LIC-Fusion: LiDAR-Inertial-Camera Odometry

Xingxing Zuo*, Patrick Geneva††, Woosik Lee†, Yong Liu*, and Guoquan Huang†

***Abstract*— This paper presents a tightly-coupled multi-sensor fusion algorithm of LiDAR-inertial-camera (LIC) odometry, which efficiently combines the IMU measurements and sparse visual and LiDAR features. To endow the proposed LIC-Fusion approach with the plug-and-play property, we also perform online spatial and temporal sensor calibration between all three sensors. The key idea of the proposed approach is to detect and track sparse edge/surf feature points over LiDAR scans and then to fuse their measurements along with the visual feature observations in the efficient MSCKF framework. We perform extensive experiments in both indoor and outdoor environments and validate that the LIC-Fusion outperforms the state-of-the-art visual-inertial odometry (VIO) and LiDAR odometry methods, in terms of localization accuracy as well as robustness to aggressive motions.**

I. Introduction and Related Work

It is essential to be able to accurately track 3D motion for autonomous vehicles and mobile perception systems. One popular solution is the inertial navigation system (INS) aided with a monocular camera, which has recently attracted significant attentions [1–6], in part because of their complimentary sensing modalities of low cost and small size. However, for obvious reasons, cameras are limited by lighting conditions. In contrast, 3D LiDAR sensors can provide more robust and accurate measurements, and are therefore also popular for robot localization and mapping [7–10] but suffer from point cloud sparsity. While they are still expensive as of today, limiting their widespread adoptions, they are expected to have dramatic cost reduction in coming years. In this work, we focus on LiDAR-inertial-camera odometry to offer an efficient and robust 3D motion tracking solution.

Fusing these multi-modal measurements, in particular, from camera and LiDAR, is often addressed within a SLAM framework [11]. For example, Zhang, Kaess, and Singh [12] associated the depth information from LiDAR to the visual features from the camera. As a result, the monocular camera can be considered as RGB-D with the augmented depth from LiDAR. Later, Zhang and Singh [13] developed a general framework for combining visual odometry (VO) and LiDAR odometry (LO), which uses a high-frequency visual odometry to estimate the ego-motion and a lower-rate LiDAR odometry which matches scans to the map in order to refine the VO estimates. Recently, Shin, Park, and Kim [14] used the depth from LiDAR in a direct visual SLAM method, where photometric errors are optimized in an iterative way. Graeter, Wilczynski, and Lauer [11] developed the LIMO algorithm, which also leverages the LiDAR for augmenting depth to visual features by fitting local planes, and was shown to perform well in autonomous driving scenarios.

Zhang and Singh [15] recently developed a laser visual-inertal odometry and mapping system fusing measurements from LiDAR, IMU and camera as we do in this paper. Specifically, they employed a sequential multi-layer processing pipeline, which is composed of three main components: IMU prediction, visual-inertial odometry, and scan matching refinement. IMU measurements are used for prediction; visual-inertial subsystem is used for ego-motion estimation, and a joint cost function of IMU error and visual feature re-projection error is minimized in an iterative way; LiDAR scan matching is performed via iterative closet point (ICP), which further refines the prior pose estimates. Furthermore, the visual-inertial odometry subsystem, and scan matching refinement subsystem will provide feedback to correct velocity and bias of IMU. Clearly, both iterative optimization and iterative closet point are performed in their work, sophisticated pipelines such as parallel processing need to be resorted for saving computational resources. Only the output pose estimation results from former subsystem is fed into the latter subsystem, thus the constraints in former subsystem can not be fully leveraged in the latter subsystem. For example, the feature reprojection errors in visual-inertial subsystem will not be directly minimized in the latter LiDAR scan matching subsystem, thus being loosely coupled.

In this paper, however, we develop a tightly-coupled, single-thread, lidar-inertial-camera (LIC) odometry algorithm with online spatial and temporal calibration between any two sensors, in order to optimally fuse these multi-modal measurements. In particular, in the proposed method, no iterative closet point or iterative optimization step are performed, rendering much computational savings. The main contributions of this work are the following:

- We develop a tightly-coupled LIC odometry (termed LIC-Fusion), which enables to efficiently estimate the 6DOF poses, meanwhile, to perform online spatial and temporal calibration between different sensors.
- The proposed method fuses inertial measurements, sparse visual features, and two different LiDAR features within the efficient multi-state constraint Kalman filter (MSCKF) framework, in which a noise model of the LiDAR feature residual is proposed to better capture the uncertainty of measurements.

*The authors are with the Institute of Cyber-System and Control, Zhejiang University, Hangzhou, China. Email: xingxingzuo@zju.edu.cn, yongliu@iipc.zju.edu.cn

†The authors are with the Department of Mechanical Engineering, University of Delaware, Newark, DE 19716, USA. Email: {ghuang,woosik}@udel.edu

††The author is with the Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716, USA. Email: pgeneva@udel.edu

- We perform extensive experimental validations of the proposed approach on real-world experiments including indoor and outdoor scenarios.

II. The Proposed LIC-Fusion

In this section, we present in detail the proposed LIC-Fusion odometry that tightly fuses LiDAR, inertial, and camera measurements to track 6DOF pose of IMU-affixed frame $\{\mathbf{I}\}$ with respect to a global reference frame $\{\mathbf{G}\}$ based on the efficient MSCKF [1]. The data flow is illustrated in Fig. 1a, showing that IMU measurements including angular velocity and linear acceleration are used for sate propagation, while features from images and LiDAR scans are used for state update.

