Exploiting Ground Plane Constraints for Visual-Inertial Navigation

Ghazaleh Panahandeh, Dave Zachariah, and Magnus Jansson
ACCESS Linnaeus Centre
KTH Royal Institute of Technology
SE-100 44 Stockholm, Sweden
{ghpa, dave.zachariah, magnus.jansson}@ee.kth.se

Abstract—In this paper, an ego-motion estimation approach is introduced that fuses visual and inertial information, using a monocular camera and an inertial measurement unit. The system maintains a set of feature points that are observed on the ground plane. Based on matched feature points between the current and previous images, a novel measurement model is introduced that imposes visual constraints on the inertial navigation system to perform 6 DoF motion estimation. Furthermore, feature points are used to impose epipolar constraints on the estimated motion between current and past images. Pose estimation is formulated implicitly in a state-space framework and is performed by a Sigma-Point Kalman filter. The presented experiments, conducted in an indoor scenario with real data, indicate the ability of the proposed method to perform accurate 6 DoF pose estimation.

Index Terms—Ego-motion estimation, vision-aided INS, ground plane feature detection, epipolar geometry.

I. INTRODUCTION

Navigation based on signals from an inertial measurement unit (IMU) has the advantage of being self-contained and robust with respect to external disturbances. With the development of MEMS technology, it is possible to implement the navigation system at low cost. The drawback of dead-reckoning based navigation, however, is that the position errors accumulate over time. The growth rate of the error can be significantly reduced if the system is aided with additional sensor information. One of the most common alternative sensors is a radio receiver, such as a global positioning system (GPS) receiver.

Integration of vision and inertial sensors is an alternative to GPS for indoor as well as outdoor navigation when the GPS signals are not available or reliable. By proper incorporation of visual information, the error growth of the inertial navigation system (INS) can be reduced [1]. However, aiding the INS with a vision sensor requires sensor-to-sensor relative transformation to be known; disregarding such an offset in the system will introduce unmodeled biases that may grow over time [2], [3]. Additionally, in vision-based pose analysis, a central challenge is that information about metric distances is lost in the camera projection that transforms the position of 3D points onto the 2D image plane [4].

In order to get more accurate and stable pose estimates, finding a proper solution for merging the IMU information with the camera image data is crucial. Depending on the image processing tools and computer vision methods used to extract the pose information from the camera images, different solutions are proposed for such a framework.

The structure of this paper is as follows. A summary of the related works is given in Section I-A. In Section I-B, we present an overview of our proposed method. Notations are introduced in Section I-C. In Section II, we provide the sensors models and describe the INS and the camera states. In Section III, the fundamental geometrical constraints of the visual inputs are derived. Measurement model of the system, which leads to the state-space equations, is given in Section IV. The structure of the used Sigma-Point Kalman filter algorithm is presented in Section V. Feature detection and outlier rejection are discussed in Section VI. Results are demonstrated in Section VII. Finally, the conclusion of the study is summarized in Section VIII.

A. Related work

Fusing inertial and vision sensors has been explored in different applications, such as vision-aided navigation systems [5], [6], mobile robotics [7], [8], and traffic applications [9]. Broadly speaking, vision-aided inertial navigation can be divided into two approaches: vision-based SLAM [10], [11] and ego-motion estimation [6], [12], [13].

In vision-based SLAM, a 3D map of observed feature points in the scene is progressively estimated jointly with the current camera pose. Using the estimated map, these methods are capable of loop-closing that cancels drift of the position estimate when returning to an already-visited location. In ego-motion estimation methods, the main goal is to recover the camera pose rather than the positions of the feature points. Hence, the computational complexity of ego-motion estimation approaches is significantly lower than that of the SLAM-based methods.

Two commonly used image processing techniques, for extracting pose information from the images, are based on feature point extraction [13] and optical flow computation [14]. Feature point extraction is usually followed by point matching between the consecutive images to obtain displacement measurements that are subsequently fused with inertial measurements. While in the optical flow based methods, changes in the images are treated as the projection of the scene velocity. As neither technique can recover the metric scale, further constraints must be imposed for motion estimation. Stereo
cameras can resolve the scale ambiguity problem by providing depth estimates of the scene [12], [15], [16]. But the accuracy of such methods is highly dependent on the relative baseline between the cameras.

More information may be exploited when navigating in structured environments, e.g., man-made buildings. In the literature, various structures have been investigated for vision-based motion estimation, e.g., horizontal or vertical lines, vanishing points and lines [4], edges of buildings or fixed objects, and artificial visual tags [17].

A few approaches make use of features located on the ground plane, which is a salient structure in these environments. For instance, Song et al. [18] proposed an algorithm for estimating the velocity of mobile robots based on the Kalman filter integration using an optical flow method for a downward-looking camera. A more flexible system for pedestrian navigation is introduced in [19] in which a camera is pointed towards the ground. Using a computer vision algorithm, 3D camera frame translation is derived that provides the velocity of the camera in the measurement model. The hardware structure of their system is, however, aided with GPS data in addition to the IMU and camera sensors. Moreover, in our previous work [13], we introduced an IMU-camera ego-motion method based on the assumption that the camera optical axis is roughly orthogonal to the ground plane. Then, in contrast to the existing approaches, the scale ambiguity problem is removed and the system is capable of 5 DoF motion estimation.

