A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation

Anastasios I. Mourikis and Stergios I. Roumeliotis*

September 28, 2006

Abstract

In this paper, we present an Extended Kalman Filter (EKF)-based algorithm for real-time vision-aided inertial navigation. The primary contribution of this work is the derivation of a measurement model that is able to express the geometric constraints that arise when a static feature is observed from multiple camera poses. This measurement model does not require including the 3D feature position in the state vector of the EKF and is optimal, up to linearization errors. The vision-aided inertial navigation algorithm we propose has computational complexity only *linear* in the number of features, and is capable of high-precision pose estimation in large-scale real-world environments. The performance of the algorithm is demonstrated in extensive experimental results, involving a camera/IMU system localizing within an urban area.

1 Introduction

In the past few years, the topic of *vision-aided inertial navigation* has received considerable attention in the research community. Recent advances in the manufacturing of MEMS-based inertial sensors have made it possible to build small, inexpensive, and very accurate Inertial Measurement Units (IMUs), suitable for pose estimation in small-scale systems such as mobile robots and unmanned aerial vehicles. These systems often operate in urban environments where GPS signals are unreliable (the "urban canyon"), as well as indoors, in space, and in several other environments where global position measurements are unavailable. The low cost, weight, and power consumption of cameras make them ideal alternatives for aiding inertial navigation, in cases where GPS measurements cannot be relied upon.

An important advantage of visual sensing is that images are high-dimensional measurements, with rich information content. Feature extraction methods can typically detect and track hundreds of features in images, which, if properly used, can result is excellent localization results. However, the high volume of data also poses a significant challenge for estimation algorithm design. When real-time localization performance is required, one is faced with a fundamental trade-off between the computational complexity of an algorithm and the resulting estimation accuracy.

In this paper we present an algorithm that is able to *optimally* utilize the localization information provided by multiple measurements of visual features. Our approach is motivated by the observation that, when a static feature is viewed from several camera poses, it is possible to define *geometric constraints* involving all these poses. The primary contribution of our work is a measurement model that expresses these constraints *without* including the 3D feature position in the filter state vector, resulting in computational complexity only *linear* in the number of features. After a brief discussion of related work in the next section, the details of the proposed estimator are presented in Section 3. In Section 4 we describe

*The authors are with the Dept. of Computer Science & Engineering, University of Minnesota, Minneapolis, MN 55455. Emails: {mourikis|stergios}@cs.umn.edu

the results of a large-scale experiment in an uncontrolled urban environment, which demonstrate that the proposed estimator enables *accurate, real-time* pose estimation. Finally, in Section 5 the conclusions of this work are drawn.

2 Related Work

One family of algorithms for fusing inertial measurements with visual feature observations follows the Simultaneous Localization and Mapping (SLAM) paradigm. In these methods, the current IMU pose, as well as the 3D positions of all visual landmarks are jointly estimated [1–4]. These approaches share the same basic principles with SLAM-based methods for camera-only localization (e.g., [5, 6], and references therein), with the difference that IMU measurements, instead of a statistical motion model, are used for state propagation. The fundamental advantage of SLAM-based algorithms is that they account for the correlations that exist between the pose of the camera and the 3D positions of the observed features. On the other hand, the main limitation of SLAM is its high computational complexity; properly treating these correlations is computationally costly, and thus performing vision-based SLAM in environments with thousands of features remains a challenging problem.

Several algorithms exist that, contrary to SLAM, estimate the pose of the camera *only* (i.e., do not jointly estimate the feature positions), with the aim of achieving real-time operation. The most computationally efficient of these methods utilize the feature measurements to derive constraints between pairs of images. For example in [7], an image-based motion estimation algorithm is applied to consecutive pairs of images, to obtain displacement estimates that are subsequently fused with inertial measurements. Similarly, in [8,9] constraints between current and previous image are defined using the epipolar geometry, and combined with IMU measurements in an Extended Kalman Filter (EKF). In [10, 11] the epipolar geometry is employed in conjunction with a statistical motion model, while in [12] epipolar constraints are fused with the dynamical model of an airplane. The use of feature measurements for imposing constraints between *pairs* of images is similar in philosophy to the method proposed in this paper. However, one fundamental difference is that our algorithm can express constraints between *multiple* camera poses, and can thus attain higher estimation accuracy, in cases where the same feature is visible in more than two images.

Pairwise constraints are also employed in algorithms that maintain a state vector comprised of multiple camera poses. In [13], an augmented-state Kalman filter is implemented, in which a sliding window of robot poses is maintained in the filter state. On the other hand, in [14], *all* camera poses are simultaneously estimated. In both of these algorithms, *pairwise* relative-pose measurements are derived from the images, and used for state updates. The drawback of this approach is that when a feature is seen in multiple images, the additional constraints between the multiple poses are discarded, thus resulting in loss of information. Furthermore, when the same image measurements are processed for computing several displacement estimates, these are not statistically independent, as shown in [15].

One algorithm that, similarly to the method proposed in this paper, directly uses the landmark measurements for imposing constraints between multiple camera poses is presented in [16]. This is a visual odometry algorithm that temporarily initializes landmarks, uses them for imposing constraints on windows of consecutive camera poses, and then discards them. This method, however, does not incorporate inertial measurements. Moreover, the correlations between the landmark estimates and the camera trajectory are not properly accounted for, and as a result, the algorithm does not provide any measure of the covariance of the state estimates.

A window of camera poses is also maintained in the Variable State Dimension Filter (VSDF) [17]. The VSDF is a hybrid batch/recursive method, that (i) uses *delayed linearization* to increase robustness against linearization inaccuracies, and (ii) exploits the sparsity of the information matrix, that naturally arises when *no dynamic motion model* is used. However, in cases where a dynamic motion model is

Algorithm 1 Multi-State Constraint Filter

Propagation: For each IMU measurement received, propagate the filter state and covariance (cf. Section 3.2).

Image registration: Every time a new image is recorded,

- augment the state and covariance matrix with a copy of the current camera pose estimate (cf. Section 3.3).
- image processing module begins operation.

Update: When the feature measurements of a given image become available, perform an EKF update (cf. Sections 3.4 and 3.5).

available (such as in vision-aided inertial navigation) the computational complexity of the VSDF is at best *quadratic* in the number of features [18].

In contrast to the VSDF, the multi-state constraint filter that we propose in this paper is able to exploit the benefits of delayed linearization while having complexity only *linear* in the number of features. By directly expressing the geometric constraints between multiple camera poses it avoids the computational burden and loss of information associated with pairwise displacement estimation. Moreover, in contrast to SLAM-type approaches, it does not require the inclusion of the 3D feature positions in the filter state vector, but still attains *optimal* pose estimation. As a result of these properties, the described algorithm is very efficient, and as shown in Section 4, is capable of high-precision vision-aided inertial navigation in real time.

3 Estimator Description

The goal of the proposed EKF-based estimator is to track the 3D pose of the IMU-affixed frame $\{I\}$ with respect to a *global frame* of reference $\{G\}$. In order to simplify the treatment of the effects of the earth's rotation on the IMU measurements (cf. Eqs. (7)-(8)), the global frame is chosen as an Earth-Centered, Earth-Fixed (ECEF) frame in this paper. An overview of the algorithm is given in Algorithm 1. The IMU measurements are processed immediately as they become available, for propagating the EKF state and covariance (cf. Section 3.2). On the other hand, each time an image is recorded, the current camera pose estimate is appended to the state vector (cf. Section 3.3). State augmentation is necessary for processing the feature measurements, since during EKF updates the measurements of each tracked feature are employed for imposing constraints between all camera poses from which the feature was seen. Therefore, at any time instant the EKF state vector comprises (i) the evolving IMU state, $\mathbf{X}_{\text{IMU}}$, and (ii) a history of up to $N_{\max}$ past poses of the camera. In the following, we describe the various components of the algorithm in detail.

3.1 Structure of the EKF state vector

The evolving IMU state is described by the vector:

$$\mathbf{X}_{\text{IMU}} = \begin{bmatrix} {}^I_G\bar{q}^T & \mathbf{b}_g{}^T & {}^G\mathbf{v}_I{}^T & \mathbf{b}_a{}^T & {}^G\mathbf{p}_I^T \end{bmatrix}^T \tag{1}$$

where ${}^I_G\bar{q}$ is the unit quaternion [19] describing the rotation from frame $\{G\}$ to frame $\{I\}$, ${}^G\mathbf{p}_I$ and ${}^G\mathbf{v}_I$ are the IMU position and velocity with respect to $\{G\}$, and finally $\mathbf{b}_g$ and $\mathbf{b}_a$ are 3×1 vectors that describe the biases affecting the gyroscope and accelerometer measurements, respectively. The IMU biases are modeled as random walk processes, driven by the white Gaussian noise vectors $\mathbf{n}_{wg}$ and $\mathbf{n}_{wa}$, respectively. Following Eq. (1), the IMU error-state is defined as:

$$\widetilde{\mathbf{X}}_{\text{IMU}} = \begin{bmatrix} \boldsymbol{\delta\theta}_I^T & \widetilde{\mathbf{b}}_g^T & {}^G\widetilde{\mathbf{v}}_I^T & \widetilde{\mathbf{b}}_a^T & {}^G\widetilde{\mathbf{p}}_I^T \end{bmatrix}^T \tag{2}$$

For the position, velocity, and biases, the standard additive error definition is used (i.e., the error in the estimate $\hat{x}$ of a quantity x is defined as $\widetilde{x} = x - \hat{x}$). However, for the quaternion a different error definition is employed. In particular, if $\hat{\bar{q}}$ is the estimated value of the quaternion $\bar{q}$, then the orientation error is described by the *error quaternion* $\delta\bar{q}$, which is defined by the relation $\bar{q} = \delta\bar{q} \otimes \hat{\bar{q}}$. In this expression, the symbol $\otimes$ denotes quaternion multiplication. The error quaternion is

$$\delta\bar{q} \quad \simeq \quad \begin{bmatrix} \frac{1}{2}\boldsymbol{\delta\theta}^T & 1 \end{bmatrix}^T \tag{3}$$

Intuitively, the quaternion $\delta\bar{q}$ describes the (small) rotation that causes the true and estimated attitude to coincide. Since attitude corresponds to 3 degrees of freedom, using $\boldsymbol{\delta\theta}$ to describe the attitude errors is a minimal representation.

