A Convex Formulation for Motion Estimation using Visual and Inertial Sensors

Mingyang Li and Anastasios I. Mourikis
Dept. of Electrical and Computer Engineering, University of California, Riverside
E-mail: mli@ee.ucr.edu, mourikis@ee.ucr.edu

***Abstract*—Most existing algorithms for vision-aided inertial navigation rely on linearization, and thus require good initial estimates of the state to operate reliably. In this paper, we present a method for computing such estimates *in absence of prior motion information*, by fusing the inertial measurements and observations of naturally-occurring point features extracted from images. Specifically, we propose a convex-minimization formulation, which is derived as an approximation to the optimal maximum-a-posteriori estimator. In this formulation, both the inertial and visual measurements are jointly used, and a robust cost function (bivariate Huber) is employed to provide robustness to outliers. Experimental results on both simulated and real-world data demonstrate that the proposed approach outperforms competing methods by a significant margin.**

I. Introduction

In recent years, the topic of motion estimation using visual and inertial sensors (termed *vision-aided inertial navigation*) has attracted considerable research interest. Numerous estimators have been proposed for fusing the measurements from a camera and an inertial measurement unit (IMU), with the vast majority of algorithms employing either an extended Kalman filter (EKF) [1]–[4], or iterative minimization [5]–[7]. A common characteristic of all these methods, which rely on linearization, is that they require an accurate initial estimate of the state for successful operation. While this is true in any linearization-based estimator, in vision-aided inertial navigation an accurate initial guess is especially important, as the initial orientation and velocity greatly influence the estimated trajectory. This makes the in-motion initialization of visual-inertial estimators a challenging problem [8]–[11].

In this work, we present a convex-minimization algorithm for computing an estimate of the trajectory using inertial measurements and monocular observations of naturally-occurring point features. The method is direct, in the sense that it does not require a prior estimate of the motion. Moreover, it employs a robust cost-function formulation, to permit reliable operation in the presence of outliers. Therefore, it can be employed for in-motion initialization of an EKF estimator, for providing an initial guess for linearization-based estimators, or for re-starting an estimator after failure.

We point out that, even though direct methods for visual-inertial motion estimation are required in many settings, prior work on the subject is limited. In [8], a loosely-coupled estimator is presented, in which the measurements of the inertial and visual sensors are processed separately. First, a structure-from-motion (SfM) algorithm is used to derive a motion estimate up to an unknown scale factor. Then, this estimate is used together with the inertial measurements to compute the scale and the orientation with respect to the horizontal plane (i.e., with respect to gravity). In this approach, obtaining an initial estimate for the motion and dealing with outliers are left to the SfM algorithm, and not addressed. By contrast, in our work the visual and inertial measurements are used jointly, for increased robustness to outliers.

A different class of methods is presented in [9], [10], [12], where the visual and inertial measurements are used to derive linear equations involving the initial IMU velocity and orientation (represented via the gravity vector expressed in the IMU frame). The linear formulation leads to analytical solutions, but does not correctly model the measurement noise and thus yields estimates that are not statistically optimal. While the approach we describe here is also suboptimal, the cost function we minimize is a close approximation to the cost function of the maximum-a-posteriori (MAP) estimator, and thus results in significantly improved accuracy (see Section V).

Specifically, our algorithm computes estimates for (i) the IMU motion and scene structure with respect to the initial IMU frame, (ii) the gravity vector expressed with respect to the initial IMU frame (equivalent to estimating the initial roll and pitch), and (iii) the accelerometer bias. We note that due to the lack of any feature points with known position, estimation with respect to a "local" frame (chosen here to coincide with the initial IMU frame) is the best one can hope for, as the global position and yaw are unobservable [2], [12]. In our approach, the orientation with respect to the initial IMU frame is computed by integrating the gyroscope measurements, while all other quantities, including the sensor's position, velocity, scene points, accelerometer bias, and gravity vector, are computed via convex minimization.

The formulation we describe draws upon results in convex SfM estimation [13], [14]. Compared to existing work in this area, in our approach the objective function being minimized additionally includes the terms arising from the inertial measurements, and contains *depth-weighted* terms for the camera reprojection errors. This depth-weighted formulation, which is made possible by a pre-processing step to compute estimates of the features' depths from the images, results in improved estimation accuracy, as shown in the experimental results.

II. Sensor Models

We here consider the case where a system comprising an IMU and a monocular camera moves while observing point features with unknown positions. We are primarily interested in the in-motion initialization problem, where the sensors record measurements over a relatively short time interval (e.g.,

a few seconds to a few tens of seconds), and our goal is to use these measurements to compute the quantities described in Section I.

