A Dual-Layer Estimator Architecture for Long-term Localization

Anastasios I. Mourikis and Stergios I. Roumeliotis
Dept. of Computer Science and Engineering, University of Minnesota
{mourikis|stergios}@cs.umn.edu *

Abstract

In this paper, we present a localization algorithm for estimating the 3D position and orientation (pose) of a moving vehicle based on visual and inertial measurements. The main advantage of the proposed method is that it provides precise pose estimates at low computational cost. This is achieved by introducing a two-layer estimation architecture that processes measurements based on their information content. Inertial measurements and feature tracks between consecutive images are processed locally in the first layer (Multi-State-Constraint Kalman filter) providing estimates for the motion of the vehicle at a high rate. The second layer comprises a bundle adjustment iterative estimator that operates intermittently so as to (i) reduce the effect of the linearization errors, and (ii) update the state estimates every time an area is re-visited and features are re-detected (loop closure). Through this process reliable state estimates are available continuously, while the estimation errors remain bounded during long-term operation. The performance of the developed system is demonstrated in large-scale experiments, involving a vehicle localizing within an urban area.

1. Introduction

In this paper, we focus on the problem of tracking the pose of a mobile platform by combining visual and inertial measurements. In particular, the main contribution of this work is a system capable of long-term, accurate, and real-time pose estimation using Inertial Measurement Unit (IMU) and monocular-camera measurements. The key characteristic of the system is its dual-layer estimation architecture (cf. Fig. 1): At the first layer, a combined visual/inertial odometry estimator fuses the visual and inertial measurements to continuously track the 3D motion of the camera. The Multi-State Constraint Kalman Filter (MSC-KF) [9] estimator, employed for this task, offers real-time performance, and reports the camera pose estimates at the IMU data rate. However, since the MSC-KF utilizes no loop-closure information, the uncertainty of the state estimates will gradually increase over time. In order to compensate for the error growth, at the second layer of the architecture we employ a least-squares Bundle-Adjustment (BA) estimator in conjunction with a loop-closure detection module. Every time a loop-closing event is detected, the re-observations of previously seen features are processed by the BA algorithm, to obtain an improved pose estimate. This estimate is then fed back to the visual/inertial odometry module, thus resulting in diminished localization errors.

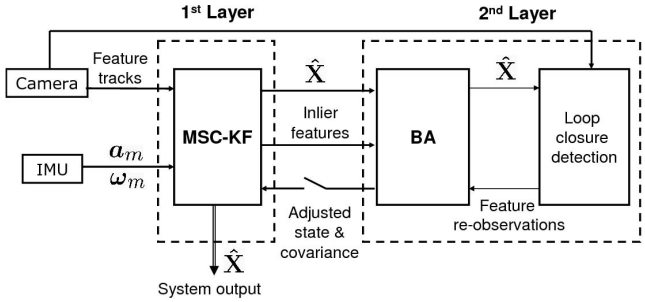

Figure 1. The block diagram of the system.

Two key properties render the proposed dual-layer architecture suitable for long-term, real-time localization: Firstly, it is important to note that the MSC-KF, utilized in the first layer, *tightly couples* the visual and inertial measurements, and has computational complexity only *linear* in the number of locally visible features. Therefore, it is able to process all the available feature measurements in real-time, producing motion estimates of high accuracy. Secondly, the use of BA for processing loop-closure information ensures that the computational burden of loop-closing is incurred *only* when a loop closure actually occurs (cf. Section 4). Since loop closing is typically an infrequent event, the additional cost in processing time is minimal compared to using the MSC-KF alone (cf. Section 5). As a result of the two aforementioned characteristics, the proposed dual-layer architecture is capable of producing pose estimates that are available both in real time, *and* with bounded long-term errors.

2. Related Work

In this section, we briefly discuss existing approaches for processing visual feature observations *and* inertial measurements. One family of such algorithms track the trajectory of the camera over time, without estimating the structure of the environment, and are typically termed visual-odometry methods. The most computationally efficient of these methods utilize the feature measurements to derive constraints between *pairs* of consecutive camera poses, and then fuse them with the inertial measurements. For example in [14],

*This work was supported by the University of Minnesota (DTC), and the National Science Foundation (EIA-0324864, IIS-0643680). Anastasios Mourikis is supported by the UMN Doctoral Dissertation Fellowship.

an image-based motion estimation algorithm is applied, to obtain displacement estimates between consecutive camera poses. Similarly, in [1, 3] constraints between the current and previous pose are defined using the epipolar geometry. In both cases, the derived constraints are combined with IMU measurements using an Extended Kalman Filter (EKF). Applying only pairwise constraints, however, is suboptimal when a feature is seen in multiple images.

