Visual-Inertial Mapping with Non-Linear Factor Recovery

Vladyslav Usenko[1], Nikolaus Demmel[1], David Schubert[1], Jörg Stückler[2] and Daniel Cremers[1]

***Abstract*—Cameras and inertial measurement units are complementary sensors for ego-motion estimation and environment mapping. Their combination makes visual-inertial odometry (VIO) systems more accurate and robust. For globally consistent mapping, however, combining visual and inertial information is not straightforward. To estimate the motion and geometry with a set of images large baselines are required. Because of that, most systems operate on keyframes that have large time intervals between each other. Inertial data on the other hand quickly degrades with the duration of the intervals and after several seconds of integration, it typically contains only little useful information.**

In this paper, we propose to extract relevant information for visual-inertial mapping from visual-inertial odometry using non-linear factor recovery. We reconstruct a set of non-linear factors that make an optimal approximation of the information on the trajectory accumulated by VIO. To obtain a globally consistent map we combine these factors with loop-closing constraints using bundle adjustment. The VIO factors make the roll and pitch angles of the global map observable, and improve the robustness and the accuracy of the mapping. In experiments on a public benchmark, we demonstrate superior performance of our method over the state-of-the-art approaches.

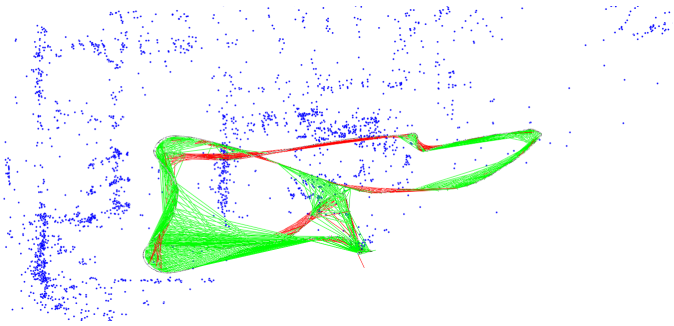


Fig. 1: Orthographic top-down projection of the map (MH_05 sequence of the EuRoC dataset [5]) rendered using the estimated gravity direction. To obtain a gravity-aligned globally consistent map, non-linear factors are recovered from the marginalization prior of the VIO and combined with keypoint-based bundle adjustment. Green lines visualize keyframe connections resulting from bundle adjustment factors and red lines connections from the recovered relative pose factors. Additionally each keyframe has a recovered factor that penalizes deviation from the gravity direction observed in VIO.

I. Introduction

Visual-inertial odometry (VIO) is a popular approach for tracking the motion of a camera in application domains such as robotics or augmented reality. By combining visual and IMU measurements, one can exploit the complementary strengths of both sensors and thereby increase accuracy and robustness. Commonly, the optimization of camera trajectory and map is performed locally on a small window of recent camera frames and IMU measurements. This approach, however, is inevitably prone to drift in the estimates.

Globally consistent optimization for visual-inertial mapping is less explored in the computer vision community. While in principle the optimization could be formulated as bundle adjustment with additional IMU measurements, this approach would quickly become computationally infeasible due to the high number of frames which would lead to a large number of optimization parameters in a naive formulation. To keep the computational burden in bounds, bundle adjustment subsamples the high-frame rate images of the camera to a smaller set of keyframes. The common choice in VIO is to preintegrate IMU measurements between consecutive frames. If we select keyframes temporally far apart to make the optimization efficient, the preintegrated IMU measurements provide only little information to constrain the trajectory due to the accumulated sensor noise. The small frame rate also affects the quality of the estimated velocities and biases from visual and inertial cues which are required for pose prediction using preintegrated IMU measurements.

We propose a novel approach that formulates visual-inertial mapping as bundle adjustment on a high-frame-rate set of visual and inertial measurements. Instead of directly optimizing the camera trajectory for all frames, we propose a hierarchical approach which first recovers a local VIO estimate at the frame rate of the camera. Once keyframes are removed and marginalized from the current local VIO optimization window, we extract non-linear factors [15] that approximate the accumulated visual-inertial information about the camera motion between keyframes. The keyframes and non-linear factors are subsequently used on the global bundle-adjustment layer.

For the VIO layer, our method uses image features designed for fast and accurate tracking, while for the mapping layer we employ distinctive but lighting and viewpoint invariant

Manuscript received: September 2, 2019; revised: November 27, 2019; accepted: December 13, 2019.

This paper was recommended for publication by Editor Eric Marchand upon evaluation of the Associate Editor and Reviewers' comments. This work was partially supported by the ERC Consolidator Grant "3D Reloaded" and the grant "For3D" by the Bavarian Research Foundation.

[1] Vladyslav Usenko, Nikolaus Demmel, David Schubert and Daniel Cremers are with the Technical University of Munich, Germany {usenko, demmeln, schubdav, cremers}@in.tum.de

[2] Jörg Stückler is with MPI for Intelligent Systems Tübingen, Germany joerg.stueckler@tuebingen.mpg.de

Digital Object Identifier (DOI): see top of this page.

keypoints that are suitable for loop closing. With this, our approach can leverage information from the IMU and short-term visual tracking at high frame rates together with keypoint matching and loop-closing at low frame rates for globally consistent mapping (Fig. 1). The factors also help to keep the map gravity-aligned, bridge between frames that do not have enough visual information. Our approach also makes the optimization problem smaller, since we do not have to estimate velocities and biases.

In summary, our contributions are:

- We propose a novel two-layered visual-inertial mapping approach that integrates keypoint-based bundle-adjustment with inertial and short-term visual tracking through non-linear factor recovery.
- As the first layer of our mapping approach we propose a VIO system which outperforms the state-of-the-art methods in terms of trajectory accuracy on the majority of the evaluated sequences. This is achieved by carefully combining appropriate components (patch tracking, landmark representation, first-estimate Jacobians, marginalization scheme) as detailed in Sec. IV.
- Unlike other state-of-the-art systems that use preintegrated IMU measurements also for mapping, we subsume high-frame rate visual-inertial information in non-linear factors extracted from the marginalization prior of the VIO layer. This results not only in a smaller optimization problem but also in better pose estimates in the resulting gravity aligned map.

