Auton Robot
DOI 10.1007/s10514-017-9642-0

Edge alignment-based visual–inertial fusion for tracking of aggressive motions

Yonggen Ling[1] · Manohar Kuse[1] · Shaojie Shen[1]

Received: 4 February 2016 / Accepted: 1 June 2017

Abstract We propose a novel edge-based visual–inertial fusion approach to address the problem of tracking aggressive motions with real-time state estimates. At the front-end, our system performs edge alignment, which estimates the relative poses in the distance transform domain with a larger convergence basin and stronger resistance to changing lighting conditions or camera exposures compared to the popular direct dense tracking. At the back-end, a sliding-window optimization-based framework is applied to fuse visual and inertial measurements. We utilize efficient inertial measurement unit (IMU) preintegration and two-way marginalization to generate accurate and smooth estimates with limited computational resources. To increase the robustness of our proposed system, we propose to perform an edge alignment self check and IMU-aided external check. Extensive statistical analysis and comparison are presented to verify the performance of our proposed approach and its usability with resource-constrained platforms. Comparing to state-of-the-art point feature-based visual–inertial fusion methods, our approach achieves better robustness under extreme motions or low frame rates, at the expense of slightly lower accuracy in general scenarios. We release our implementation as open-source ROS packages.

Keywords Visual–inertial fusion · Edge alignment · Tracking of aggressive motions · Visual–inertial odometry

1 Introduction

Real-time, robust, and accurate state estimation is the foremost important component for many autonomous robotics applications. In particularly, reliable tracking of fast and aggressive motions is essential for popular applications that involves highly dynamic mobile platforms/devices, such as aerial robotics and augmented reality (Figs. 1, 2).

As demonstrated in the literature (Baker and Matthews 2004), cameras are the ideal sensors for tracking slow to moderate motions using feature-based methods under constant lighting conditions and camera exposures. However, large image displacement caused by fast motions can seriously downgrade the feature tracking performance. Recent advances in direct dense tracking have shown good adaptability to fast motions (Engel et al. 2014; Ling and Shen 2015; Newcombe et al. 2011). These methods operate on image intensities, rather than on sparse features, to minimize the photometric cost function and make full use of all the available information within an image. Thus they essentially bypass the feature processing pipeline and eliminate some of the issues found with feature-based methods. To ensure high image quality under different lighting conditions, camera auto-exposure is usually employed due to the same physical location in space being imaged as having different intensities across frames. The photo-consistency assumption behind direct dense tracking is easily proven wrong when the lighting conditions of the environment change. In contrast to

Electronic supplementary material The online version of this article (doi:10.1007/s10514-017-9642-0) contains supplementary material, which is available to authorized users.

✉ Yonggen Ling
ylingaa@connect.ust.hk

Manohar Kuse
mpkuse@ust.hk

Shaojie Shen
eeshaojie@ust.hk

[1] Department of Electronic and Computer Engineering,
The Hong Kong University of Science and Technology,
Hong Kong, SAR, China

Published online: 01 July 2017

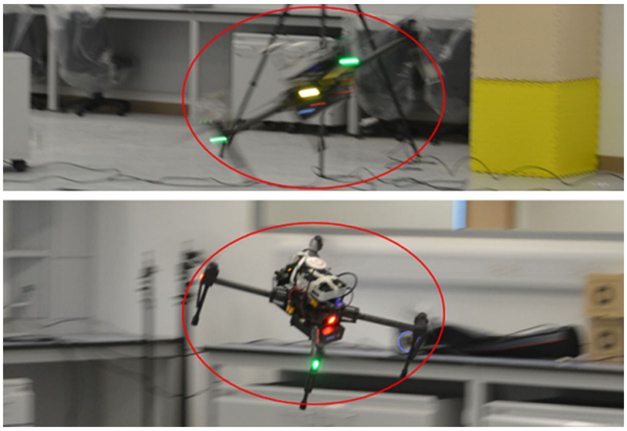


Fig. 1 Snapshots during aggressive flight. We highlight the position of the robot with *red circles*. The maximum linear velocity, linear acceleration, and angular velocity are 4.2 m/s, 9.6 m/s^2, and 245.1 °/s, respectively. Our proposed method is able to track aggressive motions even though the temporal image frequency is downsampled by several times (Color figure online)

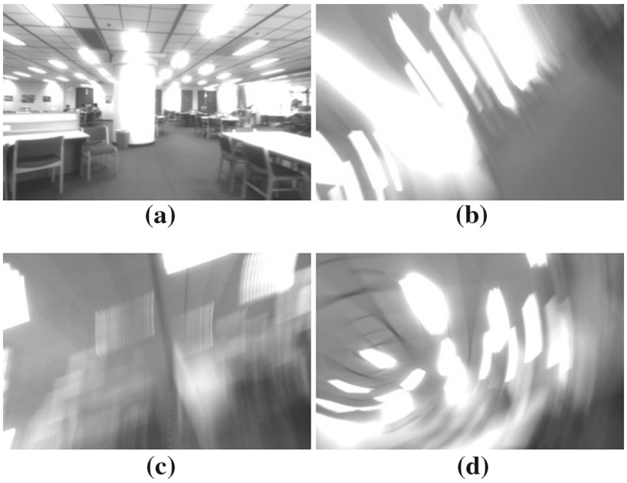


Fig. 2 Snapshots during challenging throw experiment. The maximum linear velocity is 4 m/s and maximum angular velocity is 1000 °/s. **a** Before sensor suite being thrown, **b–d** after sensor suite has been thrown. Our proposed method gives smooth and robust estimations in this experiment. See Sect. 7.3 for details

direct methods, our earlier work on edge alignment (Kuse and Shen 2016) uses the distance transform in the energy formulations and elegantly addresses the lack of photo-consistency issue. Nevertheless, it fails if the captured images undergo severe blurring. In contrast to cameras, IMUs generate noisy but outlier-free measurements, making them very effective for short-term tracking even under fast motions. On the flip side, low-cost IMUs suffer significant drift in the long run. We believe that combining the complementary nature of edge alignment and IMU measurements opens up the possibility of reliable tracking of aggressive motions.

Inspired by our earlier results towards fast motions (Ling and Shen 2015; Ling et al. 2016; Shen et al. 2013), in this work, we propose a novel approach that fuses complementary visual and inertial information for aggressive motion tracking using lightweight and off-the-shelf sensors. In contrast to existing visual–inertial fusion approaches, we explicitly address the problems of lighting variations and estimator convergence using edge alignment and graph-based nonlinear optimization. Our method uses information from a pair of calibrated stereo cameras and a MEMS IMU and runs onboard a moderate computer. The focus of this work is a semi-tightly-coupled, probabilistic, optimization-based estimation method that fuses pre-integrated IMU measurements and multi-constrained relative pose measurements from an edge alignment module. Our estimator actively searches for multi-constrained edge alignments between frames within a sliding window. This loop closure-like method enables the estimator to recover from complete loss of visual tracking and eliminate drifts after very aggressive motions. In addition, we initialize the incremental rotation for edge alignment using the angular prior from IMU measurements, which greatly improves the convergence property during aggressive motions. Extensive statistical analysis and comparison are presented to verify the performance of our proposed approach and its usability with resource-constrained platforms. Comparing to state-of-the-art point feature-based visual–inertial fusion methods, our approach achieves better robustness under extreme motions or low frame rates, at the expense of slightly lower accuracy in general scenarios. We release our code as open-source ROS packages with relevant video demonstrations available at https://github.com/ygling2008/direct_edge_imu.

The proposed system is an extension of our earlier papers (Kuse and Shen 2016; Ling et al. 2016), with improvements on edge alignment, co-estimation of IMU-camera extrinsics and IMU biases, system integration and performance evaluation. We believe that this contribution is an important milestone towards a practical visual–inertial system for tracking of aggressive motions, which would enable applications such as autonomous agile quadrotor flight and augmented reality. The remainder of the paper is organized as follows. In Sect. 2, we review the state-of-the-art scholarly work. An overview of the system is presented in Sect. 3. Notations are given in Sect. 4. Edge alignment is introduced in Sect. 5. Details of the sensor fusion framework are discussed in Sect. 6, and Sect. 7 shows implementation details and experimental evaluations. Finally, Sect. 8 draws conclusions and points out possible future extensions.

2 Related work

There has been extensive scholarly work done in relation to visual odometry, image registration, point cloud registration, and visual–inertial state estimation. Visual measurements can be calculated from different camera configurations, such as monocular (Hesch et al. 2014; Li and Mourikis 2013; Scaramuzza et al. 2014; Shen et al. 2013, 2015), stereo (Leutenegger et al. 2015), or RGB-D cameras (Huang et al. 2011). The majority of these approaches rely on detecting and tracking sparse features across multiple frames. The most well known approach is called the Kanade–Lucas–Tomasi (KLT) algorithm (Shi and Tomasi 1994; Tomasi and Kanade 1991) and uses a feature detector and a feature tracker that make full use of spatial intensity information to reduce potential matches between images. Thus, it is faster than the traditional image alignment methods. Many variants of the KLT algorithm have been developed and some of them are summarized in Baker and Matthews (2004). In Baker and Matthews (2004), an inverse compositional approach, which greatly improves the efficiency of the KLT algorithm, is presented. Other mainstream sparse feature algorithms are based on descriptors, such as FAST (Rosten and Drummond 2006), Haris (Harris and Pike 1987), Shi–Tomasi (Shi and Tomasi 1994), SIFT (Lowe 2004), and SURF (Bay et al. 2008). Sparse features are firstly detected and described, and then matched according to the distance in the feature descriptor space. Though feature-based methods are well-developed, they depend heavily on image quality for feature detection and small image displacement for feature tracking. They fail when cameras undergo aggressive motions that lead to large image movement and severe motion blur.

With recent advances in high-performance mobile computing, direct dense methods have become popular (Engel et al. 2013, 2014; Kerl et al. 2013; Newcombe et al. 2011). Kerl et al. (2013) propose a probabilistic formulation of direct dense tracking that is based on student distribution, which alleviates the influence of outliers and leads to robust estimation. Newcombe et al. (2011) present a system for real-time camera tracking and reconstruction using current commodity GPU hardware. Recent work on direct dense tracking (Engel et al. 2013, 2014) models the uncertainty on the inverse depth of pixels and exhibits amazing performance towards a large scale environment. However, these methods rely on the photo-consistency assumption, by which motion estimation can be done by following the local gradient directions to minimize the total intensity error. Direct dense methods have a small basin of attraction (as noted in Kerl et al. 2013) and are sensitive to changing lighting conditions.