However, such a configuration is restricted to a downward-looking camera which further limits the extraction of other structural information from the scene. By observing the surrounding environment and the ground plane one can impose more constraints on the camera relative position and orientation over time. Since the feature points located on the ground plane are coplanar, information about their relative position can be recovered given a pose estimate of the navigation system. Additionally, selecting ground plane features for motion estimation avoids regions of the image that may contain moving objects like human beings or moving vehicles.

B. System overview

In this paper, an IMU-camera sensor fusion system is introduced for a scenario that may find applications in mobile robot navigation, as well as personal navigation. The system performs 6 DoF ego-motion estimation by maintaining a set of observed ground feature points of a previous image frame. In contrast to other methods that use cameras with an optical axis approximately orthogonal to the ground plane, such as [13], [18], [19], no specific restrictions are imposed on the camera motion other than that the camera is continuously observing features on the plane.

When the camera optical axis is fixed to be perpendicular to the ground plane, the feature points located on the plane are at a constant orthogonal distance to the image plane. The assumption of using a camera with orthogonal optical axis to the ground plane is relaxed by virtually transforming the image plane as if it was captured by a virtual camera perpendicular to the ground plane [20]. Using the virtual camera concept, we derive a novel measurement model for ground plane observations. Additionally, by exploiting epipolar geometry between views, we can further constrain the direction of the translation vector [4], [5], [21]. Such constraints, however, cannot reduce the error growth rate along the direction of motion; hence, the observed ground plane features act as complementary constraints.

The 6 DoF ego-motion estimation problem is posed in a state-space framework and solved by a state-augmented Sigma-Point Kalman filter [22]. The maximum number of augmented feature points is a parameter that controls the size of the state vector. At each time instant when a new frame is processed, states representing past feature points are subsequently discarded and replaced by new ones. To obtain feature points on the ground plane they must first be detected. Existing detection methods are either based on ground plane textures, e.g., color cue segmentations [23], geometric assumptions on the environment [23], [24], or regions of the image plane that reliably contain the ground plane [15]. These methods either assume constrained camera motion or that the planar homography is constructed only after a manual initialization phase. Similar to [15], in this paper, the ground plane features are selected using a predefined box in the lower part of the image while the outliers are rejected using the filter’s estimates over time.

C. Notation

In the following sections scalars are denoted by lowercase letters (s), vectors by bold letters (f), and matrices by bold capitals (K).

A ⊕ B denotes the direct sum between matrices A and B. The matrix I_m denotes the m × m square identity matrix.
The coordinate frames are denoted {a} so that p^a is the position in that frame and R^b_α is the direction-cosine matrix that rotates a vector from {a} to {b}. The navigation, IMU, and the current camera frames are denoted, {n}, {b}, and {c}, respectively. {c_ℓ} is the coordinate frame of the camera capturing an image ℓ lags from the current one. Then p^n_c_ℓ is the position of the camera in {n} at that time instant.

II. INS AND CAMERA STATES

Let’s consider an IMU-camera sensor fusion system in which the camera is rigidly mounted on the IMU. The goal of the proposed algorithm is to estimate the pose of this mobile visual-inertial system with respect to the reference navigation frame {n}.

The INS estimates the position, p^n_k, velocity, v^n_k, and attitude, R^n_k, at a high rate given noisy accelerometer signals f^n and gyroscope signals ω^n. Let dt denote the sampling period, then the estimates are propagated by the following equations:

\[
\begin{align*}
\hat{p}^{n}_{k+1} &= \hat{p}^{n}_{k} + dt\hat{v}^{n}_{k} \\
\hat{v}^{n}_{k+1} &= \hat{v}^{n}_{k} + dtR^{n}_{b,n}\hat{f}^{b}_{k} + dt\hat{g}^{n} \\
R^{n}_{b,k+1} &= R^{n}_{b,k} exp \left( dt[\omega^{b}]x \right),
\end{align*}
\]

(1)
for each IMU sample \( k \). Here, \( \mathbf{g}^n \) is the local gravitation vector and \( \cdot \) denotes the skew-symmetric matrix representation of the cross product operation [25]. For the matrix exponential, \( \exp(\cdot) \), the Padé approximation can be used [26].

The position of the camera is related to the navigation states by the transformation \( \mathbf{p}^c_k = \mathbf{p}^b_k + \mathbf{R}^b_k \mathbf{p}^c_k \), where \( \mathbf{p}^b_k \) denotes the offset between the IMU and camera coordinate centers in \( \{ b \} \) and is assumed to be known by calibration [2], [3].

As depicted in Fig. 1, we approach the estimation problem using a feedback approach. Given measurements from the camera (Visual input), \( y^c \), the errors of the INS errors are estimated and corrected by a Sigma-Point Kalman filter (SPKF). Hence, to estimate parameters in the SPKF, the total system state vector is described as

\[
x^\text{ins} \triangleq [(p^b_k)^\top \ (v^b_k)^\top \ (q^b_k)^\top \ (f^b_k)^\top \ (\omega^b_k)^\top ]^\top ,
\]

where \( q^b_k \) is the unit quaternion representation of the rotation matrix \( \mathbf{R}^b_k \), and the IMU sensor biases are represented by \( f^b_k \) and \( \omega^b_k \).