We encourage the reader to watch the demonstration video and inspect the open-source implementation of the system, which is available at:

https://vision.in.tum.de/research/vslam/basalt

II. Related Work

Visual-inertial odometry: Early methods for visual-inertial odometry are primarily filter-based [11], [18]. In tightly integrated filters, the prediction step typically propagates the current camera state estimate using the IMU measurements. The state is recursively corrected based on the camera images. A significant drawback of filters is that the linearization point for the non-linear measurement and state transition models cannot be changed, once a measurement is integrated. Fixed-lag smoothers (a.k.a. optimization-based approaches) such as [13], [27] relinearize at the current states in a local optimization window of recent frames. The visual-inertial state estimation is formulated as a full bundle adjustment (BA) over keyframes and IMU measurements. The problem is reduced to a computationally manageable size by marginalization of old frames up to the recent set in the optimization window. The continuous relinearization, windowed optimization and maintenance of the marginalization prior increase the accuracy of the methods. The above methods need to discard keypoints and observations that are observed in marginalized keyframes in order to maintain the sparse structure of the marginalization prior. Hsiung et al. [9] apply non-linear factor recovery to achieve a sparse marginalization prior without discarding information about observed keypoints. This way, the approach can further refine the keypoints and achieve higher accuracy, but in contrast to our work it is limited to local BA.

Visual-inertial mapping: Only few works have tackled globally consistent mapping from visual and inertial measurements. Kasyanov et al. [12] add a pose-graph optimization layer with loop-closing on top of a keyframe-based visual-inertial odometry method [13]. The pose graph is built from the keyframes of the VIO and their relative pose estimates. In [19], the authors add inertial measurements to a keyframe-based SLAM system through IMU preintegration. The IMU measurements are preintegrated into a set of pseudo-measurements between keyframes. They notice that the accuracy of preintegrated measurements degrades over time and restrict the time between keyframes to 0.5 seconds in local BA and 3 seconds in global BA. A further shortcoming of the method is its requirement of estimating the camera velocity and IMU biases at each keyframe which is less well constrained through visual measurements than in our approach due to the strong temporal subsampling into keyframes. Schneider et al. [24] follow a similar approach in which preintegrated IMU measurements are inserted into the optimization. The approach in [20] proposes a combination of VIO and 4 degree-of-freedom (DoF) pose optimization for visual-inertial mapping. They fix 2 DoF (roll and pitch) and optimize only for the others. We also constrain roll and pitch from visual-inertial measurements. However, we extract non-linear factors in a probabilistic formulation which account for uncertainties in those values and are traded off with other information in the global probabilistic optimization.

III. Preliminaries

In this paper, we write matrices as bold capital letters (e.g. $\mathbf{R}$) and vectors as bold lowercase letters (e.g. $\boldsymbol{\xi}$). Rigid-body poses are represented as $(\mathbf{R}, \mathbf{p}) \in \mathrm{SO}(3) \times \mathbb{R}^3$ or as transformation matrices $\mathbf{T} \in \mathrm{SE}(3)$ when needed. Incrementing a rotation $\mathbf{R}$ by an increment $\boldsymbol{\xi} \in \mathbb{R}^3$ is defined as $\mathbf{R} \oplus \boldsymbol{\xi} = \mathrm{Exp}(\boldsymbol{\xi})\mathbf{R}$. The difference between two rotations $\mathbf{R}_1$ and $\mathbf{R}_2$ is calculated as $\mathbf{R}_1 \ominus \mathbf{R}_2 = \mathrm{Log}(\mathbf{R}_1\mathbf{R}_2^{-1})$ such that $(\mathbf{R} \oplus \boldsymbol{\xi}) \ominus \mathbf{R} = \boldsymbol{\xi}$. Here we use $\mathrm{Exp}\colon \mathbb{R}^3 \to \mathrm{SO}(3)$, which is a composition of the hat operator ($\mathbb{R}^3 \to \mathfrak{so}(3)$) and the matrix exponential ($\mathfrak{so}(3) \to \mathrm{SO}(3)$) and maps rotation vectors to their corresponding rotation matrices, and its inverse $\mathrm{Log}\colon \mathrm{SO}(3) \to \mathbb{R}^3$. For all other variables, such as translation, velocity and biases, we define $\oplus$ and $\ominus$ as regular addition and subtraction.

In the following we will use a state $\mathbf{s}$ that is defined as a tuple of several rotation and vector variables, and a function $\mathbf{r}(\mathbf{s})$ that depends on it and can also produce rotations and vectors as the result. An increment $\boldsymbol{\xi} \in \mathbb{R}^n$ is a stacked vector with all the increments of the variables in $\mathbf{s}$. Then, the Jacobian of the function with respect to the increment is defined as

$$\mathbf{J}_{\mathbf{r}(\mathbf{s})} = \lim_{\boldsymbol{\xi} \to \mathbf{0}} \frac{\mathbf{r}(\mathbf{s} \oplus \boldsymbol{\xi}) \ominus \mathbf{r}(\mathbf{s})}{\boldsymbol{\xi}} \,. \tag{1}$$

Here, $\mathbf{s} \oplus \boldsymbol{\xi}$ denotes that each component in $\mathbf{s}$ is incremented with the corresponding segment in $\boldsymbol{\xi}$ using the appropriate

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2019.2961227, IEEE Robotics and Automation Letters

definition of the $\oplus$ operator, and similarly for $\ominus$. The limit is done component-wise, such that the Jacobian is a matrix. For Euclidean quantities, this definition is just a normal derivative, with an extension for rotations, both as function value and as function argument. For more details and possible alternative formulations we refer the reader to [2], [4], [7].

In non-linear least squares problems, we minimize functions of the form

$$E(\mathbf{s}) = \frac{1}{2}\mathbf{r}(\mathbf{s})^\top \mathbf{W}\mathbf{r}(\mathbf{s})\,, \tag{2}$$

which is a squared norm of the sum of residuals with block-diagonal weight matrix $\mathbf{W}$. In this case, $\mathbf{r}(\mathbf{s})$ is purely vector-valued. Near the current state $\mathbf{s}$ we can use a linear approximation of the residual, which leads to

$$E(\mathbf{s} \oplus \boldsymbol{\xi}) = E(\mathbf{s}) + \boldsymbol{\xi}^\top \mathbf{J}_{\mathbf{r}(\mathbf{s})}^\top \mathbf{W}\mathbf{r}(\mathbf{s}) + \frac{1}{2}\boldsymbol{\xi}^\top \mathbf{J}_{\mathbf{r}(\mathbf{s})}^\top \mathbf{W}\mathbf{J}_{\mathbf{r}(\mathbf{s})}\boldsymbol{\xi}\,. \tag{3}$$

The optimum of this approximated energy can be attained using the Gauss-Newton increment

$$\boldsymbol{\xi}^* = -(\mathbf{J}_{\mathbf{r}(\mathbf{s})}^\top \mathbf{W}\mathbf{J}_{\mathbf{r}(\mathbf{s})})^{-1}\mathbf{J}_{\mathbf{r}(\mathbf{s})}^\top \mathbf{W}\mathbf{r}(\mathbf{s})\,. \tag{4}$$

With this, we can iteratively update the state $\mathbf{s}_{i+1} = \mathbf{s}_i \oplus \boldsymbol{\xi}^*$ until convergence.

