2016 IEEE International Conference on Robotics and Automation (ICRA)
Stockholm, Sweden, May 16-21, 2016

Visual-Inertial Direct SLAM

Alejo Concha[1], Giuseppe Loianno[2], Vijay Kumar[2], and Javier Civera[1]

Abstract— **The so-called direct visual SLAM methods have shown a great potential in estimating a semidense or fully dense reconstruction of the scene, in contrast to the sparse reconstructions of the traditional feature-based algorithms. In this paper, we propose for the first time a direct, tightly-coupled formulation for the combination of visual and inertial data.**

Our algorithm runs in real-time on a standard CPU. The processing is split in three threads. The first thread runs at frame rate and estimates the camera motion by a joint non-linear optimization from visual and inertial data given a semidense map. The second one creates a semidense map of high-gradient areas only for camera tracking purposes. Finally, the third thread estimates a fully dense reconstruction of the scene at a lower frame rate. We have evaluated our algorithm in several real sequences with ground truth trajectory data, showing a state-of-the-art performance.

I. INTRODUCTION

Simultaneous Localization and Mapping (usually referred by its acronym SLAM) has become a key technology for several potential applications like robotics, autonomous cars and augmented and virtual reality. Its goal is to estimate, from a stream of sensor data, a model of the surroundings of the sensor and its egomotion with respect to it.

In the latest decades, there has been an intense research on the use of visual sensors for SLAM, but its application has been limited by the sparsity of the scene estimation. The traditional techniques –denoted as feature-based– rely on the correspondences between image point features; that can only be reliably established for the image points that present a high degree of saliency. [1], [2] are two open-source examples of such feature-based monocular SLAM systems.

Recently, [3]–[5], have developed algorithms for real-time, online and dense scene reconstruction from monocular images. These algorithms are based, in contrast to the feature-based methods, on the direct minimization of the photometric pixel values and on a regularization function. These so-called direct methods are able to produce semidense and dense reconstructions and open the doors to a wider applicability of visual SLAM. On the other hand, their maturity is still low. For example, it was recently shown that their current accuracy is lower than the feature-based techniques [6].

Our contribution within direct SLAM is a novel algorithm for the fusion of inertial and photometric data. To our knowledge, this is the first algorithm that integrates the inertial information in direct visual SLAM in a jointly manner. The key advantage of the mono(cular)-inertial configuration with respect to the pure monocular one is the recovery of the real scale of the scene and the camera motion. This might be essential in certain applications, for example in robotics and control. We also believe that the compactness, low cost and low consumption of the hardware and the dense reconstructions we achieve might be relevant for constrained systems like quadrotors. We validated the accuracy and real-time performance of our algorithm in real visual-inertial sequences with Vicon ground truth for the trajectory.

The rest of the paper is organized as follows. Section II describes the related work. Section III presents our visual-inertial fusion for camera tracking with the Jacobians of the optimization approach detailed in the appendix VII. Section IV describes the mapping algorithm we use in our paper. Finally, section V shows the experimental results and section VI presents the conclusions.

II. RELATED WORK

SLAM from a monocular camera has a scale ambiguity that limits its use in certain applications. A usual configuration to overcome that is a stereo camera [7], [8]. However, the accuracy of the scale is limited by the ratio between the scene depth and the stereo baseline. Feature-based visual motion estimation has been also combined with wheel odometry (e.g., in [9]) and non-holonomic constraints [10], but such combination is constrained to terrestrial robots. Visual-GPS SLAM (e.g., [11]) also resolves the scale ambiguity, but it is constrained to outdoors.

The visual-inertial fusion also resolves the scale ambiguity in pure monocular SLAM. Such fusion is usually divided into two classes. Loosely-coupled approaches model the two inputs as independent (e.g., [12]). Tightly-coupled ones, that estimate jointly all the states, have proven a superior performance [13], [14] at an additional complexity in the optimization –but still within real time in modern computers. The recent works [15], [16] present two loosely-coupled direct stereo visual-inertial systems. In the first system, the inertial and stereo measurements are loosely-coupled fused with respect to the last keyframe, and in the latest one, they are fused in a –loosely coupled– pose-graph manner. Our algorithm is the first tightly-coupled one for direct SLAM methods. Also differently from [15], [16], we use monocular vision instead of stereo vision.

III. VISUAL-INERTIAL JOINT OPTIMIZATION

The main idea is to take advantage of the fast rate of the Inertial Measurement Unit (IMU) that provides inertial data and propose a joint optimization where the integration of IMU measurements is able to infer additional constraints

[1]The authors are with the I3A, Universidad de Zaragoza, Spain {alejocb, jcivera}@unizar.es

[2]The authors are with the GRASP Lab, University of Pennsylvania, 3330 Walnut Street, 19104 Phialdelphia, USA. {loiannog, kumar}@seas.upenn.edu

between camera poses. The IMU can contribute to speed up the estimation and obtain the velocity and the absolute scale information, which is not available using only the camera information.

A. Integration of IMU Measurements

Let us consider, without loss of generality, that the IMU reference frame is coincident with the camera frame.

The rotation $\boldsymbol{R}_j^w$, translation $\boldsymbol{t}_j^w$ and velocity $\boldsymbol{v}_j^w$ of the current frame j in the world reference frame w are calculated from a previous frame i using an Euler forward integration.

$$\boldsymbol{R}_j^w = \boldsymbol{R}_i^w \boldsymbol{R}_j^i \tag{1}$$

$$\boldsymbol{v}_j^w = \boldsymbol{v}_i^w + \boldsymbol{v}_{ij}^w \tag{2}$$

$$\boldsymbol{t}_j^w = \boldsymbol{t}_i^w + \boldsymbol{t}_{ij}^w \tag{3}$$

Where $\boldsymbol{R}_j^i$ is the relative rotation between the frames i and j, $\boldsymbol{v}_{ij}^w$ is the incremental velocity from i to j and $\boldsymbol{t}_{ij}^w$ is the translation vector between i and j. These three integrate the IMU measurements as follows

$$\boldsymbol{R}_j^i = \prod_{p=k}^{k+N-1} \exp_{SO(3)}\left(\left[\boldsymbol{\omega}(p) + \boldsymbol{b}_\omega(p)\right]^\wedge T\right) \tag{4}$$

$$\boldsymbol{v}_{ij}^w = \sum_{p=k}^{k+N-1} \left(\boldsymbol{R}_p^w\left(\boldsymbol{a}(p) + \boldsymbol{b}_a(p)\right) - \boldsymbol{g}\right) T \tag{5}$$

$$\begin{aligned}\boldsymbol{t}_{ij}^w = {} & N\boldsymbol{v}_i^w T + \\ & \frac{1}{2}\sum_{p=k}^{k+N-1}\left(2\left(k+N-1-p\right)+1\right)\left(\boldsymbol{R}_p^w\left(\boldsymbol{a}(p)+\boldsymbol{b}_a(p)\right)-\boldsymbol{g}\right)T^2\end{aligned} \tag{6}$$

The IMU measurements at each step p are the angular velocity $\boldsymbol{\omega} \in \mathbb{R}^3$ and the linear acceleration $\boldsymbol{a} \in \mathbb{R}^3$ in the local frame. These measurements are affected by the biases $\boldsymbol{b}_\omega \in \mathbb{R}^3$ –for the angular velocity– and $\boldsymbol{b}_a \in \mathbb{R}^3$ –for the linear acceleration. T denotes the time step size, $\boldsymbol{g} \in \mathbb{R}^3$ the gravity vector, k the time step of frame i and $k+N$ the time step of frame j. The $\exp_{SO(3)}$ operator maps an element of $so(3)$ to an element of $SO(3)$. The inverse operation is denoted with $\log_{SO(3)}$.

