Fast 3D Pose Estimation With Out-of-Sequence Measurements

Ananth Ranganathan, Michael Kaess, and Frank Dellaert
Georgia Institute of Technology
{ananth,kaess,dellaert}@cc.gatech.edu

***Abstract*— We present an algorithm for pose estimation using fixed-lag smoothing. We show that fixed-lag smoothing enables inclusion of measurements from multiple asynchronous measurement sources in an optimal manner. Since robots usually have a plurality of uncoordinated sensors, our algorithm has an advantage over filtering-based estimation algorithms, which cannot incorporate delayed measurements optimally. We provide an implementation of the general fixed-lag smoothing algorithm using square root smoothing, a technique that has recently become prominent. Square root smoothing uses fast sparse matrix factorization and enables our fixed-lag pose estimation algorithm to run at upwards of 20 Hz. Our algorithm has been extensively tested over hundreds of hours of operation on a robot operating in outdoor environments. We present results based on these tests that verify our claims using wheel encoders, visual odometry, and GPS as sensors.**

I. Introduction

Localization, also called pose estimation, is essential for a robot trying to navigate in any environment. When combined with map creation, this morphs into the Simultaneous Localization and Mapping (SLAM) problem, one of the most important problems in robotics and one that has attracted much attention in the last few years. However, in many settings, such as in outdoor environments where GPS measurements are available, pose estimation alone may suffice to enable a robot to reach its goal.

The classical solution to the pose estimation problem is the Lu-Milios algorithm [17], which performs a global optimization over relative pose constraints to obtain the optimal robot trajectory. However, the Lu-Milios algorithm is a batch algorithm that is unsuitable for real-time operation. On the other hand, filtering algorithms such as the Extended Kalman Filter (EKF) [20] estimate only the current robot pose, and hence provide estimates of poorer quality since the information from measurements can only propagate forward in time, unlike in the full Lu-Milios scheme, where new information also influences earlier estimates.

A compromise between full smoothing and the filtering is provided by the fixed-lag smoothing scheme, where the joint estimate of the poses $\{\mathbf{x}_{t-p}, \ldots, \mathbf{x}_t\}$ from time $t-p$ to the current time t is estimated at each step for a fixed interval p called the lag. At each step, the estimate of the first pose in the fixed lag, $\mathbf{x}_{t-p}$, is the best since this estimate can incorporate the largest number of future measurements. Fixed-lag smoothing is often used as an expensive filtering scheme in cases where simple filtering produces unacceptably poor estimates but full smoothing is too expensive.

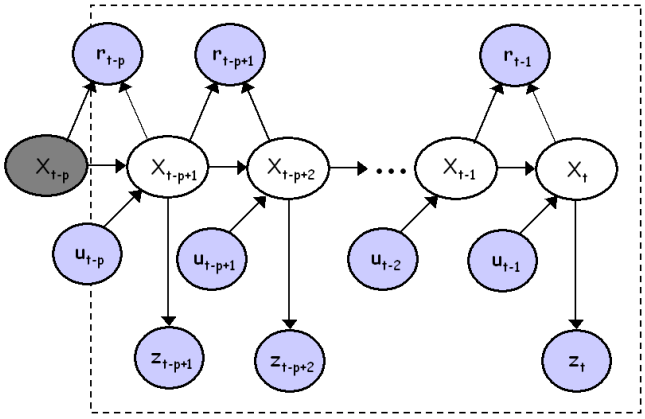


Fig. 1. Dynamic Bayes Net for fixed-lag smoothing. $\mathbf{x}_t$ is the robot state at time t, the control inputs are represented as u_i, while r_i and z_i represent the relative pose constraints (eg. odometry) and absolute pose measurements (eg. GPS) respectively. A state need not have all these types of measurements associated with it. The *active states* comprising the fixed-lag of length p are shown inside the rectangular box. The *prior state* $\mathbf{x}_{t-p}$ is shown shaded dark.

Our motivation for using fixed-lag pose estimation stems from its application to the LAGR program [16] where the robot obtains pose measurements from multiple sources such as wheel encoders, visual odometry, IMU, and GPS at varying rates. In our application, while odometry from wheel encoders is available at 20 Hz, visual odometry and GPS are only available at rates of 2-4 Hz. Hence, while incorporating a visual odometry measurement, we can take advantage of the measurements from the "future" which have already arrived from the faster sensors.

When measurements from multiple asynchronous sensor streams need to be fused, fixed-lag smoothing offers unmatched advantages over filtering, since future information from faster sensor streams can be used to improve the estimate when a time-delayed measurement from a slower stream is obtained. However, this involves incorporating out-of-sequence measurements in an efficient manner so that the algorithm maintains its real-time character. To the best of our knowledge, such an algorithm for robot localization has not been presented before.

In this paper, we consider the pose estimation problem

in the above-mentioned scenario where the robot receives measurements at varying rates from multiple sources. Further, our estimates are computed for the 3D case, where the poses are six-dimensional, consisting of three spatial coordinates and three angles, respectively. This is important since outdoor robot navigation in difficult terrain, which has recently been in the spotlight following the DARPA Grand Challenge, violates the ground plane assumption made in using 2D pose.

Our contributions through this paper are four-fold -

- We provide an optimal fixed-lag smoothing algorithm that can incorporate measurements arriving *out of* time-sequence order from different sensors
- Estimation is performed for 6D poses using Lie-Algebraic techniques on the space of all Euclidean motions, represented by the Special Euclidean group SE(3)
- Covariance estimates for the poses are also computed. This is useful, for example, in maximum likelihood data association
- The complete algorithm, including covariance estimation, runs in real time at up to 20Hz. This is possible, in part, due to our use of a square root information fixed-lag smoother, analogous to square-root information smoothing [5].

We deal with the case where the distribution on poses in the trajectory is assumed to be Gaussian. However, measurements may be non-linear as in an EKF. We consider odometry measurements from wheel encoders and visual odometry obtained from a stereo pair mounted on the robot. GPS, which unlike odometry provides absolute measurements on poses, is also incorporated into the algorithm.