Another way to estimate the states using visual measurements is an iterative closest point (ICP) based method, which directly aligns three-dimensional point clouds. Stückler and Behnke (2012) employ a method based on ICP for the alignment of point clouds obtained from an RGB-D camera, while Rusinkiewicz and Levoy (2001) present a survey of other attempts to use efficient ICP-like methods for pose estimation. A generalized formulation of ICP is proposed in Segal et al. (2005), in which ICP and point-to-plane ICP are combined into a single probabilistic framework. Fitzgibbon (2003) proposes an algorithm to align two two-dimensional point sets, and the algorithm is also extensible to three-dimensional point sets. Fitzgibbon (2003) uses the distance transform to model the point correspondence function to align two-dimensional curves. Since ICP relies on three-dimensional points clouds, its applications to monocular or stereo cameras, which generally give neither dense nor accurate three-dimensional points, are limited.

To overcome the disadvantages of approaches with vision only, visual–inertial fusion has recently gained lots of attentions. It is straightforward to apply some variations of Kalman filtering (Bloesch et al. 2015; Huang et al. 2011; Omari et al. 2015; Shen et al. 2013; Scaramuzza et al. 2014) to loosely fuse visual and inertial measurements. Huang et al. (2011) utilizes the information richness of RGB-D cameras and fuses the visual tracking and inertial measurements in an extended Kalman filter (EKF) framework. Shen et al. (2013) combines the information from a KLT tracker and IMU measurements with an unscented Kalman filter (UKF), and Omari et al. (2015) leverages the recent development of direct dense tracking and fuses it with inertial information in the EKF fashion. Scaramuzza et al. (2014) also applies EKF in a similar way to Shen et al. (2013). Intensity errors of image patches as well as inverse depth parametrization are considered in Bloesch et al. (2015). The high-level effect of such fusion is the smoothing of vision-based tracking. Even when visual tracking fails, the estimation can be done by the use of an IMU for short-term motion prediction. In loosely-coupled methods, visual measurements are usually presented in the form of relative pose transformations, while leaving the visual pose tracking as a black box. This leads to lower computational complexity at the cost of suboptimal results.

Recent developments in visual–inertial fusion indicates that tightly-coupled methods outperform their loosely-coupled counterparts in terms of estimation accuracy (Christian et al. 2015; Huang et al. 2014; Hesch et al. 2014; Leutenegger et al. 2015; Li and Mourikis 2013; Kuse and Shen 2016; Usenko et al. 2016). Shen et al. (2015), Christian et al. (2015) and Hesch et al. (2014) propose a scheme of IMU preintegration on the Lie manifold and then fuse it with monocular camera tracking information in a tightly-coupled graph-based optimization framework. The inertial measurement integration approaches in Leutenegger et al. (2015), Shen et al. (2015) and Christian et al. (2015) are slightly different, with respective pros and cons. Hesch et al. (2014), Li and Mourikis (2013) and Huang et al. (2014) put emphasis on the system's observability and build up the mathematical foundation of visual–inertial systems. Upon their analysis, they develop monocular visual–inertial systems that are high-precision as well as consistent. Besides this, Hesch et al. (2014), Li and Mourikis (2013), Dong-Si and Mourikis (2012), Heng et al. (2014) and Yang and Shen (2015) relax the

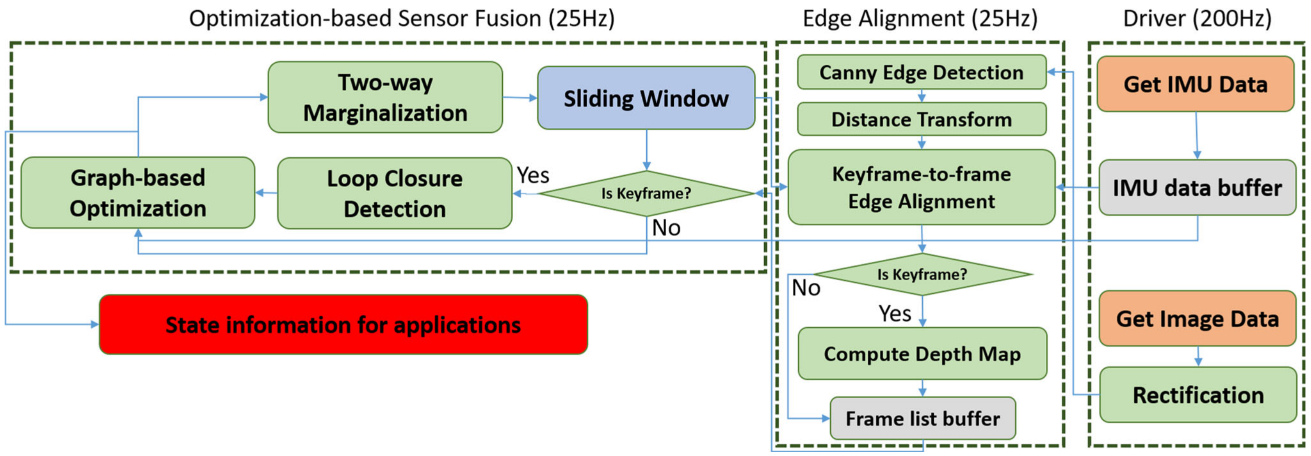


Fig. 3 The pipeline of our proposed system. It consists of three modules running in separate threads to ensure real-time availability of state estimates (denoted as *three dashed boxes*). The driver thread runs at 200 Hz, the edge alignment thread runs at 25 Hz, and the optimization-based sensor fusion thread runs at 25 Hz

assumption that the transformation between the camera and IMU (cameras extrinsics) is known. IMU-camera extrinsics are also optimized in their algorithms, which takes a step forward towards practical applications. Instead of geometric errors of sparse features, Usenko et al. (2016) propose a direct visual–inertial odometry that minimizes the intensity errors. Tightly-coupled methods consider the coupling between two types of measurements and allows the adoption of a graph optimization-based framework with the ability to iteratively re-linearize nonlinear functions. In this way, tightly-coupled approaches gain the potential to achieve better performance. On the other hand, these kinds of methods usually come with a higher computational cost as the number of variables involved in the optimization is large.

Table 1 Average computation time of the main time-consuming components of our proposed system

Component	Average computation time (ms)	Thread
Driver	1	0
Edge alignment	11	1
Canny edge detection	2	1
Distance transform	2	1
Block matching	8	1
Graph optimization	6	2
Two-way marginalization	3	2
Loop closure	28	2

3 System overview

The pipeline of our proposed system is illustrated in Fig. 3 (also see Table 1 in Sect. 7.1). Three threads run simultaneously, utilizing the multi-core architecture.

The first thread is the driver thread, which performs basic operations, such as data acquisition and image rectification.

The edge alignment (Sect. 5) thread performs key-frame-to-frame edge alignment periodically. Canny edge detection and distance transform are performed for each incoming image. The angular prior from the integration of gyroscope measurements initializes the incremental rotation. This thread also identifies instantaneous tracking performance, detects tracking failure and determines whether to add a new keyframe. A disparity map is computed using a standard block matching algorithm for every new keyframe. The visual measurements and their corresponding frames are stored in a frame list buffer for further processing by the optimization thread.

The optimization-based sensor fusion thread maintains a sliding window of states and measurements, and checks the frame list buffer periodically. If it is not empty, all the frames within the buffer will be added into the sliding window. If a keyframe is added, loop closure detection is performed to find possible visual connections between keyframes. Graph-optimization is then applied to find the maximum a posteriori estimate of all the states within the sliding window using measurements from IMU pre-integration (Sect. 6.2), multi-constrained relative pose measurements (Sect. 6.3) and the prior. A two-way marginalization scheme that selectively removes states is used to bound the computational complexity and the time interval of the IMU integration and to maximize the information stored within the sliding window (Sect. 6.5).

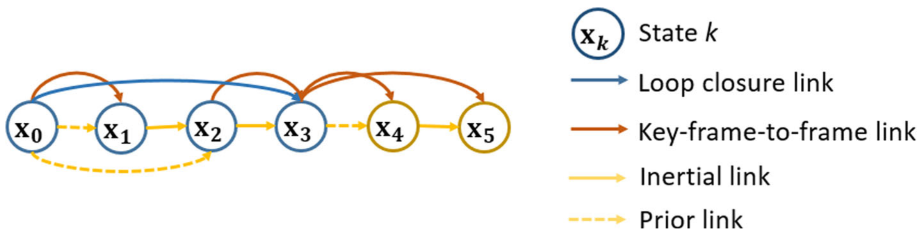


Fig. 4 Graph representation of variables ($\mathbf{x}_k = [\mathbf{p}^w_{b_k}, \mathbf{v}^w_{b_k}, \mathbf{q}^w_{b_k}, \mathbf{b}^{b_k}_a, \mathbf{b}^{b_k}_\omega]$) and constraints (inertial links, prior links, keyframe-to-frame links and loop closure links). See Sect. 6 for details about variables and constraints

4 Notations

We begin by giving notations. We consider $(\cdot)^w$ as the earth's inertial frame, $(\cdot)^{b_k}$ and $(\cdot)^{c_k}$ as the IMU body frame and camera frame while taking the kth image. We assume that the IMU-camera sensor suite is rigidly mounted, and the translation and rotation between the left camera and the IMU are $\mathbf{t}^c_b$, $\mathbf{q}^c_b$. The intrinsics of stereo cameras are calibrated beforehand. $\mathbf{p}^X_Y$, $\mathbf{v}^X_Y$ and $\mathbf{R}^X_Y$ are the 3D position, velocity and rotation of camera frame Y with respect to frame X, respectively. We also have the corresponding quaternion ($\mathbf{q}^X_Y = \left[q_x, q_y, q_z, q_w\right]$) representation for rotation. Hamilton notation is used for quaternions. The states are defined as the combinations of positions, velocities, rotations, linear acceleration biases and angular velocity biases $\mathbf{x}_k = [\mathbf{p}^w_{b_k}, \mathbf{v}^w_{b_k}, \mathbf{q}^w_{b_k}, \mathbf{b}^{b_k}_a, \mathbf{b}^{b_k}_\omega]$. For camera frame c_r (which denotes the reference frame) and camera frame c_n (which denotes the current frame), the rigid-body transformation between them is $\mathbf{T}^n_r = \{\mathbf{p}^{c_n}_{c_r}, \mathbf{R}^{c_n}_{c_r}\} \in \mathbf{SE}(3)$, where $\mathbf{p}^{c_n}_{c_r}$ and $\mathbf{R}^{c_n}_{c_r}$ are translation and rotation, respectively. Next we denote a 3D scene point i in the co-ordinate system of the camera optical center at time instance k by ${}^k\mathbf{f}_i \in \mathbb{R}^3$. The camera projection function $\Pi\colon \mathbb{R}^3 \mapsto \mathbb{R}^2$ projects the visible 3D scene point onto the image domain. The inverse projection function $\tilde{\Pi}\colon (\mathbb{R}^2, \mathbb{R}) \mapsto \mathbb{R}^3$ back-projects a pixel coordinate given the depth at this pixel co-ordinate:

$$
{}^k\mathbf{u}_i = \Pi\left({}^k\mathbf{f}_i\right) \tag{1}
$$

$$
{}^k\mathbf{f}_i = \tilde{\Pi}\left({}^k\mathbf{u}_i, Z_k\left({}^k\mathbf{u}_i\right)\right), \tag{2}
$$

where ${}^k\mathbf{u}_i \in \mathbb{R}^2$ denotes the image coordinates of the 3D point ${}^k\mathbf{f}_i$, and $Z_k({}^k\mathbf{u})$ denotes the depth of point ${}^k\mathbf{f}_i$. We use a graph structure to represent the variables (states, combination of poses and velocities) we aim to solve and constraints (links) between variables. See Fig. 4 for an illustration and Sect. 6. for details about variables and constraints.

