Sliding Window Filter with Application to Planetary Landing

Gabe Sibley
Department of Computer Science, University of Southern California, Los Angeles, California 90089
Larry Matthies
Computer Vision Group, NASA Jet Propulsion Laboratory, Pasadena, California 91109
Gaurav Sukhatme
Department of Computer Science, University of Southern California, Los Angeles, California 90089

Received 9 February 2010; accepted 21 June 2010

We are concerned with improving the range resolution of stereo vision for entry, descent, and landing (EDL) missions to Mars and other planetary bodies. The goal is to create accurate and precise three-dimensional planetary surface-structure estimates by filtering sequences of stereo images taken from an autonomous landing vehicle. We describe a sliding window filter (SWF) approach based on delayed state marginalization. The SWF can run in constant time, yet still achieve experimental results close to those of the bundle adjustment solution. This technique can scale from the offline batch least-squares solution to fast online incremental solutions. For instance, if the window encompasses all time, the solution is equivalent to full bundle adjustment; if only one time step is maintained, the solution matches the extended Kalman filter; if poses and landmarks are slowly marginalized out over time such that the state vector ceases to grow, then the filter becomes constant time, like visual odometry. Within the constant time regime, the sliding window approach demonstrates convergence properties that are close to those of the full batch solution and strictly superior to visual odometry. Experiments with real data show that ground structure estimates follow the expected convergence pattern that is predicted by theory. These experiments indicate the effectiveness of filtering long-range stereo for EDL.

1. INTRODUCTION

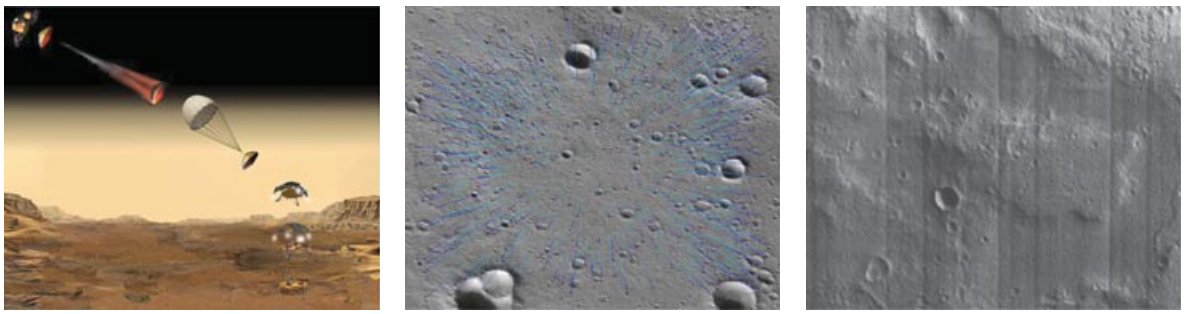
Accurate three-dimensional (3D) surface-structure estimation is a required capability for an autonomous vehicle to land on the surface of a planetary body. This is a data-fusion problem in which a sensor (undergoing uncertain, dynamic motion) must combine noisy measurements into a single underlying sensor-relative state estimate. In robotics this problem is most commonly tackled under the banner of simultaneous localization and mapping (SLAM) (Thrun, Burgard, & Fox, 2005); in computer vision the problem is referred to as structure-from-motion (SFM) (Triggs, McLauchlan, Hartley, & Fitzgibbon, 2000). This paper applies nonlinear least-squares optimization in a sliding window framework to the problem of accurate 3D planetary surface-structure estimation.

Given the entry, descent, and landing (EDL) scenario depicted in Figure 1(a), the motivation to use stereo vision instead of monocular vision is simple and practical: monocular cameras cannot easily observe depth information at the focus of expansion, which is in the center of the image for forward motion. Under normally distributed measurements [which is experimentally reasonable; see Figure 2(a)], stereo uncertainty at the focus of expansion is plotted in Figure 2(b). This highlights a fundamental advantage stereo has over a monocular setup.

Using stereo, the algorithm proposed in this paper focuses computational resources on accurately estimating surface structure by using a sliding-time window of measurements and by marginalizing out older parameters. The goal is a constant time algorithm that closely approximates the all-time *maximum a posteriori* (MAP) estimate, i.e., an estimator that achieves some notion of statistical optimality (accurately converges), efficiency (quickly reduces uncertainty), and consistency (avoids overconfidence).

Given the additive nature of measurement information, the signature of good performance for a system with a stationary, linear sensor will be proportional to a $1/m$ reduction in squared estimation error (where m is the number of measurements). This pattern is important for experimentally evaluating an estimator. Owing to motion, the structure of SLAM, and perspective projection, the problem at hand is nonlinear, and therefore error reduction will not in general follow the $1/m$ pattern. However, we will see that it is close to $1/m$ and hence that it is reasonable to use it as a metric.

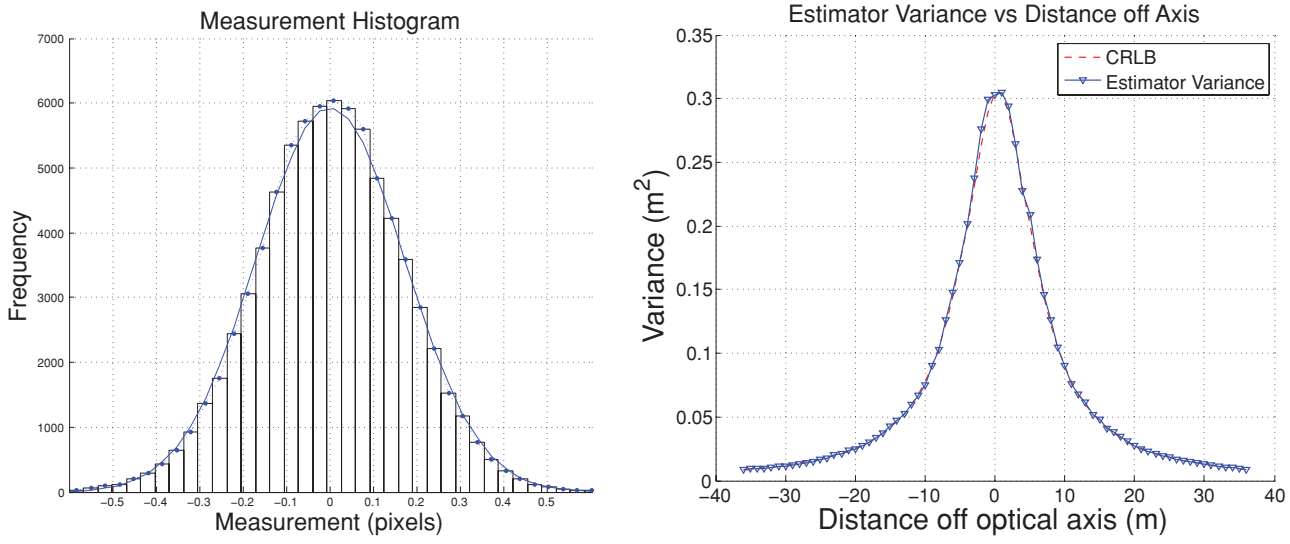
In Section 2 our derivation starts from the full nonlinear least-squares optimization perspective that considers all measurements over all time. This derivation shows how to split the estimation problem into three distinct information sources that stem from (1) measurements, (2) prior information, and (3) a process model. Section 3



(a) (b) (c)

Figure 1. (a) Graphic showing EDL for the 2011 Mars Science Laboratory mission. For the last ~1,000 m of EDL, it is envisioned that future missions will use dense-stereo data fusion for landing site hazard detection and avoidance (Image credit: NASA/JPL). (b) High-resolution MRO imagery used to simulate the final phase of EDL. (c) Example image from experiments with real data. To produce these data, a stereo sequence was taken of a simple planetary surface model built in the Robotic Embedded Systems Lab at USC.

examines the effect of marginalization on these three information sources—this examination motivates and leads directly to the sliding window approach, which is subsequently described. In Section 4 the sliding window filter (SWF) is tested in simulation and with stereo imagery captured to emulate EDL conditions for a Mars lander. These experiments show convergence in ground structure estimates that approach the best least-squares result predicted by theory. We also compare it to the extended Kalman filter (EKF), visual odometry (VO), full SLAM, and sparse bundle adjustment (BA). Compared to frame-to-frame techniques (such as VO) that do not fuse data over time, the SWF can be used to effectively extend the range resolution of stereo. In the context of EDL, this enables ground structure estimation from a greater altitude and hence more time for hazard avoidance prior to



(a) (b)

Figure 2. (a) Histogram of 77,049 measurement errors from real data indicating that a Gaussian measurement distribution on feature positions is reasonable. (b) Depth uncertainty at the focus of expansion for stereo, in which monocular approaches face difficulty. This graph shows estimator error and the CRLB after 25 frames vs. distance off the optical axis, as measured from a landing vehicle descending at 10 m/s, capturing frames at 10 Hz, starting at 100 m and stopping at 75 m. Notice that because we are fusing measurements over time, we see a motion baseline effect that leads to more certain estimates farther off the viewing axis. The horizontal FOV is ~25 deg; the stereo baseline is 1 m; image resolution is 512 × 384.

touchdown. Finally, in Section 5 we give an extensive literature review.

2. NONLINEAR LEAST-SQUARES APPROACH

To examine the underlying structure of the problem at hand, it is useful to approach it from the nonlinear least-squares optimization perspective. Compared to state-space filtering, this point of view is more in line with traditional statistical point estimation [though there are algebraic equivalences between the two views (Bell & Cathey, 1993)]. This perspective is useful for a number of reasons: first, because it highlights the fundamental minimization principle at work in least squares, which is sometimes harder to see from the state-space filtering perspective; and second, starting with the underlying probability density functions that describe our problem, it clearly shows the Gaussian probabilistic nature of SLAM—that is, SLAM is tracking a normal distribution through a large state space, a state space that changes dimension as we undertake the fundamental probabilistic operations of removing parameters via marginalization and adding parameters via error propagation and conditioning. A third reason to start from statistical point estimation is because it exposes a rich body of theory about the convergence of least-squares estimators (Dennis & Schnabel, 1996). Starting from least squares, one can easily see the connection to many important concepts such as Newton's method, Fisher information, and the Cramer–Rao lower bound (CRLB). All of these concepts have intuitive derivations starting from traditional statistical point estimation and nonlinear least squares.

2.1. Problem Formulation

The formulation below summarizes the presentation in Sibley (2006). This approach differs from other nonlinear least-squares SLAM presentations in that it focuses on clearly separating three distinct information sources that stem from (1) prior information, (2) measurements, and (3) process models (Dellaert, 2005; Thrun et al., 2005). This separation is useful for understanding the effect of incremental marginalization.

The parameter vector, $\mathbf{x} = [\mathbf{x}_\mathbf{m}^T \mathbf{x}_\mathbf{p}^T]^T$, is composed of n 3D landmarks, $\mathbf{x}_\mathbf{m} = [\mathbf{x}_{\mathbf{m}_1}^T \cdots \mathbf{x}_{\mathbf{m}_n}^T]^T$, and a temporal sequence of m six-dimensional (6D) robot poses, $\mathbf{x}_\mathbf{p} = [\mathbf{x}_{\mathbf{p}_1}^T \cdots \mathbf{x}_{\mathbf{p}_m}^T]^T$. The problem dimension is thus $\dim(\mathbf{x}) = (6m + 3n)$ and grows as the robot path increases and as new landmarks are observed.

The sensor model, $\mathbf{h}_{ij} : \mathbb{R}^{\dim(\mathbf{x})} \to \mathbb{R}^{\dim(\mathbf{z}_{ij})}$, describes the expected value the sensor will give when the ith landmark is observed from the jth pose: $\mathbf{z}_{ij} = \mathbf{h}_{ij}(\mathbf{x}_{\mathbf{m}_i}, \mathbf{x}_{\mathbf{p}_j}) + \mathbf{v}_{ij}$. We assume $\mathbf{v}_{ij} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{ij})$, giving the conditional distribution, $\mathbf{z}_{ij} \sim \mathcal{N}(\mathbf{h}_{ij}|\mathbf{x}, \mathbf{R}_{ij})$, where $\mathbf{R}_{ij}$ is the observation error covariance matrix. Note that the experimental histogram in Figure 2(a) indicates that the measurement distribution is close to Gaussian. Concatenating all the observations into the vector $\mathbf{z}$, predictions into the vector $\mathbf{h}(\mathbf{x})$, and covariances into a block diagonal matrix $\mathbf{R}$ gives $\mathbf{z} \sim \mathcal{N}(\mathbf{h}|\mathbf{x}, \mathbf{R})$.

To reflect parameters that have been marginalized out, we will need to accommodate prior information about the first pose and the map, $\hat{\mathbf{x}}_\mathbf{\Pi} \sim \mathcal{N}(\mathbf{x}_\mathbf{\Pi}, \mathbf{\Pi}^{-1})$:

$$\hat{\mathbf{x}}_\mathbf{\Pi} = \begin{bmatrix} \hat{\mathbf{x}}_\mathbf{m} \\ \hat{\mathbf{x}}_{\mathbf{p}_1} \end{bmatrix}, \quad \mathbf{\Pi} = \begin{bmatrix} \mathbf{\Pi}_\mathbf{m} & \mathbf{\Pi}_\mathbf{pm} \\ \mathbf{\Pi}_\mathbf{pm}^T & \mathbf{\Pi}_\mathbf{p} \end{bmatrix}, \tag{1}$$

where $\mathbf{\Pi}_\mathbf{p}$ is the 6×6 initial pose information matrix, $\mathbf{\Pi}_\mathbf{m}$ is the $3n \times 3n$ map prior information matrix, and $\mathbf{\Pi}_\mathbf{mp}$ is the $3n \times 6$ map-to-poses information matrix. It will become clear subsequently that this prior information comes from parameters that have been marginalized out of an incrementally operating filter.

The process model, $\mathbf{f}_j : \mathbb{R}^6 \to \mathbb{R}^6$, for a single step describes each pose in terms of the previous pose: $\mathbf{x}_{\mathbf{p}_{j+1}} = \mathbf{f}_j(\mathbf{x}_{\mathbf{p}_j}, \mathbf{u}_{j+1}) + \mathbf{w}_{j+1}$, where $\mathbf{u}_{j+1}$ is a known input command to the robot.[1] The noise vector $\mathbf{w}_{j+1}$ is additive and follows a normal distribution $\mathbf{w}_{j+1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{j+1})$, giving the conditional distribution $\mathbf{x}_{\mathbf{p}_{j+1}} \sim \mathcal{N}(\mathbf{f}_j(\mathbf{x}_{\mathbf{p}_j}, \mathbf{u}_{j+1})|\mathbf{x}_{\mathbf{p}_j}, \mathbf{Q}_{j+1})$. A simple and useful kinematic process model for $\mathbf{f}_j$ is the "compound operation," which is described in Smith, Self, and Cheeseman (1990). The 6×6 Jacobian of $\mathbf{f}_j$, $\mathbf{F}_j = \frac{\partial \mathbf{f}_j}{\partial \mathbf{x}_{\mathbf{p}_j}}|_{\mathbf{x}_{\mathbf{p}_j}, \mathbf{u}_{j+1}}$, which we will need in a moment, is also derived in Smith et al. (1990). Concatenating individual process models and covariances together, the probability density function describing the robot path is $p(\mathbf{f}(\mathbf{x})) = \mathcal{N}(\mathbf{f}(\mathbf{x}), \mathbf{Q})$.

