Stereo-Inertial Pose Estimation and Online Sensors Extrinsic Calibration

Fumin Pang and Tianmiao Wang, *M*ember, *IEEE*

***Abstract*— The fusion of visual and inertial measurement has been popular in mobile robotics community due to the complementary properties of two sensors. The combination of these two sensors offers rich texture of environment and accurate short-time motion prediction, making particularly suitable for pose estimation, especially in GPS-denied unknown environment.In this paper ,we propose a method which fuses stereo visual and inertial cues based on Multi-State Constraint Kalman Filter (MSCKF), to estimate 6DOF pose of mobile robot. Stereo vision offers real-metric perception of surrounding, giving a better initial scale estimation of the visual-inertial system. Compared with another class of methods based on batch nonlinear optimization , this filter-based method is more suitable for resource-constrained mobile platforms.On addition, the finite precision of sensors extrinsic calibration often makes estimator inconsistent.The proposed method includes extrinsic parameters in state vector to do online calibration. Experimental results on real-world datasets demonstrate that proposed method substantially improves the accuracy of pose estimation and the calibration between sensors.**

I. INTRODUCTION

Accurate 6DOF pose estimation in unknown environment is an critical component in autonomous navigation task. It is also one of the most popular topics in robotic community. Visual and inertial sensors have been more and more applied to estimate the pose of mobile robots due to their complementary properties, low cost and safety. Visual measurement offers rich texture of surrounding. Each visual feature can always be tracked by a camera from a sequence of consecutive poses, which supply with multiple constraint of camera motion. Inertial measurement can provide accurate motion prediction within a short time a higher frequency. In Simultaneous localization and mapping (SLAM), fusion of visual and inertial measurement is used to estimation robot pose and build a map of environment [1] [2]. Another pose estimation paradigm is Visual-Inertial Odometry (VIO),which merely estimate robot motion typically.

In this paper, we are interested in visual-inertial odometry. The most common VIO methods can be classified into two categories: recursive filter-based and batch-based nonlinear optimization approaches. Filter-based approaches are the earlier ones used to solve SLAM and VIO problems. In this class of method, inertial sensors are regarded as interoceptive sensors to provide linear accelerator and angular velocity, which are always integrated to form prior position and orientation estimation. Meanwhile,the visual sensors, as exteroceptive sensors at lower frequency, provide visual measurement to calculate the innovation of the filter. Thus, posterior estimation is achieved recursively. Batch based methods [3] are later ones, but achieve some impressing results in recent years [4] [5] [6]. These methods promise results of higher accuracy compared with filtering approaches as they employ re-linearization at each iteration to better deal with the nonlinearity of the measurement models. They always implement bundle adjustment in a sliding window of states, using multiple iterations to minimize cost function, which results in increased computational cost, however. Thus for a long-time running system, the lack of computational resources made recursive algorithms a favorable choice for online estimation, especially in resource-constrained systems, such as micro aerial vehicles, mobile phones, and augmented reality (AR) devices [7]. Thus, in this paper, we focus on the filter-based method.

Filter-based methods can also be divided into two main categories:loosely-coupled methods and tightly-coupled methods.Loosely-coupled ones are the most intuitive method for fusing visual and inertial measurement. This kind of method deal with visual and inertial measurement separately. Visual pipeline, always as a black box, estimate poses independently. These poses are regard as the measurement of EKF and then fused with the inertial measurement recursively [8]. Weiss and Siegwart use a monocular SLAM framework PTAM to estimate 6DOF poses of a micro aerial vehicle mounted with a down-looking camera [9] [10] . Then , the poses are fused with the prior state vector driven by inertial reading. Lynen and Achtelik implement a EKF-based framework for dealing with loosely-coupled multiple sensor Fusion [11]. This kind of method reduce computational complexity , but it ignore some crucial information. For instance, processing IMU measurements separately does not allow for optimal estimation of sensor biases,and using feature measurements for motion estimation between pairs of images ignores the correlations between consecutive timesteps [12].

Tightly-coupled methods are ones directly fuse the visual and inertial data for achieving higher precision. The most common EKF estimator to tightly fuse visual and inertial data is EKF-based SLAM, in which current camera pose and features positions are jointly included in estimator state vector [13] [14]. EKF-based SLAM suffers from the cubic computational complexity of feature number in state vector. To overcome this limitation, Mourikis and Roumeliotis come up with multi-state constraint Kalman filter(MSCKF) [12]. Instead of feature positions, a sliding window of camera poses is included in MSCKF estimator. By directly making use of the geometric constraints between multiple camera

Fumin Pang and Tianmiao Wang are with the School of Mechanical Engineering and Automation, Beihang University, Beijing, 100191 China. E-mail: fuminpang@buaa.edu.cn, itm@buaa.edu.cn

poses it avoids the computational burden and loss of information associated with pairwise displacement estimation. To improve the consistency of the MSCKF, Li and Mourikis include Camera-IMU extrinsic calibration parameter in estimator state vector, giving a more accurate result [15].

In this paper, we propose a stereo visual inertial navigation system. As we know, almost all MSCKF-based implementations are based on monocular camera-IMU rig. Stereo visual measurement can give a real metric perception of environment, which results in a better initialization of the estimator. A key contribution of proposed method is we give the derivation of stereo visual measurement model which is different from the existing monocular one. To improve the performance, the system includes online extrinsic calibration between IMU and two cameras. We test the proposed method on real-world dataset. The experimental results show that the resulting method is consistent, and that it attains substantially higher accuracy than monocular MSCKF.