Assuming that N camera poses are included in the EKF state vector at time-step k, this vector has the following form:

$$\hat{\mathbf{X}}_k = \begin{bmatrix} \hat{\mathbf{X}}_{\text{IMU}_k}^T & {}^{C_1}_{G}\hat{\bar{q}}^T & {}^G\hat{\mathbf{p}}_{C_1}^T & \dots & {}^{C_N}_{G}\hat{\bar{q}}^T & {}^G\hat{\mathbf{p}}_{C_N}^T \end{bmatrix}^T \tag{4}$$

where ${}^{C_i}_{G}\hat{\bar{q}}$ and ${}^G\hat{\mathbf{p}}_{C_i}$, $i = 1 \dots N$ are the estimates of the camera attitude and position, respectively. The EKF error-state vector is defined accordingly:

$$\widetilde{\mathbf{X}}_k = \begin{bmatrix} \widetilde{\mathbf{X}}_{\text{IMU}_k}^T & \boldsymbol{\delta\theta}_{C_1}^T & {}^G\widetilde{\mathbf{p}}_{C_1}^T & \dots & \boldsymbol{\delta\theta}_{C_N}^T & {}^G\widetilde{\mathbf{p}}_{C_N}^T \end{bmatrix}^T \tag{5}$$

3.2 Propagation

The filter propagation equations are derived by discretization of the continuous-time IMU system model, as described in the following:

3.2.1 Continuous-time system modeling

The time evolution of the IMU state is described by [20]:

$$\begin{gathered} {}^I_G\dot{\bar{q}}(t) = \tfrac{1}{2}\boldsymbol{\Omega}\big(\boldsymbol{\omega}(t)\big){}^I_G\bar{q}(t), \quad \dot{\mathbf{b}}_g(t) = \mathbf{n}_{wg}(t) \\ {}^G\dot{\mathbf{v}}_I(t) = {}^G\mathbf{a}(t), \quad \dot{\mathbf{b}}_a(t) = \mathbf{n}_{wa}(t), \quad {}^G\dot{\mathbf{p}}_I(t) = {}^G\mathbf{v}_I(t) \end{gathered} \tag{6}$$

In these expressions ${}^G\mathbf{a}$ is the body acceleration in the global frame, $\boldsymbol{\omega} = [\omega_x \;\; \omega_y \;\; \omega_z]^T$ is the rotational velocity expressed in the IMU frame, and

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} -\lfloor\boldsymbol{\omega}\times\rfloor & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{bmatrix}, \quad \lfloor\boldsymbol{\omega}\times\rfloor = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

The gyroscope and accelerometer measurements, $\boldsymbol{\omega}_m$ and $\mathbf{a}_m$ respectively, are given by [20]:

$$\boldsymbol{\omega}_m = \boldsymbol{\omega} + \mathbf{C}({}^I_G\bar{q})\boldsymbol{\omega}_G + \mathbf{b}_g + \mathbf{n}_g \tag{7}$$

$$\mathbf{a}_m = \mathbf{C}({}^I_G\bar{q})({}^G\mathbf{a} - {}^G\mathbf{g} + 2\lfloor\boldsymbol{\omega}_G\times\rfloor\,{}^G\mathbf{v}_I + \lfloor\boldsymbol{\omega}_G\times\rfloor^2\,{}^G\mathbf{p}_I) + \mathbf{b}_a + \mathbf{n}_a \tag{8}$$

where $\mathbf{C}(\cdot)$ denotes a rotational matrix, and $\mathbf{n}_g$ and $\mathbf{n}_a$ are zero-mean, white Gaussian noise processes modeling the measurement noise. It is important to note that the IMU measurements incorporate the effects of the planet's rotation, $\boldsymbol{\omega}_G$. Moreover, the accelerometer measurements include the gravitational acceleration, ${}^G\mathbf{g}$, expressed in the local frame.

Applying the expectation operator in the state propagation equations (Eq. (6)) we obtain the equations for propagating the *estimates* of the evolving IMU state:

$$
\begin{gathered}
{}^I_G\dot{\hat{\bar{q}}} = \tfrac{1}{2}\boldsymbol{\Omega}(\hat{\boldsymbol{\omega}})\,{}^I_G\hat{\bar{q}}, \quad \dot{\hat{\mathbf{b}}}_g = \mathbf{0}_{3\times 1}, \\
{}^G\dot{\hat{\mathbf{v}}}_I = \mathbf{C}_{\hat{q}}^T\hat{\mathbf{a}} - 2\lfloor\boldsymbol{\omega}_G\times\rfloor\,{}^G\hat{\mathbf{v}}_I - \lfloor\boldsymbol{\omega}_G\times\rfloor^2\,{}^G\hat{\mathbf{p}}_I + {}^G\mathbf{g} \qquad (9) \\
\dot{\hat{\mathbf{b}}}_a = \mathbf{0}_{3\times 1}, \quad {}^G\dot{\hat{\mathbf{p}}}_I = {}^G\hat{\mathbf{v}}_I
\end{gathered}
$$

where for brevity we have denoted $\mathbf{C}_{\hat{q}} = \mathbf{C}({}^I_G\hat{\bar{q}})$, $\hat{\mathbf{a}} = \mathbf{a}_m - \hat{\mathbf{b}}_a$ and $\hat{\boldsymbol{\omega}} = \boldsymbol{\omega}_m - \hat{\mathbf{b}}_g - \mathbf{C}_{\hat{q}}\boldsymbol{\omega}_G$. The linearized continuous-time model for the IMU error-state is:

$$
\dot{\widetilde{\mathbf{X}}}_{\text{IMU}} = \mathbf{F}\widetilde{\mathbf{X}}_{\text{IMU}} + \mathbf{G}\mathbf{n}_{\text{IMU}} \tag{10}
$$

where $\mathbf{n}_{\text{IMU}} = \begin{bmatrix}\mathbf{n}_g^T & \mathbf{n}_{wg}^T & \mathbf{n}_a^T & \mathbf{n}_{wa}^T\end{bmatrix}^T$ is the system noise. The covariance matrix of $\mathbf{n}_{\text{IMU}}$, $\mathbf{Q}_{\text{IMU}}$, depends on the IMU noise characteristics and is computed off-line during sensor calibration. Finally, the matrices $\mathbf{F}$ and $\mathbf{G}$ that appear in Eq. (10) are given by:

$$
\mathbf{F} = \begin{bmatrix}
-\lfloor\hat{\boldsymbol{\omega}}\times\rfloor & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\
-\mathbf{C}_{\hat{q}}^T\lfloor\hat{\mathbf{a}}\times\rfloor & \mathbf{0}_{3\times3} & -2\lfloor\boldsymbol{\omega}_G\times\rfloor & -\mathbf{C}_{\hat{q}}^T & -\lfloor\boldsymbol{\omega}_G\times\rfloor^2 \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3}
\end{bmatrix}
$$

where $\mathbf{I}_3$ is the 3×3 identity matrix, and

$$
\mathbf{G} = \begin{bmatrix}
-\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\
\mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{C}_{\hat{q}}^T & \mathbf{0}_{3\times3} \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3}
\end{bmatrix}
$$

3.2.2 Discrete-time implementation

The IMU samples the signals $\boldsymbol{\omega}_m$ and $\mathbf{a}_m$ with a period T, and these measurements are used for state propagation in the EKF. Every time a new IMU measurement is received, the IMU state estimate is propagated using 5th order Runge-Kutta numerical integration of Eqs. (9). Moreover, the EKF covariance matrix has to be propagated. For this purpose, we introduce the following partitioning for the covariance:

$$
\mathbf{P}_{k|k} = \begin{bmatrix}
\mathbf{P}_{II_{k|k}} & \mathbf{P}_{IC_{k|k}} \\
\mathbf{P}_{IC_{k|k}}^T & \mathbf{P}_{CC_{k|k}}
\end{bmatrix} \tag{11}
$$

where $\mathbf{P}_{II_{k|k}}$ is the 15×15 covariance matrix of the evolving IMU state, $\mathbf{P}_{CC_{k|k}}$ is the $6N\times 6N$ covariance matrix of the camera pose estimates, and $\mathbf{P}_{IC_{k|k}}$ is the correlation between the errors in the IMU state and the camera pose estimates. With this notation, the covariance matrix of the propagated state is given by:

$$
\mathbf{P}_{k+1|k} = \begin{bmatrix}
\mathbf{P}_{II_{k+1|k}} & \boldsymbol{\Phi}(t_k+T,t_k)\mathbf{P}_{IC_{k|k}} \\
\mathbf{P}_{IC_{k|k}}^T\boldsymbol{\Phi}(t_k+T,t_k)^T & \mathbf{P}_{CC_{k|k}}
\end{bmatrix}
$$

where $\mathbf{P}_{II_{k+1|k}}$ is computed by numerical integration of the Lyapunov equation:

$$\dot{\mathbf{P}}_{II} = \mathbf{F}\mathbf{P}_{II} + \mathbf{P}_{II}\mathbf{F}^T + \mathbf{G}\mathbf{Q}_{\text{IMU}}\mathbf{G}^T \tag{12}$$

Numerical integration is carried out for the time interval $(t_k, t_k + T)$, with initial condition $\mathbf{P}_{II_{k|k}}$. The state transition matrix $\mathbf{\Phi}(t_k + T, t_k)$ is similarly computed by numerical integration of the differential equation

$$\dot{\mathbf{\Phi}}(t_k + \tau, t_k) = \mathbf{F}\mathbf{\Phi}(t_k + \tau, t_k), \qquad \tau \in [0, T] \tag{13}$$

with initial condition $\mathbf{\Phi}(t_k, t_k) = \mathbf{I}_{15}$.

