldsymbol{p}}_{f_j}$ and

$$ \boldsymbol{W} = \boldsymbol{diag}\left\{{}^{C_1}\boldsymbol{\Sigma}_3 \;\; {}^{C_2}\boldsymbol{\Sigma}_2 \;\; {}^{C_3}\boldsymbol{\Sigma}_3 \;\; {}^{C_4}\boldsymbol{\Sigma}_2\right\} \tag{20} $$

with ${}^{C_i}\boldsymbol{\Sigma}_2 = \boldsymbol{diag}\left\{\sigma^2_{\alpha_i}, \sigma^2_{\beta_i}\right\}$. ${}^{C_i}\boldsymbol{\Sigma}_3$ is calculated in previous section.

In practice, in order to improve numerical stability, inverse depth parameterization [17] is used. After iterative optimization, we get ${}^{C_1}\hat{\boldsymbol{p}}_{f_j}$ and further ${}^{G}\hat{\boldsymbol{p}}_{f_j}$. A simulation on how different quantities of 3D measurements in sliding window effects accuracy of position estimate is showed in Fig.4. The result shows depth information can observably benefits the estimation, especially along z-axis in local coordinate .

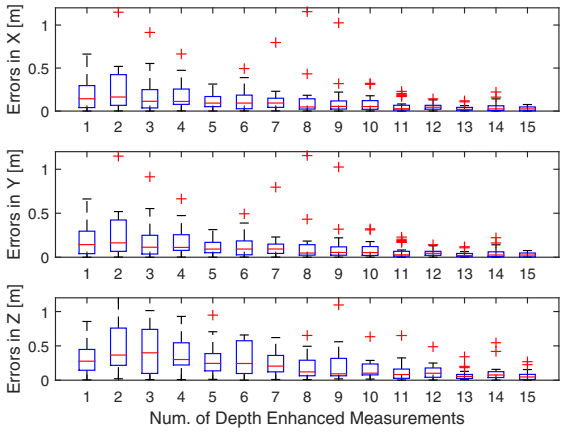

Fig. 4: Boxplot on simulation result about of 3-axis errors of $\boldsymbol{p}_{f_j}$ presented in C_1, in which different amount of 3D measurement is applied to estimate landmark position . We generate 50 features within a distance of 1-5 meters in front of cameras, and the sliding window size is 15. All feature positions are estimated with 3D measurement amounts varying from 1 to 15, respectively. The results indicate more 3D measurements reduce the measurements error, especially along z-direction in local coordinate.

E. Depth Enhanced Filter Update

Once the landmark position ${}^{G}\hat{\boldsymbol{p}}_{f_j}$ is estimated, we can calculated the measurement residuals for either 2D measurements or 3D measurements to update the system.

$$ \boldsymbol{r}_i^{(j)} = {}^{C_i}\boldsymbol{z}_i^{(j)} - {}^{C_i}\hat{\boldsymbol{z}}_i^{(j)} \tag{21} $$

where for 2D measurement, $\boldsymbol{r}_i^{(j)}$ is 2×1 and

$$ {}^{C_i}\hat{\boldsymbol{z}}_{2,i}^{(j)} = \boldsymbol{\pi}_2\left({}^{C_i}_{G}\hat{\boldsymbol{R}}\left({}^{G}\hat{\boldsymbol{p}}_{f_j} - {}^{G}\hat{\boldsymbol{p}}_{C_i}\right)\right) \tag{22} $$

and for 3D measurement, $\boldsymbol{r}_i^{(j)}$ is 3×1 and

$$ {}^{C_i}\hat{\boldsymbol{z}}_{3,i}^{(j)} = \boldsymbol{\pi}_3\left({}^{C_i}_{G}\hat{\boldsymbol{R}}\left({}^{G}\hat{\boldsymbol{p}}_{f_j} - {}^{G}\hat{\boldsymbol{p}}_{C_i}\right)\right) \tag{23} $$

