Camera Motion Estimation from RGB-D-Inertial Scene Flow

Samuel Cerezo, Javier Civera
I3A, Universidad de Zaragoza
{samueladriancerezo,jcivera}@unizar.es

Abstract

In this paper, we introduce a novel formulation for camera motion estimation that integrates RGB-D images and inertial data through scene flow. Our goal is to accurately estimate the camera motion in a rigid 3D environment, along with the state of the inertial measurement unit (IMU). Our proposed method offers the flexibility to operate as a multi-frame optimization or to marginalize older data, thus effectively utilizing past measurements. To assess the performance of our method, we conducted evaluations using both synthetic data from the ICL-NUIM dataset and real data sequences from the OpenLORIS-Scene dataset. Our results show that the fusion of these two sensors enhances the accuracy of camera motion estimation when compared to using only visual data.

1. Introduction

Autonomous navigation plays a key role in enabling robots and various other applications, including mixed reality and autonomous driving. For that, precise motion estimates derived from onboard sensors are essential. And, in this context, scene flow stands as one of the fundamental techniques for motion estimation using RGB-D or range sensors [7, 9, 24, 30, 33]. More specifically, scene flow refers to the estimation of the 3D motion field of scene points obtained from two sensor readings [31]. Although optical and scene flow have been used in numerous tasks over the years, such as motion compensation [34], object tracking [21] and object learning [22], we focus in this paper on the previously mentioned application of scene flow to odometry, *i.e.*, the estimation of the camera motion. We are also motivated by multisensor odometry and SLAM, which boost monocular-only approaches with extra accuracy and robustness, highly relevant in safety-critical robotic setups. Multisensor configurations have been widely explored in feature-based odometry and SLAM, e.g., stereo cameras [20], visual-inertial [3] or LiDAR-inertial [26], but are less explored in direct approaches that use the raw sensor measurements without feature extraction.

On the one hand, RGB-D cameras provide a practical hardware alternative to several challenges and limitations of visual odometry. Their availability at low cost has facilitated many robotics and Augmented Reality (AR) applications in the last decade. Today, RGB-D cameras stand out as one of the preferred sensors for indoor applications in robotics and AR; and their future looks promising either on their own or in combination with additional sensors. On the other hand, most commercial mobile devices are equipped with Inertial Measurement Units (IMU), which can provide large amount of information in dynamic trajectories but exhibit large drift due to noises if not fused with other information. This makes the visual-range-inertial fusion relevant, as the three modalities offer complementary characteristics.

Our contribution in this paper is a RGB-D-inertial formulation for camera motion estimation in rigid scenes. Up to our knowledge, this is the first time that inertial data is fused together with color and depth measurements to estimate camera motion based on optical flow. Specifically, we propose a tightly coupled optimization by minimizing pre-integrated inertial residuals and range constraints. As the inertial states are common between frames, we formulate the problem as a multi-frame optimization, in which past frame's states can be estimated or marginalized out into prior residuals for the inertial states. We evaluate our proposal in the synthetic ICL-NUIM dataset and in the real OpenLORIS-Scene one. The effectiveness of our fusion is shown by an error reduction of RGB-D-inertial estimation compared to RGB-D one.

2. Related Work

The first tracking system for ego-motion estimation which fuses vision and inertial measurements was presented by Armesto *et al*. [1]. In this case, the fusion is performed by considering a EKF and UKF (Extended and Unscented Kalman Filters) with multi-rate sampling of measurements. The mentioned sampling modality allows the system to work with the different rates of the sensors. In 2013, Kerl *et al*. [10] proposed a fast and accurate method to estimate the camera motion from RGB-D images. This approach es-

timates the relative motion between two consecutive frames by minimizing the photometric error. A motion prior is incorporated in the optimization, in order to guide and stabilize motion estimation in the presence of dynamic objects. Nießner *et al*. [17] developed an approach that improves the robustness of real-time 3D surface reconstruction by incorporating inertial sensing to the inter-frame alignment. As a result, they could significantly reduce the number of Iterative Closest Point (ICP) iterations required per frame. Modeling three-dimensional scene motion as a twist field, Quiroga *et al*. [23] introduced a method that encourages piecewise smooth solutions of rigid body motions. A general formulation is given to solve local and global rigid motions by jointly using intensity and depth data.

The first method to compute dense scene flow in real-time for RGB-D cameras was introduced in 2015 by Jaimez *et al*. [8]. They proposed a variational formulation where brightness and geometric consistency are imposed. Their accuracy outperforms that of previous for RGB-D flow baselines, being able to estimate non-rigid motions at 30Hz of frame rate. In the same year, Leutenegger *et al*. [15] formulated a probabilistic cost function that combines reprojection errors of landmarks with inertial terms, using stereo and monocular cameras. On the other hand, a new dense method to compute the odometry of a range sensor in real time is presented [7]. This method applies the range flow constraint equation in order to obtain the velocity of the sensor in a rigid environment. Experiments show that this approach overperforms GICP which uses the same geometric input data, whereas it achieves results similar to RDVO, which requires both geometric and photometric data.

The first tightly-coupled dense RGB-D-inertial SLAM system was proposed in 2017 by Laidlow *et al*. [13]. This system jointly optimises the camera pose, velocity, IMU biases and gravity direction while building up a globally consistent, fully dense surfel-based 3D reconstruction of the environment. In 2019, Shan *et al*. introduced VINS-RGBD [27]. The authors integrate a mapping system based on depth data and octree filtering to achieve real-time mapping. However the proposed system is applied only in ground robots. A RGB-D scene flow estimation method with global nonrigid and local rigid motion assumption is proposed by Li *et al*. in [16]. 3D motion is estimated based on the global non-rigid and local rigid assumption and spatial-temporal correlation of RGBD information. With this assumption, the interaction of motion from different parts in the same segmented region is avoided, especially the non-rigid object, e.g., a human body.

The flow formulation has been adapted to novel sensor modalities, e.g., event cameras [25], or to include additional information, such as robot dynamics in the work of Lee *et al*. [14]. Similarly to ours, the motivation in this last case is improving the robustness and accuracy of the camera motion estimation. Zhai *et al*. [32] compiled in a survey recent advances on optical and scene flow. Up to our knowledge, inertial sensing has never been integrated in flow formulations. Our work contributes to the literature presenting the first camera motion estimation from RGB-D-inertial scene flow, demonstrating its effectiveness in simulated and real public datasets.

3. Notation

Throughout this article, bold lower-case letters ($\mathbf{x}$) represent vectors and bold upper-case letters ($\mathbf{\Sigma}$) matrices. Scalars will be represented by light lower-case letters (α), scalar functions and images by light upper-case letters (J). Camera poses are represented as $\mathbf{T}_{WB} = [\mathbf{R}_{WB}, {}^{W}\mathbf{p}] \in SE(3)$ and transform points from frame B to world coordinate system W.

