Incremental Sparse Gaussian Process Regression for Continuous-time Trajectory Estimation & Mapping

Georgia Tech College of Computing

Xinyan Yan[1], Vadim Indelman[2], Byron Boots[1]

[1] College of Computing
Georgia Institute of Technology

[2] Faculty of Aerospace Engineering
Technion - Israel Institute of Technology

DEPARTMENT OF AEROSPACE ENGINEERING

1 Motivation

The problem of simultaneously recovering the location of a robot and a map of its environment from sensor readings is a fundamental challenge in robotics. We want a *continuous-time representation* of the robot trajectory, which elegantly handles asynchronous and sparse measurements, and an efficient *incremental approach* to updating the estimate of the trajectory and map.

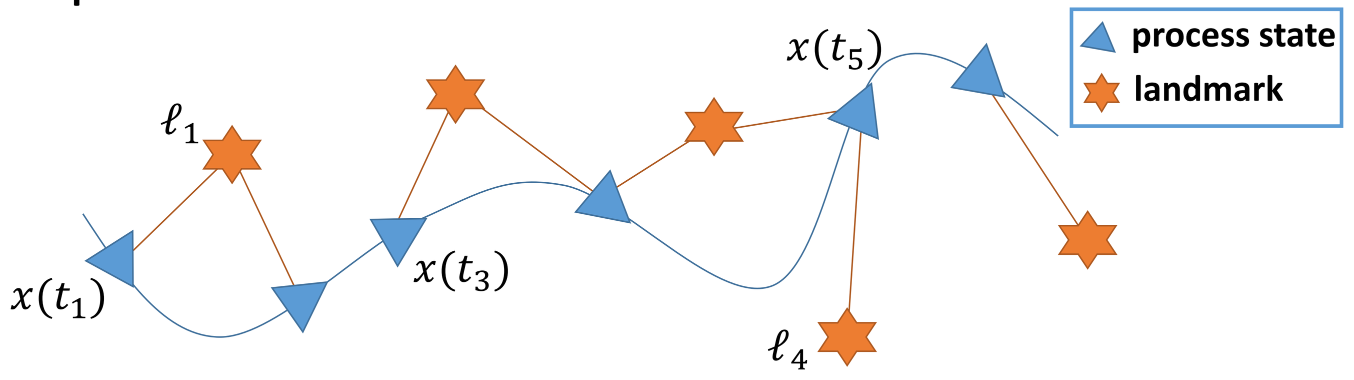


Figure 1. An example for trajectory and map.

2 Batch Trajectory Estimation Mapping as GP Regression

We take a Gaussian process regression approach to state estimation as in Tong et al.[1], where we represent robot trajectories x as functions of time t.

process model: $\quad x(t) \sim \mathcal{GP}\big(\mu(t), \mathcal{K}(t,t')\big), \quad t_0 < t, t'$

measurements: $\quad y_i = h_i(\theta_i) + n_i, \;\; n_i \sim \mathcal{N}(0, R_i), \;\; i = 1, 2, \dots, N$

Combined state of process states at measurement times and landmark locations

$$\ell = [\,\ell_1\ \ell_1 \dots \ell_M], \quad \ell \sim \mathcal{N}(d, W)$$

$$\theta \sim \mathcal{N}(\eta, \mathcal{P}), \quad \eta = [\,\mu\ d\,]^T, \quad \mathcal{P} = \begin{bmatrix} \mathcal{K} & \\ & W \end{bmatrix}$$

maximum a posteriori (MAP) estimate of the combined state:

$$\theta^* = \operatorname*{argmax}_{\theta} p(\theta|y) = \operatorname*{argmax}_{\theta} p(\theta, y) = \operatorname*{argmax}_{\theta} p(\theta) p(y|\theta)$$
$$= \operatorname{argmin}\big(\|\theta - \eta\|^2_{\mathcal{P}^{-1}} + \|y - h(\theta)\|^2_{R^{-1}}\big)$$