Linearizing about the estimates for the body pose and for the feature position, the residual of (24) can be approximated as:

$$ \boldsymbol{r}_i^{(j)} = \boldsymbol{H}^{(j)}_{\boldsymbol{x}_{\boldsymbol{B}_i}}\tilde{\boldsymbol{x}} + \boldsymbol{H}^{(j)}_{\boldsymbol{f}_i}{}^{G}\tilde{\boldsymbol{p}}_{f_j} + \boldsymbol{n}_i^{(j)} \tag{24} $$

where the matrices $\boldsymbol{H}^{(j)}_{\boldsymbol{x}_{\boldsymbol{B}_i}}$ and $\boldsymbol{H}^{(j)}_{\boldsymbol{f}_i}$ are the corresponding Jacobians of the measurement $\hat{\boldsymbol{z}}_i^{(j)}$ with respect to the state and the feature position, respectively, and ${}^{G}\tilde{\boldsymbol{p}}_{f_j}$ is the error in the position estimate of $\boldsymbol{f}_j$. $\boldsymbol{n}_i^{(j)}$ is the measurement noise with covariance either ${}^{C_i}\boldsymbol{\Sigma}_2$ or ${}^{C_i}\boldsymbol{\Sigma}_3$. $\boldsymbol{H}^{(j)}_{\boldsymbol{x}_{\boldsymbol{B}_i}}$ and $\boldsymbol{H}^{(j)}_{\boldsymbol{f}_i}$ are given by

$$ \boldsymbol{H}^{(j)}_{\boldsymbol{x}_{\boldsymbol{B}_i}} = \boldsymbol{J}^{(j)}_{\pi,i}\begin{bmatrix}\boldsymbol{0}_{3\times 15} & \boldsymbol{M}_{t_d,i} & \boldsymbol{0}_{3\times 9(i-1)} & \boldsymbol{M}_{B_i} & \boldsymbol{0}_{3\times 9(N-l)}\end{bmatrix} \tag{25} $$

$$\boldsymbol{H}_{\boldsymbol{f_i}}^{(\boldsymbol{j})} = \boldsymbol{J}_{\pi,i}^{(j)} {}_{B}^{C_0}\mathbf{R}\, {}_{G}^{B}\hat{\mathbf{R}}\left(t+\hat{t}_d\right) \tag{26}$$

where

$$\boldsymbol{M}_{t_d,i} = {}_{B}^{C_0}\mathbf{R}\left(\lfloor {}_{G}^{B}\hat{\mathbf{R}}\left({}^{G}\hat{\boldsymbol{p}}_{f_j} - {}^{G}\hat{\boldsymbol{p}}_{B_i}\right)\times\rfloor {}^{B}\boldsymbol{\omega} - {}_{G}^{B}\hat{\mathbf{R}}\, {}^{G}\hat{\boldsymbol{v}}_{B_i}\right) \tag{27}$$

$$\boldsymbol{M}_{B_i} = {}_{B}^{C_0}\mathbf{R}\, {}_{G}^{B}\hat{\mathbf{R}}\left(t+\hat{t}_d\right)\begin{bmatrix}\lfloor({}^{G}\boldsymbol{p}_{f_j} - {}^{G}\tilde{\boldsymbol{p}}_{B})\times\rfloor & -\boldsymbol{I}_3 & \mathbf{0}_3\end{bmatrix} \tag{28}$$

