Gaussian Belief Propagation for Continuous-Time SLAM

Hojune Kim, Semester Project
BSc. D-MAVT exchange student
30th July, 2024

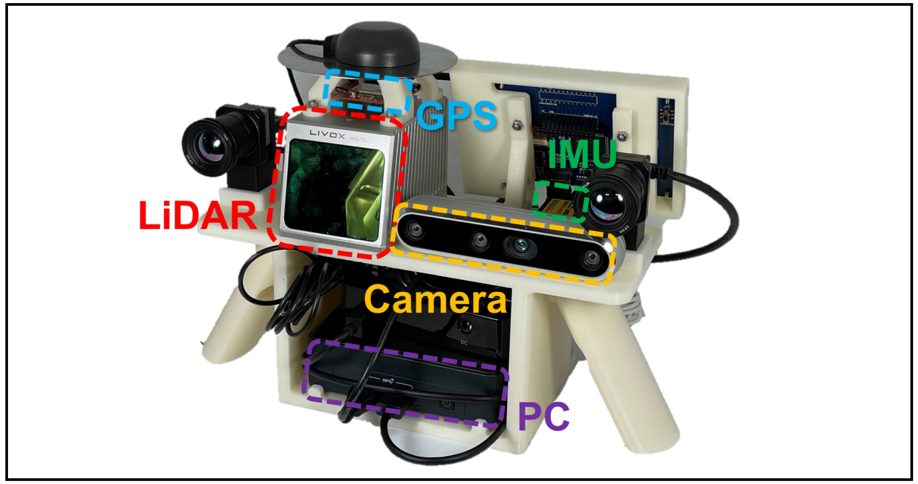
Supervisors: David Hug, Cornelius von Einem
Prof. Margarita Chli

ETH zürich

V4RL
VISION FOR ROBOTICS LAB

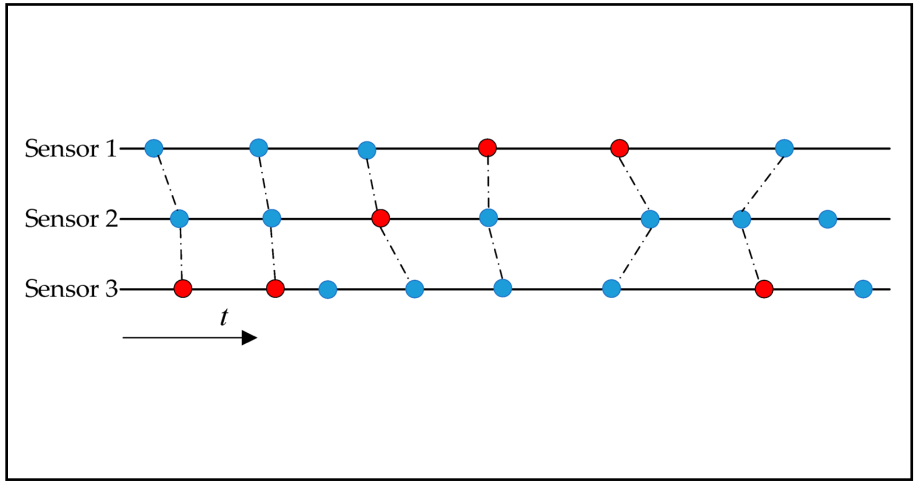
Motivation

Multi Sensor Setup



Collaborative Perception in Nature

Unsynchronized Data

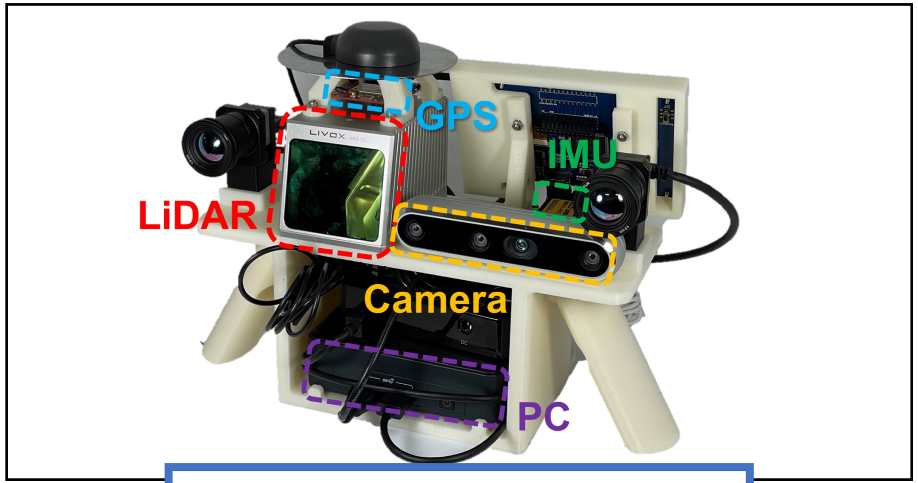


Collaborative Perception in Robotics

Lee, Dongjae, Minwoo Jung, and Ayoung Kim. "ConPR: Ongoing construction site dataset for place recognition." *arXiv preprint arXiv:2407.03684* (2024).

Motivation

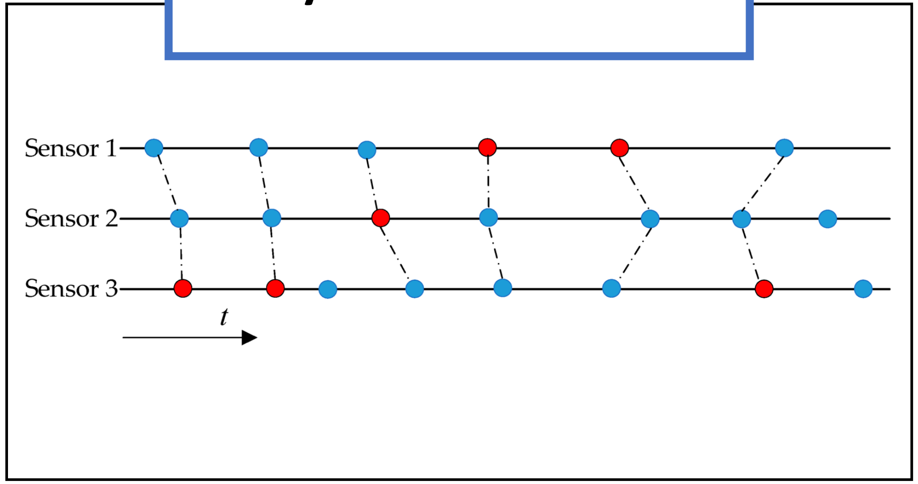
Multi Sensor Setup



Collaborative Perception in Nature

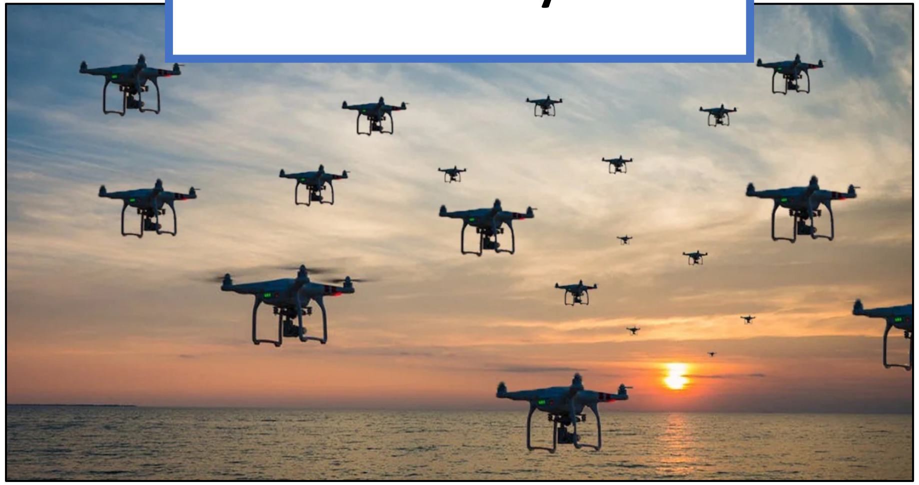
Asynchronous Data

zed Data



Distributed System

Robotics



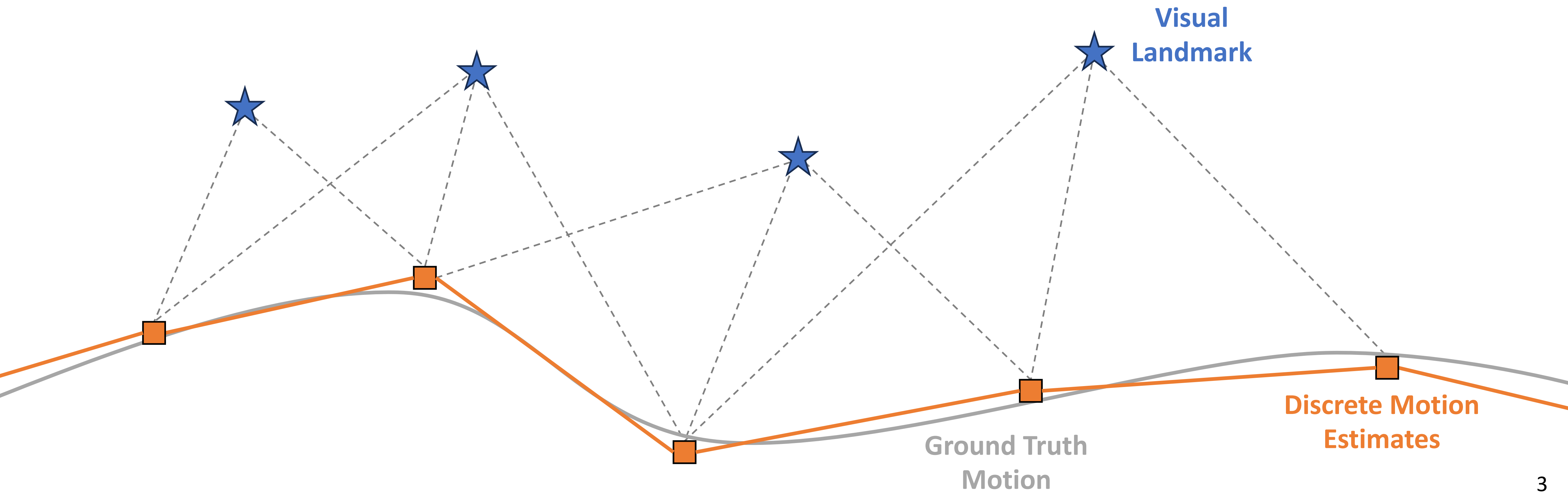
Lee, Dongjae, Minwoo Jung, and Ayoung Kim. "ConPR: Ongoing construction site dataset for place recognition." *arXiv preprint arXiv:2407.03684* (2024).