We assume that the camera intrinsics, as well as the relative spatial transformation and timing between the two sensors are known via prior calibration, for example by the approach described in [15]. In this paper, $\{I\}$ is the IMU coordinate frame, $\{C\}$ the frame of the camera, and $\{B\}$ a fixed base frame, selected to coincide with the IMU frame at the start of the motion. The observation, $\mathbf{z}_{ij}$, of the j-th feature, $\mathbf{f}_j$, in the i-th image is described by the perspective camera model[1]:

$$\mathbf{z}_{ij} = \begin{bmatrix} u_{ij} \\ v_{ij} \end{bmatrix} = \mathbf{h}\left({}^{C_i}\mathbf{p}_{f_j}\right) + \mathbf{n}_{ij} = \frac{1}{z_{ij}} \begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix} + \mathbf{n}_{ij} \quad (1)$$

where $\mathbf{n}_{ij}$ is the measurement noise vector, distributed as $\mathbf{n}_{ij} \sim \mathcal{N}\left(\mathbf{0}, \sigma_{im}^2 \mathbf{I}_2\right)$, and ${}^{C_i}\mathbf{p}_{f_j}$ is the position of the feature with respect to the camera frame at the time the i-th image is recorded, given by:

$${}^{C_i}\mathbf{p}_{f_j} = \begin{bmatrix} x_{ij} \\ y_{ij} \\ z_{ij} \end{bmatrix} = {}^{C}_{I}\mathbf{R}\, {}^{I_i}_{B}\mathbf{R}\left({}^{B}\mathbf{p}_{f_j} - {}^{B}\mathbf{p}_{I_i}\right) + {}^{C}\mathbf{p}_{I} \quad (2)$$

In the above equations, ${}^{C}\mathbf{p}_I$ and ${}^{C}_{I}\mathbf{R}$ are the *known* translation and rotation between the IMU and camera.

The IMU's gyroscopes and accelerometers measure rotational velocity, $\boldsymbol{\omega}_m$, and specific force, $\mathbf{a}_m$, respectively:

$$\boldsymbol{\omega}_m(t) = {}^{I}\boldsymbol{\omega}(t) + \mathbf{b_g} + \mathbf{n_g}(t) \quad (3)$$

$$\mathbf{a}_m(t) = {}^{B}_{I}\mathbf{R}^T(t)\left({}^{B}\mathbf{a}(t) - {}^{B}\mathbf{g}\right) + \mathbf{b_a} + \mathbf{n_a}(t) \quad (4)$$

Here ${}^{I}\boldsymbol{\omega}$ represents the IMU's rotational velocity; ${}^{B}\mathbf{a}$ is the IMU's linear acceleration; ${}^{B}\mathbf{g}$ denotes the gravity vector; $\mathbf{n_g}$ and $\mathbf{n_a}$ are the measurement noise vectors; and $\mathbf{b_g}$ and $\mathbf{b_a}$ represent the gyroscope and accelerometer biases, respectively. These biases are typically *slowly* time-varying and are usually modeled by random walk processes [1]. However, since we are here interested in motion estimation over short time periods, we consider the IMU biases as being constant. We assume that (noisy) prior estimates for the biases are available: for example, they can be assumed to be close to zero, or estimates may be available from an earlier calibration. We express this prior information by modeling the biases as Gaussian random variables, $\mathbf{b_g} \sim \mathcal{N}(\hat{\mathbf{b}}_\mathbf{g}, \mathbf{Q_g})$ and $\mathbf{b_a} \sim \mathcal{N}(\hat{\mathbf{b}}_\mathbf{a}, \mathbf{Q_a})$.

III. Estimator Formulation

A. MAP Estimation

We begin by describing the optimal MAP estimator, which serves as the basis for the derivation of our convex formulation. The state vector we seek to estimate is given by:

$$\mathbf{x} = \begin{bmatrix} \mathbf{v}_{I_1}^T & \mathbf{x}_{I_2}^T & \cdots & \mathbf{x}_{I_N}^T & {}^{B}\mathbf{p}_{f_1}^T & \cdots & {}^{B}\mathbf{p}_{f_M}^T & {}^{B}\mathbf{g}^T & \mathbf{b}_\mathbf{g}^T & \mathbf{b}_\mathbf{a}^T \end{bmatrix}^T \quad (5)$$

[1]The preceding superscript for vectors (e.g., X in ${}^{X}\mathbf{a}$) denotes the frame of reference with respect to which quantities are expressed. ${}^{X}_{Y}\mathbf{R}$ is the rotation matrix rotating vectors from $\{Y\}$ to $\{X\}$, and ${}^{X}_{Y}\bar{\mathbf{q}}$ is the corresponding unit quaternion. ${}^{X}\mathbf{p}_Y$ is the origin of $\{Y\}$ with respect to $\{X\}$. $\lfloor \mathbf{c} \times \rfloor$ denotes the skew symmetric matrix corresponding to vector $\mathbf{c}$, $\mathbf{0}$ and $\mathbf{I}$ are the zero and identity matrices respectively, and $\hat{a}$ and $\tilde{a}$ represent the estimate and error of the estimate of a variable a, respectively.

where N is the number of images recorded in the time interval of interest, M is the number features observed, and $\mathbf{x}_{I_i}$ represents the IMU state at the time instant the i-th image was recorded, consisting of the IMU orientation, position, and velocity with respect to $\{B\}$:

$$\mathbf{x}_{I_i} = \begin{bmatrix} {}^{I_i}_{B}\bar{\mathbf{q}}^T & {}^{B}\mathbf{p}_{I_i}^T & {}^{B}\mathbf{v}_{I_i}^T \end{bmatrix}^T \quad (6)$$

Note that for the first IMU state only the velocity is included in $\mathbf{x}$, since due to the definition of the base frame the first IMU position is identically zero, and ${}^{I_1}_{B}\bar{\mathbf{q}} = [0 \;\; 0 \;\; 0 \;\; 1]^T$.