A. INS error states

Let \( \delta \mathbf{p}^b_k \) and \( \delta \mathbf{v}^b_k \) denote the position and velocity errors of the INS, respectively. Parameterizing the quaternion error by Euler angles, we denote the attitude error by \( \delta \mathbf{\Theta} \in \mathbb{R}^3 \). Finally, the IMU sensor error biases are represented by \( \delta f^b_k \) and \( \delta \omega^b_k \).

The errors can be concatenated into an error state vector,

\[
\delta x^\text{ins} \triangleq [(\delta \mathbf{p}^b_k)^\top \ (\delta \mathbf{v}^b_k)^\top \ (\delta \mathbf{\Theta})^\top \ (\delta f^b_k)^\top \ (\delta \omega^b_k)^\top ]^\top.
\]

During a short period of time the errors can be modeled by a discrete-time linear state-space model

\[
\delta x_{k+1}^\text{ins} = \mathbf{F}^\text{ins}_k \delta x_k^\text{ins} + \mathbf{G}^\text{ins}_k \mathbf{w}_k \in \mathbb{R}^{15},
\]

The statistical properties of the process noise, \( \mathbf{w}_k \), is given by the noise characteristics of the IMU. We model it as a temporally uncorrelated, zero-mean process with a diagonal covariance matrix \( \mathbf{Q} \in \mathbb{R}^{12 \times 12} \). The system matrices \( \mathbf{F}^\text{ins}_k \) and \( \mathbf{G}^\text{ins}_k \) are given in Appendix A.

B. Camera error states

For small angle errors, the camera position error equals

\[
\delta \mathbf{p}^c_{\ell} = \mathbf{T} \delta x^\text{ins}, \quad \text{where} \quad \mathbf{T} \triangleq \begin{bmatrix} 1 & 0 & \mathbf{R}^b_k \mathbf{p}^c_k \\ 0 & 1 \end{bmatrix}.
\]

Given a set of the \( L \) most recent images taken at different locations \( \{ \mathbf{p}^c_{\ell} \}_{\ell=1}^L \), the camera position errors are augmented into

\[
\delta \mathbf{x}^\text{cam} = [(\delta \mathbf{p}^c_{\ell messagebox}^\top \cdots (\delta \mathbf{p}^c_{\ell messagebox})^\top ]^\top.
\]

III. GEOMETRY OF PROJECTED FEATURES

In this section, the essential geometrical constraints from the visual input are given. The obtained relations are then used in Section IV to construct the state-space model of the system. Consequently, this model is also used to formulate a SPKF that estimates the INS errors in Section V.

If \( \mathbf{\pi} \in \mathbb{R}^3 \) is an observed feature point by a camera, its position relative to the camera center \( \{ c \} \) is denoted by \( \mathbf{\pi}^c \) and its normalized pixel coordinates and homogenous coordinates on the image plane are represented by \( \mathbf{z}^c \) and \( \mathbf{z}^\text{e} \), respectively. Using the pinhole camera model [4], in the following, we will make use of the relationship between a feature point and its projection on the image plane,

\[
\mathbf{z}^\text{e} = (\mathbf{e}_z^\top \mathbf{\pi}^c)^{-1} \mathbf{\pi}^c,
\]

and

\[
\mathbf{z}^c = [1 \ 0 \ 0] (\mathbf{e}_z^\top \mathbf{\pi}^c)^{-1} \mathbf{\pi}^c,
\]

where \( \mathbf{e}_z^\top = [0 \ 0 \ 1] \). Note that scale is lost in the projection of \( \mathbf{\pi}^c \).

A. Ground plane features

Consider an arbitrary feature point \( \mathbf{\pi} \) and its position coordinatized in camera frames \( \{ c \} \) and \( \{ c_{\ell messagebox} \} \), respectively, illustrated in Fig. 2. The general transformation of the point between frames can be written as,

\[
\mathbf{\pi}^c = \mathbf{R}^c_{\ell messagebox} (\mathbf{\pi}^{c\ell messagebox} - \mathbf{t}^{c\ell messagebox}),
\]

\( \cdot \) denotes the skew-symmetric matrix representation of the cross product operation [25]. For the matrix exponential, \( \exp(\cdot) \), the Padé approximation can be used [26].

The position of the camera is related to the navigation states by the transformation \( \mathbf{p}^c_k = \mathbf{p}^b_k + \mathbf{R}^b_k \mathbf{p}^c_k \), where \( \mathbf{p}^b_k \) denotes the offset between the IMU and camera coordinate centers in \( \{ b \} \) and is assumed to be known by calibration [2], [3].

As depicted in Fig. 1, we approach the estimation problem using a feedback approach. Given measurements from the camera (Visual input), \( y^c \), the errors of the INS errors are estimated and corrected by a Sigma-Point Kalman filter (SPKF). Hence, to estimate parameters in the SPKF, the total system state vector is described as

\[
x^\text{ins} \triangleq [(p^b_k)^\top \ (v^b_k)^\top \ (q^b_k)^\top \ (f^b_k)^\top \ (\omega^b_k)^\top ]^\top ,
\]

where \( q^b_k \) is the unit quaternion representation of the rotation matrix \( \mathbf{R}^b_k \), and the IMU sensor biases are represented by \( f^b_k \) and \( \omega^b_k \).

