A Comparative Analysis of Tightly-coupled Monocular, Binocular, and Stereo VINS

Mrinal K. Paul, Kejian Wu, Joel A. Hesch, Esha D. Nerurkar, and Stergios I. Roumeliotis†

***Abstract*— In this paper, a sliding-window two-camera vision-aided inertial navigation system (VINS) is presented in the square-root inverse domain. The performance of the system is assessed for the cases where feature matches across the two-camera images are processed with or without any stereo constraints (i.e., stereo vs. binocular). To support the comparison results, a theoretical analysis on the information gain when transitioning from binocular to stereo is also presented. Additionally, the advantage of using a two-camera (both stereo and binocular) system over a monocular VINS is assessed. Furthermore, the impact on the achieved accuracy of different image-processing frontends and estimator design choices is quantified. Finally, a thorough evaluation of the algorithm's processing requirements, which runs in real-time on a mobile processor, as well as its achieved accuracy as compared to alternative approaches is provided, for various scenes and motion profiles.**

I. Introduction and Related Work

Combining measurements from an inertial measurement unit (IMU) with visual observations from a camera, known as a VINS, is a popular means for navigating within GPS-denied areas (e.g., underground, in space, or indoors). With the growing availability of such sensors in mobile devices (e.g., [1]), the research focus in VINS is gradually turning towards finding efficient, real-time solutions on resource-constrained devices. Moreover, with the recent improvements in mobile CPUs and GPUs (e.g., NVIDIA's TK1 [2]), the interest in more robust multi-camera VINS is also increasing.

Most existing VINS approaches can be classified into loosely-coupled and tightly-coupled systems. In loosely-coupled systems, either the integrated IMU data are incorporated as independent measurements into the (stereo) vision optimization (e.g., [3]), or vision-only pose estimates are used to update an extended Kalman filter (EKF) performing IMU propagation (e.g., [4]). In contrast, tightly-coupled approaches jointly optimize over all sensor measurements (e.g., [5], [6], [7]) which results in higher accuracy.

The first step towards a multi-camera VINS is to employ two cameras. Incorporating, however, a second camera to a localization system is usually computationally costly. Thus, although there exist many implementations of loosely-coupled stereo systems (e.g., [3], [8]), or approaches that use stereo only for feature initialization (e.g., [9]), only few works address the more computationally demanding tightly-coupled stereo VINS. To the best of our knowledge, Leutenegger et al. [7] and Manderson et al. [10] present the only tightly-coupled stereo VINS, which operate in real-time but only on desktop CPUs. Manderson et al. [10] employ an extension of PTAM [11] where the tracking and mapping pipelines are decoupled, and hence is inconsistent.[1] On the other hand, Leutenegger et al. [7] propose a consistent keyframe-based stereo simultaneous localization and mapping (SLAM) algorithm that performs nonlinear optimization over both visual and inertial cost terms. In order to maintain the sparsity of the system, their approach employs the following approximation: Instead of marginalizing the landmarks associated with the oldest pose in the temporal window, these are dropped from the system (fully dropped for non-keyframes and partially dropped for keyframes), rendering their approach sub-optimal.

In contrast, our method builds on and extends the work of [6], where a monocular VINS is presented in the inverse square-root form (termed as SR-ISWF). In this paper, we present both stereo and duo (binocular i.e., two independent cameras, without any stereo constrains between them) VINS, adopting the approach of [6]. We experimentally show the benefit of a stereo system over mono and duo systems, especially in challenging environments and motion profiles. We also provide a theoretical analysis on the performance gain in stereo, as compared to duo, in terms of the Cholesky factor update. Additionally, we present the impact of different image-processing frontends on VINS and show that our stereo system operates in real-time on mobile processors and achieves high accuracy, even with a low-cost commercial-grade IMU, as opposed to [7] that employs an industrial-grade IMU. Our main contributions are:

- We present the first tightly-coupled stereo VINS that operates in real-time on mobile processors.
- We present a detailed comparison between mono, duo, and stereo VINS under different scenes and motion profiles, and provide a theoretical explanation of the information gain when transitioning from duo to stereo.
- We assess the impact of different image-processing frontends on the estimation accuracy of VINS, and perform a sensitivity analysis of different frontends, with respect to the changes in feature track length. Moreover, we provide a detailed analysis of how different design

†This work was supported by Google, Project Tango.

M. K. Paul and S. I. Roumeliotis are with the Department of Computer Science and Engineering, Univ. of Minnesota, Minneapolis, MN, USA. {paulx152,stergios}@umn.edu

K. Wu is with the Department of Electrical and Computer Engineering, Univ. of Minnesota, Minneapolis, MN, USA. kejian@cs.umn.edu

J. A. Hesch and E. D. Nerurkar are with Project Tango, Google, Mountain View, CA, USA. {joelhesch, eshanerurkar}@google.com

[1] As defined in [12], a state estimator is consistent if the estimation errors are zero-mean and have covariance matrix smaller or equal to the one calculated by the filter. Since PTAM considers parts of the state vector to be perfectly known during its tracking or mapping phases, the resulting Hessian does not reflect the information and hence uncertainty of the system.

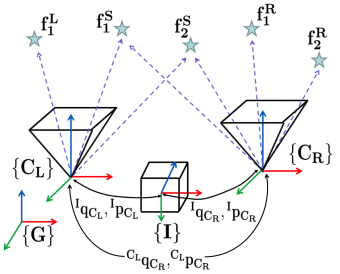

Fig. 1. Stereo camera-IMU setup, where $\{I\}$, $\{C_L\}$, $\{C_R\}$, and $\{G\}$ are the IMU, left camera, right camera, and global frames, respectively, $({}^{I}\mathbf{q}_{C_L}, {}^{I}\mathbf{p}_{C_L})$ and $({}^{I}\mathbf{q}_{C_R}, {}^{I}\mathbf{p}_{C_R})$ are the corresponding left and right IMU-camera extrinsic parameters, $({}^{C_L}\mathbf{q}_{C_R}, {}^{C_L}\mathbf{p}_{C_R})$ are the extrinsic parameters between the left and right cameras, and $\mathbf{f}_j^S$, $\mathbf{f}_j^L$, and $\mathbf{f}_j^R$ are the stereo and mono, left and right, features.

choices (e.g., optimization window-size and extrinsics representation) affect the VINS performance.

- We compare our stereo-VINS algorithm against two state-of-the-art systems: i) OKVIS [13] (an open-source implementation of [7]) and ii) ORB-SLAM2 [14] (a vision-only stereo SLAM system with loop-closures); and demonstrate its superior performance.

The rest of this paper is structured as follows: In Sec. II, we briefly overview the key components of the proposed VINS, while Sec. III describes the image-processing frontend. Sec. IV presents an overview of the estimation algorithm, highlighting the key differences between the duo and stereo systems. A theoretical explanation of the information gain in stereo, as compared to duo, is presented in Sec. V. Finally, experimental results over several datasets are shown in Sec. VI, while Sec. VII concludes the paper.

II. Vision-aided Inertial Navigation System

The key components of the proposed VINS (see Fig. 1) are briefly described hereafter.