IV. Visual-Inertial Odometry

We formulate the incremental motion tracking of the camera-IMU setup over time as fixed-lag smoothing. First, we use patch-based optical flow to track a sparse set of points in the 2D image plane between consecutive frames. This information is then used in a bundle-adjustment framework which for every frame minimizes an error that consists of point reprojection and IMU propagation terms. To maintain a fixed parameter size of the optimization problem we marginalize out old states. In the remainder of this section we will discuss these stages in more detail.

A. KLT Tracking

As a first step of our algorithm we detect a sparse set of keypoints in the frame using the FAST [22] corner detector. To track the motion of these points over a series of consecutive frames we use sparse optical flow based on KLT [14]. To achieve fast, accurate and robust tracking we combine the inverse-compositional approach as described in [1] with a patch dissimilarity norm that is invariant to intensity scaling. Several authors suggested zero-normalized cross-correlation (ZNCC) for illumination-invariant optical flow [17], [25], but we use locally-scaled sum of squared differences (LSSD) defined in [21] which is computationally less expensive than alternatives.

We formulate the patch tracking problem as estimating the transform $\mathbf{T} \in \mathrm{SE}(2)$ between two corresponding patches in two consecutive frames that minimizes the differences between the patches according to the selected norm. Essentially, we minimize a sum of squared residuals, where every residual is defined as

$$r_i(\boldsymbol{\xi}) = \frac{I_{t+1}(\mathbf{T}\mathbf{x}_i)}{\overline{I_{t+1}}} - \frac{I_t(\mathbf{x}_i)}{\overline{I_t}} \quad \forall \mathbf{x}_i \in \Omega. \tag{5}$$

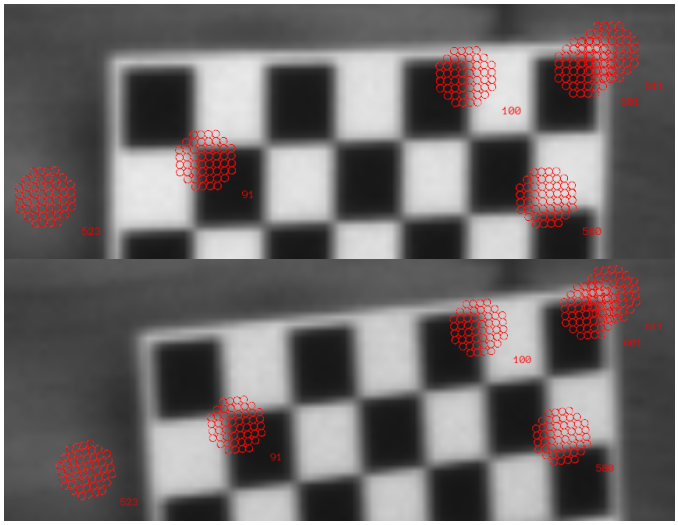


Fig. 2: Example of KLT tracks estimated by our system. Despite changes in exposure time the proposed method is able to estimate the warp in SE(2) between the patches in the images.

Here, $I_t(\mathbf{x})$ is the intensity of image t at pixel location $\mathbf{x}$. The set of image coordinates that defines the patch is denoted Ω and the mean intensity of the patch in image t is $\overline{I_t}$. A visualization of the patch and tracking results is shown in Fig. 2.

To achieve robustness to large displacements in the image we use a pyramidal approach, where the patch is first tracked on the coarsest level and then on increasingly finer levels. For outlier filtering, instead of an absolute threshold on the error, we track the patches from the current frame to the target frame and back to check consistency. Points that do not return to the initial location with the second tracking are considered as outliers and discarded.

B. Visual-Inertial Bundle Adjustment

To estimate the motion of the camera we combine error terms based on tracked feature locations from KLT tracking with IMU error terms based on preintegrated IMU measurements [8].

We use the following coordinate frames throughout the paper: W is the world frame, I is the IMU frame and C_i is the frame of camera i, where i is the index of the camera in a stereo setup. We estimate transformations $\mathbf{T}_{\mathrm{WI}} \in \mathrm{SE}(3)$ from IMU to world coordinate frame. The transformations $\mathbf{T}_{\mathrm{IC}_i}$ from camera frame i to IMU frame and the projection functions π_i are assumed to be static and known from calibration. For the formulation of reprojection errors we denote the transformations from camera i to world by $\mathbf{T}_{\mathrm{WC}_i}$. Those do not constitute additional optimization variables and are calculated using $\mathbf{T}_{\mathrm{WI}}$ and $\mathbf{T}_{\mathrm{IC}_i}$ in practice.

At different points in time, we optimize a state

$$\mathbf{s} = \{\mathbf{s}_\mathrm{k}, \mathbf{s}_\mathrm{f}, \mathbf{s}_\mathrm{l}\}\,, \tag{6}$$

where $\mathbf{s}_\mathrm{k}$ contains IMU poses for n older keyframes, $\mathbf{s}_\mathrm{f}$ contains IMU poses, velocities and biases of the m most recent

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2019.2961227, IEEE Robotics and Automation Letters

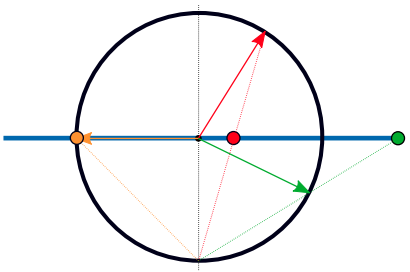


Fig. 3: Geometric interpretation of stereographic projection used to represent unit vectors. The two parameters define a point in the XY-plane of the coordinate system shown in blue. To obtain the corresponding 3D unit vector we cast a ray from $(0\ \ 0\ -1)^\top$ and find an intersection with the unit sphere shown in black. Three example points are visualized in red, green and yellow, with dashed lines representing the rays intersecting with the sphere and arrows showing the resulting unit vectors.

frames, which possibly are also keyframes if they host landmarks, and $\mathbf{s}_\mathrm{l}$ contains landmarks. A graphical representation of the problem is shown in Fig. 5 (a). Landmarks are stored relative to the keyframe where they were observed for the first time [16] and defined by a unit-length direction vector in the coordinate frame of the camera and an inverse distance to the landmark [6]. In the proposed system only keyframes host landmarks, which distinguishes them from regular frames.

1) Representation of Unit Vectors in 3D: In order to avoid the necessity of additional constraints for the optimization and to keep the number of optimiziation variables small, we parametrize the bearing vector in 3D space using a minimal representation, which is two-dimensional. In [3] the authors provide an extensive review of possible parametrizations and suggest a new parametrization based on $\mathrm{SO}(3)$ rotations that yields simple derivatives with respect to 2D increments.

