PI-VIO: Robust and Efficient Stereo Visual Inertial Odometry using Points and Lines

Feng Zheng, Grace Tsai, Zhe Zhang, Shaoshan Liu, Chen-Chi Chu, and Hongbing Hu

***Abstract*— In this paper, we present the PerceptIn Visual Inertial Odometry (PI-VIO), a tightly-coupled filtering-based stereo VIO system using both points and lines. Line features help improve system robustness in challenging scenarios when point features cannot be reliably detected or tracked, *e.g.* low-texture environment or lighting change. In addition, we propose a new lightweight filtering-based loop closing technique to reduce accumulated drift without global bundle adjustment. We formulate loop closure as EKF updates to optimally *relocate* the current sliding window maintained by the filter to past keyframes. We also present the PerceptIn Ironsides dataset, a new visual-inertial dataset, featuring high-quality synchronized stereo camera and IMU data from the Ironsides sensor [3] with various motion types and textures and millimeter-accuracy groundtruth. To validate the performance of the proposed system, we conduct extensive comparison with state-of-the-art approaches (OKVIS, VINS-MONO and S-MSCKF) using both the public EuRoC dataset and the PerceptIn Ironsides dataset.**

I. INTRODUCTION

Motion tracking is the cornerstone for a wide range of applications, such as robotics, self-driving, AR/VR, *etc*. Due to complementary properties of cameras and inertial measurement units (IMUs) and the availability of these sensors in smartphones and off-the-shelf plug-and-play devices [3], [4], visual-inertial odometry (VIO) has become popular in recent years. Well-known examples that use VIO are Apple ARKit [1] and Google ARCore [2].

There are two common ways to categorize VIO approaches. Based on *when* visual and inertial measurements are fused, VIO approaches can be divided into loosely-coupled and tightly-coupled approaches. Loosely-coupled approaches [22], [28], [37] estimate motions from images and inertial measurements, independently, and then fuse the two estimates to obtain the final estimate. Tightly-coupled approaches [11], [18], [19] fuse visual and inertial data directly at the measurement level to jointly estimate all IMU and camera states. While loosely coupling is flexible and tends to be more efficient, tightly-coupled approaches generally produce more accurate and robust motion estimates. Our proposed PI-VIO is a tightly-coupled approach.

Based on *how* visual and inertial measurements are fused, VIO approaches can be categorized into filtering-based and optimization-based approaches. Filtering based approaches [19], [33] typically employ the Extended Kalman Filter (EKF), where state propagation/prediction is made by integrating IMU measurements, and update/correction is driven by visual measurements. Contrarily, optimization based approaches [18], [27] use batch nonlinear optimization to directly minimize the errors between integrated motion from IMU measurements and camera motion estimated by the classic reprojection error minimization. Typically optimization-based approaches are more accurate but computationally more expensive due to repeated linearization. There are approaches that combines the advantages from both approaches. For example, PIRVS [41] performs EKF updates iteratively for efficient motion estimation while performing optimization at the back-end to reduce long-term drifts. Our proposed PI-VIO is a filtering-based VIO, and its accuracy as demonstrated by extensive evaluation is at the same level of state-of-the-art optimization based approaches.

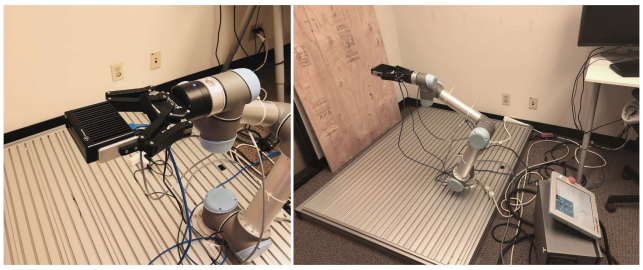


Fig. 1: The PerceptIn Ironsides dataset capture setup. The Ironsides sensor [3] outputs synchronized stereo camera and IMU data, at 60Hz and 200Hz respectively. The 6-axis robot arm with a working radius of 850 mm provides both motion and millimeter-accuracy groundtruth. The dataset contains 9 sequences, featuring various motion types and textures, making it ideal for both evaluation and development.

Most VIO approaches mentioned above only rely on point features, *e.g.* Shi-Tomasi [31], FAST [29], as intermediate image measurements. The performance of these approaches suffer considerably in low-texture environments, or in scenarios when point features can not be reliably detected or tracked, *e.g.* lighting change. Many of such low-texture environments, however, contain planar elements that are rich in linear shapes [14], and the detection of edges is less sensitive to lighting changes in nature. Therefore, in the proposed PI-VIO, in addition to point features, we extract line features as useful image measurements to increase the motion constraints available for challenging scenarios, leading to better system robustness. Both stereo points and line features are processed over a sliding window at cost only linear in the number of features, by using the Multi-State Constraint Kalman Filter [23].

†Feng Zheng, Grace Tsai, Zhe Zhang, Shaoshan Liu, Chen-Chi Chu, and Hongbing Hu are with PerceptIn, Inc., Santa Clara, CA 95054, USA.
Email: {feng.zheng, grace.tsai, zhe.zhang, shaoshan.liu, jason.chu, hongbing.hu}@perceptin.io

In addition, visual or visual-inertial odometry systems typically operate at faster speed but are more prone to drift compared to SLAM (Simultaneous Localization And Mapping) systems because odometry systems do not maintain a persistent map of the environment. Therefore, in the proposed PI-VIO, we introduce a lightweight loop closing method to reduce long-term drift without any computationally expensive map optimization (*i.e.* bundle adjustment).