A. INS error states

Let \( \delta \mathbf{p}^b_k \) and \( \delta \mathbf{v}^b_k \) denote the position and velocity errors of the INS, respectively. Parameterizing the quaternion error by Euler angles, we denote the attitude error by \( \delta \mathbf{\Theta} \in \mathbb{R}^3 \). Finally, the IMU sensor error biases are represented by \( \delta f^b_k \) and \( \delta \omega^b_k \).

Then the errors can be concatenated into an error state vector,

\[
\delta x^\text{ins} \triangleq [(\delta \mathbf{p}^b_k)^\top \ (\delta \mathbf{v}^b_k)^\top \ (\delta \mathbf{\Theta})^\top \ (\delta f^b_k)^\top \ (\delta \omega^b_k)^\top ]^\top.
\]

During a short period of time the errors can be modeled by a discrete-time linear state-space model

\[
\delta x_{k+1}^\text{ins} = \mathbf{F}^\text{ins}_k \delta x_k^\text{ins} + \mathbf{G}^\text{ins}_k \mathbf{w}_k \in \mathbb{R}^{15},
\]

The statistical properties of the process noise, \( \mathbf{w}_k \), is given by the noise characteristics of the IMU. We model it as a temporally uncorrelated, zero-mean process with a diagonal covariance matrix \( \mathbf{Q} \in \mathbb{R}^{12 \times 12} \). The system matrices \( \mathbf{F}^\text{ins}_k \) and \( \mathbf{G}^\text{ins}_k \) are given in Appendix A.

B. Camera error states

For small angle errors, the camera position error equals

\[
\delta \mathbf{p}^c_{\ell messagebox} = \mathbf{T} \delta x^\text{ins}, \quad \text{where} \quad \mathbf{T} \triangleq \begin{bmatrix} 1 & 0 & \mathbf{R}^b_k \mathbf{p}^c_k \\ 0 & 1 \end{bmatrix}.
\]

Given a set of the \( L \) most recent images taken at different locations \( \{ \mathbf{p}^c_{\ell messagebox} \}_{\ell=1}^L \), the camera position errors are augmented into

\[
\delta \mathbf{x}^\text{cam} = [(\delta \mathbf{p}^c_{\ell messagebox})^\top \cdots (\delta \mathbf{p}^c_{\ell messagebox})^\top ]^\top.
\]
This enables us to relate an observation of \( \pi \) in two different views to the motion between the views, parameterized by the rotation matrix \( R_{c}^{c_{i}} \) and the translation vector \( t_{c}^{c_{i}} \). Thus the projection of \( \pi^{c} \) is correlated with the motion of the system and therefore imposes constraints on the accumulated INS errors \( (3) \).

The feature point observed in the past frame \( \pi^{c_{i}} \) is, however, an unknown nuisance parameter. Estimating the distance of \( \pi^{c_{i}} \) is a poorly conditioned problem when it is large relative to the translation between views. The problem can be circumvented by exploiting the fact that the point belongs to the ground plane. Then \( \pi^{c_{i}} \) can be expressed in terms of camera position \( p_{c_{i}}^{n} \) and observation \( z^{c_{i}} \) \( (13) \). In order to derive the alternative parameterization of \( \pi^{c_{i}} \), we first consider a ‘virtual’ camera coordinate frame \( \{ v \} \) collocated at the center of the camera coordinate frame \( \{ c \} \); see Fig. 2. The orientation of \( \{ v \} \) is aligned with \( \{ n \} \) and its optical axis is perpendicular to the ground plane. Then the distance to the point in \( \{ v \} \) along the optical axis, \( e_{z}^{c} \pi^{v} \), is equal to the height of the camera center above the plane, \( e_{z}^{c} p_{c_{i}}^{n} \).

As the transformation between the camera frame and the virtual frame is related by a simple rotation, using (6), we can reparameterize the unknown ground plane feature point’s position \( \pi^{c_{i}} \) by

\[
\pi^{c_{i}} = R_{v}^{c} \pi^{v_{i}} = R_{v}^{c} (e_{z}^{c} \pi^{v}) z^{v_{i}} = R_{v}^{c} (e_{z}^{c} p_{c_{i}}^{n}) (R_{v}^{n} z^{c_{i}} / e_{z}^{c} R_{v}^{c} z^{c_{i}}) = e_{z}^{c} p_{c_{i}}^{n} / e_{z}^{c} R_{v}^{c} z^{c_{i}},
\]

where \( p_{c_{i}}^{n} \) is the camera position, \( z^{c_{i}} \) is the observation in homogenous coordinates and \( R_{v}^{c} \) is the rotation matrix between frames. Using (9), the general transformation (8) can now be rewritten for the ground plane feature \( \pi \) as

\[
\pi^{c} = R_{v}^{c} (\pi^{c_{i}} - t_{c}^{c_{i}}) = R_{v}^{c} \pi^{c_{i}} - R_{v}^{c} R_{c_{i}}^{n} (p_{c_{i}}^{n} - p_{c}^{n}) = R_{v}^{c} \alpha \pi^{c_{i}} - R_{v}^{c} R_{c_{i}}^{n} (p_{c_{i}}^{n} - p_{c}^{n}),
\]

where \( \alpha = e_{z}^{c} p_{c_{i}}^{n} \) and \( \beta = e_{z}^{c} R_{c_{i}}^{n} z^{c_{i}} \). This eliminates the explicit dependence on the distance to the feature point, and expresses it in terms of the camera position \( p_{c_{i}}^{n} \) and past observation \( z^{c_{i}} \).