$\lfloor \boldsymbol{c}\times\rfloor$ is the skew symmetric matrix corresponding to vector $\boldsymbol{c}$. And for more details,the nonzero blocks in $\boldsymbol{H}_{\boldsymbol{x}_{B_i}}^{(\boldsymbol{j})}$ are the Jacobians with respect to body rotation and body position , respectively. $\boldsymbol{J}_{\pi,i}^{(j)}$ is the Jacobian of the perspective model depending on 2D or 3D measurement type:

$$\boldsymbol{J}_{\pi_2,i}^{(j)} = \frac{1}{{}^{C_i}\hat{\boldsymbol{Z}}^{(j)}}\begin{bmatrix}\mathbf{1} & \mathbf{0} & -\frac{{}^{C_i}\hat{\boldsymbol{X}}^{(j)}}{{}^{C_i}\hat{\boldsymbol{Z}}^{(j)}} \\ \mathbf{0} & \mathbf{1} & -\frac{{}^{C_i}\hat{\boldsymbol{Y}}^{(j)}}{{}^{C_i}\hat{\boldsymbol{Z}}^{(j)}}\end{bmatrix} \tag{29}$$

$$\boldsymbol{J}_{\pi_3,i}^{(j)} = \frac{1}{{}^{C_i}\hat{\boldsymbol{Z}}^{(j)}}\begin{bmatrix}\mathbf{1} & \mathbf{0} & -\frac{{}^{C_i}\hat{\boldsymbol{X}}^{(j)}}{{}^{C_i}\hat{\boldsymbol{Z}}^{(j)}} \\ \mathbf{0} & \mathbf{1} & -\frac{{}^{C_i}\hat{\boldsymbol{Y}}^{(j)}}{{}^{C_i}\hat{\boldsymbol{Z}}^{(j)}} \\ \mathbf{0} & \mathbf{0} & -\frac{1}{{}^{C_i}\hat{\boldsymbol{Z}}^{(j)}}\end{bmatrix} \tag{30}$$

Stacking the residuals of all measurements either 2D or 3D , we can get:

$$\boldsymbol{r}^{(j)} = \boldsymbol{H}_{\boldsymbol{x_B}}^{(\boldsymbol{j})}\tilde{\boldsymbol{x}} + \boldsymbol{H}_{\boldsymbol{f}}^{(\boldsymbol{j})}\, {}^{G}\tilde{\boldsymbol{p}}_{f_j} + \boldsymbol{n}^{(j)} \tag{31}$$

where $\boldsymbol{r}^{(j)}$, $\boldsymbol{H}_{\boldsymbol{x_B}}^{(\boldsymbol{j})}$, $\boldsymbol{H}_{\boldsymbol{f}}^{(\boldsymbol{j})}$, and $\boldsymbol{n}^{(j)}$ are block vectors or matrices with elements $r_i^{(j)}$, $\boldsymbol{H}_{\boldsymbol{x}_{B_i}}^{(\boldsymbol{j})}$, $\boldsymbol{H}_{\boldsymbol{f_i}}^{(\boldsymbol{j})}$, and $\boldsymbol{n}_i^{(j)}$. Eq.(37) can not be directly used for measurement update in MSCKF. Because the term ${}^{G}\tilde{\boldsymbol{p}}_{f_j}$ is not part of state vector. In order to transform (37) into a standard form for update, we can compute a semi-unitary matrix $\boldsymbol{A}$ whose columns form the basis of the left nullspace of $\boldsymbol{H}_{\boldsymbol{f}}^{(\boldsymbol{j})}$, and multiply both sides of (37) by $\boldsymbol{A}$ [8] .

$$\begin{aligned}\boldsymbol{r}_{\boldsymbol{o}}^{(j)} = \boldsymbol{A}^T\boldsymbol{r}^{(j)} &= \boldsymbol{A}^T\boldsymbol{H}_{\boldsymbol{x_B}}^{(\boldsymbol{j})}\tilde{\boldsymbol{x}} + \mathbf{0} + \boldsymbol{A}^T\boldsymbol{n}^{(j)} \\ &= \boldsymbol{H}_{\boldsymbol{o}}^{(\boldsymbol{j})}\tilde{\boldsymbol{x}} + \boldsymbol{n}_0^{(j)}\end{aligned} \tag{32}$$