II. ESTIMATOR DESCRIPTION

As illustrated in Fig.1, we affix stereo camera-IMU rig body frame $\{B\}$ to IMU, to track the 6D motion with respect to a global coordinate frame, $\{G\}$. Two camera coordinate frames, CAM0 and CAM1, are $\{C_0\}$ and $\{C_1\}$.

A. MSCKF State Parametrization

The full MSCKF state representation can be partitioned into three parts. 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. We parametrize the body state at time k as the 16-dimensional vector.

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

Where ${}^B_G\bar{\mathbf{q}}_k$ is the unit quaternion representing the rotation which roatate vectors from the global frame $\{G\}$ to the body frame $\{B\}$. In this paper, all quaternions follow JPL convention [16]. ${}^G\mathbf{p}_{B,k}$ 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 vecotr repersenting 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 $\mathbf{a}_m$.

The second state part is extrinsic calibration parameters (rotation and translation) between IMU and two cameras. We parametrize it as the 14-dimensional vector.

$$\mathbf{x}_{Calib,k} := \begin{bmatrix} {}^{C_0}_B\bar{\mathbf{q}}_k{}^T & {}^B\mathbf{p}_{C_0,k}{}^T & {}^{C_1}_B\bar{\mathbf{q}}_k{}^T & {}^B\mathbf{p}_{C_1,k}{}^T \end{bmatrix}^T \tag{2}$$

Where ${}^{C_i}_B\bar{\mathbf{q}}_k$ rotate vectors from body frame $\{B\}$ to camera frame $\{C_i\}$. ${}^B\mathbf{p}_{C_i,k}$ is the original point position of camera frame $\{C_i\}$ expressed in body frame $\{B\}$, i = 0,1.

The third part of the full state is a sliding window of N past body states, in which active feature tracks were visible.

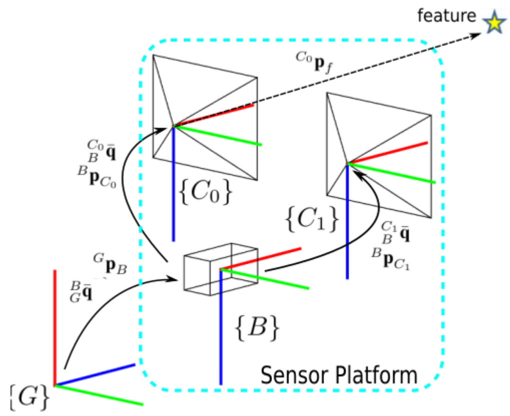

Fig. 1. Coordinate frames involved in sensor platform and visual feature.

The *i-th* body state, $i = 0...N-1$, inlcuding 6D pose and velocity, is a 10-dimensional vector.

$$\mathbf{x}_{B_i,k} := \begin{bmatrix} {}^{C_0}_{B_i}\bar{\mathbf{q}}_k{}^T & {}^B\mathbf{p}_{B_i,k}{}^T & {}^G\mathbf{v}_{B_i,k}{}^T \end{bmatrix}^T \tag{3}$$

At time k, the full state of MSCKF is a (30 + 10* N) vector consisting of current body state estimate, calibration parameters and N last body states.

$$\hat{\mathbf{x}}_k := \begin{bmatrix} \hat{\mathbf{x}}_{I,k}^T & \hat{\mathbf{x}}_{Calib,k}^T & \hat{\mathbf{x}}_{B_0,k}^T & \ldots & \hat{\mathbf{x}}_{B_{N-1},k}^T \end{bmatrix}^T \tag{4}$$

We can define *error state* of the estimator at time k:

$$\tilde{\mathbf{x}}_k := \begin{bmatrix} \tilde{\mathbf{x}}_{I,k}^T & \tilde{\mathbf{x}}_{Calib,k}^T & \tilde{\mathbf{x}}_{B_0,k}^T & \ldots & \tilde{\mathbf{x}}_{B_{N-1},k}^T \end{bmatrix}^T \tag{5}$$

where

$$\tilde{\mathbf{x}}_{I,k} := \begin{bmatrix} {}^G_B\boldsymbol{\delta\theta}_I^T & {}^G\tilde{\mathbf{p}}_{B,k}{}^T & {}^G\tilde{\mathbf{v}}_{B,k}{}^T & \tilde{\mathbf{b}}_{g,k}^T & \tilde{\mathbf{b}}_{a,k}^T \end{bmatrix}^T \tag{6}$$

$$\tilde{\mathbf{x}}_{B_i,k} := \begin{bmatrix} {}^G_{B_i}\boldsymbol{\delta\theta}_I^T & {}^G\tilde{\mathbf{p}}_{B_i,k}{}^T & {}^G\tilde{\mathbf{v}}_{B_i,k}{}^T \end{bmatrix}^T \tag{7}$$

The full error state has (27 + 9 N) dimensions. In the above, $\tilde{x}$ is the difference between the true value x and the estimated value $\bar{x}$. For position, velocity and bias, it is definde as $\tilde{x} = x - \bar{x}$. The quaternion error is definde according to

$$\boldsymbol{\delta}\bar{\boldsymbol{q}} := \hat{\bar{\boldsymbol{q}}}^{-1} \otimes \bar{\boldsymbol{q}} \approx \begin{bmatrix} \frac{1}{2}\boldsymbol{\delta\theta}^T & 1 \end{bmatrix}^T \tag{8}$$

