Exactly Sparse Delayed-State Filters for View-Based SLAM

Ryan M. Eustice, *Member, IEEE*, Hanumant Singh, *Member, IEEE*, and John J. Leonard, *Member, IEEE*

***Abstract*—This paper reports the novel insight that the simultaneous localization and mapping (SLAM) information matrix is *exactly* sparse in a delayed-state framework. Such a framework is used in view-based representations of the environment that rely upon scan-matching raw sensor data to obtain virtual observations of robot motion with respect to a place it has previously been. The exact sparseness of the delayed-state information matrix is in contrast to other recent feature-based SLAM information algorithms, such as sparse extended information filter or thin junction-tree filter, since these methods have to make approximations in order to force the feature-based SLAM information matrix to be sparse. The benefit of the exact sparsity of the delayed-state framework is that it allows one to take advantage of the information space parameterization without incurring any sparse approximation error. Therefore, it can produce equivalent results to the full-covariance solution. The approach is validated experimentally using monocular imagery for two datasets: a test-tank experiment with ground truth, and a remotely operated vehicle survey of the RMS Titanic.**

***Index Terms*—Information filters, Kalman filtering, machine vision, mobile robot motion planning, mobile robots, recursive estimation, robot vision systems, simultaneous localization and mapping (SLAM), underwater vehicles.**

I. Introduction

GOOD NAVIGATION is often a prerequisite for many of the tasks assigned to mobile robotics. This is especially true in the underwater realm where unmanned underwater vehicles (UUVs) have increasingly become part of the standard toolkit of deep-water science. The scientists who use these vehicles have come to demand that they be capable of colocating data both spatially and temporally across a range of varying applications; examples include studies of biodiversity [1], coral-reef health [2], plume tracking [3]–[5], microbathymetry mapping [6]–[8], and deep-sea archeology [9]–[11]. Since global positioning systems (GPS) signals do not penetrate the ocean surface, engineers most often resort to acoustic-beacon networks [12], [13] to meet the large-area, bounded-error, precision navigation requirements of scientists. The disadvantage of this method, however, is that it requires the deployment, calibration, and eventual recovery of the transponder net. While this is often an acceptable tradeoff for long-term deployments, it frequently is the bane of short-term surveys.

In more recent years, UUVs have seen significant advances in their dead-reckoning capabilities. The advent of sensors such as the acoustic Doppler velocity log (DVL) [14] and north-seeking fiber-optic gyro (FOG) [15] have enabled underwater vehicles to navigate with reported error bounds of less than 1% of distance traveled [16]. For shorter duration missions, this level of precision can often be quite satisfactory, but for longer duration, large-area missions, the unbounded accumulation of error is typically intolerable.

A. Visually Augmented Navigation

In an effort to overcome current underwater navigation limitations, Eustice *et al.* [17] presented a simultaneous localization and mapping (SLAM) technique for near-seafloor navigation called visually augmented navigation (VAN). Their technique incorporates pairwise camera constraints from low-overlap imagery to constrain the vehicle position estimate and "reset" the accumulated navigation drift error. In this framework, the camera provides measurements of the six-degree-of-freedom (DOF) relative coordinate transformation between poses modulo scale. The method recursively incorporates these relative-pose constraints by estimating the global poses that are consistent with the camera measurements and navigation prior. These global poses correspond to samples from the robot's trajectory acquired at image acquisition and, therefore, unlike the typical feature-based SLAM estimation problem, which keeps track of the current robot pose and an associated landmark map, the VAN state vector consists entirely of historical vehicle states corresponding to the vehicle poses at the times the images were captured. This delayed-state approach corresponds to a view-based representation of the environment (Fig. 1), which can be traced back to a batch scan-matching method by Lu and Milios [18] using laser data, a delayed-decision-making framework by Leonard and Rikoski [19] for feature initialization with sonar data, and the hybrid batch/recursive formulations by Fleischer [20] and McLauchlan [21] using camera images. In this context, scan-matching raw images results in virtual

Manuscript received September 29, 2005; revised February 27, 2006. This paper was recommended for publication by Associate Editor W. Burgard and Editor L. Parker upon evaluation of the reviewers' comments. This work was supported in part by the Center for Subsurface Sensing and Imaging Systems (CenSSIS) Engineering Research Center of the National Science Foundation under Grant EEC-9986821, in part by the Woods Hole Oceanographic Institution through a grant from the Penzance Foundation, and in part by a National Defense Science and Engineering Graduate (NDSEG) Fellowship awarded through the Department of Defense. This paper was presented in part at the IEEE International Conference on Robotics and Automation, Barcelona, Spain, April 2005.

R. Eustice was with the Joint Program in Oceanographic Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139 USA, and the Woods Hole Oceanographic Institution, Woods Hole, MA 02543 USA. He is now with the Department of Naval Architecture and Marine Engineering, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: eustice@umich.edu).

H. Singh is with the Department of Applied Ocean Physics and Engineering, Woods Hole Oceanographic Institution, Woods Hole, MA 02543 USA (e-mail: hanu@whoi.edu).

J. Leonard is with the Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: jleonard@mit.edu).

Digital Object Identifier 10.1109/TRO.2006.886264

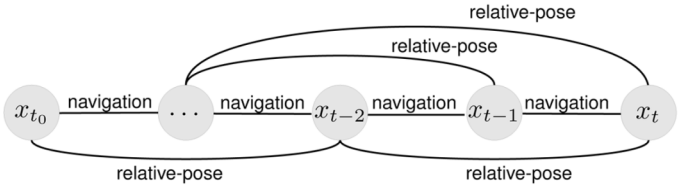


Fig. 1. System diagram for a view-based representation. The model is comprised of a graph where the nodes correspond to historical robot poses, and edges represent either Markov (navigation) or non-Markov (relative-pose) constraints.

observations of robot motion with respect to a place it has previously visited.

The VAN technique proposed the use of an extended Kalman filter (EKF) as the fusion framework for merging the navigation and camera sensor measurements. This is a well-known approach, whose application to SLAM was originally developed by Smith *et al.* [22], [23] and Moutarlier and Chatila [24]. The EKF maintains the joint correlations over all elements in the state vector and, therefore, can update estimates of all the elements involved in key events like loop closure. Maintaining these joint correlations, however, represents a significant computational burden, since each measurement update requires quadratic complexity in the size of the state vector. This limits the online use of an EKF to relatively small maps (e.g., for the VAN approach, this equates to an upper bound of approximately 100 six-vector poses).

The EKF's quadratic computational complexity has long been a recognized issue within the SLAM community, and has lead to a great deal of research being directed towards scalable large-area SLAM algorithms. Notable large-area approaches include submaps [25]–[27], postponement [28]–[30], Rao–Blackwellized particle-filtering techniques [31], [32], and covariance intersection [33]. In addition to this body of work, promising new approaches for scalable SLAM have appeared in the recent literature, and are based upon exploiting sparsity in the Gaussian "information form" [34]–[38].

B. A Scalable Framework

To our knowledge, the earliest related work that exploited the efficiency of the measurement update in the inverse covariance form was published by McLauchlan and Murray [39], in the context of recursive structure-from-motion (SFM). This work was subsequently extended to realize a hybrid batch/recursive visual SLAM implementation that unified recursive SLAM and bundle adjustment [21]. McLauchlan recognized the potential increase in efficiency that can be gained via approximations to maintain sparsity of the information matrix.

> It has long been known in the photogrammetry community, in the form of the equivalent normal formulation, that the [information] matrix ... takes a special sparse form in the context of reconstruction ... [However, in a recursive formulation] ... eliminating motion fills in the structure blocks. This has to be avoided to maintain update times proportional to n. So our *partial elimination adjustment* method is to ignore corrections that fill in zero blocks, while applying the correction to the blocks which are already nonzero.

While the consistency implications of this approximation are unknown, in practice, the method achieved results approaching those of a full batch solution for moderate-duration image sequences.

Recently, the SLAM community has also turned its attention to exploring the information parameterization for increased efficiency. In particular, published approaches include the sparse extended information filter (SEIF) [34], the thin junction-tree filter (TJTF) [35], and Treemap filters [37]. The authors of these algorithms make the important empirical observation, first noted in [34] and later proved in [40], that when the feature-based SLAM posterior is cast in the form of the extended information filter (EIF), (i.e., the dual of the EKF), many of the off-diagonal elements in the information matrix are near zero when properly normalized. These new feature-based SLAM information algorithms approximate the posterior with a sparse representation, and thereby prevent weak interlandmark links from forming. This approach (effectively) bounds the density of the information matrix and, as each author shows, allows for constant time updates. The delicate and nontrivial issue that must be dealt with, however, is "how to sparsify the information matrix?," since this approximation can lead to global map inconsistency [41], [42].

Interestingly, it is the same phenomenon that plagues both the information formulations of McLauchlan and Murray [21], [39], as well as the feature-based SLAM algorithms of Thrun *et al.* [34], Paskin [35], and Frese [37], and that is "eliminating motion fills in the structure blocks." Eliminating the robot's trajectory causes the SLAM landmark posterior to densify, destroying any sparsity [38], [43] and, hence, any efficiency associated with a sparse representation. This is why *all* feature-based SLAM information algorithms are founded upon some type of pruning strategy that removes weak constraints.

In the following, we illustrate why the feature-based SLAM information matrix is naturally dense, and therefore, why SEIF and TJTF have to *approximate* the SLAM posterior with a sparse representation. We then continue by introducing the novel insight that the information form is *exactly* sparse for a delayed-state representation. This inherent sparsity allows us to cast the delayed-state framework in an efficient representation, but without any sparse approximation error. We call this result "exactly sparse delayed-state filters (ESDFs)." Benchmark results quantifying the ESDF's efficiency with respect to the standard EKF formulation are shown for a controlled laboratory dataset. In addition, real-world results for a recent remotely operated vehicle (ROV) survey of the wreck of the RMS Titanic are presented.

II. The Information Form

A. An Alternative Parameterization of the Gaussian

The information form is often called the canonical or natural representation of the Gaussian distribution. This notion of a natural representation stems from expanding the quadratic in the

exponential of the Gaussian distribution as

$$\begin{aligned}
p(\boldsymbol{\xi}_t) &= \mathcal{N}(\boldsymbol{\xi}_t; \boldsymbol{\mu}_t, \Sigma_t) \\
&= \frac{1}{\sqrt{|2\pi\Sigma_t|}} \exp\left\{-\frac{1}{2}(\boldsymbol{\xi}_t - \boldsymbol{\mu}_t)^\top \Sigma_t^{-1}(\boldsymbol{\xi}_t - \boldsymbol{\mu}_t)\right\} \\
&= \frac{1}{\sqrt{|2\pi\Sigma_t|}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{\xi}_t^\top \Sigma_t^{-1}\boldsymbol{\xi}_t - 2\boldsymbol{\mu}_t^\top \Sigma_t^{-1}\boldsymbol{\xi}_t \right.\right. \\
&\quad \left.\left. + \boldsymbol{\mu}_t^\top \Sigma_t^{-1}\boldsymbol{\mu}_t\right)\right\} \\
&= \frac{e^{-\frac{1}{2}\boldsymbol{\mu}_t^\top \Sigma_t^{-1}\boldsymbol{\mu}_t}}{\sqrt{|2\pi\Sigma_t|}} \exp\left\{-\frac{1}{2}\boldsymbol{\xi}_t^\top \Sigma_t^{-1}\boldsymbol{\xi}_t + \boldsymbol{\mu}_t^\top \Sigma_t^{-1}\boldsymbol{\xi}_t\right\} \\
&= \frac{e^{-\frac{1}{2}\boldsymbol{\eta}_t^\top \Lambda_t^{-1}\boldsymbol{\eta}_t}}{\sqrt{|2\pi\Lambda_t^{-1}|}} \exp\left\{-\frac{1}{2}\boldsymbol{\xi}_t^\top \Lambda_t\boldsymbol{\xi}_t + \boldsymbol{\eta}_t^\top \boldsymbol{\xi}_t\right\} \\
&= \mathcal{N}^{-1}(\boldsymbol{\xi}_t; \boldsymbol{\eta}_t, \Lambda_t)
\end{aligned}$$

where

$$\Lambda_t = \Sigma_t^{-1} \quad \text{and} \quad \boldsymbol{\eta}_t = \Lambda_t\boldsymbol{\mu}_t. \tag{1}$$

The result is that rather than parameterizing the normal distribution in terms of its mean and covariance, $\mathcal{N}(\boldsymbol{\xi}_t; \boldsymbol{\mu}_t, \Sigma_t)$, it is instead parametrized in terms of its information vector and information matrix, $\mathcal{N}^{-1}(\boldsymbol{\xi}_t; \boldsymbol{\eta}_t, \Lambda_t)$ [44]. Here, "natural" refers to the fact that the exponential is parameterized directly in terms of the information vector and matrix without the need for completing the matrix square.

B. Marginalization and Conditioning

The covariance and information representations lead to very different computational characteristics with respect to the fundamental probabilistic operations of marginalization and conditioning. This is important, because these two operations appear at the core of any SLAM algorithm, for example, motion prediction and measurement updates. Table I summarizes these operations on a Gaussian distribution where we see that the covariance and information representations exhibit a dual relationship with respect to marginalization and conditioning. For example, marginalization is easy in the covariance form, since it corresponds to extracting the appropriate subblock from the covariance matrix, while in the information form, it is hard, because it involves calculating the Schur complement over the variables we wish to keep. Note that the opposite relation holds true for conditioning, which is easy in the information form and hard in the covariance form.

