<http://kth.diva-portal.org>

This is an author produced version of a paper published in *IEEE/ION Position Location and Navigation Symposium (PLANS)*, 2010.

This paper has been peer-reviewed but does not include the final publisher proof-corrections or proceedings pagination.

© 2010 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Citation for the published paper:

Dave Zachariah and Magnus Jansson.
Camera-aided inertial navigation using epipolar Points
IEEE/ION Position Location and Navigation Symposium (PLANS), 2010.
Access to the published version may require subscription.
Published with permission from: IEEE

Camera-aided inertial navigation using epipolar points

Dave Zachariah and Magnus Jansson

ACCESS Linnaeus Centre

Royal Institute of Technology (KTH)

SE-100 44 Stockholm, Sweden

Email: {dave.zachariah, magnus.jansson}@ee.kth.se

Abstract—Due to the rapid error growth of navigation systems using low-cost inertial measurement units there is a need to fuse the information with complementary sensors. In this paper a monocular camera is used to aid the system. Unlike SLAM-like approaches the problem of estimating the location of each feature point viewed in a scene is avoided, instead estimated epipolar points on the image plane are used. By maintaining a buffer of past views, the method mimics a short-term visual memory which imposes multiple constraints on the estimation problem. The result is a Sigma-Point Kalman filter in square-root form with a linear and efficient time-update. A simulation study is presented indicating the filter’s capacity to constrain the rate of error growth of an inertial navigation system. The filter may also find useful applications when fusing with additional sensors.

Index Terms—Inertial navigation, monocular camera, sensor fusion, Sigma-Point Kalman filter.

I. INTRODUCTION

Inertial navigation using a low-cost inertial measurement unit (IMU) is subject to rapid error growth that renders the navigation solution useless within a very short period of time. A significant underlying factor is the error in the gyroscope which causes a misalignment in the estimated attitude. Then the gravitational acceleration contained in the observations from the accelerometers cannot be correctly compensated in the inertial navigation system (INS) equations. Hence the need to aid the inertial navigation with complementary sensors that constrain the error or rate of error growth.

We consider a monocular camera, which is an inexpensive and passive sensor that provides a signal with a large information content. However, while humans use a vast amount of prior knowledge to extract the information for navigation, machine-based systems have often to rely on low-level feature extraction algorithms for tractability.

In [1], visual information of static feature points in the scene is fused with an inertial system to aid it in outdoor scenarios with limited or unreliable access to external localization information, e.g. due to GPS outages. Similarly, [2] used stereo cameras to aid a system in an indoor scenario, where external localization hardware is difficult to deploy and where GNSS signals are not easily obtained.

Within robotics, a similar scenario is dealt with in the Simultaneous Localization and Mapping (SLAM) problem. Over the recent years successful methods for SLAM using only a monocular camera have been proposed and demonstrated,

cf. [3]–[7]. The converse approach of aiding the monoSLAM algorithm with inertial information was used in [8] in order to improve scale consistency and hence loop closing.

For navigation purposes considered here, however, mapping is of secondary importance. Building a consistent map of feature points increases the computational complexity of the estimation problem significantly. Indeed, recent efforts have been made to sparsify maps, e.g. [9]. But common to the methods above is that they estimate static feature points in the scene, under various parameterizations, which are ‘nuisance parameters’ in the navigation problem. The approach in [10] avoids this by storing a number of past views that act as multiple constraints. However in the measurement function each feature point is reconstructed solving a nonlinear Least Squares problem. Similarly, [11] presents a view-based SLAM approach.

In this paper, we use the ideas in [12] who recover the relative pose between two views using vanishing points. But whereas they rely on an outlier rejection algorithm for classifying points on the image plane, we make use of the inertial information provided by the gyroscopes. A simulation study indicates the capacity of the proposed estimation algorithm to constrain the rate of error growth in an inertial navigation system.

II. EPIPOLES AND ROTATION UNWRAPPING

Let $\{n\}$, $\{b\}$ and $\{c\}$ denote the navigation, inertial and camera coordinate frame, respectively.

As shown in Figure 1, when image planes are parallel the epipolar point contains information about the translational movement of the origin of the camera frame, which is of interest to our estimation problem. Under pure translation, static points in the navigation frame $\{n\}$ move along lines in the image plane. Ideally all lines intersect at the epipole [13]. Therefore it is possible to estimate it by tracking feature points on the image plane, computing their lines of movement and intersection.