A. System State

At each time step k, the system maintains the following state vector:

$$\mathbf{x}'_k = \begin{bmatrix}\mathbf{x}_S^T & \mathbf{x}_k^T\end{bmatrix}^T \tag{1}$$

$$\text{with} \quad \mathbf{x}_k = \begin{bmatrix}\mathbf{x}_{C_{k-M+1}}^T & \cdots & \mathbf{x}_{C_k}^T & \mathbf{x}_P^T & \mathbf{x}_{E_k}^T\end{bmatrix}^T \tag{2}$$

where $\mathbf{x}_S$ is the state vector of the current SLAM features being estimated and $\mathbf{x}_k$ is the state vector comprising all other current states. Here $\mathbf{x}_S = \begin{bmatrix}{}^{C_m^i}\mathbf{p}_{f_1}^T & \dots & {}^{C_m^i}\mathbf{p}_{f_n}^T\end{bmatrix}^T$, with ${}^{C_m^i}\mathbf{p}_{f_j}$, for $j = 1, \dots, n$, denoting the position of the point feature $\mathbf{f}_j$ in its first observing camera frame $\{C_m^i\}$, where for the j^{th} feature, m is the time step of its first observing camera frame and $i = 0, 1$, is its first observing camera index (0 = left, 1 = right). For stereo features, if both cameras start observing the feature from the same time step, the left camera is assigned to be the first observing camera. Next, $\mathbf{x}_{C_p}$, for $p = k - M + 1, \dots, k$, represents the state vector of the cloned IMU poses at time step p, where M is the size of the sliding-window. We refer to the same stochastic cloning as in the MSCKF [5], for maintaining past IMU poses in a sliding-window estimator. Each cloned pose state is defined as

$$\mathbf{x}_{C_p} = \begin{bmatrix}{}^{I_p}\mathbf{q}_G^T & {}^{G}\mathbf{p}_{I_p}^T & t_{d_p}\end{bmatrix}^T \tag{3}$$

where ${}^{I_p}\mathbf{q}_G$ is the quaternion representing the orientation of the global frame $\{G\}$ in the IMU's frame of reference $\{I_p\}$, ${}^{G}\mathbf{p}_{I_p}$ is the position of $\{I_p\}$ in $\{G\}$, and t_{d_p} is the IMU-camera time offset (similar to the definition in [15]), at time step p. Next, the parameter state vector is defined as

$$\mathbf{x}_P = \begin{bmatrix}{}^{I}\mathbf{q}_{C_L}^T & {}^{I}\mathbf{p}_{C_L}^T & {}^{I}\mathbf{q}_{C_R}^T & {}^{I}\mathbf{p}_{C_R}^T\end{bmatrix}^T \tag{4}$$

where ${}^{I}\mathbf{q}_{C_L}$ and ${}^{I}\mathbf{q}_{C_R}$ are the quaternion representation of the orientations and ${}^{I}\mathbf{p}_{C_L}$ and ${}^{I}\mathbf{p}_{C_R}$ are the positions, of the left and right camera frames, $\{C_L\}$ and $\{C_R\}$, in the IMU's frame of reference $\{I\}$. An alternative representation of $\mathbf{x}_P$ is also considered, which consists of the left camera-IMU extrinsic and the left-right camera-camera extrinsic $({}^{C_L}\mathbf{q}_{C_R}, {}^{C_L}\mathbf{p}_{C_R})$.

$$\mathbf{x}_P = \begin{bmatrix}{}^{I}\mathbf{q}_{C_L}^T & {}^{I}\mathbf{p}_{C_L}^T & {}^{C_L}\mathbf{q}_{C_R}^T & {}^{C_L}\mathbf{p}_{C_R}^T\end{bmatrix}^T \tag{5}$$

In Sec. VI-G, we present a detailed comparison of these two representations, which supports selecting the later. Finally, $\mathbf{x}_{E_k}$ stores the current IMU biases and speed.

$$\mathbf{x}_{E_k} = \begin{bmatrix}\mathbf{b}_{g_k}^T & {}^{G}\mathbf{v}_{I_k}^T & \mathbf{b}_{a_k}^T\end{bmatrix}^T \tag{6}$$

where $\mathbf{b}_{g_k}$ and $\mathbf{b}_{a_k}$ correspond to the gyroscope and accelerometer biases, respectively, and ${}^{G}\mathbf{v}_{I_k}$ is the velocity of $\{I_k\}$ in $\{G\}$, at time step k.

The error state $\widetilde{\mathbf{x}}$ is defined as the difference between the true state $\mathbf{x}$ and the state estimate $\hat{\mathbf{x}}$ employed for linearization (i.e., $\widetilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$), while for the quaternion $\mathbf{q}$ a multiplicative error model $\widetilde{\mathbf{q}} = \mathbf{q} \otimes \hat{\mathbf{q}}^{-1} \simeq \begin{bmatrix}\frac{1}{2}\delta\boldsymbol{\theta}^T & 1\end{bmatrix}^T$ is used, where $\delta\boldsymbol{\theta}$ is a minimal representation of the attitude error.

B. Inertial Measurement Equations and Cost Terms

Given inertial measurements $\mathbf{u}_{k,k+1} = \begin{bmatrix}\boldsymbol{\omega}_{m_k}^T & \mathbf{a}_{m_k}^T\end{bmatrix}^T$, where $\boldsymbol{\omega}_{m_k}$ and $\mathbf{a}_{m_k}$ are gyroscope and accelerometer measurements, respectively, analytical integration of the continuous-time system equations (see [6]) within the time interval $\begin{bmatrix}t_k, & t_{k+1}\end{bmatrix}$ is used to determine the discrete-time system equations,

$$\mathbf{x}_{I_{k+1}} = \mathbf{f}(\mathbf{x}_{I_k}, \ \mathbf{u}_{k,k+1} - \mathbf{w}_{k,k+1}) \tag{7}$$

where $\mathbf{x}_{I_k} \triangleq \begin{bmatrix}\mathbf{x}_{C_k}^T & \mathbf{x}_{E_k}^T\end{bmatrix}^T$, and $\mathbf{w}_{k,k+1}$ is the discrete-time zero-mean white Gaussian noise affecting the IMU measurements with covariance $\mathbf{Q}_k$. Linearizing (7), around the state estimates $\hat{\mathbf{x}}_{I_k}$ and $\hat{\mathbf{x}}_{I_{k+1}}$, results in the cost term:

$$\begin{aligned}\mathcal{C}_u(\widetilde{\mathbf{x}}_{I_k}, \widetilde{\mathbf{x}}_{I_{k+1}}) = || \begin{bmatrix}\boldsymbol{\Phi}_{k+1,k} & -\mathbf{I}\end{bmatrix} \begin{bmatrix}\widetilde{\mathbf{x}}_{I_k} \\ \widetilde{\mathbf{x}}_{I_{k+1}}\end{bmatrix} \\ - (\hat{\mathbf{x}}_{I_{k+1}} - \mathbf{f}(\hat{\mathbf{x}}_{I_k}, \mathbf{u}_{k,k+1}))||^2_{\mathbf{Q}'_k}\end{aligned} \tag{8}$$

where $\mathbf{Q}'_k = \mathbf{G}_{k+1,k}\mathbf{Q}_k\mathbf{G}_{k+1,k}^T$, with $\boldsymbol{\Phi}_{k+1,k}$ and $\boldsymbol{G}_{k+1,k}$ being the corresponding IMU state and noise Jacobians.