TABLE I
SUMMARY OF MARGINALIZATION AND CONDITIONING OPERATIONS ON A GAUSSIAN DISTRIBUTION EXPRESSED IN COVARIANCE AND INFORMATION FORM

$p(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \mathcal{N}(\left[\begin{smallmatrix}\boldsymbol{\mu}_\alpha \\ \boldsymbol{\mu}_\beta\end{smallmatrix}\right], \left[\begin{smallmatrix}\Sigma_{\alpha\alpha} & \Sigma_{\alpha\beta} \\ \Sigma_{\beta\alpha} & \Sigma_{\beta\beta}\end{smallmatrix}\right]) = \mathcal{N}^{-1}(\left[\begin{smallmatrix}\boldsymbol{\eta}_\alpha \\ \boldsymbol{\eta}_\beta\end{smallmatrix}\right], \left[\begin{smallmatrix}\Lambda_{\alpha\alpha} & \Lambda_{\alpha\beta} \\ \Lambda_{\beta\alpha} & \Lambda_{\beta\beta}\end{smallmatrix}\right])$

	Marginalization $p(\boldsymbol{\alpha}) = \int p(\boldsymbol{\alpha}, \boldsymbol{\beta}) d\boldsymbol{\beta}$	**Conditioning** $p(\boldsymbol{\alpha}\|\boldsymbol{\beta}) = p(\boldsymbol{\alpha}, \boldsymbol{\beta})/p(\boldsymbol{\beta})$
Cov. Form	$\boldsymbol{\mu} = \boldsymbol{\mu}_\alpha$ $\Sigma = \Sigma_{\alpha\alpha}$	$\boldsymbol{\mu}' = \boldsymbol{\mu}_\alpha + \Sigma_{\alpha\beta}\Sigma_{\beta\beta}^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu}_\beta)$ $\Sigma' = \Sigma_{\alpha\alpha} - \Sigma_{\alpha\beta}\Sigma_{\beta\beta}^{-1}\Sigma_{\beta\alpha}$
Info. Form	$\boldsymbol{\eta} = \boldsymbol{\eta}_\alpha - \Lambda_{\alpha\beta}\Lambda_{\beta\beta}^{-1}\boldsymbol{\eta}_\beta$ $\Lambda = \Lambda_{\alpha\alpha} - \Lambda_{\alpha\beta}\Lambda_{\beta\beta}^{-1}\Lambda_{\beta\alpha}$	$\boldsymbol{\eta}' = \boldsymbol{\eta}_\alpha - \Lambda_{\alpha\beta}\boldsymbol{\beta}$ $\Lambda' = \Lambda_{\alpha\alpha}$

III. FEATURE-BASED SLAM INFORMATION FILTERS

Most SLAM approaches are feature-based, which assumes that the robot can extract an abstract representation of features in the environment from its sensor data and then use reobservation of these features for localization [22]. In this approach, a landmark map is explicitly built and maintained. The process of concurrently performing localization *and* feature map building are inherently coupled, thereby implying that the robot must then represent a joint distribution over landmarks and current pose. Using the EKF to represent these coupled errors requires maintaining the cross-correlations in the covariance matrix, in which there are quadratically many. Updating the joint correlations over map and robot leads to an $\mathcal{O}(n^2)$ complexity per update, with n being the number of landmarks in the map.

A. Sparsity Yields Efficiency

As stated earlier, some substantial papers have recently appeared in the literature in which the authors explore reparameterizing the feature-based SLAM posterior in the information form [34]–[37]. For example, Thrun *et al.* [34] make the observation that when the EIF is used for inference, measurement updates are additive and efficient. The downside of the EIF is that motion prediction is *generally* $\mathcal{O}(n^3)$; however, if the information matrix obeys a certain *sparse* structure, the EIF motion prediction can be performed in *constant* time. To obtain the requisite sparse structure, Thrun *et al.* make an important empirical observation regarding the architecture of the feature-based SLAM information matrix. They show that when properly normalized, many of the interlandmark constraints in the information matrix are redundant and weak. Based upon this insight, the methods presented in [34] and [35] try to approximate the information matrix with a sparse representation in which these weak interlandmark constraints are eliminated, allowing for efficient inference.

The delicate issue that must be dealt with in these approaches, though, is how to perform the necessary approximation step to keep the information matrix sparse. In fact, the sparsification step is an important issue, not to be glossed over, because the feature-based SLAM information matrix associated with the joint posterior over robot pose, $\mathbf{x}_t$, and landmark map $\mathbf{M}$, given sensor measurements $\mathbf{z}^t$ and control inputs $\mathbf{u}^t$ (i.e., $p(\mathbf{x}_t, \mathbf{M}|\mathbf{z}^t, \mathbf{u}^t)$) is naturally fully *dense*. As we show next, this density arises from marginalizing out past robot poses.

B. Filtering Causes Fill In

To see that marginalization results in fill in, consider the diagram shown in Fig. 2. We begin with the schematic shown to the upper left, which represents the robot $\mathbf{x}_t$ at time t connected to three landmarks $\mathbf{L}_1, \mathbf{L}_2$, and $\mathbf{L}_3$ in the context of a Markov random field (MRF) [45], [46] (a.k.a. Markov network). The shown Markov network depicts a graphical representation of the conditional independencies in the distribution $p(\mathbf{x}_t, \mathbf{L}_{1:3}|\mathbf{z}^t, \mathbf{u}^t)$, and indicates that the only constraints that

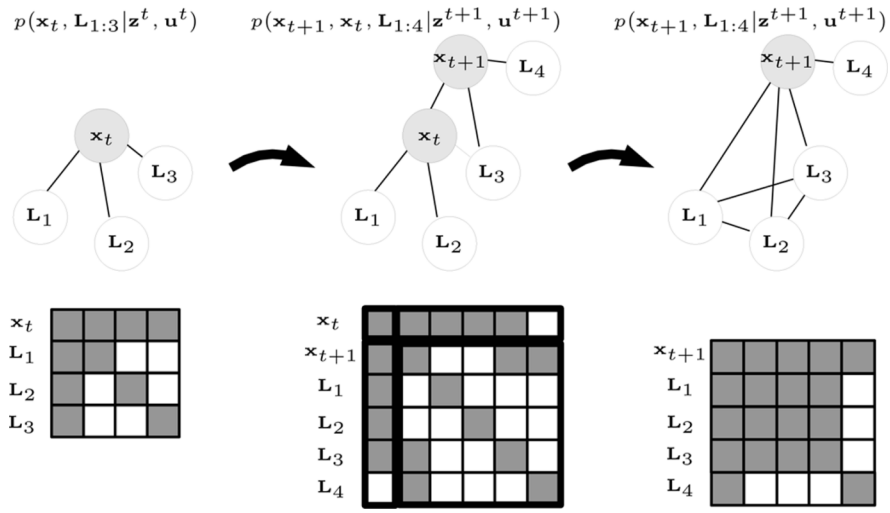


Fig. 2. Graphical explanation of why the feature-based SLAM information matrix is naturally fully dense. (left) The posterior over robot pose $\mathbf{x}_t$ and landmarks $\mathbf{L}_{1:3}$, given sensor measurements $\mathbf{z}^t$ and control inputs $\mathbf{u}^t$, is represented as a Markov network. The corresponding information matrix is shown directly below, and encodes the graphical link structure within the nonzero off-diagonal elements. (middle) The time propagation of the posterior is now shown, where the state vector has been augmented to include the robot pose at time $t+1$ (i.e., $\mathbf{x}_{t+1}$), reobservation of landmark $\mathbf{L}_3$, and observation of a new landmark $\mathbf{L}_4$. Appropriate subblocks of the information matrix have been outlined in bold to differentiate the relevant portions involved in marginalizing out the past robot pose $\mathbf{x}_t$. Referring to Table I, $\Lambda_{\alpha\alpha}$ is the lower right block, $\Lambda_{\beta\beta}$ is the upper left block, and $\Lambda_{\alpha\beta} = \Lambda_{\beta\alpha}^\top$ are the two rectangular blocks. (right) This posterior depicts the effect of marginalizing out the past robot state $\mathbf{x}_t$ with its consequent "fill in" of the information matrix.

exist are between the robot and landmarks (i.e., no interlandmark constraints appear). This lack of interlandmark constraints should be correctly interpreted to mean that each landmark is conditionally independent, given the robot pose as described in [31] and [47]. The intuition behind this comes from viewing the noise of each sensor reading as being independent, and therefore, determining each landmark position is an independent estimation problem given the *known* location of the sensor.

Directly below each Markov network in Fig. 2 is an illustration of the corresponding information matrix. Here we see that the nonzero off-diagonal elements encode the robot/landmark constraints, while the zeros in the information matrix encode the lack of direct interlandmark constraints [35]. Shown in the middle of Fig. 2 is the intermediate distribution $p(\mathbf{x}_{t+1}, \mathbf{x}_t, \mathbf{L}_{1:4}|\mathbf{z}^t, \mathbf{u}^{t+1})$. This distribution represents a time propagation of the previous distribution by augmenting the state vector to include the term $\mathbf{x}_{t+1}$ (i.e., the new robot pose at time $t+1$), reobservation of feature $\mathbf{L}_3$, and observation of a new landmark $\mathbf{L}_4$. Because the robot state evolves according to a first-order Markov process, we see that the new robot state $\mathbf{x}_{t+1}$ is only linked to the previous robot state $\mathbf{x}_t$, and that observation of the landmarks $\mathbf{L}_3$ and $\mathbf{L}_4$ add two additional constraints to $\mathbf{x}_{t+1}$. In the typical feature-based SLAM approach, only the current robot pose is estimated, and not the complete trajectory. Therefore, we always marginalize out the previous robot pose $\mathbf{x}_t$ during our time-projection step to give the distribution over current pose and map, $p(\mathbf{x}_{t+1}, \mathbf{L}_{1:4}|\mathbf{z}^t, \mathbf{u}^{t+1}) = \int p(\mathbf{x}_{t+1}, \mathbf{x}_t, \mathbf{L}_{1:4}|\mathbf{z}^t, \mathbf{u}^{t+1}) d\mathbf{x}_t$. Recalling the formula for marginalization applied to a Gaussian in the information form (see Table I), we note that it is the matrix outer product of $\Lambda_{\alpha\beta}\Lambda_{\beta\beta}^{-1}\Lambda_{\alpha\beta}^\top$ (where $\boldsymbol{\alpha} = \{\mathbf{x}_{t+1}, \mathbf{L}_{1:4}\}$ and $\boldsymbol{\beta} = \mathbf{x}_t$) that causes the information matrix to fill in and become dense. This result is shown in the rightmost graph of Fig. 2.

Intuitively, the landmarks $\mathbf{L}_1, \mathbf{L}_2, \mathbf{L}_3$, which used to be indirectly connected via a direct relationship with $\mathbf{x}_t$, must now represent that indirect relationship directly by creating new links between each other. Therefore, the penalty for a feature-based SLAM representation that always marginalizes out the robot trajectory is that the landmark Markov network becomes fully connected, and the associated information matrix becomes fully *dense* (though, as previously mentioned, [34] makes the empirical observation that many of the off-diagonal elements are relatively small).

IV. Exactly Sparse Delayed-State Filters

An alternative formulation of the SLAM problem is to use a view-based representation rather than a feature-based approach [17], [18], [48]. View-based representations do not explicitly model landmark features in the environment. Instead, the estimation problem consists of tracking the current robot pose in conjunction with a collection of historical poses sampled from the robot's trajectory. The associated posterior is then defined over a collection of delayed states [17]–[20]. In the view-based representation, raw sensor data is registered to provide virtual observations of pose displacements. For example, in [18] and [48], these virtual observations come from scan-matching raw laser range data, while in our application [17], these virtual observations come from registering overlapping optical imagery (via the Essential matrix). Algorithm 1 provides an outline of the overall ESDF algorithmic procedure whose details we discuss next.

A. State Augmentation

We begin by describing the method of state augmentation, which is how we "grow" the state vector to contain a new delayed state. This operation occurs whenever we have a new view that we wish to store. For example, in our VAN framework, we

add a delayed state for each acquired image of the environment that we wish to be able to revisit at a later time.

