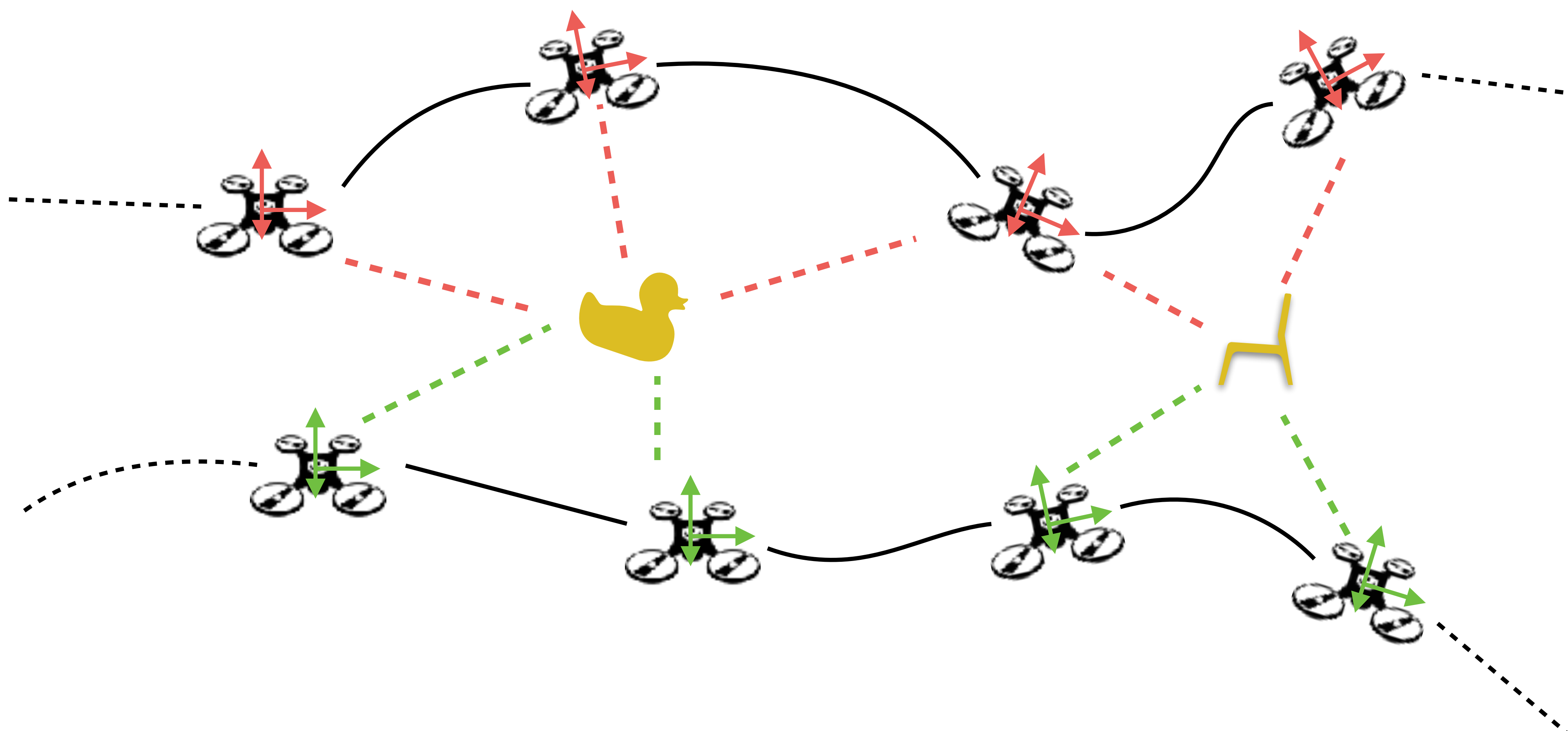
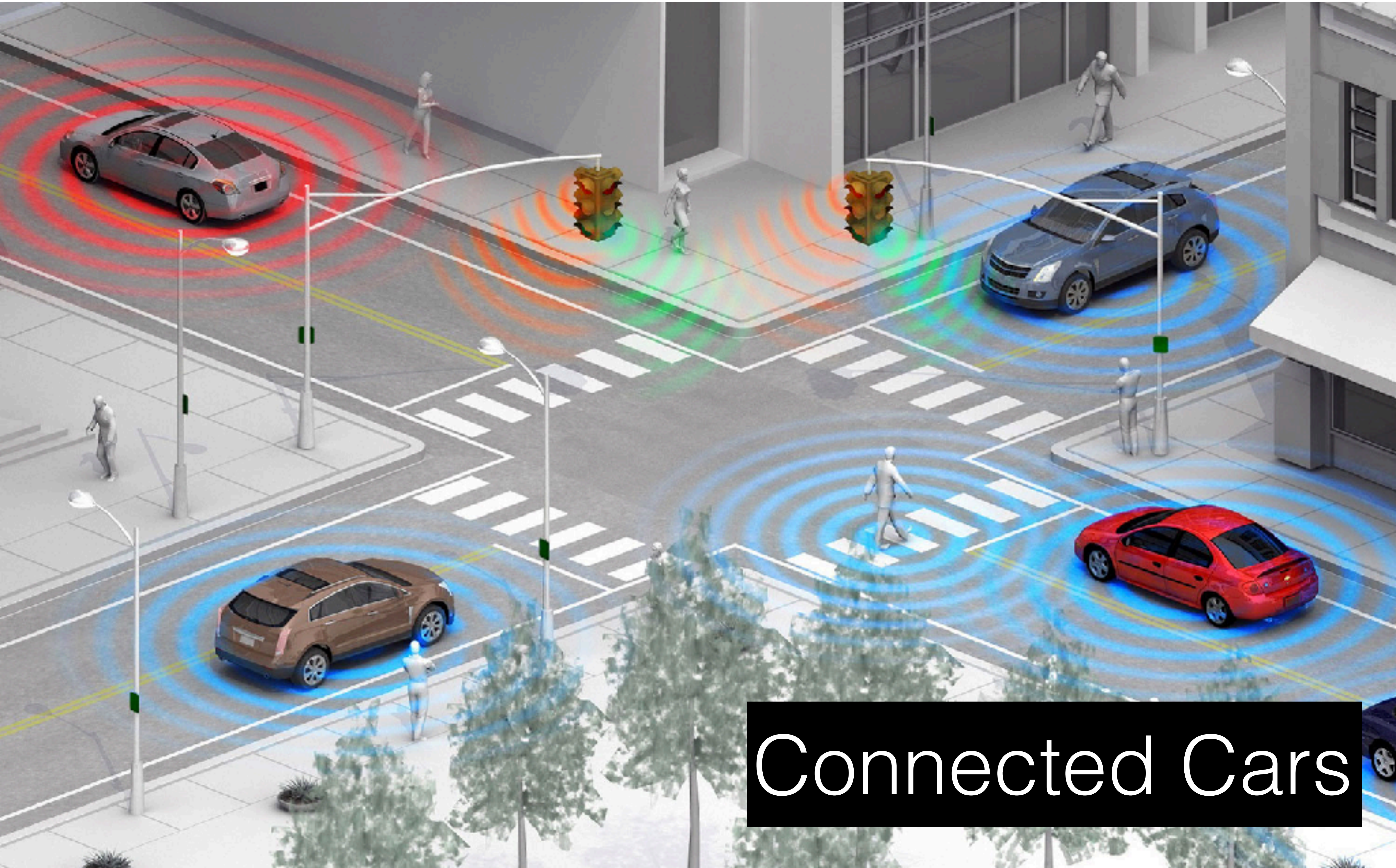
Distributed Object based SLAM

Siddharth Choudhary



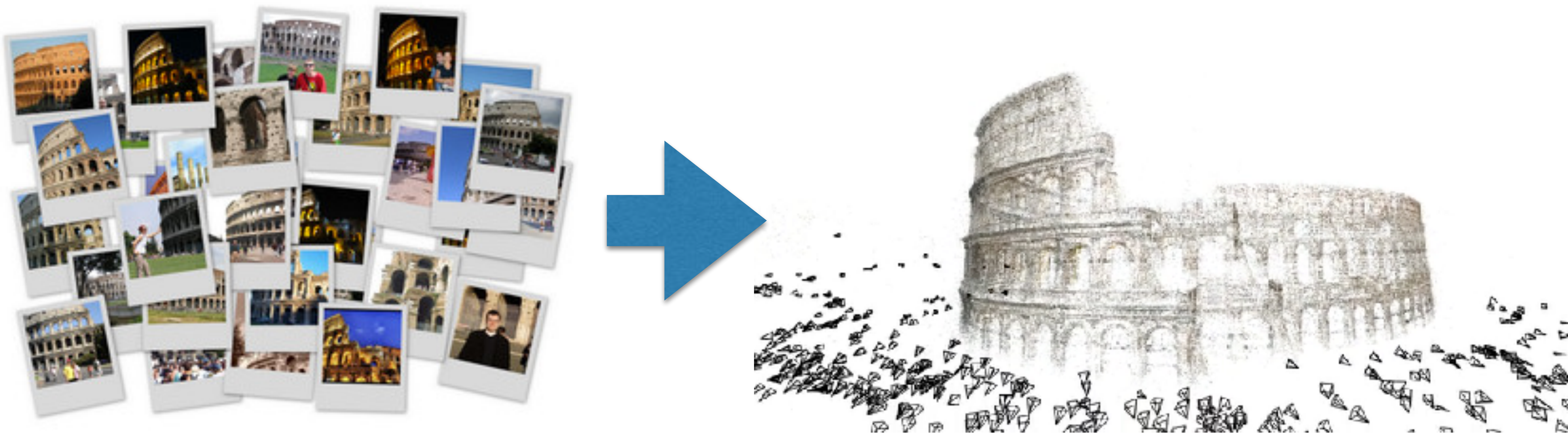
Committee Members:
Prof. Henrik I. Christensen (Advisor)
Prof. Frank Dellaert (Co-Advisor)
Prof. John Leonard
Prof. Patricio Vela
Prof. James M. Rehg

Motivation



Motivation

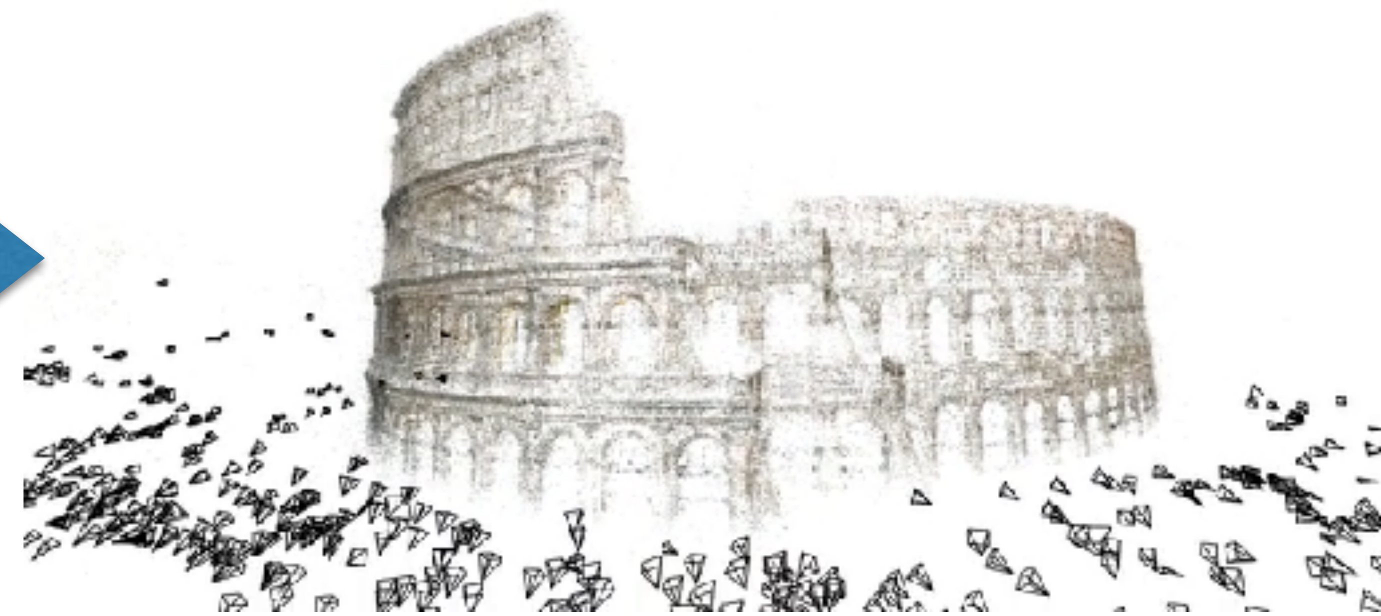
Motivation



Distributed Camera

Modeling the world from internet photo collections
Snavely et al. (IJCV '07)

Motivation



Distributed Camera

Modeling the world from internet photo collections
Snavely et al. (IJCV '07)