Now, we obtain the useful form for filter update. We assume the quantity of 2D measurements is $\mathcal{M}_2^{(j)}$, while $\mathcal{M}_3^{(j)}$ for 3D measurements. Thus $\boldsymbol{A}$ has dimension $(2\mathcal{M}_2^{(j)} + 3\mathcal{M}_3^{(j)}) \times (2\mathcal{M}_2^{(j)} + 3\mathcal{M}_3^{(j)} - 3)$ and $\boldsymbol{r}_{\boldsymbol{o}}^{(j)}$ has dimension $(2\mathcal{M}_2^{(j)} + 3\mathcal{M}_3^{(j)} - 3) \times 1$. The covariance matrix of $\boldsymbol{n}_o^{(j)}$ is $\boldsymbol{\Sigma}_{\boldsymbol{o}}^{(j)} = \boldsymbol{A}^T\boldsymbol{\Sigma}^{(j)}\boldsymbol{A}$. We can now stack all the errors $\boldsymbol{r}_{\boldsymbol{o}}^{(j)}$ for all the features selected for update.

$$\boldsymbol{r}_{\boldsymbol{o}} = \boldsymbol{H}_{\boldsymbol{o}}\tilde{\boldsymbol{x}} + \boldsymbol{n}_o \tag{33}$$

To reduce the computational complexity of the MSCKF update, a QR-decomposition of $\boldsymbol{H}_{\boldsymbol{o}}$ is employed.

Finally, we calculate the Kalman gain. And the state vector and covariance are updated following the description in [8].

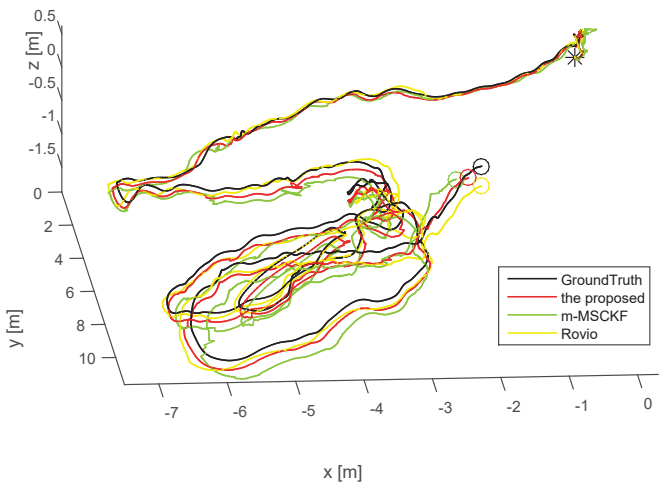

Fig. 5: Comparison on trajectory estimations of mono-MSCKF, Rovio and our method on MH_01_easy dataset.

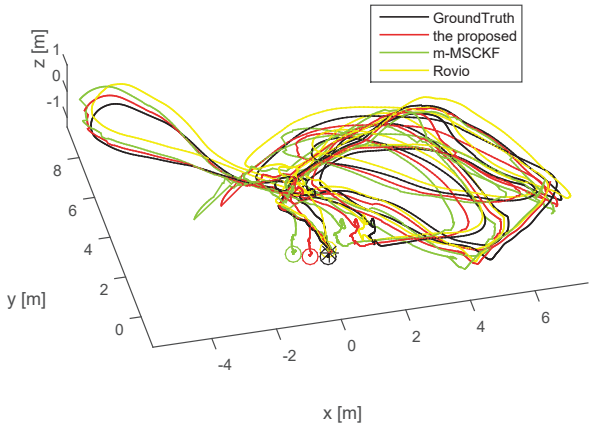

Fig. 6: Comparison on trajectory estimations of mono-MSCKF, Rovio and our method on MH_03_medium dataset.