The posterior probability of the system is $p(\mathbf{x}|\mathbf{z}) = p(\mathbf{z}|\mathbf{x})p(\mathbf{x})/p(\mathbf{z})$. The term in which we will be interested is

$$p(\mathbf{z}|\mathbf{x})p(\mathbf{x}) = \mathcal{N}\left(\begin{bmatrix} \mathbf{x}_\mathbf{\Pi} \\ \mathbf{f}(\mathbf{x}) \\ \mathbf{h}(\mathbf{x}) \end{bmatrix}, \begin{bmatrix} \mathbf{\Pi}^{-1} & & \\ & \mathbf{Q} & \\ & & \mathbf{R} \end{bmatrix}\right). \tag{2}$$

Our goal is to compute the value of $\mathbf{x}$ that maximizes this density. To do so it is first convenient to lump the sensor model, process model, and prior information terms together as follows:

$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} \mathbf{g}_\mathbf{\Pi}(\mathbf{x}) \\ \mathbf{g}_f(\mathbf{x}) \\ \mathbf{g}_z(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_\mathbf{\Pi} - \hat{\mathbf{x}}_\mathbf{\Pi} \\ \mathbf{x}_\mathbf{f} - \mathbf{f}(\mathbf{x}) \\ \mathbf{z} - \mathbf{h}(\mathbf{x}) \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} \mathbf{\Pi}^{-1} & & \\ & \mathbf{Q} & \\ & & \mathbf{R} \end{bmatrix}. \tag{3}$$

Ignoring constant terms that do not depend on $\mathbf{x}$, we see that $-\ln(p(\mathbf{z}|\mathbf{x})p(\mathbf{x}))$ is proportional to the quadratic cost function $\ell(x) = \frac{1}{2}(\mathbf{g}(\mathbf{x})^T \mathbf{C}^{-1} \mathbf{g}(\mathbf{x})) = \frac{1}{2}||\mathbf{r}(\mathbf{x})||^2$, which we aim to minimize [to see this, let $\mathbf{r}(\mathbf{x}) = \mathbf{S}\mathbf{g}(\mathbf{x})$, where $\mathbf{S}^T\mathbf{S} = \mathbf{C}^{-1}$].

[1] Using linearized error propagation, a more general model involving nonadditive noise can be easily treated. Constant velocity models can be used as well.

Using the notation $\mathbf{r}'(\mathbf{x}) = \partial \mathbf{r}/\partial \mathbf{x}$, the Gauss–Newton method defines the sequence of iterates:

$$\mathbf{x}_{i+1} = \mathbf{x}_i - (\mathbf{r}'(\mathbf{x}_i)^T \mathbf{r}'(\mathbf{x}_i))^{-1} \mathbf{r}'(\mathbf{x}_i)^T \mathbf{r}(\mathbf{x}_i), \tag{4}$$

which is locally *q-quadratically*[2] convergent to the MAP estimate for near-zero residual problems (Dennis & Schnabel, 1996). Noting that $\mathbf{r}'(\mathbf{x}_i) = \mathbf{S}\mathbf{G}_i$, where $\mathbf{G}_i$ is the Jacobian of $\mathbf{g}(\mathbf{x}_i)$, we get the system of equations $\mathbf{G}_i^T \mathbf{C}^{-1} \mathbf{G}_i \delta \mathbf{x}_i = -\mathbf{G}_i^T \mathbf{C}^{-1} \mathbf{g}(\mathbf{x}_i)$, which is the fundamental system of equations for the full SLAM problem. For notational convenience we will often omit the index i. For many problems the Gauss–Newton method is algebraically identical to the iterated EKF (Bell & Cathey, 1993). This establishes an important relationship between the parameter estimation perspective (such as BA) and filtering approaches (such as the EKF). This fact, in conjunction with an understanding of the effects of marginalization, leads to the SWF described in Section 3.

2.2. Three-Part Sparsity

The above formulation gives a natural three-way partition of the problem. It is well known that as $\boldsymbol{\ell}(\mathbf{x}_i) \to 0$, the approximate Hessian $\mathbf{G}_i^T \mathbf{C}^{-1} \mathbf{G}_i$ converges to the true Hessian. The term $\mathbf{G}_i^T \mathbf{C}^{-1} \mathbf{G}_i$ is also the Fisher information matrix, and its inverse approximates the system covariance (Bar-Shalom & Fortmann, 1988; Dennis & Schnabel, 1996).

With reference to Figures 3 and 4, which define **L**, **D**, and **H**, expanding the Jacobian

$$\mathbf{G} = \left[\frac{\partial \mathbf{g}_\pi}{\partial \mathbf{x}}^T, \frac{\partial \mathbf{g}_f}{\partial \mathbf{x}}^T, \frac{\partial \mathbf{g}_z}{\partial \mathbf{x}}^T \right]^T = \left[\mathbf{L}^T, \mathbf{D}^T, -\mathbf{H}^T \right]^T, \tag{5}$$

we see that the system matrix has three distinct terms:

$$\mathbf{G}^T \mathbf{C}^{-1} \mathbf{G} = \mathbf{L}^T \boldsymbol{\Pi} \mathbf{L} + \mathbf{D}^T \mathbf{Q}^{-1} \mathbf{D} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \tag{6}$$

and a sparse structure due to the form of **D**, **L**, and especially **H** (the structure of the Jacobians is shown in Figure 3). The three information sources (depicted in Figure 4) are (1) process-model information, (2) information from measurements, and (3) prior information.

The task is to solve $\mathbf{G}_i^T \mathbf{C}^{-1} \mathbf{G}_i \delta \mathbf{x}_i = -\mathbf{G}_i^T \mathbf{C}^{-1} \mathbf{g}(\mathbf{x}_i)$, which can also be expressed as the 2×2 system of equations

$$\begin{bmatrix} \boldsymbol{\Lambda}_\mathbf{m} & \boldsymbol{\Lambda}_\mathbf{mp} \\ \boldsymbol{\Lambda}_\mathbf{mp}^T & \boldsymbol{\Lambda}_\mathbf{p} \end{bmatrix} \begin{bmatrix} \delta \mathbf{x}_\mathbf{m} \\ \delta \mathbf{x}_\mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{g}_\mathbf{m} \\ \mathbf{g}_\mathbf{p} \end{bmatrix}, \tag{7}$$

where $\mathbf{g}_p$ and $\mathbf{g}_m$ are the right-hand-side (RHS) vectors corresponding to the robot path and map, respectively. Taking advantage of this sparse structure, the system of equations is typically solved by forward-then-backward substitution

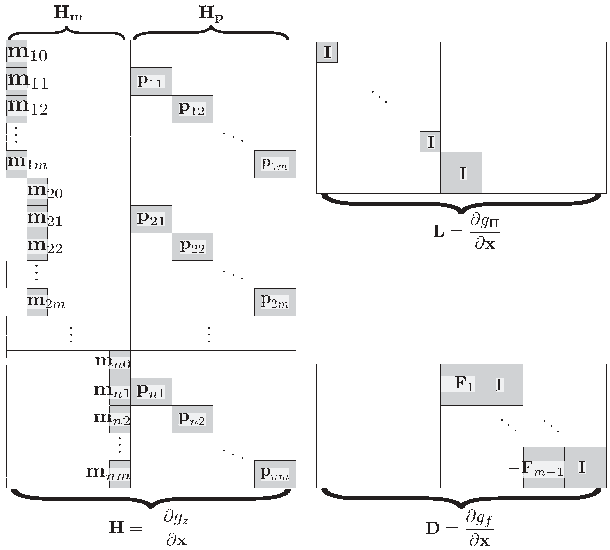


Figure 3. Structure of the Jacobians **H**, **D**, and **L**, $\mathbf{H} = [\mathbf{H}_{11}^T, \mathbf{H}_{12}^T, \ldots, \mathbf{H}_{nm}^T]^T$, where $\mathbf{H}_{ij} = \partial \mathbf{h}_{ij}/\partial \mathbf{x}$. Also, **H** has two components: $\mathbf{H}_\mathbf{p}$ is the Jacobian of **h** with respect to the pose parameters, $\mathbf{x}_\mathbf{p}$, and $\mathbf{H}_\mathbf{m}$ is the Jacobian of **h** with respect to the map parameters, $\mathbf{x}_\mathbf{m}$. Thus, $\mathbf{H}_{\mathbf{p}_{ij}} = \partial \mathbf{h}_{ij}/\partial \mathbf{x}_\mathbf{p}$ and $\mathbf{H}_{\mathbf{m}_{ij}} = \partial \mathbf{h}_{ij}/\partial \mathbf{x}_\mathbf{m}$.

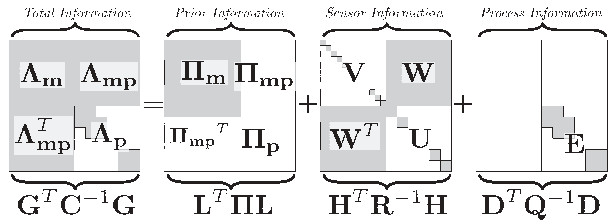
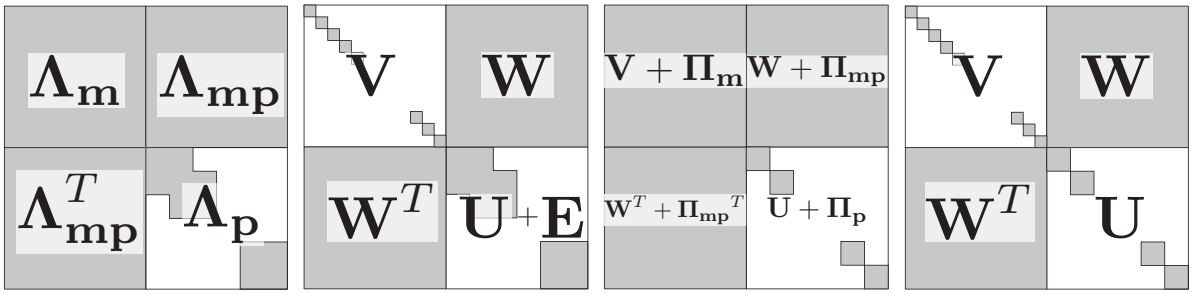


Figure 4. The sparse structure of the least-squares SLAM system matrix is due to contributions from three components: the measurement information matrix $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$, the process information matrix $\mathbf{D}^T\mathbf{Q}^{-1}\mathbf{D}$, and the prior information matrix $\mathbf{L}^T\boldsymbol{\Pi}\mathbf{L}$. The measurement information matrix has three distinct components, $\mathbf{U} = \mathbf{H}_\mathbf{p}{}^T\mathbf{R}^{-1}\mathbf{H}_\mathbf{p}$, $\mathbf{W} = \mathbf{H}_\mathbf{m}{}^T\mathbf{R}^{-1}\mathbf{H}_\mathbf{p}$, and $\mathbf{V} = \mathbf{H}_\mathbf{m}{}^T\mathbf{R}^{-1}\mathbf{H}_\mathbf{m}$. This figure also depicts the primary blocks of the Hessian—the "pose block," $\boldsymbol{\Lambda}_\mathbf{p}$, the "map block," $\boldsymbol{\Lambda}_\mathbf{m}$, and the "observation block," $\boldsymbol{\Lambda}_\mathbf{mp}$. Note that the $\boldsymbol{\Lambda}_\mathbf{mp}$ block often has a secondary sparsity pattern related to the observation sequence that can be utilized when solving large problems.

[2]Definition of *q-quadratic* convergence: Let $x_* \in \mathbb{R}, x_k \in \mathbb{R}, k = 0, 1, \ldots$ Then the sequence $\{x_k\} = \{x_0, x_1, x_2, \ldots\}$ is said to converge to x_* if $\lim_{k\to\infty} |x_k - x_*| = 0$. If there exist constants $c \geq 0$ and $\hat{k} \geq 0$ such that $\{x_k\}$ converges to x_* and for all $k \geq \hat{k}$,

$$|x_{k+1} - x_*| \leq c|x_k - x_*|^2,$$

then $\{x_k\}$ is said to be *q-quadratically* convergent (the prefix q stands for quotient). See Section 2.3, p. 20, in Dennis and Schnabel (1996) for a more detailed description.



(a) With motion model; with prior information

(b) With motion model; without prior information

(c) No motion information; with prior information

(d) No motion information; no prior information

Figure 5. The pattern of the system matrix depends on information contributed from three sources: the process model, measurements, and prior information. (a) Full SLAM structure. Including a process model but no prior from filtering (b) makes the lower right $m \times m$ "pose block" tridiagonal. Including a prior (c) can potentially cause complete fill-in of the upper left $n \times n$ "map block." Fill-in is induced when parameters are marginalized out—for instance, the fully correlated covariance matrix in EKF SLAM is precisely due to early marginalization of poses. With no process model information and no prior information (d), the problem is equivalent to photogrammetric BA.

with the Schur complement, either of the *path onto the map* or *map onto the path* (Triggs et al., 2000). See the Appendix for more on the Schur complement.

Depending on the process noise and the prior, the system matrix $\mathbf{\Lambda}$ can take on different sparsity patterns that affect the complexity of finding a solution (Frese, 2005). The possible sparsity patterns are shown in Figure 5. For instance, an infinite process noise covariance would mean that the motion model does not contribute information to the system ($\mathbf{Q}^{-1} = 0 \Longrightarrow \mathbf{E} = 0$). This would reduce the pose block of the system matrix to block diagonal, which is $O(m + n^3)$ to solve. Similarly, without prior information (i.e., $\mathbf{\Pi} = 0$) the map block is also block diagonal, which is $O(m^3 + n)$ to solve. Without information from the motion model and without prior information, the problem is equivalent to the BA problem in photogrammetry, which can be solved in either $O(m^3 + n)$ or $O(m + n^3)$ (Brown, 1976). The SWF is an incremental path-onto-map solution that also takes advantage of sparsity *and* the underlying three-part structure of the problem.

3. SLIDING WINDOW FILTER

To be useful during EDL, any proposed solution must have a computational complexity of $O(1)$ as a function of map size. The simplest way to bound computational complexity is to reduce the size of the state vector by, say, removing the oldest pose parameters or distant landmark parameters. However, if we directly remove parameters from the system we might lose information about how the parameters interact. Directly removing parameters is equivalent to conditioning and can lead to overconfidence. The correct way to remove parameters from a multidimensional normal distribution is to marginalize them out.