Motivation

Motivation

Fast information gathering
in disaster relief scenarios

Motivation

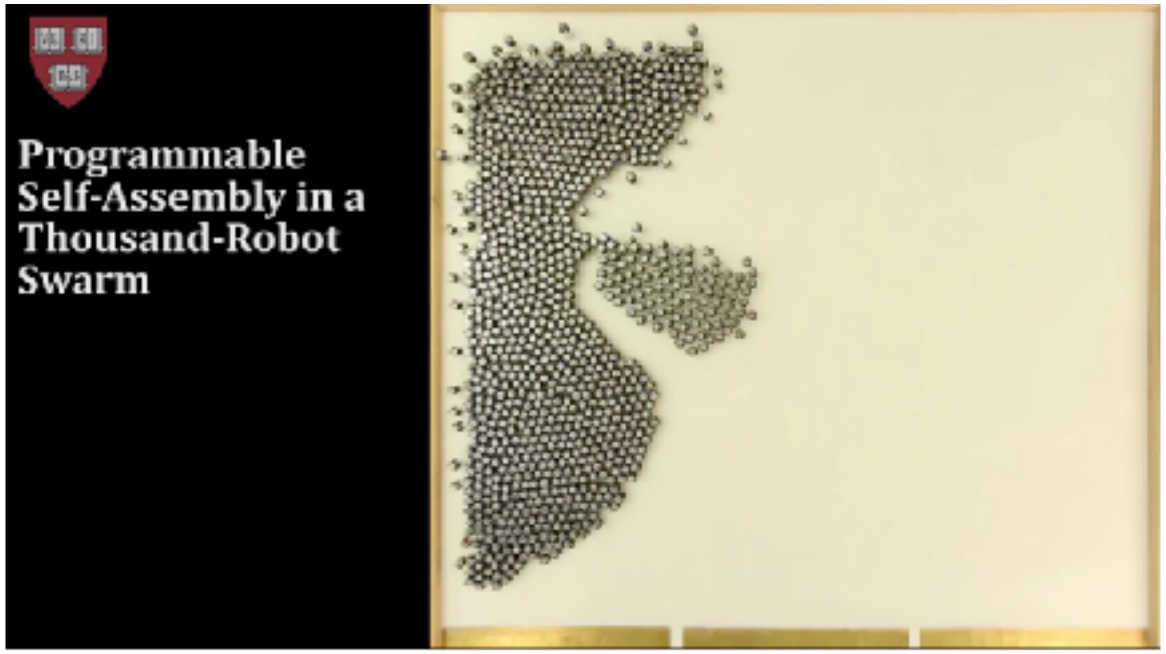
Fast information gathering
in disaster relief scenarios

Efficient coverage and monitoring

Motivation

Fast information gathering
in disaster relief scenarios

Efficient coverage and monitoring

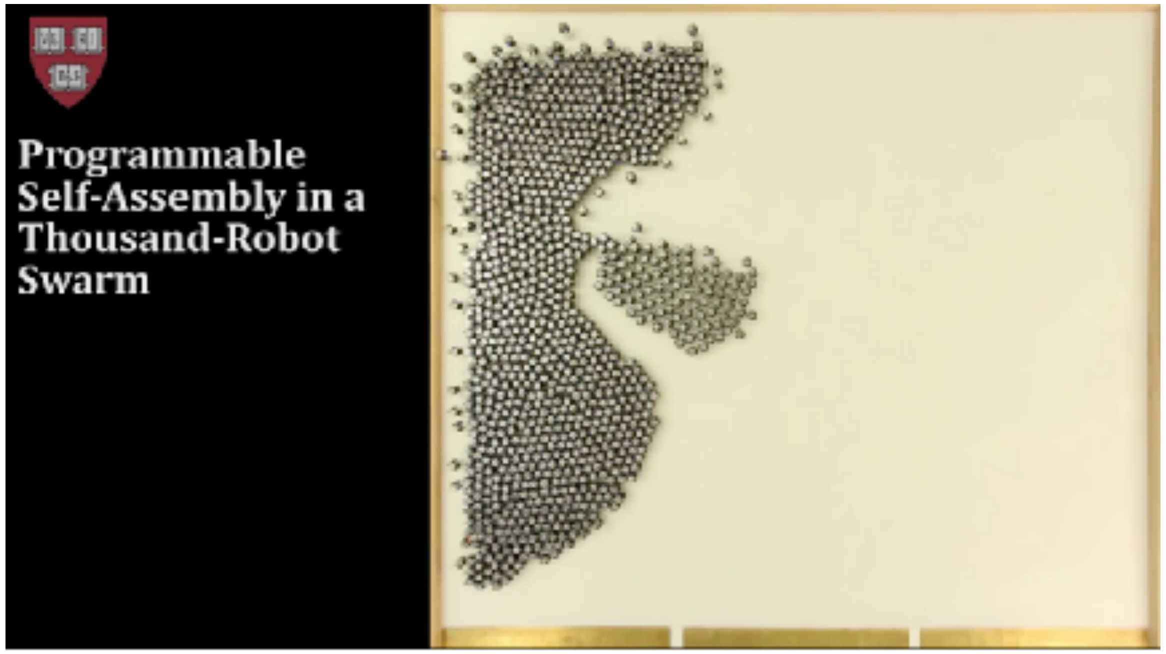


Appealing alternative to
monolithic single robot systems

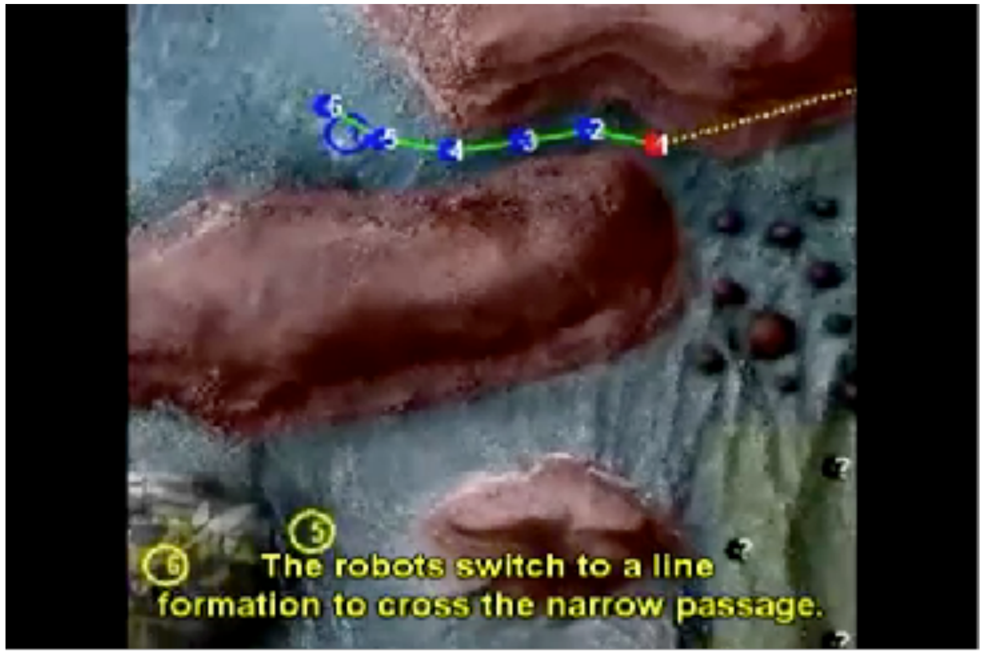
Motivation

Fast information gathering
in disaster relief scenarios

Efficient coverage and monitoring



Appealing alternative to
monolithic single robot systems



Multi Robot Search and Rescue

Motivation

“To develop autonomous, multifunctional, collaborative ensembles of agile, mobile microsystems to enhance tactical situational awareness in urban and complex terrain for small unit operations.”

Partially funded by ARL Micro Autonomous Systems And Technology (MAST) CTA Project

Motivation

"To develop autonomous, multifunctional, **collaborative ensembles** of agile, mobile microsystems to enhance tactical situational awareness in urban and complex terrain for small unit operations."

- **distributed**

Partially funded by ARL Micro Autonomous Systems And Technology (MAST) CTA Project

Motivation

"To develop autonomous, multifunctional, **collaborative ensembles** of **agile, mobile microsystems** to enhance tactical situational awareness in urban and complex terrain for small unit operations."

- **distributed**
- **resource constrained**

Partially funded by ARL Micro Autonomous Systems And Technology (MAST) CTA Project

Motivation

"To develop autonomous, multifunctional, **collaborative ensembles** of **agile, mobile microsystems** to enhance tactical **situational awareness** in urban and complex terrain for small unit operations."

- **distributed**
- **resource constrained**
- **mapping**

Partially funded by ARL Micro Autonomous Systems And Technology (MAST) CTA Project

Goal

Distributed mapping using a team of resource-constrained robots

Partially funded by ARL Micro Autonomous Systems And Technology (MAST) CTA Project

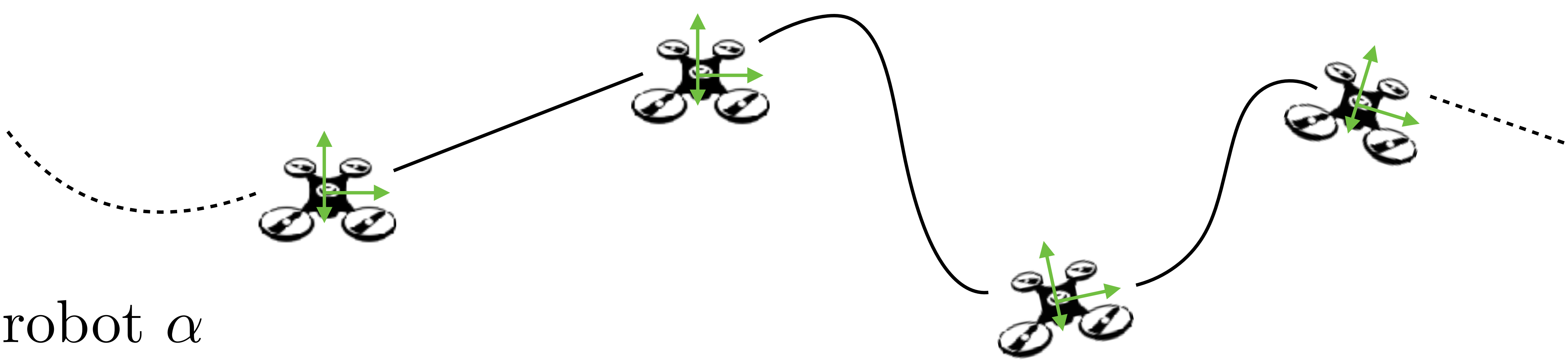
Goal

Distributed mapping using a team of resource-constrained robots

Partially funded by ARL Micro Autonomous Systems And Technology (MAST) CTA Project

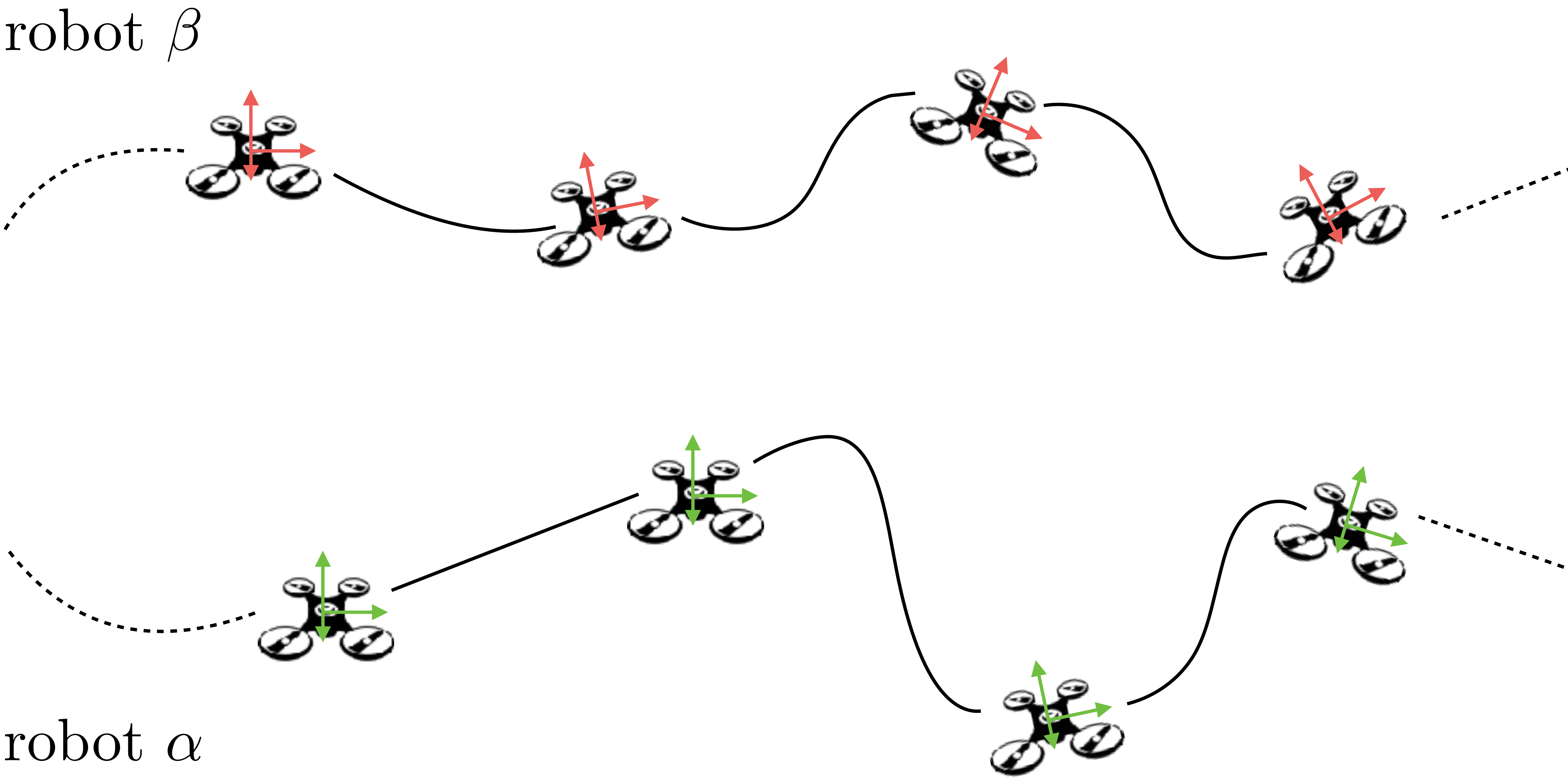
What is distributed mapping?