Accordingly, the MSCKF error state covariance $\boldsymbol{P}$ is a (27 + 9 N) $\times$ (27 + 9 N) matrix. It can be partitioned into 4 blocks:

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{P_{I,I}} & \boldsymbol{P_{I,Cali-B}} \\ \boldsymbol{P_{I,Cali-B}^T} & \boldsymbol{P_{Cali-B,Cali-B}} \end{bmatrix} \tag{9}$$

where $\boldsymbol{P_{I,I}}$ is the 15×15 covariance matrix of current body state.$\boldsymbol{P_{Cali-B,Cali-B}}$ is the combinated $(12+9N) \times (12+9N)$ covarince matrix of calibration state and past body states. $\boldsymbol{P_{I,Cali-B}}$ is the cross-covariance between the current body state and combination of calibration and past body states.

B. Filer Propagation

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

$$\boldsymbol{\omega}_m = {}^G\boldsymbol{\omega} + \boldsymbol{b}_g + \boldsymbol{n}_g \tag{10}$$

$$\boldsymbol{a}_m = {}^B_G\mathbf{R}\left({}^G\boldsymbol{a} - {}^G\boldsymbol{g}\right) + \boldsymbol{b}_a + \boldsymbol{n}_a \tag{11}$$

where ${}^B_G\mathbf{R}$ is the rotation matrix cooresponding 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. Using these measurements, we can write the dynamics of the filter state vector as:

$${}^I_G\dot{\bar{\mathbf{q}}}(t) = \frac{1}{2}\boldsymbol{\Omega}\left(\boldsymbol{\omega}_m(t) - \boldsymbol{b}_g(t) - \boldsymbol{n}_g(t)\right){}^I_G\bar{\mathbf{q}} \tag{12}$$

$${}^G\dot{\mathbf{v}}_B(t) = {}^B_G\mathbf{R}^T(t)\left(\boldsymbol{a}_m(t) - \boldsymbol{b}_a(t) - \boldsymbol{n}_a(t)\right) + {}^G\boldsymbol{g} \tag{13}$$

$${}^G\dot{\mathbf{p}}_B(t) = {}^G\mathbf{v}_B(t) \tag{14}$$

$$\dot{\boldsymbol{b}}_{\boldsymbol{g}}(t) = \boldsymbol{n}_{wg}(t) \quad \dot{\boldsymbol{b}}_{\boldsymbol{a}}(t) = \boldsymbol{n}_{wa}(t) \tag{15}$$

$${}^{C_i}_B\dot{\bar{\mathbf{q}}}(t) = \mathbf{0} \quad {}^B\dot{\mathbf{p}}_{C_i}(t) = \mathbf{0} \tag{16}$$

where $\boldsymbol{\Omega}$ is the quaternion multiplication matrix corresponding to the angular velocity vector $\boldsymbol{\omega}$. In the above, the first three equations describe the dynamics of the current body motion. The fourth line models the biases as random walk processes. The fifth line describe the fact that camera-IMU transformation dose not change in time. As for the past body states, them remain constant once the corresponding image is captured.

Equation (12)-(16) describe the the continuous-time evolution of the true states. We follow the approach described in (High-precision, consistent EKF-based visual-inertial odometry) for propagating the state estimates in a discrete-time implementation. Further, a fifth-order Runge-Kutta procedure is used for propagate quaternion.

Besides the current body position, velocity, and orientation, all other state estimates remain unchanged during propagation. We can also examine the linearized continuous-time model of the current state error state, $\tilde{\mathbf{x}}_I$:

$$\dot{\tilde{\mathbf{x}}}_I = \boldsymbol{F}\tilde{\mathbf{x}}_I + \boldsymbol{G}\boldsymbol{n}_I \tag{17}$$

where $\boldsymbol{F}$ is the continuous-time error-state transition matrix. G is given by

$$\boldsymbol{G} = \begin{bmatrix} -\boldsymbol{I_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} \\ \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} \\ \mathbf{0_3} & -{}^B_G\hat{\mathbf{R}}^T & \mathbf{0_3} & \mathbf{0_3} \\ \mathbf{0_3} & \mathbf{0_3} & \boldsymbol{I_3} & \mathbf{0_3} \\ \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \boldsymbol{I_3} \end{bmatrix} \tag{18}$$

and $\boldsymbol{n}_I = [\boldsymbol{n}_g^T \ \boldsymbol{n}_a^T \ \boldsymbol{n}_{wg}^T \ \boldsymbol{n}_{wa}^T]^T$ is the IMU process noise, which has covarince matrix $\boldsymbol{Q}_I$.

