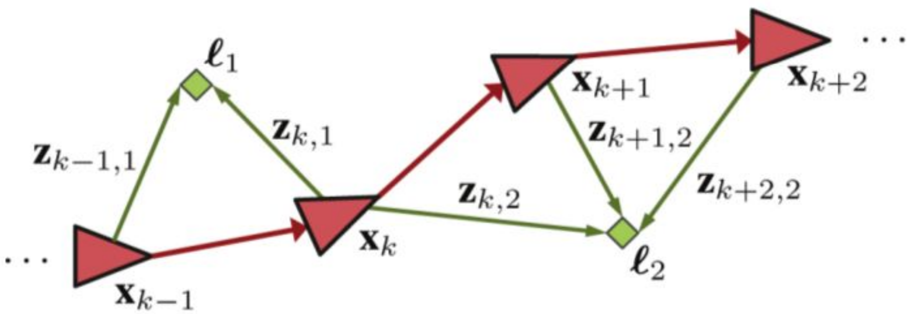
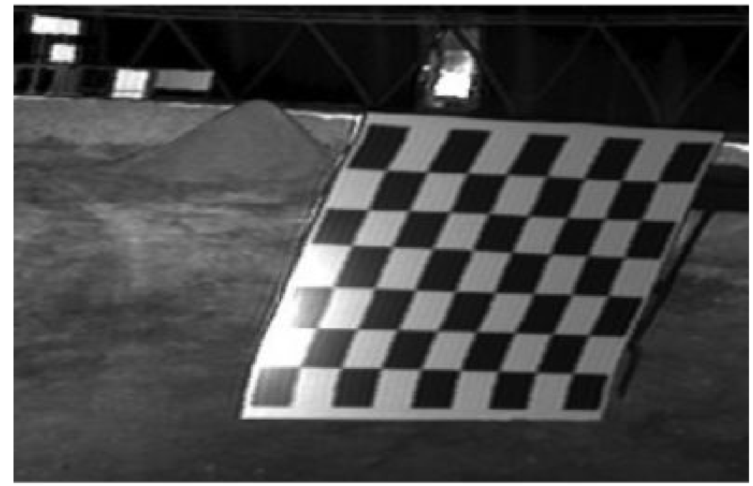
Georgia Tech Institute for Robotics and Intelligent Machines

Gaussian Processes as Continuous-time Trajectory Representations: Applications in SLAM and Motion Planning

Jing Dong
jdong@gatech.edu
2017-06-20

Discrete time SLAM

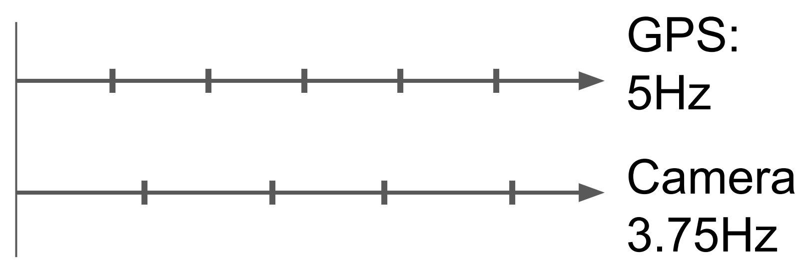




Rolling shutter effect

Downsides:

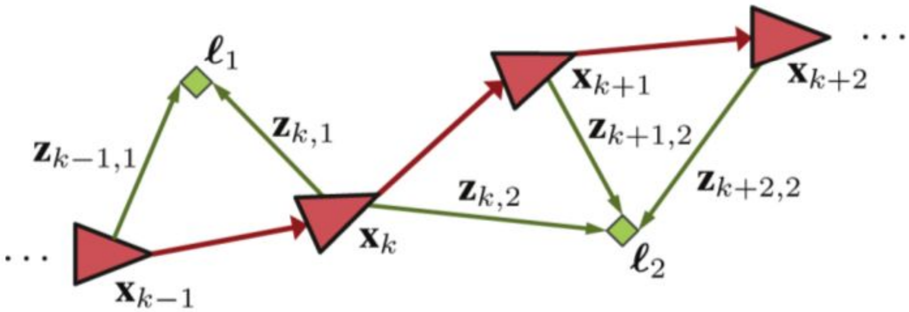
- Measurements distorted by motions
- Asynchronous measurements
- Not a compact representation

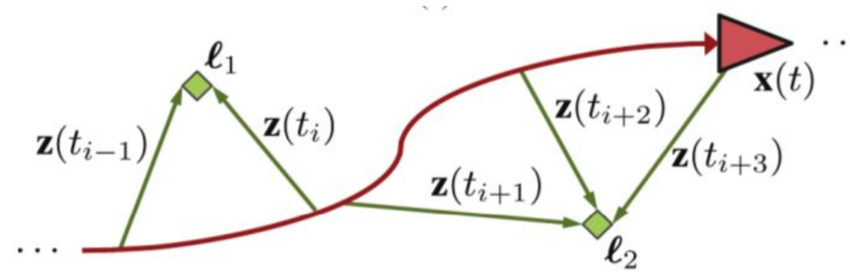


Asynchronous data

C. H. Tong, P. Furgale, and T. D. Barfoot, “Gaussian process gauss–newton for non-parametric simultaneous localization and mapping,” Intl. J. of Robotics Research, vol. 32, no. 5, pp. 507–525, 2013.

Discrete time SLAM





Downsides:

- Measurements distorted by motions
- Asynchronous measurements
- Not a compact representation

Continuous-time representation

- Linear interpolation
- Splines
- Wavelets
- Gaussian process

C. H. Tong, P. Furgale, and T. D. Barfoot, “Gaussian process gauss–newton for non-parametric simultaneous localization and mapping,” Intl. J. of Robotics Research, vol. 32, no. 5, pp. 507–525, 2013.

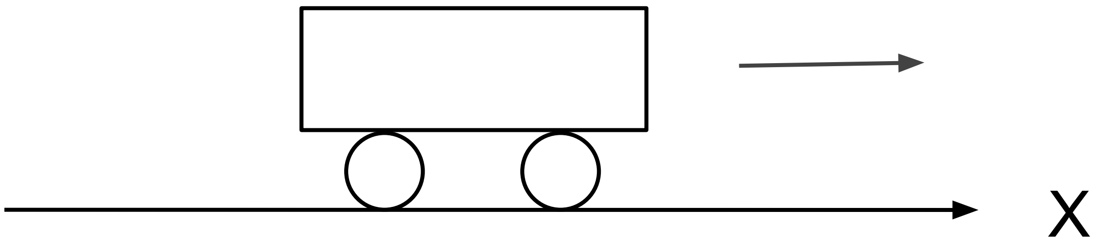
Outlines

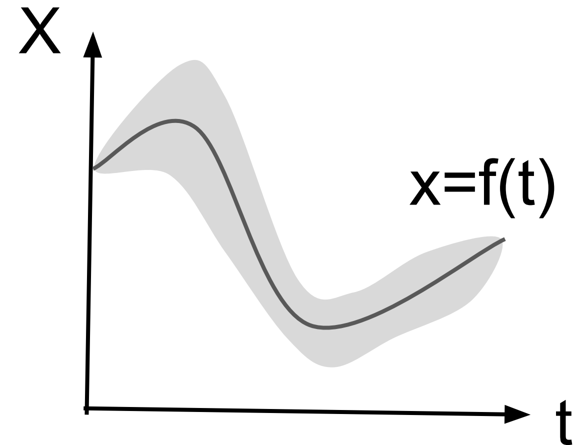
- GP as continuous-time trajectory representation
- Extend sparse GP to Lie groups
- Use sparse GP in SLAM
- Use sparse GP in motion planning

Outlines

- GP as continuous-time trajectory representation
- Extend sparse GP to Lie groups
- Use sparse GP in SLAM
- Use sparse GP in motion planning

GP as 1D continuous-time trajectory with uncertainty





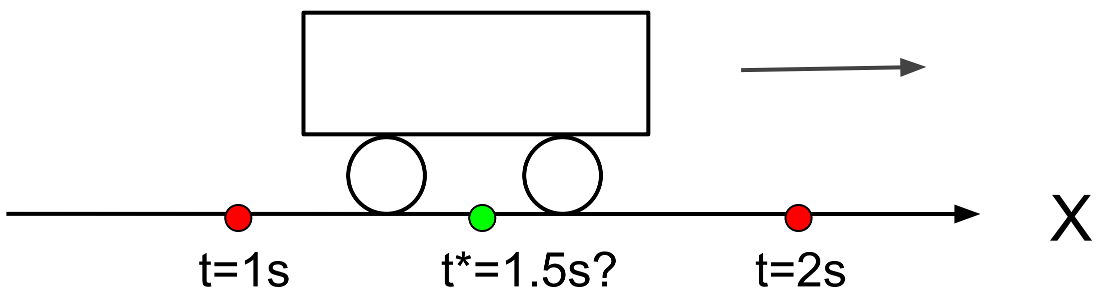
$$f(t) \sim \mathcal{GP}(m(t), k(t, t'))$$

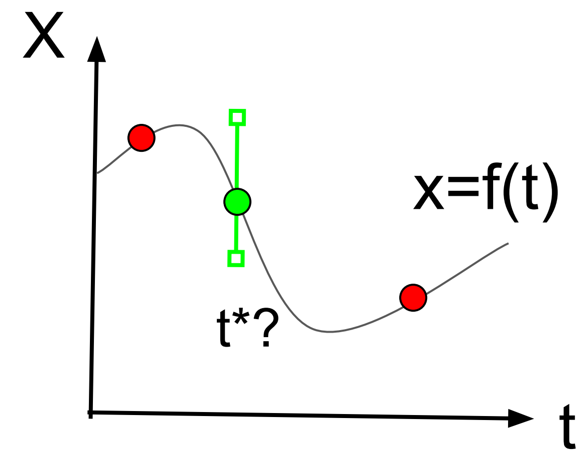
$$m(t) = \mathbb{E}(f(t))$$

$$k(t, t') = \mathbb{E}\Big(\big(f(t) - m(t)\big)\big(f(t') - m(t')\big)\Big)$$

Rasmussen, Carl Edward. "Gaussian processes for machine learning." (2006).

GP with noise-free measurements





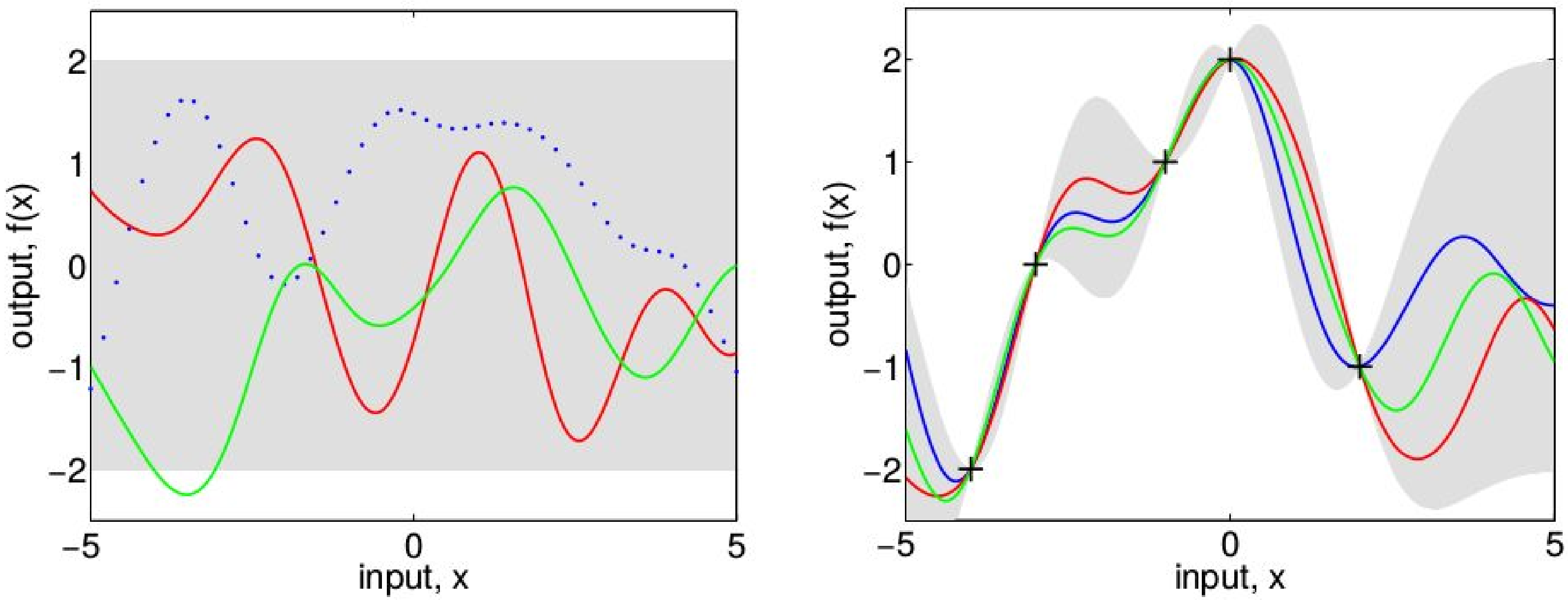
- Measurements
- Query Points

$$\begin{bmatrix} X \\ X^* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(T,T), K(T,T^*) \\ K(T^*,T), K(T^*,T^*) \end{bmatrix}\right)$$

$$X^*|T^*, T, X \sim \mathcal{N}\big(K(T^*,T)K(T,T)^{-1}X, \\ K(T^*,T^*) - K(T^*,T)K(T,T)^{-1}K(T,T^*)\big)$$

Rasmussen, Carl Edward. "Gaussian processes for machine learning." (2006).