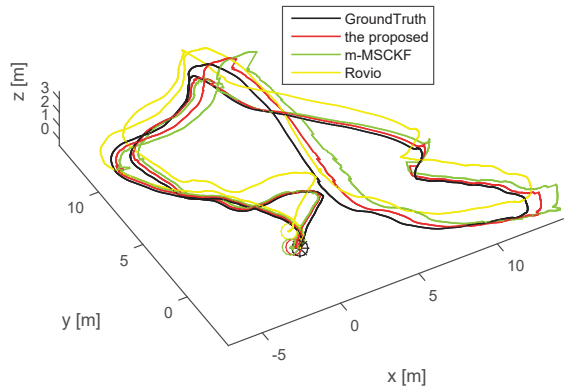

Fig. 7: Comparison on trajectory estimations of mono-MSCKF, Rovio and our method on $MH\_05\_difficult$ dataset.

TABLE I: ATE on EuRoC / ASL Dataset

Dataset	mono-MSCKF			Rovio			The Proposed		
	Mean	Median	RMSE[m]	Mean	Median	RMSE[m]	Mean	Median	RMSE[m]
MH_01_easy	0.3536	0.3433	0.3875	0.2517	0.2560	0.2761	**0.1716**	**0.1768**	**0.1936**
MH_03_medium	0.7778	0.8026	0.8511	0.4053	**0.3839**	0.4524	**0.4002**	0.4134	**0.4439**
$MH\_05\_difficult$	0.8943	0.6438	1.1091	1.0490	1.0891	1.1250	**0.4292**	**0.2984**	**0.5262**
$V2\_01\_easy$	0.2500	0.1963	0.2998	0.2383	0.2437	0.2564	**0.0959**	**0.0633**	**0.1370**
$V2\_02\_medium$	0.4825	0.4576	0.5329	0.3660	**0.2948**	**0.4296**	**0.1643**	0.4525	0.4576

III. IMPLEMENTATION DETAILS

A. Feature Detection and Match

In this paper, We extracted 60-to-200 salient point features using Oriented FAST and Rotated BRIEF (ORB) detector in each frame [18]. ORB feature maintains a image pyramid to find the best corners independent of the scale. For compromise of precision and efficiency, we use a 3-level pyramid while the scale factor between adjacent levels is 1.2. The relationship between feature pixel bearing vector uncertainty and level in which this feature is detected is

$$\sigma_\alpha = \frac{\sigma_u}{f_u}(1.2)^{level} \qquad \sigma_\beta = \frac{\sigma_v}{f_v}(1.2)^{level} \tag{34}$$

σ_u and σ_v are feature imagery coordinate isotropic uncertainties at 0-level image in pyramid. f_u and f_v are camera focal length.

Image is divide in cells of fixed size (e.g. 32×30 pixels). This results in evenly distributed features in image by limiting each cell only maintaining the strongest feature ,if it exists.

Each feature track always tries to estimate 3D position of the corresponding landmark in the frame where first detected. To make 2D-to-3D matching be available to accelerate feature search, first stereo measurements (if available) or first two $CAM0$ measurement of one feature are used to triangulate this feature's 3D position . Reprojecting the this 3D local point ,which gives a good searching guess for feature matches in succeeding images, keeps the algorithm efficient.

B. Outlier Detection

After the local feature position is achieved when it is first detected, the position uncertainty is reprojected into succeeding images

$$^{C_{cur}}\boldsymbol{S} = \boldsymbol{J_h}\,^{C_{first}}\boldsymbol{\Sigma}_p\boldsymbol{J_h}^T \tag{35}$$

where $\boldsymbol{J_h}$ is the 2×3 Jacobian of $\boldsymbol{h}(\bullet)$ which projects 3D points in the frame where the feature is first detected to current image with respect to the feature position. Than the 2σ uncertainty ellipse derivated from 2×2 matrix $^{C_{cur}}\boldsymbol{S}$ in image frame is used to predict potential matching and reject outliers.