In addition to the state estimate, the MSCKF propagates the state covariance matrix, as follows:

$$\boldsymbol{P}(t_{k+1}) = \begin{bmatrix} \boldsymbol{P_{I,I}}(t_{k+1}) & \boldsymbol{\Phi}_{I_k}\boldsymbol{P_{I,Cali-B}}(t_{k+1}) \\ \boldsymbol{P}^T_{\boldsymbol{I,Cali-B}}(t_k)\boldsymbol{\Phi}^T_{I_k} & \boldsymbol{P_{Cali-B,Cali-B}}(t_k) \end{bmatrix} \tag{19}$$

where

$$\boldsymbol{P_{I,I}}(t_{k+1}) = \boldsymbol{\Phi}_{I_k}\boldsymbol{P_{I,I}}(t_k)\boldsymbol{\Phi}^T_{I_k} + \boldsymbol{G}\boldsymbol{Q_I}\boldsymbol{G^T}\boldsymbol{\Delta T} \tag{20}$$

and $\boldsymbol{\Delta T}$ is the IMU sample period. $\boldsymbol{\Phi}_{I_k}$ is the discrete-time error-state transition matrix at k . Instead of using the most common approximation $\boldsymbol{\Phi} = \boldsymbol{I} + \boldsymbol{F}\boldsymbol{\Delta T}$, we use the closed-form matrix by [15].

C. State Augmentation

Upon recording a new image, a copy of current body rotation, position and velocity is appended to the state vector, and the covariance matrix of the MSCKF is augmented accordingly:

$$\boldsymbol{P}(t_k) \leftarrow \begin{bmatrix} \boldsymbol{I_{27+9N}} \\ \boldsymbol{J} \end{bmatrix} \boldsymbol{P}(t_k) \begin{bmatrix} \boldsymbol{I_{27+9N}} \\ \boldsymbol{J} \end{bmatrix}^T \tag{21}$$

where J is given by

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{I_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_{3\times(18+9N)}} \\ \mathbf{0_3} & \boldsymbol{I_3} & \mathbf{0_3} & \mathbf{0_{3\times(18+9N)}} \\ \mathbf{0_3} & \mathbf{0_3} & \boldsymbol{I_3} & \mathbf{0_{3\times(18+9N)}} \end{bmatrix} \tag{22}$$

D. Filter Update

When a tracked feature $\boldsymbol{f_j}$ is lost in current frame, it is selected for state update, the MSCKF estimates its position ${}^G\hat{\mathbf{p}}_f$ using an inverse depth [17] least-squares Gauss-Newton optimization [18]. The procedure takes as input N pair of camera pose and N sets of feature measurements. As for stereo visual system , each past body pose has two corresponding camera poses, and the two camera poses can calculated by

$${}^{C_i}_G\hat{\bar{\mathbf{q}}} = {}^{C_i}_B\hat{\bar{\mathbf{q}}} \otimes {}^B_G\hat{\bar{\mathbf{q}}} \quad {}^G\mathbf{p}_{C_i} = {}^G\mathbf{p}_B + {}^B_G\hat{\mathbf{R}}^T\,{}^B\mathbf{p}_{C_i} \tag{23}$$

Now that we have estimated the positions of any features which can be used in the state update, we can apply the corresponding motion constraints to the window of poses from which each feature was tracked. To present the measurement model, we consider one feature,$\boldsymbol{f}_j$, which is observed by a set of pose pairs, $\mathcal{M}_j$. $\boldsymbol{z}_{l,0}$ and $\boldsymbol{z}_{l,1}$ are observations of feature $\boldsymbol{f}_j$ from stereo camera poses $\boldsymbol{C}_{l,0}$ and $\boldsymbol{C}_{l,1}$, $l \in \mathcal{M}_j$. The 4×1 stereo visual measurement residual can be computed by

$$\boldsymbol{r}_l^{(j)} = \begin{bmatrix} \boldsymbol{z}_{l,0}^{(j)} \\ \boldsymbol{z}_{l,1}^{(j)} \end{bmatrix} - \begin{bmatrix} \hat{\boldsymbol{z}}_{l,0}^{(j)} \\ \hat{\boldsymbol{z}}_{l,1}^{(j)} \end{bmatrix} \tag{24}$$

where

$$\hat{\boldsymbol{z}}_{l,i}^{(j)} = \frac{1}{{}^{C_{l,i}}\hat{\boldsymbol{Z}}^{(j)}} \begin{bmatrix} {}^{C_{l,i}}\hat{\boldsymbol{X}}^{(j)} & {}^{C_{l,i}}\hat{\boldsymbol{Y}}^{(j)} \end{bmatrix}^T \tag{25}$$

with

$$\begin{aligned} {}^{C_{l,i}}\hat{\boldsymbol{p}}_{f_j} &= \begin{bmatrix} {}^{C_{l,i}}\hat{\boldsymbol{Z}}^{(j)} & {}^{C_{l,i}}\hat{\boldsymbol{X}}^{(j)} & {}^{C_{l,i}}\hat{\boldsymbol{Y}}^{(j)} \end{bmatrix}^T \\ &= {}^{C_{l,i}}_G\hat{\boldsymbol{R}}\left({}^G\hat{\boldsymbol{p}}_{f_j} - {}^G\hat{\boldsymbol{p}}_{C_{l,i}}\right) \end{aligned} \tag{26}$$

