Alternating-Stereo VINS: Observability Analysis and Performance Evaluation

Mrinal K. Paul, Stergios I. Roumeliotis
Google Inc.
California, USA
{mrinalkanti, stergiosr}@google.com

Abstract

One approach to improve the accuracy and robustness of vision-aided inertial navigation systems (VINS) that employ low-cost inertial sensors, is to obtain scale information from stereoscopic vision. Processing images from two cameras, however, is computationally expensive and increases latency. To address this limitation, in this work, a novel two-camera alternating-stereo VINS is presented. Specifically, the proposed system triggers the left-right cameras in an alternating fashion, estimates the poses corresponding to the left camera only, and introduces a linear interpolation model for processing the alternating right camera measurements. Although not a regular stereo system, the alternating visual observations when employing the proposed interpolation scheme, still provide scale information, as shown by analyzing the observability properties of the vision-only corresponding system. Finally, the performance gain, of the proposed algorithm over its monocular and stereo counterparts is assessed using various datasets.

1. Introduction and Related Work

With the advent of augmented (AR) and virtual reality (VR), the application of vision-aided inertial navigation systems (VINS) on mobile devices is becoming increasingly popular. As a result, the research focus in VINS is gradually shifting towards finding accurate, yet efficient, real-time solutions on resource-constrained devices. Moreover, due to recent improvements in mobile processors (e.g., [3, 4]), and the availability of multiple cameras in certain smart-phones (e.g., [5]) and AR-VR headsets (e.g., [1, 2]), the interest in more robust multi-camera VINS is also increasing.

Most existing tightly-coupled (i.e., jointly optimizing over visual and inertial cost terms) VINS approaches focus on monocular systems (e.g., [7, 10, 20, 22, 23]). Although scale is observable in monocular VINS from the inertial measurement unit (IMU)'s accelerometer, it is typically imprecise as it requires accurately subtracting the dominant gravity vector from the noisy acceleration measurements. Thus, additional scale information from stereo vision is key to achieving higher accuracy. To this end, Manderson et al. [18] propose an extension of PTAM [14] where the tracking and mapping pipelines are decoupled. On the other hand, Leutenegger et al. [16] employ a keyframe-based simultaneous localization and mapping (SLAM) algorithm that performs nonlinear optimization. Both [18] and [16], however, operate in real-time only on desktop CPUs. To the best of our knowledge, Paul et al. [21] presents the only tightly coupled stereo VINS that operates in real-time on a mobile processor. In particular, [21] extends the inverse square-root sliding-window filter of [23] to two-camera systems and shows that the additional scale information obtained from the stereo visual observations is a key factor in improving accuracy and robustness. Due to the additional image-processing requirements for the second camera, however, [21] is able to process key-frames at up to 10 Hz, which is not sufficient for tracking fast motions in low-latency demanding applications such as VR and AR.

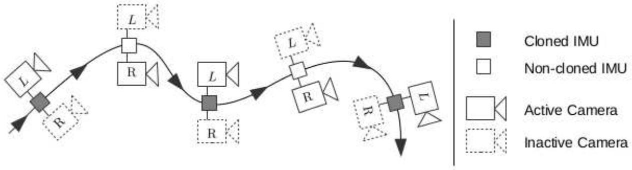


Figure 1. Alternating-stereo VINS.

To address the limitations of existing stereo systems, in this paper we present a novel alternating-stereo VINS which has CPU requirements and latency comparable to monocular VINS, yet provides scale information from the visual observations, hence achieving accuracy and robustness comparable to stereo VINS. In the proposed stereo system, the left-right cameras are triggered in an alternating fashion (see Fig. 1), while estimating the poses of the camera frame only when the left camera is active (i.e., every other image of the pair). Since, the observations from the right cameras do not correspond to any cloned frames,[1]

[1] By cloning we refer to the stochastic cloning as in the MSCKF [20]

a linear *interpolation-based motion model* is introduced to relate them to their temporally neighbouring cloned frames. By doing so, we are able to prove based on the observability analysis that the visual observations, in conjunction with the motion model, provide scale information. Additionally, we show that our system operates in real-time on mobile processors.

In summary, our main contributions are:

- We present the first alternating-stereo VINS, that combines the low-latency of a monocular VINS with the accuracy and robustness (from the visual scale information) of a stereo system. This is achieved by introducing an interpolation-based camera measurement model to process the alternating-camera observations.

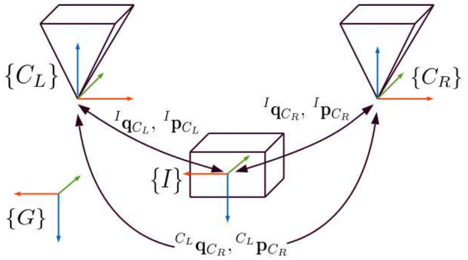
- We analyze the observability properties of the proposed alternating-stereo system when employing the interpolation scheme and show that the scale becomes observable with only visual observations.

- We perform a detailed comparison between the proposed system and its monocular and stereo counterparts, to assess its accuracy and robustness.

The rest of this paper is structured as follows: In Sec. 2, we briefly review the key components of the proposed VINS, highlighting the interpolation-based camera measurement model for the alternating observations. Sec. 3 describes the image-processing front-end, and Sec. 4 presents an overview of the estimation algorithm. In Sec. 5 we present the observability properties of the proposed system, in a vision-only setup, and show that scale becomes observable when employing the proposed interpolation scheme. Finally, experimental results over several datasets are shown in Sec. 6, while Sec. 7 concludes the paper.

2. System Description

The proposed system comprises two forward facing cameras with overlapping fields of view, where at each time step only one of the left-right cameras is capturing images. Specifically, the cameras are triggered in an alternating fashion (see Fig. 1), while cloning only on the left camera instants. The visual and inertial measurements are then fused in a tightly coupled manner, following the sliding-window approach of [23]. The key components of the proposed system (see Fig. 2) are briefly described hereafter.



$\{C_L\}$
$\{C_R\}$
${}^{I}\mathbf{q}_{C_L},\ {}^{I}\mathbf{p}_{C_L}$
${}^{I}\mathbf{q}_{C_R},\ {}^{I}\mathbf{p}_{C_R}$
$\{I\}$
$\{G\}$
${}^{C_L}\mathbf{q}_{C_R},\ {}^{C_L}\mathbf{p}_{C_R}$

Figure 2. Coordinate frames, 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, and $({}^{C_L}\mathbf{q}_{C_R}, {}^{C_L}\mathbf{p}_{C_R})$ are the left-right camera-to-camera extrinsics.

2.1. System State

At each time step k, the sliding window estimator maintains the following state vector:

$$\mathbf{x}_k = \begin{bmatrix}\mathbf{x}_S^T & \mathbf{x}_F^T\end{bmatrix}^T \tag{1}$$

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

where $\mathbf{x}_S$ contains the currently estimated SLAM features and $\mathbf{x}_F$ comprises all other current states. Here $\mathbf{x}_S = \begin{bmatrix}{}^{C_0}\mathbf{p}_{f_1}^T & \dots & {}^{C_0}\mathbf{p}_{f_n}^T\end{bmatrix}^T$, with ${}^{C_0}\mathbf{p}_{f_j}$, for $j = 1, \dots, n$, denoting the position of the feature $\mathbf{f}_j$ in its first observing camera frame $\{C_0\}$. If the feature is, however, first observed by the right camera $\{C_R\}$, it is represented with respect to the immediately previous left camera frame $\{C_L\}$. Next, $\mathbf{x}_{C_p}$, for $p = k - M + 1, \dots, k$, represents the state vector corresponding to the IMU poses at time step p, with M being the sliding-window size. Each pose state is defined as $\mathbf{x}_{C_p} = \begin{bmatrix}{}^{I_p}\mathbf{q}_G^T & {}^{G}\mathbf{p}_{I_p}^T\end{bmatrix}^T$, where ${}^{I_p}\mathbf{q}_G$ is the quaternion representing the orientation of the global frame $\{G\}$ in the IMU's frame of reference $\{I_p\}$, and ${}^{G}\mathbf{p}_{I_p}$ is the position of $\{I_p\}$ in $\{G\}$, at time step p. Next, the parameter state vector is defined as $\mathbf{x}_P = \begin{bmatrix}{}^{I}\mathbf{q}_{C_L}^T & {}^{I}\mathbf{p}_{C_L}^T\end{bmatrix}^T$, where $({}^{I}\mathbf{q}_{C_L}, {}^{I}\mathbf{p}_{C_L})$ are the extrinsic parameters between $\{C_L\}$ and $\{I\}$. The left-right camera-to-camera extrinsic parameters $({}^{C_L}\mathbf{q}_{C_R}, {}^{C_L}\mathbf{p}_{C_R})$ are, however, assumed to be known and, as shown in Sec. 5, contribute to the system scale. Finally, $\mathbf{x}_{E_k} = \begin{bmatrix}\mathbf{b}_{g_k}^T & \mathbf{b}_{a_k}^T & {}^{G}\mathbf{v}_{I_k}^T\end{bmatrix}^T$ contains gyroscope $\mathbf{b}_{g_k}$ and accelerometer $\mathbf{b}_{a_k}$ biases, as well as the velocity ${}^{G}\mathbf{v}_{I_k}$ of $\{I_k\}$ in $\{G\}$, at time step k.