Marginalizing out parameters is equivalent to applying the Schur complement to the least-squares equations (see the Appendix). For example, given the system

$$\begin{bmatrix} \mathbf{\Lambda_a} & \mathbf{\Lambda_b} \\ \mathbf{\Lambda_b}^T & \mathbf{\Lambda_c} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x_a} \\ \delta\mathbf{x_b} \end{bmatrix} = \begin{bmatrix} \mathbf{g_a} \\ \mathbf{g_b} \end{bmatrix}, \tag{8}$$

reducing the parameters $\mathbf{x_a}$ onto the parameters $\mathbf{x_b}$ gives

$$\begin{bmatrix} \mathbf{\Lambda}_a & \mathbf{\Lambda}_b \\ 0 & \mathbf{\Lambda}_c - \mathbf{\Lambda}_b^T \mathbf{\Lambda}_a^{-1} \mathbf{\Lambda}_b \end{bmatrix} \begin{bmatrix} \delta\mathbf{x}_a \\ \delta\mathbf{x}_b \end{bmatrix} = \begin{bmatrix} \mathbf{g}_a \\ \mathbf{g}_b - \mathbf{\Lambda}_b^T \mathbf{\Lambda}_a^{-1} \mathbf{g}_a \end{bmatrix}, \tag{9}$$

where the term $\mathbf{\Lambda_b}^T \mathbf{\Lambda_a}^{-1} \mathbf{\Lambda_b}$ is called the Schur complement of $\mathbf{\Lambda_a}$ in $\mathbf{\Lambda_b}$. After this forward substitution step, the smaller lower-right system $[\mathbf{\Lambda_c} - \mathbf{\Lambda_b}^T \mathbf{\Lambda_a}^{-1} \mathbf{\Lambda_b}][\delta\mathbf{x_b}] = [\mathbf{g_b} - \mathbf{\Lambda_b}^T \mathbf{\Lambda_a}^{-1} \mathbf{g_a}]$ can be solved for updates to $\mathbf{x_b}$. Solving this remaining active problem is cubic in the size of the active state vector—the SWF described below seeks to bound the growth of the active state vector using incremental marginalization.

3.1. Overview

We now give a brief synopsis of the SWF algorithm.

3.1.1. Adding New Pose Parameters

First, after completing $m - 1$ steps, the command $\mathbf{u}_m$ is used to drive the system forward via the process model, $\mathbf{x_p}_m = \mathbf{f}(\mathbf{x_p}_{m-1}, \mathbf{u}_m)$, which adds six new pose parameters to $\mathbf{x_p}$. Recall that in the Gauss–Newton method the covariance matrix is approximated by the inverse of the Hessian matrix (Bell & Cathey, 1993). Thus, after applying the process model but *before* incorporating any new measurements, we can use the Gauss–Newton method to compute an updated information matrix, which is simply the Hessian associated

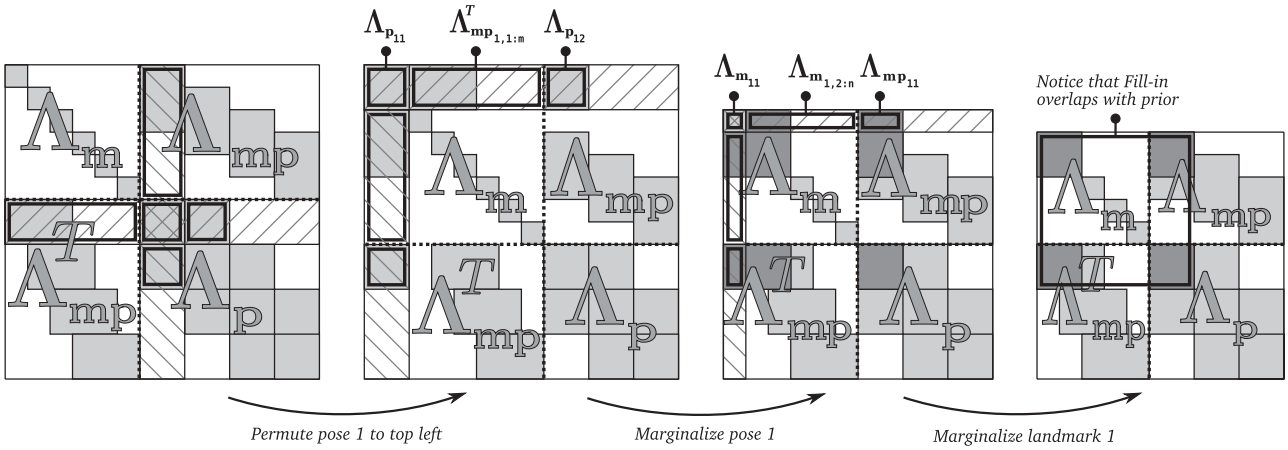


Figure 6. Evolution of the system matrix for a toy problem with four active poses and six landmarks. On the left is the system matrix after measuring landmarks 1, 2, and 3 from pose $\mathbf{x}_{\mathbf{p}_1}$; landmarks 2, 3, and 4 from pose $\mathbf{x}_{\mathbf{p}_2}$; landmarks 3, 4, and 5 from pose $\mathbf{x}_{\mathbf{p}_3}$; and 4, 5, and 6 at pose $\mathbf{x}_{\mathbf{p}_4}$. Figure 8 shows a graphical model for this system. Marginalizing out pose 1 induces conditional dependencies (fill-in) in three places: (1) the top left 6×6 of the pose block, $\mathbf{\Pi}_{\mathbf{p}}$; (2) the prior map block, $\mathbf{\Pi}_{\mathbf{m}}$, between landmarks that were visible from pose 1; and (3) the prior observation block, $\mathbf{\Pi}_{\mathbf{mp}}$, between poses and landmarks that were visible from pose 1. These places are shaded in darker gray. At this point marginalizing out landmark 1, which is not visible from any of the remaining poses, will induce no extra fill-in in $\mathbf{\Pi}$. This marginalization is depicted graphically in Figure 7.

with the MAP solution. This operation is a linearized error propagation, affects only the pose block of the information matrix, and can be computed in constant time.

3.1.2. Removing Parameters

Next, if there are now more than k poses active (for a k-step SWF), then we marginalize out the oldest pose parameters using the Schur complement. Note that marginalizing affects the RHS of the system equations. If $k = 1$, then this step transforms the state and information matrix identically to the first-order discrete EKF time step. Landmarks that are *no longer visible* from the active poses can also be marginalized out at this point. Marginalization applies the Schur complement to the system equations, which for the first pose is

$$\left[\mathbf{\Lambda}_{\setminus \mathbf{p}_1} - \mathbf{\Lambda}_{\mathbf{p}_1\mathbf{m}}^T \mathbf{\Lambda}_{\mathbf{p}_1}^{-1} \mathbf{\Lambda}_{\mathbf{p}_1\mathbf{m}}\right][\delta \mathbf{x}_{\setminus \mathbf{p}_1}] = \left[\mathbf{g}_{\setminus \mathbf{p}_1} - \mathbf{\Lambda}_{\mathbf{p}_1\mathbf{m}}^T \mathbf{\Lambda}_{\mathbf{p}_1}^{-1} \mathbf{g}_{\mathbf{p}_1}\right], \tag{10}$$

where the $\setminus \mathbf{p_1}$ notation indicates the removal of parameters from the associated vector—i.e., everything *but* the row/columns associated with $\mathbf{p_1}$. The same equations apply to landmarks.

3.1.3. Updating Parameters

Before a complete measurement update is computed, parameters are added to $\mathbf{x_m}$ to represent any newly observed landmarks (initial values are computed via stereo). Finally, all of the measurements within the time window are used to update the least-squares solution. We solve this sparse nonlinear least-squares problem with a robust Gauss–Newton method using the Huber kernel (Huber, 1964); we find that typically 4–10 iterations are needed for convergence. The Huber kernel is chosen to give robustness to outliers—it has the effect of down-weighting gross outliers. Measurements are considered outliers when their associated error goes beyond a threshold. In practice this threshold is computed from the inlier standard deviation after each iteration. Sampling methods such as random sampling and consensus (RANSAC) are also useful for outlier removal, though m-estimation was sufficient for the present work.

3.2. The Effects of Marginalization on the Three-Part System

Consider marginalizing parameters y from a Gaussian in canonical form: with reference to the Schur complement, it is easy to see that marginalizing y out will induce dependencies between other parameters that are dependent on y. Hence, marginalizing out a pose induces conditional dependencies between all the landmarks visible from that pose. This is depicted graphically in Figures 6–8. The ijth 3×6 block of the $\mathbf{\Lambda_{mp}}$ matrix encodes map-to-pose conditional dependencies and is nonzero only if the ith map landmark was visible from the jth pose—this is the "observation block" of the information matrix.

Marginalizing out the *oldest* pose from the full solution causes fill-in in three places: (1) between any landmarks that were visible from that pose, (2) between the parameters of the next oldest pose (the pose one time step after the pose being removed), and (3) between the next oldest pose and all landmarks seen by the removed pose. *Notice*

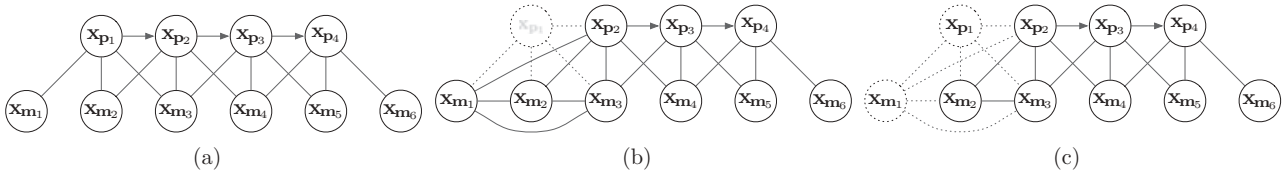


Figure 7. Graph interpretation of the marginalization example described in Figure 6. (a) The initial system, (b) the result of marginalizing out the first pose, (c) the result of marginalizing out the first landmark—removing landmarks with no active support does not cause additional fill-in.

that only $\mathbf{\Pi}$*—the part of the information matrix that expresses prior information—experiences additional fill-in*. This is important because it means that the pose block is still block tridiagonal, and the observation block is changed only along the first six columns—exactly where it overlaps with $\mathbf{\Pi_{mp}}$. Hence, when solving we can still take advantage of any sparsity patterns that may exist in $\mathbf{\Lambda_{mp}}$, just like as in BA (Hartley & Zisserman, 2000).

When landmarks are observed from a pose, this adds pose-to-landmarks conditional dependence information to the system matrix. This information encodes rigidity constraints. Marginalizing out a pose preserves this spatial rigidity by transferring its structure into a map of conditionally dependent landmarks. By marginalizing out poses, we have succeeded in removing the $O(m^3)$ cost of carrying the complete robot path in the state estimate.

We can completely bound map growth by marginalizing out landmarks that are no longer visible from any pose currently in the state vector. It is crucial that the landmark is no longer visible because, except for the oldest map-to-pose terms, all the cross-information terms in $\mathbf{\Lambda_{mp}}$ will be zero, which means that the Schur complement onto the remaining system will once again affect only the prior, $\mathbf{\Pi}$.

The key points here are that (1) marginalizing helps preserve information by transforming the way in which the system probability distribution is represented and (2) marginalizing both pose and landmark parameters that have no active support only ever affects the system prior and not the general sparsity pattern of the system equations—the prior term, $\mathbf{\Pi}$, catches all the information we marginalize out. By choosing when to marginalize poses and landmarks, the algorithm can scale from the full batch solution through the EKF solution to the incremental $O(1)$ solution. The space of solutions spanned by this approach is depicted graphically in Figure 9.

3.3. Optimality and Efficiency

If the batch filter converges to the minimum of the cost function, $\ell(\mathbf{x})$, then the resultant $\mathbf{x}$ is the minimal variance MAP estimate. This is the "best" or "optimal" estimate in the nonlinear least-squares sense with normally distributed measurements. The Gauss–Newton method has a well-known convergence theory, which, for near-zero-residual problems such as SLAM, is locally convergent (Bell & Cathey 1993; Dennis & Schnabel 1996; Ortega & Rheinboldt, 1970).

An important factor to address is estimator efficiency, that is, how well the estimator approximates a minimal variance estimate of the parameters. The information

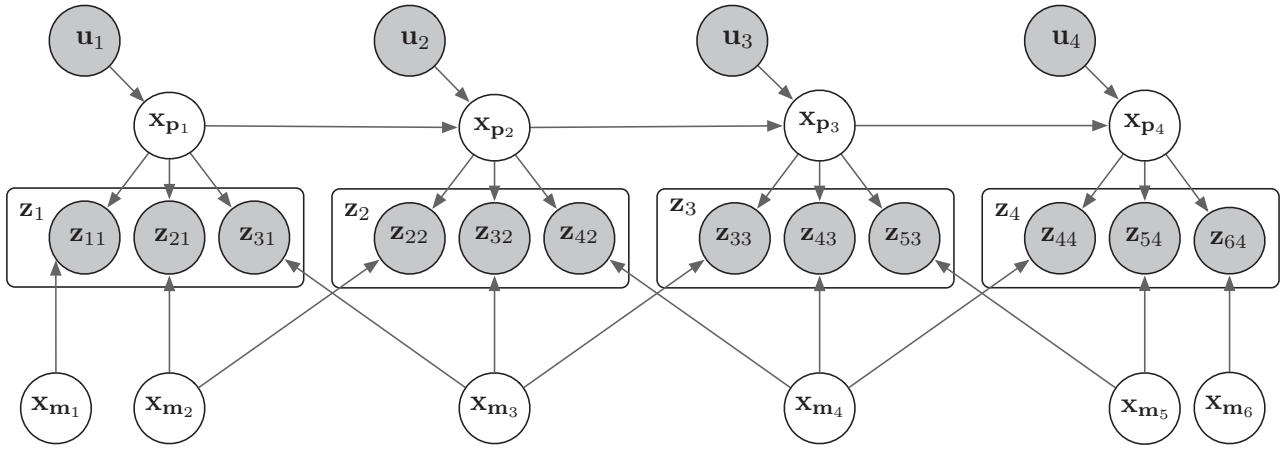


Figure 8. Graphical model for the example in Section 3.2 illustrating measuring landmarks 1, 2, and 3 from pose $\mathbf{x}_{\mathbf{p}_1}$; landmarks 2, 3, and 4 from pose $\mathbf{x}_{\mathbf{p}_2}$; landmarks 3, 4, and 5 from pose $\mathbf{x}_{\mathbf{p}_3}$; and 4, 5, and 6 at pose $\mathbf{x}_{\mathbf{p}_4}$.