The rotation matrix \( R_{v}^{c} \) is factorized into \( R_{v}^{c} = R_{v}^{n} R_{v}^{a} R_{v}^{c_{i}} R_{a}^{c_{i}} \). An estimate of the relative rotation \( R_{v}^{c_{i}} \) is given by the gyroscoops provided that the duration between the frames is relatively short so that the integrated errors are small. The IMU-camera rotation \( R_{a}^{c_{i}} \) is fixed and assumed to be calibrated. The attitude \( R_{a}^{n} \) is a navigation state \( (2) \). The rotation matrix \( R_{a}^{c_{i}} = R_{a}^{c_{i}} = \text{diag}(1, -1, -1) \), since the orientation of the virtual camera frame is fixed relative to the navigation frame \( (20) \).

**B. Epipolar constraints**

Further relative constraints can be imposed between \( \{ c \} \) and \( \{ c_{i} \} \), in periods during which the matched ground plane features are few or nonexistent. All the matched points \( \{ z_{j}^{c}, z_{j}^{c_{i}} \} \) are transformed into parallel image planes by \( R_{c_{i}}^{c} \{ z_{j}^{c_{i}}, z_{j}^{c} \} \). The displacement of this pair of points provides information about the translation from \( p_{c_{i}}^{n} \) to \( p_{c}^{n} \); the transformed matches \( \{ z_{j}^{c_{i}}, z_{j}^{c} \} \) generate lines that ideally intersect at the epipolar point \( z_{\text{epi},\ell}^{c} \) on the image plane. This point is the projection of the translation vector \( \Delta p = p_{c_{i}}^{c} - p_{c}^{c} \) onto the image plane of \( \{ c \} \); details are given in \( (21) \).

The epipolar point can be observed as a noisy measurement. Each pair of generated lines \( j \) intersect at a point \( c_{j}\ell \). The epipolar point is then estimated as a convex combination of intersections using a weighted sum,

\[
z_{\text{epi},\ell}^{c} = \sum_{j} w_{j} c_{j}\ell
\]

where \( w_{j} \) is the weight of the intersection-sum.

**C. Feature point states**

In order to impose the geometric constraint (10) the feature points on the ground plane, \( z^{c_{i}} \), matched and observed in a past frame \( \{ c_{i} \} \), are added as states. Hence, \( P_{\ell} \) points are augmented into the state vector \( m^{c_{i}} = [(z_{1}^{c_{i}})^{T} \cdots (z_{P_{\ell}}^{c_{i}})^{T}]^{T} \), that is added and removed for each incoming image depending on the feature point matches. A total number of \( P \) points are matched from \( L \) images and represented by \( m \).

**IV. MEASUREMENT MODEL**

The observations for the current frame \( \{ c \} \) are the feature points on the ground plane, matched to those observed in previous frame \( \{ c_{i} \} \). Suppose a ground plane feature point \( \pi_{i} \) has been matched. Then its coordinates on the normalized image plane is modeled by the pinhole camera model \( (4) \),

\[
z_{i}^{c} = \begin{bmatrix} I_{2} & 0 \end{bmatrix} (e_{z}^{c} \pi_{i}^{c})^{-1} \pi_{i}^{c} + n_{i},
\]

where the pixel noise \( n_{i} \) is assumed to be zero-mean with covariance matrix \( C_{i} = \sigma_{\text{pix}}^{2} I_{2} \).

The importance of parameterization of the feature points position \( \pi_{i} \), equation (10), is appeared here, since, it relates the observation to the previous frame \( z_{i}^{c_{i}} \) and camera position \( p_{c_{i}}^{n} \), both of which are states.

By stacking all \( P_{\ell} \) feature point matches corresponding to view \( \ell \), one obtains the nonlinear measurement equation

\[
y_{\text{pts,}\ell}^{c_{i}} = [z_{1}^{c_{i}} \cdots z_{P_{\ell}}^{c_{i}}]^{T} = h(\delta x^{\text{ins}}, \delta x^{\text{cam}}, m^{c_{i}} + n_{\text{pts,}\ell}).
\]

The feature points not matched to \( \ell \) are subsequently matched to a previous older frame \( \ell + 1 \), where \( \ell = 1, \ldots, L \), until at most \( P \) observations are collected. The aggregate vector of dimension \( 2P \) is denoted by \( y_{\text{pts}} \). Consequently, the error covariance matrix is \( C_{\text{pts}} = \sigma_{\text{pix}}^{2} \times I_{2P} \).
In addition, the epipolar point arising from view $\ell$ can be written as

$$z_{\text{epi},\ell} = [I_2 \ 0] (e_2^T t_{c,\ell}^c)^{-1} t_{c,\ell}^c,$$

(14)

where $t_{c,\ell}^c \equiv R_k^c R_n^c (p_{c,\ell}^n - p_c^n)$. The error covariance matrix of the observation is approximated by the dispersion of intersections,

$$C_{\text{epi},\ell} = \sum_j w_j (c_j(\ell - \hat{z}_{\text{epi},\ell}^c)(c_j(\ell - \hat{z}_{\text{epi},\ell}^c))^T.$$

When the number of intersections is small the covariance is increased to prevent the uncertainty ellipse from collapsing, $C_{\text{epi},\ell} = C_{\text{epi},\ell} + \lambda I_2$. Here $\lambda$ is a decreasing function of the number of intersections.