Lastly, we apply an additive error model for any quantity $\mathbf{x}$ as $\widetilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$, where $\widetilde{\mathbf{x}}$ is the error state and $\hat{\mathbf{x}}$ is the state estimate employed for linearization. For a quaternion $\mathbf{q}$, however, a multiplicative error model is employed as $\widetilde{\mathbf{q}} = \mathbf{q} \otimes \hat{\mathbf{q}}^{-1} \simeq \begin{bmatrix}\frac{1}{2}\delta\boldsymbol{\theta}^T & 1\end{bmatrix}^T$, where $\otimes$ indicates quaternion multiplication and $\delta\boldsymbol{\theta}$ is a minimal representation of the attitude error.

for maintaining past IMU poses in a sliding window estimator. The cloned frames are analogous to key-frames in the computer vision literature.

2.2. Inertial Measurements 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, a constraint between the consecutive inertial states can be imposed (see [23]):

$$\mathbf{x}_{I_{k+1}} = \mathbf{f}(\mathbf{x}_{I_k},\ \mathbf{u}_{k,k+1} - \mathbf{w}_{k,k+1}) \tag{3}$$

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 (3) around the state estimates $\hat{\mathbf{x}}_{I_k}$ and $\hat{\mathbf{x}}_{I_{k+1}}$ yields the inertial cost term:

$$\begin{aligned}\mathcal{C}_u(\widetilde{\mathbf{x}}_{I_k}, \widetilde{\mathbf{x}}_{I_{k+1}}) = ||\, &\begin{bmatrix}\boldsymbol{\Phi}_{k+1,k} & -\mathbf{I}\end{bmatrix}\begin{bmatrix}\widetilde{\mathbf{x}}_{I_k}\\ \widetilde{\mathbf{x}}_{I_{k+1}}\end{bmatrix}\\ &- (\hat{\mathbf{x}}_{I_{k+1}} - \mathbf{f}(\hat{\mathbf{x}}_{I_k}, \mathbf{u}_{k,k+1}))||^2_{\mathbf{Q}'_k}\end{aligned} \tag{4}$$

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 Jacobians.

2.3. Visual Measurements and Cost Terms

The measurement model for the j^{th} feature in the i^{th} $(i = L:\text{ left})$ camera is

$$\mathbf{z}_k^{i,j} = \pi({}^{C_i^{k+t}}\mathbf{p}_{f_j}) + \mathbf{n}_k^{i,j} \tag{5}$$

where $\pi(.)$ is the camera projection model, ${}^{C_i^{k+t}}\mathbf{p}_{f_j}$ is the feature position expressed in the i^{th} camera's frame of reference at the image-acquisition time $k+t$, and $\mathbf{n}_k^{i,j}$ is zero-mean, white Gaussian noise with covariance $\sigma^2\mathbf{I}_2$. Linearizing (5) around the current state estimates yields:

$$\widetilde{\mathbf{z}}_k^{i,j} = \mathbf{H}_{x,k}^{i,j}\,\widetilde{\mathbf{x}}_F + \mathbf{H}_{f,k}^{i,j}\;{}^{C_0}\widetilde{\mathbf{p}}_{f_j} + \mathbf{n}_k^{i,j} \tag{6}$$

where $\mathbf{H}_{x,k}^{i,j}$ and $\mathbf{H}_{f,k}^{i,j}$ are the corresponding Jacobians. Stacking together all N_j observations to this feature yields:

$$\widetilde{\mathbf{z}}^j = \mathbf{H}_x^j\widetilde{\mathbf{x}}_F + \mathbf{H}_f^j\,{}^{C_0}\widetilde{\mathbf{p}}_{f_j} + \mathbf{n}^j \tag{7}$$

The corresponding linearized cost term becomes:

$$\mathcal{C}_{z_j}(\widetilde{\mathbf{x}}_F, {}^{C_0}\widetilde{\mathbf{p}}_{f_j}) = ||\mathbf{H}_x^j\widetilde{\mathbf{x}}_F + \mathbf{H}_f^j\,{}^{C_0}\widetilde{\mathbf{p}}_{f_j} - \widetilde{\mathbf{z}}^j||^2_{\sigma^2\mathbf{I}_{2N_j}} \tag{8}$$

2.3.1 Interpolation-based Camera Jacobians

In the proposed system, the left-right cameras are triggered in an alternating fashion, while cloning only when the left camera is active. Thus, the Jacobians for the left camera measurements are the same as those of a monocular system (see [13] and (7)). On the other hand, since the poses corresponding to the right camera time instants are *not* included in the state vector, a motion model is needed to relate these measurements to its adjacent cloned poses. For the motion model, we choose not to involve the IMU measurements since it requires including the accelerations and rotational velocities in the state vector, consequently increasing the memory and processing requirements. Instead, we employ an interpolation-based model that avoids such issues, while maintaining indistinguishable performance.

Specifically, assuming the sensor-pair moves on approximately a straight line segment during the very small time interval between two consecutive clones ($\sim$ 60 msec), the position of the IMU frame at the right camera time instant $k+t$ is linearly interpolated from its two temporally neighboring clone positions:

$${}^G\mathbf{p}_{I_{k+t}} = (1-\lambda){}^G\mathbf{p}_{I_k} + \lambda\,{}^G\mathbf{p}_{I_{k+1}} \tag{9}$$

where λ is the interpolation ratio. The IMU rotation from time instant k to $k+1$ is defined as $\mathbf{C}(\boldsymbol{\theta}_{k+1,k}) = {}^{I_{k+1}}_{\ \ G}\mathbf{C}\,{}^{I_k}_{G}\mathbf{C}^T$. Similarly, assuming a constant axis of rotation, the orientation of $\{I_{k+t}\}$ is then equivalent to rotating $\{I_k\}$ about $\lambda\boldsymbol{\theta}_{k+1,k}$, i.e.,

$${}^{I_{k+t}}_{\ \ G}\mathbf{C} = {}^{I_{k+t}}_{\ \ I_k}\mathbf{C}\,{}^{I_k}_{G}\mathbf{C} = \mathbf{C}(\lambda\boldsymbol{\theta}_{k+1,k})\,{}^{I_k}_{G}\mathbf{C} \tag{10}$$

By employing (9) and (10), we can express the right camera measurements as:

$$\begin{aligned}\mathbf{z}_k^{R,j} &= \pi({}^{C_R^{k+t}}\mathbf{p}_{f_j}) + \mathbf{n}_k^{R,j}\\ &= \pi({}^{C_R^{k+t}}_{\ \ C_0}\mathbf{C}\,{}^{C_0}\mathbf{p}_{f_j} - {}^{\ \ I}_{C_R}\mathbf{C}^T\,{}^I\mathbf{p}_{C_R}\\ &\quad - {}^{C_R^{k+t}}_{\ \ G}\mathbf{C}({}^G\mathbf{p}_{I_{k+t}} - {}^G\mathbf{p}_{I_0}) + {}^{C_R^{k+t}}_{\ \ I_0}\mathbf{C}\,{}^I\mathbf{p}_{C_L}) + \mathbf{n}_k^{R,j}\\ &= \pi({}^{\ \ I}_{C_R}\mathbf{C}^T\mathbf{C}(\lambda\boldsymbol{\theta}_{k+1,k})\,{}^{I_k}_{G}\mathbf{C}({}^{I_0}_{G}\mathbf{C}^T\,{}^{\ \ I}_{C_L}\mathbf{C}\,{}^{C_0}\mathbf{p}_{f_j}\\ &\qquad - (1-\lambda){}^G\mathbf{p}_{I_k} - \lambda\,{}^G\mathbf{p}_{I_{k+1}} + {}^G\mathbf{p}_{I_0}\\ &\qquad + {}^{I_0}_{G}\mathbf{C}^T\,{}^I\mathbf{p}_{C_L}) - {}^{\ \ I}_{C_R}\mathbf{C}^T\,{}^I\mathbf{p}_{C_R}) + \mathbf{n}_k^{R,j}\end{aligned} \tag{11}$$

where ${}^{\ \ I}_{C_R}\mathbf{C} \triangleq {}^{\ \ I}_{C_L}\mathbf{C}\,{}^{C_L}_{C_R}\mathbf{C}$ and ${}^I\mathbf{p}_{C_R} \triangleq {}^I\mathbf{p}_{C_L} + {}^{\ \ I}_{C_L}\mathbf{C}\,{}^{C_L}\mathbf{p}_{C_R}$. Linearizing (11), the measurement model corresponding to the right camera feature observations becomes:[2]