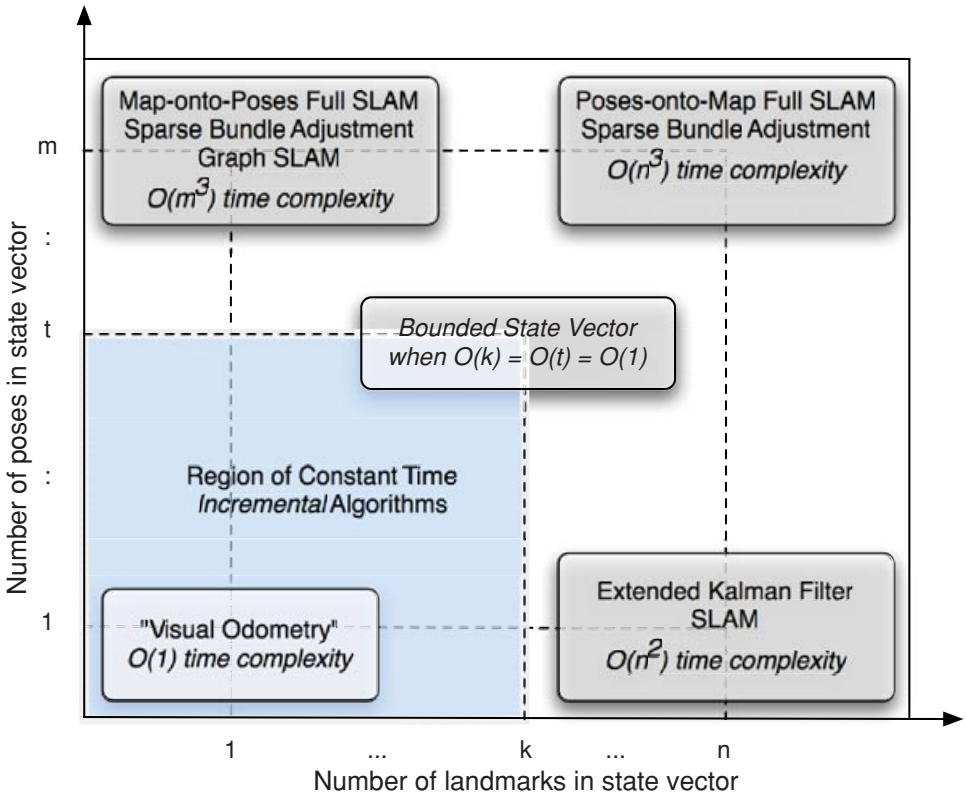


Figure 9. The space of solutions spanned with an adjustable window approach. Not considering loop closure, the constant time algorithms exhibit the potential to match the more expensive offline algorithms.

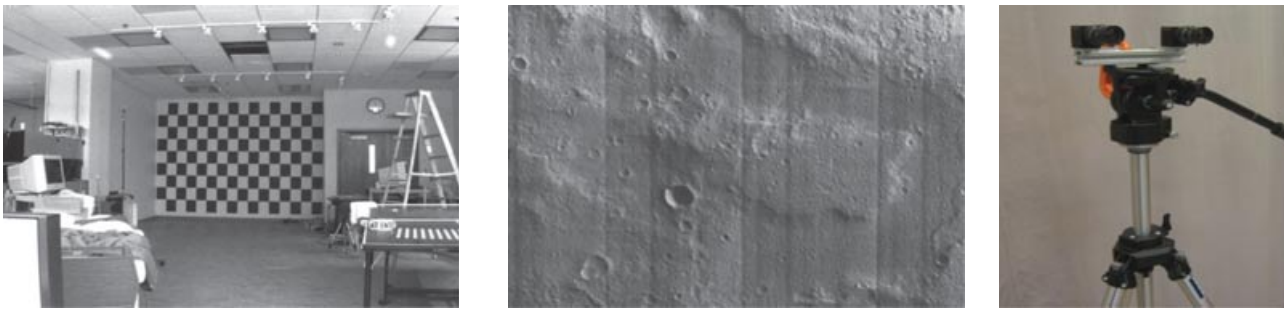
inequality, $\mathrm{cov}_{\mathbf{x}}(\mathbf{x}) \geq \mathcal{I}(\mathbf{x})^{-1}$, defines the minimal variance bound [i.e., the CRLB (DeGroot & Schervish, 2001)]. Here the matrix $\mathcal{I}(\mathbf{x})$ is the Fisher information matrix defined by the symmetric matrix whose ith, jth element is the covariance between the first partial derivatives of the cost function $\ell(\mathbf{x})$:

$$\mathcal{I}(\mathbf{x})_{,i,j} = \mathrm{cov}_{\mathbf{x}}\left(\frac{\partial \ell}{\partial \mathbf{x}_i}, \frac{\partial \ell}{\partial \mathbf{x}_j}\right). \tag{11}$$

For a multivariate normal distribution, Eq. (11) reduces to $\mathcal{I}_{i,j}(\mathbf{x}) = \mathbf{G}^T\mathbf{C}^{-1}\mathbf{G}$, which is equivalent to the Hessian matrix in the Gauss–Newton method. Evaluated at the *true* value of $\mathbf{x}$, the Gauss–Newton method *defines* the CRLB. This is why in least-squares problems such as SLAM, the Hessian (and inverse covariance) is the information matrix (Bar-Shalom & Fortmann, 1988; Sorenson, 1980). Iterative batch solutions to nonlinear problems generally give better parameter estimates than both noniterative methods and first-order recursive methods (Bierman, 1977; Jazwinski, 1970; Gelb, 1974; Maybeck, 1979; Sorenson, 1980). For example, because they do not iterate, methods such as the EKF have no guarantee of converging to the local minimum of the objective function. For convergence, one would have to use the iterative EKF, which is algebraically equivalent to the Gauss–Newton method (Bell & Cathey, 1993). Smoothing should also be employed to reduce linearization issues that plague filtering approaches (Julier, 2003).

4. EXPERIMENTAL RESULTS

The SWF was tested with real data captured to emulate Mars EDL. The data were painstakingly gathered explicitly to demonstrate convergence against surface-structure ground truth. A large flat surface was covered with orbital imagery from the Mars Reconnaissance Orbiter (MRO) High Resolution Imaging Science Experiment (HiRISE) camera, which has submeter per pixel resolution (McEwen, Eliason, Bergstrom, Bridges, Hansen, et al., 2007). This textured wall provides natural features to track. The stereo rig in Figure 10 was moved toward this surface to generate descent imagery. The rig consists of two Flea cameras from Point Grey Research with a ~10-cm baseline and narrow-field-of-view (FOV) lenses (~25 deg); gray-scale images were captured at 1,024 × 768 pixels. The cameras were calibrated and images rectified using CAHVOR camera models (Di & Li, 2004).



(a) (b) (c)

Figure 10. (a) The wall used to model planetary surfaces, with a checker pattern for ground truth in long-range stereo experiments (Sibley et al., 2006). (b) The same wall, but covered with poster-roll printouts of MRO HiRISE imagery. This model provides texture for feature detection and tracking. This image is a left view taken from the moving convergence experiment. The setup is designed to be roughly analogous to what will be seen during landing. Obviously planet surfaces are not flat, and our methods do not require a flat surface. This configuration serves to verify that our theory matches empirical results on real data. (c) The stereo rig used in these experiments.

To emulate landing conditions in a laboratory setting, we moved the cameras 15 cm from 10 to 1 m above the landing surface. Assuming a 1-m baseline for the landing vehicle, and disregarding image noise and charge-coupled device (CCD) size, this should compare to a sequence taken from 100 to 10 m, or the final stages of landing. Scaled by an order of magnitude, this corresponds to a frame every 1.5 m. Given a frame rate of 15 Hz, this represents a descent velocity of 10 m/s, which is appropriate for a lander descending on a parachute.

For ground truth, the tripod was mounted on a translation stage. The translation stage was slotted onto a measuring rule that was itself firmly attached to the floor. The tripod was then repeatedly moved to predetermined locations to collect measurements. Sequences along the same trajectory were taken with and without fiducials added (see Figure 11).

4.1. Implementation Details

If measurements are truly independent, identically distributed (i.i.d) and Gaussian, then the batch result is the best MAP solution (subject to local minima). It is therefore important to compare against the full batch least-squares result whenever possible. Before describing our results further, we first describe key implementation details behind our approach. After that we show results in which the SWF is tested with real images. For the purpose of EDL, this paper is focused on demonstrating ground structure convergence using real data. Note, however, that

(a) (b) (c)

Figure 11. (a) Illustration of the experimental apparatus that was built in the area shown in Figure 10(a). The stereo rig is fixed to a linear translation stage that moves along a measurement tape that is fixed to the ground. This allowed the creation of numerous similar trajectories, both with and without fiducials. (b) A picture of the wall shows fiducials used to establish surface-structure ground truth. The center hatch mark in each fiducial facilitates accurate subpixel saddle-point tracking. Unless otherwise noted, fiducials are *never* used in validation experiments presented. (c) Subsequence of 10 consecutive images from the moving convergence experiment. The surface model is designed to provide realistic texture for feature tracking; there is an obvious image-scale mismatch between images of the surface model (which is built from orbital views) and landing imagery.

although our goal here is to verify that experiment matches theory—which is an important demonstration of technology readiness prerequisite to any flight mission—the SWF has been applied in real-time ground robot experiments, as described in Newman, Sibley, Smith, Cummins, Harrison, et al. (2009).

4.1.1. Outlier Detection and Robust Estimation

The correspondence, data association, and outlier rejection problems are addressed using a combination of sum-absolute-difference (SAD) feature patch matching of Harris interest points (Harris & Stephens, 1988; Rosten & Drummond, 2006), Lucas–Kanade least-squares subpixel refinement (Lucas & Kanade, 1981), Moravec's consistency check (Moravec, 1980) [which has recently been rediscovered and extended by Hirschmller, Innocent, and Garibaldi, (2002) and used by Howard (2008)], and finally Huber's robust M-estimation (Huber, 1964; Rousseeuw & Leroy, 1987). Moravec's method is reminiscent of more recent methods that rely on relative geometry of landmarks (Bailey, Nebot, Rosenblatt, & Durrant-Whyte, 2000); given putative correspondences between two sets of 3D data points, it can quickly and effectively find the largest set of consistent correspondences that belong to a single rigid body transform—hence it is a useful first pass at discovering the dominant motion. This ability is what leads us to use it as a precursor to reject gross outliers. Any remaining outliers are handled by the Huber M-estimator, which converges faster with fewer outliers. Outlier detection is further improved by the fact that the M-estimator operates across the entire time window.

Moravec's method works as follows: given a set of 3D points at time t_0, another set at time t_1, and a list of n pairwise correspondences between the two sets, it first creates a "consistency" matrix. To create this matrix, first write the distance between features a and b at times t_0 and t_1 as $t_{0\bar{a}b}$ and $t_{1\bar{a}b}$. If we have the correct correspondence between these two features, then we know that their pairwise distances should be approximately the same at each time, and we can score their consistency in a $n \times n$ matrix, $\mathcal{C}_{ij} = ||t_{0ij} - t_{1ij}||$.

If $\mathcal{C}_{ij}$ is less than some threshold θ_C, then we say it is consistent. Treating $\mathcal{C}$ as an adjacency graph where only consistent pairs are neighbors, we see that the largest clique in this graph will be the largest set of self-consistent correspondences. Unfortunately, finding the maximum clique is one of the original nondeterministic polynomial-time complete (NP-complete) problems identified by Cook (1971). However, for small problems of the type we encounter, a greedy strategy is effective: simply select the maximal clique containing the node with the highest degree. Because this method detects the largest consistent set of correspondences, it is capable of finding the correct inlier set even in cases in which the inlier set is less than 50% of the data (it just has to be the largest consistent subset). It is also possible to extend Moravec's basic method to handle the greater uncertainty that plagues long-range stereo (Hirschmüller et al., 2002). In summary, to handle outliers, we have found it sufficient to employ all pair matching, followed by Moravec's method, and finally robust M-estimation using Huber's error function in the core of the SWF estimator. Our choice of the Huber kernel follows the recommendation of Zhang (1997).

4.1.2. The Importance of Feature Patch Warping

Experimentally, we found simple square-shaped patch tracking to be insufficient to achieve the desired level of convergence (Lucas & Kanade, 1981). To attain better results, we had to add a projective warping to each feature patch to align image patches before computing the subpixel feature position. To compute the patch warping between frames, we assume that each feature lies on a small locally planar 3D surface in the world, with the 3D location defined by the landmark state estimate and surface normal defined by the optical axis of the first frame in which the feature is tracked. Although this simple model is sufficient for our purposes, it would have to be extended for more general use to also estimate the patch surface normal. Still, this approach is superior to simple square patch tracking—even if landmarks do not lie on planar patches in reality.

Using this 3D plane assumption, we computed a warping homography to warp image patches from frame j back to the initial frame and then performed Lucas–Kanade least-squares, translation-only patch alignment to compute the subpixel feature location (Lucas & Kanade, 1981). We used bilinear interpolation for the warping and for the subpixel patch alignment. The subpixel patch location is refined after every Gauss–Newton step. Although this is expensive, it was necessary to attain the convergence results presented below and typically reduced error by 10%–15%.