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

$$\boldsymbol{r}_l^{(j)} = \boldsymbol{H}_{\boldsymbol{x}_{B_l}}^{(j)} \tilde{\boldsymbol{x}} + \boldsymbol{H}_{\boldsymbol{f}_l}^{(j)} {}^G\tilde{\mathbf{p}}_{f_j} + \boldsymbol{n}_l^{(j)} \tag{27}$$

where the matrices $\boldsymbol{H}_{\boldsymbol{x}_{B_l}}^{(j)}$ and $\boldsymbol{H}_{\boldsymbol{f}_l}^{(j)}$ are the corresponding Jacobians of the measurement $\hat{\boldsymbol{z}}_l^{(j)}$ with respect to the state and the feature position, respectively, and ${}^G\tilde{\mathbf{p}}_{f_j}$ is the error in the position estimate of $\boldsymbol{f}_j$. $\boldsymbol{n}_l^{(j)}$ is the 4×1 visual measurement noise with the corresponding matrix $\boldsymbol{R}_l^{(j)} = \sigma_{im}^2 \boldsymbol{I}_4$. $\boldsymbol{H}_{\boldsymbol{x}_{B_l}}^{(j)}$ is given by

$$\boldsymbol{H}_{\boldsymbol{x}_{B_l}}^{(j)} = \begin{bmatrix} \mathbf{0}_{4\times 15} & \mathbf{\Pi}_l & \mathbf{0}_{4\times 9(l-1)} & \boldsymbol{H}_{B_l} & \mathbf{0}_{4\times 9(N-l)} \end{bmatrix} \tag{28}$$

where

$$\mathbf{\Pi}_l = \begin{bmatrix} \mathbf{\Pi}_{\boldsymbol{\theta}_{l,0}} & \mathbf{\Pi}_{\boldsymbol{p}_{l,0}} & \mathbf{0}_{2\times 3} & \mathbf{0}_{2\times 3} \\ \mathbf{0}_{2\times 3} & \mathbf{0}_{2\times 3} & \mathbf{\Pi}_{\boldsymbol{\theta}_{l,1}} & \mathbf{\Pi}_{\boldsymbol{p}_{l,1}} \end{bmatrix} \tag{29}$$

$$\boldsymbol{H}_{B_l} = \begin{bmatrix} \boldsymbol{H}_{\boldsymbol{\theta}_{l,0}} & \boldsymbol{H}_{\boldsymbol{p}_{l,0}} & \mathbf{0}_{2\times 3} \\ \boldsymbol{H}_{\boldsymbol{\theta}_{l,1}} & \boldsymbol{H}_{\boldsymbol{p}_{l,1}} & \mathbf{0}_{2\times 3} \end{bmatrix} \tag{30}$$

and for more details,the nonzero blocks in matrixs are the Jacobians with respect to the two camera-IMU rotations ,camera-IMU translations, body rotation and body position , respectively:

$$\mathbf{\Pi}_{\boldsymbol{\theta}_{l,i}} = \boldsymbol{J}_l^{(j,i)} {}^{C_i}_{B}\hat{\mathbf{R}} \lfloor {}^B_G\hat{\mathbf{R}}({}^G\boldsymbol{p}_{f_j} - {}^G\tilde{\boldsymbol{p}}_B)\times \rfloor \tag{31}$$

$$\mathbf{\Pi}_{\boldsymbol{p}_{l,i}} = \boldsymbol{J}_l^{(j,i)} \tag{32}$$

$$\boldsymbol{H}_{\boldsymbol{\theta}_{l,i}} = \boldsymbol{J}_l^{(j,i)} {}^{C_i}_{B}\hat{\mathbf{R}} {}^B_G\hat{\mathbf{R}} \lfloor ({}^G\boldsymbol{p}_{f_j} - {}^G\tilde{\boldsymbol{p}}_B)\times \rfloor \tag{33}$$

$$\boldsymbol{H}_{\boldsymbol{p}_{l,i}} = -\boldsymbol{J}_l^{(j,i)} {}^{C_i}_{B}\hat{\mathbf{R}} {}^B_G\hat{\mathbf{R}} \tag{34}$$

where $\lfloor \boldsymbol{c}\times \rfloor$ is the skew symmetric matrix corresponding to vector $\boldsymbol{c}$. And $\boldsymbol{H}_{\boldsymbol{f}_l}^{(j)}$ is given by

$$\boldsymbol{H}_{\boldsymbol{f}_l}^{(j)} = \boldsymbol{J}_l^{(j,i)} \begin{bmatrix} {}^{C_0}_{B}\hat{\mathbf{R}} {}^B_G\hat{\mathbf{R}} \\ {}^{C_1}_{B}\hat{\mathbf{R}} {}^B_G\hat{\mathbf{R}} \end{bmatrix} \tag{35}$$

In the above, $\boldsymbol{J}_l^{(j,i)}$ is the Jacobian of the perspective model:

$$\boldsymbol{J}_l^{(j,i)} = \frac{1}{{}^{C_{l,i}}\hat{\boldsymbol{Z}}^{(j)}} \begin{bmatrix} \mathbf{1} & \mathbf{0} & \dfrac{{}^{C_{l,i}}\hat{\boldsymbol{X}}^{(j)}}{{}^{C_{l,i}}\hat{\boldsymbol{Z}}^{(j)}} \\ \mathbf{0} & \mathbf{1} & \dfrac{{}^{C_{l,i}}\hat{\boldsymbol{Y}}^{(j)}}{{}^{C_{l,i}}\hat{\boldsymbol{Z}}^{(j)}} \end{bmatrix} \tag{36}$$

