Journal Title
doi:0000000000

Compressed Pseudo-SLAM: Pseudoranges-Integrated Compressed Simultaneous Localisation and Mapping for UAV Navigation

Jonghyuk Kim[1], Jose Guivant[2], Martin L. Sollie[3], Torleiv H. Bryne[3] and Tor Arne Johansen[3]

[1]*Centre for Autonomous Systems, University of Technology Sydney, Australia* (E-mail: jonghyuk.kim@uts.edu.au)
[2]*School of Mechanical and Manufacturing Engineering, University of New South Wales, Australia* (E-mail: j.guivant@unsw.edu.au)
[3]*Centre for Autonomous Marine Operations and Systems, Norwegian University of Science and Technology, Norway*
(E-mail: {martin.l.sollie, torleiv.h.bryne, tor.arne.johansen}@ntnu.no)

This paper addresses the fusion of the pseudorange/pseudorange rate observations from the global navigation satellite system and the inertial-visual simultaneous localisation and mapping (SLAM) to achieve reliable navigation of unmanned aerial vehicles. This work extends the previous work on a simulation-based study (Kim et al., 2017) to a real-flight dataset collected from a fixed-wing UAV platform. The dataset consists of measurements from visual landmarks, an inertial measurement unit, and pseudorange and pseudorange rates. We propose a novel all-source navigation filter, termed a compressed pseudo-SLAM, which can seamlessly integrate all-available information in a computationally efficient way. In this framework, a local map is dynamically defined around the vehicle, updating the vehicle and local landmark states within the region. A global map includes the rest of the landmarks and is updated at a much lower rate by accumulating (or compressing) the local-to-global correlation information within the filter. It will show that the horizontal navigation error is effectively constrained with 1 satellite vehicle and 1 landmark observations. The computational cost will be analysed, demonstrating the efficiency of the method.

KEYWORDS

Aerial SLAM. Compressed filtering. Pseudorange integration. UAV Navigation.

1. INTRODUCTION. Autonomous unmanned aerial vehicles (UAVs) have attracted much attention from industries and defence over the last several decades. With the advances in the low-cost inertial sensor technology and global navigation satellite system (GNSS), the navigation solution (or position, velocity, attitude, and time) can be accurately estimated by integrating the information. This data fusion approach has enabled higher-level

autonomy, such as intelligent path-planning and decision-making (Parkinson et al., 1997; Kim et al., 2003; Skulstad et al., 2015).

One of the key challenges in achieving reliable navigation solutions has been in handling the signal interferences, both intentional (such as jamming or spoofing) or unintentional (multipath or blockage), which can degrade the satellite-based aiding to the low-cost inertial system. This interference problem has been addressed at various levels, such as the low-level design of the signal tracking-loop to the high-level fusion of redundant information. Recently, simultaneous localisation and mapping (SLAM) has drawn much attention as an alternative aiding means to the inertial systems. For example, (Kim and Sukkarieh, 2007) demonstrated the integration of inertial measurement unit (IMU) and inertial navigation system (INS) as the core odometry together with a visual perception sensor on a UAV platform. (Nützi et al., 2011) integrated monocular vision and inertial sensor on a drone platform addressing the monocular scale-ambiguity problem. (Li and Mourikis, 2013) showed visual-inertial odometry and SLAM by utilising a sliding-window based bundle-adjustment optimisation on a 6 degrees-of-freedom vehicle. (Sjanic et al., 2017) applied the expectation-maximisation technique for visual-inertial navigation of fixed-wing flight data. Recently, (Vidal et al., 2018) demonstrated a high-speed drone by combining an event camera and inertial observations.

Although the abovementioned approaches have been quite promising, achieving reliable and efficient SLAM solutions on high-speed UAV platforms has a limited success, which is largely due to the low-quality inertial odometry and the high computational complexity inherent to SLAM. In particular, the flight dataset we consider in this work has very sparse landmark observations (typically less than 3 landmarks per camera frame. Thus aiding the inertial navigation solution using the sparse visual observations under a reduced number of satellite vehicles becomes highly challenging.

To address this problem, we propose a novel all-source navigation filter, termed compressed pseudo-SLAM (CP-SLAM), which can seamlessly integrate all-available observations in a computationally efficient way. The closest work to ours is (Williams and Crump, 2012) which demonstrated the all-source navigation strategy on a UAV platform by fusing GNSS, IMU, and camera information within the SLAM framework. However, the SLAM architecture relies on a *standard* extended Kalman filter (EKF), which has a high-computational complexity of $\mathcal{O}(N^2)$ with N being the total number of registered landmarks. In our approach, a local map is dynamically defined around the vehicle, and only the local landmarks within the region are actively updated within the filter. The global landmarks are only updated at a much lower rate by compressing the local-to-global correlation information. Due to this compressed filtering approach, the complexity reduces to $\mathcal{O}(L^2)$ with L being the number of local landmarks, which is much smaller than the total number N. This work extends the previous work by the authors (Kim et al., 2017), which was a simulation-based study. The contributions of this work are twofold:

- Demonstration of the CP-SLAM for a high-speed, fixed-wing UAV dataset.
- Evaluation of the CP-SLAM performance with a reduced number of satellites and sparse visual observations.

