On the VINS Resource-Allocation Problem for a Dual-Camera, Small-Size Quadrotor

Kejian J. Wu(✉), Tien Do, Luis C. Carrillo-Arce, and Stergios I. Roumeliotis

MARS Lab, University of Minnesota, Minneapolis, MN 55455, USA
{kejian,stergios}@cs.umn.edu, {doxxx104,carrillo}@umn.edu

Abstract. In this paper, we present a novel resource-allocation problem formulation for vision-aided inertial navigation systems (VINS) for efficiently localizing micro aerial vehicles equipped with two cameras pointing at different directions. Specifically, based on the quadrotor's current speed and median distances to the features, the proposed algorithm efficiently distributes processing resources between the two cameras by maximizing the expected information gain from their observations. Experiments confirm that our resource-allocation scheme outperforms alternative naive approaches in achieving significantly higher VINS positioning accuracy when tested onboard quadrotors with severely limited processing resources.

1 Introduction and Related Work

In order for micro aerial vehicles (MAVs) to autonomously navigate within GPS-denied areas (e.g., indoors), they need to reliably and efficiently estimate their 3D position and orientation (pose) based on onboard measurements from small-size, lightweight sensors such as cameras and inertial measurement units (IMUs). Previous work on vision-aided inertial navigation systems (VINS) for quadrotors has primarily considered either *forward* or *downward*-pointing cameras in conjunction with an IMU. In [10], for example, a single forward-facing camera is employed for performing visual-inertial odometry, while in [15] a stereo pair mounted in front of the quadrotor is used for localizing. On the other hand, [5,16] focus on efficiently fusing point-feature observations from a downward-pointing camera with inertial measurements, while [6] combines optical flow with IMU data for estimating the vehicle's linear and rotational velocity.

Many quadrotors, however, (e.g., Parrot's Bebop) are equipped with *multiple cameras* pointing at *different directions* (see Fig. 1). In such cases, the downward camera is typically used for optical-flow estimation and position stabilization, while the forward one is often employed for pose determination and navigation (e.g., [3]). As shown in [12], combining visual observations from two or more cameras spanning different viewing directions can lead to significant motion-information gains. Such systems, comprising *two stereo pairs*, have been employed for determining a quadrotor's pose using all feature observations

This work was supported by the Air Force Office of Scientific Research (FA9550-10-1-0567) and the National Science Foundation (IIS-1111638).

D. Kulić et al. (eds.), *2016 International Symposium on Experimental Robotics*,
Springer Proceedings in Advanced Robotics 1, DOI 10.1007/978-3-319-50115-4_47

jointly [7], or separately [14] for first estimating each stereo-pair's pose and then combining their estimates for computing the vehicle's pose. A quadrotor localization system that employs two monocular, differently-pointing cameras is that of [4]. In particular, the optical flow from the downward camera along with the altimeter data are used for estimating the horizontal velocity and resolving the scene's scale. This information is then combined with the attitude estimates from the IMU and image data from the front camera to perform (on a remote laptop) PTAM-based localization [8] along short paths (∼16 m).

Besides the lack of a VINS that directly combines observations from two cameras with different viewing directions for estimating the pose of a MAV, very little is known about how to optimize the information gain from each camera. In particular, most prior work on feature selection for improving localization accuracy has considered one or two cameras pointing in the same direction (e.g., [2,9,18]). Moreover, existing approaches, although they solve a relaxed version of the computationally-intractable optimal problem, their processing requirements often exceed the computational resources of small-size quadrotors.

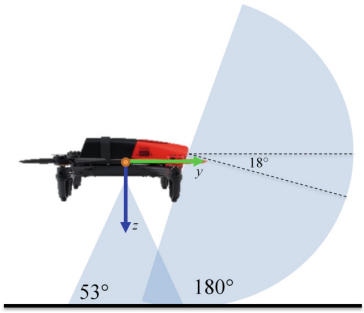

Fig. 1. The Bebop cameras' fov.

To address these limitations, the main contributions of this work are as follows:

- We introduce a novel *resource-allocation* problem formulation, which considers the vehicle's current speed and median distance to the features detected by each camera, as well as approximate models of the impact that each of these parameters has on the *expected information gain*, so as to efficiently distribute processing resources between the two cameras' observations.
- We extend the square-root inverse sliding window filter (SR-ISWF) of [17] to process visual observations from both cameras of the Bebop quadrotor.[1]
- We experimentally validate our proposed, highly efficient, resource-allocation scheme and demonstrate that it allows the VINS algorithm to achieve superior positioning accuracy as compared to alternative approaches (using only one of the two cameras, or processing the same number of features from both of them), while operating in real-time and well within (∼50% of the CPU time) the severely limited computational resources of the Bebop quadrotor.