C. Visual Measurement Equations and Cost Terms

Point features extracted from consecutive images are used as visual measurements to be processed by the estimator. The measurement model for the j^{th} feature in the i^{th} camera is

$$\mathbf{z}_k^{i,j} = \pi({}^{C_{k+t}^i}\mathbf{p}_{f_j}) + \mathbf{n}_k^{i,j} \tag{9}$$

where $\pi(.)$ is the camera projection model (including distortion), and ${}^{C_{k+t}^i}\mathbf{p}_{f_j}$ is the feature position expressed in the i^{th} ($i = 0$: left, 1 : right) camera's frame of reference at the

exact image-acquisition time instant $k+t$, $\mathbf{n}_k^{i,j}$ is zero-mean, white Gaussian noise with covariance $\sigma^2\mathbf{I}_2$. Linearizing (9) around the current state estimates yields:

$$\widetilde{\mathbf{z}}_k^{i,j} = \mathbf{H}_{x,k}^{i,j}\,\widetilde{\mathbf{x}}_k + \mathbf{H}_{f,k}^{i,j}\,{}^G\widetilde{\mathbf{p}}_{f_j} + \mathbf{n}_k^{i,j} \tag{10}$$

where $\mathbf{H}_{x,k}^{i,j}$ and $\mathbf{H}_{f,k}^{i,j}$ are the corresponding Jacobians evaluated at the state estimate $\hat{\mathbf{x}}_k$. For a monocular feature, stacking together all $N_j = N_{i,j}$ observations to it yields:

$$\widetilde{\mathbf{z}}^j = \mathbf{H}_x^j\widetilde{\mathbf{x}}_k + \mathbf{H}_f^j\,{}^G\widetilde{\mathbf{p}}_{f_j} + \mathbf{n}^j \tag{11}$$

Then, the measurements $\mathbf{z}^j$ contribute a linearized cost term:

$$\mathcal{C}_{z_j}(\widetilde{\mathbf{x}}_k, {}^G\widetilde{\mathbf{p}}_{f_j}) = ||\mathbf{H}_x^j\widetilde{\mathbf{x}}_k + \mathbf{H}_f^j\,{}^G\widetilde{\mathbf{p}}_{f_j} - \widetilde{\mathbf{z}}^j||^2_{\sigma^2\mathbf{I}_{2N_j}} \tag{12}$$

For a stereo feature, however, two sets of observations $N_{0,j}$ and $N_{1,j}$ come from each camera. Stacking together all such $N_j = N_{0,j} + N_{1,j}$ observations to this feature yields:

$$\widetilde{\mathbf{z}}^j = \begin{bmatrix}\widetilde{\mathbf{z}}^{0,j}\\ \widetilde{\mathbf{z}}^{1,j}\end{bmatrix} = \begin{bmatrix}\mathbf{H}_x^{0,j}\\ \mathbf{H}_x^{1,j}\end{bmatrix}\widetilde{\mathbf{x}}_k + \begin{bmatrix}\mathbf{H}_f^{0,j}\\ \mathbf{H}_f^{1,j}\end{bmatrix}{}^G\widetilde{\mathbf{p}}_{f_j} + \begin{bmatrix}\mathbf{n}^{0,j}\\ \mathbf{n}^{1,j}\end{bmatrix} \tag{13}$$

The corresponding linearized cost term becomes:

$$\mathcal{C}_{z_j}(\widetilde{\mathbf{x}}_k, {}^G\widetilde{\mathbf{p}}_{f_j}) = ||\mathbf{H}_x^j\widetilde{\mathbf{x}}_k + \mathbf{H}_f^j\,{}^G\widetilde{\mathbf{p}}_{f_j} - \widetilde{\mathbf{z}}^j||^2_{\sigma^2\mathbf{I}_{2N_j}} \tag{14}$$

$$= ||\begin{bmatrix}\mathbf{H}_x^{0,j}\\ \mathbf{H}_x^{1,j}\end{bmatrix}\widetilde{\mathbf{x}}_k + \begin{bmatrix}\mathbf{H}_f^{0,j}\\ \mathbf{H}_f^{1,j}\end{bmatrix}{}^G\widetilde{\mathbf{p}}_{f_j} - \begin{bmatrix}\widetilde{\mathbf{z}}^{0,j}\\ \widetilde{\mathbf{z}}^{1,j}\end{bmatrix}||^2_{\sigma^2\mathbf{I}_{2N_j}} \tag{15}$$

$$= \sum_{i=0}^{1} ||\mathbf{H}_x^{i,j}\widetilde{\mathbf{x}}_k + \mathbf{H}_f^{i,j}\,{}^G\widetilde{\mathbf{p}}_{f_j} - \widetilde{\mathbf{z}}^{i,j}||^2_{\sigma^2\mathbf{I}_{2N_{i,j}}} \tag{16}$$

D. Visual-Information Management

Three types of VINS systems are supported in our setup: i) mono, ii) duo, and iii) stereo. Where the mono system uses observations only from the left camera, the duo system processes measurements from both cameras independently, and the stereo system fuses observations from both cameras with a constraint between the commonly observed features. In all systems, we maintain two types of visual measurements: i) SLAM features and ii) MSCKF features, as in [5], [6], so as to provide high estimation accuracy while remaining computationally efficient. SLAM features are those that are added in the state vector (2) and updated across time. On the other hand, MSCKF features are those that are processed as in the MSC-KF approach [5], where the feature states are marginalized from the measurement equation (11) to generate constraints between poses. Note that, a similar approach to [6] is followed for classifying and maintaining the observed features into these two categories.

III. Image-processing Frontend

The proposed system extracts point features on consecutive images and tracks them across time. For each camera, feature extraction, tracking, and visual database management are performed independently. Two main categories of image processing frontends are explored: i) FAST [16] corner extraction and tracking using Kanade-Lucas-Tomasi (KLT) [17]-based optical flow and ii) difference of Gaussians (DOG) feature extraction with FREAK [18] descriptors and tracking using frame-to-frame matching. Henceforth, we will refer to them as KLT and FREAK frontends.

For both frontends, the feature tracking is aided with gyro-prediction (the frame-to-frame rotation estimate from the integration of the gyroscope measurements) and outliers are rejected using the 2-pt RANSAC [19]. After obtaining both left and right monocular feature tracks, stereo matching is performed. Lastly, another outlier rejection step is applied to the stereo matches and the remaining visual tracks are triangulated and prepared for the filter to process.

A. Stereo Matching

In our system, stereo matching is performed directly on the fisheye images. Where, the searched region is restricted on the curve corresponding to the epipolar line. To avoid exhaustive search along the whole epipolar line, it is assumed that the inverse depth of the features remains within an interval [0, ρ_{max}]. Also, to account for the inaccuracy of the stereo extrinsic calibration, the search region is expanded by 2 pixels on both sides, perpendicular to the epipolar line.

For the two different feature extraction pipelines under consideration, different stereo matching strategies are employed. For the FREAK pipeline, the features are already associated with descriptors and descriptor-matching is used to find stereo correspondences. For the KLT pipeline, however, sum of squared difference (SSD)-based patch matching is employed. A line patch consisting of five equidistant points on the epipolar line is used for matching, as in [20].

B. Outlier Rejection

Our system employs a cascade of outlier rejection steps for the stereo features. Besides the epipolar constraint check in stereo matching (Sec. III-A), a re-projection error test is performed on the stereo matches. Then, the depths of the stereo features are estimated using 2-view stereo triangulation and matches are rejected if the estimated depth falls outside an interval, [d_{min}, d_{max}]. Additionally, a fast 1-pt RANSAC is employed to remove the remaining outlier matches.

Specifically, among the stereo features observed at both time steps $k-1$ and k, one feature is randomly picked and (17) is used to estimate the translation, ${}^{I_k}\mathbf{p}_{I_{k-1}}$, between the corresponding IMU frames:

$$\begin{aligned}{}^{I_k}\mathbf{p}_{I_{k-1}} &= {}^I\mathbf{p}_{C^i} + {}^I\mathbf{C}_{C^i}\,{}^{C^i_k}\mathbf{p}_f \\ &\quad - {}^{I_k}\mathbf{C}_{I_{k-1}}({}^I\mathbf{C}_{C^i}\,{}^{C^i_{k-1}}\mathbf{p}_f + {}^I\mathbf{p}_{C^i})\end{aligned} \tag{17}$$

where ${}^{C^i_k}\mathbf{p}_f$ and ${}^{C^i_{k-1}}\mathbf{p}_f$ are feature positions with respect to the i^{th} camera, at time steps k and $k-1$, respectively, ${}^I\mathbf{C}_{C^i}$ and ${}^I\mathbf{p}_{C^i}$ are the IMU-camera extrinsics for the i^{th} camera, and ${}^{I_k}\mathbf{C}_{I_{k-1}}$ is the rotation between IMU frames at time steps k and $k-1$, which is reliably known from the integration of the gyroscope measurements. If both cameras [i.e., (17) for i = 0 and i = 1] generate similar estimates of ${}^{I_k}\mathbf{p}_{I_{k-1}}$, the mean estimate is used to check how many of the stereo features satisfy (17). This process is repeated until enough inliers are found or maximum iterations are reached. Next, the features are subjected to two more layers of outlier rejection in triangulation and filtering (Sec. III-C and IV-B).