Gauss-Newton method. Linearize measurements at the current estimate:

$$h_i(\bar{\theta} + \delta\theta_i) \approx h_i(\bar{\theta}_i) + \frac{\partial h_i}{\partial \theta_i}\Big|_{\bar{\theta}_i}, \quad H_i = \frac{\partial h_i}{\partial \theta_i}\Big|_{\bar{\theta}_i}$$

$$\delta\theta^* = \operatorname*{argmin}_{\delta\theta}\left(\|\bar{\theta} + \delta\theta - \eta\|^2_{\mathcal{P}^{-1}} + \|y - h(\bar{\theta}) - H\delta\theta\|^2_{R^{-1}}\right)$$

$$\underbrace{(\mathcal{P}^{-1} + H^T R^{-1} H)}_{\mathcal{I}}\ \delta\theta^* = \underbrace{H^T R^{-1}(y - \bar{h}) - \mathcal{P}^{-1}(\bar{\theta} - \eta)}_{b}$$

Solve the linear equations by *Cholesky decomposition* and *back substitution*:

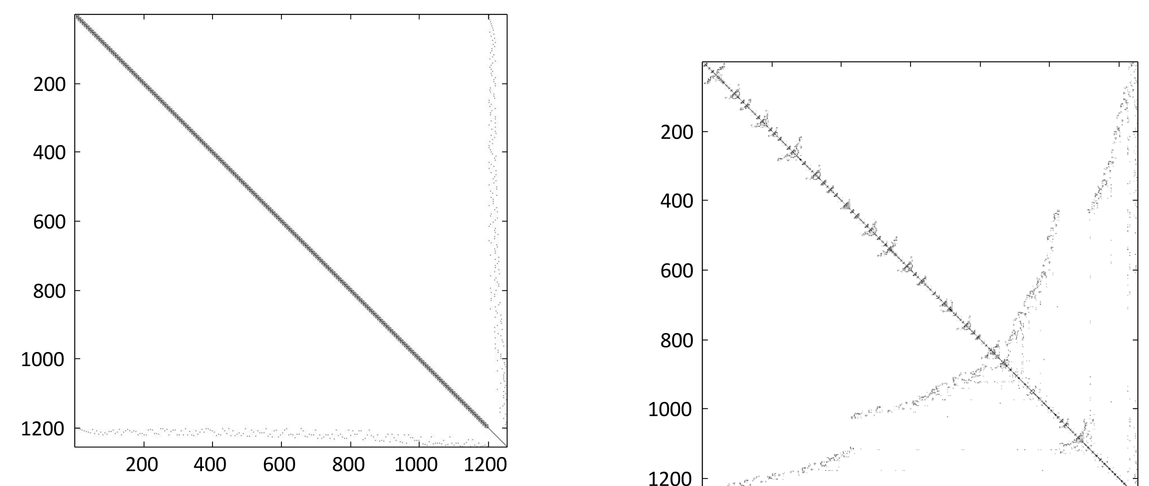
$$\mathcal{I} = \mathcal{L}^T \mathcal{L}, \quad \mathcal{L} d = b, \quad \mathcal{L}^T \delta\theta^* = d$$

Barfoot et al.[2] proved that $\mathcal{K}^{-1}$ is exactly *block-tridiagonal* when the GP is generated by linear, time-varying (LTV) stochastic differential equation (SDE):

$$\dot{x}(t) = A(t)x(t) + v(t) + F(t)w(t), \quad w(t) \sim \mathcal{GP}\big(0, Q_c \delta(t - t')\big)$$