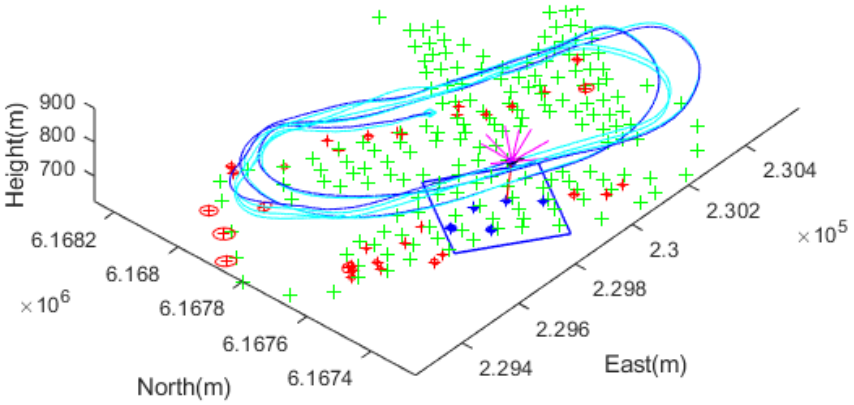


Figure 1. A snapshot[1] of the compressed pseudo-SLAM (CP-SLAM) for fixed-wing UAV navigation. The line-of-sight vectors to 9 satellite vehicles are shown as lines pointing upwards (in magenta) from the vehicle, and one visual feature observation is shown downwards (in red). The blue box shows an adaptive local-map boundary including active landmarks with uncertainty ellipses (in blue), while the red ellipses are for the global (inactive) landmark states. As the ground truth, surveyed landmark locations are shown as green '+' marks, as well as the onboard GPS/INS solution in cyan colour.

Figure 1 illustrates a snapshot (a movie clip is available[1]) of the CP-SLAM operation, showing GNSS pseudorange measurements from satellite vehicles shown as magenta-coloured lines pointing upwards from the vehicle, a visual-landmark observation as a red-coloured line pointing downwards from the vehicle, along a racehorse trajectory. Onboard navigation solutions recorded from a loosely-coupled GPS/INS system are also shown in cyan, together with the surveyed landmark positions ('+' marks in green). The key features in this work are 1) the seamless aiding of inertial navigation from the pseudoranges/pseudorange rates and the visual observations in a unified framework, and 2) the computationally-efficient compressed filtering, partitioning the map into local (in a blue box in the plot) and global (in red) ones. Thanks to the compressed approach, the global map is updated much less frequently whenever the local map is redefined, providing the computational benefits compared to the full-rate global update. This is the first UAV flight result integrating the pseudoranges and visual observations in a compressed filtering framework to the best of our knowledge.

The remainder of the article is outlined as follows. Section 2 provides the system models for the nonlinear state transition and observation used in this work, and Section 3 details the compressed pseudo-SLAM with a compressed update of the local and global map. Experimental results are provided in Section 4, analysing the filter performance and the computational complexity. Section 5 will conclude with future direction.

[1] A movie is available at https://youtu.be/sa4WTEnFd9s

2. SYSTEM MODELS. A continuous-time stochastic dynamic system with a nonlinear transition model $\mathbf{f}(\cdot)$ and an observation model $\mathbf{h}(\cdot)$ can be written as

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t)) \tag{1}$$

$$\mathbf{z}(t) = \mathbf{h}(\mathbf{x}(t), \mathbf{v}(t)), \tag{2}$$

where $\mathbf{x}(t)$ and $\mathbf{u}(t)$ are the state and control input vectors at time t, respectively, with $\mathbf{w}(t)$ being the process noise with a noise strength matrix $\mathbf{Q}$, and $\mathbf{z}(t)$ is the observation vector with $\mathbf{v}(t)$ being the observation noise with a strength matrix of $\mathbf{R}$.

In the inertial-based SLAM, the state vector consists of a vehicle state which represents the kinematics of the inertial navigation, inertial sensor errors, and map landmark states. In this work, a 17-state model is designed to estimate the vehicle position and attitude (6), velocity and sensor biases (9), receiver clock states (2), as well as the variable-size map states. The control input is the IMU measurements and drives the inertial kinematic equations. Thus the state vector $\mathbf{x} = (\mathbf{x}_v^T, \mathbf{m}^T)^T$[2] becomes,

$$\mathbf{x}_v = (\mathbf{p}^n, \mathbf{v}^n, \boldsymbol{\psi}^n, \mathbf{b}_a^b, \mathbf{b}_g^b, ct_b, ct_d) \tag{3}$$

$$\mathbf{m} = (\mathbf{m}_1^n, \mathbf{m}_2^n, \cdots, \mathbf{m}_N^n) \tag{4}$$

$$\mathbf{u} = (\mathbf{f}^b, \boldsymbol{\omega}^b), \tag{5}$$

where

- Vehicle position in the navigation frame[3] $\mathbf{p}^n = (x, y, z)$
- Vehicle velocity $\mathbf{v}^n = (v_x, v_y, v_z)$
- Vehicle attitude (Euler angles in roll, pitch and yaw) $\boldsymbol{\psi}^n = (\phi, \theta, \psi)$
- Accelerometer bias in the body frame[4] $\mathbf{b}_a^b = (b_{ax}, b_{ay}, b_{az})$
- Gyroscope bias $\mathbf{b}_g^b = (b_{gx}, b_{gy}, b_{gz})$
- Receiver clock bias ct_b with c being the speed of light
- Receiver clock drift ct_d
- Map position $\mathbf{m}_i^n = (m_{ix}, m_{iy}, m_{iz})$
- Accelerometer measurement $\mathbf{f}^b = (f_x, f_y, f_z)$
- Gyroscope measurement $\boldsymbol{\omega}^b = (\omega_x, \omega_y, \omega_z)$.

2.1. *Dynamic Model.* The system dynamic model consists of the kinematic equations of the inertial navigation driven by the IMU measurements which are the specific force (or the sum of dynamic acceleration of the vehicle and gravity) and angular rate. The IMU bias errors are modelled as a random walk process, while the map states are modelled as a

[2] The transpose notation is omitted henceforth for simplicity

[3] A superscript-n denotes a navigation frame. In this work, MGA94 (Map Grid of Australia 1994) is used as a local-fixed, local-tangent frame with North, East and Down (NED) (m)

[4] A superscript-b denotes a body frame attached to the vehicle.