2 Technical Approach

Assume that a quadrotor is equipped with two cameras, namely a forward-facing camera, $\{C_f\}$, and a downward-pointing camera, $\{C_d\}$.[2] Features are extracted

[1] Although the two cameras' fov have a small overlap, we do not match features between them as the different camera characteristics make such process unreliable.

[2] Note that although the ensuing presentation focuses on the specific (forward and downward) configuration of the cameras onboard the Bebop quadrotor used in our

and tracked across image sequences for each camera independently. Selecting the most informative features for localization (i.e., so as to minimize the posterior covariance of the quadrotor's pose estimates), would require solving the following optimization problem:

$$\min_{u_i,v_j\in\{0,1\}} \ \mathrm{tr}\left(\mathbf{P}^{\ominus -1} + \sum_{i\in\mathcal{S}_f} u_i \frac{1}{\sigma_f^2}\mathbf{H}_i^T\mathbf{H}_i + \sum_{j\in\mathcal{S}_d} v_j \frac{1}{\sigma_d^2}\mathbf{H}_j^T\mathbf{H}_j\right)^{-1} \qquad (1)$$
$$\text{s.t.} \quad \sum_{i\in\mathcal{S}_f} u_i + \sum_{j\in\mathcal{S}_d} v_j \leq \gamma$$

where $\mathbf{P}^{\ominus}$ denotes the prior covariance of the quadrotor's pose estimates, $\mathbf{H}_i$ is the feature i's measurement Jacobian with respect to the pose state, σ_f and σ_d are the measurement noise standard deviations, $\mathcal{S}_f$ and $\mathcal{S}_d$ are the sets of features observed by C_f and C_d, respectively, and γ is the maximum number of features that can be processed at each time step.

Since (1) is an integer programming problem with NP complexity, prior approaches (e.g., [1]) relax it by ignoring $\mathbf{P}^{\ominus}$ and allowing $u_i, v_j \in [0\ 1]$ so that it becomes convex and can be cast as a semidefinite program (SDP). Its cost, however, remains prohibitively high $O((|\mathcal{S}_f|+|\mathcal{S}_d|)^3)$. Although alternative approximate formulations achieve lower complexity ($O((|\mathcal{S}_f|+|\mathcal{S}_d|)^2)$ for [2] and $O(\gamma(|\mathcal{S}_f|+|\mathcal{S}_d|))$ for [18]), their processing requirements are still quite high for the Bebop's limited resources.

For this reason, in this work we introduce further approximations to (1) so as to derive a constant-cost solution. Specifically, in order to avoid explicitly evaluating each feature's measurement Jacobian, we focus on the *expected value* of the original cost function in (1), over a particular distribution (to be specified later on) of the positions of the features viewed by each camera:

$$\mathbb{C}(\lambda) = \ \mathbb{E}\left[\mathrm{tr}\left(\lambda\frac{1}{\sigma_f^2}\mathbf{H}_i^T\mathbf{H}_i + (1-\lambda)\frac{1}{\sigma_d^2}\mathbf{H}_j^T\mathbf{H}_j\right)^{-1}\right], \quad \lambda\in[0\ 1]. \qquad (2)$$

Furthermore, by employing Jensen's inequality and the fact that the function $\mathrm{tr}(\mathbf{X}^{-1})$ is convex [1], it is straightforward to show that $\mathbb{C}(\lambda)$ in (2) has the following lower bound:

$$\begin{aligned}\mathbb{C}_{lb}(\lambda) &= \mathrm{tr}\left(\mathbb{E}\left[\lambda\frac{1}{\sigma_f^2}\mathbf{H}_i^T\mathbf{H}_i + (1-\lambda)\frac{1}{\sigma_d^2}\mathbf{H}_j^T\mathbf{H}_j\right]\right)^{-1} \\ &= \mathrm{tr}\left(\lambda\mathbb{E}\left[\frac{1}{\sigma_f^2}\mathbf{H}_i^T\mathbf{H}_i\right] + (1-\lambda)\mathbb{E}\left[\frac{1}{\sigma_d^2}\mathbf{H}_j^T\mathbf{H}_j\right]\right)^{-1} \qquad (3)\end{aligned}$$

experiments, our approach is applicable to any dual-camera system with arbitrary geometric configuration.

By defining the *expected information gain* of a feature measurement, from the forward or downward-pointing camera, with respect to the pose state as:

$$\hat{\boldsymbol{\mathcal{I}}}_f = \mathbb{E}\left[\frac{1}{\sigma_f^2}\mathbf{H}_i^T\mathbf{H}_i\right], \quad \hat{\boldsymbol{\mathcal{I}}}_d = \mathbb{E}\left[\frac{1}{\sigma_d^2}\mathbf{H}_j^T\mathbf{H}_j\right] \tag{4}$$

and substituting into the relaxed cost function $\mathbb{C}_{lb}(\lambda)$ in (3), our proposed optimization problem can be written as:

$$\min_{\lambda\in[0\ \ 1]} \ \mathrm{tr}\left(\lambda\hat{\boldsymbol{\mathcal{I}}}_f + (1-\lambda)\hat{\boldsymbol{\mathcal{I}}}_d\right)^{-1} \tag{5}$$

which represents a resource-allocation problem between the two cameras. Note that, once the optimal percentage of resources is allocated to each camera (i.e., the optimal value λ^* is obtained), we then select features *within each camera* by employing the approach of [9]; i.e., we enforce uniform feature extraction during image processing and select the ones with the longest tracks.

