Autonomous Robots (2019) 43:1895–1915
https://doi.org/10.1007/s10514-019-09834-7

A lightweight and scalable visual-inertial motion capture system using fiducial markers

Guoping He[1] · **Shangkun Zhong**[1] · **Jifeng Guo**[1]

Received: 12 June 2017 / Accepted: 12 January 2019 / Published online: 19 February 2019

Abstract
Accurate localization of a moving object is important in many robotic tasks. Often an elaborate motion capture system is used to realize it. While high precision is guaranteed, such a complicated system is costly and limited to specified small size workspace. This paper describes a lightweight and scalable visual-inertial approach, which leverages paper printable, known size and unknown pose, artificial landmarks, as called fiducials, to obtain motion state estimates, including pose and velocity. Visual-inertial joint optimization using incremental smoother over factor graph and the IMU preintegration technique make our method efficient and accurate. No special hardware is required except a monocular camera and an IMU, making our system lightweight and easy to deploy. Using paper printable landmarks, as well as the efficient incremental inference algorithm, renders it nearly constant-time complexity and scalable to large-scale environment. We perform an extensive evaluation of our method on public datasets and real-world experiments. Results show our method achieves accurate state estimates and is scalable to large-scale environment and robust to fast motion and changing light condition. Besides, our method has the ability to recover from intermediate failure.

Keywords Motion capture system · Visual-inertial · Fiducial based · IMU preintegration · Incremental smoothing

1 Introduction

Many robotic tasks such as path planning and navigation need an accurate localization. Generally these tasks need the moving object's pose information relative to a global coordinate and sometimes the velocity estimate is also needed. One common way to obtain the pose estimate of the moving object is using a set of motion capture system such as Vicon and OptiTrack. These systems are usually highly accurate while they are costly and limited to a fixed small size workspace, making it uneasy to access and low scalability to large-scale environment. Furthermore, the complicated pre-deploy operations of these system make it uneasy to use.

An alternative to motion capture system is Visual simultaneous localization and mapping (VSLAM) or a lighter one, Visual odometry (VO). Many excellent VSLAM and VO systems have been proposed in recent years, such as PTAM, SVO, ORB-SLAM, LSD-SLAM and DSO (Klein and Murray 2007; Forster et al. 2014; Mur-Artal et al. 2015; Engel et al. 2014, 2017). While these systems can obtain a good performance, they can only provide pose estimation relative to the initial coordinate. Furthermore, for the monocular visual SLAM, the inherent scale-ambiguity makes it fail to stabilize the scale and keep drifting over time and render it complicated to compute the normalized depth. Scaled sensors, such as stereo or depth camera, however, can only provide reliable measurements in limited range. Even some strategies introduced, such as loop detection and local pose optimization on similarity transformation rather than rigid body transformation, it can only correct the scale drift instead of estimating the absolute scale factor.

One method of rectifying scale drift completely is to introduce absolute scale information. It's naturally to fuse visual information with IMU measurements as do in Leutenegger et al. (2014), Concha et al. (2016) and Usenko et al. (2016). It works well while it seems a bit complicate

✉ Guoping He
heguoping@hit.edu.cn

Shangkun Zhong
shangkun.zhong@foxmail.com

Jifeng Guo
guojifeng@hit.edu.cn

[1] Department of Aerospace Engineering, Harbin Institute of Technology, Harbin HL451, China

to obtain the visual measurements and the visual cues are not robust enough for trakcing, no matter with feature-based method or direct method. We aim at facilitating this process and improving the tracking robustness by using fiducials. On the contrary, methods in Sementille and Rodello (2004), Fiala (2005), Lim and Lee (2009) and Olson (2011) exploit fiducials without combining with IMU measurements and is consequently not robust to fast motion and the absence of fiducials.

In this paper, we proposed a lightweight and scalable visual-inertial motion capture system. Our system incorporates two threads that run in parallel, the tag detector and the smoother. We exploit unknown pose printable artificial landmarks with known size information, which further mitigate the complexity to obtain visual measurements, and tightly fuse inertial data with visual cues, especially the known size information, in a compact form. We use AprilTags (Olson 2011) as our fiducials. These tags can be robustly tracked and can provide unique identification number which can handle loop closure implicitly avoiding the need of additional loop detection step. By using incremental smoothing over the factor graph (Kaess et al. 2011) and IMU pre-integration technique (Forster et al. 2017), our method can provide efficient and accurate motion estimation. To this end, we summarize our main contributions as follow:

- We present a lightweight and scalable fiducials-based monocular visual-inertial motion capture system. Visual-inertial are optimized jointly in a tight way using incremental smoother over factor graph. A tag factor based on AprilTag is constructed.
- Extensive evaluation results both on public datasets and real-world experiments are provided to demonstrate the performance of our system. Results show that our system is robust to motion blur, changing light conditions and outperforms the filtering-based method and more scalable to large-scale environment (indoor/outdoor).
- We propose a intermediate failure recovery mechanism using IMU measurements and fiducials. Experimental results show that our system has the ability to recover from intermediate failure.

The rest of the paper is organized as follows. In Sect. 2, we discuss related work. Section 3.2 introduces the tag detector. Section 3.3 formulates the joint visual-inertial optimization problem, where the nonlinear error terms from IMU and fiducial measurements are described in-depth with the Jacobians of the tag reprojection error detailed in the "Appendix" section. Section 3.4 describes the adaptive measurement uncertainty calculation method. Section 3.5 introduces the incremental smoothing. Section 4 describes the intermediate failure recovery mechanism. We evaluate our proposed system in Sect. 5 by deploying it both on public datasets and real-world experiments. Finally, a discussion about runtime and a conclusion are made in Sects. 6 and 7, respectively (Fig. 1).

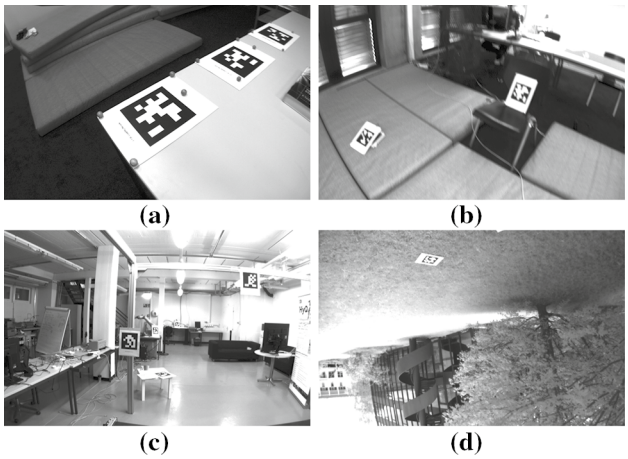


Fig. 1 Images extracted from the 4 real-world datasets we used, *table* (**a**), *dataset_1* (**b**), *cube* (**c**) and *pavillon* (**d**). The AprilTags are distributed in the scene randomly with unknown pose

2 Related work

Fiducial-based localization has been well studied in past years while most of these algorithms, like (Sementille and Rodello 2004; Fiala 2005; Lim and Lee 2009; Olson 2011), only use camera without fusing with IMU measurements thus usually only position or (and) orientation estimates can be obtained, not full motion state. Besides, most of them aim at virtual reality or augmented reality not for moving object's motion capture. In Faessler et al. (2014), infrared LEDs are used as fiducials. These fiducials are mounted on the target object with known relative positions and captured by external camera. Since no motion model is used, it can only provide 6DOF relative pose estimation, no velocity estimation. Moreover, its workspace is limited to the field of view of the off-board camera. In our work, on the contrary, the camera is mounted on the moving object and the fiducials are fixed in the environment with unknown pose thus it has no workspace limitations and can do all computations online. In addition, our paper printable fiducial is much more accessible and unexpensive than the infrared LEDs. Also in Qiu et al. (2015), modulated LED lights, distributed arbitrarily in the environment, are used to aid indoor localization. Each LED has a unique modulation waveform, while it's not easy to use compared to paper printable AprilTags we used since the waveform should be selected carefully to ensure robust data-association.

One advantage of fiducials is that it can provide physical size information with which the scale becomes observable,

solving the scale drift problem completely. There are also some works that learn objects size online (Botterill et al. 2013; Gálvez-López et al. 2016) or exploit prior size information (Frost et al. 2016) to correct scale drift, instead of using object's size information directly. While these methods are unconstrained by artificial landmarks, they are limited to specific object and exist much uncertainty in size information. Instead in our approach, the size of fiducials are definite. Besides, we combine fiducial measurements with IMU data, making our approach more robust to fast motion and the absence of fiducials.

Providing unique identification number is the other advantage of fiducials. Many SLAM system use loop detection mechanism to rectify scale drift, such as Mur-Artal et al. (2015) and Hauke et al. (2010). It is indeed an effective way but also exists many limitations. One of the limitations is that a good loop-closure is dependent on a reliable place recognition method and often the bag of words is used, which brings much more complexity and is not robust enough to fast motion and changing light condition. Moreover, it doesn't work if the trajectory contains no closed loops and the worst is that sometimes the scale drift is so severe that even detecting loop is infeasible. However, with fiducials' unique identification number, we can handle the loop closure implicitly and reduce complexity extensively.

On the other hand, combining IMU measurements with fiducial visual cues further improve the robustness to fast motion. Compared to fiducial-free visual-inertial systems, like (Mourikis and Roumeliotis 2007; Concha et al. 2016; Usenko et al. 2016), fiducial-based visual-inertial motion capture system facilitate the process of extracting visual cues, as all we need to do is detecting fiducials, no descriptor , no matching. Besides, the fiducial visual cues are more robust for data association. In this regard, the most related to our work is Neunert et al. (2016). It fuses fiducial cues with IMU measurements by EKF. It has a good performance in small workspace with a few fiducials while it get worse if there are dozens of fiducials, as the states expand quickly over time and the program becomes very slow and is prone to divergence. In our system, we use incremental smoothing algorithm. With it, the problem remains nearly constant-time complexity and we can obtain full accuracy. To further aid high-rate performance, we utilize IMU preintegration technique, which can reduce the number of states that must be added to the smoother significantly.

3 Visual-inertial joint optimization

3.1 Notations

Throughout the paper, we will write scalars as small letters (e.g. l), vectors as bold small letters (e.g. $\boldsymbol{p}$) and matrices as bold capital letters (e.g. $\boldsymbol{R}$). A capital pre-subscript of a vector denotes the coordinate frame that the quantity is expressed in. The trailing subscript describes the direction of the vector from its origin to its destination. The tracked moving object and the fiducials are represented relative to an world frame W where we think the z axis coincides with the negative gravity vector. We distinguish camera frame as C, fiducial frame as A and IMU sensor frame as B. We also think that IMU sensor frame coincides with the moving object body frame. For instance, ${}_W\boldsymbol{p}_{WB}$ represent a vector, expressed in world frame, pointing from the origin of world frame to the origin of IMU sensor frame. The rotation matrix is denoted as $\boldsymbol{R} \in SO_3$. $\boldsymbol{R}_{WB}$ represents the rotation from IMU sensor frame to world frame. The transformation matrix, denoted as $\boldsymbol{T} \in \mathfrak{se}(3)$, includes both rotation and translation, such as:

$$\boldsymbol{T}_{WB} = \begin{bmatrix} \boldsymbol{R}_{WB} & {}_W\boldsymbol{p}_{WB} \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix} \tag{1}$$

We denote the homogeneous coordinate of a 3-dimension point as $\boldsymbol{l} = \begin{bmatrix} \boldsymbol{l}_{1:3}^T & 1 \end{bmatrix}^T$.

The tangent space of the $SO(3)$ is denoted as $\mathfrak{so}(3)$, which is also called *Lie algebra* and coincides with the 3×3 skew symmetric matrices. We can get the corresponding skew symmetric matrix for every vector in $\mathbb{R}^3$ using the *hat* operator as:

$$\boldsymbol{\omega}^\wedge = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}^\wedge = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega & 0 \end{bmatrix} \in \mathfrak{so}(3) \tag{2}$$

The opposite operator of $(.)^\wedge$ is denoted as $(.)^\vee$, which map a skew symmetric matrix to a vector. Thus if we have $\boldsymbol{\omega}^\wedge = \boldsymbol{S}$, then we get $\boldsymbol{\omega} = \boldsymbol{S}^\vee$. As we operate our states on vectors directly, we adopt "vectorized" versions of the exponential and logarithm map between $\mathfrak{so}(3)$ and $SO(3)$:

$$\begin{aligned} &\mathrm{Exp} : \mathbb{R}^3 \to SO(3); \quad \boldsymbol{\phi} \to \boldsymbol{R} = \exp(\boldsymbol{\phi}^\wedge) \\ &\mathrm{Log} : SO(3) \to \mathbb{R}^3; \quad \boldsymbol{R} \to \boldsymbol{\phi} = \log(\boldsymbol{R})^\vee \end{aligned} \tag{3}$$

3.2 AprilTag detector

We choose AprilTags (Olson 2011), which are 2-dimensional and printable bar codes, as our fiducials. In the past years, kinds of fiducial systems have been proposed (Fiala 2010). The reason for choosing ApriltTag is that it's robust for tracking and can provide a initial relative pose for the smoother. Furthermore, it has open efficient C++

implementations. For fair comparison with Neunert et al. (2016), we also use the implementation in cv2cg.[1] Each tag has 4 corners and a unique identification number (tag ID, e.g. tag57 means the tag's ID is 57). For each tag, we define a coordinate frame A_j which coincides with the geometrical center of the tag and where z_i axis is perpendicular to the tag plane. Here, the subscript j represents the tag id. Let a be the side length of the square tag, then we can obtain the homogeneous physical coordinates $\boldsymbol{l}_j^n (n = 1, 2, 3, 4)$ of its 4 corners in the tag j frame, that are $(-\frac{a}{2}, -\frac{a}{2}, 0, 1)$, $(\frac{a}{2}, -\frac{a}{2}, 0, 1)$, $(\frac{a}{2}, \frac{a}{2}, 0, 1)$ and $(-\frac{a}{2}, \frac{a}{2}, 0, 1)$ in counterclockwise order starting from the lower left corner. We implement the tag detection in a separate detector thread. This detector thread will provide each detected tag's 4 corner pixel coordinates as well as the initial relative pose to the smoother thread.

