Visual-based simultaneous localization and mapping and global positioning system correction for geo-localization of a mobile robot

2011 Meas. Sci. Technol. 22 124003

(http://iopscience.iop.org/0957-0233/22/12/124003)

Meas. Sci. Technol. **22** (2011) 124003 (9pp) doi:10.1088/0957-0233/22/12/124003

Visual-based simultaneous localization and mapping and global positioning system correction for geo-localization of a mobile robot

Sid Ahmed Berrabah[1,2], **Hichem Sahli**[2] **and Yvan Baudoin**[1]

[1] Royal Military Academy of Belgium (RMA), Av. de la Renaissance 30, B1000 Brussels, Belgium
[2] Vrije Universiteit Brussel (VUB), Pleinlaan 2, 1050 Brussels, Belgium

E-mail: sidahmed.berrabah@rma.ac.be

Received 15 February 2011, in final form 10 August 2011
Published 15 November 2011
Online at stacks.iop.org/MST/22/124003

Abstract
This paper introduces an approach combining visual-based simultaneous localization and mapping (V-SLAM) and global positioning system (GPS) correction for accurate multi-sensor localization of an outdoor mobile robot in geo-referenced maps. The proposed framework combines two extended Kalman filters (EKF); the first one, referred to as the integration filter, is dedicated to the improvement of the GPS localization based on data from an inertial navigation system and wheels' encoders. The second EKF implements the V-SLAM process. The linear and angular velocities in the dynamic model of the V-SLAM EKF filter are given by the GPS/INS/Encoders integration filter. On the other hand, the output of the V-SLAM EKF filter is used to update the dynamics estimation in the integration filter and therefore the geo-referenced localization. This solution increases the accuracy and the robustness of the positioning during GPS outage and allows SLAM in less featured environments.

Keywords: geo-localization, simultaneous localization and mapping

(Some figures in this article are in colour only in the electronic version)

1. Introduction

To be able to navigate in its environment, a mobile robot is required to infer its current position in relation to the outside world using onboard sensory readings. For outdoor applications, the global positioning system (GPS) could be used to compute the robot's position in a geo-referenced map of the environment. However, it is well known that GPS systems are subject to several sources of errors, among them, ionosphere and troposphere delays, signal multi-path, number of visible satellites, satellite geometry/shading, etc. A typical GPS receiver, for civil applications, provides 6–12 m accuracy, depending on the number of available satellites. This accuracy can be reduced to 1 m when using a differential GPS (DGPS) system which employs a second receiver at a fixed location to compute corrections to the GPS satellite measurements.

Several solutions have been proposed in the literature to increase the accuracy of GPS localization by integrating data from other sensors. In particular, inertial navigation systems (INS) [1–3] and/or wheel encoders [4, 5] have often been used. This integration usually makes use of a Kalman filter (KF). Based on an error model of the different navigation system parameters, a KF solution may be capable of providing a reliable estimate of the position, velocity and attitude components of the moving platform [6]. Such a solution for the integration of GPS and INS has been successfully used in practice. However, the accuracy of these systems decreases drastically during long outage of the GPS receiver.

On the other hand, for local navigation in unknown outdoor environments, simultaneous localization and mapping (SLAM) techniques have been developed allowing robots to build up a map of their environment while at the same time keeping a track of their current location [8, 11].

Combining GPS and SLAM has also been addressed in the literature. Lee *et al* [23] used the GPS and digital road map information as prior constraints to aid their SLAM algorithm in data association and loop closure. A similar idea was introduced in [24], where the authors built a stereo camera-based topological/metric hierarchical SLAM for vehicle localization in urban environments. When available, the data from GPS are fused with the visual estimation using a Kalman filter. Yang *et al* [26] developed a SLAM-aided GPS/INS navigation system. In their algorithm, if the GPS information is available, the SLAM-aided system works in the way of INS/GPS, and at the same time, online building and updating the landmark-based map using INS/GPS solution. If the GPS data are not available, the generated map is used to constrain the INS errors. In [26], Asmar introduced the VisSLAM approach combining vision and INS using an EKF filter.

In this study, we propose a different multi-sensor-based framework combining visual SLAM and integrated GPS/INS/Encoders filtering for outdoor robot localization. The filters are combined in a feedback loop. Compared to [24–26], the proposed method corrects the GPS measurement (using the SLAM output and INS and encoders data) before using it to localize the built map and the SLAM estimation is helped by the integration filter.

The following sections give a detailed description of the proposed framework and discuss the obtained results for each part of the algorithm.

2. Global algorithm

In this work, the used robot is the 'ROBUDEM' robot (figure 1), equipped with a camera, a GPS, an INS and wheel encoders. We define a local global coordinate system G (figure 1) formed from a plane tangent to the Earth's surface and fixed to a specific location with known geodetic coordinates (in our case, it is supposed to be the initial robot position for a null initialization of the covariance matrix in the SLAM process, see section 4). The X axis points toward the east, the Y axis points toward the north and the Z axis points vertically upward. We also define an inertial coordinate frame L related to the INS sensor, a coordinate system C for the camera and a platform (robot) frame R (figure 1). The axes of these frames are parallel with the XY plane parallel to the ground and the X axis points toward the robot direction. The Z axis points vertically upward. For geo-localization, a conventional coordinate frame called 'Earth-centered Earth-fixed (ECEF or ECF)' is used. This frame has its origin at the center of the Earth (figure 1). The X axis passes through the equator at the prime meridian. The Z axis passes through the North Pole. The Y axis can be determined by the right-hand rule to be passing through the equator at 90° longitude. The geodetic coordinates expressed in terms of latitude Φ, longitude Γ and altitude Ψ can be converted into ECEF coordinates (x^E, y^E, z^E) using the following formulas:

$$\begin{aligned} x^E &= (\beth + \Theta)\cos(\Psi)\cos(\Gamma) \\ y^E &= (\beth + \Theta)\cos(\Psi)\sin(\Gamma) \\ z^E &= (\beth(1-e^2) + \Theta)\sin(\Psi), \end{aligned} \tag{1}$$