What is distributed mapping?

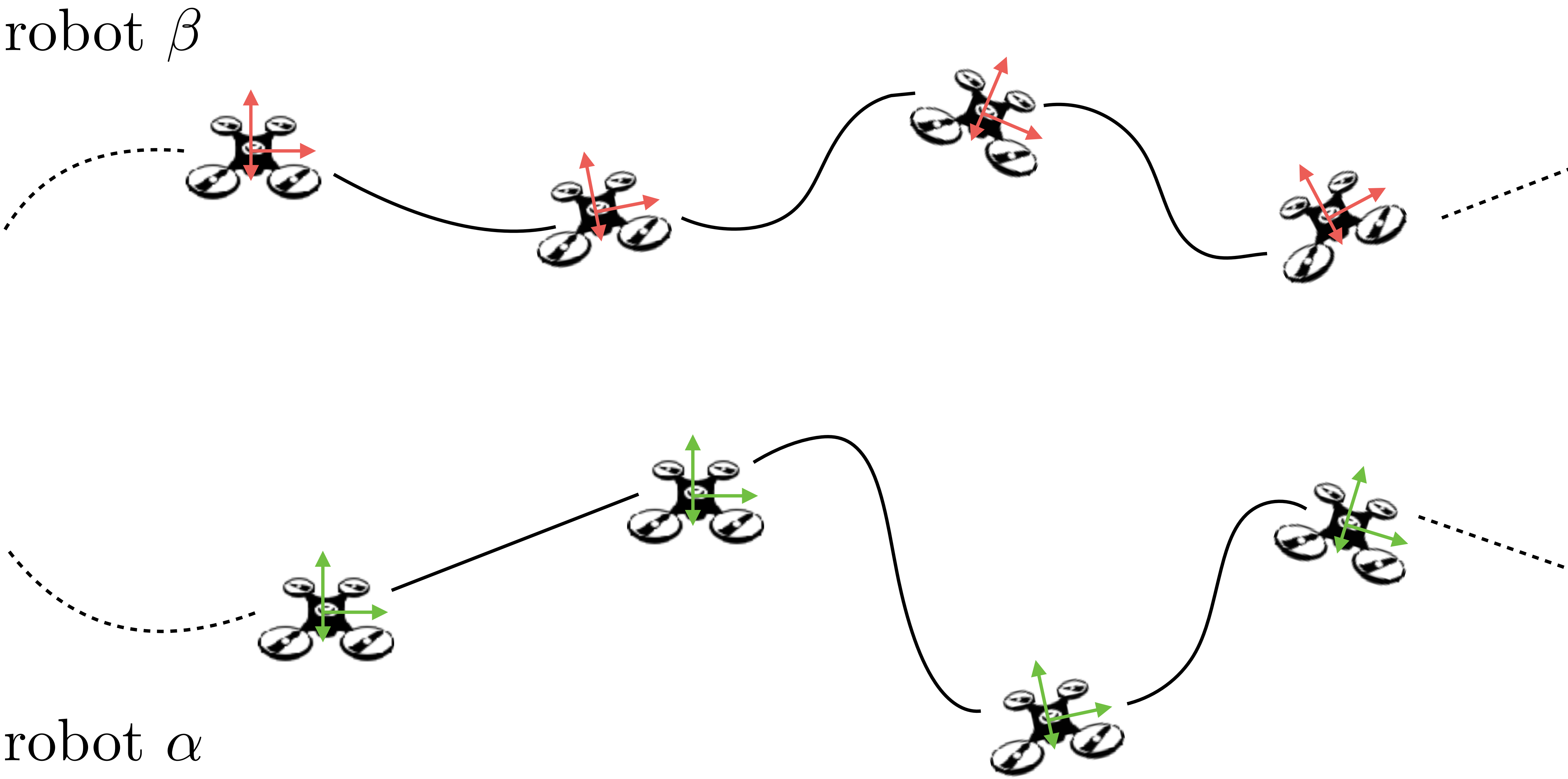


robot α

What is distributed mapping?

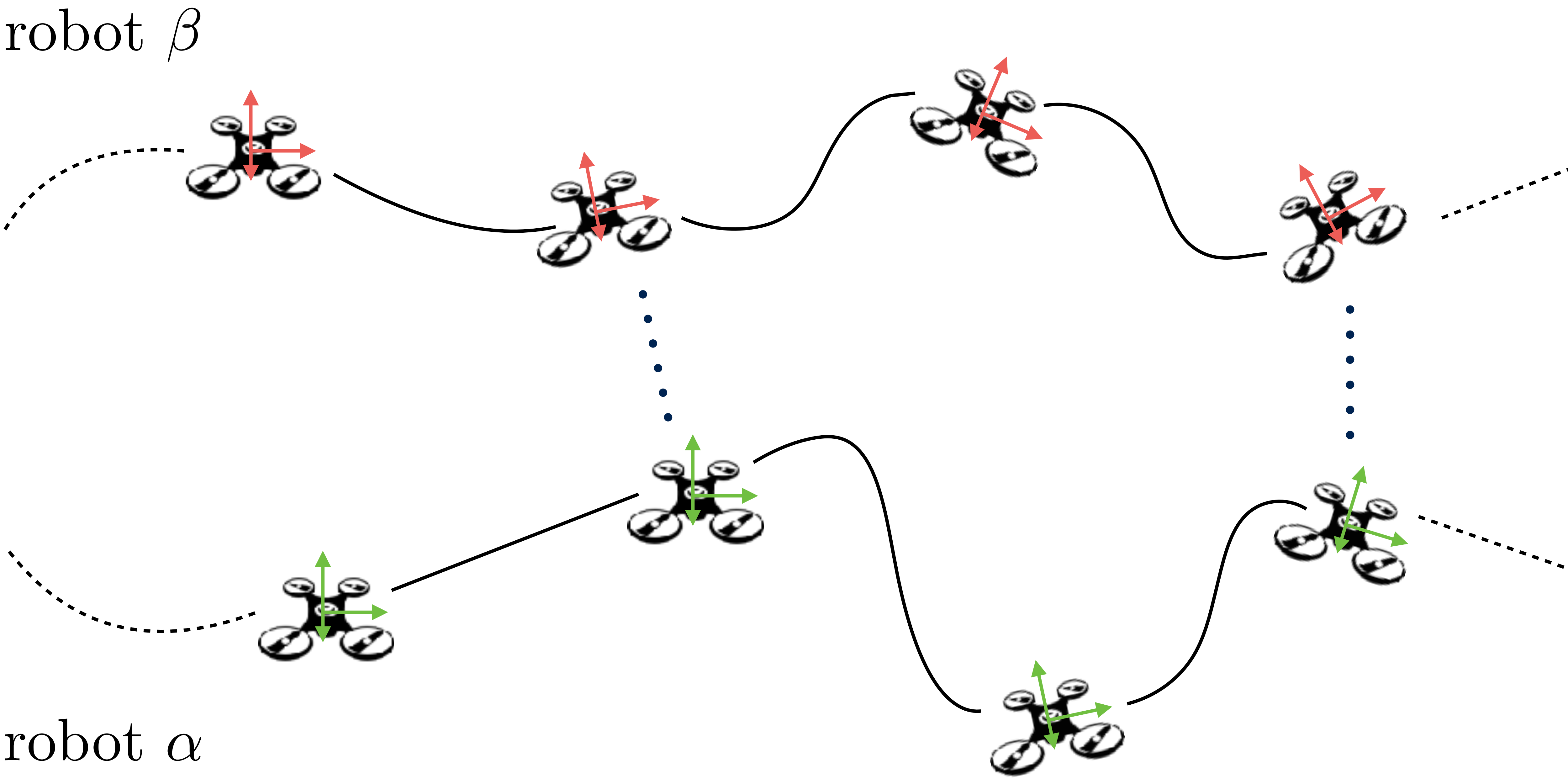


What is distributed mapping?



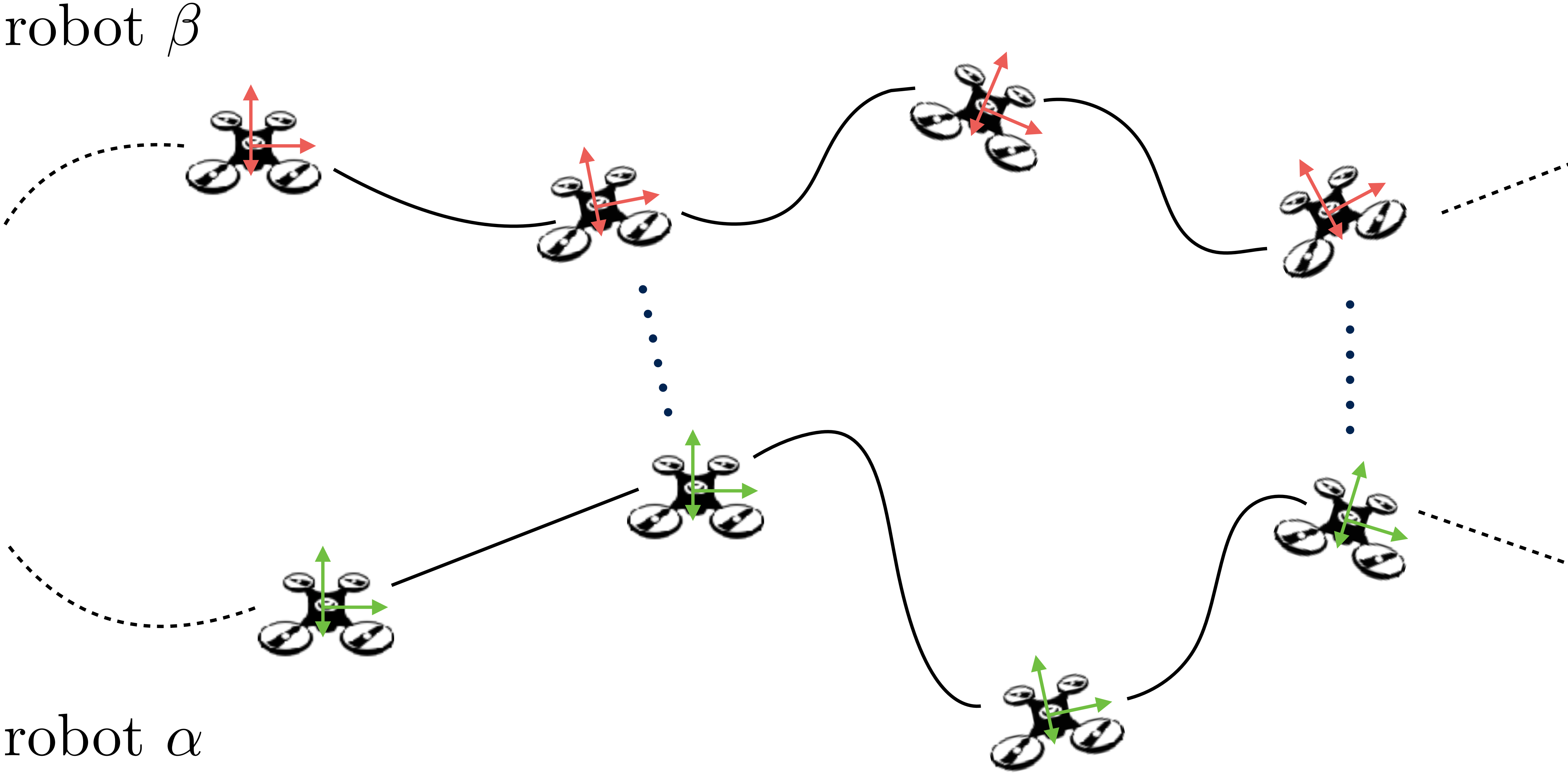
each robot estimates its own trajectory in the local coordinate frame

What is distributed mapping?