3.3 Batch visual cues with inertial terms

The states of the system at image time t_i comprise the moving object states $\boldsymbol{x}_R^i$, described by the IMU orientation $\mathrm{Log}(\boldsymbol{R}_{WB}^i)$, position ${}_W\boldsymbol{p}_{WB}^i$, velocity ${}_W\boldsymbol{v}^i$, gyro biase $\boldsymbol{b}_g^i$ and accelerometer bias $\boldsymbol{b}_a^i$, and the visible tag poses $\{\boldsymbol{x}_F^j\}_{j\in\mathcal{J}_{K_i}}$:

$$\boldsymbol{x}_i = \left[\boldsymbol{x}_R^{i\,T} \ \{\boldsymbol{x}_F^{j\,T}\}_{j\in\mathcal{J}_{K_i}}\right]^T \tag{4}$$

where the indices of tags visible at frame K_i are written as the set $\mathcal{J}_{K_i}$. The moving object state is:

$$\boldsymbol{x}_R^i = \left[\left[\mathrm{Log}(\boldsymbol{R}_{WB}^i)\right]^T \ {}_W\boldsymbol{p}_{WB}^{i\,T} \ {}_W\boldsymbol{v}^{i\,T} \ \boldsymbol{b}_g^{i\,T} \ \boldsymbol{b}_a^{i\,T}\right]^T \tag{5}$$

And the pose of the jth fiducials seen at image time t_i is:

$$\boldsymbol{x}_F^j = \left[[\mathrm{Log}(\boldsymbol{R}_{WA_j})]^T \ {}_W\boldsymbol{p}_{WA_j}^T\right]^T \tag{6}$$

Let $\mathcal{K}_s$ denote the set of all frames up to image time t_s. As incremental full smoothing is used, we estimate the states at all image time:

$$\boldsymbol{\chi}_s = \left[\{\boldsymbol{x}_R^{i\,T}\} \ \{\boldsymbol{x}_F^{j\,T}\}\right]_{i\in\mathcal{K}_s\, j\in\mathcal{J}_{\mathcal{K}_s}} \tag{7}$$

where $\mathcal{J}_{\mathcal{K}_s}$ are the set of all tags in which every fiducial is visible at least in one frame in $\mathcal{K}_s$.

We formulate the visual-inertial system in one joint optimization problem thus the cost function $J(\boldsymbol{\chi}_s)$ contains both (weighted) visual reprojection errors $J_{re}(\boldsymbol{\chi}_s)$ and (weighted) IMU residual terms $J_{imu}(\boldsymbol{\chi}_s)$:

$$J(\boldsymbol{\chi}_s) = J_{imu}(\boldsymbol{\chi}_s) + J_{re}(\boldsymbol{\chi}_s) \tag{8}$$

The IMU residual terms are

$$J_{imu}(\boldsymbol{\chi}_s) = \sum_{i\in\mathcal{K}_s} \left(\boldsymbol{e}_{imu}^i\right)^T \boldsymbol{W}_{imu}^i \boldsymbol{e}_{imu}^i \tag{9}$$

[1] https://code.google.com/archive/p/cv2cg/.

The visual reprojection errors are

$$J_{re}(\boldsymbol{\chi}_s) = \sum_{i\in\mathcal{K}_s} \sum_{j\in\mathcal{J}_{\mathcal{K}_s}} \left(\boldsymbol{e}_{re}^{i,j}\right)^T \boldsymbol{W}_{re}^{i,j} \boldsymbol{e}_{re}^{i,j} \tag{10}$$

where $\boldsymbol{W}_{re}^{i,j}$ and $\boldsymbol{W}_{imu}^i$ represent the information matrix (inverse covariance matrix) of the respective tag measurement and the ith IMU residual.

3.3.1 IMU model and preintegration

An IMU typically measures 3-axis linear acceleration ${}_B\boldsymbol{a}_m$ and 3-axis angular velocity ${}_B\boldsymbol{\omega}_m$ of its body frame B with respect to inertial (world) frame W, where the subscript m denotes the measured value. We assume that the IMU measurements contain slowly random walk gyro bias $\boldsymbol{b}_g$ and accelerometer bias $\boldsymbol{b}_a$ and additive, zero-mean white Gaussian noise for a gyro and accelerometer sensor, respectively, ($\boldsymbol{n}_a$ and $\boldsymbol{n}_g$). Thus for the real angular velocity ${}_B\boldsymbol{\omega}$ in the IMU frame and the real acceleration ${}_W\boldsymbol{a}$ in world frame, we have:

$$\begin{aligned}
{}_B\boldsymbol{\omega} &= {}_B\boldsymbol{\omega}_m - \boldsymbol{n}_g - \boldsymbol{b}_g \\
{}_W\boldsymbol{a} &= \boldsymbol{R}_{WB} \cdot \left({}_B\boldsymbol{a}_m - \boldsymbol{n}_a - \boldsymbol{b}_a\right) + {}_W\boldsymbol{g} \\
\dot{\boldsymbol{b}}_g &= \boldsymbol{n}_{bg} \\
\dot{\boldsymbol{b}}_a &= \boldsymbol{n}_{ba}
\end{aligned} \tag{11}$$

The IMU states typically are composed of the orientation of the IMU body frame to world frame $\boldsymbol{R}_{WB} \in SO(3)$, body velocity ${}_W\boldsymbol{v}$ and position ${}_W\boldsymbol{p}_{WB}$. The evolution of the states can be inferred from the continuous IMU kinetics:

$$\begin{aligned}
\dot{\boldsymbol{R}}_{WB} &= \boldsymbol{R}_{WB} \cdot \left({}_B\boldsymbol{\omega}_m - \boldsymbol{n}_g - \boldsymbol{b}_g\right)^\wedge \\
{}_W\dot{\boldsymbol{v}} &= \boldsymbol{R}_{WB} \cdot \left({}_B\boldsymbol{a}_m - \boldsymbol{n}_a - \boldsymbol{b}_a\right) + {}_W\boldsymbol{g} \\
{}_W\dot{\boldsymbol{p}}_{WB} &= {}_W\boldsymbol{v}
\end{aligned} \tag{12}$$

where ${}_W\boldsymbol{g}$ is the gravity vector expressed in world frame.

The IMU measurements are sampled at regular interval $\Delta t_k = t_{k+1} - t_k$. In time interval $[t_k, t_{k+1}]$, the acceleration and rotational rate are assumed to remain constant. Hence the discrete evolution of the IMU states can be computed via first order Euler integration:

$$\begin{aligned}
\boldsymbol{R}_{WB}^{k+1} &= \boldsymbol{R}_{WB}^k \mathrm{Exp}\left(\left({}_B\boldsymbol{\omega}_m^k - \boldsymbol{b}_g^k\right)\Delta t_k\right) \\
{}_W\boldsymbol{v}^{k+1} &= {}_W\boldsymbol{v}^k + {}_W\boldsymbol{g}\Delta t_k + \boldsymbol{R}_{WB}^k \cdot \left({}_B\boldsymbol{a}_m^k - \boldsymbol{b}_a^k\right)\Delta t_k \\
{}_W\boldsymbol{p}_{WB}^{k+1} &= {}_W\boldsymbol{p}_{WB}^k + {}_W\boldsymbol{v}^k\Delta t_k \\
&\quad + \frac{1}{2}\left({}_W\boldsymbol{g} + \boldsymbol{R}_{WB}^k \cdot \left({}_B\boldsymbol{a}_m^k - \boldsymbol{b}_a^k\right)\right)\Delta t_k^2
\end{aligned} \tag{13}$$

where the superscript k represents the navigation state at time-step t_k.

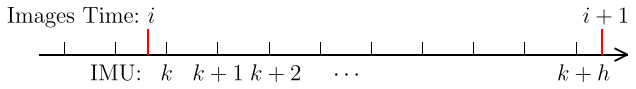


Fig. 2 Different rates for IMU and camera

To implement the optimization approach, the residue of states with respect to the IMU measurements should be explored. As the IMU rate is high, it's not efficient to add new states in the estimation at every new IMU measurement. We use the recently proposed IMU preintegration technique Forster et al. (2017) to further aid high-rate performance. As shown in Fig. 2, there are $h-1$ IMU measurements between two consecutive image time t_i and t_{i+1}. All these IMU measurements can be integrated as a compound measurement, namely preintegrated measurment $\Delta\boldsymbol{R}_{i\,i+1}$, $\Delta\boldsymbol{p}_{i\,i+1}$, $\Delta\boldsymbol{v}_{i\,i+1}$, which constrains the motion between two consecutive frames:

$$\begin{aligned}
\Delta\boldsymbol{R}_{i\,i+1} &= \prod_{z=k-1}^{k+h} \mathrm{Exp}\left(\left(\boldsymbol{\omega}_m^z - \boldsymbol{b}_g^i\right)\Delta t_z\right) \\
\Delta\boldsymbol{v}_{i\,i+1} &= \sum_{z=k-1}^{k+h} \Delta\boldsymbol{R}_{iz}\cdot\left(\boldsymbol{a}_m^z - \boldsymbol{b}_a^i\right)\Delta t_z \\
\Delta\boldsymbol{p}_{i\,i+1} &= \sum_{z=k-1}^{k+h} \left[\Delta\boldsymbol{v}_{iz}\Delta t_z + \frac{1}{2}\Delta\boldsymbol{R}_{iz}\cdot\left(\boldsymbol{a}_m^z - \boldsymbol{b}_a^i\right)\Delta t_z^2\right]
\end{aligned} \tag{14}$$

where we drop the coordinate frame subscripts for readability (the notations keep consistent throughout the paper). For the Δt_z, we have

$$\Delta t_z = \begin{cases} t_k - t_i & z = k-1 \\ t_{z+1} - t_z & \text{others} \\ t_{i+1} - t_{k+h} & z = k+h \end{cases} \tag{15}$$

It is assumed that the IMU random-walk biases remain constant during the two image time $[t_i, t_{i+1}]$ during IMU preintegration:

$$\begin{aligned}
\boldsymbol{b}_g^i &= \boldsymbol{b}_g^k = \cdots = \boldsymbol{b}_g^{k+h} = \boldsymbol{b}_g^{i+1} \\
\boldsymbol{b}_a^i &= \boldsymbol{b}_a^k = \cdots = \boldsymbol{b}_a^{k+h} = \boldsymbol{b}_a^{i+1}
\end{aligned} \tag{16}$$

while during optimization, the bias $\bar{\boldsymbol{b}} = \left[\bar{\boldsymbol{b}}_g;\bar{\boldsymbol{b}}_a\right]$ more likely changes by a small amount $\delta\boldsymbol{b} = \left[\delta\boldsymbol{b}_g;\delta\boldsymbol{b}_a\right]$:

$$\boldsymbol{b} \leftarrow \bar{\boldsymbol{b}} + \delta\boldsymbol{b} \tag{17}$$

Using first-order expansion on the bias changes and combining Eq. (14), we can get the predicted state as:

$$\begin{aligned}
\boldsymbol{R}_{i+1} &= \boldsymbol{R}_i\Delta\bar{\boldsymbol{R}}_{i\,i+1}\mathrm{Exp}\left(\boldsymbol{J}^g_{\Delta\bar{\boldsymbol{R}}_{i\,i+1}}\delta\boldsymbol{b}_g^i\right) \\
\boldsymbol{v}_{i+1} &= \boldsymbol{v}_i + \boldsymbol{g}\Delta t_{i\,i+1} \\
&\quad + \boldsymbol{R}_i\left(\Delta\bar{\boldsymbol{v}}_{i\,i+1} + \boldsymbol{J}^g_{\Delta\bar{\boldsymbol{v}}_{i\,i+1}}\delta\boldsymbol{b}_g^i + \boldsymbol{J}^a_{\Delta\bar{\boldsymbol{v}}_{i\,i+1}}\delta\boldsymbol{b}_a^i\right) \\
\boldsymbol{p}_{i+1} &= \boldsymbol{p}_i + \boldsymbol{v}_i\Delta t_{i\,i+1} + \frac{1}{2}\boldsymbol{g}\Delta\tilde{t}_{i\,i+1}^2 \\
&\quad + \boldsymbol{R}_i\left(\Delta\bar{\boldsymbol{p}}_{i\,i+1} + \boldsymbol{J}^g_{\Delta\bar{\boldsymbol{p}}_{i\,i+1}}\delta\boldsymbol{b}_g^i\boldsymbol{J}^a_{\Delta\bar{\boldsymbol{p}}_{i\,i+1}}\delta\boldsymbol{b}_a^i\right)
\end{aligned} \tag{18}$$

where $\Delta t_{i\,i+1} = t_{i+1} - t_i$. Considering the quality of the IMU, sometimes the sample time interval Δt_k is not a constant. Besides, the IMU may not be synchronized with the camera. To this end, we have

$$\Delta\tilde{t}_{i\,i+1}^2 = \begin{cases} \Delta t_{i\,i+1}^2 & \Delta t_k \text{ is constant and synchronized} \\ \sum_{z=k-1}^{k+h}\left[\Delta t_z^2 + \Delta t_z\sum_{d=k-1}^{z}\Delta t_d\right] & \text{others} \end{cases} \tag{19}$$

Hence, we get the IMU error term $\boldsymbol{e}_{imu}^i$ between image time t_i and t_{i+1}:

$$\begin{aligned}
\boldsymbol{e}_{imu}^i &= \left[\boldsymbol{e}_{\Delta R}^{i\,T}, \boldsymbol{e}_{\Delta v}^{i\,T}, \boldsymbol{e}_{\Delta p}^{i\,T}\right]^T \\
\boldsymbol{e}_{\Delta R}^i &= \mathrm{Log}\left[\left(\Delta\bar{\boldsymbol{R}}_{i\,i+1}\mathrm{Exp}\left(\boldsymbol{J}^g_{\Delta\bar{\boldsymbol{R}}_{i\,i+1}}\delta\boldsymbol{b}_g^i\right)\right)^T\boldsymbol{R}_i^T\boldsymbol{R}_{i+1}\right] \\
\boldsymbol{e}_{\Delta v}^i &= \boldsymbol{R}_i^T\cdot\left(\boldsymbol{v}_{i+1} - \boldsymbol{v}_i - \boldsymbol{g}\Delta t_{i\,i+1}\right) \\
&\quad - \Delta\bar{\boldsymbol{v}}_{i\,i+1} - \boldsymbol{J}^g_{\Delta\bar{\boldsymbol{v}}_{i\,i+1}}\delta\boldsymbol{b}_g^i - \boldsymbol{J}^a_{\Delta\bar{\boldsymbol{v}}_{i\,i+1}}\delta\boldsymbol{b}_a^i \\
\boldsymbol{e}_{\Delta p}^i &= \boldsymbol{R}_i^T\cdot\left(\boldsymbol{p}_{i+1} - \boldsymbol{p}_i - \boldsymbol{v}_i\Delta t_{i\,i+1} - \frac{1}{2}\boldsymbol{g}\Delta\tilde{t}_{i\,i+1}^2\right) \\
&\quad - \Delta\bar{\boldsymbol{p}}_{i\,i+1} - \boldsymbol{J}^g_{\Delta\bar{\boldsymbol{p}}_{i\,i+1}}\delta\boldsymbol{b}_g^i - \boldsymbol{J}^a_{\Delta\bar{\boldsymbol{p}}_{i\,i+1}}\delta\boldsymbol{b}_a^i
\end{aligned} \tag{20}$$

where $\Delta\bar{\boldsymbol{R}}_{i\,i+1} = \Delta\boldsymbol{R}_{i\,i+1}\left(\bar{\boldsymbol{b}}_g^i\right)$, $\Delta\bar{\boldsymbol{v}}_{i\,i+1} = \Delta\boldsymbol{v}_{i\,i+1}\left(\bar{\boldsymbol{b}}_g^i, \bar{\boldsymbol{b}}_a^i\right)$ and $\Delta\bar{\boldsymbol{p}}_{i\,i+1} = \Delta\boldsymbol{p}_{i\,i+1}\left(\bar{\boldsymbol{b}}_g^i, \bar{\boldsymbol{b}}_a^i\right)$. The Jacobians $\boldsymbol{J}^g_{(\cdot)}$ and $\boldsymbol{J}^a_{(\cdot)}$ account for first-order approximation of the changing biases without recomputing during optimization. Both IMU preintegration and Jocabians can be computed iterately with high efficiency as IMU measurements arrive Forster et al. (2017).