random constant due to their stationary nature, yielding,

$$\dot{\mathbf{p}}^n = \mathbf{v}^n \tag{6}$$

$$\dot{\mathbf{v}}^n = \mathbf{C}_b^n(\mathbf{f}^b - \mathbf{b}_a^b) - 2\boldsymbol{\omega}_{ie}^n \times \mathbf{v}^n + \mathbf{g}^n(\mathbf{p}^n) \tag{7}$$

$$\dot{\boldsymbol{\psi}}^n = \mathbf{E}_b^n(\boldsymbol{\omega}^b - \mathbf{b}_g^b) \tag{8}$$

$$\dot{\mathbf{b}}_a^b = \mathbf{w}_a, \quad \dot{\mathbf{b}}_g^b = \mathbf{w}_g \tag{9}$$

$$\dot{ct}_b = ct_d, \quad \dot{ct}_d = w_d \tag{10}$$

$$\dot{\mathbf{m}}_{1\cdots N}^n = \mathbf{0}, \tag{11}$$

where

- $\boldsymbol{\omega}_{ie}^n$ is the Earth rotation rate in the navigation frame
- $\mathbf{g}^n(\mathbf{p}^n)$ is the gravitational acceleration
- $\mathbf{w}_a, \mathbf{w}_g, w_d$ are noise processes for accelerometers, gyroscopes, and receiver clock drift respectively
- $\mathbf{C}_b^n$ and $\mathbf{E}_b^n$ are the direction cosine matrix transforming a vector in the body frame to the one in the navigation frame, and the matrix transforming a body rate to an Euler angle rate, respectively,

$$\mathbf{C}_b^n = \begin{bmatrix} c_\theta c_\psi & -c_\phi s_\psi + s_\phi s_\theta c_\psi & s_\phi s_\psi + c_\phi s_\theta c_\psi \\ c_\theta s_\psi & c_\phi c_\psi + s_\phi s_\theta s_\psi & -s_\phi c_\psi + c_\phi s_\theta s_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix}, \; \mathbf{E}_b^n = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{bmatrix},$$

  where $s_{(\cdot)}$, $c_{(\cdot)}$, and $t_{(\cdot)}$ are shorthand notations for $\sin(\cdot)$, $\cos(\cdot)$, and $\tan(\cdot)$, respectively.

Please note that this utilises a simplified version of the strap-down inertial navigation equations, aiming for the low-cost inertial system applications. The effects of the Earth curvature on the attitude calculation are not included, although it will be straightforward to add the curvature terms in the model.

2.2. *Observation Model.* In all-source aiding strategy, the observations consist of all-available aiding information which are, in this work, a set of pseudoranges (ρ) and pseudorange rates ($\dot{\rho}$) measured from a GNSS receiver, and/or a set of range r, bearing ϕ and elevation θ observations from a camera sensor,

$$\rho_i = \sqrt{(X_i - x)^2 + (Y_i - y)^2 + (Z_i - z)^2} + ct_b + v_\rho \tag{12}$$

$$\dot{\rho}_i = l_{xi}(V_{xi} - v_x) + l_{yi}(V_{yi} - v_y) + l_{zi}(V_{zi} - v_z) + \dot{ct}_b + v_{\dot{\rho}} \tag{13}$$

$$r_j = \sqrt{(m_{jx} - x)^2 + (m_{jy} - y)^2 + (m_{jz} - z)^2} + v_r \tag{14}$$

$$\phi_j = \arctan\frac{(m_{jy} - y)}{(m_{jx} - x)} + v_\phi \tag{15}$$

$$\theta_j = \arctan\frac{(m_{jz} - z)}{\sqrt{(m_{jx} - x)^2 + (m_{jy} - y)^2}} + v_\theta, \tag{16}$$

where (X_i, Y_i, Z_i) denotes the i^{th} satellite vehicle's position computed from the ephemeris data, which is converted from the ECEF (Earth-Centred, Earth-Fixed) frame to the navigation frame, with v_ρ being the noise process, and (l_{xi}, l_{yi}, l_{zi}) is the line-of-sight vector of the i^{th} SV seen from the vehicle which projects the relative velocity along the line-of-sight direction. For the range, bearing and elevation measurement, we first compute the relative position vector $((m_{jx} - x), (m_{jy} - y), (m_{jz} - z))$ of the j^{th}-landmark seen from the vehicle, and then transform it to the polar coordinate quantities. $(v_\rho, v_{\dot{\rho}}, v_r, v_\phi, v_\theta)$ represent the observation noise processes. A new landmark position is initialised by utilising the inverse function of Eqs. (14) to (16) with the estimated vehicle state and observation information.

3. COMPRESSED PSEUDO-SLAM. Using the state and observation models described in the previous section, and their corresponding linearised state and observation models $(\mathbf{F}, \mathbf{H})$, a *standard* extended Kalman filter can predict and update the state and covariance matrix $(\mathbf{P})$ as detailed in (Kim and Sukkarieh, 2007). In a *compressed* filtering framework, the state vector is further partitioned into 1) a local map state including the vehicle state and local map, and 2) a global map for the remainder landmarks (Kim et al., 2017; Guivant, 2017). The underlying key assumption is that the cross-correlation information is *only* dependent on the local states, thus enabling the accumulation (thus compression) during the local filtering cycles. Consequently the compressed cross-correlation can be used to update the global state at a much lower rate. This assumption is quite valid for the downward-looking camera configuration as in this work where the camera field-of-view naturally defines the minimum boundary of the local map. The partitioned state estimates and covariance matrix become,

$$\left[\begin{array}{c} \hat{\boldsymbol{x}}_L \\ \hline \hat{\boldsymbol{x}}_G \end{array}\right] = \left[\begin{array}{c} \hat{\boldsymbol{x}}_v \\ \hat{\boldsymbol{m}}_{\boldsymbol{L}} \\ \hline \hat{\boldsymbol{m}}_G \end{array}\right] \tag{17}$$

$$\left[\begin{array}{c|c} \mathbf{P}_{LL} & \mathbf{P}_{LG} \\ \hline \mathbf{P}_{GL} & \mathbf{P}_{GG} \end{array}\right] = \left[\begin{array}{cc|c} \mathbf{P}_{vv} & \mathbf{P}_{vm} & \mathbf{P}_{LG} \\ \mathbf{P}_{mv} & \mathbf{P}_{mm} & \\ \hline \multicolumn{2}{c|}{\mathbf{P}_{GL}} & \mathbf{P}_{GG} \end{array}\right], \tag{18}$$

where the local estimate $\hat{\boldsymbol{x}}_L$ includes the vehicle $\hat{\boldsymbol{x}}_v$ and local landmarks $\hat{\boldsymbol{m}}_L$, while the remainder landmarks are stacked as $\hat{\boldsymbol{m}}_G$.

