ATLANTIS PRESS
Advances in Computer Science Research, volume 71
4th International Conference on Machinery, Materials and Information Technology Applications (ICMMITA 2016)

Keyframe-Based Tightly-Coupled SLAM with RGBD Camera and IMU

Yiming Zhang[1, a], Kui Li[2, b] and Wei Wang[1, c]

[1]Beihang University, Beijing 100191, China.

[2]State Key Laboratory of Geo-information Engineering, Xi'an Research Institute of Surveying and mapping, Xi'an 710054.

[a]zhangyiming@buaa.edu.cn, [b]eric.lee_buaa@buaa.edu.cn, [c]wongwei@buaa.edu.cn

Keywords: visual SLAM, IMU, keyframe-based, tightly coupled.

Abstract. In this paper, we propose a novel inertial-assisted slam system intended for low-cost micro aerial vehicles (MAVs). The system sensor assembly consists of color camera, depth camera and a inertial measurement unit (IMU) with three-axis accelerometers/gyroscopes. We use IMU data to enable probability-based predetermined operations rather than hypothesis testing iterations to accelerate pose calculation and obtain its optimal estimation through nonlinear optimization. We illustrate the performance of our system by hovering a MAV in a GPS-denied environment which position accuracy can reach 3.9cm RMSE in a 189 second flight and its robustness is also illustrated in a complex indoor environment.

Introduction

As portable computing devices, such as smart phones, smart glasses and other devices, become more ubiquitous, there is an interest to provide such devices with localization and mapping capabilities. Localization can be partially addressed by relying on systems that use global positioning system (GPS) signals or triangulation of cell tower signals to calculate position. However, such services tend to be limited to use outdoors, since GPS signals or cell tower signals may be blocked within buildings. Moreover, commercial localization and mapping services are generally unable to provide accuracy higher than several meters with respect to position.

Visual-based inertial navigation systems rely on information obtained from images and inertial measuring devices in order to achieve localization and mapping. Since visual-based inertial navigation systems do not require signals from GPS or cell towers, such systems may be used indoors where GPS and cell signals cannot reach or are unavailable due to obstacle or interference. Furthermore, visual-based inertial navigation systems enable very high position accuracy, e.g., on the order of centimeters. However, visual-based inertial navigation systems are typically computationally intensive as they need to process large amounts of image data acquired from an image detector, such as a camera, and inertial measurement unit (IMU), all in real-time. In addition, to achieve highly accurate measurements of position, a history of information related to previous poses (positions and orientations), inertial measurements and image features is typically stored, thus requiring device to use a substantial amount of memory and consequently large computation time to process this information.

System overview.

The visual-inertial fusion approaches found in the literature can be categorized to follow two approaches. In loosely-coupled systems, e.g. [1], the IMU measurements are incorporated as independent inclinometer and relative yaw measurements into the stereo vision optimization. Weiss et al. [2] use vision-only pose estimates as updates to an EKF with indirect IMU propagation. Also in [3], relative stereo pose estimates are integrated into a factor-graph containing inertial terms and absolute GPS measurements. Such methods limit the complexity, but disregard correlations amongst internal states of different sensors.

Fig 1 Synchronized RGB, depth and IMU hardware

Notation and Definitions. (1) Notation: We employ the following notation throughout this work: F_A denotes a reference frame A; vectors expressed in it are written as p_A or optionally as p_A^{BC}, with B and C as start and end points, respectively. A transformation between frames is represented by a homogeneous transformation matrix T_{AB} that transforms the coordinate representation of homogeneous points from F_B to F_A. Its rotation matrix part is written as C_A^B; the corresponding quaternion is written as $q_A^B = [\epsilon^T \quad \eta^T] \in S^3$, ϵ and η representing the imaginary and real parts. We adopt the notation introduced in Barfoot et al. [1]: concerning the quaternion multiplication $q_A^C = q_A^B \otimes q_B^C$. (2) Frames: The performance of the proposed method is evaluated using a stereo-camera/IMU setup schematically depicted in Figure 3. Inside the tracked body that is represented relative to an inertial frame, F_W, we distinguish camera frames, F_C, and the IMU-sensor frame, F_S.