In this work we use a parametrization based on stereographic projection that given 2D coordinates $(u, v)^\top$ generates a unit-length bearing vector

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \eta u \\ \eta v \\ \eta - 1 \end{pmatrix}, \quad \eta = \frac{2}{1 + u^2 + v^2}. \tag{7}$$

This parametrization is efficient as it only uses simple operations such as multiplication and division (compared to trigonometric operations needed in [6]) and is defined for all u and v. A geometric interpretation is shown in Fig. 3. The only direction vector that cannot be represented with finite u, v is the negative Z-direction $\begin{pmatrix} 0 & 0 & -1 \end{pmatrix}^\top$. However, this is not a drawback in practice, as cameras usually have a limited field of view and cannot see points behind them.

2) Reprojection Error: The first cue we can use for motion estimation is the reprojection error. When point i that is hosted in frame $h(i)$ is detected in target frame t at image coordinates $\mathbf{z}_{it}$, the residual is defined as

$$\mathbf{r}_{it} = \mathbf{z}_{it} - \pi_{c(t)}(\mathbf{T}_t^{-1}\mathbf{T}_{h(i)}\mathbf{q}_i(u, v, d)), \tag{8}$$

$$\mathbf{q}_i(u, v, d) = \begin{pmatrix} x(u,v) & y(u,v) & z(u,v) & d \end{pmatrix}^\top, \tag{9}$$

where $c(t)$ is the index of the camera used to take frame t. The pose $\mathbf{T}_t$ denotes $\mathbf{T}_{\mathrm{WC}_{c(t)}}$ at the time when frame t has been taken, and similarly for $\mathbf{T}_{h(i)}$. The first three entries of the homogeneous point coordinates $\mathbf{q}_i(u, v, d)$ are computed from the minimal representation (u, v) as described in Sec. IV-B1, with an additional fourth entry d, the inverse distance. Since the projection function is independent of scale we do not have to normalize $\mathbf{q}_i$, which makes this formulation numerically stable even when d is close or equal to zero.

3) IMU Error: The second cue for motion estimation is the IMU data. To deal with high frequency of the IMU measurements we preintegrate several consecutive IMU measurements into a pseudo-measurement. When adding an IMU factor between frame i and frame j, we compute pseudo-measurement $\Delta\mathbf{s} = (\Delta\mathbf{R}, \Delta\mathbf{v}, \Delta\mathbf{p})$ similar to [8], which we can use to formulate the residuals as

$$\mathbf{r}_{\Delta\mathbf{R}} = \mathrm{Log}\left(\Delta\tilde{\mathbf{R}}\mathbf{R}_j^\top\mathbf{R}_i\right), \tag{10}$$

$$\mathbf{r}_{\Delta\mathbf{v}} = \mathbf{R}_i^\top(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t) - \Delta\tilde{\mathbf{v}}, \tag{11}$$

$$\mathbf{r}_{\Delta\mathbf{p}} = \mathbf{R}_i^\top(\mathbf{p}_j - \mathbf{p}_i - \frac{1}{2}\mathbf{g}\Delta t^2) - \Delta\tilde{\mathbf{p}}, \tag{12}$$

where $\mathbf{g}$ is the gravity vector and $\mathbf{R}$ and $\mathbf{p}$ denote the rotation and translation components of $\mathbf{T}_{\mathrm{WI}}$, respectively. These residuals have to be weighted with appropriate covariance matrix $\mathbf{\Sigma}_{ij}$, which can be calculated recursively. For more detailed information about the underlying physical model of the IMU and preintegration theory we refer the reader to the supplementary material.

4) Optimization and Partial Marginalization: For each new frame we minimize a non-linear energy that consists of reprojection terms, IMU terms and a marginalization prior E_m

$$E = \sum_{\substack{i\in\mathcal{P} \\ t\in\mathrm{obs}(i)}} \mathbf{r}_{it}^\top\mathbf{\Sigma}_{it}^{-1}\mathbf{r}_{it} + \sum_{(i,j)\in\mathcal{C}} \mathbf{r}_{ij}^\top\mathbf{\Sigma}_{ij}^{-1}\mathbf{r}_{ij} + E_\mathrm{m}. \tag{13}$$

The reprojection errors are summed over the set of points $\mathcal{P}$ and for each point i over the set $\mathrm{obs}(i)$ of frames where the point is observed, including its host frame. The set $\mathcal{C}$ contains pairs of frames which are connected by IMU factors.

The energy E is optimized using the Gauss-Newton algorithm. To constrain the problem size we fix the number of keyframe poses and consecutive states that we optimize at every iteration. When a new frame is added, there are n pose-only keyframes in $\mathbf{s}_\mathrm{k}$ and the m newest frames including the newly added one in $\mathbf{s}_\mathrm{f}$. After optimizing, we perform a partial marginalization of the state to prevent the problem size from growing.

Two possible scenarios for marginalization are shown in Fig. 5. In the first one we marginalize out the oldest non-keyframe. In this case we drop the landmark factors that have this frame as a target to maintain the sparsity of the problem. In the second case we have a new keyframe, so we marginalize out velocity and biases for this frame and one old keyframe with corresponding landmarks.

In both cases the marginalization is done on the linearized Markov blanket of the variables we want to remove, where the Markov blanket is a collection of incident states to those variables. The linearization $\mathbf{H}$ and $\mathbf{b}$ represent a distribution of the estimated state in the vector space of the increment $\boldsymbol{\xi}$. If we split the increment $\boldsymbol{\xi} = [\boldsymbol{\xi}_\alpha^\top, \boldsymbol{\xi}_\beta^\top]^\top$ into variables $\boldsymbol{\xi}_\alpha$ to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2019.2961227, IEEE Robotics and Automation Letters

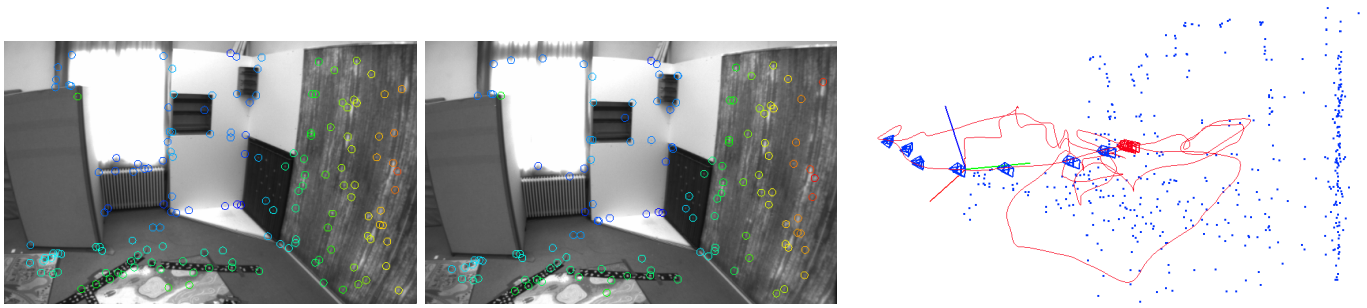


Fig. 4: Visual-inertial odometry subsystem proposed in Section IV. Projections of the landmarks with color-coded inverse distance used for estimating the position of the current frame are shown on the left. The results of local visual-inertial bundle adjustment are shown on the right. Keyframe poses with the associated landmarks are visualized in blue, current states and the estimated trajectory are visualized in red. Information about the keyframe poses in the local window is approximated using a set of non-linear factors as described in Section V and reused for global mapping.