Stacking the residuals of all $\mathcal{M}_j$measurements of this feature, we can get:

$$\boldsymbol{r}^{(j)} = \boldsymbol{H}_{\boldsymbol{x}_B}^{(j)} \tilde{\boldsymbol{x}} + \boldsymbol{H}_{\boldsymbol{f}}^{(j)} {}^G\tilde{\mathbf{p}}_{f_j} + \boldsymbol{n}^{(j)} \tag{37}$$

where $\boldsymbol{r}^{(j)}$, $\boldsymbol{H}_{\boldsymbol{x}_B}^{(j)}$, $\boldsymbol{H}_{\boldsymbol{f}}^{(j)}$, and $\boldsymbol{n}^{(j)}$ are block vectors or matrices with elements $r_l^{(j)}$, $\boldsymbol{H}_{\boldsymbol{x}_{B_l}}^{(j)}$, $\boldsymbol{H}_{\boldsymbol{f}_l}^{(j)}$, and $\boldsymbol{n}_l^{(j)}$, for $\boldsymbol{l} \in \mathcal{M}_j$. And $\boldsymbol{R}_l^{(j)} = \sigma_{im}^2 \boldsymbol{I}_{4\mathcal{M}_j}$. Equation(37) can not be directly used for measurement update in MSCKF. Because the term $\boldsymbol{H}_{\boldsymbol{f}}^{(j)} {}^G\tilde{\mathbf{p}}_{f_j}$ is not part of state vector. In order to transform Eq.(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 $\boldsymbol{H}_{\boldsymbol{f}}^{(j)}$, and multiply both sides of Eq.(37) by $\boldsymbol{A}$.

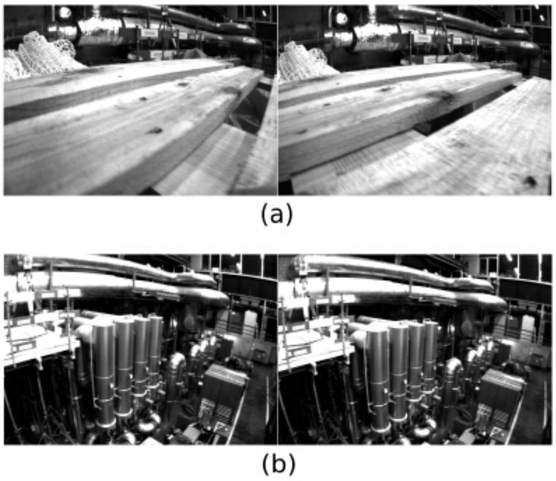

Fig. 2. Sample images from dataset used in pose estimation experiment. (a) is the first pair of stereo images. (b) is the last pair.

$$\begin{aligned} \boldsymbol{r}_{\boldsymbol{o}}^{(j)} = \boldsymbol{A}^T \boldsymbol{r}^{(j)} &= \boldsymbol{A}^T \boldsymbol{H}_{\boldsymbol{x}_B}^{(j)} \tilde{\boldsymbol{x}} + \mathbf{0} + \boldsymbol{A}^T \boldsymbol{n}^{(j)} \\ &= \boldsymbol{H}_{\boldsymbol{o}}^{(j)} \tilde{\boldsymbol{x}} + \boldsymbol{n}_0^{(j)} \end{aligned} \tag{38}$$

Now , we obtain the useful form for filter update. $\boldsymbol{H}_{\boldsymbol{x}_B}^{(j)}$ has full column rank, accordingly, A has dimension $4\mathcal{M}_j \times (4\mathcal{M}_j - 3)$ and $\boldsymbol{r}_{\boldsymbol{o}}^{(j)}$ has dimension $(4\mathcal{M}_j - 3) \times 1$. The covariance matrix of $\boldsymbol{n}_o^{(j)}$ is $\boldsymbol{R}_{\boldsymbol{o}}^{(j)} = \boldsymbol{A}^T \boldsymbol{R}^{(j)} \boldsymbol{A}$. We can now stack all the errors $\boldsymbol{r}_{\boldsymbol{o}}^{(j)}$ for all the features selected for update.

$$\boldsymbol{r}_{\boldsymbol{o}} = \boldsymbol{H}_{\boldsymbol{o}} \tilde{\boldsymbol{x}} + \boldsymbol{n}_o \tag{39}$$

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

$$\boldsymbol{H}_{\boldsymbol{o}} = \begin{bmatrix} \boldsymbol{Q}_0 & \boldsymbol{Q}_1 \end{bmatrix} \begin{bmatrix} \boldsymbol{H}_T \\ \mathbf{0} \end{bmatrix} \tag{40}$$

where $\boldsymbol{Q}_0$, $\boldsymbol{Q}_1$ are unitary matrices and $\boldsymbol{H}_T$ is an uppertriangular matrix. Substituting this result into (39) , we obtain:

$$\begin{bmatrix} \boldsymbol{Q}_0^T \boldsymbol{r}_o \\ \boldsymbol{Q}_1^T \boldsymbol{r}_o \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}_T \\ \mathbf{0} \end{bmatrix} \tilde{\boldsymbol{x}} + \begin{bmatrix} \boldsymbol{Q}_0^T \boldsymbol{n}_0 \\ \boldsymbol{Q}_1^T \boldsymbol{n}_0 \end{bmatrix} \tag{41}$$