3.3.2 The tag-camera model

The camera model is assumed to be a pinhole camera model $\boldsymbol{\pi} : \mathbb{R}^3 \rightarrow \mathbb{R}^2$, which maps a 3-dimension point $\boldsymbol{p}_C$ expressed in camera reference C to pixel coordinates in an image:

$$\boldsymbol{\pi}\left(\boldsymbol{p}_c\right) = \begin{bmatrix} f_x\frac{x_C}{z_C} + c_x \\ f_y\frac{x_C}{z_C} + c_y \end{bmatrix} \quad \boldsymbol{p}_C = \left[x_C, y_C, z_C\right] \tag{21}$$

where f_x and f_y are camera's focal length and c_x and c_y are camera's principle points which should be given as a prior.

We undistort the extracted tag corner pixel coordinates before added to the smoother hence the distortion model is unnecessary in our method. IMU and camera are considered rigidly attached and the transformation $\boldsymbol{T}_{CB} = \left[\boldsymbol{R}_{CB}\ {}_C\boldsymbol{t}_{CB}\right]$ between IMU and camera can be calibrated by the open source tool (Furgale et al. 2014).

Once the estimator receives an image, current camera pose can be predicted from IMU measurments and tag corners in corresponding tag frame can be projected into the image plane. So the reprojection error of each tag corner is:

$$\begin{aligned} \boldsymbol{e}_{re}^{i,j,n} &= \boldsymbol{z}^{i,j,n} - \boldsymbol{\pi}\left(\boldsymbol{T}_{CA_j}^{i} \cdot \boldsymbol{l}_j^n\right) \quad (n = 1,2,3,4) \\ \boldsymbol{T}_{CA_j}^{i} &= \boldsymbol{T}_{CB}\left(\boldsymbol{T}_{WB}^{i}\right)^{-1}\boldsymbol{T}_{WA_j} \end{aligned} \tag{22}$$

as defined above, the superscript i represents that the image is received at time-step t_i, and j is the tag id and n is nth tag point in the tag frame A_j. The tag pose $\boldsymbol{T}_{WA_j}$ can be initialized with the prediction of IMU pose and the initial transformation matrix $\boldsymbol{T}_{CA_j}$ obtained from the tag detector thread. The Jacobians of the tag reprojection error with respect to the moving object pose $\boldsymbol{T}_{WB}^{i}$ and the tag pose $\boldsymbol{T}_{WA_j}$, needed in implementation, are given in the "Appendix" section.

Each observed tag would provide 4 corner measurements which constraint the camera pose and tag pose relative to the world reference. We stack all reprojection errors of these 4 corners to a column vector hence we obtain the final reprojection error for each tag:

$$\boldsymbol{e}_{re}^{i,j} = \left[\boldsymbol{e}_{re}^{i,j,1\ T}\ \boldsymbol{e}_{re}^{i,j,2\ T}\ \boldsymbol{e}_{re}^{i,j,3\ T}\ \boldsymbol{e}_{re}^{i,j,4\ T}\right]^T \tag{23}$$

3.4 Measurement uncertainty

Intuitively, greater velocity renders greater measurement uncertainty. To this end, we compute measurement uncertainty σ_m adaptively as

$$\sigma_m = \begin{cases} \alpha v_i & \sigma_m > 2 \text{ pixels} \\ 2 & \sigma_m \leq 2 \text{ pixels} \end{cases} \tag{24}$$

where v_i is the initial estimated velocity magnitude (generally predicted by IMU preintegration) at image time t_i and α is the scale factor. In our implementation, $\alpha = 2$.

Beside the measurement uncertainty, for each tag, we need to set the initial translational and angular uncertainty. Referring to Fig. 3, the initial position uncertainty for each axis of the tag is approximately computed as

$$\sigma_t = \frac{\beta_1 \sigma_m}{l} * s \tag{25}$$

The initial angular uncertainty of the tag is approximately computed as

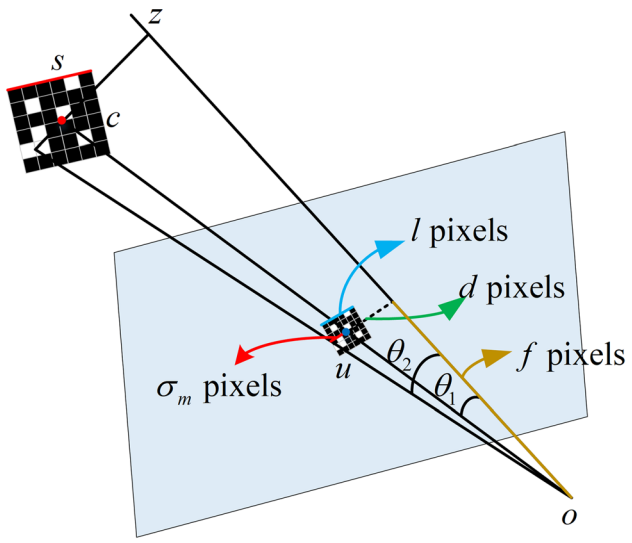


Fig. 3 Computation of the approximate measurement uncertainty. o is the camera centre and oz is the optical axis. c is the initial estimation of the tag centre in the scene corresponding to the pixel u. s is the physical size of the tag and l is corresponding projection length in the image. d is the pixel distance of u deviated from the pixel corresponding to the camera centre

$$\begin{aligned} \sigma_r &= \Delta\theta = \theta_2 - \theta_1 \\ &= \beta_2 * \left[\arctan\left(\frac{d+\sigma_m}{f}\right) - \arctan\left(\frac{d}{f}\right)\right] \end{aligned} \tag{26}$$

where β_1 and β_2 are the scale factor. In our implementation, $\beta_1 = \beta_2 = 5$.

3.5 Incremental smoother

The smoother thread minimize the cost function of Eq. (8). Commonly, nonlinear optimization methods are used, such as Gauss–Newton iterations or the Levenberg–Marquardt algorithm. Given the initial estimate, Gauss–Newton finds an update from the normal equation:

$$\boldsymbol{J}^T\boldsymbol{W}\boldsymbol{J}\boldsymbol{\delta} = -\boldsymbol{J}^T\boldsymbol{W}\boldsymbol{e} \tag{27}$$

where $\boldsymbol{e}$ represents the residual errors and $\boldsymbol{\delta}$ are the updates for the states. $\boldsymbol{W}$ is the information matrix. The general Jacobian $\boldsymbol{J}$ is composed of the Jacobian of IMU residual with respect to IMU states and the Jacobian of reprojection error with respect to current body pose and tag pose in the world frame.

Full smoothing method, also called batch optimization, keeps estimating entire history of the states. It can guarantee highest accuracy. However, it becomes infeasible quickly in computation as the states expand over time. iSAM2 (Kaess et al. 2011) is an efficient incremental smoothing algorithm over the graphical model. It uses a novel data structure,

Table 1 Dataset characteristics

Name	Image	IMU data	Tag	Duration (s)	DF rate[a] (%)	GT[b]	Comment
table	1269	12,690	3	63	8.82	Vicon 6D	–
dataset_1	1190	11,890	3	59.5	17.14	Vicon 6D	Fast motion and blur
cube	4907	49,070	34	245	2.06	–	2 rounds, ~ 70 m/round
pavillon	8690	86,879	34	434	11.19	–	3 rounds, ~ 80 m/round, indoor and outdoor

[a] DF Rate means detection failure rate, that is the percentage of images in which no tag is detected.

[b] GT means ground truth. – means no ground truth

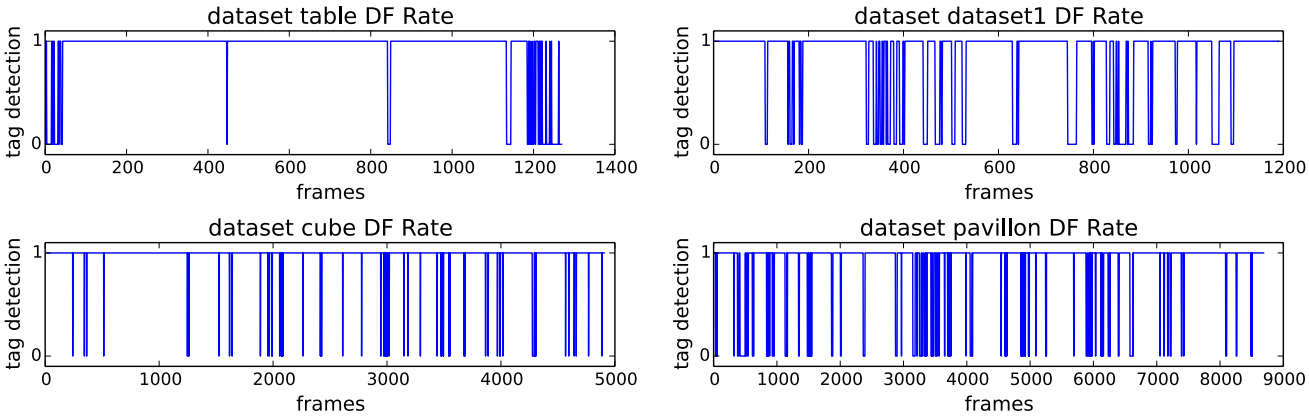


Fig. 4 The detection failure statistics of the 4 public datasets. The value is set to 1 if one or more tags are detected and 0 if no tag is detected

Bayes tree, to encode a factored probability density. Within Bayes tree, variables can be reordered at every incremental update that keeps minimal fill-in in the square root information matrix. Besides, iSAM2 performs a partial state update exploiting the fact that the new measurements have only a local effect. All these make iSAM2 retain sparseness and nearly constant-time computation while still obtain full accuracy. In our experiments, we use GTSAM4.0 (Dellaert 2012)[2] to perform incremental smoothing and implement IMU preintegration.

4 Intermediate failures recovery

As IMU noise integration expands quickly, it is considered a intermediate track failure if it fails to detect tags for a period of time. Once a failure is detected, we activate the intermediate failure recovery mechanism. In the recovery mechanism, we process every frame to detect the tags which have been present in the optimizer. Frames successfully detect the re-observed tags are called recovery frames. Two consecutive recovery frames are used for recovery. The re-observed tags can provide prior pose information. The IMU bias is assumed to be a constant during the failure period. According Eq. (18), we can compute the initial velocity of the first recovery frame by

$$\boldsymbol{v}_i = \frac{\boldsymbol{p}_{i+1} - \boldsymbol{p}_i - \frac{1}{2}\boldsymbol{g}\Delta t_{i\,i+1}^2 - \boldsymbol{R}_i \Delta\tilde{\boldsymbol{p}}_{i\,i+1}}{\Delta t_{i\,i+1}} \tag{28}$$

where $\boldsymbol{p}_{i+1}$ and $\boldsymbol{p}_i$ are the positions of the two recovery frames and can be provided by the re-observed tags. $\boldsymbol{R}_i$ is the orientation of the first recovery frame which also can be obtained from the known pose tags. $\Delta\tilde{\boldsymbol{p}}_{i\,i+1}$ is the IMU preintegration incorporating bias updates

$$\Delta\tilde{\boldsymbol{p}}_{i\,i+1} = \Delta\bar{\boldsymbol{p}}_{i\,i+1} + \boldsymbol{J}^{g}_{\Delta\bar{\boldsymbol{p}}_{i\,i+1}} \delta\boldsymbol{b}^{i}_{g} + \boldsymbol{J}^{a}_{\Delta\bar{\boldsymbol{p}}_{i\,i+1}} \delta\boldsymbol{b}^{i}_{a} \tag{29}$$

For the reason of consistence, we use one tag to compute $\boldsymbol{v}_i$. We may detect several tags in the recovery frames while only the tag that has minimum re-projection error is selected for recovery. Once we get all initial states of the first recovery frame, we can add the tag factor to the optimizer with the prior information and keep tracking again.

[2] https://bitbucket.org/gtborg/gtsam/.

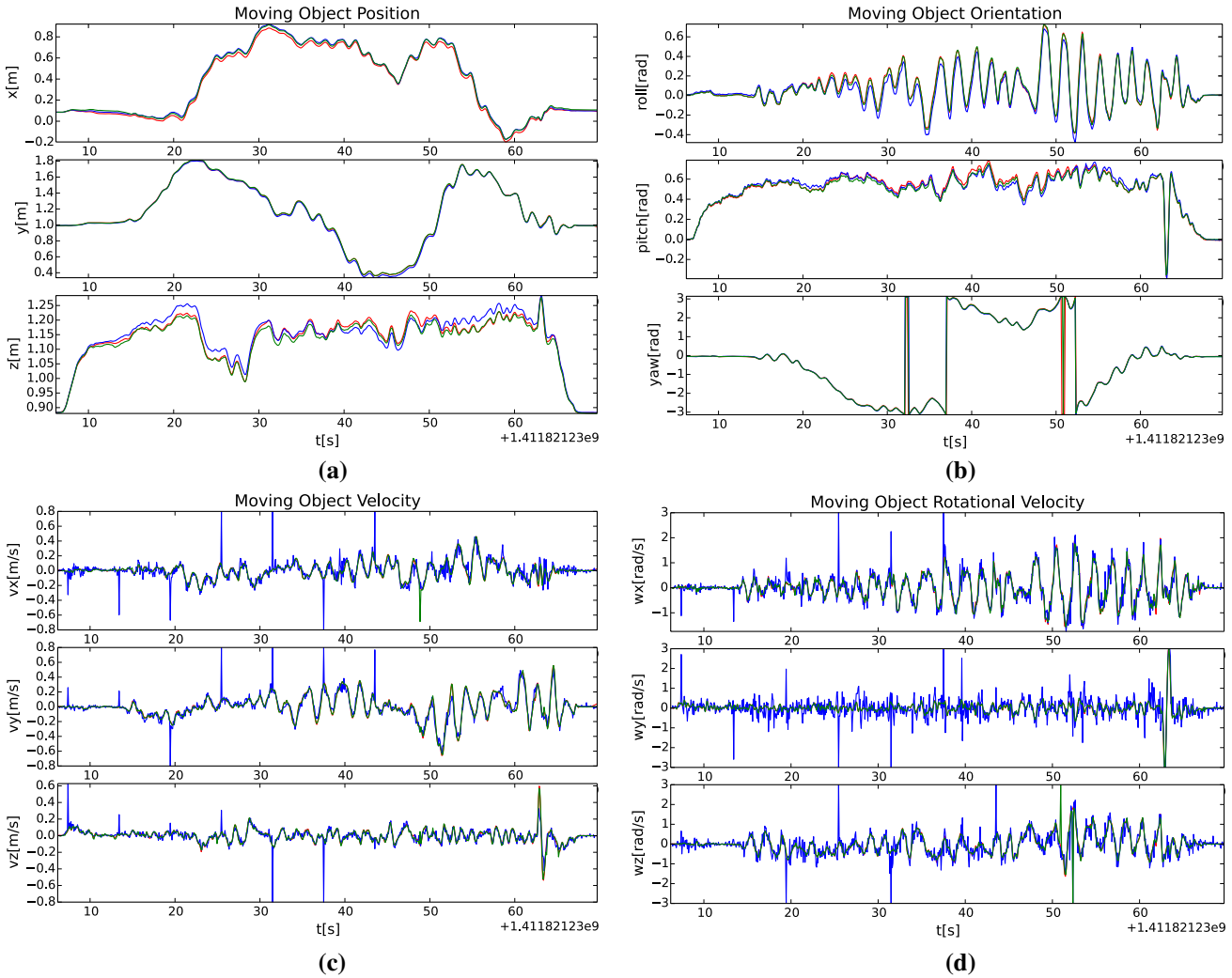


Fig. 5 Comparison of our proposed approach (red), vicon ground truth (blue) and the EKF method (green) in Neunert et al. (2016) on *table* sequence. **a** Positions. **b** Orientations. **c** Linear velocities. **d** Rotational velocities. For better comparison, all velocity estimates are calculated by finite differences of the position/orientation data. The orientation plots are discontinuous due to the wrap-around at $\pm\pi$. As the *table* sequence is easy, the results obtained from our approach and the EKF method in Neunert et al. (2016) are almost same and nearly overlap the ground truth

5 Experiments and results

Our system incorporates two threads that run in parallel: the tag detector thread and the smoother thread. The tag detector thread provide visual measurements, including 4 corner pixel coordinates and the unique tag id of each detected tag, and the initial tag pose relative to the camera for the smoother thread. The smoother thread use iSAM2 to optimize all states at each image time in nearly constant-time.