3.3 State Augmentation

Upon recording a new image, the camera pose estimate is computed from the IMU pose estimate as:

$${}^C_G\hat{\bar{q}} = {}^C_I\bar{q} \otimes {}^I_G\hat{\bar{q}}, \quad \text{and} \quad {}^G\hat{\mathbf{p}}_C = {}^G\hat{\mathbf{p}}_I + \mathbf{C}_{\hat{q}}^T\,{}^I\mathbf{p}_C \tag{14}$$

where ${}^C_I\bar{q}$ is the quaternion expressing the rotation between the IMU and camera frames, and ${}^I\mathbf{p}_C$ is the position of the origin of the camera frame with respect to $\{I\}$, both of which are known. This camera pose estimate is appended to the state vector, and the covariance matrix of the EKF is augmented accordingly:

$$\mathbf{P}_{k|k} \leftarrow \begin{bmatrix} \mathbf{I}_{6N+15} \\ \mathbf{J} \end{bmatrix} \mathbf{P}_{k|k} \begin{bmatrix} \mathbf{I}_{6N+15} \\ \mathbf{J} \end{bmatrix}^T \tag{15}$$

where the Jacobian $\mathbf{J}$ is derived from Eqs. (14) as:

$$\mathbf{J} = \begin{bmatrix} \mathbf{C}\left({}^C_I\bar{q}\right) & \mathbf{0}_{3\times 9} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 6N} \\ \lfloor \mathbf{C}_{\hat{q}}^T\,{}^I\mathbf{p}_C \times\rfloor & \mathbf{0}_{3\times 9} & \mathbf{I}_3 & \mathbf{0}_{3\times 6N} \end{bmatrix} \tag{16}$$

3.4 Measurement Model

We now present the measurement model employed for updating the state estimates, which is the primary contribution of this paper. Since the EKF is used for state estimation, for constructing a measurement model it suffices to define a residual, $\mathbf{r}$, that depends linearly on the state errors, $\widetilde{\mathbf{X}}$, according to the general form:

$$\mathbf{r} = \mathbf{H}\widetilde{\mathbf{X}} + \text{noise} \tag{17}$$

In this expression $\mathbf{H}$ is the measurement Jacobian matrix, and the noise term must be zero-mean, white, and *uncorrelated* to the state error, for the EKF framework to be applied.

To derive our measurement model, we are motivated by the fact that viewing a static feature from multiple camera poses results in constraints involving all these poses. In our work, the camera observations are grouped per *tracked feature*, rather than *per camera pose* where the measurements were recorded (the latter is the case, for example, in methods that compute pairwise constraints between poses [7,13,14]). All the measurements of the same 3D point are used to define a constraint equation (cf. Eq. (26)), relating all the camera poses at which the measurements occurred. This is achieved *without* including the feature position in the filter state vector.

We present the measurement model by considering the case of a *single* feature, f_j, that has been observed from a set of M_j camera poses $({}^{C_i}_G\bar{q}, {}^G\mathbf{p}_{C_i})$, $i \in \mathcal{S}_j$. Each of the M_j observations of the feature is described by the model:

$$\mathbf{z}_i^{(j)} = \frac{1}{{}^{C_i}Z_j} \begin{bmatrix} {}^{C_i}X_j \\ {}^{C_i}Y_j \end{bmatrix} + \mathbf{n}_i^{(j)}, \qquad i \in \mathcal{S}_j \tag{18}$$

where $\mathbf{n}_i^{(j)}$ is the 2×1 image noise vector, with covariance matrix $\mathbf{R}_i^{(j)} = \sigma_{\text{im}}^2 \mathbf{I}_2$. The feature position expressed in the camera frame, ${}^{C_i}\mathbf{p}_{f_j}$, is given by:

$$ {}^{C_i}\mathbf{p}_{f_j} = \begin{bmatrix} {}^{C_i}X_j \\ {}^{C_i}Y_j \\ {}^{C_i}Z_j \end{bmatrix} = \mathbf{C}({}^{C_i}_G\bar{q})({}^G\mathbf{p}_{f_j} - {}^G\mathbf{p}_{C_i}) \tag{19} $$

where ${}^G\mathbf{p}_{f_j}$ is the 3D feature position in the global frame. Since this is unknown, in the first step of our algorithm we employ least-squares minimization to obtain an estimate, ${}^G\hat{\mathbf{p}}_{f_j}$, of the feature position. This is achieved using the measurements $\mathbf{z}_i^{(j)}$, $i \in \mathcal{S}_j$, and the filter estimates of the camera poses at the corresponding time instants (cf. Appendix).

Once the estimate of the feature position is obtained, we compute the measurement residual:

$$ \mathbf{r}_i^{(j)} = \mathbf{z}_i^{(j)} - \hat{\mathbf{z}}_i^{(j)} \tag{20} $$

where

$$ \hat{\mathbf{z}}_i^{(j)} = \frac{1}{{}^{C_i}\hat{Z}_j} \begin{bmatrix} {}^{C_i}\hat{X}_j \\ {}^{C_i}\hat{Y}_j \end{bmatrix} \quad , \quad \begin{bmatrix} {}^{C_i}\hat{X}_j \\ {}^{C_i}\hat{Y}_j \\ {}^{C_i}\hat{Z}_j \end{bmatrix} = \mathbf{C}({}^{C_i}_G\hat{\bar{q}})({}^G\hat{\mathbf{p}}_{f_j} - {}^G\hat{\mathbf{p}}_{C_i}) $$

Linearizing about the estimates for the camera pose and for the feature position, the residual of Eq. (20) can be approximated as:

$$ \mathbf{r}_i^{(j)} \simeq \mathbf{H}_{\mathbf{X}_i}^{(j)} \widetilde{\mathbf{X}} + \mathbf{H}_{f_i}^{(j)}\, {}^G\widetilde{\mathbf{p}}_{f_j} + \mathbf{n}_i^{(j)} \tag{21} $$

In the preceding expression $\mathbf{H}_{\mathbf{X}_i}^{(j)}$ and $\mathbf{H}_{f_i}^{(j)}$ are the Jacobians of the measurement $\mathbf{z}_i^{(j)}$ with respect to the state and the feature position, respectively, and ${}^G\widetilde{\mathbf{p}}_{f_j}$ is the error in the position estimate of f_j. The Jacobians are given by:

$$ \mathbf{H}_{\mathbf{X}_i}^{(j)} = \begin{bmatrix} \mathbf{0}_{2\times15} & \mathbf{0}_{2\times6} & \dots & \underbrace{\mathbf{J}_i^{(j)} \lfloor {}^{C_i}\hat{\mathbf{X}}_{f_j} \times \rfloor \quad -\mathbf{J}_i^{(j)}\mathbf{C}({}^{C_i}_G\hat{\bar{q}})}_{\text{Jacobian wrt pose i}} & \dots \end{bmatrix} \tag{22} $$

and

$$ \mathbf{H}_{f_i}^{(j)} = \mathbf{J}_i^{(j)}\mathbf{C}({}^{C_i}_G\hat{\bar{q}}) \tag{23} $$

In the preceding expressions $\mathbf{J}_i^{(j)}$ is the Jacobian matrix

$$ \mathbf{J}_i^{(j)} = \nabla_{{}^{C_i}\hat{\mathbf{p}}_{f_j}} \mathbf{z}_i^{(j)} = \frac{1}{{}^{C_i}\hat{Z}_j} \begin{bmatrix} 1 & 0 & -\frac{{}^{C_i}\hat{X}_j}{{}^{C_i}\hat{Z}_j} \\ 0 & 1 & -\frac{{}^{C_i}\hat{Y}_j}{{}^{C_i}\hat{Z}_j} \end{bmatrix} $$

By stacking the residuals of all M_j measurements of this feature, we obtain:

$$ \mathbf{r}^{(j)} \simeq \mathbf{H}_{\mathbf{X}}^{(j)} \widetilde{\mathbf{X}} + \mathbf{H}_f^{(j)}\, {}^G\widetilde{\mathbf{p}}_{f_j} + \mathbf{n}^{(j)} \tag{24} $$

where $\mathbf{r}^{(j)}$, $\mathbf{H}_{\mathbf{X}}^{(j)}$, $\mathbf{H}_f^{(j)}$, and $\mathbf{n}^{(j)}$ are block vectors or matrices with elements $\mathbf{r}_i^{(j)}$, $\mathbf{H}_{\mathbf{X}_i}^{(j)}$, $\mathbf{H}_{f_i}^{(j)}$, and $\mathbf{n}_i^{(j)}$, for $i \in \mathcal{S}_j$. Since the feature observations in different images are independent, the covariance matrix of $\mathbf{n}^{(j)}$ is $\mathbf{R}^{(j)} = \sigma_{\text{im}}^2 \mathbf{I}_{2M_j}$.