GP with noise-free measurements

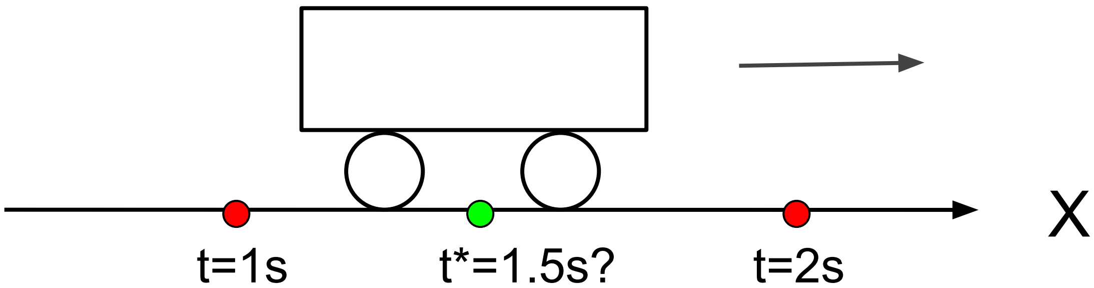


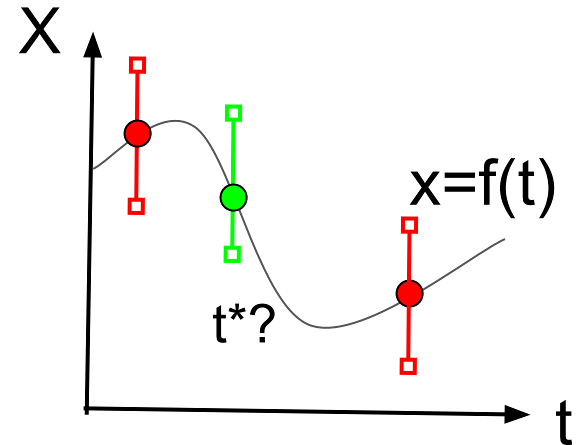
(a), prior (b), posterior

Figure 2.2: Panel (a) shows three functions drawn at random from a GP prior; the dots indicate values of y actually generated; the two other functions have (less correctly) been drawn as lines by joining a large number of evaluated points. Panel (b) shows three random functions drawn from the posterior, i.e. the prior conditioned on the five noise free observations indicated. In both plots the shaded area represents the pointwise mean plus and minus two times the standard deviation for each input value (corresponding to the 95% confidence region), for the prior and posterior respectively.

Rasmussen, Carl Edward. "Gaussian processes for machine learning." (2006).

GP with noisy measurements





Measurement function:

$$y = x + n, n \sim \mathcal{N}(0, \sigma)$$

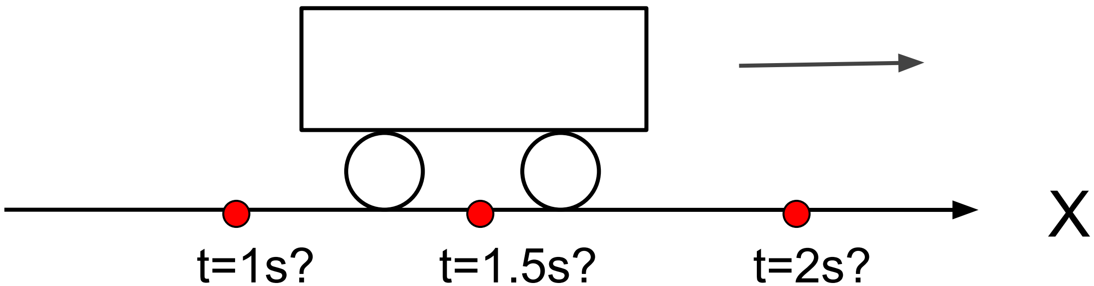
- Measurements (red)
- Query Points (green)

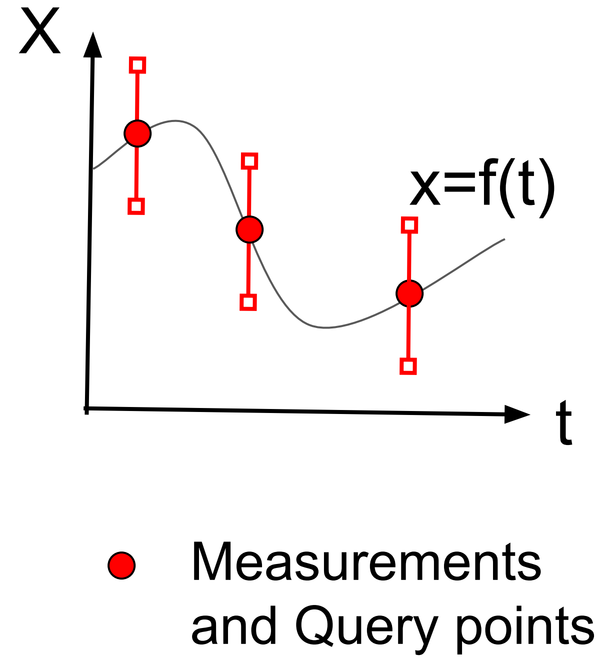
$$\begin{bmatrix} Y \\ X^* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(T,T) + \sigma I, K(T,T^*) \\ K(T^*,T), K(T^*,T^*) \end{bmatrix}\right)$$

$$X^*|T^*,T,Y \sim \mathcal{N}\big(K(T^*,T)[K(T,T)+\sigma I]^{-1}Y,\\ K(T^*,T^*)-K(T^*,T)[K(T,T)+\sigma I]^{-1}K(T,T^*)\big)$$

Rasmussen, Carl Edward. "Gaussian processes for machine learning." (2006).

GP with noisy measurements





Measurement function:

$$y = x + n, n \sim \mathcal{N}(0, \sigma)$$

Measurement and Query share same time stamps:

$$\begin{bmatrix} Y \\ X \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K + \sigma I, K \\ K, K \end{bmatrix}\right), K = K(T, T)$$

$$X|Y \sim \mathcal{N}\left(K[K + \sigma I]^{-1} Y, K - K[K + \sigma I]^{-1} K\right)$$

Rasmussen, Carl Edward. "Gaussian processes for machine learning." (2006).

Another perspective ...

Measurement function:

$$y = x + n, n \sim \mathcal{N}(0, \sigma)$$

Mean value: Maximum A Posteriori Estimation

$$\begin{aligned}\bar{X} &= \underset{X}{\operatorname{argmax}}\, P(X)P(Y|X) \\ &= \underset{X}{\operatorname{argmax}} \Big(\exp(\frac{1}{2}X^\top K^{-1}X) \prod_i \exp\big(\frac{(x_i - y_i)^2}{\sigma}\big)\Big) \\ &= \underset{X}{\operatorname{argmin}} \Big(\frac{1}{2}X^\top K^{-1}X + \frac{1}{2}(X-Y)^\top \sigma^{-1} I (X-Y)\Big)\end{aligned}$$

$$\frac{\partial J}{\partial X} = 0 \quad \Rightarrow \quad \bar{X} = (K^{-1} + \sigma^{-1}I)^{-1}\sigma^{-1}Y$$

$\Updownarrow$ Sherman–Morrison–Woodbury formula

$$\bar{X} = K(K + \sigma I)^{-1}Y$$

Rasmussen, Carl Edward. "Gaussian processes for machine learning." (2006).

Nonlinear measurement function and high DOF...

Problem definition:

$$\boldsymbol{x}(t) \sim \mathcal{GP}(\boldsymbol{\mu}(t), \mathcal{K}(t, t'))$$

$$\mathbf{z}_i = \mathbf{h}_i(\boldsymbol{x}(t_i)) + \mathbf{n}_i, \mathbf{n}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_i)$$

Mean value: MAP Estimation

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmax}} \left\{ \frac{1}{2} \parallel \boldsymbol{x} - \boldsymbol{\mu} \parallel^2_{\mathcal{K}} + \frac{1}{2} \parallel \mathbf{h}(\boldsymbol{x}) - \mathbf{z} \parallel^2_{\boldsymbol{\Sigma}} \right\}$$

Gauss-Newton method

$$(\mathcal{K}^{-1} + \mathbf{H}^\top \boldsymbol{\Sigma}^{-1} \mathbf{H}) \delta \boldsymbol{x}^* = \mathcal{K}^{-1}(\boldsymbol{\mu} - \overline{\boldsymbol{x}}) + \mathbf{H}^\top \boldsymbol{\Sigma}^{-1}(\mathbf{z} - \mathbf{h})$$

$$\overline{\boldsymbol{x}} \leftarrow \overline{\boldsymbol{x}} + \delta \boldsymbol{x}^* \text{ until convergence.}$$

T. Barfoot, C. H. Tong, and S. Sarkka, “Batch continuous-time trajectory estimation as exactly sparse gaussian process regression,” Robotics: Science and Systems (RSS), 2014.

Sparsity and Efficiency

Gauss-Newton step

$$(\mathcal{K}^{-1} + \mathbf{H}^\top \mathbf{\Sigma}^{-1} \mathbf{H}) \delta x^* = \mathcal{K}^{-1}(\boldsymbol{\mu} - \overline{\boldsymbol{x}}) + \mathbf{H}^\top \mathbf{\Sigma}^{-1}(\mathbf{z} - \mathbf{h})$$

Inverse Kernel (Prior)

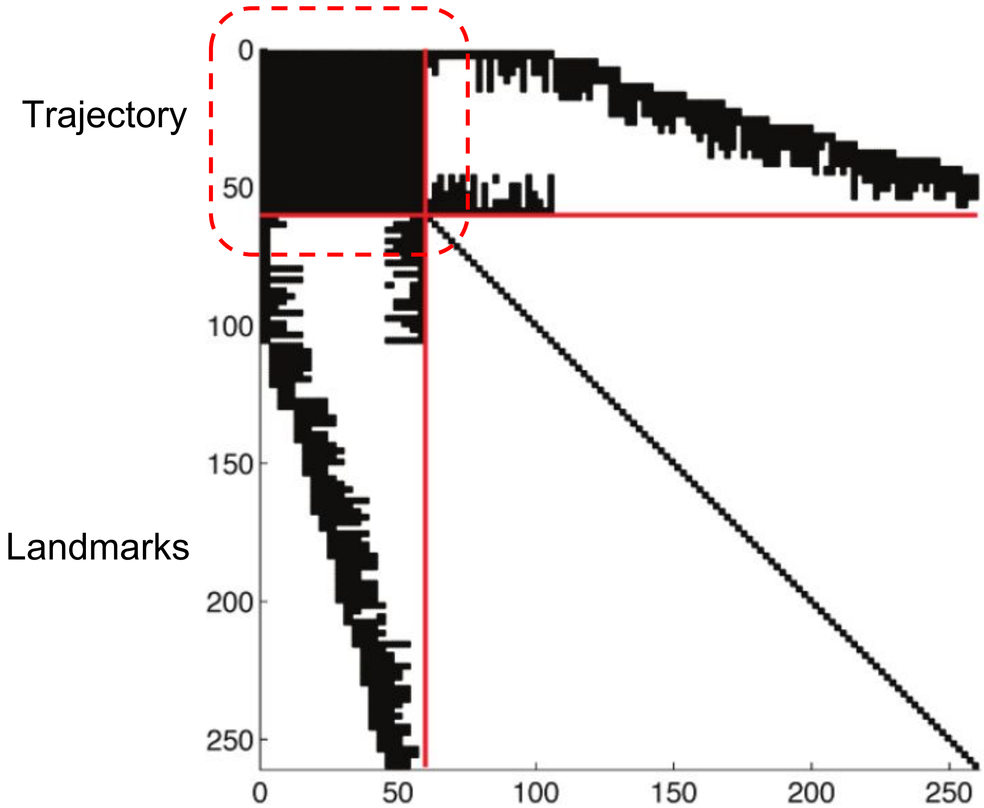
Measurements (Likelihood)

- Particular kernel is needed to make sparsity

T. Barfoot, C. H. Tong, and S. Sarkka, "Batch continuous-time trajectory estimation as exactly sparse gaussian process regression," Robotics: Science and Systems (RSS), 2014.

Sparse information matrix

- Information matrix: $\mathcal{K}^{-1} + \mathbf{H}^\top \mathbf{\Sigma}^{-1} \mathbf{H}$
- Generally the inverse kernel matrix $\mathcal{K}^{-1}$ is not sparse



C. H. Tong, P. Furgale, and T. D. Barfoot, "Gaussian process gauss–newton for non-parametric simultaneous localization and mapping," Intl. J. of Robotics Research, vol. 32, no. 5, pp. 507–525, 2013.