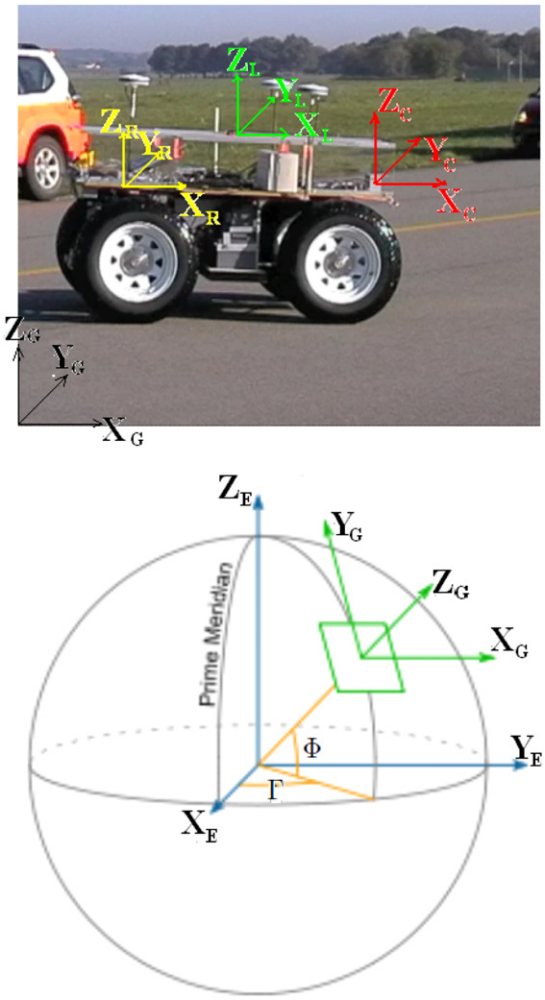


Figure 1. The ROBUDEM robot and coordinate frames.

where $\beth = a/\sqrt{(1 - e^2\sin^2(\Psi))}$ is the distance from the surface to the Z axis along the ellipsoid normal. a and e^2 are the semi-major axis and the square of the first numerical eccentricity of the ellipsoid, respectively ($a = 6356\,752.3142$ m and $e^2 = 6.694\,379\,990\,14 \times 10^{-3}$).

The conversion of the coordinates of a location (x^E, y^E, z^E) in the ECEF frame to the coordinates (x^G, y^G, z^G) in the local global coordinate G fixed to a location O with geodetic coordinates (Ψ, Γ, Θ) and the ECEF coordinates (x_O^E, y_O^E, z_O^E) is computed by

$$\begin{bmatrix} x^G \\ y^G \\ z^G \end{bmatrix} = \begin{bmatrix} -\mathrm{s}(\Gamma_O) & \mathrm{c}(\Gamma_O) & 0 \\ -\mathrm{s}(\Psi_O)\mathrm{c}(\Gamma_O) & -\mathrm{s}(\Psi_O)\mathrm{s}(\Gamma_O) & \mathrm{c}(\Psi_O) \\ \mathrm{c}(\Psi_O)\mathrm{c}(\Gamma_O) & \mathrm{c}(\Psi_O)\mathrm{s}(\Gamma_O) & \mathrm{s}(\Psi_O) \end{bmatrix} \times \begin{bmatrix} x^E - x_O^E \\ y^E - y_O^E \\ z^E - z_O^E \end{bmatrix}, \tag{2}$$

where $\mathrm{c}(\cdot)$ and $\mathrm{s}(\cdot)$ stand for $\cos(\cdot)$ and $\sin(\cdot)$, respectively.

In our application all measurements and computations are transformed into the G coordinate system. For simplicity, the subscript G is omitted in the following.

The state vector of the robot $\mathbf{x}_r$ is defined with the 3D position vector $\mathbf{r} = (x_r, y_r, z_r)$ in the world frame coordinates and the robot's orientations yaw, roll and pitch $(\omega_r, \theta_r, \varphi_r)$:

$$\mathbf{x}_r = \begin{bmatrix} x_r \\ y_r \\ z_r \\ \omega_r \\ \theta_r \\ \varphi_r \end{bmatrix}.$$

The dynamic model or motion model is the relationship between the robot's past state, $\mathbf{x}_r^{t-1}$, and its current state, $\mathbf{x}_r^t$, given a control input u^t:

$$\mathbf{x}_r^t = \mathbf{f}\left(\mathbf{x}_r^{t-1}, u^t, \mathbf{v}^t\right), \tag{3}$$

where $\mathbf{f}$ is a function representing the mobility, kinematics and dynamics of the robot (transition function) and $\mathbf{v}$ is a random vector describing the unmodeled aspects of the vehicle (process noise such as wheel sleep or odometry error).

The system dynamic model of the robot, considering the control u as identity, is given by

$$\mathbf{x}_r^t = \begin{bmatrix} x_r^t \\ y_r^t \\ z_r^t \\ \omega_r^t \\ \theta_r^t \\ \varphi_r^t \end{bmatrix} = \begin{bmatrix} x_r^{t-1} + (v^{t-1} + V)\cos\left(\omega_r^{t-1}\right)\Delta t \\ y_r^{t-1} + (v^{t-1} + V)\sin\left(\omega_r^{t-1}\right)\Delta t \\ z_r^{t-1} \\ \omega_r^{t-1} + \left(\dot{\omega}_r^{t-1} + \Omega\right)\Delta t \\ \theta_r^{t-1} \\ \varphi_r^{t-1} \end{bmatrix}, \tag{4}$$

where v and $\dot{\omega}$ are the linear and the angular velocities, respectively, and V and Ω are the Gaussian distributed perturbations to the robot's linear and angular velocity, respectively.

Figure 2 illustrates the proposed framework for robot localization, where two extended Kalman filters (EKF) are combined in a feedback loop. The first EKF, referred to as integration EKF (EKF-I), is dedicated to the improvement of the GPS localization based on data from an inertial navigation system (INS) and wheels' encoders. The second EKF implements the V-SLAM process. Both EKFs exchange motion parameters for better localization.

The built V-SLAM algorithm uses an EKF to represent a visual feature-based map. The linear v and angular $\dot{\omega}$ velocities in the dynamic model of the V-SLAM algorithm are given by the GPS/INS/Encoders integration process. On the other hand, the output of the V-SLAM is used to update the dynamics estimation in the EKF-I, and therefore the geo-referenced localization.

The proposed solution will allow increasing accuracy and robustness of the positioning during GPS outage as well as using fewer features for the V-SLAM.