Existing Challenges

Discrete Time SLAM(Simultaneous Localization and Mapping)

Conventional consolidated theory

Assume **synchronized inputs** and **steady sensor acquisition rates**



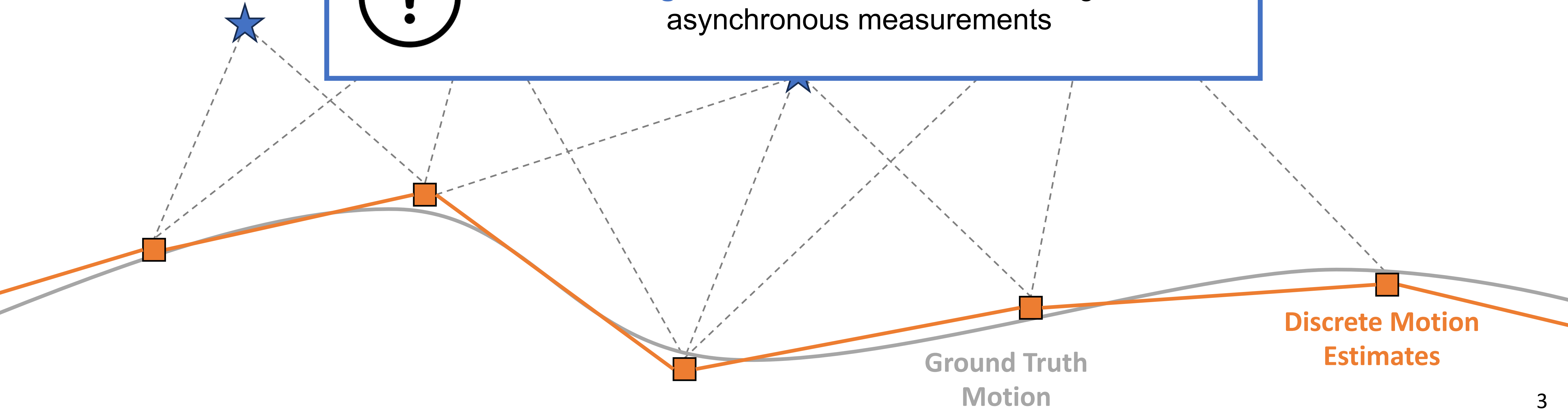
Existing Challenges

Discrete Time SLAM(Simultaneous Localization and Mapping)

Conventional consolidated theory

Assume **synchronized inputs** and **steady sensor acquisition rates**

Tailored algorithms needed to deal with high rate or asynchronous measurements



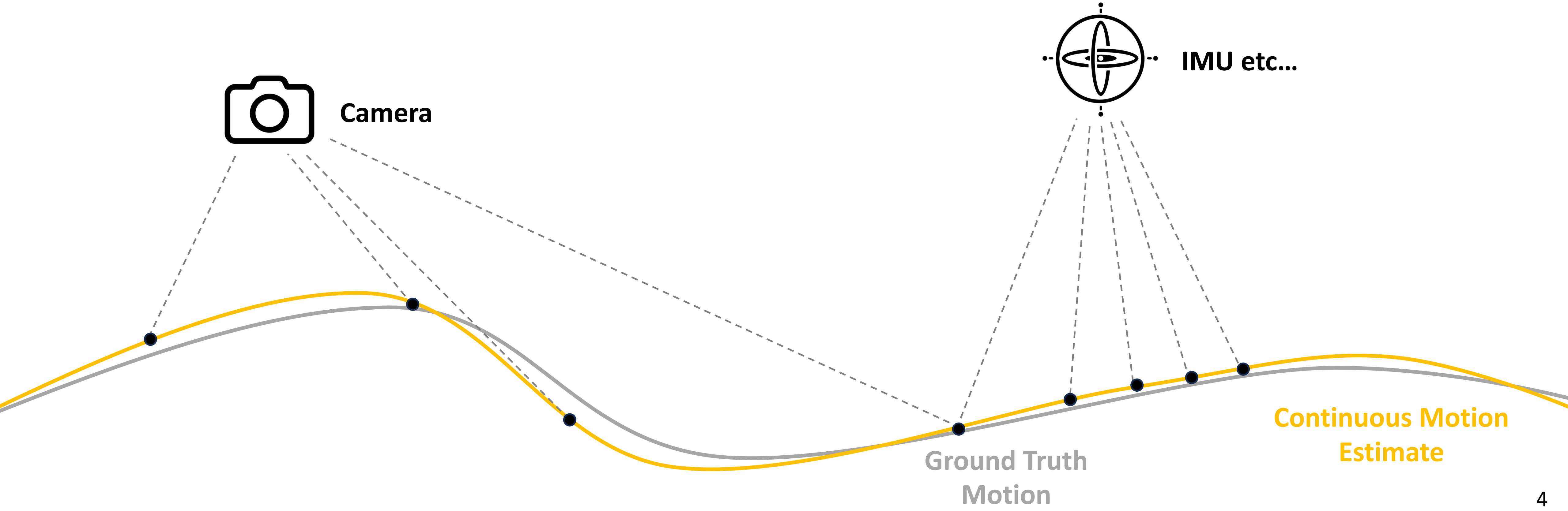
Existing Approach

Continuous Time SLAM(Simultaneous Localization and Mapping)

Intuitive fusion of **asynchronous** or continuous **high-rate sensor data**

No need of adding a new optimization variable for each new measurement

Provide **analytic derivatives**

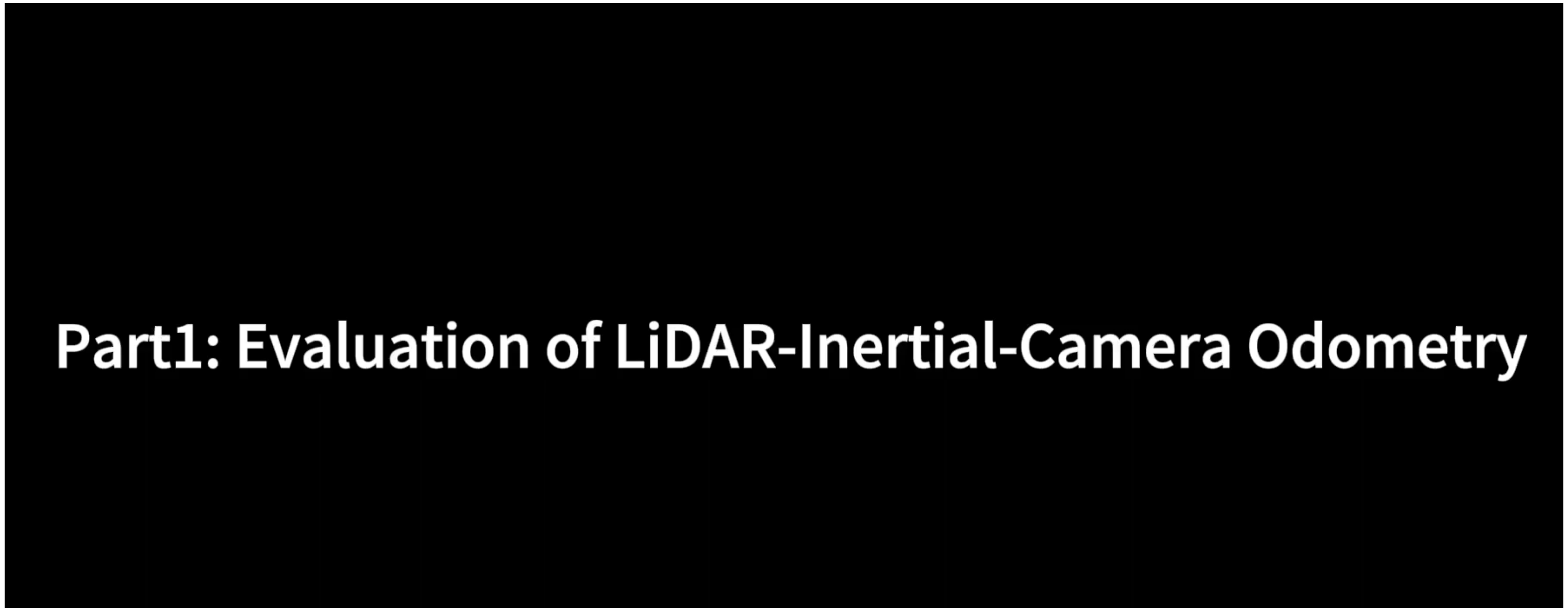


Existing Challenges

Non-linear Least Squares(NLLS) Solver

Many sensor fusion algorithm developed in continuous time

However, most of the methods utilize **NLLS(Non-linear Least Squares)** optimizer which is a **centralized solver**



Lang, Xiaolei, et al. "Coco-LIC: continuous-time tightly-coupled LiDAR-inertial-camera odometry using non-uniform B-spline." *IEEE Robotics and Automation Letters* (2023).