5 Edge alignment

In this section, we introduce our formulation for relative camera motion estimation, which we refer to as the edge alignment formulation. It is based on the minimization of the geometric error term at each edge pixel to obtain an estimate of the rigid body transform between two frames, ie., to find a pose (rotation and translation matrix) such that the edges of the two images align. This is in contrast to previous direct methods, notably the one proposed by Kerl et al. (2013), which minimizes the photometric error at every pixel. The energy formulation we propose in this work is the sum of the squared distances between transformed-projected (on current frame) co-ordinates of the edge-pixels from the reference frame and the nearest edge-pixels in the current frame.

5.1 Formulation

For convenience of notation we derive our energy formulation using $\mathbf{R}$ and $\mathbf{p}$ as the alias to $\mathbf{R}^{c_n}_{c_r}$ and $\mathbf{p}^{c_n}_{c_r}$ for a particular instance of the reference and current frames. Our proposed geometric energy function is the sum of the distances of the re-projections (of edge points from the reference image) and nearest edge points in the current image:

$$
f(\mathbf{R}, \mathbf{p}) = \sum_i \min_j D^2\left(\Pi[\mathbf{R}\,{}^r\mathbf{f}_i + \mathbf{p}],\ {}^n\mathbf{u}_j\right), \tag{3}
$$

where $D\colon (\mathbb{R}^2, \mathbb{R}^2) \mapsto \mathbb{R}$ denotes the Euclidean distance between those points. The best estimates for the rigid transform can be obtained by solving the following optimization problem:

$$
\begin{aligned}
&\underset{\mathbf{R},\mathbf{p}}{\text{minimize}} \quad f(\mathbf{R}, \mathbf{p}) \\
&\text{subject to} \quad \mathbf{R} \in \mathbf{SO}(3).
\end{aligned} \tag{4}
$$

We relax the geometric energy function by restricting it only to edge points. In this approach, we observe that, if the image points corresponding to edge points in the reference image (denoted by ${}^r\mathbf{e}_i \in \mathbb{R}^2$ with corresponding 3D point ${}^r\mathbf{E}_i$) are pre-selected, then the function $\min_j D(\mathbf{u}_i, \mathbf{u}_j)$ is exactly the definition of the distance transform (Felzenszwalb and Huttenlocher 2012). We denote the distance transform of the edge-map of the current image as $V^{(n)}\colon \mathbb{R}^2 \mapsto \mathbb{R}$. Thus, the energy terms for an edge-pixel of the reference frame are given by

$$
\upsilon_{\mathbf{e}_i}(\mathbf{R}, \mathbf{p}) = V^{(n)}\left(\Pi\left[\mathbf{R}\,\tilde{\Pi}\left({}^r\mathbf{e}_i, Z_r\left({}^r\mathbf{e}_i\right)\right) + \mathbf{p})\right]\right). \tag{5}
$$

To summarize, the relaxed energy function is

$$f(\mathbf{R}, \mathbf{p}) = \sum\nolimits_{\forall \mathbf{e}_i} \left(\upsilon_{\mathbf{e}_i}(\mathbf{R}, \mathbf{p})\right)^2 \tag{6}$$

and our goal is to solve for $\mathbf{R}^*$ and $\mathbf{t}^*$:

$$\begin{aligned} &arg \min_{\mathbf{R},\mathbf{p}} \quad f(\mathbf{R}, \mathbf{p}) \\ &\text{subject to} \quad \mathbf{R} \in \mathbf{SO}(3). \end{aligned} \tag{7}$$

5.2 Optimization on Lie group manifolds

Since (5) is highly nonlinear with respect to $\mathbf{R}$ and $\mathbf{p}$, we linearize it on the Lie group manifolds $\mathbf{SE}(3)$ with respect to $\boldsymbol{\xi} = (\delta\mathbf{p}, \delta\boldsymbol{\theta}) \in \mathfrak{se}(3)$, which is the minimum dimension error representation,

$$\upsilon_{\mathbf{e}_i}(\mathbf{R}, \mathbf{p}, \boldsymbol{\xi}) = \upsilon_{\mathbf{e}_i}(\mathbf{R}, \mathbf{p}) + \mathbf{J}_{i\,|\boldsymbol{\xi}=\mathbf{0}} \cdot \boldsymbol{\xi}, \tag{8}$$

where $\mathbf{J}_i$ is the Jacobian matrix of $\upsilon_{\mathbf{e}_i}(\mathbf{R}, \mathbf{p})$ with respect to $\boldsymbol{\xi}$ at $\boldsymbol{\xi} = \mathbf{0}$. Lie group $\mathbf{SE}(3)$ and Lie algebra $\mathfrak{se}(3)$ can be linked by an exponential map and logarithm map. More details can be found in Ma et al. (2012).

Unlike in our early work Kuse and Shen (2016), which provides a strong theoretical guarantee on convergence, we adopt the Gaussian–Newton method to solve (7). We empirically find that the Gaussian–Newton method works in most cases and it converges to the local minimum quickly so the real-time requirement of our proposed system is satisfied. To get rid of the disturbance caused by convergence, we have mechanisms to detect and reject failure of edge alignment (Sects. 5.3, 6.4).

Following the scheme of the Gaussian–Newton approach, we iteratively solve (7) using the linearization (8) around the current estimate $\hat{\mathbf{T}} = \{\hat{\mathbf{R}}, \hat{\mathbf{p}}\}$ and then perform incremental updates until convergence:

$$\hat{\mathbf{T}} \leftarrow \hat{\mathbf{T}} \otimes \exp(\boldsymbol{\xi}). \tag{9}$$

Substituting (8) into (7) and then taking the derivative with respect to $\boldsymbol{\xi}$ and setting it to zero leads to the following system:

$$\mathbf{J}^{\mathrm{T}}\mathbf{J}\boldsymbol{\xi} = -\mathbf{J}^{\mathrm{T}}\boldsymbol{\varsigma}, \tag{10}$$

where $\mathbf{J}$ is a Jacobian matrix that is formed by stacking Jacobians $\mathbf{J}_i$ and $\boldsymbol{\varsigma}$ is the corresponding vector that is formed by stacking $\upsilon_{\mathbf{e}_i}(\mathbf{R}, \mathbf{p})$ together.

As has been observed by Kerl et al. (2013), weighting large residues can help alleviate the effect of outliers arising due to reflections, occlusions, disocclusions and edge-map misses. We use the Laplacian weighting term given by $w(\upsilon_{\mathbf{e}_i}(\xi)) = e^{-\upsilon_{\mathbf{e}_i}(\xi)}$ and rewrite (10) as a weighted formulation:

$$\mathbf{J}^{\mathrm{T}}\mathbf{W}\mathbf{J}\boldsymbol{\xi} = -\mathbf{J}^{\mathrm{T}}\mathbf{W}\boldsymbol{\varsigma}, \tag{11}$$

where $\mathbf{W}$ is a diagonal matrix that encodes the Laplacian weights. Figure 5 shows the reprojections of edges-pixels as the Gaussian–Newton optimization progresses.

5.3 Edge alignments self check

The proposed edge alignment abstracts the image as edges and optimizes a function based on the distance transform for the relative pose for the key-frame-to-current-frame. This results in an increased convergence basin and robustness towards changing lighting conditions. Inspite of its robustness, under certain extreme situations, edge alignment tends to produce estimates with high uncertainty. This is detrimental to the overall performance of the system and detection of such an event is crucial. For example, aggressive motions can cause severe motion-blur in captured images, the effect being that the Canny edge detection module results in temporally inconsistent edges. For another example, when captured images undergo changing lighting conditions, the detected edges may also be inconsistent among consecutive images.

We propose to use the average reprojected distance as the criterion for the self check and reporting of failures. We evaluate the value of the cost function (6) at the final iteration ($f(\mathbf{R}^*, \mathbf{p}^*)$) divided by the number of edge pixels. An appropriate threshold is set to detect failure of convergence.

6 Sliding window-based sensor fusion

Given two time instants that correspond to two images, we can write the IMU propagation model for position, velocity and rotation with respect to the earth's inertial frame:

$$\begin{aligned} \mathbf{p}^w_{b_{k+1}} &= \mathbf{p}^w_{b_k} + \mathbf{v}^w_{b_k}\Delta t - \mathbf{g}^w \Delta t^2/2 + \mathbf{R}^w_{b_k}\boldsymbol{\alpha}^k_{k+1} \\ \mathbf{v}^w_{b_{k+1}} &= \mathbf{v}^w_{b_k} + \mathbf{R}^w_{b_k}\boldsymbol{\beta}^k_{k+1} - \mathbf{g}^w \Delta t \\ \mathbf{q}^w_{k+1} &= \mathbf{q}^w_k \otimes \mathbf{q}^k_{k+1}, \end{aligned} \tag{12}$$

where Δt is the interval between two image acquisitions, and $\mathbf{g}^w = [0\ 0\ 9.8]$ is the gravity vector in the earth's inertial frame. $\boldsymbol{\alpha}^k_{k+1}$, $\boldsymbol{\beta}^k_{k+1}$ and $\mathbf{q}^k_{k+1}$ are obtained by integrating the IMU measurements between time instants k and $k+1$, with the definition detailed in Sect. 6.2.