We summarize our contributions as follow:

- To the best of our knowledge, the proposed PI-VIO approach is the first tightly-coupled filtering based stereo VIO that uses both point and line features.
- We introduce a new lightweight filtering-based loop closing approach formulated as an EFK update, which optimally *relocates* the current sliding window maintained by the filter to the detected loops.
- We conduct extensive evaluation of our PI-VIO with comparison to state-of-the-art open-source VIO approaches including OKVIS [18], VINS-MONO [27], and the recent S-MSCKF [33] using both the EuRoC dataset and our PerceptIn Ironsides dataset.
- We release the PerceptIn Ironsides dataset captured using the Ironsides [3], a high-quality device with synchronized stereo camera and IMU data, with millimeter-accuracy groundtruth from robot arm. The dataset will be available at *https:to-be-filled-in*.

II. RELATED WORK

In this section, we review the state of art of odometry or SLAM approaches in terms of line or edge features and loop closure.

PL-SLAM [26] builds on top of ORB-SLAM [25] and extend its formulation to handle both point and line correspondences in monocular setup. In a similar vein, another joint point and line based work [14], termed PL-SLAM as well, aims at stereo camera setting, and also proposes a bag-of-words (BoW) place recognition method using both point and line descriptors for loop detection. In [34], Tarrio and Pedre introduce an edge-based visual odometry for a monocular camera, with simple extension to using rotation prior obtained from gyroscope as regularization term within edge alignment error minimization. Most recently, Ling *et al.* present a tightly-coupled optimization-based VIO by edge alignment in the distance transform domain [20].

Within rich body of filtering-based VIO literatures, there are not many works using edge or line features. One of the earliest work along this line is [17], which uses only line observations to update the filter and also conducts observability analysis. In [38], the authors extend [17] with a new line parameterization which is shown to exhibit better linearity properties and support rolling-shutter cameras. The edge parametrization introduced in [39] allows non-straight contours. Similar to [17] and [38], we use straight line segments.

Direct methods, such as LSD-SLAM [10], DSO [9], rely on image intensities at high-gradient regions, which include but are not limited to image region of features and edges. Usenko *et al.* [35] extends the vision-only formulation of LSD-SLAM to tightly-couple with IMU by minimizing a combined photometric and inertial energy functional. ROVIO [5], [6] is a direct filtering-based VIO method, using photometric error of image patches as innovation term in the EKF update step. Our usage of line features, to some extent, lies in between direct and feature-based methods. Despite the advantage of feature-free operation, direct methods rely on brightness constancy assumption, and hence sensitive to environment lighting change and camera gain and exposure settings. In contrast, in particular to ROVIO, we use point reprojection error and point-to-line distance as the filter update innovation.

Drift is an inhere issue in SLAM and odometry methods. Loop closure has proven to be effective to correct drift, and state-of-art approaches typically employ global pose graph optimization [25], [27], [14]. In particular, VINS-MONO [27] introduces a two-step loop close procedure: (1) tightly-coupled relocalization which aligns the sliding window with past poses, and (2) global pose graph optimization. Our lightweight loop close resembles the first step employed by VINS-MONO, except that we realize it in a filtering framework and we exclude global pose graph optimization for efficiency. To our best knowledge, it is the first tightly-coupled filtering-based loop close. Furthermore, our proposed PI-VIO handles both stereo point and line features and loop closure in a consistent filtering-based framework.

III. ESTIMATOR DESCRIPTION

The backbone of our estimator is the Multi-State Constraint Kalman Filter (MSCKF) [23]. The key idea of MSCKF is to maintain and update a sliding window of camera poses using feature track observations without including features in the filter state. Instead, 3D feature positions are estimated via least-squares multi-view triangulation and subsequently marginalized, which resembles structureless BA to some extent. The advantage of doing this is considerable reduction of computational cost, making MSCKF's complexity linear in the number of features, instead of cubic like EKF-SLAM [8].

We introduce two types of EKF updates: (1) joint point and line feature update to cope with challenging scenarios and enhance robustness, and (2) loop closing update to reduce accumulated drift. Filter consistency is ensured by using the right nullspace of the observability Gramian to modify state transition matrix and observation matrix at each propagation and update step following [15].

A. State Parameterization

We follow [23] and define the evolving IMU state as follows:

$$\mathbf{X}_B = [\ {}^B_G\mathbf{q}^T \quad \mathbf{b}_g{}^T \quad {}^G\mathbf{v}_B^T \quad \mathbf{b}_a{}^T \quad {}^G\mathbf{p}_B^T \quad {}^B_C\mathbf{q}^T \quad {}^B\mathbf{p}_C^T\ ]^T \quad (1)$$

where ${}^B_G\mathbf{q}$ is the unit quaternion representing the rotation from the global frame $\{G\}$ to the IMU body frame $\{B\}$, ${}^G\mathbf{p}_B$ and ${}^G\mathbf{v}_B$ are the IMU position and velocity in the global frame, and $\mathbf{b}_g$ and $\mathbf{b}_a$ denote gyroscope and accelerometer

biases. Optionally, we include IMU extrinsics ${}^{B}_{C}\mathbf{q}$ and ${}^{B}\mathbf{p}_{C}$ in the state, which represent the rotation and the translation between the IMU body frame $\{B\}$ and the camera frame $\{C\}$.

At time k, the full state of our estimator consists of the current IMU state estimate and N camera poses

$$\hat{\mathbf{X}}_k = [\ \hat{\mathbf{X}}_{B_k}^T \quad \hat{\mathbf{X}}_{C_1}^T \quad ... \quad \hat{\mathbf{X}}_{C_N}^T\ ]^T \tag{2}$$

where $\hat{\mathbf{X}}_C = [{}^{C}_{G}\hat{\mathbf{q}}^T \ {}^{G}\hat{\mathbf{p}}_C^T]^T$ represents the camera pose estimate.