1) Adding a Delayed State: Assume for the moment that our estimate at time t is described by the following distribution expressed in both covariance and information form:

$$p(\mathbf{x}_t, \mathbf{M}|\mathbf{z}^t, \mathbf{u}^t) = \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_{x_t}\\ \boldsymbol{\mu}_M\end{bmatrix}, \begin{bmatrix}\Sigma_{x_t x_t} & \Sigma_{x_t M}\\ \Sigma_{M x_t} & \Sigma_{MM}\end{bmatrix}\right) = \mathcal{N}^{-1}\left(\begin{bmatrix}\boldsymbol{\eta}_{x_t}\\ \boldsymbol{\eta}_M\end{bmatrix}, \begin{bmatrix}\Lambda_{x_t x_t} & \Lambda_{x_t M}\\ \Lambda_{M x_t} & \Lambda_{MM}\end{bmatrix}\right).$$

This distribution represents a map $\mathbf{M}$ and current robot state $\mathbf{x}_t$, given all measurements $\mathbf{z}^t$ and control inputs $\mathbf{u}^t$. Here, the map variable $\mathbf{M}$ is used in a general sense; for example, it could represent a collection of delayed states or a set of landmark features in the environment. For now, we do not care, because we want to show what happens when we augment our representation to include the time-propagated robot state $\mathbf{x}_{t+1}$, obtaining the distribution $p(\mathbf{x}_{t+1}, \mathbf{x}_t, \mathbf{M}|\mathbf{z}^t, \mathbf{u}^{t+1})$, which can be factored as

$$\begin{aligned} &p(\mathbf{x}_{t+1}, \mathbf{x}_t, \mathbf{M}|\mathbf{z}^t, \mathbf{u}^{t+1})\\ &\quad = p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{M}, \mathbf{z}^t, \mathbf{u}^{t+1}) p(\mathbf{x}_t, \mathbf{M}|\mathbf{z}^t, \mathbf{u}^{t+1})\\ &\quad \overset{\text{Markov}}{=} p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_{t+1}) p(\mathbf{x}_t, \mathbf{M}|\mathbf{z}^t, \mathbf{u}^t). \end{aligned} \tag{2}$$

In (2), we factored the posterior into the product of a probabilistic state transition multiplied by our prior using the common assumption that the robot state evolves according to a first-order Markov process. Equation (3) describes the general nonlinear discrete-time Markov robot motion model we assume, and (4) its first-order linearized form where F is the Jacobian evaluated at $\boldsymbol{\mu}_{x_t}$ and $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathrm{Q})$ is the white process noise

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_{t+1}) + \mathbf{w}_t \tag{3}$$

$$\approx \mathbf{f}(\boldsymbol{\mu}_{x_t}, \mathbf{u}_{t+1}) + \mathrm{F}(\mathbf{x}_t - \boldsymbol{\mu}_{x_t}) + \mathbf{w}_t. \tag{4}$$

Note that, in general, our robot state description $\mathbf{x}_t$ consists of both pose (i.e., position and orientation) *and* kinematic components (e.g., body-frame velocities, angular rates).

2) Augmentation in the Covariance Form: Under the linearized approximation (4), the augmented state distribution (2) is also Gaussian, and in covariance form its result is given by [22]

$$p(\mathbf{x}_{t+1}, \mathbf{x}_t, \mathbf{M}|\mathbf{z}^t, \mathbf{u}^{t+1}) = \mathcal{N}(\boldsymbol{\mu}'_{t+1}, \Sigma'_{t+1})$$

$$\text{where} \quad \boldsymbol{\mu}'_{t+1} = \begin{bmatrix}\mathbf{f}(\boldsymbol{\mu}_{x_t}, \mathbf{u}_{t+1})\\ \boldsymbol{\mu}_{x_t}\\ \boldsymbol{\mu}_M\end{bmatrix}$$

$$\Sigma'_{t+1} = \begin{bmatrix}(\mathrm{F}\Sigma_{x_t x_t}\mathrm{F}^\top + \mathrm{Q}) & \mathrm{F}\Sigma_{x_t x_t} & \mathrm{F}\Sigma_{x_t M}\\ \Sigma_{x_t x_t}\mathrm{F}^\top & \Sigma_{x_t x_t} & \Sigma_{x_t M}\\ \Sigma_{M x_t}\mathrm{F}^\top & \Sigma_{M x_t} & \Sigma_{MM}\end{bmatrix}. \tag{5}$$

The lower-right 2×2 subblock of Σ'_{t+1} corresponds to the covariance between the delayed-state element $\mathbf{x}_t$ and the map $\mathbf{M}$, and has remained unchanged from the prior. Meanwhile, the first row and column contain the cross-covariances associated with the time propagated robot state $\mathbf{x}_{t+1}$, which includes the effect of the process model.

3) Augmentation in the Information Form: Having obtained the delayed-state distribution in covariance form, we can now transform (5) to its information form (6). This requires inversion of the 3×3 block covariance matrix Σ'_{t+1} whose tedious derivation we omit here, though note that (6) can be verified by the fact that $\Lambda'_{t+1}\Sigma'_{t+1} = \mathrm{I}$ and $\boldsymbol{\eta}'_{t+1} = \Lambda'_{t+1}\boldsymbol{\mu}'_{t+1}$

$$\begin{aligned} &p(\mathbf{x}_{t+1}, \mathbf{x}_t, \mathbf{M}|\mathbf{z}^t, \mathbf{u}^{t+1})\\ &\quad = \mathcal{N}^{-1}(\boldsymbol{\eta}'_{t+1}, \Lambda'_{t+1})\\ &\boldsymbol{\eta}'_{t+1} = \begin{bmatrix}\mathrm{Q}^{-1}\left(\mathbf{f}\left(\boldsymbol{\mu}_{x_t}, \mathbf{u}_{t+1}\right) - \mathrm{F}\boldsymbol{\mu}_{x_t}\right)\\ \boldsymbol{\eta}_{x_t} - \mathrm{F}^\top\mathrm{Q}^{-1}\left(\mathbf{f}\left(\boldsymbol{\mu}_{x_t}, \mathbf{u}_{t+1}\right) - \mathrm{F}\boldsymbol{\mu}_{x_t}\right)\\ \boldsymbol{\eta}_M\end{bmatrix}\\ &\Lambda'_{t+1} = \begin{bmatrix}\mathrm{Q}^{-1} & -\mathrm{Q}^{-1}\mathrm{F} & \boxed{0}\\ -\mathrm{F}^\top\mathrm{Q}^{-1} & \Lambda_{x_t x_t} + \mathrm{F}^\top\mathrm{Q}^{-1}\mathrm{F} & \Lambda_{x_t M}\\ \underbrace{\boxed{0}}_{\text{key result}} & \Lambda_{M x_t} & \Lambda_{MM}\end{bmatrix}. \end{aligned} \tag{6}$$

4) Markovity Yields Exact Sparseness: Equation (6) provides a key insight into the structure of the information matrix regarding delayed states. We see that augmenting our state vector to include the time-propagated robot state $\mathbf{x}_{t+1}$ introduces shared information only between it and the previous robot state $\mathbf{x}_t$. Moreover, the shared information between $\mathbf{x}_{t+1}$ and the map $\mathbf{M}$ is *always* zero, irrespective of what $\mathbf{M}$ abstractly represents (i.e., regardless of whether $\mathbf{M}$ represents a set of landmarks or a collection of delayed states, the result will always be zero). This sparsity in the augmented state information matrix is a direct consequence of the Markov property associated with the state transition probability $p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_{t+1})$, which states that $\mathbf{x}_{t+1}$ is only conditionally dependent upon its previous state $\mathbf{x}_t$. In terms of a graphical Markov network, we can trivially arrive at the same sparsity pattern as (6) by recognizing that the time-propagated state $\mathbf{x}_{t+1}$ is only linked to its parent node $\mathbf{x}_t$ via the state-transition probability $p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_{t+1})$, as per (2) and, therefore, is conditionally independent of $\mathbf{M}$ (i.e., shares no links).

By induction, a key property of state augmentation in the information form is that if we continue to augment our state vector with additional delayed states, the information matrix will exhibit a block tridiagonal structure linking each delayed state with only the post and previous states

$$\begin{bmatrix}\Lambda_{x_{t+1}x_{t+1}} & \Lambda_{x_{t+1}x_t} & & & \\ \Lambda^\top_{x_{t+1}x_t} & \Lambda_{x_t x_t} & \Lambda_{x_t x_{t-1}} & & \\ & \Lambda^\top_{x_t x_{t-1}} & \Lambda_{x_{t-1}x_{t-1}} & \Lambda_{x_{t-1}x_{t-2}} & \\ & & \ddots & \ddots & \ddots\end{bmatrix}. \tag{7}$$

Hence, the view-based SLAM delayed-state information matrix is *naturally* sparse without having to make any approximations.

B. Measurement Updates

One of the very attractive properties of the information form is that measurement updates are constant-time and additive in an EIF [34]. This is in contrast to an EKF's quadratic complexity *per* update. Assume the following general nonlinear measurement function (8) and its first-order linearized form (9):

$$\mathbf{z}_t = \mathbf{h}(\boldsymbol{\xi}_t) + \mathbf{v}_t \tag{8}$$

$$\approx \mathbf{h}(\bar{\boldsymbol{\mu}}_t) + \mathrm{H}(\boldsymbol{\xi}_t - \bar{\boldsymbol{\mu}}_t) + \mathbf{v}_t \tag{9}$$

where $\boldsymbol{\xi}_t$ is the predicted state vector distributed according to $\boldsymbol{\xi}_t \sim \mathcal{N}(\bar{\boldsymbol{\mu}}_t, \bar{\Sigma}_t) \equiv \mathcal{N}^{-1}(\bar{\boldsymbol{\eta}}_t, \bar{\Lambda}_t)$, $\mathbf{v}_t$ is the white measurement noise $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathrm{R})$, and H is the Jacobian evaluated at $\bar{\boldsymbol{\mu}}_t$. The EKF covariance update requires computing the Kalman gain and updating $\bar{\boldsymbol{\mu}}_t$ and $\bar{\Sigma}_t$ via [44]

$$\begin{aligned} \mathrm{K} &= \bar{\Sigma}_t \mathrm{H}^\top (\mathrm{H}\bar{\Sigma}_t \mathrm{H}^\top + \mathrm{R})^{-1} \\ \boldsymbol{\mu}_t &= \bar{\boldsymbol{\mu}}_t + \mathrm{K}(\mathbf{z}_t - \mathbf{h}(\bar{\boldsymbol{\mu}}_t)) \\ \Sigma_t &= (\mathrm{I} - \mathrm{KH})\bar{\Sigma}_t (\mathrm{I} - \mathrm{KH})^\top + \mathrm{KRK}^\top. \end{aligned} \tag{10}$$

This calculation nontrivially modifies all elements in the covariance matrix resulting in quadratic computational complexity per update [22]. In contrast, the corresponding EIF update is given by [34]

$$\begin{aligned} \boldsymbol{\eta}_t &= \bar{\boldsymbol{\eta}}_t + \mathrm{H}^\top \mathrm{R}^{-1}(\mathbf{z}_t - \mathbf{h}(\bar{\boldsymbol{\mu}}_t) + \mathrm{H}\bar{\boldsymbol{\mu}}_t) \\ \Lambda_t &= \bar{\Lambda}_t + \mathrm{H}^\top \mathrm{R}^{-1} \mathrm{H}. \end{aligned} \tag{11}$$

1) ESDF Updates Are Constant-Time: Equation (11) shows that the information matrix is additively updated by the outer product term $\mathrm{H}^\top \mathrm{R}^{-1} \mathrm{H}$. In general, this outer product modifies all elements of the predicted information matrix $\bar{\Lambda}_t$; however, a key observation is that the SLAM Jacobian, H, is always sparse [34]. For example, in the VAN framework, pairwise registration of images I_i and I_j provides a relative-pose measurement (modulo scale) between states $\mathbf{x}_i$ and $\mathbf{x}_j$ resulting in a sparse Jacobian of the form

$$\mathrm{H} = \left[0 \cdots \quad \frac{\partial \mathbf{h}}{\partial \mathbf{x}_i} \quad \cdots 0 \cdots \quad \frac{\partial \mathbf{h}}{\partial \mathbf{x}_j} \quad \cdots 0 \right].$$

As a result, only the four block elements corresponding to $\mathbf{x}_i$ and $\mathbf{x}_j$ of the information matrix need to be modified. In particular, the information in the diagonal blocks $\bar{\Lambda}_{x_i x_i}$ and $\bar{\Lambda}_{x_j x_j}$ is increased, while new information appears at $\bar{\Lambda}_{x_i x_j}$ and its symmetric counterpart $\bar{\Lambda}_{x_j x_i}$. This new off-diagonal information reflects the addition of a new edge (i.e., constraint) into the corresponding Markov network, linking the nodes $\mathbf{x}_i$ and $\mathbf{x}_j$.

2) ESDF Updates Use Linear Storage: Putting (7) together with (11), we see that an important consequence of the delayed-state framework is that the total number of *nonzero* off-diagonal elements in the information matrix is *linear* in the number of delayed states and relative-pose constraints for a bounded graph structure (Fig. 3). Hence, without any approximation, a view-based representation is exactly sparse and, furthermore, requires only linear storage. In our application, we control the degree of sparsity by bounding the number of image registrations that the robot may attempt per state augmentation. In other words, the robot is only allowed to hypothesize k possible candidate images (where $k = 5$ in our application) for attempted registration with the current view; this leads to at most $2nk$ non-Markov off-diagonal constraints in the resulting information matrix.

3) ESDF Update and Measurement Correlation: As a side note, it is worth pointing out that (8) assumes that the measurements are corrupted by time-independent noise. Since scan-matching methods rely upon registering raw data, this criterion may be violated if data is reused. In our VAN framework, relative-pose measurements are generated by pairwise registration of images with common overlap. As typical underwater optical survey trajectories consist of a boustrophedon pattern and low frame rates (to reduce the amount of power expended on illumination), this implies that overall spatial image overlap tends to be low. Therefore, we assume that most pairwise camera measurements in our application are weakly (if at all) correlated, as derived in [49] and, hence, we do not directly enforce the exclusion of data reuse.[1] For the general case, however, measurement independence should be ensured by using a set of raw data correspondences only once, so that scan-matching measurements remain statistically independent.