Existing Challenges

Non-linear Least Squares(NLLS) Solver

Many sensor fusion algorithm developed in continuous time

However, most of the methods utilize **NLLS(Non-linear Least Squares)** optimizer which is a **centralized solver**

Mathematical expressions for spline-related residuals are often **complex and prone to mistakes**, which can lead to **suboptimal performance**

Standard NLLS optimizers **do not model uncertainties** and **do not easily extend to distributed computations** across multiple agents.

Lang, Xiaolei, et al. "Coco-LIC: continuous-time tightly-coupled LiDAR-inertial-camera odometry using non-uniform B-spline." *IEEE Robotics and Automation Letters* (2023).

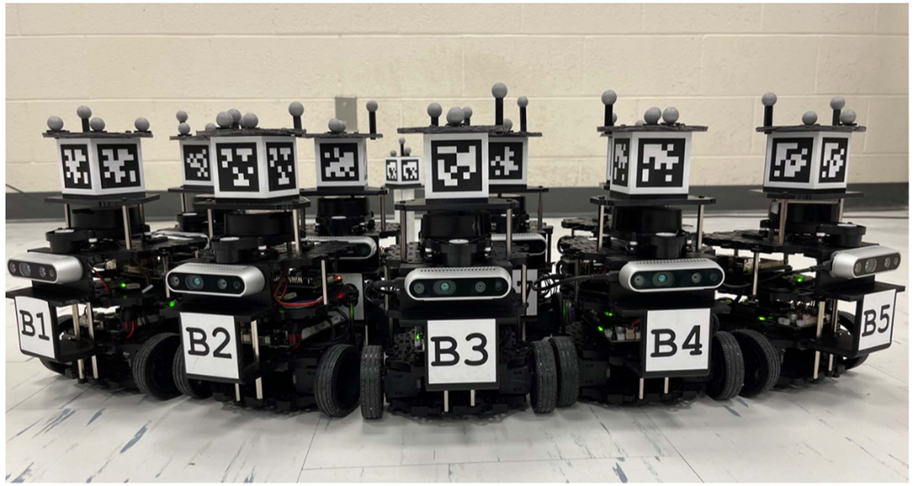
Existing Approach

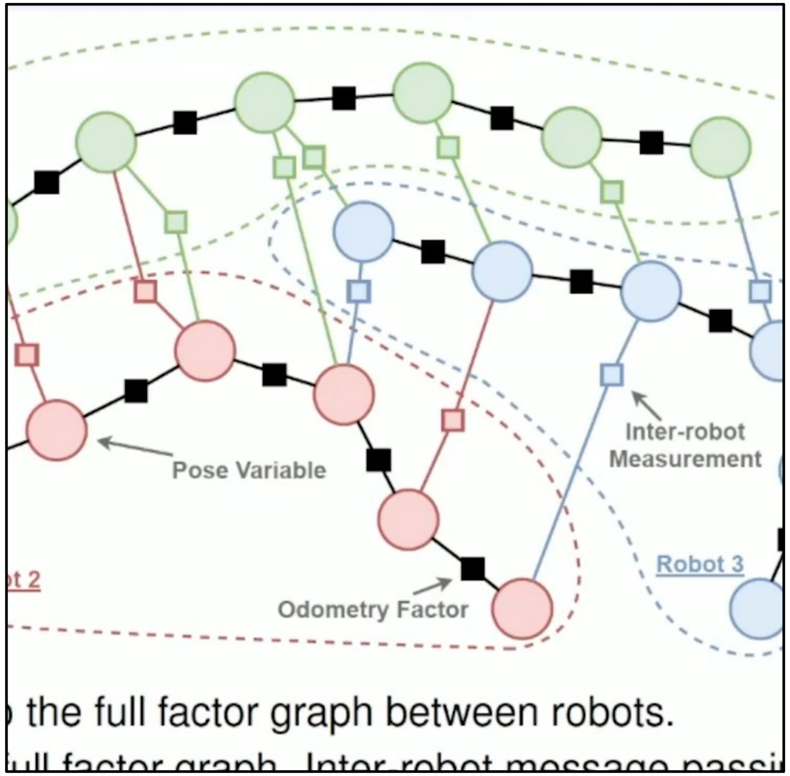
Message Passing(Belief Propagation)

Robot web for distributed many-device localization by Gaussian Belief Propagation

Decentralized solver by **propagating the belief** through the **message passing iteratively**

No need of adding a new optimization variable for each new measurement

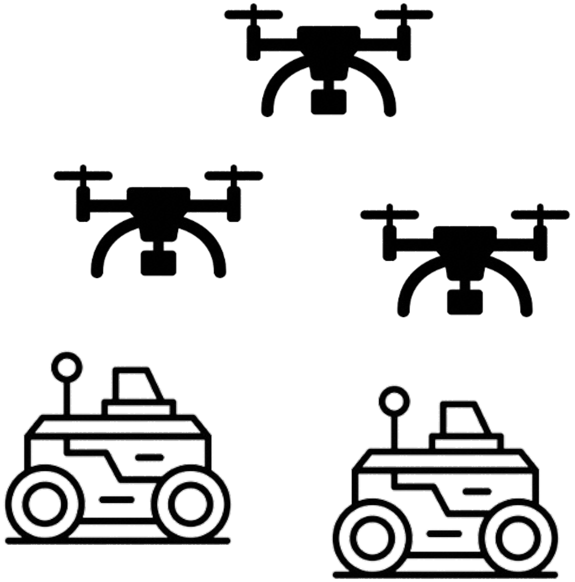




Murai, Riku, et al. "A robot web for distributed many-device localisation." *IEEE Transactions on Robotics* (2023).

Our Method

Continuous Time Gaussian Belief Propagation(GBP)



Distributed System

Various sensor setup

Inter-robot measurement

> Asynchronous data naturally occurs

> Efficient distributed computation is needed

Continuous Time SLAM

+

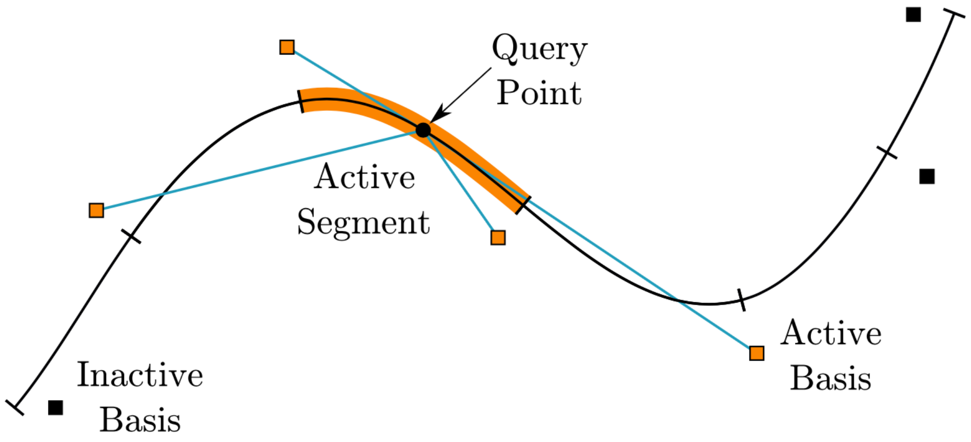
Gaussian Belief Propagation

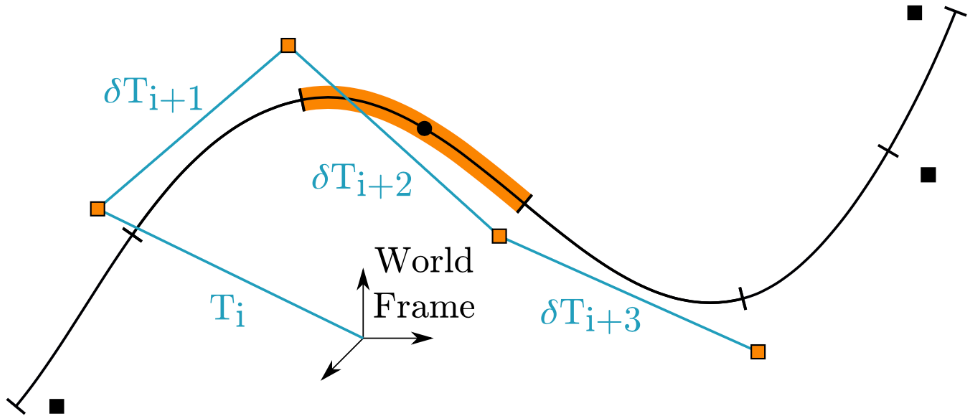
Our Method

Continuous Time Parameterization

Each **spline segments** is associated with multiple basis **control points**

Transformation of the query point is computed by **combining the increments** between control points