Note that since the state estimate, $\mathbf{X}$, is used to compute the feature position estimate (cf. Appendix), the error ${}^G\widetilde{\mathbf{p}}_{f_j}$ in Eq. (24) is correlated with the errors $\widetilde{\mathbf{X}}$. Thus, the residual $\mathbf{r}^{(j)}$ is not in the form of

Eq. (17), and cannot be directly applied for measurement updates in the EKF. To overcome this problem, we define a residual $\mathbf{r}_o^{(j)}$, by projecting $\mathbf{r}^{(j)}$ on the left nullspace of the matrix $\mathbf{H}_f^{(j)}$. Specifically, if we let $\mathbf{A}$ denote the unitary matrix whose columns form the basis of the left nullspace of $\mathbf{H}_f$, we obtain:

$$\mathbf{r}_o^{(j)} = \mathbf{A}^T(\mathbf{z}^{(j)} - \hat{\mathbf{z}}^{(j)}) \simeq \mathbf{A}^T\mathbf{H}_{\mathbf{X}}^{(j)}\widetilde{\mathbf{X}} + \mathbf{A}^T\mathbf{n}^{(j)} \tag{25}$$

$$= \mathbf{H}_o^{(j)}\widetilde{\mathbf{X}}^{(j)} + \mathbf{n}_o^{(j)} \tag{26}$$

Since the $2M_j \times 3$ matrix $\mathbf{H}_f^{(j)}$ has full column rank, its left nullspace is of dimension $2M_j - 3$. Therefore, $\mathbf{r}_o^{(j)}$ is a $(2M_j - 3) \times 1$ vector. This residual is *independent* of the errors in the feature coordinates, and thus EKF updates can be performed based on it. Eq. (26) defines a *linearized* constraint between all the camera poses from which the feature f_j was observed. This expresses all the available information that the measurements $\mathbf{z}_i^{(j)}$ provide for the M_j states, and thus the resulting EKF update is optimal, except for the inaccuracies caused by linearization.

It should be mentioned that in order to compute the residual $\mathbf{r}_o^{(j)}$ and the measurement matrix $\mathbf{H}_o^{(j)}$, the unitary matrix $\mathbf{A}$ does not need to be explicitly evaluated. Instead, the projection of the vector $\mathbf{r}$ and the matrix $\mathbf{H}_{\mathbf{X}}^{(j)}$ on the nullspace of $\mathbf{H}_f^{(j)}$ can be computed very efficiently using Givens rotations [21], in $O(M_j^2)$ operations. Additionally, since the matrix $\mathbf{A}$ is unitary, the covariance matrix of the noise vector $\mathbf{n}_o^{(j)}$ is given by:

$$E\{\mathbf{n}_o^{(j)}\mathbf{n}_o^{(j)T}\} = \sigma_{\text{im}}^2\mathbf{A}^T\mathbf{A} = \sigma_{\text{im}}^2\mathbf{I}_{2M_j-3}$$

The residual defined in Eq. (25) is not the only possible expression of the geometric constraints that are induced by observing a static feature in M_j images. An alternative approach would be, for example, to employ the epipolar constraints that are defined for each of the $M_j(M_j-1)/2$ pairs of images. However, the resulting $M_j(M_j-1)/2$ equations would still correspond to only $2M_j - 3$ independent constraints, since each measurement is used multiple times, rendering the equations statistically correlated. Our experiments have shown that employing linearization of the epipolar constraints results in a significantly more complex implementation, and yields inferior results compared to the approach described above.

3.5 EKF Updates

In the preceding section, we presented a measurement model that expresses the geometric constraints imposed by observing a static feature from multiple camera poses. We now present in detail the update phase of the EKF, in which the constraints from observing multiple features are used. EKF updates are triggered by one of the following two events:

- When a feature that has been tracked in a number of images is no longer detected, then all the measurements of this feature are processed using the method presented in Section 3.4. This case occurs most often, as features move outside the camera's field of view.

- Every time a new image is recorded, a copy of the current camera pose estimate is included in the state vector (cf. Section 3.3). If the maximum allowable number of camera poses, N_{max}, has been reached, at least one of the old ones must be removed. Prior to discarding states, all the feature observations that occurred at the corresponding time instants are used, in order to utilize their localization information. In our algorithm, we choose $N_{\text{max}}/3$ poses that are evenly spaced in time, starting from the second-oldest pose. These are discarded after carrying out an EKF update using the constraints of features that are common to these poses. We have opted to always keep the oldest pose in the state vector, because the geometric constraints that involve poses further back in time typically correspond to larger baseline, and hence carry more valuable positioning information. This approach was shown to perform very well in practice.

We hereafter discuss the update process in detail. Consider that at a given time step the constraints of L features, selected by the above two criteria, must be processed. Following the procedure described in the preceding section, we compute a residual vector $\mathbf{r}_o^{(j)}$, $j = 1 \ldots L$, as well as a corresponding measurement matrix $\mathbf{H}_o^{(j)}$, $j = 1 \ldots L$ for each of these features (cf. Eq. (25)). By stacking all residuals in a single vector, we obtain:

$$\mathbf{r}_o = \mathbf{H}_{\mathbf{X}}\widetilde{\mathbf{X}} + \mathbf{n}_o \tag{27}$$

where $\mathbf{r}_o$ and $\mathbf{n}_o$ are vectors with block elements $\mathbf{r}_o^{(j)}$ and $\mathbf{n}_o^{(j)}$, $j = 1 \ldots L$, respectively, and $\mathbf{H}_{\mathbf{X}}$ is a matrix with block rows $\mathbf{H}_{\mathbf{X}}^{(j)}$, $j = 1 \ldots L$.

Since the feature measurements are statistically independent, the noise vectors $\mathbf{n}_o^{(j)}$ are uncorrelated. Therefore, the covariance matrix of the noise vector $\mathbf{n}_o$ is equal to $\mathbf{R}_o = \sigma_{\text{im}}^2 \mathbf{I}_d$, where $d = \sum_{j=1}^{L}(2M_j - 3)$ is the dimension of the residual $\mathbf{r}_o$. One issue that arises in practice is that d can be a quite large number. For example, if 10 features are seen in 10 camera poses each, the dimension of the residual is 170. In order to reduce the computational complexity of the EKF update, we employ the QR decomposition of the matrix $\mathbf{H}_{\mathbf{X}}$ [9]. Specifically, we denote this decomposition as

$$\mathbf{H}_{\mathbf{X}} = \begin{bmatrix}\mathbf{Q}_1 & \mathbf{Q}_2\end{bmatrix}\begin{bmatrix}\mathbf{T}_H \\ \mathbf{0}\end{bmatrix}$$

where $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are unitary matrices whose columns form bases for the range and nullspace of $\mathbf{H}_{\mathbf{X}}$, respectively, and $\mathbf{T}_H$ is an upper triangular matrix. With this definition, Eq. (27) yields:

$$\mathbf{r}_o = \begin{bmatrix}\mathbf{Q}_1 & \mathbf{Q}_2\end{bmatrix}\begin{bmatrix}\mathbf{T}_H \\ \mathbf{0}\end{bmatrix}\widetilde{\mathbf{X}} + \mathbf{n}_o \Rightarrow \tag{28}$$

$$\begin{bmatrix}\mathbf{Q}_1^T\mathbf{r}_o \\ \mathbf{Q}_2^T\mathbf{r}_o\end{bmatrix} = \begin{bmatrix}\mathbf{T}_H \\ \mathbf{0}\end{bmatrix}\widetilde{\mathbf{X}} + \begin{bmatrix}\mathbf{Q}_1^T\mathbf{n}_o \\ \mathbf{Q}_2^T\mathbf{n}_o\end{bmatrix} \tag{29}$$

From the last equation it becomes clear that by projecting the residual $\mathbf{r}_o$ on the basis vectors of the range of $\mathbf{H}_{\mathbf{X}}$, we retain all the useful information in the measurements. The residual $\mathbf{Q}_2^T\mathbf{r}_o$ is only noise, and can be completely discarded. For this reason, instead of the residual shown in Eq. (27), we employ the following residual for the EKF update:

$$\mathbf{r}_n = \mathbf{Q}_1^T\mathbf{r}_o = \mathbf{T}_H\widetilde{\mathbf{X}} + \mathbf{n}_n \tag{30}$$

In this expression $\mathbf{n}_n = \mathbf{Q}_1^T\mathbf{n}_o$ is a noise vector whose covariance matrix is equal to $\mathbf{R}_n = \mathbf{Q}_1^T\mathbf{R}_o\mathbf{Q}_1 = \sigma_{\text{im}}^2\mathbf{I}_r$, with r being the number of columns in $\mathbf{Q}_1$. The EKF update proceeds by computing the Kalman gain:

$$\mathbf{K} = \mathbf{P}\mathbf{T}_H^T\left(\mathbf{T}_H\mathbf{P}\mathbf{T}_H^T + \mathbf{R}_n\right)^{-1} \tag{31}$$

while the correction to the state is given by the vector

$$\Delta\mathbf{X} = \mathbf{K}\mathbf{r}_n \tag{32}$$

Finally, the state covariance matrix is updated according to:

$$\mathbf{P}_{k+1|k+1} = \left(\mathbf{I}_\xi - \mathbf{K}\mathbf{T}_H\right)\mathbf{P}_{k+1|k}\left(\mathbf{I}_\xi - \mathbf{K}\mathbf{T}_H\right)^T + \mathbf{K}\mathbf{R}_n\mathbf{K}^T \tag{33}$$

where $\xi = 6N + 15$ is the dimension of the covariance matrix.

It is interesting to examine the computational complexity of the operations needed during the EKF update. The residual $\mathbf{r}_n$, as well as the matrix $\mathbf{T}_H$, can be computed using Givens rotations in $O(r^2d)$

operations, without the need to explicitly form $\mathbf{Q}_1$. On the other hand, Eq. (33) involves multiplication of square matrices of dimension ξ, an $O(\xi^3)$ operation. Therefore, the cost of the EKF update is $\max(O(r^2 d), O(\xi^3))$. If, on the other hand, the residual vector $\mathbf{r}_o$ was employed, without projecting it on the range of $\mathbf{H}_{\mathbf{X}}$, the computational cost of computing the Kalman gain would have been $O(d^3)$. Since typically $d \gg \xi, r$, we see that the use of the residual $\mathbf{r}_n$ results in substantial savings in computation.