GPs generated by LTV-SDEs

- linear time-varying stochastic differential equations (LTV-SDEs)

$$\dot{\boldsymbol{x}}(t) = \mathbf{A}(t)\boldsymbol{x}(t) + \mathbf{u}(t) + \mathbf{F}(t)\mathbf{w}(t),$$
$$\mathbf{w}(t) \sim \mathcal{GP}(\mathbf{0}, \mathbf{Q}_C\delta(t - t')),$$

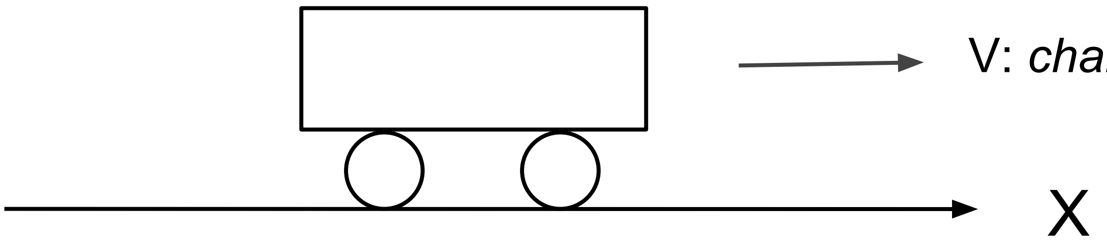
- solution:

$$\boldsymbol{\mu}(t) = \mathbf{\Phi}(t, t_0)\boldsymbol{\mu}_0 + \int_{t_0}^{t} \mathbf{\Phi}(t, s)\mathbf{u}(s)ds$$
$$\boldsymbol{\mathcal{K}}(t, t') = \mathbf{\Phi}(t, t_0)\boldsymbol{\mathcal{K}}_0\mathbf{\Phi}(t', t_0)^\top + \int_{t_0}^{\min(t,t')} \mathbf{\Phi}(t, s)\mathbf{F}(s)\mathbf{Q}_C\mathbf{F}(s)^\top\mathbf{\Phi}(t', s)^\top ds$$

- Inverse kernel matrix is tridiagonal block-wise sparse if the GP is generated by a LTV-SDE

T. Barfoot, C. H. Tong, and S. Sarkka, “Batch continuous-time trajectory estimation as exactly sparse gaussian process regression,” Robotics: Science and Systems (RSS), 2014.

Constant velocity LTV-SDEs



- Inject white noise in acceleration

$$\ddot{x}(t) = w(t)$$

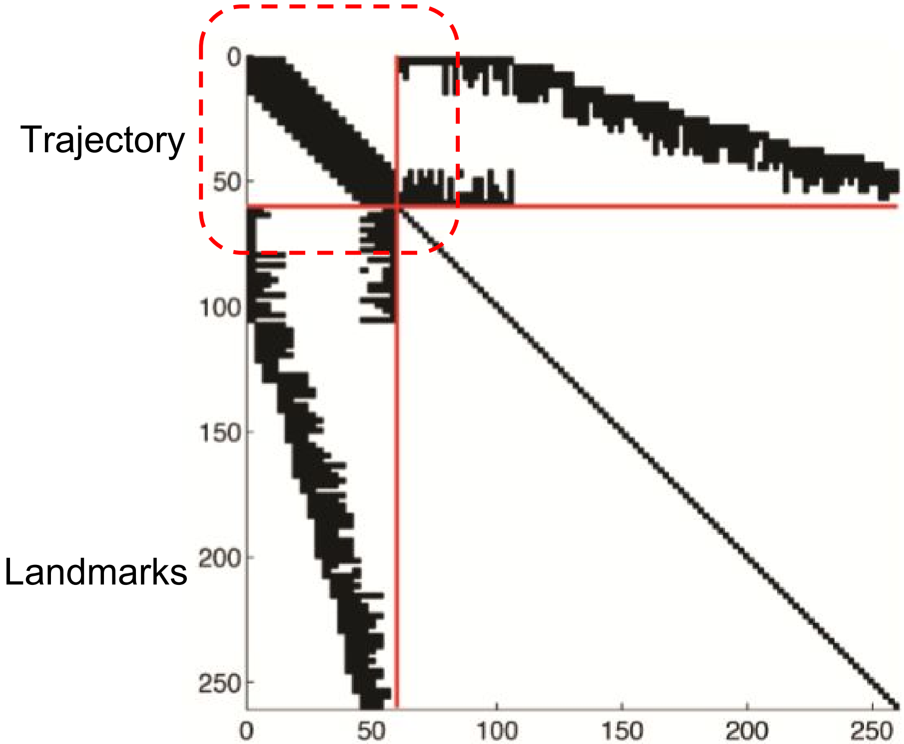
- Rewrite LTV-SDE

$$\mathbf{x}(t) = \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} \quad \mathbf{A}(t) = \begin{bmatrix} \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \mathbf{u}(t) = \mathbf{0}, \quad \mathbf{F}(t) = \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix}$$

$$\boxed{\begin{gathered} \dot{\boldsymbol{x}}(t) = \mathbf{A}(t)\boldsymbol{x}(t) + \mathbf{u}(t) + \mathbf{F}(t)\mathbf{w}(t), \\ \mathbf{w}(t) \sim \mathcal{GP}(\mathbf{0}, \mathbf{Q}_C \delta(t - t')), \end{gathered}}$$

T. Barfoot, C. H. Tong, and S. Sarkka, “Batch continuous-time trajectory estimation as exactly sparse gaussian process regression,” Robotics: Science and Systems (RSS), 2014.

Factor Graph view



Tri-diagonal inverse kernel

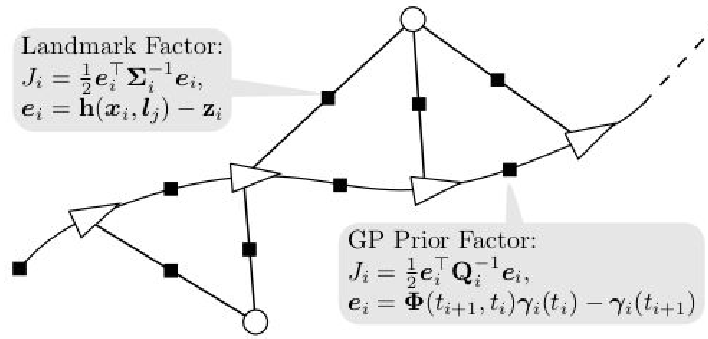


Fig. 3: A factor graph of an example STEAM problem containing GP prior factors and landmark measurements factors. Landmarks are illustrated with open circles.

T. Barfoot, C. H. Tong, and S. Sarkka, “Batch continuous-time trajectory estimation as exactly sparse gaussian process regression,” Robotics: Science and Systems (RSS), 2014.
Dong, Jing, Byron Boots, and Frank Dellaert. "Sparse Gaussian Processes for Continuous-Time Trajectory Estimation on Matrix Lie Groups." arXiv preprint arXiv:1705.06020 (2017).

Const-time Interpolation

- Any time could be interpolated by nearby two states
- Time complexity O(1)
- Measurements at any time fused in graph by interpolated factor

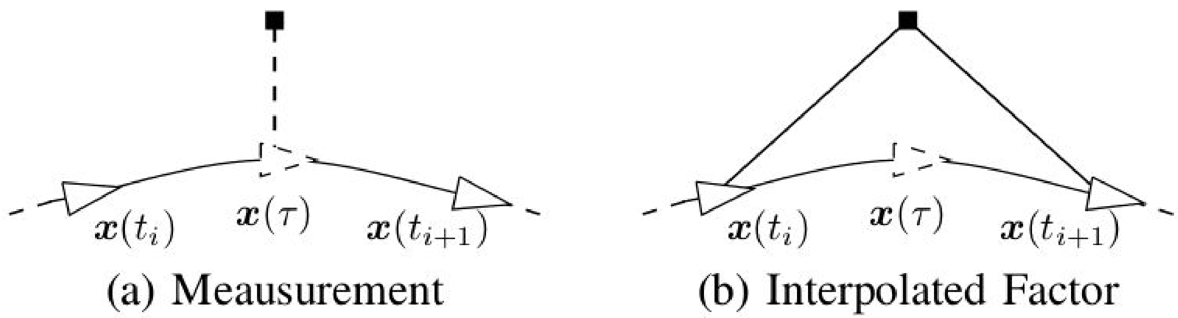
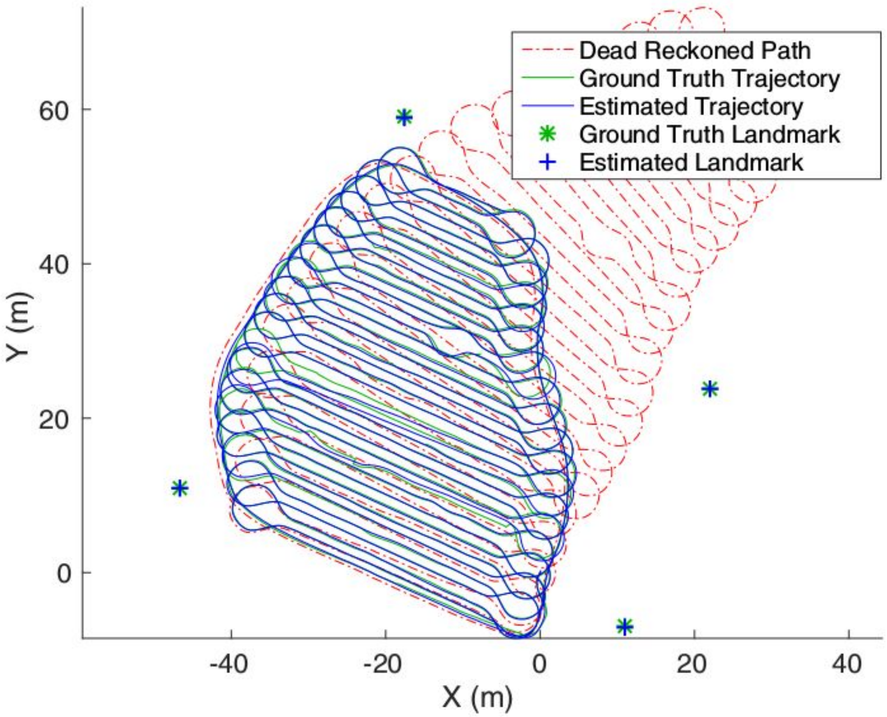


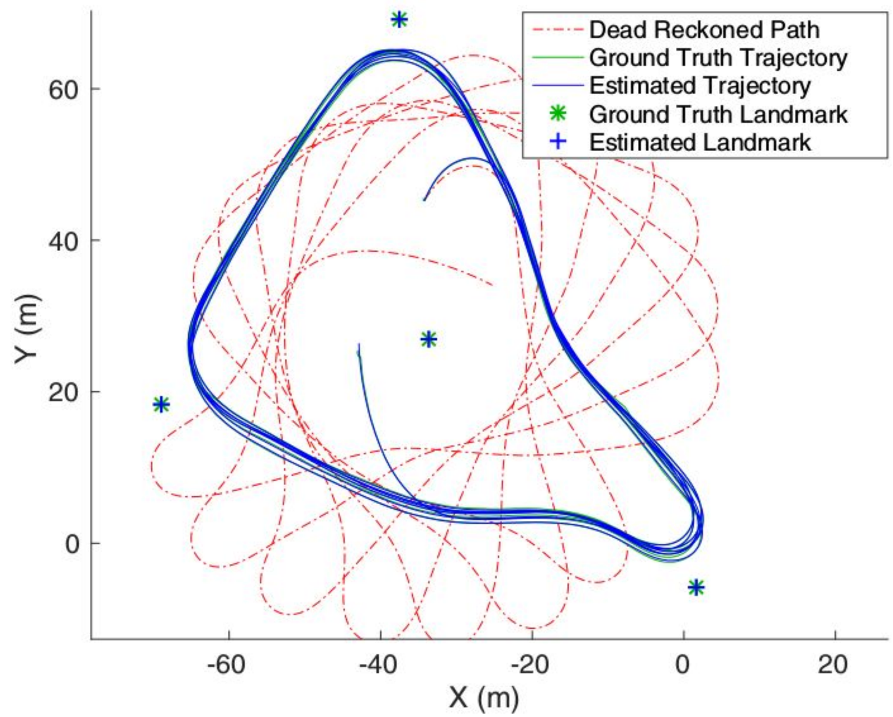
Fig. 2: (a) Measurement at time τ, dashed line indicates it's not an actual factor. (b) The interpolated factor encodes measurement at time τ.

T. Barfoot, C. H. Tong, and S. Sarkka, “Batch continuous-time trajectory estimation as exactly sparse gaussian process regression,” Robotics: Science and Systems (RSS), 2014.
Dong, Jing, Byron Boots, and Frank Dellaert. "Sparse Gaussian Processes for Continuous-Time Trajectory Estimation on Matrix Lie Groups." arXiv preprint arXiv:1705.06020 (2017).