C. Triangulation and Preparation of Visual Tracks

In the stereo system, the visual feature tracks are classified into two categories: monocular and stereo. If within the current optimization window, any two left-right feature tracks

are linked by at-least 2 stereo matches, all measurements from the two tracks are combined into a single feature track, classified as stereo track. If any left track matches multiple right tracks, the right track with the maximum number of matches is chosen. The rest of the tracks from both cameras are then classified as monocular tracks. After classification, the features are triangulated using all observations to them from both cameras. During triangulation, the individual and mean re-projection errors of all observations in a track are checked. If the errors exceed a threshold, the track is rejected as an outlier. Additionally, for the stereo tracks, if the combined mean re-projection error is larger than the corresponding left-right track errors, the stereo track association is considered erroneous and the associated left-right tracks are re-classified and processed as monocular.

IV. Estimation Algorithm

In this section, the main steps of the SR-ISWF algorithm are briefly described. Though the proposed system supports both SLAM and MSCKF features, for brevity and clarity of presentation we focus only on the MSCKF features.

At each time step k, the objective is to minimize the cost term $\mathcal{C}_k^{\oplus}$ that contains all the information available so far:

$$\mathcal{C}_k^{\oplus} = \mathcal{C}_{k-1} + \mathcal{C}_u + \mathcal{C}_{\mathbb{Z}_M} \tag{18}$$

where $\mathcal{C}_u$ [see (8)] represents the cost term arising from the IMU measurement $\mathbf{u}_{k-1,k}$, and $\mathcal{C}_{\mathbb{Z}_M} = \sum_{j=1}^{N_M} \mathcal{C}_{z_j}$ [see (12)] from the visual measurements to the N_M MSCKF features. In the mono system, $\mathcal{C}_{\mathbb{Z}_M}$ consists of the feature observations from the left camera only, hence equals $\mathcal{C}_{\mathbb{Z}_{M_0}}$. In the duo system, however, it incorporates observations from both cameras, i.e., $\mathcal{C}_{\mathbb{Z}_M} = \mathcal{C}_{\mathbb{Z}_{M_0}} + \mathcal{C}_{\mathbb{Z}_{M_1}}$. Lastly, for the stereo system, $\mathcal{C}_{\mathbb{Z}_M}$ comprises of monocular observations from each camera, as well as, stereo observations of features viewed by both cameras, i.e., $\mathcal{C}_{\mathbb{Z}_M} = \mathcal{C}_{\mathbb{Z}_{M_0}} + \mathcal{C}_{\mathbb{Z}_{M_1}} + \mathcal{C}_{\mathbb{Z}_{M_s}}$, where $\mathcal{C}_{\mathbb{Z}_{M_s}}$ is the cost term due to the N_s stereo observations, that is

$$\mathcal{C}_{\mathbb{Z}_{M_s}} = \sum_{i=0}^{1}\sum_{j=1}^{N_s} \mathcal{C}_{z_{i,j}}(\mathbf{f}_{i,j}) \qquad \text{s.t.} \quad \mathbf{f}_{0,j} \equiv {}^0\mathbf{T}_1 \oplus \mathbf{f}_{1,j} \tag{19}$$

where ${}^0\mathbf{T}_1$ is the transformation between the two camera frames and $\mathbf{f}_{i,j}$ is the j^{th} feature observed by the i^{th} camera. All the prior information obtained from the previous time step is contained in

$$\mathcal{C}_{k-1}(\widetilde{\mathbf{x}}_{k-1}) = \|\mathbf{R}_{k-1}\widetilde{\mathbf{x}}_{k-1} - \mathbf{r}_{k-1}\|^2 \tag{20}$$

where $\mathbf{R}_{k-1}$ and $\mathbf{r}_{k-1}$ are the prior information *factor* matrix and residual vector, respectively, and $\widetilde{\mathbf{x}}_{k-1} \triangleq \mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}$ is the error state from time step $k-1$ [see (2)].

A. Propagation

In the sliding-window, a new pose state $\mathbf{x}_{I_k}$ [see (7)] is appended to the current state vector at each time step k:

$$\mathbf{x}_k^{\ominus} = \begin{bmatrix}\mathbf{x}_{k-1}^T & \mathbf{x}_{I_k}^T\end{bmatrix}^T \tag{21}$$

using the IMU measurement $\mathbf{u}_{k-1,k}$. Hence, as in [6], the cost term, which initially comprised only $\mathcal{C}_{k-1}$, becomes

$$\begin{aligned}\mathcal{C}_k^{\ominus}(\widetilde{\mathbf{x}}_k^{\ominus}) &= \mathcal{C}_{k-1}(\widetilde{\mathbf{x}}_{k-1}) + \mathcal{C}_u(\widetilde{\mathbf{x}}_{I_{k-1}}, \widetilde{\mathbf{x}}_{I_k}) \\ &= \|\mathbf{R}_k^{\ominus}\widetilde{\mathbf{x}}_k^{\ominus} - \mathbf{r}_k^{\ominus}\|^2\end{aligned} \tag{22}$$

B. Marginalization and Covariance Factor Recovery

To maintain constant computational complexity, at time step k, the SR-ISWF marginalizes the oldest clone $\widetilde{\mathbf{x}}_{C_{k-M}}$, and the extra IMU states $\widetilde{\mathbf{x}}_{E_{k-1}}$ from the previous time step. For marginalization, the same strategy as in [6] is used with the resultant cost term after marginalization being:

$$\mathcal{C}_k^M(\widetilde{\mathbf{x}}_k) = \min_{\widetilde{\mathbf{x}}_k^M} \; \mathcal{C}_k^{\ominus}(\widetilde{\mathbf{x}}_k^M, \widetilde{\mathbf{x}}_k) = \left\|\mathbf{R}_k^R\widetilde{\mathbf{x}}_k - \mathbf{r}_k^R\right\|^2 \tag{23}$$

where $\widetilde{\mathbf{x}}_k^M$ are the marginalized states, $\widetilde{\mathbf{x}}_k$ are the remaining states after marginalization [see (2)], and $\mathbf{R}_k^R$ is upper-triangular. Next, as in [6], the covariance factor is recovered and used in Mahalanobis-distance-test-based outlier rejection.

C. Update

During the update step the pose-constraint information from the MSCKF feature measurements are incorporated.