3. GPS/INS/Encoders integration

The GPS measurements are called pseudo-ranges (instead of ranges) since the estimated times of transmission are corrupted

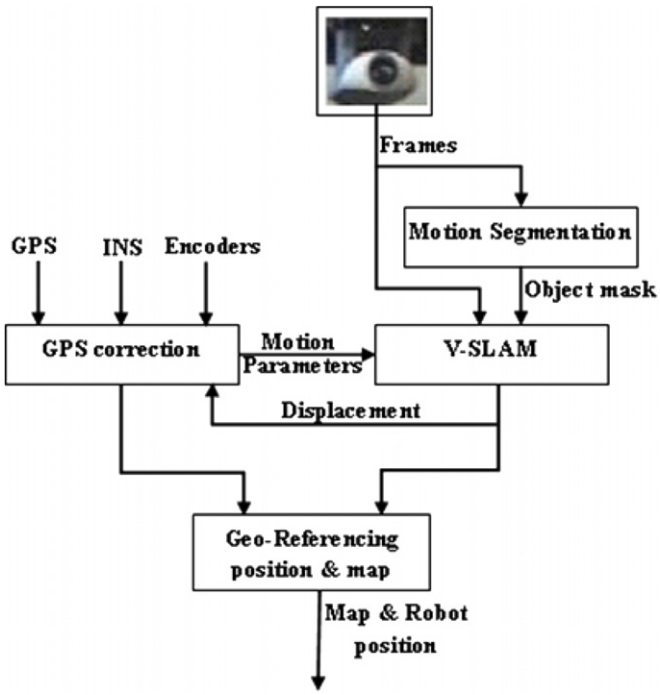


Figure 2. The proposed framework for robot localization.

by different biases. The positioning equations for n_s satellites in sight at time instant t can be defined as

$$r_i^t = \sqrt{\left(X_i^t - x_r^t\right)^2 + \left(Y_i^t - y_r^t\right)^2 + \left(Z_i^t\right)^2} + b^t + w_i^t \tag{5}$$

for $i = 1, \ldots, n_s$, where r_i^t is the pseudo-range between the GPS receiver and the ith satellite, $[X_i^t, Y_i^t, Z_i^t]^T$ is the position of the ith satellite, b^t is the GPS receiver clock offset in meters, w_i^t is the measurement error and $[x_r, y_r]$ is the vehicle position to be estimated (the vehicle altitude is $z_r = 0$ in our application).

The GPS clock offset dynamic model is defined by

$$\dot{b}^t = d^t + v_b^t, \qquad \dot{d}^t = v_d^t \tag{6}$$

where v_b^t and v_d^t are the noise on GPS measurements.

The inertial navigation system (INS) is a self-contained navigation technique in which measurements provided by accelerometers and gyroscopes are used to track the position and orientation of the robot relative to a known starting point, orientation and velocity. INS typically contains three orthogonal rate-gyroscopes and three orthogonal accelerometers, measuring angular velocity and linear acceleration, respectively.

Usually, INS can only provide an accurate solution for a short period of time. As the acceleration is integrated twice to obtain the position, any error in the acceleration measurement will also be integrated and will cause a bias on the estimated velocity and a continuous drift on the position estimate by the INS.

The accelerometers deliver a nongravitational acceleration (also referred to as the specific force f^R) and the gyrometers measure the rotation rate of the sensor cluster Ω^{LG} in order to keep track of the vehicle orientation.

The differential equations relating the measured quantities to the dynamics are defined as follows:

$$\dot{v}^{EG} = R^{RG} f^R + g^G - (\Omega^{EG} + 2\Omega^{LE})v^E - (\Omega^{LE})^2 p^G, \quad (7)$$

where R^{RG} is the rotation matrix from the R frame to the local geographic frame W, Ω^{LE} is the rotation rate from the L frame to the E frame, Ω^{EG} is the rotation rate from the E frame to the G frame, v^E is the velocity relative to the E frame and g^G is the gravitational acceleration.

The location p^G of the vehicle in the G frame is given by

$$\dot{p}^G = \begin{pmatrix} \dot{\Psi}_r \\ \dot{\Gamma}_r \end{pmatrix} = \begin{pmatrix} \frac{1}{R_\Psi} & 0 \\ 0 & \frac{1}{R_\Gamma \cos(\Psi)} \end{pmatrix}, \quad (8)$$

where Ψ_r and Γ_r are the latitude and longitude of the vehicle, R_Ψ is the Earth's radius of curvature in the meridian and R_Γ is the transverse radius.

The state model describing the INS error dynamics can be obtained by linearizing equations (7) and (8) around the INS estimation (see [22] for more details):

$$\begin{aligned} \delta\dot{p}^G &= S(\Omega^{EG}) \wedge \delta p^G + \delta\dot{v}^{EG} \\ \delta\dot{v}^{EG} &= S(\delta\rho) \wedge f^R - S(\Omega^{EG} + 2\Omega^{LE}) \wedge \delta v^{EG} \\ &\quad + \delta g^G - S(\delta\Omega^{EG} + 2\Omega^{LE}) \wedge v^{EG} \\ \delta\dot{\rho} &= -\delta\Omega^{LE} - S(\Omega^{LG}) \wedge \delta\rho + R^{RG}\delta\Omega^{LE}, \end{aligned} \quad (9)$$

where $S(\cdot)$ is the skew-symmetric matrix and $\wedge$ denotes the cross product.

The linear speed v and the yaw rate (angular velocity) $\dot{\omega}$ of the vehicle at time t can be computed based on the wheel encoders as follows:

$$\begin{aligned} v &= \frac{\alpha_r^t R_r + \alpha_l^t R_l}{2} \\ \dot{\omega} &= \frac{\alpha_r^t R_r + \alpha_l^t R_l}{L_{rl}}, \end{aligned} \quad (10)$$

where α_r^t and α_l^t are the angular velocities of the right and left rear wheels, respectively, and R_r and R_l their corresponding radii. L_{rl} is the distance between rear wheels.

The EKF-I filter prediction is done using equation (9) to estimate the state vector composed of the vehicle position $\mathbf{x}_r$, the linear and angular velocities, and the GPS bias term

$$\mathbf{x}_{\text{EKF-I}} = [\mathbf{x}_r, p^G, v, \dot{\omega}, b, d]^T.$$

For the update of the filter, the linear and angular velocities are estimated as an average between the sensor measurements and the V-SLAM estimations (as described in the following section).

4. Visual SLAM for localization

4.1. Visual SLAM formulation

The SLAM problem is tackled as a stochastic problem and it has been addressed with approaches based on Bayesian filtering. The most well-known Bayesian filters for treating the SLAM problem are (i) the extended Kalman filter (EKF) [7–9] where the belief is represented by a Gaussian distribution, and (ii) the particle filters [10, 11] where the belief is represented by multiple values (particles). Whenever a landmark is observed by the robot's on-board sensors, the system determines whether it has been already registered and updates the filter.