We use the error-state representation in order to minimally parameterize orientation in 3 degrees of freedom (DOF) and to avoid singularities [32]. Specifically, for the position, velocity, and biases, the standard additive error is employed, while for the orientations, the compositional update $\mathbf{q} = \boldsymbol{\delta}\mathbf{q} \otimes \hat{\mathbf{q}}$ is used, where $\boldsymbol{\delta}\mathbf{q}$ is the 3DOF error quaternion as follows

$$\boldsymbol{\delta}\mathbf{q} = [\frac{1}{2}\boldsymbol{\delta\theta}\ 1]^T \tag{3}$$

B. EKF Propagation

Whenever a new IMU measurement is received, it is used to propagate the EKF state and covariance estimates. We use the standard continuous-time IMU kinematics model as follows

$$\begin{aligned}
{}^{B}_{G}\dot{\hat{\mathbf{q}}} &= \frac{1}{2}\Omega(\hat{\boldsymbol{\omega}}){}^{B}_{G}\hat{\mathbf{q}} && (4)\\
\dot{\hat{\mathbf{b}}}_g &= \mathbf{0}_{3\times1} && (5)\\
{}^{G}\dot{\hat{\mathbf{v}}} &= R({}^{G}_{B}\hat{\mathbf{q}})\,\hat{\mathbf{a}} + {}^{G}\mathbf{g} && (6)\\
\dot{\hat{\mathbf{b}}}_a &= \mathbf{0}_{3\times1} && (7)\\
{}^{G}\dot{\hat{\mathbf{p}}}_B &= {}^{G}\hat{\mathbf{v}} && (8)\\
{}^{B}_{C}\dot{\hat{\mathbf{q}}} &= \mathbf{0}_{3\times1} && (9)\\
{}^{B}\dot{\hat{\mathbf{p}}}_C &= \mathbf{0}_{3\times1} && (10)
\end{aligned}$$

where $\hat{\boldsymbol{\omega}}$ and $\hat{\mathbf{a}}$ are angular velocity and linear acceleration from gyroscope and accelerometer respectively with biases removed, R denotes the corresponding rotation matrix of the quaternion, and $\Omega(\hat{\boldsymbol{\omega}}) \in \mathbb{R}^{4\times4}$ is the skew-symmetric matrix formed from the angular rate

$$\Omega(\hat{\boldsymbol{\omega}}) = \begin{bmatrix} -[\hat{\boldsymbol{\omega}}_\times] & \hat{\boldsymbol{\omega}} \\ -\hat{\boldsymbol{\omega}}^T & 0 \end{bmatrix} \tag{11}$$

Our discrete-time implementation employs 4th order Runge-Kutta numerical method. For sake of simplicity, we omit the description of state transition matrix and covariance propagation. Interested readers please refer to [23].

C. Measurement Model for Point Features

In MSCKF, all the *continuous* measurements of the same 3D point, *i.e.* feature tracks, are used to update all involved camera poses that observe the point. The residual is the standard reprojection error:

$$\mathbf{r}_{f_i} = \mathbf{z}_{f_i} - \hat{\mathbf{z}}_{f_i} \tag{12}$$

where $\mathbf{z}_{f_i} = [u_i\ v_i]^T$ is the observation of the *i-th* feature in the image, while $\hat{\mathbf{z}}_{f_i}$ is the predicted measurement of the feature from projecting its estimated 3D position ${}^{G}\hat{\mathbf{p}}_{f_i} = [{}^{G}\hat{X}_i\ {}^{G}\hat{Y}_i\ {}^{G}\hat{Z}_i]^T$ in global frame into the image based on the camera pose estimated and the projection model as follows

$$\hat{\mathbf{z}}_{f_i} = \pi(\ R({}^{C}_{G}\hat{\mathbf{q}})({}^{G}\mathbf{p}_{f_i} - {}^{G}\hat{\mathbf{p}}_C)\ ) \tag{13}$$

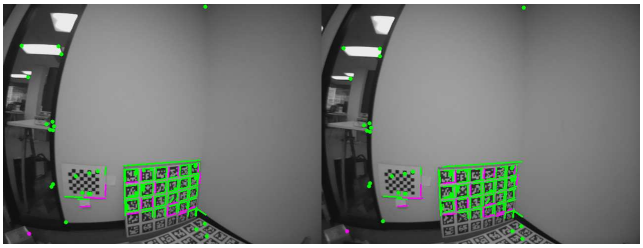
where π is the pinhole projection model ($\pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$)

$$\pi(\mathbf{p}) = \frac{1}{Z}\begin{bmatrix} X \\ Y \end{bmatrix} \tag{14}$$

We then linearize Eq. 12 about the estimates for the camera pose and for the feature position, and calculate the Jacobians with respect to the state and the feature position as $H_{X_{f_i}}$ and H_{f_i} respectively following [23]. After that, we marginalize the feature position via nullspace projection to de-correlate it with the state.