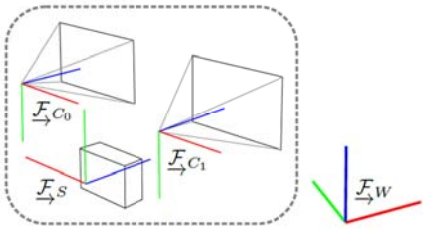

Fig 2 Coordinate frames involved in the hardware setup used: two cameras are placed as a stereo setup with respective frames F_C. IMU data is acquired in F_S,. F_S is estimated with respect to F_W.

(3) States: The variables to be estimated comprise the robot states at the image times (index k) $\boldsymbol{X}_R^k$ and landmarks $\boldsymbol{X}_L^c$. $\boldsymbol{X}_R$ holds the robot position in the inertial frame $\boldsymbol{P}_W^{WS}$, the body orientation quaternion $\boldsymbol{q}_{WS}$, the velocity in inertial frame $\boldsymbol{V}_W^{WS}$, as well as the biases of the gyroscopes $\boldsymbol{b}_g$ and the biases of the accelerometers $\boldsymbol{b}_a$. Thus, $\boldsymbol{X}_R$ is written as:

$$\boldsymbol{X}_R = \left[\boldsymbol{P}_W^{WS^T} \middle| \boldsymbol{q}_{WS}{}^T \middle| \boldsymbol{V}_W^{WS^T} \middle| \boldsymbol{b}_g{}^T \middle| \boldsymbol{b}_a{}^T\right]^T \in \mathbb{R}^3 \times S^3 \times \mathbb{R}^3 \tag{1}$$

Furthermore, we use a partition into the pose states, $\boldsymbol{X}_T = \left[\boldsymbol{P}_W^{WS^T} \middle| \boldsymbol{q}_{WS}{}^T\right]^T$ and the speed/bias states $\boldsymbol{X}_{SB} = \left[\boldsymbol{V}_W^{WS^T} \middle| \boldsymbol{b}_g{}^T \middle| \boldsymbol{b}_a{}^T\right]^T$. Landmarks are represented in homogeneous coordinates as in [4], in order to allow seamless integration of close and very far landmarks: $\boldsymbol{X}_L = \left[l_x \middle| l_y \middle| l_z \middle| l_w\right]^T \in \mathbb{R}^4$.

System Architecture. We try to descripted our system architecture as two parts, front-end and back-end. As shown in Figure 3, Front-end module is in charge of calculating the pose information between consequence frames. While back-end module is responsible for optimization. It contains bundle adjustment and maintains a keyframes package and feature points map.

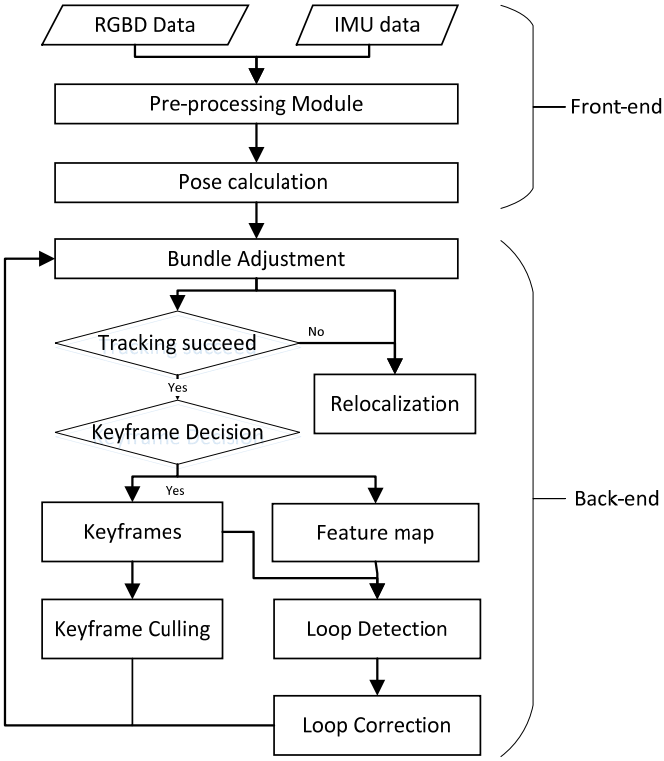

Fig 3 System architecture

IMU accelerated pose calculation

We get RGB, depth and synchronized IMU data from the hardware. According to the intrinsic and extrinsic parameters of RGB and depth camera, A RGB point cloud is generated. In the pre-process module, we successfully combine IMU, a FAST feature detector and a rotated BRIEF feature descriptor to approximate the rotation-invariant ORB features, used in large angular velocity estimation. The pipeline mainly consists of five modules: re-mapping, FAST feature scoring, image smoothing, non-maximal suppression and BRIEF descriptor generation.