4.2. Results

4.2.1. Static Convergence

Owing to obvious difficulties in establishing ground truth, experiments such as this typically cannot report landmark-estimate error results with real data. To evaluate landmark convergence, we first established ground truth by computing the batch solution over all frames using the fiducials. We then used the converged result to find a parametric model of the wall, against which we compute 3D map error as the shortest distance from a 3D landmark to the model. From this we were able to determine that the wall was indeed very flat (with average residuals of $\sim$0.1 mm from a single plane fit), and hence we found that using a plane as a ground-truth model was sufficient. Obviously planet surfaces are not flat, and our methods do not require a flat surface—it is merely an experimental convenience to establish that our theoretical understanding matches experimental results. Figure 12 shows convergence results for a

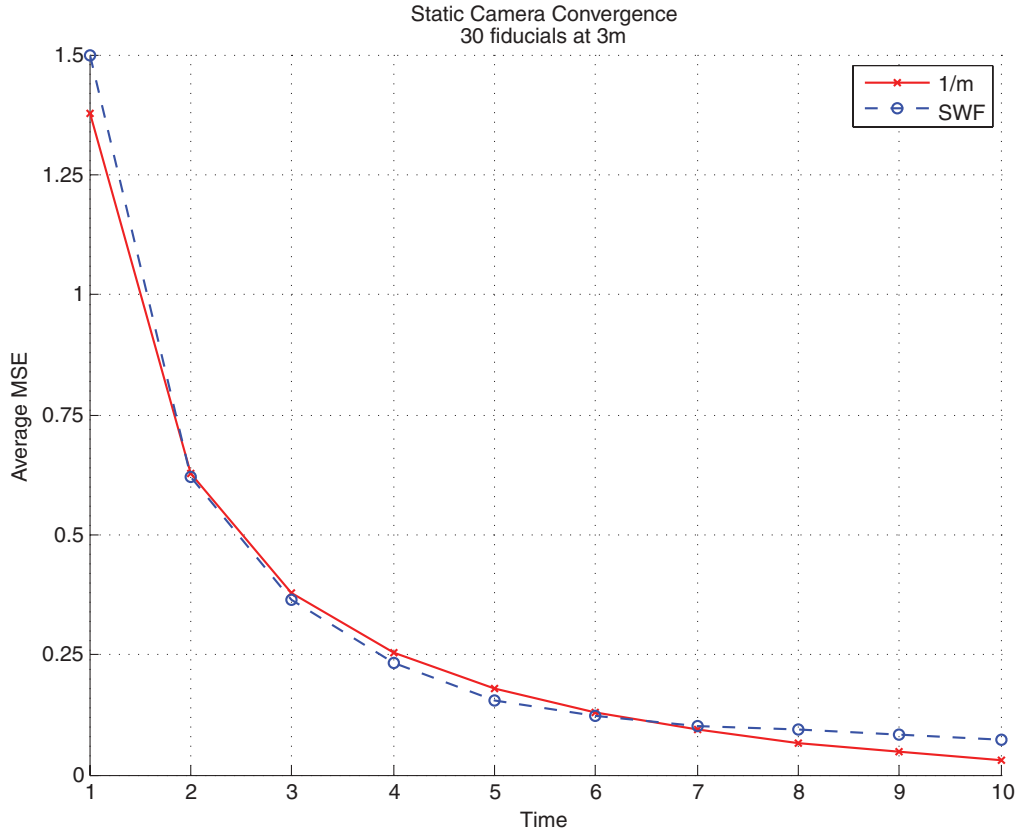


Figure 12. For validation, this graph shows landmark-convergence results for a sequence with stationary cameras tracking 30 fiducials (note that pose is still being estimated). This graphic indicates that the SWF performance is independent of the motion baseline effects or reduced range uncertainty effects that might be responsible for the better than $1/m$ convergence shown in Figure 13.

sequence with no camera motion with fiducials. This baseline experiment demonstrates convergence independent of any motion baseline effect or reduced-range uncertainty effect.

4.2.2. Moving Convergence

Our next experiments were designed assuming EDL systems with approximately 1-m stereo baselines. It is assumed that, depending on camera FOV and resolution, stereo ground ranging will commence anywhere from 300 m down to 50 m. Using a ~10-cm baseline, we moved the cameras at 15-cm steps from 10 to 1 m above the surface. Scaled by an order of magnitude, this corresponds to descent from 100 m down to 10 m.

Convergence results for this sequence are shown in Figure 13. This sequence tracks the same 150 features throughout. This is important because it means that we can predict that the uncertainty should follow an approximately $1/m$ curve, and hence we can use the plot of uncertainty to check for landmark convergence. It is apparent that we have achieved slightly better than $1/m$ convergence—which is to be expected given that range uncertainty naturally reduces as the distance to the surface shrinks and that there is a slight motion baseline effect. Note that the 3-, 4-, and 5-frame SWFs come close to matching the batch filter. This demonstrates the phenomenon that we are interested in—namely MAP landmark convergence for EDL.

4.2.3. Run Time

One aim of the SWF is to achieve constant run time by incremental marginalization. Figure 14(a) shows run times for full SLAM, a 10-frame SWF, and 20-frame SWF. Each frame measures ~20 features as the camera moves, and feature tracks survive ~10 frames. Figure 14(b) shows the average state-vector size for a moving system with an average feature-track length of 4.8 frames. This result shows the expected constant run time, independent of frame number or map size. For the 10-frame SWF, the average number of marginalizations is ~2 landmarks per frame.

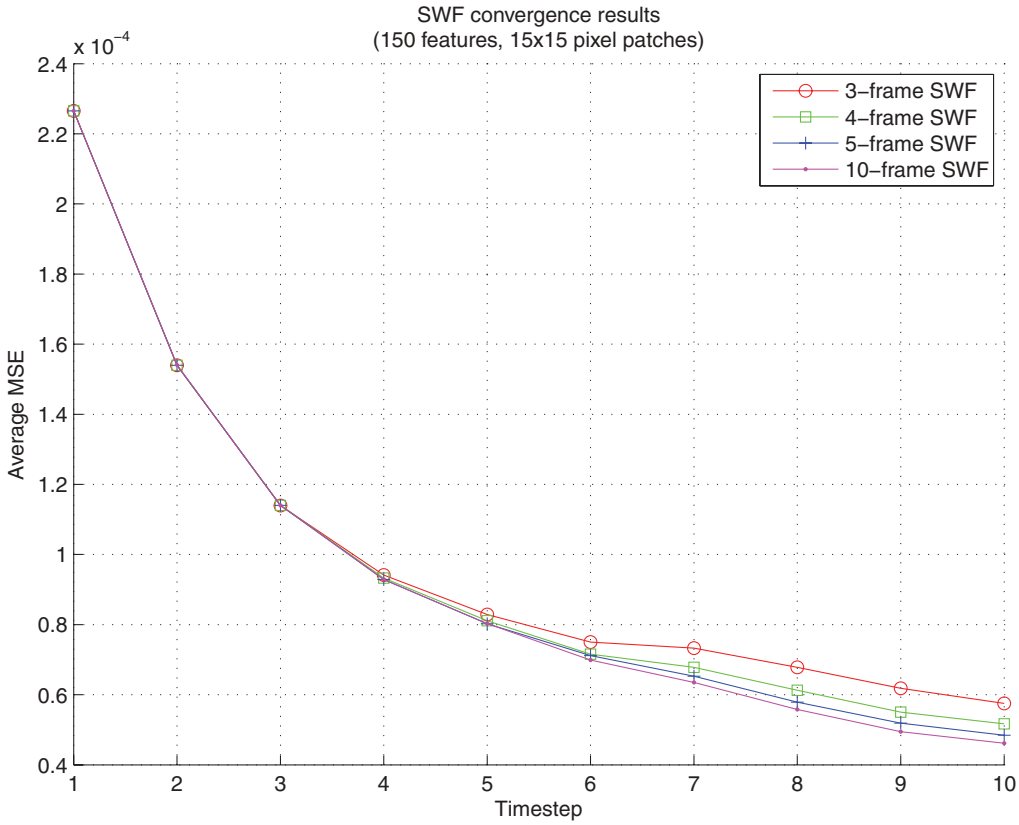


Figure 13. Convergence results for moving cameras for the Mars EDL experiment without fiducials. The graph shows comparison between full SLAM and the SWF of various window sizes. Because the SWF scales to match the full solution, the 10-frame estimate here represents the optimal least-squares result (as in BA or full SLAM). 150 features are tracked over 10 frames and fused with a SWF of 3, 4, 5, and 10 frames. This plot demonstrates slightly better than $1/m$ convergence because range uncertainty is also shrinking as the cameras get closer to the wall. The uncertainty is also reduced due to the motion baseline effect explained in Figure 2(b). The static camera case is shown in Figure 12 in order to indicate that convergence is not due to range-uncertainty reduction or a motion baseline effect.

4.2.4. Effect of Marginalization

Marginalization is beneficial in that it allows capturing the effect of past measurements as prior information (Frese, 2007). To appreciate this, consider what happens if we simply delete parameters from the estimator instead of marginalizing them out. As is apparent in Figure 12, the error converges according to $1/k$ just as we would expect the batch estimator to do. However, as shown in Newman et al. (2009), after k steps, *the error stops converging* as we delete information from the back of the filter. With such deleting and $k = 1$, we end up with a solution that is nearly identical to VO (Matthies & Shafer, 1987; Nister, Naroditsky, & Bergen, 2004; Olson, Matthies, Schoppers, & Maimone, 2001). The SWF comes close to matching the full solution and outperforms VO in terms of accuracy. For instance, over 10 frames, SWF error is ∼76% less than VO, and the standard deviation is reduced by a factor of ∼3.5. Together with the run-time results shown in Figure 14, the SWF can be seen as strictly superior to VO: it has the same computational complexity as VO yet it (1) shows convergence comparable to the batch solution and (2) does not suffer from stationary drift.

4.2.5. Effect of Early Marginalization

Early marginalization can also be detrimental. To demonstrate this, our next experiment shows the benefit of extended smoothing vs. filtering. Figure 15 shows the SWF and the full SLAM solution for sideways motion with landmarks coming into and going out of view. This sequence shows that the short-window SWF can suffer from rolling information into the prior too early. Just like the EKF (which is a 1-frame SWF), the 2-frame SWF suffers early linearization errors caused by marginalizing poses out too early. The 10-frame SWF on the other hand continues to relinearize old poses and is able to match the batch filter

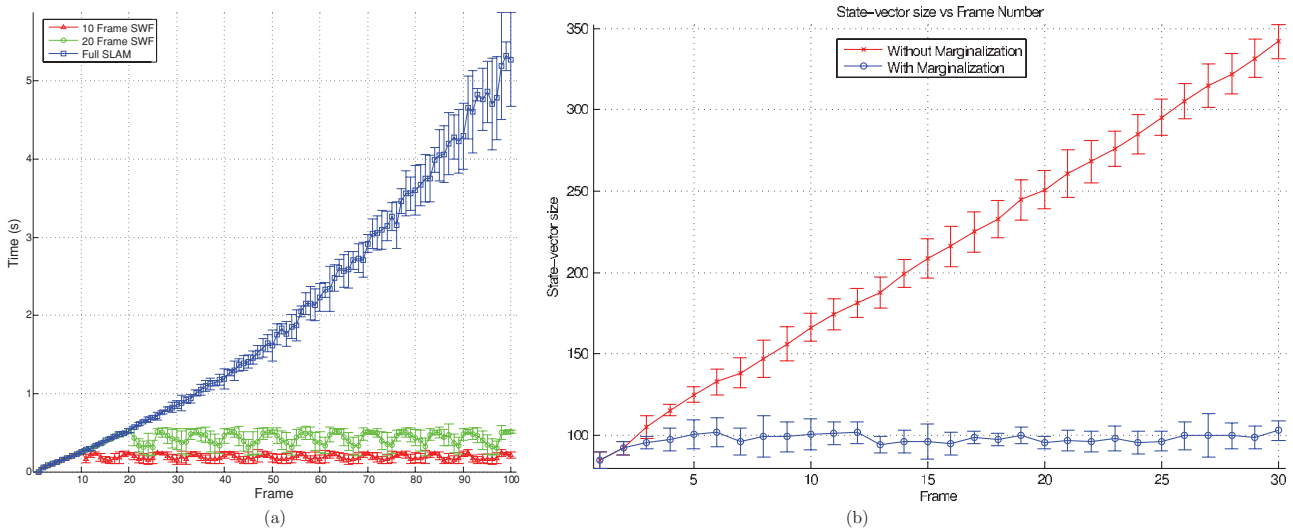


Figure 14. (a) Run time averaged over 20 runs comparing full SLAM and sliding windows with 10 and 20 frames. This shows the expected constant run time, independent of frame number. For simulation, added image noise is 0.5 pixels standard deviation. The problem tracks 76 landmarks; the average size of the state vector is ∼132 for the 10-frame filter and ∼172 for the the 20-frame filter. Each frame measures ∼20 features, and feature tracks last ∼10 frames. (b) A similar experiment showing the average state-vector size over five runs for a 2-frame SWF, both with and without marginalization. This demonstrates expected bounded state-vector size. Here the average feature track is ∼5 frames, ∼28 features are visible each frame, and measurement noise is 0.5 pixels standard deviation.

and avoid divergence. A better parameterization, such as inverse depth (Montiel, Civera, & Davison, 2006), can help alleviate this kind of linearization error.

Note that marginalization and linearization are related: marginalization of a parameter requires linearization about that parameter—this fact can lead to incorrect marginal distributions if the parameters are not well estimated before being marginalized out. By waiting to marginalize poses, the SWF benefits from delaying linearization until pose parameters have converged. Although it is not an issue for EDL, note that marginalization makes it difficult to reobserve landmarks, which effectively precludes the SWF from handling loop closures.

4.2.6. Average Feature-Track Length

The next experiment we describe is aimed at determining whether there is an optimal choice of sliding window size and whether the average feature-track length can aid in selecting a good window size. It is natural to ask whether there is a relation between sliding window size and the average feature-track length. For instance, if we look over a range of sliding window sizes, is there a plateau near the average track length? One might expect feature-track length to play an important role in overall estimator convergence. Figure 16 examines average converged map mean squared error (MSE) as a function of (1) average feature track length and (2) sliding window size. We see that the final map MSE is largely independent of SWF window size, which indicates that rolling measurements up into prior information is not problematic for this sequence and that even short-window-length filters perform well (e.g., 2 frames).

Although it is intuitive that MSE should depend on window size and average track length, we see that longer track length is more important than a longer sliding window. Further, it is not apparent from this result that sliding window size can be determined from a given average feature-track length. Although short windows can lead to decent convergence results, the major benefit of longer windows is the avoidance of inconsistency and estimator divergence that can arise from premature marginalization as, for example, in Figure 15.

4.2.7. Sparse Marginalization

In this experiment we compare full marginalization to the sparse marginalization technique used in the variable state dimension filter (VSDF) (McLauchlan, 1999; McLauchlan & Murray, 1995). Instead of applying the full Schur complement, the sparse VSDF applies only the diagonal elements of the Schur complement:

$$\begin{aligned}&\left[\mathbf{\Lambda}_{\backslash \mathbf{p}_1} - \operatorname{diag}\left(\mathbf{\Lambda}_{\mathbf{p}_1\mathbf{m}}^T \mathbf{\Lambda}_{\mathbf{p}_1}^{-1} \mathbf{\Lambda}_{\mathbf{p}_1\mathbf{m}}\right)\right][\delta \mathbf{x}_{\backslash \mathbf{p}_1}] \\ &\quad = \left[\mathbf{g}_{\backslash \mathbf{p}_1} - \mathbf{\Lambda}_{\mathbf{p}_1\mathbf{m}}^T \mathbf{\Lambda}_{\mathbf{p}_1}^{-1} \mathbf{g}_{\mathbf{p}_1}\right], \end{aligned} \tag{12}$$

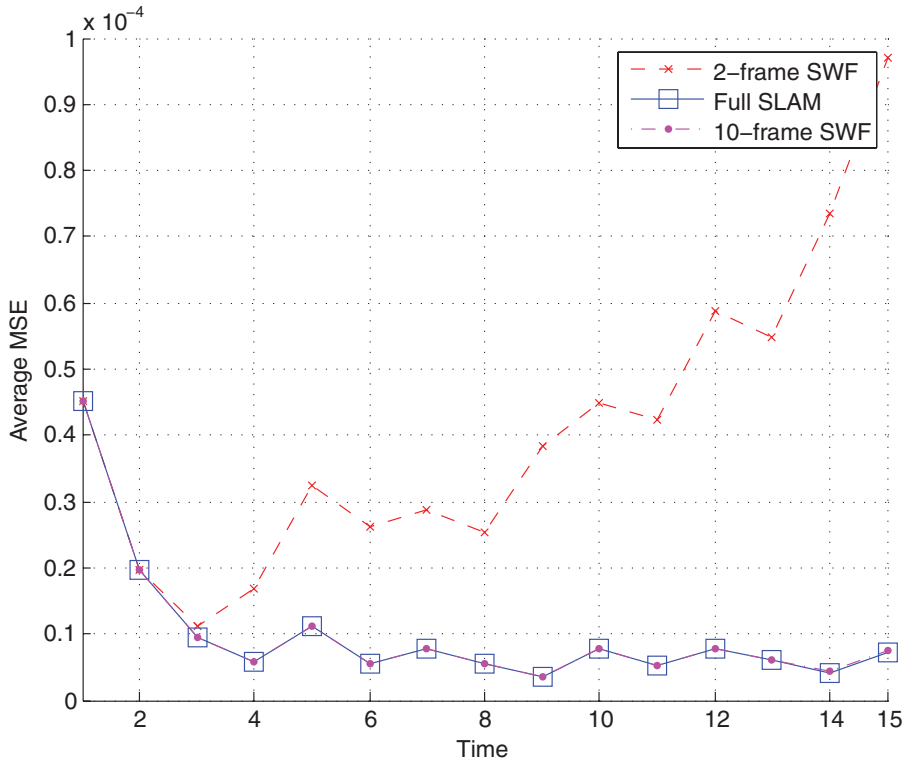


Figure 15. SWF compared to full SLAM for sideways motion. Each frame overlaps ~85%, with ~25 features per frame. This sequence shows the danger of rolling information into the prior too early.