The MAP estimate for $\mathbf{x}$ can be obtained by solving the minimization problem:

$$\begin{array}{ll} \text{minimize} & c_{IMU}(\mathbf{x}) + c_{cam}(\mathbf{x}) + c_{prior}(\mathbf{x}) \\ \text{subject to} & \|{}^{B}\mathbf{g}\|_2 = g \end{array} \quad (7)$$

where g is the known norm of the gravity vector, and c_{IMU}, c_{cam}, and c_{prior} represent the cost functions corresponding to the IMU measurements (used as process-model information), the camera measurements, and the prior, respectively:

$$c_{IMU}(\mathbf{x}) = \sum_{i=1}^{N-1} \|\mathbf{x}_{I_{i+1}} - \mathbf{f}\left(\mathbf{x}_{I_i}, \mathbf{b_g}, \mathbf{b_a}, \mathbf{a}_m, \boldsymbol{\omega}_m\right)\|_{\mathbf{Q}_i} \quad (8)$$

$$c_{cam}(\mathbf{x}) = \sum_{\{i,j\} \in \mathcal{S}} \|\mathbf{z}_{ij} - \mathbf{h}\left(\mathbf{x}_{I_i}, \mathbf{f}_j\right)\|_{\sigma_{im}^2 \mathbf{I}_2} \quad (9)$$

$$c_{prior}(\mathbf{x}) = \|\mathbf{b_g} - \hat{\mathbf{b}}_\mathbf{g}\|_{\mathbf{Q}_g} + \|\mathbf{b_a} - \hat{\mathbf{b}}_\mathbf{a}\|_{\mathbf{Q}_a} \quad (10)$$

In the above equations, $\mathcal{S}$ is a set containing all the pairs of indices $\{i, j\}$ that describe the feature measurements, $\mathbf{f}(\cdot)$ represents the IMU propagation equation, which is implemented as described in [1], $\mathbf{Q}_i$ is the discrete-time process-noise covariance matrix, and we use the notation $\|\mathbf{y}\|_\mathbf{W} = \mathbf{y}^T\mathbf{W}^{-1}\mathbf{y}$.

The minimization problem (7) is a nonlinear least-squares problem, and can be solved via Gauss-Newton or Levenberg-Marquardt minimization [5], [6], [9]. However, due to the non-convex nature of the problem, for these minimization methods to converge (close) to the global minimum of the objective function, a good initial guess for $\mathbf{x}$ is required. In what follows, we describe how such an initial guess can be obtained by formulating and solving a convex minimization problem.

B. Convex formulation

One of the reasons making the minimization problem in (7) non-convex is the fact that the IMU orientation is included in the unknown $\mathbf{x}$. To address this, in our formulation we obtain an estimate for the orientation of all the IMU poses relative to $\{B\}$ (i.e., relative to the initial state) by integrating the gyroscope measurements [1], and subsequently treat it as known. Given these estimates for the relative orientation, the remaining quantities that need to be estimated are described by the vector:

$$\mathbf{x_r} = \begin{bmatrix} \mathbf{v}_{I_1}^T & \mathbf{x}_{I_2}^{\star T} & \cdots & \mathbf{x}_{I_N}^{\star T} & {}^{B}\mathbf{p}_{f_1}^T & \cdots & {}^{B}\mathbf{p}_{f_M}^T & {}^{B}\mathbf{g}^T & \mathbf{b}_\mathbf{a}^T \end{bmatrix}^T \quad (11)$$

where $\mathbf{x}_{I_i}^\star = [{}^{B}\mathbf{p}_{I_i}^T \;\; {}^{B}\mathbf{v}_{I_i}^T]^T$, for $i = 2, \ldots, N$. The estimate for $\mathbf{x_r}$ is computed via convex minimization, as described next.

When the relative orientation between different time instants is considered known, the IMU measurements can be used to

define *linear* constraints involving the IMU position and velocity, the gravity vector, and the accelerometer bias. Specifically, as shown in [1], we can write:

$$ {}^B\mathbf{p}_{I_{i+1}} = {}^B\mathbf{p}_{I_i} + {}^B\mathbf{v}_{I_i}\Delta t_i + {}^B\mathbf{g}\frac{\Delta t_i^2}{2} + {}^B_{I_i}\hat{\mathbf{R}}\left(\mathbf{y}_i - \mathbf{P}_i\,\mathbf{b_a}\right) + \mathbf{w}_{p_i} \tag{12} $$

$$ {}^B\mathbf{v}_{I_{i+1}} = {}^B\mathbf{v}_{I_i} + {}^B\mathbf{g}\Delta t_i + {}^B_{I_i}\hat{\mathbf{R}}\left(\mathbf{s}_i - \mathbf{V}_i\,\mathbf{b_a}\right) + \mathbf{w}_{v_i} \tag{13} $$

where $\Delta t_i = t_{i+1} - t_i$ represents the time interval between image i and $i+1$, $\mathbf{w}_{p_i}$ and $\mathbf{w}_{v_i}$ are error vectors whose joint covariance matrix, $\mathbf{Q}_{w_i}$, can be computed as a function of the IMU noise parameters, and

$$ \mathbf{y}_i = \int_{t_i}^{t_{i+1}}\int_{t_i}^{\tau} {}^{I_i}_{I_\eta}\hat{\mathbf{R}}\,\mathbf{a}_m(\eta)d\eta d\tau \quad \mathbf{P}_i = \int_{t_i}^{t_{i+1}}\int_{t_i}^{\tau} {}^{I_i}_{I_\tau}\hat{\mathbf{R}}d\eta d\tau $$

$$ \mathbf{s}_i = \int_{t_i}^{t_{i+1}} {}^{I_i}_{I_\tau}\hat{\mathbf{R}}\,\mathbf{a}_m(\tau)dt \qquad \mathbf{V}_i = \int_{t_i}^{t_{i+1}} {}^{I_i}_{I_\tau}\hat{\mathbf{R}}dt $$

Equations (12) and (13) can be used to define the IMU-related term in the objective function as a quadratic cost function:

$$ c'_{IMU}(\mathbf{x_r}) = \sum_{i=1}^{N-1} \|\mathbf{A}_i\mathbf{x_r} - \mathbf{c}_i\|_{\mathbf{Q}_{w_i}} \tag{14} $$

where $\mathbf{A}_i$ and $\mathbf{c}_i$ are constant matrices and vectors, respectively, computed from (12) and (13).