As compared to (1), the relaxed optimization problem in (5) has only one scalar variable λ. Furthermore, since the matrices $\hat{\boldsymbol{\mathcal{I}}}_f$ and $\hat{\boldsymbol{\mathcal{I}}}_d$ have a fixed size and can be efficiently computed, (5) can be solved in constant time that only depends on the matrices' size, regardless of the number of features available from each camera. Thus, and in order to reduce complexity, we first assume that the features' positions can be accurately triangulated from their first two observations, and then used for localizing the rest of the camera poses in the estimator's optimization window [17]. Moreover, we ignore the cameras' orientation, i.e., their (i) roll and pitch, as they are observable and can be precisely estimated [typically, with root mean square error (RMSE) of 0.1°], and (ii) yaw, as its impact over a short time horizon (i.e., the 1 s corresponding to the remaining 4 poses in the estimator's sliding window) is very small for any error, due to the gyro noise, to become significant. As a result of these relaxations, the measurement Jacobian $\mathbf{H}_i$ is now determined with respect to only the downward-camera's position state,[3] and hence the size of the information matrices $\hat{\boldsymbol{\mathcal{I}}}_f$ and $\hat{\boldsymbol{\mathcal{I}}}_d$ becomes 3-by-3. Based on these approximations, in what follows, we present a closed-form expression for evaluating these two matrices.

In order to compute the expected information gain from each feature measurement, as defined in (4), we introduce certain simplifying assumptions about the spatial distribution of the features observed by the two cameras. We start by parameterizing every feature i with respect to the camera s, where $s \in \{f, d\}$ is the camera index, by its spherical coordinates (the azimuth angle ϕ_i, the polar angle θ_i, and the distance ρ_i). Assuming that all features are (i) located on a spherical cap of radius equal to the median distance, ρ_s, of the features currently observed, and (ii) uniformly distributed over the angles ϕ_i and θ_i, i.e.,

$$\rho_i = \rho_s, \quad \phi_i \sim \mathrm{U}[0, 2\pi], \quad \theta_i \sim \mathrm{U}[0, \theta_{Ms}] \tag{6}$$

[3] Without loss of generality, we choose the quadrotor's frame of reference to be the one of the downward camera.

where θ_{Ms} equals half of the field of view (fov) of the camera s, it can be shown (see Appendix A) that the expected information gain becomes:

$$\hat{\mathcal{I}}_f = {}^{C_d^{k'}}_{C_d^k}\mathbf{R}^T {}^{C_f}_{C_d}\mathbf{R}^T \mathbf{D}_f {}^{C_f}_{C_d}\mathbf{R} {}^{C_d^{k'}}_{C_d^k}\mathbf{R}, \qquad \hat{\mathcal{I}}_d = {}^{C_d^{k'}}_{C_d^k}\mathbf{R}^T \mathbf{D}_d {}^{C_d^{k'}}_{C_d^k}\mathbf{R} \tag{7}$$

$$\text{with} \qquad \mathbf{D}_s = \frac{\tan\theta_{Ms}}{\rho_s^2 \theta_{Ms} \sigma_s^2} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{\tan^2\theta_{Ms}}{3} \end{bmatrix}, \quad s \in \{f, d\} \tag{8}$$

where k' and k denote the time steps when a feature measurement is considered and when it is first observed, respectively, ${}^{C_d^{k'}}_{C_d^k}\mathbf{R}$ represents the rotation matrix between the downward-camera frames corresponding to these two time steps, and ${}^{C_f}_{C_d}\mathbf{R}$ is the extrinsic-calibration rotation matrix between the forward and downward cameras. By substituting (7) into (5), the cost function becomes:

$$\begin{aligned} \mathbb{C}_{lb}(\lambda) &= \operatorname{tr}\left(\lambda {}^{C_d^{k'}}_{C_d^k}\mathbf{R}^T {}^{C_f}_{C_d}\mathbf{R}^T \mathbf{D}_f {}^{C_f}_{C_d}\mathbf{R} {}^{C_d^{k'}}_{C_d^k}\mathbf{R} + (1-\lambda) {}^{C_d^{k'}}_{C_d^k}\mathbf{R}^T \mathbf{D}_d {}^{C_d^{k'}}_{C_d^k}\mathbf{R} \right)^{-1} \\ &= \operatorname{tr}\left(\lambda {}^{C_f}_{C_d}\mathbf{R}^T \mathbf{D}_f {}^{C_f}_{C_d}\mathbf{R} + (1-\lambda)\mathbf{D}_d \right)^{-1} = \frac{f(\lambda)}{g(\lambda)} \end{aligned} \tag{9}$$