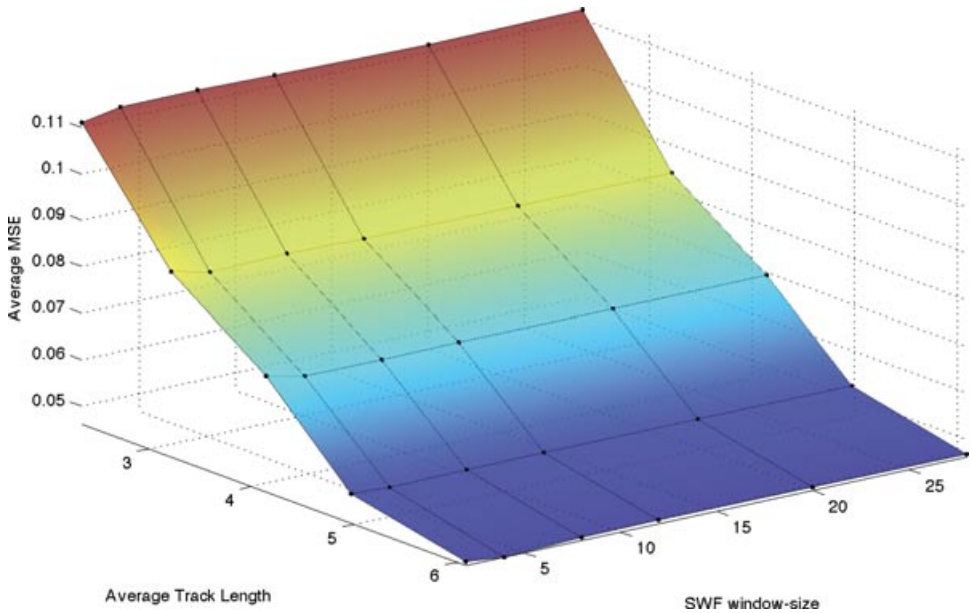


Figure 16. Converged estimator error vs. feature-track length and sliding window size. Each point on the surface represents average final map MSE for that trial point. In all trials approximately 30 features per frame are measured. This indicates that track length has a large impact on estimator error, whereas SWF window size is less important.

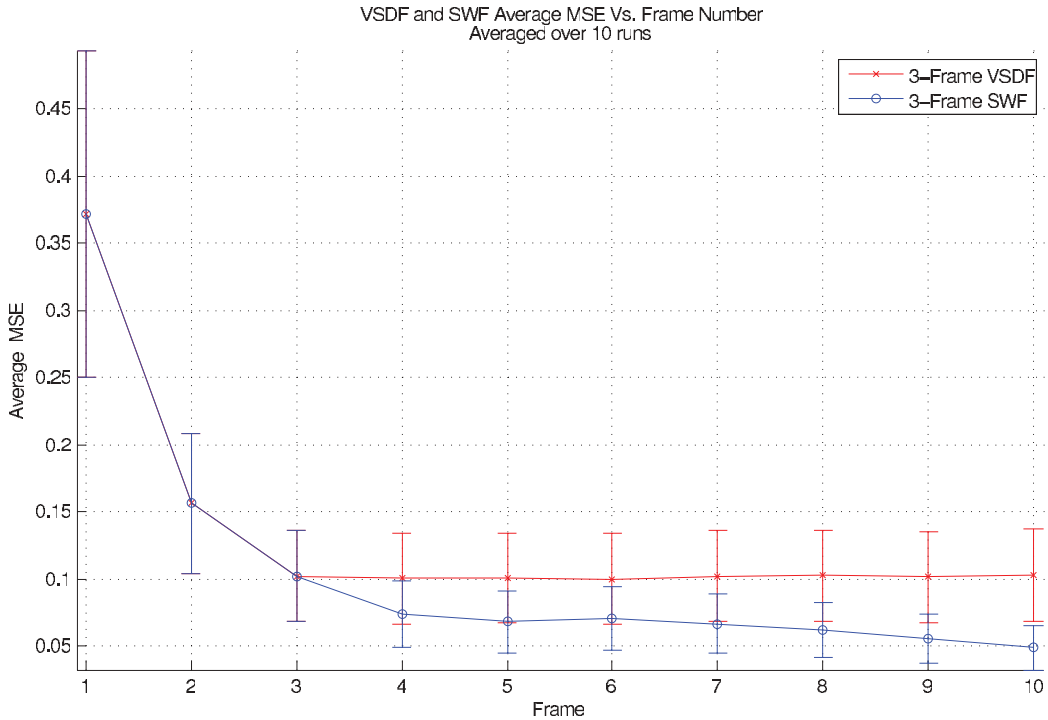


Figure 17. Convergence results comparing VSDF and SWF, both using three active frames. To expose the expected convergence rate, all features are measured over all frames.

which has the desirable effect of avoiding fill-in in the prior matrices. Figure 17 examines the effect of sparse marginalization. It is apparent from this result that marginalization as in Eq. (12) does not transform measurement information into prior information as effectively as full marginalization does. In fact, it seems that a majority of the usefulness of carrying a prior comes from the off-diagonal terms—i.e., the conditional dependencies between parameters are valuable. This is not surprising given the well-known importance of cross-covariance in EKF SLAM (Newman, 1999).

5. RELATED WORK

This work was motivated by results from the photogrammetry community, dating back to the late 1950s (Brown, 1958; Mikhail, 1983) and later derivatives such as the VSDF (McLauchlan, 1999; McLauchlan & Murray, 1995), VO (Matthies & Shafer, 1987; Nister et al., 2004), and EKF SLAM (Smith et al., 1990). The techniques of photogrammetry were gradually adopted or rediscovered as VO and SFM in the computer vision community (Fitzgibbon & Zisserman, 2004; Matthies & Shafer, 1987; Triggs et al., 2000) and SLAM in the robotics community (Lu & Milios, 1997; Thrun et al., 2005). Nonlinear least-squares optimization is the tool at the heart of these methods, and very similar problems arise in surveying (Wolf & Ghilani, 1997), orbit determination (Tapley, Schutz, & Born, 2004), photogrammetry (Brown, 1958; Mikhail, 1983), and geodesy (Vanicek & Krakiwsky, 1986). Indeed, there is a rich history of applying estimation theory and nonlinear least-squares to the task of *relative* spatial estimation—a history[3] that dates back most prominently to Gauss (1821, 1995).

It is useful to divide the literature into three camps: (1) "full SLAM" methods related to the batch nonlinear least-squares problem described in Section 2.1, (2) approaches comparable to or based on the EKF, and (3) nonparametric methods based on sampling (Doucet, Andrieu, & Godsill, 2000; Fox, Burgard, & Thrun, 1999; Montemerlo & Thrun, 2003). The latter is sufficiently different from our own approach that we do not discuss it further, except to note that there exist a number of particle filter visual-SLAM systems (Elinas & Little, 2005; Pupilli & Calway, 2005; Qian & Chellappa, 2004) and to note that the primary benefit of particle filters is their ability to track multimodal distributions. In this light it is interesting to note parenthetically that with reversible data association, SWFs can retroactively change mode estimates as new information comes online (Bibby & Reid, 2007). In real-world dynamic scenes, ambiguous situations rarely persist for long, and hence reversible data association within a sliding time window is one

[3]See Sorenson and Stubberud (1968) for a brief but excellent history of estimation theory.

possible mechanism to overcome the pitfalls of unimodal estimates (Bibby & Reid, 2007). This same flexibility means that measurements can be included at any instance within the sliding time window, which allows for out-of-sequence measurement updates (Ranganathan, Kaess, & Dellaert, 2007).

5.1. Full SLAM Methods

Since the original development of the SWF (Sibley, 2006), a few similar techniques have been developed in the computer vision literature based on BA (Engels, Stewenius, & Nister, 2006; Mouragnon, Lhuillier, Dhome, Dekeyse, & Sayd, 2006) and fixed-lag smoothing techniques (Kaess, & Dellaert, 2006; Mourikis & Roumeliotis, 2007). The high frame rates achieved in Engels et al. (2006) are largely due to short feature-track length, which is very different from planetary landing in which features can easily be tracked for 30–40 frames; further, the effect of marginalization and including prior information is not addressed, and it is assumed that fixing old frames is reasonable. Because frames are removed and only certain key frames are kept, the results cannot converge to the batch solution over all measurements. Similarly, the results of Mouragnon et al. (2006) do not include all data but instead use only a select subset of key frames and hence cannot match full SLAM. In contrast, the SWF attempts to match the full solution by rolling parameters into prior information. In Kaess and Dellaert (2006) and Mourikis and Roumeliotis (2007), even though details are not provided, the estimation method appears to be similar, because a sparse information-form smoother is used.

Brown's photogrammetric BA is the original image-based batch maximum likelihood solution to the full SLAM problem from the iterative nonlinear least-squares perspective (Brown, 1958). Brown's sparse (and therefore, fast) solution to BA does not include dense prior information or a process model. The work by Mikhail (1983) gives an incremental/recursive algorithm that can include arbitrary functional relationships between parameters (e.g., a process model) as well as prior information matrices. However, to facilitate faster run times Mikhail employs the same sparse optimizations as Brown. Brown's sparse system of equations does not capture the temporal evolution of the probability density function if there is prior information induced by marginalization.

Graph-SLAM (Thrun et al., 2005), exactly sparse delayed state filters (ESDSF) (Eustice, Singh, Leonard, Walter, & Ballard, 2005), smoothing and mapping (SAM) (Dellaert, 2005; Kaess, 2008), and recent work of Konolige and Agrawal (2007) are all examples of nonlinear least-squares techniques similar to BA. SAM solves the system equations efficiently by variable reordering, which is also a well-known technique in photogrammetry (Triggs et al., 2000). Graph-SLAM is an offline solution and is typically tackled with available numerical sparse solvers (Galassi, Davies, Theiler, Gough, Jungman, et al., 2003; Press, Flannery, Teukolsky, & Vetterling, 1992).

Both Graph-SLAM and ESDSFs factor the map onto the path, thereby producing a pose graph, which can then be solved for the optimal robot trajectory. Fast pose-graph optimization methods are a recent development (Frese & Duckett, 2003; Grisetti, Stachniss, Grzonka, & Burgard, 2007; Olson, Leonard, & Teller, 2006). By finding the maximum likelihood configuration of a sequence of interrelated poses, these approaches can solve impressively large global SLAM problems. Note, however, that pose-graph methods do not compute optimal landmark-structure estimates and instead focus on computing the vehicle trajectory.

The VSDF (McLauchlan, 1999; McLauchlan & Murray, 1995) tries to combine the benefits of batch least squares with those of recursive estimation. Both the SWF and the VSDF are very similar to Mikhail's "unified adjustment" technique (Mikhail, 1983). Mikhail's work is a general and complete treatment of least-squares adjustment, whereas the SWF and VSDF are specific examples applied to incremental SLAM and SFM. The VSDF is a mixed formulation, taking inspiration from the sparse Levenberg–Marquardt method used in BA (Hartley & Zisserman, 2000; More, 1978) and also from the EKF used in SLAM (Smith et al., 1990). For computational efficiency, the VSDF ignores conditional dependencies that are induced from marginalizing out old parameters, and similar to Brown's BA, it also ignores conditional dependencies that exist between adjacent pose parameters—especially the block tridiagonal matrix structure of the process block. In comparison, the least-squares formulation for full SLAM captures this information naturally. Neglecting conditional dependencies can be detrimental as it can lead to divergence (Newman, 1999).

The work of Deans (2005) is also inspired by the least-squares approach and, like the SWF, aims at online operation by focusing computation on the most recent set of measurements while removing other parameters from consideration. However, instead of incrementally marginalizing the solution pose by pose, the formulation breaks the problem into sets of adjacent batch problems. The effect of marginalizing out landmarks is not explored, nor is the overall evolution of the structure of the system equations.

5.2. Single-Pose Methods

In this section we describe methods similar to the original EKF SLAM solution (Smith & Cheeseman, 1986) that only keep the current pose active in the state vector. In the context of EDL, an EKF approach was employed in Trawny, Mourikis, Roumeliotis, Johnson, and Montgomery (2007). Credit for the first frame-rate, vision-based EKF SLAM implementation goes to Davison (2003). Unfortunately, the computation and storage costs of the EKF SLAM is $O(n^2)$ in the number of landmarks—a problem that many authors have addressed with relative submapping and decorrelation techniques (Bosse, Newman, Leonard, & Teller, 2002;

Csorba, 1997; Frese & Duckett, 2003; Paskin, 2003; Thrun, Koller, Ghahmarani, & Durrant-Whyte, 2002a). Most approaches make use of the observation that conditional dependencies between landmarks are often negligible and hence that the information matrix (inverse covariance matrix) is nearly sparse (Paskin, 2003; Thrun et al., 2002a). This leads naturally to approximate decorrelation techniques in which only a subset of cross-correlation terms are maintained at any given time (Julier, 2003; Paskin, 2003; Thrun et al., 2002a).

