Incremental Sparse GP Regression for Continuous-time Trajectory Estimation & Mapping

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

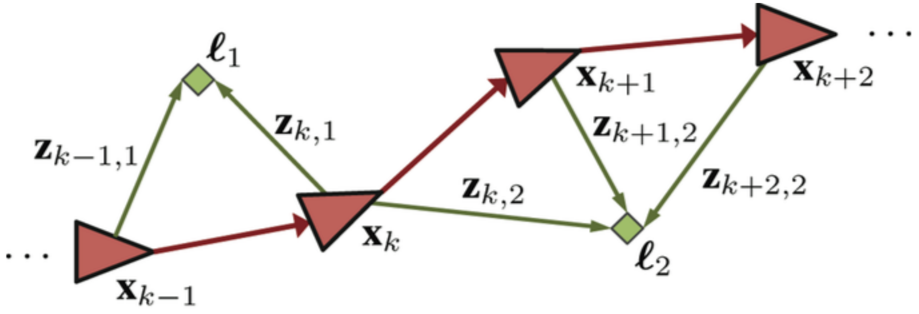
Georgia Tech College of Computing

TECHNION Israel Institute of Technology

International Symposium on Robotics Research (ISRR), September 2015

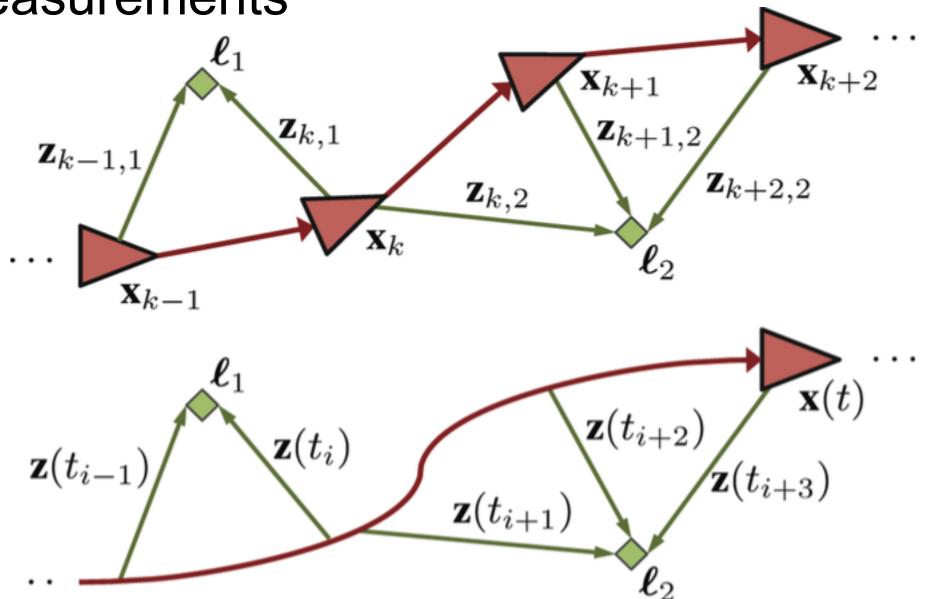
Introduction

- SLAM – fundamental problem in robotics
- Challenges include long term autonomy - how to operate online as more data is accumulated?
- Progress in recent years:
  - **Sparsity-aware smoothing approaches** (e.g. g2o, iSAM)
  - **Incremental smoothing** (iSAM2.0):
    - Identify and update only relevant **part** of the state
    - Fast, incremental
    - **But** – **discrete time** formulation



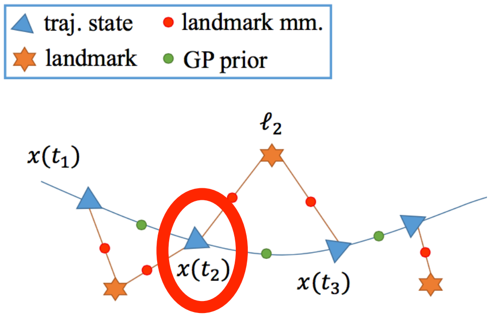
Continuous-Time SLAM via GP Regression

- Gaussian Processes have been recently incorporated within SLAM [Tong et al., IJRR '13]
  - Continuous time representation
  - Provide the ability to interpolate states while still using all measurements
  - Can be realized efficiently by exploiting sparsity of the inverse kernels
  - Naturally handles asynchronous measurements
- Key drawback: **batch** optimization



Contribution

- **We combine the benefits of Incremental smoothing (iSAM2.0) with the benefits of Continuous-time GP-SLAM**
- This leads to:
  - State interpolation yields a major reduction in running time
  - Minor impact on accuracy



$$f(\boldsymbol{\theta}) = \prod_i f_i(\boldsymbol{\theta}_i)$$

$$f_j(\boldsymbol{\theta}_j) \propto \exp\{-\frac{1}{2}\|\mathbf{h}_k(\boldsymbol{\theta}_k + \delta\boldsymbol{\theta}_k) - \mathbf{y}_k\|^2_{\mathbf{R}_k}\}$$

$$f_j(\boldsymbol{\theta}_j) \propto \exp\{-\frac{1}{2}\|\mathbf{h}_k\left(\bar{\mathbf{x}}(\tau)\right) + \mathbf{H}_k\mathcal{K}(\tau)\mathcal{K}^{-1}\delta\mathbf{x} - \mathbf{y}_k\|^2_{\mathbf{R}_k}\}$$