So far, we have described the measurement model for monocular camera. One tricky part is the estimation of 3D feature positions, which is typically computed by multi-view triangulation in least-squares fashion. There has to be enough baseline among the cameras observing the same feature in order to do the triangulation. Therefore, monocular MSCKF cannot estimate the 3D positions of features nor do EKF updates while being static or undergoes rotation dominant motion. This motivates us to adopt the more practical stereo camera setup to overcome this limitation, from which we can also get the true scale. For stereo feature measurements, we employ a simple yet effective representation, similar to [33],

$$\hat{\mathbf{z}}_{f_i} = \begin{bmatrix} \pi({}^{C_1}\hat{\mathbf{p}}_{f_i}) \\ \pi({}^{C_2}\hat{\mathbf{p}}_{f_i}) \end{bmatrix} = \begin{bmatrix} \pi(\ R({}^{C_1}_{G}\hat{\mathbf{q}})({}^{G}\mathbf{p}_{f_i} - {}^{G}\hat{\mathbf{p}}_{C_1})\ ) \\ \pi(\ R({}^{C_2}_{G}\hat{\mathbf{q}})({}^{G}\mathbf{p}_{f_i} - {}^{G}\hat{\mathbf{p}}_{C_2})\ ) \end{bmatrix} \tag{15}$$



(a) Stereo frame 579

(b) Stereo frame 1262

Fig. 2: Stereo point features and line features. Magenta: new features. Green: tracked features. Line features help improve system robustness in challenging scenarios, e.g. low-texture environment (a) and motion blur (b). These two stereo frames are from PerceptIn Ironsides dataset PI_3058. (Best viewed in color or electronically.)

where $\hat{\mathbf{z}}_{f_i} \in \mathbb{R}^4$, ${}^{C_1}\hat{\mathbf{p}}_{f_i}$ and ${}^{C_2}\hat{\mathbf{p}}_{f_i}$ are the estimated 3D positions of the same feature point in left and right camera coordinates respectively, and $\hat{\mathbf{X}}_{C_1} = [{}^{C_1}_{G}\hat{\mathbf{q}}^T \ {}^{G}\hat{\mathbf{p}}^T_{C_1}]^T$ and $\hat{\mathbf{X}}_{C_2} = [{}^{C_2}_{G}\hat{\mathbf{q}}^T \ {}^{G}\hat{\mathbf{p}}^T_{C_2}]^T$ are stereo camera poses at the same timestamp. Note that the stereo camera is assumed to be calibrated beforehand, and the camera extrinsics relating the left and the right cameras is assumed to be constant.

D. Measurement Model for Line Features

We now present the measurement model of line features for updating the state estimates. We denote a line l_i in image using point-normal form, $l_i = [\mathbf{z}_{l_i} \ \vec{\mathbf{n}}_{l_i}]$, where $\mathbf{z}_i$ is any point on the line and $\vec{\mathbf{n}}_{l_i} \in \mathbb{R}^{2\times 1}$ is a unit vector denoting line's normal direction in image space. For a 3D line, L_j, we over-parameterize it by using two 3D endpoints, ${}^{G}L_j = [{}^{G}\mathbf{p}_b \ {}^{G}\mathbf{p}_e]$, where ${}^{G}\mathbf{p}_b$ and ${}^{G}\mathbf{p}_e$ are the beginning and ending endpoints on the 3D line in the global frame.

For the line feature residual, $\mathbf{r}_{l_i} \in \mathbb{R}^{2\times 1}$, we use the point to line distance, as follows

$$\mathbf{r}_{l_i} = \begin{bmatrix} (\mathbf{z}_{l_i} - \hat{\mathbf{z}}_{l_{ib}}) \cdot \vec{\mathbf{n}}_{l_i} \\ (\mathbf{z}_{l_i} - \hat{\mathbf{z}}_{l_{ie}}) \cdot \vec{\mathbf{n}}_{l_i} \end{bmatrix} \tag{16}$$

where $\hat{\mathbf{z}}_{l_{ib}} \in \mathbb{R}^{2\times 1}$ and $\hat{\mathbf{z}}_{l_{ie}} \in \mathbb{R}^{2\times 1}$ are the 2D projections of the beginning and ending endpoints on the 3D line, and $\cdot$ represents dot product. To conform to the standard form of EKF residual as in Eq. 12, we simplify Eq. 16 to

$$\mathbf{r}_{l_i} = \begin{bmatrix} \vec{\mathbf{n}}^T_{l_i}\mathbf{z}_{l_i} - \vec{\mathbf{n}}^T_{l_i}\hat{\mathbf{z}}_{l_{ib}} \\ \vec{\mathbf{n}}^T_{l_i}\mathbf{z}_{l_i} - \vec{\mathbf{n}}^T_{l_i}\hat{\mathbf{z}}_{l_{ie}} \end{bmatrix} \tag{17}$$

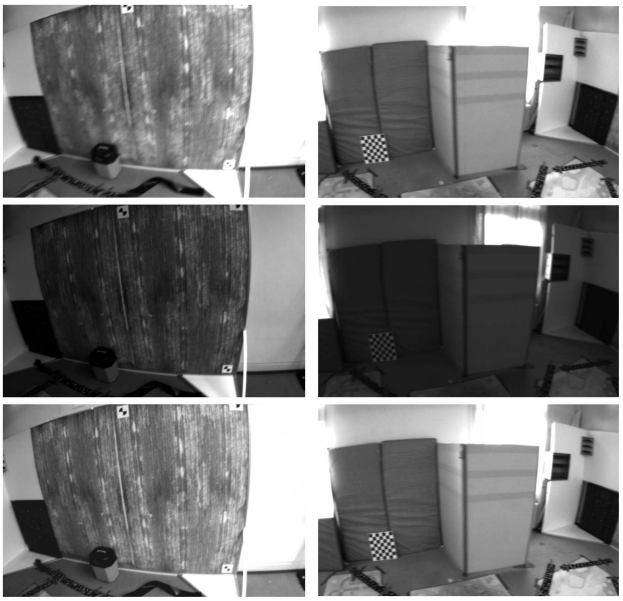
Note $\vec{\mathbf{n}}^T_{l_{ib}}\mathbf{z}_{l_{ib}}$ produces a scalar number, thus one 3D endpoint results in one dimensional residual. This is desirable as line features can only provide useful constraints in the normal direction. Therefore, every 3D line represented by two endpoints produces a two dimensional residual. The over-parameterization makes sure that, if the projected 3D line and the observed line do not perfectly align, at least in one dimension of the residual it will not be zero. This holds even when one projected endpoint is accidentally on the observed line.