The fundamental matrix $\mathbf{F} \in \mathbb{R}^{3 \times 3}$ is the algebraic representation of epipolar geometry and is of rank 2 [14]. Let $\Delta \mathbf{p}$ be the translation vector and $\mathbf{R}_{c_j}^{c_i}$ the rotation matrix between two views. Then the fundamental matrix equals up to scale, $\mathbf{F} \simeq \mathbf{A}^{-\top} [\Delta \mathbf{p}]_{\times} \mathbf{R}_{c_j}^{c_i} \mathbf{A}^{-1}$, where $\mathbf{A}$ is the camera calibration matrix containing intrinsic camera parameters and $[\cdot]_{\times}$ is the

Fig. 1. (a) Translation from camera center c_0 to c . Epipole, shown by a circle in current frame, is the vanishing point of the direction of translation. (b) Under pure translational movement, the image planes are parallel and points move along lines radiating from the epipole.

Fig. 2. Image points $\mathbf{z}$ represented as unit vectors $\mathbf{u}$.

skew-symmetric matrix representation of a vector in the cross product operation. Let $\mathbf{y}_p$ denote the epipole in projective coordinates. It lies in the null space of $\mathbf{F}$, i.e. $\mathbf{F}\mathbf{y}_p = \mathbf{0}$. Under pure translation this yields $\Delta\mathbf{p} \times \mathbf{A}^{-1}\mathbf{y}_p = \mathbf{0}$ or equivalently, the translation vector $\Delta\mathbf{p}$ and the (normalized) vector of the epipole $\mathbf{A}^{-1}\mathbf{y}_p$ are parallel. Hence measuring $\mathbf{y}_p$ will give information about $\Delta\mathbf{p}$.

However, in general the camera does not move under pure translation. Then the movement of a feature point on the image plane contains both a translational and a rotational component and the computation of the epipole using the method above is not valid. One way to circumvent this problem is to use the gyroscopes that provide information about the orientation in order to effectively preserve parallel image planes.

There is a one-to-one mapping, $\tau : \mathbb{R}^2 \rightarrow \mathbb{R}^3$, from the image plane and the half-sphere facing it, illustrated in Figure 2. An image point $\mathbf{z} \in \mathbb{R}^2$ can then be represented as a unit vector $\mathbf{u} \in \mathbb{R}^3$. Suppose an image point $\mathbf{z}_0$ from a past view $\{c_0\}$ has been detected. Then it can be transformed to a point $\mathbf{z}'_0$ on a plane parallel to that in current view $\{c\}$ by $\mathbf{z}'_0 = \tau^{-1}(\mathbf{u}'_0)$ where $\mathbf{u}'_0 = \mathbf{R}_{c_0}^c \mathbf{u}_0$ and $\mathbf{u}_0 = \tau(\mathbf{z}_0)$, thereby ‘unwrapping’ the effect of rotation.

In practice the gyroscopes contain noise that will lead to an accumulated error of the relative orientation encoded in $\mathbf{R}_{c_0}^c$. It will be assumed, however, that the order of error growth is moderate so that the error of the computed epipole between the current and past view is small.

Fig. 3. Feedback approach. Filter estimates the navigation error states $\delta\mathbf{x}$ and corrects the navigation solution.

III. PROCESS MODEL

A. Navigation error states

The navigation states of interest are position $\mathbf{p}^n$, velocity $\mathbf{v}^n$ and orientation $\mathbf{R}_b^n$. The time-evolution of the states is given by the inertial navigation system equations and driven by the external signals from the IMU.

In order to fuse the sensor information in a tractable way, a feedback approach to the estimation problem is taken. Figure 3 gives a conceptual outline. Instead of estimating the navigation states, the errors $(\delta\mathbf{p}^n, \delta\mathbf{v}^n, \delta\boldsymbol{\theta})$, where the attitude error is expressed as Euler angles, are estimated and fed back to correct the solution of the inertial navigation system equations. The advantage is that the evolution of the errors can be approximated by a linear process model, provided the velocity and attitude errors are not too large.

The accelerometer and gyroscopes of the IMU contain systematic errors, or biases, denoted $\delta\mathbf{f}^b$ and $\delta\boldsymbol{\omega}^b$, respectively. These nuisance parameters are estimated along with the navigation error states. Let dt be the sampling period, then the linearized, discrete-time error process model can be written as

$$\begin{aligned}\delta\mathbf{p}_{k+1}^n &= \delta\mathbf{p}_k^n + dt\delta\mathbf{v}_k^n \\ \delta\mathbf{v}_{k+1}^n &= \delta\mathbf{v}_k^n + dt[\hat{\mathbf{R}}_{b,k}^n \tilde{\mathbf{f}}_k^b] \times \delta\boldsymbol{\theta}_k \\ &\quad + dt\hat{\mathbf{R}}_{b,k}^n \delta\mathbf{f}_k^b + dt\hat{\mathbf{R}}_{b,k}^n \mathbf{n}_{f,k} \\ \delta\boldsymbol{\theta}_{k+1} &= \delta\boldsymbol{\theta}_k - dt\hat{\mathbf{R}}_{b,k}^n \delta\boldsymbol{\omega}_k^b - dt\hat{\mathbf{R}}_{b,k}^n \mathbf{n}_{\omega,k},\end{aligned}\tag{1}$$

where the sensor noise terms $\mathbf{n}_f$ and $\mathbf{n}_\omega$ are modeled as zero-mean, white Gaussian processes.