A. State Vector

The state vector of the proposed method includes the IMU state $\mathbf{x}_I$ at time k, the extrinsics between IMU and camera $\mathbf{x}_{calib\_C}$, the extrinsics between IMU and LiDAR $\mathbf{x}_{calib\_L}$, a sliding window of local IMU clones at the past m image times $\mathbf{x}_C$, and a sliding window of local IMU clones at the past n LiDAR scan times $\mathbf{x}_L$. Note that both the IMU clones for image and LiDAR scan are the past local IMU states, which won't evolve over time and will be updated by the visual feature measurement or LiDAR feature measurement. The total state vector is:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_I^\top & \mathbf{x}_{calib\_C}^\top & \mathbf{x}_{calib\_L}^\top & \mathbf{x}_C^\top & \mathbf{x}_L^\top \end{bmatrix}^\top \tag{1}$$

where

$$\mathbf{x}_I = \begin{bmatrix} {}^{I_k}_{G}\bar{q}^\top & \mathbf{b}_g^\top & {}^G\mathbf{v}_{I_k}^\top & \mathbf{b}_a^\top & {}^G\mathbf{p}_{I_k}^\top \end{bmatrix}^\top \tag{2}$$

$$\mathbf{x}_{calib\_C} = \begin{bmatrix} {}^{C}_{I}\bar{q}^\top & {}^C\mathbf{p}_I^\top & t_{d_C} \end{bmatrix}^\top \tag{3}$$

$$\mathbf{x}_{calib\_L} = \begin{bmatrix} {}^{L}_{I}\bar{q}^\top & {}^L\mathbf{p}_I^\top & t_{d_L} \end{bmatrix}^\top \tag{4}$$

$$\mathbf{x}_C = \begin{bmatrix} {}^{I_{a_1}}_{G}\bar{q}^\top & {}^G\mathbf{p}_{I_{a_1}}^\top & \cdots & {}^{I_{a_m}}_{G}\bar{q}^\top & {}^G\mathbf{p}_{I_{a_m}}^\top \end{bmatrix}^\top \tag{5}$$

$$\mathbf{x}_L = \begin{bmatrix} {}^{I_{b_1}}_{G}\bar{q}^\top & {}^G\mathbf{p}_{I_{b_1}}^\top & \cdots & {}^{I_{b_n}}_{G}\bar{q}^\top & {}^G\mathbf{p}_{I_{b_n}}^\top \end{bmatrix}^\top \tag{6}$$

where ${}^{I_k}_{G}\bar{q}$ is the JPL quaternion [16] relating to the rotation matrix ${}^{I_k}_{G}\mathbf{R} \in \mathbb{R}_{3\times3}$ from global reference frame $\{G\}$ to local frame $\{I_k\}$ of IMU at time stamp t_k, ${}^G\mathbf{v}_{I_k}$ and ${}^G\mathbf{p}_{I_k}$ represent the IMU velocity and position in the global frame, respectively. $\mathbf{b}_g$ and $\mathbf{b}_a$ are the gyroscope and accelerometer biases. ${}^{C}_{I}\bar{q}$ and ${}^C\mathbf{p}_I$ represent the rigid-body transformation between the IMU frame $\{C\}$ and the IMU frame $\{I\}$, analogously ${}^{L}_{I}\bar{q}$ and ${}^L\mathbf{p}_I$ are the transformation between camera frame and LiDAR frame. Considering the coupled system composed by these three sensors, timestamps are typically obtained for each image, IMU sample, and LiDAR scan, and the unknown time offsets between different sensors generally exist due to different sensor's latency, missed data, clock skew, and data transmission dealy [17].

Fig. 1b shows one example of the time offset which arises because of data transmission delay of camera images or LiDAR scans. In the figure, the upper plot denotes the physical sampling time instants. The lower plot represents the labeled timestamp of measurement by computer. Once a measurement is received by the computer, it will be labeled with a timestamp. However the labeled timestamp of Camera/LiDAR deviates from the true physical sampling time instant by t_{dC} or t_{dL}. In order to utilize the heterogeneous measurements obtained from three different sensors, the time offsets must be known. Therefore, we estimate the unknown time offset t_{dC} between the camera and IMU, t_{dL} between the LiDAR and IMU. We use the IMU time as time reference, and align the labeled timestamps of image and LiDAR scan, t_C, t_L with the IMU timestamp, t_I, by $t_I = t_C + t_{dC} = t_L + t_{dL}$. It should be noted the t_{dC} or t_{dL} here may have a positive or negative value. For the case illustrated in Fig 1b, it has a negative value.

B. IMU Propagation

The IMU measurements used for propagation in our EKF estimator, which predicts the states at given time. The continuous-time dynamics of the IMU state $\mathbf{x}_I$ can be described as [16]:

$$\begin{aligned} {}^{I_k}_{G}\dot{\bar{q}}(t) &= \frac{1}{2}\mathbf{\Omega}\left({}^{I_k}\boldsymbol{\omega}(t)\right){}^{I_k}_{G}\bar{q}(t) \\ {}^G\dot{\mathbf{p}}_{I_k}(t) &= {}^G\mathbf{v}_{I_k}(t) \\ {}^G\dot{\mathbf{v}}_{I_k}(t) &= {}^{I_k}_{G}\mathbf{R}(t)^\top {}^{I_k}\mathbf{a}(t) + {}^G\mathbf{g} \\ \dot{\mathbf{b}}_g(t) &= \mathbf{n}_{wg}, \quad \dot{\mathbf{b}}_a(t) = \mathbf{n}_{wa}(t) \end{aligned} \tag{7}$$

In the above expression, we define $\mathbf{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} -\lfloor\boldsymbol{\omega}\rfloor & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & \mathbf{0} \end{bmatrix}$, $\lfloor\cdot\rfloor$ as a conversion function to skew symmetric matrix, ${}^{I_k}\boldsymbol{\omega}$ and ${}^{I_k}\mathbf{a}$ represent the angular velocity and linear acceleration in local IMU frame, and ${}^G\mathbf{g}$ denotes the gravitational acceleration in global frame.