We perform five experiments to evaluate the proposed motion capture system. In the first experiment, we compare our method with EKF-based method (Neunert et al. 2016) on public datasets. We perform detailed numerical analyses to show the accuracy and robustness of our system. A series of tests are then carried out to further evaluate the robustness to changing lighting conditions and motion blur on 4 groups of real-world datasets. We then further evaluate the scalability and the ability to of big loop closure on larger-scale datasets. A real-world experiment is performed to test the ability of intermediate failures recovery. In the last experiment, we do comparisons to show the effectiveness of the adaptive measurement uncertainty calculation method, described in Sect. 3.4, and the generalization of our system.

5.1 Comparisons on public datasets

We have evaluated our approach in 4 different public sequences provided by Neunert et al. (2016)[3] (see Fig. 1).

[3] https://bitbucket.org/adrlab/rcars/wiki/Home.

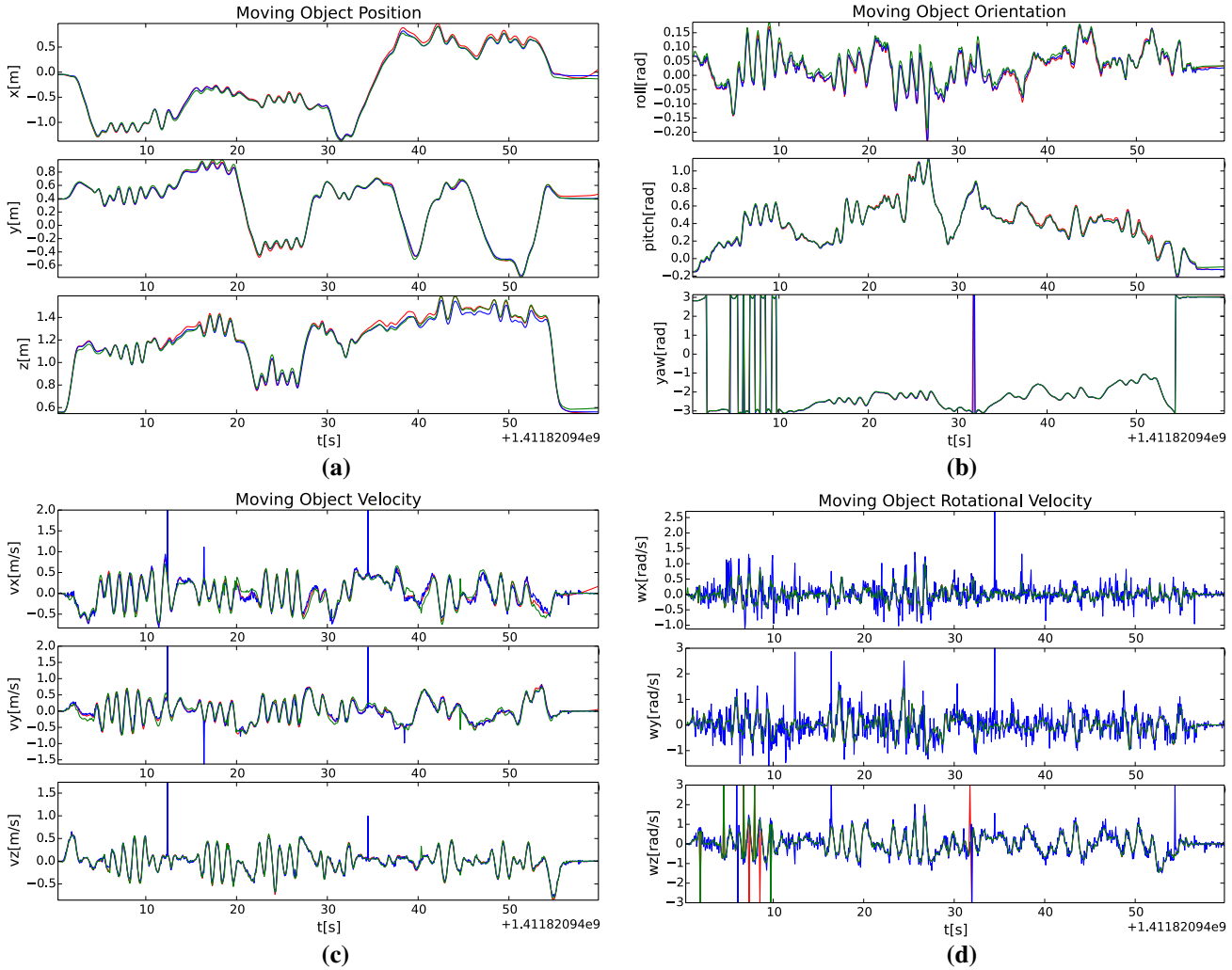


Fig. 6 Comparison of our proposed approach (red), vicon ground truth (blue) and the EKF method (green) in Neunert et al. (2016) on *dataset_1* sequence. **a** Positions. **b** Orientations. **c** Linear velocities. **d** Rotational velocities. For better comparison, all velocity estimates are calculated by finite differences of the position/orientation data. The orientation plots are discontinuous due to the wrap-around at $\pm\pi$. The *dataset_1* dataset has been made artifically difficult with sparse tag coverage and fast motion. Our method get slightly better estimates in orientations, as the results of our method almost overlap the ground truth. These results show that our approach is robust to fast motion and the absence of tags

The main characteristics of these 4 datasets are summarized in Table 1. The *dataset_1* sequence is a challenging dataset in which the AprilTags are sparsely distributed with a high tag detection failure rate of 17.14%, as shown in Fig. 4. Furthermore, Fig. 1b shows it suffers from fast motion blur. The *cube* and *pavillon* sequences, two large scale datasets moving from basement to ground level or from indoors to outdoors, contain round trips and are subject to changing light condition. We make comparisons with vicon ground truth and the estimated results of Neunert et al. (2016) on the first two datasets, and use loop closure to validate the accuracy of our approach in large-scale environment on the last two datasets. To test the generalization ability of our approach, all these 4 datasets use the same parameters in our experiments.

5.1.1 Table and dataset_1

We use two datasets, *table* and *dataset_1*, which have 6DOF ground truth, to evaluate the performance of motion estimation. We compare our motion estimation with both of Vicon ground truth and the results of Neunert et al. (2016).

In the *table* dataset, three tags are placed flat on a table at the same orientation. Comparisons shown in Fig. 5 indicate our approach obtain an accurate estimation. As shown in Fig. 5a, b, in the most of the trajectory our estimates nearly overlap the ground truth, and the maximum position error

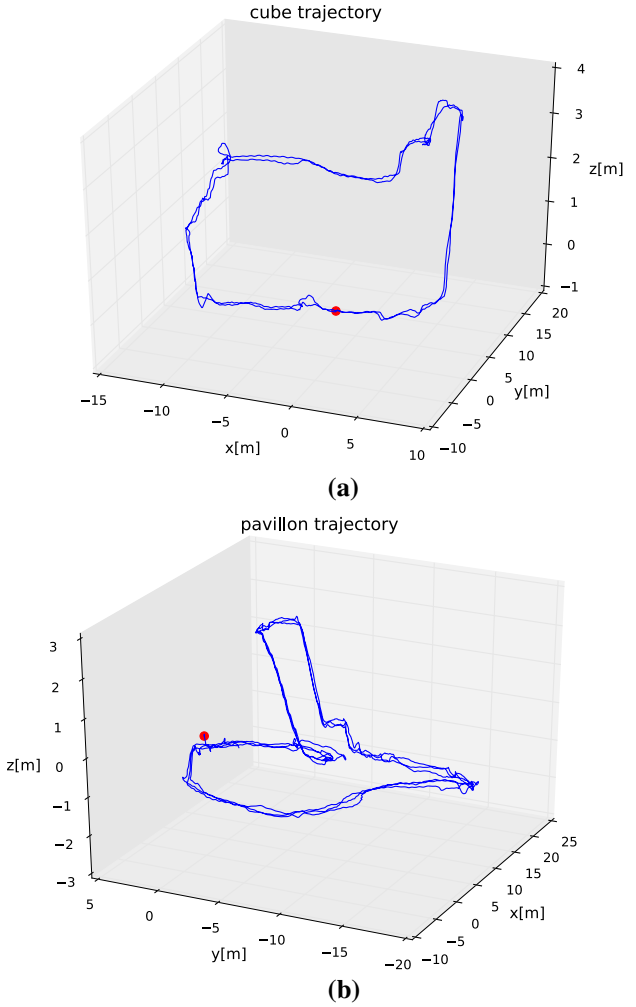


Fig. 7 The trajectory estimates for our approach on *cube* **a** sequence and *pavillon* **b** sequence. The starting position is marked by bold read dot. Both of the two trajectory estimates get good loop closures, showing the scalability to large-scale environment and the robustness to changing lighting conditions

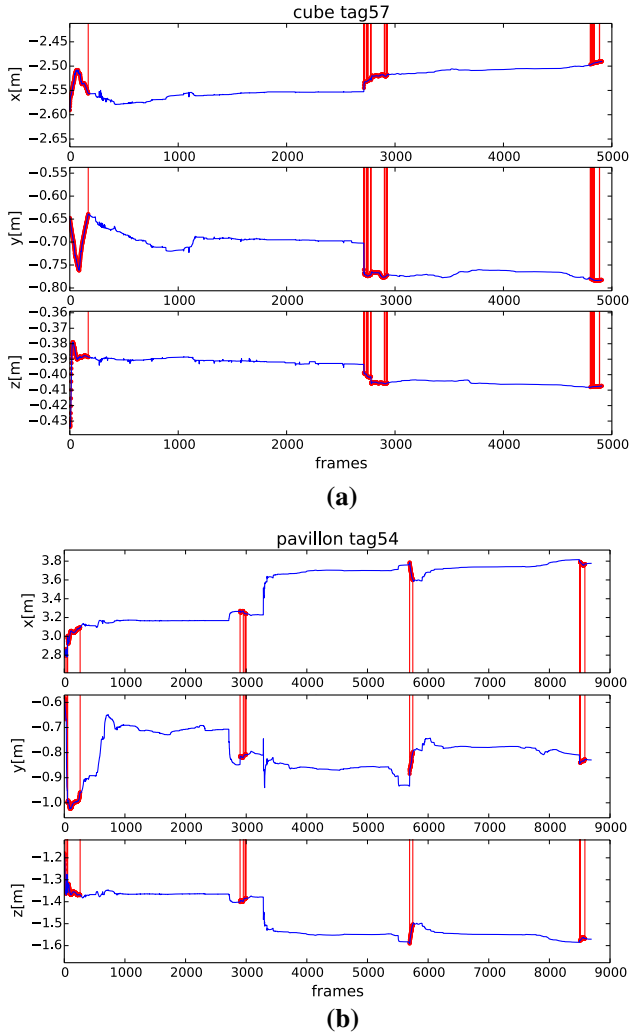


Fig. 8 Position plots for tag, whose id is 57, in *cube* (**a**) and tag 54 in *pavillon* (**b**). As smoothing method is used, the tag positions will be kept in the smoother since first observation. Actually the tags are not visible over the entire trajectory. Frames in which the tags are real visible are highlighted by bold red line and through this we can clearly see when loops are closed. These results validate that our approach can treat loop closures well explicitly and can obtain accurate estimations in large-scale and changing lighting conditions environment

in all axes is within 4 cm and the maximum orientation error in all axes is less than 5°. Figure 5c, d show the linear and rotational velocity estimates. We can see our estimates agree well with the ground truth, while it is noteworthy that compared to the ground truth and the estimated results in Neunert et al. (2016), our estimated results are outlier free as there exit no occasional peaks. Also we find that our full incremental smoothing method and filtering method in Neunert et al. (2016) almost have the same accuracy. One probable reason is that the *table* dataset is relatively simple and the workspace size is relatively small so the advantages of incremental smoothing is not reflected. Difference will come up in the more challenging *dataset_1* dataset and especially in the last two large-scale datasets.