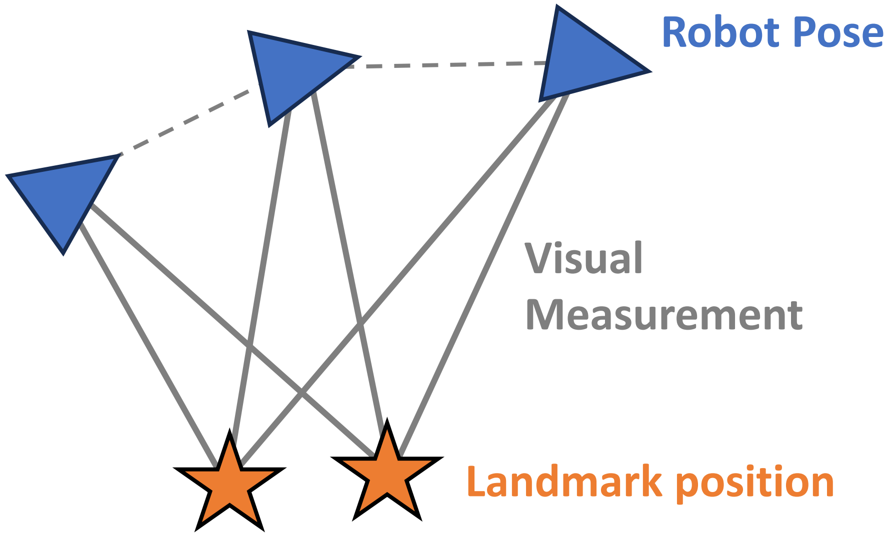


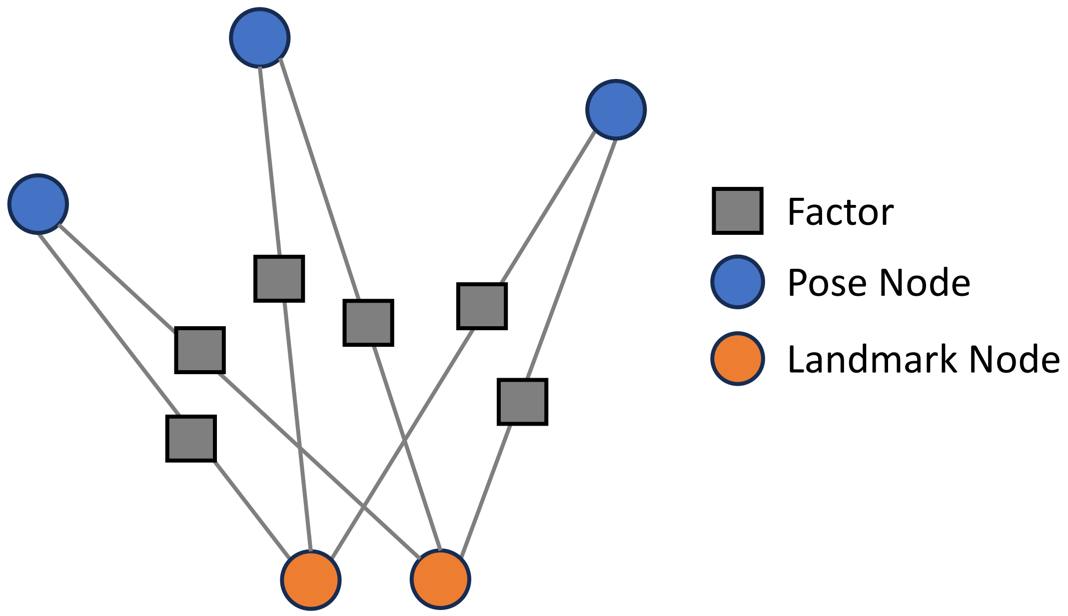
Hug, David, et al. "Continuous-time stereo-inertial odometry." *IEEE Robotics and Automation Letters* 7.3 (2022): 6455-6462.

Our Method

Factor Graph

SLAM problem can be represented with **Factor Graph**, consisted by **node** and **factor**





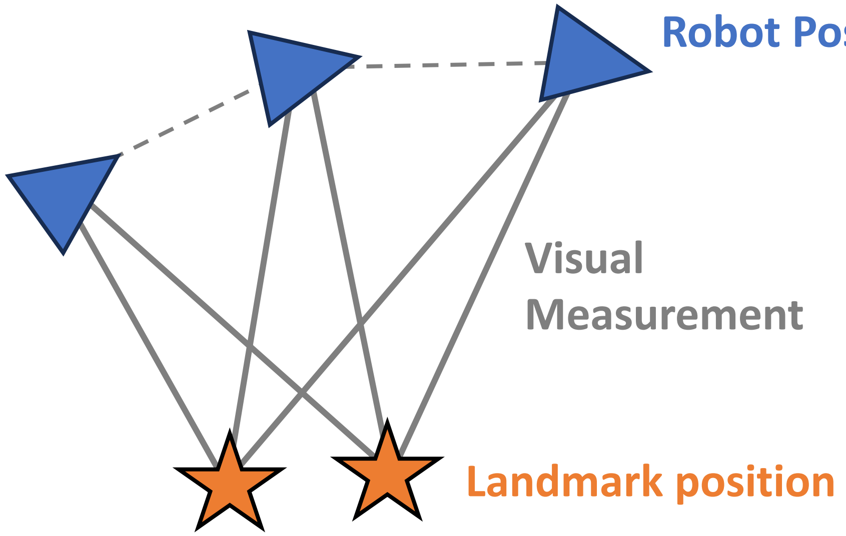
Factro Graph

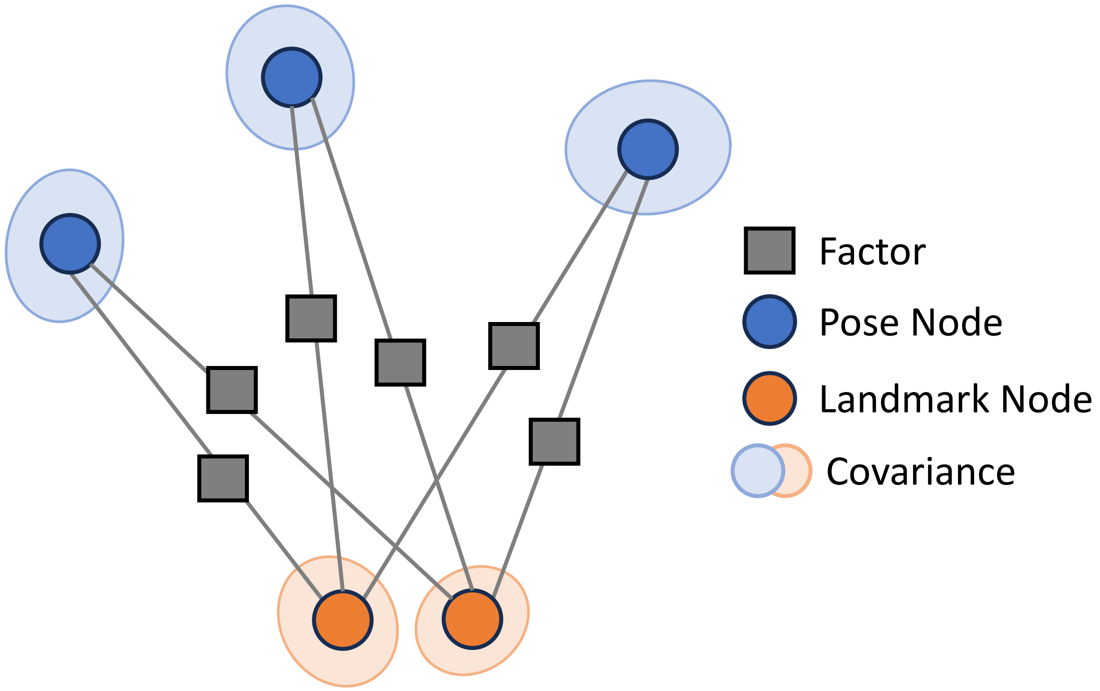
Our Method

Factor Graph

SLAM problem can be represented with **Factor Graph**, consisted by **node** and **factor**

Stochastic factor graph by considering the **covariance** of the nodes





Factro Graph

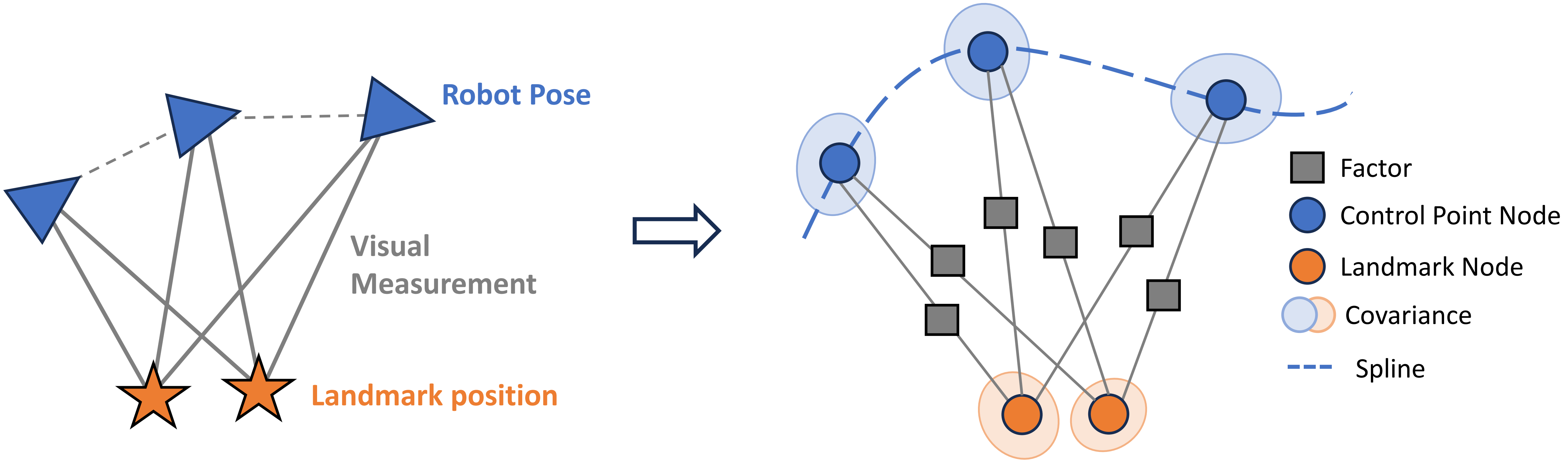
Our Method

Factor Graph

SLAM problem can be represented with **Factor Graph**, consisted by **node** and **factor**