At two successive sampling time, we combine the 3D position of a given feature point in the previous image with the motion from IMU to predict its 2D projection in the next image. Then we use a circle-window to search for the real corresponding feature point. For another matching scheme considering the rectified stereo frames at the same sampling time, we are able to search the corresponding features directly on the same row of the second image.

The 2D-3D correspondences are used for motion estimation, since 3D information can be recovered from the pointcloud. Each correspondence can be represented by $c_i = \{p_i, u_i\}$, where $p_i = [x_i, y_i, z_i]^T$ is the 3D coordinate in the previous frame and $u_i = [u_i, v_i, 1]^T$ is the 2D image (undistorted and rectified) pixel index in the current frame. Then we have:

$$\lambda_i u_i = K(Rp_i + t)$$
$$K = \begin{bmatrix} f_x & 0 & u_c \\ 0 & f_y & v_c \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

where K is the camera intrinsic parameters, R and t are the rotation and translation between two successive camera frames, and λ_i is the unknown scale factor.

Assume that we have n feature correspondences $c_1, c_2, \ldots, c_n$ and the exact rotation R from IMU readings. Let $R = [r_1, r_2, r_3]^T$ and $t = [t_x, t_y, t_z]^T$ respectively. Then we have $\lambda_i = r_3^T p_i + t_z$. Replace λ_i with t and we get:

$$\begin{bmatrix} (u_1 - u_c)r_3^T p_1 - f_x r_1^T p_1 \\ (v_1 - v_c)r_3^T p_1 - f_y r_2^T p_1 \\ \vdots \\ (u_n - u_c)r_3^T p_n - f_x r_1^T p_n \\ (v_n - v_c)r_3^T p_n - f_y r_2^T p_n \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_c - u_1 \\ 0 & f_y & v_c - v_1 \\ \vdots & \vdots & \vdots \\ f_x & 0 & u_c - u_n \\ 0 & f_y & v_c - v_n \end{bmatrix} t \tag{3}$$

Then we can obtain a least square solution to t with a minimal n of 2.

We shall now use IMU data to make the pose calculation more accuracy and fast. As IMU has a high orientation accuracy in short time. So the initial guess of rotation R is provided by IMU. Then we use Longest Successive Consistency (LONSC) method to reject outliers. We will explain the details of this algorithm and compare it to the traditional well-known outlier rejection method – RANSAC. The basic idea of LONSC is to estimate motion for every two successive correspondences stored in a 1D array and regard the longest successive motion-consistent sequence (SMS) as an inlier set. These inliers set will be used to estimate a motion model which can identify most of other inliers. The LONSC algorithm architecture is shown in Algorithm 1.

Algorithm 1 Motion estimation based on LONSC

Require:
The sequence of correspondences {c1, c2, ...,
The rotation R provided by IMU;
Ensure:
The set of inliers;
The estimated translation t;

```
MLen = MT ail = CLen = 0, T = 0, Φ = ∅;
for i = 2 to nc do
  if R, T and ci are consistent then
    CLen = CLen + 1;
    if CLen > MLen then
      MLen = CLen, MTail = i;
    end if
  else
    Use c_{i−1} and c_i to estimate T;
    CLen = 1;
  end if
end for
Use {c_{MTail−MLen+1}, ..., c_{MTail}} to estimate ^t̂;
for i = 1 to n_c do
  if R, t̂ and c_i are consistent then
    Add c_i to Φ;
  end if
end for
Use Φ to estimate t;
return Φ, t;
```

So far, the front-end get the relative rotation and translation between two consequence frames. As there is no costly RANSAC step involved—another advantage of tight IMU involvement. For the subsequent optimization, a bounded set of camera frames is maintained, i.e. poses with associated images taken at that time instant; all landmarks visible in these images are kept in the local map. As illustrated in Figure 4, we distinguish two kinds of frames: we introduce a temporal window of the S most recent frames including the current frame; and we use a number of N keyframes that may have been taken far in the past. For keyframe selection, we use a simple heuristic: if the ratio between the image area spanned by matched points versus the area spanned by all detected points falls below 50 to 60%, the frame is labeled keyframe.