$$\begin{aligned}\widetilde{\mathbf{z}}_k^{R,j} &= \mathbf{H}_{\boldsymbol{\pi}}^{j,k}\Bigg(\mathbf{H}_f^{j,k}\;{}^{C_0}\widetilde{\mathbf{p}}_{f_j} + \begin{bmatrix}\mathbf{H}_{\mathbf{p}_{k+1}}^{j,k} & \mathbf{H}_{\boldsymbol{\theta}_{k+1}}^{j,k}\end{bmatrix}\begin{bmatrix}{}^G\widetilde{\mathbf{p}}_{I_{k+1}}\\ {}^{I_{k+1}}\widetilde{\boldsymbol{\theta}}_G\end{bmatrix}\\ &\quad + \begin{bmatrix}\mathbf{H}_{\mathbf{p}_k}^{j,k} & \mathbf{H}_{\boldsymbol{\theta}_k}^{j,k}\end{bmatrix}\begin{bmatrix}{}^G\widetilde{\mathbf{p}}_{I_k}\\ {}^{I_k}\widetilde{\boldsymbol{\theta}}_G\end{bmatrix} + \begin{bmatrix}\mathbf{H}_{\mathbf{p}_0}^{j,k} & \mathbf{H}_{\boldsymbol{\theta}_0}^{j,k}\end{bmatrix}\begin{bmatrix}{}^G\widetilde{\mathbf{p}}_{I_0}\\ {}^{I_0}\widetilde{\boldsymbol{\theta}}_G\end{bmatrix}\\ &\quad + \begin{bmatrix}\mathbf{H}_{\mathbf{p}_x}^{j,k} & \mathbf{H}_{\boldsymbol{\theta}_x}^{j,k}\end{bmatrix}\begin{bmatrix}{}^I\widetilde{\mathbf{p}}_{C_L}\\ {}^I\widetilde{\boldsymbol{\theta}}_{C_L}\end{bmatrix}\Bigg) + \mathbf{n}_k^{R,j} \qquad (12)\\ &= \mathbf{H}_{x,k}^{R,j}\,\widetilde{\mathbf{x}}_F + \mathbf{H}_{f,k}^{R,j}\;{}^{C_0}\widetilde{\mathbf{p}}_{f_j} + \mathbf{n}_k^{R,j} \qquad (13)\end{aligned}$$

where

$$\begin{aligned}\mathbf{H}_{\boldsymbol{\pi}}^{j,k} &= \frac{1}{\gamma_j^2}\begin{bmatrix}\gamma_j & 0 & -\alpha_j\\ 0 & \gamma_j & -\beta_j\end{bmatrix},\ {}^{C_R^{k+t}}\mathbf{p}_{f_j} \triangleq \begin{bmatrix}\alpha_j\\ \beta_j\\ \gamma_j\end{bmatrix}\\ \mathbf{H}_{\mathbf{p}_0}^{j,k} &= \mathbf{M} \triangleq {}^{\ \ I}_{C_R}\mathbf{C}^T\mathbf{C}(\lambda\boldsymbol{\theta})\,{}^{I_k}_{G}\mathbf{C}\\ \mathbf{H}_{\boldsymbol{\theta}_0}^{j,k} &= -\mathbf{M}\,{}^{I_0}_{G}\mathbf{C}^T\lfloor{}^{\ \ I}_{C_L}\mathbf{C}\,{}^{C_0}\mathbf{p}_{f_j} + {}^I\mathbf{p}_{C_L}\rfloor\\ \mathbf{H}_{\mathbf{p}_{k+1}}^{j,k} &= -\lambda\mathbf{M}, \qquad \mathbf{H}_{\mathbf{p}_k}^{j,k} = (\lambda-1)\mathbf{M}\end{aligned}$$

[2] We use IMU integration to find the linearization point as it provides a higher accuracy state estimate as compared to interpolation.

$$
\begin{aligned}
\mathbf{H}^{j,k}_{\boldsymbol{\theta}_{k+1}} &= \mathbf{H}_{\boldsymbol{\theta}} \triangleq \lambda {}^{I}_{C_R}\mathbf{C}^T \lfloor \mathbf{C}(\lambda\boldsymbol{\theta}) {}^{I_k}_{G}\mathbf{C}\boldsymbol{\xi} \rfloor \\
\mathbf{H}^{j,k}_{\boldsymbol{\theta}_k} &= -\mathbf{H}_{\boldsymbol{\theta}} {}^{I_{k+1}}_{G}\mathbf{C} {}^{I_k}_{G}\mathbf{C}^T + {}^{I}_{C_R}\mathbf{C}^T \mathbf{C}(\lambda\boldsymbol{\theta}) \lfloor {}^{I_k}_{G}\mathbf{C}\boldsymbol{\xi} \rfloor \\
\mathbf{H}^{j,k}_{\mathbf{p}_x} &= \mathbf{M} {}^{I_0}_{G}\mathbf{C}^T - {}^{I}_{C_R}\mathbf{C}^T \\
\mathbf{H}^{j,k}_{\boldsymbol{\theta}_x} &= \mathbf{M} {}^{I_0}_{G}\mathbf{C}^T \lfloor {}^{I}_{C_L}\mathbf{C} {}^{C_0}\mathbf{p}_{f_j} \rfloor - {}^{I}_{C_R}\mathbf{C}^T \lfloor {}^{I_{k+t}}_{G}\mathbf{C}\boldsymbol{\xi} - {}^{I}\mathbf{p}_{C_L} \rfloor \\
\mathbf{H}^{j,k}_{f} &= \mathbf{M} {}^{I_0}_{G}\mathbf{C}^T {}^{I}_{C_L}\mathbf{C} \qquad (14)
\end{aligned}
$$

with $\lfloor . \rfloor$ denoting the skew-symmetric matrix, $\boldsymbol{\theta} \triangleq \boldsymbol{\theta}_{k+1,k}$, and $\boldsymbol{\xi} \triangleq {}^{I_0}_{G}\mathbf{C}^T {}^{I}_{C_L}\mathbf{C} {}^{C_0}\mathbf{p}_{f_j} - {}^{G}\mathbf{p}_{I_{k+t}} + {}^{G}\mathbf{p}_{I_0} + {}^{I_0}_{G}\mathbf{C}^T {}^{I}\mathbf{p}_{C_L}$.

Note that, despite its more complicated expressions for the Jacobians, (13) has identical structure to the linearized measurement model in (6) corresponding to the left camera feature observations. Thus, it can be employed with any monocular VINS estimator (e.g., [23]).

2.3.2 Interpolation Ratio Computation

In our experiments, we employ different interpolation factors for the translation λ_t and rotation λ_θ terms, assuming varying velocities. Specifically, from (9) and (10) we have:

$$
{}^{G}\mathbf{p}_{I_{k+t}} - {}^{G}\mathbf{p}_{I_k} = \lambda_t ({}^{G}\mathbf{p}_{I_{k+1}} - {}^{G}\mathbf{p}_{I_k}) \qquad (15)
$$

$$
\begin{aligned}
{}^{I_{k+t}}_{G}\mathbf{C} &= \mathbf{C}(\lambda_\theta \boldsymbol{\theta}_{k+1,k}) {}^{I_k}_{G}\mathbf{C} \approx (\mathbf{I}_3 - \lfloor \lambda_\theta \boldsymbol{\theta} \rfloor) {}^{I_k}_{G}\mathbf{C} \\
&\Rightarrow \mathbf{I}_3 - {}^{I_{k+t}}_{G}\mathbf{C} {}^{I_k}_{G}\mathbf{C}^T = \lambda_\theta \lfloor \boldsymbol{\theta} \rfloor \qquad (16)
\end{aligned}
$$

where ${}^{G}\mathbf{p}_{I_{k+t}}$ and ${}^{I_{k+t}}_{G}\mathbf{C}$ are obtained from IMU integration. Then, λ_t and λ_θ are estimated in a least-squares (LS) sense, i.e., $\lambda_t = \frac{\mathbf{b}_1^T \mathbf{a}_1}{\mathbf{b}_1^T \mathbf{b}_1}$ and $\lambda_\theta = \frac{\mathbf{b}_2^T \mathbf{a}_2}{\mathbf{b}_2^T \mathbf{b}_2}$, where $\mathbf{a}_1 \triangleq {}^{G}\mathbf{p}_{I_{k+t}} - {}^{G}\mathbf{p}_{I_k}, \mathbf{b}_1 \triangleq {}^{G}\mathbf{p}_{I_{k+1}} - {}^{G}\mathbf{p}_{I_k}, \mathbf{a}_2 \triangleq \begin{bmatrix} \mathbf{A}_{(1,2)} & \mathbf{A}_{(1,3)} & \mathbf{A}_{(2,1)} & \mathbf{A}_{(2,3)} & \mathbf{A}_{(3,1)} & \mathbf{A}_{(3,2)} \end{bmatrix}^T$, and $\mathbf{b}_2 \triangleq \begin{bmatrix} \mathbf{B}_{(1,2)} & \mathbf{B}_{(1,3)} & \mathbf{B}_{(2,1)} & \mathbf{B}_{(2,3)} & \mathbf{B}_{(3,1)} & \mathbf{B}_{(3,2)} \end{bmatrix}^T$, with $\mathbf{A} = \mathbf{I}_3 - {}^{I_{k+t}}_{G}\mathbf{C} {}^{I_k}_{G}\mathbf{C}^T$ and $\mathbf{B} = \lfloor \boldsymbol{\theta} \rfloor$.