Another benefit of this measurement model is that the line feature Jacobian becomes extremely easy to calculate and feature marginalization can be done in the same way as point features. Under the chain rule, we can derive the Jacobian for each line as follows

$$\mathbf{H}_{l_i} = \begin{bmatrix} \vec{\mathbf{n}}^T_{l_i}\mathbf{H}_{l_{ib}} \\ \vec{\mathbf{n}}^T_{l_i}\mathbf{H}_{l_{ie}} \end{bmatrix} \tag{18}$$

where $\mathbf{H}_{l_{ib}} \in \mathbb{R}^{2\times 3}$ can be calculated in the same way as point feature Jacobian H_{f_i} and the "point" here is the beginning endpoint of the line. Likewise, $\mathbf{H}_{l_{ie}} \in \mathbb{R}^{2\times 3}$ is the "point" Jacobian of the line ending endpoint. Note that $\vec{\mathbf{n}}^T_{l_i}\mathbf{H}_{l_{ib}} \in \mathbb{R}^{1\times 3}$, hence $\mathbf{H}_{l_i} \in \mathbb{R}^{2\times 3}$. Similarly, the Jacobian of line feature with respect to the state can be derived as

$$\mathbf{H}_{X_{l_i}} = \begin{bmatrix} \vec{\mathbf{n}}^T_{l_i}\mathbf{H}_{X_{l_{ib}}} \\ \vec{\mathbf{n}}^T_{l_i}\mathbf{H}_{X_{l_{ie}}} \end{bmatrix} \tag{19}$$



(a) Stereo (b) Temporal

Fig. 3: Examples of histogram matching to deal with stereo and temporal brightness mismatch. (a) Stereo histogram matching: top - stereo left image, middle - original stereo right image, bottom - stereo right image after histogram matching. The frames are from EuRoC V2_03_difficult dataset. (b) Temporal histogram matching: top - left image, middle - original left image at the next timestamp, bottom - left image at the next timestamp after histogram matching. The frames are from EuRoC V1_03_difficult dataset. It is obvious that middle images in (a) and (b) exhibit strong brightness difference compared to the top ones, and the histogram matching results at the bottom match the top ones well in terms of overall brightness and distribution. Intensity based feature tracking and matching, *e.g.* KLT optical flow [21], can significantly benefit from this operation.

where $\mathbf{H}_{X_{l_{ib}}}$ and $\mathbf{H}_{X_{l_{ie}}}$ are the Jacobians of the line's beginning and ending endpoints with respect to the state, and they share the same formula as point features.

To extend the measurement model to the stereo setting is straightforward. We follow the way we use for stereo point features, and represent the stereo line residual, $\mathbf{r}_{l_i} \in \mathbb{R}^4$, as follows

$$\mathbf{r}_{l_i} = \begin{bmatrix} \vec{\mathbf{n}}^T_{l_i,1}\mathbf{z}_{l_i,1} - \vec{\mathbf{n}}^T_{l_i,1}\hat{\mathbf{z}}_{l_{ib},1} \\ \vec{\mathbf{n}}^T_{l_i,1}\mathbf{z}_{l_i,1} - \vec{\mathbf{n}}^T_{l_i,1}\hat{\mathbf{z}}_{l_{ie},1} \\ \vec{\mathbf{n}}^T_{l_i,2}\mathbf{z}_{l_i,2} - \vec{\mathbf{n}}^T_{l_i,2}\hat{\mathbf{z}}_{l_{ib},2} \\ \vec{\mathbf{n}}^T_{l_i,2}\mathbf{z}_{l_i,2} - \vec{\mathbf{n}}^T_{l_i,2}\hat{\mathbf{z}}_{l_{ie},2} \end{bmatrix} \tag{20}$$

E. EKF Update: Point and Line Features

We adopt a similar update strategy as [23]: whenever a point and/or line feature is no longer tracked, or the sliding window size exceeds the predefined maximum size, EKF update is triggered. Point and line features are subsequently marginalized since their positions are directly correlated with

the state estimate $\hat{X}$. This makes the algorithm complexity linear in the number of features. The marginalization is performed by using the left nullspace of feature Jacobian, which cancels out the feature term in the linearized residual. We then stack the transformed residuals and the state Jacobians of both points and lines to form the final residual and observation matrix.

F. EKF Update: Loop Closure

To reduce accumulated drift while being efficient for resource constrained platforms which cannot afford global BA, we present here a new lightweight loop closing method which is formulated as an EKF update. As will be described in Section V, when a new camera state is added to the sliding window, we perform keyframe selection and trigger loop detection in a parallel thread if selected. If a loop is detected while the keyframe is still in the sliding window, loop closing update is triggered. Otherwise, the keyframe is added to the database along with feature descriptors and 3D positions.

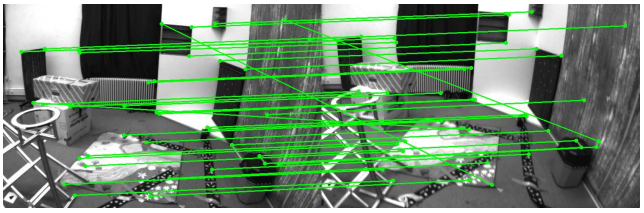
Since loop detection establishes feature matches between the current keyframe and the loop closing keyframe, we use feature positions from the loop closing keyframe for EKF update instead of triangulating them based on current sliding window poses. The update procedure is almost the same as the update with point features, except that we treat 3D positions of such loop matched features as prior knowledge and thus do not perform either feature Jacobian calculation or marginalization. This makes sense given such "map" points have been marginalized along with the loop closing keyframe when it slides out of the sliding window and is inserted into the loop detection database in the past. Note that this is similar to [24], [41] except that their maps are either pre-built or online estimated via the costly bundle adjustment. To our best knowledge, our introduction of "map" based EKF update for loop closing is novel.