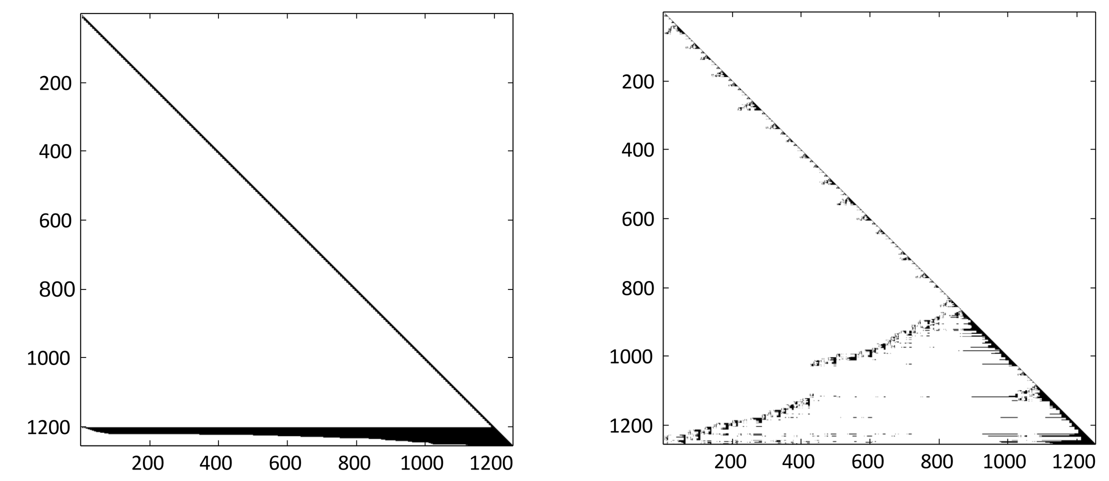
3 Variable Reordering

Even though $\mathcal{P}^{-1}$ is sparse, the Cholesky factor $\mathcal{L}$ may have a lot of *fill-in*. Good heuristic reordering methods like *SYMAMD* are able to reduce *fill-in* dramatically, which leads to significant improvements in terms of both time and space complexity.



(a) XL ordering (b) SYMAMD ordering

Figure 2. Information matrix $\mathcal{I}$ with different orderings.



(a) XL ordering (b) SYMAMD ordering

Figure 3. Cholesky factor $\mathcal{L}$ with different orderings.

4 Bayes Tree for Fast Incremental Updates

In order to incrementally update the Gaussian process combined state efficiently, we utilize a *Bayes tree* as in Kaess et al[3]. The *Bayes tree* leverages a factor graph interpretation of the problem to directly update the Cholesky factor $\mathcal{L}$ with just-in-time relinearization while maintaining sparsity.