The *hat* operator $(\cdot^\wedge)$ is used to convert a 3×1 vector in a an element of of $so(3)$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}^\wedge = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}, \tag{7}$$

while the opposite operation is performed using the *vee* operator $(\cdot^\vee)$

$$\begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}^\vee = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}. \tag{8}$$

The IMU biases are modeled as random walk processes with variances $\boldsymbol{\eta}_a$ and $\boldsymbol{\eta}_\omega$ for the acceleration $\boldsymbol{a}$ and angular velocity $\boldsymbol{\omega}$ respectively

$$\boldsymbol{b}_a(k+1) = \boldsymbol{b}_a(k) + T\boldsymbol{\eta}_a, \tag{9}$$

$$\boldsymbol{b}_\omega(k+1) = \boldsymbol{b}_\omega(k) + T\boldsymbol{\eta}_\omega. \tag{10}$$

They frame i is assumed to be fixed and only the current frame j is optimized. This assumption is made due to the computational cost of direct visual SLAM. Specifically, without this assumption, the Jacobians of the photometric error have to be computed in every iteration and we could not take advantage of the inverse compositional approach.

B. State vector and residual

The state vector $\boldsymbol{x}$ is composed of 15 variables:

$$\boldsymbol{x} = \begin{bmatrix} \left[\log_{SO(3)}(\boldsymbol{R}_j^w)^\vee\right]^\top & \boldsymbol{t}_j^{w\top} & \boldsymbol{v}_j^{w\top} & \boldsymbol{b}_\omega^\top & \boldsymbol{b}_a^\top \end{bmatrix}^\top, \tag{11}$$

The residual $\boldsymbol{r}$ to minimize is divided into two parts, referred to as the photometric $\boldsymbol{r}_{ph}$ and IMU –inertial– $\boldsymbol{r}_{imu}$ residual

$$\boldsymbol{r} = \begin{bmatrix} \boldsymbol{r}_{ph}^\top & \boldsymbol{r}_{imu}^\top \end{bmatrix}^\top \tag{12}$$

The IMU residual is

$$\boldsymbol{r}_{imu} = \begin{bmatrix} \log_{SO(3)}\left(\boldsymbol{R}_j^{i\top}\left(\boldsymbol{R}_i^w\right)^\top \boldsymbol{R}_j^w\right)^\vee \\ \left(\boldsymbol{t}_j^w - \boldsymbol{t}_{ij}^w - \boldsymbol{t}_i^w\right) \\ \left(\boldsymbol{v}_j^w - \boldsymbol{v}_{ij}^w - \boldsymbol{v}_i^w\right) \\ \boldsymbol{b}_\omega^i - \boldsymbol{b}_\omega \\ \boldsymbol{b}_a^i - \boldsymbol{b}_a \end{bmatrix} \tag{13}$$

The intensity residual is

$$\boldsymbol{r}_{ph} = \sum_{k=1}^{n} \boldsymbol{I}_i\left(\pi\left(\boldsymbol{T}_w^i \boldsymbol{p}_t^k\right)\right) - \boldsymbol{I}_j\left(\pi\left(\boldsymbol{T}_w^j \boldsymbol{p}_t^k\right)\right) \tag{14}$$

$$\boldsymbol{T}_w^j = \begin{bmatrix} \boldsymbol{R}_w^j & \boldsymbol{t}_w^j \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix}, \tag{15}$$

where π is the pinhole camera model and $\boldsymbol{p}_t$ is each one of the n 3D points tracked from i to j. $\boldsymbol{I}_i$ is the image taken at step k and $\boldsymbol{I}_j$ the image taken at step $k+N$.

C. Visual-Inertial Direct Tracking

The visual-inertial thread minimizes the residual of eq (12) using standard Gauss-Newton optimization

$$\boldsymbol{J}^\top \boldsymbol{\Lambda} \boldsymbol{J} \boldsymbol{\delta} = -\boldsymbol{J}^\top \boldsymbol{\Lambda} \cdot \boldsymbol{r}, \tag{16}$$

where $\boldsymbol{\delta}$ are the updates for the pose and the IMU $\boldsymbol{\delta} = \left[\boldsymbol{\delta}_P^\top, \boldsymbol{\delta}_I^\top\right]^\top$ and $\boldsymbol{\Lambda}$ is the inverse covariance. The general Jacobian $\boldsymbol{J}$ is composed of the Jacobian of the IMU residual with respect to the IMU parameters (biases and velocity) $\boldsymbol{J}_I^{r_{imu}}$, the Jacobian of the IMU error with respect to the current pose of the camera $\boldsymbol{J}_P^{r_{imu}}$ and the Jacobian of the photometric error with respect to the current pose $\boldsymbol{J}_P^{r_{ph}}$. Note than $\boldsymbol{J}_I^{r_{ph}} = 0$. $\boldsymbol{J}_P^{r_{ph}}$ will be defined in section III-D and $\boldsymbol{J}_I^{r_{imu}}$ and $\boldsymbol{J}_P^{r_{imu}}$ will be defined in the appendix (see section VII).

The full Jacobian is formed as follows

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}_P^{r_{ph}} & 0 \\ \boldsymbol{J}_P^{r_{imu}} & \boldsymbol{J}_I^{r_{imu}} \end{bmatrix}, \tag{17}$$

and reparametrized for convenience as

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}_P^r & \boldsymbol{J}_I^r \end{bmatrix}, \tag{18}$$

where $\boldsymbol{J}_P^r = \begin{bmatrix} \boldsymbol{J}_P^{r_{ph}} \\ \boldsymbol{J}_P^{r_{imu}} \end{bmatrix}$ and $\boldsymbol{J}_I^r = \begin{bmatrix} \boldsymbol{J}_I^{r_{ph}} \\ \boldsymbol{J}_I^{r_{imu}} \end{bmatrix}$.