Figure 6 shows the comparison results of the *dataset_1*. As it shows, we get a highly reliable estimate though there exists blur and the tags are sparsely distributed in this dataset. Compared to Neunert et al. (2016), our algorithm represents slightly more robust performance to fast motion and the absence of tags. While we get almost the same maximum position error in x and y axis (about 6 cm), our maximum position error (within 3 cm) is obviously smaller than Neunert et al. (2016) (over 8 cm) in y axis. Advantages are more obvious in orientation as our approach is more accurate than Neunert et al. (2016) nearly over the entire trajectory in roll (less than 1° in most of the trajectory vs. ~2.5°). The maximum orientation error in three axes is less than 2°, which

Table 2 Positions of the tag57 in *cube* sequence around the two loop closures

Frame	x (m)	y (m)	z (m)
2711	− 2.552870	− 0.701140	− 0.393095
2712	− 2.552870	− 0.701140	− 0.393095
2713	**− 2.544910**	**− 0.760228**	**− 0.398702**
2714	− 2.539093	− 0.767750	− 0.398894
2715	− 2.534418	0.766083	− 0.398725
4800	− 2.497738	− 0.778147	− 0.408248
4801	− 2.497738	− 0.778147	− 0.408248
4802	**− 2.496689**	**− 0.779559**	**− 0.408046**
4803	− 2.496669	− 0.779577	− 0.408049
4804	− 2.496669	− 0.779577	− 0.408049

The bold rows indicate the frames where the loops are closed. These data show that the positions are stable at loop closures

Table 3 Positions of the tag54 in *pavillon* sequence around the three loop closures

Frame	x (m)	y (m)	z (m)
2897	3.264784	− 0.847778	− 1.401861
2898	3.264784	− 0.847778	− 1.401861
2899	**3.271305**	**− 0.817749**	**− 1.401375**
2900	3.264874	− 0.814845	− 1.398138
2901	3.263837	− 0.814247	− 1.397623
5693	3.767093	− 0.933071	− 1.585778
5694	3.767093	− 0.933071	− 1.585778
5695	**3.785211**	**− 0.884717**	**− 1.590338**
5696	3.781866	− 0.874544	− 1.588108
5697	3.777749	− 0.869719	− 1.585975
8503	3.792405	− 0.842300	− 1.576825
8504	3.792405	− 0.842300	− 1.576825
8505	**3.792388**	**− 0.842298**	**− 1.576815**
8506	3.791885	− 0.840557	− 1.577049
8507	3.790774	− 0.839990	− 1.576753

The bold rows indicate the frames where the loops are closed. These data show that the positions are stable at loop closures

is fairly accurate. As for the velocity estimates, we can see from the Fig. 6c, d that our estimates exist least occasional peaks.

5.1.2 Cube and pavillon

We stress that our approach is scalable since it is not limited to small size indoor environment and can also work well in large-scale and outdoor environment. Compared to Neunert et al. (2016), which gets slower and becomes infeasible in large-scale environment, our approach can retain

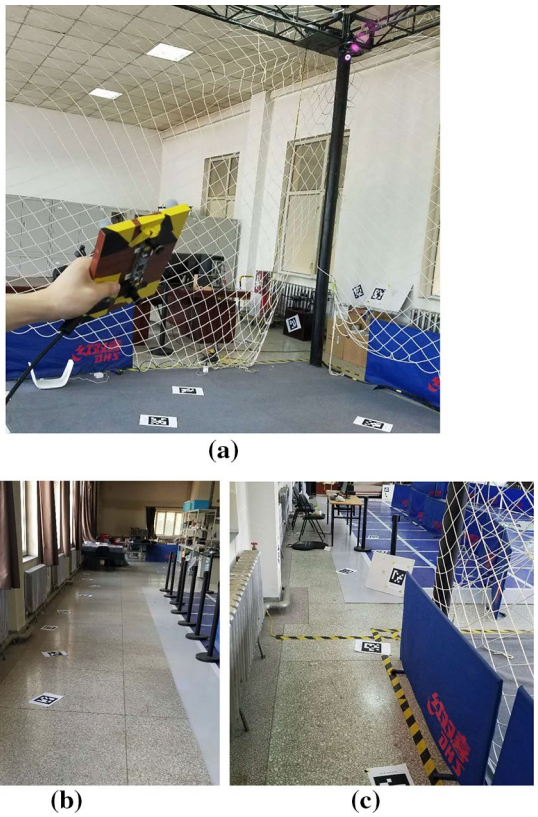


Fig. 9 The scene for data collection. **a** We adjust the exposure time to obtain different lighting conditions and shake the handheld camera in different degree to obtain different velocity. **b** and **c** The scene for large-scale data collection. AprilTags are distributed along a roundabout trajectory

nearly constant-time complexity. Here *cube* and *pavillon* sequences, two large-scale datasets, are used to validate the scalability and the robustness to changing light condition of the presented method. As no motion capture system can cover such a large area, we use loop closure to evaluate our estimation. More specifically, the reprojection error at loop closure before optimized and the position offset of the tag after optimized at the first time we reobserve it after a round are used.

Figure 7 shows the estimated trajectories of the *cube* and *pavillon* sequences. As we can see, our method can obtain a high-quality loop closure, verifying the accuracy and the scalability to large-scale environment. As Neunert et al. (2016) doesn't provide the detailed performance and the trajectory estimates of these two large-scale datasets, we don't make comparisons here. In Fig. 8, we plot two tag position estimations of the *cube* and *pavillon* sequences, respectively, (tag57 and tag54). As we use incremental smoothing method and the two selected tags are visible from the begining of the sequences, these two tag poses are kept being optimized over the entire trajectory. In fact, they are not visible in all

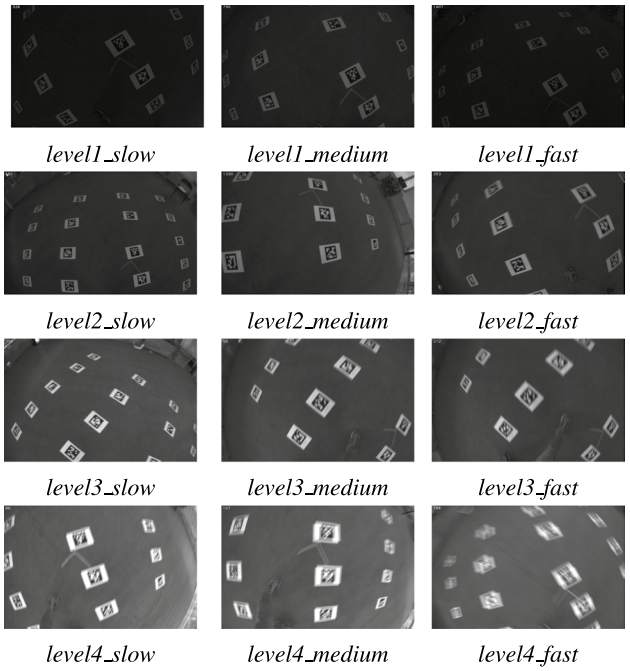


Fig. 10 Images extracted from the 4 groups of datasets. **a–c** correspond to slow, medium, fast of the level1 lighting condition, and **d–f** to level2, **g–i** to level3, **j–l** to level4

Table 4 Translational and angular accuracy in the 4 group datasets

Velocity	Lighting			
	level1	level2	level3	level4
Slow				
v_{max} (m/s)	1.643	1.523	1.517	2.451
ω_{max} (rad/s)	3.271	3.399	3.489	4.547
Trans. RMSE (m)	0.026	0.040	0.039	0.063
Ang. RMSE (deg)	1.363	1.795	2.131	4.534
Medium				
v_{max} (m/s)	2.903	2.532	2.790	Failed
ω_{max} (rad/s)	5.345	4.960	5.724	Failed
Trans. RMSE (m)	0.051	0.042	0.020	Failed
Ang. RMSE (deg)	3.727	2.804	3.311	Failed
Fast				
v_{max}(m/s)	3.641	4.229	3.849	Failed
ω_{max} (rad/s)	6.273	8.951	8.886	Failed
Trans. RMSE (m)	0.039	0.049	0.058	Failed
Ang. RMSE (deg)	3.875	4.867	5.500	Failed

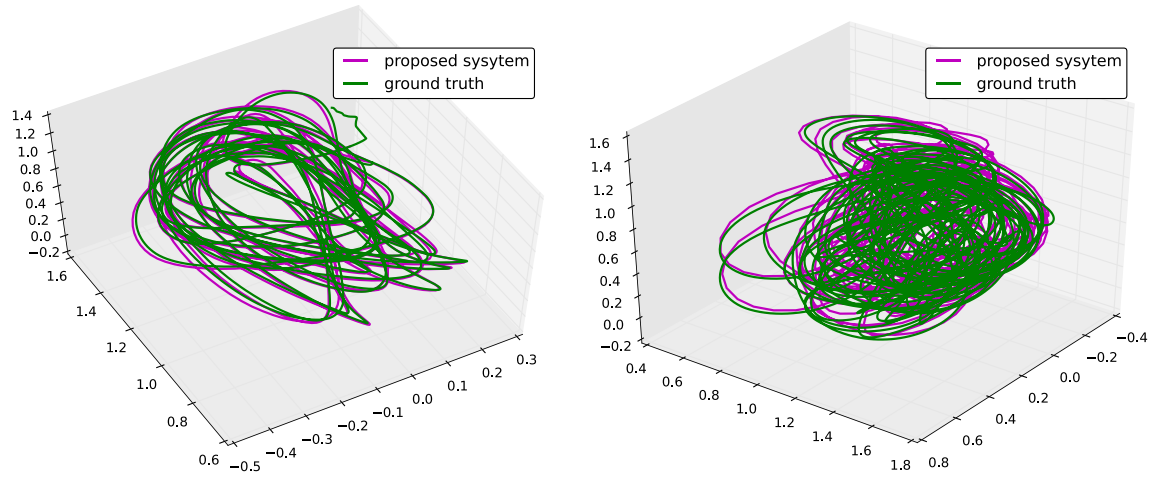
(Considering the errors induced by coordinate frames and time alignment between ground truth and the estimated results, occasional data missing in the ground truth, the RMSEs could be even smaller)

images instead they can only be observed in part of the sequences which are highlighted by bold red line in Fig. 8. From Fig. 8, we can see clearly when these two tags are reobserved again, at which the loops are closed. The *cube* dataset includes two rounds. Each round trip until loop closure is about 70m long. Two loop closures are detected for the tag57. The average position offset over these two loop closure is about 3 cm (~6 cm in the first loop closure and almost zero in second loop closure) thus the relative error is around 0.043%. The average reprojection error is about 93.7 pixels (175.1 pixels in first loop closure and 12.3 pixels in second loop closure). While the reprojection error is large in the first loop closure, it decreases drastically, indicating high stability of our method. For the *pavillon* dataset, three rounds are included. Each round trip until loop closure is about 80m. Three loop closure are detected for the tag54. The average position offset over the three loop closure is about 2.73 cm (3 cm, 5.2 cm in the first two loop closures, and almost zero in the last loop closure) thus the relative offset is around 0.034%. The average reprojection error is about 28.8 pixels (28.2 pixels, 45.8 pixels and 12.3 pixels in the three loop closure, respectively). We notice that there is a jump (position offset is about 40 cm) between frame 3000 and frame 4000 in Fig 8b. We guess it is the long-term tag detection failure that causes this jump. As shown in Fig. 4, images in which no tag is detected gather together around the jump. Tables 2 and 3 give the detailed position data around the loop closures, where the frames at which the loops are closed are bold. The position fluctuation before and after loop closure is very small, indicating that our method can handle the loop closures well explicitly and can obtain an accurate estimation in large-scale environment, despite the changing lighting conditions.

5.2 Robustness experiment

This experiment is designed to further evaluate the robustness to changing lighting conditions and motion blur. We collect 4 groups of datasets and each one corresponds to a level of lighting condition, i.e. level1, level2, level3 and level4. Each group has 3 datasets collected under 3 different velocity, i.e. slow, medium and fast. All datasets are collected by a handheld visual-inertial sensor, as shown in Fig. 9a, and provided with ground truth by OptiTrack motion capture system at rate of 100 Hz. The IMU rate is set to 500 Hz and the camera rate is set to 25 Hz. We shake the visual-inertial sensor in different degree to obtain different velocity, i.e. smoothly for the slow dataset and violently for the fast dataset. Exposure time is fixed to different values to obtain various lighting conditions. Figure 10 shows the extracted images from these 4 groups of datasets. As we can see, the level1 is very dark and becomes brighter and brighter to level4. Low exposure time renders dark pictures, but mitigates motion blur.

We evaluate the accuracy of our system in the 4 group datasets by comparing with OptiTrack ground truths. Table 4 shows the translational and angular Root Mean Square Error



(a) Trajectory of *level1_low* **(b)** Trajectory of *level2_fast*

Fig. 11 Overhead plots of the OptiTrack ground truths

Fig. 12 Images extracted from the *large_2* dataset. Since we fixed the exposure time, the lighting condition along the trajectory varies very much. Areas beside the outdoor window are very bright while very dark in the area far away from the window

Table 5 Dataset characteristics

Name	Image	IMU data	Tag	Duration (s)	Comment
large_1	3952	79,788	122	158.9963	1 lap, ~ 125 m
large_2	5281	106,425	122	212.0881	2 laps, ~ 125 m/lap

(RMSE) of each dataset. Figure 11 shows two overhead trajectory plots of OptiTrack ground truths. Our system successfully processes all these datasets, except *level4_medium* and *level4_fast*, in which the motion blur is so extreme that most of the frames can't detect tag successfully. Results show that our system is very robust to changing lighting conditions and is capable to survive from motion blur in a degree. In comparison, our system is more robust to changing lighting conditions than motion blur.

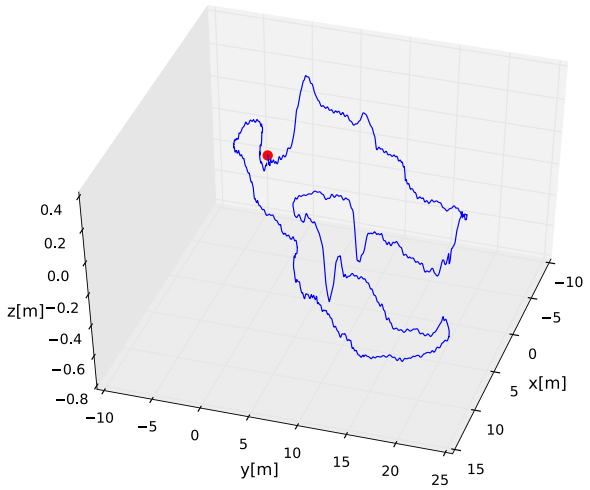
5.3 Larger-scale datasets

To further assess the ability for loop closure of our system, we test on the other two larger-scale datasets. These two large-scale dataset, also collected by the handheld visual-inertial sensor (shown in Fig. 9a), are named by *large_1* and *large_2*. Figure 9b, c show the scene for data collection. The *large_1* has one lap and the *large_2* has two laps, with 125m/lap. Since we fixed the exposure time, the two datasets suffer a lot from lighting condition changing. As show in Fig. 12, areas beside the outdoor window are very bright while areas away are very dark. 122 tags are distributed along a roundabout trajectory to construct a big loop. For the convenience of loop closure evaluation, we marked the start point of the trajectory when collected datum and end the trajectory (almost) at the start point intendedly. More detailed information of the two datasets are listed in Table 5.

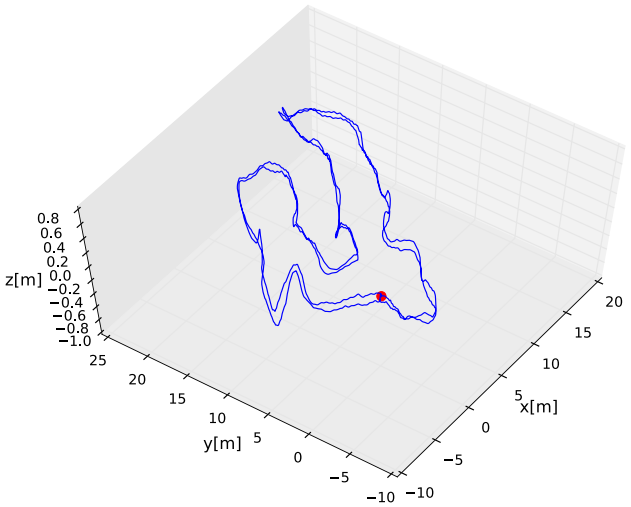
Figure 13a shows the estimated trajectory of *large_1* and Fig. 13b shows the estimated trajectory of *large_2*. The red points are the start points of the trajectories. From the results, we can see both of trajectories almost exactly end at the start points as expected, indicating that our system close the loop very well. Figure 14 shows the tag332 position plot. We can see the tag position is stable when closed loop although there are some fluctuations up and down within 10 cm. These two experiments again validate that our system is scalable to large-scale environment and robust to lighting condition changing.