The slow-varying sensor biases are modeled as integrated white noise:

$$\begin{aligned}\delta\mathbf{f}_{k+1}^b &= \delta\mathbf{f}_k^b + dt\mathbf{w}_{\delta f,k} \\ \delta\boldsymbol{\omega}_{k+1}^b &= \delta\boldsymbol{\omega}_k^b + dt\mathbf{w}_{\delta \omega,k}\end{aligned}\tag{2}$$

The navigation error states can be compactly written as

$$\delta\mathbf{x}_{k+1} = \mathbf{F}_k \delta\mathbf{x}_k + \mathbf{G}_k \mathbf{n}_k \in \mathbb{R}^{15}.\tag{3}$$

B. Camera position error state

Let $\mathbf{p}_c^b$ denote the offset between inertial frame and camera frame. Then the current camera position is $\boldsymbol{\rho} = \mathbf{p}^n + \mathbf{R}_b^n \mathbf{p}_c^b$ and the position error expressed in $\{n\}$,

$$\delta\boldsymbol{\rho} = \delta\mathbf{p}^n + [\hat{\mathbf{R}}_b^n \mathbf{p}_c^b] \times \delta\boldsymbol{\theta} \in \mathbb{R}^3.\tag{4}$$

Hence there is a linear relation between current camera position error and navigation error states $\delta\boldsymbol{\rho} = \mathbf{T}\delta\mathbf{x}$, where

$$\mathbf{T} \triangleq (\mathbf{I}_3 \quad \mathbf{0}_3 \quad [\hat{\mathbf{R}}_b^n \mathbf{p}_c^b]_\times \quad \mathbf{0}_{3 \times 6}). \quad (5)$$

A recorded camera pose has no dynamics, therefore the error process model is simply $\delta\boldsymbol{\rho}_{k+1} = \delta\boldsymbol{\rho}_k$.

C. Complete error state model

The more epipoles used, the more constraints we can impose on the estimation problem. This requires, however, a memory of past camera images, their positions and errors $\{\delta\boldsymbol{\rho}^{(1)}, \dots, \delta\boldsymbol{\rho}^{(P)}\}$ in order to compare with the current image and position.

The error state vector is augmented as

$$\delta\bar{\mathbf{x}}_k \triangleq \text{vec}([\delta\mathbf{x}_k \quad \delta\boldsymbol{\rho}_k^{(1)} \dots \delta\boldsymbol{\rho}_k^{(P)}]) \in \mathbb{R}^{15+3P}. \quad (6)$$

Since computational resources are finite, the maximum size of the memory is set to $P_{\max}$. New camera positions will be added and old ones removed. Hence the process is a sliding window of views.

The total error state model can be written as $\delta\bar{\mathbf{x}}_{k+1} = \bar{\mathbf{F}}_k \delta\bar{\mathbf{x}}_k + \bar{\mathbf{G}}_k \mathbf{n}_k$, where

$$\bar{\mathbf{F}}_k = \begin{pmatrix} \mathbf{F}_k & \mathbf{0}_{15 \times 3P} \\ \mathbf{0}_{3P \times 15} & \mathbf{I}_{3P} \end{pmatrix}, \bar{\mathbf{G}}_k = \begin{pmatrix} \mathbf{G}_k \\ \mathbf{0}_{3P \times 12} \end{pmatrix} \quad (7)$$

and the covariance matrix of the process noise is $\mathbf{Q}_k$.

IV. MEASUREMENT MODEL

The relative orientation between the current view and a past view i must be updated in order to unwrap the rotation and compute the corresponding epipole $\mathbf{y}^{(i)} \in \mathbb{R}^2$. This can be done efficiently using quaternions by storing the relative orientation in the inertial frames $\{b\}$, that can be transformed into $\{c\}$ when needed.

Let $\mathbf{q}_b^{b_i}$ denote the quaternion that expresses the rotation from the current inertial frame $\{b\}$ to the past frame $\{b_i\}$. Each time a view is recorded a new vector $\mathbf{q}_b^{b_j} = [0 \quad 0 \quad 0 \quad 1]^\top$ is augmented to the matrix $\boldsymbol{\Theta} = [\mathbf{q}_b^{b_1} \dots \mathbf{q}_b^{b_P}]$. All orientations are updated in one operation, $\boldsymbol{\Theta}_k = \boldsymbol{\Omega}(\tilde{\boldsymbol{\omega}}_k) \boldsymbol{\Theta}_{k-1}$, where $\tilde{\boldsymbol{\omega}}_k$ is the sampled angular velocity and $\boldsymbol{\Omega}$ is the quaternion update matrix [15]. Hence the update can be done for each IMU sample simultaneously with the attitude update in the INS equations.