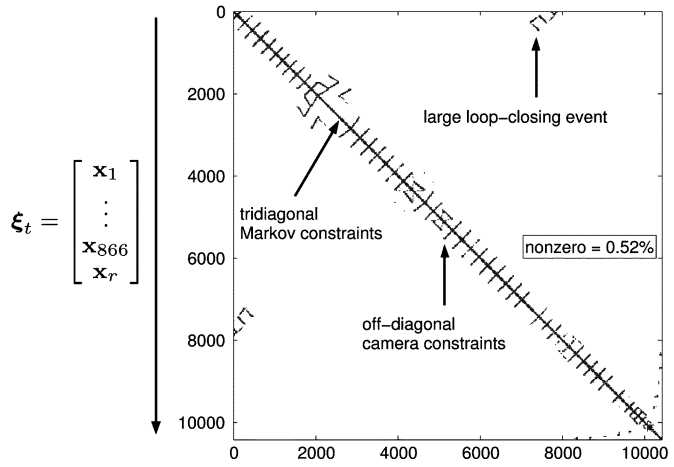


Fig. 3. Topology of the view-based SLAM information matrix. This figure highlights the exact sparsity of view-based SLAM using the information matrix from the RMS Titanic survey of Section VI-B. In all, there are 867 delayed states where each state is a 12-vector consisting of six pose and six kinematic components. The resulting information matrix is a $10\,404 \times 10\,404$ matrix with only 0.52% nonzero elements.

C. Motion Prediction

Motion prediction corresponds to a time propagation of the robot's state from time t to time $t+1$. In (6), we derived an expression in the information form for the augmented distribution containing the time predicted robot state $\mathbf{x}_{t+1}$, and its previous state $\mathbf{x}_t$; in other words, $p(\mathbf{x}_{t+1}, \mathbf{x}_t, \mathbf{M} | \mathbf{z}^t, \mathbf{u}^{t+1})$. To derive the time propagated distribution $p(\mathbf{x}_{t+1}, \mathbf{M} | \mathbf{z}^t, \mathbf{u}^{t+1})$, note that all that is required is to simply marginalize out the previous state $\mathbf{x}_t$ from (6). Referring to Table I for marginalization of a Gaussian in the information form, we have[2]

$$\begin{aligned} & p(\mathbf{x}_{t+1}, \mathbf{M} | \mathbf{z}^t, \mathbf{u}^{t+1}) \\ &\quad = \int p(\mathbf{x}_{t+1}, \mathbf{x}_t, \mathbf{M} | \mathbf{z}^t, \mathbf{u}^{t+1}) d\mathbf{x}_t \\ &\quad = \mathcal{N}^{-1}(\bar{\boldsymbol{\eta}}_{t+1}, \bar{\Lambda}_{t+1}) \\ & \bar{\boldsymbol{\eta}}_{t+1} = \begin{bmatrix} \mathrm{Q}^{-1} \left(\mathbf{f}\left(\boldsymbol{\mu}_{x_t}, \mathbf{u}_{t+1}\right) - \mathrm{F}\boldsymbol{\mu}_{x_t} \right) \\ \boldsymbol{\eta}_M \end{bmatrix} \\ &\qquad - \begin{bmatrix} -\mathrm{Q}^{-1}\mathrm{F} \\ \Lambda_{M x_t} \end{bmatrix} \Omega^{-1} \boldsymbol{\eta}^\star_{x_t} \\ &\quad = \begin{bmatrix} \mathrm{Q}^{-1}\mathrm{F}\Omega^{-1}\boldsymbol{\eta}_{x_t} + \Psi\left(\mathbf{f}\left(\boldsymbol{\mu}_{x_t}, \mathbf{u}_{t+1}\right) - \mathrm{F}\boldsymbol{\mu}_{x_t}\right) \\ \boldsymbol{\eta}_M - \Lambda_{M x_t} \Omega^{-1} \boldsymbol{\eta}^\star_{x_t} \end{bmatrix} \end{aligned}$$

[1] [49] provides a derivation showing that the correlation between two pairwise image registrations sharing a common image is null (low) if the shared feature set is empty (small).

[2] The simplification of Ψ employs the matrix inversion lemma

$$(\mathrm{A} + \mathrm{BCB}^\top)^{-1} = \mathrm{A}^{-1} - \mathrm{A}^{-1}\mathrm{B}(\mathrm{B}^\top \mathrm{A}^{-1}\mathrm{B} + \mathrm{C}^{-1})^{-1}\mathrm{B}^\top \mathrm{A}^{-1}.$$

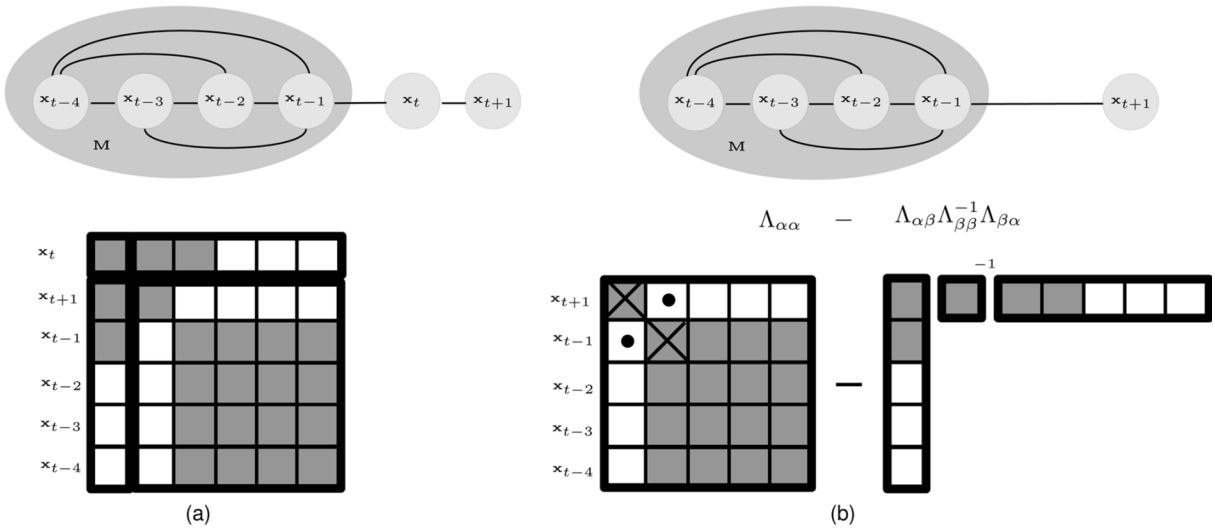


Fig. 4. ESDF motion prediction is constant-time. Shown above is a graphical illustration of the effect of motion prediction within a delayed-state framework. (a) The Markov network for a segregated collection of delayed states. The view-based "map" $\mathbf{M}$ is composed of the set $\mathbf{M} = \{\mathbf{x}_{t-4}, \mathbf{x}_{t-3}, \mathbf{x}_{t-2}, \mathbf{x}_{t-1}\}$, which is a collection of delayed states that are interlinked by camera constraints. The previous and predicted robot states, $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$, respectively, are serially linked to the map. Below the Markov network is a schematic showing the nonzero structure (colored in gray) of the associated information matrix. (b) Recalling from Table I the expression for marginalization of a Gaussian in information form, we see that the rightmost schematic illustrates this operation graphically. The end result is that only the states that were linked to $\mathbf{x}_t$ (i.e., $\mathbf{x}_{t-1}$ and $\mathbf{x}_{t+1}$) are affected by the marginalization operation, as indicated by the cross-hairs and black dots superimposed on $\Lambda_{\alpha\alpha}$.

$$\begin{aligned}\bar{\Lambda}_{t+1} &= \begin{bmatrix} \mathrm{Q}^{-1} & 0 \\ 0 & \Lambda_{MM} \end{bmatrix} - \begin{bmatrix} -\mathrm{Q}^{-1}\mathrm{F} \\ \Lambda_{Mx_t} \end{bmatrix} \Omega^{-1} \begin{bmatrix} -\mathrm{F}^\top\mathrm{Q}^{-1} & \Lambda_{x_tM} \end{bmatrix} \\ &= \begin{bmatrix} \Psi & \mathrm{Q}^{-1}\mathrm{F}\Omega^{-1}\Lambda_{x_tM} \\ \Lambda_{Mx_t}\Omega^{-1}\mathrm{F}^\top\mathrm{Q}^{-1} & \Lambda_{MM} - \Lambda_{Mx_t}\Omega^{-1}\Lambda_{x_tM} \end{bmatrix} \end{aligned} \tag{12}$$

and

$$\begin{aligned}\boldsymbol{\eta}_{x_t}^\star &= \boldsymbol{\eta}_{x_t} - \mathrm{F}^\top\mathrm{Q}^{-1}\left(\mathbf{f}\left(\boldsymbol{\mu}_{x_t}, \mathbf{u}_{t+1}\right) - \mathrm{F}\boldsymbol{\mu}_{x_t}\right) \\ \Omega &= \Lambda_{x_tx_t} + \mathrm{F}^\top\mathrm{Q}^{-1}\mathrm{F} \\ \Psi &= \mathrm{Q}^{-1} - \mathrm{Q}^{-1}\mathrm{F}\Omega^{-1}\mathrm{F}^\top\mathrm{Q}^{-1} \\ &= \mathrm{Q}^{-1} - \mathrm{Q}^{-1}\mathrm{F}\left(\mathrm{F}^\top\mathrm{Q}^{-1}\mathrm{F} + \Lambda_{x_tx_t}\right)^{-1}\mathrm{F}^\top\mathrm{Q}^{-1} \\ &= \left(\mathrm{Q} + \mathrm{F}\Lambda_{x_tx_t}^{-1}\mathrm{F}^\top\right)^{-1}.\end{aligned}$$

1) ESDF Prediction Is Constant-Time: An important consequence of the delayed-state framework is that (12) can be implemented in constant-time. To see this, we refer to Fig. 4, which illustrates the effect of motion prediction for a collection of delayed states. We begin with the Markov network of Fig. 4(a), showing a segregated collection of delayed states. Our view-based "map" corresponds to the set of states $\mathbf{M} = \{\mathbf{x}_{t-4}, \mathbf{x}_{t-3}, \mathbf{x}_{t-2}, \mathbf{x}_{t-1}\}$, which have an interconnected dependence due to camera measurements, while the states $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$ are only serially connected, and correspond to the previous and predicted robot states, respectively. Referring back to Table I, we see that Fig. 4(b) illustrates the effect of marginalization on the information matrix. We note that since $\mathbf{x}_t$ is only serially connected to $\mathbf{x}_{t+1}$ and $\mathbf{x}_{t-1}$, marginalizing it out only requires modifying the information blocks associated with these elements (i.e., $\Lambda'_{x_{t+1}x_{t+1}}$ and $\Lambda'_{x_{t-1}x_{t-1}}$, denoted with cross-hairs, and the symmetric blocks $\Lambda'_{x_{t+1}x_{t-1}} = \Lambda'^\top_{x_{t-1}x_{t+1}}$, denoted with black dots). Therefore, since only a fixed portion of the information matrix is ever involved in the calculation of (12), motion prediction can be performed in constant-time. This is an important result, since, in practice, the fusion of asynchronous navigation sensor measurements (e.g., odometry, compass) implies that prediction is typically a high-bandwidth operation (e.g., $\mathcal{O}$ (10 Hz) or more).

D. State Recovery

The information form of the Gaussian is parameterized by its information vector and information matrix, $\boldsymbol{\eta}_t$ and Λ_t, respectively. However, the expressions for motion prediction (12) and measurement update (11) additionally require subelements from the state mean vector $\boldsymbol{\mu}_t$, so that the nonlinear models (3) and (8) can be linearized. Therefore, in order for the information form to be a computationally efficient parameterization for delayed states, we also need to be able to easily recover portions of the state mean vector. Fortunately, this is the case due to the sparse structure of the information matrix Λ_t.

1) Full State Recovery: Naïve recovery of the state estimate through matrix inversion results in cubic complexity and destroys any efficiency gained over the EKF. Fortunately, closer inspection reveals that the recovery of the state mean $\boldsymbol{\mu}_t$ can be posed more efficiently as solving a sparse, symmetric, positive-definite, linear system of equations

$$\Lambda_t\boldsymbol{\mu}_t = \boldsymbol{\eta}_t. \tag{13}$$

Such systems can be solved via the classic iterative method of conjugate gradients (CG) [50]. In general, CG can solve this system in n iterations with $\mathcal{O}(n)$ cost per iteration, where n is the size of the state vector (i.e., $\mathcal{O}(n^2)$ total cost), and often in

many fewer iterations if the initialization is good [51]. In addition, since the state mean $\boldsymbol{\mu}_t$ typically does not change significantly with each measurement update (excluding key events like loop closure), this relaxation can take place over *multiple time steps* using a fixed number of iterations per update, as pioneered by Duckett *et al.* [52] and Thrun *et al.* [34]. The caveat being that a fixed number of iterations does not necessarily guarantee convergence and, hence, optimal state recovery within an individual time step [53].