In the compressed prediction step, the vehicle-to-map correlation is compressed as

$$\mathbf{P}_{vm}(k) = \mathbf{P}_{vm}(0) \prod_{s=1}^{k} \mathbf{F}(s), \; \mathbf{F}(s) = \mathbf{I} + \Delta T \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}_v} \right), \tag{19}$$

where $\boldsymbol{F}(s)$ is the Jacobian matrix at a discrete step-s with a step time ΔT. In this work, the prediction rate is 400Hz driven by the IMU data, and the correlation information is accumulated until a local-to-global update event occurs, which is at much lower rate of around 0.5Hz in this work. This is a significant computational gain in the SLAM prediction as only the vehicle-dependent Jacobian matrix of 17×17 needs to be compounded at a full rate, compared to the full correlation update of $17 \times N$ or full covariance of $N \times N$.

When observations are available (either pseudoranges/rates at 1Hz or visual landmarks at 25Hz), the local filter states are first updated, then the cross-correlation term $\mathbf{P}_{LG}$ and

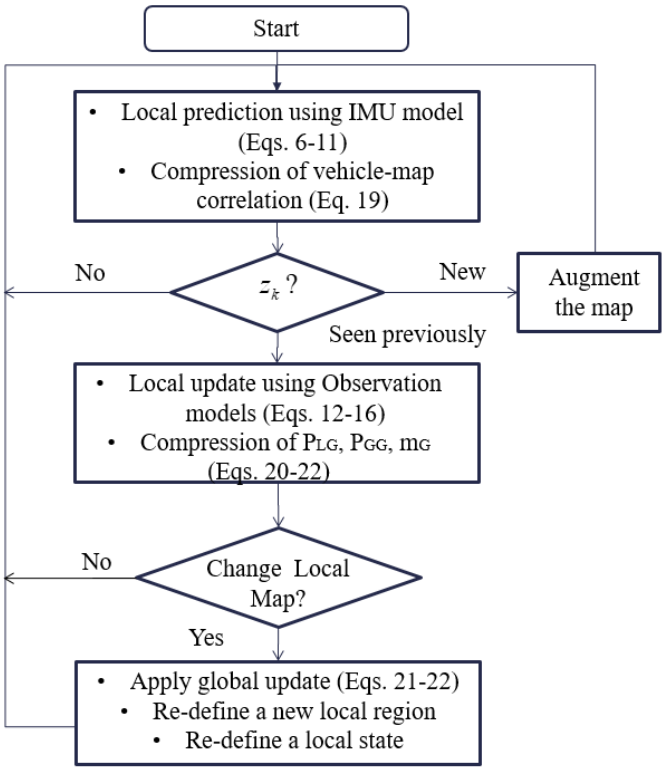


Figure 2. The flow diagram of the CP-SLAM algorithm.

finally the full global states. In compressed filtering, the accumulated cross-correlation at discrete time k has a closed-form expression given the initial correlation $\mathbf{P}_{LG}(0)$ as

$$\mathbf{P}_{LG}(k) = \mathbf{P}_{LG}(0) \prod_{s=1}^{k} \left(\mathbf{I} - \mathbf{P}_{LL}(s)\mathbf{H}^T(s)\mathbf{S}^{-1}(s)\mathbf{H}(s)\right), \tag{20}$$

where the terms inside the product are all available from the local filter update – the observation Jacobian $\mathbf{H}(s)$, the innovation covariance $\boldsymbol{S}(s)$ and the local covariance $\boldsymbol{P}_{LL}(s)$.

By using this compressed correlation $\mathbf{P}_{LG}(k)$, the global map and covariance matrix can be recovered as,

$$\mathbf{P}_{GG}(k) = \mathbf{P}_{GG}(0) - \sum_{s=1}^{k} \mathbf{P}_{LG}^T(i)\mathbf{H}^T(s)\mathbf{S}_i^{-1}(s)\mathbf{H}(s)\mathbf{P}_{LG}(s) \tag{21}$$

$$\hat{\mathbf{m}}_G(k) = \hat{\mathbf{m}}_G(0) + \sum_{s=1}^{k} \mathbf{P}_{LG}^T(s)\mathbf{H}^T(s)\mathbf{S}^{-1}(s)\boldsymbol{\nu}(s), \tag{22}$$

where $\boldsymbol{\nu}(s)$ represents the local innovation during the local update. Figure 2 summarises the algorithmic flow diagram with the key relevant equations discussed in this section. More