4. IMU Model and Motion Integration

4.1. Inertial preintegration

An IMU consists typically of an accelerometer and a three-axis gyroscope, and measures the angular velocity ${}^{B}\boldsymbol{\omega}$ and linear acceleration ${}^{B}\mathbf{a}$ of the sensor in the body reference frame B. We will denote the IMU measurement at time k as ${}^{B}\tilde{\boldsymbol{\omega}}_k$ and ${}^{B}\tilde{\mathbf{a}}_k$. IMU measurements are affected by additive white noise $\boldsymbol{\eta}^g$, $\boldsymbol{\eta}^a \in \mathbb{R}^3$ and two slowly varying gyroscope and accelerometer bias $\mathbf{b}^g$ and $\mathbf{b}^a \in \mathbb{R}^3$ respectively. Finally, the acceleration measurement is affected by gravity ${}^{W}\mathbf{g}$. This model is formulated by Eq. (1) and (2).

$$ {}^{B}\tilde{\boldsymbol{\omega}}_k = {}^{B}\boldsymbol{\omega}_k + \mathbf{b}^g_k + \boldsymbol{\eta}^g_k \tag{1} $$

$$ {}^{B}\tilde{\mathbf{a}}_k = \mathbf{R}^{\top}_{WB}\left({}^{W}\mathbf{a}_k - {}^{W}\mathbf{g}\right) + \mathbf{b}^a_k + \boldsymbol{\eta}^a_k \tag{2} $$

We use pre-integrated inertial residuals as proposed by Forster *et al.* [4]. We compute an inital guess for $\mathbf{b}^g$ as the difference of an estimate ${}^{B}\boldsymbol{\omega}_k$ of the angular velocity between two consecutive frames, and the direct measurement of the gyroscope ${}^{B}\tilde{\boldsymbol{\omega}}_k$ that includes the bias, *i.e.*, $\hat{\mathbf{b}}^g_k = {}^{B}\tilde{\boldsymbol{\omega}}_k - {}^{B}\boldsymbol{\omega}_k$, where the angular velocity estimate ${}^{B}\boldsymbol{\omega}_k$ can be computed by relative motion estimation between point clouds, divided by the time increment between them. We set the initial seed for $\mathbf{b}^a_k$ to zero.

Following [4], we can use the relative motion increment $\Delta\mathbf{v}_{ij} \doteq \mathbf{R}^{\top}_i(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij})$ in order to obtain a first gravity vector estimation

$$ \hat{\mathbf{g}} = \frac{\mathbf{v}_j - \mathbf{v}_i}{\Delta t_{ij}} - \frac{\mathbf{R}_i \Delta\mathbf{v}_{ij}}{\Delta t_{ij}} \tag{3} $$

4.2. Noise propagation

The covariance matrix of the raw IMU measurements noise $\mathbf{\Sigma}_{\boldsymbol{\eta}} \in \mathbb{S}_+^6$ [1] is composed by sub-matrices $\mathbf{\Sigma}_{\boldsymbol{\omega}}, \mathbf{\Sigma}_{\mathbf{a}} \in \mathbb{S}_+^3$

$$\mathbf{\Sigma}_{\boldsymbol{\eta}} = \left[\begin{array}{c|c} \mathbf{\Sigma}_{\boldsymbol{\omega}} & \mathbf{0}_{3\text{x}3} \\ \hline \mathbf{0}_{3\text{x}3} & \mathbf{\Sigma}_{\mathbf{a}} \end{array}\right] \tag{4}$$

where $\mathbf{0}_{3\text{x}3}$ is a 3x3 matrix for which all its elements are equal to zero.

Following the computation of the preintegrated noise covariance in [4], we consider the matrix $\boldsymbol{\eta}_{ik}^{\Delta} \doteq \left[\delta\boldsymbol{\phi}_{ik}^\top, \delta\mathbf{v}_{ik}^\top, \delta\mathbf{p}_{ik}^\top\right]^\top$ and defining the IMU measurement noise $\boldsymbol{\eta}_k^d \doteq \left[\boldsymbol{\eta}_k^{gd}, \boldsymbol{\eta}_k^{ad}\right]^\top$, the noise is propagated as

$$\mathbf{\Sigma}_{ij} = \mathbf{A}_{j-1}\mathbf{\Sigma}_{ij-1}\mathbf{A}_{j-1}^\top + \mathbf{B}_{j-1}\mathbf{\Sigma}_{\boldsymbol{\eta}}\mathbf{B}_{j-1}^\top \tag{5}$$

with initial conditions $\mathbf{\Sigma}_{ii} = \mathbf{0}_{9\times 9}$ and $\mathbf{A}_{j-1} \in \mathbb{R}^{9\times 9}$, $\mathbf{B}_{j-1} \in \mathbb{R}^{9\times 6}$ defined as

$$\mathbf{A}_{j-1} = \begin{bmatrix} \Delta\tilde{\mathbf{R}}_{j-1j}^\top & \mathbf{0}_{3\text{x}3} & \mathbf{0}_{3\text{x}3} \\ -\Delta\tilde{\mathbf{R}}_{ij-1}(\tilde{\mathbf{a}}_{j-1} - \mathbf{b}_i^a)^\wedge \Delta t & \mathbf{I}_3 & \mathbf{0}_{3\text{x}3} \\ -\frac{1}{2}\Delta\tilde{\mathbf{R}}_{ij-1}(\tilde{\mathbf{a}}_{j-1} - \mathbf{b}_i^a)^\wedge \Delta t^2 & \mathbf{I}_3\Delta t & \mathbf{I}_3 \end{bmatrix}$$

$$\mathbf{B}_{j-1} = \begin{bmatrix} \mathbf{J}_r^{j-1}\Delta t & \mathbf{0}_{3\text{x}3} \\ \mathbf{0}_{3\text{x}3} & \Delta\tilde{\mathbf{R}}_{ij-1}\Delta t \\ \mathbf{0}_{3\text{x}3} & \frac{1}{2}\Delta\tilde{\mathbf{R}}_{ij-1}\Delta t^2 \end{bmatrix}$$

where $\mathbf{I}_3$ stands for the identity matrix of size 3.

For the computation of the above matrices, the preintegrated expressions in [4] were used. The matrix $\mathbf{\Sigma}_{ij} \in \mathbb{S}_+^9$ is composed by nine sub-matrices of dimension 3 by 3 each

$$\mathbf{\Sigma}_{ij} = \left[\begin{array}{c|c|c} \mathbf{\Sigma}_{\Delta\boldsymbol{\phi}_{ij}} & \mathbf{\Sigma}_{\Delta\boldsymbol{\phi}_{ij}\Delta\mathbf{v}_i} & \mathbf{\Sigma}_{\Delta\boldsymbol{\phi}_{ij}\Delta\mathbf{p}_{ij}} \\ \hline \mathbf{\Sigma}_{\Delta\mathbf{v}_i\Delta\boldsymbol{\phi}_{ij}} & \mathbf{\Sigma}_{\Delta\mathbf{v}_i} & \mathbf{\Sigma}_{\Delta\mathbf{v}_i\Delta\mathbf{p}_{ij}} \\ \hline \mathbf{\Sigma}_{\Delta\mathbf{p}_{ij}\Delta\boldsymbol{\phi}_{ij}} & \mathbf{\Sigma}_{\Delta\mathbf{p}_{ij}\Delta\mathbf{v}_i} & \mathbf{\Sigma}_{\Delta\mathbf{p}_{ij}} \end{array}\right] \tag{6}$$

The above matrix will be important in the calculation of inertial residuals in the next section.