To solve it, we perform Gauss-Newton optimization and use the Schur complement to accelerate it. The Jacobian $\boldsymbol{J}$ yields the following Hessian approximation

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{J}_P^{r\top} \boldsymbol{\Lambda} \boldsymbol{J}_P^r & \boldsymbol{J}_P^{r\top} \boldsymbol{\Lambda} \boldsymbol{J}_I^r \\ \boldsymbol{J}_P^r \boldsymbol{\Lambda} \boldsymbol{J}_I^{r\top} & \boldsymbol{J}_I^{r\top} \boldsymbol{\Lambda} \boldsymbol{J}_I^r. \end{bmatrix} \tag{19}$$

The linear system is therefore

$$\begin{bmatrix} \boldsymbol{H}_{PP}^r & \boldsymbol{H}_{PI}^r \\ \boldsymbol{H}_{PI}^{r\top} & \boldsymbol{H}_{II}^r \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}_P \\ \boldsymbol{\delta}_I \end{bmatrix} = \begin{bmatrix} \boldsymbol{B}_P \\ \boldsymbol{B}_I \end{bmatrix}, \tag{20}$$

where $\boldsymbol{B}_I = -\boldsymbol{J}_I^{r\top} \boldsymbol{\Lambda} r$ and $\boldsymbol{B}_P = -\boldsymbol{J}_P^{r\top} \boldsymbol{\Lambda} r$, $\boldsymbol{\delta}_I$ and $\boldsymbol{\delta}_P$ are the updates for the IMU and pose parameters respectively.

The system is solved using the Schur complement rearrangement. The lower part of the system of the eq. (20) results in the smaller equation system for the IMU update

$$\left(\boldsymbol{H}_{II} - \boldsymbol{H}_{PI}^{r\top} {\boldsymbol{H}_{PP}^r}^{-1} \boldsymbol{H}_{PI}^r \right) \boldsymbol{\delta}_I = \boldsymbol{B}_I - \boldsymbol{H}_{PI}^{r\top} {\boldsymbol{H}_{PP}^r}^{-1} \boldsymbol{B}_P, \tag{21}$$

This linear system is typically solved using Cholesky factorization or SVD decomposition. Once the update for the IMU parameters is found we can solve the upper system for the pose update:

$$\boldsymbol{\delta}_P = {\boldsymbol{H}_{PP}^r}^{-1} \boldsymbol{B}_P - {\boldsymbol{H}_{PP}^r}^{-1} \boldsymbol{H}_{PI}^r \boldsymbol{\delta}_I. \tag{22}$$

D. Visual Tracking Jacobian

The transformation from the current camera frame to the global reference frame $\boldsymbol{T}_w^j$ is estimated based on the photometric reprojection error $\boldsymbol{r}_{ph}$ using the inverse compositional approach [17]. The photometric error for the 3D points $\boldsymbol{p}_t$ is

$$\boldsymbol{r}_{ph} = \sum_{k=1}^{n} \left(\boldsymbol{I}_i \left(\pi \left(\hat{\boldsymbol{T}} \boldsymbol{T}_w^i \boldsymbol{p}_t^k \right) \right) - \boldsymbol{I}_j \left(\pi \left(\boldsymbol{T}_w^j \boldsymbol{p}_t^k \right) \right) \right). \tag{23}$$

The tracking thread only uses a subset of image points, composed of high-gradient points –and superpixel contours if desired. Section IV details how the 3D position for those is obtained. We seek to estimate the transformation $\hat{\boldsymbol{T}}$ from the closest keyframe $\boldsymbol{I}_i$ to the current frame $\boldsymbol{I}_j$. The seed for T_w^j comes from the raw inertial readings before the optimization.

For the optimization we use a minimal parametrization for the camera pose. The rotation $\boldsymbol{R}$ is mapped to the tangent space *so(3)* of the rotation group *SO(3)* at the identity. The increments for the camera pose –the angular increment $\delta\boldsymbol{\omega}$ and the increment for the translation $\delta\boldsymbol{t}$– are defined as follows:

$$\hat{\boldsymbol{T}} = \begin{bmatrix} \exp_{SO(3)}([\delta\boldsymbol{\omega}]^\wedge) & \delta\boldsymbol{t} \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix}, \tag{24}$$

The update for the current camera pose $\boldsymbol{T}_w^j$ is as follows

$$\boldsymbol{T}_w^j = \hat{\boldsymbol{T}}^{-1} \boldsymbol{T}_w^j. \tag{25}$$

The Jacobian of the photometric residual $\boldsymbol{J}_P^{r_{ph}}$ is obtained via the chain rule:

$$\boldsymbol{J}_P^{r_{ph}} = \frac{\partial \boldsymbol{r}_{ph}}{\partial \boldsymbol{\varepsilon}} = \boldsymbol{J}_F^{r_{ph}} \boldsymbol{J}_{T_w^i}^F \boldsymbol{J}_{\varepsilon}^{T_w^i}, \tag{26}$$

where $\boldsymbol{J}_F^{r_{ph}}$ are the image gradients of the reference keyframe, $\boldsymbol{J}_{T_w^i}^F$ is the derivative of the projection model with respect to $\boldsymbol{T}_w^i$ and $J_{\varepsilon}^{\boldsymbol{T}_w^i}$ is the derivative of the transformation $\boldsymbol{T}_w^i$ with respect to the motion $\boldsymbol{\varepsilon} = (\delta\boldsymbol{\omega}, \delta\boldsymbol{t})$.

Using the inverse compositional approach the Jacobians are always referred to the last keyframe, not being updated until a new frame becomes a keyframe –$\boldsymbol{T}_w^i$, $\boldsymbol{p}_t$ and $\boldsymbol{J}_F^{r_{ph}}$ are constant during the optimization. This accelerates significantly the camera tracking.

To bootstrap our system we integrate the inertial measurements and we estimate the first map –obtaining the real scale– using the approach of section IV. After the initialization, we apply the joint optimization explained in the section III.

IV. Mapping

A. Semidense Mapping

Rapid camera motions require high-frequency map updates for the camera to be tracked successfully. Similarly to [18]–[20], our system maintains a semidense map of high-gradient pixels that is quickly updated and serves for camera tracking. The semidense map is not only used for tracking, but also for the estimation of the fully dense map described in section IV-B.

The inverse depth ρ for every high-gradient pixel $\boldsymbol{u}^*$ in a keyframe $\boldsymbol{I}_i$ is estimated by minimizing its photometric error $\boldsymbol{r}_{ph}^o$ with respect to several overlapping views $\boldsymbol{I}_o$. The specific optimization is

$$\hat{\rho} = \arg\min_{\rho} \boldsymbol{r}_{ph}^o, \tag{27}$$

with

$$\boldsymbol{r}_{ph}^o = \sum_o \| (\boldsymbol{I}_i(\boldsymbol{s}_{u^*}) - \boldsymbol{I}_o(G(\boldsymbol{s}_{u^*}, \boldsymbol{T}_i, \boldsymbol{T}_o, \rho))) \|_2^2. \tag{28}$$

$\boldsymbol{s}_{u^*}$ are the pixel coordinates of the template (7x7 pixels square) around the pixel $\boldsymbol{u}^*$ and G is the function that backprojects the template from the new keyframe $\boldsymbol{I}_i$ to the 3D world and projects it back to each overlapping image $\boldsymbol{I}_o$.

The reader is referred to [20] for more details on the semidense mapping algorithm used in this paper.

B. Superpixels Mapping

We use the algorithm in [20] to extract the piecewise planar textureless regions to fill the semidense map of the previous section. First, we segment a keyframe $\boldsymbol{I}_i$ into a set of superpixels $\mathscr{S}_k = \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_i, \ldots, \boldsymbol{s}_m\}$ using the algorithm of [21].