The gyroscope and accelerometer biases $\mathbf{b}_g$ and $\mathbf{b}_a$ are modeled as random walk, which are driven by the white Gaussian noises $\mathbf{n}_{wg}$ and $\mathbf{n}_{wa}$, respectively. The states described in preceding section are propagated over time by the IMU measurements $\boldsymbol{\omega}_m$ and $\mathbf{a}_m$. Actually, among all the states, only the IMU states $\mathbf{x}_I$ will evolve over this propagation step. We linearize the propagation step at current estimates, and then propagate the states and corresponding covariance. Since the state covariance is correlated, and it will evolve over this propagation [1].

C. State Augmentation

Every time the system receives a new image or LiDAR scan, we will propagate the IMU state to this time, clone this IMU state and append it as a new state to existing state vector. In order to calibrate the time offsets between different sensors, we will propagate up to IMU time $\hat{t}_{Ik}$, which is supposed to be the physical IMU time when image or LiDAR scan comes. Such as a new LiDAR scan is received with timestamp t_{Lk}, we will propagate up to $\hat{t}_{Ik} = t_{Lk} + \hat{t}_{dL}$, and argument the state vector to include this new cloned state estimate,

$$\hat{\mathbf{x}}_{Lk}(\hat{t}_{Ik}) = \begin{bmatrix} {}^{I_k}\hat{\bar{q}}(\hat{t}_{Ik})^\top & {}^G\hat{\mathbf{p}}_{I_k}(\hat{t}_{Ik})^\top \end{bmatrix} \tag{8}$$

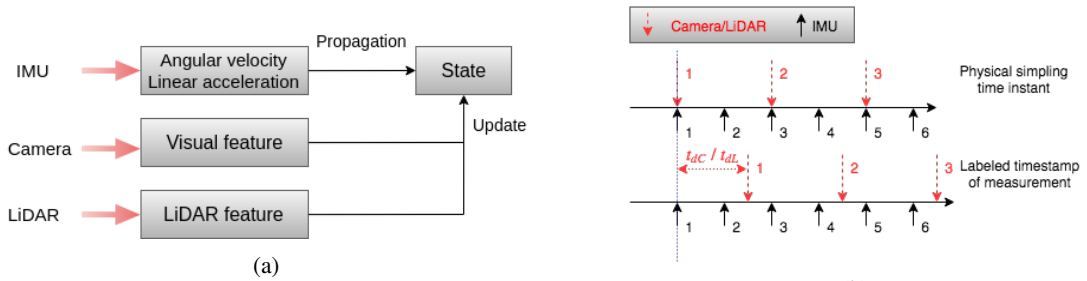

Fig. 1: Data flow of *LIC-fusion*, measurements from three different sensors are utilized in the EKF based estimator (a). Illustration of the time offset between camera/LiDAR and IMU (b).

As the covariance are correlated, the state covariance will evolve over this augmentation:

$$\mathbf{P}(\hat{t}_{Ik}) \leftarrow \begin{bmatrix} \mathbf{P}(\hat{t}_{Ik}) & \mathbf{P}(\hat{t}_{Ik})\mathbf{J}_{Ik}(\hat{t}_{Ik})^\top \\ \mathbf{J}_{Ik}(\hat{t}_{Ik})\mathbf{P}(\hat{t}_{Ik}) & \mathbf{J}_{Ik}(\hat{t}_{Ik})\mathbf{P}(\hat{t}_{Ik})\mathbf{J}_{Ik}(\hat{t}_{Ik})^\top \end{bmatrix} \tag{9}$$

where $\mathbf{J}_{Ik}(\hat{t}_{Ik})$ is the Jacobian of the new cloned state with respect to the state already existing in the state vector Eq. 1:

$$\mathbf{J}_{Ik}(\hat{t}_{Ik}) = \frac{\partial \delta \mathbf{x}_{Lk}(\hat{t}_{Ik})}{\partial \delta \mathbf{x}} = \begin{bmatrix} \mathbf{J}_I & \mathbf{J}_{calib\_C} & \mathbf{J}_{calib\_L} & \mathbf{J}_C & \mathbf{J}_L \end{bmatrix} \tag{10}$$

In the above expression, $\mathbf{J}_I$ denotes the Jacobian with respect to the IMU state $\mathbf{x}_I$,

$$\mathbf{J}_I = \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times9} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times9} & \mathbf{I}_{3\times3} \end{bmatrix} \tag{11}$$