Results

- 2D SLAM cases, $\boldsymbol{x}(t) = [x(t), y(t), \theta(t)]^\top$

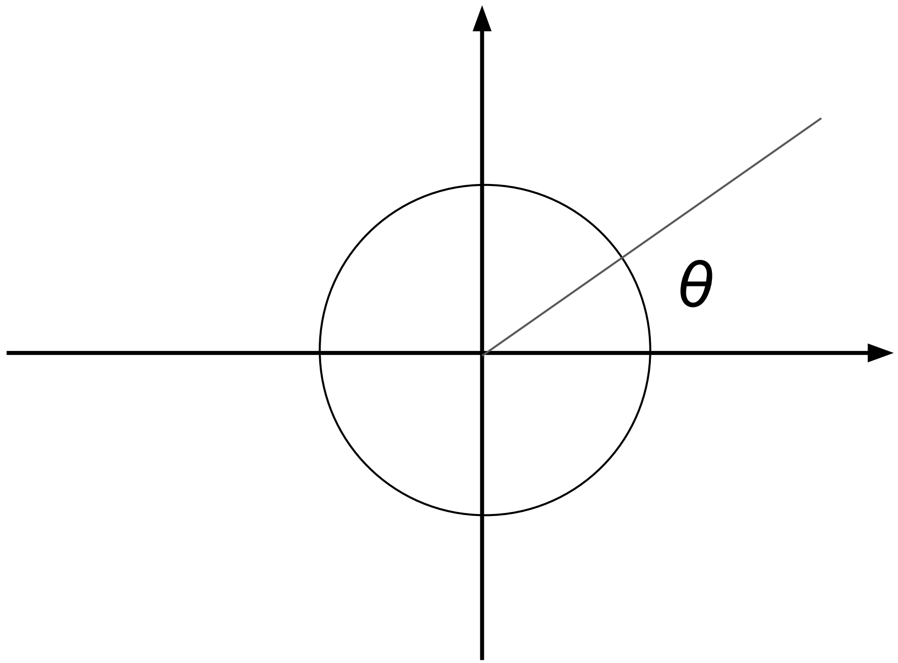




Outlines

- GP as continuous-time trajectory representation
- Extend sparse GP to Lie groups
- Use sparse GP in SLAM
- Use sparse GP in motion planning

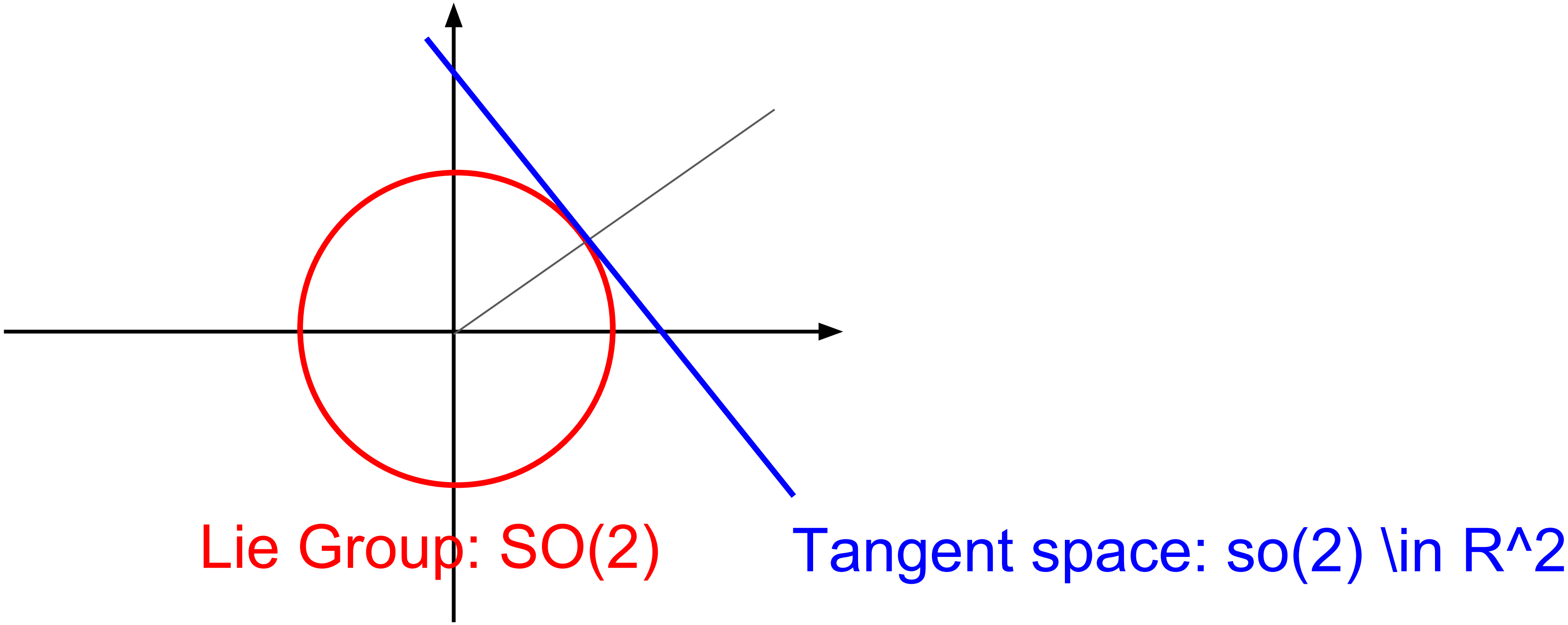
Example Lie Group: SO(2)



$$\begin{bmatrix} \cos(\theta), \sin(\theta) \\ -\sin(\theta), \cos(\theta) \end{bmatrix}$$

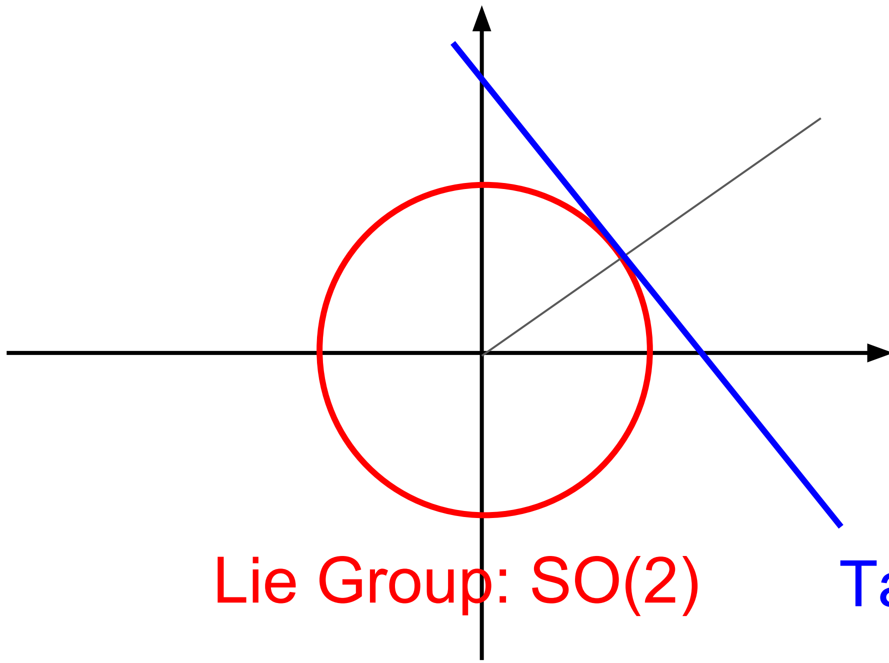
S. Anderson and T. D. Barfoot, “Full STEAM ahead: Exactly sparse gaussian process regression for batch continuous-time trajectory estimation on SE (3),” (IROS), 2015.
Dong, Jing, Byron Boots, and Frank Dellaert. "Sparse Gaussian Processes for Continuous-Time Trajectory Estimation on Matrix Lie Groups." arXiv preprint arXiv:1705.06020 (2017).

Gaussian Distribution on SO(2)



S. Anderson and T. D. Barfoot, “Full STEAM ahead: Exactly sparse gaussian process regression for batch continuous-time trajectory estimation on SE (3),” (IROS), 2015.
Dong, Jing, Byron Boots, and Frank Dellaert. "Sparse Gaussian Processes for Continuous-Time Trajectory Estimation on Matrix Lie Groups." arXiv preprint arXiv:1705.06020 (2017).

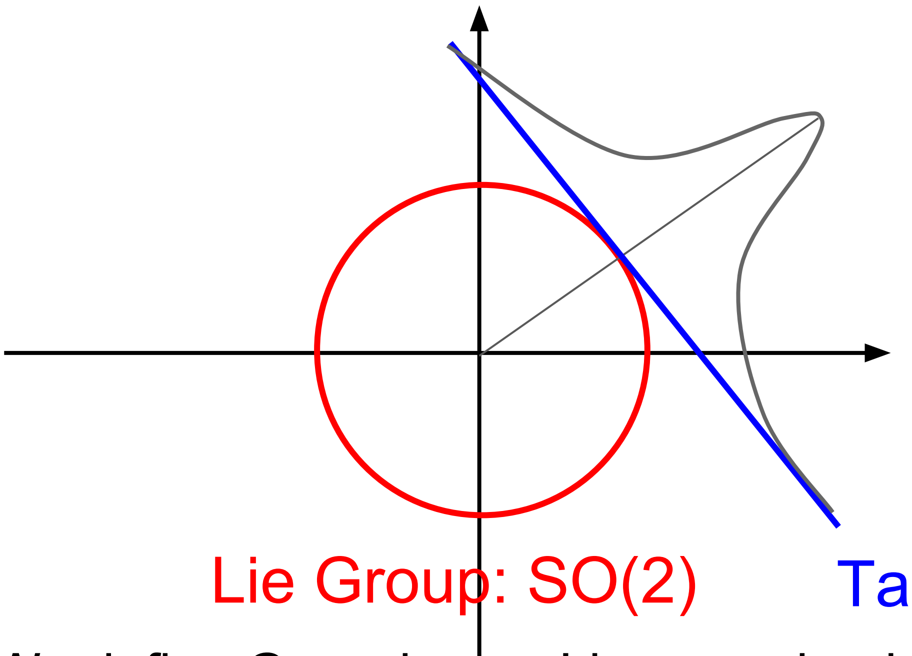
Gaussian Distribution on SO(2)



Expmap: so(2) -> SO(2)
Logmap: SO(2) -> so(2)

S. Anderson and T. D. Barfoot, “Full STEAM ahead: Exactly sparse gaussian process regression for batch continuous-time trajectory estimation on SE (3),” (IROS), 2015.
Dong, Jing, Byron Boots, and Frank Dellaert. "Sparse Gaussian Processes for Continuous-Time Trajectory Estimation on Matrix Lie Groups." arXiv preprint arXiv:1705.06020 (2017).

Gaussian Distribution on SO(2)



Expmap: so(2) -> SO(2)
Logmap: SO(2) -> so(2)

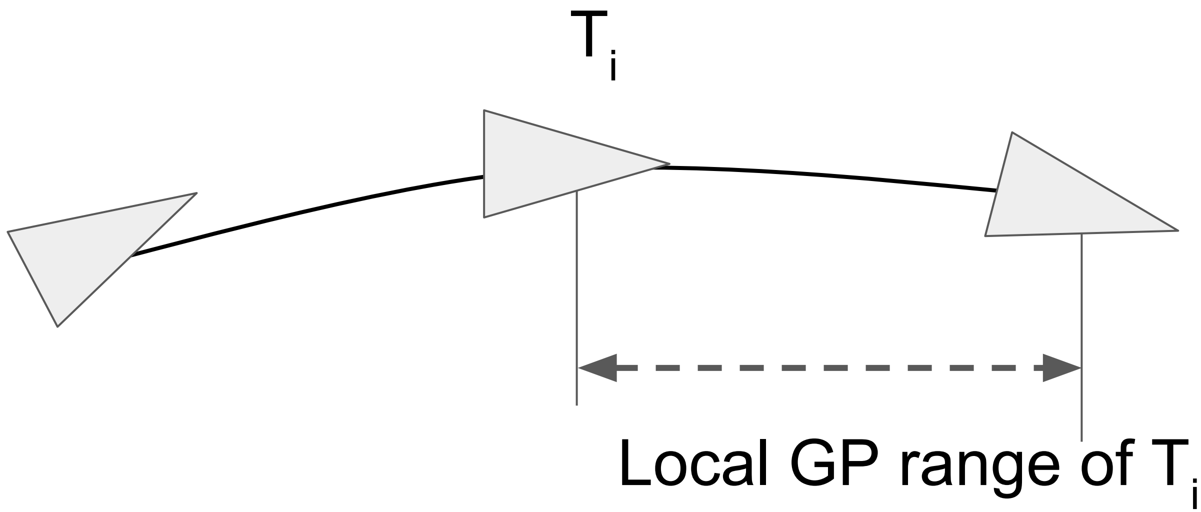
- We define Gaussian on Lie group by defining Gaussian on tangent space, then map back by exponential map

$$\widetilde{\mathbf{R}} = \mathbf{R}\exp(\epsilon^\wedge), \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \Sigma)$$

- Note that tangent space is defined on a particular θ on Lie group

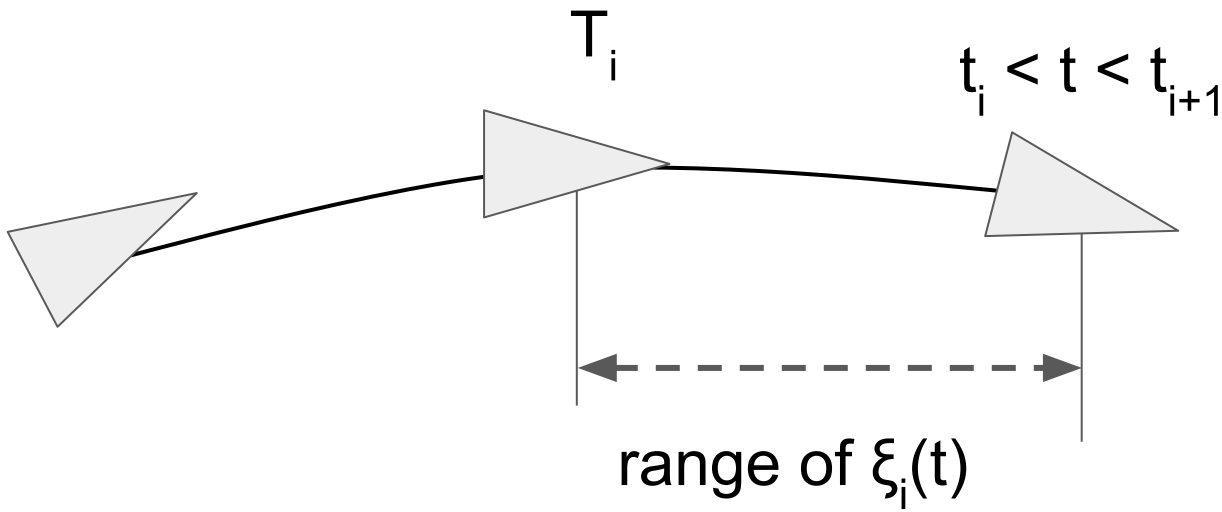
S. Anderson and T. D. Barfoot, “Full STEAM ahead: Exactly sparse gaussian process regression for batch continuous-time trajectory estimation on SE (3),” (IROS), 2015.
Dong, Jing, Byron Boots, and Frank Dellaert. "Sparse Gaussian Processes for Continuous-Time Trajectory Estimation on Matrix Lie Groups." arXiv preprint arXiv:1705.06020 (2017).