1) Mono System Update: Following [6], first an orthonormal factorization is performed on the cost term in (12), to split it into two terms: i) $\mathcal{C}_{z_j^1}$, containing information about the feature's position and ii) $\mathcal{C}_{z_j^2}$, containing information about the poses which observed the feature, i.e.,

$$\mathcal{C}_{z_j}(\widetilde{\mathbf{x}}_k, {}^G\widetilde{\mathbf{p}}_{f_j}) = ||\mathbf{F}_1^j\widetilde{\mathbf{x}}_k + \mathbf{R}_f^{j\,G}\widetilde{\mathbf{p}}_{f_j} - \widetilde{\mathbf{z}}_1^j||^2_{\sigma^2\mathbf{I}_3} \tag{24}$$

$$+ ||\mathbf{F}_2^j\widetilde{\mathbf{x}}_k - \widetilde{\mathbf{z}}_2^j||^2_{\sigma^2\mathbf{I}_{2N_j-3}} \tag{25}$$

$$= \mathcal{C}_{z_j^1}(\widetilde{\mathbf{x}}_k, {}^G\widetilde{\mathbf{p}}_{f_j}) + \mathcal{C}_{z_j^2}(\widetilde{\mathbf{x}}_k) \tag{26}$$

where $\mathbf{F}_l^j$, $\mathbf{R}_f^j$, and $\widetilde{\mathbf{z}}_l^j$ $(l = 1, 2)$ are the corresponding Jacobians and residuals. Next, the feature states are marginalized by dropping $\mathcal{C}_{z_j^1}$ while the pose-constraints from the $\mathcal{C}_{z_j^2}$ terms are incorporated in the cost term $\mathcal{C}_k^M$. To do so, the Jacobians are stacked together and, as in [6], a thin-QR factorization [21] is performed, resulting in the cost term:

$$\mathcal{C}_k^{\oplus}(\widetilde{\mathbf{x}}_k) = \mathcal{C}_k^M(\widetilde{\mathbf{x}}_k) + \mathcal{C}_{\mathbb{Z}_M}(\widetilde{\mathbf{x}}_k) = \|\mathbf{R}_k^{\oplus}\widetilde{\mathbf{x}}_k - \mathbf{r}_k^{\oplus}\|^2 \tag{27}$$

where $\mathcal{C}_{\mathbb{Z}_M}(\widetilde{\mathbf{x}}_k) = \sum_{j=1}^{N_M} \mathcal{C}_{z_j^2}(\widetilde{\mathbf{x}}_k)$, with N_M being the number of MSCKF features. Finally, (27) is minimized with respect to the error state vector and the solution for $\widetilde{\mathbf{x}}_k$ is used to update the state.

$$\min_{\widetilde{\mathbf{x}}_k} \mathcal{C}_k^{\oplus}(\widetilde{\mathbf{x}}_k) = \min_{\widetilde{\mathbf{x}}_k} \|\mathbf{R}_k^{\oplus}\widetilde{\mathbf{x}}_k - \mathbf{r}_k^{\oplus}\|^2 \tag{28}$$

$$\hat{\mathbf{x}}_k^{\oplus} = \hat{\mathbf{x}}_k + \widetilde{\mathbf{x}}_k \tag{29}$$

2) Duo System Update: As compared to the mono system, the only difference here is that, the cost term $\mathcal{C}_{\mathbb{Z}_M}$ comprises two sets of measurements, one from each camera:

$$\mathcal{C}_{\mathbb{Z}_M}(\widetilde{\mathbf{x}}_k) = \sum_{i=0}^{1}\sum_{j=1}^{N_{M_i}} \mathcal{C}_{z_{i,j}^2}(\widetilde{\mathbf{x}}_k) \tag{30}$$

where N_{M_i} is the number of features for the i^{th} camera.

3) Stereo System Update: For the stereo system, there are two types of visual measurements: monocular and stereo. The monocular measurements can come from either camera and are processed independently in the same way as in duo. For the stereo measurements in (13), however, the cost terms (14) come from both cameras. The combined Jacobians $\mathbf{H}_f^j$ are then factorized similarly as in mono (26). In our optimized implementation of the factorization, we stack

together the measurement Jacobians in (15) and maintain the block-diagonal shape of $\mathbf{H}_x^j$, by interleaving the Jacobians from each camera in the clone order that they were observed.

V. Information Gain from Duo to Stereo

For features that are observed from both cameras, the transition from duo to stereo systems is due to the addition of the constraint in (19), i.e., that the two sets of observations correspond to the same feature. Such additional constraint results in information gain. To show that, we start with the cost term in (14) and apply a two-step factorization. Similar to mono, the two $\mathbf{H}_f^{i,j}$ terms in (16) are first factorized:

$$\begin{aligned}\mathcal{C}_{z_j}(\widetilde{\mathbf{x}}_k, {}^G\widetilde{\mathbf{p}}_{f_j}) &= \sum_{i=0}^{1} ||\mathbf{F}_1^{i,j}\widetilde{\mathbf{x}}_k + \mathbf{R}_f^{i,j\,G}\widetilde{\mathbf{p}}_{f_j} - \widetilde{\mathbf{z}}_1^{i,j}||^2_{\sigma^2\mathbf{I}_3} \\ &\quad + \sum_{i=0}^{1} ||\mathbf{F}_2^{i,j}\widetilde{\mathbf{x}}_k - \widetilde{\mathbf{z}}_2^{i,j}||^2_{\sigma^2\mathbf{I}_{2N_{i,j}-3}} \\ &= \mathcal{C}_{z_j^1}(\widetilde{\mathbf{x}}_k, {}^G\widetilde{\mathbf{p}}_{f_j}) + \mathcal{C}_{z_j^2}(\widetilde{\mathbf{x}}_k) \qquad (31)\end{aligned}$$

where $\mathbf{F}_l^{i,j}$, $\mathbf{R}_f^{i,j}$, and $\widetilde{\mathbf{z}}_l^{i,j}(l=1,2)$ are the corresponding Jacobians and residuals. In what follows, the two Jacobians $\mathbf{R}_f^{i,j}$ are then combined into $\mathbf{J}_f^j$ [see (32)] and another orthonormal factorization is performed on it to absorb the remaining information regarding the poses from the stereo measurements. To do so, a square orthonormal matrix $\mathbf{Q}_j$, partitioned as $\mathbf{Q}_j = \begin{bmatrix}\mathbf{P}_j & \mathbf{V}_j\end{bmatrix}$ is considered, where the 3 columns of $\mathbf{P}_j$ span the column space of $\mathbf{J}_f^j$, while the 3 columns of $\mathbf{V}_j$, are its left nullspace.

$$\begin{aligned}\mathcal{C}_{z_j^1}(\widetilde{\mathbf{x}}_k, {}^G\widetilde{\mathbf{p}}_{f_j}) &= || \begin{bmatrix}\mathbf{F}_1^{0,j}\\ \mathbf{F}_1^{1,j}\end{bmatrix}\widetilde{\mathbf{x}}_k + \begin{bmatrix}\mathbf{R}_f^{0,j}\\ \mathbf{R}_f^{1,j}\end{bmatrix}{}^G\widetilde{\mathbf{p}}_{f_j} - \begin{bmatrix}\widetilde{\mathbf{z}}_1^{0,j}\\ \widetilde{\mathbf{z}}_1^{1,j}\end{bmatrix}||^2_{\sigma^2\mathbf{I}_6} \\ &= ||\mathbf{J}_x^j\widetilde{\mathbf{x}}_k + \mathbf{J}_f^{j\,G}\widetilde{\mathbf{p}}_{f_j} - \widetilde{\mathbf{z}}_1^j||^2_{\sigma^2\mathbf{I}_6} \qquad (32) \\ &= ||\mathbf{Q}_j^T\mathbf{J}_x^j\widetilde{\mathbf{x}}_k + \mathbf{Q}_j^T\mathbf{J}_f^{j\,G}\widetilde{\mathbf{p}}_{f_j} - \mathbf{Q}_j^T\widetilde{\mathbf{z}}_1^j||^2_{\sigma^2\mathbf{I}_6} \\ &= ||\mathbf{P}_j^T\mathbf{J}_x^j\widetilde{\mathbf{x}}_k + \mathbf{P}_j^T\mathbf{J}_f^{j\,G}\widetilde{\mathbf{p}}_{f_j} - \mathbf{P}_j^T\widetilde{\mathbf{z}}_1^j||^2_{\sigma^2\mathbf{I}_3} \\ &\quad + ||\mathbf{V}_j^T\mathbf{J}_x^j\widetilde{\mathbf{x}}_k - \mathbf{V}_j^T\widetilde{\mathbf{z}}_1^j||^2_{\sigma^2\mathbf{I}_3} \\ &= ||\mathbf{F}_3^j\widetilde{\mathbf{x}}_k + \mathbf{R}_f^{j\,G}\widetilde{\mathbf{p}}_{f_j} - \widetilde{\mathbf{z}}_3^j||^2_{\sigma^2\mathbf{I}_3} + ||\mathbf{F}_4^j\widetilde{\mathbf{x}}_k - \widetilde{\mathbf{z}}_4^j||^2_{\sigma^2\mathbf{I}_3} \qquad (33) \\ &= \mathcal{C}_{z_j^3}(\widetilde{\mathbf{x}}_k, {}^G\widetilde{\mathbf{p}}_{f_j}) + \mathcal{C}_{z_j^4}(\widetilde{\mathbf{x}}_k) \qquad (34)\end{aligned}$$