$\mathbf{J}_{calib\_L}$ is the Jacobian with respect to the extrinsics (including time offset) between IMU and LiDAR,

$$\mathbf{J}_{calib\_L} = \begin{bmatrix} \mathbf{0}_{6\times6} & \mathbf{J}_{t_{dL}} \end{bmatrix}, \ \mathbf{J}_{t_{dL}} = \begin{bmatrix} {}^{Ik}\hat{\boldsymbol{\omega}}^\top & {}^{G}\hat{\mathbf{v}}_{Ik}^\top \end{bmatrix}^\top \tag{12}$$

where ${}^{Ik}\hat{\boldsymbol{\omega}}^\top$ denotes the local angular velocity of IMU at time $\hat{t}_{Ik}$, and ${}^{G}\hat{\mathbf{v}}_{Ik}^\top$ is the global linear velocity of IMU at time $\hat{t}_{Ik}$. Since $\hat{t}_{Ik}$ is not exactly aligned with one IMU measurement timestamp at most of time, we will interpolate the IMU measurements for computing ${}^{Ik}\hat{\boldsymbol{\omega}}^\top$ and ${}^{G}\hat{\mathbf{v}}_{Ik}^\top$. $\mathbf{J}_{calib\_C}$, $\mathbf{J}_C$, $\mathbf{J}_L$ are the Jacobian with respect to extrinsics between IMU and camera, clones at camera time, clones at LiDAR time, respectively. All of them should be zero matrix. The dependence of the new cloned IMU state on t_{dL} is modeled via the Jacobian $\mathbf{J}_{t_{dL}}$, and t_{dL} will be updated in the EKF update step by measurements. Likewise, the time offset between IMU and camera can be modeled and estimated when new image is coming. In this simple fashion, we can explicitly perform online temporal calibration by jointly estimating the time offsets and other states in the state vector.

D. Measurement Models

1) LiDAR Feature Measurement: As there are lots of points in an incoming LiDAR scan, we cannot use all points in the cloud and remain computational efficiency. Therefore, we only utilize some edge and surf features as in [7]. For the LiDAR scans which are related to a cloned IMU state in the sliding window, we will project their features into the oldest scan in the sliding window to find the closest neighbor features of the same class using a KD-tree for fast indexing [18]. For an example, we project one feature point ${}^{L_{l+1}}\mathbf{p}_{fi}$ in the second oldest LiDAR scan L_{l+1} to the oldest LiDAR scan L_l, the projected point is denoted as ${}^{L_l}\mathbf{p}_{fi}$.

$${}^{L_l}\mathbf{p}_{fi} = {}^{L_l}_{L_{l+1}}\mathbf{R}\,{}^{L_{l+1}}\mathbf{p}_{fi} + {}^{L_l}\mathbf{p}_{L_{l+1}} \tag{13}$$

where ${}^{L_l}_{L_{l+1}}\mathbf{R}$, ${}^{L_l}\mathbf{p}_{L_{l+1}}$, are the relative rotation and translation between two LiDAR frames, which can be computed from the states in the state vector:

$$\begin{aligned} {}^{L_l}_{L_{l+1}}\mathbf{R} &= {}^{L}_{I}\mathbf{R}\,{}^{I_l}_{G}\mathbf{R}\left({}^{L}_{I}\mathbf{R}\,{}^{I_{l+1}}_{G}\mathbf{R}\right)^\top \\ {}^{L_l}\mathbf{p}_{L_{l+1}} &= {}^{L}_{I}\mathbf{R}\,{}^{I_l}_{G}\mathbf{R}\left({}^{G}\mathbf{p}_{I_{l+1}} - {}^{G}\mathbf{p}_{I_l} + {}^{I_{l+1}}_{G}\mathbf{R}^\top {}^{I}\mathbf{p}_L\right) + {}^{L}\mathbf{p}_I \\ {}^{I}\mathbf{p}_L &= -{}^{L}_{I}\mathbf{R}^\top {}^{L}\mathbf{p}_I \end{aligned} \tag{14}$$

For the projected edge features ${}^{L_l}\mathbf{p}_{fi}$, we will find its two corresponding edge features in the oldest scan, ${}^{L_l}\mathbf{p}_{fj}$, ${}^{L_l}\mathbf{p}_{fk}$, which are supposed to be sampled from the same physical edge as ${}^{L_l}\mathbf{p}_{fi}$. ${}^{L_l}\mathbf{p}_{fj}$ is the closet edge feature in the oldest LiDAR scan which is composed with many rings, here we assume that ${}^{L_l}\mathbf{p}_{fj}$ is on r_{th} ring, then we will find the other closet edge feature ${}^{L_l}\mathbf{p}_{fk}$ on neighboring ring $r-1$ or $r+1$. The measurement residual of edge feature ${}^{L_{l+1}}\mathbf{p}_{fi}$ is the distance between its projected feature point ${}^{L_l}\mathbf{p}_{fi}$ and the straight line represented by two points ${}^{L_l}\mathbf{p}_{fj}$, ${}^{L_l}\mathbf{p}_{fk}$ [7]:

$$r({}^{L_{l+1}}\mathbf{p}_{fi}) = \frac{\left\| \left({}^{L_l}\mathbf{p}_{fi} - {}^{L_l}\mathbf{p}_{fj}\right) \times \left({}^{L_l}\mathbf{p}_{fi} - {}^{L_l}\mathbf{p}_{fk}\right) \right\|_2}{\left\| {}^{L_l}\mathbf{p}_{fj} - {}^{L_l}\mathbf{p}_{fk} \right\|_2} \tag{15}$$

where $\|\cdot\|_2$ represents the 2-norm of a matrix, and $\times$ denotes the cross product of two vector. We linearize the above distance measurement at current estimate by:

$$\begin{aligned} r({}^{L_{l+1}}\mathbf{p}_{fi}) &= h(\mathbf{x}) + n_r \\ &= h(\hat{\mathbf{x}}) + \mathbf{H}_{\mathbf{x}}\big|_{\mathbf{x}=\hat{\mathbf{x}}}\,\delta\mathbf{x} + n_r \end{aligned} \tag{16}$$