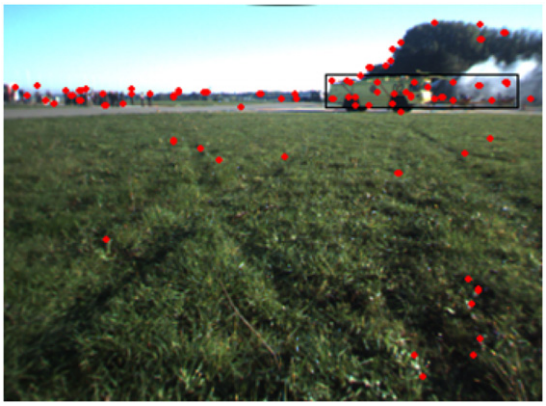


Figure 3. Features detected in a scene with moving objects.

Usually the features used in vision-based localization algorithms are salient and distinctive objects are detected from images. Typical features might include regions, edges, object contours, corners, etc. In our work, the map features are obtained using the SIFT feature detector [12]. These features are invariant to image scale, rotation and change in illumination [13].

To deal with the problem of SLAM in dynamic scenes with a moving object, we use a previously developed motion segmentation algorithm [14] to remove outlier features which are associated with moving objects. In other words, the detected features which correspond to the moving parts in the scene are not considered in the built map. The approach in [14] uses a Gaussian mixture model background subtraction approach to detect the moving objects' mask and a Markov random field framework to optimize the detected masks based on the space and time dependences that moving objects impose on a frame pixel. The algorithm starts by estimating and compensating the camera motion. In another paper, we will show how we exploit the estimated 3D motion in the SLAM process for the 2D camera motion compensation. For more details, the reader is referred to [14].

To deal with the reliability of the detected and tracked features, we use a bounding box around the moving objects (figure 3), and the newly detected features should be detected and matched in at least j consecutive frames (in our application, $j = 5$) before being added to the features' map.

Features are represented in the system state vector by their 3D location in the local world coordinate system G:

$$\mathbf{m}_i = (m_{1,i}, m_{2,i}, m_{3,i})^T.$$

The observation model of the EKF-SLAM is given by

$$\mathbf{z}^t = \left[z_1^t, z_2^t\right]^T = \mathbf{h}(\mathbf{m}^t) + \mathbf{w}^t, \quad (11)$$

where $\mathbf{z}^t$ is the observation vector at time t and $\mathbf{h}$ is the observation model. The vector $\mathbf{z}_i^t$ is an observation at instant t of the ith landmark location $\mathbf{m}_i^t$ relative to the robot's location

$\mathbf{x}_r^t$. Using a perspective projection, the observation model in the robot coordinate system is obtained as follows:

$$\mathbf{z}_i^t = \mathbf{h}\left(\mathbf{m}_i^t\right) = \begin{bmatrix} o_x + f\dfrac{{m_{1,i}^t}^R}{{m_{3,i}^t}^R} \\ o_y + f\dfrac{{m_{2,i}^t}^R}{{m_{3,i}^t}^R} \end{bmatrix}, \tag{12}$$

where o_x and o_y are the image center coordinates and f is the focal length of the camera.

$\mathbf{m}_i^R = \left(m_{1,i}^R, m_{2,i}^R, m_{3,i}^R\right)^T$ are the coordinates of the feature i in the robot coordinate frame R. They are related to $\mathbf{m}_i$ by

$$\mathbf{m}_i^R = \begin{pmatrix} \cos(\omega_r) & -\sin(\omega_r) & 0 \\ \sin(\omega_r) & \cos(\omega_r) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} m_{1,i}^t - x_r \\ m_{2,i}^t - y_r \\ m_{3,i}^t - h \end{pmatrix}, \tag{13}$$

where h is the height of the camera.

In EKF-based SLAM approaches, the environment is represented by a stochastic map $\mathfrak{M} = (\mathbf{x}, \mathbf{P})$, where $\mathbf{x}$ is the estimated state vector, consisting of the n_r states representing the robot, $\mathbf{x}_r^t$, and the n states describing the observed landmarks, $\mathbf{m}_i^t, i = 1, \ldots, n$, and $\mathbf{P}$ is the estimated covariance matrix, where all the correlations between the elements of the state vector are defined:

$$\mathbf{x}^t = \begin{bmatrix} \mathbf{x}_r^t \\ \mathbf{m}_1^t \\ \vdots \\ \mathbf{m}_n^t \end{bmatrix}$$

$$\mathbf{P}^t = \begin{bmatrix} \mathbf{P}_{rr}^t & \mathbf{P}_{r1}^t & \cdots & \mathbf{P}_{rn}^t \\ \mathbf{P}_{1r}^t & \mathbf{P}_{11}^t & \cdots & \mathbf{P}_{1n}^t \\ \vdots & \vdots & \ddots & \cdots \\ \mathbf{P}_{nr}^t & \mathbf{P}_{n1}^t & \cdots & \mathbf{P}_{nn}^t \end{bmatrix}. \tag{14}$$

The sub-matrices $\mathbf{P}_{rr}^t$, $\mathbf{P}_{ri}^t$ and $\mathbf{P}_{ii}^t$ are, respectively, the robot to robot, robot to feature, and feature to feature covariances. The sub-matrices $\mathbf{P}_{ij}^t$ are the feature to feature cross-correlations. $\mathbf{x}$ and $\mathbf{P}$ will change in dimension as features are added or deleted from the map.

The extended Kalman filter consists of two steps:

(a) The prediction step (equations (15)), which estimates the system state according to the state transition function $\mathbf{f}$ (equation (4)) and the covariance matrix $\mathbf{P}$ to reflect the increase in uncertainty in the state due to noise $\mathbf{Q}$ (unmodeled aspects of the system). The linear v and the angular $\dot{\omega}$ velocities are estimated by the GPS/INS/Encoder integration KEF filter:

$$\mathbf{x}^{t|t-1} = \begin{bmatrix} \mathbf{f}(\mathbf{x}_r^{t-1|t-1}, u = 0) \\ \mathbf{m}_1^{t-1|t-1} \\ \vdots \end{bmatrix},$$
$$\mathbf{P}^{t|t-1} = \mathbf{F}\mathbf{P}^{t-1|t-1}F^T + \mathbf{Q}^{t-1} \tag{15}$$

where $\mathbf{F} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}|_{\mathbf{x}^{t-1|t-1}} = \mathrm{diag}\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}_r}|_{\mathbf{x}_r^{t-1|t-1}}, I\right)$ is the Jacobian of $\mathbf{f}$ with respect to the state vector $\mathbf{x}$ and $\mathbf{Q}$ is the process noise covariance.