The pose estimation algorithm presented here has been deployed on a robot operating in difficult and varied outdoor terrain. It has operated consistently for many hundreds of hours. We present selected results obtained during the robot's operation using our pose estimator.

In the rest of the paper, we first discuss related work and subsequently provide a formal description of fixed-lag smoothing in Section III. This is followed by an exposition in Section IV of the square root information smoother applied to the fixed-lag case. Section V provides the details for incorporating out-of-sequence measurements into the fixed-lag algorithm. Section VI discusses the particular state representation, state transition model, and measurement models used by us. We provide experimental results and analysis using these models in Section VII and some concluding remarks in Section VIII.

II. Related Work

We view pose estimation as the problem of computing the robot trajectory from relative constraints between, and also absolute measurements, on the poses. This view was pioneered by Lu and Milios [17], and further developed in various contexts [14], [6]. All Lu-Milios type approaches operate in batch mode and hence, are not real-time. Even fast linear algebra-based smoothing techniques, such as [5], ultimately become too slow due to the linear increase in the number of states over time. This is especially severe in our case since measurements arrive at rates of up to 20Hz, implying 20 additional states in the smoothing problem every second. Of the various smoothing techniques, recent work by Agrawal [1] is of particular interest as it extends the Lu-Milios technique to the 3D pose estimation using Lie Groups. This is also an approach we take here, but our algorithm also operates in real-time and can handle asynchronous measurement streams, features that Agrawal's work does not have.

Filtering approaches [10], especially those based on particle filters [9], provide real time operation at the expense of some estimation quality, and have been very popular. However, extending filtering algorithms to account for multiple measurement streams with possible delay is harder, and provides estimates that are much poorer than smoothing. The optimal solution for the delayed measurements case was provided by Bar-Shalom [2].

The first fixed-lag smoothing algorithm was developed by Rauch [19]. The use of fixed-lag smoothing for incorporating out-of-sequence measurements has been proposed before [3]. However, our use of it in conjunction with square-root smoothing is novel. While we only consider the linearized case of fixed-lag smoothing, non-parametric methods such as Sequential Monte Carlo [4] can also be applied.

III. Fixed-lag Smoothing for Pose Estimation

We now describe the use of fixed-lag smoothing for robot pose estimation. The fixed-lag problem consists of finding the posterior on the p most recent robot states given all previous measurements, where p is the length of the lag. The Dynamic Bayes Net (DBN) for fixed-lag smoothing is shown in Figure 1. We term the states in the lag $\mathbf{x}_{t-p+1:t}$ as the *active states* and the most recent state $\mathbf{x}_t$ as the *head of the lag*. The model includes three types of input that we consider - the control inputs u_i, relative pose constraints r_i such as those from wheel encoders and visual odometry, and absolute state measurements z_i such as those provided by a GPS sensor. In the discussion below, we denote all these inputs at time i collectively as Z_i to simplify the notation.

The desired posterior can be obtained by marginalizing over the state which occurs just before the start of the lag. Denoting the current time by t, this is the state at time $t-p$, $\mathbf{x}_{t-p}$. This state, denoted the *prior state*, captures all the information contained in the previous measurements up to time $t-p$. The fixed-lag posterior can hence be written as

$$p(\mathbf{x}_{t-p+1:t}|Z_{1:t}) = \int_{\mathbf{x}_{t-p}} p(\mathbf{x}_{t-p+1:t}|\mathbf{x}_{t-p}, Z_{t-p+1:t}) \, p(\mathbf{x}_{t-p}|Z_{1:t-p}) \quad (1)$$

where $Z_{1:t}$ is the set of measurements $\{Z_1, \ldots, Z_t\}$. Note that (1) is simply the Bayes filter equation for the fixed-lag case. The prior distribution $p(\mathbf{x}_{t-p}|Z_{1:t-p})$ is the filtering estimate of $\mathbf{x}_{t-p}$ when it was the head of the lag. Applying Bayes law to the first term in the integrand on the right side, we

obtain

$$p(\mathbf{x}_{t-p+1:t}|\mathbf{x}_{t-p}, Z_{t-p+1:t}) \propto p(Z_{t-p+1:t}|\mathbf{x}_{t-p}, \mathbf{x}_{t-p+1:t}) \, p(\mathbf{x}_{t-p+1:t}|\mathbf{x}_{t-p}) \quad (2)$$

The two terms on the right side of (2), which denote respectively the joint measurement likelihood and the prior on states, can in turn be factorized using the Markov assumption

$$p(Z_{t-p+1:t}|\mathbf{x}_{t-p}, \mathbf{x}_{t-p+1:t}) = \prod_{i=t-p+1}^{t} p(Z_i|\mathbf{x}_{i-1}, \mathbf{x}_i) \quad (3)$$

$$p(\mathbf{x}_{t-p+1:t}|\mathbf{x}_{t-p}) = \prod_{i=t-p}^{t-1} p(\mathbf{x}_{i+1}|\mathbf{x}_i) \quad (4)$$

Fixed-lag state estimation consists of estimating the posterior in (1) recursively using the sensor models $p(Z_i|\mathbf{x}_{i-1}, \mathbf{x}_i)$ and the motion model $p(\mathbf{x}_{i+1}|\mathbf{x}_i)$. A graphical illustration of the algorithm is given in Figure 2

IV. Square Root Information Smoothing

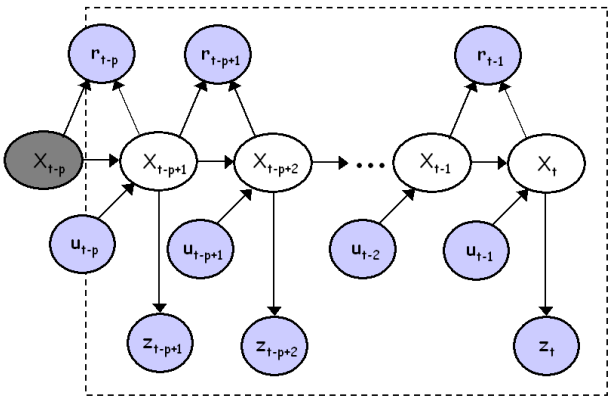
While the above discussion provided an abstract view of fixed-lag smoothing, we now present details for real-time computation of the fixed-lag posterior using square root information smoothing.