5.4 Intermediate failure recovery experiment

We test the recovery mechanism on several real-world datasets. These datasets are designed to suffer from intermediate



(a) Estimated trajectory of *large_1* (1 lap).



(b) Estimated trajectory of *large_2* (2 laps).

Fig. 13 The red point are the start points of the trajectories. Both of the estimated trajectories end almost at the start point (the red points) and close the loop well (Color figure online)

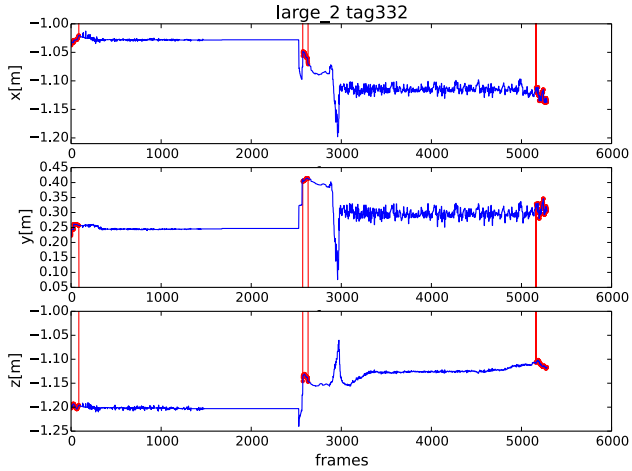


Fig. 14 Position plots for tag332 in *large_2* dataset. As smoothing method is used, the tag position will be kept in the smoother since first observation. Actually the tags are not visible over the entire trajectory. Frames in which the tags are observable are highlighted by bold red line and through this we can clearly see when loops are closed. We can see the tag position is stable at loop closure although some fluctuations exist

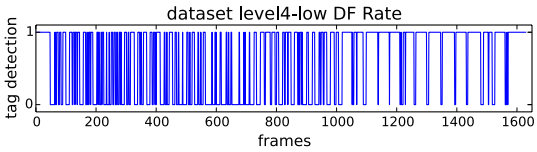


Fig. 15 The detection failure statistics of the *level4_low* dataset. The value is set to 1 if one or more tags are detected and 0 if no tag is detected (DF Rate means detection failure rate)

failures. Figure 16 shows the evaluation results on the *level4_low* dataset. This dataset suffers from serious motion blur and more than 39.9% of the frames cant observe the tag, as shown in Fig. 15. In our implementation, we judge it a intermediate failure if it fails to detect tags more than 1 s. As the frame rate is 25 Hz, so we judge it failed if more than 25 consecutive frames cant observe any tag. One intermediate failure is detected in *level4_low* dataset and is circled by black boxes in Fig. 16. We can see the discontinuity from the enlarged insets in Fig. 16a, b at which an intermediate failure is detected and our system can recovers from it and gets a good performance. Figure 17 displays the estimated trajectory of the *level4_low* dataset and its comparison against the OptiTrack ground truth. The bold black curve shows the discontinuity, indicating that there is an intermediate failure and our system can recovers from it successfully.

Figure 18 displays the estimated trajectory of the other challenging dataset *data_4_failure*, in which 4 intermediate failures are detected. Table 6 shows the detailed position fluctuations at the 4 intermediate failures. Our system can recover from all of the intermediate failures, verifying that our system has the ability to recover form intermediate failures.

5.5 Uncertainty experiment

In this experiment, we aim to study the uncertainty difference provided by the fiducial observations distributed head-on and side and evaluate the adaptive measurement uncertainty calculation method, as described in Sect. 3.4. For comparison, we also provide the results of using constant measurement uncertainty with $\sigma_m = 10$ pixels, $\sigma_t = 0.1$ m, $\sigma_r = 0.02$ rad.

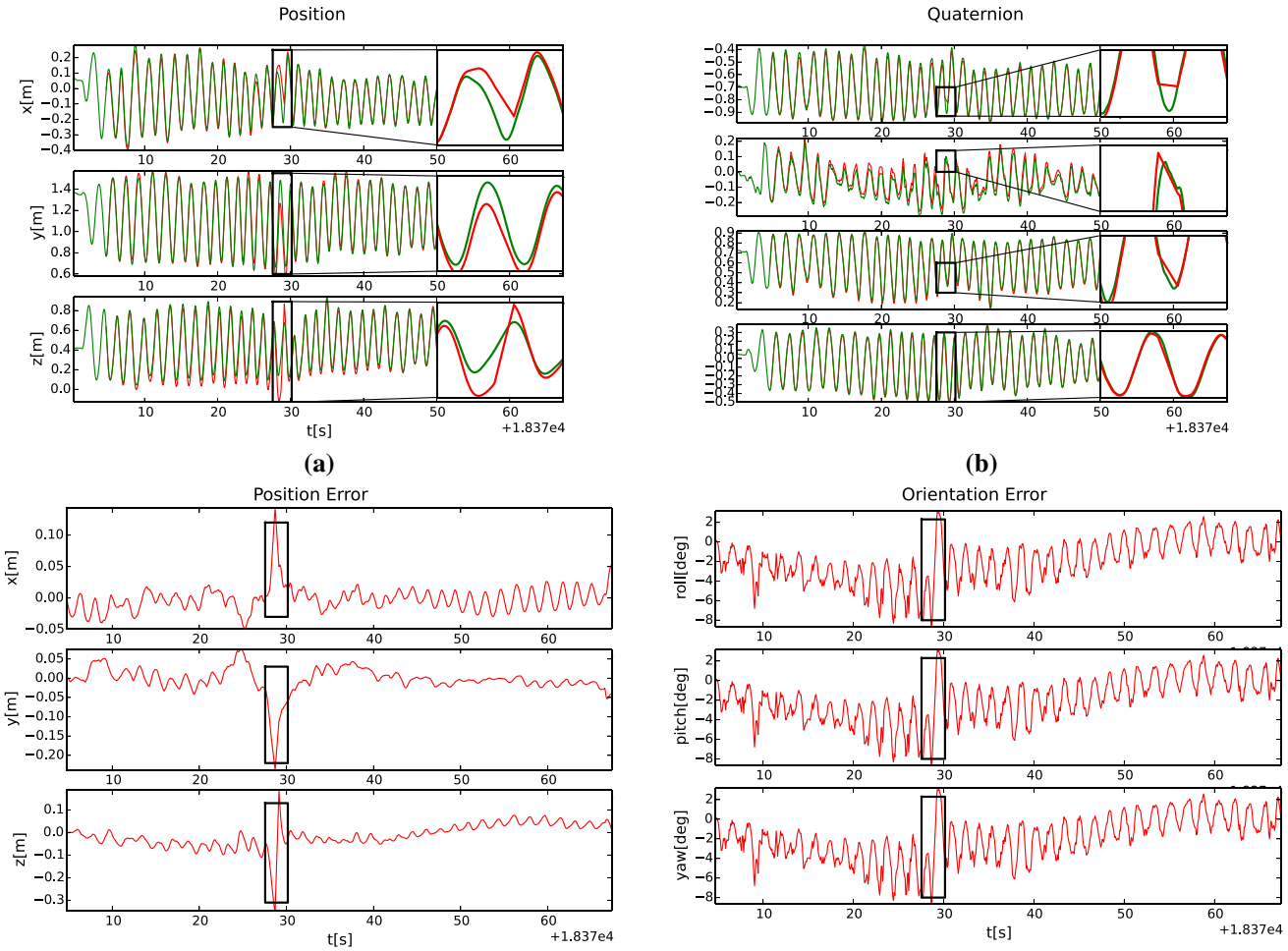


Fig. 16 Evaluation results for the intermediate failure recovery. *Level4_low* dataset is used. **a** Comparison of position between our estimated result (red) and the OptiTrack ground truth (green). **b** Comparison of quaternion between our estimated result (red) and the OptiTrack ground truth (green).The enlarged insets in **a** and **b** show the jump and recovery at intermediate failure. **c** Position error. **d** Orientation error.These results indicate that our system has the ability to recover from intermediate failure

We firstly test on a designed dataset. Figure 19a shows the scene for data collection. Five rows of tags are deployed in the scene and each row has 4 tags. We shake the sensor above the tags and the sensor is tracked by the OptiTrack motion capture system. Figure 19b shows one image from the dataset. Generally, the projections of the tags in row 3 are located in the center of the image and row 1 and row 5 are located in the side of the image. Besides, tags in row 1 are tilted most and tags in row 5 tilted least.

Figure 20 show the translational and angular errors of using constant measurement uncertainty. For comparison, we use one row of tags in each tests, e.g. *Row1* means only use the first row of tags and *All* means use all rows of tags in the optimization. Table 7 shows translational and angular RMSE for each test. From the results, we can see generally tags located in the centre provide less uncertainty compared those located in the side and in consequence get better estimated results.

Figure 21 and Table 8 show the evaluation results for using adaptive measurement uncertainty calculation method described in Sect. 3.4. Compared to the results in Fig. 20 and Table 7, we get better estimates in more cases, showing the effectiveness of the adaptive method. It's noteworthy that using all tags do not get a better estimated result compared

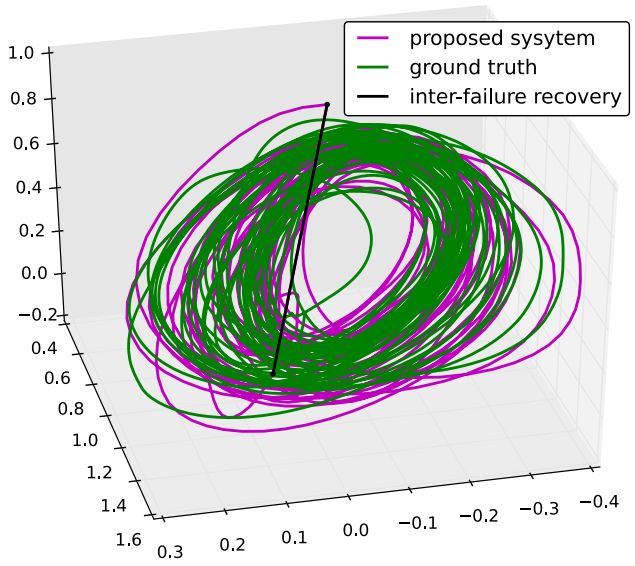


Fig. 17 Comparison of trajectory between our estimated result and the OptiTrack ground truth. The bold black curve shows the jump and successful recovery at intermediate failure

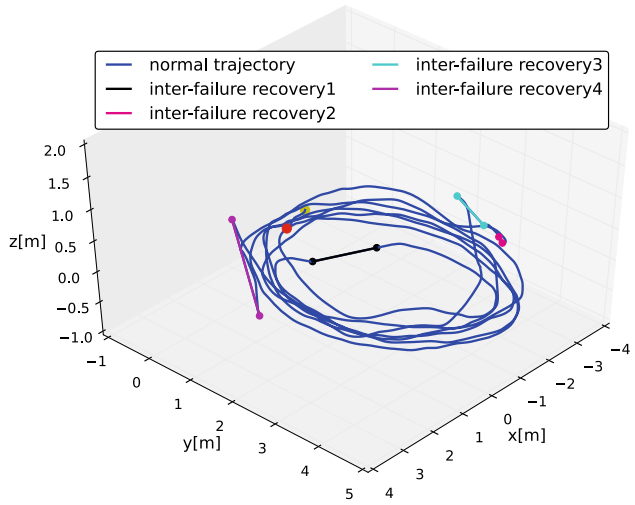


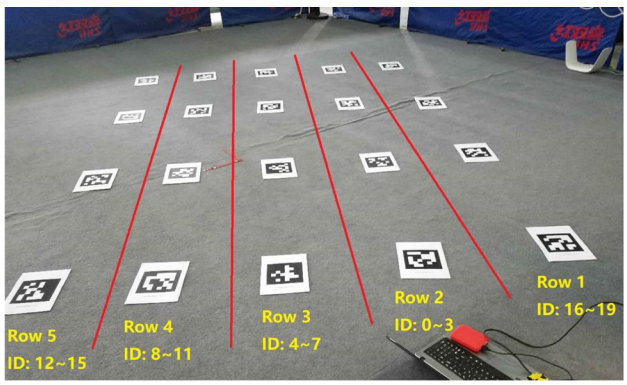
Fig. 18 Estimated trajectory of the challenging *data_4_failure* dataset. 4 intermediate failures are detected in this dataset. Our system can recover from all of the intermediate failures

to using Row3 only, indicating some bad tags can deteriorate the accuracy.

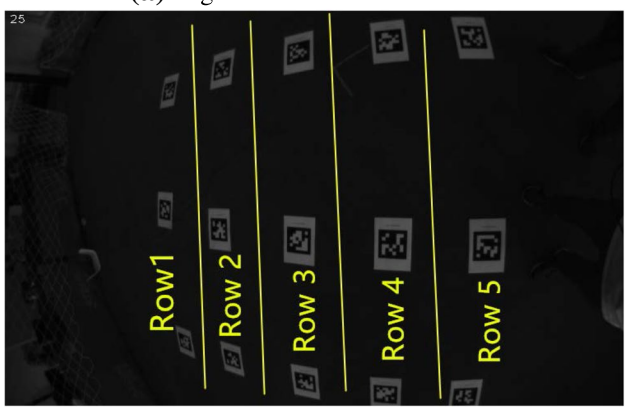
For further evaluation of the adaptive measurement uncertainty calculation method, we performed more comparison tests on the above-mentioned 4 group datasets and the comparison results are shown in Table 9. We can see in some cases the estimated results are improved a lot while get a litter worse in other cases. Generally, it's better to use the adaptive measurement uncertainty calculation method

Table 6 Position fluctuations at the 4 intermediate failures

Time (s)	x (m)	y (m)	z (m)
19281.8823	− 0.74035	0.435518	− 0.66973
19281.9223	**− 0.74326**	**0.431206**	**− 0.67159**
19286.3701	**− 1.60074**	**1.367213**	**− 0.35913**
19286.4502	**− 1.65833**	**1.401088**	**− 0.35258**
19318.3861	**− 2.89895**	3.299355	0.128106
19318.4262	**− 2.87388**	**3.323892**	**0.129766**
19318.9471	**− 2.9441**	**3.363206**	**0.023628**
19318.9871	− 2.93238	3.369428	0.022237
19320.7502	− 3.43447	2.665631	− 0.06244
19320.7903	**− 3.42381**	**2.634428**	**− 0.06254**
19321.4314	**− 3.10769**	**2.203388**	**0.361591**
19321.4715	− 3.07627	2.264048	0.369181
19331.0482	3.17122	1.459582	1.401288
19331.0883	**3.175377**	**1.44239**	**1.425113**
19333.8131	**2.573646**	**1.539521**	**− 0.25155**
19333.8531	2.542245	1.512037	− 0.26681

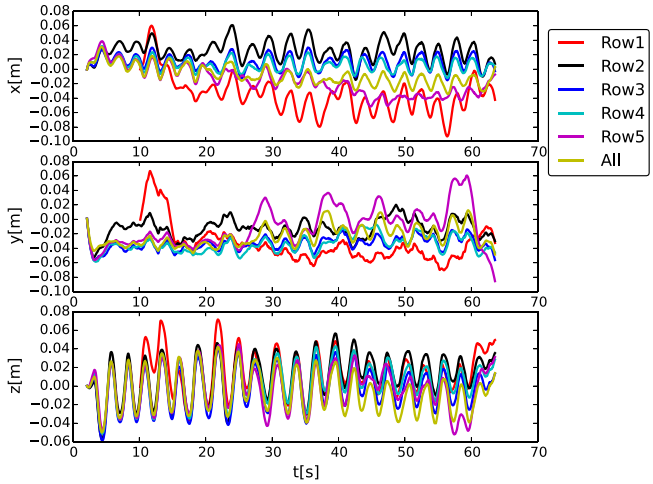


(a) Tags distribution in the scene.