where $\mathbf{H}_{\mathbf{x}}$ is the Jacobian of the distance with respect to the states in the state vector. n_r is modeled as white Gaussian with covariance of C_r. The non-zero elements in $\mathbf{H}_{\mathbf{x}}$ are only the items related to the cloned poses ${}^{I_l}_{G}\bar{q}$, ${}^{G}\mathbf{p}_{I_l}$, ${}^{I_{l+1}}_{G}\bar{q}$, ${}^{G}\mathbf{p}_{I_{l+1}}$

and the rigid transformation between IMU and LiDAR ${}^{L}_{I}\bar{q}, {}^{L}\mathbf{p}_{I}$. Thus we have:

$$\mathbf{H_x} = \frac{\partial \delta r({}^{L_{l+1}}\mathbf{p}_{fi})}{\partial {}^{L_l}\delta \mathbf{p}_{fi}} \frac{\partial {}^{L_l}\delta \mathbf{p}_{fi}}{\partial \delta \mathbf{x}} \tag{17}$$

the non-zero elements in $\frac{\partial {}^{L_l}\delta \mathbf{p}_{fi}}{\partial \delta \mathbf{x}}$ are:

$$\begin{aligned}
\frac{\partial {}^{L_l}\delta \mathbf{p}_{fi}}{\partial {}^{I_l}_{G}\delta \boldsymbol{\theta}} &= {}^{L}_{I}\mathbf{R} \lfloor {}^{I_l}_{G}\mathbf{R} {}^{I_{l+1}}_{G}\mathbf{R}^{\top} {}^{L}_{I}\mathbf{R} {}^{L_{l+1}}\mathbf{p}_{fi} \rfloor \\
&\quad + {}^{L}_{I}\mathbf{R} \lfloor {}^{I_l}_{G}\mathbf{R} \left({}^{G}\mathbf{p}_{I_{l+1}} - {}^{G}\mathbf{p}_{I_l} + {}^{I_{l+1}}_{G}\mathbf{R}^{\top} {}^{I}\mathbf{p}_{L} \right) \rfloor \\
\frac{\partial {}^{L_l}\delta \mathbf{p}_{fi}}{\partial {}^{G}\delta \mathbf{p}_{I_l}} &= -{}^{L}_{I}\mathbf{R} {}^{I_l}_{G}\mathbf{R} \\
\frac{\partial {}^{L_l}\delta \mathbf{p}_{fi}}{\partial {}^{I_{l+1}}_{G}\delta \boldsymbol{\theta}} &= -{}^{L}_{I}\mathbf{R} {}^{I_l}_{G}\mathbf{R} {}^{I_{l+1}}_{G}\mathbf{R}^{\top} \lfloor {}^{L}_{I}\mathbf{R}^{\top} {}^{L_{l+1}}\mathbf{p}_{fi} + {}^{I}\mathbf{p}_{L} \rfloor \\
\frac{\partial {}^{L_l}\delta \mathbf{p}_{fi}}{\partial {}^{G}\delta \mathbf{p}_{I_{l+1}}} &= {}^{L}_{I}\mathbf{R} {}^{I_l}_{G}\mathbf{R} \\
\frac{\partial {}^{L_l}\delta \mathbf{p}_{fi}}{\partial {}^{L}_{I}\delta \boldsymbol{\theta}} &= \lfloor {}^{L_l}_{L_{l+1}}\mathbf{R} \left({}^{L_{l+1}}\mathbf{p}_{fi} - {}^{L}\mathbf{p}_{I} \right) \rfloor \\
&\quad - {}^{L_l}_{L_{l+1}}\mathbf{R} \lfloor {}^{L_{l+1}}\mathbf{p}_{fi} - {}^{L}\mathbf{p}_{I} \rfloor \\
\frac{\partial {}^{L_l}\delta \mathbf{p}_{fi}}{\partial {}^{L}\delta \mathbf{p}_{I}} &= -{}^{L_l}_{L_{l+1}}\mathbf{R} + \mathbf{I}_{3\times 3}
\end{aligned} \tag{18}$$

In order to perform EKF update, we need to know the explicit covariance C_r of the distance measurement. As this measurement is not directly obtained from sensors, we propagate the covariance of raw measurements (point) in LiDAR scan to C_r at the current estimate state. Assuming the covariance of point ${}^{L_{l+1}}\mathbf{p}_{fi}, {}^{L_l}\mathbf{p}_{fj}, {}^{L_l}\mathbf{p}_{fk}$ are $\mathbf{C}_i, \mathbf{C}_j, \mathbf{C}_k$ respectively, C_r can be computed as:

$$\begin{aligned}
C_r &= \sum_{x=i,j,k} \mathbf{J}_x \mathbf{C}_x \mathbf{J}_x^{\top}, \quad \mathbf{J}_i = \frac{\partial \delta r({}^{L_{l+1}}\mathbf{p}_{fi})}{\partial {}^{L_l}\delta \mathbf{p}_{fi}} {}^{L_l}_{L_{l+1}}\mathbf{R} \\
\mathbf{J}_j &= \frac{\partial \delta r({}^{L_{l+1}}\mathbf{p}_{fi})}{\partial {}^{L_l}\delta \mathbf{p}_{fj}}, \quad \mathbf{J}_k = \frac{\partial \delta r({}^{L_{l+1}}\mathbf{p}_{fi})}{\partial {}^{L_l}\delta \mathbf{p}_{fk}}
\end{aligned} \tag{19}$$

Due to limit space, the Jacobian of $r({}^{L_{l+1}}\mathbf{p}_{fi})$ with respect to edge features $\frac{\partial \delta r({}^{L_{l+1}}\mathbf{p}_{fi})}{\partial {}^{L_l}\delta \mathbf{p}_{fi}}, \frac{\partial \delta r({}^{L_{l+1}}\mathbf{p}_{fi})}{\partial {}^{L_l}\delta \mathbf{p}_{fj}}, \frac{\partial \delta r({}^{L_{l+1}}\mathbf{p}_{fi})}{\partial {}^{L_l}\delta \mathbf{p}_{fk}}$ are omitted here.