3.6 Discussion

We now study some of the properties of the described algorithm. As shown in the previous section, the filter's computational complexity is *linear* in the number of observed features, and at most *cubic* in the number of states that are included in the state vector. Thus, the number of poses that are included in the state is the most significant factor in determining the computational cost of the algorithm. Since this number is a selectable parameter, it can be tuned according to the available computing resources, and the accuracy requirements of a given application. If required, the length of the filter state can be also adaptively controlled during filter operation, to adjust to the varying availability of resources.

One source of difficulty in recursive state estimation with camera observations is the nonlinear nature of the measurement model. Vision-based motion estimation is very sensitive to noise, and, especially when the observed features are at large distances, false local minima can cause convergence to inconsistent solutions [22]. The problems introduced by nonlinearity have been addressed in the literature using techniques such as Sigma-point Kalman filtering [23], particle filtering [4], and the inverse depth representation for features [24]. Two characteristics of the described algorithm increase its robustness to linearization inaccuracies: (i) the inverse feature depth parametrization used in the measurement model (cf. Appendix) and (ii) the *delayed linearization* of measurements [17]. By the algorithm's construction, multiple observations of each feature are collected, prior to using them for EKF updates, resulting in more accurate evaluation of the measurement Jacobians.

One interesting observation is that in typical image sequences, most features can only be reliably tracked over a small number of frames ("opportunistic" features), and only few can be tracked for long periods of time, or when re-visiting places (persistent features). This is due to the limited field of view of cameras, as well as occlusions, image noise, and viewpoint changes, that result in failures of the feature tracking algorithms. As previously discussed, if all the poses in which a feature has been seen are included in the state vector, then the proposed measurement model is *optimal*, except for linearization inaccuracies. Therefore, for realistic image sequences, the proposed algorithm is able to optimally use the localization information of the opportunistic features. Moreover, we note that the state vector $\mathbf{X}_k$ is *not* required to contain *only* the IMU and camera poses. If desired, the persistent features can be included in the filter state, and used for SLAM. This would further improve the attainable localization accuracy, within areas with lengthy loops.

4 Experimental results

The algorithm described in the preceding sections has been tested extensively both in simulation and with real data. In this section, some representative results are discussed.

4.1 Simulation results

Several simulation tests have been carried out, in order to verify the performance of the proposed algorithm. We here present results demonstrating the consistency of the multi-state constraint filter. In Fig. 1(a), the estimated trajectory of the camera for a typical simulation run is plotted. For this particular trial, 1000 visual features are randomly placed on the walls of a 12m×12m simulated room, and the camera is moving in a circular trajectory of radius 3m. The camera is recording images at 1Hz, while moving at a velocity of 0.1m/sec. In Figs. 1(b-d), the estimation errors for the IMU position, orientation, and velocity are shown (blue lines), for 10 trials under this simulation setup. These errors are compared against the corresponding $\pm 3\sigma$ bounds computed using the estimated covariance (red enveloping lines). From these plots it becomes clear that the errors are commensurate with the computed covariance, and thus the estimator is *consistent*. Although the results shown in Fig. 1 pertain to this particular simulation setup, the results are typical; through numerous simulations we have verified that the estimates produced by the multiple-state constraint filter are generally consistent. Given the severe nonlinearities that arise in vision-based estimation, this is a very significant property of the algorithm.

4.2 Real-world experiments

In order to verify the ability of the proposed algorithm to operate in a real-world setting, we have also conducted experiments with real image sequences. We here present two experiments, one carried out indoors, and one outdoors. In both cases, the system used is comprised of a Pointgrey FireFly camera, registering images of resolution 640 × 480 pixels and an Inertial Science ISIS IMU, providing inertial measurements at a rate of 100Hz. During both experiments, all data was stored on a computer, and processing was done off-line. For the results shown here, feature extraction and matching was performed using the SIFT algorithm [25]. In the filter state vector a maximum of 30 camera poses were maintained. Since features were rarely tracked for more than 30 images, this number was sufficient for utilizing most of the available constraints between states.

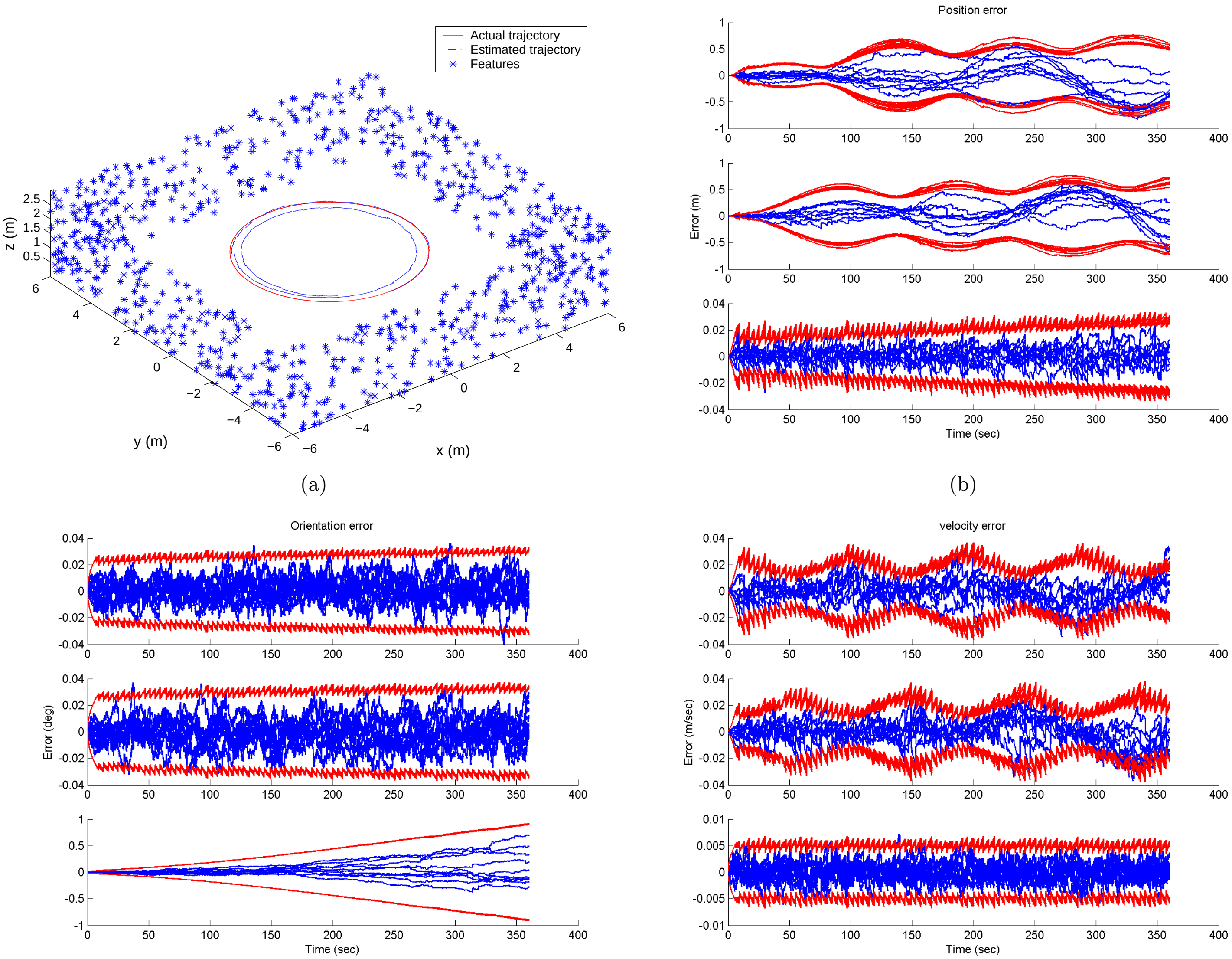
4.2.1 Indoor Experiment

For the indoor experiment, the camera/IMU system was moved manually inside a university office, on a 84m-long trajectory. Some example images from the sequence of 688 frames (recorded at 2 Hz) are shown in Fig. 2, while the complete dataset can be found online at [26]. The estimated trajectory of the IMU can be seen in Fig. 3. In this plot, the initial position of the IMU is denoted by a red square, while the final position estimate is denoted by a star. Even though ground truth for the entire duration of the experiment is not available, it is known that during this motion the system was returned to its initial position, ${}^{G}\mathbf{p}_{\text{init}} = [0 \;\; 0 \;\; 0]^T$m, twice: once at $t = 220$sec, and once at the end of the trajectory. At these two time instants the position estimates are equal to ${}^{G}\hat{\mathbf{p}}_1 = [-0.12 \;\; 0.20 \quad -0.02]^T$m and ${}^{G}\hat{\mathbf{p}}_{\text{final}} = [-0.10 \;\; 0.36 \;\; -0.03]^T$m, respectively. It is important to note that these estimates correspond to errors smaller than 0.5% of the travelled distance. Moreover, the position errors agree with the estimated 3σ values, for the position estimate, which are shown in Fig. 4. In this figure, the 3σ values corresponding to the estimated covariance for the IMU position, attitude, and velocity, are plotted.

4.2.2 Outdoor Experiment

In order to test the algorithm in an uncontrolled environment, we have also conducted an outdoor experiment, during which the camera/IMU system was placed on a car, moving on the streets of a residential area in Minneapolis, MN. Some example images from the image sequence (recorded at 3 Hz) are shown in Fig. 5, and a video of all 1598 images, which were stored in about 9 minutes of driving, can be found online at [26]. Even though images were only recorded at 3Hz due to limited hard disk space on the test system, the estimation algorithm is able to process the dataset at 14Hz, on a single core of an Intel T7200 processor (2GHz clock rate). During the experiment, a total of 142903 features were successfully tracked and used for EKF updates, along a 3.2km-long trajectory. A GPS sensor was not available during the experiment, and therefore no ground-truth trajectory data exists. However, the quality of the position estimates can be evaluated using a map of the area.