$$\begin{aligned}\text{with}\quad \mathbf{F}_3^j &\triangleq \mathbf{P}_j^T\mathbf{J}_x^j \quad \mathbf{F}_4^j \triangleq \mathbf{V}_j^T\mathbf{J}_x^j \quad \mathbf{R}_f^j \triangleq \mathbf{P}_j^T\mathbf{J}_f^j \\ \widetilde{\mathbf{z}}_3^j &\triangleq \mathbf{P}_j^T\widetilde{\mathbf{z}}_1^j \quad \widetilde{\mathbf{z}}_4^j \triangleq \mathbf{V}_j^T\widetilde{\mathbf{z}}_1^j \qquad (35)\end{aligned}$$

Hence, (31) becomes:

$$\mathcal{C}_{z_j}(\widetilde{\mathbf{x}}_k, {}^G\widetilde{\mathbf{p}}_{f_j}) = \mathcal{C}_{z_j^3}(\widetilde{\mathbf{x}}_k, {}^G\widetilde{\mathbf{p}}_{f_j}) + \mathcal{C}_{z_j^4}(\widetilde{\mathbf{x}}_k) + \mathcal{C}_{z_j^2}(\widetilde{\mathbf{x}}_k)$$

Here, $\mathcal{C}_{z_j^2}$ and $\mathcal{C}_{z_j^3}$ are similar to the cost terms in (26) and $\mathcal{C}_{z_j^4}$ is the cost term that conveys the extra information factor $\mathbf{F}_4^j$ obtained from the stereo features, as compared to duo.

It is well known that any monocular system has seven unobservable dof (3 for global translation, 3 for global orientation, and 1 for scale), while the same is true for the duo system. The stereo system, however, directly provides metric information and hence makes the scale observable. As evident, the additional factor $\mathbf{F}_4^j$ is the one providing the scale information.[2]

[2]A formal proof of the directions along which $\mathbf{F}_4^j$ provides information is omitted due to lack of space.

VI. Experimental Results

In this section, a brief description of the experimental setup is provided, followed by a performance assessment of the mono, duo, and stereo VINS, using different image-processing frontends and design choices. The capability of the proposed stereo system for real-time operation on a commercial-grade mobile processor is also assessed. Lastly, comparisons to alternative approaches are provided.

A. Experimental Setup

For our experiments, the wide stereo setup of Fig. 2 was used. The stereo-rig is equipped with two Chameleon-2 camera sensors, with PT-02118BMP lenses. The cameras are global shutter and the lenses are fixed focal fisheye lenses with 165° field of view (FOV). The baseline between the cameras is 26 cm and they capture 640×480 images at 15 Hz. A commercial grade Invensense MPU-9250 IMU is used, which reports inertial measurements at 100 Hz. The cameras are triggered simultaneously by the IMU. The full pipeline runs real-time on the NVIDIA Jetson TK1 [2] board, which is equipped with a Tegra TK1 mobile processor, featuring a Kepler GPU and a quad-core ARM Cortex-A15 CPU.

B. Datasets and Ground-truth Estimation

In what follows, we first present our evaluation results on 7 of our indoor datasets.[3] To assess the effect of various conditions on certain pipelines, these were recorded under different lighting conditions, motion profiles, and scenes. The datasets are classified into three categories:

- Regular (Datasets 1-3): The purpose of these datasets is to assess how much drift the filter accumulates in a regular scene over a long period of time. These are long (0.8 km, 1.1 km, and 1.1 km), multi-floor, mostly texture-rich, and well-lit, with few relatively darker and feature-deprived stairways.
- Textureless (Datasets 4-5): The purpose of these datasets is to assess the robustness of different image-processing frontends in challenging scenes. These are also long (1.2 km and 1.1 km), single floor, and composed of dark, long, textureless corridors.
- Fast-motion (Datasets 6-7): The purpose of these is to assess the robustness of the image-processing frontends to fast motions. These are short (105.4 m and 88.4 m long), but contain very fast and arbitrary motions.

Furthermore, we present results for the EuRoC MAV datasets [22], which are classified into the i) Machine Hall (MH) and ii) VICON room datasets.

As a measure of positioning accuracy, the percentage of root mean square error (RMSE) over the distance travelled is used. For ground-truth, the batch least squares (BLS) solution of [23] with all visual (from both cameras) and inertial measurements is used. The stereo BLS implementation performs loop-closures and hence accumulates negligible drift over long trajectories (RMSE is around 0.05%).

[3]The datasets are available at this link:
`http://mars.cs.umn.edu/research/stereo_vins.php`

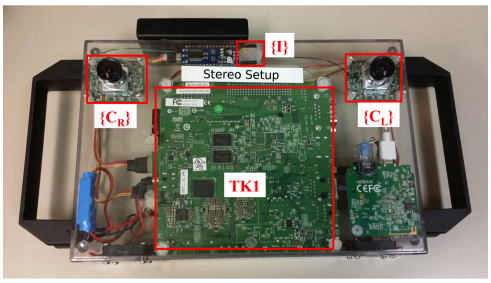

Fig. 2. Experimental setup.

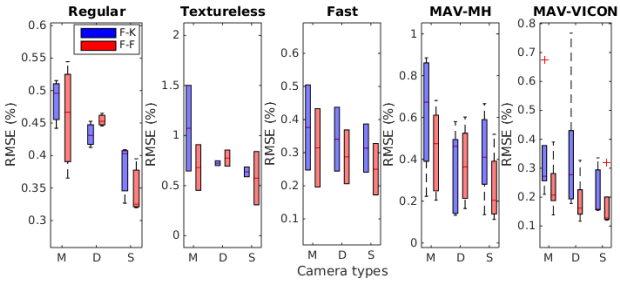

Fig. 3. Comparison between mono, duo, and stereo VINS using both FAST-KLT (F-K) and FREAK-to-FREAK (F-F) pipelines, for both our datasets and MAV datasets (M = mono, D = duo, S = stereo).

C. Comparison Between Mono, Duo, and Stereo VINS

The goal of this experiment is to assess the accuracy of the proposed stereo system, as well as to compare it against its monocular and binocular counterparts. For a fair comparison, the same feature budget (the number of features processed in each update) was used in both mono and stereo systems (20 SLAM features and 50 MSCKF features). For the duo system, the same features in the stereo were used, while dropping the stereo constraints. The sliding-window size, M, was set to 5. The comparison results for both image-processing frontends are presented in Fig. 3 as a box-and-whisker plot. Fig. 3 indicates that from mono to duo the estimation accuracy usually increases, since additional measurements from a second camera are included. The duo setup, however, does not always guarantee an improvement in performance, as is in the case of the FREAK pipeline in the low-texture datasets. On the other hand, from duo to stereo a significant performance gain is always observed, confirming the findings of Sec. V.