Stochastic factor graph by considering the **covariance** of the nodes

Derive **spline** with control points in **continuous time**



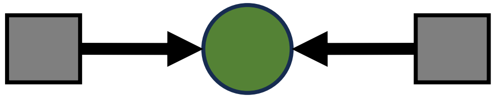
Factro Graph

Our Method

Gaussian Belief Propagation

Canonical form of Gaussian distribution: $\mathcal{N}(\mu, \Sigma) = \mathcal{N}^{-1}(\eta, \Lambda)$ $\eta = \Sigma^{-1}\mu,$ $\Lambda = \Sigma^{-1}$

1. Node Update



$$\eta_{n_j} = \sum_{f_i \in N(n_j)} \eta_{f_i \to n_j}$$

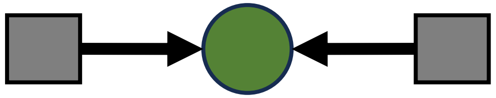
How does my neighboring factors believe about me

Our Method

Gaussian Belief Propagation

Canonical form of Gaussian distribution: $\mathcal{N}(\mu, \Sigma) = \mathcal{N}^{-1}(\eta, \Lambda)$ $\eta = \Sigma^{-1}\mu,$ $\Lambda = \Sigma^{-1}$

1. Node Update

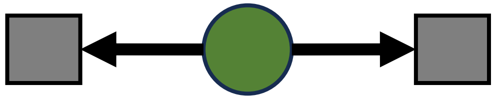


$$\eta_{n_j} = \sum_{f_i \in N(n_j)} \eta_{f_i \to n_j}$$

How does my neighboring factors believe about me

$\Downarrow$

2. Node-to-Factor Message



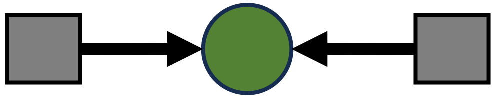
Propagate the updated belief to neighboring factors

Our Method

Gaussian Belief Propagation

Canonical form of Gaussian distribution: $\mathcal{N}(\mu, \Sigma) = \mathcal{N}^{-1}(\eta, \Lambda)$ $\eta = \Sigma^{-1}\mu,$ $\Lambda = \Sigma^{-1}$

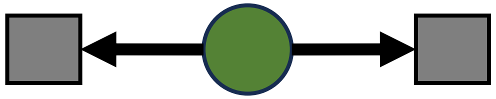
1. Node Update



$$\eta_{n_j} = \sum_{f_i \in N(n_j)} \eta_{f_i \to n_j}$$

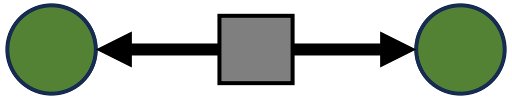
How does my neighboring factors believe about me

2. Node-to-Factor Message



Propagate the updated belief to neighboring factors

3. Factor-to-Node Message



$$\eta'_{f_i \to n_a} = \eta_a^0 - \mathbf{\Lambda}'^{\top}_{aa} \mathbf{\Lambda}'^{-1}_{bb} \eta'_b$$

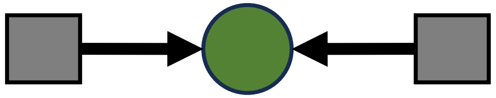
Marginalize the probability distribution for neighboring nodes

Our Method

Gaussian Belief Propagation

Canonical form of Gaussian distribution: $\mathcal{N}(\mu, \Sigma) = \mathcal{N}^{-1}(\eta, \Lambda)$ $\quad \eta = \Sigma^{-1}\mu, \quad \Lambda = \Sigma^{-1}$

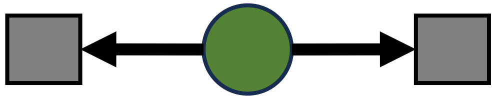
1. Node Update



$$\eta_{n_j} = \sum_{f_i \in N(n_j)} \eta_{f_i \to n_j}$$

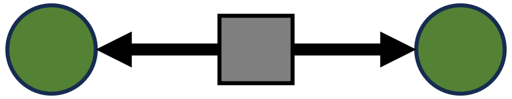
How does my neighboring factors believe about me

2. Node-to-Factor Message



Propagate the updated belief to neighboring factors

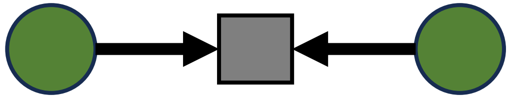
3. Factor-to-Node Message



$$\eta'_{f_i \to n_a} = \eta_a^0 - \boldsymbol{\Lambda}'^{\top}_{aa} \boldsymbol{\Lambda}'^{-1}_{bb} \eta'_b$$

Marginalize the probability distribution for neighboring nodes

4. Factor Update



$$\eta_i^0 = -\bar{\mathbf{J}}_i^{0,\top} \bar{\mathbf{r}}_i^0$$

Evaluate the residual and Jacobian through the cost function

Project Goal

Gaussian Belief Propagation for Continuous time SLAM

=> “**Hyperion**” beta version has developed

Verified with SE(3) poses, but **not with the visual features**

First step : **fix the bugs** for the visual feature nodes

Practical Goal : **verify in real world** dataset => **Apriltag Grid**

Hug, David, Ignacio Alzugaray, and Margarita Chli. "Hyperion-A fast, versatile symbolic Gaussian Belief Propagation framework for Continuous-Time SLAM." *arXiv preprint arXiv:2407.07074*(2024).