where $f(\lambda)$ and $g(\lambda)$ are quadratic and cubic polynomial functions, respectively, of $\lambda \in [0\ 1]$. To minimize (9), we first compute all the stationary points of the unconstrained optimization problem, which requires solving the quartic equation $f'(\lambda)g(\lambda) - g'(\lambda)f(\lambda) = 0$. Then, the optimal solution λ^* is the one that yields the minimal cost value $\mathbb{C}_{lb}(\lambda^*)$ among all feasible ($\lambda^* \in [0\ 1]$) stationary points, computed in closed form, together with the boundary values 0 and 1. Figure 2 (left) illustrates the optimal values of λ^* for different median feature distances ρ_s. As evident, three regions emerge: (I) for $\rho_f/\rho_d \geq 2$ and (III) for $\rho_f/\rho_d \leq 1.15$ where all processing is allocated to the downward or forward camera, respectively, while in region (II) features from both cameras are processed.

At this point, we should note that the preceding formulation does not consider the impact of the quadrotor's *motion* on the expected information gain. In particular, due to the limited fov and close distance to the ground, the track length of the downward-camera's features is quite limited as compared to the front one's. Moreover, reliably tracking features from the downward camera becomes exceedingly difficult as the quadrotor's speed increases [see Fig. 2 (right)]. To account for the track length's impact, we modify the cost function in (9) as:

$$\mathbb{C}'_{lb}(\lambda) = \operatorname{tr}\left(\lambda \psi_f {}^{C_f}_{C_d}\mathbf{R}^T \mathbf{D}_f {}^{C_f}_{C_d}\mathbf{R} + (1-\lambda)\psi_d \mathbf{D}_d \right)^{-1} \tag{10}$$

where ψ_f and ψ_d are the expected feature-track lengths (minus 2, since the first two observations are used for triangulating the feature and do not provide information for localizing the cameras) expressed as functions of the quadrotor's speed based on prior data [see Fig. 2 (right)]. This modification is motivated by the fact that, in general, the longer a feature track is, the more information it will provide to the sliding-window estimator for determining the camera's

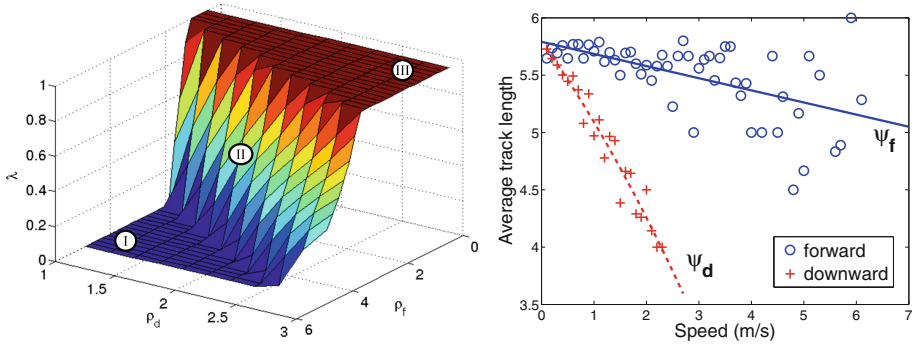

Fig. 2. (left) Optimal resource allocation λ^* for $\rho_f \in [0.2\ 5]$ m and $\rho_d \in [1\ 3]$ m. (right) Average feature track lengths for each camera at different speeds. The blue-solid and red-dashed lines are the fitted linear and quadratic functions ψ_f and ψ_d, respectively.

position. Besides this consideration, and due to the fact that the downward-camera's feature tracking becomes unreliable for speeds higher than 2 m/s, we only consider features from the forward camera during such fast motions.

Lastly, once the number of features that will be processed from each camera is determined, i.e., $\lambda^*\gamma$ and $(1-\lambda^*)\gamma$ for the forward and downward camera, respectively, we employ the method of [9] for selecting the most informative ones within each camera.[4]

3 Experimental Results

To examine the impact of the proposed resource-allocation algorithm on the localization accuracy of VINS, we compared our approach against three naive allocation schemes using as testing platform a MAV. Specifically, the Bebop quadrotor carries an IMU, a 180° fov forward camera with resolution downsampled to 300×264, a 53° fov downward camera with resolution downsampled to 320×240, and a 800 MHz ARM-based dual-core processor. Approximately 200 FAST corners [13] are extracted from the images, and tracked using the Kanade-Lucas-Tomasi (KLT) algorithm [11] across time at a frequency of 15 Hz. The SR-ISWF estimator [17] maintains a sliding window of 6 poses, selected at 5 Hz.