In Fig. 6, the estimated trajectory is plotted on a map of the neighborhood where the experiment took place. We observe that this trajectory follows the street layout quite accurately and, additionally,



(a)

(b)

(c)

(d)

Figure 1: (a) The trajectory used for the simulation experiments. The solid red line denotes the actual trajectory, the dash-dotted blue line represents the estimated trajectory for a single trial, while the feature positions are shown with asterisks. (b-d) The position, attitude, and velocity errors (blue lines), respectively, and the $\pm 3\sigma$ bounds (red enveloping lines) for for 10 simulation trials.

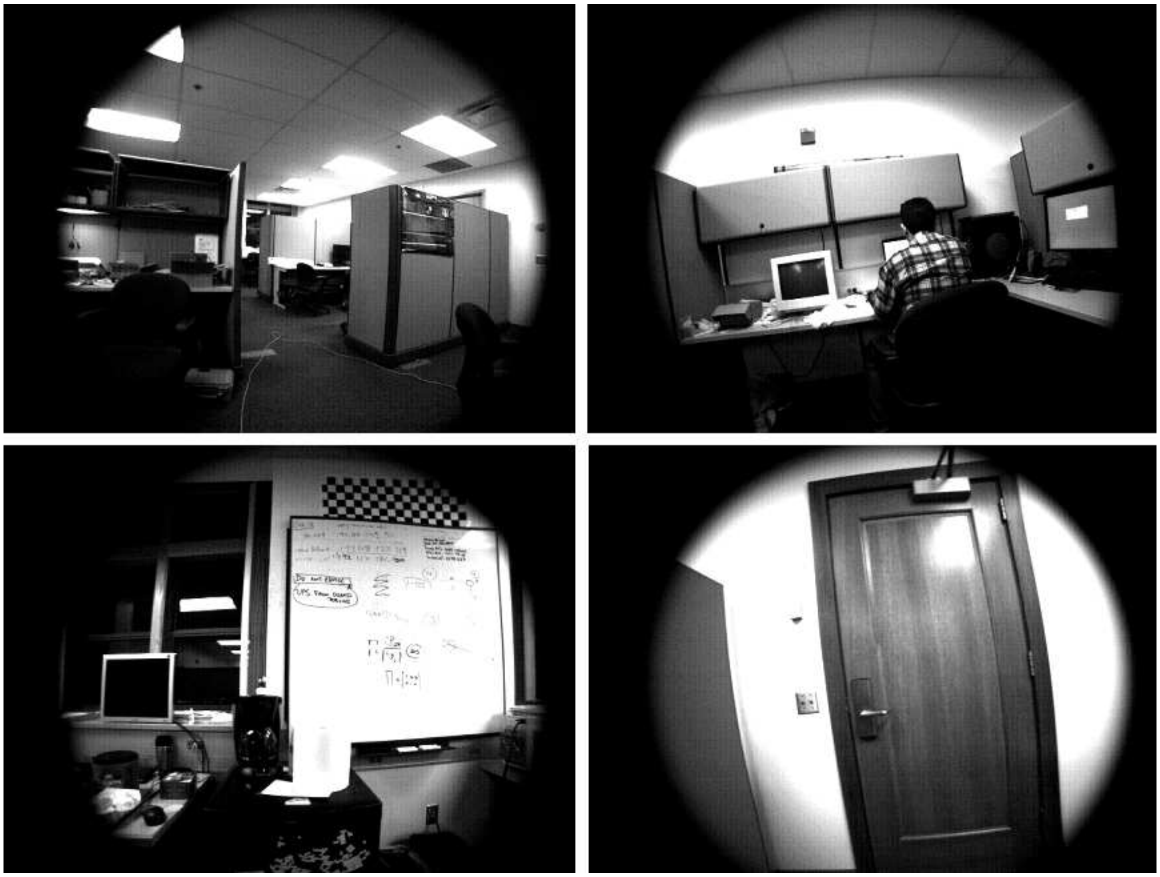


Figure 2: Some example images from the sequence recorded during the indoor experiment. Note the severe distortion caused by the use of a 4mm-lens with a relatively wide field of view (55 degrees). This lens also causes the image to be projected in the central part of the CCD sensor, resulting in "blacked-out" areas in outer parts of the images.

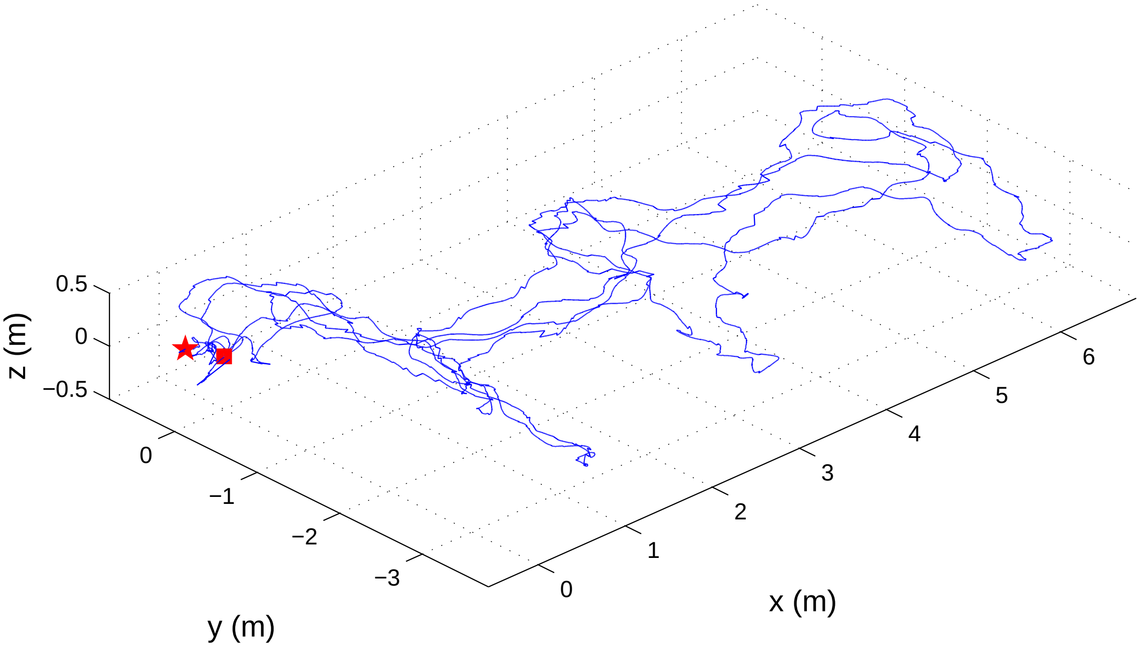
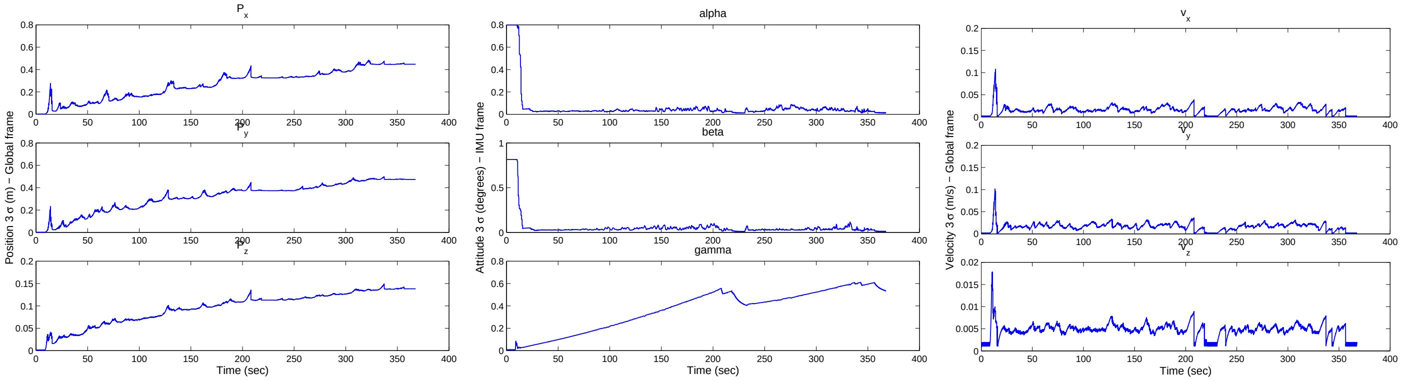


Figure 3: The estimated 3D trajectory of the IMU. The initial position is denoted by a red square, while the final one is denoted by a red star.



(a)

(b)

(c)

Figure 4: The 3σ bounds for the errors in the position, attitude, and velocity, for the indoor experiment. The plotted values are 3-times the square roots of the corresponding diagonal elements of the state covariance matrix.

the position errors that can be inferred from this plot agree with the 3σ bounds shown in Fig 7(a). The final position estimate, expressed with respect to the starting pose, is $\hat{\mathbf{X}}_{\text{final}} = [-7.92 \quad 13.14 \quad -0.78]^T$m. From the initial and final parking spot of the vehicle it is known that the true final position expressed with respect to the initial pose is approximately $\mathbf{X}_{\text{final}} = [0 \quad 7 \quad 0]^T$m. Thus, the final position error is approximately 10m in a trajectory of 3.2km, i.e., an error of 0.31% of the travelled distance. This is remarkable, given that the algorithm does *not* utilize loop closing, and uses no prior information (for example, non-holonomic constraints or a street map) about the car motion. Moreover, it is worth pointing out that the camera motion is almost parallel to the optical axis, a condition which is particularly adverse for image-based motion estimation algorithms [22]. In Figs. 7(b) and 7(c), the 3σ bounds for the errors in the IMU attitude and velocity along the three axes are shown. From these, we observe that the algorithm obtains accuracy (3σ) better than 1^o for attitude, and better than 0.35m/sec for velocity in this particular experiment.