Contrary to the aforementioned approaches, the MSC-KF algorithm, which is employed in the first layer of the proposed localization architecture, uses the feature measurements to impose constraints between *all* consecutive poses from which a feature is seen. This is similar in spirit to visual-odometry approaches that use bundle adjustment over a sliding window of camera poses [7, 10]. Similarly to the MSC-KF, these approaches temporarily initialize features, use them for imposing constraints on windows of consecutive camera poses, and then discard them. These approaches, however, employ only a *loose* coupling between the visual and inertial measurements: the IMU rotational velocity measurements are used to independently compute attitude estimates, which are subsequently fused with the results of the visual odometry module. In this case, the resulting estimates are suboptimal, because the IMU biases are not updated. A loose coupling of visual and inertial measurements is also employed in the system presented in [12], which uses multiple cameras for visual odometry, and then fuses the result with that of pure IMU-based pose tracking. In contrast, in the MSC-KF the visual and inertial measurements are fused in a *tightly coupled* formulation, which results in increased accuracy.

All the aforementioned approaches only process the measurements for motion tracking, and do *not* utilize loop closure information. On the other end of the spectrum lie Simultaneous Localization and Mapping (SLAM) algorithms, which jointly estimate the current IMU pose, as well as the 3D positions of all landmarks. The estimation is typically carried out by an EKF [11, 13, 15]. The fundamental advantage of EKF-SLAM algorithms is that, because the feature positions are maintained in the filter state vector, feature re-observations that occur when the camera re-visits an area can be readily processed. However, EKF-based SLAM methods have computational complexity *quadratic* in the *total* number of features estimated. Most importantly, even though the camera observes only a small number of features at each time instant, the covariance matrix for *all* the features in the state vector needs to be updated at every time step. Thus the quadratic cost of updating the entire state is incurred at every time step.

In contrast to EKF-based SLAM, in the BA formulation, which is employed for processing loop closure information in our system, the information (i.e., Hessian) matrix remains naturally sparse. Feature observations introduce new terms involving *only* the particular feature and the camera poses from which it was observed. Thus, the computational cost of updating the Hessian with the new measurement information is constant. Additionally, solving the system is postponed until a loop closure occurs. We note that a batch algorithm for estimating the camera trajectory using visual and inertial measurements is also presented in [15]. However, in that work the constraints that are introduced by the IMU measurements on the position, attitude, and velocity are treated as independent, which is an approximation. In our work, the full correlation structure between these constraints is properly accounted for (cf. (20)), resulting in improved estimation accuracy.

Algorithm 1 MSC-KF

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 a feature track is lost, perform an EKF update (cf. Section 3.4).

3. First Layer: MSC-KF

In this section, we describe the Multi-State constraint-Kalman filter (MSC-KF), which is used in the first layer of the localization system (cf. Fig. 1). As mentioned in the Introduction, the purpose of this layer is to efficiently and accurately track the camera motion, using the visual and inertial measurements. The MSC-KF is chosen for this task, because it has computational complexity only linear in the number of local features (thus attaining real-time performance), and can utilize the camera-motion constraints due to the feature measurements in a statistically optimal fashion. Specifically, the filter's design is motivated by the observation that, when a static feature is viewed from multiple camera poses, its measurements can be used to define *constraints* involving all these poses. The MSC-KF employs a measurement model that expresses these constraints *without* including the 3D feature position in the filter state vector, as explained in Section 3.4.

An overview of the MSC-KF algorithm is given in Algorithm 1. We consider a system consisting of an IMU and a camera, in which the transformation between the two is known and constant. The MSC-KF tracks the 3D pose of the IMU-affixed frame $\{I\}$ with respect to a *global frame* of reference $\{G\}$. In our work, $\{G\}$ is chosen as an Earth-Centered, Earth-Fixed (ECEF) frame, which allows us to easily account for the effects of the earth's rotation on the IMU measurements (cf. Eqs. (5)-(6)). In the MSC-KF, the IMU measurements are processed immediately as they be-

come 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 allows us to create a state vector comprising a sliding window of the N latest camera poses. During EKF updates, the measurements of each tracked feature are used for imposing constraints between these poses. In the following, we describe the various components of the MSC-KF algorithm (for a more detailed description, the interested reader is referred to [9]).

3.1. Structure of the EKF state vector

At any time instant, the MSC-KF state vector comprises (i) the evolving IMU state, $\mathbf{X}_{\text{IMU}}$, and (ii) a sliding window of N past camera poses. The IMU state vector is:

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

where ${}^I_G\bar{q}$ is the unit quaternion 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 (1), the IMU error-state is defined as[1]:

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

where $\boldsymbol{\delta\theta}_I$ is the 3×1 IMU attitude-error vector, defined by:

$${}^I_G\bar{q} = \delta\bar{q}\otimes {}^I_G\hat{\bar{q}}, \text{ where } \delta\bar{q} \simeq \begin{bmatrix} \frac{1}{2}\boldsymbol{\delta\theta}_I^T & 1 \end{bmatrix}^T \quad (3)$$

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:

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

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.