Each 3D point $\boldsymbol{p}_t^k$ from the semidense map is then projected on the keyframe $u^* = F(\boldsymbol{T}_w^i \boldsymbol{p}_t^k)$. The 3D points $\boldsymbol{p}_t^k$ are assigned to the superpixels if their projections $\boldsymbol{u}^*$ lie within a threshold ξ of the superpixel contour.

The 3D points associated to the contour of every superpixel $\boldsymbol{p}_t^k \in \mathscr{C}(\boldsymbol{s}_i)$ are used to robustly fit a plane π_i using singular value decomposition.

Again, we refer the reader to [20] for more details on the 3D superpixels estimation.

V. EXPERIMENTS

A. Experimental setup

The experimental tests have been performed in the GRASP Lab [22] at The University of Pennsylvania. The considered working area is a volume of $5 \times 4 \times 5$ m^3. The Vicon motion capture system[1], composed of 20 T040 cameras, provides a ground truth for our rotation and translation estimates [22].

The camera-IMU system setup is composed by Matrix Vision mvBlueFOX-MLC[2] camera and a Microstrain 3DM-GX4-25 IMU[3] as shown in Figure 1. The image processing runs on 752×480 pixels at 30 Hz, whereas the IMU rate has been set to 100 Hz. The entire solution has been developed using the ROS[4] framework. The visual-inertial direct tracking thread (section III-C) runs at 30 Hz while the 2 mapping threads (section IV) run at a lower rate. The computer used is a 3.5 GHz Intel Core i7-3770K processor and 8.0 GB of RAM memory.

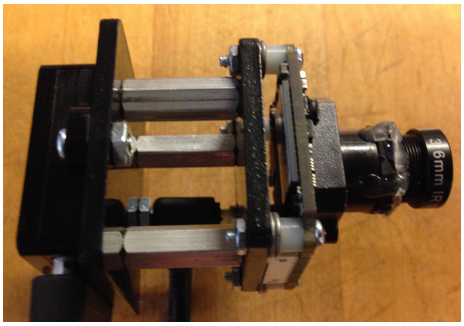


Fig. 1: The sensor setup used for the experiments.

B. Results

We have evaluated our algorithm in 10 different sequences of visual and inertial data. The sequences were recorded with a hand-held camera while the first author walked around the room. Their duration goes from 20 to 55 seconds.

Table I shows the mean translational errors. Notice how the errors are of the same order than those reported for direct SLAM methods (for example, LSD-SLAM in [19] or in the results of [6]). But in our case we are able to recover the real scale of the scene and the trajectory thanks to the inertial fusion. For further details, figures 2 and 3 show respectively the angular and translational errors for each coordinate on the 10 experiments.

Experiment	1	2	3	4	5	6	7	8	9	10
Error [cm]	6.6	6.3	11.6	6.3	9.3	10.6	7.4	3.2	10.8	6.2

TABLE I: Mean trajectory error.

Figure 4 shows the computational cost of our algorithm –vertical axis– for every frame –horizontal axis– and for every sequence –in different colors. 99% of the frames have a computational cost under 33 milliseconds, which is our frame period. We attribute the cost spikes in our results to other high priority tasks on the processor (we do not use a real-time OS). In any case, the sequences were played in real time, showing that our algorithm is resilient to a few lost frames.

Figure 5 shows some results of our algorithm. The top row

[1] http://www.vicon.com
[2] http://www.matrix-vision.com/
[3] http://www.microstrain.com/
[4] http://www.ros.org/

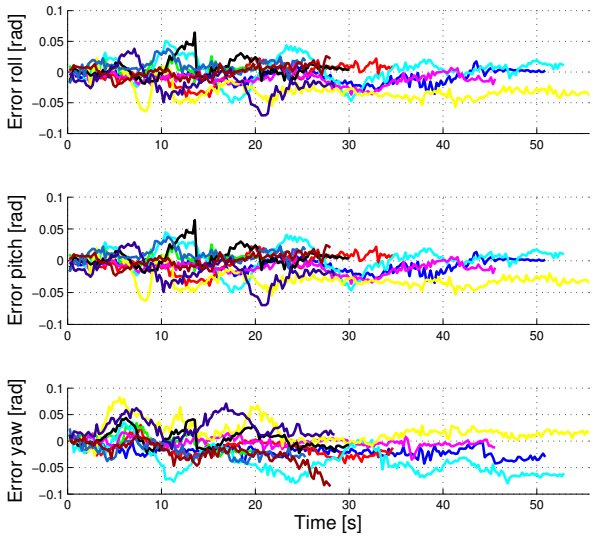


Fig. 2: Angular error for our 10 sequences. Each color is a different sequence.

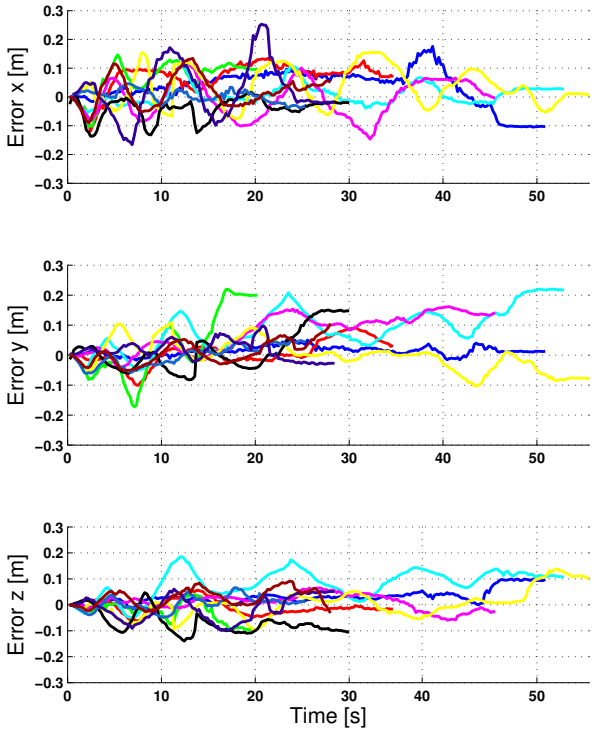


Fig. 3: Translational error for our 10 sequences. Each color is a different sequence.

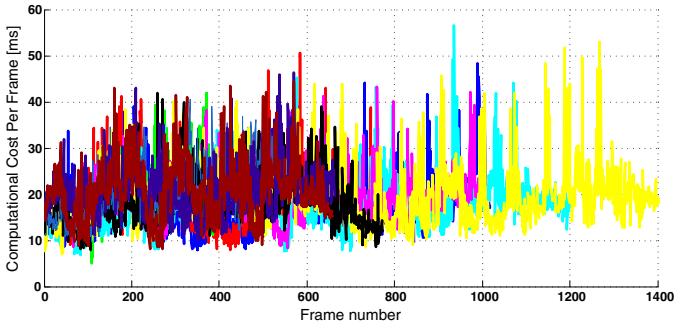


Fig. 4: Computational cost for our 10 sequences. Each color is a different sequence.

shows four sample images, and the bottom row several views of the estimated reconstruction and the camera trajectory in red.