6.1 State estimation formulation

We set the initial position and yaw angle to be zero, and define the full state vector as:

$$\mathcal{X} = \left[\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{t}^c_b, \mathbf{q}^c_b\right],$$

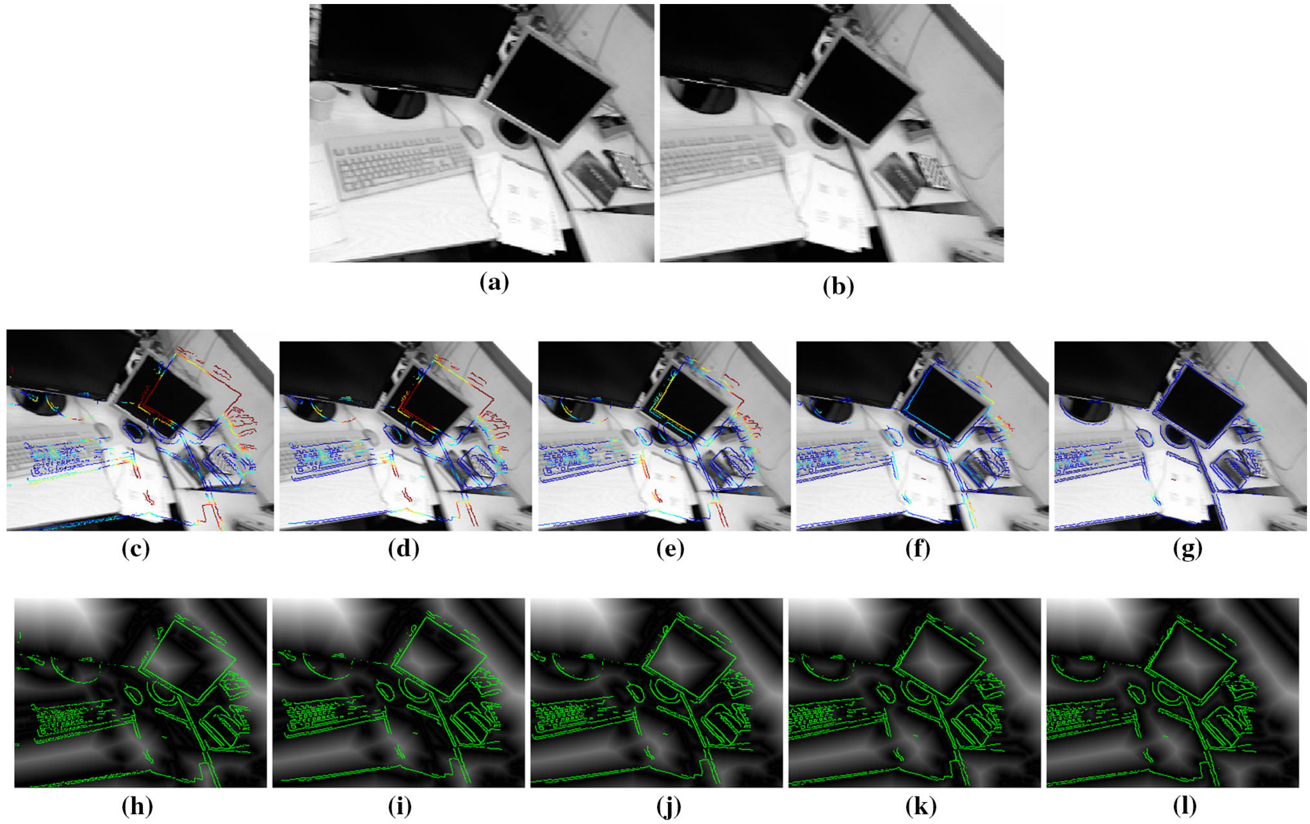


Fig. 5 Reprojections of edge-pixels in the reference frame onto the current frame as the Gaussian–Newton optimization progresses. The *middle row* shows the reprojections on the current gray image. They are false *colored* to represent $v_{e_i}(\xi)$. The last row shows reprojections on the distance transform image of the edge-map of the current frame. Note that the current frame and the reference frame are about 160 ms apart (5 frames) and the Gaussian–Newton method progress is shown without pyramids, with the initial guess as the identity. Viewing in *color* is recommended. **a** Reference image I_r , **b** current image I_n, **c** iteration 0, **d** iteration 2, **e** iteration 4, **f** iteration 6, **g** iteration 8, **h** iteration 0, **i** iteration 2, **j** Iteration 4, **k** iteration 6, **l** iteration 8 (Color figure online)

where $\mathbf{x}_k = [\mathbf{p}^w_{b_k}, \mathbf{v}^w_{b_k}, \mathbf{q}^w_{b_k}, \mathbf{b}^{b_k}_a, \mathbf{b}^{b_k}_\omega]$. We aim to obtain a maximum a posteriori (MAP) estimate by minimizing the sum of the Mahalanobis norm of the weighted visual measurement residuals, inertial measurement residuals and the prior:

$$\begin{aligned}\underset{\mathcal{X}}{\text{minimize}}\ &||\mathbf{b}_p - \mathbf{H}_p\mathcal{X}||^2 + \sum_{k\in S_i} ||r_{S_i}\left(\hat{\mathbf{z}}^k_{k+1}, \mathcal{X}\right)||^2_{\mathbf{P}^k_{k+1}} \\ &+ \sum_{(i,j)\in S_c} ||r_{S_c}\left(\hat{\mathbf{z}}^j_i, \mathcal{X}\right)||^2_{\left(\mathbf{W}^j_i\right)^{-1}\mathbf{P}^j_i} \end{aligned} \tag{13}$$

where S_i and S_c are the set of inertial and visual measurements respectively, $r_{S_i}(\hat{\mathbf{z}}^k_{k+1}, \mathcal{X})$ is the residual function that measures the residual between the inertial measurements and $\mathcal{X}$ with covariance $\mathbf{P}^k_{k+1}$, while $r_{S_c}(\hat{\mathbf{z}}^j_i, \mathcal{X})$ is the residual function that measures the reprojection error between the visual measurements and $\mathcal{X}$ with covariance $\mathbf{P}^j_i$. Since visual measurements are subject to failure, we added a diagonal matrix $\mathbf{W}^j_i$ to weight the influence of $r_{S_c}(\hat{\mathbf{z}}^j_i, \mathcal{X})$. $\mathbf{b}_p$ and $\mathbf{H}_p$ are the prior of the states, which will be detailed in the two-way marginalization section (Sect. 6.5). Inertial measurements are obtained by IMU preintegration (Sect. 6.2) and visual measurements are obtained by multi-constrained edge alignments (Sect. 6.3).

6.2 IMU preintegration

We adopt the IMU preintegration approach proposed in Yang and Shen (2016). The linear acceleration $\mathbf{a}^{b_t}$ and angular velocity $\boldsymbol{\omega}^{b_t}$ at time t are modeled as

$$\mathbf{a}^{b_t} = \mathbf{a}^{b_t *} + \mathbf{b}^{b_t}_a + \mathbf{n}^{b_t}_a \tag{14}$$

$$\boldsymbol{\omega}^{b_t} = \boldsymbol{\omega}^{b_t *} + \mathbf{b}^{b_t}_\omega + \mathbf{n}^{b_t}_\omega \tag{15}$$

where $\mathbf{a}^{b_t *}$ and $\boldsymbol{\omega}^{b_t *}$ are true values, $\mathbf{b}^{b_t}_a$ and $\mathbf{b}^{b_t}_\omega$ are slowly varying biases which are modeled as Gaussian random walks, and $\mathbf{n}^{b_t}_a$ and $\mathbf{n}^{b_t}_\omega$ are additive Gaussian white noises.

The integration from IMU measurements between time instants k and $k+1$ is

$$\hat{\mathbf{z}}^k_{k+1} = \begin{bmatrix} \hat{\boldsymbol{\alpha}}^k_{k+1} \\ \hat{\boldsymbol{\beta}}^k_{k+1} \\ \hat{\mathbf{q}}^k_{k+1} \end{bmatrix} = \begin{bmatrix} \int_{t\in[k,k+1]} \mathbf{R}^k_t \mathbf{a}^{b_t} dt^2 \\ \int_{t\in[k,k+1]} \mathbf{R}^k_t \mathbf{a}^{b_t} dt \\ \int_{t\in[k,k+1]} \frac{1}{2} \begin{bmatrix} -\lfloor \boldsymbol{\omega}^{b_t} \times \rfloor & \boldsymbol{\omega}^{b_t} \\ -\boldsymbol{\omega}^{b_t\,T} & \mathbf{0} \end{bmatrix} \mathbf{q}^k_t dt \end{bmatrix}. \tag{16}$$

The residual function between the states and the IMU integration is defined as

$$r_{S_i}\left(\hat{\mathbf{z}}^k_{k+1}, \mathcal{X}\right) = \begin{bmatrix} \delta\boldsymbol{\alpha}^k_{k+1} \\ \delta\boldsymbol{\beta}^k_{k+1} \\ \delta\boldsymbol{\theta}^k_{k+1} \\ \delta\mathbf{b}^{b_k}_a \\ \delta\mathbf{b}^{b_k}_\omega \end{bmatrix}$$
$$= \begin{bmatrix} \mathbf{R}^{b_k}_w \left(\mathbf{p}^w_{b_{k+1}} - \mathbf{p}^w_{b_k} - \mathbf{v}^w_{b_k}\Delta t + \mathbf{g}^w \frac{\Delta t^2}{2}\right) - \hat{\boldsymbol{\alpha}}^k_{k+1} \\ \mathbf{R}^{b_k}_w \left(\mathbf{v}^w_{b_{k+1}} - \mathbf{v}^w_{b_k} + \mathbf{g}^w \Delta t\right) - \hat{\boldsymbol{\beta}}^k_{k+1} \\ 2\left[\left(\hat{\mathbf{q}}^k_{k+1}\right)^{-1} \left(\mathbf{q}^w_{b_k}\right)^{-1} \mathbf{q}^w_{b_{k+1}}\right]_{xyz} \\ \mathbf{b}^{b_{k+1}}_a - \mathbf{b}^{b_k}_a \\ \mathbf{b}^{b_{k+1}}_\omega - \mathbf{b}^{b_k}_\omega \end{bmatrix}. \tag{17}$$

The covariance $\mathbf{P}^k_{k+1}$ can be calculated by iteratively linearizing the continuous-time dynamics of the error term and then updating it with discrete-time approximation:

$$\begin{aligned} \mathbf{P}^k_{t+\delta t} &= (\mathbb{I} + \mathbf{F}_t\delta t) \cdot \mathbf{P}^k_t \cdot (\mathbb{I} + \mathbf{F}_t\delta t)^{\mathrm{T}} \\ &\quad + (\mathbb{I} + \mathbf{G}_t\delta t) \cdot \mathbf{Q}_t \cdot (\mathbb{I} + \mathbf{G}_t\delta t)^{\mathrm{T}}, \end{aligned} \tag{18}$$

with the initial condition $\mathbf{P}^k_k = \mathbf{0}$. $\mathbf{F}_t$ and $\mathbf{G}_t$ are the state transition Jacobians with respect to the states and the IMU measurement noise, respectively. Detailed derivations can be found in Yang and Shen (2016). IMU preintegration forms constraints between consecutive state variables (inertial links in the graph model, see Fig. 4).