When a new image is obtained, its feature points are extracted and compared with the feature points in view i , producing a set of matched pairs $\{(\mathbf{z}_j, \mathbf{z}_{i,j})\}_j$. Then $\mathbf{R}_b^{b_i}$, obtained from the corresponding quaternion, is used to compute the rotation matrix $\mathbf{R}_{c_i}^c = (\mathbf{R}_b^c \mathbf{R}_b^{b_i} \mathbf{R}_c^b)^*$ and unwrap the rotation $\mathbf{z}_{i,j}' = \tau^{-1}(\mathbf{R}_{c_i}^c \tau(\mathbf{z}_{i,j}))$. This produces a set of matched feature points on parallel image planes $\{(\mathbf{z}_j, \mathbf{z}_{i,j}')\}_j$.

Each pair generates a line and ideally all of them intersect at the epipole. However, there are a host of error sources that arise from the inertial and visual sensors, as well as from image processing, that lead to noise on the measured epipole $\mathbf{y}^{(i)}$, which is modeled as a zero-mean additive white process $\mathbf{n}_y$. For robustness, the intersection $\mathbf{c}_j$ is computed for

each pair of lines. L valid lines would result in $L(L-1)/2$ intersections. The epipole is then computed as a weighted average of all intersections,

$$\mathbf{y}^{(i)} = \sum_j w_j \mathbf{c}_j. \quad (8)$$

In addition, the spread of intersections $\mathbf{c}_j$ yields a measure of the uncertainty of the measured epipole. The measurement noise covariance matrix can be approximated as

$$\hat{\mathbf{R}}_{\mathbf{y}^{(i)}} = \sum_j w_j (\mathbf{c}_j - \mathbf{y}^{(i)})(\mathbf{c}_j - \mathbf{y}^{(i)})^*. \quad (9)$$

The weights w_j , which need not be identical to those used in the weighted average, are set according to some measure of the reliability of the intersection $\mathbf{c}_j$, e.g. as a function of the distance between the points that generate the lines $\|\mathbf{z}_j - \mathbf{z}_{i,j}'\|_2$. A smaller distance results in a line that is more sensitive to noise. The further away the feature point is in $\{n\}$, the smaller the displacement on the image plane. In the limit, such points are invariant on the image plane to translational movement.

Since image processing is prone to outliers, the above procedure for computing the epipole requires some method of outlier rejection to suppress extreme intersections $\mathbf{c}_j$, e.g. arising from near parallel lines.

As explained in Section II, under pure translational movement the epipole coincides with the vanishing point of the direction of the translation vector, which is related to the navigation states. The translation vector between view i and the current view, expressed in $\{c\}$, can be written in terms of the error states $(\delta\mathbf{p}^n, \delta\boldsymbol{\theta}, \delta\boldsymbol{\rho}^{(i)})$ as

$$\begin{aligned} \Delta\mathbf{p}^{(i)} &= \mathbf{R}_b^c \mathbf{R}_n^b (\boldsymbol{\rho}^{(i)} - \boldsymbol{\rho}) \\ &= \mathbf{R}_b^c \hat{\mathbf{R}}_n^b \mathbf{C}(\delta\boldsymbol{\theta}) \\ &\quad \cdot [\hat{\boldsymbol{\rho}}^{(i)} - \hat{\mathbf{p}}^n + \delta\mathbf{p}^n - \delta\boldsymbol{\rho}^{(i)} - \mathbf{C}^*(\delta\boldsymbol{\theta}) \hat{\mathbf{R}}_b^n \mathbf{p}_c^b], \end{aligned} \quad (10)$$

where $\mathbf{C}$ is the direction-cosine matrix.¹ When projected on the image plane using a pinhole camera model gives the epipole on parallel planes:

$$\mathbf{y}^{(i)} = \bar{\mathbf{A}}([0 \quad 0 \quad 1] \Delta\mathbf{p}^{(i)})^{-1} \Delta\mathbf{p}^{(i)} + \mathbf{n}_y \quad (11)$$

where $\bar{\mathbf{A}}$ is a truncated camera calibration matrix and $\mathbf{n}_y$ is measurement noise.

The computed epipoles are stacked as a vector $\bar{\mathbf{y}} = \text{vec}([\mathbf{y}^{(1)} \dots \mathbf{y}^{(P)}])$ with a block diagonal measurement noise covariance matrix $\mathbf{R}_{\bar{\mathbf{y}}} = \hat{\mathbf{R}}_{\mathbf{y}^{(1)}} \oplus \dots \oplus \hat{\mathbf{R}}_{\mathbf{y}^{(P)}} \in \mathbb{R}^{2P \times 2P}$.