The joint probability of variables is factored as

$$f(\theta) = \prod_i f_i(\theta_i)$$

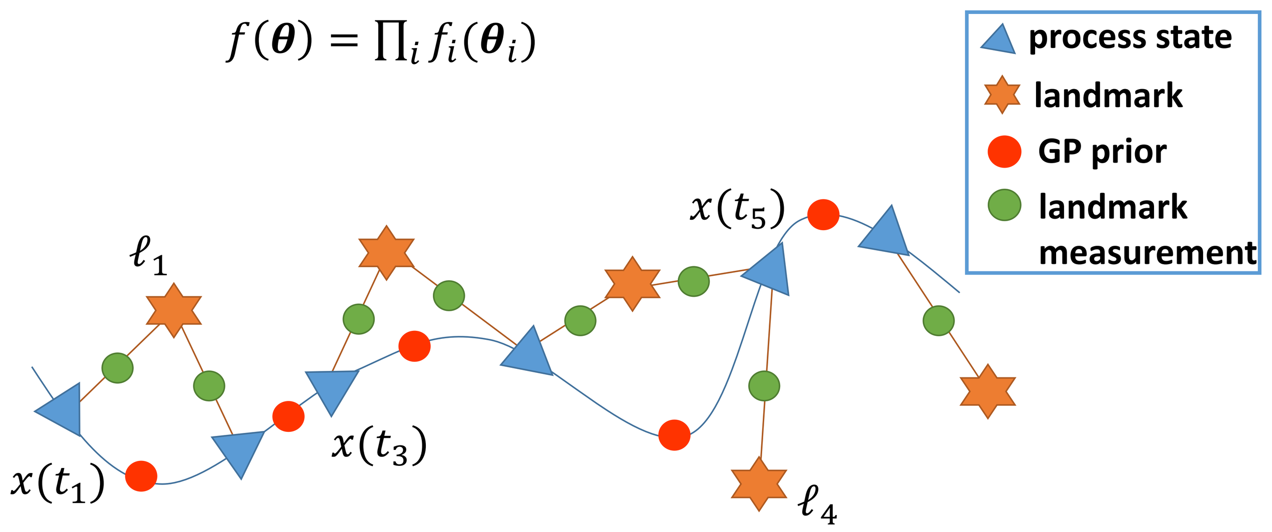


Figure 4. Trajectory and map with factors.

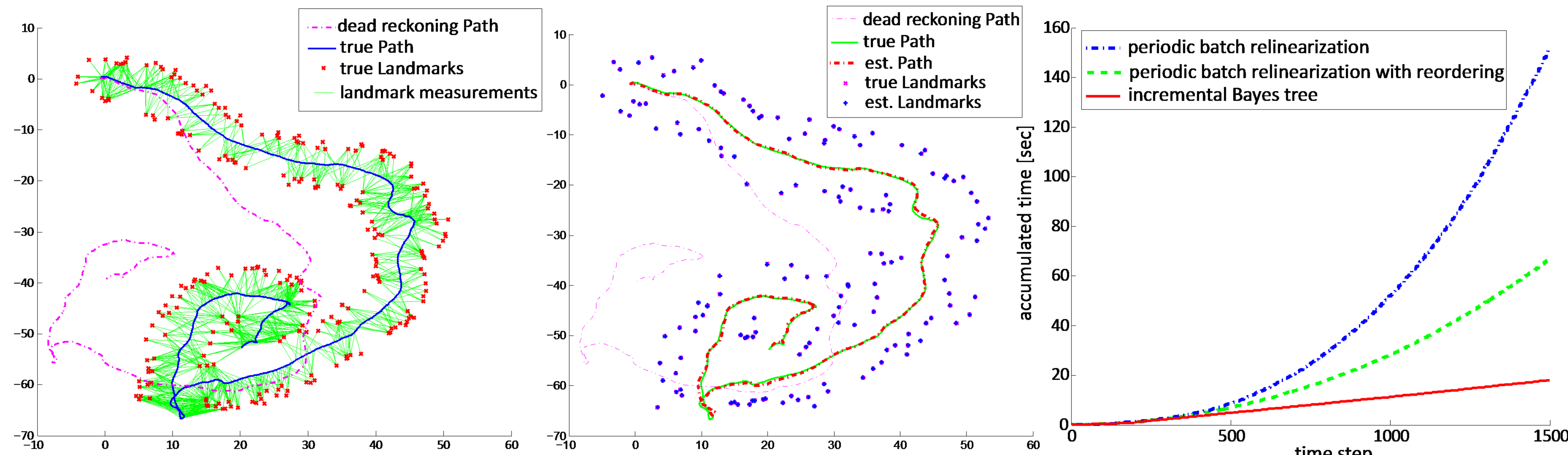
The Gaussian process priors result from the underlying process model generated from Gaussian process. When the GP is assumed to be generated by LTV SDE, the process states are first-order Markovian, even though we are using a continuous-time prior.

5 Experimental Results

We performed two experiments to illustrate some of the properties of the algorithm. Two types of measurements are included; (1) odometry measurement: robot-oriented distance and heading, and (2) range measurement: distance between the robot and a landmark.

5.1 Synthetic SLAM Exploration Task

The data consists of an exploration task with 1,500 poses, 151 landmarks, 1,499 odometry measurements and 1,500 range measurements.