With the covariances of states and measurements, we compute the Mahalanobis distance as:

$$r_m = r({}^{L_{l+1}}\mathbf{p}_{fi})^{\top} \left(\mathbf{H_x} \mathbf{P}(\hat{t}_{Ik}) \mathbf{H_x}^{\top} + C_r \right)^{-1} r({}^{L_{l+1}}\mathbf{p}_{fi})$$

where r_m should subject to $\boldsymbol{\chi}^2$ distribution. If r_m is smaller than a threshold, measurement $r({}^{L_{l+1}}\mathbf{p}_{fi})$ will be used for EKF update.

Similarly, for the projected surf features ${}^{L_l}\mathbf{p}_{fi}$, we will find its corresponding three surf features, ${}^{L_l}\mathbf{p}_{fj}, {}^{L_l}\mathbf{p}_{fk}, {}^{L_l}\mathbf{p}_{fl}$, which are supposed to be sampled on the same physical plane as ${}^{L_l}\mathbf{p}_{fi}$. The measurement residual of surf feature ${}^{L_{l+1}}\mathbf{p}_{fi}$ will be the distance between its projected feature point ${}^{L_l}\mathbf{p}_{fi}$ and the plane represented by ${}^{L_l}\mathbf{p}_{fj}, {}^{L_l}\mathbf{p}_{fk}, {}^{L_l}\mathbf{p}_{fl}$. The covariance propagation to the distance measurement, linearization operation and Mahalanobis distance test are akin to the edge feature elaborated here.

2) Visual Feature Measurement: Once a new image is received, we will augment the state with a cloned IMU state, and extract FAST feature from the image which will then be tracked them across sequential images by KLT optical flow. Once visual feature measurements lost or reach the size of the sliding window, we will use these measurements and the related cloned poses to initialize the visual feature in 3D space by triangulation [1]. The residual of visual feature measurement is the reprojection error. For visual measurements $\mathbf{z}_i$ of a 3D visual feature ${}^{G}\mathbf{p}_{fi}$ the linearized residual will be:

$$\begin{aligned}
\mathbf{r}(\mathbf{z}_i) &= h(\mathbf{x}, {}^{G}\mathbf{p}_{fi}) + n_r \\
&= h(\hat{\mathbf{x}}, {}^{G}\hat{\mathbf{p}}_{fi}) + \mathbf{H_x}\Big|_{\mathbf{x}=\hat{\mathbf{x}}} \delta \mathbf{x} \\
&\quad + \mathbf{H_f}|_{{}^{G}\mathbf{p}_{fi}={}^{G}\hat{\mathbf{p}}_{fi}} {}^{G}\delta \mathbf{p}_{fi} + \mathbf{n}_r
\end{aligned} \tag{20}$$

where $\mathbf{H}_f$ is the Jacobian of visual feature measurement with respect to the 3D feature ${}^{G}\mathbf{p}_{fi}$. Since our measurements are a function of ${}^{G}\hat{\mathbf{p}}_{fi}$, see Eq. 20, we leverage the null space operation [1] to remove its dependency. After the null space operation, we have:

$$\mathbf{r}_o(\mathbf{z}_i) = \mathbf{H_{xo}} \delta \mathbf{x} + \mathbf{n}_{ro} \tag{21}$$

It should be noted that the Jacobian with respect to the rigid transformation between IMU and camera ${}^{C}_{I}\bar{q}, {}^{C}\mathbf{p}_{I}$ is non-zero, which means the transformation between IMU and camera can be calibrated online.

E. EKF update

After linearizing the LiDAR feature and visual feature measurements at current estimate, we can perform MSCKF update. By stacking all measurement residuals (which can origin from LiDAR feature or visual feature) in one residual vector, we have:

$$\mathbf{r} = \mathbf{H_x} \delta \mathbf{x} + \mathbf{n} \tag{22}$$

where r and n are vectors with block elements of residual and noise in Eq. 16 or Eq. 21. We treat all the measurements as statistically independent, thus the noise vector $\mathbf{n}$ are uncorrelated. Since number of measurements are much larger than the states, we employ Givens rotation [19] to perform the measurement compression for computational efficiency, which can also be conducted by the QR decomposition. After the measurement compression, we obtain:

$$\mathbf{r}_c = \mathbf{T}_H \delta \mathbf{x} + \mathbf{n}_c \tag{23}$$

The rows of compressed Jacobain $\mathbf{T}_H$ should equals to the dimension of the related state vector $\mathbf{x}$.