[1] Throughout this paper $\widehat{x}$ denotes the estimate of a quantity x, and $\widetilde{x}$ denotes the error in this estimate, defined as $\widetilde{x} = x - \widehat{x}$. Moreover, $\mathbf{I}_N$ denotes the $N\times N$ identity matrix, $\mathbf{C}(\bar{q})$ denotes the rotation matrix corresponding to a quaternion $\bar{q}$, the symbol $\otimes$ denotes quaternion multiplication, and $\lfloor x\times\rfloor$ denotes the skew symmetric matrix corresponding to the 3×1 vector x. Finally, the preceding superscript for a quantity x, e.g., ${}^A x$, denotes the frame of reference in which the quantity is expressed.

3.2. IMU Propagation

Propagation of the IMU state is carried out by numerical integration of the continuous-time IMU system model. The gyroscope and accelerometer measurements, $\boldsymbol{\omega}_m$ and $\mathbf{a}_m$ respectively, are given by [2]:

$$\boldsymbol{\omega}_m = {}^I\boldsymbol{\omega} + \mathbf{C}({}^I_G\bar{q})\,{}^G\boldsymbol{\omega}_e + \mathbf{b}_g + \mathbf{n}_g \quad (5)$$

$$\mathbf{a}_m = \mathbf{C}({}^I_G\bar{q})({}^G\mathbf{a} - {}^G\mathbf{g} + 2\lfloor {}^G\boldsymbol{\omega}_e\times\rfloor {}^G\mathbf{v}_I + \lfloor {}^G\boldsymbol{\omega}_e\times\rfloor^2\,{}^G\mathbf{p}_I) + \mathbf{b}_a + \mathbf{n}_a \quad (6)$$

where ${}^I\boldsymbol{\omega}$ is the IMU rotational velocity, ${}^G\mathbf{a}$ is the IMU body acceleration, ${}^G\mathbf{g}$ and ${}^G\boldsymbol{\omega}_e$ are the gravitational acceleration and the earth rotation vector respectively, and finally $\mathbf{n}_g$ and $\mathbf{n}_a$ are zero-mean, white Gaussian measurement noise processes. Given the IMU measurements at time-steps t_k and $t_{k+1} = t_k + T$, propagation of the IMU state estimate is carried out by 5-th order Runge-Kutta integration of the continuous-time IMU system model [9]:

$$\dot{\hat{\mathbf{X}}}_{\text{IMU}} = f(\hat{\mathbf{X}}_{\text{IMU}}, \boldsymbol{\omega}_m, \mathbf{a}_m) \quad (7)$$

in the time interval $[t_k, t_{k+1}]$. Moreover, the covariance matrix of the MSC-KF has to be propagated. For this purpose, we introduce the following partitioning for the covariance:

$$\mathbf{P}_{k|k} = \begin{bmatrix} \mathbf{P}_{II_{k|k}} & \mathbf{P}_{IC_{k|k}} \\ \mathbf{P}_{IC_{k|k}}^T & \mathbf{P}_{CC_{k|k}} \end{bmatrix} \quad (8)$$

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:

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

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

$$\dot{\mathbf{P}}_{II} = \mathbf{F}\mathbf{P}_{II} + \mathbf{P}_{II}\mathbf{F}^T + \mathbf{G}\mathbf{Q}_{\text{IMU}}\mathbf{G}^T \quad (9)$$

In this equation, $\mathbf{F}$ and $\mathbf{G}$ are the Jacobians of the system model with respect to the IMU error state and the process noise, respectively, and $\mathbf{Q}_{\text{IMU}}$ is the covariance matrix of the process noise. Numerical integration is carried out for the time interval $[t_k, t_{k+1}]$, with initial condition $\mathbf{P}_{II_{k|k}}$. The state transition matrix $\boldsymbol{\Phi}(t_{k+1}, t_k)$ is similarly computed by numerical integration of the differential equation

$$\dot{\boldsymbol{\Phi}}(t_k + \tau, t_k) = \mathbf{F}\boldsymbol{\Phi}(t_k + \tau, t_k), \quad \tau\in[0, T] \quad (10)$$

with initial condition $\boldsymbol{\Phi}(t_k, t_k) = \mathbf{I}_{15}$. We point out that the computational complexity of IMU propagation is linear in

the number of camera poses in the MSC-KF state vector, N, and is therefore extremely efficient, with each propagation requiring only a few microseconds of processing time.

3.3. State Augmentation

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

$$ {}^C_G\hat{\bar{q}} = {}^C_I\bar{q} \otimes {}^I_G\hat{\bar{q}}, \quad \text{and} \quad {}^G\hat{\mathbf{p}}_C = {}^G\hat{\mathbf{p}}_I + \mathbf{C}^T({}^I_G\hat{\bar{q}})\,{}^I\mathbf{p}_C \quad (11) $$

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 [9].

3.4. MSC-KF Measurement Model

We now present the measurement model employed for updates in the MSC-KF. Since the *Extended* form of the Kalman filter is used, 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:

$$ \mathbf{r} = \mathbf{H}\widetilde{\mathbf{X}} + \text{noise} \quad (12) $$

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.