Finally, figures 6 and 7 illustrates the difference between the semidense and dense maps estimated by our algorithm. The top row in both images displays again sample frames from the sequence, and the bottom row a close view of the semidense and dense maps estimated by our algorithm. Notice in the figures 6(e) and 7(e) that the semidense map only contains the high gradient regions of the images, being of little use for some applications like robotic navigation. Notice the higher point density of the dense map in the figures 6(f) and 7(f) , where the textureless areas have been reconstructed using a piecewise planar prior.

As shown in [23], a TV-regularization of a photometric map produces low quality results in low texture scenes. We also run a TV-regularization on the experiments of this paper and the qualitative accuracy of the maps was lower than the piecewise-planar approach we use.

VI. CONCLUSIONS

We have presented in this paper a tightly-coupled visual-inertial SLAM algorithm for real-time, online, dense scene reconstruction and camera tracking. Our main contribution is the joint optimization of the IMU measurements and visual data using direct methods. Our full SLAM system is divided into 3 threads; a low-cost mapping thread produces semi-dense reconstructions for real-time camera tracking, a camera tracking thread responsible for the visual-inertial fusion, and a high-quality dense mapping thread producing dense maps at low frame rate. Our dense mapping algorithm uses a piecewise planar approximation based on 3D superpixels.

We have validated the accuracy and real-time performance of our approach with several real sequences, comparing the estimated pose with respect to a Vicon motion capture system. To our knowledge this is the first real-time, tightly-coupled, direct visual SLAM approach. Given the low cost and compactness of the mono-inertial hardware setting, our approach can be relevant for low-cost, low-consumption, lightweight robots like some quadrotors, or for augmented and virtual reality applications.

VII. APPENDIX

A. Background

The following result relates infinitesimal increments in so(3) with right hand-multiplications [14]:

$$\exp_{SO(3)}\left([\boldsymbol{\theta}+\delta\boldsymbol{\theta}]^{\wedge}\right)=\exp_{SO(3)}\left([\boldsymbol{\theta}]^{\wedge}\right)\exp_{SO(3)}\left([\boldsymbol{J}_r(\boldsymbol{\theta})\delta\boldsymbol{\theta}]^{\wedge}\right) \tag{29}$$

where $\boldsymbol{J}_r(\boldsymbol{\theta})$ is the $SO(3)$ Jacobian.

$$\boldsymbol{J}_r(\boldsymbol{\theta})=\boldsymbol{I}-\frac{1-cos(||\boldsymbol{\theta}||)}{||\boldsymbol{\theta}||^2}\boldsymbol{\theta}^{\wedge}+\frac{||\boldsymbol{\theta}||-sin(||\boldsymbol{\theta}||)}{||\boldsymbol{\theta}||^3}(\boldsymbol{\theta}^{\wedge})^2 \tag{30}$$

A similar first-order approximation holds for the logarithm as shown in [14]

$$\log_{SO(3)}\left(\exp_{SO(3)}\left([\boldsymbol{\theta}]^{\wedge}\right)\exp_{SO(3)}\left([\delta\boldsymbol{\theta}]^{\wedge}\right)\right)^{\vee}=\boldsymbol{\theta}+\boldsymbol{J}_r(\boldsymbol{\theta})^{-1}\delta\boldsymbol{\theta}. \tag{31}$$

The following relation holds for SO(3)

$$\exp_{SO(3)}\left([\boldsymbol{\theta}]^{\wedge}\right)\boldsymbol{R}=\boldsymbol{R}\exp_{SO(3)}\left(\left[\boldsymbol{R}^{\top}\boldsymbol{\theta}\right]^{\wedge}\right) \tag{32}$$

B. Jacobians

a) **Jacobians of the IMU residual with respect to the camera pose $J_P^{r_{imu}}$:**
Definition of the rotation error with respect to the increment

$$\begin{aligned}&\boldsymbol{r}_{imu}[1]\left(\boldsymbol{R}_j^w\exp_{SO(3)}\left(\left[\delta\boldsymbol{\theta}_j\right]^{\wedge}\right)\right)=\log_{SO(3)}\left(\boldsymbol{R}_j^{i\top}\boldsymbol{R}_i^{w\top}\boldsymbol{R}_j^w\exp_{SO(3)}\left(\left[\delta\boldsymbol{\theta}_j\right]^{\wedge}\right)\right)^{\vee}=\\&\log_{SO(3)}\left(\left(\prod_{p=k}^{k+N-1}\exp_{so(3)}\left([\boldsymbol{\omega}(p)+\boldsymbol{b}_{\boldsymbol{\omega}}(p)]^{\wedge}T\right)\right)^{\top}\boldsymbol{R}_i^{w\top}\boldsymbol{R}_j^w\exp_{SO(3)}\left(\left[\delta\boldsymbol{\theta}_j\right]^{\wedge}\right)\right)^{\vee}\end{aligned} \tag{33}$$

By $\boldsymbol{r}_{imu}[1]$ we denote the first term of the IMU error (the rotation error in this case). Rearranging terms and using eq. (31)

$$\begin{aligned}&\boldsymbol{r}_{imu}[1]\left(\boldsymbol{R}_j^w\exp_{SO(3)}\left(\left[\delta\boldsymbol{\theta}_j\right]^{\wedge}\right)\right)=\boldsymbol{r}_{imu}[1](\boldsymbol{R}_j^w)\\&+\boldsymbol{J}_r(\boldsymbol{r}_{imu}[1](\boldsymbol{R}_j^w))^{-1}\delta\boldsymbol{\theta}_j\end{aligned} \tag{34}$$

$$\frac{\partial\boldsymbol{r}_{imu}[1]\left(\boldsymbol{R}_j^w\exp_{SO(3)}\left(\left[\delta\boldsymbol{\theta}_j\right]^{\wedge}\right)\right)}{\partial\delta\boldsymbol{\theta}_j}=\boldsymbol{J}_r(\boldsymbol{r}_{imu}[1](\boldsymbol{R}_j^w))^{-1}. \tag{35}$$

Note that $\frac{\partial\boldsymbol{r}_{imu}[2]\left(\boldsymbol{R}_j^w\exp_{SO(3)}\left(\left[\delta\boldsymbol{\theta}_j\right]^{\wedge}\right)\right)}{\partial\delta\boldsymbol{\theta}_j}=\frac{\partial\boldsymbol{r}_{imu}[3]\left(\boldsymbol{R}_j^w\exp_{SO(3)}\left(\left[\delta\boldsymbol{\theta}_j\right]^{\wedge}\right)\right)}{\partial\delta\boldsymbol{\theta}_j}=\frac{\partial\boldsymbol{r}_{imu}[4]\left(\boldsymbol{R}_j^w\exp_{SO(3)}\left(\left[\delta\boldsymbol{\theta}_j\right]^{\wedge}\right)\right)}{\partial\delta\boldsymbol{\theta}_j}=\frac{\partial\boldsymbol{r}_{imu}[5]\left(\boldsymbol{R}_j^w\exp_{SO(3)}\left(\left[\delta\boldsymbol{\theta}_j\right]^{\wedge}\right)\right)}{\partial\delta\boldsymbol{\theta}_j}=\boldsymbol{0}_{3\times3}$.