V. ESTIMATION FRAMEWORK

The system forms a state-space model with a linear process model and a nonlinear measurement model:

$$\begin{aligned} \delta\bar{\mathbf{x}}_{k+1} &= \bar{\mathbf{F}}_k \delta\bar{\mathbf{x}}_k + \bar{\mathbf{G}}_k \mathbf{n}_k \\ \bar{\mathbf{y}}_k &= \mathbf{h}_k(\delta\bar{\mathbf{x}}_k) + \mathbf{n}_{y,k}. \end{aligned} \quad (12)$$

State estimation is done with respect to a Mean Square Error criterion within a class of linear estimators, following the

¹If the attitude errors $\delta\boldsymbol{\theta}$ are small, the following approximation can be made $\mathbf{C}(\delta\boldsymbol{\theta}) \approx (\mathbf{I}_3 - [\delta\boldsymbol{\theta}]_\times)$.

measurement and time-update form of the Kalman filter. The measurement update

$$\delta \hat{\mathbf{x}}_{k|k} = \delta \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\bar{\mathbf{y}}_k - \hat{\mathbf{y}}_{k|k-1}) \quad (13)$$

requires second-order moments for computing the Kalman gain matrix $\mathbf{K}_k = \langle \delta \mathbf{x}_k, \mathbf{e}_k \rangle \langle \mathbf{e}_k \rangle^{-2}$, where $\mathbf{e}_k$ is the innovation. These statistics are not easily obtained when using a highly nonlinear measurement function, and are here approximated using a Sigma-Point approach that uses deterministic sampling [16], [17]. The result is a Sigma-Point Kalman filter with a linear and efficient time update. The estimation procedure is summarized in Algorithm 1.

Algorithm 1 Estimation using Sigma-Point Kalman Filter, with weights $\{w^c\}$, $\{w^m\}$ and η . Number of error states $N = 15 + 3P$.

```

1: Initialize  $\hat{\mathbf{x}}_{0|-1}$ ,  $\mathbf{P}_{0|-1}$  and  $\delta \hat{\mathbf{x}}_{0|-1} = \mathbf{0}_{15 \times 1}$ 
2: for  $k = 0, \dots$  do
3:   Update INS equations and  $\Theta_k$ 
4:   if New image  $\exists$  then
5:     Extract feature points  $\{\mathbf{z}_j\}$ 
6:     Match and rotation unwrap past feature points  $\{\mathbf{z}_{i,j}\}$ 
7:     Compute epipoles  $\mathbf{y}^{(i)}$  and  $\hat{\mathbf{R}}_{\mathbf{y}^{(i)}}$  using  $\{(\mathbf{z}_j, \mathbf{z}_{i,j})\}_j$ 
8:   end if
9:   if  $\{\mathbf{y}^{(i)}\} \neq \emptyset$  then
10:    Form  $\bar{\mathbf{y}}_k$  and  $\mathbf{R}_{\bar{\mathbf{y}},k}$ 
11:    %Generate sigma points and prediction:
12:     $\delta \mathcal{X}_{k|k-1}^{(l)} = \mathbf{0}_{N \times 1} \pm \eta \cdot \text{col}[\mathbf{P}_{k|k-1}^{1/2}]_l$ 
13:     $\mathbf{y}_{k|k-1}^{(l)} = \mathbf{h}_k(\delta \mathcal{X}_{k|k-1}^{(l)})$ 
14:     $\hat{\mathbf{y}}_{k|k-1} = \sum_{l=0}^{2N} w_l^m \mathbf{y}_{k|k-1}^{(l)}$ 
15:    %Measurement update:
16:     $\mathbf{R}_{\mathbf{e},k} = \sum_{l=0}^{2N} w_l^c (\mathbf{y}_{k|k-1}^{(l)} - \hat{\mathbf{y}}_{k|k-1})$ 
17:     $\cdot (\mathbf{y}_{k|k-1}^{(l)} - \hat{\mathbf{y}}_{k|k-1})^* + \mathbf{R}_{\bar{\mathbf{y}},k}$ 
18:     $\mathbf{R}_{\delta \mathbf{x}, \mathbf{e}, k} = \sum_{l=0}^{2N} w_l^c \delta \mathcal{X}_{k|k-1}^{(l)} (\mathbf{y}_{k|k-1}^{(l)} - \hat{\mathbf{y}}_{k|k-1})^*$ 
19:     $\mathbf{K}_k = \mathbf{R}_{\delta \mathbf{x}, \mathbf{e}, k} \mathbf{R}_{\mathbf{e}, k}^{-1}$ 
20:     $\delta \hat{\mathbf{x}}_{k|k} = \delta \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\bar{\mathbf{y}}_k - \hat{\mathbf{y}}_{k|k-1})$ 
21:     $\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{R}_{\mathbf{e}, k} \mathbf{K}_k^*$ 
22:    Use  $\delta \hat{\mathbf{x}}_{k|k}$  to correct state estimates  $\hat{\mathbf{x}}_k$ 
23:     $\mathbf{S}_k := \mathbf{P}_{k|k}$ 
24:   else
25:     $\mathbf{S}_k := \mathbf{P}_{k|k-1}$ 
26:   end if
27:   if New image  $\exists$  then
28:     Record view position  $\hat{\rho}^{(i)}$ 
29:     Decimate and augment view orientations in  $\Theta_k$ 
30:     Decimate and augment view error states in  $\delta \hat{\mathbf{x}}_k$ 
31:     Update  $\mathbf{S}_k$  correspondingly
32:   end if
33:   %Time update:
34:    $\delta \hat{\mathbf{x}}_{k+1|k} = \mathbf{0}_{N \times 1}$ 
35:    $\mathbf{P}_{k+1|k} = \bar{\mathbf{F}}_k \mathbf{S}_k \bar{\mathbf{F}}_k^* + \bar{\mathbf{G}}_k \mathbf{Q}_k \bar{\mathbf{G}}_k^*$ 
36: end for