D. Impact of Different Image-processing Frontends

In the proposed system, two categories of image-processing frontends, KLT-based and FREAK-matching-based, are considered. The key benefit of the KLT pipeline is that it extracts better geometric information by providing longer feature tracks. Such tracks, however, could drift over time, causing significant performance degradation, especially in the case of stereo. The FREAK pipeline, on the other hand, usually generates drift-free tracks since FREAK DOGs are relatively well localized and have distinguishable descriptors associated with them. But such features suffer from shorter track length, as the DOG extrema have low repeatability in consecutive image frames. In this section, we will assess which one of these factors (having longer vs. drift-free tracks) affects the estimation accuracy more.

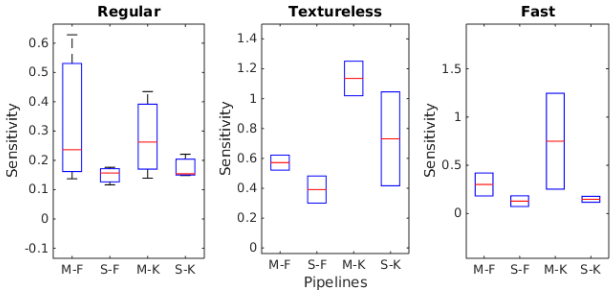

Fig. 4. Sensitivity of different pipelines (M-F = mono FREAK, S-F = stereo FREAK, M-K = mono KLT, S-K = stereo KLT) to changes in the maximum track length.

We start by noting that, although in the regular datasets KLT achieves very long feature tracks (8.1 vs. 3.7 for FREAK), the track length drops close to the FREAK tracks for the textureless and fast-motion datasets (3.8 vs. 3.3). On the other hand, the FREAK-matching pipeline uses descriptors that are more invariant to scene changes, hence maintaining almost a constant average track length over all types of datasets. Also, being a matching-based approach, large displacement of features due to fast-motion do not hurt the tracking performance as much, where KLT struggles to capture such large displacements. Consequently, as shown in Fig. 3, both mono and stereo systems benefit significantly by using the FREAK pipeline. For the duo pipeline, the FREAK frontend either performs better (for fast motion) or comparable to KLT. These results suggest that, although track length is an important factor for acquiring better geometric information, having drift-free tracks (though shorter) is even more important.

E. Sensitivity with respect to Maximum Track Length

To improve our understanding of how track length affects the estimation performance, a sensitivity analysis is performed on both monocular and stereo VINS systems with respect to changes in the maximum track-length. We compared both KLT and FREAK-based feature trackers, while varying the maximum allowed track length from 4 to 10 (at 5 Hz cloning). The track length sensitivity is defined as the maximum variation in RMSE due to the variation in the maximum track length. The results shown in Fig. 4 indicate that, compared to the FREAK frontend, KLT has a higher sensitivity to changes in track length, justifying the worse performance of KLT in challenging datasets (see Fig. 3). Also, as compared to stereo, the mono system shows more sensitivity towards changes in track length since for monocular features the triangulation baseline depends heavily on the track length, where, stereo features typically have sufficient baseline irrespective of the track length.

F. Impact of Sliding-Window Size

In this experiment, the influence of the sliding-window size, M, on the estimation accuracy is assessed. The error distribution, while varying M (5, 7, and 10), is presented in Fig. 5 for different pipelines and datasets. In terms of processing time, $M = 5$ is around 1.5 (than $M = 7$) to 2.1 (than $M = 10$) times faster. As expected, in the regular

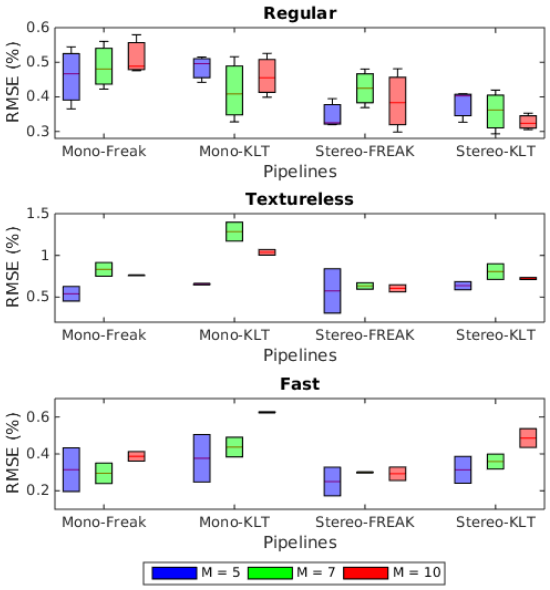

Fig. 5. Performance comparison of different pipelines by varying the number of clones in the sliding-window, M.

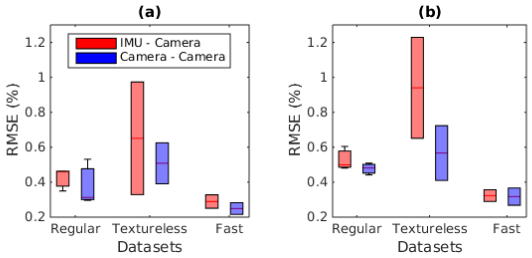

Fig. 6. Impact of the two representations of the right camera extrinsic, with respect to i) IMU (IMU-Camera) and ii) left camera (Camera-to-Camera), for (a) stereo and (b) duo systems.

TABLE I
COMPARISON: TIMING RESULTS PER FILTER UPDATE (MS)

Pipelines	Filter Update	Total Pipeline
Mono-KLT	12.6362	36.1116
Mono-FREAK	12.3767	52.6017
Mono-GPUFREAK	9.41638	39.9897
Duo-KLT	14.3675	74.5584
Duo-FREAK	17.5211	94.4892
Duo-GPUFREAK	13.543	65.2674
Stereo-KLT	14.6614	93.2688
Stereo-FREAK	17.7651	98.9529
Stereo-GPUFREAK	13.5359	70.7481

datasets with the KLT frontend, the error drops as the sliding-window size increases, for both mono and stereo systems. Interestingly, the opposite happens for the FREAK pipeline and also for the KLT pipeline in the textureless and the fast-motion datasets. In such cases the average track length is close to or less than 4 clones, hence, even if the window size is increased, on average only few features have long enough tracks to benefit from the longer optimization window. Also, by doubling the optimization window size (from 5 to 10), the condition number in such cases usually increases by a factor of 1.4, which might contribute to this loss of performance.

G. Impact of Extrinsic Representations

As stated in Sec. II-A, the sensors' extrinsics can be represented in two ways [see (4) and (5)]. The most commonly used option is (4), for its ease of implementation (especially in multi-camera systems). By employing this representation, however, the two IMU-camera extrinsics are optimized independently, without posing any constraint on the camera-to-camera extrinsics. On the other hand, in (5) the camera-to-camera extrinsics are explicitly represented, allowing them to have a strong initial prior (for stereo, the camera-to-camera extrinsics can be estimated offline very accurately) so that they do not vary rapidly during optimization.[4] Fig. 6 compares the performance of the stereo and duo systems using the FREAK frontend with both representations. As evident, (5) always outperforms (4), especially in the stereo system and regular-motion datasets.

H. Computational Performance

Table I compares the processing times in milliseconds (ms) of the mono, duo, and stereo systems for both KLT (NEON optimized) and FREAK (both NEON and CUDA optimized) frontends, running on NVIDIA Jetson TK1. For this test, M was set to 5 and the cloning rate was fixed to 5 Hz. Since same filter update budgets were maintained for all systems, the filter execution times were comparable (only 16-40% increase from mono to stereo). The duo and stereo systems, however, both have the overhead of an additional feature extraction and tracking step, followed by stereo matching (for the stereo system). Thus, their image-processing frontends consume most of the computational resources making their overall time almost double of the mono system. The image-processing frontend, however, can be offloaded to special-purpose hardware (e.g., Movidius VPU [24], Hexagon DSP [25]), in which case the gain in performance from mono to stereo will be realized with only minimal processing overhead.