In the relative submap approaches, each submap computes an independent EKF solution and submaps are related to each other via a global map of maps (Bosse et al., 2002; Castellanos, Montiel, Neira, & Tardos, 1999; Csorba, 1997; Leonard & Feder, 1999). Julier has shown that to bound robot covariance requires maintaining and updating $O(n)$ cross-correlation terms between the submaps (Julier, 2003). This view is supported by Frese and Duckett (2003), who describes an approach using multigrid relaxation methods that solve the nonlinear least-squares minimization problem in the context of relaxation on a graph. The compressed EKF (CEKF) (Guivant & Nebot, 2001) estimates a subportion of the map and can achieve $O(1)$ updates within a submap. However, propagating changes to the entire map still requires $O(n^2)$ in the number of landmarks. Other approximate constant time algorithms (Knight, Davison, & Reid, 2001; Newman, Leonard, & Rikoski, 2003) use the idea of "postponement," which is similar to the delayed state approach.

The sparse extended-information filter (SEIF) is an approach that aims for constant time complexity (Thrun et al., 2002a). In SEIFs, poses are rolled up into the prior and only the most recent pose is ever active in the state vector. The conditional dependencies that this induces are then dealt with via a "sparsification" routine that deletes weak links between parameters. Hence, like the VSDF, SEIF can end up ignoring conditional dependencies that are induced when the pose history is rolled up into the most current pose estimate. As a result, SEIFs have been shown to be inconsistent (Eustice et al., 2005). Because SEIFs and DSFs use the information formulation, the state and covariance are not directly accessible, and calculating them requires inverting the entire system information matrix [an $O(n^3)$ operation]. SEIFs address this via an approximation that searches for sets of conditionally independent parameters (a Markov blanket) (Thrun, Koller, Ghahmarani, & Durrant-Whyte, 2002b), which can then be extracted without solving for the remaining parameters.

Other algorithms closely related are the thin junction tree filters (TJTF) (Paskin, 2003) and TreeMap (Frese & Schröder, 2006), both of which roll up the process information into the prior and hence do not smooth over the robot path. Both techniques make explicit the connection to graphical modeling (as does GraphSLAM) as an underlying tool for thinking about the SLAM problem. TJTF in particular is a beautiful example of modern inference techniques on graphical models.

With graphical models, information filters or plain least squares such as the Gauss–Newton method, the concept of smoothing an nth-order Markov model is readily apparent—it is clear that the process model encodes conditional dependencies between adjacent poses and that the estimation operates over the n poses in the path. An equivalent solution can be found via the Kalman smoother (Bell & Cathey, 1993; Triggs et al., 2000). However, the Kalman smoother relies on (potentially) nonintuitive notions such as "backward filtering" and "future estimates" and is generally less accessible. With least squares and the relationship to graphical models, smoothing becomes intuitive. Indeed, as Jazwinski notes, the development of smoothing filters was based on least squares in the first place (Jazwinski, 1970).

6. DISCUSSION

That the nonlinear least-squares formulation is not the standard approach for SLAM is perhaps due to the historical need for fast recursive algorithms in online estimation, which lead naturally to the Kalman filter. Unfortunately, the Kalman filter is optimal only for linear problems. To improve the Kalman filter beyond linear problems, one has to add both iterative linearization and smoothing (Bierman, 1977; Gelb, 1974; Jazwinski, 1970; Maybeck, 1979; Sorenson, 1980). As a nonsmoothing, noniterative solution to a nonlinear least-squares problem, the EKF is suboptimal when compared to the full batch SLAM solution. Methods that take the EKF as their baseline for comparison are unlikely to match the batch estimator. On the other hand, a Gauss–Newton batch estimator over k time steps, such as the SWF, is equivalent to an "order-k" iterated extended Kalman *smoother* (Bell, 1994; Kailath, Sayed, & Hassibi, 2000). Here "order" refers to the number of delayed states within the smoother (and not to higher order moment matching or Taylor series expansion). The traditional EKF SLAM solution is simply an order-1 filter. For significantly nonlinear problems, higher order filters will yield better results than lower order filters (Bierman, 1977). The SWF is an approximation to the full SLAM optimization problem that aims to (1) capture the benefits of higher order smoothing and (2) bridge the gap with filtering approaches.

If there is one key lesson learned in this endeavor it is this: we find the optimization approach to estimation greatly superior to filtering. Below are just a few reasons for this *opinion*:

1. Optimization avoids the many pitfalls of filtering, such as early linearization, violation of the Markov assumption, misrepresentation of state uncertainty, covariance inflation "tweaking," and all sorts of ensuing inconsistency issues (Julier, 2003).

2. Optimization is often faster than filtering approaches because there is no need to carry and invert large dense covariance matrices. For instance, key-frame BA outpaces the EKF in modern SLAM systems (Engels et al., 2006; Klein & Murray, 2008; Sibley, Mei, Reid, & Newman, 2010).
3. Nonlinear least-squares optimization is easier to relate to first principles, e.g., the principle of energy minimization (Strang, 1986).
4. The optimization approach lends itself to fast, efficient development cycles in that it relies primarily on one's ability to articulate physically motivated quadratic cost functions.
5. The optimization approach with priors strictly subsumes order-1 filtering—that is, order-1 filtering is a specific form of optimization.

Even though nonlinear least squares was in active use at NASA Jet Propulsion Laboratory in the 1960s, the Kalman filter gained popularity during the Apollo program due to its perceived computational advantage (Schmidt, 1981). Given the extraordinary increase in computer processing power since then, and armed with a better understanding of nonlinear least-squares optimization, we find little reason to employ fully recursive filtering approaches such as the EKF—at least for the kind of problem described here.

One idea left for the future is the possibility of "smart-marginalization" that, instead of rolling delayed states into a dense prior, replaces the prior by a conservative approximation that is efficient to invert. This could be done in a number of ways, such as covariance intersection (Julier & Uhlmann, 2007), a Chow–Liu tree (Chow & Liu, 1968), or simple covariance inflation (Guivant & Nebot, 2003). The motivation behind this idea is twofold because the nonlinear least-squares approach shows that (1) marginalization is potentially dangerous as it locks in linearization errors and (2) even without priors, VO and key-frame BA work very well (Klein & Murray, 2008; Nister et al., 2004). Given these two points, within the overall optimization framework, it is not clear how important it is to keep dense prior information matrices.

7. CONCLUSION

This paper describes a SWF and experimental results of applying it to the problem of stereo vision ground structure estimation during EDL. The efficacy of our approach relies on delayed-state marginalization. Experiments show convergence in surface landmark estimates that match the result predicted by theory. By combining many measurements to efficiently reduce uncertainty, the SWF effectively extends the range resolution of stereo. In the context of EDL, this enables ground structure estimation from greater altitude and hence more time for hazard avoidance prior to touchdown. By tuning the sliding window, the algorithm can scale from exhaustive batch solutions to fast incremental solutions; if the window encompasses all time, the solution is equivalent to BA—if only one time step is maintained, the solution is equivalent to the iterated EKF solution. The SWF is superior to VO in that it has the same computational complexity as VO, yet it shows better convergence and does not suffer from stationary drift.

8. APPENDIX: MATRIX INVERSION LEMMA AND THE SCHUR COMPLEMENT

Below are some useful properties of block matrices and how they are used in recursive filtering. First we look at the problem of computing the inverse of a matrix in terms of its submatrices. This derivation leads to the matrix inversion lemma and makes use of an operation called the Schur complement, which for normal distributions in canonical form turns out to be equivalent to marginalization. The matrix inversion lemma and marginalization via the Schur complement are useful for understanding recursive linear estimation theory. For instance, deriving the Kalman filter from the Bayes rule is greatly simplified by reference to the matrix inversion lemma, and the time step in state estimation is an application of marginalization. Let us say that $\mathbf{M}$ is a large square matrix:

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}$$

that we wish to invert and we know that $\mathbf{A}$ and $\mathbf{D}$ are square and invertible. The first thing is to notice the two following simple matrix multiplications that allow us to triangularize $\mathbf{M}$: first, the following left multiplication creates an upper-right triangular system:

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{C}\mathbf{A}^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \boldsymbol{\Delta}_{\mathbf{A}} \end{bmatrix},$$

and the following right multiplication creates a lower-left triangular system:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{A}^{-1}\mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C} & \boldsymbol{\Delta}_{\mathbf{A}} \end{bmatrix}.$$

The term $\boldsymbol{\Delta}_{\mathbf{A}} = \mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}$ is called the Schur complement of $\mathbf{A}$ in $\mathbf{M}$. Similarly, we can complement $\mathbf{D}$ instead of $\mathbf{A}$:

$$\begin{bmatrix} \mathbf{I} & -\mathbf{B}\mathbf{D}^{-1} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Delta}_{\mathbf{D}} & \mathbf{0} \\ \mathbf{C} & \mathbf{D} \end{bmatrix},$$

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{D}^{-1}\mathbf{C} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Delta}_{\mathbf{D}} & \mathbf{B} \\ \mathbf{0} & \mathbf{D} \end{bmatrix},$$

where $\boldsymbol{\Delta}_{\mathbf{D}} = \mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C}$ is the Schur complement of $\mathbf{D}$ in $\mathbf{M}$.

Combining the above gives two different ways to block diagonalize $\mathbf{M}$:

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{C}\mathbf{A}^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{A}^{-1}\mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Delta}_\mathbf{A} \end{bmatrix},$$

$$\begin{bmatrix} \mathbf{I} & -\mathbf{B}\mathbf{D}^{-1} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{D}^{-1}\mathbf{C} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Delta}_\mathbf{D} \end{bmatrix}.$$

Using these we can reexpress the original matrix $\mathbf{M}$ in terms of a lower-left block triangular component, a block diagonal component, and an upper-right block triangular component. That is,

$$\begin{aligned} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{C}\mathbf{A}^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Delta}_\mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{A}^{-1}\mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{I} & \mathbf{B}\mathbf{D}^{-1} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Delta}_\mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{D}^{-1}\mathbf{C} & \mathbf{I} \end{bmatrix}, \end{aligned}$$

which greatly simplifies computing the inverse because the middle term is block diagonal. For instance,

$$\begin{aligned} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}^{-1} &= \begin{bmatrix} \mathbf{I} & -\mathbf{A}^{-1}\mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Delta}_\mathbf{A}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{C}\mathbf{A}^{-1} & \mathbf{I} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}\mathbf{\Delta}_\mathbf{A}^{-1}\mathbf{C}\mathbf{A}^{-1} & -\mathbf{A}^{-1}\mathbf{B}\mathbf{\Delta}_\mathbf{A}^{-1} \\ -\mathbf{\Delta}_\mathbf{A}^{-1}\mathbf{C}\mathbf{A}^{-1} & \mathbf{\Delta}_\mathbf{A}^{-1} \end{bmatrix}, \end{aligned} \quad \text{(A.1)}$$

and equivalently,

$$\begin{aligned} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}^{-1} &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{D}^{-1}\mathbf{C} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{D}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Delta}_\mathbf{D}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{B}\mathbf{D}^{-1} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{\Delta}_\mathbf{D}^{-1} & -\mathbf{\Delta}_\mathbf{D}^{-1}\mathbf{B}\mathbf{D}^{-1} \\ -\mathbf{D}^{-1}\mathbf{C}\mathbf{\Delta}_\mathbf{D}^{-1} & \mathbf{\Delta}_\mathbf{D}^{-1} + \mathbf{D}^{-1}\mathbf{C}\mathbf{\Delta}_\mathbf{D}^{-1}\mathbf{B}\mathbf{D}^{-1} \end{bmatrix}. \end{aligned} \quad \text{(A.2)}$$

Equating various terms of Eqs. (A.1) and (A.2) yields the different forms of the matrix inverse lemma, one of which is

$$(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} = \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1}.$$

This lemma is one of the primary tricks used to manipulate systems of equations that appear in least-squares methods. For instance, it is used to get between the Gauss–Newton method and the Kalman filter. The Schur complement, when applied to an inverse covariance or information matrix, is equivalent to marginalizing a normal distribution. The matrix inversion lemma can also be used to improve the computational complexity of forward substitution in the SWF from $O(n^3)$ to $O(n^2)$.

In the SWF, note that the reduced system, $[\mathbf{\Lambda_p} - \mathbf{\Lambda}_\mathbf{mp}^T\mathbf{\Lambda}_\mathbf{m}^{-1}\mathbf{\Lambda_{mp}}][\mathbf{x_p}] = [\mathbf{g_p} - \mathbf{\Lambda}_\mathbf{mp}^T\mathbf{\Lambda}_\mathbf{m}^{-1}\mathbf{g_m}]$, involves inverting the map, which would naively be $O(n^3)$. Similar to the standard Kalman filter result, this cost can be reduced to $O(n^2)$ by noting that

$$\begin{aligned} \mathbf{\Lambda}_\mathbf{m}^{-1} &= (\mathbf{\Pi_m} + \mathbf{V})^{-1} \\ &= \left(\mathbf{\Sigma}_\mathbf{m}^{-1} + \mathbf{V}\right)^{-1} \\ &= \left(\mathbf{\Sigma}_\mathbf{m}^{-1} + \mathbf{M}\mathcal{D}\mathbf{M}^T\right)^{-1} \\ &= \mathbf{\Sigma_m} - \mathbf{\Sigma_m}\mathbf{M}(\mathcal{D}^{-1} + \mathbf{M}^T\mathbf{\Sigma_m}\mathbf{M})^{-1}\mathbf{M}^T\mathbf{\Sigma_m}, \end{aligned}$$

where $\Sigma_\mathbf{m} = \mathbf{\Pi}_\mathbf{m}^{-1}$ and $\mathbf{M}\mathcal{D}\mathbf{M}^T$ is a suitable decomposition of $\mathbf{V}$ (which is efficient to compute because $\mathbf{V}$ is block diagonal).

REFERENCES

Bailey, T., Nebot, E., Rosenblatt, J., & Durrant-Whyte, H. (2000, April). Data association for mobile robot navigation: A graph theoretic approach. In Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, CA (pp. 2512–2517).

Bar-Shalom, Y., & Fortmann, T. E. (1988). Tracking and data association. Boston, MA: Academic Press.

Bell, B. M. (1994). The iterated Kalman smoother as a Gauss–Newton method. SIAM Journal on Optimization, 4(3), 626–636.

Bell, B. M., & Cathey, F. W. (1993). The iterated Kalman filter update as a Gauss–Newton method. IEEE Transactions on Automatic Control, 38(2), 294–297.

Bibby, C., & Reid, I. (2007, June). Simultaneous localisation and mapping in dynamic environments (SLAMIDE) with reversible data association. In Proceedings of Robotics: Science and Systems, Atlanta, GA.

Bierman, G. J. (1977). Factorization methods for discrete sequential estimation. Boston, MA: Academic Press.

Bosse, M., Newman, P., Leonard, J., & Teller, S. (2002). An Atlas framework for scalable mapping. MIT Marine Robotics Laboratory.

Brown, D. (1958). A solution to the general problem of multiple station analytical stereo triangulation (Tech. Rep. RCP-MTP, Data Reduction Technical Report No. 43). Patrick Air Force Base, FL (also designated as AFMTC 58-8).

Brown, D. (1976, July). The bundle adjustment—Progress and prospects. In XIIIth Congress of the International Society for Photogrammetry, Helsinki, Finland.