Local GP on SO(2)



S. Anderson and T. D. Barfoot, “Full STEAM ahead: Exactly sparse gaussian process regression for batch continuous-time trajectory estimation on SE (3),” (IROS), 2015.
Dong, Jing, Byron Boots, and Frank Dellaert. "Sparse Gaussian Processes for Continuous-Time Trajectory Estimation on Matrix Lie Groups." arXiv preprint arXiv:1705.06020 (2017).

Local GP on SO(2)



- We define a local linear variable $\xi_i(t)$ around T_i, which locally meets a zero-mean Gaussian process defined by kernel $K(t_i, t)$

$$T(t) = T_i \exp(\boldsymbol{\xi}_i(t)^\wedge), \quad \boldsymbol{\xi}_i(t) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\mathcal{K}}(t_i, t))$$

$$\boldsymbol{\xi}_i(t) \doteq \log(T_i^{-1} T(t))^\vee$$

S. Anderson and T. D. Barfoot, “Full STEAM ahead: Exactly sparse gaussian process regression for batch continuous-time trajectory estimation on SE (3),” (IROS), 2015.
Dong, Jing, Byron Boots, and Frank Dellaert. "Sparse Gaussian Processes for Continuous-Time Trajectory Estimation on Matrix Lie Groups." arXiv preprint arXiv:1705.06020 (2017).

Constant Velocity LVT-SDE

- Const body frame velocity SDE: nonlinear

$$\dot{\boldsymbol{\varpi}}(t) = \mathbf{w}(t),$$

- Equivalence to LTV-SDE on loca variable

$$\boldsymbol{\xi}_i(t) \doteq \log(T_i^{-1} T(t))^{\vee}$$

$$\dot{\boldsymbol{\xi}}_i(t) = \boldsymbol{\mathcal{J}}_r(\boldsymbol{\xi}_i(t))^{-1} \boldsymbol{\varpi}(t) \quad \dot{\boldsymbol{\xi}}_i(t) \approx \boldsymbol{\varpi}(t)$$

- Local constant velocity LTV-SDE

$$\ddot{\boldsymbol{\xi}}(t) = \mathbf{w}(t)$$

S. Anderson and T. D. Barfoot, “Full STEAM ahead: Exactly sparse gaussian process regression for batch continuous-time trajectory estimation on SE (3),” (IROS), 2015.
Dong, Jing, Byron Boots, and Frank Dellaert. "Sparse Gaussian Processes for Continuous-Time Trajectory Estimation on Matrix Lie Groups." arXiv preprint arXiv:1705.06020 (2017).

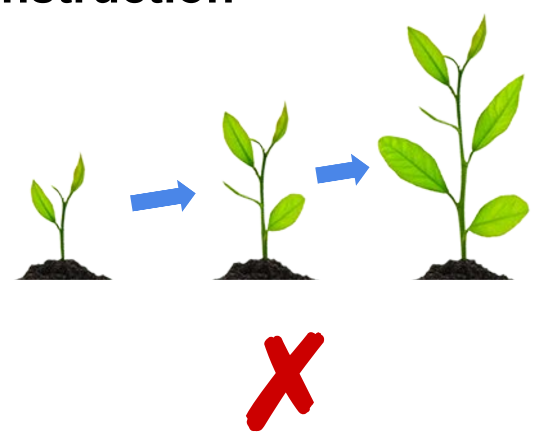
Outlines

- GP as continuous-time trajectory representation
- Extend sparse GP to Lie groups
- Use sparse GP in SLAM
- Use sparse GP in motion planning

Background

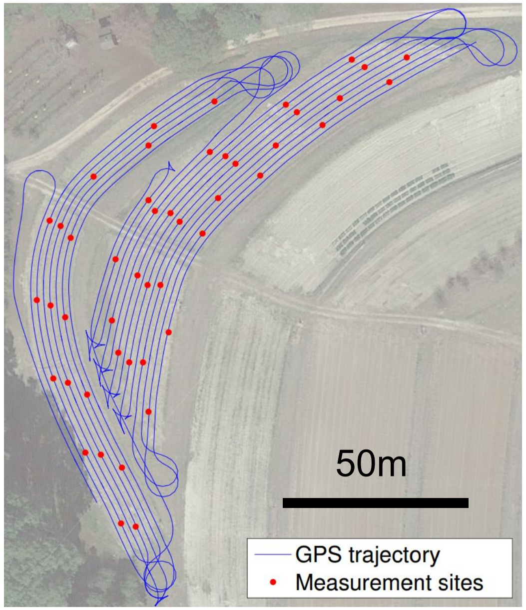
- Application: autonomous crop monitoring using UAV / UGV
- Get crop height / size / color / etc.
- Requirement: use low-cost sensors: camera, GPS, etc.

- Existing techniques: Structure from Motion (SfM): 3D reconstruction
- Difficulties: dynamic scene with crop growing: **4D reconstruction**



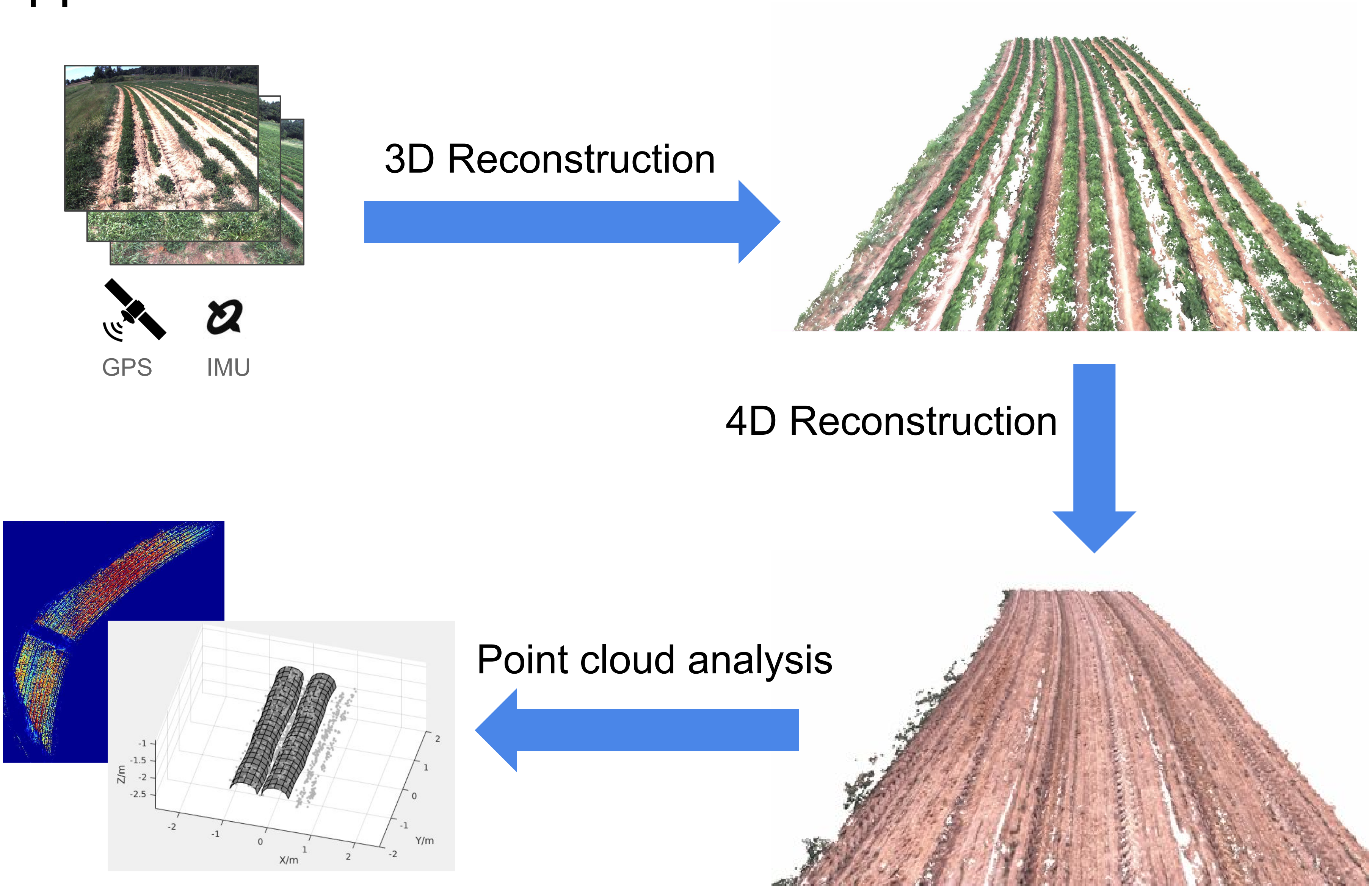
Dataset

- Dataset: color images + GPS + IMU, on ground vehicle and UAV
- Ground dataset: Twice per week for 3 months: total 23 sessions
- With ground truth height / leaf chlorophyll at multiple sampling sites



Dong, Jing, et al. "4D Crop Monitoring: Spatio-Temporal Reconstruction for Agriculture." ICRA (2017).

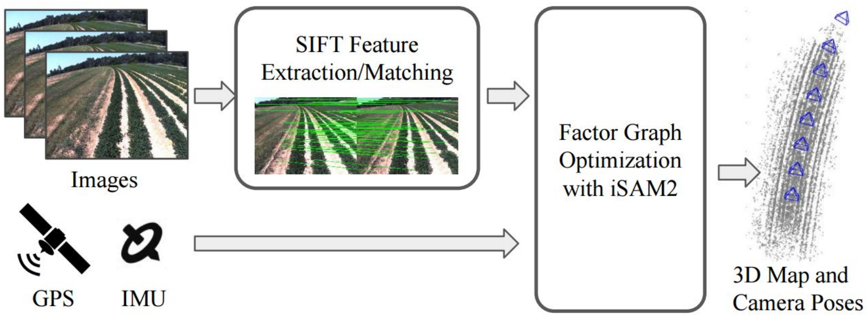
Approach

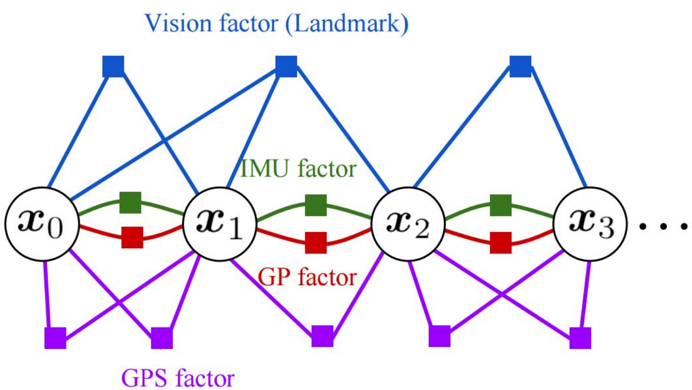


Dong, Jing, et al. "4D Crop Monitoring: Spatio-Temporal Reconstruction for Agriculture." ICRA (2017).

1. Multi-Sensor Fusion SLAM

- Multi-sensor (camera, IMU, GPS) fusion
- Using factor graph and maximum a posterior (MAP) estimation
- Optional dense reconstruction by PMVS



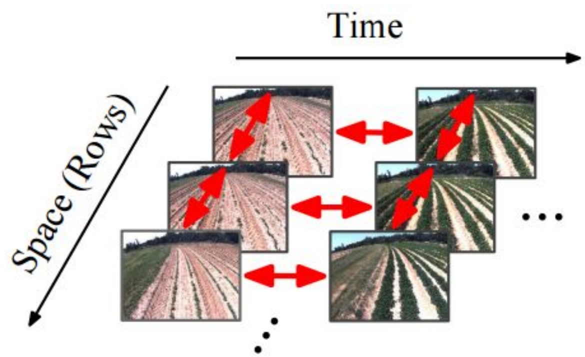


3D factor graph

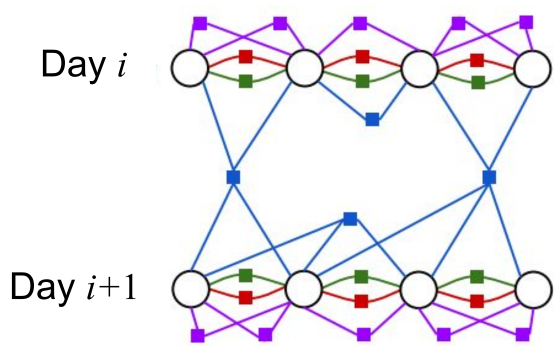
Dong, Jing, et al. "4D Crop Monitoring: Spatio-Temporal Reconstruction for Agriculture." ICRA (2017).