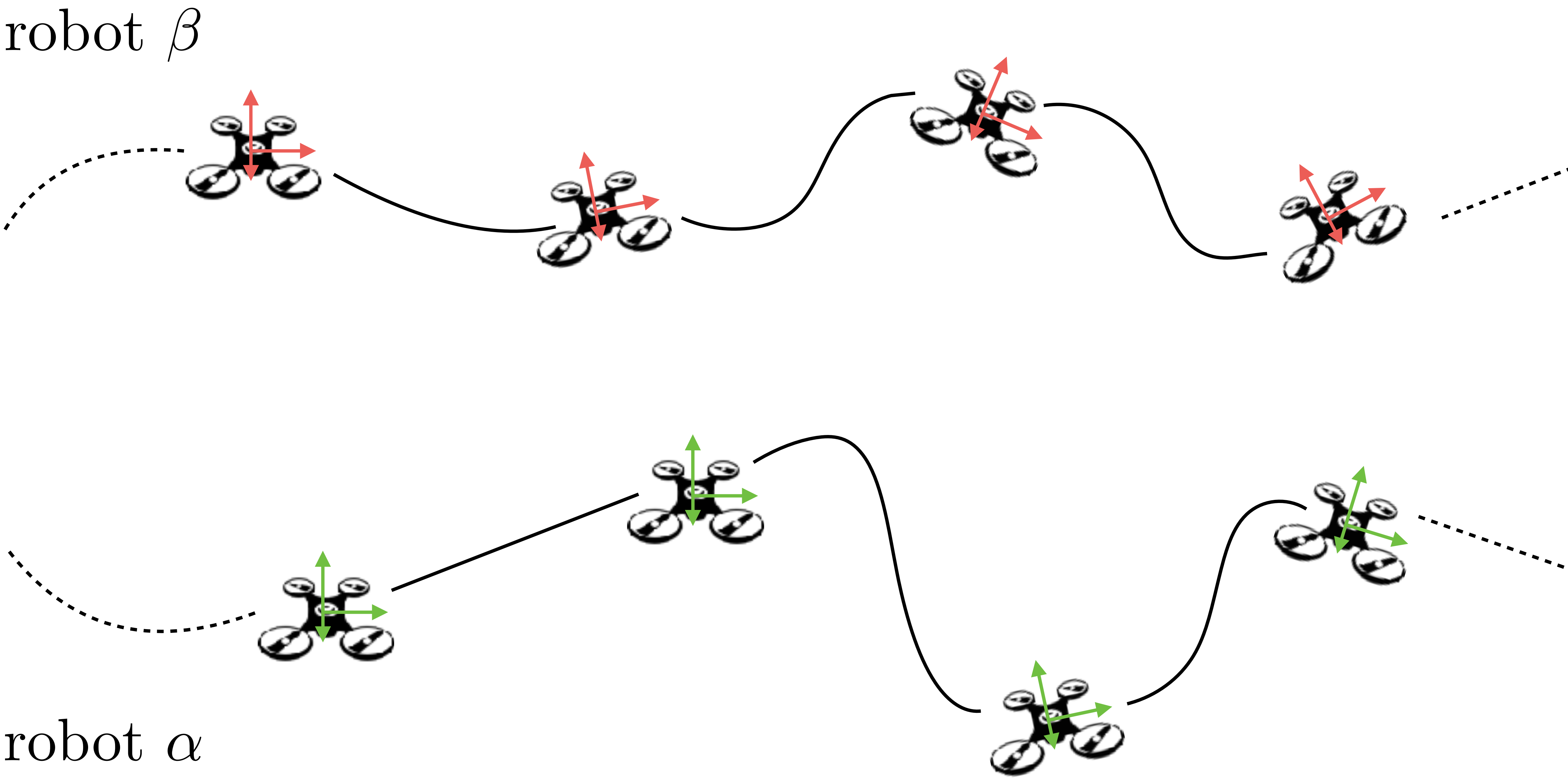
each robot communicates with the neighboring robots

What is distributed mapping?



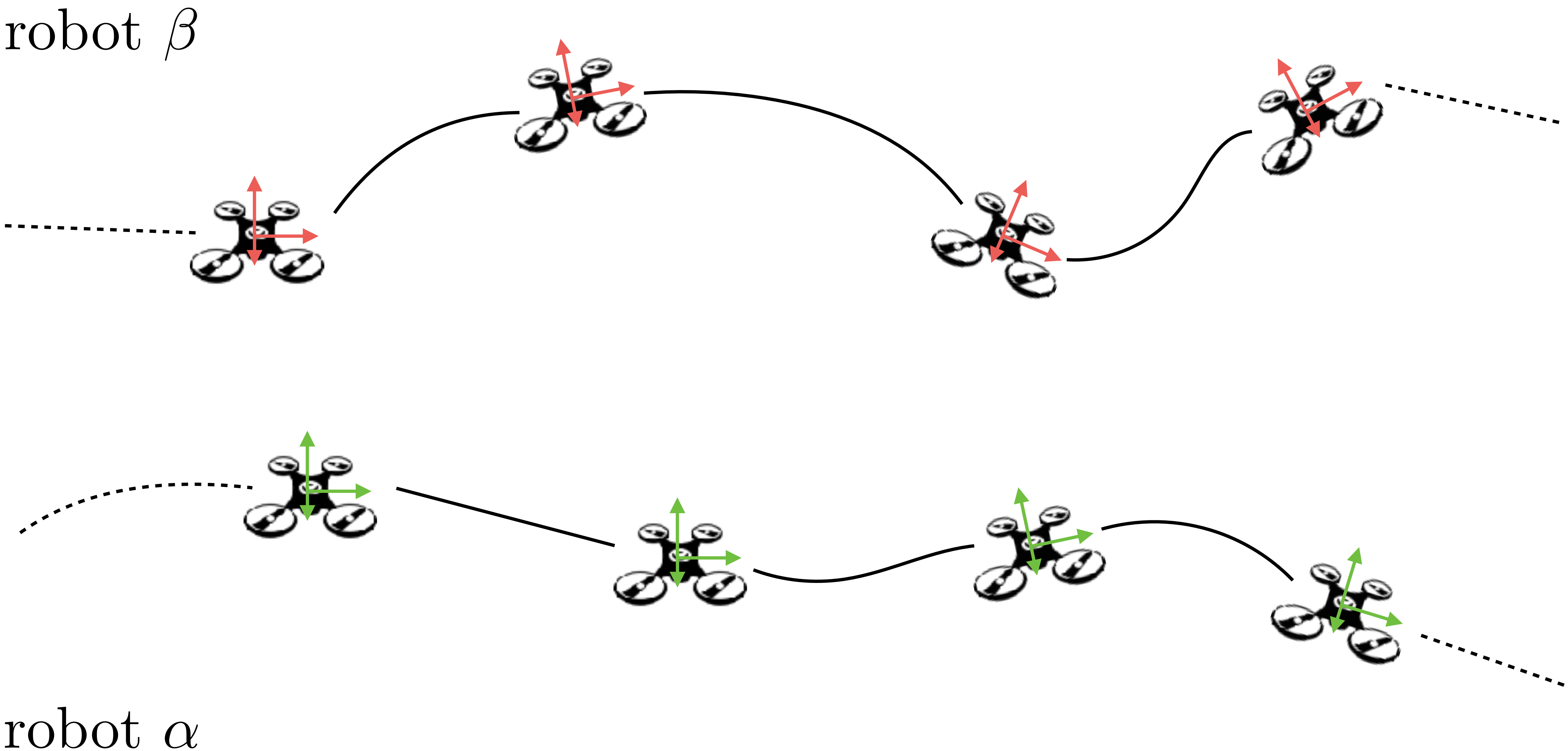
each robot communicates with the neighboring robots

What is distributed mapping?



each robot optimizes its own trajectory given the new information
from the neighboring robots

What is distributed mapping?



after the optimization, the resulting trajectory of each robot is globally consistent

Problem Statement

Cooperative estimation of 3D robot trajectories from **relative pose measurements**, with the following constraints:

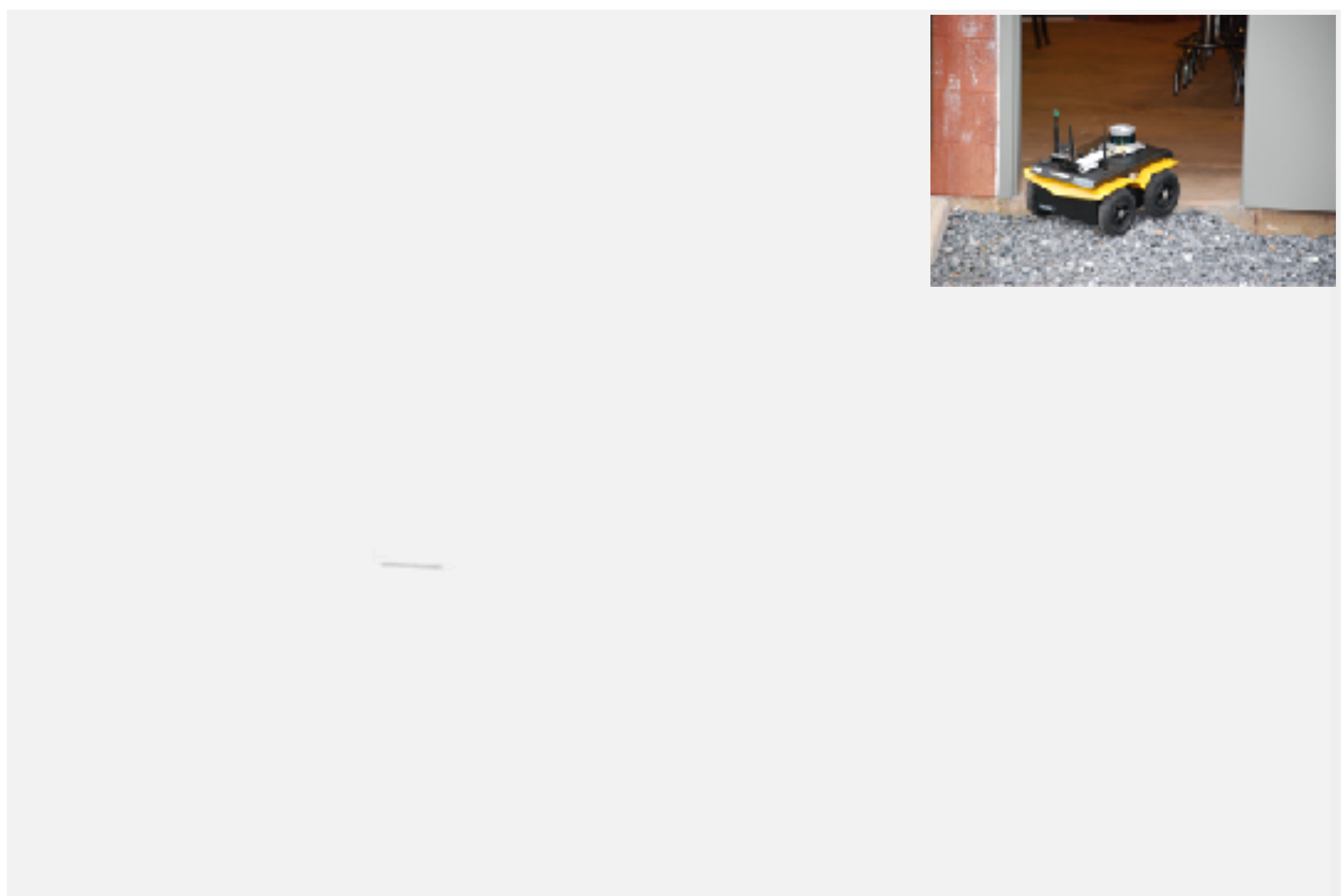
1. Communication only occurs during **rendezvous**.

Indoor mapping with Jackal robots

Problem Statement

Cooperative estimation of 3D robot trajectories from **relative pose measurements**, with the following constraints:

1. Communication only occurs during **rendezvous**.

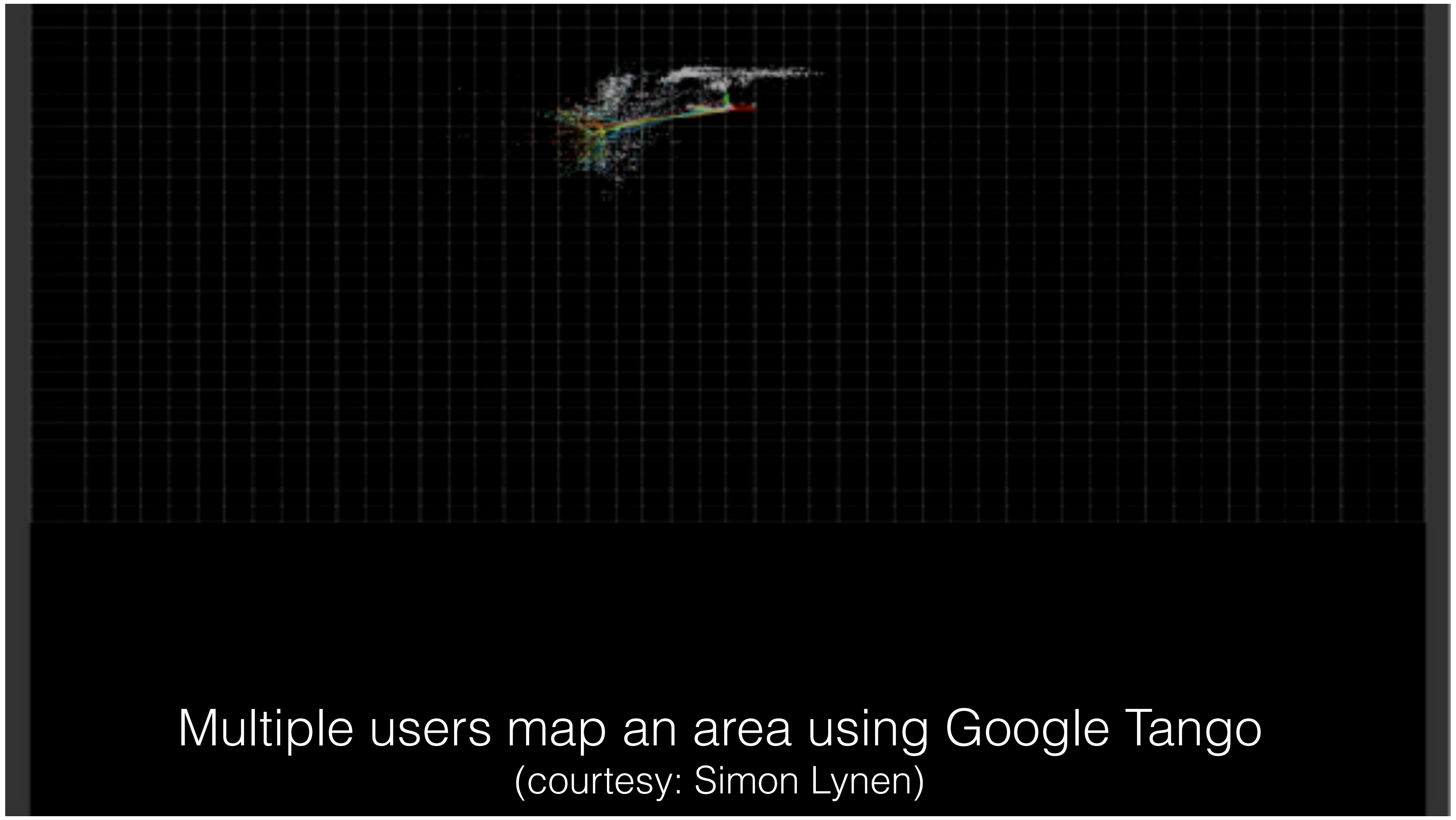


Indoor mapping with Jackal robots

Problem Statement

Cooperative estimation of 3D robot trajectories from **relative pose measurements**, with the following constraints:

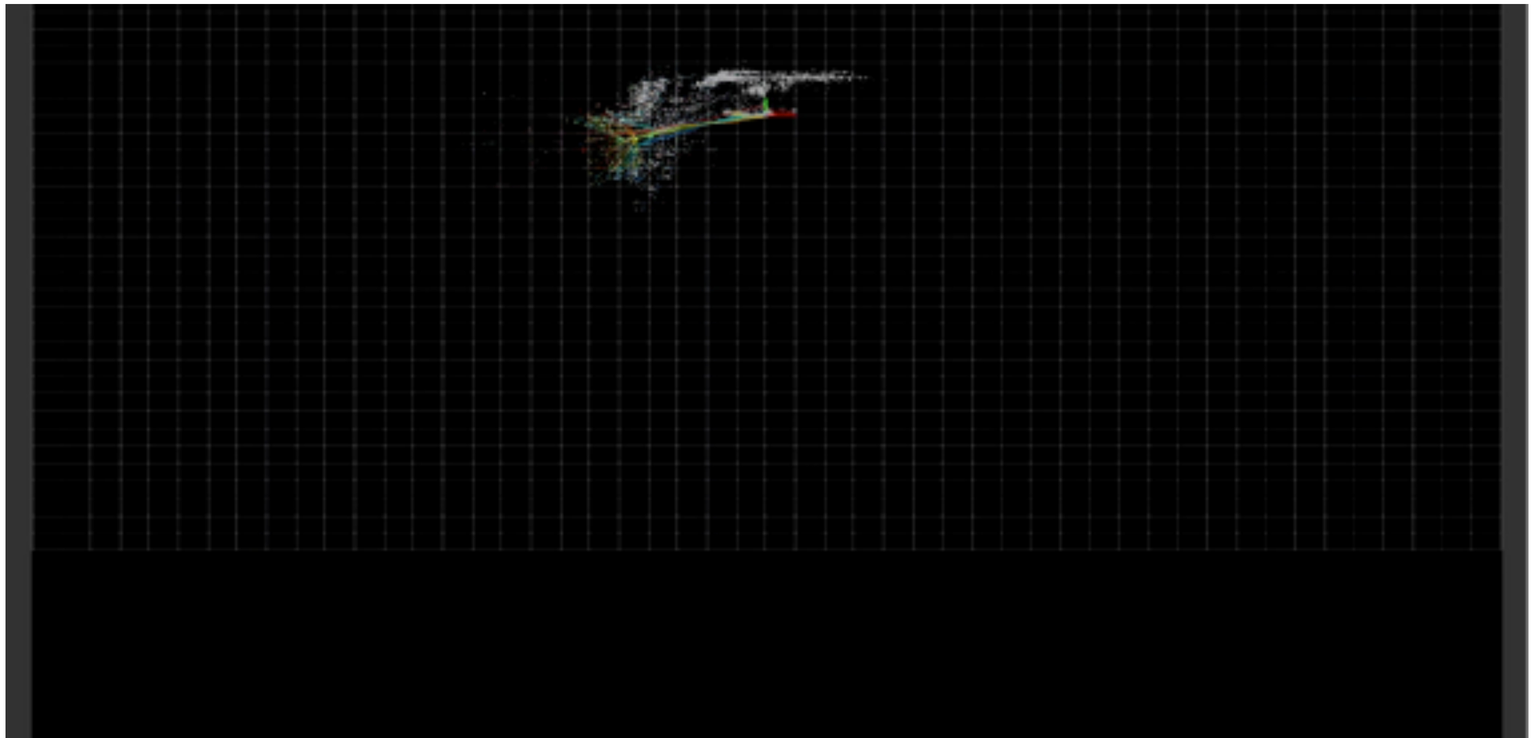
1. Communication only occurs during **rendezvous**.
2. Data exchange must be minimal (due to **limited bandwidth** and **privacy**).



Problem Statement

Cooperative estimation of 3D robot trajectories from **relative pose measurements**, with the following constraints:

1. Communication only occurs during **rendezvous**.
2. Data exchange must be minimal (due to **limited bandwidth** and **privacy**).



Multiple users map an area using Google Tango
(courtesy: Simon Lynen)

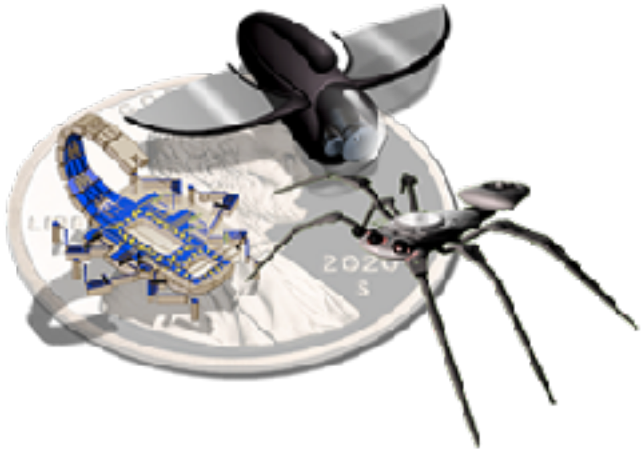
Problem Statement

Cooperative estimation of 3D robot trajectories from **relative pose measurements**, with the following constraints:

1. Communication only occurs during **rendezvous**.

2. Data exchange must be minimal (due to **limited bandwidth** and **privacy**).

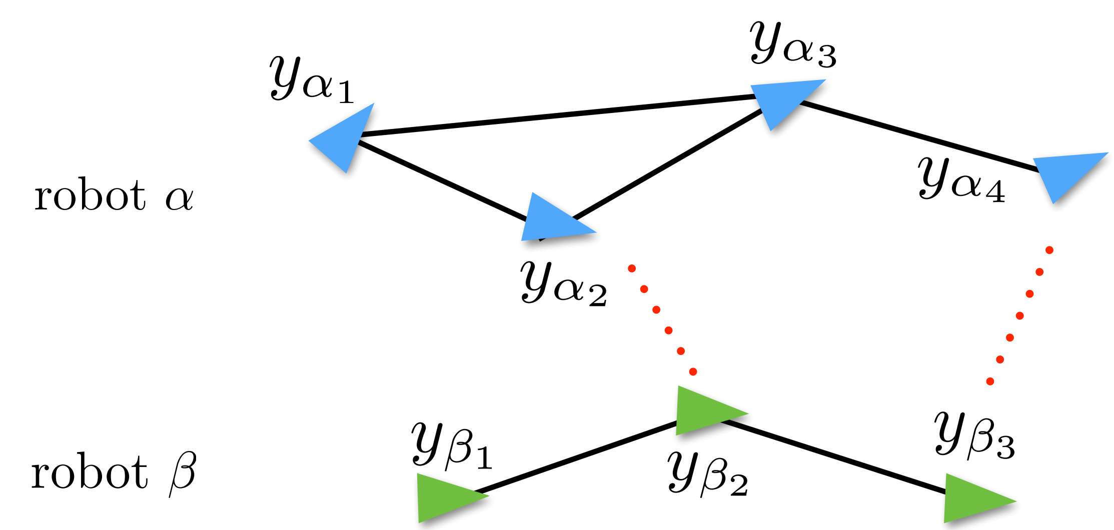
3. **Memory required** by each robot is minimal.



Contributions

Contributions

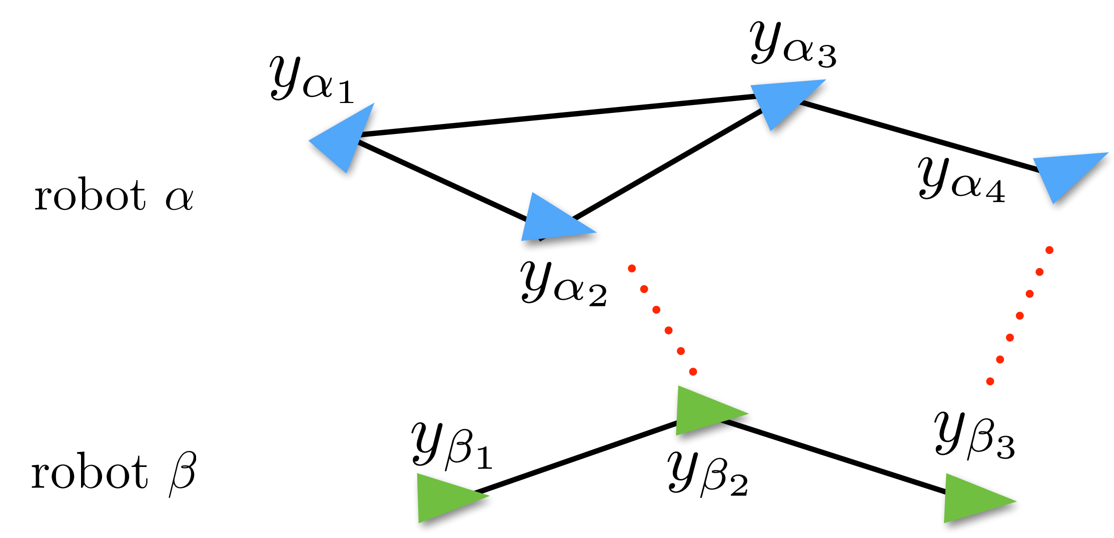
- **Distributed inference algorithm**: split computation of trajectory estimation algorithm among teammates (Choudhary et al., ICRA 2016)



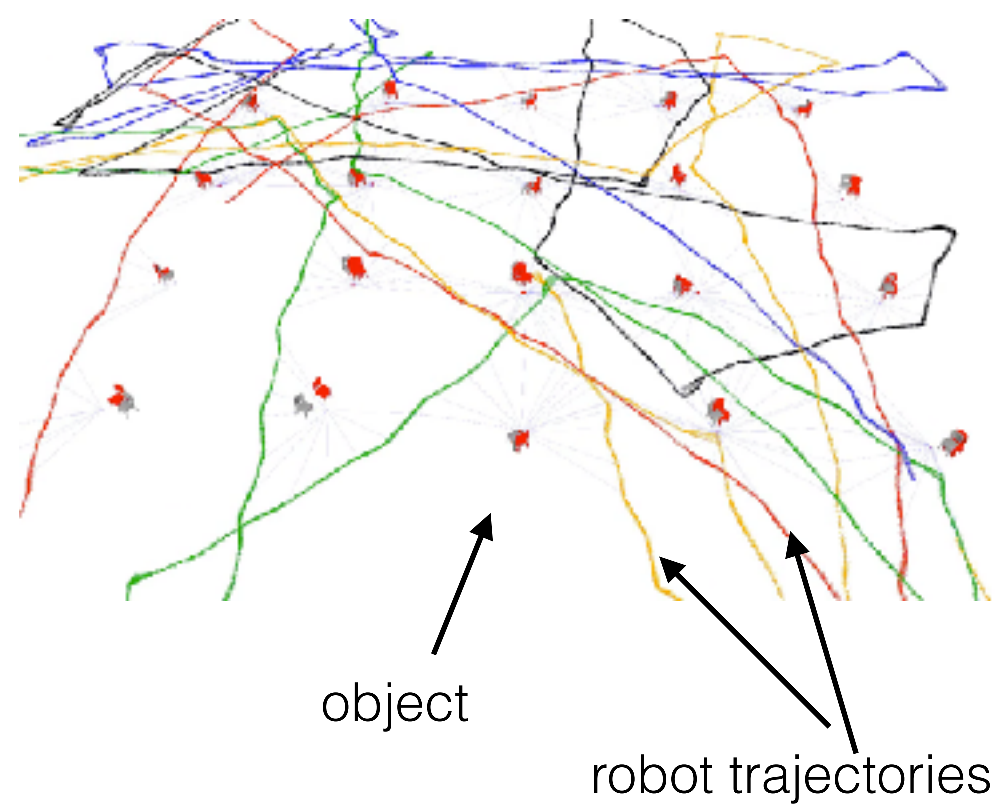
Distributed Trajectory Estimation with Privacy and Communication Constraints: a Two-Stage Distributed Gauss-Seidel Approach
Choudhary et al. (ICRA 2016)