Turning our attention to the camera-related terms in the objective function, we see that by substituting (1) into (9), we can write the original cost function c_{cam} as:

$$ c_{cam}(\mathbf{x_r}) = \sum_{\{i,j\}\in\mathcal{S}} \left(\frac{(u_{ij}\,z_{ij} - x_{ij})^2 + (v_{ij}\,z_{ij} - y_{ij})^2}{\sigma_{im}^2 z_{ij}^2} \right) \tag{15} $$

It is important to point out that the coordinates x_{ij}, y_{ij}, and z_{ij} are *linear* functions of the IMU position ${}^B\mathbf{p}_{I_i}$ and feature position ${}^B\mathbf{p}_{f_j}$, as shown in (2). Therefore, the above cost function is a summation of quadratic-over-quadratic terms, which, in general, is not convex. To approximate it by a convex function, let us consider that an estimate of the feature depth in each image, $\hat{z}_{ij}$, is available (see Section IV). Then (15) can be approximated by:

$$ c'_{cam}(\mathbf{x_r}) = \sum_{\{i,j\}\in\mathcal{S}} \frac{1}{\sigma_{im}^2\hat{z}_{ij}} \left(\frac{(u_{ij}\,z_{ij} - x_{ij})^2 + (v_{ij}\,z_{ij} - y_{ij})^2}{z_{ij}} \right) \tag{16} $$

Since all feature depths z_{ij} are positive, each of the functions in the above summation is convex (it is the perspective of a quadratic function), and therefore $c'_{cam}(\mathbf{x_r})$ is convex.

To understand the effects of approximating the cost function in (15) by the one in (16), we note that each of the summands in (16) equals the corresponding summand in (15), weighted by $z_{ij}/\hat{z}_{ij}$. Therefore, using the approximate cost function in (16) is tantamount to introducing a weighting of each feature measurement's cost by $z_{ij}/\hat{z}_{ij}$. Minimizing this cost function would be equivalent to solving a weighted least-squares problem, where the weight of each term is not the "ideal" one required for MAP estimation, but one close to it.

We now define the following minimization problem for $\mathbf{x_r}$:

$$ \begin{aligned} \text{minimize} \quad & c'_{IMU}(\mathbf{x_r}) + c'_{cam}(\mathbf{x_r}) + \|\mathbf{b_a} - \hat{\mathbf{b}}_\mathbf{a}\|_{\mathbf{Q}_a} \\ \text{subject to} \quad & \|{}^B\mathbf{g}\|_2 \leq g, \quad z_{ij} \geq 0, \forall ij \end{aligned} \tag{17} $$

where the first two terms in the objective function represent the information provided by the IMU and the camera measurements, respectively, and the term $\|\mathbf{b_a} - \hat{\mathbf{b}}_\mathbf{a}\|_{\mathbf{Q}_a}$ represents the prior information about the accelerometer bias. Compared to the exact formulation in (7), the above formulation has an approximate cost function, and uses a relaxed version of the gravity-norm constraint. While these approximations mean that the solution of (7) will in general not be identical to the solution of (17), the latter problem is a *convex* one, since it involves a convex objective function and convex constraints. Therefore, the global minimum of (17) can be found using standard optimization tools, without requiring a prior estimate for the solution.

C. Initialization in the presence of outliers

One key challenge when using vision for state estimation is the presence of outlier measurements. If the problem formulation in (17) was used with feature measurements that included outliers (e.g., wrong correspondences or features on moving objects), the estimation result would be unreliable. In recursive estimation, where a prior distribution for the state is available, outliers can be identified and discarded using standard statistical tests such as Mahalanobis-distance gating [1]. Since we here perform initialization in the absence of a motion prior, these methods are not applicable.

To deal with possible outliers in the camera measurements, we replace the cost function shown in (16) by a "robust" one, which exhibits lower sensitivity to outliers. Specifically, we employ the bivariate Huber function [13], to write:

$$ c''_{cam}(\mathbf{x_r}) = \sum_{\{i,j\}\in\mathcal{S}} \frac{m(u_{ij}\,z_{ij} - x_{ij}, z_{ij}) + m(v_{ij}\,z_{ij} - y_{ij}, z_{ij})}{\sigma_{im}^2\hat{z}_{ij}} \tag{18} $$

where

$$ m(\alpha,\beta) = \begin{cases} \frac{\alpha^2}{\beta} & |\alpha| \leq \kappa\beta \\ 2\kappa|\alpha| - \kappa^2\beta & |\alpha| > \kappa\beta \end{cases} \tag{19} $$

Here κ is a threshold, chosen so that the function enters its linear branch for residuals larger than $3\sigma_{im}$. If the residuals are small, the bivariate Huber function yields a cost function identical to that in (16). However, for large measurement residuals (likely for outliers), the bivariate Huber results in a smaller penalty term added to the objective function. In this way, outliers do not have large effects on the solution. Note that the bivariate Huber is a convex function, and therefore by using $c''_{cam}(\mathbf{x_r})$ in (17), instead of $c'_{cam}(\mathbf{x_r})$, the problem remains convex.