For simplicity, we consider the case of a *single* feature, f_j, that has been observed from the N camera poses, $\Pi_i = \{{}^{C_i}_G\bar{q}, {}^G\mathbf{p}_{C_i}\}$, in the MSC-KF state vector. Each of the N observations of the feature is described by the perspective (nonlinear) measurement model:

$$ \mathbf{z}_i^{(j)} = h({}^G\mathbf{p}_{f_j}, \Pi_i) + \mathbf{n}_i^{(j)}, \quad i = 1 \dots N \quad (13) $$

where $\mathbf{n}_i^{(j)}$ is the 2×1 image noise vector, with covariance matrix $\mathbf{R}_i^{(j)} = \sigma_{\text{im}}^2\mathbf{I}_2$. Since the feature position ${}^G\mathbf{p}_{f_j}$ is unknown, in the first step of the MSC-KF algorithm we employ least-squares minimization (intersection) to obtain an estimate, ${}^G\hat{\mathbf{p}}_{f_j}$, of the feature position [9]. Once this estimate is obtained, we compute the measurement residual:

$$ \mathbf{r}_i^{(j)} = \mathbf{z}_i^{(j)} - \hat{\mathbf{z}}_i^{(j)} = \mathbf{z}_i^{(j)} - h({}^G\hat{\mathbf{p}}_{f_j}, \hat{\Pi}_i) \quad (14) $$

Linearizing about the estimates of the camera pose and the feature position, the residual of (14) is approximated as:

$$ \mathbf{r}_i^{(j)} \simeq \mathbf{H}_{\mathbf{X}_i}^{(j)}\widetilde{\mathbf{X}} + \mathbf{H}_{f_i}^{(j)}{}^G\widetilde{\mathbf{p}}_{f_j} + \mathbf{n}_i^{(j)} \quad (15) $$

where $\mathbf{H}_{\mathbf{X}_i}^{(j)}$ 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. By stacking the residuals of all N measurements of this feature, we obtain:

$$ \mathbf{r}^{(j)} \simeq \mathbf{H}_{\mathbf{X}}^{(j)}\widetilde{\mathbf{X}} + \mathbf{H}_f^{(j)}{}^G\widetilde{\mathbf{p}}_{f_j} + \mathbf{n}^{(j)} \quad (16) $$

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 $i = 1 \dots N$. Since the feature observations in different images are independent, the covariance matrix of $\mathbf{n}^{(j)}$ is $\mathbf{R}^{(j)} = \sigma_{\text{im}}^2\mathbf{I}_{2N}$.

Recall at this point that the state estimate, $\hat{\mathbf{X}}$, was used to compute the feature position estimate. Therefore, the error ${}^G\widetilde{\mathbf{p}}_{f_j}$ in (16) is correlated with the errors $\widetilde{\mathbf{X}}$. Consequently the residual $\mathbf{r}^{(j)}$ is *not* in the form of Eq. (12), and cannot be directly applied for 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 $\mathbf{U}$ denote the unitary matrix whose columns form the basis of the left nullspace of $\mathbf{H}_f$, we obtain:

$$ \mathbf{r}_o^{(j)} = \mathbf{U}^T(\mathbf{z}^{(j)} - \hat{\mathbf{z}}^{(j)}) \simeq \mathbf{U}^T\mathbf{H}_{\mathbf{X}}^{(j)}\widetilde{\mathbf{X}} + \mathbf{U}^T\mathbf{n}^{(j)} \quad (17) $$

$$ = \mathbf{H}_o^{(j)}\widetilde{\mathbf{X}}^{(j)} + \mathbf{n}_o^{(j)} \quad (18) $$

For computational efficiency, this projection is carried out in $O(N^2)$ operations using Givens rotations [5], and without explicitly forming $\mathbf{U}$. Since the $2N \times 3$ matrix $\mathbf{H}_f^{(j)}$ has full column rank, its left nullspace is of dimension $2N - 3$. Therefore, $\mathbf{r}_o^{(j)}$ is a $(2N - 3) \times 1$ vector. This residual is *independent* of the errors in the feature coordinates, and thus EKF updates can be performed based on it. Eq. (18) 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 N states, and thus the resulting EKF update is optimal, except for the inaccuracies caused by linearization.

It is important to note that the residual defined in (17) is not the only possible expression of the geometric constraints that are induced by observing a static feature in N images. An alternative approach would be, for example, to employ the epipolar constraints that are defined between $2N - 3$ pairs of the images, or to use the multi-linear constraints defined by the N measurements directly [6]. However, the resulting constraints are highly nonlinear, and moreover, they are not statistically independent, since each measurement is used in defining multiple constraints. Our experiments have shown that employing linearization of these constraints yields inferior results compared to the approach described above.