4.3. Gravity vector representation

As the gravity modulus is known, a reasonable representation is by its directional vector. Unit-norm direction vectors belong to the S^2 manifold, which has only two degrees of freedom. As S^2 does not form a Lie group, we follow the parametrization proposed in [6]. The explicit expressions are detailed in [19].

[1] By $\mathbb{S}_+^n = \{\mathbf{\Sigma} \in \mathbb{R}^{n\times n} \mid \mathbf{\Sigma} = \mathbf{\Sigma}^\top, \mathbf{\Sigma} \succeq 0\}$ we denote the set of $n \times n$ symmetric positive semidefinite matrices.

4.4. Optical flow and velocity constraint

We adopt the standard assumption that the local intensity image patterns are approximately constant, at least in the short period of time between two frames of a video [2]. This constraints the motion in the image as in the following

$$0 = \frac{\partial I}{\partial t} + \frac{\partial I}{\partial u}\dot{u} + \frac{\partial I}{\partial v}\dot{v} \tag{7}$$

where $(\dot{u}, \dot{v})$ is the optical flow in image units (pixels/s). Under the common assumption of a rigid scene, we formulate the point velocities in terms of the camera motion. Let $Z : \Omega \to \mathbb{R}$ be a depth image provided by a 3D range camera where Ω is the image domain. Following the work by Spies *et al.* [29], the range flow constraint is as follows

$$\dot{Z} = (\dot{w}) = \frac{\partial Z}{\partial t} + \frac{\partial Z}{\partial u}\dot{u} + \frac{\partial Z}{\partial v}\dot{v} \tag{8}$$

This equation reflects that the total derivative of the depth can be calculated from the optical flow and the partial derivatives of Z. Following [7] and using the pin-hole model, we obtain the range flow constraint in Eq. (9). Here f_x, f_y are the focal length values, expressed in pixels, while $\dot{x}, \dot{y}, \dot{z}$ are in camera coordinates.

$$\begin{aligned} -\frac{\partial Z}{\partial t} \quad = \quad & \left(1 + \frac{x f_x}{z^2}\frac{\partial Z}{\partial u} + \frac{y f_y}{z^2}\frac{\partial Z}{\partial v}\right)(v_z + y\omega_x - x\omega_y) \\ & + \frac{f_x}{z}\frac{\partial Z}{\partial u}(-v_x + y\omega_z - z\omega_y) \\ & + \frac{f_y}{z}\frac{\partial Z}{\partial v}(-v_y - x\omega_z + z\omega_x) \end{aligned} \tag{9}$$

The above constraint for the camera velocity will be used for our visual residuals, detailed in next section.

5. Camera Motion from RGB-D-I Flow

This section presents our approach to integrating inertial measurements with RGB-D scene flow to estimate camera motion. We begin by defining the state, which varies according to different operating modes, primarily depending on the number of frames considered. We then proceed to formulate the cost function to be optimized. Finally, this section concludes with the marginalization process, through which we retain the information of removed states.

5.1. State definition

Our goal is to track the state $\mathbf{x}$ of a sensing device equipped with an IMU and a RGB-D camera. This state consists basically of the device velocities, IMU biases and gravity vector at different moments of time. We assume that the IMU is synchronized with the camera, as it is shown in Fig. 1.

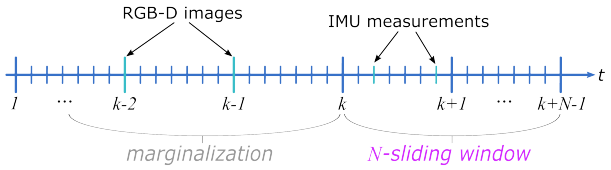

Figure 1. Illustration of the temporal notation for RGB-D images, IMU measurements and marginalization and optimization windows.

In a general case, the system performs the optimization over N frames, what we called a *N-sliding window*. As the sensing device moves through the trajectory, we marginalize out the old states for the optimization to remain compact. The state at step $k+N-1$ is defined as

$$\mathbf{x} = \left[\mathbf{v}_k^\top, \boldsymbol{\omega}_k^\top, \cdots, \mathbf{v}_{k+N-1}^\top, \boldsymbol{\omega}_{k+N-1}^\top, \mathbf{g}^\top, \mathbf{b}^{g\top}, \mathbf{b}^{a\top}\right]^\top \tag{10}$$

where $\{\mathbf{v}_l\}_{l=k}^{k+N-1} \in \mathbb{R}^3$ are the linear velocities in each frame, $\{\boldsymbol{\omega}_l\}_{l=k}^{k+N-1} \in \mathbb{R}^3$ are the angular velocities in each frame, $\mathbf{g} \in \mathrm{S}^2$ contain the two degrees of freedom of the gravity direction and $\mathbf{b}^g, \mathbf{b}^a \in \mathbb{R}^3$ are the gyroscope and accelerometer bias, respectively.

As mentioned, we include the gravity direction as part of the state. Under the traditional absolute formulations for visual-inertial state estimation, this variable is removed from the state by aligning the global reference frame to the gravity direction during system initialization. But, as a consequence, the rest of the variables in the state are tied to a gravity-aligned absolute frame. In turn, by including the gravity vector in the local camera frames we remove this dependence and all states are relative. As an additional benefit, it becomes possible to explicitly re-estimate the gravity direction during normal system operation and thus avoid coupling gravity and absolute orientation errors. In order to improve the observability of the state (in particular of the accelerometer bias), we assume a known gravity magnitude (981 cm/s^2) and only optimize the gravity direction.

5.2. Cost function

We formulate an optimization problem over the state $\mathbf{x}$ for which the camera velocity consistency is imposed as well as those terms corresponding to the pre-integration of the IMU readings. The joint optimization problem will consist on minimizing a cost function $J(\mathbf{x})$ which is the summation of terms associated to the inertial measurements J_i as well as to the camera measurements J_c. Our state estimate $\hat{\mathbf{x}}$ will be the one that minimizes the cost function $J(\mathbf{x})$.