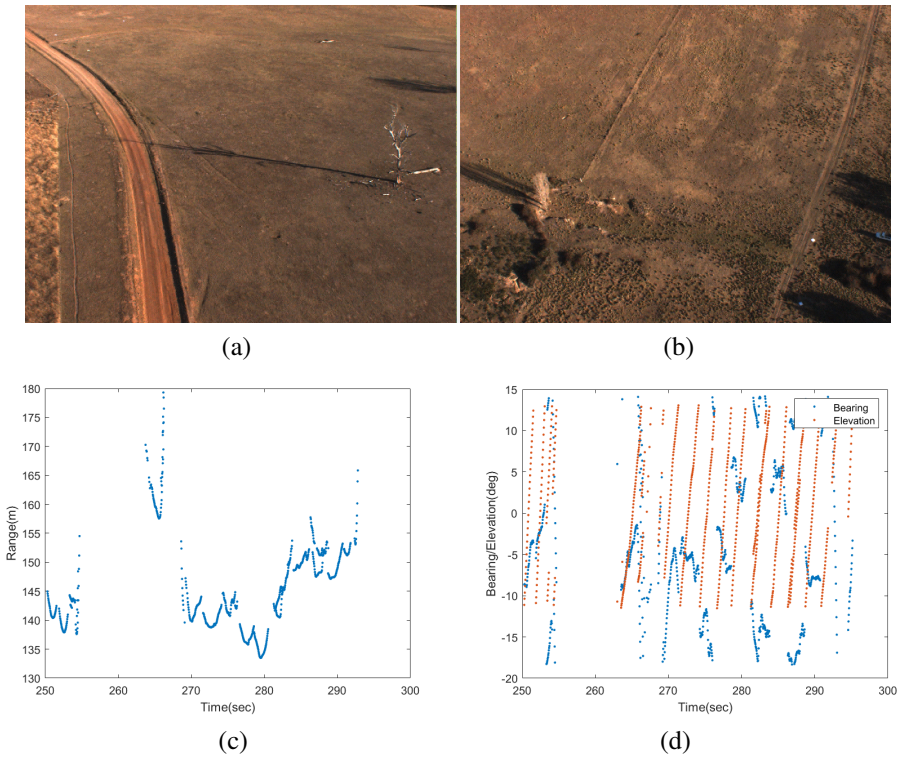
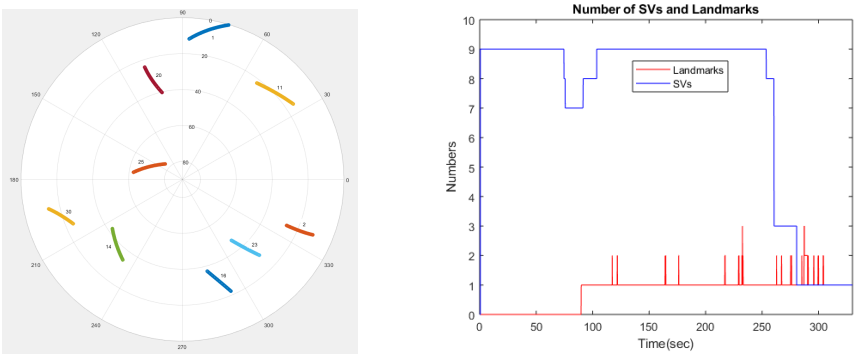


Figure 3. Two aerial images seen from the UAV (a)(b) . (b) An artificial/white visual landmark is seen at the right-bottom corner of the right image, which is a white plastic sheet of $1m \times 1m$ from which the *range* was estimated. (c) Extracted range and (d) bearing/elevation measurements from the detected visual landmarks. It can be observed that landmarks are detected from the top of the camera image (with a minimum elevation angle) and exit the view at the bottom of the image (with a maximum elevation angle). This also manifests in the range showing 'V'-like patterns, which stems from the fact that the range becomes maximum when a landmark is first or last detected, while minimum when it is directly under the vehicle.

discussions are provided in the following results section such as the data association and local region selection methods.

4. EXPERIMENTAL RESULTS.

4.1. *Environment and Sensors.* A flight dataset recorded from a fixed-wing UAV platform (Kim and Sukkarieh, 2007) is used to verify the method. An IMU sensor is a low-grade sensor with a gyroscope bias stability of $10^\circ/hr$, delivering the outputs at 400 Hz. A GPS receiver from Canadian Marconi Communication (now Novatel) was used to record the pseudorange and pseudorange rate measurements at 1 Hz and the satellite ephemeris data. The receiver provides integrated-carrier-phase (ICP) measurements, which are used to compute the pseudorange rates. DGPS correction messages (RTCM) were broadcasted from the ground station at the time of the experiment. A Sony monochrome camera (with a frame rate of 25 Hz) was installed in a down-looking configuration. Please refer to (Kim



(a) (b)

Figure 4. (a) The sky view plot of the satellite vehicle (SVs) during the flight experiment. (b) A dropout of the satellites from 250 seconds was simulated. The number of visual landmarks is shown together. It can be observed that the number of visual landmarks is quite low, changing from 0 to 3, and thus requiring additional pseudorange information to calibrate the low-cost IMU.

and Sukkarieh, 2007) for more details of the onboard sensor configurations together with a flight control computer that recorded the data.

Figures 3(a) and 3(b) show sample aerial images collected from the camera, showing a white plastic-sheet landmark laid on the ground at the right-bottom corner of Figure 3(b). Artificial visual landmarks are made of white plastic sheets ($1m \times 1m$). They are installed on the ground, whose positions are pre-surveyed using a pair of real-time-kinematic GPS receivers. An intensity-based blob-detection algorithm is used, followed by template matching to detect square-shaped landmarks. Once detected, the known size of white pixels is used to compute the distance (range) to the landmark. The resulting (range, bearing, elevation) with uncertainty are reported/recorded in the camera coordinates system as shown in Figures 3(c) and 3(d). It can be observed that landmarks are first detected from the top of the camera image (or with the minimum elevation angle in the camera coordinates), tracked, and disappeared at the bottom of the image (or with the maximum elevation angle). This can be seen in the increasing elevation angles in Figure 3(d). The range also shows a 'V'-like pattern, which stems from the fact that the distance to a landmark becomes minimum when the landmark lies directly under the vehicle. The detected landmarks are associated with the SLAM map using the joint-compatibility branch and bound (JCBB) algorithm, which can effectively match a set of landmarks with the map. In this work, however, the number of landmarks detected in each image frame is quite low $(1-2)$, and thus the association performance is close to the nearest neighbour (NN) method.