3.5. Outlier Rejection

Prior to using each feature's measurements for updates, an outlier rejection test is performed. Specifically, for each feature the Mahalanobis distance:

$$ d = \mathbf{r}_o^{(j)T}\left(\mathbf{H}_o^{(j)}\mathbf{P}\mathbf{H}_o^{(j)T} + \sigma_{\text{im}}^2\mathbf{I}_{2N-3}\right)^{-1}\mathbf{r}_o^{(j)} \quad (19) $$

is computed, and compared against the 95-th percentile of the χ^2 cumulative distribution function with $2N-3$ degrees of freedom. If d is smaller than this threshold, the feature is accepted as an inlier, and used in the updates.

Note that, in contrast to outlier rejection based on vision alone, in this outlier rejection scheme the MSC-KF state estimate is used as a prior, to help identify outliers. Additionally, it is important to observe that *all* the measurements of the feature are simultaneously used for the rejection test. As a result, features that correspond to slowly-moving objects, or whose tracking is unreliable, can be more easily detected and discarded. These properties arise from the tight coupling of the visual and inertial measurements, implemented by the MSC-KF. Finally, we note that in our dual-layer localization architecture, outlier rejection is carried out in the first layer by the MSC-KF, where the covariance matrix is directly available. Thus, the cost of computing the covariance matrix for outlier rejection in BA can be avoided, resulting in computational savings.

3.6. MSC-KF computational complexity

The computational complexity of applying the measurement model described in Section 3.4 is quadratic in N for each feature, and is dominated by the cost of the projection operation. If at time-step t_k we use M_k features for updates, the total computational cost of applying the measurement model is $O(M_k N^2)$. Moreover, in [9] it is shown that once all residual vectors $\mathbf{r}_o^{(j)}$ have been computed, the update can also be carried out at computational cost $\max(O(M_k N^2), O(N^3))$. What is important here is that the cost is *linear* in the number of features, which is typically much larger than N (typically a few tens of states are kept in the MSC-KF sliding window, while hundreds of features are processed at each time step). This property enables the MSC-KF to operate in real-time, while processing all the available feature measurements. Because the measurement model described in Section 3.4 is optimal up to linearization, all the motion constraints provided by the feature tracks are utilized. As a result, the MSC-KF provides combined visual/inertial odometry of high accuracy.

4. Second Layer: Closing Loops

The MSC-KF, presented in the preceding section, only processes *local* motion information, in the form of IMU measurements and features tracked in consecutive images. The localization information available when the camera revisits an area, is not utilized. For this reason, we employ a second layer of estimation (cf. Fig. 1), whose main purpose is to detect loop closures, and use the corresponding measurements for improving the state estimates. As discussed in Section 2, a simple approach for achieving this would be to include in the MSC-KF state vector a number of landmarks, similarly to SLAM, and to use the re-observations of these landmarks for improving the estimation accuracy. However, this approach has two limitations: First, even if we knew in advance which landmarks will be re-observed, their inclusion in the EKF state vector would require updating their position estimates (and the associated covariance matrix) *every* time a filter update takes place. This would incur a significant computational cost. Secondly, and most importantly, we typically *cannot* predict which landmarks will be re-observed in the future. As a result, we would need to maintain a large number of landmarks in the state vector, many of which would never be seen again.

For these reasons, we have opted for a different approach when processing loop-closure information. In particular, a separate module of our system uses the recorded images, as well as the history of camera-pose estimates, to detect when an area is re-visited (cf. Fig 1). Since the MSC-KF estimates are typically very accurate (e.g., errors less than 0.5% of the distance traveled) detecting *candidate* loop-closures along the trajectory can be performed very efficiently, based on a simple distance criterion. Once a candidate location is identified, only then images are processed to detect features observed during both visits. These feature re-observations are subsequently processed in a BA algorithm, along with the IMU measurements and the features that passed the Mahalanobis gating test in the MSC-KF[2]. The main benefit of this approach is that the processing is essentially trajectory-driven. The computational cost of loop closing is incurred *only* when loop closing occurs, which is typically an infrequent event.

In addition to using loop-closure information, the use of an iterative BA algorithm leads to improved linearization. Since the MSC-KF algorithm is an EKF-based estimator, it linearizes the measurements only once, and the gradual buildup of linearization errors can eventually lead to inconsistent estimates. To reduce the effect of linearization inaccuracies, BA can be run intermittently (or continuously, as a background process), even when no loop closure occurs, and its results can be used to "reset" the MSC-KF state and covariance estimates, and remove any accumulated linearization errors.

4.1. Bundle Adjustment

We now describe the formulation of a batch Maximum a Posteriori (MAP) estimator for processing the inertial and visual measurements. We consider the case in which K IMU measurements and K images are available, recorded at every time-step in the interval[3] $[t_1, t_K]$. The MAP estimate for all

[2] Although this architecture is independent of the type of visual features used, we note for completeness that in our implementation Harris corners are used when there is little change between images (e.g., tracking features for the MSC-KF), while SIFT keypoints are used for wide-baseline matching (e.g., loop closure).