The results shown here demonstrate that the proposed algorithm is capable of operating in a real-world environment, and producing very accurate pose estimates in real-time. We should point out that in the dataset presented here several moving objects appear, such as cars, pedestrians, and trees whose leaves move in the wind. The algorithm is able to discard the outliers which arise from visual features detected on these objects, using a simple Mahalanobis distance test. Robust outlier rejection is facilitated by the fact that multiple observations of each feature are available, and thus visual features that do not correspond to static objects become easier to detect. As a final remark, we note that the described method can be used either as a stand-alone pose estimation algorithm, or combined with additional sensing modalities to provide increased accuracy. For example, if a GPS sensor was available during this experiment, its measurements could be used to compensate for position drift.

5 Conclusions

In this paper we have presented an EKF-based estimation algorithm for real-time vision-aided inertial navigation. The main contribution of this work is the derivation of a measurement model that is able to express the geometric constraints that arise when a static feature is observed from multiple camera poses. This measurement model does not require including the 3D feature positions in the state vector of the EKF, and is optimal, up to the errors introduced by linearization. The resulting EKF-based pose estimation algorithm has computational complexity *linear* in the number of features, and is capable of very accurate pose estimation in large-scale real environments. In this paper the presentation has only focused on fusing inertial measurements with visual measurements from a monocular camera. However, the approach is general and can be adapted to different sensing modalities both for the proprioceptive, as well as for the exteroceptive measurements (e.g., for fusing wheel odometry and laser scanner data).

Acknowledgements

This work was supported by the University of Minnesota (DTC), the NASA Mars Technology Program (MTP-1263201), and the National Science Foundation (EIA-0324864, IIS-0643680). The authors would like to thank Faraz Mirzaei for his invaluable help with the experiments.

Appendix

To compute an estimate of the position of a tracked feature f_j we employ *intersection* [27]. To avoid local minima, and for better numerical stability, during this process we use an inverse-depth parametrization of the feature position [24]. In particular, if $\{C_n\}$ is the camera frame in which the feature was observed

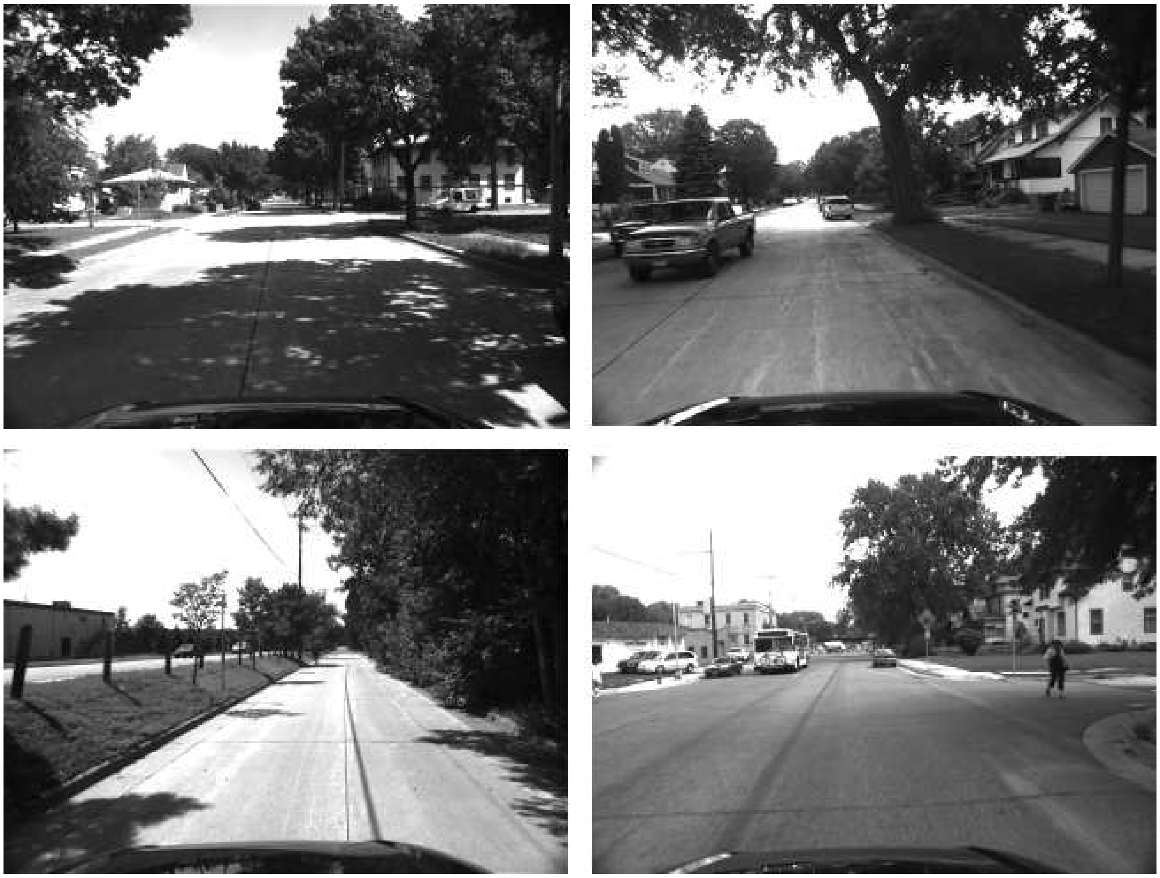


Figure 5: Some images from the dataset used for the experiment. The entire video sequence can be found at [26].

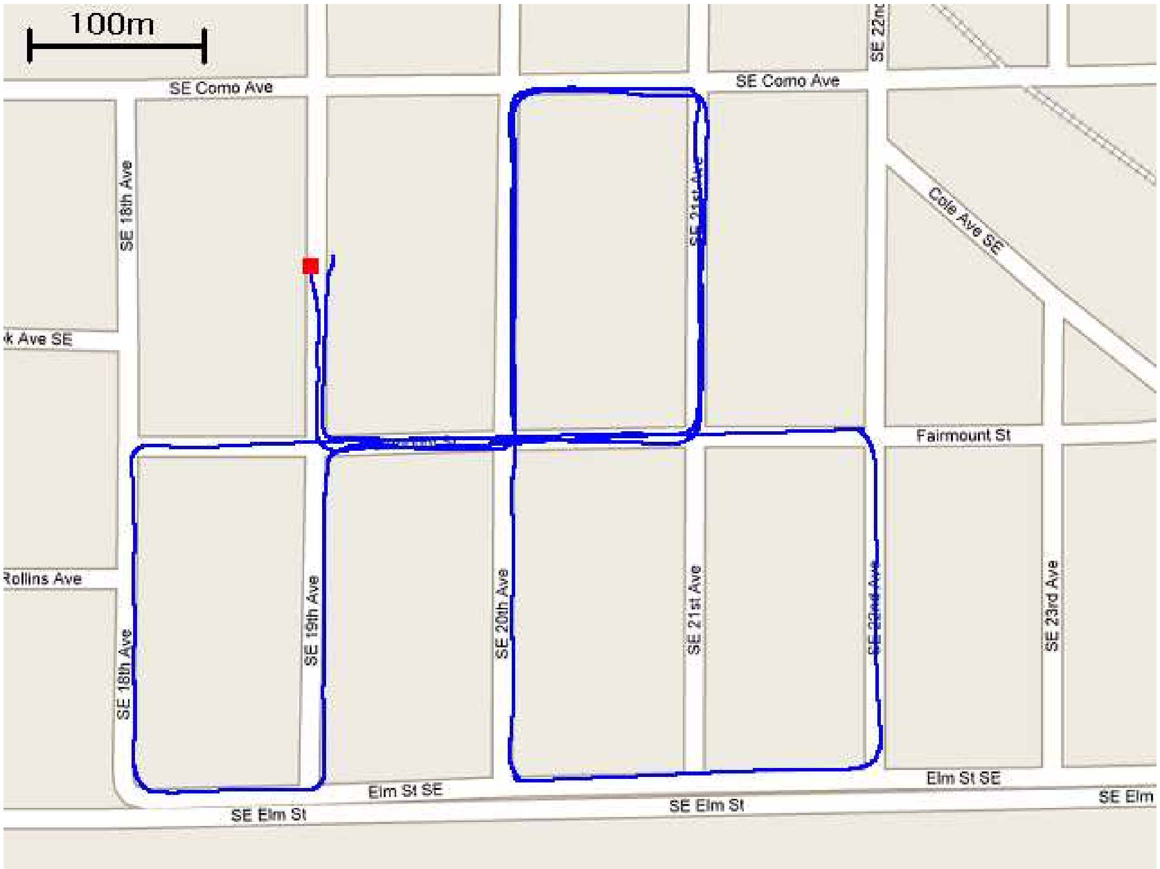
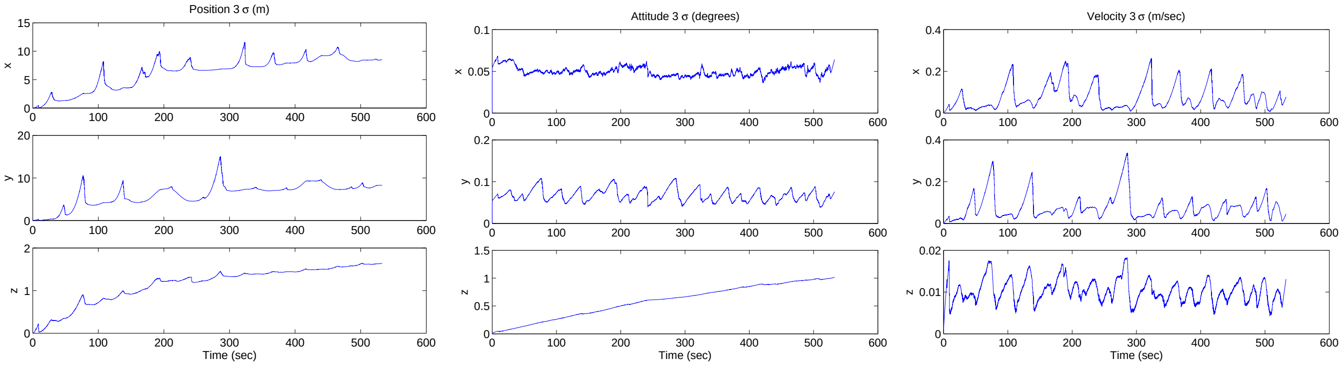


Figure 6: The estimated trajectory overlaid on a map of the area where the experiment took place. The initial position of the car is denoted by a red square, and the scale of the map is shown on the top left corner.