4.2. *Flight Trajectory and Vertical Stabilisation.* The flight segment has around 6 minutes of duration from the taking off, along the multiple racehorse tracks (each track is about 5 km long). The average flight speed is around 120 km/hr, and the average altitude is 150 m above the ground. Figure 4(a) shows the sky plot of the visible SVs in an azimuth-elevation form showing 9 visible SVs. Figure 4(b) shows the change of the number of satellite vehicles during the flight, with a simulated drop to 3 from 250 seconds and then 1 from 280 seconds to test the performance of the system. The number of visual landmarks (in red) is also shown, which changes from 0 to 3 during the flight. These sparse visual

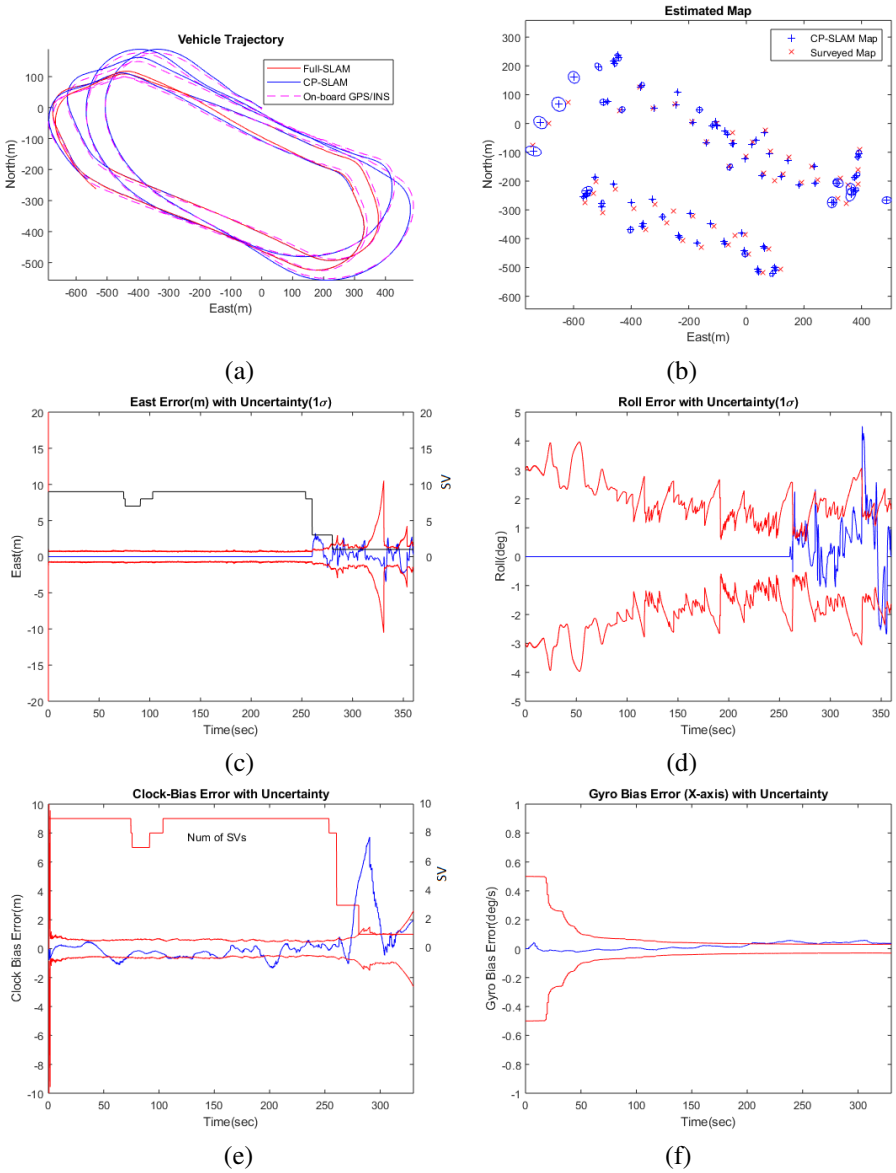


Figure 5. The estimation results from the compressed pseudo-SLAM (CP-SLAM): (a) the vehicle trajectory compared with Full-SLAM and on-board GPS/INS solution, (b) the map with uncertainty (10σ for visual clarity), (c) the East positional error (m) with uncertainty (10σ), (d) the roll angular error (deg) with uncertainty (1σ), (e) the receiver clock-bias error (m) with uncertainty (1σ), and (f) the x-gyro bias error with uncertainty (1σ). The CP-SLAM trajectory shows a consistent result compared to Full-SLAM. The navigational errors and corresponding uncertainties show larger errors when the number of SVs drops to 3 and 1. However, thanks to the continuous SLAM aiding, the navigational/clock/sensor errors are all constrained adequately.

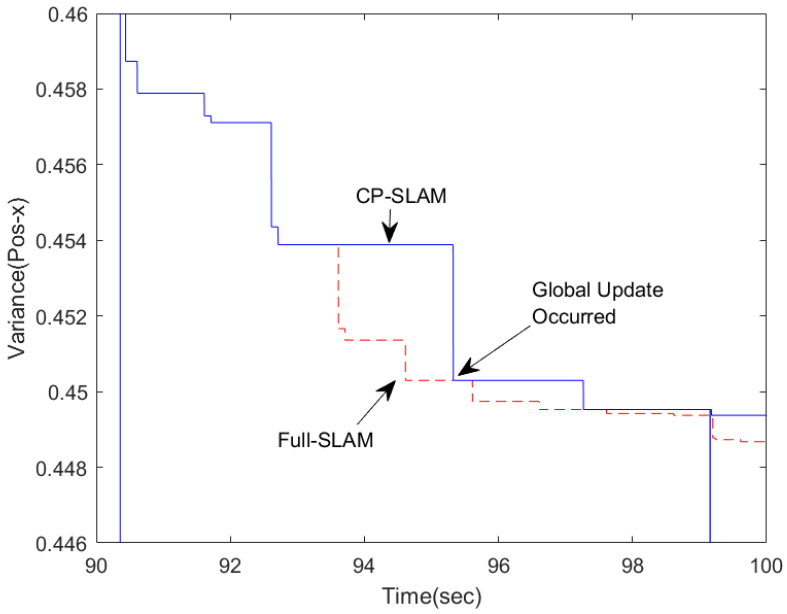


Figure 6. Comparison of the evolutions of covariance of CP-SLAM (in solid blue) and Full-SLAM (in dashed red), for the landmark-1. The landmark-1 is first registered at around 90.5 seconds, and the covariance of CP-SLAM and Full-SLAM change in the same way as the landmark is within the active map. From 93 seconds, however, the landmark becomes inactive as it is out of the local map, making the covariance of CP-SLAM unchanged. In contrast, the Full-SLAM continuously reduces the covariance. Due to the global updates at around 95 and 99 seconds, the whole map is corrected, making the CP-SLAM results back to optimal, thanks to the compressed fusion method.

observations make the height aiding of the inertial navigation system particularly challenging due to the large range errors inherent to the visual sensing. In this work, we stabilise the vertical height using the additional barometric-altitude information. The absolute and relative pressure measurements from the air data system are converted to barometric height and climb-rate and then used to aid the vertical inertial outputs using a simple α-β filter. Although this is a suboptimal approach, it simplifies the SLAM filter-tuning process with reliable performance and is thus adopted in this work.