Note that other higher-order interpolation schemes (e.g., B-splines [17], GP interpolation [6]) can also be employed, but their gain in accuracy is negligible, as compared to linear interpolation, for short time duration and hence, does not justify the processing overhead.

3. Image-processing Front-end

The proposed system extracts and tracks point features on consecutive alternating images (see Fig. 1). The tracking algorithm is, however, indifferent to whether the image is provided by the left or the right camera and processes the alternating image stream as if they are coming from a monocular system. Specifically, a descriptor-based tracking-by-matching strategy (similar to [21]) is employed. As a first step, a 3D-to-2D matching is performed against the local SLAM map, followed by a gyro-aided (i.e., using a rotation-only prediction from the integrated gyroscope measurements) 2D-to-2D matching to associate the remaining features with the previous 2D feature tracks. Next, outliers are rejected using the 2-pt RANSAC [15] and the Mahalanobis distance test. The inlier tracks are then triangulated, using all observations from all viewing cameras, and processed by the estimator. After triangulation, outliers are rejected by checking both the individual and mean re-projection errors for all observations in a track. Additionally, when right-camera measurements are present in a track, if the track's mean re-projection error is larger than the corresponding left track error, it is considered to have erroneous stereo associations and is marked as an outlier.

4. Estimation Algorithm

In what follows, we describe the main steps of the estimation algorithm. At each time step k, the objective is to minimize the cost term $\mathcal{C}_k^{\oplus} = \mathcal{C}_{k-1} + \mathcal{C}_u + \mathcal{C}_{\mathbb{Z}}$ that contains all available information so far, where $\mathcal{C}_u$ [see (4)] represents the cost term arising from the IMU measurement $\mathbf{u}_{k-1,k}$, $\mathcal{C}_{\mathbb{Z}}$ from the SLAM visual measurements, and $\mathcal{C}_{k-1}$ from the prior information obtained from the previous time step, with $\mathcal{C}_{k-1}(\widetilde{\mathbf{x}}_{k-1}) = \|\mathbf{R}_{k-1}\widetilde{\mathbf{x}}_{k-1} - \mathbf{r}_{k-1}\|^2$, where $\mathbf{R}_{k-1}$ and $\mathbf{r}_{k-1}$ are the prior information *factor* matrix and residual vector, respectively.

At each time step k, the current state vector $\widetilde{\mathbf{x}}_{k-1}$ is first propagated by appending a new pose state $\mathbf{x}_{I_k}$ [see (3)] to it, as $\mathbf{x}_k^{\ominus} = \begin{bmatrix} \mathbf{x}_{k-1}^T & \mathbf{x}_{I_k}^T \end{bmatrix}^T$. Following [23], the cost term, which initially comprised only $\mathcal{C}_{k-1}$, then becomes

$$
\mathcal{C}_k^{\ominus}(\widetilde{\mathbf{x}}_k^{\ominus}) = \mathcal{C}_{k-1}(\widetilde{\mathbf{x}}_{k-1}) + \mathcal{C}_u(\widetilde{\mathbf{x}}_{I_{k-1}}, \widetilde{\mathbf{x}}_{I_k}) \qquad (17)
$$

To maintain constant computational complexity, at each time step k, 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 are marginalized. For marginalization, as in [23], a state permutation followed by a QR factorization [11] is applied, with the resulting cost term after marginalization being:

$$
\mathcal{C}_k^{M}(\widetilde{\mathbf{x}}_k^{R}) = \min_{\widetilde{\mathbf{x}}_k^{M}} \mathcal{C}_k^{\ominus}(\widetilde{\mathbf{x}}_k^{M}, \widetilde{\mathbf{x}}_k^{R}) = \left\|\mathbf{R}_k^{R}\widetilde{\mathbf{x}}_k^{R} - \mathbf{r}_k^{R}\right\|^2 \qquad (18)
$$

where $\widetilde{\mathbf{x}}_k^{M}$ are the marginalized states, $\widetilde{\mathbf{x}}_k^{R}$ are the remaining states after marginalization [see (1)], while $\mathbf{R}_k^{R}$ and $\mathbf{r}_k^{R}$ are the corresponding upper-triangular information *factor* and residual, respectively. After marginalization, new SLAM feature states $\mathbf{x}_S^N$ are added to the state vector as $\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_k^{R^T} & \mathbf{x}_S^{N^T} \end{bmatrix}^T$. The new SLAM feature observations $\mathbb{Z}_N$ and re-observations $\mathbb{Z}_R$ of existing SLAM features are then used to perform updates.

$$
\mathcal{C}_k^{\oplus}(\widetilde{\mathbf{x}}_k) = \mathcal{C}_k^{M}(\widetilde{\mathbf{x}}_k^{R}) + \mathcal{C}_{\mathbb{Z}}(\widetilde{\mathbf{x}}_k) = \|\mathbf{R}_k^{\oplus}\widetilde{\mathbf{x}}_k - \mathbf{r}_k^{\oplus}\|^2 \qquad (19)
$$

where $\mathcal{C}_{\mathbb{Z}}(\widetilde{\mathbf{x}}_k) = \mathcal{C}_{\mathbb{Z}_N}(\widetilde{\mathbf{x}}_k) + \mathcal{C}_{\mathbb{Z}_R}(\widetilde{\mathbf{x}}_k^{R}) = \sum_{j=1}^{N_{NS}} \mathcal{C}_{z_j}(\widetilde{\mathbf{x}}_k) + \sum_{j=1}^{N_R} \mathcal{C}_{z_j}(\widetilde{\mathbf{x}}_k^{R})$, with N_{NS} and N_R being the number of new SLAM features and SLAM re-observations, respectively.

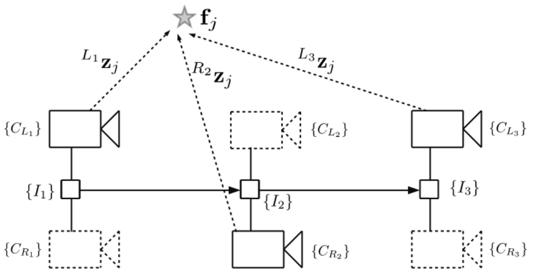


Figure 3. System setup, where for the i^{th} time step $\{I_i\}$, $\{C_{L_i}\}$, and $\{C_{R_i}\}$ denote the IMU, left camera, and right camera frames, respectively, and $\mathbf{f}_j$, with $j = 1, \ldots, 5$, are the features observed by the active camera frames $\{C_{L_1}\}$, $\{C_{R_2}\}$, and $\{C_{L_3}\}$, the observations being ${}^{L_1}\mathbf{z}_j$, ${}^{R_2}\mathbf{z}_j$, and ${}^{L_3}\mathbf{z}_j$.

Finally, (19) is minimized with respect to the error state vector and the solution for $\widetilde{\mathbf{x}}_k$ is used to update the state.

$$\min_{\widetilde{\mathbf{x}}_k} \mathcal{C}_k^{\oplus}(\widetilde{\mathbf{x}}_k) = \min_{\widetilde{\mathbf{x}}_k} \|\mathbf{R}_k^{\oplus}\widetilde{\mathbf{x}}_k - \mathbf{r}_k^{\oplus}\|^2 \tag{20}$$

At the next time step $k+1$, a new clone pose will be added to the sliding window and the same process will be repeated, with $\hat{\mathbf{x}}_k^{\oplus} = \hat{\mathbf{x}}_k + \widetilde{\mathbf{x}}_k$ and $\mathbf{R}_k^{\oplus}$ serving as the corresponding prior state estimate and information factor, respectively.

5. Observability Analysis

In this section, we study the observability properties (i.e., gauge freedom analysis) of the linearized vision-only version of the proposed system and show that compared to its monocular counterpart, scale becomes observable. For simplicity, we employ the minimal setup of Fig. 3, depicting 3 consecutive camera frames in a left-right alternating fashion. Since we are considering a vision-only system, for the purpose of the observability analysis, we address the case where 5 static features comprise the scene and are detected by all 3 consecutive camera frames. Furthermore, we assume that the pose of the first camera frame is known and all other frames are expressed with respect to it. Note that, the extension of the following analysis to the general case of m poses and n features is straightforward.

For easiness of presentation, we first study a monocular system and determine the unobservable direction corresponding to the scale. Then, we examine the proposed alternating-stereo system and prove that the null direction vanishes.