For the translation, we define the translation error $\boldsymbol{r}_{imu}[2]$ with respect to the SE(3) increment

$$\boldsymbol{r}_{imu}[2]\left(\boldsymbol{t}_j^w+\boldsymbol{R}_j^w\delta\boldsymbol{t}_j^w\right)=\left(\boldsymbol{t}_j^w+\boldsymbol{R}_j^w\delta\boldsymbol{t}_j^w-\boldsymbol{t}_{ij}-\boldsymbol{t}_i^w\right) \tag{36}$$

$$\frac{\partial\boldsymbol{r}_{imu}[2](\boldsymbol{t}_j^w+\boldsymbol{R}_j^w\delta\boldsymbol{t}_j^w)}{\partial\delta\boldsymbol{t}_j^w}=\boldsymbol{R}_j^w \tag{37}$$

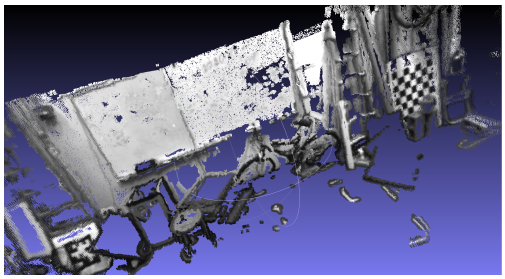
Note that $\frac{\partial\boldsymbol{r}_{imu}[1]\left(\boldsymbol{t}_j^w+\boldsymbol{R}_j^w\delta\boldsymbol{t}_j^w\right)}{\partial\delta\boldsymbol{t}_j^w}=\frac{\partial\boldsymbol{r}_{imu}[3]\left(\boldsymbol{t}_j^w+\boldsymbol{R}_j^w\delta\boldsymbol{t}_j^w\right)}{\partial\delta\boldsymbol{t}_j^w}=\frac{\partial\boldsymbol{r}_{imu}[4]\left(\boldsymbol{t}_j^w+\boldsymbol{R}_j^w\delta\boldsymbol{t}_j^w\right)}{\partial\delta\boldsymbol{t}_j^w}=\frac{\partial\boldsymbol{r}_{imu}[5]\left(\boldsymbol{t}_j^w+\boldsymbol{R}_j^w\delta\boldsymbol{t}_j^w\right)}{\delta\boldsymbol{t}_j^w}=\boldsymbol{0}_{3\times3}$.

(a) Sample frame

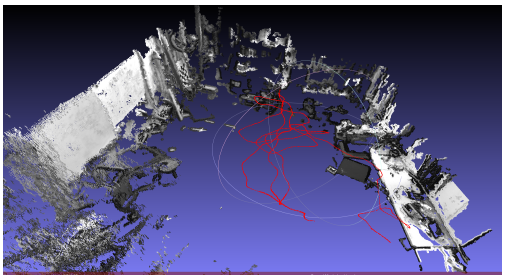
(b) Sample frame

(c) Sample frame

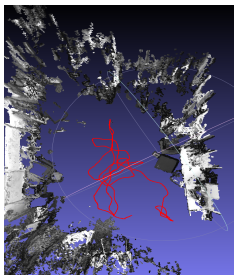
(d) Sample frame



(e) Close view.



(f) Further view.



(g) Top view.

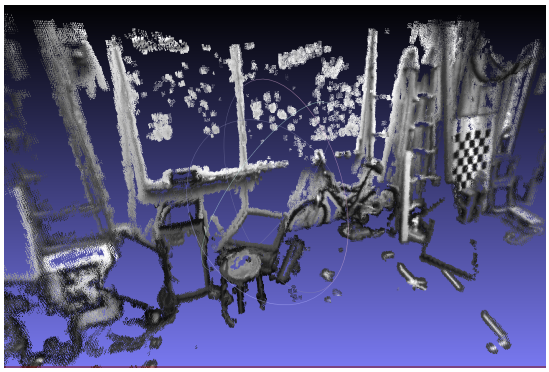
Fig. 5: Sample images and views of the reconstructed scene for the largest of our experiments, covering almost the whole GRASP Lab Vicon room. The estimated trajectory of the camera is shown in red. Notice, qualitatively, the accuracy and density of the estimated map.

(a) Sample frame

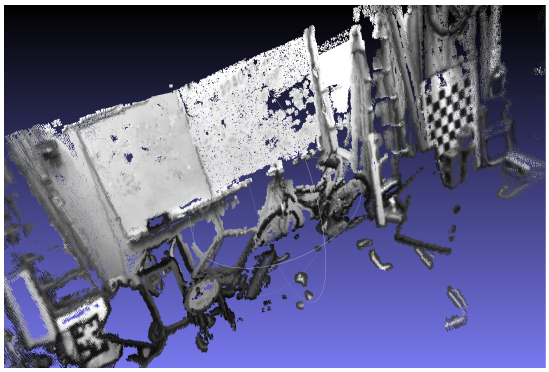
(b) Sample frame

(c) Sample frame

(d) Sample frame



(e) Semidense map



(f) Piecewise-planar semidense map.

Fig. 6: Sample frames for one of our experiments, and close-ups of the semidense map used for camera tracking and the piecewise planar dense map. Notice how, in the latest, the low-textured areas have been correctly reconstructed.

b) **Jacobian of the IMU residual with respect to IMU parameters $J_I^{r_{imu}}$:**

- Jacobian with respect to the velocity in the frame j

$$\frac{\partial \boldsymbol{r}_{imu}[3]\left(\boldsymbol{v}_j^w+\delta v_j\right)}{\partial \delta \boldsymbol{v}_j^w}=\boldsymbol{I}_{3\times 3}. \tag{38}$$

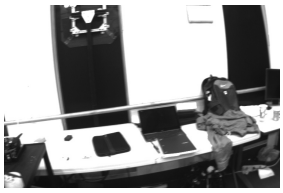
Note that $\frac{\partial \boldsymbol{r}_{imu}[1](\boldsymbol{v}_j^w+\delta \boldsymbol{v}_j^w)}{\partial \delta \boldsymbol{v}_j^w} = \frac{\partial \boldsymbol{r}_{imu}[2](\boldsymbol{v}_j^w+\delta \boldsymbol{v}_j^w)}{\partial \delta v_j^w} = \frac{\partial \boldsymbol{r}_{imu}[4](\boldsymbol{v}_j^w+\delta \boldsymbol{v}_j^w)}{\partial \delta \boldsymbol{v}_j^w} = \frac{\partial \boldsymbol{r}_{imu}[5](\boldsymbol{v}_j^w+\delta \boldsymbol{v}_j^w)}{\partial \delta \boldsymbol{v}_j^w} = \boldsymbol{0}_{3\times 3}.$

- Jacobian of the rotation error $\boldsymbol{r}_{imu}[1]$ with respect to the angular velocity bias $\boldsymbol{b}_\omega$

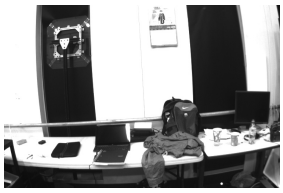
$$\boldsymbol{r}_{imu}[1](\boldsymbol{b}_w+\delta \boldsymbol{b}_w)= \log_{SO(3)}\left(\left(\prod_{p=k}^{k+N-1}\exp_{SO(3)}\left(\left(\boldsymbol{\omega}(p)+\boldsymbol{b}_{\boldsymbol{\omega}}(p)+\delta \boldsymbol{b}_{\boldsymbol{\omega}}\right)^{\wedge}T\right)\right)^{\top}\boldsymbol{R}_i^{w\top}\boldsymbol{R}_j^w\right)^{\vee}. \tag{39}$$

Applying eq. (29) in the increments and using eq. (32) to move the all increment terms to the

(a) Sample frame.