Considering a constant velocity model for the smooth camera motion:

$$\left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}_r}\right|_{\mathbf{x}_r^{t-1|t-1}} = \begin{bmatrix} 1 & 0 & -\sin\left(\omega_r^{t-1}\right)(v^{t-1} + V)\Delta t \\ 0 & 1 & \cos\left(\omega_r^{t-1}\right)(v^{t-1} + V)\Delta t \\ 0 & 0 & 1 \end{bmatrix}. \tag{16}$$

(b) The update step uses the current measurement to improve the estimated state, and therefore the uncertainty represented by $\mathbf{P}$ is reduced:

$$\mathbf{x}^{t|t} = \mathbf{x}^{t|t-1} + \mathbf{W}^t\varepsilon^t$$
$$\mathbf{P}^{t|t} = \mathbf{P}^{t|t-1} - \mathbf{W}^t\mathbf{S}^t\mathbf{W}^{t\,T}, \tag{17}$$

where

$$\mathbf{W}^t = \mathbf{P}^{t|t-1}\mathbf{H}^T(\mathbf{S}^t)^{-1}$$
$$\mathbf{S}^t = \mathbf{H}\mathbf{P}^{t|t-1}\mathbf{H} + \mathbf{U}^t \tag{18}$$
$$\varepsilon = \mathbf{z}^t - \mathbf{h}(\mathbf{x}^{t|t-1}).$$

$\mathbf{Q}$ and $\mathbf{U}$ are block-diagonal matrices (obtained empirically) defining the error covariance matrices characterizing the noise in the model and the observations, respectively. $\mathbf{H}$ is the Jacobian of the measurement model $\mathbf{h}$ with respect to the state vector. A measurement of feature $\mathbf{m}_i$ is not related to the measurement of any other feature so

$$\frac{\partial \mathbf{h}_i}{\partial \mathbf{x}} = \begin{bmatrix} \dfrac{\partial \mathbf{h}_i}{\partial \mathbf{x}_r} & 0 & \cdots & \dfrac{\partial \mathbf{h}_i}{\partial \mathbf{m}_i} & 0 & \cdots \end{bmatrix},$$

where $\mathbf{h}_i$ is the measurement model for the ith feature.

4.2. Initialization

Several approaches have been proposed for the estimation of the initial state of the EKF-SLAM. Deans [15] combined Kalman filter and bundle adjustment in filter initialization, obtaining accurate results at the expense of increasing filter complexity. In [8], Davison uses an A4 piece of paper as a landmark to recover the metric information of the scene. Then, whenever a scene feature is observed, a set of depth hypotheses are made along its direction. In subsequent steps, the same feature is seen from different positions reducing the number of hypotheses and leading to accurate landmark pose estimation. Solà *et al* [16] proposed a 3D bearing-only SLAM algorithm based on EKF filters, in which each feature is represented by a sum of Gaussians.

In our application, to estimate the 3D position of the detected features, we use an approach based on epipolar geometry. This geometry represents the geometric relationship between multiple viewpoints of a rigid body and it depends on the internal parameters and relative positions of the camera. The essence of the epipolar geometry is illustrated in figure 4 [24].

The fundamental matrix $\mathbb{F}$ (a 3×3 matrix of rank 2) encapsulates this intrinsic geometry. It describes the relationship between matching points: if a landmark $\tilde{\mathbf{M}}$ is imaged as $\mathbf{m}$ in the first view, and $\mathbf{m}'$ in the second, then the image points satisfy the relation $\mathbf{m}^T\mathbb{F}\mathbf{m}' = 0$ called the

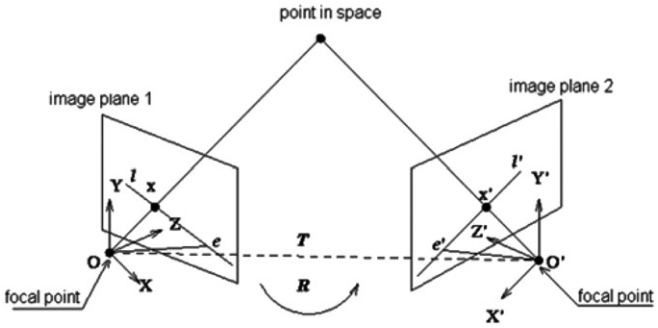


Figure 4. Illustration of the epipolar geometry.

epipolar constraint. $\mathbf{m}$ lies on the epipolar line $\mathbb{F}\mathbf{m}'$ and so the two rays back-projected from image points $\mathbf{m}$ and $\mathbf{m}'$ lie in a common epipolar plane. Since they lie in the same plane, they will intersect at one point. This point is the reconstructed 3D scene point $\mathbf{M}$.

Analytically, the depth of the 3D point corresponding to x and x' can be calculated by the following equation:

$$\mathbf{z} = \frac{(\mathbf{e} \times \mathbf{m})(\mathbf{m} \times \mathbf{m}')}{\| \mathbf{m} \times \mathbf{m}' \|^2} \tag{19}$$

where $\mathbf{e}$ is the epipole at the first view satisfying the relation $\mathbb{F}\mathbf{e} = 0$.

The fundamental matrix $\mathbb{F}$ is independent of scene structure and can be computed from correspondences of imaged scene points, without requiring knowledge of the cameras' internal parameters or relative pose. Given a set of n pairs of image correspondences $(\mathbf{m}_j, \mathbf{m}'_j)$, $j = 1, \ldots, n$, we compute the rotation matrix $\mathbf{R}$ between the two views and translation vector $\mathbf{t}$ such that the epipolar error (equation (20)) is minimized. For the minimization, we use the random sample Cconsensus (RANSAC) algorithm [21]:

$$\min_{\mathbb{F}} \sum_{j=1}^{n} \mathbf{m}'_j \mathbb{F} \mathbf{m}_j. \tag{20}$$