4.3. *CP-SLAM Performance.* Figures 1 and 5 illustrate the operation of the CP-SLAM system and the estimation results, respectively. Figures 5(a) compares the estimated vehicle positions from CP-SLAM (in solid blue), non-compressed Full-SLAM (in solid red), and on-board GPS/INS solution (dashed red). It can be observed that the CP-SLAM and Full-SLAM show almost identical accuracy. Figure 5(b) also compares the estimated map from the CP-SLAM and the ground truth (surveyed map) with a total of 85 landmarks and uncertainty, showing consistent mapping performance. Figure 5(c) presents the navigational errors (East position error and Roll angle error) with 1σ uncertainties, and Figure 5(f) shows the receiver clock bias error and the IMU gyroscope bias error (x-axis) with uncertainties. It can be observed that when the number of SVs is sufficiently large (above 4), the pseudorange measurements dominate the estimation performance. In contrast, when

the number of SVs drops to 3 and 1, the errors increase, but the errors are adequately constrained within the uncertainty bound thanks to the continuous visual-aiding. Figure 6 compares the evolution of the map covariance from CP-SLAM (in solid blue) and Full-SLAM (in dashed red). For the plot's clarity, only the result of landmark-1 is shown for the first 10 seconds from the initial registration. It can be seen that, during $90.5 - 93$ seconds, the landmark is within the local map, thus actively updated in both CP-SLAM and Full-SLAM. However, from 93 seconds onwards, the landmark becomes inactive as it is out of the local map, making CP-SLAM's covariance unchanged. In contrast, the Full-SLAM continuously reduces the covariance. Due to the global updates at around 95 and 99 seconds, the whole map is corrected, making the CP-SLAM results back to optimal, thanks to the compressed fusion method. This clearly demonstrates CP-SLAM's benefits, which can run the global update less frequently without sacrificing the performance. This result is consistent with the results (Guivant, 2017) (Please refer to Figure 3 in page 13), which utlised simulated multiple mobile robots showing that the CP-SLAM methods can be generalisable to various robot navigation problems.

4.4. *Computational Complexity.* Figure 7(a) presents the change of the total number of landmarks (in blue) and the number of local landmarks (in red) registered in the CP-SLAM filter. The final state vector dimension is 272, consisting of the vehicle state 17 and the landmark map 255, which corresponds to 85 landmarks with 3D position for each landmark. The number of local landmarks is maintained within 20 throughout the flight, although it increases towards the end of the flight. This increase was due to the higher altitude of the vehicle resulting in a bigger local map and thus more local landmarks. Figure 7(b) compares the computational times of the filter update from CP-SLAM (in blue) and Full-SLAM (in red), showing the improved computational performance of CP-SLAM. The Full-SLAM update's computational complexity is $\mathcal{O}(N^2)$ with N being the total number of landmarks. At the same time, the CP-SLAM has the complexity of $\mathcal{O}(L^2), (L << N)$ with L being the local map size, which is less than the total map size. We can see that the pseudorange update times are nearly constant with 1 ms on average, while the local landmark update times increase towards 1.5 ms. However, the local-go-global updates sometimes show peaks during the map transitions, which was affected by the additional data association and sorting process. During the last 60 seconds of the GNSS dropout period, the update time is dominated by the visual updates. This result confirms that the compressed filtering approach is suitable for the real-time processing showing on average 1.5 ms processing time (using Matlab in Intel i5-core CPU with 1.7 GHz).

Figures 7(c) and 7(d) further compare the correlation structures of the Full-SLAM and CP-SLAM. Due to the sparse visual observations and landmarks, the correlation coefficient matrices become sparse (the top-left corner blocks are for the vehicle states). The Full-SLAM contains a much wider band-matrix structure than the CP-SLAM, particularly towards the right-bottom blocks, as expected. The CP-SLAM shows more off-diagonal elements, which are due to the frequent reshuffling of the landmarks into the local and global ones but has a smaller band-diagonal pattern due to the lower number of local landmarks. It can be noted that Full-SLAM registers the landmarks in temporal order, while CP-SLAM re-orders the map based on spatial closeness. Depending on the application, we can apply a more optimal strategy to define the local map. For example, this work utilises a moving local region along with the vehicle, while (Guivant, 2017) used a grid-based and fixed local region resulting in more block-diagonal structures. A further investigation might lead to a more sparse structure and thus a more efficient filtering algorithm.