stay in the system and variables $\boldsymbol{\xi}_\beta$ to be marginalized, we can compute the parameters of the new distribution using the Schur complement,

$$\mathbf{H}^{\mathrm{m}}_{\alpha\alpha} = \mathbf{H}_{\alpha\alpha} - \mathbf{H}_{\alpha\beta}\mathbf{H}^{-1}_{\beta\beta}\mathbf{H}_{\beta\alpha}\,, \tag{14}$$

$$\mathbf{b}^{\mathrm{m}}_{\alpha} = \mathbf{b}_{\alpha} - \mathbf{H}_{\alpha\beta}\mathbf{H}^{-1}_{\beta\beta}\mathbf{b}_{\beta}\,, \tag{15}$$

where we have split the original $\mathbf{H}$ and $\mathbf{b}$ into

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{\alpha\alpha} & \mathbf{H}_{\alpha\beta} \\ \mathbf{H}_{\beta\alpha} & \mathbf{H}_{\beta\beta} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_{\alpha} \\ \mathbf{b}_{\beta} \end{bmatrix}. \tag{16}$$

$\mathbf{H}^{\mathrm{m}}_{\alpha\alpha}$ and $\mathbf{b}^{\mathrm{m}}_{\alpha}$ now define an energy term that only depends on $\boldsymbol{\xi}_\alpha$ and can be added to the total energy at the next iteration.

We use first-estimate Jacobians [10] to maintain the nullspace properties of the linearized marginalization prior. As soon as a variable becomes a part of the marginalization prior, its linearization point is fixed, and the Jacobian used to calculate $\mathbf{H}$ and $\mathbf{b}$ is evaluated at this linearization point, while the residuals are calculated at the current state estimate. Residuals already in the marginalization term have to be linearly approximated, thus not $\mathbf{b}^{\mathrm{m}}_{\alpha}$, but $\mathbf{b}^{\mathrm{m}}_{\alpha}+\mathbf{H}^{\mathrm{m}}_{\alpha\alpha}\boldsymbol{\delta}_\alpha$ is added to the Gauss-Newton optimization once $\boldsymbol{\xi}_\alpha$ deviates by $\boldsymbol{\delta}_\alpha$ from the state used to calculate the residuals in $\mathbf{b}^{\mathrm{m}}_{\alpha}$.

V. Visual-Inertial Mapping

The fixed-lag smoothing method for visual-inertial odometry (Fig. 4) presented in the previous section accumulates drift in the estimate due to the fixed linearization points outside the optimization window. A typical approach to eliminate such drift is to detect loop closures and incorporate loop-closing constraints into the optimization. We propose a two-layered approach which runs our visual-inertial odometry on the lower layer and bundle-adjustment on the visual-inertial mapping layer, where we additionally use non-linear factors that summarize the keyframe pose information from the odometry layer. BA optimizes the camera poses of keyframes and positions of keypoints. We implicitly detect loop closures using keypoint matching and achieve globally consistent mapping.

A. Global Map Optimization

To get statistically independent observations we detect and match ORB [23] features (distinct from VIO points) between the keyframes in the global map optimization. This allows us to use the reprojection error function as defined in Eq. (8). Combining this reprojection error with the error terms from the recovered non-linear factors yields the objective function:

$$E^{\mathrm{G}}(\mathbf{s}) = \sum_{\substack{i\in\mathcal{P} \\ t\in\mathrm{obs}(i)}} \mathbf{r}_{it}^{\top}\boldsymbol{\Sigma}_{it}^{-1}\mathbf{r}_{it} + E_{\mathrm{nfr}}(\mathbf{s}), \tag{17}$$

where $E_{\mathrm{nfr}}(\mathbf{s})$ collects the error terms by the recovered non-linear factors. These factors and their recovery are detailed in the following. The state $\mathbf{s}$ that we optimize on this global optimization layer includes the keyframe poses and the positions of the new landmarks (parametrized as in Sec. IV-B1).

We interface the global map optimization with the VIO layer at the keyframe poses. When a keyframe is marginalized out from the VIO we save the linearization of the Markov blanket (Fig. 5 (c)) and marginalize all other variables except of keyframe poses. From this marginalization prior, we recover a set of non-linear factors on the keyframe poses that approximate the distribution stored in it.

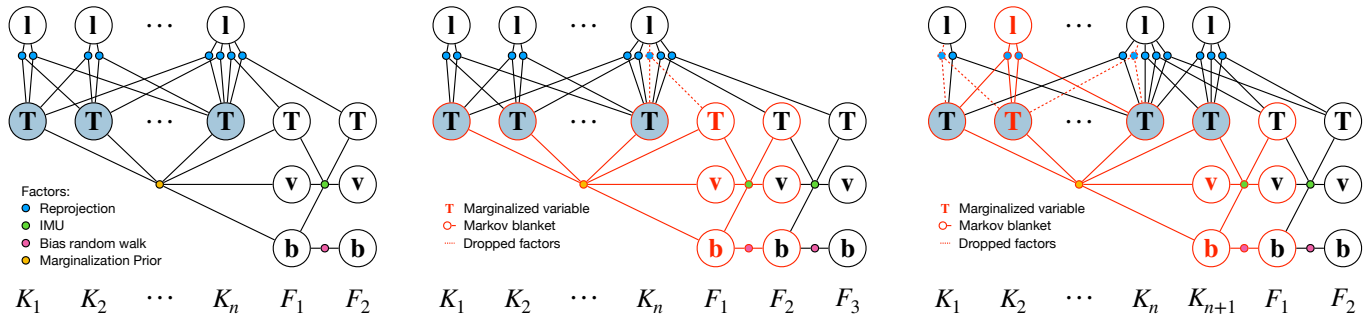
B. Non-Linear Factor Recovery

Non-linear factor recovery (NFR [15]) approximates a dense distribution stored in the linearized Markov blanket of the original factor graph with a different set of non-linear factors that yield a sparse factor graph topology. While the initial aim of NFR is to keep the computational complexity of SLAM optimization bounded, we use it to transfer information accumulated during VIO to our globally consistent visual-inertial map optimization.