2. 4D Data Association and Reconstruction

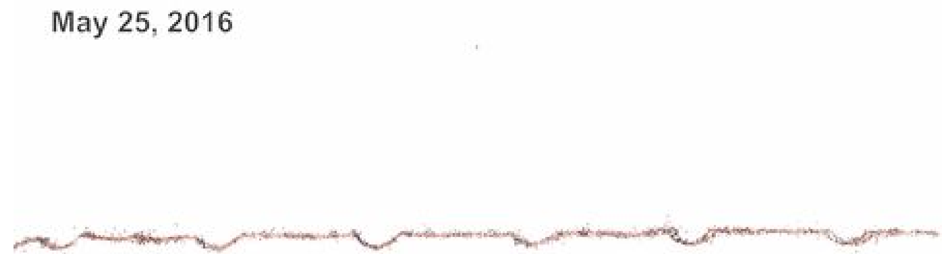
- View-invariant feature matching by bounded search and homography warping
- Spatio-temporal (4D) factor graph optimization

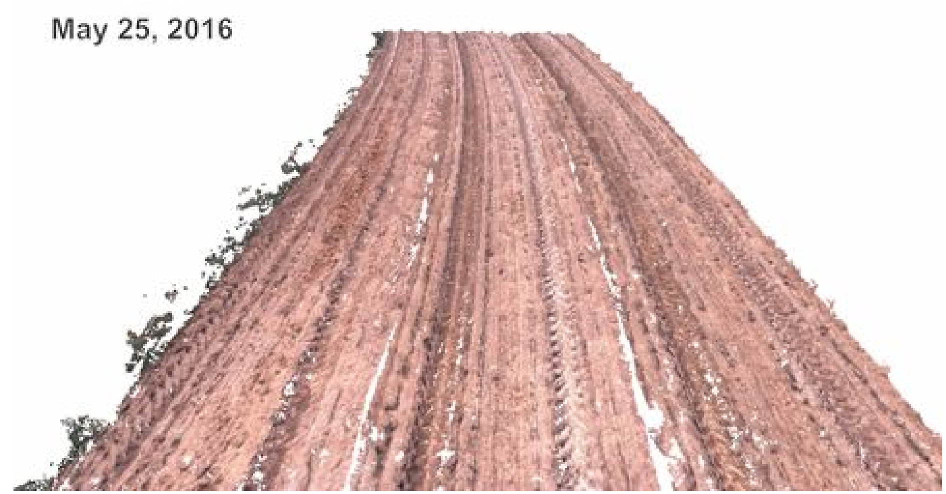


View-invariant robust feature matching



Spatio-temporal (4D) factor graph





Dong, Jing, et al. "4D Crop Monitoring: Spatio-Temporal Reconstruction for Agriculture." ICRA (2017).

Results

https://youtu.be/BgLlLlsKWzI

Dong, Jing, et al. "4D Crop Monitoring: Spatio-Temporal Reconstruction for Agriculture." ICRA (2017).

Outlines

- GP as continuous-time trajectory representation
- Extend sparse GP to Lie groups
- Use sparse GP in SLAM
- Use sparse GP in motion planning

Introduction

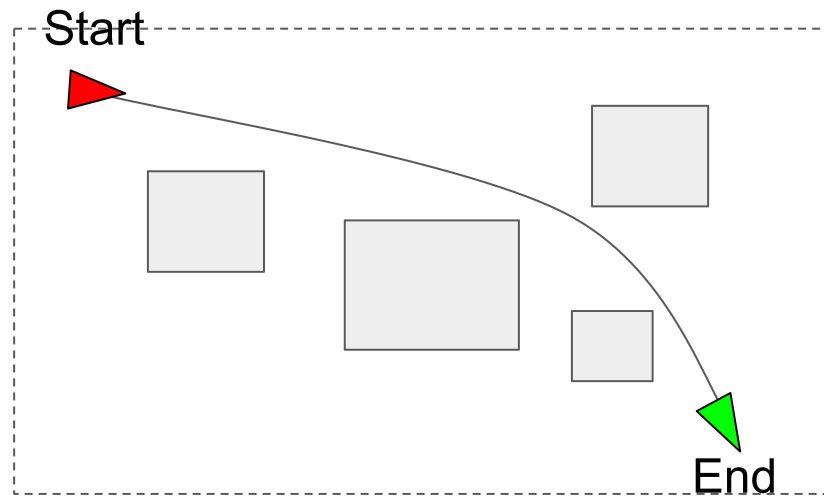
- Planning on high DOF system is difficult
- Sampling based approaches like RRT and RPM are slow due to high DOF

- State-of-the-art approaches are mostly optimization based
    - Energy reduction (Zucker et al, 2013 IJRR, Ratliff et al, 2009 ICRA)
    - Constrained optimization (Schluman et al, 2013 RSS)
    - Factor Graph and message passing (Toussaint et al, 2010 Book)

- Main Contributions:
    - Formulate motion planning as continuous trajectory estimation by GP
    - Enable efficient motion planning by sparse GP and iSAM2

Motion Planning as Probabilistic inference

- Two requirements of motion planning:
  - *Feasibility*: trajectory is collision-free
  - *Optimality*: smooth output by reducing the energy used or trajectory length



- Calculate the trajectory by maximize the probability

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, P(\boldsymbol{\theta}) P(\mathbf{c} = 0 | \boldsymbol{\theta})$$

Prior distribution: enforce smoothness

Collision-free likelihood: enforce feasibility

Dong, Jing, et al. "Motion Planning as Probabilistic Inference using Gaussian Processes and Factor Graphs." *Robotics: Science and Systems (RSS)* (2016).

Collision-free Likelihood

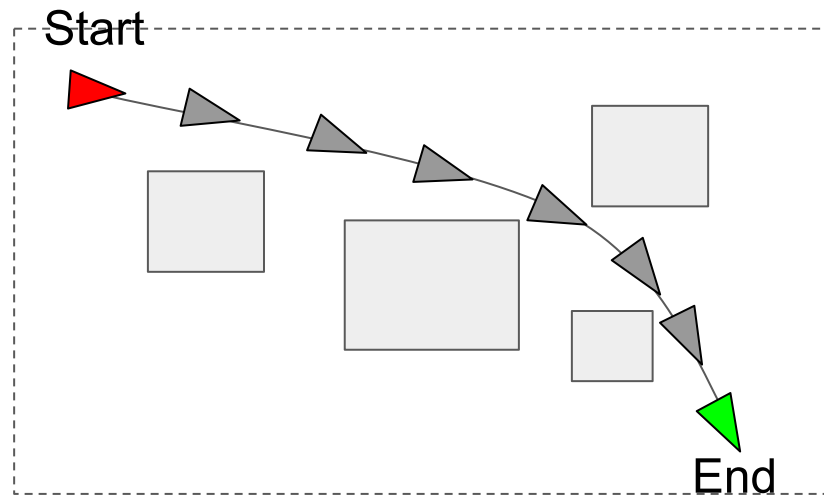
- Collision-free likelihood is calculated at each discretized points of trajectory

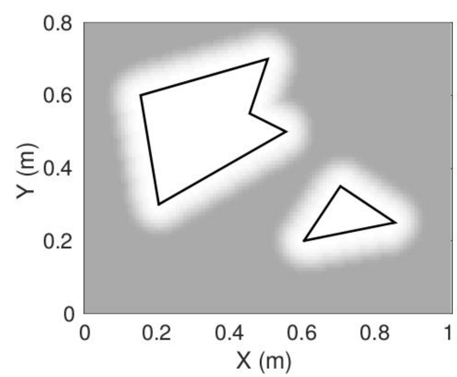
$$L(\boldsymbol{\theta}|\mathbf{c}=0) = \prod_i L(\boldsymbol{\theta_i}|c_i=0)$$

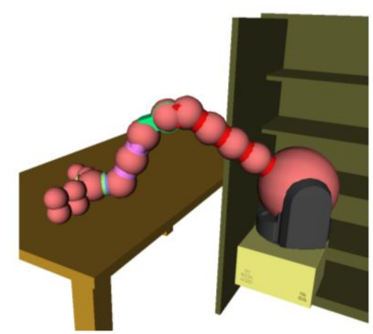
- Collision-free likelihood is defined by exponential family

$$L(\boldsymbol{\theta_i}) = \exp(-\frac{1}{2}h(\boldsymbol{\theta_i}))$$

- Obstacle cost function is defined by hinge loss function on signed distance
  - Signed distance field is pre-calculated
  - Robot body use sphere model to approximate





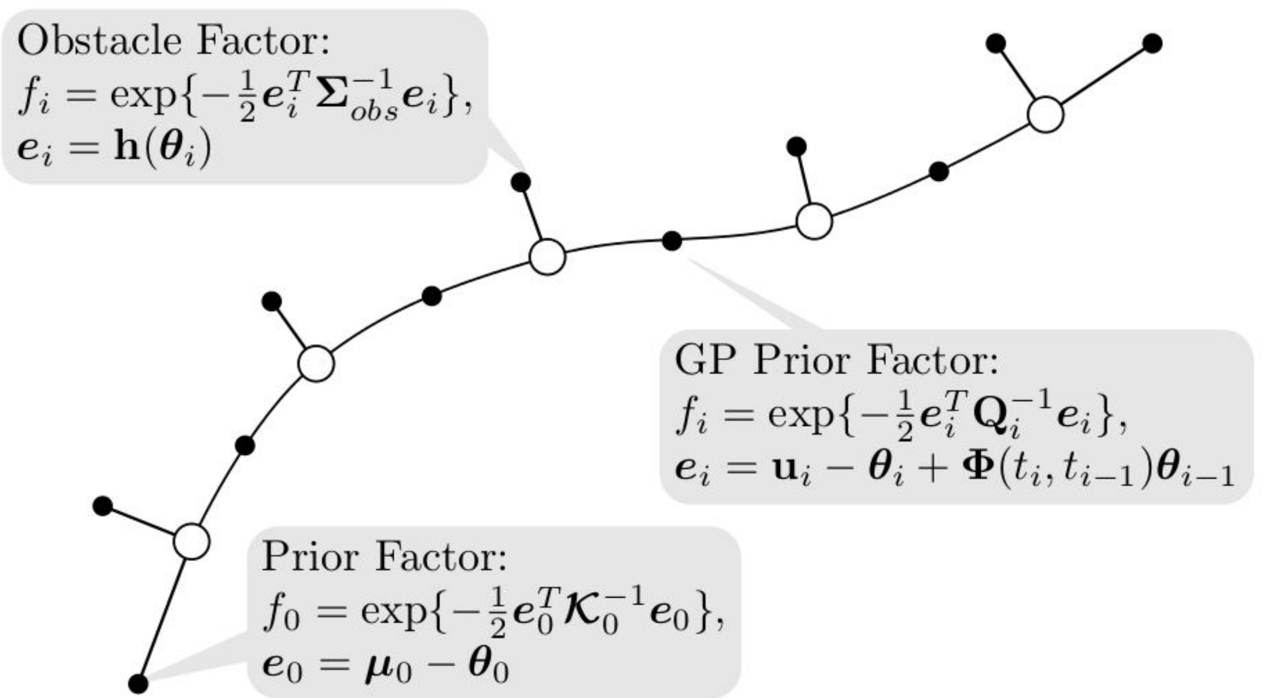


Dong, Jing, et al. "Motion Planning as Probabilistic Inference using Gaussian Processes and Factor Graphs." *Robotics: Science and Systems (RSS)* (2016).

GP prior and Factor graph

- Trajectory is represented by a GP
- Trajectory prior use constant velocity GP prior
- Additional prior on start/goal configuration (Since they are known)

- Inference problem represented by a factor graph



Dong, Jing, et al. "Motion Planning as Probabilistic Inference using Gaussian Processes and Factor Graphs." *Robotics: Science and Systems (RSS)* (2016).