The observed epipolar points from $L$ views are stacked, yielding the nonlinear measurement equation

$$\mathbf{y}_{\text{epi}}^c = [(z_{\text{epi},1})^T \cdots (z_{\text{epi},L})^T]^T + \mathbf{n}_{\text{epi}}$$

$$= \mathbf{h} (\delta \mathbf{x}^\text{ins}, \delta \mathbf{x}^\text{cam}) + \mathbf{n}_{\text{epi}},$$

(15)

where the noise covariance matrix $C_{\text{epi}}$ is the block-diagonal concatenation of each matrix $C_{\text{epi},\ell}$.

V. Estimation Framework

The joint state-space model for the INS and camera error states, along with the measurements (13) and (15), is compactly written as

$$\delta \mathbf{x}_{k+1} = \mathbf{F}_k \delta \mathbf{x}_k + \mathbf{G}_k \mathbf{w}_k$$

$$\mathbf{y}_{k}^c = \mathbf{h}(\delta \mathbf{x}_k, \mathbf{m}_k) + \mathbf{n}_k,$$

(16)

where the covariance matrix of $\mathbf{n}_k$, $C_{n,k}$, is the concatenation of $C_{n,k}$ and $C_{\text{epi}}$. We model the statistics of the feature point vector $\mathbf{m}$ (Section III-C) based on the following simplifying assumptions: Its expected value is taken as the observed coordinates on the image plane, $\mathbf{m}$, and its covariance matrix is $C_{\mathbf{m}} = \sigma_{\text{pix}}^2 \times I_{2p}$. Further, $\mathbf{m}$ and $\delta \mathbf{x}$ are assumed to be uncorrelated.

Since the process model is linear, the predicted errors are zero with error covariance matrix propagated as

$$\mathbf{P}_{k+1} = \mathbf{F}_k \mathbf{P}_{k} \mathbf{F}_k^T + \mathbf{G}_k \mathbf{Q} \mathbf{G}_k^T,$$

where

$$\mathbf{F}_k \triangleq \begin{bmatrix} \mathbf{F}_{k}^\text{ins} & 0 \\ 0 & I_{4L} \end{bmatrix} \text{ and } \mathbf{G}_k \triangleq \begin{bmatrix} \mathbf{G}_{k}^\text{ins} \\ 0_{3L \times 12L} \end{bmatrix}.$$

When a new image is recorded an error state $\delta \mathbf{p}_{c}^n$ is augmented to $\delta \mathbf{x}_{\text{cam}}$ (5), and the error covariance matrix is updated as

$$\mathbf{P}' = \begin{bmatrix} \mathbf{P} & \mathbf{P} \mathbf{T}^T \\ \mathbf{TP} & \mathbf{T} \mathbf{P} \mathbf{T}^T \end{bmatrix}$$

(17)

where $\mathbf{T} = [\mathbf{T} \ \mathbf{0}]$. When the buffer of $L$ views is full, the oldest state and its corresponding rows and columns of the covariance matrix are decimated.

The SPKF is set to perform a measurement update when ground plane features have been detected and matched. Based on the state-space model, (16), the SPKF estimates the errors linearly as

$$\delta \mathbf{x}_k = \mathbf{K}_k (\mathbf{y}_k^c - \hat{\mathbf{y}}_k^c),$$

(18)

where $\hat{\mathbf{y}}_k^c$ is the measurement prediction and $\mathbf{K}_k$ is the Kalman gain matrix. An overview of the algorithm is given in Algorithm 1.

The joint statistics of $\delta \mathbf{x}$ and $\mathbf{m}$ are propagated as $2N + 1$ sigma points, where $N$ is the total number of states $N = 15 + 3L + 2P$. Let the joint state vector be denoted by $\mathbf{x} = [\delta \mathbf{x}^T \ \mathbf{m}^T]^T$ and $\hat{\mathbf{x}} = [\mathbf{0}^T \ \hat{\mathbf{m}}^T]^T$ be the prediction with error covariance matrix $\mathbf{P}^- \oplus C_{\mathbf{m}}$. The sigma points $\mathbf{x}_j = [\delta \mathbf{x}_j^T \ \mathbf{M}_j]^T$ are then generated using the matrix square-root of the joint error covariance matrix. By propagating the sigma points through the measurement equations (13) and (15), the correlations between the error states $\delta \mathbf{x}$ and observations $\mathbf{y}_k^c$ can be approximated [22]. These correlations form the basis of the Kalman gain.