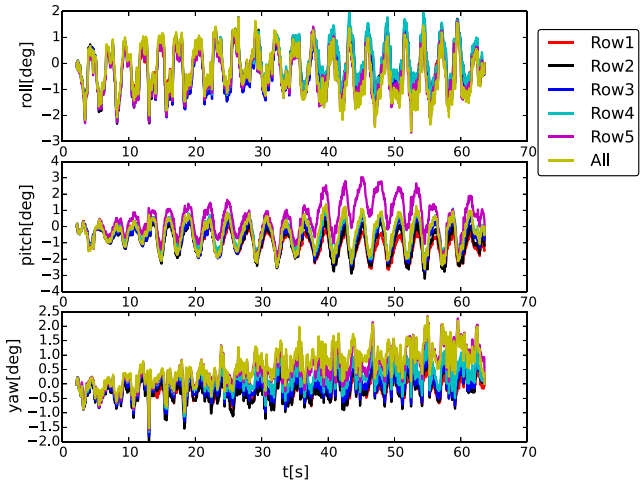


(b) One image extracted from the dataset.

Fig. 19 Generally, the projections of the tags in row 3 are located in the center of the image and row l and row 5 are located in the side of the image. Besides, tags in row 1 are tilted most and tags in row 5 tilted least



(a) Comparisons of translational error.



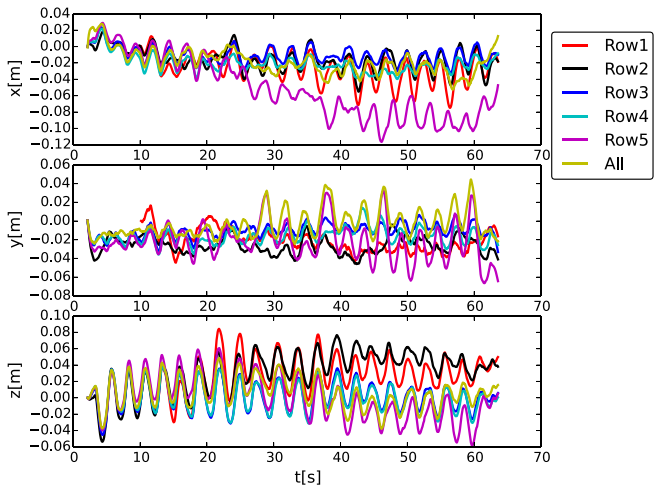
(b) Comparisons of angular error.

Fig. 20 Analysis of accuracy for using different row of tags (e.g, *Row1* means only use the first row of tags and *All* means use all rows of tags in the optimization). Because of the alignment error between ground truth and estimated results, we cut off the first ~8 s for *Row1* for better comparison. Constant measurement uncertainties are used with $\sigma_m = 10$ pixels, $\sigma_r = 0.01$ rad, $\sigma_t = 0.2$ m

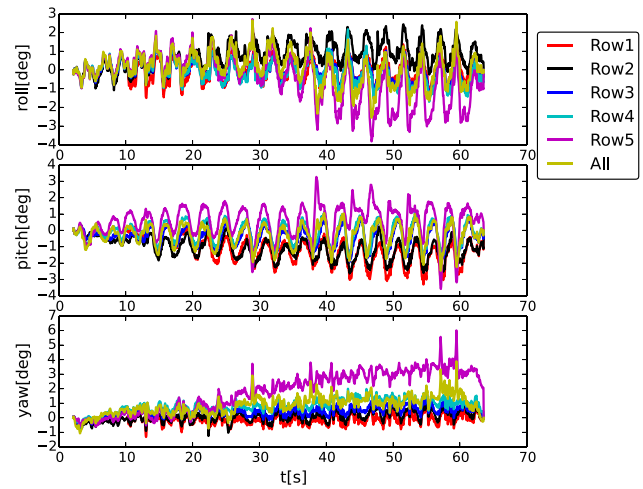
Table 7 RMSEs for using different row of tags

	Row1	Row2	Row3	Row4	Row5	All
Trans. RMSE (m)	0.067	0.042	0.043	0.043	0.044	0.038
Ang. RMSE (deg)	1.458	1.514	1.384	1.272	1.627	1.572

Constant measurement uncertainty with $\sigma_m = 10$ pixels, $\sigma_r = 0.01$ rad, $\sigma_t = 0.2$ m are used



(a) Comparisons of translational error.



(b) Comparisons of angular error.

Fig. 21 Analysis of accuracy for using different row of tags (e.g, *Row1* means only use the first row of tags and *All* means use all rows of tags in the optimization). Because of the alignment error between ground truth and estimated results, we cut off the first ~8s for *Row1* for better comparison. Measurement uncertainties are calculated by the adaptive method

Table 8 RMSEs for using different row of tags

	Row1	Row2	Row3	Row4	Row5	All
Trans. RMSE (m)	0.055	0.053	0.027	0.034	0.074	0.034
Ang. RMSE (deg)	1.657	1.686	1.062	1.310	2.822	1.537

Measurement uncertainty are calculated by the adaptive method

Table 9 Comparison of using constant measurement uncertainty with $\sigma_m = 10$ pixels, $\sigma_r = 0.01$ rad, $\sigma_t = 0.2$ m versus using adaptive calculation method

Dataset	RMSE	Adaptive	Constant
level1_slow	Trans. RMSE (m)	0.026	0.024
	Ang. RMSE (deg)	1.363	1.197
level1_medium	Trans. RMSE (m)	0.051	0.047
	Ang. RMSE (deg)	3.727	3.491
level1_fast	Trans. RMSE (m)	0.039	0.036
	Ang. RMSE (deg)	3.875	3.372
level2_low	Trans. RMSE (m)	0.040	0.064
	Ang. RMSE (deg)	1.795	1.819
level2_medium	Trans. RMSE (m)	0.042	0.043
	Ang. RMSE (deg)	2.804	2.777
level2_fast	Trans. RMSE (m)	0.049	0.081
	Ang. RMSE (deg)	4.867	4.910
level3_slow	Trans. RMSE (m)	0.039	0.042
	Ang. RMSE (deg)	2.131	1.900
level3_medium	Trans. RMSE (m)	0.020	0.024
	Ang. RMSE (deg)	3.311	3.572
level3_fast	Trans. RMSE (m)	0.058	0.052
	Ang. RMSE (deg)	5.500	5.669
level4_low	Trans. RMSE (m)	0.063	0.056
	Ang. RMSE (deg)	4.534	4.440

than to use the constant measurement uncertainty. We want to stress that we don't take much time on parameters tuning for better results and both of these measurement uncertainty can get a good performance, showing the generalization of our system.

6 Discussion

As incremental smoothing technique is used in our system, it can achieve nearly constant-time complexity. In our implementation, it runs at 9–10 frames per second on a 2.7 GHz Intel Core i7-6820HQ processor. While the filter method in Neunert et al. (2016) is fast in small size workspace with a few tags, it gets slower over time and almost aborted in our hardware configuration on *cube* and *pavillon* datasets. Notice that we add every image in the sequence to the smoother without introducing the notion of keyframe. Indeed, if we process one frame every $N + 1$ frames and discard the in-between N frames directly, it becomes several times faster and runs at real-time. We test different N values, from 0 to 4, and the results are given in Figs. 22 and 23. As we can see, the estimation accuracy is just reduced slightly and we still get a reliable result until $N = 4$. We believe it will be more accurate and efficient if a more sophisticated keyframe selection strategy is used.

7 Conclusion and future work

In this work, we proposed a lightweight and scalable visual-inertial motion capture system. Using cheaply paper printable fiducials and no special hardware requirements except a monocular camera and an IMU make our system lightweight and easy to deploy. By tightly fusing IMU measurements with fiducial visual cues using incremental full smoothing and IMU preintegration technique, we can obtain fairly accurate motion estimation and achieve nearly constant-time complexity, which renders our system scalable to large-scale and outdoor environment. Our system is divided into two threads: a tag detector detects tags in every image and provides initial relative tag pose, and a smoother performs visual-inertial joint optimization. We perform extensive evaluations to validate the accuracy, scalability to large-scale environment and robustness to changing light conditions and motion blur of our system. The intermediate failure experiment shows that our system has the ability to recover from intermediate failures. The uncertainty experiment shows the effectiveness of the adaptive measurement uncertainty calculation method. Besides, we show that using constant measurement uncertainty without complicated parameter tuning can also get a good performance, showing the generalization of our system. Since the current system processes all frames which makes it doesn't meet the real-time requirement, our future

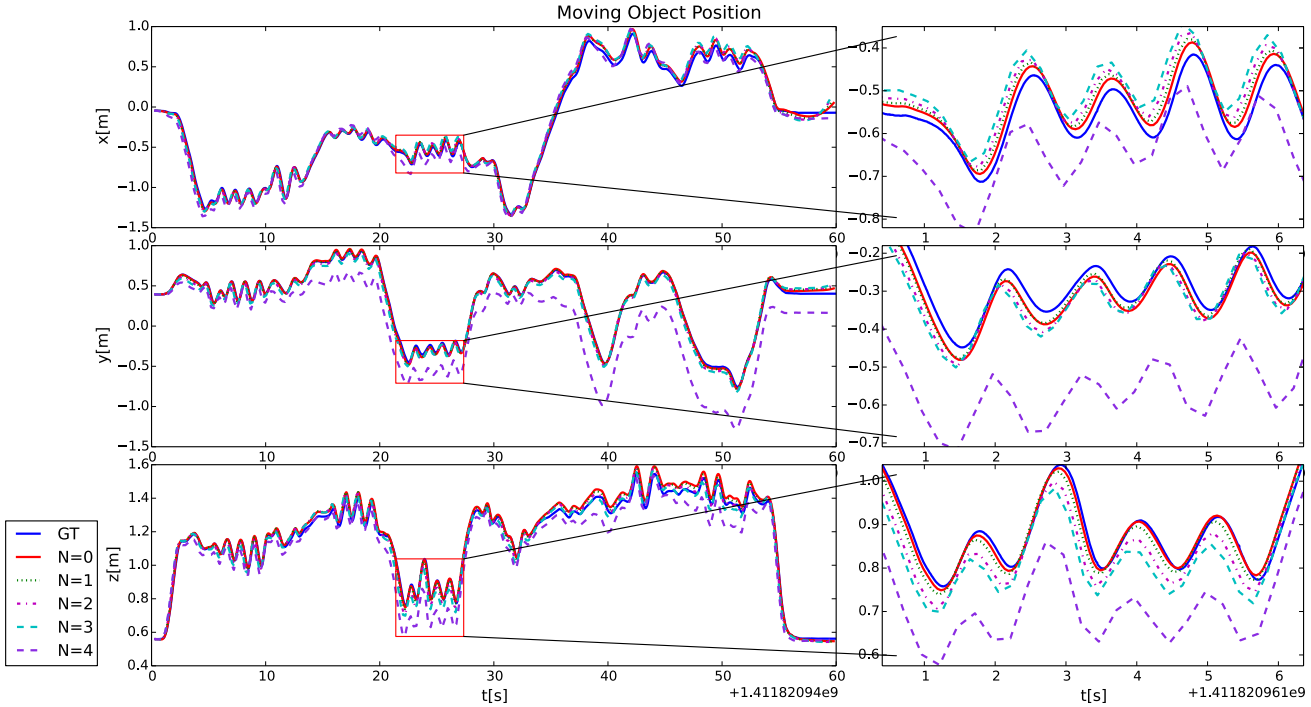


Fig. 22 Moving object position plots of the *dataset_1* for different frame discard strategy. Each curve is one strategy. The area within the red boxes in the left images are zoomed in in the right images. N means we process one frame every $N+1$ frames and discard the in-between N frames directly. GT means ground truth. These results show we can still obtain a good estimate until $N=4$ and it is times faster and runs at real-time

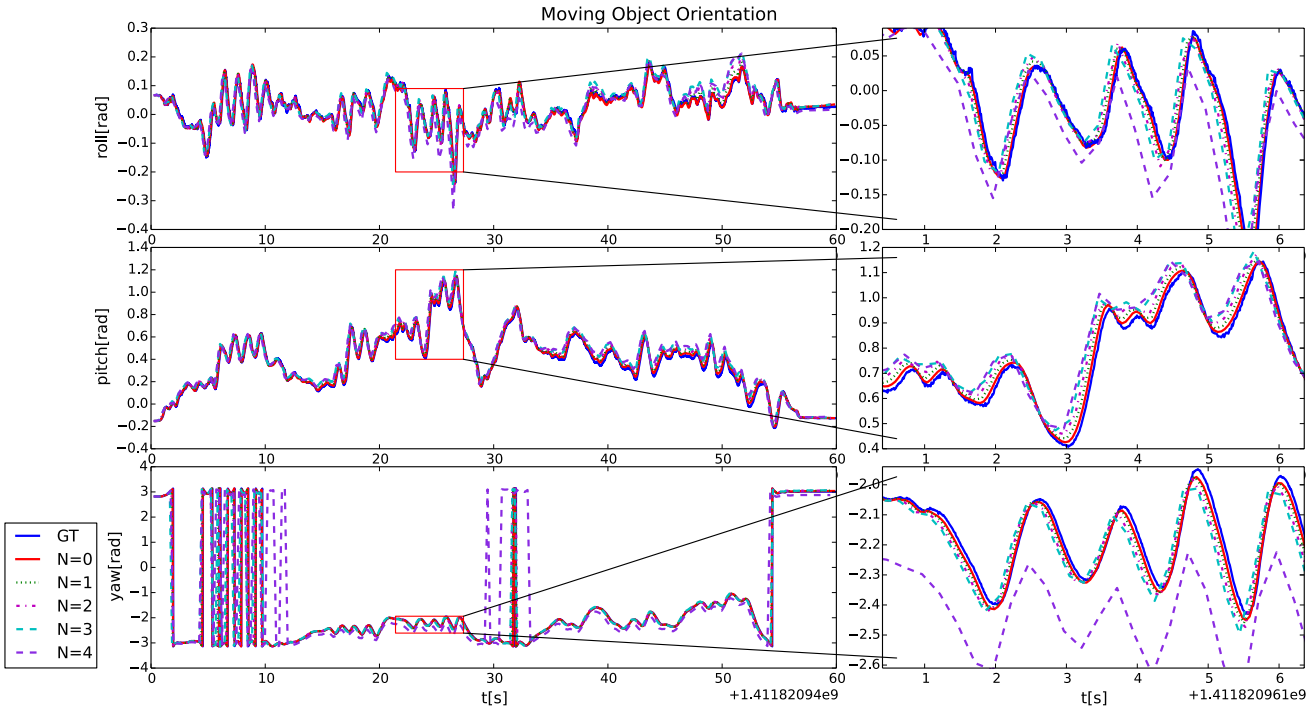


Fig. 23 Moving object orientation plots of the *dataset_1* for different frame discard strategy. Each curve is one strategy. The area within the red boxes in the left images are zoomed in in the right images. N means we process one frame every $N+1$ frames and discard the in-between N frames directly. GT means ground truth. These results show we can still obtain a good estimate until $N=4$ and it is times faster and runs at real-time

work involves developing a sophisticated keyframe selection strategy to make our system run at real-time.