```

Note that after each measurement, $\delta \hat{\mathbf{x}}_{k+1|k}$ is set to zero and only the error covariance matrix $\mathbf{P}_{k+1|k}$ needs to be propagated during time updates. Since the view positions have no dynamics the system matrices $\bar{\mathbf{F}}_k$ and $\bar{\mathbf{G}}_k$ have a sparse structure that can be exploited in the propagation. In addition, since the computation of epipoles can be performed

independently for each view i the parallelism can be exploited for fast implementations.

For numerical stability, the filter is implemented in ‘array’ or square-root form [18], [19]. The error covariance matrices are propagated as lower-triangular matrix square-roots $\mathbf{P}^{1/2}$ which ensures positive definiteness. This has the additional advantage for the Sigma-Point Kalman filter that there is no need to factorize the error covariance matrix by Cholesky decomposition since it is already in the desired form.

The time update of $\mathbf{P}^{1/2}$ is done by means of a QR-factorization² and the measurement update by rank-1 Cholesky downdates³. Augmenting a new view error state $\delta \rho$, or decimating an old one when the number of recorded views reaches $P_{\max}$, modifies $\mathbf{P}$ correspondingly. In square-root form, however, this update $\mathbf{P}^{1/2} \rightarrow \bar{\mathbf{P}}^{1/2}$, where $\bar{\mathbf{P}}^{1/2}$ is the modified matrix, has to be specified more clearly.

When augmenting a state the modified matrix can be expressed as

$$\bar{\mathbf{P}}^{1/2} = \begin{pmatrix} \mathbf{P}^{1/2} & \mathbf{0} \\ \mathbf{T}' \mathbf{P}^{1/2} & \mathbf{0} \end{pmatrix} \quad (14)$$

where $\mathbf{T}' \triangleq [\mathbf{T} \quad \mathbf{0}_{3 \times 3P}]$.

When decimating the oldest view error state, consider the matrix

$$\mathbf{P}^{1/2} = \begin{pmatrix} \mathbf{X} & \mathbf{0} & \mathbf{0} \\ \mathbf{Y} & \mathbf{U} & \mathbf{0} \\ \mathbf{Z} & \mathbf{V} & \mathbf{W} \end{pmatrix} \quad (15)$$

where $\mathbf{X}$, $\mathbf{U}$ and $\mathbf{W}$ are lower-triangular matrices and $\mathbf{P}^{1/2}$ has been decomposed so that $\mathbf{Y} \in \mathbb{R}^{3 \times 15}$ and $\mathbf{U} \in \mathbb{R}^{3 \times 3}$ contain the correlations of the old state. Then the decimated matrix becomes

$$\bar{\mathbf{P}}^{1/2} = \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{B} & \mathbf{C} \end{pmatrix} \quad (16)$$

where $\mathbf{A} = \mathbf{X}$, $\mathbf{B} = \mathbf{Z}$ and $\mathbf{C}$ is the solution of $\mathbf{C}\mathbf{C}^* = \mathbf{V}\mathbf{V}^* + \mathbf{W}\mathbf{W}^*$, obtained by a lower-triangular Cholesky decomposition.

VI. SIMULATIONS

A. Setup

In order to test the capabilities of the filter, a simulation study of the system in a bounded environment was conducted. A user-defined trajectory is generated using cubic splines to ensure continuous acceleration. The outputs are ideal IMU signals sampled at 100 Hz. Biases and stochastic noise are added as constants and additive white Gaussian noise processes, respectively. Their parameters are given in Table I, with the standard deviation specified on a per sample basis. The gyroscope bias alone contributes to a error drift of each attitude state by approximately $0.1^\circ/\text{s}$.