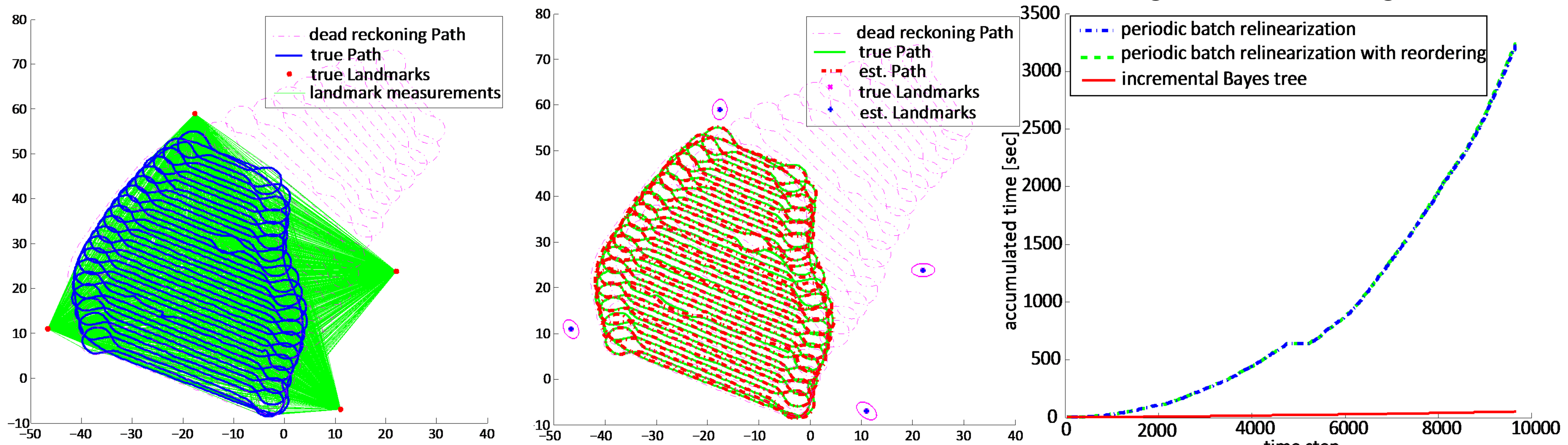
(a) True trajectory and landmarks (b) Estimated trajectory and landmarks (c) Comparison of accumulated time

Figure 5. Solving the *simultaneous trajectory estimation and mapping* problem on a synthetic dataset.

5.2 Autonomous Lawnmower

This freely available range-only SLAM dataset is collected from an autonomous lawn-mowing robot. The robot travelled 1.9km, occupied 9,658 poses, and received 3,529 range measurements.

Figure 6. lawn-mowing robot



(a) True trajectory and landmarks (b) Estimated trajectory and landmarks (c) Comparison of accumulated time

Figure 7 Solving the *simultaneous trajectory estimation and mapping* problem on a real-world dataset.

6 Conclusion

We have introduced an incremental sparse Gaussian process regression algorithm that elegantly combines the benefits of continuous-time Gaussian process-based approaches while simultaneously leveraging state-of-the-art innovations from incremental discrete-time algorithms for smoothing and mapping.

References

[1] Chi Hay Tong, Paul Furgale, and Timothy D Barfoot. Gaussian process gauss–newton for non-parametric simultaneous localization and mapping. The International Journal of Robotics Research, 32(5):507–525, 2013.

[2] Tim Barfoot, Chi Hay Tong, and Simo Sarkka. Batch continuous-time trajectory estimation as exactly sparse gaussian process regression. In Proceedings of Robotics: Science and Systems, Berkeley, USA, July 2014.

[3] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J.J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. Intl. J. of Robotics Research, IJRR, 31(2):217–236, Feb 2012.