We assume Gaussian motion and measurement models, as is standard in the SLAM literature. The motion model is given as $\mathbf{x}_{i+1} = f_i(\mathbf{x}_i, u_i) + w_i$, where w_i is zero-mean Gaussian noise with covariance matrix Λ_i. Similarly, the measurement equation for a relative constraint between poses is given as $r_k = h_k(\mathbf{x}_{i_k}, \mathbf{x}_{j_k}) + v_k$, and that for an absolute measurement on the state, such as GPS, is given as $z_k = g_k(\mathbf{x}_{i_k}) + s_k$ where v_k and s_k are normally distributed zero-mean measurement noise with covariances Γ_k and Σ_k respectively. Denoting the active states $\mathbf{x}_{t-p+1:t}$ by X, the problem becomes one of non-linear least squares minimization to compute X^*

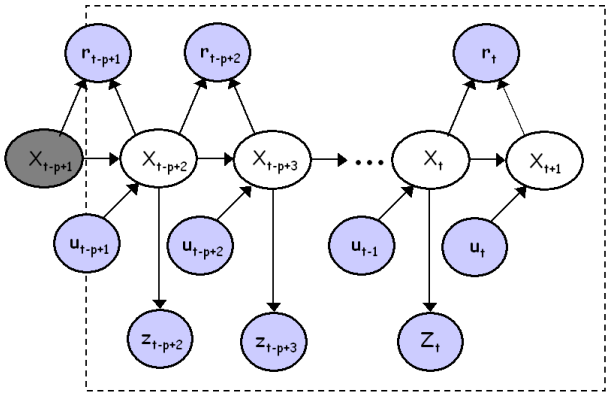
$$X^* = \underset{X}{\text{argmin}} \left\{ \sum_{i=t-p}^{t-1} ||f_i(\mathbf{x}_i, u_i) - \mathbf{x}_{i+1}||_{\Lambda_i} + \sum_k ||h_k(\mathbf{x}_{i_k}, \mathbf{x}_{j_k}) - r_k||_{\Gamma_k} + \sum_k ||g_k(\mathbf{x}_{i_k}) - z_k||_{\Sigma_k} + ||\mathbf{x}_{t-p} - \mu_{t-p}||_{P_{t-p}} \right\} \quad (5)$$

where (μ_{t-p}, P_{t-p}) is the Gaussian prior on the prior state $\mathbf{x}_{t-p}$, and we use the notation $||e||_\Sigma = e^T \Sigma^{-1} e$ for the squared Mahalanobis distance given a covariance matrix Σ. Note that all the terms in the above equation correspond to the integrand in the Bayes filter (1). Further, the prior on $\mathbf{x}_{t-p}$ is the filtering distribution on that state from p time steps ago, when $\mathbf{x}_{t-p}$ was the head of the lag, and so need not be calculated here.

In practice, only linearized versions of the motion and measurement models are considered, with multiple iterations involving relinearization and optimization being used to obtain convergence to the minimum. In the following, we will assume that either a good linearization point is available or that we are working on one iteration of a non-linear optimization method. Each iteration produces an estimate of the incremental change to the current values of the state needed to reach the minimum, i.e. if the current state is X,



(a)



(b)

Fig. 2. Illustration of fixed-lag smoothing. (a) The initial smoothing problem with fixed-lag length p, prior state $\mathbf{x}_{t-p}$, and head of the lag $\mathbf{x}_t$ (b) When a new measurement arrives, a new state $\mathbf{x}_{t+1}$ is added and $\mathbf{x}_{t-p+1}$ becomes the new prior state. Since the states move to the left in the DBN after the addition of each measurement, the prior distribution at time t is simply the filtering distribution on the head of the lag at time $t-p$. For example, the filtering distribution on $\mathbf{x}_{t+1}$ in the figure will be the prior distribution for the smoothing problem at time $t+p+1$.

an iteration produces an increment $\delta X^\star$ such that $X + \delta X^\star$ is closer to the minimum state than X. It can be shown [5] that the linearized version results in a sparse linear least squares problem in $\delta X^\star$

$$\delta X^\star = \underset{\delta X}{\text{argmin}} \left\{ \sum_{i=t-p}^{t-1} ||F_i \delta\mathbf{x}_i + I_d \delta\mathbf{x}_{i+1} - a_i||_{\Lambda_i} + \sum_{k=1} ||H_1^{i_k} \delta\mathbf{x}_{i_k} + H_2^{j_k} \delta\mathbf{x}_{j_k} - b_k||_{\Gamma_k} + ||G_k^i - c_k||_{\Sigma_k} + ||\mathbf{x}_{t-p} - \mu_{t-p}||_{P_{t-p}} \right\} \quad (6)$$

where F_i is the Jacobian of $f_i(.)$ at the linearization point $\mathbf{x}_i^0$ and $a_i \triangleq \mathbf{x}_{i+1}^0 - f_i(\mathbf{x}_i^0, u_i)$ is the odometry prediction error. The other terms are defined analogously, for example G_k^i is the Jacobian of $g_k(.)$ with respect to a change in $\mathbf{x}_{i_k}$, evaluated at the linearization point $\mathbf{x}_{i_k}^0$, and $c_k \triangleq z_k - g_k(\mathbf{x}_{i_k}^0)$ is the measurement prediction error. We use the $d \times d$ identity matrix I_d , d being the dimension of $\mathbf{x}_i$, to avoid treating $\delta\mathbf{x}_{i+1}$ in a special way.

The solution to the linear least squares problem (6) is given

by the linear system $J\delta X - b = 0$, where J is the system Jacobian matrix obtained by assembling the individual measurement Jacobians, and b is the vector of all the measurement errors.