By linearization of the residual function of a non-linear least squares problem Eq. (2), we obtain a multivariate Gaussian distribution $p(\mathbf{s}) \sim N(\boldsymbol{\mu}_{\mathrm{o}}, \mathbf{H}_{\mathrm{o}}^{-1})$ in which the mean $\boldsymbol{\mu}_{\mathrm{o}}$ equals the state estimate. We want to construct another distribution $p_{\mathrm{a}}(\mathbf{s}) \sim N(\boldsymbol{\mu}_{\mathrm{a}}, \mathbf{H}_{\mathrm{a}}^{-1})$ that well approximates the original distribution with a sparser factor graph topology.

We follow NFR [15] and minimize the Kullback-Leibler divergence (KLD) between the recovered distribution and the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2019.2961227, IEEE Robotics and Automation Letters



(a) (b) (c)

Fig. 5: Factor graphs. (a) After marginalizing a frame, the system consists of n older keyframes $K_1 \dots K_n$ and the $m-1$ most recent frames F_1 and F_2 (which could potentially also host landmarks and hence be keyframes). After a new frame has been added, the oldest velocity $\mathbf{v}$ and the oldest bias $\mathbf{b}$ are marginalized. If they do not belong to a keyframe (b), the whole frame including its pose $\mathbf{T}$ is marginalized. If they belong to a keyframe (c), another keyframe is selected for marginalization, including the landmarks hosted in it and its pose. In both cases, reprojection factors where the target frame is the marginalized frame are dropped. In the latter case, reprojection factors from the marginalized frame to F_2 are dropped to allow relinearization. Note that not all possible combinations of host and target frames for reprojection factors are shown.

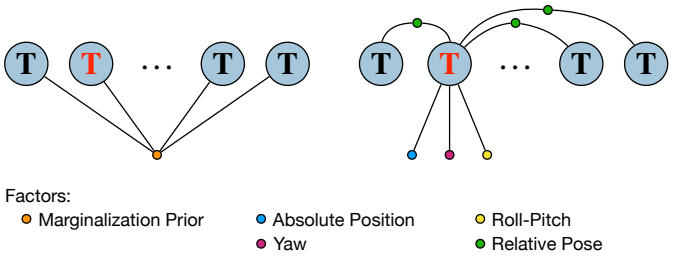


Fig. 6: Visualization of non-linear factor recovery. Left: Densely connected factor from marginalization saved from the VIO before removing a keyframe pose. Right: Extracted non-linear factors that approximate the distribution stored in the original factor.

original distribution. More formally, we minimize

$$D_{\mathrm{KL}}(p(\mathbf{s})||p_{\mathrm{a}}(\mathbf{s})) = \frac{1}{2}\left(\langle \mathbf{H}_{\mathrm{a}}, \boldsymbol{\Sigma}_{\mathrm{o}}\rangle - \log\det(\mathbf{H}_{\mathrm{a}}\boldsymbol{\Sigma}_{\mathrm{o}}) + ||\mathbf{H}_{\mathrm{a}}^{\frac{1}{2}}(\boldsymbol{\mu}_{\mathrm{a}} - \boldsymbol{\mu}_{\mathrm{o}})||^2 - d\right), \tag{18}$$

where $\boldsymbol{\Sigma}_{\mathrm{o}} = \mathbf{H}_{\mathrm{o}}^{-1}$ and d is constant.

For the ith non-linear factor that we want to recover, we need to define a residual function such that $\mathbf{r}_i(\mathbf{s}, \mathbf{z}_i) = \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{H}_i^{-1})$. NFR estimates the pseudo measurements $\mathbf{z}_i$ and information matrices $\mathbf{H}_i$ for the factors. Choosing $\mathbf{z}_i$ such that $\mathbf{r}_i(\boldsymbol{\mu}_{\mathrm{o}}, \mathbf{z}_i) = \mathbf{0}$ induces $\boldsymbol{\mu}_{\mathrm{a}} = \boldsymbol{\mu}_{\mathrm{o}}$ which makes the third term of (18) vanish. To estimate $\mathbf{H}_i$ we define

$$\mathbf{J}_{\mathrm{r}} = \begin{bmatrix} \vdots \\ \mathbf{J}_i \\ \vdots \end{bmatrix} \quad \mathbf{H}_{\mathrm{r}} = \begin{bmatrix} \ddots & & 0 \\ & \mathbf{H}_i & \\ 0 & & \ddots \end{bmatrix}, \tag{19}$$

where $\mathbf{J}_{\mathrm{r}}$ stacks the Jacobians of the defined residual functions with respect to the state, and $\mathbf{H}_{\mathrm{r}}$ is a block diagonal matrix that consists of the $\mathbf{H}_i$ for the corresponding residual functions. This allows us to write $\mathbf{H}_{\mathrm{a}} = \mathbf{J}_{\mathrm{r}}^{\top}\mathbf{H}_{\mathrm{r}}\mathbf{J}_{\mathrm{r}}$, and consequently, we can recover the information matrices $\mathbf{H}_i$ by minimizing

$$D_{\mathrm{KL}}(\mathbf{H}_{\mathrm{r}}) = \langle \mathbf{J}_{\mathrm{r}}^{\top}\mathbf{H}_{\mathrm{r}}\mathbf{J}_{\mathrm{r}}, \boldsymbol{\Sigma}_{\mathrm{o}}\rangle - \log\det(\mathbf{J}_{\mathrm{r}}^{\top}\mathbf{H}_{\mathrm{r}}\mathbf{J}_{\mathrm{r}}). \tag{20}$$

For full-rank and invertible $\mathbf{J}_{\mathrm{r}}$, [15], [9] showed that the following closed-form solution exists,

$$\mathbf{H}_i = (\{\mathbf{J}_{\mathrm{r}}\boldsymbol{\Sigma}_{\mathrm{o}}\mathbf{J}_{\mathrm{r}}^{\top}\}_i)^{-1}, \tag{21}$$

where $\{\}_i$ denotes the corresponding diagonal block.

C. Non-Linear Factors for Distribution Approximation

When we need to marginalize out a keyframe as shown in Fig. 5 (c), we save the current linearization and marginalize out everything except the keyframe poses. This gives us a factor that densely connects all keyframe poses in the optimization window. We use it to recover non-linear factors between the marginalized keyframe and all other keyframes as shown in Fig. 6. We define the following residual functions:

$$\mathbf{r}_{\mathrm{rel}}(\mathbf{s}, \mathbf{z}_{\mathrm{rel}}) = \mathrm{Log}(\mathbf{z}_{\mathrm{rel}}\mathbf{T}_j^{-1}\mathbf{T}_i), \tag{22}$$

$$\mathbf{r}_{\mathrm{rp}}(\mathbf{s}, \mathbf{z}_{\mathrm{rp}}) = \lfloor \mathbf{z}_{\mathrm{rp}}\mathbf{R}_i^{-1}(0,0,-1)^{\top}\rfloor_{xy}, \tag{23}$$