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \; J(\mathbf{x}) = \arg\min_{\mathbf{x}} \; \left(J_c(\mathbf{x}) + J_i(\mathbf{x})\right) \tag{11}$$

We begin by developing the term J_c. For a pair of consecutive frames i and j, the velocity constraint in Eq. (9) results in the linear constraint

$$\mathbf{r}_c = \mathbf{W}\mathbf{A}\mathbf{x} - \mathbf{W}\mathbf{B} \tag{12}$$

where $\mathbf{A}$ contains the weights of the coefficients that multiply the state vector $\mathbf{x}$ in the velocity constraint, and the matrix $\mathbf{B}$ contains the temporal derivatives of the per-pixel depths (inverted in sign). The linearization that is applied to derive the range flow constraint in Eq. (9) assumes differentiability of the depth images and small scene displacement. Therefore, we implement an adaptive mask on the image, in order to discard those pixels belonging to edges and prone to have high depth derivatives. This mask is represented in a diagonal matrix $\mathbf{W}$, which also has the weights associated with the uncertainty of each equation. For details on these aspects, the reader is referred to [7]. Using the residual in Eq. (12), the visual cost is expressed as

$$J_c = \mathbf{r}_c^\top \boldsymbol{\Sigma}_c^{-1} \mathbf{r}_c \tag{13}$$

Having developed the first term of the cost function, we now turn our attention to the development of J_i, which will be composed by several terms. Firstly, from Eq. (1) we can derive the residual associated to the angular velocity estimate, as follows

$$\mathbf{r}_\omega = {}^B\boldsymbol{\omega} - \left({}^B\tilde{\boldsymbol{\omega}} - \mathbf{b}^g\right) \tag{14}$$

Assuming constant biases, as we compute flow for a small number of frames, the residual for the preintegrated linear velocity term is defined in [4] as

$$\mathbf{r}_{\Delta\mathbf{v}_i} = \mathbf{R}_i^\top(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t) - \Delta\tilde{\mathbf{v}}_{ij} \tag{15}$$

The residual for biases are made by penalising changes as the sliding window moves *i.e.*, $\mathbf{r}_{ba} = \mathbf{b}_0^a - \mathbf{b}^a$ and $\mathbf{r}_{bg} = \mathbf{b}_0^g - \mathbf{b}^g$, where $\mathbf{b}_0^a$ and $\mathbf{b}_0^g$ are the initial estimates.

Using the submatrices $\boldsymbol{\Sigma}_{\Delta\mathbf{v}_i}, \boldsymbol{\Sigma}_{\boldsymbol{\omega}}, \boldsymbol{\Sigma}_{\mathbf{a}} \in \mathbb{S}_+^3$ from Eq. (6), we can now define the term J_i as follows

$$\begin{aligned} J_i \;=\; & \mathbf{r}_{\Delta\mathbf{v}_i}^\top \boldsymbol{\Sigma}_{\Delta\mathbf{v}_i}^{-1} \mathbf{r}_{\Delta\mathbf{v}_i} + \mathbf{r}_\omega^\top \boldsymbol{\Sigma}_{\boldsymbol{\omega}}^{-1} \mathbf{r}_\omega \\ & + \mathbf{r}_{bg}^\top \boldsymbol{\Sigma}_{\boldsymbol{\omega}}^{-1} \mathbf{r}_{bg} + \mathbf{r}_{ba}^\top \boldsymbol{\Sigma}_{\mathbf{a}}^{-1} \mathbf{r}_{ba} \end{aligned} \tag{16}$$

Up to this point we have defined the cost function, by means of J_c and J_i. In the general case $J(\mathbf{x})$ will be made up depending on the number of frames in each case. These cases will be detailed below.

5.3. Operating Modes

We have mentioned that the cost function $J(\mathbf{x})$ will be formed as a function of the number of frames (N) in the sliding window while the camera is moving along the trajectory. Fig. 2 illustrates this situation graphically.

Depending on the value of N, the system will have different *operating modes* which will be detailed in this section.

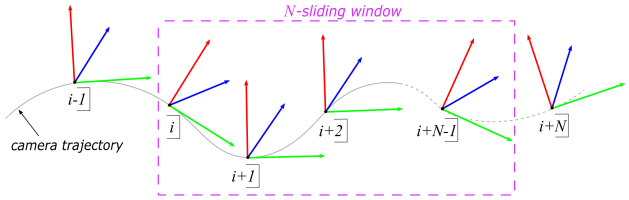

Figure 2. Optimization using a sliding window over N frames while the camera is moving over the trajectory.

We start with the factor graph illustrated in Fig. 3(a), where only frames i and j are available. Here, $N = 2$ and any variable has been marginalized yet. Since we only have two frames, the cost function will be $J = J_i^{ij} + J_c^{ij}$. The first term is associated to the inertial measurements and the second one to the camera measurements. Super-indices ^{ij}. denote that the corresponding term is built up by frames i and j. The state $\mathbf{x}$ contains in this case the velocities $\mathbf{v}_i, \boldsymbol{\omega}_i, \mathbf{v}_j$ and $\boldsymbol{\omega}_j$ as well as the gravity ${}^W\mathbf{g}$ and biases $\mathbf{b}^g$ and $\mathbf{b}^a$.

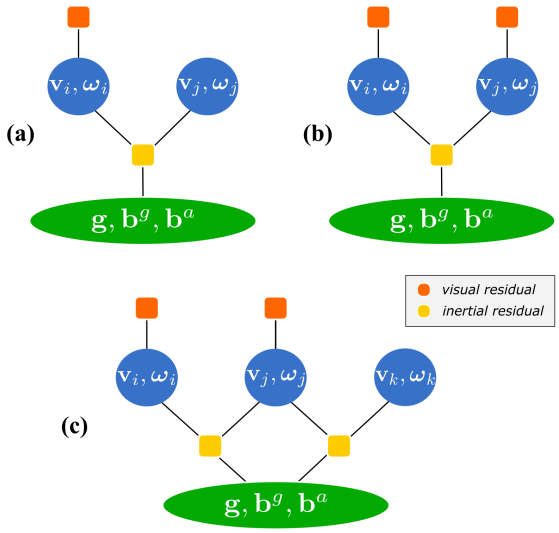

Figure 3. Factor graph representation using different modes of operation. Blue and green shapes contains the variables to be estimated. (a) Taking two frames, only one visual and one inertial residual are used. (b) Here we take three frames, so there are two visual residual. Only one inertial residual is used. (c) Is the same situation as before but in this case two inertial residual are used. The difference between (b) and (c) is the inertial constraint imposed by the last aggregate frame.

When a new frame ($k-$frame) is added to the window, $N = 3$ and the associated graph is shown in Fig. 3(b). In this case the state remains the same as before but the cost changes to $J(\mathbf{x}) = J_i^{ij} + J_c^{ij} + J_c^{jk}$, *i.e.*, only visual information is added to the cost function.