Contributions

- **Distributed inference algorithm**: split computation of trajectory estimation algorithm among teammates (Choudhary et al., ICRA 2016)



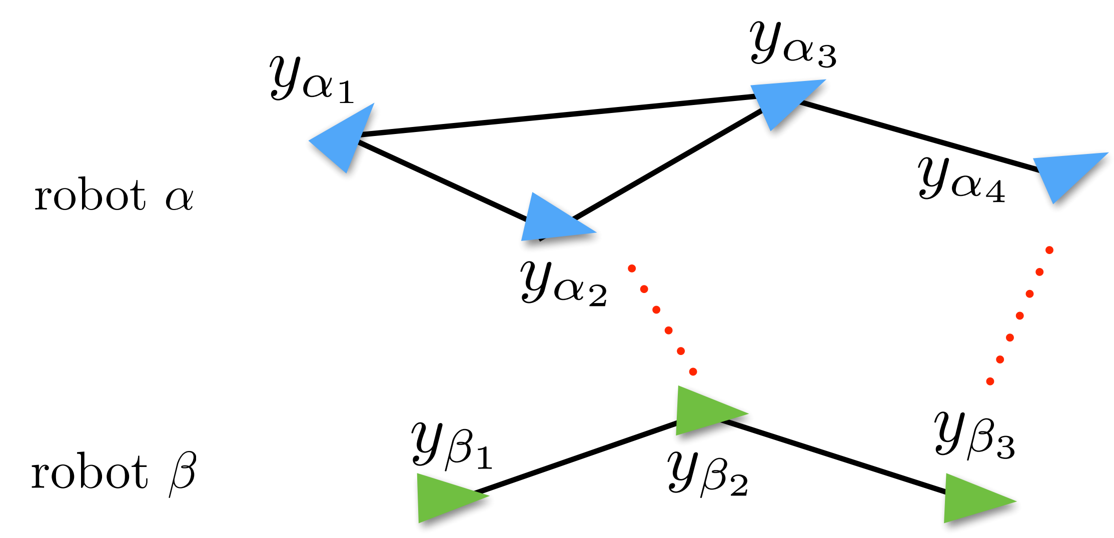
- **High level map representation**: teammates reason in terms of objects (Choudhary et al., ISER 2016)



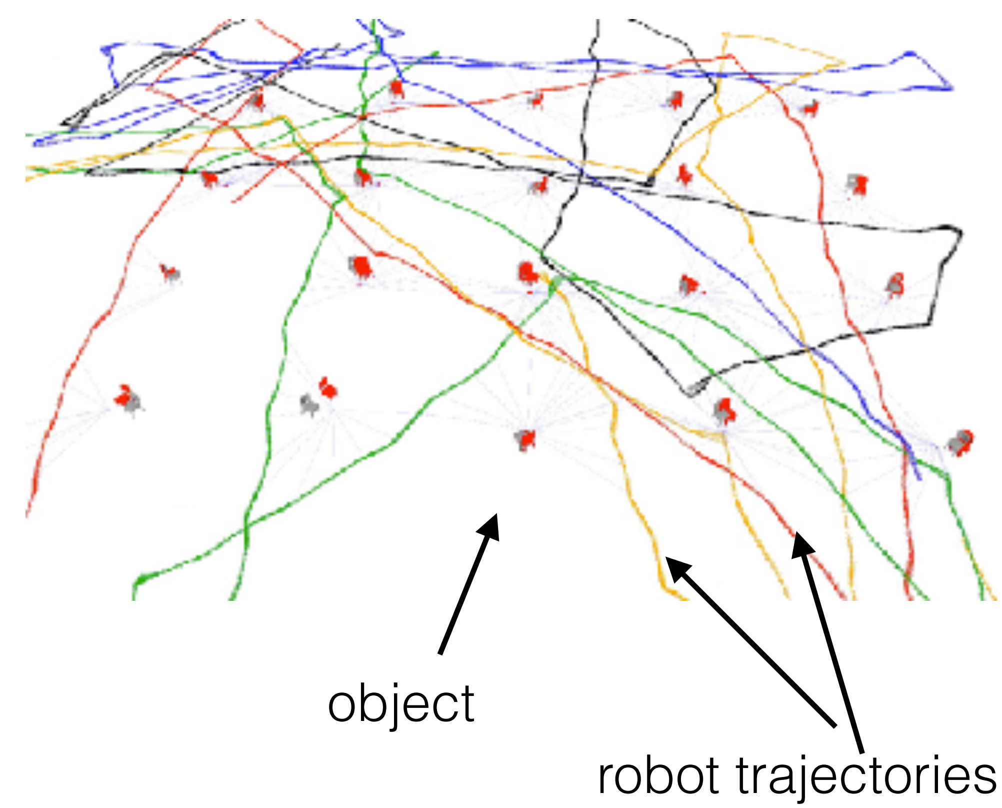
Multi Robot Object-based SLAM
Choudhary et al. (ISER 2016)

Contributions

- **Distributed inference algorithm**: split computation of trajectory estimation algorithm among teammates (Choudhary et al., ICRA 2016)



- **High level map representation**: teammates reason in terms of objects (Choudhary et al., ISER 2016)



Multi Robot Object-based SLAM
Choudhary et al. (ISER 2016)

Thesis Statement

*Using **objects** as landmarks in a **distributed SLAM** framework and leveraging a **state of art distributed optimizer** both **reduce** the **communication bandwidth and the memory** used by each robot, outputs a **human understandable map** and improves the **robustness and scalability** of distributed SLAM.*

Thesis Statement

Using ***objects*** *as landmarks* ***distributed SLAM***

state of art distributed optimizer

reduce *the* ***communication bandwidth and the memory***

outputs a ***human understandable map***

improves the ***robustness and scalability***

Thesis Statement

state of art distributed optimizer

+

*Using **objects** as landmarks*

distributed SLAM

1. ***reduce*** *the **communication bandwidth and the memory***
2. *improves the **robustness and scalability***
3. *outputs a **human understandable map***

Outline

1. Distributed Gauss-Seidel Approach
   (*state of art distributed optimizer*)

2. Distributed Object-based SLAM with Known Object Models
   (*using objects as landmarks*)

3. Distributed Object-based SLAM with Joint Object Modeling and Mapping
   (*jointly model objects along with SLAM*)

4. Conclusions and Future Work

Outline

1. Distributed Gauss-Seidel Approach
   (*state of art distributed optimizer*)

2. Distributed Object-based SLAM with Known Object Models
   (*using objects as landmarks*)

3. Distributed Object-based SLAM with Joint Object Modeling and Mapping
   (*jointly model objects along with SLAM*)

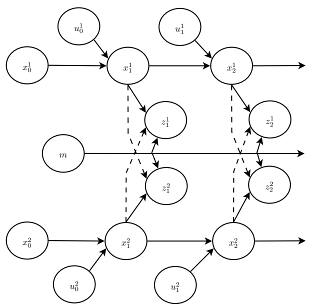
4. Conclusions and Future Work

Distributed Trajectory Estimation

Distributed Trajectory Estimation

- Roumeliotis and Bekey (TRO 2002)
- Thrun and Liu (ISRR 2003)
- A. Howard (IJRR 2006)
- Carlone et al. (JIRS 2011)

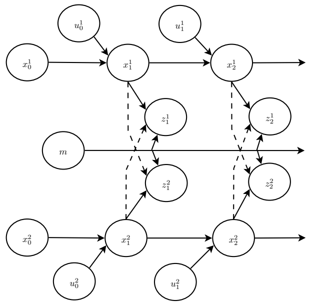
Distributed Filtering



Distributed Trajectory Estimation

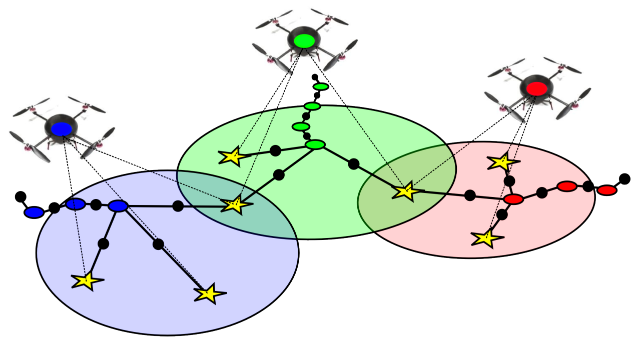
- Roumeliotis and Bekey (TRO 2002)
- Thrun and Liu (ISRR 2003)
- A. Howard (IJRR 2006)
- Carlone et al. (JIRS 2011)

Distributed Filtering



- Andresson and Nygards (ICRA 2008)
- Nerurkar et al. (ICRA 2009)
- Franceschelli and Gasparri (ICRA 2010)
- Kim et al. (ICRA 2010)
- Cunningham et al. (IROS 2010, ICRA 2013)
- Indelman et al. (IJRR 2012)
- Dong et al. (ICRA 2015)
- Paull et al. (ICRA 2015)

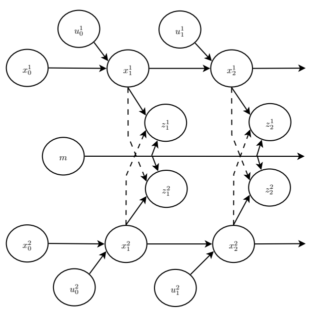
Distributed Smoothing



Distributed Trajectory Estimation

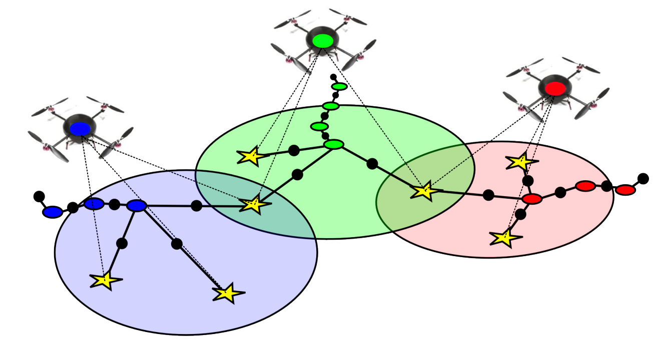
- Roumeliotis and Bekey (TRO 2002)
- Thrun and Liu (ISRR 2003)
- A. Howard (IJRR 2006)
- Carlone et al. (JIRS 2011)

Distributed Filtering



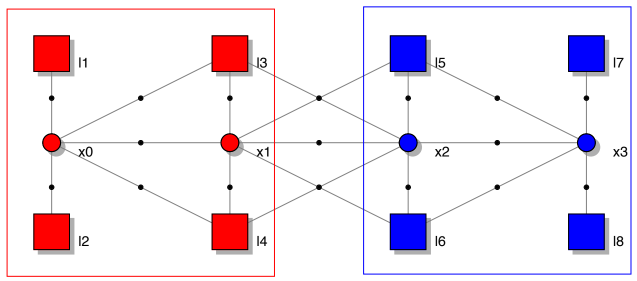
- Andresson and Nygards (ICRA 2008)
- Nerurkar et al. (ICRA 2009)
- Franceschelli and Gasparri (ICRA 2010)
- Kim et al. (ICRA 2010)
- Cunningham et al. (IROS 2010, ICRA 2013)
- Indelman et al. (IJRR 2012)
- Dong et al. (ICRA 2015)
- Paull et al. (ICRA 2015)

Distributed Smoothing



- Leonard and Feder (JOE 2001)
- Leonard and Newman (IJCAI 2003)
- Bosse et al (IJRR 2004)
- Ni et al. (ICRA 2007)

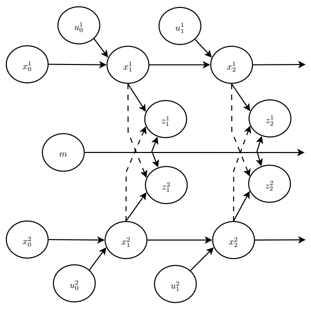
Submapping



Distributed Trajectory Estimation

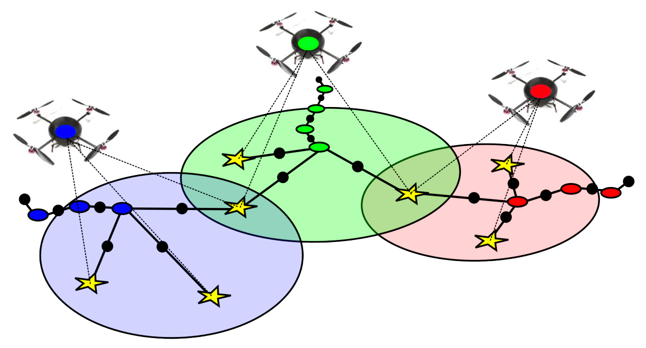
- Roumeliotis and Bekey (TRO 2002)
- Thrun and Liu (ISRR 2003)
- A. Howard (IJRR 2006)
- Carlone et al. (JIRS 2011)