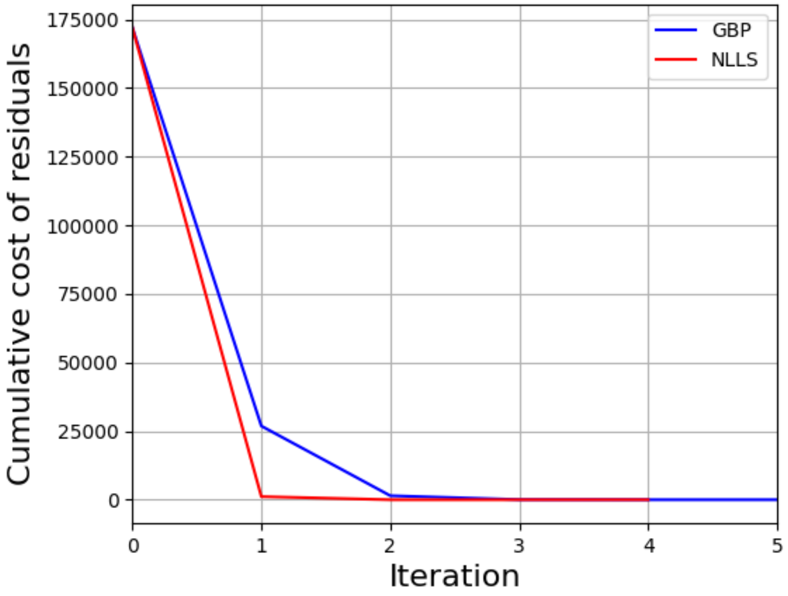
Verify in Simulation

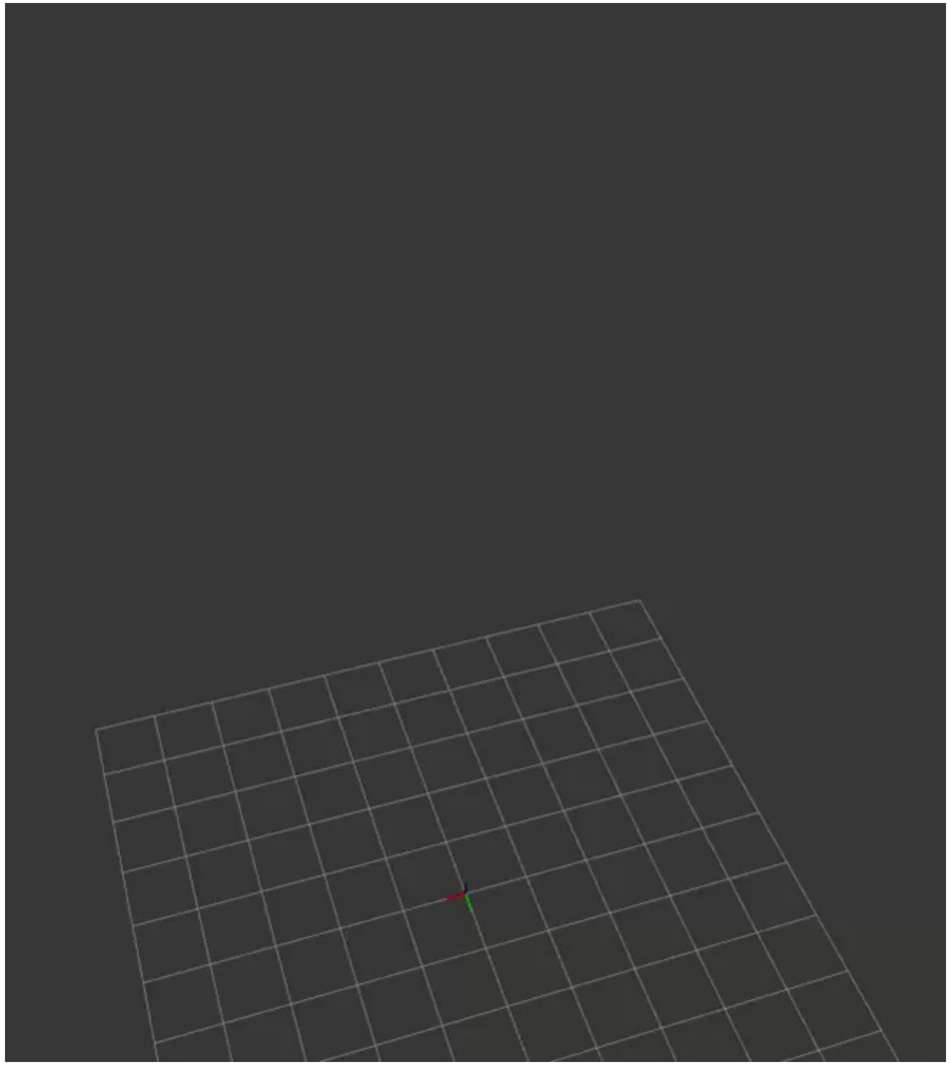
Debugging... Fixed Bugs

Verified visual feature nodes by simulating the simple triangulation scenario

Perturbed pose and landmark converges to the GT

GBP is slightly slower than NLLS due to **stochastic inference with covariance**





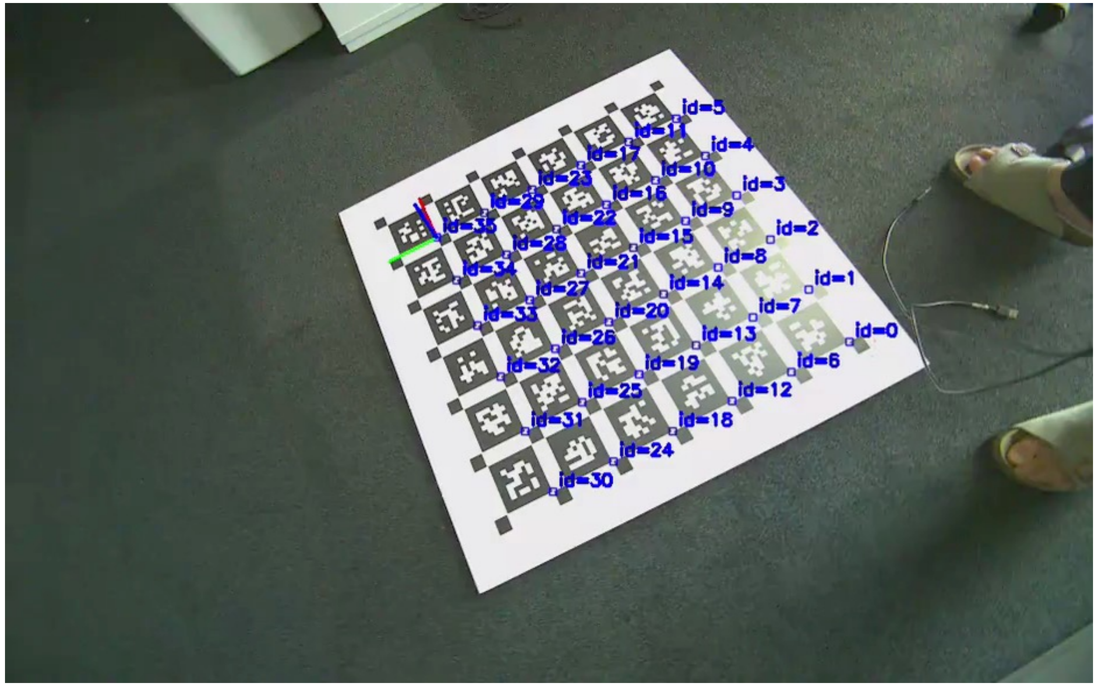
Dataset

Configuration

6x6 Apriltag Grid Stereo Camera Dataset

GT pose : extracted by **PnP(Perspective-n-Point)** from corner detections

GT Landmarks : extracted by **bundle adjustment** from GT pose



Dataset

Problem Setup and Metrics

Problem setup

Fix first stereo frame as a reference frame

Optimize other **disturbed pose and landmark** node with **only stereo images**

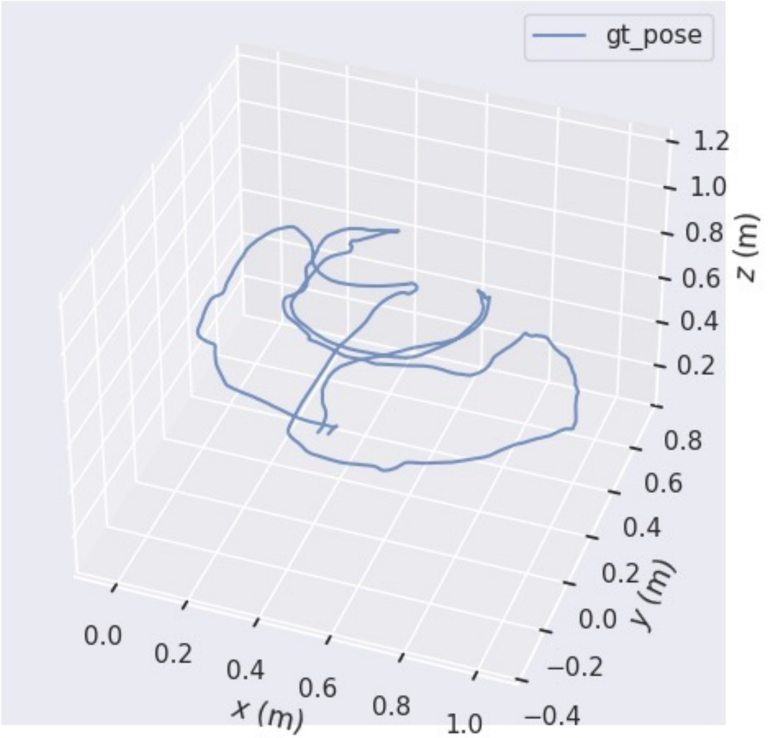
Time : Discrete vs Continuous

Solver : GBP(using Hyperion) vs NLLS(using Ceres)

Evaluation Metric

Landmark : root mean square error in translation

Pose : root mean square error in rotation and translation



Result

Discrete Time – Residual Weight

Residual of the factor is **weighted** by **square-root information matrix**($\boldsymbol{\Omega}_m$)

Bias & scale error occurred with the equal weight

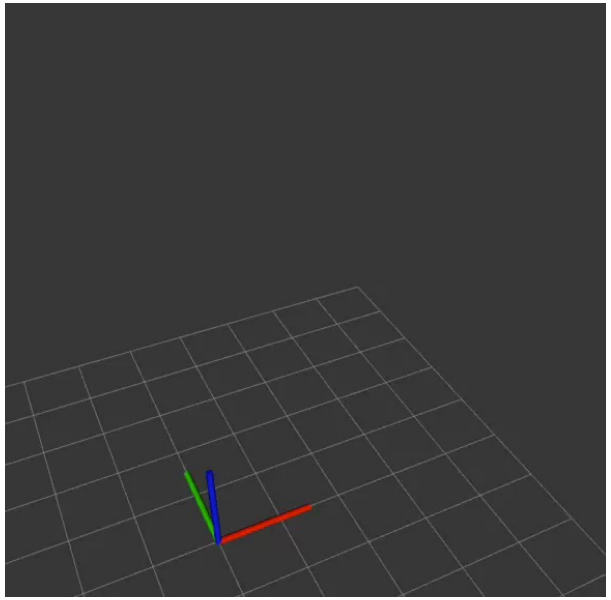
Become stable after reducing the weight except the first frame factors to **focus on the reference frame**

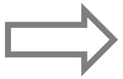
$$\boldsymbol{r}(t,\boldsymbol{\theta}_s) = \widehat{\boldsymbol{m}}(t,\boldsymbol{\theta}_s) \boxminus_{\boldsymbol{\mu}} \boldsymbol{m}(t)$$

$$\|\boxed{\bar{\boldsymbol{r}}}\|^2 = \bar{\boldsymbol{r}}^T\bar{\boldsymbol{r}} = \boldsymbol{r}^T\boldsymbol{\Omega}_m^T\boxed{\boldsymbol{\Omega}_m}\boldsymbol{r}$$

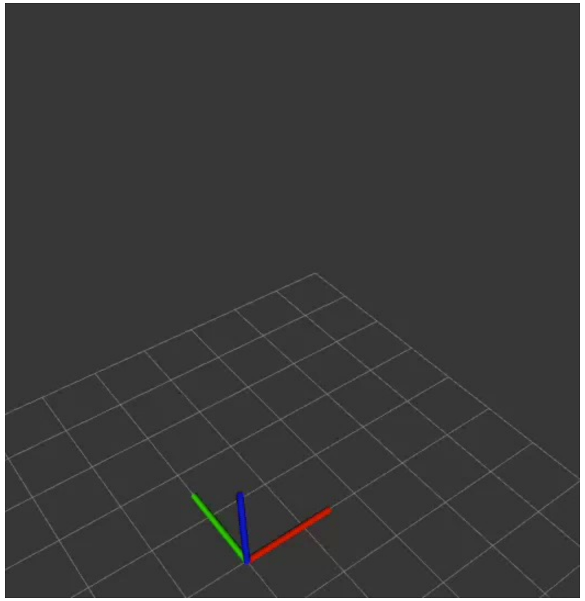
weighted residual   **square-root information matrix**