The last case is when we add a new inertial term (between j and k frames), therefore $J(\mathbf{x}) = J_i^{ij} + J_c^{ij} + J_c^{jk} + J_i^{jk}$. This situation is represented in Fig. 3(c). Here the state changes and the velocities $\mathbf{v}_k$ and $\boldsymbol{\omega}_k$ are added.

In the general case, the state $\mathbf{x} \in \mathbb{R}^{6N+8}$ is defined as

$$\mathbf{x} = \left[\mathbf{v}_i^\top, \boldsymbol{\omega}_i^\top, \ldots, \mathbf{v}_{i+N-1}^\top, \boldsymbol{\omega}_{i+N-1}^\top, \mathbf{g}^\top, \mathbf{b}^{g\top}, \mathbf{b}^{a\top}\right]^\top \tag{17}$$

and the cost function $J(\mathbf{x})$ can be expressed compactly as follows

$$J(\mathbf{x}) = \sum_{p=i}^{i+N-1} \left(\mathbf{r}_{c_p}^\top \boldsymbol{\Sigma}_{c_p}^{-1} \mathbf{r}_{c_p} + \mathbf{r}_{\Delta\mathbf{v}_p}^\top \boldsymbol{\Sigma}_{\Delta\mathbf{v}_p}^{-1} \mathbf{r}_{\Delta\mathbf{v}_p}\right) + \mathbf{r}_{bg}^\top \boldsymbol{\Sigma}_{\boldsymbol{\omega}}^{-1} \mathbf{r}_{bg} + \mathbf{r}_{ba}^\top \boldsymbol{\Sigma}_{\mathbf{a}}^{-1} \mathbf{r}_{ba} + \sum_{l=i}^{i+N} \mathbf{r}_{\omega_l}^\top \boldsymbol{\Sigma}_{\boldsymbol{\omega}}^{-1} \mathbf{r}_{\omega_l} \tag{18}$$

5.4. Marginalization

We mentioned that, as the sliding window moves, information from previous states is marginalized out. Let consider the case in Fig. 4, in which we want to perform 3-frames-sliding-window optimization.

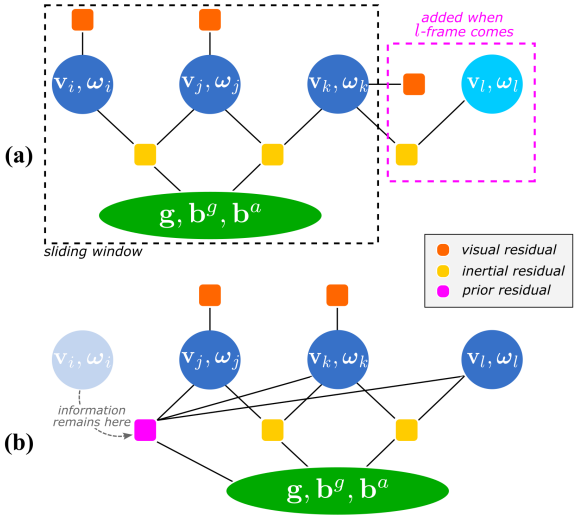

Figure 4. Factor graph using sliding window with containing 3 frames. Blue and green shapes contains the variables to be estimated. (a) When a new frame comes, both visual and inertial residual is added and the marginalization is done. (b) After marginalization, a new prior residual is added on the cost function.

Fig. 4(a) illustrates the graph after we have optimized the states at time steps i, j and k, and a new frame ($l-$frame) arrives. The state estimate at this point, $\mathbf{x}_M$, can be decomposed into two parts: the variables that we want to marginalize $\mathbf{x}_\alpha = [\mathbf{v}_i^\top, \boldsymbol{\omega}_i^\top]^\top \in \mathbb{R}^6$ and the starting point for the next step $\mathbf{x}_\beta =$

$$\left[\mathbf{v}_j^\top, \boldsymbol{\omega}_j^\top, \mathbf{v}_k^\top, \boldsymbol{\omega}_k^\top, \mathbf{v}_l^\top, \boldsymbol{\omega}_l^\top, \mathbf{g}^\top, \mathbf{b}^{g\top}, \mathbf{b}^{a\top}\right]^\top.$$

$$\mathbf{x}_M = \left[\begin{array}{c:c} \mathbf{x}_\alpha^\top & \mathbf{x}_\beta^\top \end{array}\right]^\top \tag{19}$$

Consider Eq. (20), where we have the cost function before marginalization $J(\mathbf{x})$, and a new term J_p, which stands for the marginalization priors and accounts for the information associated to the marginalized variables

$$J^*(\mathbf{x}) \doteq J(\mathbf{x}) + J_p \tag{20}$$

Using second-order Taylor approximation, the cost $J(\mathbf{x})$ can be expressed as follows

$$J(\mathbf{x}) \approx J(\mathbf{x}_0) + \cancel{\nabla J(\mathbf{x}_0)\mathbf{r}} + \frac{1}{2}\mathbf{r}^\top \mathbf{H}(\mathbf{x}_0)\mathbf{r} \tag{21}$$

The above approximation is calculated around a state $\mathbf{x}_0$, where a minimum is achieved, *i.e.* $\nabla J(\mathbf{x}_M)\mathbf{r} = 0$ ($\mathbf{r} = \mathbf{x} - \mathbf{x}_0$). The Hessian $\mathbf{H}$ contains the second derivatives of the cost function with respect to the state variables, therefore it encodes how every state variable affects the others.

We denote as α the block of variables we would like to marginalize, and β the block of variables we would like to keep. When marginalizing a set α of variables, we gather all factors dependent on them as well as the connected variables β. This is done by means of the Schur Complement as follows

$$\mathbf{H}^* = \mathbf{H}_{\beta\beta} - \mathbf{H}_{\alpha\beta}^\top \mathbf{H}_{\alpha\alpha}^{-1} \mathbf{H}_{\alpha\beta} \tag{22}$$

Fig. 5 graphically illustrates how the $\alpha-$block (variables $\mathbf{v}_i$ and $\boldsymbol{\omega}_i$) is removed from $\mathbf{H}$ but the information is preserved in the new matrix $\mathbf{H}^*$.

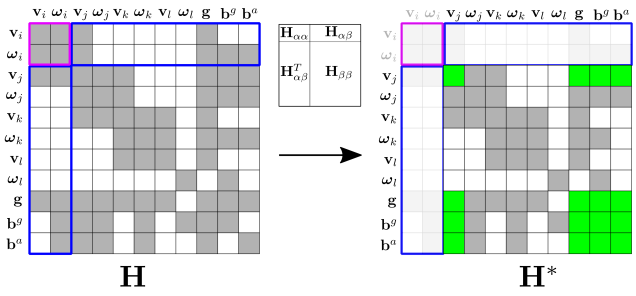

Figure 5. Marginalization example. We start with a Hessian matrix $\mathbf{H}$ after optimization with $N = 4$. We want to marginalize $\mathbf{v}_i$ and $\boldsymbol{\omega}_i$. The marginalized Hessian matrix $\mathbf{H}^*$ corresponds to the Schur complement of $\mathbf{H}_{\alpha\alpha}$. This calculation transfers the information constraints of the variable being eliminated to its adjacent nodes, adding shared information between these variables (green cells).