Additionally, a couple of recently developed multilevel relaxation SLAM algorithms have appeared in the literature that propose linear asymptotic complexity. These new techniques, by Konolige [51] and Frese *et al.* [54], propose to achieve the computational reduction by subsampling poses and performing the relaxation over multiple spatial resolutions. Borrowing multigrid relaxation techniques pioneered in the early 1970s for solving discretized partial differential equations (PDEs) [55], the key idea is that spatial subsampling improves relaxation convergence rates. Frese *et al.* further take advantage of this by applying their multilevel method incrementally over time.

2) Partial State Recovery: An important observation regarding the expressions for motion prediction (12) and measurement updates (11) is that they only require knowing *subsets* of the state mean $\boldsymbol{\mu}_t$. In light of this, we note that rather than always solving for the complete state mean vector $\boldsymbol{\mu}_t$, we can partition (13) into two sets of coupled equations as

$$\begin{bmatrix} \Lambda_{\ell\ell} & \Lambda_{\ell b} \\ \Lambda_{b\ell} & \Lambda_{bb} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu}_\ell \\ \boldsymbol{\mu}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{\eta}_\ell \\ \boldsymbol{\eta}_b \end{bmatrix}. \tag{14}$$

This partitioning of $\boldsymbol{\mu}_t$ into what we call the "local portion" of the map, $\boldsymbol{\mu}_\ell$, and the "benign portion," $\boldsymbol{\mu}_b$, allows us to suboptimally solve for local portions of the map in constant-time. By holding our current estimate for $\boldsymbol{\mu}_b$ fixed, we can solve (14) for an estimate of $\boldsymbol{\mu}_\ell$ as

$$\hat{\boldsymbol{\mu}}_\ell = \Lambda_{\ell\ell}^{-1}(\boldsymbol{\eta}_\ell - \Lambda_{\ell b}\hat{\boldsymbol{\mu}}_b). \tag{15}$$

Equation (15) provides us with a method for recovering an estimate of the local map $\hat{\boldsymbol{\mu}}_\ell$, provided that our estimate for the benign portion $\hat{\boldsymbol{\mu}}_b$ is a decent approximation to the actual mean, $\boldsymbol{\mu}_b$. Furthermore, note that only a *subset* of $\hat{\boldsymbol{\mu}}_b$ is actually required in the calculation of $\hat{\boldsymbol{\mu}}_\ell$ corresponding to the nonzero elements in the sparse matrix $\Lambda_{\ell b}$. In terms of Thrun *et al.*'s notation [34], this active subset, denoted $\boldsymbol{\mu}_b^+$, represents the Markov blanket of $\boldsymbol{\mu}_\ell$ and corresponds to elements that are directly connected to $\boldsymbol{\mu}_\ell$ in the associated Markov network. Therefore, calculation of the local map $\hat{\boldsymbol{\mu}}_\ell$ only requires an estimate of the *locally* connected delayed-state network $\hat{\boldsymbol{\mu}}_b^+$, and does not depend upon passive elements in the benign portion of the map.

In particular, we use (15) to provide an accurate and constant-time approximation for recovering the robot mean during motion prediction (12), and during incorporation of high-bandwidth navigation sensor measurements (11). Since the robot state is only serially connected to the map, $\Lambda_{\ell b}$ has only one nonzero block element (Section IV-C). Therefore, solving for the robot mean is constant-time. Note, though, that (15) will only provide a good approximation so long as the active mean estimate $\hat{\boldsymbol{\mu}}_b^+$ is accurate. In the case that it is not (e.g., as a result of loop closure), then the true full mean $\boldsymbol{\mu}_t$ should be recovered via (13).

E. Data Association

Traditionally, the problem of data association is addressed by evaluating the likelihood of a measurement for different correspondence hypotheses [56]. However, obtaining the requisite prior requires marginalizing out all elements in the state estimate except for the subset of variables of interest (i.e., cubic complexity in the information form, see Table I). To sidestep this difficulty, Thrun *et al.* [34], [57] instead proposed using conditional likelihoods based upon extracting elements within the appropriate Markov blanket. Their method inverted this submatrix to obtain a conditional covariance subblock that they then used for data association. While [34] and [57] reported success using this conditional covariance, it can be shown to yield overconfident likelihood estimates [58].

Alternatively, Eustice *et al.* [58] derived a method for obtaining conservative estimates for the marginal covariances. Their technique stems from posing the relationship $\Lambda_t\Sigma_t = \mathrm{I}$ as a sparse system of linear equations $\Lambda_t\Sigma_{*i} = \mathbf{e}_i$, where Σ_{*i} and $\mathbf{e}_i$ denote the ith columns of the covariance and identity matrices, respectively. They show that this relationship allows for efficient determination of the robot's covariance column and, based upon this, offer a novel algorithm for inferring conservative marginal covariances useful for data association.

For our VAN application, we employ the technique of [58] to infer the map marginal covariances and associated robot cross-correlation. This allows us to compute relative Euclidean distances and first-order uncertainty estimates between the current robot pose and all other stored poses (i.e., linear time complexity). Based upon this, we infer the probability of overlap and select the k most likely views ($k = 5$ in our application) as candidates for registration [49].

Algorithm 1: Summary of the ESDF algorithm as implemented within the context of VAN.

Require $\boldsymbol{\eta}_{t_0}, \Lambda_{t_0}, \boldsymbol{\mu}_{t_0}$ {*a priori* robot estimate}

1: AUGMENT_FLAG $\leftarrow 0$

2: **loop** {perform SLAM}

3: $\quad \mathbf{u}_{t+1} \leftarrow$ current control input

4: $\quad$ **if** AUGMENT_FLAG **then**

5: $\quad\quad$ Time-propagate state from t to $t+1$ via augmentation

$\quad\quad\quad \bar{\boldsymbol{\eta}}_{t+1}, \bar{\Lambda}_{t+1} \leftarrow (6)|_{\boldsymbol{\eta}_t, \Lambda_t, \boldsymbol{\mu}_t, \mathbf{u}_{t+1}}$

6: $\quad\quad$ AUGMENT_FLAG $\leftarrow 0$

7: $\quad$ **else**

8: $\quad\quad$ Time-propagate state from t to $t+1$ via motion prediction

$\quad\quad\quad \bar{\boldsymbol{\eta}}_{t+1}, \bar{\Lambda}_{t+1} \leftarrow (12)|_{\boldsymbol{\eta}_t, \Lambda_t, \boldsymbol{\mu}_t, \mathbf{u}_{t+1}}$

9: **end if**

10: Perform partial-state recovery for robot

$\bar{\boldsymbol{\mu}}_{t+1} \leftarrow (15)|_{\bar{\boldsymbol{\eta}}_{t+1},\bar{\Lambda}_{t+1},\boldsymbol{\mu}_t}$

11: **if** navigation sensor measurement $\mathbf{z}_{t+1}$ **then**

12: Apply measurement update

$\boldsymbol{\eta}_{t+1}, \Lambda_{t+1} \leftarrow (11)|_{\mathbf{z}_{t+1},\bar{\boldsymbol{\eta}}_{t+1},\bar{\Lambda}_{t+1},\bar{\boldsymbol{\mu}}_{t+1}}$

13: Perform partial-state recovery for robot

$\boldsymbol{\mu}_{t+1} \leftarrow (15)|_{\boldsymbol{\eta}_{t+1},\Lambda_{t+1},\bar{\boldsymbol{\mu}}_{t+1}}$

14: **else if** new image frame I_{t+1} **then**

15: AUGMENT_FLAG $\leftarrow 1$

16: Extract and encode image interest points (Section VI).

17: Propose the k most likely candidate views for attempted registration with I_{t+1} (Section IV-E).

18: $\mathbf{z}_{t+1} \leftarrow \emptyset$ {initialize measurement vector}

19: **for** $i = 1$ to k **do**

20: Attempt image registration between I_i and I_{t+1}.

21: **if** registration success **then**

22: Add pose constraint to measurement vector

$\mathbf{z}_{t+1} \leftarrow \left[\mathbf{z}_i^\top \ \mathbf{z}_{t+1}^\top\right]^\top$

23: **end if**

24: **end for**

25: **if** $\mathbf{z}_{t+1} \neq \emptyset$ **then**

26: Apply measurement update

$\boldsymbol{\eta}_{t+1}, \Lambda_{t+1} \leftarrow (11)|_{\mathbf{z}_{t+1},\bar{\boldsymbol{\eta}}_{t+1},\bar{\Lambda}_{t+1},\bar{\boldsymbol{\mu}}_{t+1}}$

27: Perform full-state recovery

$\boldsymbol{\mu}_{t+1} \leftarrow (13)|_{\boldsymbol{\eta}_{t+1},\Lambda_{t+1},\bar{\boldsymbol{\mu}}_{t+1}}$

28: Update data association bounds (Section IV-E).

29: **end if**

30: **end if**

31: **end loop**

V. Discussion

A. Connection to Lu–Milios

The concept of a view-based map representation has strong roots going back to a seminal paper by Lu and Milios [18]. Their approach sidestepped difficulties associated with feature segmentation and representation by doing away with an explicit feature-based parameterization of the environment. Rather, their technique indirectly represented a physical map via a collection of global robot poses and raw scan data. To determine the global poses, they formulated the nonlinear optimization problem as one of estimating a set of global robot poses consistent with the relative-pose constraints obtained by scan matching and odometry. They then solved this sparse nonlinear optimization problem in an batch-iterative fashion. Our ESDF framework essentially attempts to recursively solve the same problem. Note, though, that in the ESDF framework, the nonlinear relative-pose constraints are only linearized *once* about the current state when the measurement is incorporated via (11), while in the noncausal Lu–Milios batch formulation, they are *relinearized* around the current best estimate of the state at each iteration of the nonlinear optimization. This implies that while the ESDF solution can be performed recursively, it will be more prone to linearization error.

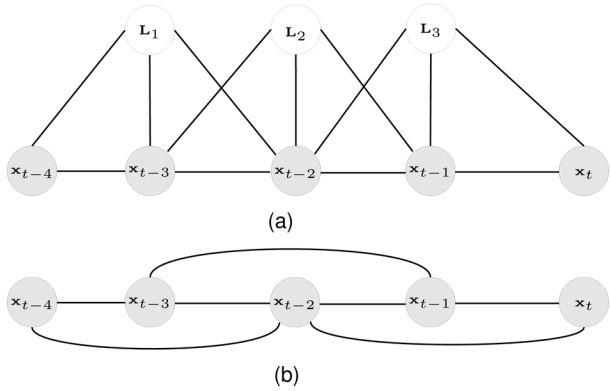


Fig. 5. Depiction of the ESDF's connection to feature-based SLAM. Conceptually, view-based SLAM can be idealized as marginalizing out the landmarks (i.e., $\mathbf{L}_1, \mathbf{L}_2, \mathbf{L}_3$), which in turn causes edges to appear between spatially proximal samples from the robot's trajectory (i.e., $\mathbf{x}_{t-4}, \ldots, \mathbf{x}_t$). (a) The SLAM posterior over landmarks and trajectory as represented by a Markov network. (b) The corresponding delayed-state Markov network after marginalizing out the landmarks.

B. Connection to Feature-Based SLAM

Another interesting theoretical connection involves relating the delayed-state SLAM framework to feature-based SLAM. In Section III, we saw that the feature-based SLAM information matrix is naturally dense as a result of marginalizing out the robot's trajectory. On a similar train of thought, conceptually we can view the off-diagonal elements appearing in the delayed-state SLAM information matrix as being a result of marginalizing out the *landmarks* (Fig. 5). Since landmarks are only ever locally observed, they only create links to spatially close robot states. Therefore, each time we eliminate a landmark, it introduces a new off-diagonal entry into the information matrix that links all robot states that observed that landmark.

Interestingly, this same type of constraint phenomenon also appears in photogrammetry and, in particular, in large-scale, batch, bundle-adjustment techniques [59]. These techniques are based upon a partitioned Levenberg–Marquardt algorithm that takes advantage of the inherent sparsity between camera and 3-D feature constraints to efficiently solve the batch reconstruction problem. Their central component is based upon eliminating 3-D-structure equations to yield a coupled set of equations over camera poses that can be solved and then back-substituted to recover the associated 3-D structure. This strategy of

3-D structure elimination to yield a coupled set of equations over cameras, is reminiscent of the conceptual notion of marginalizing out the landmarks in SLAM to a graph over poses. Therefore, loosely speaking, the VAN ESDF framework represents an online, linearized formulation of this same camera recovery problem.

C. Connection to FastSLAM

Our approach has an interesting relationship with other recent SLAM algorithms that are based upon Rao–Blackwellized particle filters [31], [47]. Montemerlo *et al.*'s factored solution to SLAM (FastSLAM) exploits the property that individual feature estimates are conditionally independent, given perfect knowledge of the vehicle trajectory. Different possible instantiations of the trajectory are represented as particles in a particle filter, where each trajectory has its own set of estimated feature locations. If each particle in FastSLAM did not represent the *complete* vehicle trajectory, then the conditional independence assumptions they exploit would no longer apply. Our approach also exploits this same conditional independence property, and must keep a history of vehicle poses in the state vector to maintain sparsity. Our method, however, does not estimate feature locations explicitly, but rather applies constraints derived from measurements of the same features at multiple poses to compute updates to the entire vehicle trajectory. The extension of FastSLAM to accommodate such pose constraints presents an interesting question and warrants future research, especially in the context of dealing with the issue of particle depletion [60].