Further, once $\boldsymbol{r}_o^{(j)}$ and $\boldsymbol{H}_o^{(j)}$ are calculated in (38), we proceed to carry out a Mahalanobis gating test for the residual $\boldsymbol{r}_o^{(j)}$. Here a 95th percentile of the χ^2 test with $(2\mathcal{M}_2^{(j)} + 3\mathcal{M}_3^{(j)} - 3)$ degrees of freedom is made.

IV. EXPERIMENTS

We implemented and tested both algorithms as a single thread program on a Lenovo laptop with a 2.4 GHz Intel Core i7-5500 CPU and 12 GB of DDR3L RAM. The experiments are mainly on the EuRoC MAV dataset recorded by ETH Autonomous Systems Lab. This dataset is recorded using VI-sensor and includes data streams from stereo camera and IMU with accurate state groundtruth [19].

A. Pose Estimation

We compare the the proposed method with monocular-based MSCKF (mono-MSCKF) and Rovio, another popular visual-inertial odometry based on EKF [20]. We implemented a mono-MSCKF method based [8] and carried out indoor experiment in different scenarios. In

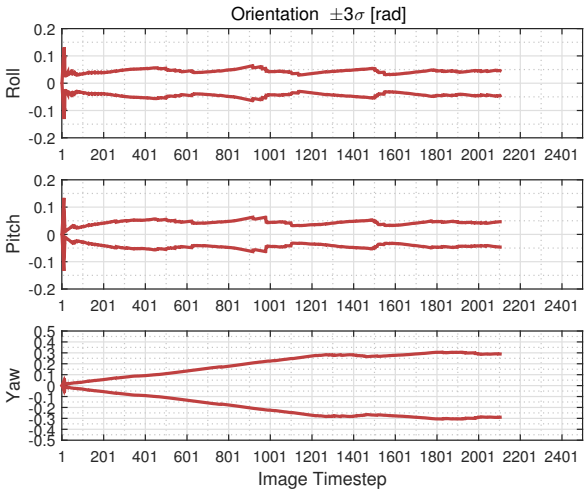

Fig. 8: Estimated orientation $\pm 3\sigma$ bounds of Roll, Pitch and Yaw

Fig.5-7, groundtruth and trajectories estimated from all three methods on MH_01_easy, MH_03_medium and $MH\_05\_different$ dataset. TABLE.1 contains the absolute trajectory error (ATE) between the estimated and the reference trajectory on five EuRoc datasets which contain either fast or slow motion in either light or dark scenes from different methods. Means, medians and RMSEs are listed. In all cases both trajectories were aligned by a rigid transform that minimizes their distance. These comparison results demonstrate the proposed method is robust to different motion and scenarios. Compared to mono-MSCKF and

Rovio, the proposed method has a higher accuracy on 6D pose estimation.

B. Orientation Observability

The results shown in Fig.8 show $\pm 3\sigma$ bounds corresponding to orientation estimate. The plots reveal the proposed estimator has correct observability properties for VINS. Roll and pitch are observable, the error bounds do not increase indefinitely.In contrast, those for the position and yaw do continuously increase because they are not observable [21].

C. Bias Estimation

In Fig.9, IMU biases are estimated. We can see the gyroscope biases exhibit a better convergence than the accelerometer biases. This result may come from the more direct link of rotational rates to visual errors [22].

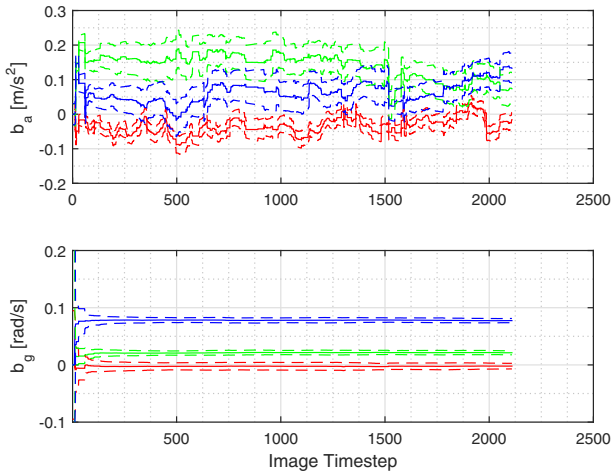

Fig. 9: Estimated IMU biases with their . Top: accelerometer biases(red:x, green:y, blue:z). Bottom: gyroscope biases(red:x, green:y, blue:z).