A solution to the linear system can be obtained using the QR factorization [12] of the measurement Jacobian $J \in \mathbb{R}^{m \times n}$:

$$J = Q \begin{bmatrix} R \\ 0 \end{bmatrix} \tag{7}$$

where $Q \in \mathbb{R}^{m \times m}$ is orthogonal, and $R \in \mathbb{R}^{n \times n}$ is upper triangular. Note that the information matrix $\mathcal{I} \triangleq J^T J$, which is the inverse of the covariance, can also be expressed in terms of this factor R as $\mathcal{I} = R^T R$, which we therefore call the *square root information matrix* R. We rewrite the least-squares problem, noting that multiplying with the orthogonal matrix Q^T does not change the norm:

$$\begin{aligned} ||Q \begin{bmatrix} R \\ 0 \end{bmatrix} \delta X - b||^2 &= || \begin{bmatrix} R \\ 0 \end{bmatrix} \delta X - \begin{bmatrix} d \\ e \end{bmatrix} ||^2 \\ &= ||R\delta X - d||^2 + ||e||^2 \end{aligned} \tag{8}$$

where we defined $[d, e]^T \triangleq Q^T b$. The first term $||R\delta X - d||^2$ vanishes for the least-squares solution δX^*, leaving the second term $||e||^2$ as the residual of the least-squares problem.

The square root factor allows us to efficiently recover the mean of the active states in the fixed-lag posterior (1) at any given time. This is simply achieved by back-substitution using the current factor R and right hand side d to obtain an update for all states X based on

$$R\delta X^\star = d \tag{9}$$

Since R is a sparse matrix due to the fact that the active states are not fully connected in the DBN (Figure 1), the mean state estimates can be obtained in $O(p)$ time complexity, where p is the length of the lag as in Section III.

While the state update using δX^* involves simple addition in most cases, this in inappropriate in our case due to the use of 6D pose, which involves a 3D orientation. The 6D pose lies in the Lie group SE(3), which in turn can be viewed as a semi-direct product of the 3D Euclidean Lie group E(3) and the 3D orientation Lie group SO(3). We effect the update in this product space through use of the Lie group exponential maps of E(3) and SO(3) respectively. The exponential map in SO(3), for instance, is the well-known Rodriguez's formula [8]. As our focus here is fixed-lag smoothing, we do not discuss the update equations further, but instead direct the reader to [21] and [1] for further details.

Given the above discussion, we can now summarize the square root information fixed-lag smoothing algorithm. At each step, we linearize and obtain the linear equation involving the Jacobian $J\delta X - b = 0$. The equation is solved using QR factorization, and a back-substitution step using the R matrix, to obtain the minimum state $X = \mathbf{x}_{t-p+1:t}$. When a new measurement is encountered, we add a new state corresponding to this measurement as $\mathbf{x}_{t+1}$ and the oldest active state $\mathbf{x}_{t-p+1}$ becomes the new prior state. The whole process is then repeated. Fixed-lag smoothing using the square root information matrix is illustrated in Figure 3.

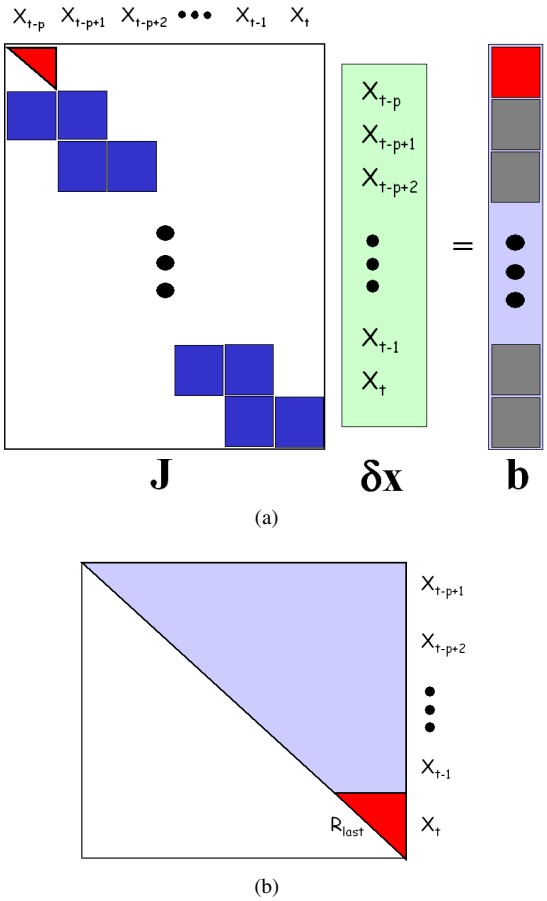


Fig. 3. Illustration of the fixed-lag algorithm using square root information smoothing. (a) The fixed-lag posterior is linearized to obtain a linear system $J\delta X = b$ involving the Jacobian matrix. The system shown here corresponds to Figure 2(a). The prior distribution on $\mathbf{x}_{t-p}$ is imposed using the error computed from (9). (b) The linear system is solved using QR factorization followed by back-substitution. The R matrix for the above system is shown here. The marginal covariance of the head of the lag $\mathbf{x}_t$ is given by $\left(R_{last}^T R_{last}\right)^{-1}$. This corresponds to the filtering distribution on $\mathbf{x}_t$ and is saved at each step, since it is also the prior on $\mathbf{x}_t$ when it becomes the prior state at time $t + p$.

A. Computing Marginal Covariances

The square root information smoother enables an easy estimate of the marginal covariances of the robot poses. Covariances are useful, for example, for providing a maximum likelihood solution to the data association problem in SLAM [15], and also for providing location uncertainty to probabilistic path planning algorithms [13].