6.3 Multi-constrained edge alignments

Edge alignment (Sect. 5) is performed between the latest keyframe within the sliding window and the latest incoming frame (also referred to as the current frame). The resultant visual measurements are named key-frame-to-frame links in the graph model (Fig. 4). In addition, since significant drifts may occur after aggressive motions, we introduce a local loop closure module for recovery. Once a new keyframe is added, loop closure detection is performed to seek possible visual measurements between existing key-frames within the sliding window and the new keyframe using edge alignment. Note that the cross check is adopted to avoid wrong loop closure. If and only if the two corresponding estimated rigid-body transformations are consistent, the cross check is passed. The outputs from the loop closure detection module are denoted as loop closure links in the graph model (Fig. 4).

Suppose the visual measurement between reference frame i and aligned frame j obtained from edge alignment is $\hat{\mathbf{z}}^j_i = \mathbf{T}^{j\,*}_i$. The residual function is defined as

$$r_{S_c}\left(\hat{\mathbf{z}}^j_i, \mathcal{X}\right) = \begin{bmatrix} \delta\mathbf{p}^j_i \\ \delta\boldsymbol{\theta}^j_i \end{bmatrix}$$
$$= \begin{bmatrix} \mathbf{R}^{b_j}_w \left(\mathbf{p}^w_{b_i} - \mathbf{p}^w_{b_j}\right) - \mathbf{R}^b_c \left(\mathbf{R}^{c_j}_{c_i} \mathbf{t}^c_b + \hat{\mathbf{p}}^{c_j}_{c_i}\right) - \mathbf{t}^b_c \\ 2\left[\left(\mathbf{q}^c_b \hat{\mathbf{q}}^{c_j}_{c_i} \mathbf{q}^c_b\right)^{-1} \left(\mathbf{q}^w_{b_j}\right)^{-1} \mathbf{q}^w_{b_i}\right]_{xyz} \end{bmatrix}, \tag{19}$$

where $\hat{\mathbf{q}}^j_i$ is the quaternion representation of $\hat{\mathbf{R}}^j_i$, and vice versa. It can be derived mathematically that the corresponding covariance $\mathbf{P}^j_i$ is the inverse of the Hessian matrix $\mathbf{J}^{\mathrm{T}}\mathbf{W}\mathbf{J}$ at the final Gaussian–Newton iteration.

6.4 IMU-aided external check

Since the IMUs provide noisy but outlier-free measurements, the estimation using IMU preintegration is short-term reliable. Moreover, though we can tune the related parameters so that the edge alignment exhibits good performance, it fails as the surroundings are complicated and unknown in advance. Additionally, tuning the parameters is not an easy job as different external conditions may result in different parameter settings. We propose to use the IMU preintegration to threshold the performance of instantaneous edge alignment. The characteristics of an IMU can be easily calculated offline and are supposed to be known prior to the starting of the system. We can detect possible false edge alignment according to the difference between the IMU preintegration and edge alignment estimate. We ignore the visual measurements from edge alignment if they are not consistent with the IMU preintegration (both for the rotation and translation estimation). An IMU-aided external check is a vitally important step towards a practical and robust system.

We declare edge alignment estimates as failures if either of the criteria (Sects. 5.3, 6.4) fail.

6.5 Two-way marginalization

Due to the limited memory and computational resources of the system, we can only maintain a certain number of states and measurements within the sliding window. We convert states that carry less information into priors $\boldsymbol{\Lambda}_p$, $\mathbf{b}_p\}$ by marginalization, where $\boldsymbol{\Lambda}_p = \mathbf{H}^T_p\mathbf{H}_p$. Note that the effectiveness of loop closure (Sect. 6.3) and drift elimination depends on whether an older state is kept within the sliding window. For this reason, unlike traditional approaches, which only

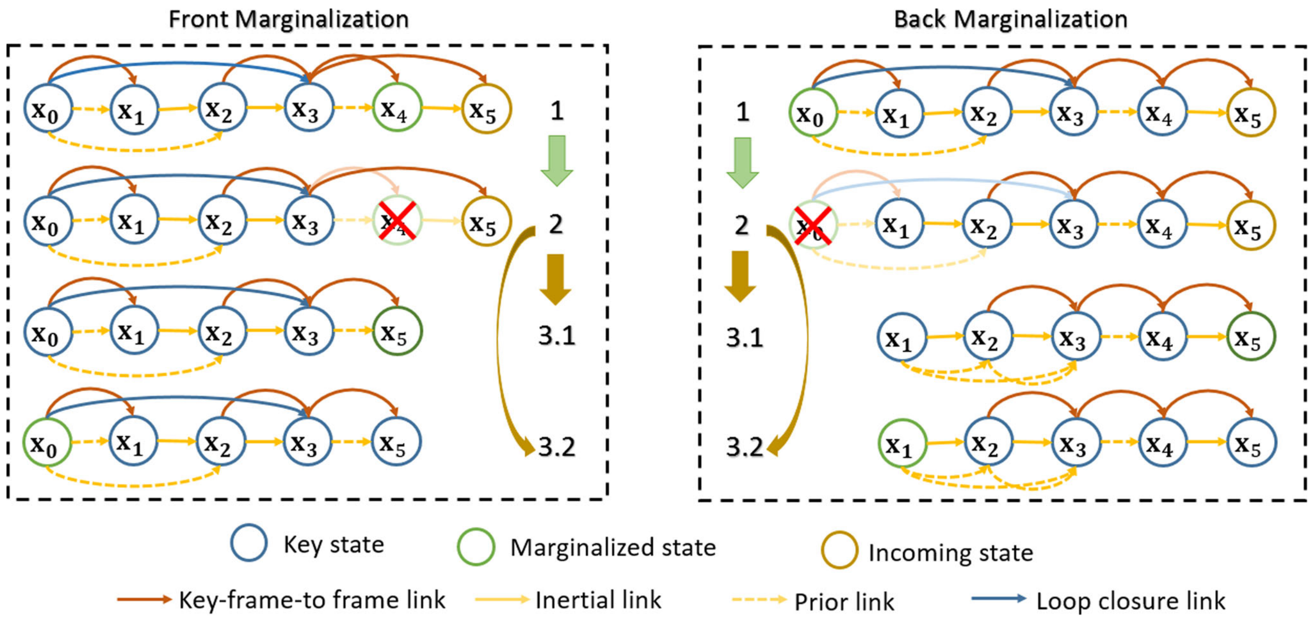


Fig. 6 The process of our two-way marginalization, which marginalizes all the available information (motion estimates from edge alignment, inertial measurement, loop closure relation and prior) into a new prior and maintains bounded computational complexity. Front marginalization marginalizes the second newest state, while back marginalization marginalizes the oldest state within the sliding window (Color figure online)

marginalize old states, we use the two-way marginalization scheme that was first introduced in our earlier work, Shen et al. (2014), to selectively remove old or more recent states in order to enlarge the covered regions of the sliding window.

Figure 6 illustrates the process of our two-way marginalization. Front marginalization removes the second newest state, while back marginalization removes the oldest state. Blue circles represent key states, green circles represent the states to be marginalized and brown circles represent the incoming states. The relation between states and frames is that states include poses and velocities, while frames include poses and images. Each frame has its corresponding state and vice versa. States are linked by IMU preintegration (inertial link), incremental edge alignment (tracking link), loop closure (loop closure link) and the prior (prior link). To perform front marginalization, the second newest state is first linked with the incoming state (step 1) and then marginalized out (step 2). For back marginalization, the oldest state is simply marginalized out (steps 1–2). After marginalization, the third step decides which state is to be marginalized in the next round (front marginalization or back marginalization). Mathematically, to marginalize a specific state, we remove all links related to it and then add the removed links into a prior:

$$\mathbf{\Lambda}_p = \mathbf{\Lambda}_p + \sum_{k \in S_i^-} \left(\mathbf{H}_{k+1}^k\right)^{\mathrm{T}} \left(\mathbf{P}_{k+1}^k\right)^{-1} \mathbf{H}_{k+1}^k + \sum_{(i,k) \in S_c^-} \left(\mathbf{H}_i^k\right)^{\mathrm{T}} \left(\mathbf{P}_i^k\right)^{-1} \mathbf{H}_i^k \tag{20}$$

$$\mathbf{b}_p = \mathbf{b}_p + \sum_{k \in S_i^-} \left(\mathbf{H}_{k+1}^k\right)^{\mathrm{T}} \left(\mathbf{P}_{k+1}^k\right)^{-1} r_{S_i}\left(\hat{\mathbf{z}}_{k+1}^k, \mathcal{X}\right) + \sum_{(i,k) \in S_c^-} \left(\mathbf{H}_i^k\right)^{\mathrm{T}} \left(\mathbf{P}_i^k\right)^{-1} r_{S_c}\left(\hat{\mathbf{z}}_i^j, \mathcal{X}\right), \tag{21}$$

where S_i^- and $S_c{}^-$ are the set of removed IMU preintegration measurements and visual measurements, respectively. The prior is then marginalized via the Schur complement (Sibley et al. 2010).

The criteria to select whether to use front or back marginalization are based on the edge alignment performance. If the edge alignment is good and the second newest state is near to the current keyframe, the second newest state will be marginalized in the next round. Otherwise, the oldest state will be marginalized if it fails.

Note that our two-way marginalization is fundamentally different from traditional keyframe-based approaches that simply drop non-keyframes. We preserve all the information (IMU and edge alignment) from non-keyframes by only performing marginalization after the newest state comes, and the system is then updated (step 1 in front marginalization). Also, by marginalization, we ensure that the time period for each IMU preintegration is bounded in order to bound the accumulated error in the IMU measurements. Two-way marginalization preserves the relations between states and serves as the prior links in the graph model (Fig. 4).

6.6 Optimization with robust norm

Based on the residual functions defined in (17) and (19), we operate on the error state and optimize (13) using the Gaussian–Newton method, which iteratively minimizes

$$\min_{\delta\mathcal{X}} \;\; ||\mathbf{b}_p - \mathbf{H}_p\mathcal{X}||^2 + \sum_{k\in S_i} ||r_{S_i}\left(\hat{\mathbf{z}}_{k+1}^k, \mathcal{X}\right) + \mathbf{H}_{k+1}^k \delta\mathcal{X}||^2_{\mathbf{P}_{k+1}^k} + \sum_{(i,j)\in S_c} ||r_{S_c}\left(\hat{\mathbf{z}}_i^j, \mathcal{X}\right) + \mathbf{H}_i^j \delta\mathcal{X}||^2_{\left(\mathbf{W}_i^j\right)^{-1}\mathbf{P}_i^j} \tag{22}$$

and then updates

$$\hat{\mathcal{X}} = \hat{\mathcal{X}} \oplus \delta\mathcal{X} \tag{23}$$

until convergence. $\mathbf{H}_{k+1}^k$ and $\mathbf{H}_i^j$ are the Jacobian matrices of the inertial measurements and visual measurements with respect to the states.