D. Connection to High Data Rates

Finally, note that a view-based representation is still applicable, even with much higher perceptual data rates (e.g., laser scans, video). While the ESDF framework is general, and supports trajectory sampling at any rate, for practical reasons, it may be prudent to decimate the trajectory in order to control the rate of growth of the resulting state vector. The key idea is that we are not required to sample the trajectory at our perceptual update rate, but rather, we can sample it at an appropriate spatial decimation that is sufficient for relocalization.

For example, in our underwater VAN application, a digital still image is collected every few seconds from a down-looking monocular camera. Since this typically results in sequential frame overlap of the order of 15%–35%, we include all frames into our view-based map representation. However, in the general case where video frame rates are available, we can selectively sample key frames from the video sequence to serve as spatial "anchor points" in a view-based map. Reobservation of these key frames (coupled with successful image registration) provides a zero-drift spatial measurement of robot motion allowing for loop closure. Furthermore, we can exploit the higher frame rates to get an improved estimate of visual odometry by performing a local bundle adjustment over all frames occurring between temporally consecutive anchor images (Fig. 6). This would provide a more rigid constraint between sampled poses than a simple pairwise registration would [61]. While this may not make optimal use of the intersample data, it represents a practical online compromise.

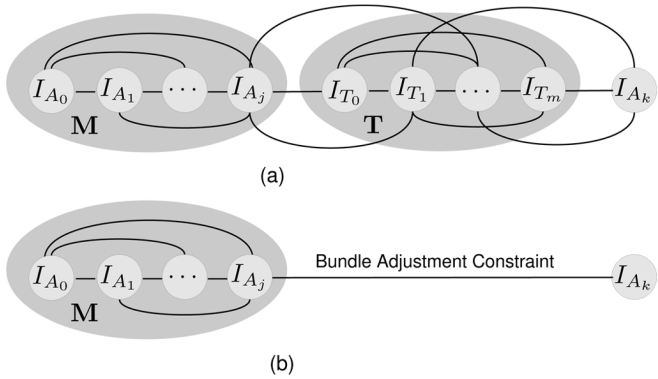


Fig. 6. Extension of view-based SLAM to video frame rates. (a) Our collection of anchor images $\{I_{A_0}, \ldots, I_{A_j}\}$ represents a subsampling of the available video image sequence, and serves as our view-based spatial map $\mathbf{M}$. Given higher frame rates, we can exploit the additional views between temporally consecutive anchor images I_{A_j} and I_{A_k} to get an improved estimate of incremental motion. The improved motion estimate comes from a local bundle adjustment that includes the temporary frame set $\mathbf{T} = \{I_{T_0}, \ldots, I_{T_m}\}$. (b) The result is a serial constraint between I_{A_j} and I_{A_k} that is more rigid than a single pairwise measurement between the pair.

VI. Results

This section presents experimental and real-world results proving both the scalability and efficiency of the ESDF information framework. Note that for each dataset, all processing was done using Matlab R13 running on an Intel 3.4 GHz Pentium-4 desktop with 2048 MB of RAM. For the purposes of benchmark comparison, we employed the full state-recovery technique of (13) after every camera measurement, and otherwise used the constant-time partial state-recovery method of (15) to recover the robot state.

Camera constraints were generated using a state-of-the-art feature-based image registration approach [62] founded upon the following.

- Extract a combination of both Harris [63] and SIFT [64] interest points from each image. For the Harris points, we exploit our navigation prior to apply an orientation normalization to the interest regions by warping via the infinite homography [62], and then compactly encode using Zernike moments [65].
- Establish putative correspondences between overlapping candidate image pairs based upon similarity and a pose-constrained correspondence search [17].
- Employ a statistically robust least median of squares (LMedS) [66] registration methodology with regularized sampling [67] to extract a consistent inlier correspondence set. For this task, we use a six-point Essential matrix algorithm [68] as the motion-model constraint.
- Solve for a relative-pose estimate using the inlier correspondence set and Horn's relative orientation algorithm [69] initialized with samples from our orientation prior.
- Carry out a two-view maximum-likelihood estimate (MLE) refinement based upon minimizing the reprojection error over all inliers [62]. This returns the optimal 5-DOF relative-pose constraint (i.e., azimuth, elevation, Euler roll, Euler pitch, Euler yaw) and first-order parameter covariance (using the standard assumption of 1-pixel,

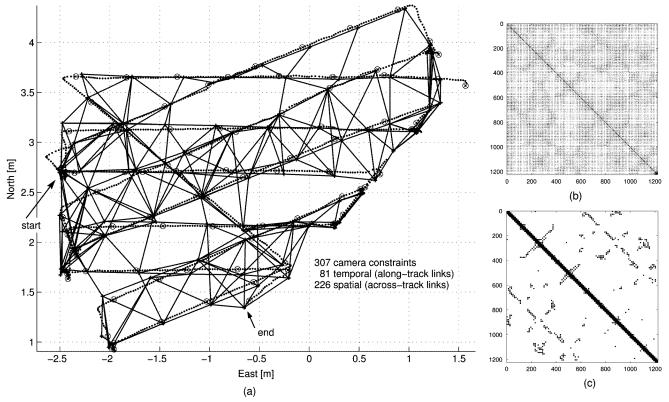


Fig. 7. Contrast of the exact sparsity of the ESDF information matrix versus the density of the full-covariance matrix. (a) Spatial topology of a 101-image sequence of underwater images collected from the JHU ROV—in all, there are 307 camera constraints. The ground-truth trajectory is shown as a dotted black line with circles depicting ESDF trajectory samples; the recovered VAN pose-constraint network is depicted in black. The 3σ VAN covariance ellipses are unviewable at this scale with a standard deviation of less than 2 cm. Note that the recovered trajectory exhibits a slight systematic discrepancy from the ground truth; we believe this is due to an unaccounted-for bias in the ground-truth experimental setup. (b) Nonzero elements of the covariance matrix; all elements above a normalized correlation score of 10% are shown. (c) Nonzero elements of the information matrix. Note that the covariance matrix has $1224^2 = 1\,498\,176$ nonzero elements, while the information matrix contains only 60 048. The covariance matrix and information matrix are numerically equivalent, however, the information matrix is exactly sparse.

isotropic, independent and identically distributed (i.i.d.) noise for extracted interest points).

For further details on VAN's systems-level image processing, see [49].

A. Laboratory Validation: EKF versus ESDF

In this section, we demonstrate the efficiency of the ESDF information framework as compared with the standard EKF-based formulation.

1) Experimental Setup: The experimental setup consisted of a downward-looking digital still camera mounted to an underwater, moving, pose-instrumented ROV at the Johns Hopkins University (JHU) Hydrodynamic Test Facility [70]. Their vehicle [71] is instrumented with a typical suite of oceanographic dead-reckoning navigation sensors capable of measuring heading, attitude, XYZ bottom-referenced Doppler velocities, and a pressure sensor for depth. The vehicle and test facility are also equipped with a high frequency acoustic long-baseline (LBL) system that provides centimeter-level bounded error XY vehicle positions used for validation purposes only. A simulated seafloor environment was created by placing textured carpet, riverbed rocks, and landscaping boulders on the tank floor and was appropriately scaled to match a rugged seafloor environment with considerable 3-D scene relief. See [49] for further experimental details.

2) Experimental Results: Fig. 7 shows the result of estimating the ROV delayed states associated with a 101-image sequence using a full covariance EKF and sparse ESDF. For this experiment, the vehicle started near the top-left corner of the plot at (-2.5, 2.75) and then drove a course consisting of two grid-based surveys, one oriented SW to NE, and the other W to E. Fig. 7(a) shows the spatial XY pose topology, 3σ confidence bounds (unviewable at this scale), and link network of camera constraints; links correspond to image pairs that were successfully registered. Fig. 7(b) and (c) compare the densities associated with the EKF covariance matrix versus the ESDF information matrix. Note that while the EKF correlation matrix is dense, the information matrix exhibits a sparse tridiagonal structure with the number of off-diagonal elements being linear in the number of camera constraints. In all, there are 307 camera measurements (81 temporal/226 spatial), and each delayed state is a 12-vector consisting of six pose and six kinematic components. Therefore, 102 delayed states (101 images plus the robot) results in a 1224×1224 information matrix containing $12^2(102 + 2 \cdot 101) + 6^2(2 \cdot 226) = 60\,048$ nonzero elements, as shown. We found the EKF and ESDF solutions to be numerically equivalent and, furthermore, that the ESDF only required 4% of the storage of the EKF for this experiment.

Turning our attention now to filter efficiency, in Fig. 8, we compare the prediction and update times of the EKF to those of the ESDF. In particular, we see that prediction is essentially a constant-time operation for both filters (in actuality, the EKF requires linear time complexity for prediction, but the slope is indiscernible at this scale). However, Fig. 8(b) shows that ESDF updates are orders of magnitude more efficient than corresponding EKF updates, and moreover, that they become *more* efficient relative to the EKF as the number of delayed states *increases*. This increase in relative efficiency with increasing state size results from a *decreasing* density in the information matrix. Also, note that this impressive computational reduction is despite the fact that we are using Matlab's "left-divide" capability to solve (13) (essentially a form of LU decomposition with forward and backward substitution). Hence, the ESDF's results could be even better if we implemented the iterative multilevel state-recovery techniques of [51] and [54]. In summary, for this 101-image sequence, data collection took a total of 17 min, EKF processing required 29 min, and ESDF estimation was just over 1 min (i.e., 17 times faster than real-time). (These numbers are for the estimation time only and exclude any image processing time.)

B. Real-World Results: Scalability

In this section, we present experimental results validating the large-area scalability of our ESDF framework.

1) Experimental Setup: The wreck of the RMS Titanic was surveyed during the summer of 2004 by the deep-sea ROV Hercules [72] operated by the Institute for Exploration of the Mystic Aquarium. The ROV was equipped with a standard suite of oceanographic dead-reckon navigation sensors comparable to the JHU vehicle suite. In addition, Hercules also had onboard a calibrated stereo rig consisting of two downward-looking 12-bit digital still cameras that collected imagery at a rate of 1 frame every 8 s. Note, however, that the results being presented here were produced using imagery from *one* camera only—the purpose of this self-imposed restriction to a monocular sequence is to demonstrate the general applicability of our VAN methodology.

2) Experimental Results: In Fig. 9, we see a time progression of the camera constraints and vehicle trajectory estimate with

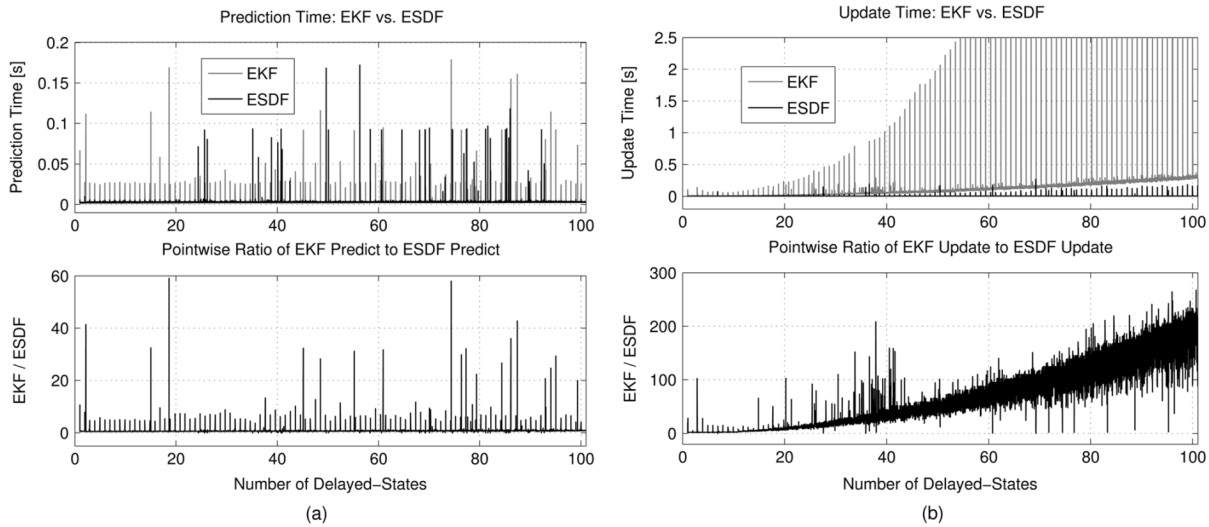


Fig. 8. Time comparison of EKF versus ESDF filtering operations using the JHU dataset. (a) The top figure shows both the EKF and ESDF prediction times in seconds versus the number of delayed-state entries, while the bottom figure shows their pointwise ratio. From the plots, we can gather that, for all practical purposes, prediction is a constant-time operation for both filters (in actuality, the EKF requires linear time complexity while the ESDF is constant-time, however, their respective slopes are indiscernible at this scale). (b) The same plot layout as before, but now we show the update times for each filter. The y-axis of the upper graph has been clipped from its nominal range of $[0, 15]$ to show detail. For benchmark comparison, we employed the full-state recovery technique of (13) after every camera measurement (using Matlab's "left divide" capability). Note that even despite this, the ESDF becomes more efficient relative to the EKF with increasing state size due to the decreasing density of the information matrix.