The camera is simulated using the pinhole camera model. The visible feature points in $\{n\}$ are projected on the image

²It is obtained from factorization of $\mathcal{A}^* = \mathbf{U} \begin{pmatrix} \mathbf{P}_{k+1|k}^{*/2} \\ \mathbf{0} \end{pmatrix}$ where $\mathcal{A} \triangleq [\mathbf{F}_k \mathbf{P}_{k|k}^{1/2} \quad \mathbf{G}_k \mathbf{Q}_k^{1/2}]$.

³ $\mathbf{P}_{k|k}^{*/2} = \text{cholupdate}\{\mathbf{P}_{k|k-1}^{1/2}, \mathbf{U}, -1\}$, where $\mathbf{U} = \mathbf{K}_k \mathbf{R}_{\mathbf{e}, k}^{1/2}$.

TABLE I
IMU SENSOR NOISE PARAMETERS.

	Bias	Std.
Acc. [m/s ²]	2×10^{-2}	1×10^{-2}
Gyro. [rad/s]	2×10^{-3}	5×10^{-3}

plane at a rate of 10 Hz and rounded to integers as coordinates on a grid of 752×480 pixels. The origins of the camera and inertial frames were offset by 5×10^{-2} m. The camera calibration matrix containing the intrinsic camera parameters with focal length $f = 5 \times 10^{-3}$ m and pixel width $\Delta_x = \Delta_y = 6 \times 10^{-6}$ m.

Next, the estimation filter parameters are specified: The maximum length of the view buffer was set to $P_{\max} = 10$ frames, corresponding to a visual memory of 1 second. The weights in the Sigma-Point filter are set as $w_l^c = w_l^m = \frac{1}{2(N+\lambda)}$ for $l = 1, \dots, 2N$. For $l = 0$, $w_0^c = \frac{\lambda}{N+\lambda} + (1 - \alpha^2 + \beta)$ and $w_0^m = \frac{\lambda}{N+\lambda}$. Here $\lambda = \alpha^2(N + \kappa) - N$, with parameters $\alpha = 0.1$, $\beta = 2$ and $\kappa = 0$ that also determine the spread of the sigma points through the weight $\eta = \sqrt{N + \lambda}$.

The weights $\{w_j\}$ used for computing the epipole are set according to a measure of the reliability of the pair of lines used for each intersection $\mathbf{c}_j$. Here the minimum distance between the points that generate the pair of lines is used $\gamma_j \triangleq \min(\|\mathbf{z}_k - \mathbf{z}'_{i,k}\|_2, \|\mathbf{z}_l - \mathbf{z}'_{i,l}\|_2)$. The weights are then set as $w_j = w'_j / \sum_k w'_k$ where

$$w'_j = \begin{cases} 1 & \text{if } \gamma_j \geq \gamma_{\text{thres}} \\ 0 & \text{otherwise} \end{cases}$$

and the threshold γ_{thres} is set to 10 pixels.

Since image processing is prone to outliers, unreliable intersections have to be rejected completely. Here pairs of lines with slopes that differ by less than 30% are rejected.

B. Results

The simulated trajectory is shown in Figures 4 and 5. It consists of 3 laps with crossing paths, spanning 239 seconds in total. 200 feature points are sprinkled along the trajectory at varying heights. The system does not exhibit roll but the motion in the plane contains both left and right turns which excites changes in heading. In order to excite more angular motion the system changes elevation during the first 100 seconds, inducing changes in the pitch.

The filter is initialized with the correct position, velocity and attitude. Note that since navigation is conducted in a relative reference frame, initial position and heading are correct by definition. The initial error covariance matrix $\mathbf{P}_{0|-1} = \text{diag}(6.25 \times 10^{-2} \cdot \mathbf{I}_3, 0.1 \cdot \mathbf{I}_3, 2.5 \times 10^{-4} \cdot \mathbf{I}_3, 2.25 \times 10^{-4} \cdot \mathbf{I}_3, 6.25 \times 10^{-7} \cdot \mathbf{I}_3)$, which corresponds to a standard deviation of the initial attitude estimate to about 1° .

The result of a realization of the trajectory is shown in Figure 6. Whereas inertial navigation alone would quickly result in a position error of several hundred meters, the camera-aided filter is capable of keeping the estimate on track on the order of minutes.

Fig. 4. Simulated trajectory with 200 feature points.

Fig. 5. Simulated trajectory along z-axis.

Fig. 6. Estimated trajectory.

Fig. 7. Accelerometer bias estimates.

The filter was given a ‘cold start’ without any initial zero-velocity updates in a stationary mode and bias estimates initially set to zero. Examples of the evolution of the bias estimates are given in Figures 7 and 8. For the accelerometers, the estimate of δf_z quickly converges since it is parallel to the gravitation vector and there is slow or no motion in that direction. Similarly the estimates of $\delta\omega_\phi$ and $\delta\omega_\theta$ converge rapidly.