To increase the robustness of our proposed system, $\mathbf{W}_i^j$ changes in each iteration to further eliminate the possible outliers in edge alignment that pass the DE-A self check (Sect. 5.3) and IMU-aided external check (Sect. 6.4). $\mathbf{W}_i^j$ is computed according to the Huber norm thresholding on the current estimate

$$(\mathbf{W}_i^j)_{ul} = \begin{cases} \mathbb{I}_{3\times3}, & \text{if } ||\mathbf{R}_0^j\left(\mathbf{p}_i^0 - \mathbf{p}_j^0\right) - \hat{\mathbf{t}}_i^j|| \le c_t \\ \frac{c_t}{||\mathbf{R}_0^j\left(\mathbf{p}_j^0 - \mathbf{p}_i^0\right) - \hat{\mathbf{t}}_i^j||}\mathbb{I}_{3\times3}, & \text{otherwise} \end{cases} \tag{24}$$

$$(\mathbf{W}_i^j)_{lr} = \begin{cases} \mathbb{I}_{3\times3}, & \text{if } ||2\left[\left(\hat{\mathbf{q}}_i^j\right)^{-1}\left(\mathbf{q}_j^0\right)^{-1}\mathbf{q}_i^0\right]_{xyz}|| \le c_a \\ \frac{c_a}{||2\left[\left(\hat{\mathbf{q}}_i^j\right)^{-1}\left(\mathbf{q}_j^0\right)^{-1}\mathbf{q}_i^0\right]_{xyz}||}\mathbb{I}_{3\times3}, & \text{otherwise,} \end{cases} \tag{25}$$

where $(\mathbf{W}_i^j)_{ul}$ is the upper left 3×3 matrix of $\mathbf{W}_i^j$, $(\mathbf{W}_i^j)_{lr}$ is the lower right 3×3 matrix of $\mathbf{W}_i^j$, $\mathbb{I}_{3\times3}$ is an identity matrix, and c_t and c_a are the given translation and angular threshold, respectively.

7 Experiments

For sensing, we use a VI-sensor[1] which consists of a MEMS IMU and two global shutter cameras with a fronto-parallel stereo configuration. A power efficient small-form factor computer, the Intel NUC[2] with a dual-core CPU i5-4250U running at 1.3 GHz and 16 GB RAM is used for the computing needs. All the algorithms are developed in C++ with ROS as the interfacing robotics middleware. The IMU generates data at 200 Hz and the stereo camera produces time synchronized data at 25 Hz.

7.1 Real-time implementation

To achieve real-time performance we set the finest resolution for edge alignment to be 320 × 240. To estimate the depth map from the stereo camera, we use a block matching algorithm implemented in OpenCV (StereoBM). Since, the proposed edge alignment requires depth values at edge pixels only, a simple stereo block matching suffices for our needs. We adopt image pyramids (with three levels) in the edge alignment to handle the large image displacement caused by fast motion and increase the speed of convergence for the underlying iterative optimization procedure. We set the size of the sliding window to be 30. The threshold of the average reprojection distance of the edge alignment self check is set to 5. For the local loop closure module, we firstly do the cross check of the edge alignment at the coarsest level, and ignore the candidates that fail this test. We then do the cross check of the edge alignment with full image pyramids for the remaining candidates. Meanwhile, we restrict the number of cross checks with full image pyramids so as to limit the maximum time spent on the loop closure module. The computing times of each component are summarized in Table 1.

We do not impose a global prior (like fixing the oldest pose) when solving Eqs. (13) and (22). Instead, we solve Eqs. (13) and (22) without any global prior (the resultant equations may not be well constrained, we thus use perturbed Cholesky decomposition that ensures positive definiteness to solve them). The obtained positions and yaw angles of states in the sliding window after the iteration are subtracted by the position and the yaw angle difference of the oldest pose before and after the iteration. We do NOT enforce a global prior as the pitch and roll angle of the oldest state in the sliding window are observable. The initial position and yaw angle of the oldest pose at time instant b_0 before the iteration are zero. Prior matrix $\mathbf{\Lambda}_p$ and prior vector $\mathbf{b}_p$ obtained in the marginalization step are relative priors between states.

7.2 Tracking in changing lighting conditions

We record an image sequence which spans different rooms. The path spans rooms that are dimly lit, followed by a rather featureless corridor, which leads to a room lit by sunlight (brightly lit), followed again by a corridor that is dimly lit and a bright corridor. In such a situation, where ambient brightness is changing, it is not appropriate to disable the camera auto exposure. The charge couple devices (CCDs) in cameras usually have a low dynamic range. The auto-exposure

[1] http://www.skybotix.com/.

[2] http://www.intel.com/content/www/us/en/nuc/overview.html.

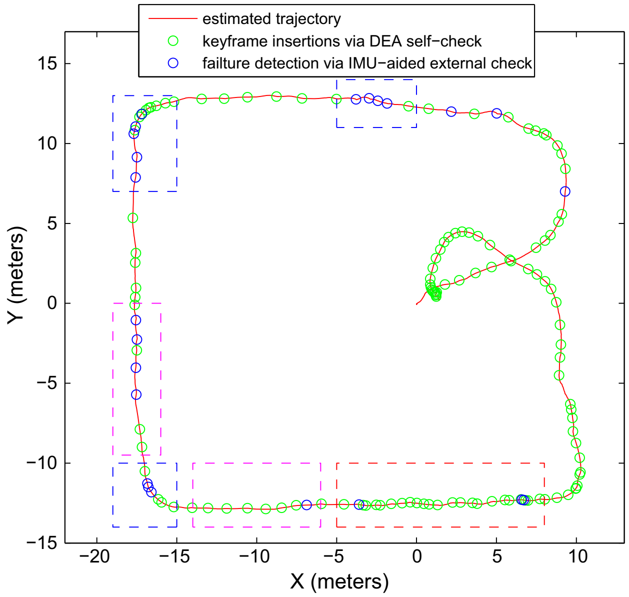


Fig. 7 The estimated trajectory of a walk around a circular path with different lighting conditions (Color figure online)

module of the camera adjusts the exposure time to match the mid-tone of the scene to the mid-tone of the captured image by means of an internal exposure module. This module is present in almost every camera and allows it to produce better image quality. Disabling this camera module to satisfy the photo-consistency assumption is detrimental to the overall image quality. For example, fixing the exposure in a dimly lit room and using this exposure setting in a brighter room causes severe degradation of image quality.

We note that the previous visual odometry algorithms which rely on the photo-consistency assumption fail in this challenging environment. The dense approach proposed by Kerl et al. (2013) fails to produce any meaningful results due to the violation of the photo-consistency assumption. Furthermore, the feature based algorithms also fail in this situation because of the rather featureless corridors.

Figure 7 presents the estimated output of our estimator. The red curve represents the estimated trajectory, green circles represent the locations at which new keyframes are inserted by the edge alignment self check, and blue circles represent the locations at which edge alignment is detected as failed by the IMU-aided external check. Some of the images captured during this experiment are shown in Fig. 8a–d. The total distance travelled is about 120 m and the final drift is about 1.4 m. More details can be found in the accompanying video.

The segments within blue dashed boxes indicate the locations at which the captured surroundings transform from an indoor corridor to an outdoor corridor or from an outdoor corridor to an indoor corridor (Fig. 8a, b). Since lighting conditions change rapidly and greatly, edge detection is not consistent between frames and results in alignment failure. Our proposed system is able to detect this alignment failure by the IMU-aided external check, and this increases the robustness of the system.

The segments within the purple dashed boxes indicate the locations at which the captured surroundings are featureless (Fig. 8c). Since our estimator tracks incremental motions based on edges instead of sparse features, the alignment succeeds in most cases. Though occasional failures exist (see the blue circles), our system overcomes them by the IMU-aided external check.

The segments within the red dashed boxes indicate the locations at which the lighting conditions of the captured surroundings change rapidly and alternately (Fig. 8d). The changing and alternating lighting conditions are caused by the transitions between the glass and borders of the windows. We also notice that there are strong reflections on the windows on both sides. Our edge alignment module inserts keyframes frequently to handle these cases (see the green circles). Again, the IMU-aided external check is required (see the blue circles).

7.3 Throw it!

In this experiment, we test the proposed system with extreme experimental conditions. This experiment is firstly designed for demonstration of the superior tracking performance of

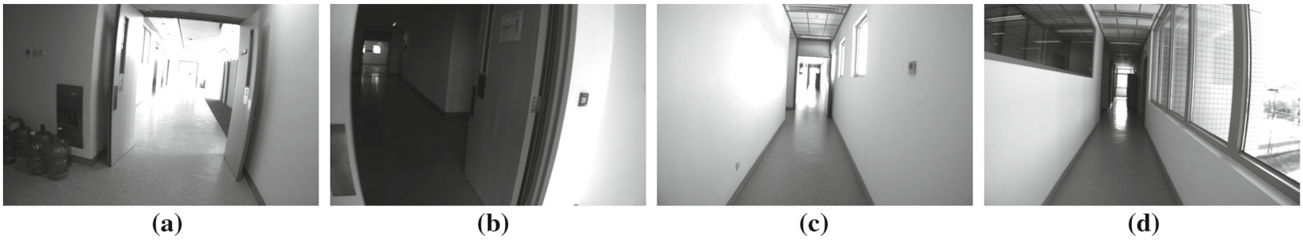

(a)
(b)
(c)
(d)

Fig. 8 Part of captured images during a walk around a circular path with various lighting conditions. **a** Transition from indoor corridor to outdoor corridor, **b** transition from outdoor corridor to indoor corridor, **c** featureless and structured surroundings, **d** fast changing lighting conditions and strong reflection on windows

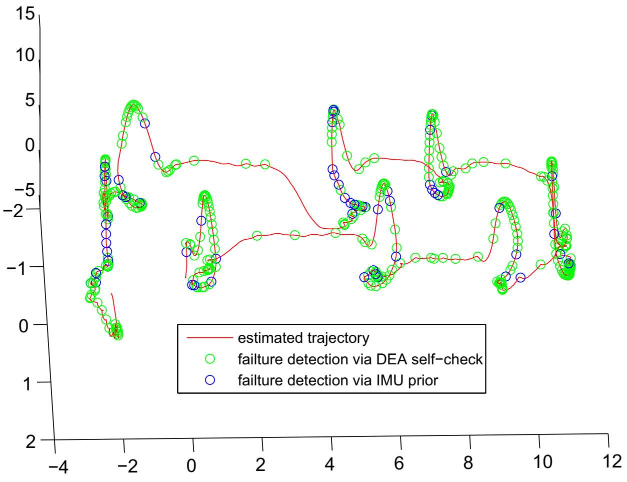


Fig. 9 The estimated trajectory of a walk around a circular path. We throw the VI-sensor while walking (Color figure online)

our previous work (Ling and Shen 2015). We play back the recorded data and redo this experiment using the proposed system in this work. In this challenging experiment, we throw the VI-sensor while walking (Fig. 2). The total walking distance is about 50 m and the final position drift is about 2.24 m. VI-sensor is thrown eight times in total. The estimation results of our proposed method are shown in Fig. 9. From the figure, we see that our estimator can successfully track the motions of these eight throws, resulting in a smooth estimated trajectory. Though edge alignment in our proposed system is able to handle large image displacement caused by challenging motions, it fails when the motions become more and more aggressive (captured images become more and more blurry, see the green and blue circles for indications). Inertial measurements are the last resort that provide crucial links between consecutive states to ensure continuous operation of the estimator. Moreover, failure detection via edge alignment self-check (highlighted with green circles, detailed in Sect. 5.3) and failure detection via IMU prior (highlighted with blue circles, detailed in Sect. 6.4) are of vital importance to the smoothness of the estimated trajectory. In all cases, our local loop closure is able to largely eliminate drifts after throwing (Sect. 6.3).