V. CONCLUSIONS

In this paper, we have presented a depth enhanced visual-inertial odometry system based on MSCKF. By integrating visual feature and depth information ,if available, a 3D meaurement model of landmark position is derivated, which reduces the estimation uncertainty. Combining 2D and 3D measurements results in more accurate estimation of landmark position. MSCKF update model is modified to gain compatibility of 2D and 3D measurement.

Experiments under datasets recorded in different scenarios underline the robustness of the proposed method on pose estimate. Compared to mono-MSCKF and other popular open-source method, the proposed gets competitive accuracy.

REFERENCES

[1] S. Weiss and R. Siegwart, "Real-time metric state estimation for modular vision-inertial systems," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 4531–4537.

[2] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 6, 2007.

[3] C. Roussillon and S. Lacroix, "High rate-localization for high-speed all-terrain robots," in *Communications, Computing and Control Applications (CCCA), 2012 2nd International Conference on*. IEEE, 2012, pp. 1–8.

[4] E. S. Jones and S. Soatto, "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach," *The International Journal of Robotics Research*, vol. 30, no. 4, pp. 407–430, 2011.

[5] J. Kelly, S. Saripalli, and G. S. Sukhatme, "Combined visual and inertial navigation for an unmanned aerial vehicle," in *Field and Service Robotics*. Springer, 2008, pp. 255–264.

[6] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[7] M. Li, "Visual-inertial odometry on resource-constrained systems," 2014.

[8] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 3565–3572.

[9] M. Li and A. I. Mourikis, "High-precision, consistent ekf-based visual–inertial odometry," *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013.

[10] J. Zhang, M. Kaess, and S. Singh, "Real-time depth enhanced monocular odometry," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 4973–4980.

[11] M. N. Galfond, "Visual-inertial odometry with depth sensing using a multi-state constraint kalman filter," Ph.D. dissertation, Massachusetts Institute of Technology, 2014.

[12] N. Trawny and S. I. Roumeliotis, "Indirect kalman filter for 3d attitude estimation," *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep*, vol. 2, p. 2005, 2005.

[13] M. Li and A. I. Mourikis, "3-d motion estimation and online temporal calibration for camera-imu systems," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 5709–5716.

[14] F. Pang and T. Wang, "Stereo-inertial pose estimation and online sensors extrinsic calibration," in *Robotics and Biomimetics (ROBIO), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1377–1382.

[15] A. W. Stroupe, M. C. Martin, and T. Balch, "Merging gaussian distributions for object localization in multi-robot systems," in *Experimental Robotics VII*. Springer, 2001, pp. 343–352.

[16] L. E. Clement, V. Peretroukhin, J. Lambert, and J. Kelly, "The battle for filter supremacy: A comparative study of the multi-state constraint kalman filter and the sliding window filter," in *Computer and Robot Vision (CRV), 2015 12th Conference on*. IEEE, 2015, pp. 23–30.

[17] J. Montiel, J. Civera, and A. J. Davison, "Unified inverse depth parametrization for monocular slam," *analysis*, vol. 9, p. 1, 2006.

[18] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. IEEE, 2011, pp. 2564–2571.

[19] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[20] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct ekf-based approach," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 298–304.

[21] J. Kelly and G. S. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *The International Journal of Robotics Research*, vol. 30, no. 1, pp. 56–79, 2011.

[22] S. M. Weiss, "Vision based navigation for micro helicopters," Ph.D. dissertation, 2012.