$$\mathbf{r}_{\mathrm{pos}}(\mathbf{s}, \mathbf{z}_{\mathrm{pos}}) = \mathbf{z}_{\mathrm{pos}} - \mathbf{p}_i, \tag{24}$$

$$\mathbf{r}_{\mathrm{yaw}}(\mathbf{s}, \mathbf{z}_{\mathrm{yaw}}) = \lfloor \mathbf{R}_i\mathbf{z}_{\mathrm{yaw}}\rfloor_{y}, \tag{25}$$

where with $\lfloor\rfloor_{xy}$ we denote x and y components of the vector and with $\mathbf{z}$ we denote the recovered measurements from the estimated state at the time of linearization. In our case $\mathbf{z}_{\mathrm{rel}} = \mathbf{T}_i^{-1}\mathbf{T}_j \in \mathrm{SE}(3)$, $\mathbf{z}_{\mathrm{rp}} = \mathbf{R}_i \in \mathrm{SO}(3)$, $\mathbf{z}_{\mathrm{pos}} = \mathbf{p}_i \in \mathbb{R}^3$ and $\mathbf{z}_{\mathrm{yaw}} = \mathbf{R}_i^{-1}\begin{pmatrix}1 & 0 & 0\end{pmatrix}^{\top} \in \mathbb{R}^3$.

We recover pairwise relative-pose factors between the keyframe that we will remove and all other current VIO keyframes. For that keyframe we also recover roll-pitch, absolute position and yaw factors (Fig. 6). This gives us a full-rank invertible Jacobian $\mathbf{J}_{\mathrm{r}}$ which means that we can use Eq. (21) for recovering information matrices for the factors.

Since yaw and absolute position are 4 unobservable states of the VIO, the only information we have there comes from

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2019.2961227, IEEE Robotics and Automation Letters

Sequence	MH_01	MH_02	MH_03	MH_04	MH_05	V1_01	V1_02	V1_03	V2_01	V2_02
VI DSO [26], mono	**0.06**	**0.04**	0.12	**0.13**	0.12	0.06	0.07	**0.10**	**0.04**	0.06
OKVIS [13] mono	0.34	0.36	0.30	0.48	0.47	0.12	0.16	0.24	0.12	0.22
OKVIS [13] stereo	0.23	0.15	0.23	0.32	0.36	**0.04**	0.08	0.13	0.10	0.17
VINS FUSION [20] mono	0.18	0.09	0.17	0.21	0.25	0.06	0.09	0.18	0.06	0.11
VINS FUSION [20] stereo	0.24	0.18	0.23	0.39	0.19	0.10	0.10	0.11	0.12	0.10
IS VIO [9] stereo	**0.06**	0.06	0.10	0.24	0.19	0.06	0.10	0.26	0.08	0.21
Proposed VIO, stereo	0.07	0.06	**0.07**	**0.13**	**0.11**	**0.04**	**0.05**	**0.10**	**0.04**	**0.05**
VI SLAM [12] mono, KF	0.25	0.18	0.21	0.30	0.35	0.11	0.13	0.20	0.12	0.20
VI SLAM [12] stereo, KF	0.11	0.09	0.19	0.27	0.23	0.04	0.05	0.11	0.10	0.18
VI ORB-SLAM [19], mono, KF	**0.07**	0.08	0.09	0.22	**0.08**	**0.03**	0.03	X	**0.03**	0.04
Pure BA, stereo, KF	0.09	0.08	**0.05**	0.27	0.16	0.04	0.03	X	0.04	0.04
BA + Identity Factors, stereo, KF	0.08	0.07	X	0.34	0.15	0.04	0.03	0.56	0.05	0.04
Proposed VI Mapping, stereo, KF	0.08	**0.06**	**0.05**	**0.10**	**0.08**	0.04	**0.02**	**0.03**	**0.03**	**0.02**

TABLE I: RMS ATE of the estimated trajectory in meters on the EuRoC dataset for several different methods. In the upper part we summarize the results for the VIO methods that run optimization in a local window and estimate the pose of every camera frame. In the lower part we evaluate mapping methods that operate on all keyframes and perform global map optimization. In both evaluations the proposed system shows the lowest error on the majority of the sequences and outperforms the competitors. Note: The V2_03 sequence is excluded from the comparison because it has more than 400 missing frames for one of the cameras.

the initial prior on the start pose. As we do not need this information for the global map we drop yaw and absolute position factors, and only take relative pose and roll-pitch factors for the map optimization. With these factors, the energy terms $E^{\mathrm{G}}_{\mathrm{nfr}}$ become

$$E^{\mathrm{G}}_{\mathrm{nfr}}(\mathbf{s}) = \sum_{(i,j)\in\mathcal{R}} \mathbf{r}_{ij}^{\top}\mathbf{H}_{ij}\mathbf{r}_{ij} + \sum_{i\in\mathcal{P}} \mathbf{r}_{i}^{\top}\mathbf{H}_{i}\mathbf{r}_{i}, \tag{26}$$

where $\mathcal{R}$ is a set of all relative pose factors and $\mathcal{P}$ is the set of all roll-pitch factors.

VI. Evaluation

To evaluate the presented approach we conduct evaluation on the EuRoC dataset [5] and compare it to other state-of-the-art systems. We present the evaluation for both our VIO subsystem and our full visual-inertial mapping approach. Our VIO runs the optimization in a local window of frames and provides a pose for every tracked frame, while the mapping system performs global map optimization for keyframes that were selected by the VIO. To measure the accuracy of the evaluated systems, we use the root mean square (RMS) of the absolute trajectory error (ATE) after aligning the estimates with ground truth.

a) System parameters: At the KLT tracking stage the image is divided into a regular grid with the cell size of 50 pixels. For each cell that has no point tracked from the previous frame, one feature point with the best FAST response is extracted (if it exceeds the threshold). With the resolution of the EuRoC dataset it results in 80-120 features tracked by the system at every point in time. At the VIO level we use a window of 7 old keyframes (poses) and 3 latest temporal states (poses, velocities and biases). The newest temporal state is selected as a keyframe if less than 70% of the KLT features are connected to the currently tracked points in the local map.

b) Accuracy: The results of the evaluation are summarized in Table I. When considering visual-inertial odometry methods our system shows the best performance on eight out of ten sequences while the closest competitor (VI DSO [26]) shows the best results on five.

To evaluate the mapping part we compare it to the visual-inertial version of ORB-SLAM [19], where the vision subsystem is very similar to the one proposed in our mapping layer (ORB keypoints). The main difference lies in the inertial part where ORB-SLAM uses preintegrated measurements between keyframes, while we use recovered non-linear factors that summarize IMU and visual tracking on the VIO layer.

The proposed system clearly outperform ORB-SLAM on the "machine hall" sequences where the large scale of the environment results in large time intervals between keyframes. On the "Vicon room" sequences the difference is smaller, since the rapid motion of the MAV that carries the camera in a small room results in many keyframes with small time intervals between them.