Castellanos, J., Montiel, J., Neira, J., & Tardos, J. (1999). The SPmap: A probabilistic framework for simultaneous localization and map building. IEEE Transactions on Robotics and Automation, 15, 948–952.

Chow, C., & Liu, C. (1968). Approximating discrete probability distributions with dependence trees. IEEE Transactions on Information Theory, 14(3), 462–467.

Cook, S. A. (1971, May). The complexity of theorem-proving procedures. In Third Annual ACM Symposium on Theory of Computing (pp. 151–158).

Csorba, M. (1997). Simultaneous localization and map building. Ph.D. thesis, University of Oxford, Oxford, UK.

Davison, A. J. (2003, October). Real-time simultaneous localisation and mapping with a single camera. In International Conference on Computer Vision (p. 1403).

Deans, M. C. (2005). Bearings-only localization and mapping. Ph.D. thesis, School of Computer Science, Carnegie Mellon University.

DeGroot, M. H., & Schervish, M. J. (2001). Probability and statistics. Boston, MA: Addison Wesley.

Dellaert, F. (2005, June). Square root SAM. In Proceedings of Robotics: Science and Systems, Boston, MA (pp. 1181–1203).

Dennis, J. J., & Schnabel, R. B. (1996). Numerical methods for unconstrained optimization and nonlinear equations. Philadelphia: Society for Industrial and Applied Mathematics.

Di, K., & Li, R. (2004). CAHVOR camera model and its photogrammetric conversion for planetary applications. Journal of Geophysical Research, 109, 9.

Doucet, A., Andrieu, C., & Godsill, S. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. Statistics and Computing, 10(3), 197–208.

Elinas, P., & Little, J. J. (2005). σMCL: Monte-Carlo localization for mobile robots with stereo vision. In Robotics: Science and Systems (pp. 373–380).

Engels, C., Stewenius, H., & Nister, D. (2006). Bundle adjustment rules. In Photogrammetric Computer Vision.

Eustice, R., Singh, H., Leonard, J., Walter, M., & Ballard, R. (2005). Visually navigating the RMS Titanic with SLAM information filters. In Robotics: Science and Systems, (pp. 57–64).

Fitzgibbon, A. W., & Zisserman, A. (2004). Automatic camera recovery for closed or open image sequences. Freiburg, Germany: Springer.

Fox, D., Burgard, W., & Thrun, S. (1999). Markov localization for mobile robots in dynamic environments. Journal of Artificial Intelligence Research, 11, 391–427.

Frese, U. (2005, April). A proof for the approximate sparsity of SLAM information matrices. In Proceedings of the IEEE International Conference on Robotics and Automation, Barcelona, Spain (pp. 331–337).

Frese, U. (2007, April). Efficient 6-DOF SLAM with treemap as a generic backend. In Proceedings of the International Conference on Robotics and Automation, Rome, Italy.

Frese, U., & Duckett, T. (2003, August). A multigrid approach for accelerating relaxation-based SLAM. In Proceedings IJCAI Workshop on Reasoning with Uncertainty in Robotics (RUR 2003), Acapulco, Mexico (pp. 39–46).

Frese, U., & Schröder, L. (2006, October). Closing a million-landmarks loop. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China (pp. 5032–5039).

Galassi, M., Davies, J., Theiler, J., Gough, B., Jungman, G., Booth, M., & Rossi, F. (2003). GNU Scientific Library Reference Manual. Network Theory, Ltd.

Gauss, C. (1821). Theoria combinationis observationum erroribus minimis obnoxiae. Commentationes societatis regiae scientiarum Gottingensis recentiores, 5, 6–93.

Gauss, C. F. (1995). Theory of the combination of observations least subject to error (modern translation). Philadelphia: Society for Industrial and Applied Mathematics.

Gelb, A. (1974). Applied optimal estimation. Cambridge, MA: MIT Press.

Grisetti, G., Stachniss, C., Grzonka, S., & Burgard, W. (2007, June). A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In Proceedings Robotics: Science and Systems, Atlanta, GA.

Guivant, J., & Nebot, E. (2001). Optimization of the simultaneous localization and map-building algorithm for real-time implementation. IEEE Transactions on Robotics and Automation, 17(3), 242–257.

Guivant, J., & Nebot, E. (2003). Solving computational and memory requirements of feature-based simultaneous localization and mapping algorithms. IEEE Transactions on Robotics and Automation, 19(4), 749–755.

Harris, C., & Stephens, M. (1988, August). A combined corner and edge detector. In Proceedings of the Fourth Alvey Vision Conference, Manchester, UK (pp. 147–151).

Hartley, R., & Zisserman, A. (2000). Multiple view geometry in computer vision. Cambridge, UK: Cambridge University Press.

Hirschmüller, H., Innocent, P., & Garibaldi, J. (2002). Fast, unconstrained camera motion estimation from stereo without tracking and robust statistics. In International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore.

Howard, A. (2008, September). Real-time stereo visual odometry for autonomous ground vehicles. In IEEE Conference on Robots and Systems (IROS), Nice, France.

Huber, P. J. (1964). Robust estimation of a location parameter. Annals of Mathematical Statistics, 35(2), 73–101.

Jazwinski, A. H. (1970). Stochastic processes and filtering theory. New York: Academic Press.

Julier, S. J. (2003, September). The stability of covariance inflation methods for SLAM. In International Conference on Intelligent Robots and Systems, Taipei, Taiwan (pp. 2749–2754).

Julier, S. J., & Uhlmann, J. K. (2007). Using covariance intersection for SLAM. Robotics and Autonomous Systems, 55, 3–20.

Kaess, M. (2008). Incremental smoothing and mapping. Ph.D. thesis, Georgia Institute of Technology.

Kaess, M., & Dellaert, F. (2006). Visual SLAM with a multi-camera rig (Tech. Rep. GIT-GVU-06-06). Georgia Institute of Technology.

Kailath, T., Sayed, A. H., & Hassibi, B. (2000). Linear estimation. Upper Saddle River, NJ: Prentice Hall.

Klein, G., & Murray, D. (2008, October). Improving the agility of keyframe-based SLAM. In European Conference on Computer Vision, Marseille, France.

Knight, J. G. H., Davison, A. J., & Reid, I. D. (2001, October). Constant time SLAM using postponement. In Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (pp. 405–413).

Konolige, K., & Agrawal, M. (2007, April). Frame-frame matching for realtime consistent visual mapping. In 2007 IEEE International Conference on Robotics and Automation, Rome, Italy.

Leonard, J., & Feder, H. (1999, October). A computationally efficient method for large-scale concurrent mapping and localization. In D. K. J. Hollerbach and D. Koditschek (Eds.), International Symposium on Robotics Research, Snowbird, UT.

Lu, F., & Milios, E. (1997). Globally consistent range scan alignment for environment mapping. Autonomous Robots, 4(4), 333–349.

Lucas, B. D., & Kanade, T. (1981, August). An iterative image registration technique with an application to stereo vision. In Proceedings of the 7th International Joint Conference on Artificial Intelligence, Vancouver, Canada (pp. 674–697).

Matthies, L., & Shafer, S. (1987). Error modelling in stereo navigation. IEEE Journal of Robotics and Automation, 3(3), 239–248.

Maybeck, P. S. (1979). Stochastic models, estimation, and control, vol. 141 of Mathematics in Science and Engineering. Boston, MA: Academic Press.

McEwen, A. S., Eliason, E. M., Bergstrom, J. W., Bridges, N. T., Hansen, C. J., Delamere, W. A., Grant, J. A., Gulick, V. C., Herkenhoff, K. E., Keszthelyi, L., Kirk, R. L., Mellon, M. T., Squyres, S. W., Thomas, N., & Weitz, C. M. (2007). Mars reconnaissance orbiter's high resolution imaging science experiment (HiRISE). Journal of Geophysical Research, 112, 44.

McLauchlan, P. F. (1999). The variable state dimension filter applied to surface-based structure from motion. University of Surrey.

McLauchlan, P. F., & Murray, D. W. (1995, June). A unifying framework for structure and motion recovery from image sequences. In International Conference on Computer Vision, Cambridge, MA (pp. 314–320).

Mikhail, E. M. (1983). Observations and least squares. Rowman & Littlefield.

Montemerlo, M., & Thrun, S. (2003). FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In International Joint Conferences on Artificial Intelligence (pp. 1151–1156).

Montiel, J., Civera, J., & Davison, A. J. (2006, August). Unified inverse depth parametrization for monocular SLAM. In Proceedings of Robotics: Science and Systems, Philadelphia, PA (pp. 81–88).

Moravec, H. (1980). Obstacle avoidance and navigation in the real world by a seeing robot rover. Ph.D. thesis, Stanford University.

More, J. (1978). The Levenberg–Marquardt algorithm: Implementation and theory. Lecture Notes in Mathematics, 630, 105–116.

Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyse, F., & Sayd, P. (2006, June). Real time localization and 3D reconstruction. In Proceedings of Computer Vision and Pattern Recognition, New York.

Mourikis, A., & Roumeliotis, S. (2007, April). A multi-state constraint Kalman filter for vision-aided inertial navigation. In Proceedings of the IEEE International Conference on Robotics and Automation, Rome, Italy (pp. 3565–3572).

Newman, P. (1999). On the structure and solution of the simultaneous localisation and map building problem. Ph.D. thesis, University of Sydney.

Newman, P., Leonard, J. J., & Rikoski, R. J. (2003, October). Towards constant-time SLAM on an autonomous underwater vehicle using synthetic aperture sonar. In Proceedings of the Eleventh International Symposium on Robotics Research, Siena, Italy.

Newman, P., Sibley, G., Smith, M., Cummins, M., Harrison, A., Mei, C., Posner, I., Shade, R., Schroeter, D., Murphy, L., Churchill, W., Cole, D., & Reid, I. (2009). Navigating, recognizing and describing urban spaces with vision and lasers. International Journal of Robotics Research, 1, 1–28.

Nister, D., Naroditsky, O., & Bergen, J. (2004, July). Visual odometry. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC (pp. 652–659).

Olson, C. F., Matthies, L. H., Schoppers, M., & Maimone, M. W. (2001, May). Stereo egomotion improvements for robust rover navigation. In Proceedings of the IEEE Conference on Robotics and Automation, Seoul, Korea (pp. 1099–1104).

Olson, E., Leonard, J., & Teller, S. (2006, May). Fast iterative alignment of pose graphs with poor initial estimates. In Proceedings of the IEEE International Conference on Robotics and Automation, Orlando, FL (pp. 2262–2269).

Ortega, J., & Rheinboldt, W. C. (1970). Iterative solution of nonlinear equations in several variables. New York: Academic Press.

Paskin, M. A. (2003, August). Thin junction tree filters for simultaneous localization and mapping. In Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico (pp. 1157–1164).

Press, W. H., Flannery, B. P., Teukolsky, S. A., & Vetterling, W. T. (1992). Numerical recipes in C: The art of scientific computing. Cambridge, UK: Cambridge University Press.

Pupilli, M., & Calway, A. (2005, September). Real-time camera tracking using a particle filter. In Proceedings of the British Machine Vision Conference, Oxford, UK.

Qian, G., & Chellappa, R. (2004). Structure from motion using sequential Monte Carlo methods. International Journal of Computer Vision, 59, 5–31.

Ranganathan, A., Kaess, M., & Dellaert, F. (2007, October). Fast 3D pose estimation with out-of-sequence measurements. In IEEE Conference on Intelligent Robots and Systems, San Diego, CA.

Rosten, E., & Drummond, T. (2006, May). Machine learning for high-speed corner detection. In European Conference on Computer Vision, Graz, Austria.

Rousseeuw, P. J., & Leroy, A. M. (1987). Robust regression and outlier detection. Hoboken, NJ: Wiley.

Schmidt, S. F. (1981). The Kalman filter: Its recognition and development for aerospace applications. Journal of Guidance and Control, 4(1), 4–7.

Sibley, G. (2006). Sliding window filters for SLAM (Tech. Rep. CRES-06-004). Center for Robotics and Embedded Systems, University of Southern California.

Sibley, G., Mei, C., Reid, I., & Newman, P. (2010). Vast scale outdoor navigation using adaptive relative bundle adjustment. International Journal of Robotics Research, 29, 958–980.

Sibley, G., Sukhatme, G., & Matthies, L. (2006, August). The iterated sigma point Kalman filter with applications to long range stereo. In Robotics: Science and Systems, Philadelphia, PA (pp. 263–270).

Smith, R. C., & Cheeseman, P. (1986). On the representation and estimation of spatial uncertainty. International Journal of Robotics Research, 5(4), 56–68.

Smith, R. C., Self, M., & Cheeseman, P. (1990). Estimating uncertain spatial relationships in robotics. In I. J. Cox and G. T. Wilfong (Eds.), Autonomous robot vehicles (pp. 167–193). Berlin, Germany: Springer-Verlag.

Sorenson, H. W. (1980). Parameter estimation: Principles and problems. New York: Marcel Dekker.

Sorenson, H. W., & Stubberud, A. R. (1968). Non-linear filtering by approximation of the posteriori density. International Journal of Control, 8(1), 33–51.

Strang, G. (1986). Introduction to applied mathematics. Cambridge, UK: Cambridge University Press.

Tapley, B., Schutz, B., & Born, G. H. (2004). Statistical orbit determination. Boston, MA: Academic Press.

Thrun, S., Burgard, W., & Fox, D. (2005). Probabilistic robotics. Cambridge, MA: MIT Press.

Thrun, S., Koller, D., Ghahmarani, Z., & Durrant-Whyte, H. (2002a, December). Simultaneous mapping and localization with sparse extended information filters: Theory and initial results. In Workshop on the Algorithmic Foundations of Robotics, Nice, France.

Thrun, S., Koller, D., Ghahmarani, Z., & Durrant-Whyte., H. (2002b, December). SLAM updates require constant time. In Workshop on the Algorithmic Foundations of Robotics, Nice, France.

Trawny, N., Mourikis, A., Roumeliotis, S., Johnson, A., & Montgomery, J. (2007). Vision-aided inertial navigation for pin-point landing using observations of mapped landmarks. Journal of Fied Robotics, 25(5), 357–378.

Triggs, B., McLauchlan, P. F., Hartley, R. I., & Fitzgibbon, A. W. (2000). Bundle adjustment—A modern synthesis. In ICCV '99: Proceedings of the International Workshop on Vision Algorithms (pp. 298–372). London: Springer-Verlag.

Vanicek, P., & Krakiwsky, E. (1986). Geodesy—The concepts. New York: Elsevier Science Publishing Co.

Wolf, P. R., & Ghilani, C. D. (1997). Adjustment computations: statistics and least squares in surveying and GIS. Wiley-Interscience.

Zhang, Z. (1997). Parameter estimation techniques: A tutorial with application to conic fitting. Image and Vision Computing Journal, 15(2676), 59–76.