There are some degenerate scenarios for visual features such as texture-less environment, which we hope to leverage the LiDAR's ability during these outages. However, in the cases where the states are not well-constrained by the obtained measurements, we compute the sum of the row elements in $\mathbf{T}_H$, such as for the i_{th} row: $s_i = \mathrm{sum}(\mathbf{T}_{\mathrm{H}}[\mathrm{i}, :])$ If s_i is smaller than a given threshold, which means this direction (dimension) of the state vector is degenerate, we

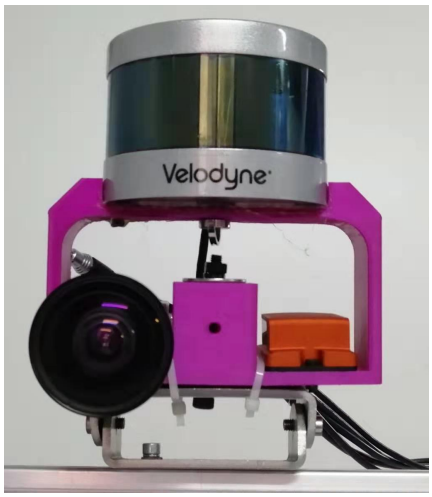

Fig. 2: The self-assembled LiDAR-inertial camera rig with camera, IMU, and sixteen beam LiDAR.

will remove the i_{th} row elements from $\mathbf{T}_H$ for robustness and accuracy of the coupled system. After removing the i_{th} row elements in $\mathbf{T}_H$, the corresponding direction of the state vector will not be updated. This cut-off operation in filter is similar with the solution remapping operation in the optimization-based method [20]. The solution remapping operation determines and separates degenerate directions in the state space, and only partially update the state in well-conditioned directions. By the cut-off operation described, the coupled system should not be corrupted by noise in the degenerate scenarios for LiDAR measurements or camera measurements.

III. Experimental Results

To test the performance of the proposed algorithms, several experiments were performed both in outdoors and indoors environments. In this section, the localization performance and computation time test are presented. Our system is composed of IMU, camera, and sixteen channel LiDAR. The sensor rig is shown in Fig. 2, which uses an Xsens MTi-300 AHRS IMU, Velodyne VLP-16 LiDAR, and the global-shutter pointgrey camera. The extrinsic of the system are manually measured and are refined online, with initial guesses of the time offsets between the sensors set to zero.

A. Outdoor Tests

We evaluated the system on a collected dataset from a custom built mobile robot platform. The robot was equipped with a RTK GPS which is used as the groundtruth for comparison of the different odometry methods. We compare against and implementation of the standard MSCKF [1], and LOAM LiDAR odometry [7]. We note that we compare against the global output of the LOAM algorithm that leverages indirect loop-closure information through cloud registrations to its constructed global map.

Fig. 3 shows outdoor test results of LIC, MSCKF and LOAM. The length of the trajectory is around 800 meters and recorded for 4 minutes. Around 170 seconds into the dataset

TABLE I: Outdoor Experimental RMSE Averages

Datasets		LIC	MSCKF	LOAM
Outdoor	Average Error(m)	4.06	10.75	23.08
	1 sigma(m)	3.42	3.56	2.63

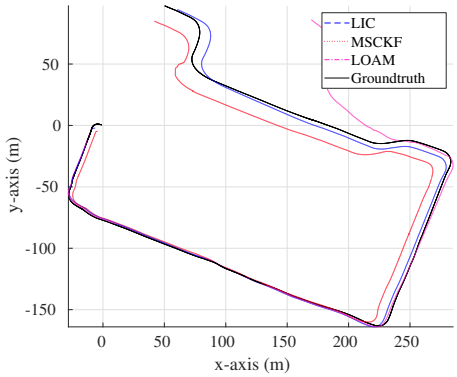

Fig. 3: Top view trajectories of real dataset showing the LIC (blue), MSCKF (red), LOAM (pink) and RTK GPS groundtruth (black)

LOAM starts diverging while both LIC and MSCKF showed stable localization. Each algorithm was run six different times to account for randomness in their different RANSAC methods and provide a representative evaluation of expected typical performance. The average absolute trajectory error (ATE) of each method is presented in Fig.4, in which the trajectories where aligned to the RTK groundtruth using the "best fit" transform that minimized the overall trajectory error. The proposed LIC showed a 2.5 decrease in the averagee error as compared to the standard MSCKF, and 5 decreased when compared to LOAM. We also saw that the drift of LIC grows much slower over time as compared to the other two methods and maintains the smallest error for most of the trajectory. The average error and 1 sigma of the test results are in Table I. These results show that the proposed system is able to localize with high accuracy and the fusion of different sensing modalities (that being camera, inertial, and LiDAR) allows for an increase of robustness to when single sensor odometry algorithms fail.

B. Indoor Tests

The system was then tested on a series of indoor datasets in various lighting conditions and motion profiles. Since groundtruth was not avalible indoors, we returned the sensor

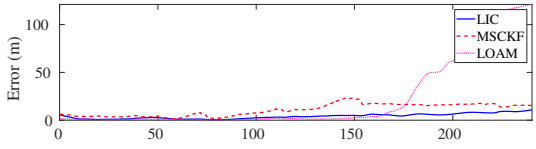

Fig. 4: RMS errors of the LIC (blue), MSCKF (red) and LOAM (pink) over the duration of the trajectory.

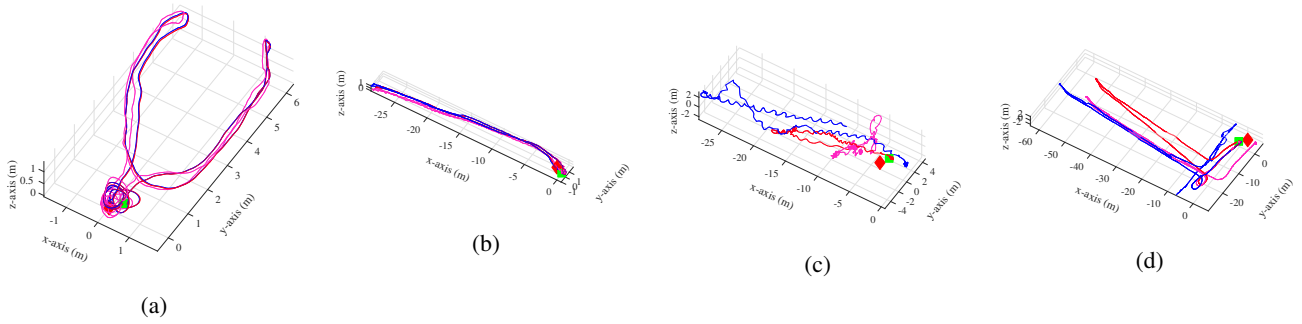

Fig. 5: Isometric views of the indoor datasets, sub-captions correspond to the respective dataset (a), (b), (c), and (d).