Considering $\mathbf{x}_\beta$ obtained from Eq. (19) and the new state $\mathbf{x}^*$, the term $\mathbf{r}_p$ can be expressed as $\mathbf{r}_p = \mathbf{x}^* - \mathbf{x}_\beta$, where

$$\mathbf{x}^* = \left[\mathbf{v}_j^\top, \boldsymbol{\omega}_j^\top, \mathbf{v}_k^\top, \boldsymbol{\omega}_k^\top, \mathbf{v}_l^\top, \boldsymbol{\omega}_l^\top, \mathbf{g}^\top, \mathbf{b}^{g\top}, \mathbf{b}^{a\top}\right]^\top$$

The new term J_p can now be defined as follows

$$J_p = \mathbf{r}_p^\top \mathbf{H}^* \mathbf{r}_p \tag{23}$$

Finally, the result is obtained by minimizing the cost function $J^*(\mathbf{x})$ expressed in Eq. (20).

6. Experiments

6.1. Setup

Public benchmarks that provide IMU, color and depth images are scarce. We chose to evaluate our proposal on an extended version of the living room sequences in the ICL-NUIM dataset [5]. ICL-NUIM is a synthetic photo-realistic dataset that provides ground truth poses as well as 3D scene models to benchmark reconstruction and/or localization approaches. As ICL-NUIM does not provide IMU data, in a manner similar to [11], we fit splines to the ground truth poses to simulate continuous trajectories and simulated IMU measurements from them. We use the IMU model described in [18] and the same IMU parameters as [12]. We also evaluated our RGB-D-inertial flow in the OpenLORIS-Scene datasets [28], in which data are collected in real-world indoor scenes, for multiple times in each place to include natural scene changes in everyday scenarios. RGB-D images and IMU measurements from a RealSense D435i are provided. The ground truth trajectory was recorded by an OptiTrack MCS, that tracked artificial markers deployed on the Segway robot used to record the data.

As metrics, we use the Root-Mean-Square-Error (RMSE) for the velocities $\mathbf{v}$ and $\boldsymbol{\omega}$ and biases $\mathbf{b}^a$ and $\mathbf{b}^g$. For the gravity vector, we use the angle $\theta_\mathbf{g} = \cos^{-1}\left(\hat{\mathbf{g}} \cdot \mathbf{g}_{gt} / \|\hat{\mathbf{g}}\| \|\mathbf{g}_{gt}\|\right)$ between the ground truth and estimated gravity direction.

6.2. Results

ICL-NUIM. In this experiment we use the living room sequences in the ICL-NUIM dataset, and we run our RGB-D-I flow based method against the so-called DIFODO [7], based on RGB-D flow. Note that, as [7] does not provide code, we used our own implementation based on the description in the paper. We run both scene flow methods for different estimation modes: 2-frames, 3-frames, 4-frames and 5-frames. Marginalization is not done here. The results are obtained over the entire dataset, taking as starting frame one out of every 2 which gives us more than 400 subsequences. In this experiment we have made 10 runs on the complete dataset. The results (specifically, the mean ± the standard deviation of the RMSE for the estimated states) are shown in Table 1, in which the best result per estimation mode is boldfaced. Note that using inertial measurements improves the accuracy in both linear and angular velocity estimates. We can also observe that the errors are higher for

	2-frames		3-frames			4-frames		5-frames		
	RGB-D (DIFODO*)	RGB-D-I (ours)	RGB-D (**)	RGB-D-I (ours)	RGB-D-I (ours+M)	RGB-D (**)	RGB-D-I (ours)	RGB-D (**)	RGB-D-I (ours)	RGB-D-I (ours+M)
$\text{RMSE}_{\mathbf{v}}$ $[cm/s]$	8.020 ±4.989	**7.897 ±4.699**	0.565 ±0.508	0.524 ±0.442	**0.523 ±0.250**	0.575 ±0.514	**0.533 ±0.447**	0.582 ±0.520	0.538 ±0.452	**0.535 ±0.245**
$\text{RMSE}_{\boldsymbol{\omega}}$ $[rad/s]$	0.168 ±0.079	**0.037 ±0.022**	0.00113 ±0.00081	0.00107 ±0.00072	**0.00091 ±0.00052**	0.00114 ±0.00083	**0.00108 ±0.00074**	0.00116 ±0.00085	0.00109 ±0.00075	**0.00094 ±0.00054**
$\text{RMSE}_{\mathbf{b}^a}$ $[cm/s^2]$	-	0.177 ±0.026	-	**0.214 ±0.083**	0.243 ±0.112	-	0.227 ±0.141	-	0.217 ±0.171	**0.199 ±0.080**
$\text{RMSE}_{\mathbf{b}^g}$ $[rad/s]$	-	0.031 ±0.025	-	**0.048 ±0.100**	0.049 ±0.012	-	0.086 ±0.168	-	**0.098 ±0.209**	0.103 ±0.073
$\theta_{\mathbf{g}}$ $[rad]$	-	0.372 ±0.308	-	0.299 ±0.236	**0.168 ±0.089**	-	0.273 ±0.204	-	0.275 ±0.215	**0.167 ±0.086**

Table 1. Error Metrics on ICL- NUIM for different operating modes. (DIFODO*) stands for our implementation of the method in [7]. (**) indicates the output of our DIFODO* implementation between pairs of frames.