IV. Feature Depth Estimation

The objective function being minimized requires estimates, $\hat{z}_{ij}$, for the depths of the features in the images. For simplicity, one may use a constant value (e.g., the expected average depth of scene features) for all terms $\hat{z}_{ij}$. However, improved accuracy can be achieved by using pre-processing step to obtain better depth estimates, as described next.

A. Depth Estimation in Image Pairs

We begin by employing pairs of images to compute (scaled) estimates for the feature depths. Specifically, let us consider a feature, $\mathbf{f}_j$, observed in images i and k. The coordinates of $\mathbf{f}_j$ in camera frames $\{C_i\}$ and $\{C_k\}$ are related by:

$$^{C_k}\mathbf{p}_{f_j} = {}^{C_k}_{C_i}\mathbf{R}\,{}^{C_i}\mathbf{p}_{f_j} + {}^{C_k}\mathbf{p}_{C_i} \tag{20}$$

Ignoring the noise, we can write the above equation as:

$$z_{kj}\begin{bmatrix} u_{kj} \\ v_{kj} \\ 1 \end{bmatrix} = z_{ij}\,{}^{C_k}_{C_i}\mathbf{R}\begin{bmatrix} u_{ij} \\ v_{ij} \\ 1 \end{bmatrix} + \begin{bmatrix} {}^{C_k}x_{C_i} \\ {}^{C_k}y_{C_i} \\ {}^{C_k}z_{C_i} \end{bmatrix} \tag{21}$$

Solving the third row of the above equation for z_{kj}, and substituting in the first two rows, we obtain:

$$z_{ij}\underbrace{\left(\bar{w}_{ij}\begin{bmatrix} u_{kj} \\ v_{kj} \end{bmatrix} - \begin{bmatrix} \bar{u}_{ij} \\ \bar{v}_{ij} \end{bmatrix}\right)}_{\mathbf{k}_{ik,j}} = \begin{bmatrix} {}^{C_k}x_{C_i} \\ {}^{C_k}y_{C_i} \end{bmatrix} - {}^{C_k}z_{C_i}\underbrace{\begin{bmatrix} u_{kj} \\ v_{kj} \end{bmatrix}}_{\mathbf{z}_{kj}} \tag{22}$$

where we have used the notation

$$\begin{bmatrix} \bar{u}_{ij} \\ \bar{v}_{ij} \\ \bar{w}_{ij} \end{bmatrix} = {}^{C_k}_{C_i}\mathbf{R}\begin{bmatrix} u_{ij} \\ v_{ij} \\ 1 \end{bmatrix}$$

Stacking the equations in (22) for all features (i.e., all j) observed in images i and k, we obtain:

$$\underbrace{\begin{bmatrix}\mathbf{A}_{ik}^{xy} & \mathbf{a}_{ik}^{z} & \mathbf{A}_{ik}^{f}\end{bmatrix}}_{\mathbf{A}_{ik}}\underbrace{\begin{bmatrix} {}^{C_k}x_{C_i} \\ {}^{C_k}y_{C_i} \\ {}^{C_k}z_{C_i} \\ \boldsymbol{\rho}_i \end{bmatrix}}_{\mathbf{x}_{ik}} = \mathbf{0}, \quad \text{with } \boldsymbol{\rho}_i = \begin{bmatrix} z_{i1} \\ \vdots \\ z_{iM} \end{bmatrix} \tag{23}$$

where

$$\mathbf{A}_{ik}^{xy} = \begin{bmatrix} \mathbf{I}_2 \\ \vdots \\ \mathbf{I}_2 \end{bmatrix}, \quad \mathbf{a}_{ik}^{z} = \begin{bmatrix} -\mathbf{z}_{k1} \\ \vdots \\ -\mathbf{z}_{kM} \end{bmatrix}, \quad \mathbf{A}_{ik}^{f} = -\mathbf{Diag}(\mathbf{k}_{ik,j}) \tag{24}$$

If no noise was present, we could compute the camera motion and the features' depths (up to scale) by finding the vector that satisfies $\mathbf{A}_{ik}\mathbf{x}_{ik} = \mathbf{0}$ (this is the right singular vector of $\mathbf{A}_{ik}$ corresponding to the smallest singular value). In the presence of noise and outliers, however, this solution is not robust. To address this issue, we formulate a minimization problem based on the Huber cost function.

First, since the scale of $\mathbf{x}_{ik}$ can not be determined using monocular-camera measurements alone, we must enforce a scale factor on the solution. To this end, we set one of ${}^{C_k}x_{C_i}$, ${}^{C_k}y_{C_i}$, or ${}^{C_k}z_{C_i}$ to be equal to one. Specifically, if the camera motion has a significant component along the optical axis, we set ${}^{C_k}z_{C_i} = 1$, otherwise we set either ${}^{C_k}x_{C_i}$ or ${}^{C_k}y_{C_i}$ equal to one, based on the dominant motion direction of the feature points on the image. To detect whether the motion along the optical axis is significant, we note that when ${}^{C_k}z_{C_i}$ is zero, then all vectors $\mathbf{k}_{ik,j}$, $j = 1, \ldots, M$, are collinear (see (22)). Therefore, we can test whether ${}^{C_k}z_{C_i}$ is significant by testing for the collinearity of these vectors.