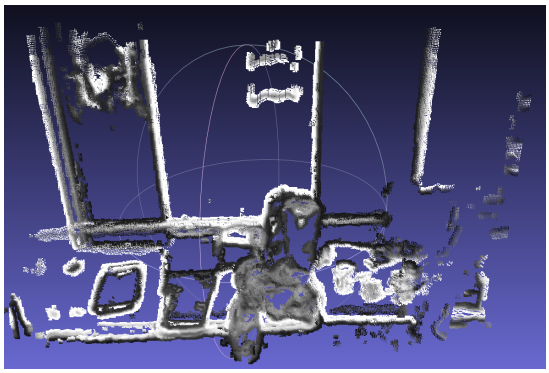


(b) Sample frame.

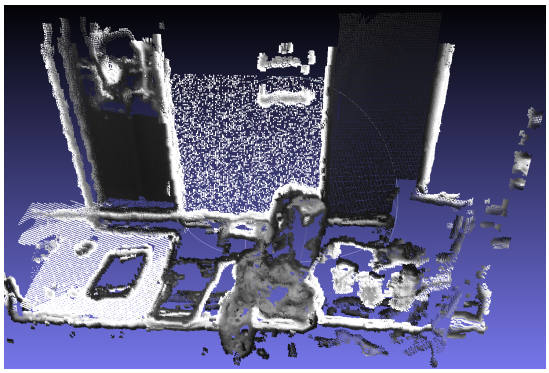


(c) Sample frame.

(d) Sample frame.



(e) Semidense map.



(f) Piecewise-planar semidense map.

Fig. 7: Sample frames for one of our experiments, and close-ups of the semidense map used for camera tracking and the piecewise planar dense map. Notice how, in the latest, the low-textured areas have been correctly reconstructed.

right side, assuming that the increments are small $\exp_{SO(3)}\left([\boldsymbol{a}]^{\wedge}\right)\exp_{SO(3)}\left([\boldsymbol{b}]^{\wedge}\right)=\exp_{SO(3)}\left([\boldsymbol{a}+\boldsymbol{b}]^{\wedge}\right)$, rearranging terms and applying eq. (31) we obtain

$$\begin{aligned}&\boldsymbol{r}_{imu}[1]\left(\boldsymbol{b}_w+\delta\boldsymbol{b}_\omega\right)=\\&\boldsymbol{r}_{imu}[1]-\boldsymbol{J}_r\left(\boldsymbol{r}_{imu}[1]\right)^{-1}*\exp_{SO(3)}\left([\boldsymbol{r}_{imu}[1]]^{\wedge}\right)^{\top}\boldsymbol{A}\delta\boldsymbol{b}_w\end{aligned}\tag{40}$$

where

$$\boldsymbol{A}=\sum_{p=k}^{k+N-1}\left[\left[\prod_{m=p+1}^{k+N-1}\exp_{SO(3)}\left([\boldsymbol{\omega}(m)+\boldsymbol{b}_\omega(m)]^{\wedge}T\right)\right]^{\top}\boldsymbol{J}_r^p*T\right]\tag{41}$$

$$\frac{\partial\boldsymbol{r}_{imu}[1]\left(\boldsymbol{b}_\omega+\delta\boldsymbol{b}_\omega\right)}{\partial\delta\boldsymbol{b}_\omega}=-\boldsymbol{J}_r(\boldsymbol{r}_{imu}[1])^{-1}*\exp_{SO(3)}\left([\boldsymbol{r}_{imu}[1]]^{\wedge}\right)^{\top}\boldsymbol{A}\tag{42}$$

Where

$$\boldsymbol{J}_r^p=\boldsymbol{J}_r\left((\boldsymbol{\omega}(p)+\boldsymbol{b}_\omega(p))T\right).$$

- Jacobian of the position error $\boldsymbol{r}_{imu}[2]$ with respect to the angular velocity bias $\boldsymbol{b}_w$. Applying eq. (29) in the increments and using eq. (32) to move all increment terms to the right side, assuming that the increments are small $\exp_{SO(3)}\left([\boldsymbol{a}]^{\wedge}\right)\exp_{SO(3)}\left([\boldsymbol{b}]^{\wedge}\right)=\exp_{SO(3)}\left([\boldsymbol{a}+\boldsymbol{b}]^{\wedge}\right)$, using a first order approximation for the exponential $\exp_{SO(3)}\left([\boldsymbol{a}]^{\wedge}\right)=\boldsymbol{I}_{3\times3}+[\boldsymbol{a}]^{\wedge}$ and rearranging terms we obtain

$$\frac{\partial\boldsymbol{r}_{imu}[2]\left(\boldsymbol{b}_\omega+\delta\boldsymbol{b}_\omega\right)}{\partial\delta\boldsymbol{b}_\omega}=\frac{1}{2}\sum_{p=k}^{k+N-1}\boldsymbol{C}\left(\boldsymbol{a}(p)+\boldsymbol{b}_a(p)\right)^{\wedge}\boldsymbol{B}*T^2\tag{43}$$

$$\boldsymbol{B}=\sum_{l=k}^{p-1}\left[\left[\prod_{m=l+1}^{p-1}\exp_{SO(3)}\left([\boldsymbol{\omega}(m)+\boldsymbol{b}_\omega(m)]^{\wedge}T\right)\right]^{\top}\boldsymbol{J}_r^l*T\right]\tag{44}$$

$$\boldsymbol{C}=\prod_{q=k}^{p-1}(2(k+N-1-p)+1)\boldsymbol{R}_i^w\exp_{SO(3)}\left([\boldsymbol{\omega}(q)+\boldsymbol{b}_\omega(q)]^{\wedge}T\right)$$