Qualitative results of reconstructed maps are shown in Fig. 1. With the proposed system we are able to reconstruct globally consistent gravity-aligned maps and recover keyframe poses even for segments where no matches between detected ORB features can be estimated.

c) Factor Weighting: To evaluate the importance of the extracted factors and their proper weighting in the final mapping results we consider two alternative implementations. In the first one we do not use any factors and rely purely on the BA with ORB features. In the second one we extract the factors, but use identity weights (i.e. $\mathbf{H}_{ij} = \mathbf{H}_i = \mathbf{I}$ in Eq. (26)) for all of them, which is a typical approach for pose graph optimization [19], [20]. The evaluation results presented in Table I show that the system with the factor weights recovered according to Sec. V results in better accuracy and robustness when compared to those alternatives.

d) Timing: The main source of timing improvement for the mapping stage is the fact that for a global optimization requires a 2.5 smaller state (no velocity or biases) compared to the naive IMU integration. In absolute numbers we test our system on an Intel E5-1620 CPU (4 cores, 8 virtual cores). Our implementation is highly parallel and utilizes all available

Total	Factor Extraction	Keypoint detection	Matching and Triangulation	Optimization (10 iterations)
52.8	3.6	6.4	23.1	19.7

TABLE II: Mean processing time in **milliseconds** of the mapping subsystem on EuRoC sequences normalized (divided) by the number of keyframes in the map.

CPU resources. For the VIO the average time per frame on the EuRoC sequences is 7.83 ms (largest: 9.4 ms on MH_02; smallest: 5.5 ms in V1_03). On average 11.5% of the frames are selected as keyframes and proceed to the mapping stage.

The timing of the mapping stage is provided in Table II. In particular, for the MH_05 sequence (see Fig. 1, 2273 stereo frames, 114 seconds) the processing takes 19.2 seconds for VIO and 9.7 seconds for mapping for the entire sequence (around 4x faster than real-time playback).

VII. Conclusions

In this paper we present a novel approach for visual-inertial mapping that combines the strengths of highly accurate visual-inertial odometry with globally consistent keyframe-based bundle adjustment. We achieve this in a hierarchical framework that successively recovers non-linear factors from the VIO estimate that summarize the accumulated inertial and visual information between keyframes. VIO is formulated as fixed-lag smoothing which optimizes a set of active recent frames in a sliding window and keeps past information in marginalization priors. The accumulated VIO information between keyframes is extracted and retained for the visual-inertial mapping when a keyframe falls outside the window and is marginalized.

Compared to alternative approaches that use preintegrated IMU measurements between keyframes our system shows better trajectory estimates on a public benchmark. This formulation has the potential to reduce the computational cost of optimization by reducing the dimensionality of the state space and enable large-scale visual-inertial mapping. Integrating information from other sensor modalities or extending the system for multi-camera settings are interesting directions for future research.

References

[1] S. Baker and I. Matthews, "Equivalence and efficiency of image alignment algorithms," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Comput. Soc, 2001.

[2] T. Barfoot, *State Estimation for Robotics*. Cambridge University Press, 2017.

[3] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 10, pp. 1053–1072, sep 2017.

[4] M. Bloesch, H. Sommer, T. Laidlow, M. Burri, G. Nützi, P. Fankhauser, D. Bellicoso, C. Gehring, S. Leutenegger, M. Hutter, and R. Siegwart, "A primer on the differential calculus of 3D orientations," *arXiv:1606.05285 [cs.RO]*, jun 2016.

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

[6] J. Civera, A. Davison, and J. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE Transactions on Robotics (TRO)*, vol. 24, no. 5, pp. 932–945, oct 2008.

[7] E. Eade, "Lie groups for computer vision," *Technical Report, Cambridge University*, 2014.

[8] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation," in *Proc. of Robotics: Science and Systems (RSS)*. Robotics: Science and Systems Foundation, jul 2015.

[9] J. Hsiung, M. Hsiao, E. Westman, R. Valencia, and M. Kaess, "Information sparsification in visual-inertial odometry," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, oct 2018.

[10] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "A first-estimates Jacobian EKF for improving SLAM consistency," in *Experimental Robotics*. Springer Berlin Heidelberg, 2009, pp. 373–382.

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

[12] A. Kasyanov, F. Engelmann, J. Stückler, and B. Leibe, "Keyframe-based visual-inertial online SLAM with relocalization," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, sep 2017.

[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 (IJRR)*, vol. 34, no. 3, pp. 314–334, dec 2014.

[14] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI)*, 1981, pp. 674–679.

[15] M. Mazuran, W. Burgard, and G. D. Tipaldi, "Nonlinear factor recovery for long-term SLAM," *The International Journal of Robotics Research (IJRR)*, vol. 35, no. 1-3, pp. 50–72, jun 2015.

[16] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid, "RSLAM: A system for large-scale mapping in constant-time using stereo," *International Journal of Computer Vision (IJCV)*, vol. 94, no. 2, pp. 198–214, jun 2010.

[17] J. Molnár, D. Chetverikov, and S. Fazekas, "Illumination-robust variational optical flow using cross-correlation," *Computer Vision and Image Understanding*, vol. 114, no. 10, pp. 1104–1114, oct 2010.

[18] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, apr 2007.

[19] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular SLAM with map reuse," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, apr 2017.

[20] T. Qin, J. Pan, S. Cao, and S. Shen, "A general optimization-based framework for local odometry estimation with multiple sensors," *arXiv:1901.03638 [cs.RO]*, jan 2019.

[21] N. Roma, J. Santos-Victor, and J. Tomé, "A comparative analysis of cross-correlation matching algorithms using a pyramidal resolution approach," in *Series in Machine Perception and Artificial Intelligence*. World Scientific, may 2002, pp. 117–142.

[22] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 32, no. 1, pp. 105–119, jan 2010.

[23] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to sift or surf," in *Proc. of the International Conference on Computer Vision (ICCV)*. Washington, DC, USA: IEEE Computer Society, 2011, pp. 2564–2571.

[24] T. Schneider, M. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski, and R. Siegwart, "Maplab: An open framework for research in visual-inertial mapping and localization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1418–1425, jul 2018.

[25] F. Steinbrücker, T. Pock, and D. Cremers, "Advanced data terms for variational optic flow estimation," in *Proceedings of the Vision, Modeling, and Visualization Workshop (VMV)*, Braunschweig, Germany, 2009.

[26] L. V. Stumberg, V. Usenko, and D. Cremers, "Direct sparse visual-inertial odometry using dynamic marginalization," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, may 2018.

[27] V. Usenko, J. Engel, J. Stückler, and D. Cremers, "Direct visual-inertial odometry with stereo cameras," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, may 2016.