IV. IMAGE PROCESSING

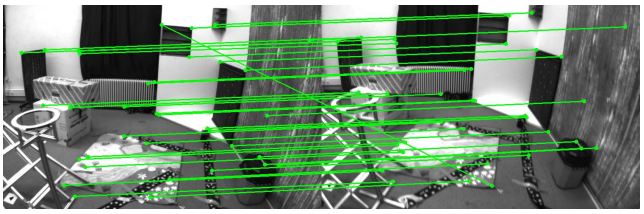
In this section, we describe our image processing pipeline for detection and tracking of point and line features. An example is shown in Fig. 2.

For each new image, we track existing point features via KLT optical flow (OF) [21] and for non-tracked image regions new features are detected via FAST feature detector [29]. We enforce uniform distribution of features in image by spatial binning and maintain a fixed number of high response features in each bin. To cope with fast motion, we obtain initial guess for optical flow using relative rotation computed from gyroscope measurements. We use KLT OF for stereo feature matching as well similar to [33] for efficiency. To reject outliers in both stereo and temporal matching, we use 2-point RANSAC and space-time circular matching [16].

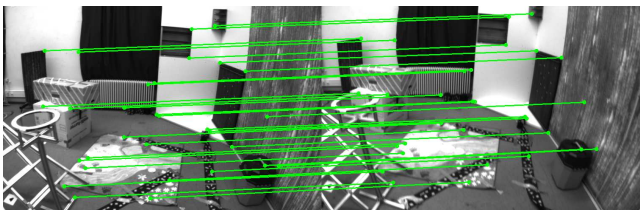
For line features, we use Line Segment Detector (LSD) [36] to extract line segments. For each line segment detected, we extract binary descriptor using Line Band Descriptor (LBD) [40]. Both stereo and temporal matching of line features are based on LBD descriptor matching. To ensure best matches, we perform 4-way consistency check, i.e. left to right, right to left, previous to current, and current to previous. Furthermore, we prune putative matches by checking length and orientation of lines.



(a) Raw feature matches



(b) After fundamental matrix RANSAC removal



(c) After PnP RANSAC removal

Fig. 4: Two-step outlier rejection for loop detection feature matches. The frames are from EuRoC V2_02_medium dataset. (Best viewed in color.)

To further enhance the robustness of feature tracking and matching, we introduce a fast brightness check between stereo and temporally consecutive images based on their mean brightnesses, and perform histogram matching to ensure consistent brightness and contrast across stereo and temporal images if necessary. An example is shown in Fig. 3. This considerably boosts stereo feature matching and temporal tracking performance under unfavorable conditions, *e.g.* auto-exposure mismatch between stereo cameras, dramatic lighting change. This is in contrast to using histogram equalization for each frame as done in [27] and [41], which incurs more computation and disregards brightness consistency between stereo and temporal images, and may result in over-enhancement.

V. LOOP DETECTION

In this section, we describe our loop detection approach. For each new image, we do keyframe selection based on the number of features tracked and the pose distance to existing keyframes in the loop detection database. If a keyframe is selected, we extract ORB descriptors [30] for loop detection. Our loop detection is implemented based on DBoW2 [12] which is both fast and reliable, and it runs in a parallel thread to the main VIO thread. For candidate loops, similar to [27],

we perform two-step outlier rejection: 2D-2D fundamental matrix test and 3D-2D PnP test both within the RANSAC framework. This outlier rejection strategy is effective as shown in Fig. 4. If the number of inlier feature matches is above the pre-defined threshold, we mark loop detected and trigger loop close EKF update as described in Sec. III-F. If the current keyframe does not contain loops, we add it to the database when it is marginalized from the active sliding window maintained by the filter along with its pose, feature 2D and 3D positions, and descriptors. We set a maximum number of keyframes in the database considering memory requirement and detection speed to make sure that it returns result within one frame.

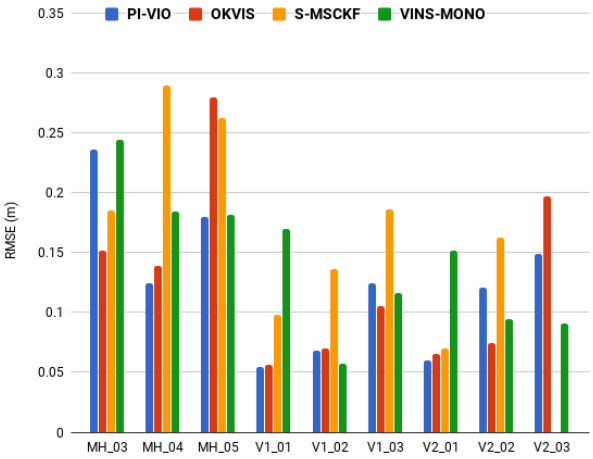
VI. EXPERIMENTS

In this section, we introduce two experiments to demonstrate the performance of the proposed PI-VIO approach. Both experiments compare PI-VIO to competitive state-of-the-art VIO approaches including OKVIS [18], VINS-MONO [27], and S-MSCKF [33]. OKVIS and VINS-MONO are optimization based tightly coupled VIO systems, while S-MSCKF is a tightly-coupled filtering-based stereo VIO system closely related to us. Both OKVIS and S-MSCKF support stereo camera hence we run them in stereo mode, while VINS-MONO is a monocular system. The first experiment is conducted with the public EuRoC MAV dataset [7], while the second is with our new PerceptIn Ironsides dataset. As all comparison approaches contain more or less non-determinism, *e.g.* due to RANSAC, we repeat all experiments five times and report median numbers.