5.1. Monocular System

In the monocular system, the two left camera measurements for the j^{th} feature can be written as,

$$\begin{aligned}
{}^{L_1}\mathbf{z}_j &= \pi({}^{C_{L_1}}\mathbf{p}_{f_j}) + \mathbf{n}_{L_1,j} && (21)\\
{}^{L_3}\mathbf{z}_j &= \pi\left({}^{C_{L_3}}\mathbf{p}_{f_j}\right) + \mathbf{n}_{L_3,j} \\
&= \pi\left({}^{C_{L_3}}_{C_{L_1}}\mathbf{C}\left({}^{C_{L_1}}\mathbf{p}_{f_j} - {}^{C_{L_1}}\mathbf{p}_{C_{L_3}}\right)\right) + \mathbf{n}_{L_3,j} && (22)
\end{aligned}$$

where ${}^{C_{L_1}}\mathbf{p}_{f_j}$ and ${}^{C_{L_3}}\mathbf{p}_{f_j}$ are the positions of the j^{th} feature in the left camera frames $\{C_{L_1}\}$ and $\{C_{L_3}\}$, respectively, $\mathbf{n}_{L_1,j}$ and $\mathbf{n}_{L_3,j}$ are zero-mean white Gaussian noises, and $({}^{C_{L_3}}_{C_{L_1}}\mathbf{C}, {}^{C_{L_1}}\mathbf{p}_{C_{L_3}})$ is the pose of $\{C_{L_3}\}$ with respect to $\{C_{L_1}\}$. Linearizing (21) and (22) yields:

$$\begin{aligned}
{}^{L_1}\widetilde{\mathbf{z}}_j &= {}^{L_1}\mathbf{\Pi}_j\widetilde{\mathbf{f}}_j + \mathbf{n}_{L_1,j} && (23)\\
{}^{L_3}\widetilde{\mathbf{z}}_j &= {}^{L_3}\mathbf{\Pi}_j{}^3\mathbf{C}\widetilde{\mathbf{f}}_j - {}^{L_3}\mathbf{\Pi}_j{}^3\mathbf{C}\widetilde{\mathbf{p}}_3 \\
&\quad + {}^{L_3}\mathbf{\Pi}_j{}^3\mathbf{C}\lfloor\mathbf{f}_j - \mathbf{p}_3\rfloor{}^3\mathbf{C}^T\widetilde{\boldsymbol{\theta}}_3 + \mathbf{n}_{L_3,j} && (24)
\end{aligned}$$

where ${}^{L_3}\mathbf{\Pi}_j$ and ${}^{L_1}\mathbf{\Pi}_j$ are the Jacobians of the perspective projection functions, while we simplified notation, $\mathbf{f}_j \triangleq {}^{C_{L_1}}\mathbf{p}_{f_j}$, $\mathbf{p}_3 \triangleq {}^{C_{L_1}}\mathbf{p}_{C_{L_3}}$, and ${}^3\mathbf{C} \triangleq {}^{C_{L_3}}_{C_{L_1}}\mathbf{C} = \mathbf{C}({}^{C_{L_3}}\boldsymbol{\theta}_{C_{L_1}})$ with $\boldsymbol{\theta}_3 \triangleq {}^{C_{L_3}}\boldsymbol{\theta}_{C_{L_1}}$. From (23) and (24), the Jacobian corresponding to the error state $\widetilde{\mathbf{x}}_j = \begin{bmatrix}\widetilde{\mathbf{p}}_3^T & \widetilde{\boldsymbol{\theta}}_3^T & \widetilde{\mathbf{f}}_j^T\end{bmatrix}^T$ is,

$$\mathbf{H}_j = \begin{bmatrix} {}^{L_1}\mathbf{\Pi}_j & \mathbf{0}_{2\text{x}3} \\ \mathbf{0}_{2\text{x}3} & {}^{L_3}\mathbf{\Pi}_j{}^3\mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \\ -\mathbf{I}_3 & \lfloor\mathbf{f}_j - \mathbf{p}_3\rfloor{}^3\mathbf{C}^T & \mathbf{I}_3 \end{bmatrix} \tag{25}$$

By stacking together the Jacobians for all five features, from only the left camera, we get the Jacobian $\mathbf{H} = \mathbf{D}\mathbf{M}$ corresponding to the error state $\widetilde{\mathbf{x}} = \begin{bmatrix}\widetilde{\mathbf{p}}_3^T & \widetilde{\boldsymbol{\theta}}_3^T & \widetilde{\mathbf{f}}_1^T & \widetilde{\mathbf{f}}_2^T & \widetilde{\mathbf{f}}_3^T & \widetilde{\mathbf{f}}_4^T & \widetilde{\mathbf{f}}_5^T\end{bmatrix}^T$, where

$$\begin{aligned}
\mathbf{D} &\triangleq \text{BlkDiag}({}^{L_1}\mathbf{\Pi}_1, {}^{L_3}\mathbf{\Pi}_1{}^3\mathbf{C}, \ldots, {}^{L_1}\mathbf{\Pi}_5, {}^{L_3}\mathbf{\Pi}_5{}^3\mathbf{C}) \\
\mathbf{M} &\triangleq \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \ldots & \mathbf{0}_3 \\ -\mathbf{I}_3 & \lfloor\mathbf{f}_1 - \mathbf{p}_3\rfloor{}^3\mathbf{C}^T & \mathbf{I}_3 & \ldots & \mathbf{0}_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \ldots & \mathbf{I}_3 \\ -\mathbf{I}_3 & \lfloor\mathbf{f}_5 - \mathbf{p}_3\rfloor{}^3\mathbf{C}^T & \mathbf{0}_3 & \ldots & \mathbf{I}_3 \end{bmatrix}
\end{aligned} \tag{26}$$

The unobservable direction of the system corresponds to the nullspace of the Jacobian matrix $\mathbf{H}$. To analyze this nullspace, we use the fact that the rank of the product of two matrices $\mathbf{D}$ and $\mathbf{M}$ is given by (see (4.5) in [19]):

$$\text{rank}(\mathbf{D}\mathbf{M}) = \text{rank}(\mathbf{M}) - \dim(\mathcal{N}(\mathbf{D}) \cap \mathcal{R}(\mathbf{M})) \tag{27}$$

where $\mathcal{R}(.)$ and $\mathcal{N}(.)$ represents the range and nullspace of a matrix, respectively. Hereafter, we first show that $\mathbf{M}$ is of full column rank, using Gaussian elimination. Specifically, subtracting every odd row from its next even row and rearranging the rows of $\mathbf{M}$ yields:

$$\mathbf{M} \rightarrow \left[\begin{array}{cc|c} -\mathbf{I}_3 & \lfloor\mathbf{f}_1 - \mathbf{p}_3\rfloor{}^3\mathbf{C}^T & \\ -\mathbf{I}_3 & \lfloor\mathbf{f}_2 - \mathbf{p}_3\rfloor{}^3\mathbf{C}^T & \\ -\mathbf{I}_3 & \lfloor\mathbf{f}_3 - \mathbf{p}_3\rfloor{}^3\mathbf{C}^T & \mathbf{0}_{15} \\ -\mathbf{I}_3 & \lfloor\mathbf{f}_4 - \mathbf{p}_3\rfloor{}^3\mathbf{C}^T & \\ -\mathbf{I}_3 & \lfloor\mathbf{f}_5 - \mathbf{p}_3\rfloor{}^3\mathbf{C}^T & \\ \hline \multicolumn{2}{c|}{\mathbf{0}_{15\text{x}6}} & \mathbf{I}_{15} \end{array}\right] \tag{28}$$

Then, by negating the first block-row and adding it to the next 4 block-rows we get,

$$\mathbf{M} \to \left[\begin{array}{ll|c} \mathbf{I}_3 & \lfloor \mathbf{p}_3 - \mathbf{f}_1 \rfloor {}^3\mathbf{C}^T & \\ \mathbf{0}_3 & \lfloor \mathbf{f}_2 - \mathbf{f}_1 \rfloor {}^3\mathbf{C}^T & \\ \mathbf{0}_3 & \lfloor \mathbf{f}_3 - \mathbf{f}_1 \rfloor {}^3\mathbf{C}^T & \mathbf{0}_{15} \\ \mathbf{0}_3 & \lfloor \mathbf{f}_4 - \mathbf{f}_1 \rfloor {}^3\mathbf{C}^T & \\ \mathbf{0}_3 & \lfloor \mathbf{f}_5 - \mathbf{f}_1 \rfloor {}^3\mathbf{C}^T & \\ \hline \multicolumn{2}{c|}{\mathbf{0}_{15\text{x}6}} & \mathbf{I}_{15} \end{array}\right] \tag{29}$$