A conservative estimate of the marginal covariance of a state can be obtained from the filtering distribution when the state is the head of the lag. At this point, the portion of the R matrix that corresponds to the head state is the last triangular piece, denoted R_{last} as shown in Figure 3(b). Due to the conditional independence constraints encoded by the R matrix, the marginal covariance of the newest state is simply $\left(R_{last}^T R_{last}\right)^{-1}$. This covariance estimate is conservative since it can only shrink as the state moves up through the

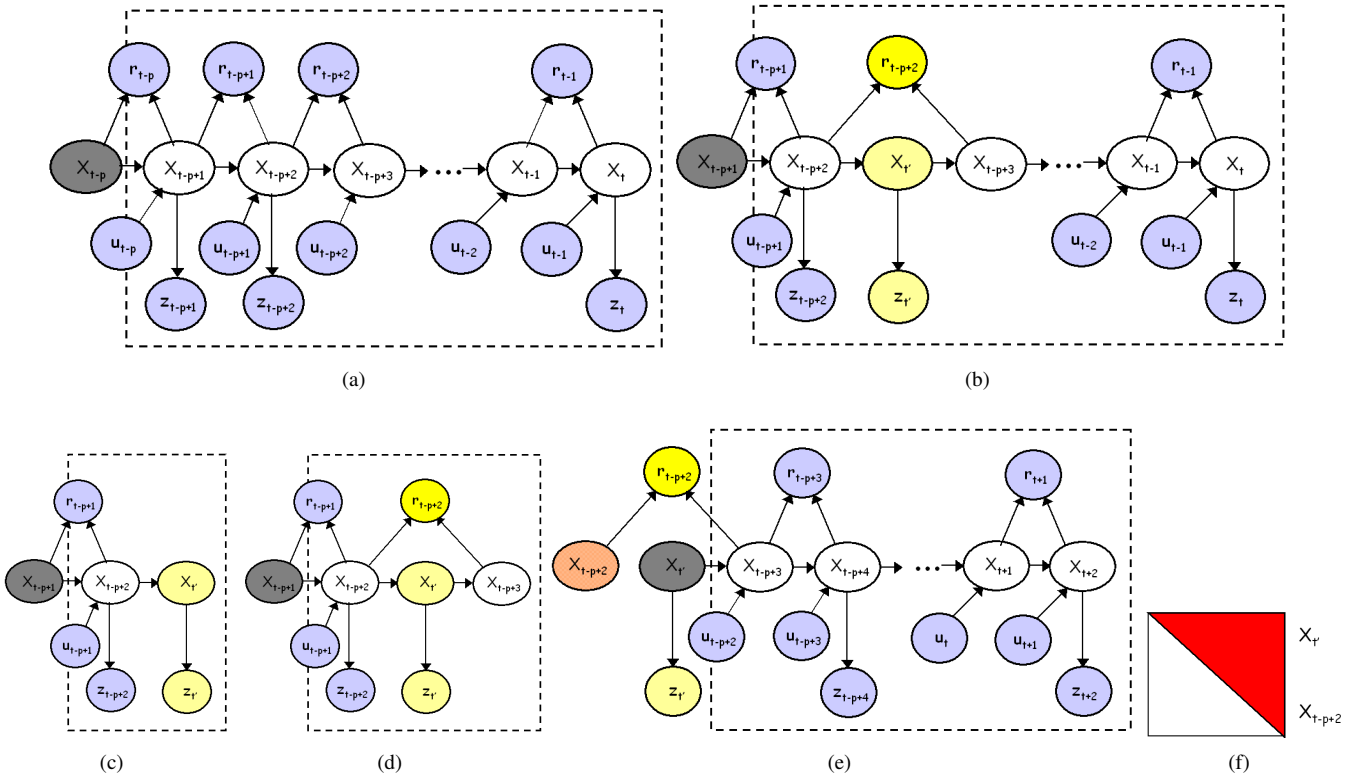


Fig. 4. Illustration of out-of-sequence measurement handling. (a) The initial smoothing problem with prior state $\mathbf{x}_{t-p}$, and head state $\mathbf{x}_t$. (b) A delayed, asynchronous absolute pose measurement arrives with time stamp t' and has to be inserted along with its state $\mathbf{x}_{t'}$ (shown in light yellow) into the DBN. The stored filtering distributions of all the poses following $\mathbf{x}_{t'}$ are now invalid and have to be recalculated. (c)-(d)This is done incrementally by solving a number of smoothing problems starting with states up to $\mathbf{x}_{t'}$ and adding a state until we get to the complete fixed-lag problem in (b). Each of these smoothing steps has a state with an invalidated filtering distribution at the head of the lag. Note that the relative measurement r_{t-p+2} in (b), shown in yellow, spans across $\mathbf{x}_{t'}$. (e) When $\mathbf{x}_{t'}$ becomes the prior state as shown here, the state preceding it, $\mathbf{x}_{t-p+2}$ (in red), which should have been discarded, is involved in a relative constraint with an active state and so has to be included in the calculations. (f) This is done by using a joint prior distribution on $\mathbf{x}_{t-p+2}$ and $\mathbf{x}_{t'}$ (R matrix shown) computed from (c). In general, there may be more than one state like $\mathbf{x}_{t-p+2}$.

set of active states and future measurements provide more information on it.

While it is possible to obtain marginal covariances for all the states in the fixed-lag at each time step, the operation is no longer $O(1)$ as above, but takes $O(p^2)$, p being the length of the fixed-lag.

V. Incorporating Out-of-Sequence Measurements

When multiple asynchronous measurement streams with different time delays are present, this introduces significant complications to the vanilla fixed-lag smoother presented in the previous section. Firstly, a time delayed measurement causes the stored filtering distributions of the active states that follow the measurement to be invalidated, as shown in Figure 4(b). Since these distributions are used as priors in the Bayes filter (1) in future steps, this problem is henceforth called the *prior invalidation* problem. Secondly, introduction of asynchronous absolute pose measurements can give rise to constraints between active states and those that have already been discarded, as shown in Figure 4(e). We will call this the *fixed-lag overflow* problem.