Duality between inference and energy minimization

Trajectory Prior — Collision-free Likelihood

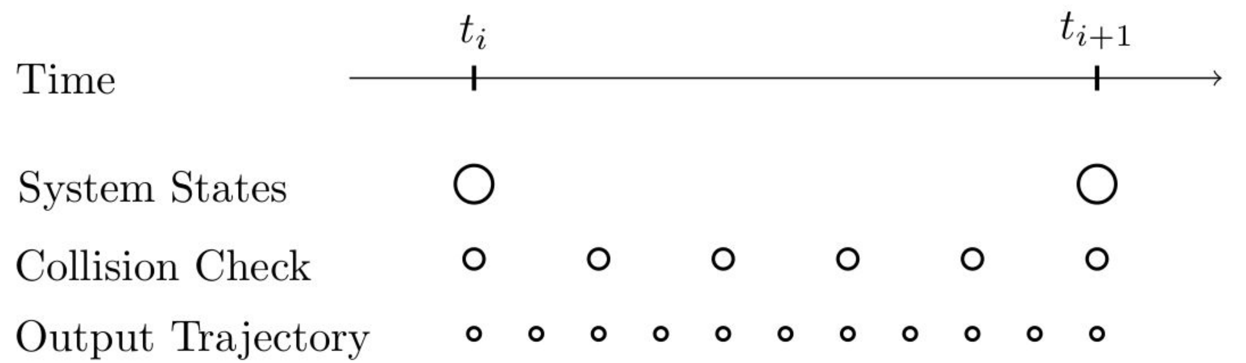
$$\begin{aligned}
\boldsymbol{\theta}^* &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left\{ P(\boldsymbol{\theta}) \prod_i P(c_i|\boldsymbol{\theta}_i) \right\}, \\
&= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left\{ -\log \left(P(\boldsymbol{\theta}) \prod_i P(c_i|\boldsymbol{\theta}_i) \right) \right\}, \\
&= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left\{ \frac{1}{2} \parallel \boldsymbol{\theta} - \boldsymbol{\mu} \parallel^2_{\mathcal{K}} + \frac{1}{2} \parallel \boldsymbol{h}(\boldsymbol{\theta}) \parallel^2_{\boldsymbol{\Sigma}_{obs}} \right\}
\end{aligned}$$

Smoothness Cost — Collision Cost

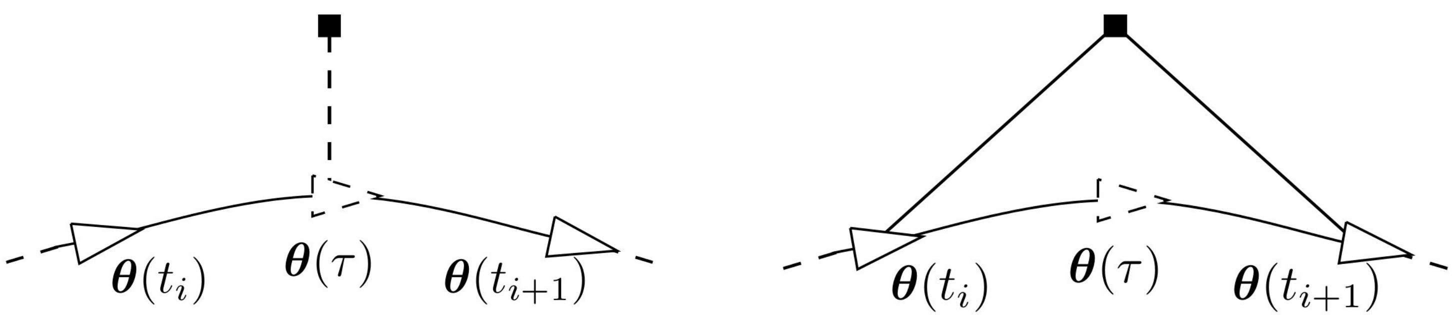
Dong, Jing, et al. "Motion Planning as Probabilistic Inference using Gaussian Processes and Factor Graphs." *Robotics: Science and Systems (RSS)* (2016).

Efficient planning: Use GP interpolation

- GP enable us to query any time of interest
- So we can check collision cost at more points with less states to optimize



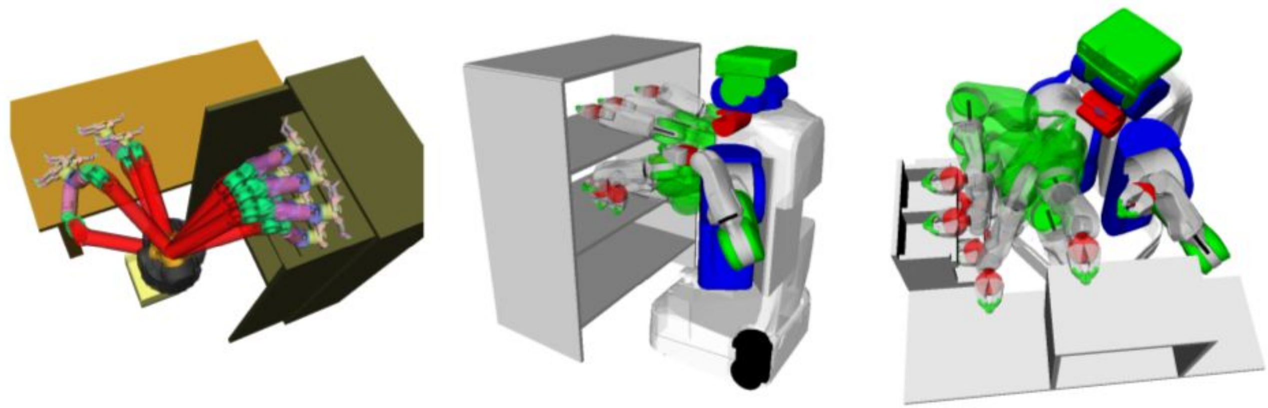
- A single query state is interpolated by near by two states
- Interpolated collision cost factor is binary



Dong, Jing, et al. "Motion Planning as Probabilistic Inference using Gaussian Processes and Factor Graphs." *Robotics: Science and Systems (RSS)* (2016).

Evaluation: Batch planning

- Dataset: 7-DOF WAM arm and 7-DOF PR2, total 114 problems



- Criteria: Success rate, average run-time, maximum run-time

Table I.A Results for 24 planning problems on the 7-DOF WAM arm.

	GPMP2_inter	GPMP2_no-inter	TrajOpt-101	TrajOpt-11	GPMP	CHOMP	STOMP
Success Rate (%)	**95.8**	91.7	91.7	20.8	**95.8**	75.0	40.0
Average Time to Success (s)	0.068	0.120	0.323	**0.027**	0.590	1.337	6.038
Maximum Time to Success (s)	0.112	0.217	0.548	**0.033**	1.322	6.768	22.971

Table I.B Results for 90 planning problems on PR2's 7-DOF right arm.

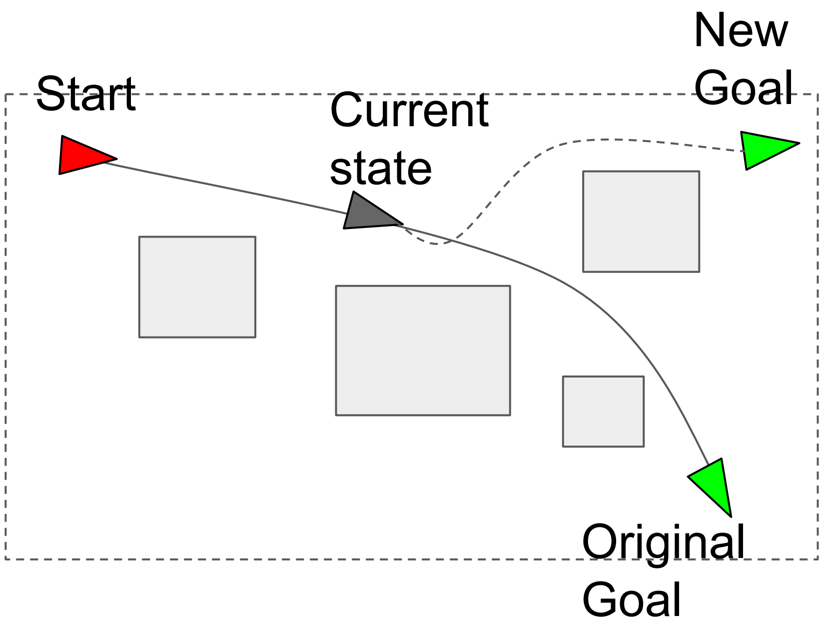
	GPMP2_inter	GPMP2_no-inter	TrajOpt-51	TrajOpt-11	GPMP	CHOMP	STOMP
Success Rate (%)	**94.4**	88.9	93.3	88.9	47.8	80.0	52.4
Average Time to Success (s)	**0.033**	0.053	0.860	0.168	1.134	4.999	7.586
Maximum Time to Success (s)	**0.083**	0.120	4.827	0.455	9.516	44.623	95.679

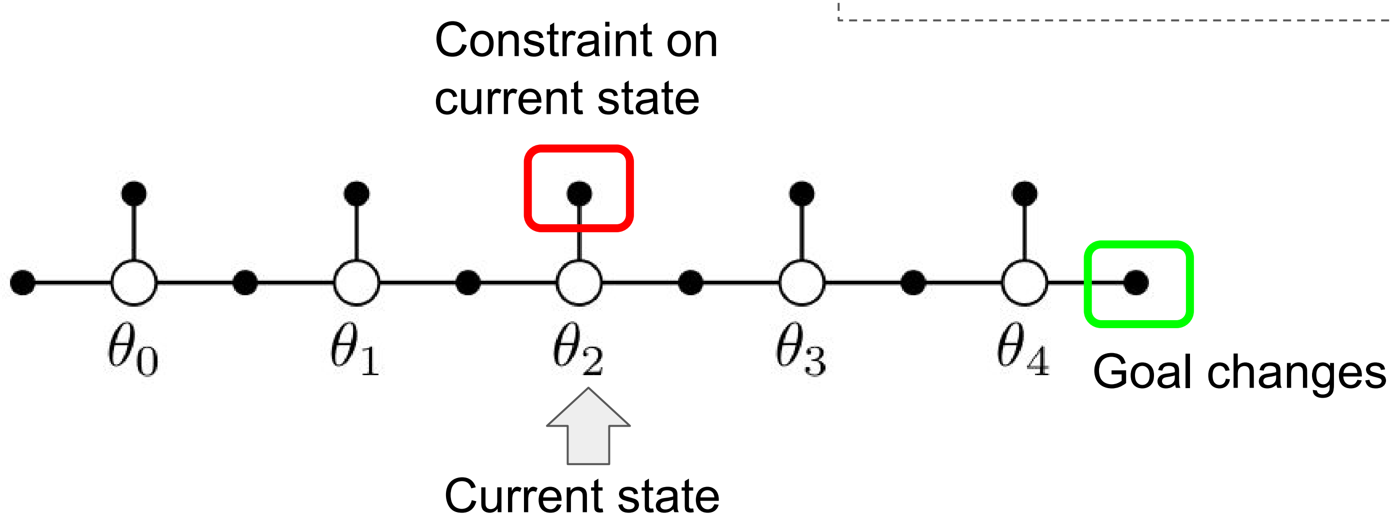
Dong, Jing, et al. "Motion Planning as Probabilistic Inference using Gaussian Processes and Factor Graphs." *Robotics: Science and Systems (RSS)* (2016).

Efficient planning: Replanning

- Example replanning problem: given current state and a new goal
- Naive approach: plan from scratch given current (as start) and goal states

- Fact: planning problem changes little





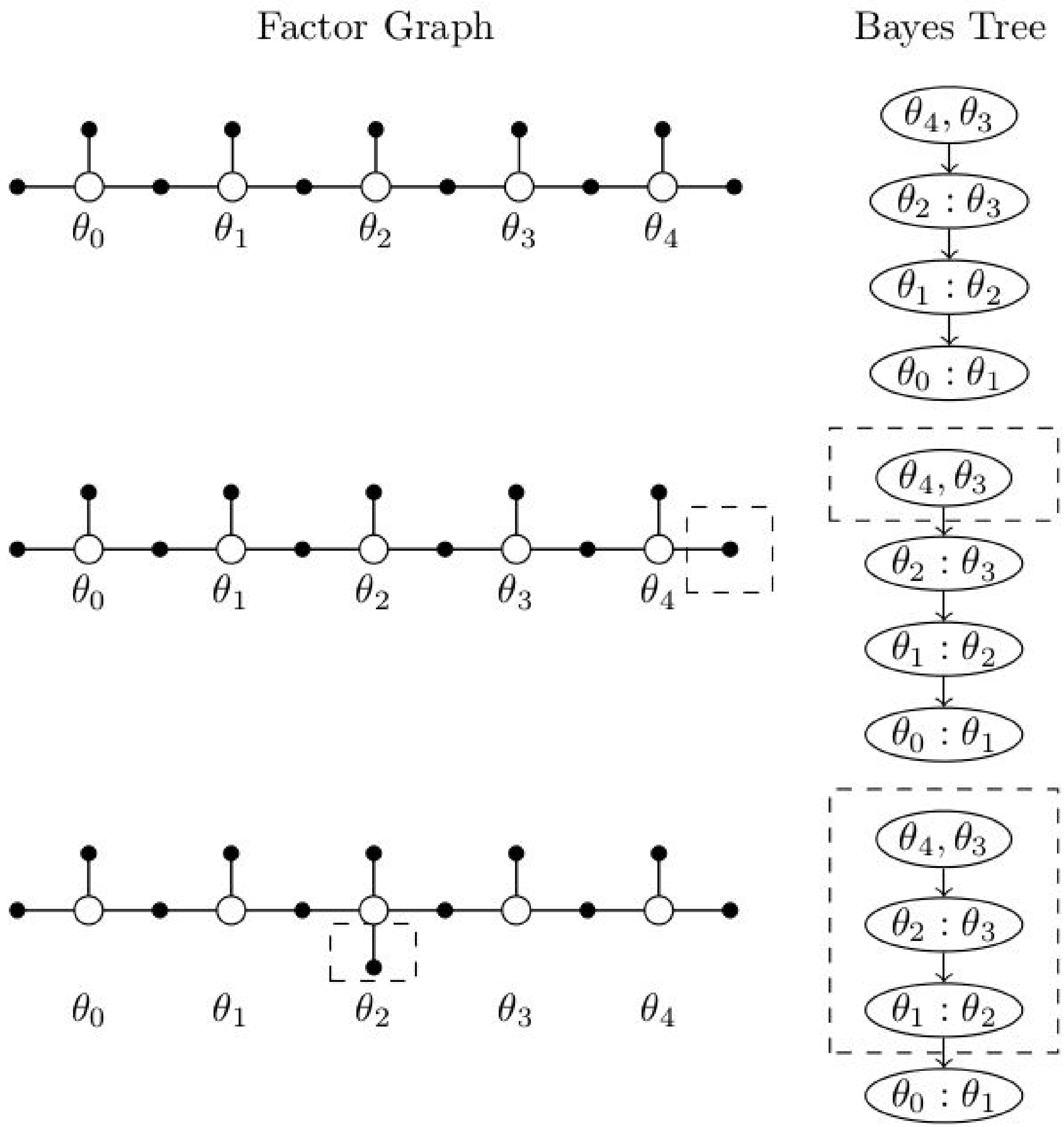
Dong, Jing, et al. "Motion Planning as Probabilistic Inference using Gaussian Processes and Factor Graphs." *Robotics: Science and Systems (RSS)* (2016).
Mukadam, Mustafa, et al. "Simultaneous Trajectory Estimation and Planning via Probabilistic Inference." *Robotics: Science and Systems (RSS)* (2017).