Algorithm 1 Sigma-Point Kalman Filter

1: Initialize $\mathbf{P}_0$
2: for $k = 0, \ldots$ do
3: Update INS state estimates and $\{\mathbf{R}_{c,\ell}\}$
4: if new image $\exists$ then
5: Extract and match feature points $\{z_{c,\ell}^k\}$
6: Detect feature points on ground plane
7: Estimate epipoles and covariance matrices
8: end if
9: if $\hat{\mathbf{y}}_k^c$ and/or $\mathbf{y}_{\text{epi}}^c$ exists then
10: Generate sigma points and prediction:
11: $\mathbf{X}_j = \mathbf{x}_j^c \pm \eta \cdot \text{column}[(\mathbf{P}_k \oplus C_{\mathbf{m}})^{1/2}]_j$
12: $\mathbf{Y}_j = \mathbf{h}(\delta \mathbf{x}_j, \mathbf{M}_j)$
13: $\hat{\mathbf{y}}_k^c = \sum_{j=0}^{2N} w_j^n \mathbf{y}_j$
14: Measurement update:
15: $C_{e,k} = \sum_{j=0}^{2N} w_j^n (\mathbf{Y}_j - \hat{\mathbf{y}}_k^c)(\mathbf{Y}_j - \hat{\mathbf{y}}_k^c)^T + C_{n,k}$
16: $\mathbf{D}_k = \sum_{j=0}^{2N} w_j^n \delta \mathbf{x}_j (\mathbf{Y}_j - \hat{\mathbf{y}}_k^c)^T$
17: $\mathbf{K}_k = \mathbf{D}_k C_{e,k}^{-1}$
18: $\delta \mathbf{x}_k = \mathbf{K}_k (\mathbf{y}_k^c - \hat{\mathbf{y}}_k^c)$
19: $\mathbf{P}_k = \mathbf{P}_k - \mathbf{K}_k C_{e,k} \mathbf{K}_k^T$
20: Use $\delta \mathbf{x}_k$ to correct state estimates
21: $\mathbf{P} := \mathbf{P}_k$
22: end if
23: $\mathbf{P} := \mathbf{P}_k$
24: end if
25: if new image $\exists$ then
26: Record view position $\mathbf{p}_{c,\ell}^n$
27: Decimate and augment error states $\delta \mathbf{p}_{c,\ell}^n$
28: Update $\mathbf{P}$ correspondingly
29: end if
30: Prediction:
31: $\mathbf{P}_{k+1} = \mathbf{F}_k \mathbf{P}_{k} \mathbf{F}_k^T + \mathbf{G}_k \mathbf{Q} \mathbf{G}_k^T$
32: end for

The weights in the SPKF are set as $w_l^n = w_0^n = \frac{1}{2(N+\lambda)}$ for $l = 1, \ldots, 2N$. For $l = 0$, $w_0 = \frac{\lambda}{N+\lambda} + (1 - \alpha^2 + \beta)$ and $w_0^n = \frac{\lambda}{N+\lambda}$. Here $\lambda = \alpha^2 (N + \kappa) - N$, with parameters set to $\alpha = 0.1$, $\beta = 2$ and $\kappa = 0$ that also determine the spread of the sigma points through the weight $\eta = \sqrt{N+\lambda}$. 
VI. FEATURE DETECTION AND OUTLIER REJECTION

The image processing module of the system consists of feature extraction and matching, i.e., identifying distinguishable features between successive image frames. We used the speeded up robust features (SURF) [27], which is invariant to image scale and rotation. The MATLAB Computer Vision Toolbox implementation of SURF is used for feature detection and matching. The algorithm is capable of identifying corresponding features in successive images, despite poor lighting conditions in the test environment.

Fig. 3a shows an example of SURF feature detection and matching in the test environment. Depending on the hardware structure, different vision-based ground plane feature detection approaches are studied in the literature [15], [23], [24]. In this paper, for simplicity, a method similar to [15] is used for the ground plane feature detection. That is, among all the detected and matched features, the ground plane features are sampled from a square-shaped region of interest in the lower part of the image. For the sake of illustration, the square-shaped region together with the selected and matched features are transformed in the virtual image plane in Fig. 3b, which can be interpreted as the top view image of the ground plane, Section III-A.

In order to detect and reject mismatches or very noisy observations, we employ a residual-based outlier rejection before using the observed points in the image plane in the measurement update (18). For each observed point \( z^c_j \) the following test was performed,

\[
(z^c_j - \hat{z}^c_j)^\top C_{e,j}^{-1}(z^c_j - \hat{z}^c_j) < \gamma, \tag{19}
\]

where \( \hat{z}^c_j \) is the prediction and \( C_{e,j} \) is the corresponding submatrix of \( C_e \). Points that fail the test are rejected and removed from the observation vector.

VII. RESULTS

The described algorithm in the preceding sections has been evaluated both in simulations and real experiments. Both the simulation and real experiments have verified the 6 DoF motion estimations, however, due to the space limitations, only the results of one real experiment are given in this section.

A. Setup

The hardware used for this experiment is shown in Fig. 4a. The camera is an AVT Guppy monochrome camera that is rigidly mounted on top of a MicroStrain 3DMGX2 IMU with sampling rate 250 Hz. Images are captured with resolution 752 × 480 pixels, where the camera sampling rate is 10 Hz. The camera internal calibration parameters and the IMU-camera calibration parameters are both estimated using [28].