Distributed Filtering



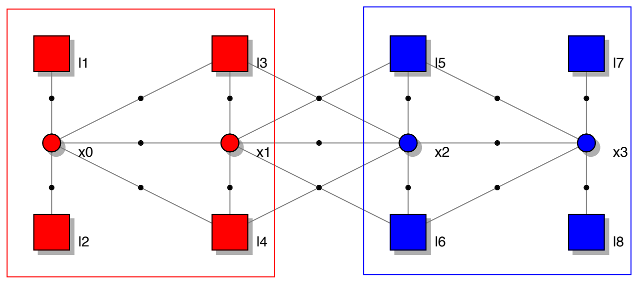
- Andresson and Nygards (ICRA 2008)
- Nerurkar et al. (ICRA 2009)
- Franceschelli and Gasparri (ICRA 2010)
- Kim et al. (ICRA 2010)
- Cunningham et al. (IROS 2010, ICRA 2013)
- Indelman et al. (IJRR 2012)
- Dong et al. (ICRA 2015)
- Paull et al. (ICRA 2015)

Distributed Smoothing



- Leonard and Feder (JOE 2001)
- Leonard and Newman (IJCAI 2003)
- Bosse et al (IJRR 2004)
- Ni et al. (ICRA 2007)

Submapping



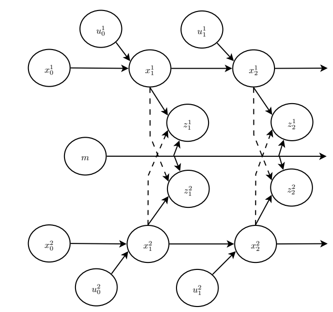
- Anderson et al. (SIAM Journal of Disc. Math. 2010)
- Calafiore et al. (TSMC 2012)
- Barooah and Hespanha (Control Systems Magazine 2007)
- Aragues et al. (System and Control Letters 2012)
- Thunberg et al. (CDC 2011)
- Tron and Vidal (CDC 2009)

Sensor Network Localization

Distributed Trajectory Estimation

- Roumeliotis and Bekey (TRO 2002)
- Thrun and Liu (ISRR 2003)
- A. Howard (IJRR 2006)
- Carlone et al. (JIRS 2011)

Distributed Filtering



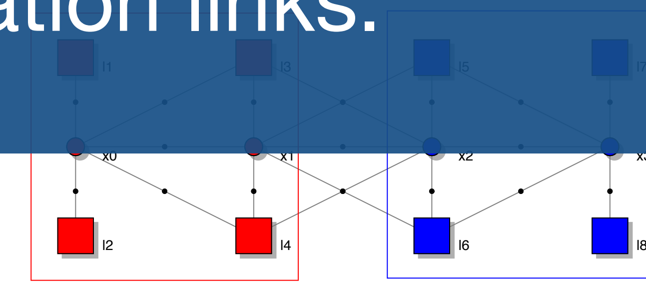
- Andresson and Nygards (ICRA 2008)
- Nerurkar et al. (ICRA 2009)
- Franceschelli and Gasparri (ICRA 2010)
- Kim et al. (ICRA 2010)
- Cunningham et al. (IROS 2010, ICRA 2013)
- Indelman et al. (IJRR 2012)
- Dong et al. (ICRA 2015)
- Paull et al. (ICRA 2015)

Distributed Smoothing

The state of art in robotics requires communication cost quadratic in the number of communication links.

- Leonard and Feder (JOE 2001)
- Leonard and Newman (IJCAI 2003)
- Bosse et al (IJRR 2004)
- Ni et al. (ICRA 2007)

Submapping



- Anderson et al. (SIAM Journal of Disc. Math. 2010)
- Calafiore et al. (TSMC 2012)
- Barooah and Hespanha (Control Systems Magazine 2007)
- Aragues et al. (System and Control Letters 2012)
- Thunberg et al. (CDC 2011)
- Tron and Vidal (CDC 2009)

Sensor Network Localization

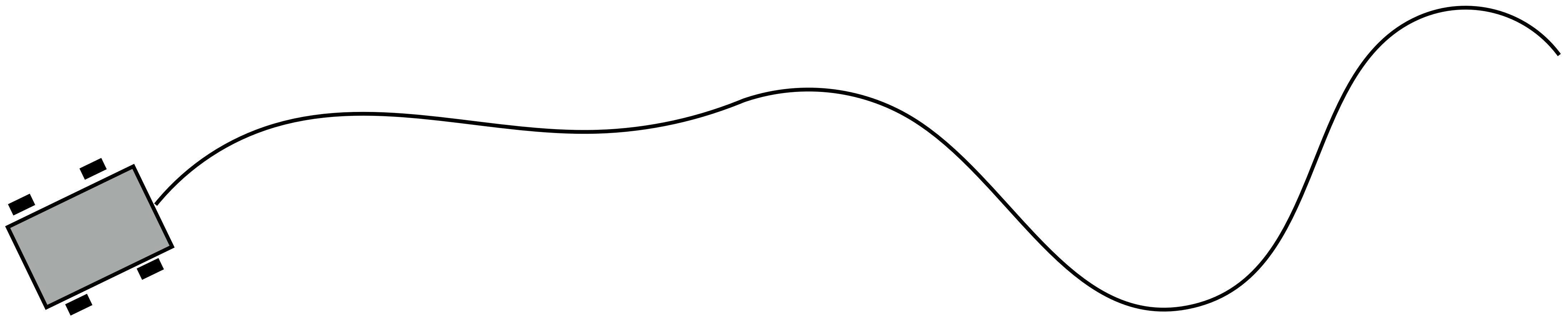
DDF-SAM: Fully distributed SLAM using constrained factor graphs
Cunningham et al. (IROS 2010)

Our Approach

Trajectory estimation as Pose Graph Optimization

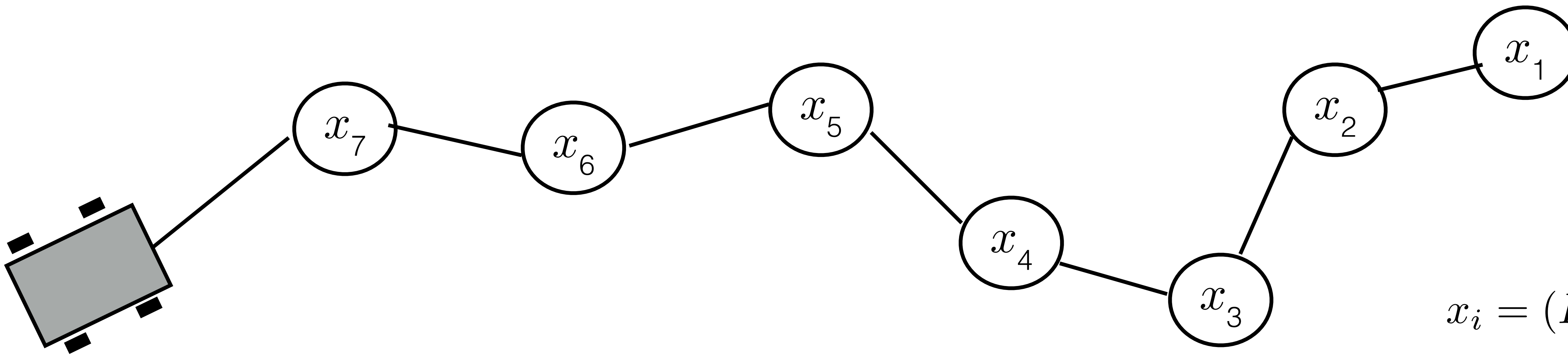
Our Approach

Trajectory estimation as Pose Graph Optimization



Our Approach

Trajectory estimation as Pose Graph Optimization



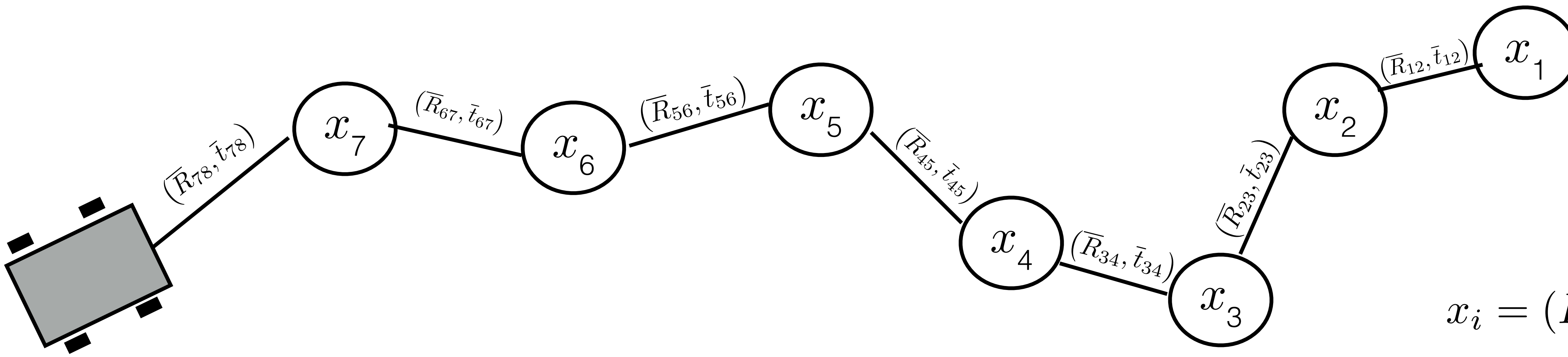
$x_i = (R_i, t_i) \in SE(3)$

we represent a smooth trajectory using a finite set of 3D poses

c.f. Luca Carlone

Our Approach

Trajectory estimation as Pose Graph Optimization



$$x_i = (R_i, t_i) \in SE(3)$$

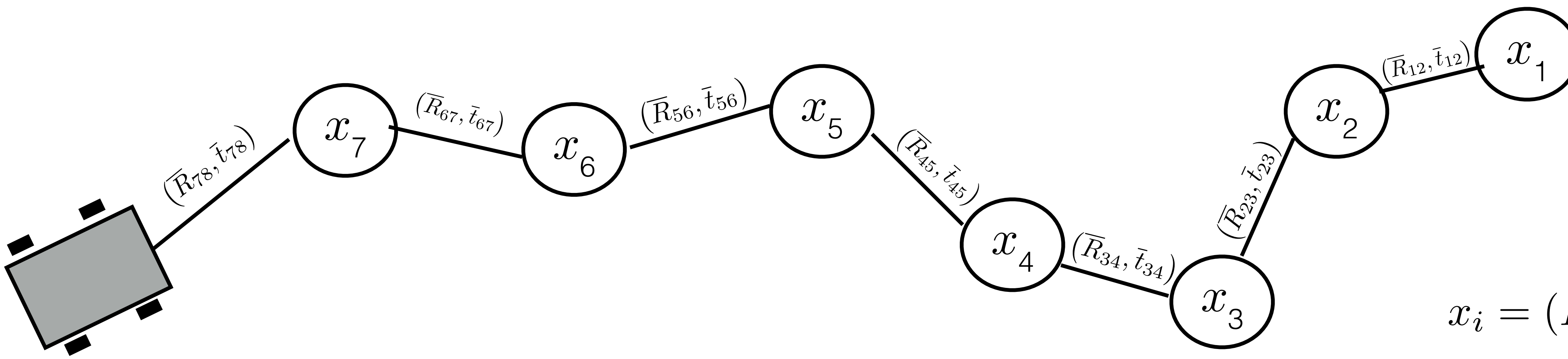
$$\overline{\boldsymbol{z}}_{ij} = (\overline{R}_{ij}, \bar{t}_{ij}) \quad \begin{cases} \overline{R}_{ij} = R_i^T R_j \ . \ noise \\ \bar{t}_{ij} = R_i^T (t_j - t_i) + noise \end{cases}$$

Edges in the graph correspond to relative pose measurements between pairs of poses

c.f. Luca Carlone

Our Approach

Trajectory estimation as Pose Graph Optimization



$$x_i = (R_i, t_i) \in SE(3)$$

$$\bar{\boldsymbol{z}}_{ij} = (\overline{R}_{ij}, \bar{t}_{ij}) \quad \begin{cases} \overline{R}_{ij} = R_i^T R_j \ . \ noise \\ \bar{t}_{ij} = R_i^T (t_j - t_i) + noise \end{cases}$$

$$\boxed{\min_{t_i, R_i} \sum_{(i,j)\in\mathcal{E}} \omega_t^2 \|t_j - t_i - R_i \bar{t}_{ij}\|^2 + \omega_R^2 \|R_j - R_i \overline{R}_j^i\|_F^2}$$