Fig. 10(a) showing the final 3-D pose-constraint network. In particular, Fig. 9(c) depicts a large loop-closing event, whereby the vehicle successfully relocalized by correctly registering four image pairs out of 64 hypothesized candidates. This was *after* having lost bottom-lock Doppler velocity measurements for an extended period of time. In all, the vehicle traversed a (3-D) path length of 3.4 km over the course of a 344 min survey with a (faster than real-time) total ESDF estimation time of less than 39 min (excluding image processing time). The resulting convex hull of the final mapped region encompasses an area over 3100 m^2, with a total of 866 images used to provide 3494 camera-generated relative-pose constraints.

While there is no ground truth for this dataset, the resulting pose network qualitatively appears to be consistent in that the recovered vehicle trajectory forms the outline of a ship's hull. To quantitatively corroborate the recovered pose-network accuracy, we pairwise triangulated scene structure using only the saved pairwise image correspondences and the final VAN estimated vehicle poses. The results are shown in Fig. 10. Note that the histograms of Fig. 10(d) and (e) contain two error measures and that the y-axis has been clipped to show fine detail. The first measure (white) is the triangulation error based upon the relative-pose camera measurements used by the ESDF filter. This should serve as a baseline for the best possible pairwise triangulation error, since each pose measure is the result of a two-view bundle adjustment. The second measure (black) is the triangulation error based upon the final VAN estimated poses. Scale for both measures has been set by the VAN estimate.

Note that the VAN triangulated errors are more widely distributed than the pairwise bundle-adjusted poses. This is, however, to be expected since VAN's global estimate takes into account all measured camera constraints. The "outliers" are due to poor triangulation resulting from residual error in the global VAN estimate. Again, this error is to be expected, since VAN is not directly enforcing structure consistency, only pose consistency. In fact, because VAN is enforcing only pose consistency, the overall coherence of the point clouds in Fig. 10(b) and (c) (less than 7.5 cm of triangulation error) corroborates the global consistency of VAN's pose estimates. This result is even more

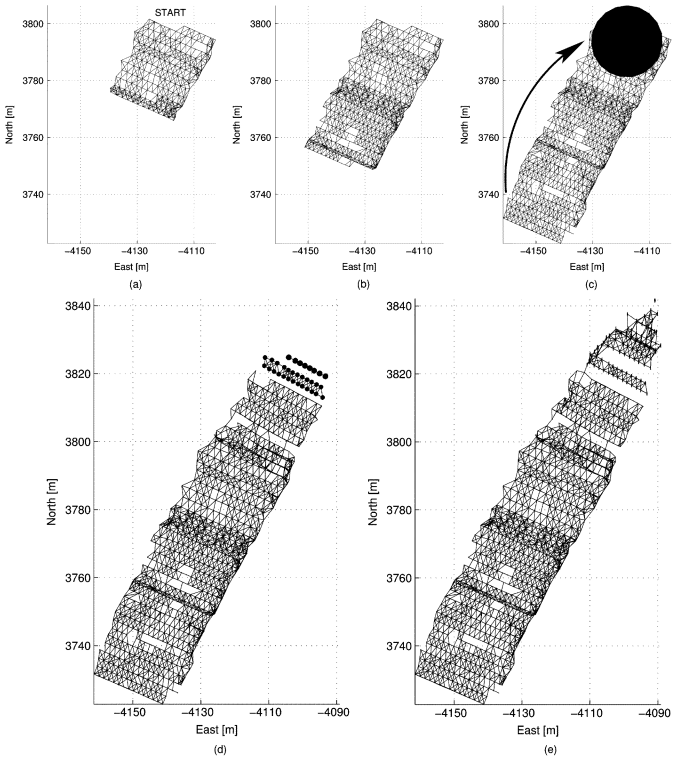


Fig. 9. Time evolution of the RMS Titanic pose constraint network. Successfully registered camera links are shown in gray, and 3σ covariance bounds are depicted in black. Time progression begins with the upper left plot and proceeds from left to right: images 1–200, 1–400, 1–600, 1–800, all. Note the large loop-closing event that occurs in (c) when the vehicle returns to the bow of the ship (depicted by the black arrow) after having traveled from the stern with the camera turned off.

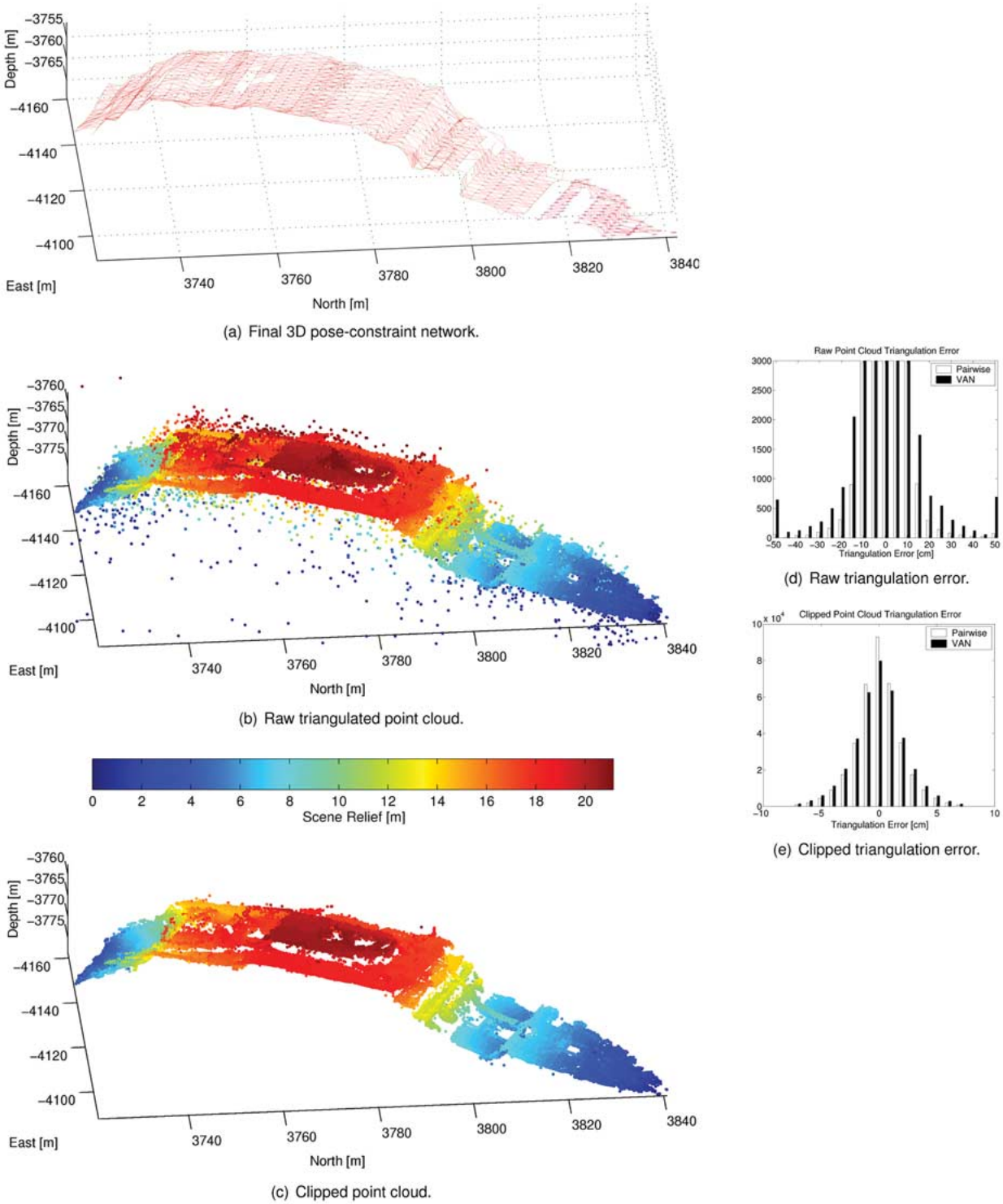


Fig. 10. Triangulated structure for the RMS Titanic as computed from the final VAN pose estimate and saved pairwise correspondences. Triangulated 3-D points are defined as the midpoint of the minimum perpendicular distance between two corresponding camera rays. Since structure is triangulated on a pairwise basis, redundant 3-D points may occur. (a) Oblique view of the final 3-D VAN pose-constraint network associated with using 866 images to provide 3494 camera constraints; 3σ bounds are unviewable at this scale. Green links represent temporally consecutive registered image pairs, while red links represent spatially registered image pairs. (b) Raw VAN triangulated points rendered in 3-D (467 512 points in total). (c) Reduced set of triangulated data (363 799 points) for which we have thrown away all points having a triangulation error greater than 7.5 cm. (d)–(e) Histograms of the triangulation error (i.e., the minimum perpendicular distance) for all points across all established camera pairs.

impressive when taking into consideration the fact that VAN does not explicitly enforce consistency of structure, only consistency of poses. This adds further evidence that VAN's global pose estimates are near-ideal. As an aside, note that the quality of VAN's results suggests that it can serve as a recursive scalable solution to large-area SFM, since the estimated pose and triangulated structure should provide a good initialization point in an optimal batch bundle-adjustment step.

VII. Conclusion

In conclusion, this paper presented the insight that the delayed-state view-based SLAM information matrix is exactly sparse and, furthermore, that this sparsity is a direct consequence of retaining historical trajectory samples. Moreover, while the EKF covariance formulation requires quadratic storage, the number of nonzero off-diagonal elements in the ESDF information matrix is *linear* in the number of measured relative-pose constraints. This sparse matrix structure allows for efficient full state recovery via recently proposed multilevel relaxation methods, while approximate partial state recovery allows motion prediction and navigation updates to be performed in constant time. Finally, we demonstrated the efficiency and large-area applicability of the ESDF framework by presenting vision-based 6-DOF SLAM results for both laboratory and real-world experiments.

Acknowledgment

The authors wish to thank their colleagues, Prof. L. Whitcomb and Dr. J. Kinsey, for their collaboration in collecting the JHU tank dataset. They are also grateful to Dr. R. Ballard for providing them with the RMS Titanic dataset. Finally, they would like to thank Prof. S. Teller for his discussions regarding large-area, scalable SLAM.

References

[1] J. Reynolds, R. Highsmith, B. Konar, C. Wheat, and D. Doudna, "Fisheries and fisheries habitat investigations using undersea technology," in *Proc. IEEE/MTS OCEANS Conf. Exhib.*, Honolulu, HI, Nov. 2001, vol. 2, pp. 812–820.

[2] H. Singh, R. Armstrong, F. Gilbes, R. Eustice, C. Roman, O. Pizarro, and J. Torres, "Imaging coral I: Imaging coral habitats with the SeaBED AUV," *J. Subsurface Sensing Tech. Appls.*, vol. 5, no. 1, pp. 25–42, Jan. 2004.

[3] B. Fletcher, "Chemical plume mapping with an autonomous underwater vehicle," in *Proc. IEEE/MTS OCEANS Conf. Exhib.*, Honolulu, HI, Nov. 2001, vol. 1, pp. 508–512.

[4] C. German, D. Connelly, R. Prien, D. Yoerger, M. Jakuba, A. Bradley, T. Shank, H. Edmonds, and C. Langmuir, "New techniques for hydrothermal exploration: In situ chemical sensors on AUVs—Preliminary results from the Lau Basin," in *EOS: Trans. Amer. Geophys. Union Fall Meet. Supplement*, Dec. 2004, p. A190+, Abstract.

[5] M. Jakuba, D. Yoerger, A. Bradley, C. German, C. Langmuir, and T. Shank, "Multiscale, multimodal AUV surveys for hydrothermal vent localization," in *Proc. Int. Symp. Unmanned Untethered Subm. Technol.*, Durham, NH, 2005, to be published.

[6] D. Yoerger, A. Bradley, B. Walden, H. Singh, and R. Bachmayer, "Surveying a subsea lava flow using the Autonomous Benthic Explorer (ABE)," *Int. J. Syst. Sci.*, vol. 29, no. 10, pp. 1031–1044, Oct. 1998.

[7] D. Yoerger, A. Bradley, M. Cormier, W. Ryan, and B. Walden, "High resolution mapping of a fast spreading mid-ocean ridge with the autonomous benthic explorer," in *Proc. Int. Symp. Unmanned Untethered Subm. Technol.*, Durham, NH, Aug. 1999, CD-ROM.

[8] D. Yoerger, D. Kelley, and J. Delaney, "Fine-scale three-dimensional mapping of a deep-sea hydrothermal cent site using the Jason ROV system," *Int. J. Robot. Res.*, vol. 19, no. 11, pp. 1000–1014, Nov. 2000.

[9] D. Yoerger, H. Singh, L. Whitcomb, J. Catteau, J. Adams, B. Foley, and D. Mindell, "High resolution mapping for deep water archeology," in *Proc. Annu. Meet. Soc. Historical Archaeol.*, 1998, CD-ROM.