We discard term $\boldsymbol{Q}_1^T \boldsymbol{r}_o$,because it is only noise. And we re-define error term used for update:

$$\boldsymbol{r}_{\boldsymbol{n}} := \boldsymbol{Q}_0^T \boldsymbol{r}_o = \boldsymbol{H}_{\boldsymbol{T}} \tilde{\boldsymbol{x}} + \boldsymbol{Q}_0^T \boldsymbol{n}_0 = \boldsymbol{H}_{\boldsymbol{T}} \tilde{\boldsymbol{x}} + \boldsymbol{n}_n \tag{42}$$

Then, the covariance matrix of n n is $\boldsymbol{R}_n = \boldsymbol{Q}_0^T \boldsymbol{R}_o \boldsymbol{Q}_0$. Now, we arrive at the final form of measurement equation. We can finally formulate the Kalman gain and correction equations to obtain the updated estimates for the filter state

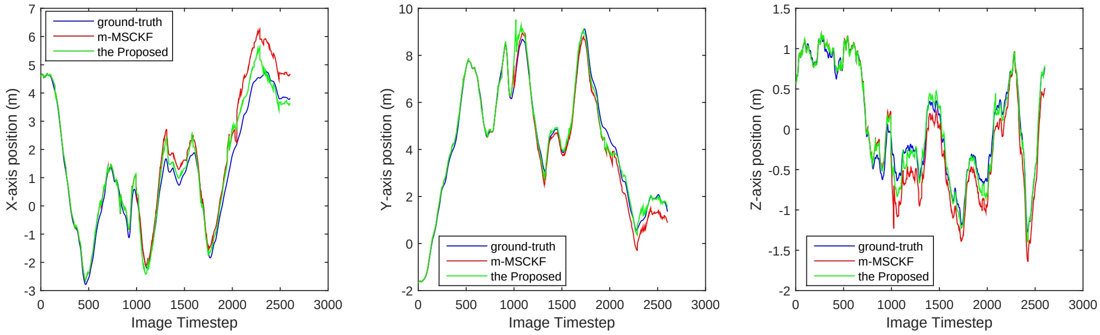

Fig. 3. Experimental results: comparison of ground-truth (blue), trajectories estimated by m-MSCKF (red) and the proposed method (green)

and covariance:

$$\boldsymbol{K} = \boldsymbol{P}_{k+1}\boldsymbol{H}_T^T(\boldsymbol{H}_T\boldsymbol{P}_{k+1}\boldsymbol{H}_T^T + \boldsymbol{R}_n)^{-1} \quad (43)$$

$$\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_{k+1} + \boldsymbol{K}\boldsymbol{r}_n \quad (44)$$

$$\boldsymbol{P}_{k+1} \leftarrow (\boldsymbol{I}_{27+9N} - \boldsymbol{K}\boldsymbol{H}_T)\boldsymbol{P}_{k+1}(\boldsymbol{I}_{27+9N} - \boldsymbol{K}\boldsymbol{H}_T)^T + \boldsymbol{K}\boldsymbol{R}_n\boldsymbol{K}^T \quad (45)$$

III. EXPERIMENTS

We tested the proposed method and compared it with the monocluar-based MSCKF method 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 [19]. The carefully calibrated extrinsic parameters and millimeter accurate position ground-truth is available in this dataset.

We implemented and tested both algorithms in MATLAB 2014b on a Lenovo laptop with a 2.4 GHz Intel Core i7 processor and 12 GB of DDR3L RAM. We extracted between 100 and 200 salient point features using Oriented FAST and Rotated BRIEF (ORB) detector [20] of OpenCV from the stereo pairs and tracked them temporally using Kanade-Lucas-Tomasi (KLT) tracking [21]. Outliers are rejected using a 5-point Random Sample Consensus (RANSAC) . We conducted two experiments on the dataset to compare the accuracy of the the proposed method with the monocular-based one and show the result of the calibration between sensors. We discuss both of these in turn.

A. Pose Estimation

To compare the performance between monocular-based MSCKF (m-MSCKF) and the proposed method, we implemented a m-MSCKF method based [12] and carried out indoor experiment using real-world dataset. The total duration of the experiment is 130s and the trajectory length is approximately 59 m. The IMU sample rate is 200Hz and images are recorded at 20 Hz. Fig.2 shows sample images from the dataset.

In Fig.3, the estimated poses are displayed in X, Y and Z axis separately. It also gives a comparison between the m-MSCKF method. According to the ground truth, the final position is $[3.8050 \quad 1.3780 \quad 0.7572]^T$m. The final position estimation given by the m-MSCKF method is $[4.6609 \quad 0.9028 \quad 0.5055]^T$m, while the proposed method gives $[3.6275 \quad 1.5840 \quad 0.7833]^T$m. The average error between the trajectory estimated by proposed method and the ground truth is $[0.2810 \ \ 0.1804 \ \ 0.0659]^T$m in three axis. The results show that the proposed method owns better precision than m-MSCKF method.

B. Sensors Calibration

In order to validate the proposed filter algorithm is effective for estimating the IMU-camera transformation while pose estimation, we added initial alignment error for translation and rotaion to the known extrinsic calibration between sensors.