Now, consider the 6x3 block $\left[(\lfloor \mathbf{f}_4 - \mathbf{f}_1 \rfloor {}^3\mathbf{C}^T)^T \;\; (\lfloor \mathbf{f}_5 - \mathbf{f}_1 \rfloor {}^3\mathbf{C}^T)^T\right]^T$. Here, in general, each of the two block rows has rank 2, but since the features $\mathbf{f}_1$, $\mathbf{f}_4$, and $\mathbf{f}_5$ are not colinear, the 2 block rows do not have a common nullspace. In other words, the matrix formed by these two blocks has full column rank. Therefore, after applying appropriate Gaussian-elimination the block becomes $[\mathbf{I}_3 \;\; \mathbf{0}_3]^T$; i.e.,

$$\mathbf{M} \to \left[\begin{array}{ll|c} \mathbf{I}_3 & \lfloor \mathbf{p}_3 - \mathbf{f}_1 \rfloor {}^3\mathbf{C}^T & \\ \mathbf{0}_3 & \lfloor \mathbf{f}_2 - \mathbf{f}_1 \rfloor {}^3\mathbf{C}^T & \\ \mathbf{0}_3 & \lfloor \mathbf{f}_3 - \mathbf{f}_1 \rfloor {}^3\mathbf{C}^T & \mathbf{0}_{15} \\ \mathbf{0}_3 & \qquad \mathbf{I}_3 & \\ \mathbf{0}_3 & \qquad \mathbf{0}_3 & \\ \hline \multicolumn{2}{c|}{\mathbf{0}_{15\text{x}6}} & \mathbf{I}_{15} \end{array}\right] \tag{30}$$

This block is then used in subsequent Gaussian-elimination to the 2^{nd} column blocks to yield:

$$\mathbf{M} \to \left[\begin{array}{ll|c} \mathbf{I}_3 & \mathbf{0}_3 & \\ \mathbf{0}_3 & \mathbf{0}_3 & \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_{15} \\ \mathbf{0}_3 & \mathbf{I}_3 & \\ \mathbf{0}_3 & \mathbf{0}_3 & \\ \hline \multicolumn{2}{c|}{\mathbf{0}_{15\text{x}6}} & \mathbf{I}_{15} \end{array}\right] \to \begin{bmatrix} \mathbf{I}_{21} \\ \mathbf{0}_{9\text{x}21} \end{bmatrix} \tag{31}$$

Hence, $\mathbf{M}_{30\text{x}21}$ is a full column rank matrix of rank 21.

Now, $\mathbf{D}_{20\text{x}30}$ is a block diagonal matrix, where each diagonal block is of rank 2. So, $\mathbf{D}$ has a 10 dimensional nullspace spanned by $\boldsymbol{\eta}_{1,j} = [\mathbf{0}_{1\text{x}3} \; \dots \; \mathbf{f}_j^T \; \dots \; \mathbf{0}_{1\text{x}3}]^T$ and $\boldsymbol{\eta}_{2,j} = [\mathbf{0}_{1\text{x}3} \; \dots \; (\mathbf{f}_j - \mathbf{p}_3)^T \; \dots \; \mathbf{0}_{1\text{x}3}]^T$, where $j = 1, 2, \dots, 5$.

From the expression of $\mathbf{M}$ in (26) and the basis $\{\boldsymbol{\eta}_{1,j}, \boldsymbol{\eta}_{2,j}\}$, $j = 1, 2, \dots, 5$, of the nullspace of $\mathbf{D}$, it can be shown that (proof omitted due to lack of space) there exists only one linearly independent direction $\boldsymbol{\eta}$ in $\mathcal{N}(\mathbf{D}) \cap \mathcal{R}(\mathbf{M})$, where[3]

$$\begin{aligned} \boldsymbol{\eta} &= \sum_{i=1}^{2}\sum_{j=1}^{5} \boldsymbol{\eta}_{i,j} = [\mathbf{f}_1^T \;\; (\mathbf{f}_1 - \mathbf{p}_3)^T \;\; \dots \;\; \mathbf{f}_5^T \;\; (\mathbf{f}_5 - \mathbf{p}_3)^T]^T \\ &= \mathbf{M}_{(:,1:3)}\mathbf{p}_3 + \mathbf{M}_{(:,7:9)}\mathbf{f}_1 + \dots + \mathbf{M}_{(:,19:21)}\mathbf{f}_5 \end{aligned} \tag{32}$$

Therefore, $\text{rank}(\mathbf{H}) = \text{rank}(\mathbf{M}) - \dim(\boldsymbol{\eta}) = 21 - 1 = 20$, and $\mathbf{H}$ has an one dimensional nullspace $\mathcal{N}(\mathbf{H}) = \text{span}([\mathbf{p}_3^T \; \mathbf{0}_{1\text{x}3} \; \mathbf{f}_1^T \; \dots \; \mathbf{f}_5^T]^T)$, which is the unobservable direction corresponding to scale (see [24]).

5.2. Alternating-Stereo System

In the alternating-stereo system, in addition to the left camera observations in (21) and (22) for the j^{th} feature, the right camera also contributes a measurement:

$${}^{R_2}\mathbf{z}_j = \pi\left({}^{C_{R_2}}\mathbf{p}_{f_j}\right) + \mathbf{n}_{R_2,j} \tag{33}$$

with ${}^{C_{R_2}}\mathbf{p}_{f_j} = {}^{C_R}\mathbf{p}_{C_L} + {}^{C_{R_2}}_{C_{L_1}}\mathbf{C}\left({}^{C_{L_1}}\mathbf{p}_{f_j} - {}^{C_{L_1}}\mathbf{p}_{C_{L_2}}\right)$, where ${}^{C_{R_2}}_{C_{L_1}}\mathbf{C} = {}^{C_R}_{C_L}\mathbf{C}{}^{C_{L_2}}_{C_{L_1}}\mathbf{C}$, ${}^{C_{R_2}}\mathbf{p}_{f_j}$ is the position of the j^{th} feature in the right camera frame $\{C_{R_2}\}$, $\mathbf{n}_{R_2,j}$ is zero-mean white Gaussian noise, $({}^{C_{L_2}}_{C_{L_1}}\mathbf{C}, {}^{C_{L_1}}\mathbf{p}_{C_{L_2}})$ is the pose of the corresponding left camera frame $\{C_{L_2}\}$ with respect to $\{C_{L_1}\}$, and $({}^{C_R}_{C_L}\mathbf{C}, {}^{C_R}\mathbf{p}_{C_L})$ is the known left-right camera extrinsics. Now, in the alternating-stereo system only left camera frames are cloned.[4] Hence, we introduce a linear interpolation model, i.e., ${}^{C_{L_2}}_{C_{L_1}}\mathbf{C} = \mathbf{C}(\lambda\boldsymbol{\theta}_3)$, ${}^{C_{L_1}}\mathbf{p}_{C_{L_2}} = \lambda\mathbf{p}_3$, to relate the right camera measurements with the two adjacent left clones.

$${}^{R_2}\mathbf{z}_j = \pi({}^{C_R}\mathbf{p}_{C_L} + {}^{C_R}_{C_L}\mathbf{C}\mathbf{C}(\lambda\boldsymbol{\theta}_3)(\mathbf{f}_j - \lambda\mathbf{p}_3)) + \mathbf{n}_{R_2,j} \tag{34}$$

Linearizing (34) yields:

$$\begin{aligned} {}^{R_2}\widetilde{\mathbf{z}}_j = {}& {}^{R_2}\boldsymbol{\Pi}_j {}^{2R}\mathbf{C}\widetilde{\mathbf{f}}_j - \lambda {}^{R_2}\boldsymbol{\Pi}_j {}^{2R}\mathbf{C}\widetilde{\mathbf{p}}_3 \\ &+ \lambda {}^{R_2}\boldsymbol{\Pi}_j {}^{2R}\mathbf{C}\lfloor \mathbf{f}_j - \mathbf{p}_2 \rfloor {}^2\mathbf{C}^T\widetilde{\boldsymbol{\theta}}_3 + \mathbf{n}_{R_2,j} \end{aligned} \tag{35}$$

where ${}^{R_2}\boldsymbol{\Pi}_j$ is the perspective projection Jacobian, $\mathbf{p}_2 \triangleq {}^{C_{L_1}}\mathbf{p}_{C_{L_2}}$, ${}^{2R}\mathbf{C} \triangleq {}^{C_{R_2}}_{C_{L_1}}\mathbf{C}$, and ${}^2\mathbf{C} \triangleq {}^{C_{L_2}}_{C_{L_1}}\mathbf{C}$. From (23), (35), and (24), combining measurements from both left and right cameras, the Jacobian corresponding to the error state $\widetilde{\mathbf{x}}_j$ is:

$$\mathbf{H}_j = \begin{bmatrix} \mathbf{0}_{2\text{x}3} & \mathbf{0}_{2\text{x}3} & \mathbf{D}_{1,j} \\ -\lambda\mathbf{D}_{2,j} & \lambda\mathbf{D}_{2,j}\lfloor \mathbf{f}_j - \mathbf{p}_2 \rfloor {}^2\mathbf{C}^T & \mathbf{D}_{2,j} \\ -\mathbf{D}_{3,j} & \mathbf{D}_{3,j}\lfloor \mathbf{f}_j - \mathbf{p}_3 \rfloor {}^3\mathbf{C}^T & \mathbf{D}_{3,j} \end{bmatrix} \tag{36}$$