TECHNION Israel Institute of Technology

Georgia Tech College of Computing

Contribution

- We combine the benefits of Incremental smoothing (iSAM2.0) with the benefits of Continuous-time GP-SLAM

Algorithm 2 Incremental Sparse GP Regression via the Bayes tree with Gaussian Process Priors (BTGP)

Assign the sets of *affected* variables, variables involved in *new factors*, and *relinearized* variables to empty sets, $\boldsymbol{\theta}_{aff} := \boldsymbol{\theta}_{nf} := \boldsymbol{\theta}_{rl} := \varnothing$.

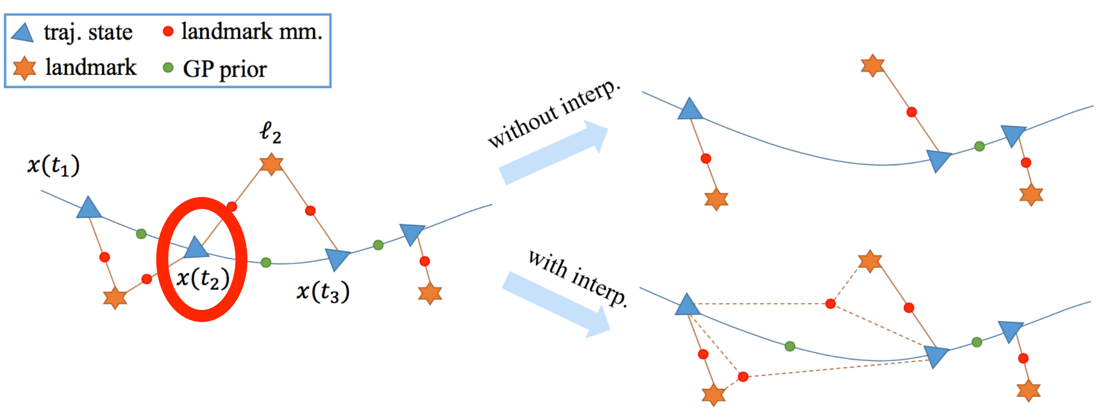
while collecting data **do**

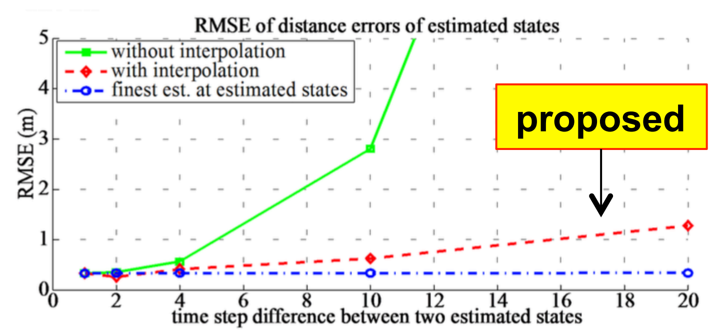
1. Collect measurements, store as new factors. Set $\boldsymbol{\theta}_{nf}$ to the set of variables involved in the *new factors*. If $\mathbf{x}(\tau) \in \boldsymbol{\theta}_{nf}$ is a missing state, replace it by newby states (Eq. 19); If $\mathbf{x}(\tau) \in \boldsymbol{\theta}_{nf}$ is a new state to estimate, a GP prior (Eq. 23) is stored, and $\boldsymbol{\theta}_{nf} := \boldsymbol{\theta}_{nf} \cup \mathbf{x}_{i-1}$.
2. For all $\theta_i \in \boldsymbol{\theta}_{aff} = \boldsymbol{\theta}_{rl} \cup \boldsymbol{\theta}_{nf}$, remove the corresponding cliques and ascendants up to the root of the Bayes tree.
3. Relinearize the factors, using interpolation when missing states are involved (Eq. 30).
4. Add the cached marginal factors from the orphaned sub-trees of the removed cliques.
5. Eliminate the graph by a new ordering into a Bayes tree, attach back orphaned sub-trees.
6. Partially update estimate from the root, stop when updates are below a threshold.
7. Collect variables, for which the difference between the current estimate and the previous linearization point is above a threshold, into $\boldsymbol{\theta}_{rl}$.

end while

Contribution

- **We combine the benefits of Incremental smoothing (iSAM2.0) with the benefits of Continuous-time GP-SLAM**
- This leads to:
  - State interpolation yields a major reduction in running time
  - Minor impact on accuracy





Contribution

- **We combine the benefits of Incremental smoothing (iSAM2.0) with the benefits of Continuous-time GP-SLAM**
- This leads to:
  - State interpolation yields a major reduction in running time
  - Minor impact on accuracy