The fixed-lag overflow problem can be overcome by modifying the Bayes filter (1) so that we marginalize over not just the prior state but all inactive states that have constraints linking them to active states. We denote this set of states by $\mathbf{x}^o_{t-p}$. In Figure 4(e), $\mathbf{x}^o_{t-p}$ consists of just the single state $\mathbf{x}_{t-p+2}$ (shown in red). However, in general, because a delayed measurement may involve states separated by more than one state, and because multiple delayed measurements may be present, $\mathbf{x}^o_{t-p}$ will be larger in size. Thus, the Bayes filter (1) can now be modified to marginalize over this set of states in addition to the prior state

$$p(\mathbf{x}_{t-p+1:t}|Z_{1:t}) = \int_{\mathbf{x}^o_{t-p},\mathbf{x}_{t-p}} p(\mathbf{x}^o_{t-p},\mathbf{x}_{t-p}|Z_{1:t-p}) \; p(\mathbf{x}_{t-p+1:t}|\mathbf{x}^o_{t-p},\mathbf{x}_{t-p},Z_{t-p+1:t}) \quad (10)$$

and an analysis similar to Section III follows.

For the square-root information case, computing the marginal distributions of all the states in $\mathbf{x}^o_{t-p}$ for use in (10) at each time step is expensive. Instead, we note that all the states in $\mathbf{x}^o_{t-p}$ precede $\mathbf{x}_{t-p}$, and denote the oldest state by $\mathbf{x}_{o_p}$. The marginalization can now be performed over the contiguous states $\mathbf{x}_{o_p:t-p}$ which can again be obtained by storing the joint distribution on these states when $\mathbf{x}_{t-p}$ was the head of the lag.

Referring to Figure 4(e), marginalization is performed over the two contiguous states $\mathbf{x}_{t-p+2}$ and $\mathbf{x}_{t'}$, the joint distribution on which is computed from the smoothing solution in the

previous step shown in Figure 4(c). The R matrix corresponding to the joint covariance on these states in shown in Figure 4(f). In the case where there are additional states between $\mathbf{x}_{t-p+2}$ and $\mathbf{x}_{t'}$ that are *not involved* in any delayed measurements, these are also included in the marginalization. This is because computing the marginal distribution on $\mathbf{x}_{t-p+2}$ is more expensive compared to integrating over the joint Gaussian distribution on the states $\mathbf{x}_{t-p+2:t'}$. Additionally, as the intervening states are not involved in any measurements, their distributions are not affected by the computation in (10). Since the number of states in this joint prior cannot be more than p, the length of the fixed-lag, this implies that any measurement delayed excessively cannot be processed. However, in practice, this is not an issue since p can be set to account for measurement time delays of the system.

A time delayed measurement invalidates the stored joint distributions corresponding to states following the measurement. These joint distributions are nedded for use as future prior distributions as described above. This problem is illustrated in Figure 4(b). The invalid distributions have to be recalculated by successively performing smoothing steps starting from the state corresponding to the delayed measurement and adding one state at a time. This is shown in Figures 4(c)-(d).

While this procedure for computing the joint prior distributions may seem expensive, it is in practice quite fast since all the states have good initial estimates from previous iterations (except for $\mathbf{x}_{t'}$) and 1-2 steps of the iterative optimization usually suffice to compute each new prior. Furthermore, time delayed measurements are usually infrequent.

VI. State Representation and Models

We now provide details of the robot state representation, motion model and measurement models used to obtain the results presented in the following section. Note that the fixed-lag algorithm is in no way restricted to this particular state representation but is applicable to any valid representation with the associated motion and measurement models.

We take the robot state to be the 8-tuple

$$\mathbf{x} \quad = \quad (x, y, z, \theta, \phi, \psi, v, w) \tag{11}$$

where (x, y, z) is the location of the robot, (θ, ϕ, ψ) is the (yaw, pitch, roll) orientation of the robot, and v and w are the linear and angular yaw velocities respectively. The state transition dynamics are modeled by the linear system $\mathbf{x}_{t+1} = f(\mathbf{x}_t) + Q$, where f is the non-linear prediction function and Q is the Gaussian prediction noise. The prediction function models the pitch ϕ, roll ψ and angular velocity ω as constants C_{pitch}, C_{roll} and C_ω respectively with some added Gaussian noise. Hence, the state transition equation can be written out as

$$\mathbf{x}_{t+1} = \begin{bmatrix} yaw_{t+1} \\ pitch_{t+1} \\ roll_{t+1} \\ x_{t+1} \\ y_{t+1} \\ z_{t+1} \\ v_{t+1} \\ \omega_{t+1} \end{bmatrix} = \begin{bmatrix} yaw_t + \omega_t * dt \\ C_{pitch} \\ C_{roll} \\ x_t + v_t * (\cos yaw_t) * dt \\ y_t + v_t * (\sin yaw_t) * dt \\ z_t \\ v_t \\ C_\omega \end{bmatrix} + \begin{bmatrix} 0 \\ \sigma_p \\ \sigma_r \\ 0 \\ 0 \\ \sigma_z \\ \sigma_v \\ \sigma_\omega \end{bmatrix} \tag{12}$$

so that the motion model $p\left(\mathbf{x}_{t+1}|\mathbf{x}_t\right)$ is given by the Gaussian $\mathcal{N}(Fx_t, F\Sigma F^T + Q)$, where F is the Jacobian matrix of $f(\mathbf{x})$ evaluated at $\mathbf{x}_t$, and Σ is the covariance matrix on x_t. Note that this is the standard constant velocity model used widely in the literature, for instance in car tracking [7].

We consider measurements in the form of odometry, visual odometry, and GPS. The measurement models for these are also assumed to be Gaussian distributed

$$\begin{aligned} p(z_i|x_i) \quad &\propto \quad \mathcal{N}(\mathbf{x}_i, \Sigma_{gps}) \\ p(r_k|\mathbf{x}_{i_k}, \mathbf{x}_{j_k}) \quad &\propto \quad \mathcal{N}\left(h\left(\mathbf{x}_{i_k}, \mathbf{x}_{j_k}\right), \Sigma_k\right) \end{aligned} \tag{13}$$

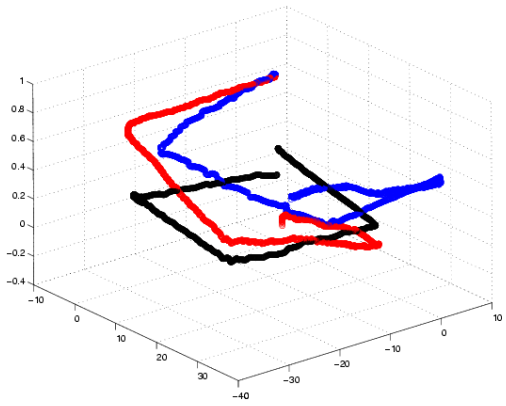
where $h(.)$ is the non-linear map that computes the 6D rigid transformation between two robot states.