Assuming for the sake of presentation that we have used ${}^{C_k}z_{C_i} = 1$, we subsequently compute $\mathbf{x}_{ik}$ by solving the following least-soft-square minimization problem [16]:

$$\min_{\mathbf{x}_{ik}^\star,\boldsymbol{\zeta}} \frac{1}{2}\left\|\underbrace{\begin{bmatrix}\mathbf{A}_{ik}^{xy} & \mathbf{A}_{ik}^{f}\end{bmatrix}}_{\mathbf{A}_{ik}^\star}\underbrace{\begin{bmatrix} {}^{C_k}x_{C_i} \\ {}^{C_k}y_{C_i} \\ \boldsymbol{\rho}_i \end{bmatrix}}_{\mathbf{x}_{ik}^\star} + \mathbf{a}_{ik}^{z} + \boldsymbol{\zeta}\right\|_2^2 + \lambda\|\boldsymbol{\zeta}\|_1 \tag{25}$$

In [16] it is shown that solving the above problem, which includes the "regulation term" $\lambda\|\boldsymbol{\zeta}\|_1$, is equivalent to minimizing the sum of the Huber functions of each element in $\mathbf{A}_{ik}^\star\mathbf{x}_{ik}^\star + \mathbf{a}_{ik}^z$, with λ being the threshold of the Huber function. The minimization problem in (25) is convex, and can be efficiently solved by an iterative algorithm. Due to the sparsity of the involved matrices, the computational cost per iteration is only linear in the size of $\mathbf{x}_{ik}^\star$.

B. Computing consistent depth estimates for all images

By completing the process described above for each pair of consecutive images, we have an estimate of the scaled depths of the features in each image, $\boldsymbol{\rho}_i$, $i = 1, \ldots, N$. The difficulty lies in the fact that a different scale factor, s_i, exists in each image. In order to obtain consistent scale among all images, we solve a set of ℓ_1-minimization problems.

Specifically, the feature depths in image i are related to the elements of the vector $\boldsymbol{\rho}_i$ by: $z_{ij} = s_i\rho_{ij}$, $j = 1, \ldots, M$. Using this relationship in (21), we can write $s_k\rho_{kj} = s_i\rho_{ij}\bar{w}_{ij} + {}^{C_k}z_{C_i}$, $j = 1, \ldots, M$. Eliminating ${}^{C_k}z_{C_i}$ from these M equations, we obtain:

$$s_k\begin{bmatrix} \rho_{k2} - \rho_{k1} \\ \vdots \\ \rho_{kM} - \rho_{k1} \end{bmatrix} - s_i\begin{bmatrix} \rho_{i2}\bar{w}_{i2} - \rho_{i1}\bar{w}_{i1} \\ \vdots \\ \rho_{iM}\bar{w}_{iM} - \rho_{i1}\bar{w}_{i1} \end{bmatrix} = \mathbf{0} \tag{26}$$

Based on the above equation we formulate a series of minimization problems in order to obtain the scale factors for all images. For the first image, we select the factor s_1 such that the median estimated depth of the features matches the expected average depth of the scene. Subsequently, we recursively compute the scale factors for all remaining images, by solving, for $i = 1, \ldots, N-1$, the following minimization problem:

$$\min_{s_{i+1}}\left\|\frac{s_{i+1}}{s_i}\begin{bmatrix} \rho_{(i+1)2} - \rho_{(i+1)1} \\ \vdots \\ \rho_{(i+1)M} - \rho_{(i+1)1} \end{bmatrix} - \begin{bmatrix} \rho_{i2}\bar{w}_{i2} - \rho_{i1}\bar{w}_{i1} \\ \vdots \\ \rho_{iM}\bar{w}_{iM} - \rho_{i1}\bar{w}_{i1} \end{bmatrix}\right\|_1$$

The solution to this problem can be computed in closed form, as shown in [17].

We stress that the overall computational cost of the process for obtaining feature-depth estimates is *linear* in the number of image features. As shown in the simulation results presented in the following section, this process leads to an improvement in the accuracy of the convex estimator, at a low additional cost. In certain cases, however, depth estimation is intrinsically unreliable, e.g., when the features are all at large distances compared to the baseline of the camera motion. These cases can be explicitly detected using the results of (25). To avoid using inaccurate estimates of depth, the depth estimation is disabled in these cases, and the average scene depth is used as the initial guess for all features' depths in (18).

TABLE I: RMS errors in the simulations

	CE dep. est.	CE	Martinelli 2014	MAP
Vel. (m/s)	0.055	0.072	0.189	0.032
Ori. (deg)	0.411	0.430	0.511	0.396

V. Experiments

A. Simulations

1) Motion estimation accuracy: We first present the results of Monte-Carlo simulations showing the accuracy of the proposed method compared to alternatives. We simulate a scenario where a camera-IMU system records images and inertial measurements over a 3.2-sec long time interval. Eight images are recorded, in each of which an average of 50 features are observed, with feature depths uniformly distributed between 2 and 12 meters. Inertial measurements are available at 100 Hz. We conducted 100 Monte-Carlo trials, where in each trial the camera trajectory was randomly generated, while the feature positions, IMU biases, and measurement noises were independently sampled from their corresponding pdfs. The noise parameters were selected to be identical to those of the experimental setup described in Section V-B.