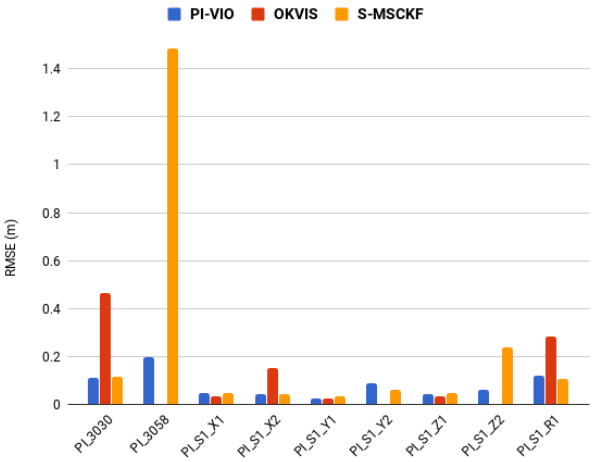
A. EuRoC MAV Dataset

The EuRoC dataset contains elven sub-datasets in three categories (MH, V1, V2) collected on-board a Micro Aerial Vehicle (MAV). We select nine from them so that each category contains 3 datasets. For comparison approaches, we use their default parameters, as they all have been carefully tuned for EuRoC dataset. In addition, we keep the global loop closing functionality on for VINS-MONO, as we want to compare with it in its best form. Evaluation results are shown in Fig. 5a. It is evident that PI-VIO is among the best performing methods, leading results in MH_04, MH_05, V1_01, and V2_01.

For V2_03 dataset, S-MSCKF produces poor results, mentioned in [33] as well, for the reason that "the continuous inconsistency in brightness between the stereo images causes failures in stereo feature matching". Hence, its result is not reported in Fig. 5a. In contrast, the histogram matching employed by PI-VIO makes it robust to this challenging scenario, as demonstrated in Fig. 3. In addition, V2_03 has about 400 missing frames in the left camera data, resulting in OKVIS tracking failure. After we prune extra frames from the right camera data, OKVIS runs well. Note that we use the original V2_03 dataset for PI-VIO evaluation as our approach is robust enough to handle frame drop in either stereo or temporal frames. VINS-MONO is not affected as it is a monocular approach and uses only left camera data.



(a) EuRoC dataset



(b) PerceptIn Ironsides dataset

Fig. 5: Absolute trajectory RMSE (Root Mean Square Error) results of our PI-VIO and competing approaches including OKVIS, S-MSCKF, and VINS-MONO on both the public EuRoC dataset and our new PerceptIn Ironsides dataset. Note that we exclude VINS-MONO from the second evaluation. (Best viewed in color.)

B. PerceptIn Ironsides Dataset

To further evaluate the performance of PI-VIO, we introduce a new public dataset. The dataset is recorded by PerceptIn Ironsides [3] in a robot arm platform as shown in Fig. 1. We collect in total 9 sub-datasets, featuring a wide range of motions and environmental conditions, from controlled slow motion around each axis under good visual conditions to fast random motion with motion blur and low texture. We provide the entire datasset in two formats, ROS bag and zipped format, similar to the EuRoC dataset. The groundtruth provided by the robot arm platform is up to millimeter accuracy and precisely synced with the Ironsides sensor. Therefore, the PerceptIn Ironsides dataset is ideal for

both VIO/SLAM evaluation and development.

The comparison result is shown in Fig. 5b. We tune parameters of all approaches to make them perform well as much as possible. Our PI-VIO is consistently among the top two best performing approaches. S-MSCKF results are close to us, except dataset PI_3058, where we show significantly better results due to the usage of additional line features. PI_3058 is the most challenging one in the Ironsides dataset with fast motion and low texture in many parts of the sequence, making it hard for VIO approaches which rely on only point features. For OKVIS, it performs well for easy sequences (PI_S1_X1, PI_S1_Y1, and PI_S1_Z1) whose dominant motions are slow translation. However, OKVIS produces poor results for PI_3058, PI_S1_Y2, and PI_S1_Z2, hence we omit reporting those numbers. We notice that OKVIS's feature matching suffers from repetitive textures in the scene. Note that we exclude VINS-MONO from this comparison for two reasons: (1) its distortion model is not exactly compatible with the Ironsides calibration, and (2) rotation dominant motion at the beginning in many datasets leads to poor initialization. To improve monocular SLAM initialization, delayed initialization till enough parallax and model selection between fundamental matrix and homography could help [13], [25].

VII. CONCLUSIONS

In this work, we have presented PerceptIn Visual Inertial Odometry (PI-VIO), a new tightly-coupled filtering-based stereo visual inertial odometry approach using both point and line features. Line features help improve system robustness in point-scarce scenarios, *e.g.* low texture and changing light. Both stereo points and line features are processed over a sliding window at cost only linear in the number of features. To reduce drift, which is inherent in any odometry approach, we have introduced a new lightweight loop closing method naturally formulated as EKF updates to optimally *relocate* the current sliding window maintained by the filter to past keyframes. All of them (point features, line features, and loop closure) are handled in a common filtering-based framework.

We have also presented PerceptIn Ironsides dataset, a new public visual-inertial dataset, featuring high-quality synchronized stereo camera and IMU data from the Ironsides sensor with various motion types and textures and millimeter-accuracy groundtruth. The extensive evaluation against competitive state-of-the-art approaches using this new dataset and the public EuRoC dataset clearly demonstrate the superior performance of the proposed PI-VIO approach.