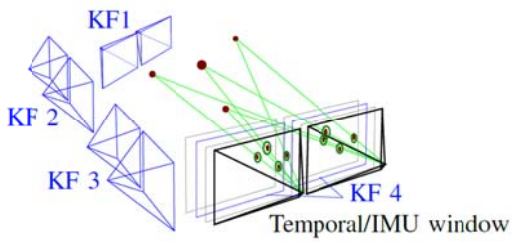

Fig 4 Frames kept for matching and subsequent optimization

Tight coupled nonlinear optimization

We seek to formulate the visual-inertial localization and mapping problem as one joint optimization of a cost function $J(X)$ containing both the (weighted) reprojection errors $\boldsymbol{e}_r$ and the temporal error term from the IMU $\boldsymbol{e}_s$:

$$J(x)=\sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{j\in O(i,k)} e_r^{i,j,k^T} W_r^{i,j,k} + \sum_{k=1}^{K-1} e_s^{k^T} W_s^k e_s^k \tag{4}$$

where k denotes the camera frame index, and j denotes the landmark index. The indices of landmarks visible in the kth frame are written as the set $\mathcal{J}(X)$. Furthermore, $\boldsymbol{W}_r^{i,j,k}$ represents the information matrix of the respective landmark measurement, and $\boldsymbol{W}_s^k$ the information of the kth IMU error.

Inherently, the purely visual SLAM has 6 Degrees of Freedom (DoF) that need to be held fixed during optimization, i.e. the absolute pose. The combined visual-inertial problem has only 4 DoF, since gravity renders two rotational DoF observable. This complicates fixation. We want to freeze yawing around the gravity direction (world z-axis), as well as the position, typically of the first pose (index k_1). Thus, apart from setting position changes to zero, $\delta\boldsymbol{p}_W^{WS^k} = 0_{3\times1}$, we also postulate $\delta\boldsymbol{\alpha}^{k1} = [\delta\alpha_1^{k1}|\delta\alpha_2^{k1}|0]^T$.

In the following, we will present the (standard) reprojection error formulation. Afterwards, an overview on IMU kinematics combined with bias term modeling is given, upon which we base the IMU error term.

1) Reprojection Error Formulation:

We use a rather standard formulation of the reprojection error adapted with minor modifications

$$\mathrm{e}_r^{i,j,k} = z^{i,j,k} - h_i\left(T_{C_iS}T_{SW}^k l_W^{WL,j}\right) \tag{5}$$

Hereby $h_i(\cdots)$ denotes the camera projection model and $z^{i,j,k}$ stands for the measurement image coordinates. The error Jacobians with respect to minimal disturbances follow directly.

2) IMU Kinematics

Under the assumption that the measured effects of the Earth's rotation is small compared to the gyroscope accuracy, we can write the IMU kinematics combined with simple dynamic bias models as:

$$\begin{aligned}
\dot{p}_W^{WS} &= v_W^{WS} \\
\dot{q}_{WS} &= \frac{1}{2}\Omega\left(\tilde{\omega}_S^{WS}, w_g, b_g\right)q_{WS} \\
\dot{v}_W^{WS} &= C_{WS}(\tilde{a}_W^{WS} + w_a - b_a) + g_W \\
\dot{b}_g &= w_{b_g} \\
\dot{b}_a &= -\frac{1}{\tau}b_a + w_{b_a}
\end{aligned} \tag{6}$$

where the elements $\boldsymbol{w} = \left[w_g^T, w_a^T, w_{b_g}^T, w_{b_a}^T\right]^T$ are each uncorrelated zero-mean Gaussian white noise processes. $\tilde{a}_W^{WS}$ are accelerometer measurements and g_W the Earth's gravitational acceleration vector. In contrast to the gyro bias modeled as random walk, we use the time constant $\tau > 0$ to model

the accelerometer bias as bounded random walk. The matrix Ω is formed from the estimated angular rate $\tilde{\omega}_S^{WS} = \omega_W^{WS} + w_g - b_g$, with gyro measurement $\tilde{\omega}_S^{WS}$.