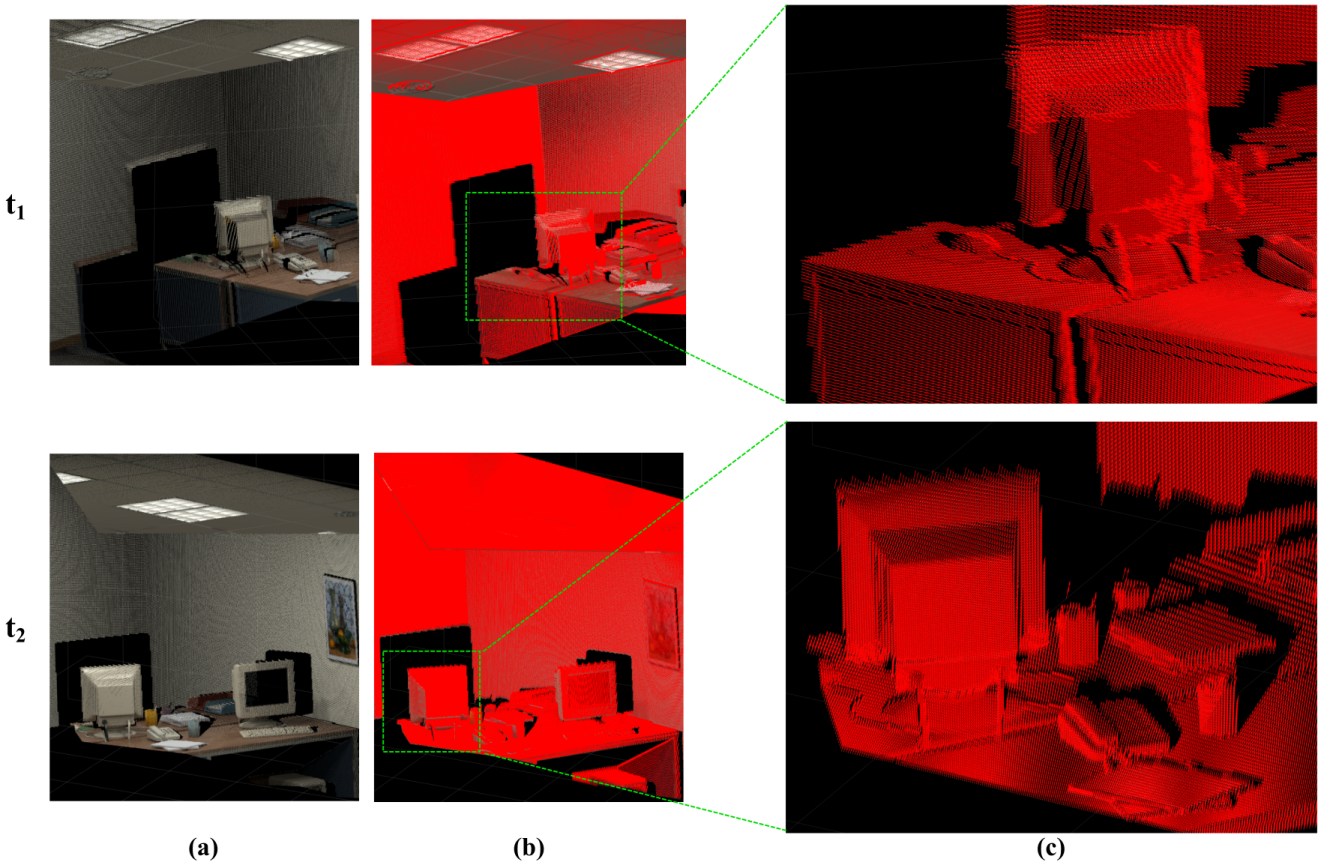

Figure 6. Motion estimation in an office scene from the ICL-NUIM dataset in two different times t_1 and t_2. (a) 3D representation of the scene. (b) Motion estimation of the objects in the scene. Every velocity is represented by a red arrow on each point. (c) Zoomed-in areas.

the 2-frames case than for the rest, which shows how the information from additional frames is leveraged to estimate the inertial states. Note also how the standard deviation of the errors is reduced when inertial sensing is used, which indicates a higher robustness in challenging cases.

ICL-NUIM + Marginalization. In this experiment, we show the effect of the marginalization in the living room sequences of ICL-NUIM, using two different sliding windows: 3 frames and 5 frames. In both cases we marginalize one frame only. The results are shown in the columns of Table 1 titled as *RGB-D-I (ours+M)*. It can be seen how the gravity vector estimation is improved in comparison to the case without marginalization. Also, a small improvement occurs in the linear and angular velocities errors.

	2-frames		3-frames		4-frames		5-frames	
	RGB-D (**)	RGB-D-I (ours)	RGB-D (*DIFODO**)	RGB-D-I (ours)	RGB-D (**)	RGB-D-I (ours)	RGB-D (**)	RGB-D-I (ours)
$\text{RMSE}_{\mathbf{v}}$ $[cm/s]$	36.993 ±11.295	**34.008 ±8.438**	22.453 ±9.029	**22.054 ±8.766**	22.453 ±9.029	**21.808 ±8.646**	22.453 ±9.029	**21.924 ±8.769**
$\text{RMSE}_{\boldsymbol{\omega}}$ $[rad/s]$	**0.165 ±0.090**	0.184 ±0.102	0.172 ±0.091	**0.169 ±0.092**	0.172 ±0.091	**0.168 ±0.092**	0.172 ±0.091	**0.167 ±0.092**
$\text{RMSE}_{\mathbf{b}^a}$ $[cm/s^2]$	-	0.121 ±0.001	-	0.102 ±0.028	-	0.097 ±0.014	-	0.113 ±0.024
$\text{RMSE}_{\mathbf{b}^g}$ $[rad/s]$	-	0.015 ±0.001	-	0.015 ±0.001	-	0.015 ±0.001	-	0.015 ±0.001
$\theta_{\mathbf{g}}$ $[rad]$	-	1.550 ±0.440	-	0.150 ±0.104	-	0.115 ±0.072	-	0.095 ±0.061

Table 2. Error Metrics on OpenLORIS-Scene for different operating modes. (DIFODO*) stands for our implementation of the method in [7]. (**) indicates the output of our DIFODO* implementation between pairs of frames.

Fig. 6 shows qualitative results for motion estimation. The office scene consists on a computer on a desk and luminaires. Fig. 6(a) shows a point cloud extracted from the RGB-D data, and Fig. 6(b) displays the scene flow (red arrows represent the velocity in each point). For better appreciation, Fig. 6(c) zooms in some areas. Observe that our approach estimates a smooth flow even in textureless areas such as the background wall.

OpenLORIS-Scene. In this experiment we consider the office-1 sequence in the OpenLORIS-Scene Dataset, where the robot moves along a U-shape route. As in previous experiments, we compare our RGB-D-I flow-based motion estimation against RGB-D-only motion estimation in 4 different optimization modes: 2-frames, 3-frames, 4-frames and 5-frames, all of them without marginalization. The experiment is performed over the entire dataset, using one frame out of every five as starting point, which gave us more than 400 experiments. Table 2 shows the mean ± the standard deviation of the RMSE for the results of such experiments. As in the synthetic case, it can be observed that using inertial measurements improves the estimation results in both linear and angular velocities. It can also be observed how, as the optimization window grows, the errors of the inertial states are also smaller.

7. Conclusions

In this work we present a novel camera motion estimation based on RGB-D-I scene flow. Specifically, we formulate the fusion of RGB-D and inertial data as a joint optimization using scene flow residuals and pre-integrated IMU residuals, weighted by their corresponding covariances. We also consider the marginalization of old states in order to keep a compact optimization. We evaluated our approach on a synthetic dataset, ICL-NUIM, and on a real dataset, OpenLORIS, both publicly available. Our results quantify the improvement that inertial fusion can offer to RGB-D scene flow techniques.