[3] To simplify the presentation in this section, we assume that IMU measurements and images are concurrently recorded. In a real implementation, however, IMU measurements are most often available at a higher rate than images. This case is treated analogously, by performing multiple propaga-

IMU states and all feature positions can be determined by minimizing the cost function:

$$J = \left|\left|\mathbf{X}_{\text{IMU}_1} - \mathbf{Z}_1\right|\right|_{\mathbf{R}_{\text{prior}}} + \sum_{\ell,j} \left|\left|\mathbf{z}_\ell^{(j)} - h({}^G\mathbf{p}_{f_j}, \Pi_\ell)\right|\right|_{\mathbf{R}_\ell^{(j)}} + \sum_{\ell=1}^{K-1} \left|\left|\mathbf{X}_{\text{IMU}_{\ell+1}} - \phi(\mathbf{X}_{\text{IMU}_\ell}, \boldsymbol{\omega}_m, \mathbf{a}_m)\right|\right|_{\mathbf{Q}_\ell} \quad (20)$$

where $\left|\left|x\right|\right|_A$ denotes the matrix-weighted norm $x^T A^{-1} x$. The three terms in this cost function correspond to the following types of information that is available to the system:

- The first term in J expresses the prior information about the initial state of the IMU. Typically, we have an estimate for the pose and velocity of the IMU at the start of the system's operation, while for the IMU biases such prior information is obtained from the sensor specifications, or by sensor calibration. In (20) the prior estimate and its covariance are denoted by $\mathbf{Z}_1$ and $\mathbf{R}_{\text{prior}}$, respectively.
- The second term in (20) is the weighted squared error between the actual and predicted feature measurements, and expresses the constraints due to the visual observations. This term is the cost that is typically minimized by photogrammetric bundle-adjustment algorithms [16]. We note that the indices ℓ and j in this term assume appropriate values to index all the available feature measurements. This includes both the feature tracks provided by the MSC-KF, as well as the feature re-observations that are detected by the loop-closure module.
- The last term in (20) expresses the constraints due to the IMU process model. Each of the $K-1$ summands is the weighted difference between the estimated IMU state at time-step $t_{\ell+1}$, and the IMU state predicted using the inertial measurements. To compute this predicted state (denoted by $\phi(\mathbf{X}_{\text{IMU}_\ell}, \boldsymbol{\omega}_m, \mathbf{a}_m)$), we numerically integrate the IMU system model over the time interval $[t_\ell, t_{\ell+1}]$, starting from the estimate $\mathbf{X}_{\text{IMU}_\ell}$. The covariance matrix $\mathbf{Q}_\ell$, which expresses the uncertainty of the IMU-state-change estimate, is similarly computed by numerically integrating the Lyapunov equation (cf. (9)), starting from a zero initial value. The Jacobian of the term $\phi(\cdot)$, needed by the iterative minimization algorithm, is computed by numerically integrating (10), starting with the identity matrix as an initial value.

In order to minimize the cost function J with respect to all IMU states and all feature positions, we employ Gauss-Newton iterative minimization. Since the vast majority of the features observed are tracked in a small number of frames, in each iteration we utilize the technique of first marginalizing out all features, solving for the IMU states, and then back-substituting for the feature positions, similarly to [4]. This leads to a sparse structure for the Hessian matrix of the system, which we solve using sparse skyline Cholesky factorization [16]. Because the iterative minimization uses as an initial guess the MSC-KF output, which is typically very accurate, only a few iterations (usually 3-4) are required for convergence.

tion steps in the computation of *each* of the summands of the third term in (20).

4.2. Feedback to the First Layer

Once the minimization has converged, the IMU and camera state estimates contained in the current MSC-KF sliding window are fed back to the first layer. Moreover, the corresponding covariance matrix is computed and replaces the current MSC-KF covariance matrix. The computation of the covariance matrix can be sped up significantly, by taking into consideration the properties of Cholesky factorization. Specifically, from the Gauss-Newton iteration, the Cholesky factor of the Hessian matrix corresponding to the history of all IMU states is available:

$$\mathbf{A} = \mathbf{R}^T\mathbf{R} \Rightarrow \begin{bmatrix} \mathbf{A}_{oo} & \mathbf{A}_{oa} \\ \mathbf{A}_{oa}^T & \mathbf{A}_{aa} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{oo}^T & \mathbf{0} \\ \mathbf{R}_{oa}^T & \mathbf{R}_{aa}^T \end{bmatrix} \begin{bmatrix} \mathbf{R}_{oo} & \mathbf{R}_{oa} \\ \mathbf{0} & \mathbf{R}_{aa} \end{bmatrix}$$

where blocks denoted by the subscript "oo" correspond to older poses, blocks denoted by "aa" correspond to the poses that are currently active in the MSC-KF sliding window, and "ao" corresponds to the cross-terms between these. Employing the properties of the block-matrix inversion and substituting from the above expression, the covariance matrix of the active states is computed as:

$$\mathbf{P}_{aa} = \left(\mathbf{A}_{aa} - \mathbf{A}_{oa}^T\mathbf{A}_{oo}^{-1}\mathbf{A}_{oa}\right)^{-1} = \mathbf{R}_{aa}^{-1}\mathbf{R}_{aa}^{-T} \quad (21)$$

Since $\mathbf{R}_{aa}$ is already available, the cost of computing $\mathbf{P}_{aa}$ is simply that of inverting a triangular matrix and multiplying it with its transpose.