Acknowledgements Special thanks are given to Zheming Liu and Junlin Song for their help in data collection.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Appendix

A. Background

In the following, we wish to derive the partial derivatives of the transformation $\boldsymbol{\pi}(\boldsymbol{T}\cdot\boldsymbol{l})$ with respect to the pose $\boldsymbol{T}$. According to Hauke (2012), this can be calculated using the smooth path $\boldsymbol{T}(t)=\boldsymbol{T}\mathrm{Exp}(\delta\boldsymbol{\xi})$:

$$\begin{aligned}\frac{\partial\boldsymbol{\pi}(\boldsymbol{T}\mathrm{Exp}(\delta\boldsymbol{\xi})\cdot\boldsymbol{l})}{\partial\delta\boldsymbol{\xi}} &= \left.\frac{\partial\boldsymbol{\pi}(\boldsymbol{q})}{\partial\boldsymbol{q}}\right|_{\boldsymbol{q}=\boldsymbol{T}\cdot\boldsymbol{l}} \left.\frac{\partial\boldsymbol{T}\mathrm{Exp}(\delta\boldsymbol{\xi})\cdot\boldsymbol{l}}{\partial\delta\boldsymbol{\xi}}\right|_{\delta\boldsymbol{\xi}=\boldsymbol{0}} \\ &= \boldsymbol{J}_r\boldsymbol{T}\begin{bmatrix}\boldsymbol{I}_{3\times3} & -\boldsymbol{l}^{\wedge}_{1:3}\\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3}\end{bmatrix}\end{aligned} \tag{30}$$

where $\boldsymbol{q}=\boldsymbol{T}\mathrm{Exp}(\delta\boldsymbol{\xi})\cdot\boldsymbol{l}$, $\delta\boldsymbol{\xi}=\begin{bmatrix}\delta\boldsymbol{\rho} & \delta\boldsymbol{\phi}\end{bmatrix}^T$, $\delta\boldsymbol{\xi}\in\mathfrak{se}(3)$, $\delta\boldsymbol{\phi}\in\mathfrak{so}(3)$ and $\delta\boldsymbol{\rho}\in\mathbb{R}^3$. $\boldsymbol{J}_r$ denotes the Jacobian matrix of the pinhole camera model with respect to the 3-dimension landmark point coordinates expressed in the camera frame. We will also use the exponential map property:

$$\mathrm{Exp}(-\delta\boldsymbol{\phi})^T=\mathrm{Exp}(\delta\boldsymbol{\phi}) \tag{31}$$

B. Jacobians

This section provides the Jacobians of the tag reprojection error with respect to the moving object pose $\boldsymbol{T}_{WB}$ and the tag pose $\boldsymbol{T}_{WA}$. The tag reprojection error of the nth corner in the jth tag at image time t_i is

$$\boldsymbol{e}^{i,j,n}_{re}=\boldsymbol{z}^{i,j,n}-\boldsymbol{\pi}\left(\boldsymbol{T}_{CB}\left(\boldsymbol{T}^i_{WB}\right)^{-1}\boldsymbol{T}_{WA_j}\cdot\boldsymbol{l}^n_j\right) \tag{32}$$

1. Jacobian of the tag reprojection error with respect to the moving object pose $\boldsymbol{T}_{WB}$:

The reprojection error with respect to the rotational increment is:

$$\begin{aligned}&\boldsymbol{e}^{i,j,n}_{re}\left(\boldsymbol{R}^i_{WB}\mathrm{Exp}\left(\delta\boldsymbol{\phi}^i_R\right)\right)\\ &=\boldsymbol{z}^{i,j,n}-\boldsymbol{\pi}\left(\boldsymbol{T}^i_{CB}\begin{bmatrix}\left(\boldsymbol{R}^i_{WB}\mathrm{Exp}\left(\delta\boldsymbol{\phi}^i_R\right)\right)^T & -\left(\boldsymbol{R}^i_{WB}\mathrm{Exp}\left(\delta\boldsymbol{\phi}^i_R\right)\right)^T{}_W\boldsymbol{p}^i_{WB}\\ \boldsymbol{0}_{3\times3} & 1\end{bmatrix}\boldsymbol{T}_{WA_j}\cdot\boldsymbol{l}^n_j\right)\\ &=\boldsymbol{z}^{i,j,n}-\boldsymbol{\pi}\left(\boldsymbol{T}^i_{CB}\begin{bmatrix}\mathrm{Exp}\left(-\delta\boldsymbol{\phi}^i_R\right)\boldsymbol{R}^{i\,T}_{WB}\left({}_W\boldsymbol{l}^n_{j\,1:3}-{}_W\boldsymbol{p}^i_{WB}\right)\\ 1\end{bmatrix}\right)\end{aligned} \tag{33}$$

where ${}_W\boldsymbol{l}^n=\begin{bmatrix}{}_W\boldsymbol{l}^n_{1:3} & 1\end{bmatrix}^T=\boldsymbol{T}_{WA_j}\cdot\boldsymbol{l}^n_j$. We can get the Jacobian of the tag reprojection error with respect to the moving object orientation using Eqs. (30) and (31):

$$\frac{\partial\boldsymbol{e}^{i,j,n}_{re}}{\partial\delta\boldsymbol{\phi}^i_R}=-\boldsymbol{J}_{r\,j,n}\boldsymbol{T}^i_{CB}\begin{bmatrix}\left(\boldsymbol{R}^{i\,T}_{WB}\left({}_W\boldsymbol{l}^n_{j\,1:3}-{}_W\boldsymbol{p}^i_{WB}\right)\right)^{\wedge}\\ \boldsymbol{0}_{1\times3}\end{bmatrix} \tag{34}$$

The reprojection error with respect to the translational increment is:

$$\begin{aligned}&\boldsymbol{e}^{i,j,n}_{re}\left({}_W\boldsymbol{p}^i_{WB}+\delta_W\boldsymbol{p}^i_{WB}\right)\\ &\quad=\boldsymbol{z}^{i,j,n}-\boldsymbol{\pi}\left(\boldsymbol{T}^i_{CB}\begin{bmatrix}\boldsymbol{R}^{i\,T}_{WB}\left({}_W\boldsymbol{l}^n_{j\,1:3}-{}_W\boldsymbol{p}^i_{WB}-\delta_W\boldsymbol{p}^i_{WB}\right)\\ 1\end{bmatrix}\right)\end{aligned} \tag{35}$$

and the Jacobian of the tag reprojection error with respect to the moving object translation is:

$$\frac{\partial\boldsymbol{e}^{i,j,n}_{re}}{\partial\delta_W\boldsymbol{p}^i_{WB}}=\boldsymbol{J}_{r\,j,n}\boldsymbol{T}^i_{CB}\begin{bmatrix}\boldsymbol{R}^{i\,T}_{WB}\\ \boldsymbol{0}_{1\times3}\end{bmatrix} \tag{36}$$

2. Jacobian of the tag reprojection error with respect to the tag pose $\boldsymbol{T}_{WA}$: The reprojection error with respect to the $SE(3)$ increment is:

$$\begin{aligned}&\boldsymbol{e}^{i,j,n}_{re}\left(\boldsymbol{T}_{WA_j}\mathrm{Exp}\left(\delta\boldsymbol{\xi}^j_F\right)\right)\\ &\quad=\boldsymbol{z}^{i,j,n}-\boldsymbol{\pi}\left(\boldsymbol{T}_{CB}\left(\boldsymbol{T}^i_{WB}\right)^{-1}\boldsymbol{T}_{WA_j}\mathrm{Exp}\left(\delta\boldsymbol{\xi}^j_F\right)\cdot\boldsymbol{l}^n_j\right)\end{aligned} \tag{37}$$

so we can get the Jacobian of the tag reprojection error with respect to the tag pose using Eq. (30):

$$\frac{\partial\boldsymbol{e}^{i,j,n}_{re}}{\partial\delta\boldsymbol{\xi}}=-\boldsymbol{J}_{r\,j,n}\boldsymbol{T}^i_{CB}\left(\boldsymbol{T}^i_{WB}\right)^{-1}\boldsymbol{T}^j_{WA}\begin{bmatrix}\boldsymbol{I}_{3\times3} & -\boldsymbol{l}^{\wedge}_{j\,1:3}\\ \boldsymbol{0}_{1\times3} & \boldsymbol{0}_{1\times3}\end{bmatrix} \tag{38}$$

References

Botterill, T., Mills, S., & Green, R. (2013). Correcting scale drift by object recognition in single-camera slam. *IEEE Transactions on Cybernetics*, *43*(6), 1767–1780.

Concha, A., Loianno, G., Kumar, V., & Civera, J. (2016). Visual-inertial direct slam. In *IEEE international conference on robotics and automation* (pp. 1331–1338).

Dellaert, F. (2012). *Factor graphs and gtsam: A hands-on introduction*. Atlanta: Georgia Institute of Technology.
Engel, J., Schps, T., & Cremers, D. (2014). Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision (ECCV)* (pp. 834–849).
Engel, J., Koltun, V., & Cremers, D. (2017). Direct sparse odometry. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, *PP*(99), 1–1.
Faessler, M., Mueggler, E., Schwabe, K., & Scaramuzza, D. (2014). A monocular pose estimation system based on infrared leds. In *IEEE international conference on robotics and automation* (pp. 907 – 913).
Fiala, M. (2005). Artag, a fiducial marker system using digital techniques. In *IEEE computer society conference on computer vision and pattern recognition (CVPR)* (Vol. 2, pp. 590–596).
Fiala, M. (2010). Designing highly reliable fiducial markers. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, *32*(7), 1317–24.
Forster, C., Pizzoli, M., & Scaramuzza, D. (2014). Svo: Fast semi-direct monocular visual odometry. In *IEEE international conference on robotics and automation* (pp. 15–22).
Forster, C., Carlone, L., Dellaert, F., & Scaramuzza, D. (2017). On-manifold preintegration for real-time visual-inertial odometry. *IEEE Transactions on Robotics*, *33*(1), 1–21.
Frost, D. P., Khler, O., & Murray, D. W. (2016). Object-aware bundle adjustment for correcting monocular scale drift. In *IEEE international conference on robotics and automation* (pp. 4770–4776).
Furgale, P., Rehder, J., & Siegwart, R. (2014). Unified temporal and spatial calibration for multi-sensor systems. In *IEEE/RSJ international conference on intelligent robots and systems* (pp. 1280–1286).
Gálvez-López, D., Salas, M., Tardós, J. D., & Montiel, J. (2016). Real-time monocular object slam. *Robotics & Autonomous Systems*, *75*(PB), 435–449.
Hauke, S. (2012). *Local accuracy and global consistency for efficient slam*. London: Imperial College London.
Hauke, S., Montiel, J. M. M., & Davison, A. (2010). Scale drift-aware large scale monocular slam. In *Robotics: Science and systems*
Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J. J., & Dellaert, F. (2011). isam2: Incremental smoothing and mapping using the bayes tree. *International Journal of Robotics Research*, *31*(2), 216–235.
Klein, G., & Murray, D. (2007). Parallel tracking and mapping for smallar workspaces. In *IEEE and ACM international symposium on mixed and augmented reality* (pp. 1–10).
Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., & Furgale, P. (2014). Keyframe-based visual-inertial odometry using nonlinear optimization. *International Journal of Robotics Research*, *34*(3), 314–334.
Lim, H., & Lee, Y. S. (2009). Real-time single camera slam using fiducial markers. In *Iccas-sice* (pp. 177–182).
Mourikis, A. I., & Roumeliotis, S. I. (2007). A multi-state constraint kalman filter for vision-aided inertial navigation. In *IEEE international conference on robotics and automation* (pp. 3565–3572).
Mur-Artal, R., Montiel, J. M. M., & Tards, J. D. (2015). Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, *31*(5), 1147–1163.
Neunert, M., Bloesch, M., & Buchli, J. (2016). An open source, fiducial based, visual-inertial motion capture system. *Epigenetics Official Journal of the Dna Methylation Society*, *7*(7), 710–9.
Olson, E. (2011). Apriltag: A robust and flexible visual fiducial system. In *IEEE international conference on robotics and automation* (pp. 3400–3407).
Qiu, K., Zhang, F., & Liu, M. (2015). Visible light communication-based indoor environment modeling and metric-free path planning. In *IEEE international conference on automation science and engineering* (pp. 200–205).
Sementille, A. C., & Rodello, I. (2004). A motion capture system using passive markers. In *Vrcai 2004, ACM siggraph international conference on virtual reality continuum and ITS applications in industry, Nanyang technological university, Singapore* (pp. 440–447).
Usenko, V., Engel, J., Stuckler, J., & Cremers, D. (2016). Direct visual-inertial odometry with stereo cameras. In *IEEE international conference on robotics and automation* (pp. 1885–1892).

Guoping He received the B.E from Harbin Institute of Technology in 2015. He is currently a Master student since 2015 at the Department of Astronautics of Harbin Institute of Technology. His research interests include robot navigation, computer vision, visual (inertial) odometry and SLAM.

Shangkun Zhong received the B.E from Harbin Institute of Technology in 2015. He is currently a Master student since 2015 at the Department of Astronautics of Harbin Institute of Technology. His research interests include vision based navigation and sensor fusion.

Jifeng Guo is a Professor at the Department of Astronautics, Harbin Institute of Technology. He received the Ph.D degrees in Aeronautical and Astronautical Science and Technology from Harbin Institute of Technology in 2007. He is now the deputy director of the Department of Astronautics of Harbin Institute of Technology. His current research interests include space on-orbit servicing technology, intelligent sensing and autonomous planning.