Fig. 5. The LAGR robot with two stereo camera pairs and a GPS antenna on top used in the experiments.

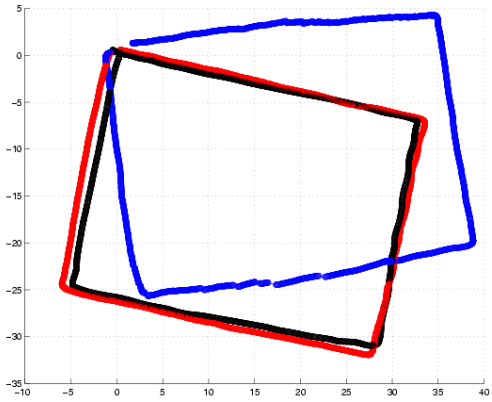
VII. Results

We have implemented the fixed-lag smoothing pose estimator using the motion and measurement models described in the previous section, on a robot with two stereo camera pairs (Figure 5). Odometry information from wheel encoders is available from the robot at the rate of 20Hz while GPS measurements arrive at 3-4Hz with a latency of about 0.3 seconds. Visual odometry, computed from the stereo pairs using the algorithm described in [18], can be computed at a rate of 4Hz but is only available with a latency of 0.5 secs due to communication and computation delays. Hence, the measurement streams are both asynchronous and contain time delays, both of which are difficult cases for normal pose estimators, but are handled by our algorithm.

Under the LAGR program [16], our pose estimator has clocked hundreds of robot hours in varied outdoor terrain,



(a)



(b)

Fig. 6. Two views of the rectangular trajectories for the first experiment - the proposed fixed lag algorithm (red), EKF filtering (blue), and Lu-Milios smoothing (black). The start location is at the top right in (b) and the robot traversed the rectangle in an anti-clockwise direction. The total number of states in the run was 2788. Note that the z-axis has a much larger scale.

including under forest cover where GPS measurements are unreliable, and in hilly terrain where full 6D pose is essential. We present results from some of these runs below. All results were obtained using a lag of length $p = 25$ which corresponds to approximately 1.25 secs on the robot.

We first present an experiment that emphasizes the need for processing delayed measurements optimally. The robot was made to traverse a rectangular path and ended its run at its initial location. The trajectories obtained using simple a EKF filtering algorithm, our fixed-lag smoothing, and full Lu-Milios smoothing are shown in Figure 6. The robot's wheels slip at the start of its trajectory resulting in an orientation error. The fixed-lag and smoothing algorithm cope with this by incorporating GPS measurements, which are delayed by about 0.3 seconds each. The EKF algorithm, however, cannot accomodate these measurements optimally. In this case, we modified the filter to consider the GPS measurement as corresponding to the current state. This results in EKF state estimate lagging behind the GPS measurements, and forever trying to "catch up". The resulting trajectory is skewed as

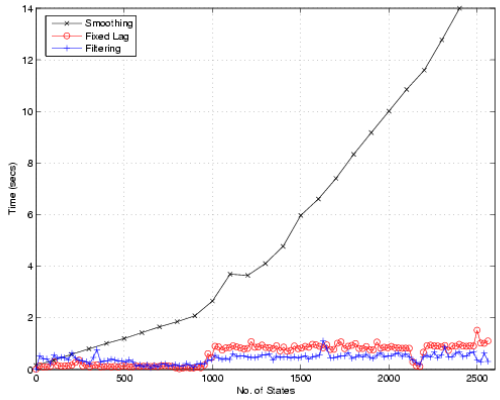


Fig. 7. Timing information for the run in Figure 6 displayed in seconds for a given length of the full trajectory. Since fixed-lag smoothing (red) and filtering (blue) are independent of the length of the trajectory, their computation time per time step remains almost constant, while full smoothing (black) requires $O(n)$ time, where n is the number of states. The times for fixed-lag smoothing and filtering have been exaggerated by a factor of 20 for viewing convenience.

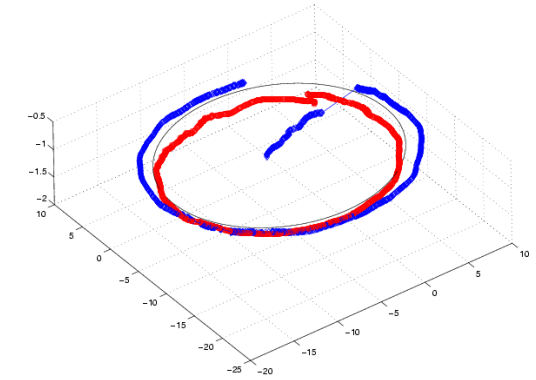
can be seen in Figure 6. Fixed-lag smoothing with a lag of 25 gives an almost identical result to Lu-Milios smoothing.

Figure 7 gives timing information for run in the above experiment, where the runtime for filtering and fixed-lag smoothing have been multiplied by 20 for viewing convenience on the linear scale. As expected, the time required for full smoothing increases linearly with number of states while the time required for fixed-lag smoothing remains almost constant. This time is almost the same as that required for filtering up to a constant factor. Both filtering and fixed-lag smoothing are almost two orders of magnitude faster than full Lu-Milios style smoothing. This provides evidence for the real-time applicability of fixed-lag smoothing, especially when used in conjunction with square-root smoothing.