- Jacobian of the velocity error $\boldsymbol{r}_{imu}[3]$ with respect to the angular velocity bias $\boldsymbol{b}_w$. Similarly to the previous Jacobian we obtain

$$\frac{\partial\boldsymbol{r}_{imu}[3]\left(\boldsymbol{b}_\omega+\delta\boldsymbol{b}_\omega\right)}{\partial\delta\boldsymbol{b}_\omega}=\sum_{p=k}^{k+N-1}\boldsymbol{D}\left(\boldsymbol{a}(p)+\boldsymbol{b}_a(p)\right)^{\wedge}\boldsymbol{B}*T\tag{45}$$

$$\boldsymbol{D}=\prod_{q=k}^{p-1}\boldsymbol{R}_i^w\exp_{SO(3)}\left[(\boldsymbol{\omega}(q)+\boldsymbol{b}_\omega(q)]^{\wedge}T\right)$$

- Jacobian of the angular velocity bias error $\boldsymbol{r}_{imu}[4]$ with respect to the angular velocity bias $\boldsymbol{b}_w$

$$\frac{\partial\boldsymbol{r}_{imu}[4]\left(\boldsymbol{b}_\omega+\delta\boldsymbol{b}_\omega\right)}{\partial\delta\boldsymbol{b}_\omega}=-\boldsymbol{I}_{3\times3}\tag{46}$$

- Jacobian of the acceleration bias error $\boldsymbol{r}_{imu}[5]$ with respect to the angular velocity bias $\boldsymbol{b}_\omega$

$$\frac{\partial\boldsymbol{r}_{imu}[5](\boldsymbol{b}_\omega+\delta\boldsymbol{b}_\omega)}{\partial\delta\boldsymbol{b}_\omega}=\boldsymbol{0}_{3\times3}\tag{47}$$

- Jacobian of the rotation error $\boldsymbol{r}_{imu}[1]$ with respect to the acceleration bias $\boldsymbol{b}_a$

$$\frac{\partial\boldsymbol{r}_{imu}[1]\left(\boldsymbol{b}_\omega+\delta\boldsymbol{b}_\omega\right)}{\partial\delta\boldsymbol{b}_\omega}=\boldsymbol{0}_{3\times3}\tag{48}$$

- Jacobian of the position error $\boldsymbol{r}_{imu}[2]$ with respect to the acceleration bias $\boldsymbol{b}_a$

$$\frac{\partial \boldsymbol{r}_{imu}[2](\boldsymbol{b}_a+\delta\boldsymbol{b}_a)}{\partial \delta \boldsymbol{b}_a} = -\frac{1}{2}\sum_{p=k}^{k+N-1}(2(k+N-1-p)+1)\left(\boldsymbol{R}_p^w\right)T^2 \tag{49}$$

- Jacobian of the velocity error $\boldsymbol{r}_{imu}[3]$ with respect to the acceleration bias $\boldsymbol{b}_a$

$$\frac{\partial \boldsymbol{r}_{imu}[3](\boldsymbol{b}_a+\delta\boldsymbol{b}_a)}{\partial \delta \boldsymbol{b}_a} = -\sum_{p=k}^{k+N-1}\left(\boldsymbol{R}_p^w\right)T, \tag{50}$$

- Jacobian of the angular velocity bias error $\boldsymbol{r}_{imu}[4]$ with respect to the acceleration bias $\boldsymbol{b}_a$

$$\frac{\partial \boldsymbol{r}_{imu}[4](\boldsymbol{b}_a+\delta\boldsymbol{b}_a)}{\partial \delta \boldsymbol{b}_a} = \boldsymbol{0}_{3\times 3} \tag{51}$$

- Jacobian of the acceleration bias error $\boldsymbol{r}_{imu}[5]$ with respect to the acceleration bias $\boldsymbol{b}_a$

$$\frac{\partial \boldsymbol{r}_{imu}[5](\boldsymbol{b}_a+\delta\boldsymbol{b}_a)}{\partial \delta \boldsymbol{b}_a} = -\boldsymbol{I}_{3\times 3} \tag{52}$$

ACKNOWLEDGMENTS

This research has been partially funded by the Spanish government (projects DPI2012-32168 and DPI2015-67275), the Aragón regional government (Grupo DGA T04-FSE) and the University of Zaragoza (JIUZ-2015-TEC-03). In addition, this work was supported by the ARL grant W911NF-08-2-0004, ONR grants N00014-07-1-0829, N00014-14-1-0510, N00014-09-1-1051, N00014-09-1-103, NSF grant IIS-1426840, IIS-1138847, DARPA grants HR001151626, HR0011516850, and TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA

REFERENCES

[1] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2007.

[2] R. Mur-Artal, J. M. M. 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, October 2015*, 2015.

[3] J. Stühmer, S. Gumhold, and D. Cremers, "Real-time dense geometry from a handheld camera," in *Pattern Recognition*, 2010, pp. 11–20.

[4] G. Graber, T. Pock, and H. Bischof, "Online 3d reconstruction using convex optimization," in *2011 IEEE International Conference on Computer Vision Workshops*, 2011, pp. 708–711.

[5] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *2011 IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 2320–2327.

[6] R. Mur-Artal and J. D. Tardós., "Probabilistic semi-dense mapping from highly accurate feature-based monocular slam," in *Robotics: Science and Systems*, 2015.

[7] C. Mei, G. Sibley, M. Cummins, P. M. Newman, and I. D. Reid, "A constant-time efficient stereo slam system." in *BMVC*, 2009, pp. 1–11.

[8] T. Pire, T. Fischer, J. Civera, P. D. Cristóforis, and J. J. Berlles, "Stereo parallel tracking and mapping for robot localization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015.

[9] J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel, "1-point ransac for EKF filtering: Application to real-time structure from motion and visual odometry," *Journal of Field Robotics*, vol. 27, no. 5, pp. 609–631, October 2010.

[10] D. Scaramuzza, "1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints," *International journal of computer vision*, vol. 95, no. 1, pp. 74–85, 2011.

[11] D. Schleicher, L. M. Bergasa, M. Ocaña, R. Barea, and E. López, "Real-time hierarchical gps aided visual slam on urban environments," in *ICRA'09. IEEE International Conference on Robotics and Automation, 2009.* IEEE, 2009, pp. 4381–4386.

[12] 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 *2012 IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 957–964.

[13] 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.

[14] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation," in *IEEE Robotics: Science and Systems*, 2015.

[15] S. Omari, M. Bloesch, P. Gohl, and R. Siegwart, "Dense visual-inertial navigation system for mobile robots," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 2634–2640.

[16] L. Ma, J. M. Falquez, S. McGuire, and G. Sibley, "Large scale dense visual inertial slam," in *International Symposium on Experimental Robotics*, 2015.

[17] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *Int. J. Comput. Vision*, vol. 56, no. 3, pp. 221–255, 2004.

[18] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[19] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular slam," in *ECCV 2014*, 2014, pp. 834–849.

[20] A. Concha and J. Civera, "DPPTAM: Dense piecewise-planar tracking and mapping from a monocular sequence," in *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems"*, 2015.

[21] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[22] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The Grasp Multiple Micro–UAV Test Bed," *IEEE Robotics and Automation Magazine*, vol. 17, no. 3, pp. 56–65, 2010.

[23] A. Concha, W. Hussain, L. Montano, and J. Civera, "Manhattan and piecewise-planar constraints for dense monocular mapping," in *Robotics:Science and Systems*, 2014.