4.3. Marginalization of Old States

An important issue is that the computational cost of BA increases with the number of states in the estimated trajectory (due to sparsity, the increase is approximately linear in time). Thus, for very long trajectories the computational burden can become intractable. To address this problem, we can choose to permanently marginalize out certain older poses and the features seen from these poses. By limiting the number of estimated states, this process allows the processing time for BA to remain bounded. Clearly, after marginalization the linearization of the measurements that involve the removed states is not recomputed, and hence marginalization leads to an approximation of the cost function. However, if only older, "mature" states (i.e., states for which the estimates are deemed accurate) are removed, the approximation will be very good. Finally, we note that once a pose is marginalized, we no longer have the ability to close loops using this pose. Therefore, care should be taken in order to always maintain a set of poses in areas that are likely to be revisited by the vehicle.

5. Experimental results

The presented localization system has been applied for estimating the trajectory of a vehicle moving in an urban environment. The experimental setup comprised a Pointgrey FireFly camera, registering images with resolution 640×480 pixels, and an Inertial Science ISIS IMU, providing inertial measurements at 100Hz. During the experiments all data were stored on a computer and processed off-line. The runtimes reported in the following exclude feature extraction and tracking. We hereafter present results from two experiments to demonstrate the system's performance.

Experiment 1: In this experiment the vehicle drove for about 16 minutes, covering a distance of approximately 7.6 km. Images were processed at a rate of 7.5 Hz, and an average of 800 features were tracked in each image. These feature measurements were processed by the MSC-KF, in which a sliding window of 30 camera poses was maintained. In order to demonstrate the localization accuracy attainable by the tightly-coupled visual/inertial odometry when used without feature re-detection, in this dataset loop-closing was *not* applied. The BA module is run every 500 images, for reducing the buildup of linearization errors. To limit the computational burden of the iterative minimization process, permanent marginalization of older poses is applied in the BA, so as to keep the maximum number of actively optimized camera states to 1000.

With these settings, the MSC-KF requires approximately 100 msec of processing time per image, while BA requires approximately 2.2 sec per iteration (the algorithms run on a 2GHz processor). Since BA runs every 500 images, and typically requires 3 iterations, this implies that the additional cost of carrying out the batch optimization is about 6.6 sec for every 50 sec of processing time taken up by the MSC-KF, a mere 13% overhead. This small additional cost demonstrates the benefits of using the dual-layer localization architecture: the BA guarantees that the state estimates are very close to the globally optimal MAP result (a small difference is expected due to the marginalization), while the overhead over the simple local processing of the MSC-KF is minimal.

In Fig. 2, the estimated trajectory is shown, and compared to the GPS ground truth. The estimate is plotted in white (a red square indicates the starting position), while the GPS measurements are denoted by blue dots. Both are superimposed on a satellite image of the area where the experiment took place. Comparison of the estimated trajectory with the GPS measurements shows that the position error remains below 30 m throughout the trajectory. For a trajectory of length 7.6 km, this corresponds to an error of less than 0.4% of the distance traveled. Note also that this level of accuracy is achieved *without* utilizing any additional localization information (e.g., knowledge of the map, vehicle wheel odometry, or kinematic model of the car) and by processing images at a relatively low rate (7.5 Hz). This further demonstrates the benefits of fusing visual with inertial measurements.

Experiment 2: In this case, the car covered a distance of 3.2 km within 9 minutes and five loop closures were detected. Due to hardware limitations, in this experiment images could only be recorded at 3 Hz. Since the vehicle often revisited the same areas, a number of loop-closure events were considered. In our implementation, candidate loop-closure sections of the trajectory are identified based on two criteria: (i) spatial closeness of the trajectory, and (ii) motion in approximately the same direction. To limit the search space, candidate loop closures are sought only in portions of the trajectory where the vehicle either stops or turns. Once the candidate loop-closure locations are identified, three equally spaced images are chosen in each of the matching trajectory segments. In these images, SIFT keypoints are detected and matched [8]. If keypoints are reliably matched in all the images (both within each segment and across the current and previous segments), they are then passed for processing to the BA algorithm.

The estimated trajectory is shown in Fig 3, superimposed on a satellite image of the area where the experiment took place. Even though GPS ground truth was not available, the quality of the estimated trajectory can be evaluated based on how closely it follows the road pattern. Additionally, the final position of the vehicle with respect to its starting point was known, and this allowed us to compute the final position error: 6.3 m in the ground plane, and 0.8 m in altitude. For a trajectory of 3.2 km in length, this corresponds to less than 0.2% of the distance traveled. When no loop-closing features were used, the final error magnitude was approximately 10 m [9], which shows the benefits of utilizing loop-closure information.