where $\mathbf{D}_{1,j} \triangleq {}^{L_1}\boldsymbol{\Pi}_j$, $\mathbf{D}_{2,j} \triangleq {}^{R_2}\boldsymbol{\Pi}_j {}^{2R}\mathbf{C}$, and $\mathbf{D}_{3,j} \triangleq {}^{L_3}\boldsymbol{\Pi}_j {}^3\mathbf{C}$. By stacking together the Jacobians for all five feature measurements, we get the Jacobian $\mathbf{H} = \mathbf{D}\mathbf{M}$ corresponding to the error state $\widetilde{\mathbf{x}}$, where

$$\mathbf{D} \triangleq \text{BlkDiag}(\mathbf{D}_{1,1}, \mathbf{D}_{2,1}, \mathbf{D}_{3,1}, \dots, \mathbf{D}_{1,5}, \mathbf{D}_{2,5}, \mathbf{D}_{3,5})$$

$$\mathbf{M} \triangleq \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \dots & \mathbf{0}_3 \\ -\lambda\mathbf{I}_3 & \lambda\lfloor \mathbf{f}_1 - \mathbf{p}_2 \rfloor {}^2\mathbf{C}^T & \mathbf{I}_3 & \dots & \mathbf{0}_3 \\ -\mathbf{I}_3 & \lfloor \mathbf{f}_1 - \mathbf{p}_3 \rfloor {}^3\mathbf{C}^T & \mathbf{I}_3 & \dots & \mathbf{0}_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & & \mathbf{I}_3 \\ -\lambda\mathbf{I}_3 & \lambda\lfloor \mathbf{f}_5 - \mathbf{p}_2 \rfloor {}^2\mathbf{C}^T & \mathbf{0}_3 & \dots & \mathbf{I}_3 \\ -\mathbf{I}_3 & \lfloor \mathbf{f}_5 - \mathbf{p}_3 \rfloor {}^3\mathbf{C}^T & \mathbf{0}_3 & \dots & \mathbf{I}_3 \end{bmatrix} \tag{37}$$

[3] Using MATLAB notations.

[4] Note that, if we cloned at every frame the alternating scheme will lose scale. Specifically, it will introduce 2 additional block columns in the $\mathbf{M}$ matrix [see (26)], resulting in $\dim(\mathcal{N}(\mathbf{H})) = \dim(\mathcal{N}(\mathbf{D}) \cap \mathcal{R}(\mathbf{M}) = 1$.

Similarly to the monocular system, it can be shown that $\mathbf{M}_{45\text{x}21}$ is a full column rank matrix of rank 21 and $\mathbf{D}_{30\text{x}45}$ has a 15 dimensional nullspace that is spanned by: $\boldsymbol{\eta}_{1,j} = [\mathbf{0}_{1\text{x}3} \ \ldots \ \mathbf{f}_j^T \ \ldots \ \mathbf{0}_{1\text{x}3}]^T$, $\boldsymbol{\eta}_{2,j} = [\mathbf{0}_{1\text{x}3} \ \ldots \ (\mathbf{f}_j - \mathbf{p}_3)^T \ \ldots \ \mathbf{0}_{1\text{x}3}]^T$, and $\boldsymbol{\eta}_{3,j} = [\mathbf{0}_{1\text{x}3} \ \ldots \ (\mathbf{f}_j - \lambda\mathbf{p}_3 + \mathbf{u})^T \ \ldots \ \mathbf{0}_{1\text{x}3}]^T$, where $j = 1, 2, \ldots, 5$ and $\mathbf{u} \triangleq {}^{2R}\mathbf{C}^{T\,C_R}\mathbf{p}_{C_L}$. In this case, it can be shown that there exists no nonzero vector in $\mathcal{N}(\mathbf{D}) \cap \mathcal{R}(\mathbf{M})$, i.e., $\mathcal{N}(\mathbf{D}) \cap \mathcal{R}(\mathbf{M}) = \{\mathbf{0}_{45\text{x}1}\}$. Therefore, $\text{rank}(\mathbf{H}) = \text{rank}(\mathbf{M}) - 0 = 21$, and $\mathbf{H}_{30\text{x}21}$ has no nullspace. Since $\mathbf{H}$ has no null direction, unlike the monocular system, scale becomes observable in the proposed interpolation-based alternating-stereo system with just the visual observations. This can be easily verified by multiplying the Jacobian $\mathbf{H}$ with the scale direction $[\mathbf{p}_3^T \ \mathbf{0}_{1\text{x}3} \ \mathbf{f}_1^T \ \ldots \ \mathbf{f}_5^T]^T$, and confirming that the result is not zero, due to the fact that $\mathcal{N}(\mathbf{D}) \cap \mathcal{R}(\mathbf{M}) = \{\mathbf{0}_{45\text{x}1}\}$.

To summarize, processing the alternating visual observations requires a motion model for describing the camera poses between key-frames. The motion model along with the stereo constraint allow us to acquire scale information, whose accuracy depends on how well the motion model approximates the device's actual motion. In practice, during the short time between frames, the assumptions of smooth motion and small rotation typically hold and hence the proposed linear interpolation model is adequate. Finally, we should note that the scale will be observable for higher-order models too as the presence of the stereo baseline ${}^{C_R}\mathbf{p}_{C_L}$ in the nullspace $\boldsymbol{\eta}_{3,j}$ of $\mathbf{D}$, will prevent the observability matrix from losing rank.

6. Experimental Results

For our experiments, the wide stereo rig depicted in Fig. 4 was used, which contains two global shutter Chameleon-2 camera sensors with PT-02118BMP fixed-focus, 165° field of view (FOV), fisheye lenses. The baseline between the cameras is 19.3 cm and they capture VGA-resolution images at 25 Hz. A commercial-grade Invensense MPU-9250 IMU is used to measure inertial data at 100 Hz. The cameras are triggered in an alternating fashion, using an Arduino Nano micro-controller, and are time-synced with the IMU. The full pipeline runs in real-time on the NVIDIA Jetson TK1 [3] board, which is equipped with a Tegra TK1 mobile processor, featuring a Kepler GPU and a quad-core ARM Cortex-A15 CPU.

In what follows, we present our evaluation results on 6 hand-held indoor sequences with varied motion profiles, captured using the device described above. Furthermore, we also present results for the 30 Hz EuRoC MAV [9] datasets, which consist of 11 indoor sequences recorded onboard a micro-aerial vehicle under various motion profiles and scene illuminations. For assessing the positioning accuracy, we compute the root mean square error (RMSE) of each trajectory against VICON ground-truth.

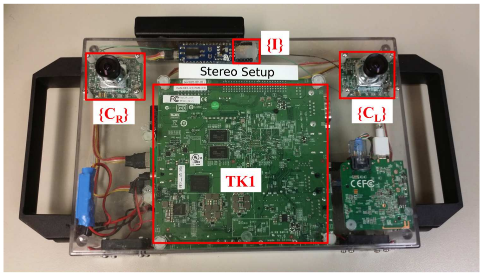


Figure 4. Stereo device.

6.1. Configurations Considered

In Secs. 6.2 and 6.3, we evaluate the accuracy and computational performance, respectively, of the proposed alternating-stereo system against its monocular and stereo counterparts. Since the proposed system performs image-processing at f Hz ($f = 25$ Hz and 30 Hz for our and the EuRoC datasets, respectively) and filter updates at $f/2$ Hz, for a thorough comparison, we included monocular and stereo systems operating at both $f/2$ Hz and f Hz. The optimization window size, M is set to 10 for these comparisons. For the f Hz cases, however, using the same window size usually reduces the baseline[5] and thus does not always guarantee better performance over its $f/2$ Hz counterparts. Therefore, for fairness, we also included the f Hz monocular and stereo systems with $M = 20$ in our comparison.

6.2. Accuracy Comparison

The RMSE results for the aforementioned systems are summarized in Fig. 5 with a box-and-whisker plot. As expected, stereo achieves better performance than mono and estimation accuracy usually improves with higher image-processing frequency and optimization window size ($M = 20$ vs. $M = 10$). The f Hz systems with $M = 10$, however, do not necessarily achieve better accuracy than their $f/2$ Hz counterparts, due to the smaller effective baseline. As evident from Fig. 5, the proposed alternating-stereo system always performs better than the $M = 10$ mono, while it either outperforms or is on par with the $M = 20$ mono. Interestingly, the proposed system attains better accuracy than even the $f/2$ Hz stereo. This is due to the fact that besides providing scale and spanning the same key-frames as the $f/2$ Hz stereo does, the alternating-stereo has access to typically longer feature tracks (owing to the f Hz feature tracking) and to additional feature observations from the alternating frames. The alternating-stereo also exhibits similar or better performance than the f Hz, $M = 10$ stereo, because of having comparable feature tracks with a longer effective baseline. The $M = 20$ stereo, however, performs the best, since it uses every frame as

[5] Baseline due to motion.