For testing our method, we collected two building-scale datasets (path length ∼200 m each) while manually flying the quadrotor at fast speeds (up to 6 m/s) through open spaces, with features far away from the forward camera, as well as during slow motions, including rotations in place, while navigating through narrow passages with nearby scenes. Since the Bebop's processing resources are quite

[4] Through experimentation, [9] has been shown to offer a very efficient and accurate metric for assessing the expected information gain from each feature.

limited, we allowed the SR-ISWF to process up to 20 features and compared the achieved localization accuracy against the batch least-squares (BLS) estimates (computed offline) for the following configurations: (i) *f-only*: 20 MSCKF features are used from only the forward camera;[5] (ii) *f-SLAM*: 10 MSCKF and 10 SLAM features are used from the forward camera; (iii) *fd-EF*: resources are *equally* distributed between the two cameras by *fixing* the number of MSCKF features processed by each of them to 10 (20 total); and (iv) the proposed *fd-D* where 20 MSCKF features are dynamically selected from the two cameras.

The resource-allocation results of the proposed approach are depicted in Fig. 3, where the optimal λ^* that minimizes the cost function in (10) is plotted, along with the speed of the quadrotor, against time. As evident, our resource-allocation scheme is able to properly adjust to the different motions and scene distances. Specifically, when the quadrotor is flying fast (e.g., during time steps 210–260), only the forward camera is used ($\lambda = 1$) since no features can be reliably tracked across the downward-camera's images. On the other hand, when the quadrotor navigates through narrow passages with nearby scenes (e.g., many times between time steps 400 and 750), observations from both cameras are used ($0 < \lambda < 1$). Lastly, when the quadrotor rotates in place and the scene observed by the forward camera is distant (e.g., many times between time steps 350 and 600), only the downward camera is used ($\lambda = 0$) to maximize the positioning accuracy.

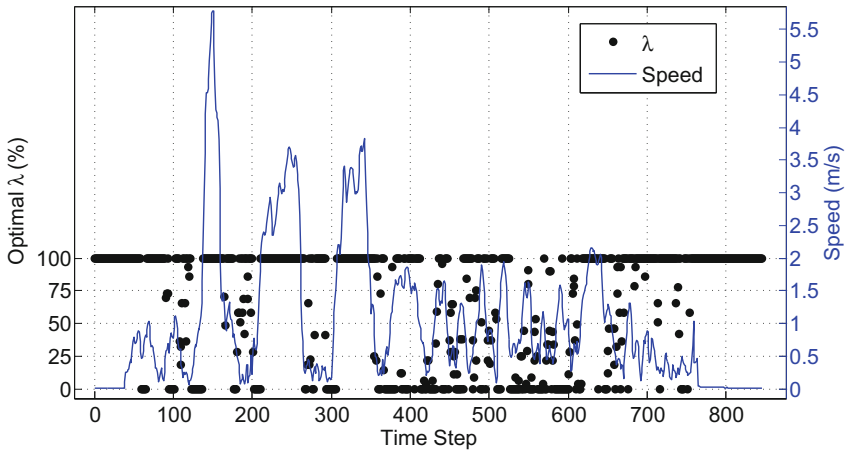

Fig. 3. The percentage of resources allocated to the forward camera (i.e., optimal value of λ, shown as black dots) plotted along with the speed of the quadrotor (solid blue line) against time steps, each of duration 0.2 s.

[5] MSCKF features are marginalized by the SR-ISWF for performing visual-inertial odometry without including their estimates in the filter's state; see [17] for details.

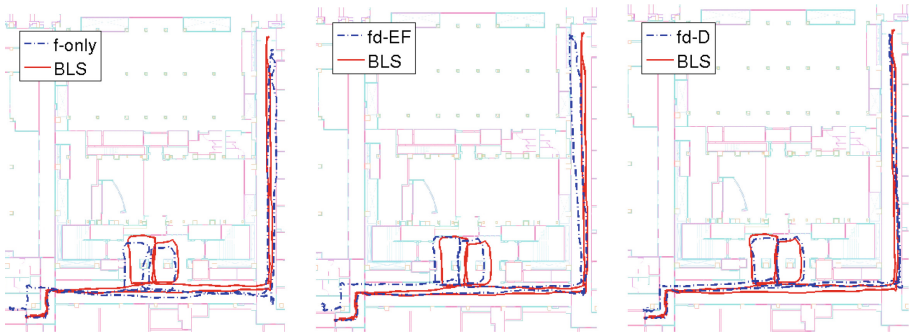

Fig. 4. Estimated trajectories for three of the four resource-allocation schemes considered against the BLS groundtruth overlaid on the building's blueprint.

Table 1. VINS RMSE for the 4 resource-allocation schemes considered.