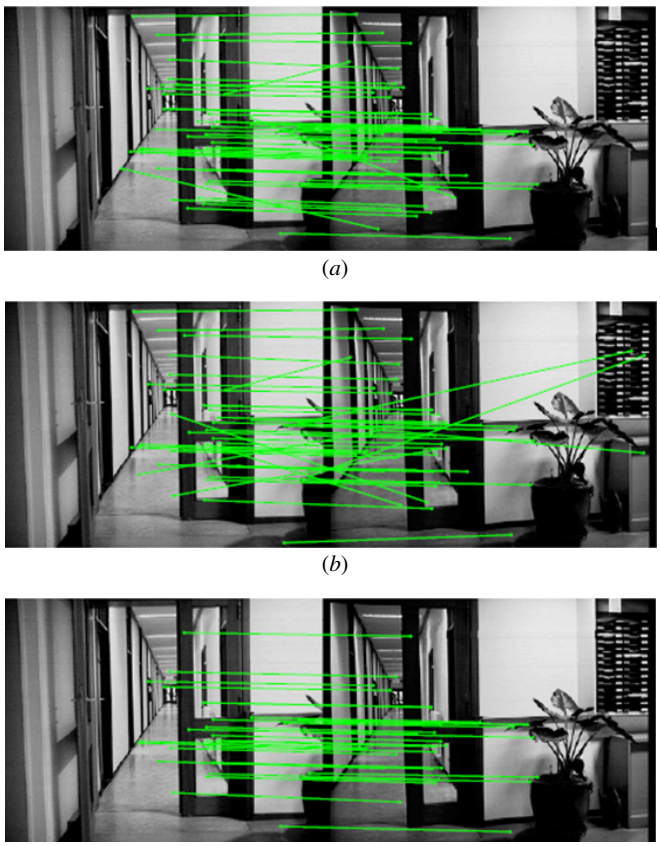
4.3. Feature matching

At step t, the onboard sensor obtains a set of measurements $\mathbf{z}_i^t$ $(i = 1, \ldots, k)$ of k environment features. Feature matching corresponds to data association, also known as the correspondence problem, which consists in determining the origin of each measurement, in terms of the map features $\mathbf{m}_j$, $j = 1, \ldots, n$. In our implementation, the measurement $\mathbf{z}_i^t$ can be considered corresponding to the feature j if the following equation is satisfied:

$$D^2 = D_{ij}^2 + D_{\text{desc}}^2 + D_{\text{epi}}^2 < th \tag{21}$$

where D_{ij}^2 is the Mahalanobis distance between the new detected feature i and the map features j, D_{desc}^2 is the Euclidean distance between the descriptor vectors of the features i and j, and D_{epi}^2 is the distance of the feature i from the epipolar line induced by the feature j.

Figure 5 illustrates the effectiveness and accuracy of the proposed approach for feature matching given by equation (21) (figure 5(*c*)), compared to other techniques using matching based on Mahalanobis distance with consistency hypothesis (figure 5(*a*)) and matching based on Euclidean distance between feature descriptors (figure 5(*b*)).



(*a*)

(*b*)

(*c*)

Figure 5. SIFT feature matching. (*a*) Feature matching based on the Mahalanobis distance with consistency hypothesis. (*b*) Feature matching based on the Euclidean distance between feature descriptors. (*c*) Feature matching based on equation (21).

4.4. SLAM in large-scale areas

One of the problems of the current state-of-the-art SLAM approaches and particularly vision-based approaches is mapping large-scale areas. Relevant shortcomings of this problem are, on the one hand, the computational burden, which limits the applicability of the EKF-based SLAM in large-scale real time applications and, on the other hand, the use of linearized solutions which compromises the consistency of the estimation process. The computational complexity of the EKF stems from the fact that the covariance matrix $\mathbf{P}$ represents every pairwise correlation between the state variables. Incorporating an observation of a single feature will necessarily have an effect on every other state variable. This makes the EKF computationally infeasible for SLAM in large environment.

Methods like network coupled feature maps [17], sequential map joining [18] and the constrained local submap filter (CRSF) [19] have been proposed to solve the problem of SLAM in large spaces by breaking the global map into submaps. This leads to a sparser description of the correlations between map elements. When the robot moves out of one

submap, it either creates a new submap or relocates itself in a previously defined submap. By limiting the size of the local map, this operation is constant time per step. Local maps are joined periodically into a global absolute map in an $O(N^2)$ step. Each approach reduces the computational requirement of incorporating an observation to constant time. However, these computational gains come at the cost of slowing down the overall rate of convergence.

The constrained relative submap filter [19] proposes to maintain the local map structure. Each map contains links to other neighboring maps, forming a tree structure (where loops cannot be represented). The method converges by revisiting the local maps and updating the links through correlations. On the other side, in the hierarchical SLAM [20], the links between local maps form an adjacency graph. This method allows us to reduce the computational time and memory requirements and to obtain accurate metric maps of large environments in real time.

To solve the problem of SLAM in large spaces, in our study, we propose a procedure to break the global map into submaps by building a global representation of the environment based on several size-limited local maps built using the previously described approach. The global map is a set of robot positions where new local maps started (i.e. the base references of the local maps). The base frame for the global map is the robot position at instant t_0.

Each local map is built as follows: at a given instant t_k, a new map is initialized using the current vehicle location, $\mathbf{x}_r^{t_k}$, as the base reference $B_k = \mathbf{x}_r^{t_k}$, $k = 0, 1, \ldots$ being the local map order. Then, the vehicle performs a limited motion acquiring sensor information about the L_i neighboring environment features.

The kth local map is defined by

$$\mathfrak{M}_k = (\mathbf{x}_k, \mathbf{P}_k),$$

where $\mathbf{x}_k$ is the state vector in the base reference B_k of the L_k detected features and $\mathbf{P}_k$ is their covariance matrix estimated in B_k.

The decision to start building a new local map at an instant t_k is based on two criteria: the number of features in the current local map and the scene cut detection result. The instant t_k is called a key instant. In our application, we defined two thresholds for the number of features in the local maps: a lower Th^- and a higher Th^+ threshold. A key instant is selected if the number of features n^k in the current local map k is bigger than the lower threshold and a scene cut has been detected or the number of features has reached the higher threshold. This allows keeping reasonable dimensions of the local maps and avoiding building too small maps.

Formally, the global map is defined as

$$\mathfrak{M}_G^B = \left(\bar{\mathbf{x}}_r^0, \bar{\mathbf{x}}_r^1, \bar{\mathbf{x}}_r^2, \ldots\right),$$

where $\bar{\mathbf{x}}_r^k$ are the robot coordinates in B_0, where it decides to build the local map $\mathfrak{M}_k$ at instant t_k:

$$\begin{pmatrix} \bar{\mathbf{x}}_r^k \\ 1 \end{pmatrix} = \mathcal{T}_{k\to 0} \cdot \begin{pmatrix} \mathbf{x}_r^k \\ 1 \end{pmatrix} \tag{22}$$

$t_0 = 0$ and $\bar{\mathbf{x}}_r^0 = \bar{\mathbf{x}}_r^{t_0} = (0, 0, 0)$.