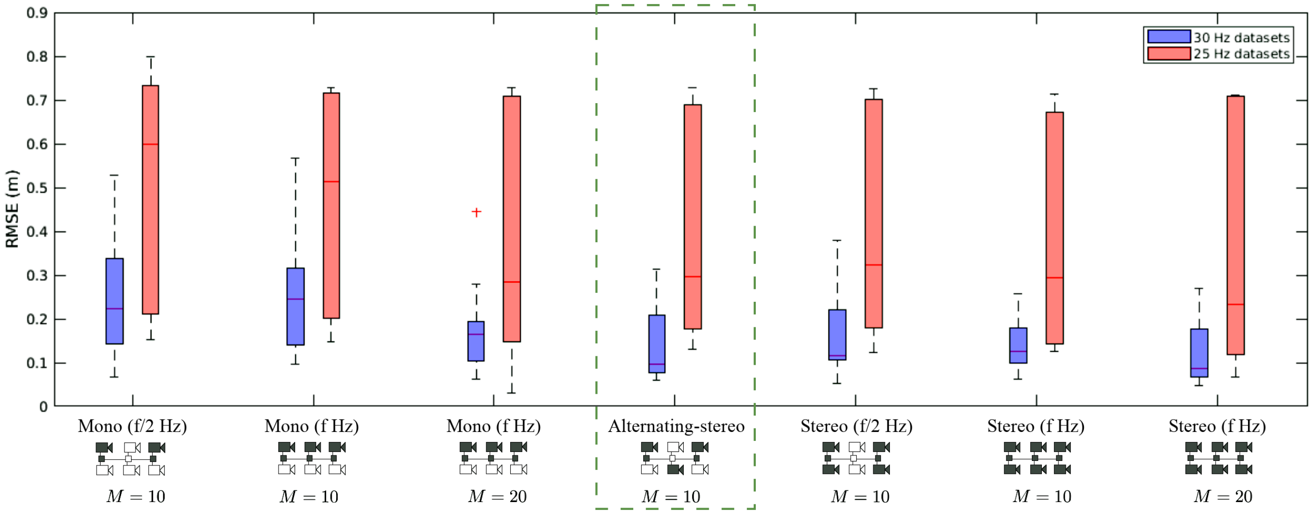


Figure 5. RMSE comparison between different configurations. f is the cloning frequency and M is the optimization window size.

key-frame, while maintaining similar feature tracks and temporal span as the alternating-stereo. Note that compared to the 30 Hz datasets, the 25 Hz datasets demonstrate relatively lower gain from the interpolation model in the alternating-stereo scheme. This is to be expected, since in this case the system needs to interpolate through a longer time interval (80 msec as compared to 60 msec) and hence the linear approximation is less often valid.

Lastly, we assessed the impact of our proposed LS-based interpolation ratio (see Sec. 2.3.2) over the time-based approach of [8, 12] and found our method to be 0.03 m more accurate in terms of median RMSE.

6.3. Computational Performance

Table 1 compares the processing times of the proposed system on the NVIDIA Jetson TK1 [3] with the aforementioned mono and stereo systems. For brevity, only the timing results from our 25 Hz datasets are shown. As evident, compared to the mono systems, stereo requires more than double the CPU to perform image-processing due to the additional feature extraction, tracking, and stereo-matching steps. In terms of filter update time, stereo requires ~ 1.3 times more CPU than mono, as it needs to process additional feature observations from the second camera. Furthermore, since the f Hz systems perform image-processing and filter updates twice as often compared to the $f/2$ Hz systems, their processing requirements are also double. Lastly, doubling the optimization window size M increases the filter update time by a factor of ~ 2.8 and the total time by ~ 1.4 times.

The proposed alternating-stereo system performs feature extraction and tracking at f Hz and filter updates at $f/2$ Hz, resulting in similar filter update time but almost double image-processing time as compared to the $f/2$ Hz mono. Nonetheless, since alternating-stereo does not require an additional stereo-matching step, it still performs faster than the $f/2$ Hz stereo. Lastly, we note that the proposed system runs in real-time on the NVIDIA Jetson TK1; in-fact, besides the $f/2$ Hz mono, it is the only other real-time system in our comparison.

Table 1. Comparison: Timing Results (msec)

Pipelines		Filter update (per key-frame)	Image-proc. (per key-frame)	Total pipeline (per 1 sec data)
Mono	$f/2$ Hz, M=10	14.06	30.23	666.75
	f Hz, M=10	14.32	27.53	1264.20
	f Hz, M=20	38.72	21.53	1728.49
Stereo	$f/2$ Hz, M=10	17.81	71.89	1372.41
	f Hz, M=10	18.77	67.18	2609.26
	f Hz, M=20	52.81	72.99	3710.50
Alternating-stereo		11.16	50.38	977.60

7. Conclusion

In this paper, we present a novel alternating-stereo VINS which enjoys the low latency of a monocular system, while acquiring scale information from visual observations analogously to a stereo system. To do so, we introduce an alternating cloning strategy along with an interpolation-based camera measurement model (that can be employed by any visual-inertial estimator), for efficiently processing the non-cloned camera observations. Additionally, we analyze the observability properties of the proposed system and show that scale becomes observable from the visual observations under the employed interpolation-based motion model. Finally, the paper provides accuracy comparison of the proposed VINS against its monocular and stereo counterparts and shows that, in terms of estimation accuracy, the alternating-stereo system either outperforms or is on par with the monocular and stereo VINS that have comparable or higher computational requirements.

References

[1] HTC VIVE VR headset, https://www.vive.com/us.

[2] Microsoft HoloLens augmented reality headset, https://www.microsoft.com/en-us/hololens.

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

[4] NVIDIA Jetson TX2, https://www.nvidia.com/en-us/autonomous-machines/embedded-systems-dev-kits-modules.

[5] Samsung Galaxy Note 8 smart-phone, https://www.samsung.com/us/galaxy/note8.

[6] T. D. Barfoot, C. H. Tong, and S. Särkkä. Batch continuous-time trajectory estimation as exactly sparse gaussian process regression. In *Proc. of the Robotics: Science and Systems Conference*, Berkeley, CA, July 14 – 16 2014.

[7] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart. Robust visual inertial odometry using a direct EKF-based approach. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 298–304, Hamburg, Germany, Sept. 28 – Oct. 2 2015.

[8] M. Bosse and R. Zlot. Continuous 3d scan-matching with a spinning 2d laser. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 4312–4319, Kobe, Japan, May 12 – 17 2009.

[9] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart. The EuRoC micro aerial vehicle datasets. *International Journal of Robotics Research*, 35(10):1157–1163, Sept. 2016.

[10] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza. SVO: Semidirect visual odometry for monocular and multicamera systems. *IEEE Transactions on Robotics*, 33(2):249–265, Apr. 2017.

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

[12] 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 14 – 16 2014.

[13] 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*, 30(1):158–176, Feb. 2014.

[14] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. of the IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 225–234, Nava, Japan, Nov.13 – 16 2007.

[15] 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*, pages 16.1–16.11, Dundee, Scotland, Aug. 29 - Sept. 2 2011.

[16] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale. Keyframe-based visual-inertial odometry using nonlinear optimization. *International Journal of Robotics Research*, 34(3):314–334, Feb. 2015.

[17] S. Lovegrove, A. Patron-Perez, and G. Sibley. Spline fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras. In *Proc. of the British Machine Vision Conference*, Bristol, UK, Sept. 9 – 13 2013.

[18] T. Manderson, F. Shkurti, and G. Dudek. Texture-aware SLAM using stereo imagery and inertial information. In *Proc. of the Conference on Computer and Robot Vision*, BC, Canada, June 1 – 3 2016.

[19] C. D. Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM: Society for Industrial and Applied Mathematics, 2001.

[20] 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*, pages 3482–3489, Rome, Italy, Apr. 10–14 2007.

[21] M. K. Paul, K. J. Wu, J. A. Hesch, E. D. Nerurkar, and S. I. Roumeliotis. A comparative analysis of tightly-coupled monocular, binocular, and stereo vins. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 165–172, Singapore, May 29 – June 3 2017.

[22] T. Qin, P. Li, and S. Shen. VINS-Mono: A robust and versatile monocular visual-inertial state estimator. *CoRR*, abs/1708.03852, Aug. 13 2017.

[23] K. J. 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 the Robotics: Science and Systems*, Rome, Italy, July 13 – 17 2015.

[24] K. J. Wu and S. I. Roumeliotis. Unobservable directions of vins under special motions. Technical report, University of Minnesota, Dept. of Comp. Sci. & Eng., Sept. 2016.