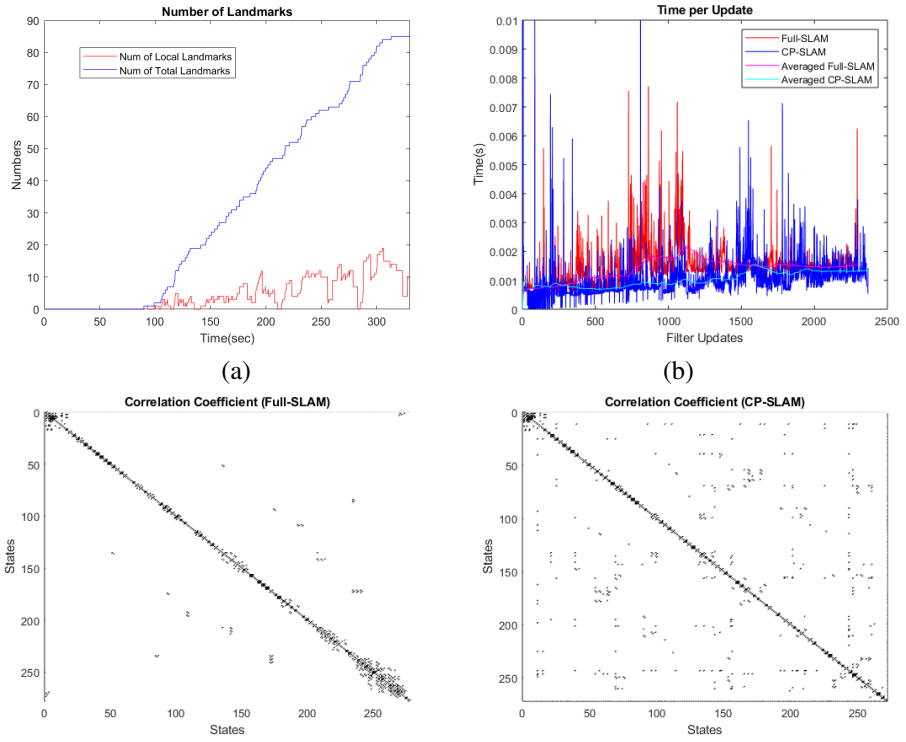


Figure 7. (a) The comparison of the total number of landmarks registered (in blue) and the number of local landmarks in CP-SLAM (in red), and (b) the comparison of the update time of the Full-SLAM (in red) and CP-SLAM (in blue). The smoothed average values of the update times clearly show the benefits of the CP-SLAM. (c) The Full-SLAM's correlation-coefficient matrices and (d) CP-SLAM (the vehicle states correspond to the top-left elements in the matrix). Both matrices show strong diagonal correlation patterns but with sparse off-diagonal structures due to the loop-closures. CP-SLAM shows more off-diagonal elements, which are due to the frequent reshuffling of the landmarks into the local and global ones, but has a smaller band-diagonal pattern due to the lower number of local landmarks.

5. CONCLUSIONS. This article presented a novel compressed pseudo-SLAM algorithm which seamlessly integrates pseudorange/pseudorange rates and inertial-visual SLAM to achieve continuous and reliable UAV navigation in a partially GNSS-denied environment. A flight dataset recorded from a high-speed, fixed-wing UAV platform was used to demonstrate the method. The offline processed results using Matlab showed sustained and reliable navigation solutions even under a single SV and sparse landmark observations while calibrating the IMU and the receiver clock errors. The compressed implementation improved the computational complexity achieving the filter update time of less than 1.5 ms on average for the state dimension of 272, consisting of 17 for the vehicle states and 255 for the map (or 85 landmarks). This result confirms the feasibility of the method for real-time implementation, which is a future work using a drone platform. The performance can be further improved by investigating the satellites and landmarks' optimal sensing geometry and designing an efficient local-map strategy. Future work is to address how to

optimally select the ground landmarks given a set of satellites in the cluttered environment to complement the low GNSS accuracy, for example, the height.

Acknowledgment Acknowledgement to ARC Discovery Project grant (DP200101640) and Research Council of Norway grant numbers 223254 and 250725. The authors thank Australian Centre for Field Robotics (ACFR) for using the flight dataset in this paper.

REFERENCES

Kim, J.; Cheng, J.; Guivant, J.; Nieto, J.:Compressed fusion of GNSS and inertial navigation with simultaneous localization and mapping, in *IEEE Aerospace and Electronic Systems Magazine*, vol. 32, no. 8, pp. 22–36, 2017.

Parkinson, B.: Origins, evolution, and future of satellite navigation, in *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 1, pp. 11–25, 1997.

Kim, J.; Sukkarieh, S,; Wishart, S.: Real-time navigation, guidance and control of a UAV using low-cost sensors, in*International Conference of Field and Service Robotics*, Yamanashi, Japan, 2003, pp. 95–100.

Skulstad, R.; Syversen, C.; Merz, M.; Sokolova, N.; Fossen, T.; Johansen, T.: Autonomous net recovery of fixed-wing UAV with single-frequency carrier-phase differential GNSS," in *IEEE Aerospace and Electronic Systems Magazine*, vol. 30, no. 5, pp. 18–27, May 2015.

Kim J.; Sukkarieh, S.: Real-time implementation of airborne inertial-SLAM, in *Robotics and Autonomous Systems*, vol. 55, no. 1, pp. 62–71, 2007.

Nützi, G.;Weiss, S.; Scaramuzza, D.; Siegwart, R.: Fusion of IMU and vision for absolute scale estimation in monocular SLAM, in *Journal of intelligent & robotic systems*, vol. 61, no. 1-4, pp. 287–299, 2011.

Li, M. ; Mourikis, A. I.: High-precision, consistent EKF-based visual-inertial odometry, in *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013.

Sjanic, Z.; Skoglund, M. A.; Gustafsson, F.: EM-SLAM with inertial/visual applications, in *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 1, pp. 273–285, Feb 2017.

Vidal, A. R.; Rebecq, H.; Horstschaefer, T.; Scaramuzza, D.: Ultimate SLAM? combining events, images, and imu for robust visual slam in hdr and high-speed scenarios, in *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 994–1001, 2018.

Williams, P., and Crump, M., "All-source navigation for enhancing UAV operations in GPS-denied environments," in *Proceedings of the 28th International Congress of the Aeronautical Sciences.*, 2012.

Guivant, J. E.: The generalized compressed Kalman filter, in *Robotica*, vol. 35, no. 8, p. 1639–1669, 2017.

Kim, J.; Sukkarieh, S.: Airborne simultaneous localisation and map building," in *Robotics and Automation, 2003. Proceedings. IEEE International Conference on*, vol. 1, pp. 406–411, 2003.