To evaluate the performance of the proposed method with real data, a controlled test environment was set up in an underground hall at the university of KTH. Some of the sample images from the test environment are given in Fig. 4b. The underground location of the test environment provides a controlled level of radio interference for a reference commercial UWB system [29] used as a ground truth for evaluating the results. The accuracy of the UWB system is about 15 cm under line-of-sight and favorable geometrical configurations of the tags and sensors. When line-of-sight is lost, however, the UWB position estimates become unreliable. During the experiments all data were stored on a computer and processing was done off-line.

B. Experimental validation

In this experiment 3081 image frames were recorded for about 5 minutes. The IMU and camera were placed at the bottom of a trolley, moving in a closed loop trajectory. The system was initialized at an estimated height of 21 cm above the ground. The roll and pitch were estimated using the gravitational force during the first samples when the system was stationary [25].

In the implementation, the number of views in memory was set to \( L = 5 \). The measurement noise level \( \sigma_{\text{pix}} \) was set to...
be equivalent to 10 pixels. The outlier rejection level $\gamma = 6$ and any measured epipole outside of the visible region of the image plane was rejected. The maximum number of ground plane feature point states was set to $P = 8$.

The estimated trajectory along with the UWB data, used as the ground truth, is plotted in Fig. 5a over the map of the test environment. Although, the UWB sensors were not in line-of-sight during a substantial part of the trajectory, the results do show that the SPKF is capable of drastically reducing the growth rate of the accumulated INS errors. The trajectory follows the UWB position estimates and the system stops approximately 50 cm from the reference.

Furthermore, the estimated height of the mobile system is shown in Fig. 5b, which is crucial in the measurement model of the ground plane feature points. The result indicates that with sufficient matches and translation the height errors can be tracked by the SPKF during this period of time.

The $3\sigma$ level for the estimated error in the IMU position, orientation, and velocity along the three axis are shown in Fig. 6. The $\sigma$ values are the square root of the computed corresponding diagonal elements of the states error covariance matrix that provides a representation of the SPKF’s estimate uncertainty. According to Fig. 6a, the filter consistency is also verified by the height error, approximately 5 cm error, at the end of the experiment. Given that no height update was done in the estimation process, the vision-based error correction of the INS is significant. Expectedly, the positional error along the $x$ and $y$ axis grows slowly as the system has no access to the absolute position update.

The provided results demonstrate the potential application of the proposed method in both the pedestrian and robotics applications for 3D pose estimations. Since, in many robotics applications the height is fixed, the height information can be added as a pseudomeasurement to the SPKF, which will dramatically improve the 2D pose estimation for long term applications. Alternatively, a barometer could provide such information for pedestrians to maintain the stability of longer trajectories.

Since the image of three non-collinear points determines a unique affine transformation in the plane, at least three non-collinear points are necessary to be selected in images to define the ground plane. Although increasing the number of feature points improves the stability of the estimated trajectory, the number of points must be kept reasonably low to reduce the complexity of the filter.

Video clips and supplemental materials of the experiment are available at http://www.ee.kth.se/~ghpa/plans2012.

VIII. CONCLUSION

An IMU-camera sensor fusion approach has been used to construct a vision-aided inertial navigation system by tracking salient features of planar terrain. In the proposed algorithm, a new measurement model has been derived based on the virtual image of a downward-looking camera where the real camera movement is not restricted. The proposed algorithm is not only capable of velocity estimation comparable to optical flow methods but also accurate 3D pose estimation. The problem of nonlinearity of the state-space model of the system has been handled by the state-augmented Sigma-Point Kalman filter. The presented results from the real experiments indicate that the method is capable of correcting the INS solution even with a few number of features while the position error growth rate is significantly reduced. As a future work, instead of using a fixed shaped region in the image plane for ground plane feature detection, we are investigating a more flexible and reliable method for ground plane feature detection, outlier rejection, and obstacle removal.

APPENDIX A

\[
F^\text{ins}_k = \begin{bmatrix}
I_3 & dtI_3 & 0_3 & 0_3 & 0_3 \\
0_3 & I_3 & dt[R_{b,k}^n] & dtR_{b,k}^n & 0_3 \\
0_3 & 0_3 & I_3 & 0_3 & -dtR_{b,k}^n \\
0_3 & 0_3 & 0_3 & I_3 & 0_3 \\
0_3 & 0_3 & 0_3 & 0_3 & I_3
\end{bmatrix}
\]

and

\[
G^\text{ins}_k = \begin{bmatrix}
0_3 & 0_3 & 0_3 & 0_3 \\
dt[R_{b,k}^n] & 0_3 & 0_3 & 0_3 \\
0_3 & -dtR_{b,k}^n & 0_3 & 0_3 \\
0_3 & 0_3 & dtI_3 & 0_3 \\
0_3 & 0_3 & 0_3 & dtI_3
\end{bmatrix}.
\]
ACKNOWLEDGMENT

The authors would like to thank Dr. Alessio De Angelis for his assistance with data collections and installation of the the reference system.

REFERENCES