Notice that, to the best of our knowledge, this experiment is the toughest testing for a visual–inertial estimator that has ever been reported.

7.4 Performance on the EuRoC MAV dataset

We compare our proposed method and other state-of-the-art approaches on the public EuRoC MAV dataset (Burri et al. 2016). The complexity of the sequences in this dataset varies in terms of trajectory length, flight dynamics, and illumination conditions. The reference methods are OKVIS (Leutenegger et al. 2015) and ROVIO (Bloesch et al. 2015). Both OKVIS and ROVIO contain the default parameters for the EuRoC MAV dataset in their open-source implementations. Since we use stereo cameras in our proposed system, for fair comparison, we set the “doStereoInitialization” flag to be true in ROVIO, and also use stereo cameras in OKVIS. To separate the effects of local loop closure and integrating IMUs, our proposed system is tested on four settings. For “Edge-Only” setting, neither local loop closure nor IMU measurements are used; for “Edge+Loop” setting, local loop closure is used; for “Edge+IMU” setting, IMU measurements are used; for “Edge+IMU+Loop” setting, both local loop closure and IMU measurements are used. The accuracy of the estimated position and orientation is measured using the average relative rotation error (ARE-rot) and the average relative translation error (ARE-trans) proposed in Geiger et al. (2012). The summaries are shown in Tables 2 and 3. No data means the concerned method fails to converge at some point in the sequence.

OKVIS, a tightly coupled feature-based approach, is the best in terms of ARE-rot and ARE-trans. Nevertheless, it fails to track in the V2_03_difficult sequence. The other approaches are able to track all the sequences successfully. The ARE-rot and ARE-trans of “Edge+IMU” are smaller than “Edge+IMU+Loop” in most of the sequences. This is because local loop closure usually causes a noticeable pose correction to the latest estimate, which is unfriendly for the relative metrics of ARE-rot and ARE-trans. The same for the comparison of “Edge-Only” and “Edge+Loop”. In terms of the ARE-rot, our proposed method (“Edge+IMU+Loop”) is better than ROVIO. However, for the ARE-trans, ROVIO obtains smaller errors than our approach. The reason is that the estimation of rotation is not related to the scene depth, its error only depends on the number of pixels that well-constraints the rotation. ARE-trans greatly depends on scene depth, thus ROVIO, which is a tight-coupled approach that jointly optimizes the poses and the scene depth, performs better than our method.

7.5 Discussions on convergence basin

One advantage of our proposed method compared to dense tracking based on image intensities is that the convergence basin is larger. Put differently, what it means is, with the proposed formulation, the iterations converge even for a rather poor initial guess. We evaluate this property via skipping frames (downsampling the image temporal frequency). Suppose the origin image temporal frequency is f_n and the number of skipped frames is s_m. The downsampled temporal frequency is

$$f_n' = \frac{f_n}{1 + s_m}. \tag{26}$$

Table 2 Average relative angle error (ARE-rot, deg/m) of different approaches on the EuRoC MAV dataset. The best results are given in bold. No data means the concerned method fails to converge at some point in the sequence

Sequence	OKVIS	ROVIO	Edge+IMU+Loop	Edge+IMU	Edge+Loop	Edge Only
MH_01_easy	**0.006715**	0.014446	0.009401	0.008921	0.015630	0.015630
MH_02_easy	**0.006412**	0.014243	0.009286	0.010063	0.016520	0.010659
MH_03_medium	**0.007525**	0.011873	0.009198	0.009005	0.013628	0.011600
MH_04_difficult	**0.005876**	0.011875	0.012522	0.011266	0.020757	0.019770
MH_05_difficult	**0.004875**	0.009175	0.011338	0.010784	0.018719	0.014592
V1_01_easy	**0.024244**	0.034842	0.035249	0.035605	0.129194	0.106800
V1_02_medium	**0.042781**	0.054126	0.044258	0.046482	0.054274	0.050339
V1_03_difficult	**0.049682**	0.067790	0.057715	0.059629	0.069350	0.060957
V2_01_easy	**0.018585**	0.026823	0.018803	0.018513	0.026975	0.025404
V2_02_medium	**0.040857**	0.057114	0.053272	0.056820	0.069222	0.065446
V2_03_difficult	–	0.075503	**0.064237**	0.066026	0.087980	0.084640

Table 3 Average relative translation error (ARE-trans, m/m) of different approaches on the EuRoC MAV dataset. The best results are given in bold. No data means the concerned method fails to converge at some point in the sequence

Sequence	OKVIS	ROVIO	Edge+IMU+Loop	Edge+IMU	Edge+Loop	Edge Only
MH_01_easy	**0.000346**	0.000915	0.001023	0.000982	0.001041	0.001041
MH_02_easy	**0.000375**	0.001063	0.001106	0.001253	0.001401	0.001083
MH_03_medium	**0.000548**	0.001289	0.001884	0.001675	0.002500	0.002563
MH_04_difficult	**0.000440**	0.003783	0.003586	0.002986	0.004070	0.003987
MH_05_difficult	**0.000426**	0.001271	0.002179	0.002014	0.003459	0.003357
V1_01_easy	**0.000786**	0.001543	0.002883	0.002835	0.003666	0.003571
V1_02_medium	**0.001311**	0.002322	0.003494	0.003982	0.006836	0.006216
V1_03_difficult	**0.001204**	0.002152	0.002475	0.002509	0.004593	0.004055
V2_01_easy	**0.000523**	0.001018	0.002081	0.002083	0.002509	0.002508
V2_02_medium	**0.001018**	0.001794	0.002667	0.002720	0.004530	0.003304
V2_03_difficult	–	**0.002270**	0.002656	0.002496	0.007265	0.006052

We use the most difficult sequence (V2_03_difficult) of the EuRoc MAV dataset to give a detailed assessment of our system. Since OKVIS fails to track this sequence, we exclude it in this comparison. We compare our system with ROVIO. The same as the previous experiment, we set the "doStereoInitialization" flag to be true in ROVIO for fair comparison. Our system runs with four settings ("Edge-Only", "Edge+Loop", "Edge+IMU", "Edge+IMU+Loop"). Details are shown in Tables 4 and 5. No data means the concerned method fails to converge at some point in the sequence. ARE-rot and ARE-trans metrics are used for comparison. ROVIO loses track if the number of skipped frames is equal to or more than 5 while our system loses track if the number of skipped frames is equal to or more than 8. The integration of the IMU has a significant improvement on the tracking accuracy and robustness. Firstly, it provides an initial pose estimate for edge alignment, especially for the rotation estimate, which greatly reduces the risks of trapping in wrong local regions. Secondly, the fusion formulation involving IMU measurements optimizes velocities, which helps to bound the poses according to the differentiation equation. Finally, IMU measurements are noisy but outlier-free. The integration of IMU measurements is a good reference to detect tracking failure of the edge alignment module. The local loop module seems to be useless in terms of ARE-rot and ARE-trans metrics. However, it relocalizes the latest pose when the edge alignment fails and the prediction from IMU measurements is not accurate after long-term integration. It also helps to bound the poses and velocities within the sliding window. As a result, the following poses to be estimated will be bounded according to the differentiation equation (see that the sections of the skipped number are more than 4).

7.6 Tracking in an outdoor environment with more complex textures and less prominent edge data

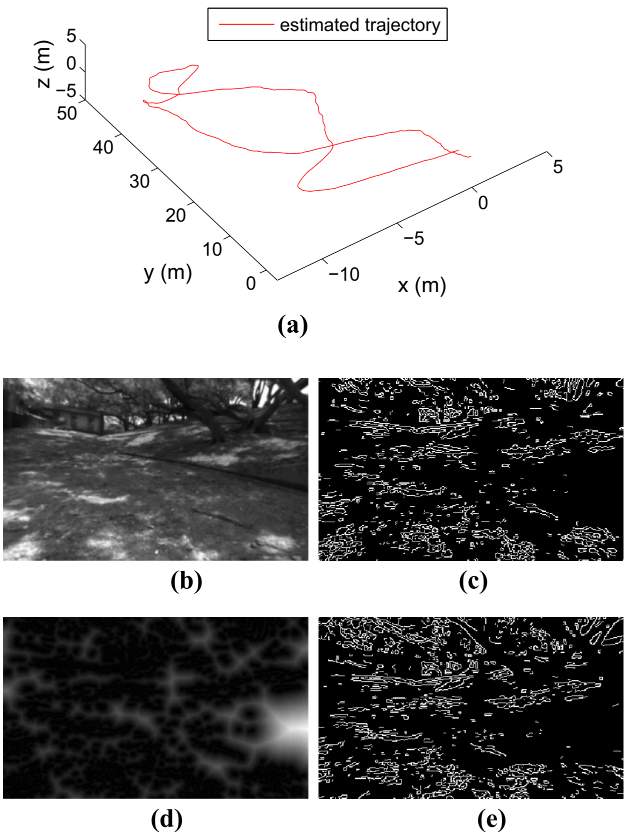
We further test our system performance in an outdoor environment with more complex textures and less prominent edge data. There are trees, grass and shadows in the test sequence. Our system is able to handle this tracking sequence. The estimated trajectory is shown in Fig. 10a. One of the captured images is shown in Fig. 10b. Corresponding edges and distance transform are shown in Fig. 10c, d. Reference keyframe

Table 4 Comparison between different methods with different numbers of skipped frames in the V2_03_difficult sequence of the EuRoc MAV dataset. Error metrics are average relative angle error (ARE-rot, deg/m). The best results are given in bold. No data means the concerned method fails to converge at some point in the sequence

Skipped Number	ROVIO	Edge+IMU+Loop	Edge+IMU	Edge+Loop	Edge-Only
1	0.120224	0.069523	**0.067445**	0.093351	0.088024
2	0.139539	0.076872	**0.074963**	–	–
3	0.182557	0.094589	**0.091109**	–	–
4	0.233408	0.094585	**0.079288**	–	–
5	–	0.115803	**0.115535**	–	–
6	–	0.123581	**0.123147**	–	–
7	–	**0.152680**	–	–	–