Four different algorithms are compared: i) the proposed convex-minimization estimator (CE) using the depth-estimation approach described in Section IV, ii) the proposed convex-minimization estimator where all depth estimates $\hat{z}_{ij}$ are set to the average scene depth, iii) the linear estimator of [10], and iv) the optimal MAP estimator, implemented using the Levenberg-Marquardt algorithm. To evaluate the performance of the different approaches, we compute the root mean squared (RMS) errors over all the trials, for the initial velocity ${}^B\mathbf{v}_{I_1}$ and for the initial orientation (the latter is computed as the angle between the true and estimated gravity vectors ${}^B\mathbf{g}$). Since in the method of [10] outliers are not modeled, we do not generate any outliers in the simulations, to ensure a fair comparison. Outlier features do exist in the real-world experiment described in Section V-B.

For visualization purposes, Fig. 1 shows the initial-velocity estimation errors in ten representative simulation trials, while the error statistics for both the initial velocity and orientation over all 100 Monte-Carlo trials are shown in Table I. As expected, the MAP estimator reports the most accurate estimates, since it minimizes the "exact" cost function. On the other hand, both of the proposed convex formulations outperform the method of [10] by a significant margin. This is due to the fact that the approach of [10] does not model the measurement noise properly, while the proposed formulations are derived from the MAP estimator, and better characterize the relative accuracy of the measurements. Among the two convex-optimization based methods, the one in which a pre-processing step is used to estimate the feature depths achieves higher accuracy. However, the difference between the two approaches is not dramatic. This suggests that the main factor limiting the performance of the convex-optimization approach is the separate estimation of the relative rotation, rather than the approximation of the camera-cost function.

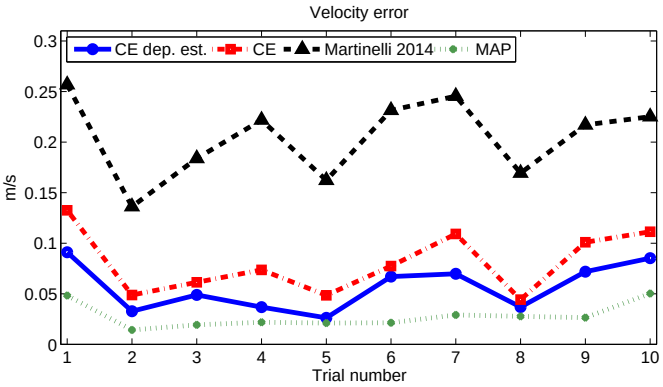

Fig. 1: Estimation errors of the initial velocity in ten representative Monte-Carlo simulation trials.

2) Initializing the iterative MAP estimator: We next compare the performance of the proposed convex formulation against that of [10], in terms of how reliably it can initialize the iterative MAP estimator. To obtain challenging scenarios we generated only 20 features per image on average, and reduced the duration of the data to 2 sec. We generated features with depths uniformly distributed between d and $2d$ in each trial. To examine the effect of scene depth, d was varied between 3 m and 11 m, with 30 Monte-Carlo trials run for each setting.

In each trial, the MAP estimator was initialized by the estimates of (i) the proposed method with depth estimation and (ii) the linear formulation of [10], and run to convergence. We would like to examine whether the iterations converge to the global minimum in each trial. However, a provably globally optimal solution is difficult to obtain. Therefore, we instead ran the MAP estimator initialized by the *ground-truth* in each trial, and treated the resulting solution as the "ideal" one. To evaluate the results of the two initialization methods in (i) and (ii) above, we compare the cost function after convergence to the cost of the "ideal" solution, and if it is more than 1% larger, the trial is considered unsuccessful.

Fig. 2 shows the percentage of unsuccessful trials for the two methods compared. It becomes clear that when the MAP estimator is initialized using the estimates of [10], a significant proportion of trials are unsuccessful, i.e., the MAP estimator converges to a local minimum of the cost function. This percentage increases as features are placed farther away from the camera, since this makes the estimation problem harder (the baseline is smaller compared to the scene depth). On the other hand, the proposed method results in significantly improved performance (only four unsuccessful trials out of 270). Interestingly, the performance of the algorithm does not appear to be significantly affected by increasing scene depth. This is important, as it permits operation in a wider range of environmental conditions.

B. Real-world Experiment

In this section, we provide results from a real-world experiment, conducted with an IMU-camera platform in an indoor environment. This platform comprised an Xsens MTi-G IMU and a Bumblebee2 stereo camera (only one camera was used). During the experiment, the platform moved for about two minutes at an average velocity of $0.4\,\mathrm{m/s}$. The measurements

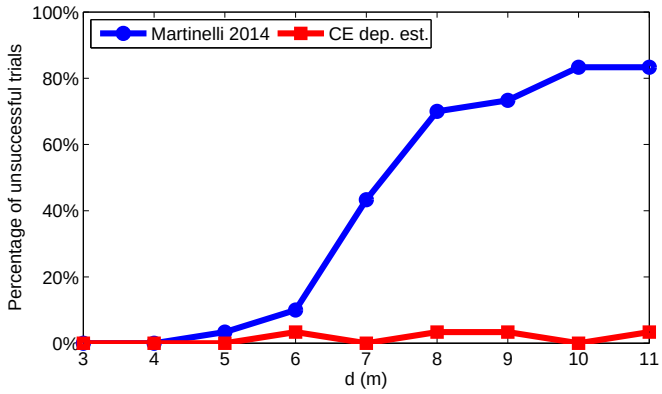

Fig. 2: Percentage of unsuccessful MAP solutions in the Monte-Carlo simulations.