Figure 6. Closing the loop.

The transformation matrix $\mathcal{T}_{k\to 0}$ is obtained by successive transformations:

$$\mathcal{T}_{k\to 0} = \mathcal{T}_{1\to 0}.\mathcal{T}_{2\to 1} \ldots \mathcal{T}_{k\to k-1}, \tag{23}$$

where $\mathcal{T}_{i\to i-1} = (\mathbf{R}|\mathbf{t})$ is the transformation matrix corresponding to the rotation $\mathbf{R}$ and translation $\mathbf{t}$ between B_i and B_{i-1}:

$$\mathcal{T}_{i\to i-1} = \begin{pmatrix} \cos\left(\omega_r^{t_i}\right) & -\sin\left(\omega_r^{t_i}\right) & 0 & x_r^{t_i} \\ \sin\left(\omega_r^{t_i}\right) & \cos\left(\omega_r^{t_i}\right) & 0 & y_r^{t_i} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{24}$$

For feature matching at instant t, the robot uses the local map with the closest base frame to its current location:

$$\arg\min_i \left(\bar{\mathbf{x}}_r^k - \bar{\mathbf{x}}_r^t\right), \tag{25}$$

where $\bar{\mathbf{x}}_r^t$ is the robot position at instant t in B_0.

The local maps are considered as nodes in a topological representation. Based on its current position, the robot selects the local map on which the feature matching will be done. If a matching is detected the two local maps are fused in one local map. Since the relative reference frames of both maps are known, the main goal of the algorithm is to transform one of the maps and its features into the reference system of the other one:

$$\mathfrak{M}_{i+j} = (\mathbf{x}_{i+j}, \mathbf{P}_{i+j}), \tag{26}$$

where $\mathbf{x}_{i+j}$ and $\mathbf{P}_{i+j}$ represent the state vector and the covariance resultant of the fusion of the maps $\mathfrak{M}_i$ and $\mathfrak{M}_j$ in the reference frame of the map $\mathfrak{M}_i$.

5. Experimental results

Figure 6 shows an example for the detection of loops using the SLAM process. In this example, the robot wanders twice across a defined path (real path drawn in red in the figure). At the first round, the position error exceeds 5 m and the uncertainty around the robot position reaches 6.2 m (cyan ellipses in the figure). After loop detection, the uncertainty is

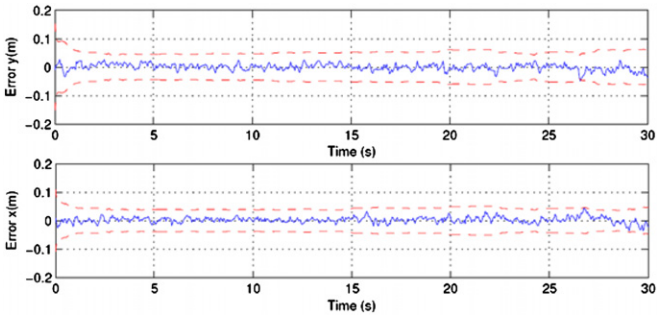


Figure 7. Robot position errors and the corresponding 2σ variance bounds.

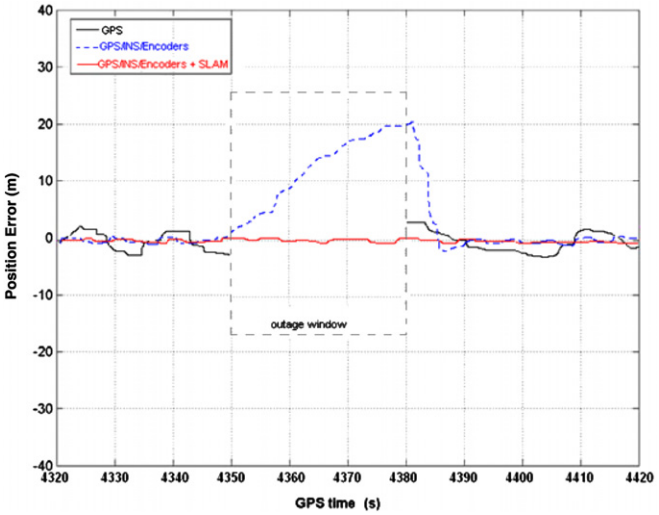


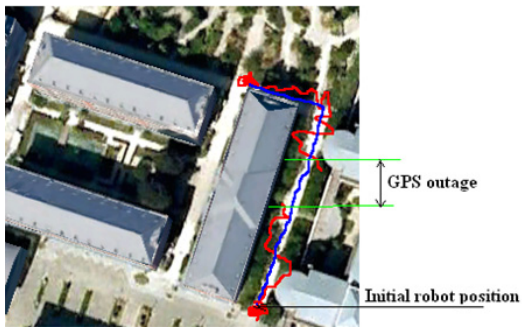
Figure 8. Robot localization in the case of GPS outage.

reduced. During the second round, the position error is limited to a maximum of 1 m and the uncertainty to a maximum of 2 m (magenta ellipses in the figure) before the detection of the closure of the loop.

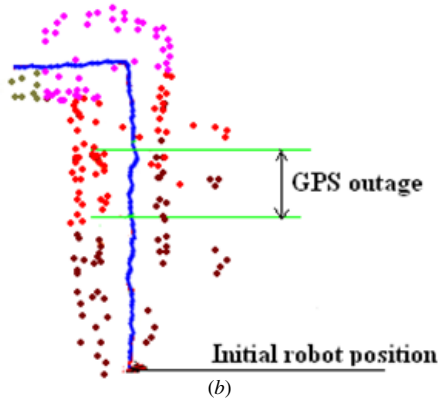
Figure 7 represents the robot position error and its corresponding 2σ variance bounds obtained by the proposed algorithm. Position errors are plotted as x and y distances of the robot location.

The proposed framework has been tested in real environments. Different GPS outages were simulated and analyzed. Figure 8 illustrates an example where the GPS signal was lost for 30 s. The black curve shows that the GPS localization error is 3.6 m. The dashed blue curve shows that even if the integration of GPS, INS and wheels encoders data reduces the error on the robot position to less than 1 m, it is not reliable during the GPS outage where the error grows continuously, while the proposed framework combining GPS/INS/Encoders localization and visual SLAM localization remains stable even during the GPS outage (red curve).