TABLE II: Indoor Experimental RMSE Averages

	MSCKF	LIC-fusion	LOAM [7]
Indoor-A (39m)	0.99	0.9776	0.66
Indoor-B (86m)	1.55	1.0369	0.46
Indoor-C (55m)	49.94	1.5454	2.44
Indoor-D (189m)	46.03	3.6782	5.99

platform to the inital location and evaluate the start-end error. Table II, summarizes the results, and shows that the proposed LIC is able to localize with high accuracy. The indoor C dataset is interesting since it has high angular velocities and linear accelerations with high levels of motion blur. Our system is able to localize in this, while the other methods have large amounts of errors.

IV. Conclusions and Future Work

In this paper, we have developed a tightly coupled multi-sensor fusion algorithm for LiDAR-inertial-camera odoemtry (i.e., LIC-Fusion) along with online spatial and temporal calibration of the three sensors. The proposed approach detects and tracks sparse edge/surfl feature points over LiDAR scans and then fuses these measurements along with the visual features extracted from monocular images in the efficient MSCKF framework. As a result, by taking advantages of different sensing modalities, the proposed LIC-Fusion odometry is able to provide accurate and robust 6DOF motion tracking in 3D in different environments and under different motions. In the future, we will investigate how to efficiently integrate loop closure constraints obtained from both LiDAR and camera into the LIC-Fusion in order to bound navigation errors.

References

[1] A. I. Mourikis and S. I. Roumeliotis. "A multi-state constraint Kalman filter for vision-aided inertial navigation". In: *Proceedings of the IEEE International Conference on Robotics and Automation*. Rome, Italy, 2007, pp. 3565–3572.

[2] T. Qin, P. Li, and S. Shen. "Vins-mono: A robust and versatile monocular visual-inertial state estimator". In: *IEEE Transactions on Robotics* 34.4 (2018), pp. 1004–1020.

[3] R. Mur-Artal and J. D. Tardós. "Visual-inertial monocular SLAM with map reuse". In: *IEEE Robotics and Automation Letters* 2.2 (2017), pp. 796–803.

[4] G. Huang, K. Eckenhoff, and J. Leonard. "Optimal-State-Constraint EKF for Visual-Inertial Navigation". In: *Proc. of the International Symposium on Robotics Research*. (to appear). Sestri Levante, Italy, 2015.

[5] S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart. "Keyframe-based visual-inertial slam using nonlinear optimization". In: *Proceedings of Robotis Science and Systems (RSS) 2013* (2013).

[6] Z. Huai and G. Huang. "Robocentric Visual-Inertial Odometry". In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*. Madrid, Spain, 2018.

[7] J. Zhang and S. Singh. "LOAM: Lidar Odometry and Mapping in Real-time." In: *Robotics: Science and Systems*. Vol. 2. 2014, p. 9.

[8] C. Park, S. Kim, P. Moghadam, C. Fookes, and S. Sridharan. "Probabilistic surfel fusion for dense lidar mapping". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2418–2426.

[9] T. Shan and B. Englot. "LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 4758–4765.

[10] J. Behley and C. Stachniss. "Efficient surfel-based SLAM using 3D laser range data in urban environments". In: *Robotics: Science and Systems (RSS)*. 2018.

[11] J. Graeter, A. Wilczynski, and M. Lauer. "LIMO: Lidar-Monocular Visual Odometry". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2018, pp. 7872–7879.

[12] J. Zhang, M. Kaess, and S. Singh. "Real-time depth enhanced monocular odometry". In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2014, pp. 4973–4980.

[13] J. Zhang and S. Singh. "Visual-lidar odometry and mapping: Low-drift, robust, and fast". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 2174–2181.

[14] Y.-S. Shin, Y. S. Park, and A. Kim. "Direct Visual SLAM using Sparse Depth for Camera-LiDAR System". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 1–8.

[15] J. Zhang and S. Singh. "Laser–visual–inertial odometry and mapping with high robustness and low drift". In: *Journal of Field Robotics* 35.8 (2018), pp. 1242–1264.

[16] N. Trawny and S. I. Roumeliotis. *Indirect Kalman Filter for 3D Attitude Estimation*. Tech. rep. University of Minnesota, Dept. of Comp. Sci. & Eng., Mar. 2005.

[17] M. Li and A. I. Mourikis. "Online temporal calibration for camera–IMU systems: Theory and algorithms". In: *The International Journal of Robotics Research* 33.7 (2014), pp. 947–964.

[18] M. De Berg, M. Van Kreveld, M. Overmars, and O. Schwarzkopf. "Computational geometry". In: *Computational geometry*. Springer, 1997, pp. 1–17.

[19] G. H. Golub and C. F. Van Loan. *Matrix computations*. Vol. 3. JHU press, 2012.

[20] J. Zhang, M. Kaess, and S. Singh. "On degeneracy of optimization-based state estimation problems". In: *2016 IEEE International Conference on Robotics and Automation*. IEEE. 2016, pp. 809–816.