References

[1] Leopoldo Armesto, Josep Tornero, and Markus Vincze. Fast ego-motion estimation with multi-rate fusion of inertial and vision. *The International Journal of Robotics Research*, 26 (6):577–589, 2007. 1

[2] S. S. Beauchemin and J. L. Barron. The computation of optical flow. *ACM Computing Surveys*, 27(3):433–466, 1995. 3

[3] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021. 1

[4] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-Manifold Preintegration for Real-Time Visual–Inertial Odometry. *IEEE Transactions on Robotics*, 33(1):1–21, 2017. 2, 3, 4

[5] Ankur Handa, Thomas Whelan, John McDonald, and Andrew J. Davison. A benchmark for rgb-d visual odometry, 3d reconstruction and slam. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1524–1531, 2014. 6

[6] Christoph Hertzberg, René Wagner, Udo Frese, and Lutz Schröder. Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds. *Information Fusion*, 14(1):57–77, 2013. 3

[7] Mariano Jaimez and Javier Gonzalez-Jimenez. Fast visual odometry for 3-d range sensors. *IEEE Transactions on Robotics*, 31(4):809–822, 2015. 1, 2, 3, 4, 6, 7, 8

[8] Mariano Jaimez, Mohamed Souiai, Javier Gonzalez-Jimenez, and Daniel Cremers. A primal-dual framework for real-time dense rgb-d scene flow. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 98–104. IEEE, 2015. 2

[9] Mariano Jaimez, Javier Monroy, Manuel Lopez-Antequera, and Javier Gonzalez-Jimenez. Robust planar odometry based on symmetric range flow and multiscan alignment. *IEEE Transactions on Robotics*, 34(6):1623–1635, 2018. 1

[10] Christian Kerl, Jurgen Sturm, and Daniel Cremers. Robust odometry estimation for RGB-D cameras. In *2013 IEEE In-*

ternational Conference on Robotics and Automation, pages 3748–3754, Karlsruhe, Germany, 2013. IEEE. 1

[11] Christian Kerl, Jörg Stückler, and Daniel Cremers. Dense continuous-time tracking and mapping with rolling shutter rgb-d cameras. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2264–2272, 2015. 6

[12] Tristan Laidlow, Michael Bloesch, Wenbin Li, and Stefan Leutenegger. Dense rgb-d-inertial slam with map deformations. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6741–6748, 2017. 6

[13] Tristan Laidlow, Michael Bloesch, Wenbin Li, and Stefan Leutenegger. Dense RGB-D-inertial SLAM with map deformations. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6741–6748, Vancouver, BC, 2017. IEEE. 2

[14] Keuntaek Lee, Jason Gibson, and Evangelos A Theodorou. Aggressive perception-aware navigation using deep optical flow dynamics and pixelmpc. *IEEE Robotics and Automation Letters*, 5(2):1207–1214, 2020. 2

[15] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334, 2015. 2

[16] Xiuxiu Li, Yanjuan Liu, Haiyan Jin, Lei Cai, and Jiangbin Zheng. RGBD Scene Flow Estimation with Global Nonrigid and Local Rigid Assumption. *Discrete Dynamics in Nature and Society*, 2020:1–9, 2020. 2

[17] Matthias Nießner, Angela Dai, and Matthew Fisher. Combining inertial navigation and icp for real-time 3d surface reconstruction. In *Eurographics (Short Papers)*, pages 13–16. Citeseer, 2014. 2

[18] Janosch Nikolic, Paul Furgale, Amir Melzer, and Roland Siegwart. Maximum likelihood identification of inertial sensor noise model parameters. *IEEE Sensors Journal*, 16(1): 163–176, 2016. 6

[19] Matas Nitsche, Facundo Pessacg, and Javier Civera. Visual-inertial teach and repeat. *Robotics and Autonomous Systems*, 131:103577, 2020. 3

[20] Taihú Pire, Thomas Fischer, Gastón Castro, Pablo De Cristóforis, Javier Civera, and Julio Jacobo Berlles. S-ptam: Stereo parallel tracking and mapping. *Robotics and Autonomous Systems*, 93:27–42, 2017. 1

[21] Lorenzo Porzi, Markus Hofinger, Idoia Ruiz, Joan Serrat, Samuel Rota Bulo, and Peter Kontschieder. Learning multi-object tracking and segmentation from automatic annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6846–6855, 2020. 1

[22] Alessandro Prest, Christian Leistner, Javier Civera, Cordelia Schmid, and Vittorio Ferrari. Learning object class detectors from weakly annotated video. In *2012 IEEE Conference on computer vision and pattern recognition*, pages 3282–3289. IEEE, 2012. 1

[23] Julian Quiroga, Thomas Brox, Frédéric Devernay, and James Crowley. Dense Semi-rigid Scene Flow Estimation from RGBD Images. In *Computer Vision – ECCV 2014*, pages 567–582. Springer International Publishing, Cham, 2014. Series Title: Lecture Notes in Computer Science. 2

[24] Antoni Rosinol, John J Leonard, and Luca Carlone. Nerf-slam: Real-time dense monocular slam with neural radiance fields. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023. 1

[25] Yannick Schnider, Stanisław Woźniak, Mathias Gehrig, Jules Lecomte, Axel Von Arnim, Luca Benini, Davide Scaramuzza, and Angeliki Pantazi. Neuromorphic optical flow and real-time implementation with event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4128–4137, 2023. 2

[26] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Daniela Rus. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 5135–5142. IEEE, 2020. 1

[27] Zeyong Shan, Ruijian Li, and Sören Schwertfeger. RGBD-Inertial Trajectory Estimation and Mapping for Ground Robots. *Sensors*, 19(10):2251, 2019. 2

[28] Xuesong Shi, Dongjiang Li, Pengpeng Zhao, Qinbin Tian, Yuxin Tian, Qiwei Long, Chunhao Zhu, Jingwei Song, Fei Qiao, Le Song, Yangquan Guo, Zhigang Wang, Yimin Zhang, Baoxing Qin, Wei Yang, Fangshi Wang, Rosa H. M. Chan, and Qi She. Are we ready for service robots? the OpenLORIS-Scene datasets for lifelong SLAM. In *2020 International Conference on Robotics and Automation (ICRA)*, pages 3139–3145, 2020. 6

[29] Hagen Spies, Bernd Jähne, and John L. Barron. Range flow estimation. *Comput. Vis. Image Underst.*, 85(3):209–231, 2002. 3

[30] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021. 1

[31] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, pages 722–729 vol.2, Kerkyra, Greece, 1999. IEEE. 1

[32] Mingliang Zhai, Xuezhi Xiang, Ning Lv, and Xiangdong Kong. Optical flow and scene flow estimation: A survey. *Pattern Recognition*, 114:107861, 2021. 2

[33] Tianwei Zhang, Huayan Zhang, Yang Li, Yoshihiko Nakamura, and Lei Zhang. Flowfusion: Dynamic dense rgb-d slam based on optical flow. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7322–7328. IEEE, 2020. 1

[34] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based optical flow using motion compensation. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018. 1