In terms of processing requirements, in this dataset the MSC-KF requires approximately 34 msec of processing time per image (this is less than in the previous case, because fewer features are tracked, and for fewer frames on average), and processes the entire dataset of 1596 images in approximately 54 sec. For this dataset no marginalization of older poses is employed, and BA requires approximately 2.4 sec per iteration, when processing all the 1596 states simultaneously. BA is once again run every 500 states, requiring a total processing time of 15.4 sec for the entire run. Thus, the processing overhead is 28%, compared to using the MSC-KF alone. We deem this overhead to be small, considering that, by using the entire two-layer architecture instead of the first layer alone, we obtain the benefits of (i) using loop closure information, and (ii) obtaining the globally optimal MAP estimate for the trajectory.

6. Conclusions

In this paper, we have presented a localization system that tightly integrates measurements from a camera and an inertial measurement unit. The system follows a two-layer architecture, in which the first layer carries out combined visual/inertial odometry in real time, while the second layer

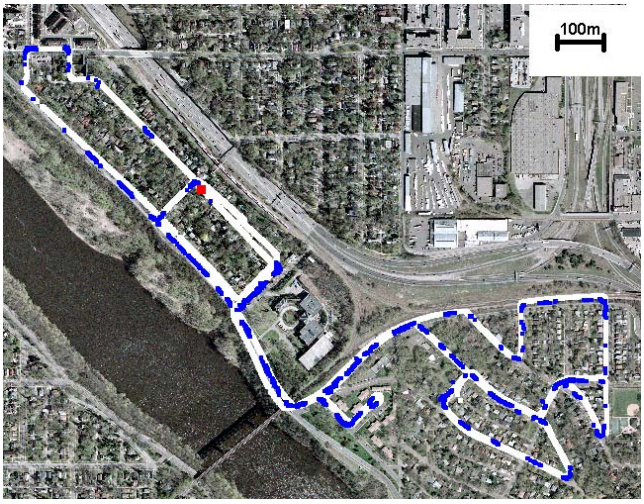

Figure 2. The estimated trajectory overlaid on a satellite image of the area where the experiment took place. The initial position of the vehicle is denoted by a red square. The blue dots represent the GPS measurements, which are often unavailable, due to the dense foliage.

Figure 3. The estimated trajectory overlaid on a satellite image of the area where the experiment took place. The initial position of the vehicle is denoted by a red square.

intermittently employs a batch nonlinear minimization algorithm (bundle adjustment) for imposing loop-closure constraints. As key advantages of the proposed two-layer architecture we can identify: (i) The estimates are available in real time, and at the IMU data rate, (ii) The estimation errors remain bounded when loops are detected, thus enabling long-term localization, and (iii) The computational overhead of utilizing loop-closure constraints is minimal, even though loop closure significantly improves localization accuracy. These properties render the proposed architecture suitable for large-scale localization applications. As a final remark, we note that if additional measurements (e.g., GPS) are available to the system, these can be readily used to improve the localization accuracy, without any modification of the system's architecture.

References

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

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

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

[4] C. Engels, H. Stewenius, and D. Nister. Bundle adjustment rules. In *Proceedings of the Photogrammetric Computer Vision Conference*, pages 266–271, Bonn, Germany, September 20-22 2006.

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

[6] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[7] K. Konolige, M. Agrawal, and J. Sola. Large-scale visual odometry for rough terrain. In *Proceedings of the International Symposium on Research in Robotics*, Hiroshima, Japan, November 26-29 2007.

[8] D. G. Lowe. Distinctive image features from scale-ivnariant keypoints. *International Journal of Computer Vision*, 60(2):91–100, 2004.

[9] A. I. Mourikis and S. I. Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3565–3572, Rome, Italy, April 2007.

[10] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1):3–20, January 2006.

[11] L. L. Ong, M. Ridley, J. H. Kim, E. Nettleton, and S. Sukkarieh. Six DoF decentralised SLAM. In *Proceedings of the Australasian Conference on Robotics and Automation*, pages 10–16, Brisbane, Australia, December 2003.

[12] T. Oskiper, Z. Zhiwei, S. Samarasekera, and R. Kumar. Visual odometry system using multiple stereo cameras and inertial measurement unit. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, Minneapolis, MN, June 17-22 2007.

[13] P. Pinies, T. Lupton, S. Sukkarieh, and J. Tardos. Inertial aiding of inverse depth SLAM using a monocular camera. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2797–2802, Rome, Italy, April 2007.

[14] S. I. Roumeliotis, A. E. Johnson, and J. F. Montgomery. Augmenting inertial navigation with image-based motion estimation. In *IEEE International Conference on Robotics and Automation*, pages 4326–33, Washington D.C., 2002.

[15] D. Strelow. *Motion estimation from image and inertial measurements*. PhD thesis, Carnegie Mellon University, 2004.

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