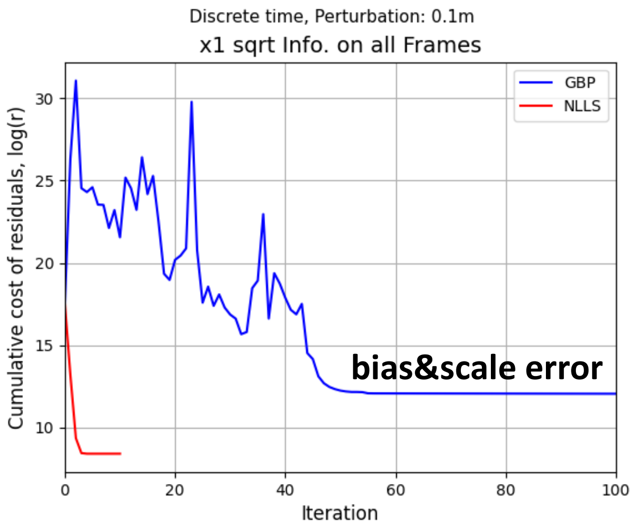
Identity $\boldsymbol{\Omega}_m$ for all factor residuals

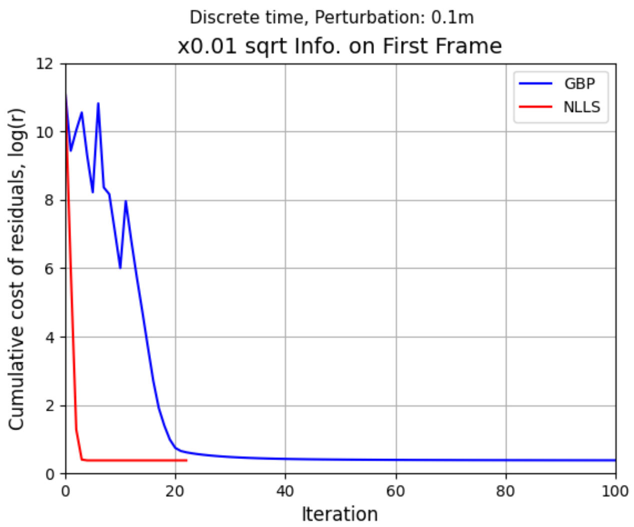




$0.01\times\boldsymbol{\Omega}_m$ except the factors of first frame







*Video Legend
Blue: GBP
Red: NLLS(optimized)
Green: GT
Arrow: Pose
Square: Landmark

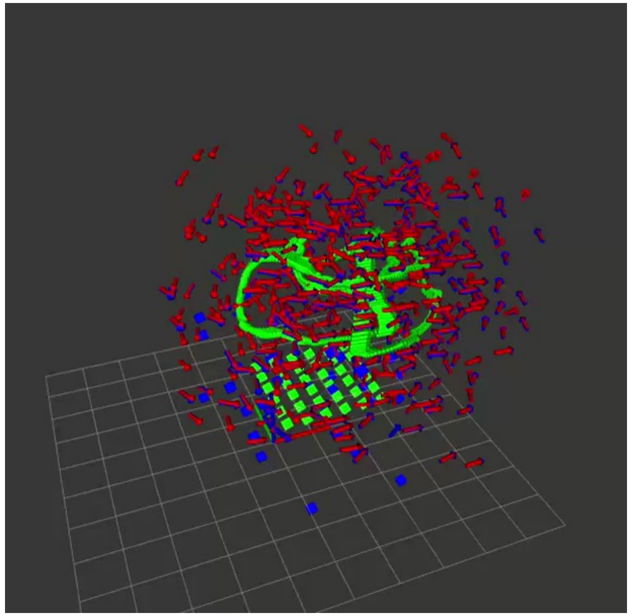
Result

Discrete Time – Perturbation

Tested various **perturbed initialization setup**

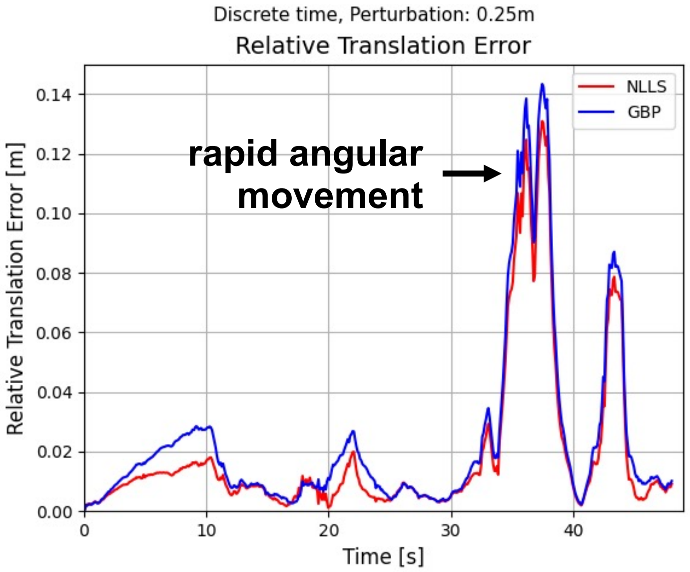
GBP showed more robustness than NLLS, converging with up to 0.5 m/rad noise on the pose and landmarks

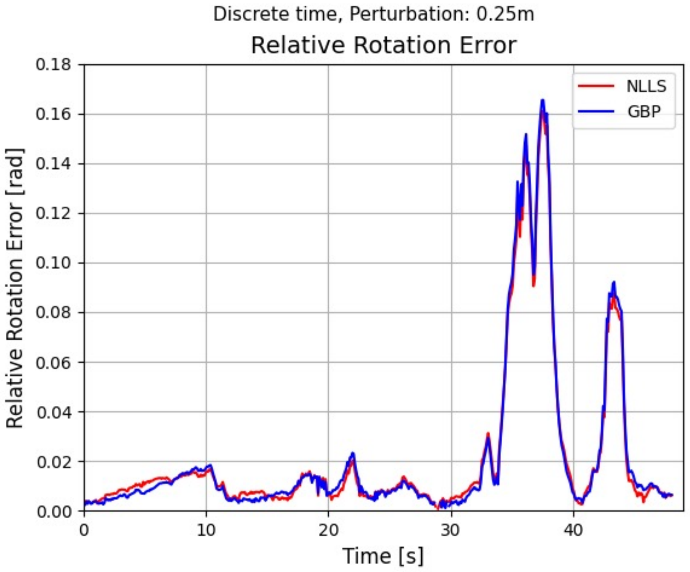
	Perturbation [m/rad*]		
	0.1	0.25	0.5
NLLS landmark RMSE [mm]	2.79	2.79	**X (Diverged)**
GBP landmark RMSE [mm]	2.95	8.11	**8.62**



*Video Legend
Blue: GBP
Red: NLLS(optimized)
Green: GT
Arrow: Pose
Square: Landmark

Perturbed pose($\pm 0.5\mathrm{m/rad}$) and landmark($\pm 0.25\mathrm{m}$) initialization





*ex) Perturbation 0.1(pose ± 0.1 m, ± 0.1 rad, landmark ± 0.1 m)

Result

Discrete Time – Initial Covariance

Tested various **initial covariance setup** on GBP nodes

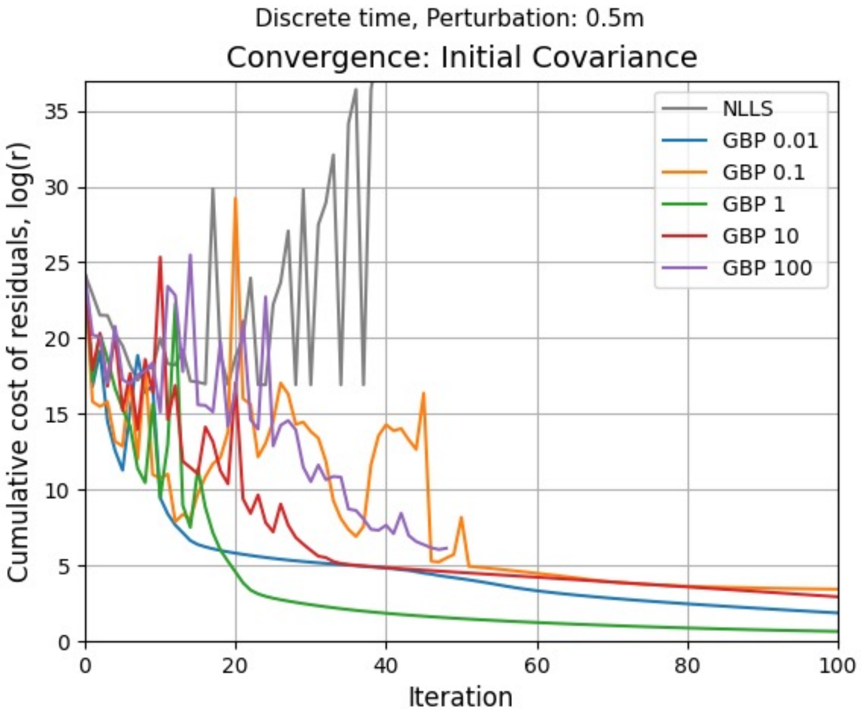
Initial covariance affects the **rate of convergence** and **stability**

Depending on the initial covariance, the result might fall into the suboptimal solution

Perturbation 0.5 [m/rad]