RMSE (m)	f-only	f-SLAM	fd-EF	fd-D
Dataset 1	2.96	2.38	2.67	**1.22**
Dataset 2	2.72	2.70	2.87	**2.11**

In order to assess the impact on the VINS localization accuracy, the root mean square error (RMSE) of the estimated 3D position for each of the four resource-allocation schemes considered is shown in Table 1, while the estimated trajectories for three of them, as well as the BLS groundtruth, overlaid on the building's blueprint are depicted in Fig. 4. As evident from Table 1, by adjusting the allocation of processing resources based on the vehicle's speed and the median distance to each camera's corresponding scene, significant gains in accuracy (0.59–1.16 m lower RMSE) are realized as compared to when using only one of the two cameras, or processing the same number of features from each of them.[6] This key finding is also visually confirmed by the trajectories shown in Fig. 4 where the one estimated by the SR-ISWF when employing the proposed dynamic resource-allocation scheme best aligns with the BLS groundtruth.

Lastly, we note that the dual-camera SR-ISWF runs onboard the Bebop quadrotor and takes less than 100 ms per estimate. Specifically, 6 ms for FAST feature extraction, 36 ms for KLT tracking, 2 ms for RANSAC, and 50 ms for a SR-ISWF update. Since the filter runs at 5 Hz, the overall processing takes ∼500 ms of every second. The remaining processing is reserved for future autonomous navigation tasks such as obstacle detection/avoidance, path planning, and exploration. Videos of the presented experiments can be found at http://mars.cs.umn.edu/research/dual_camera_quadrotor.php.

[6] We do not evaluate the RMSE for the case of only downward-pointing camera since the quadrotor's CPU cannot perform image processing at the high frame rates (40 Hz) required for tracking features at high speeds (6 m/s).

4 Conclusions

In this paper, we considered the problem of visual-information selection for efficiently localizing a dual-camera MAV. In particular, instead of addressing the computationally-intractable problem of selecting the most informative feature measurements, we focused on optimally distributing processing resources between the two cameras. To this end, we introduced a novel problem formulation that seeks to maximize the expected information gain based on each camera's characteristics (fov and noise standard deviation), their geometric configuration, the median features' distance, and the vehicle's speed. Moreover, by employing simplifying assumptions about the spatial distribution of the features viewed by each camera, we showed that the optimal solution to the resource-allocation problem can be found in constant time, by solving, in closed form, the quartic equation resulting from the optimality conditions. Our approach was tested experimentally using a small-size quadrotor flying indoors over a wide range of motions and scene distances. In all cases considered, the proposed resource-allocation scheme allowed the VINS algorithm to operate in real time while achieving positioning accuracy superior to that of naive approaches that employ only one of the two cameras, or equally distribute the quadrotor's processing resources among them.

Appendix A

In order to compute the expected information matrices in (7), we start by deriving the measurement Jacobian $\mathbf{H}_i$, appearing in (4), at time step k'. Consider a feature i, observed by the camera s, $s \in \{f, d\}$, whose position, $\mathbf{p}_i$, with respect to the camera frame $\{C_s^{k'}\}$, is:

$$ {}^{C_s^{k'}}\mathbf{p}_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} \rho_i \sin\theta_i \cos\phi_i \\ \rho_i \sin\theta_i \sin\phi_i \\ \rho_i \cos\theta_i \end{bmatrix} \tag{11} $$

where $[x_i, y_i, z_i]^T$ and $[\phi_i, \theta_i, \rho_i]^T$ are the feature's Cartesian and spherical coordinates, respectively. The camera measures the perspective projection of feature i:

$$ \mathbf{z} = \pi\left({}^{C_s^{k'}}\mathbf{p}_i\right) + \mathbf{n}_i = \begin{bmatrix} \frac{x_i}{z_i} \\ \frac{y_i}{z_i} \end{bmatrix} + \mathbf{n}_i, \qquad {}^{C_s^{k'}}\mathbf{p}_i = {}^{C_s^{k'}}_{C_s^{k}}\mathbf{R}({}^{C_s^{k}}\mathbf{p}_i - {}^{C_s^{k}}\mathbf{p}_{C_s^{k'}}) \tag{12} $$

where $\mathbf{n}_i$ is the measurement noise and ${}^{C_s^{k}}\mathbf{p}_i$ denotes the feature's position with respect to the first-observing camera frame, $\{C_s^k\}$, at time step k, while ${}^{C_s^{k'}}_{C_s^{k}}\mathbf{R}$ and ${}^{C_s^{k}}\mathbf{p}_{C_s^{k'}}$ represent the rotation matrix and translation vector, respectively, between the camera frames at the corresponding time steps k and k'. Based on (12), the measurement Jacobian with respect to the camera's position is:

$$ \mathbf{H}_i = \frac{\partial\pi\left({}^{C_s^{k'}}\mathbf{p}_i\right)}{\partial {}^{C_s^{k'}}\mathbf{p}_i} \frac{\partial {}^{C_s^{k'}}\mathbf{p}_i}{\partial {}^{C_s^{k}}\mathbf{p}_{C_s^{k'}}} = -\frac{1}{\rho_i \cos\theta_i} \begin{bmatrix} 1 & 0 & -\tan\theta_i \cos\phi_i \\ 0 & 1 & -\tan\theta_i \sin\phi_i \end{bmatrix} {}^{C_s^{k'}}_{C_s^{k}}\mathbf{R} \tag{13} $$

which leads to the following information matrix:

$$\frac{1}{\sigma_s^2}\mathbf{H}_i^T\mathbf{H}_i = \frac{1}{\sigma_s^2\rho_i^2\cos^2\theta_i}{}^{C_s^{k'}}_{C_s^k}\mathbf{R}^T \begin{bmatrix} 1 & 0 & -\tan\theta_i\cos\phi_i \\ 0 & 1 & -\tan\theta_i\sin\phi_i \\ -\tan\theta_i\cos\phi_i & -\tan\theta_i\sin\phi_i & \tan^2\theta_i \end{bmatrix} {}^{C_s^{k'}}_{C_s^k}\mathbf{R} \tag{14}$$

By employing the assumptions about the features' distribution in (6), and substituting (14) into (4), yields:

$$\begin{aligned}
\hat{\boldsymbol{\mathcal{I}}}_s &= \frac{1}{\sigma_s^2\rho_s^2}{}^{C_s^{k'}}_{C_s^k}\mathbf{R}^T\mathbb{E}\left\{\frac{1}{\cos^2\theta_i}\begin{bmatrix} 1 & 0 & -\tan\theta_i\cos\phi_i \\ 0 & 1 & -\tan\theta_i\sin\phi_i \\ -\tan\theta_i\cos\phi_i & -\tan\theta_i\sin\phi_i & \tan^2\theta_i \end{bmatrix}\right\} {}^{C_s^{k'}}_{C_s^k}\mathbf{R} \\
&= \frac{1}{\sigma_s^2\rho_s^2}{}^{C_s^{k'}}_{C_s^k}\mathbf{R}^T\int_0^{\theta_{Ms}}\int_0^{2\pi}\frac{1}{2\pi\theta_{Ms}\cos^2\theta_i} \\
&\qquad \begin{bmatrix} 1 & 0 & -\tan\theta_i\cos\phi_i \\ 0 & 1 & -\tan\theta_i\sin\phi_i \\ -\tan\theta_i\cos\phi_i & -\tan\theta_i\sin\phi_i & \tan^2\theta_i \end{bmatrix} d\phi_i d\theta_i \; {}^{C_s^{k'}}_{C_s^k}\mathbf{R} \\
&= {}^{C_s^{k'}}_{C_s^k}\mathbf{R}^T\frac{\tan\theta_{Ms}}{\rho_s^2\theta_{Ms}\sigma_s^2}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{\tan^2\theta_{Ms}}{3} \end{bmatrix} {}^{C_s^{k'}}_{C_s^k}\mathbf{R} = {}^{C_s^{k'}}_{C_s^k}\mathbf{R}^T\mathbf{D}_s{}^{C_s^{k'}}_{C_s^k}\mathbf{R}
\end{aligned} \tag{15}$$

Note that $\mathbf{H}_i$ in (13), and hence $\hat{\boldsymbol{\mathcal{I}}}_s$ in (15), is expressed with respect to the position state, ${}^{C_s^k}\mathbf{p}_{C_s^{k'}}$, of the camera s [see (13)]. Therefore, and since we chose the system's state to comprise the downward-camera's position, ${}^{C_d^k}\mathbf{p}_{C_d^{k'}}$, the expected information gain from the corresponding feature observations is obtained by directly setting $s = d$ in (15), i.e.,

$$\hat{\boldsymbol{\mathcal{I}}}_d = {}^{C_d^{k'}}_{C_d^k}\mathbf{R}^T\mathbf{D}_d{}^{C_d^{k'}}_{C_d^k}\mathbf{R} \tag{16}$$

On the other hand, the forward-camera's measurement Jacobian also depends on the extrinsics of the two cameras, i.e.,

$$\mathbf{H}_j = \frac{\partial\pi\left({}^{C_f^{k'}}\mathbf{p}_i\right)}{\partial{}^{C_f^{k'}}\mathbf{p}_i}\frac{\partial{}^{C_f^{k'}}\mathbf{p}_i}{\partial{}^{C_f^k}\mathbf{p}_{C_f^{k'}}}\frac{\partial{}^{C_f^k}\mathbf{p}_{C_f^{k'}}}{\partial{}^{C_d^k}\mathbf{p}_{C_d^{k'}}}, \quad \text{where} \quad \frac{\partial{}^{C_f^k}\mathbf{p}_{C_f^{k'}}}{\partial{}^{C_d^k}\mathbf{p}_{C_d^{k'}}} = {}^{C_f}_{C_d}\mathbf{R} \tag{17}$$

results from the geometric relationship between the two cameras across time steps k and k'. By comparing (17) to (13), the forward-camera's Jacobian is obtained by first setting $s = f$ in (13), and then multiplying it, from the right, with the extrinsic-calibration rotation matrix ${}^{C_f}_{C_d}\mathbf{R}$. Consequently, the expected information gain from the forward camera becomes:

$$\hat{\boldsymbol{\mathcal{I}}}_f = {}^{C_f}_{C_d}\mathbf{R}^T{}^{C_f^{k'}}_{C_f^k}\mathbf{R}^T\mathbf{D}_f{}^{C_f^{k'}}_{C_f^k}\mathbf{R}{}^{C_f}_{C_d}\mathbf{R} \tag{18}$$