For simplicity and due to limited space, in this section, we just discuss the calibration of transformation between IMU and CAM0. Because the calibration of IMU and CAM1 is in a similarity situation. The initial alignment error for translation is set to ${}^B\tilde{\boldsymbol{p}}_{C_0} = [-0.03 \quad 0.10 \quad -0.09]^T$m with a standard deviation of $[0.0548 \quad 0.0447 \quad 0.0458]^T$m in each axis. The initial alignment error for rotation is set to ${}^{C_0}_{B}\boldsymbol{\delta\theta} = [5° \quad 6° \quad -4°]^T$ with a standard deviation of $[3.3914° \quad 3.8455° \quad 3.1917°]^T$ in each axis of rotation. Consequently, the filter state vector and error-state covariance matrix are initialized according to the process described in Section II.

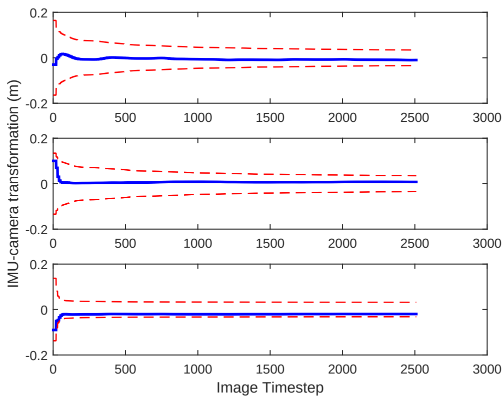

Fig. 4. IMU-camera translation error and 3σ bounds. Translation along axes x, y, and z.

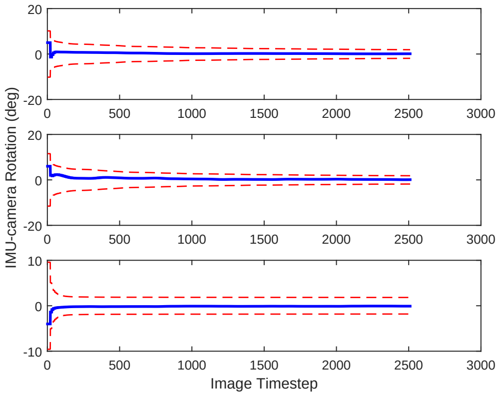

Fig. 5. IMU-camera rotation error and 3σ bounds. Rotation about axes x (roll), y (pitch), and z (yaw).

TABLE I
FINAL UNCERTAINTY (3σ) OF THE IMU-CAMERAS PARAMETERS AFTER 2500 IMAGE TIMESTEP

3σ		x(m)	y(m)	z(m)	r(°)	p(°)	y(°)
CAM0	Initial	0.16	0.13	0.134	10.17	11.53	9.57
	Final	0.03	0.03	0.03	1.90	1.82	1.81
CAM1	Initial	0.28	0.23	0.19	9.41	12.63	12.06
	Final	0.03	0.03	0.03	1.90	1.83	1.84

In Figs. 4 and 5, the IMU-camera clibration errors and their 3σ bounds for the 6-DOF transformation between the IMU and the camera are shown.

Fianlly, we list the initial and final uncertanty of the IMU-camera parameters (transition and rotation), including CAM0 and CAM1 in Table I. The results shown here demonstrate that the proposed calibration is capable of operating in a real-world environment and this method is effective.

IV. CONCLUSIONS

In this paper, we presented a filter-based visual inertial sensor fusion algorithm for pose estimation. The main contribution is the derivation of a measurement model of stereo visual observation. And no other method of attempting to stereo visual information with inertial sensor in a MSCKF framework has been found on literature yet. On addition, the proposed method estimates estrinsic calibration parameters between sensors to improve the performamce. The experimental results indicate that our method has a better accuracy than a self-implemented m-MSCKF ant it is effective to estimate sensor extrinsic calibration online.

REFERENCES

[1] E. Eade and T. Drummond, “Scalable monocular slam,” in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, vol. 1. IEEE, 2006, pp. 469–476.

[2] D. Strelow and S. Singh, “Motion estimation from image and inertial measurements,” *The International Journal of Robotics Research*, vol. 23, no. 12, pp. 1157–1195, 2004.

[3] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, “Bundle adjustmenta modern synthesis,” in *International workshop on vision algorithms*. Springer, 1999, pp. 298–372.

[4] 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.

[5] H. Strasdat, J. Montiel, and A. J. Davison, “Scale drift-aware large scale monocular slam,” *Robotics: Science and Systems VI*, 2010.

[6] R. Mur-Artal, J. Montiel, and J. D. Tardós, “Orb-slam: a versatile and accurate monocular slam system,” *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

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

[8] M. Kleinert and S. Schleith, “Inertial aided monocular slam for gps-denied navigation,” in *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2010 IEEE Conference on*. IEEE, 2010, pp. 20–25.

[9] G. Klein and D. Murray, “Parallel tracking and mapping for small ar workspaces,” in *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*. IEEE, 2007, pp. 225–234.

[10] 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.

[11] S. Lynen, M. Achtelik, S. Weiss, M. Chli, and R. Siegwart, “A robust and modular multi-sensor fusion approach applied to mav navigation,” in *Proc. of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2013.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] J. Sola, “Quaternion kinematics for the error-state kf,” *Laboratoire dAnalyse et dArchitecture des Systemes-Centre national de la recherche scientifique (LAAS-CNRS), Toulouse, France, Tech. Rep*, 2012.

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

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

[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*, 2016. [Online]. Available: http://ijr.sagepub.com/content/early/2016/01/21/0278364915620033.abstract

[20] 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.

[21] S. Birchfield, “Klt: An implementation of the kanade-lucas-tomasi feature tracker,” 2007.