[10] H. Singh, J. Adams, D. Mindell, and B. Foley, "Imaging underwater for archeology," *Amer. J. Field Archaeol.*, vol. 27, no. 3, pp. 319–328, 2000.

[11] R. Ballard, L. Stager, D. Master, D. Yoerger, D. Mindell, L. Whitcomb, H. Singh, and D. Piechota, "Iron age shipwrecks in deep water off Ashkelon, Israel," *Amer. J. Archaeol.*, vol. 106, no. 2, pp. 151–168, Apr. 2002.

[12] M. Hunt, W. Marquet, D. Moller, K. Peal, W. Smith, and R. Spindel, "An acoustic navigation system," Woods Hole Oceanogr. Inst., Tech. Rep. WHOI-74-6, Dec. 1974.

[13] P. Milne, *Underwater acoustic positioning systems*. Houston, TX: Gulf, 1983.

[14] *Acoustic Doppler Current Profiler: Principles of Operation a Practical Primer*, 2nd ed. San Diego, CA: RD Instruments, Jan. 1996.

[15] T. Gaiffe, "U-Phins: A FOG-based inertial navigation system developed specifically for AUV navigation and control," in *Proc. Int. Conf. Underwater Intervention*, New Orleans, LA, Feb. 2002, CD-ROM.

[16] L. Whitcomb, D. Yoerger, and H. Singh, "Advances in Doppler-based navigation of underwater robotic vehicles," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1999, vol. 1, pp. 399–406.

[17] R. Eustice, O. Pizarro, and H. Singh, "Visually augmented navigation in an unstructured environment using a delayed state history," in *Proc. IEEE Int. Conf. Robot. Autom.*, New Orleans, LA, Apr. 2004, vol. 1, pp. 25–32.

[18] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Auton. Robots*, vol. 4, pp. 333–349, Apr. 1997.

[19] J. Leonard and R. Rikoski, "Incorporation of delayed decision making into stochastic mapping," in *Experimental Robotics VII*, ser. Lecture Notes in Control and Information Sciences. New York: Springer-Verlag, 2001, vol. 271, pp. 533–542.

[20] S. Fleischer, "Bounded-error vision-based navigation of autonomous underwater vehicles," Ph.D. dissertation, Stanford Univ., Stanford, CA, May 2000.

[21] P. McLauchlan, "A batch/recursive algorithm for 3D scene reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Hilton Head, SC, 2000, vol. 2, pp. 738–743.

[22] R. Smith, M. Self, and P. Cheeseman, "A stochastic map for uncertain spatial relationships," in *Proc. Int. Symp. Robot. Res.*, 1988, pp. 467–474.

[23] ——, "Estimating uncertain spatial relationships in robotics," in *Autonomous Robot Vehicles*I. Cox and G. Wilfong, Eds. New York: Springer-Verlag, 1990, pp. 167–193.

[24] P. Moutarlier and R. Chatila, "An experimental system for incremental environment modeling by an autonomous mobile robot," in *Proc. Int. Symp. Exp. Robot.*, Montreal, QC, Canada, Jun. 1989, pp. 327–346.

[25] J. Leonard and H. Feder, "Decoupled stochastic mapping," *IEEE J. Ocean. Eng.*, vol. 26, no. 4, pp. 561–571, Oct. 2001.

[26] M. Bosse, P. Newman, J. Leonard, and S. Teller, "An Atlas framework for scalable mapping," *Int. J. Robot. Res.*, vol. 23, pp. 1113–1139, Dec. 2004.

[27] J. Leonard and P. Newman, "Consistent, convergent, and constant-time SLAM," in *Proc. Int. Joint Conf. Artif. Intell.*, Acapulco, Mexico, Aug. 2003, pp. 1143–1190.

[28] A. Davison, "Mobile robot navigation using active vision," Ph.D. dissertation, Univ. Oxford, Oxford, U.K., 1999.

[29] S. Williams, "Efficient solutions to autonomous mapping and navigation problems," Ph.D. dissertation, Univ. Sydney, Sydney, Australia, 2001.

[30] J. Guivant and E. Nebot, "Optimization of the simultaneous localization and map building algorithm for real time implementation," *IEEE Trans. Robot. Autom.*, vol. 17, no. 3, pp. 242–257, Jun. 2001.

[31] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proc. AAAI Nat. Conf. Artif. Intell.*, Edmonton, AB, Canada, 2002, pp. 593–598.

[32] M. Montemerlo, "FastSLAM: A factored solution to the simultaneous localization and mapping problem with unknown data association," Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, PA, Jun. 2003.

[33] S. Julier and J. Uhlmann, "Building a million beacon map," in *Proc. SPIE Sensor Fusion, Decentralized Control Robot. Syst. IV*, G. McKee and P. Schenker, Eds., Oct. 2001, vol. 4571, pp. 10–21.

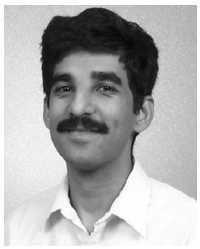
[34] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous localization and mapping with sparse extended information filters," *Int. J. Robot. Res.*, vol. 23, no. 7–8, pp. 693–716, Jul.–Aug. 2004.

[35] M. Paskin, "Thin junction tree filters for simultaneous localization and mapping," in *Proc. Int. Joint Conf. Artif. Intell.*, San Francisco, CA, 2003, pp. 1157–1164.

[36] U. Frese and G. Hirzinger, "Simultaneous localization and mapping—A discussion," in *Proc. IJCAI Workshop: Reasoning with Uncertainty in Robot.*, Seattle, WA, 2001, pp. 17–26.
[37] U. Frese, "Treemap: An O(Log N) algorithm for simultaneous localization and mapping," in *Spatial Cognition IV*, C. Freksa, Ed. New York: Springer-Verlag, 2004.
[38] R. Eustice, H. Singh, and J. Leonard, "Exactly sparse delayed-state filters," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, 2005, pp. 2428–2435.
[39] P. McLauchlan and D. Murray, "A unifying framework for structure and motion recovery from image sequences," in *Proc. IEEE Int. Conf. Comput. Vis.*, Boston, MA, 1995, pp. 314–320.
[40] U. Frese, "A proof for the approximate sparsity of SLAM information matrices," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, 2005, pp. 331–337.
[41] R. Eustice, M. Walter, and J. Leonard, "Sparse extended information filters: Insights into sparsification," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2005, pp. 641–648.
[42] M. Walter, R. Eustice, and J. Leonard, "A provably consistent method for imposing sparsity in feature-based SLAM information filters," in *Proc. Int. Symp. Robot. Res.*, San Francisco, CA, Oct. 2005, CD-ROM.
[43] F. Dellaert, "Square root SAM," in *Proc. Robot.: Sci. Syst.*, Cambridge, MA, Jun. 2005, pp. 177–184.
[44] Y. Bar-Shalom, X. Rong Li, and T. Kirubarajan, *Estimation With Applications to Tracking and Navigation*. New York: Wiley, 2001.
[45] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann, 1988.
[46] Y. Weiss and W. Freeman, "Correctness of belief propagation in Gaussian graphical models of arbitrary topology," *Neural Comput.*, vol. 13, no. 10, pp. 2173–2200, 2001.
[47] K. Murphy, "Bayesian map learning in dynamic environments," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 1999.
[48] J. Gutmann and K. Konolige, "Incremental mapping of large cyclic environments," in *Proc. IEEE Int. Symp. Comput. Intell. Robot. Autom.*, Monterey, CA, Nov. 1999, pp. 318–325.
[49] R. Eustice, "Large-area visually augmented navigation for autonomous underwater vehicles," Ph.D. dissertation, Mass. Inst. Technol./Woods Hole Oceanogr. Inst. Joint Prog., Woods Hole, MA, Jun. 2005.
[50] J. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-CS-94-125, Aug. 1994.
[51] K. Konolige, "Large-scale map-making," in *Proc. AAAI Nat. Conf. Artif. Intell.*, San Jose, CA, 2004, pp. 457–463.
[52] T. Duckett, S. Marsland, and J. Shapiro, "Learning globally consistent maps by relaxation," in *Proc. IEEE Int. Conf. Robot. Autom.*, San Francisco, CA, Apr. 2000, pp. 3841–3846.
[53] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1992.
[54] U. Frese, P. Larsson, and T. Duckett, "A multilevel relaxation algorithm for simultaneous localization and mapping," *IEEE Trans. Robot.*, vol. 21, no. 2, pp. 1–12, Apr. 2005.
[55] A. Brandt, "Multi-level adaptive solutions to boundary-value problems," *Math. Comput.*, vol. 31, pp. 333–390, 1977.
[56] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Trans. Robot. Autom.*, vol. 17, no. 3, pp. 229–241, Jun. 2001.
[57] Y. Liu and S. Thrun, "Results for outdoor-SLAM using sparse extended information filters," in *Proc. IEEE Int. Conf. Robot. Autom.*, Sep. 2003, vol. 1, pp. 1227–1233.
[58] R. Eustice, H. Singh, J. Leonard, M. Walter, and R. Ballard, "Visually navigating the RMS Titanic with SLAM information filters," in *Proc. Robot.: Sci. Syst.*, Cambridge, MA, Jun. 2005, pp. 57–64.
[59] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment—A modern synthesis," in *Vision Algorithms: Theory and Practice*, W. Triggs, A. Zisserman, and R. Szeliski, Eds. New York: Springer-Verlag, 2000, pp. 298–375.
[60] C. Stachniss, G. Grisetti, and W. Burgard, "Recovering particle diversity in a Rao–Blackwellized particle filter for SLAM after actively closing loops," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, Apr. 2005, pp. 655–660.
[61] Z. Zhang and Y. Shan, "Incremental motion estimation through local bundle adjustment," Microsoft Res., Tech. Rep. MSR-TR-01-54, May 2001.
[62] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
[63] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Vis. Conf.*, Manchester, U.K., 1988, pp. 147–151.
[64] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[65] O. Pizarro, "Large scale structure from motion for autonomous underwater vehicle surveys," Ph.D. dissertation, Mass. Inst. Technol./Woods Hole Oceanogr. Inst. Joint Prog., Woods Hole, MA, Sep. 2004.
[66] P. Rousseeuw and A. Leroy, *Robust Regression and Outlier Detection*. New York: Wiley, 1987.
[67] Z. Zhang, "Determining the epipolar geometry and its uncertainty: A review," *Int. J. Comput. Vis.*, vol. 27, no. 2, pp. 161–198, 1998.
[68] O. Pizarro, R. Eustice, and H. Singh, "Relative pose estimation for instrumented, calibrated imaging platforms," in *Proc. Digital Image Comput. Appls.*, Sydney, Australia, Dec. 2003, pp. 601–612.
[69] B. Horn, "Relative orientation," *Int. J. Comput. Vis.*, vol. 4, no. 1, pp. 59–78, Jan. 1990.
[70] J. Kinsey, D. Smallwood, and L. Whitcomb, "A new hydrodynamics test facility for UUV dynamics and control research," in *Proc. IEEE/MTS OCEANS Conf. Exhib.*, Sep. 2003, vol. 1, pp. 356–361.
[71] D. Smallwood, R. Bachmayer, and L. Whitcomb, "A new remotely operated underwater vehicle for dynamics and control research," in *Proc. Int. Symp. Unmanned Untethered Subm. Technol.*, Durham, NH, 1999, pp. 370–377.
[72] D. Coleman, R. Ballard, and T. Gregory, "Marine archaeological exploration of the Black Sea," in *Proc. IEEE/MTS OCEANS Conf. Exhib.*, Sep. 2003, vol. 3, pp. 1287–1291.

Ryan M. Eustice (S'00–M'05) received the B.S. degree in mechanical engineering from Michigan State University, East Lansing, in 1998, and the Ph.D. degree in ocean engineering from the Massachusetts Institute of Technology/Woods Hole Oceanographic Institution Joint Program, Woods Hole, MA, in 2005.

Currently, he is an Assistant Professor with the Department of Naval Architecture and Marine Engineering, University of Michigan, Ann Arbor. His research interests are in the areas of navigation and mapping, underwater computer vision and image processing, and autonomous underwater vehicles.



Hanumant Singh (S'87–M'95) received the B.S. degree as a distinguished graduate in computer science and electrical engineering from George Mason University, Fairfax, VA, in 1989, and the Ph.D. degree from the Massachusetts Institute of Technology, Cambridge, MA/Woods Hole Oceanographic Institution (WHOI), Woods Hole, MA, joint program in 1995.

He has been a member of the staff at WHOI since 1995, where his research interests include high-resolution imaging underwater and issues associated with docking, navigation, and the architecture of underwater vehicles.

John J. Leonard (S'87–M'87) received the B.S.E. degree in electrical engineering and science from the University of Pennsylvania, Philadelphia, in 1987, and the D.Phil. degree in engineering science from the University of Oxford, Oxford, U.K., in 1994.

Currently, he is an Associate Professor of Mechanical and Ocean Engineering in the Department of Mechanical Engineering, Massachusetts Institute of Technology (MIT), Cambridge. He is also a member of the MIT Computer Science and Artificial Intelligence Laboratory (CSAIL). His research addresses the problems of navigation, mapping, and persistent autonomy for autonomous mobile robots operating in unstructured environments.