I. Comparison with OKVIS and ORB-SLAM2

Fig. 7 compares the performance of the proposed stereo system (with the FREAK-matching frontend) against two state-of-the-art stereo SLAM systems: i) OKVIS [13] and ii) ORB-SLAM2 [14]. For the default configuration of [13] (maximum 400 keypoints, 5 keyframes, 3 most current frames), the median RMSE (% over distance travelled) on the MAV datasets was 0.21%, where the proposed algorithm (maximum 400 keypoints, window size 5) resulted in a median RMSE of 0.12%. For the datasets used in this paper, however, the difference was significantly larger: 1.62% median RMSE in OKVIS vs. 0.31% in ours. Since our datasets are mostly exploration type, OKVIS needs to select more frequent keyframes, causing shorter baselines and hence lower accuracy. In terms of computation time the difference is more pronounced: Our algorithm (for fairness we compare the non-CUDA frontend) performs 19.05 times (452.81 ms vs. 23.77 ms per frame, on MAV datasets) to 5.49 times (180.94 ms vs. 32.98 ms per frame, on our datasets) faster than OKVIS on the Jetson TK1. These timing results preclude the use of OKVIS on mobile processors, while our

[4]Among the 6 dof of the IMU-camera extrinsic calibration, the 3 dof of position are typically the ones less accurately estimated since acquiring information for them requires very fast rotations that cause image blur.

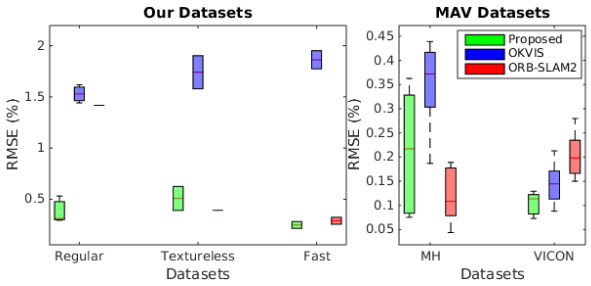

Fig. 7. Accuracy comparison with OKVIS and ORB-SLAM2.

proposed stereo system is capable of running in real-time at 10 Hz.

ORB-SLAM2, on the other hand, fails (loses track and fails to re-localize or re-localizes with a wrong loop-closure) in most of our long datasets (works only in 2 out of 5). It also performs worse in the fast-motion and MAV-VICON datasets. Only in the medium-scale MAV-MH datasets ORB-SLAM2 takes advantage of the loop-closures and and performs better than our algorithm. Processing these loop-closures and running a parallel bundle adjustment thread, however, costs ORB-SLAM2 in processing time. On Jetson TK1 it takes around 174 ms per frame, which is 5-7 times slower than our algorithm.

VII. Conclusions

This paper presents a two-camera extension and analysis of the SR-ISWF [6] for high-precision, real-time (up to 10 Hz cloning rate on mobile ARM processors) VINS. In particular, we provided a detailed comparison between the mono, duo, and stereo systems, along with a theoretical explanation of the superior performance of stereo over duo. Additionally, we assessed the robustness of different image-processing frontends and the importance of feature-drift over track-length. Moreover, we showed that descriptor-matching-based frontends are more robust than KLT-based, especially in challenging scenes and motion profiles. Also, we showed that although for regular scenes (with longer track lengths) a larger optimization window-size increases the estimation accuracy, for shorter tracks (due to challenging scenes or motions, or from the FREAK-matching frontend) both mono and stereo systems perform better with a smaller window size. Furthermore, for stereo systems, we also presented a novel outlier rejection strategy and an alternative extrinsic representation. Lastly, we demonstrated that the proposed stereo system outperforms OKVIS [13] and the ORB-SLAM2 [14] both in terms of accuracy and processing efficiency.

References

[1] Project Tango, https://get.google.com/tango/.

[2] NVIDIA Jetson TK1, http://www.nvidia.com/object/jetson-tk1-embedded-dev-kit.html.

[3] K. Konolige, M. Agrawal, and J. Sola, "Large-scale visual odometry for rough terrain," in *Robotics Research*. Springer, 2010, pp. 201–212.

[4] 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 *Proc. of the IEEE International Conference on Robotics and Automation*, Saint Paul, MN, May14 – 18 2012, pp. 957–964.

[5] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proc. of the IEEE International Conference on Robotics and Automation*, Rome, Italy, Apr. 10–14 2007, pp. 3482–3489.

[6] K. Wu, A. Ahmed, G. Georgiou, and S. I. Roumeliotis, "A square root inverse filter for efficient vision-aided inertial navigation on mobile devices," in *Proc. of Robotics: Science and Systems*, Rome, Italy, July 13 – 17 2015.

[7] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," *International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, Feb. 2015.

[8] J.-P. Tardif, M. George, M. Laverne, A. Kelly, and A. Stentz, "A new approach to vision-aided inertial navigation," in *Proc. of the IEEE International Conference on Intelligent Robots and Systems*, Taipei, Oct.18 – 22 2010, pp. 4161–4168.

[9] N. de Palzieux, T. Ngeli, and O. Hilliges, "Duo-vio: Fast, light-weight, stereo inertial odometry," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 2237–2242.

[10] T. Manderson, F. Shkurti, and G. Dudek, "Texture-aware SLAM using stereo imagery and inertial information," in *Proceedings of the 2016 Conference on Computer and Robot Vision*, May 2016.

[11] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE, 2007, pp. 225–234.

[12] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[13] OKVIS: Open Keyframe-based Visual-Inertial SLAM, http://ethz-asl.github.io/okvis.

[14] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras," *arXiv preprint arXiv:1610.06475*, 2016.

[15] C. Guo, D. G. Kottas, R. DuToit, A. Ahmed, R. Li, and S. I. Roumeliotis, "Efficient visual-inertial navigation using a rolling-shutter camera with inaccurate timestamps," in *Proc. of the Robotics: Science and Systems Conference*, Berkeley, CA, July 12 – 16 2014.

[16] M. Trajković and M. Hedley, "Fast corner detection," *Image and vision computing*, vol. 16, no. 2, pp. 75–87, feb 1998.

[17] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. of the International Joint Conference on Artificaial Intelligence*, Vancouver, British Columbia, Aug. 24–28 1981, pp. 674–679.

[18] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast retina keypoint," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, College Park, MD, June 16–21 2012, pp. 510–517.

[19] L. Kneip, M. Chli, and R. Siegwart, "Robust real-time visual odometry with a single camera and an IMU," in *Proc. of the British Machine Vision Conference*, Dundee, Scotland, August 29 - September 2 2011, pp. 16.1–16.11.

[20] J. Engel, J. Sturm, and D. Cremers, "Semi-dense visual odometry for a monocular camera," in *Proc. of the IEEE International Conference on Computer Vision*, Sydney, Australia, Dec.1 – 8 2013, pp. 1449–1456.

[21] G. Golub and C. Van Loan, *Matrix Computations*, 4th ed. Johns Hopkins University Press, 2013.

[22] 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*, vol. 35, no. 10, pp. 1157–1163, Sept. 2016.

[23] C. X. Guo, K. Sartipi, R. C. DuToit, G. A. Georgiou, R. Li, J. O'Leary, E. D. Nerurkar, J. A. Hesch, and S. I. Roumeliotis, "Large-scale cooperative 3D visual-inertial mapping in a Manhattan world," in *2016 IEEE International Conference on Robotics and Automation*, Stockholm, Sweden, May 16 – 21 2016, pp. 1071–1078.

[24] Movidius Vision Processing Unit (VPU), https://www.movidius.com/solutions/vision-processing-unit.

[25] Qualcomm Hexagon DSP Processor, https://developer.qualcomm.com/software/hexagon-dsp-sdk.