Table 5 Comparison between different methods with different numbers of skipped frames in the V2_03_difficult sequence of the EuRoc MAV dataset. Error metrics are average translation error (ARE-trans, m/m). The best results are given in bold. No data means the concerned method fails to converge at some point in the sequence

Skipped Number	ROVIO	Edge+IMU+Loop	Edge+IMU	Edge+Loop	Edge-Only
1	**0.004429**	0.011473	0.011459	0.012937	0.012262
2	**0.007028**	0.017675	0.015349	–	–
3	**0.011486**	0.028945	0.022221	–	–
4	**0.021452**	0.039633	0.031632	–	–
5	–	**0.039131**	0.039562	–	–
6	–	**0.065406**	0.065720	–	–
7	–	**0.108776**	–	–	–



(a)

(b) **(c)**

(d) **(e)**

Fig. 10 Our system is able to track in an outdoor environment with more complex textures and less prominent edge data. **a** The estimated trajectory, **b** one of the captured images, **c** the edges detected in the current frame, **d** the distance transform of the current frame, **e** the reference keyframe edges. More tracking details can be found in the supplementary video: https://1drv.ms/u/s!ApzRxvwAxXqQmgX66v7srdWZNvAs

edges are shown in Fig. 10e. The total travel distance is about 120 m and the final position drift is about 1.5 m. More tracking details can be found in the supplementary video: https://1drv.ms/u/s!ApzRxvwAxXqQmgX66v7srdWZNvAs.

8 Conclusions and future work

We propose a novel and robust real-time system for state estimation of aggressive motions. Our system is designed specifically for aggressive quadrotor flights or other applications in which aggressive motions are encountered (such as augmented reality). We employ a novel edge-tracking formulation for visual relative pose estimation. We also propose a semi-tightly coupled probabilistic framework for fusion of sensor states over a sliding window. The multi-thread framework enables a fast and stable estimate with only the CPU of an off-the-shelf computing platform. Experiments have verified the performance of our system and its potential for use in embedded system applications.

We note that the tightly-coupled methods, that jointly optimize poses and point depth, outperform our semi-tightly coupled approach if their front-end trackers work well. As the future work, we will integrate the front-end edge tracker and back-end tightly-coupled optimization in a whole framework to achieve better performance. The main challenge is to handle the greatly increased system complexity. A probabilistic formulation of the edge alignment will also be considered.

Acknowledgements The authors acknowledge the funding support from HKUST internal Grant R9341 and HKUST institutional scholarship. We would like to thank all AURO reviewers for their exceptionally useful reviews.

References

Baker, S., & Matthews, I. (2004). Lucas–Kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, *56*(3), 221–255.

Bay, H., Tuytelaars, T., Ess, A., & Gool, L. V. (2008). Speeded up robust features. In *Computer vision and image understanding*.

Bloesch, M., Omari, S., Hutter, M.,& Roland, S. (2015). Robust visual inertial odometry using a direct EKF-based approach. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems*.

Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., et al. (2016). The EuRoC micro aerial vehicle datasets. *The International Journal of Robotics Research*, *35*(10), 1157–1163.

Christian, F., Luca, C., Frank, D., & Davide, S. (2015). IMU preintegration on manifold for efficient visual–inertial maximum-a-posteriori estimation. In *Proceedings of the robotics: Science and system*.

Dong-Si, T., & Mourikis, A. I. (2012). Estimator initialization in vision-aided inertial navigation with unknown camera-IMU calibration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Engel, J., Schöps, T.,& Cremers, D. (2014). LSD-SLAM: Large-scale direct monocular SLAM. In *European conference on computer vision.*

Engel, J., Sturm, J., & Cremers, D. (2013). Semi-dense visual odometry for a monocular camera. In: *Proceedings of the IEEE international conference computer vision*, Sydney.

Felzenszwalb, P. F., & Huttenlocher, D. P. (2012). Distance transforms of sampled functions. *Theory of Computing*, *8*(1), 415–428.

Fitzgibbon, A. (2003). Robust registration of 2D and 3D point sets. *Image and Vision Computing*, *21*(14), 1145–1153.

Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Conference on computer vision and pattern recognition*.

Harris, C. G., & Pike, J. M. (1987). 3D positional integration from image sequences. In *Proceedings of the Alvey vision conference*, Cambridge.

Heng, L., Lee, G. H., & Pollefeys, M. (2014). Self-calibration and visual SLAM with a multi-camera system on a micro aerial vehicle. In *Proceedings of Robotics: Science and Systems*. Berkeley, CA.

Hesch, J. A., Kottas, D. G., Bowman, S. L., & Roumeliotis, S. I. (2014). Consistency analysis and improvement of vision-aided inertial navigation. *IEEE Transactions on Robotics*, *30*(1), 158–176.

Huang, A. S., Bachrach, A., Henry, P., Krainin, M., Maturana, D., Fox, D., & Roy, N. (2011). Visual odometry and mapping for autonomous flight using an RGB-D camera. In *Proceedings of the international symposium of robotics research*, Flagstaff, AZ.

Huang, G., Kaess, M., & Leonard, J. J. (2014). Towards consistent visual–inertial navigation. In *Proceedings of the IEEE international conference on robotics and automation*, Hong Kong.

Kerl, C., Sturm, J., & Cremers, D. (2013). Robust odometry estimation for RGB-D cameras. In *Proceedings of the IEEE international conference on robotics and automation*.

Kuse, M., & Shen, S. (2016). Robust camera motion estimation using direct edge alignment and sub-gradient method. In *Proceedings of the IEEE international conference on robotics and automation*.

Leutenegger, S., Furgale, P., Rabaud, V., Chli, M., Konolige, K., & Siegwart, R. (2015). Keyframe-based visual–inertial using nonlinear optimization. *The International Journal of Robotics Research, 34*(3), 314–334.

Li, M., & Mourikis, A. (2013). High-precision, consistent EKF-based visual-inertial odometry. *The International Journal of Robotics*, *32*(6), 690–711.

Ling, Y., Liu, T., & Shen, S. (2016). Aggressive quadrotor flight using dense visual–inertial fusion. In *Proceedings of the IEEE international conference on robotics and automation*.

Ling, Y., & Shen, S. (2015). Dense visual–inertial odometry for tracking of aggressive motions. In *Proceedings of the IEEE international conference on robotics and biomimetics*.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Ma, Y., Soatto, S., Kosecka, J., & Sastry, S. S. (2012). *An invitation to 3-d vision: From images to geometric models* (Vol. 26). Berlin: Springer.

Newcombe, R. A., Lovegrove, S., & Davison, A. J. (2011). DTAM: Dense tracking and mapping in real-time. In *IEEE international conference on computer vision* (pp. 2320–2327).

Omari, S., Bloesch, M., Gohl, P., & Siegwart, R. (2015). Dense visual–inertial navigation system for mobile robots. In *Proceedings of the IEEE international conference on robotics and automation*.

Rosten, E., & Drummond, T. (2006). Machine learning for high-speed corner detection. In *IEEE conference on European conference on computer vision*.

Rusinkiewicz, S., & Levoy, M. (2001). Efficient variants of the ICP algorithm. In *International conference on 3-D imaging and modeling* (pp. 145–152).

Scaramuzza, D., Achtelik, M., Doitsidis, L., Fraundorfer, F., Kosmatopoulos, E., Martinelli, A., et al. (2014). Vision-controlled micro flying robots: From system design to autonomous navigation and mapping in GPS-denied environments. *IEEE Robotics & Automation Magazine, 21*(3), 26–40.

Shi, J., & Tomasi, C. (1994). Good features to track. In *IEEE conference on computer vision and pattern recognition*.

Segal, A., Haehnel, D., & Thrun, S. (2005). Generalized-ICP. In *Robotics: Science and systems*.

Shen, S., Michael, N., & Kumar, V. (2015). Tightly-coupled monocular visual–inertial fusion for autonomous flight of rotorcraft MAVs. In *Proceedings of the IEEE international conference on robotics and automation*, Seattle, WA.

Shen, S., Mulgaonkar, Y., Michael, N., & Kumar, V. (2013). Vision-based state estimation and trajectory control towards high-speed flight with a quadrotor. In *Proceedings of robotics: Science and systems*, Berlin.

Shen, S., Mulgaonkar, Y., Michael, N., & Kumar, V. (2014). Initialization-free monocular visual–inertial estimation with application to autonomous MAVs. In *Proceedings of the international symposium on experimental robotics*, Morocco.

Sibley, G., Matthies, L., & Sukhatme, G. (2010). Sliding window filter with application to planetary landing. *Journal of Field Robotics, 27*(5), 587–608.

Stückler, J., & Behnke, S. (2012). Model learning and real-time tracking using multi-resolution surfel maps. In *Association for the advancement of artificial intelligence*.

Tomasi, C., & Kanade, T. (1991). Detection and tracking of point features. In *Carnegie Mellon University Technical Report CMU-CS-91-132*.

Usenko, V., Engel, J., Stuckler, J., & Cremers, D. (2016). Direct visual–inertial odometry with stereo cameras. In *Proceedings of the IEEE international conference on robotics and automation*.

Yang, Z., & Shen, S. (2015). Monocular visual–inertial fusion with online initialization and camera-IMU calibration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Yang, Z., & Shen, S. (2016). *Tightly-coupled visual–inertial sensor fusion based on IMU pre-integration*, Technical report. Hong Kong University of Science and Technology. http://www.ece.ust.hk/~eeshaojie/vins2016zhenfei.pdf.

Yonggen Ling is a Ph.D. candidate in Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology. His main research interest focus on visualinertial fusion, direct dense tracking, visual tracking, robust estimation, dense mapping, and autonomous navigation.

Manohar Kuse is from Mumbai, India. He is currently a Ph.D. candidate at Robotics Institute (RI) of Hong Kong University of Science and Technology (HKUST) in Hong Kong. His research focus on Visual Navigation of UAVs. Before joining his Ph.D. studies he was a Research Assistant at European Organization for Nuclear Research (CERN) in Geneva, Switzerland.

Shaojie Shen received his B.Eng. degree in Electronic Engineering (Honors Research Option) from the Hong Kong University of Science and Technology in 2009. He received his M.S. in Robotics and Ph.D. in Electrical and Systems Engineering in 2011 and 2014, respectively, all from the University of Pennsylvania. He joined the Department of Electronic and Computer Engineering at the Hong Kong University of Science and Technology in September 2014 as an Assistant Professor. His research interests are in the areas of robotics and unmanned aerial vehicles, with focus on state estimation, sensor fusion, localization and mapping, and autonomous navigation in complex environments.