Figure 9 shows an example of the robot localization in a real environment. The blue/light curve represents the obtained robot path using the proposed algorithm, and the red/dark curve is the GPS data. The initial GPS position is the mean of the GPS measurements during a few minutes of initialization. Figure 9(*b*) represents the built local maps. Each local map



(*a*)



(*b*)

Figure 9. Robot localization in a real environment. (*a*) Robot path superimposed on a geo referenced map. (*b*) The built local maps.

is represented by a different color. In this experiment, the maximum number of features in the local maps is fixed to 60 features.

6. Conclusion

In this paper, we presented an algorithm for robot localization in georeferenced images. The proposed algorithm combines two localization techniques, one based on a GPS/INS/Wheel encoders integration approach and the other based on a visual SLAM approach.

The obtained results are interesting and for a future work we want to constrain the feature matching for the closure of the loop based on global positioning and study the influence of the GPS outage on the closing of the loop.

References

[1] Wolf R, Eissfeller B and Hein G W 1997 A Kalman filter for the integration of a low cost INS and an attitude GPS *Proc. Int. Symp. on Kinematic Systems in Geodesy, Geomatics and Navigation* pp 143–50

[2] Shin E H 2001 Accuracy improvement of low cost INS/GPS for land applications *MSc Thesis* Department of Geomatics Engineering, University of Calgary

[3] Abdel-Hamid W 2005 Accuracy enhancement of integrated MEMS IMU/GPS systems for land vehicular navigation applications *PhD Thesis* Department of Geomatics Engineering, University of Calgary

[4] Spangenberg M, Calmettes V and Tourneret J-Y 2007 Fusion of GPS, INS and odometric data for automotive navigation *15th European Signal Processing Conf. (EUSIPCO 2007) (Poznan, Poland, 3–7 September 2007)* pp 886–90

[5] Gao J, Petovello M G and Cannon M E 2006 Development of precise GPS/INS/wheel speed sensor/yaw rate sensor integrated vehicular positioning system *Proc. ION NTM-06 (Monterey, CA, January 2006)* pp 1–13
[6] El-Sheimy N 2004 Inertial surveying and INS/GPS integration *ENGO 623 Lecture Notes* Geomatics Department, University of Calgary
[7] Folkesson J, Jensfelt P and Christensen H 2005 Graphical SLAM using vision and the measurement subspace *IEEE/JRS Int. Conf. on Intelligent Robotics and Systems (IROS) (Edmonton, August 2005)* pp 325–30
[8] Davison J, Reid I D, Molton N D and Stasse O 2007 MonoSLAM: real-time single camera SLAM *IEEE Trans. Pattern Anal. Mach. Intell.* **29** 1052–67
[9] Fenwick J W, Newman P M and Leonard J J 2002 Cooperative concurrent mapping and localization *IEEE Int. Conf. on Robotics and Automation (Washington)* pp 1810–7
[10] Sim R, Elinas P, Griffin M and Little J J 2005 Vision-based SLAM using the Rao–Blackwellised particle filter *Proc. IJCAI Workshop on Reasoning with Uncertainty in Robotics* pp 9–16
[11] Stachniss C, Grisetti G and Burgard W 2005 Recovering particle diversity in a Rao–Blackwellized particle filter for slam after actively closing loops *IEEE Int. Conf. on Robotics and Automation* pp 667–72
[12] Lowe D G 2004 Distinctive image features from scale-invariant keypoints *Int. J. Comput. Vis.* **60** 91–110
[13] Mikolajczyk K and Schmid C 2003 A performance evaluation of local descriptors *IEEE Comput. Sci. Conf. on Computer Vision and Pattern Recognition, CVPR '03* vol 2 pp 257–64
[14] Berrabah S A, De Cubber G, Enescu V and Sahli H 2006 MRF-based foreground detection in image sequences from a moving camera *Proc. Int. Conf. on Image Processing, ICIP2006 (Atlanta, GA, 8–11 October 2006)* pp 1125–8
[15] Deans M and Hebert M 2000 Experimental comparison of techniques for localization and mapping using a bearing-only sensor *7th Int. Symp. on Experimental Robotics (Honolulu, HI, December 2000)* vol 271 pp 395–404
[16] Solà J, Lemaire T, Devy M, Lacroix S and Monin A 2005 Delayed vs undelayed landmark initialization for bearing only SLAM *Proc. IEEE Int. Conf. on Robotics and Automation Workshop on SLAM (Barcelona, Spain, April 2005)* pp 2499–504
[17] Bailey I 2002 Mobile robot localization and mapping in extensive outdoor environments *PhD Thesis* Australian Centre for Field Robotics, University of Sydney, Australia
[18] Trados J D, Neira J, Newman P and Leonard J 2002 Robust mapping and localization in indoor environments using sonar data *Int. J. Robot. Res.* **21** 311–30
[19] Williams S B 2001 Efficient solutions to autonomous mapping and navigation problems *PhD Thesis* Australian Centre for Field Robotics, University of Sydney, Australia
[20] Estrada C, Neira J and Tardos J D 2005 Hierarchical SLAM: real-time accurate mapping of large environments *IEEE Trans. Robot.* **21** 588–96
[21] Nistr D 2003 Preemptive RANSAC for live structure and motion estimation *Proc. IEEE Int. Conf. on Computer Vision (ICCV'2003)* pp 199–206
[22] Farrel J A and Barth M 1999 *The Global Positioning System and Inertial Navigation* (New York: McGraw-Hill)
[23] Lee K W, Wijesoma S and Guzman J I 2007 A constrained SLAM approach to robust and accurate localization of autonomous ground vehicles *Robot. Auton. Syst.* **55** 527–40
[24] Schleicher D, Bergasa L M, Ocaña M, Barea R and López E 2009 Real-time hierarchical GPS aided visual SLAM on urban environments 2009 *IEEE Int. Conf. on Robotics and Automation (Kobe, Japan, 12–17 May 2009)* pp 4381–6
[25] Cao M-L and Cui P-Y 2007 Simultaneous localization and mapping aided INS/GPS navigation system *J. Chin. Inertial Technol.* **4** 4–13
[26] Asmar D 2006 Vision-inertial SLAM using natural features in outdoor environments *PhD Thesis* University of Waterloo, Canada