So the linearized error dynamics take the form

$$\delta\dot{x}_R = F_c(\overline{x}_R)\delta x_R + G(\overline{x}_R)w \tag{7}$$

where G is straight-forward to derive and:

$$F_c = \begin{bmatrix} \mathbf{0}_{3\times3} & [\overline{\mathbf{C}}_{WS}\,{}_S\overline{\mathbf{v}}]^\times & \overline{\mathbf{C}}_{WS} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \overline{\mathbf{C}}_{WS} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & -\overline{\mathbf{C}}_{SW}[{}_W g]^\times & -[{}_S\varpi]^\times & -[{}_S\overline{v}]^\times & -I_3 \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\frac{1}{\tau}I_3 \end{bmatrix} \tag{8}$$

3) Formulation of the IMU Measurement Error Term

Due to the difference in measurement rates with camera measurements taken at time steps k and k + 1, as well as faster IMU-measurements that are not synchronized with the camera measurements in general. We need the IMU error term $e_s^k\left(X_R^k, X_R^{k+1}, Z_s^k\right)$ to be a function of robot states at steps k and k+1 as well as of all the IMU measurements in between these time instances (comprising accelerometer and gyro readings) summarized as Z_k^s. Hereby we have to assume an approximate normal conditional probability density f for given robot states at camera measurements k and k + 1:

$$f\left(e_s^k \middle| X_R^k, X_R^{k+1}\right) \approx \mathcal{N}(0, R_s^k) \tag{9}$$

For the state prediction $\hat{X}_R^{k+1}(X_R^k, Z_k^s)$ with associated conditional covariance $P\left(\delta\hat{X}_R^{k+1} \middle| X_R^k, Z_s^k\right)$, the IMU prediction error term can now be written as:

$$e_s^k\left(X_R^k, X_R^{k+1}, Z_s^k\right) = \begin{bmatrix} \hat{P}_W^{WS^{k+1}} - P_W^{WS^{k+1}} \\ 2\left[\hat{q}_{WS}^{k+1} \otimes q_{WS}^{k+1^{-1}}\right]_{1:3} \\ \hat{X}_{sb}^{k+1} - X_{sb}^{k+1} \end{bmatrix} \in \mathbb{R}^{15} \tag{10}$$

This is simply the difference between the prediction based on the previous state and the actual state—except for orientation, where we use a simple multiplicative minimal error.

Experiment Results

We present experimental results using a custom-built sensor prototype as shown in Figure, which provides VGA color images and dual IR camera depth camera with 6 cm baseline synchronized to the IMU (BMX055) measurements. The proposed method runs in real-time for all experiments on a standard laptop (2.2 GHz Quad-Core Intel Core i5, 8 Gb RAM). We use g2o [5] as an optimization framework. A precise intrinsic and extrinsic calibration of the camera with respect to the IMU using [6] was available beforehand. The IMU characteristics used (Table I) are slightly more conservative than specified.

Table 1 IMU CHARACTERISTICS

Gyroscopes			Accelerometers		
RMS	360	°/h	RMS	4	mg
RNSD	0.014	°/s/hz@10hz	PSD	150	ug/hz
Resolution	0.004	°/s	Resolution	0.98	mg

We adopt the evaluation scheme of [7]: for many starting times, the ground truth and estimated trajectories are aligned and the error is evaluated for increasing distances travelled from there. Our tightly-coupled algorithm is evaluated against ground truth. To ensure that only the estimation algorithms are being compared, we fix the feature correspondences for all algorithms to the ones derived from the tightly-coupled approach.