Heading ψ is harder to estimate in rigid motion along the horizontal plane perpendicular to the gravitation vector. The rotation unwrapping effectively removes the orientation information between two views. Unless angular motion is excited along ϕ and θ the estimate of $\delta\omega_\psi$ will not converge. One possible solution is to extend the measurement function to use the relative orientation information encoded in the vanishing points that are invariant to translation and therefore will be stationary on the epipolar line after rotation unwrapping.

For evaluation of the filter, 10 realizations of the same trajectory were generated. The navigation state errors are plotted in Figure 9, 10, and 11, along with the $\pm 3\sigma$ -levels, where σ is obtained from the diagonal elements of the filter’s error covariance matrices and hence represents its uncertainty of the estimate. The results are indicative of the error tracking capacity of the filter to constrain the rate of error growth. The filter can not, however, consistently track the heading error over long period of time when moving rigidly along the horizontal plane.

VII. CONCLUSIONS

A camera-aided inertial navigation system has been presented, exploiting a short-term visual memory that imposes multiple constraints using epipolar points between the current and past views. The estimation algorithm is a recursive Sigma-Point Kalman filter in square-root form with a linear and efficient time-update. The results from a simulation study are indicative of the filter’s capacity to constrain the rate of error

Fig. 8. Gyroscope bias estimates.

Fig. 9. Position errors, 10 realizations.

Fig. 10. Velocity errors, 10 realizations.

Fig. 11. Attitude errors, 10 realizations.

growth and keep the navigation state estimates on track during a time period in which the estimates of an unaided low-cost inertial navigation system would be rendered useless.

Work is currently being conducted to validate the performance for real data in an indoor scenario. The system can be extended to fuse more sensors, such as GNSS or UWB positioning, and may find useful applications in different navigation scenarios.

REFERENCES

- [1] M. George and S. Sukkarieh, "Inertial navigation aided by monocular camera observations of unknown features," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 3558–3564, Apr. 2007.
- [2] S. Ebcin and M. Veth, "Tightly-coupled image-aided inertial navigation using the unscented kalman filter," in *Proc. of the 20th International Technical Meeting of the Satellite Division of the Institute of Navigation ION GNSS 2007*, pp. 1851–1860, Sept. 2007.
- [3] A. Davison, "Real-time simultaneous localization and mapping with a single camera," in *Proc. Int. Conf. Comput. Vis.*, 2003.
- [4] J. Montiel, J. Civera, and A. Davison, "Unified inverse depth parametrization for monocular slam," in *Proc. Robotics: Science and Systems*, 2006.
- [5] R. Munguia and A. Grau, "Monocular slam for visual odometry," in *Proc. IEEE Int. Symp. Intelligent Signal Processing (WISP)*, Oct. 2007.
- [6] A. Davison, I. Reid, N. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.
- [7] S. Holmes, G. Klein, and D. Murray, "A square root unscented kalman filter for visual monoslam," in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 3710–3716, May 2008.
- [8] P. Pinies, T. Lupton, S. Sukkarieh, and J. Tardos, "Inertial aiding of inverse depth slam using a monocular camera," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 2797–2802, Apr. 2007.
- [9] P. Pinies and J. Tardos, "Large-scale slam building conditionally independent local maps: Application to monocular vision," *IEEE Transactions on Robotics*, vol. 24, pp. 1094–1106, Oct. 2008.
- [10] A. Mourikis and S. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 3565–3572, Apr. 2007.
- [11] K. Konolige and M. Agrawal, "Frameslam: From bundle adjustment to real-time visual mapping," *IEEE Transactions on Robotics*, vol. 24, pp. 1066–1077, Oct. 2008.
- [12] D. Burschka and E. Mair, "Direct pose estimation with a monocular camera," in *Proc. Robot Vision, Second International Workshop, RobVis 2008*, pp. 440–453, Feb. 2008.
- [13] O. Faugeras, Q.-T. Loung, and T. Papadopoulos, *The Geometry of Multiple Images*. The MIT Press, 2001.
- [14] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.
- [15] J. Farrell and M. Barth, *The Global Positioning System and Inertial Navigation*. McGraw-Hill, 1999.
- [16] S. Julier and J. Uhlmann, "A new extension of the kalman filter to nonlinear systems," in *Proc. of Signal Processing, Sensor Fusion, and Target Recognition VI*, pp. 182–193, Apr. 1997.
- [17] E. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC*, pp. 153–158, Oct. 2000.
- [18] T. Kailath, A. Sayed, and B. Hassibi, *Linear Estimation*. Prentice Hall, 2000.
- [19] R. Van der Merwe and E. Wan, "The square-root unscented kalman filter for state and parameter-estimation," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 3461–3464, May 2001.