TABLE II: RMS errors in the real-world experiment

	CE dep. est.	Martinelli 2014	MAP
Vel. (m/s)	0.096	0.208	0.058
Ori. (deg.)	0.315	0.470	0.250

of the IMU and camera were recorded at 100 Hz and 2.5 Hz, respectively. To process images, the Shi-Tomasi algorithm [18] was used to extract features, and feature matching was done by normalized cross-correlation. To compare the performance of the methods, we break the dataset into 30 non-overlapping 3-sec windows, and perform estimation separately in each.

We here report the results of three methods: i) the proposed method, with depth estimation used only when it is deemed reliable (see Section IV-B), ii) the method of [10], and iii) the MAP algorithm. Since [10] does not tolerate outliers, features with large reprojection errors (larger than $3\sigma_{im}$ when used in our method) were not used for [10]. To compare the results of these approaches, we computed an approximate ground truth for the entire dataset via the method of [15]. While this ground truth will likely contain some errors, these are significantly smaller than the errors of each of the estimators used here, as [15] uses the entire history of measurements to obtain estimates at each point in time.

Table II shows the RMS errors of the three algorithms, averaged over the 30 time windows. Similarly to the results in the simulations, Table II shows that the proposed convex approach outperforms the method of [10] by a significant margin, even though the latter uses "outlier-free" data. The performance of the proposed convex optimization is closer to that of the optimal MAP estimator (which is in fact initialized by the result of the convex minimization), than to that of [10].

VI. Conclusion

In this paper, we propose a novel algorithm for estimating motion using visual and inertial measurements. The method does not require a prior estimate of the motion, and can operate in the presence of outliers. Due to these properties, the proposed approach can be employed for in-motion initialization of an EKF estimator, for providing an initial guess for linearization-based estimators, or for re-starting an estimator after failure. Our simulations and real-world testing demonstrate that the proposed method outperforms that of [10], which is the best previous solution proposed for this problem. In our ongoing work, we are investigating methods for attaining better performance (i.e., smaller approximations) within the convex formulation proposed here.

Acknowledgments

This work was supported by the National Science Foundation (grant no. IIS-1117957, IIS-1253314 and IIS-1316934).

References

[1] M. Li and A. I. Mourikis, "High-precision, consistent EKF-based visual-inertial odometry," *International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, May 2013.

[2] E. Jones and S. Soatto, "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach," *International Journal of Robotics Research*, vol. 30, no. 4, pp. 407–430, Apr. 2011.

[3] J. Kelly and G. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *International Journal of Robotics Research*, vol. 30, no. 1, pp. 56–79, Jan. 2011.

[4] S. Weiss, M. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environment," in *Proceedings of the IEEE International Conference on Robotics and Automation*, St Paul, MN, May 2012, pp. 957–964.

[5] H.-P. Chiu, S. Williams, F. Dellaert, S. Samarasekera, and R. Kumar, "Robust vision-aided navigation using sliding-window factor graphs," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Kalsruhe, Germany, May 2013, pp. 46–53.

[6] J. Michot, A. Bartoli, and F. Gaspard, "Bi-objective bundle adjustment with application to multi-sensor SLAM," in *International Symposium 3D Data Processing, Visualization and Transmission*, Paris, France, May 2010.

[7] T. Dong-Si and A. I. Mourikis, "Motion tracking with fixed-lag smoothing: Algorithm and consistency analysis," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Shanghai, China, May 2011, pp. 5655 – 5662.

[8] L. Kneip, S. Weiss, and R. Siegwart, "Deterministic initialization of metric state estimation filters for loosely-coupled monocular vision-inertial systems," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, CA, Sept 2011, pp. 2235–2241.

[9] T. Dong-Si and A. I. Mourikis, "Estimator initialization in vision-aided inertial navigation with unknown camera-IMU calibration," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura, Portugal, Oct. 2012, pp. 1064–1071.

[10] A. Martinelli, "Closed-form solution of visual-inertial structure from motion," *International Journal of Computer Vision*, vol. 106, no. 2, pp. 138–152, Jan. 2014.

[11] T. Lupton and S. Sukkarieh, "Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 61–76, Feb. 2012.

[12] A. Martinelli, "Vision and IMU data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 44 –60, Feb. 2012.

[13] C. Zach and M. Pollefeys, "Practical methods for convex multi-view reconstruction," in *Proceedings of the European Conference on Computer Vision*, Heraklion, Crete, Greece, Sept. 2010, pp. 354–367.

[14] Q. Ke and T. Kanade, "Robust l1 norm factorization in the presence of outliers and missing data by alternative convex programming," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 1, 2005, pp. 739–746.

[15] M. Li, H. Yu, X. Zheng, and A. I. Mourikis, "High-fidelity sensor modeling and self-calibration in vision-aided inertial navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Hong Kong, June 2014.

[16] D. Wang, H. Lu, and M.-H. Yang, "Least soft-threshold squares tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2371–2378.

[17] Q. Ke and T. Kanade, "Robust subspace computation using l1 norm," Carnegie Mellon University, Tech. Rep., 2003.

[18] J. Shi and C. Tomasi, "Good features to track," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Seattle, WA, June 1994, pp. 593–600.