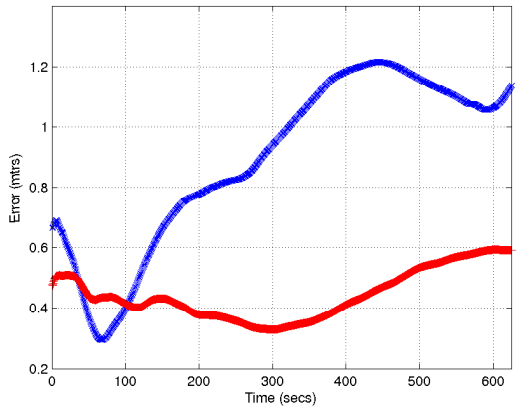
The second experiment validates the robustness of the algorithm to missing measurements. The robot in this case was made to traverse a circular trajectory of known radius (Figure 8(a)). We analysed the localization error of our algorithm under a scenario where the GPS signal is lost midway through the run but is regained after an interval. When GPS is regained, this causes a discontinuity in the EKF estimate due to the large measurement error. Fixed-lag smoothing spreads this error over all the active states, and hence displays more robustness. Figure 8(b) gives the localization error and demonstrates that the error from fixed-lag smoothing is less than half of the filtering error at the end of the run.

VIII. Conclusion

We presented a fixed-lag smoothing algorithm for pose estimation that can deal with multiple asynchronous measurement sources in the presence of time delays. We provided an implementation for the general fixed-lag smoother in the case of linearized motion and measurement models. This implementation uses sparse QR factorization to produce a fast square-root smoothing algorithm. Our algorithm has been extensively tested on a robot in a wide variety of terrain and



(a)



(b)

Fig. 8. (a) The circular path traversed by the robot is shown with the ground-truth in black, the filtering estimate in blue and fixed-lag estimate in red. GPS is lost midway through the run and causes a jump in the filtering estimate when it is regained. (b) Average localization error for the run with the error from the filtering estimate in blue and fixed-lag estimate in red.

we presented results that confirm the robustness and real-time applicability of the method.

Integration of GPS measurements has been done using a Gaussian measurement model. However, GPS measurements are highly non-linearly distributed in practice due to multipath effects from surroundings, particularly under forest cover or in urban canyons. In practice, to enable the use of a Gaussian model, we gate GPS measurements and discard those that fall outside the gating radius. This approach is adopted in the interest of speed and also accuracy, since linearizing a function encoding multi-path effects is impractical. The use of sequential Monte Carlo methods instead of square-root smoothing would rectify this problem [11].

The extension of our technique to SLAM is relatively straight-forward. The fixed-lag posterior for SLAM consists not only of p robot states but also includes the current estimates of locations of landmarks in the environment, which make up the map. Loop closings can be handled in a similar manner to EKF SLAM but with more accurate estimates. The application of fixed-lag smoothing to SLAM and an analysis of its performance is part of future work.

ACKNOWLEDGEMENTS

This work was funded in part by DARPA under the IPTO/LAGR program (contract #FA8650-04-C-7131). The authors are grateful to Matthew Powers, David Wooden, and Dongshin Kim for conducting many of the experiments.

REFERENCES

[1] M. Agrawal. A Lie algebraic approach for consistent pose registration for motion estimation. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2006.

[2] Y. Bar-Shalom. Update with out-of-sequence measurements in tracking: Exact solution. *Signal and Data Processing of Small Targets 2000, Proceedings of SPIE*, 4080:541–556, 2000.

[3] S. Challa, R.J. Evans, X. Wang, and J. Legg. A fixed-lag smoothing solution to out-of-sequence information fusion problems. *Communications In Information and Systems*, 2(4):325–348, 2002.

[4] T.C. Clapp and S.J. Godsill. Fixed-lag smoothing using sequential importance sampling. In J.M. Bernardo, J.O. Berger, A.P. Dawid, and A.F.M. Smith, editors, *Bayesian Statistics VI*, pages 743–752, 1999.

[5] F. Dellaert. Square Root SAM: Simultaneous location and mapping via square root information smoothing. In *Robotics: Science and Systems (RSS)*, 2005.

[6] F. Dellaert and M. Kaess. Square Root SAM: Simultaneous localization and mapping via square root information smoothing. *Intl. J. of Robotics Research*, 25(12), Dec 2006.

[7] F. Dellaert and C.E. Thorpe. Robust car tracking using Kalman filtering and Bayesian templates. In *Intl. Soc. Opt. Eng. (SPIE)*, volume 3207, October 1997.

[8] O.D. Faugeras and Q.T. Luong. *The geometry of multiple images*. The MIT press, Cambridge, MA, 2001. with contributions from T. Papadopoulo.

[9] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte Carlo Localization – Efficient position estimation for mobile robots. In *Proc.* 16^{th} *AAAI National Conference on AI*, 1999.

[10] D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello. Bayesian filtering for location estimation. *IEEE Pervasive Computing*, pages 10–19, July 2003.

[11] A. Giremus, A. Doucet, A.-C. Escher, and J.-Y. Tourneret. Nonlinear Filtering Approaches for INS/GPS Integration. In *European Signal and Image Processing Conference*, pages 873–876, 2004.

[12] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, third edition, 1996.

[13] J. P. Gonzalez and A. Stentz. Planning with uncertainty in position: An optimal and efficient planner. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 2435–2442, 2005.

[14] J.-S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *IEEE Intl. Symp. on Computational Intelligence in Robotics and Automation (CIRA)*, pages 318–325, November 2000.

[15] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Fast incremental smoothing and mapping with efficient data association. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Rome, Italy, April 2007.

[16] Learning Applied to Ground Robots (LAGR). http://www.darpa.mil/ipto/programs/lagr/vision.htm.

[17] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, pages 333–349, April 1997.

[18] K. Ni and F. Dellaert. Stereo tracking and three-point/one-point algorithms - a robust approach in visual odometry. In *Intl. Conf. on Image Processing (ICIP)*, 2006.

[19] H. Rauch. Solutions to the linear smoothing problem. *IEEE Transactions on Automatic Control*, 8(4):371–372, 1963.

[20] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in Robotics. In I. Cox and G. Wilfong, editors, *Autonomous Robot Vehicles*, pages 167–193. Springer-Verlag, 1990.

[21] C.J. Taylor and D.J. Kriegman. Minimization on the Lie group SO(3) and related manifolds. Technical Report 9405, Yale University, New Haven, CT, April 1994.