Efficient replanning: Bayes tree and iSAM2

Original graph

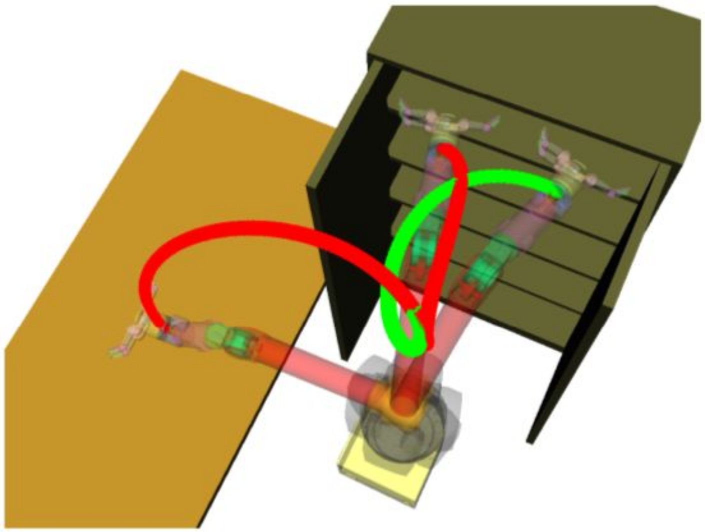
Goal changes

Constraint on current state



Evaluation: Replanning

- Dataset: 7-DOF WAM arm and 7-DOF PR2, total 60 problems



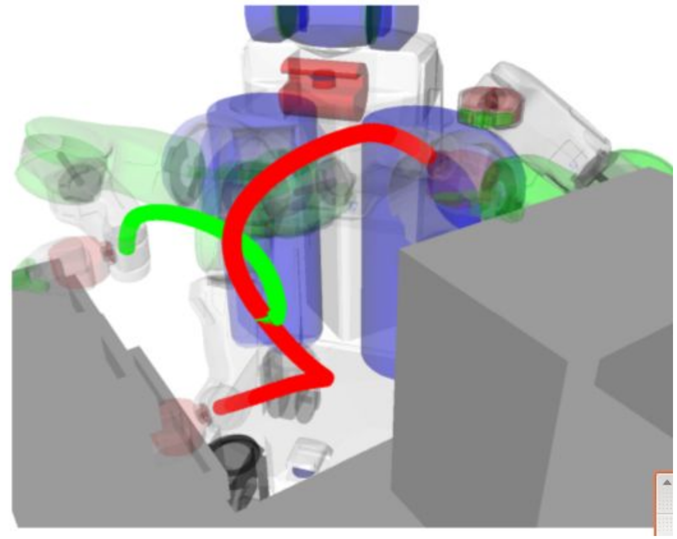


Table II.A Results for 32 replanning problems on WAM.

	iGPMP2	GPMP2
Success Rate (%)	90.6	**100.0**
Average Time to Success (ms)	**2.38**	30.21
Maximum Time to Success (ms)	**3.92**	46.60

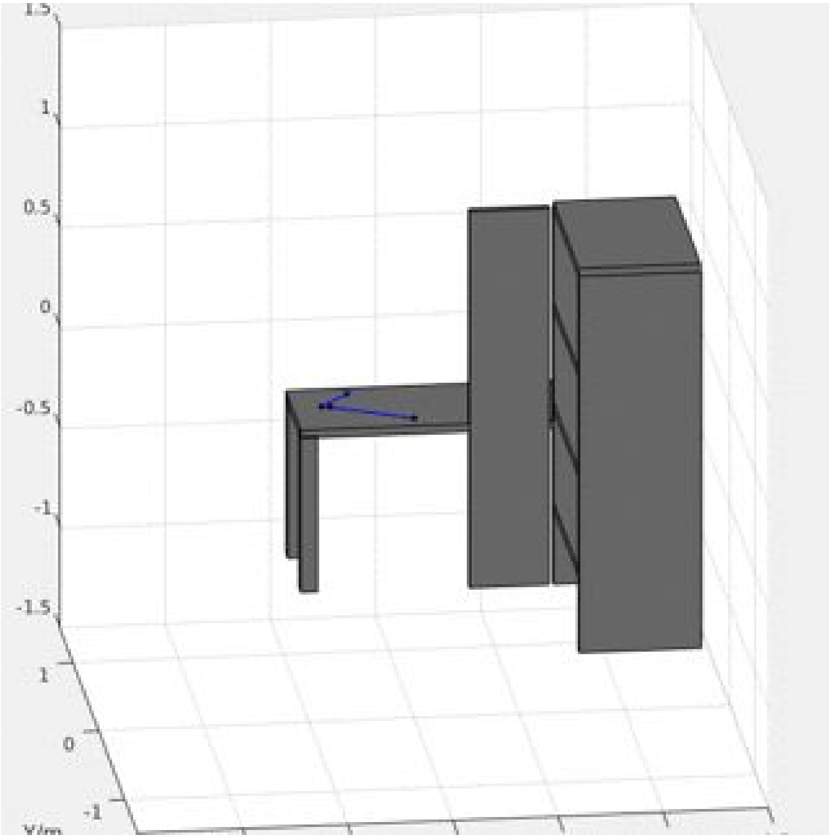
Table II.B Results for 28 replanning problems on PR2.

	iGPMP2	GPMP2
Success Rate (%)	75.0	**96.4**
Average Time to Success (ms)	**4.27**	26.70
Maximum Time to Success (ms)	**6.67**	58.84

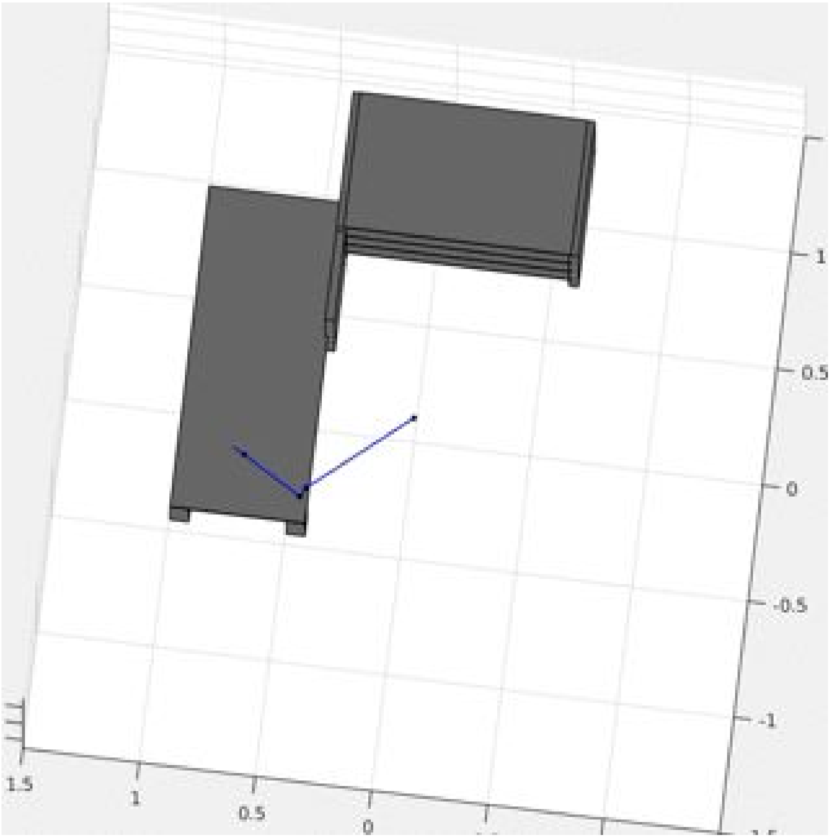
Dong, Jing, et al. "Motion Planning as Probabilistic Inference using Gaussian Processes and Factor Graphs." *Robotics: Science and Systems (RSS)* (2016).

Evaluation: Replanning

- Dataset: 7-DOF WAM arm and 7-DOF PR2, total 60 problems



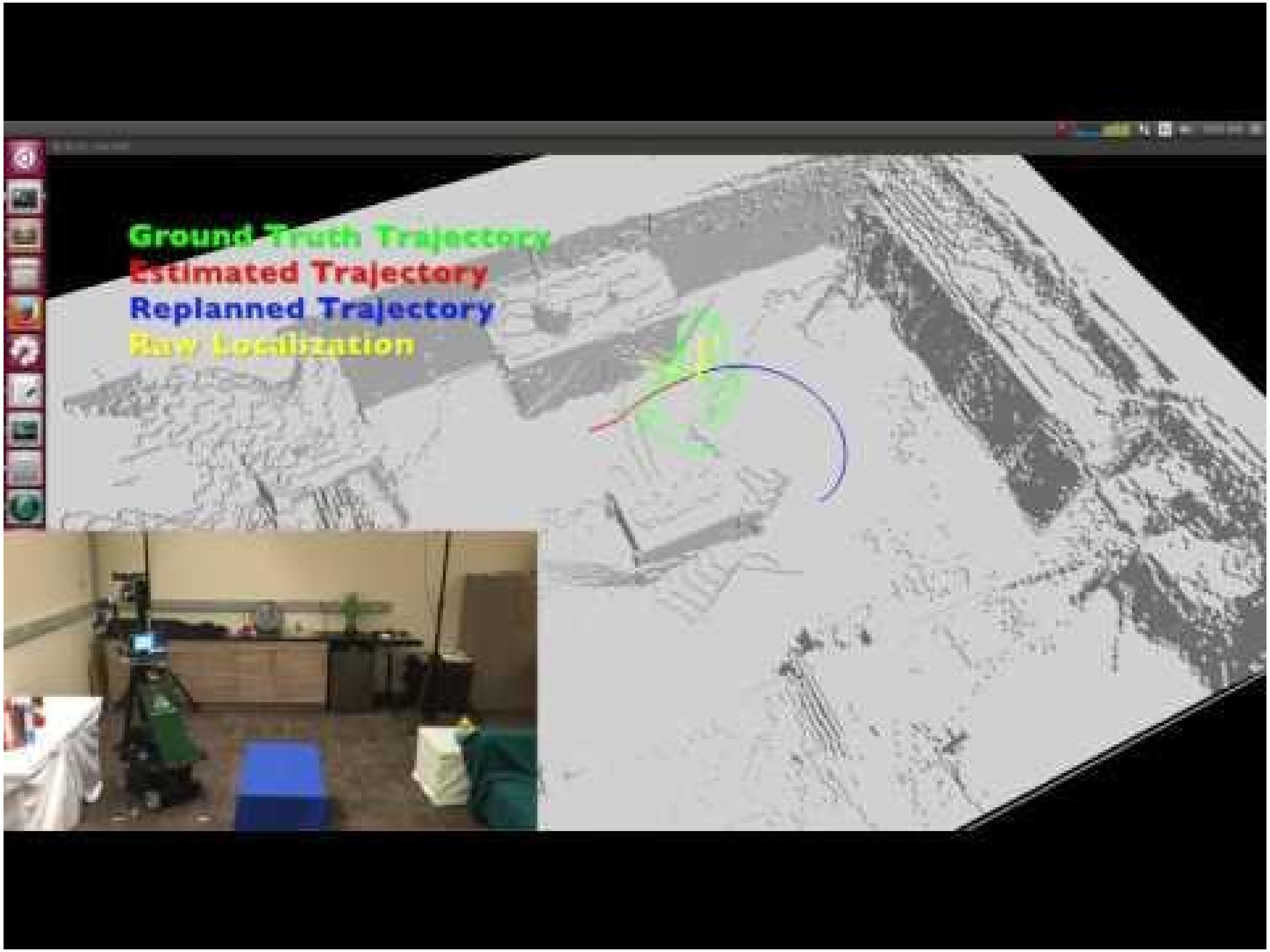
https://github.com/gtrll/gpmp2/blob/master/doc/pics/wam_replan_2.gif



https://github.com/gtrll/gpmp2/blob/master/doc/pics/wam_replan_1.gif

Dong, Jing, et al. "Motion Planning as Probabilistic Inference using Gaussian Processes and Factor Graphs." *Robotics: Science and Systems (RSS)* (2016).

Real-time Replanning on Robots



https://youtu.be/lyayNKV1eAQ

Mukadam, Mustafa, et al. "Simultaneous Trajectory Estimation and Planning via Probabilistic Inference." *Robotics: Science and Systems (RSS)* (2017).

Thanks!