ACKNOWLEDGMENT

Thanks Yen-Cheng Liu for calibrating Ironsides devices and collecting the dataset. Thanks Chandrahas Jagadish Ramalad for preparing the PerceptIn Ironsides dataset for release.

REFERENCES

[1] "Apple ARKit." [Online]. Available: https://developer.apple.com/arkit/
[2] "Google ARCore." [Online]. Available: https://developers.google.com/ar/
[3] "PerceptIn Ironsides: Visual Inertial Computing Module." [Online]. Available: https://www.perceptin.io/ironsides
[4] "PerceptIn Old Ironsides: Visual Inertial Module." [Online]. Available: https://www.perceptin.io/old-ironsides
[5] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct ekf-based approach," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2015, pp. 298–304.
[6] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended kalman filter based visual-inertial odometry using direct photometric feedback," *The International Journal of Robotics Research*, vol. 36, no. 10, pp. 1053–1072, 2017.
[7] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *The International Journal of Robotics Research*, 2016.
[8] A. Davison, I. Reid, N. Molton, and O. Stasse, "MonoSLAM: real-time single camera SLAM." *Pattern Anal. Mach. Intell. (PAMI), IEEE Trans.*, vol. 29, no. 6, pp. 1052–67, 2007.
[9] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, March 2018.
[10] J. Engel, T. Schoeps, and D. Cremers, "LSD-SLAM: Large-Scale Direct Monocular SLAM," pp. 834–849, 2014.
[11] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, Feb 2017.
[12] D. Galvez-Lopez and J. D. Tardos, "Bags of Binary Words for Fast Place Recognition in Image Sequences," *IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 28, no. 5, pp. 1188–1197, 2012.
[13] S. Gauglitz, C. Sweeney, J. Ventura, M. Turk, and T. Hllerer, "Live tracking and mapping from both general and rotation-only camera motion," in *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Nov 2012, pp. 13–22.
[14] R. Gomez-Ojeda, F. Moreno, D. Scaramuzza, and J. G. Jiménez, "PL-SLAM: a stereo SLAM system through the combination of points and line segments," *CoRR*, vol. abs/1705.09479, 2017.
[15] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Consistency analysis and improvement of vision-aided inertial navigation," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 158–176, Feb 2014.
[16] B. Kitt, A. Geiger, and H. Lategahn, "Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme," in *Intelligent Vehicles Symposium (IV)*, 2010.
[17] D. G. Kottas and S. I. Roumeliotis, "Efficient and consistent vision-aided inertial navigation using line observations," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 1540–1547.
[18] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," *Int. J. Rob. Res.*, vol. 34, no. 3, pp. 314–334, 2015.
[19] M. Li and A. I. Mourikis, "High-precision, consistent ekf-based visual-inertial odometry," *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013.
[20] Y. Ling, M. Kuse, and S. Shen, "Edge alignment-based visual–inertial fusion for tracking of aggressive motions," *Autonomous Robots*, vol. 42, no. 3, pp. 513–528, Mar 2018.
[21] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, ser. IJCAI'81. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1981, pp. 674–679.
[22] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to mav navigation," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 3923–3929.
[23] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, April 2007, pp. 3565–3572.
[24] A. I. Mourikis, N. Trawny, S. I. Roumeliotis, A. E. Johnson, A. Ansar, and L. Matthies, "Vision-aided inertial navigation for spacecraft entry, descent, and landing," *IEEE Transactions on Robotics*, vol. 25, no. 2, pp. 264–280, April 2009.
[25] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM:

A Versatile and Accurate Monocular SLAM System," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, 2015.

[26] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "PL-SLAM: Real-time monocular visual SLAM with points and lines," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 4503–4508.

[27] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *arXiv preprint arXiv:1708.03852*, 2017.

[28] A. Ranganathan, M. Kaess, and F. Dellaert, "Fast 3d pose estimation with out-of-sequence measurements," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2007, pp. 2486–2493.

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

[30] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 2564–2571, 2011.

[31] J. Shi and C. Tomasi, "Good features to track," in *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Jun 1994, pp. 593–600.

[32] J. Solà, "Quaternion kinematics for the error-state kalman filter," *CoRR*, vol. abs/1711.02508, 2017.

[33] K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor, and V. Kumar, "Robust stereo visual inertial odometry for fast autonomous flight," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 965–972, April 2018.

[34] J. J. Tarrio and S. Pedre, "Realtime edge-based visual odometry for a monocular camera," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 702–710.

[35] V. Usenko, J. Engel, J. Stckler, and D. Cremers, "Direct visual-inertial odometry with stereo cameras," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 1885–1892.

[36] R. G. von Gioi, J. Jakubowicz, J. M. Morel, and G. Randall, "LSD: A Fast Line Segment Detector with a False Detection Control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 722–732, April 2010.

[37] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments," in *2012 IEEE International Conference on Robotics and Automation*, May 2012, pp. 957–964.

[38] H. Yu and A. I. Mourikis, "Vision-aided inertial navigation with line features and a rolling-shutter camera," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2015, pp. 892–899.

[39] ——, "Edge-based visual-inertial odometry," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2017, pp. 6670–6677.

[40] L. Zhang and R. Koch, "An efficient and robust line segment matching approach based on lbd descriptor and pairwise geometric consistency," *J. Vis. Comun. Image Represent.*, vol. 24, no. 7, pp. 794–805, Oct. 2013. [Online]. Available: http://dx.doi.org/10.1016/j.jvcir.2013.05.006

[41] Z. Zhang, S. Liu, G. Tsai, H. Hu, C.-C. Chu, and F. Zheng, "PIRVS: An Advanced Visual-Inertial SLAM System with Flexible Sensor Fusion and Hardware Co-Design," in *2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, Australia, May 20 - May 25*, 2018.