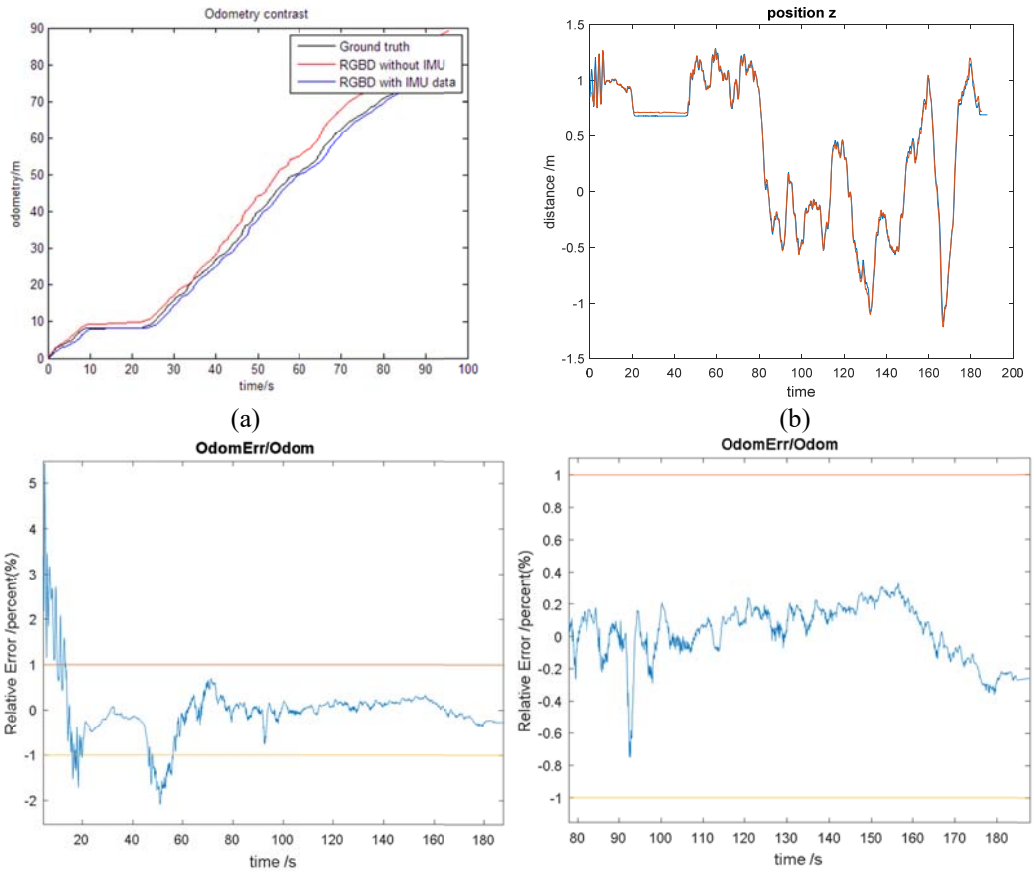

Fig 5 experiment result from a 300m^2 room flight

Table 2 Statistical Error Analysis

RMSE	Mean	Medium	Std	Min	Max
3.93cm	0.3cm	-0.73	3.63cm	-10.1cm	17.1cm

As an experiment, the sensor is fastened on a quad-drone while flying in a 300 m^2 room. This sequence exhibits challenging high dynamic, lighting and texture conditions while flying through lights and box. The odometry comparison plot in Figure demonstrates the applicability of the proposed method in such scenarios with a loop-closure error of 17.1 cm, and its stand error can reach 3.63 cm. As can be seen from figure 5 (a), our tightly coupled visual-IMU SLAM perform better than pure visual SLAM.

Conclusion

This paper presents a method of tightly integrating inertial measurements into keyframe-based visual SLAM. The combination of error terms in the non-linear optimization is motivated by error statistics available for both keypoint detection and IMU readings—thus superseding the need for any tuning parameters. Using the proposed approach, we obtain global consistency of the gravity direction and robust outlier rejection employing the IMU kinematics motion model. At the same time, all the benefits of keyframe based nonlinear optimization are obtained, such as pose keeping in stand-still. Results obtained using a stereo-camera and IMU sensor demonstrate real-time operation of the proposed framework while exhibiting increased accuracy and robustness over vision-only or a loosely coupled approach.

Acknowledgments

This research is supported and funded by the National Natural Science Foundation of China (L142200032) and Long-term development strategic research of china Engineering Science and

Technology (2014-zxq-01), the State Key Laboratory of Geo-Information Engineering (NO.SKLGIE2015-M-2-3). The authors would appreciate the support and fund.

References

[1] K. Konolige, M. Agrawal, and J. Sola. Large-scale visual odometry for rough terrain. In Robotics Research, pages 201–212. Springer, 2011.

[2] S. Weiss, M.W. Achtelik, S. Lynen, M. Chli, and R. Siegwart. Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments. In Proc. of the IEEE International Conference on Robotics and Automation (ICRA), 2012

[3] A. Ranganathan, M. Kaess, and F. Dellaert. Fast 3d pose estimation with out-of-sequence measurements. In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2007.

[4] P. T. Furgale. Extensions to the Visual Odometry Pipeline for the Exploration of Planetary Surfaces. PhD thesis, University of Toronto, 2011.

[5] R. K¨ummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2011.

[6] P. T. Furgale, J. Rehder, and R. Siegwart. Unified temporal and spatial calibration for multi-sensor systems. In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2013.

[7] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012.