Initial Covariance Coefficient

	0.01	0.1	1	10	100
GBP landmark RMSE [mm]	22.4	20.6	**8.62**	42.0	84.0



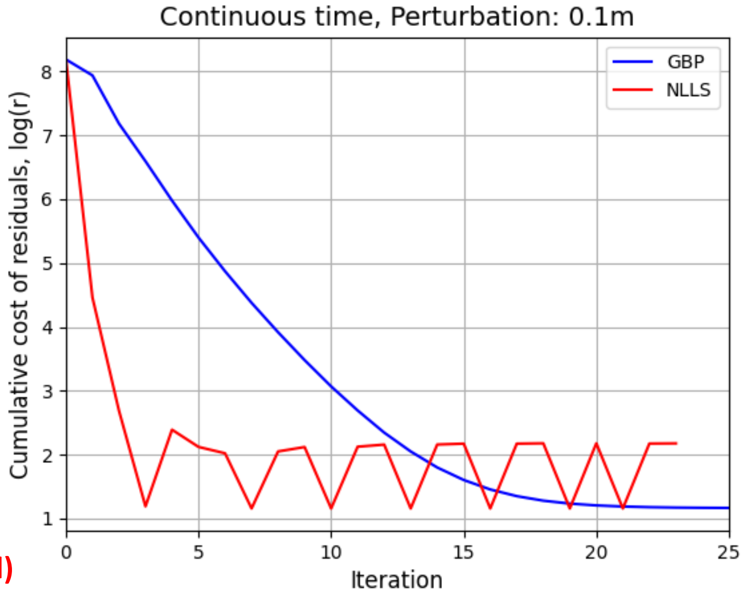
Result

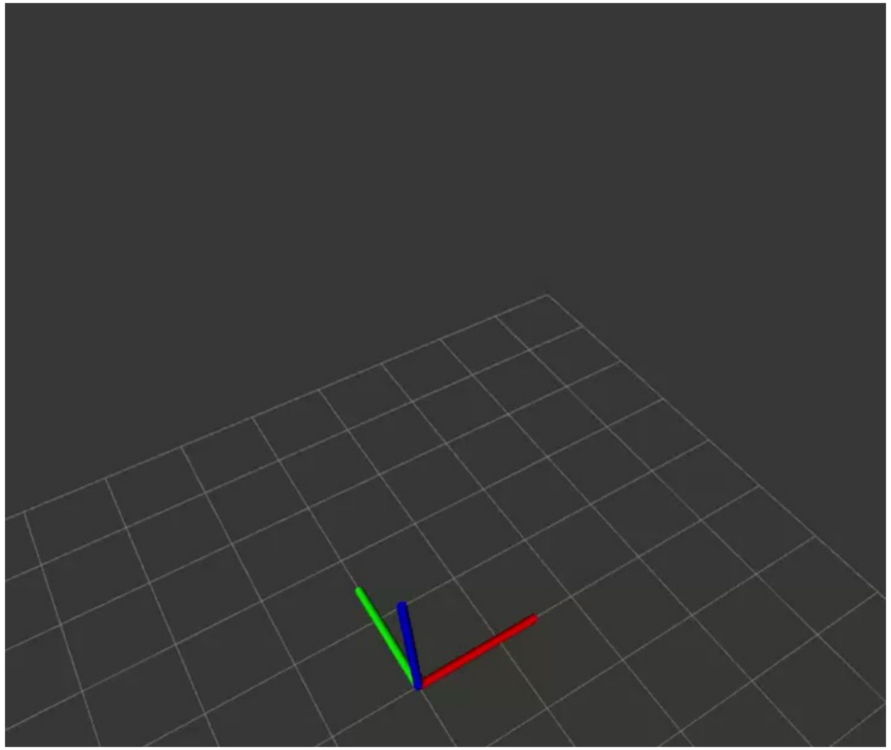
Continuous Time

4 Degree of Z-Spline, set control points to half of the data frames

Consumes more time to evaluate than discrete time, since **factor-node connection is quadrupled**
(4 control points required to represent the pose)

Diverges with perturbation grater than 0.1 m/rad In both NLLS and GBP





*Video Legend
Blue: GBP
Red: NLLS(optimized)
Green: GT
Arrow: Pose
Square: Landmark

Perturbation 0.1 [m/rad]

	Landmark RMSE [m]	Pose translation RMSE [m]	Pose rotation RMSE [rad]
NLLS	2.22 e-3	**0.038**	**0.040**
GBP	**1.79 e-3**	0.045	0.043

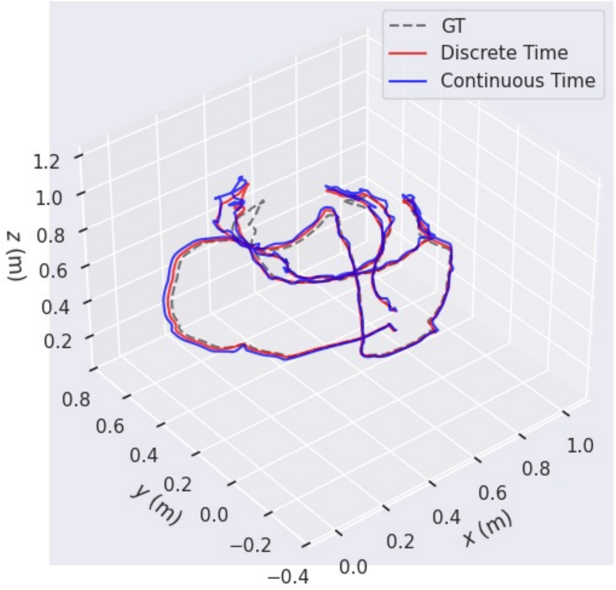
Result

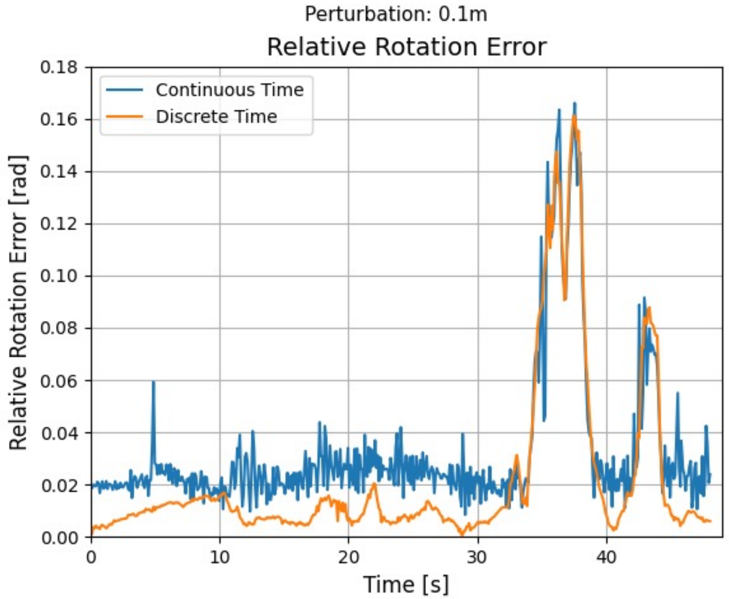
Discrete Time vs Continuous Time

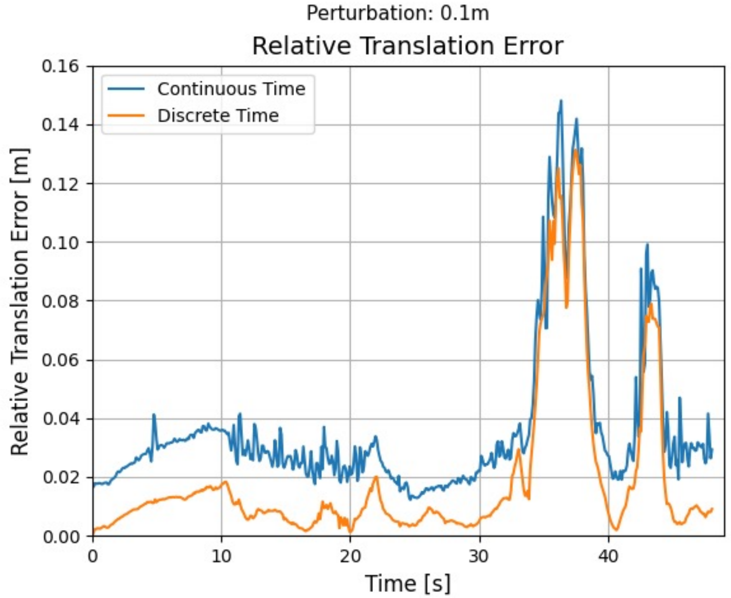
CT+GBP and DT+GBP showed **similar optimized solution in small perturbation**

Discrete Time representation was more robust to perturbation in **synchronized data**

	Perturbation 0.1 [m/rad]		
	Landmark RMSE [m]	Pose translation RMSE [m]	Pose rotation RMSE [rad]
DT+GBP	2.95 e-3	**0.034**	**0.040**
CT+GBP	**1.79 e-3**	0.045	0.043







Conclusion / Outlook

- Gaussian belief propagation(GBP) has been successfully **verified for both continuous and discrete time** batch optimization
- **GBP demonstrates superior robustness** against perturbations compared to the non-linear least square solver in **discrete time**

- **Hyperparameter** : The rate of convergence of GBP depends on the parameters
  -> implement adaptive parameter strategy
- **Practicality** : Is GBP+CT promising for multi-sensor SLAM?
  -> evaluate in benchmark datasets
- **Time consumption**: How about in large graph problems?
  -> propagate belief based on their covariance

Q&A

Thank you!