Lastly, employing the geometric relationship ${}^{C_f^{k'}}_{C_f^k}\mathbf{R}\,{}^{C_f}_{C_d}\mathbf{R} = {}^{C_f}_{C_d}\mathbf{R}\,{}^{C_d^{k'}}_{C_d^k}\mathbf{R}$ in (18) results in the expression for $\hat{\boldsymbol{\mathcal{I}}}_f$ shown in (7).

References

1. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, New York (2004)
2. Davison, A.J.: Active search for real-time vision. In: Proceedings of the IEEE International Conference on Computer Vision, Beijing, China, pp. 66–73, 17–21 October 2005
3. Do, T., Carrillo-Arce, L.C., Roumeliotis, S.I.: Autonomous flights through image-defined paths. In: Proceedings of the International Symposium of Robotics Research, Sestri Levante, Italy, 12–15 September 2015
4. Engel, J., Sturm, J., Cremers, D.: Camera-based navigation of a low-cost quadrotor. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Algarve, Portugal, pp. 2815–2821, 7–12 October 2012
5. Forster, C., Pizzoli, M., Scaramuzza, D.: SVO: fast semi-direct monocular visual odometry. In: Proceedings of the IEEE International Conference on Robotics and Automation, Hong Kong, China, pp. 15–22, 31 May–5 June 2014
6. Grabe, V., Bülthoff, H.H., Scaramuzza, D., Giordano, P.R.: Nonlinear ego-motion estimation from optical flow for online control of a quadrotor UAV. Int. J. Robot. Res. **34**(8), 1114–1135 (2015)
7. Heng, L., Lee, G.H., Pollefeys, M.: Self-calibration and visual SLAM with a multi-camera system on a micro aerial vehicle. Autonomous Robots **39**(3), 259–277 (2015)
8. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, Japan, pp. 225–234, 13–16 November 2007
9. Kottas, D.G., DuToit, R.C., Ahmed, A., Guo, C.X., Georgiou, G., Li, R., Roumeliotis, S.I.: A resource-aware vision-aided inertial navigation system for wearable and portable computers. In: Proceedings of the IEEE International Conference on Robotics and Automation, Hong Kong, China, pp. 6336–6343, 31 May– 5 June 2014
10. Loianno, G., Mulgaonkar, Y., Brunner, C., Ahuja, D., Ramanandan, A., Chari, M., Diaz, S., Kumar, V.: Smartphones power flying robots. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Hamburg, Germany, pp. 1256–1263, 28 September– 2 October 2015
11. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the International Joint Conference on Artificial Intelligence, Vancouver, British Columbia, pp. 674–679, 24–28 August 1981
12. Pless, R.: Using many cameras as one. In: Proceeding of the IEEE International Conference on Computer Vision and Pattern Recognition, Madison, WI, pp. 11–18, 16–22 June 2003
13. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 430–443. Springer, Heidelberg (2006). doi:10.1007/11744023_34

14. Schauwecker, K., Zell, A.: On-board dual-stereo-vision for the navigation of an autonomous MAV. J. Intell. Robot. Syst. **74**(1–2), 1–16 (2014)
15. Shen, S., Mulgaonkar, Y., Michael, N., Kumar, V.: Vision-based state estimation and trajectory control towards high-speed flight with a quadrotor. In: Proceedings of the Robotics: Science and Systems, Berlin, Germany, 24–28 June 2013
16. Weiss, S., Achtelik, M.W., Lynen, S., Achtelik, M.C., Kneip, L., Chli, M., Siegwart, R.: Monocular vision for long-term micro aerial vehicle state estimation: a compendium. J. Field Robot. **30**(5), 803–831 (2013)
17. Wu, K.J., Ahmed, A., Georgiou, G., Roumeliotis, S.I.: A square root inverse filter for efficient vision-aided inertial navigation on mobile devices. In: Proceedings of Robotics: Science and Systems, Rome, Italy, 13–17 July 2015
18. Zhang, G., Vela, P.A.: Good features to track for visual SLAM. In: Proceedings of the IEEE InternationaL Conference on Computer Vision and Pattern Recognition, Boston, MA, pp. 1373–1382, 7–12 June 2015