(a) (b) (c)

Figure 7: The 3σ bounds for the errors in the position, attitude, and velocity. The plotted values are 3-times the square roots of the corresponding diagonal elements of the state covariance matrix. Note that the EKF state is expressed in ECEF frame, but for plotting we have transformed all quantities in the initial IMU frame, whose x axis is pointing approximately south, and its y axis east.

for the first time, then the feature coordinates with respect to the camera at the i-th time instant are:

$$ {}^{C_i}\mathbf{p}_{f_j} = \mathbf{C}({}^{C_i}_{C_n}\bar{q})\,{}^{C_n}\mathbf{p}_{f_j} + {}^{C_i}\mathbf{p}_{C_n}, \qquad i \in \mathcal{S}_j \tag{34} $$

In this expression $\mathbf{C}({}^{C_i}_{C_n}\bar{q})$ and ${}^{C_i}\mathbf{p}_{C_n}$ are the rotation and translation between the camera frames at time instants n and i, respectively. Eq. (34) can be rewritten as:

$$ \begin{aligned} {}^{C_i}\mathbf{p}_{f_j} &= {}^{C_n}Z_j \left(\mathbf{C}({}^{C_i}_{C_n}\bar{q}) \begin{bmatrix} \frac{{}^{C_n}X_j}{{}^{C_n}Z_j} \\ \frac{{}^{C_n}Y_j}{{}^{C_n}Z_j} \\ 1 \end{bmatrix} + \frac{1}{{}^{C_n}Z_j}\,{}^{C_i}\mathbf{p}_{C_n} \right) & (35) \\ &= {}^{C_n}Z_j \left(\mathbf{C}({}^{C_i}_{C_n}\bar{q}) \begin{bmatrix} \alpha_j \\ \beta_j \\ 1 \end{bmatrix} + \rho_j\ {}^{C_i}\mathbf{p}_{C_n} \right) & (36) \\ &= {}^{C_n}Z_j \begin{bmatrix} h_{i1}(\alpha_j, \beta_j, \rho_j) \\ h_{i2}(\alpha_j, \beta_j, \rho_j) \\ h_{i3}(\alpha_j, \beta_j, \rho_j) \end{bmatrix} & (37) \end{aligned} $$

In the last expression h_{i1}, h_{i2} and h_{i3} are scalar functions of the quantities $\alpha_j, \beta_j, \rho_j$, which are defined as:

$$ \alpha_j = \frac{{}^{C_n}X_j}{{}^{C_n}Z_j}, \qquad \beta_j = \frac{{}^{C_n}Y_j}{{}^{C_n}Z_j}, \qquad \rho_j = \frac{1}{{}^{C_n}Z_j}, \tag{38} $$

Substituting from Eq. (37) into Eq. (18) we can express the measurement equations as functions of α_j, β_j and ρ_j only:

$$ \mathbf{z}_i^{(j)} = \frac{1}{h_{i3}(\alpha_j, \beta_j, \rho_j)} \begin{bmatrix} h_{i1}(\alpha_j, \beta_j, \rho_j) \\ h_{i2}(\alpha_j, \beta_j, \rho_j) \end{bmatrix} + \mathbf{n}_i^{(j)} \tag{39} $$

Given the measurements $\mathbf{z}_i^{(j)}, i \in \mathcal{S}_j$, and the estimates for the camera poses in the state vector, we can obtain estimates for $\hat{\alpha}_j, \hat{\beta}_j$, and $\hat{\rho}_j$, using Gauss-Newton least-squares minimization. Then, the global feature position is computed by:

$$ {}^{G}\hat{\mathbf{p}}_{f_j} = \frac{1}{\hat{\rho}_j}\mathbf{C}^T({}^{C_n}_{G}\hat{\bar{q}}) \begin{bmatrix} \hat{\alpha}_j \\ \hat{\beta}_j \\ 1 \end{bmatrix} + {}^{G}\hat{\mathbf{p}}_{C_n} \tag{40} $$

We note that during the least-squares minimization process the camera pose estimates are treated as known constants, and their covariance matrix is ignored. As a result, the minimization can be carried out very efficiently, at the expense of the optimality of the feature position estimates. Recall, however, that up to a first-order approximation, the errors in these estimates do *not* affect the measurement residual (cf. Eq. (25)). Thus, no significant degradation of performance is inflicted.

References

[1] J. W. Langelaan, "State estimation for autonomous flight in cluttered environments," Ph.D. dissertation, Stanford University, Department of Aeronautics and Astronautics, 2006.

[2] D. Strelow, "Motion estimation from image and inertial measurements," Ph.D. dissertation, Carnegie Mellon University, 2004.

[3] L. L. Ong, M. Ridley, J. H. Kim, E. Nettleton, and S. Sukkarieh, "Six DoF decentralised SLAM," in *Australasian Conf. on Robotics and Automation*, Brisbane, Australia, December 2003, pp. 10–16.

[4] E. Eade and T. Drummond, "Scalable monocular SLAM," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 17-26 2006, pp. 469 – 476.

[5] A. Chiuso, P. Favaro, H. Jin, and S. Soatto, "Structure from motion causally integrated over time," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 523–535, April 2002.

[6] A. J. Davison and D. W. Murray, "Simultaneous localisation and map-building using active vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 865 – 880, July 2002.

[7] S. Roumeliotis, A. Johnson, and J. Montgomery, "Augmenting inertial navigation with image-based motion estimation," in *IEEE International Conference on Robotics and Automation*, Washington D.C., 2002, pp. 4326–33.

[8] D. D. Diel, "Stochastic constraints for vision-aided inertial navigation," Master's thesis, MIT, January 2005.

[9] D. S. Bayard and P. B.Brugarolas, "An estimation algorithm for vision-based exploration of small bodies in space," in *American Control Conference*, June 8-10 2005, pp. 4589 – 4595.

[10] S. Soatto, R. Frezza, and P. Perona, "Motion estimation via dynamic vision," *IEEE Transactions on Automatic Control*, vol. 41, no. 3, pp. 393–413, March 1996.

[11] S. Soatto and P. Perona, "Recursive 3-d visual motion estimation using subspace constraints," *IEEE Transactions on Automatic Control*, vol. 22, no. 3, pp. 235–259, 1997.

[12] R. J. Prazenica, A. Watkins, and A. J. Kurdila, "Vision-based kalman filtering for aircraft state estimation and structure from motion," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, no. AIAA 2005-6003, San Fransisco, CA, Aug. 15-18 2005.

[13] R. Garcia, J. Puig, P. Ridao, and X. Cufi, "Augmented state Kalman filtering for AUV navigation," in *IEEE International Conference on Robotics and Automation*, Washington D.C., 2002, pp. 4010–4015.

[14] R. Eustice, H. Singh, J. Leonard, M. Walter, and R. Ballard, "Visually navigating the RMS Titanic with SLAM information filters," in *Proceedings of Robotics: Science and Systems*, Cambridge, MA, June 2005.

[15] A. I. Mourikis and S. I. Roumeliotis, "On the treatment of relative-pose measurements for mobile robot localization," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Orlando, FL, May 15-19 2006, pp. 2277 – 2284.

[16] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *Journal of Field Robotics*, vol. 23, no. 1, pp. 3–20, January 2006.

[17] P. McLauchlan, "The variable state dimension filter," Centre for Vision, Speech and Signal Processing, University of Surrey, UK, Tech. Rep., 1999.

[18] M. C. Deans, "Maximally informative statistics for localization and mapping," in *IEEE International Conference on Robotics and Automation*, Washington D.C., May 2002, pp. 1824–1829.

[19] W. G. Breckenridge, "Quaternions proposed standard conventions," JPL, Tech. Rep. INTEROFFICE MEMORANDUM IOM 343-79-1199, 1999.

[20] A. B. Chatfield, *Fundamentals of High Accuracy Inertial Navigation.* Reston, VA: AIAA, 1997.

[21] G. Golub and C. van Loan, *Matrix computations.* The Johns Hopkins University Press, London, 1996.

[22] J. Oliensis, "A new structure-from-motion ambiguity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 7, pp. 685–700, July 2000.

[23] A. Huster, "Relative position sensing by fusing monocular vision and inertial rate sensors," Ph.D. dissertation, Department of Electrical Engineering, Stanford University, 2003.

[24] A. D. J. Montiel, J. Civera, "Unified inverse depth parametrization for monocular slam," in *Proceedings of Robotics: Science and Systems*, Philadelphia, PA, June 2006.

[25] D. G. Lowe, "Distinctive image features from scale-ivnariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–100, 2004.

[26] http://www.cs.umn.edu/∼mourikis/icra07video.htm.

[27] B. Triggs, P. McLauchlan, R. Hartley, and Fitzgibbon, "Bundle adjustment – a modern synthesis," in *Vision Algorithms: Theory and Practice.* Springer Verlag, 2000, pp. 298–375.