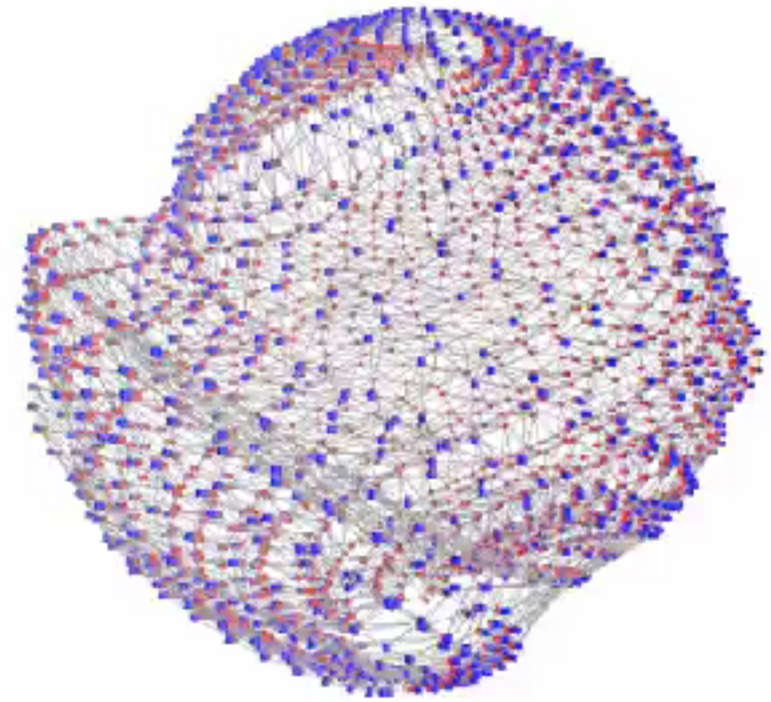
pose graph optimization problem

c.f. Luca Carlone

Our Approach

Trajectory estimation as Pose Graph Optimization:

Related work: iterative optimization

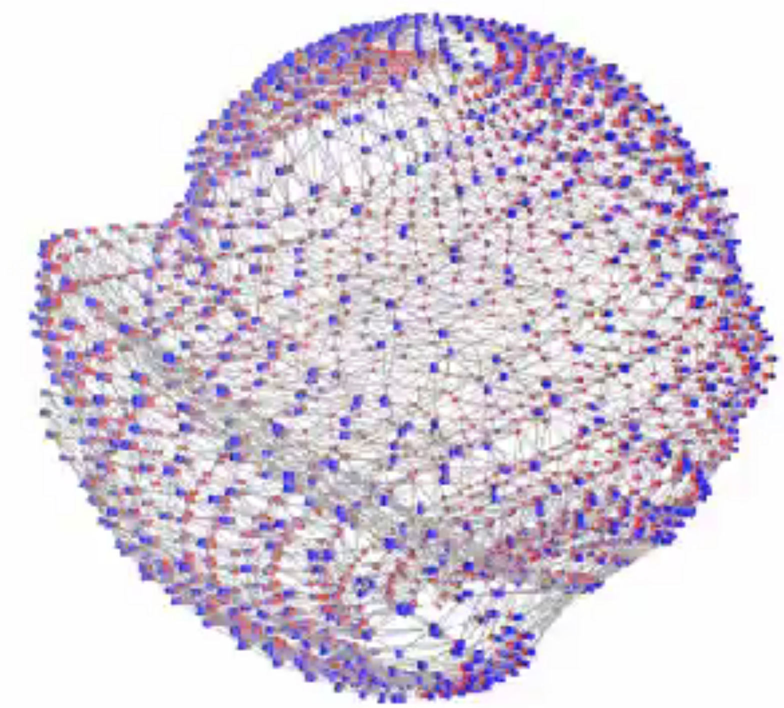


SLAM - TORO - Sphere Optimization
courtesy: Cyril Stachniss

Our Approach

Trajectory estimation as Pose Graph Optimization:

Related work: iterative optimization

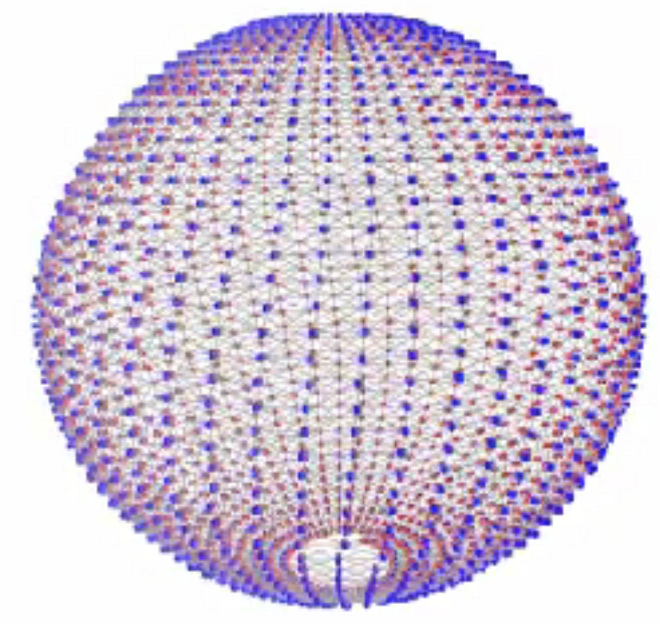


SLAM - TORO - Sphere Optimization
courtesy: Cyril Stachniss

Our Approach

Trajectory estimation as Pose Graph Optimization:

Related work: iterative optimization

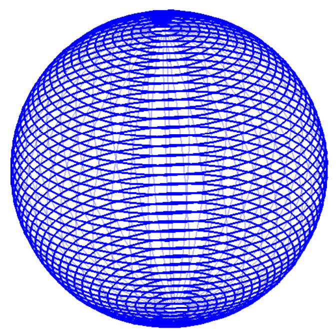


SLAM - TORO - Sphere Optimization
courtesy: Cyril Stachniss

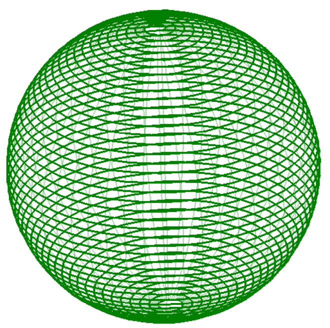
Our approach: Two stage [Carlone et al. (ICRA 2015)]

solve rotation ⟶ solve poses

- Each phase requires solving a linear system
  - shown to have near optimum results
  - robust to bad initialization



Estimate

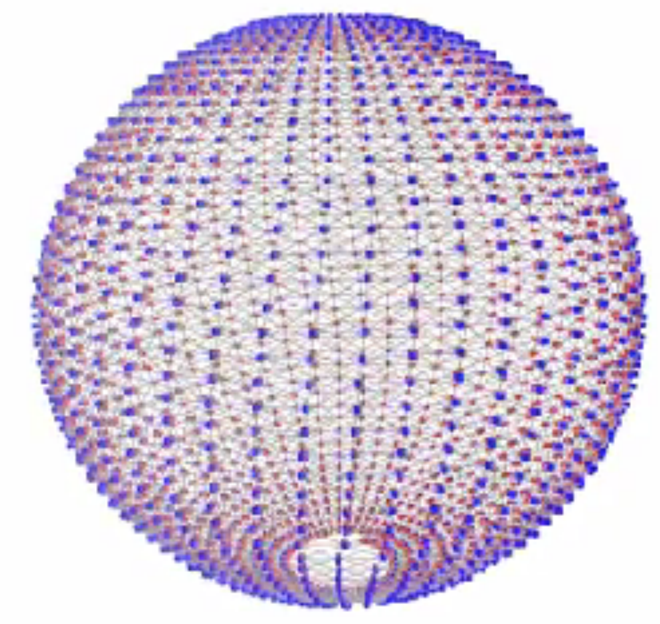


Optimum

Initialization Techniques for 3D SLAM: a Survey on Rotation Estimation and its Use in Pose Graph Optimization
Carlone et al. (ICRA 2015)

Our Approach

Trajectory estimation as Pose Graph Optimization:



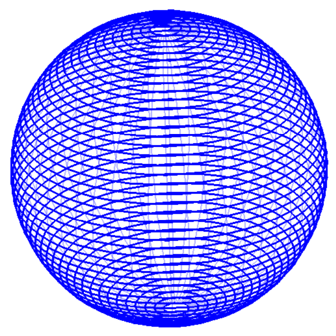
SLAM - TORO - Sphere Optimization
courtesy: Cyril Stachniss

Related work: iterative optimization

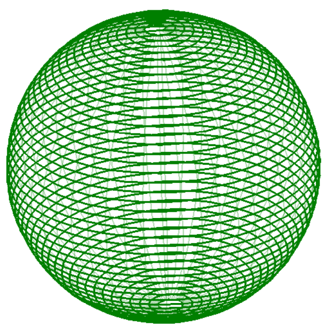
Our approach: Two stage [Carlone et al. (ICRA 2015)]

solve rotation ⟶ solve poses

- Each phase requires solving a linear system
  - shown to have near optimum results
  - robust to bad initialization



Estimate



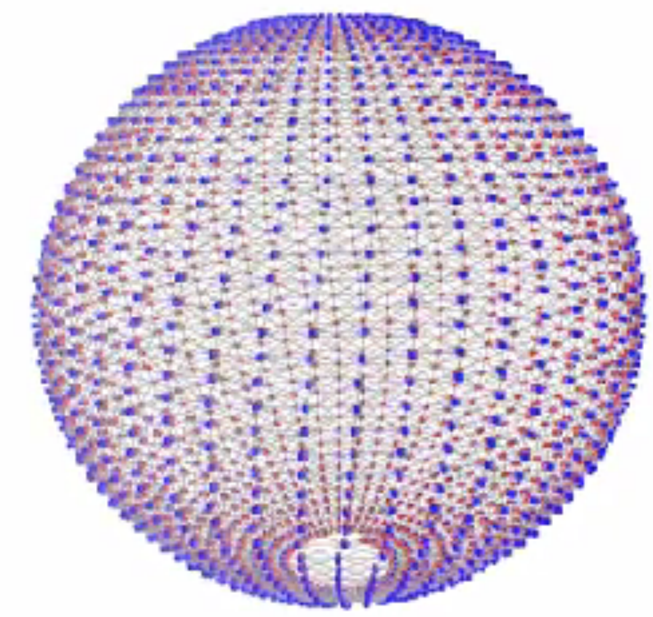
Optimum

Initialization Techniques for 3D SLAM: a Survey on Rotation Estimation and its Use in Pose Graph Optimization
Carlone et al. (ICRA 2015)

Our Approach

Trajectory estimation as Pose Graph Optimization:

Related work: iterative optimization

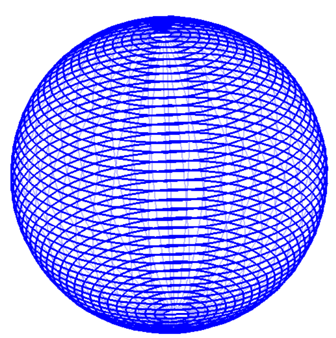


SLAM - TORO - Sphere Optimization
courtesy: Cyril Stachniss

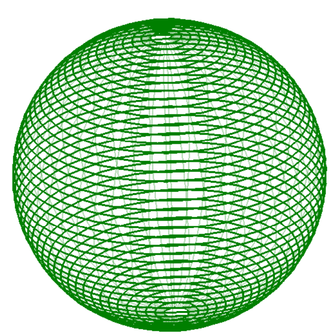
Our approach: Two stage [Carlone et al. (ICRA 2015)]

solve rotation $\longrightarrow$ solve poses

- Each phase requires solving a linear system
- We use the **Gauss-Seidel algorithm as distributed linear solver**



Estimate

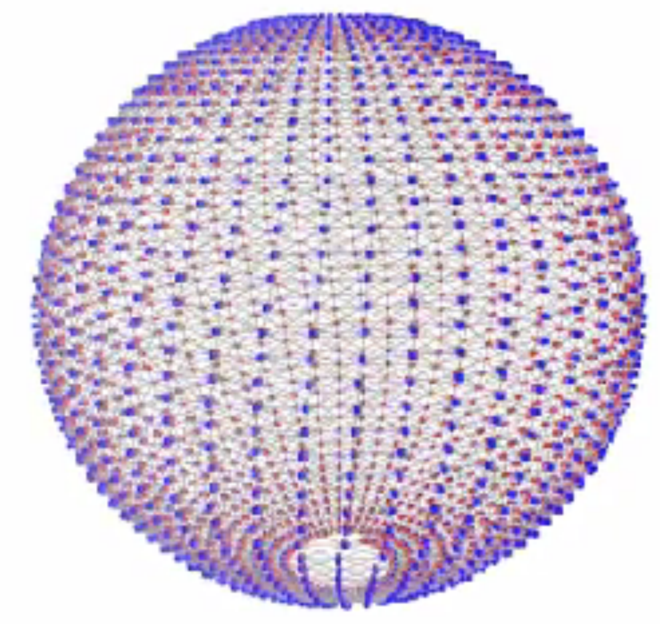


Optimum

Initialization Techniques for 3D SLAM: a Survey on Rotation Estimation and its Use in Pose Graph Optimization
Carlone et al. (ICRA 2015)

Our Approach

Trajectory estimation as Pose Graph Optimization:



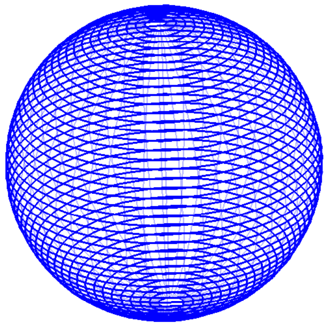
SLAM - TORO - Sphere Optimization
courtesy: Cyril Stachniss

Related work: iterative optimization

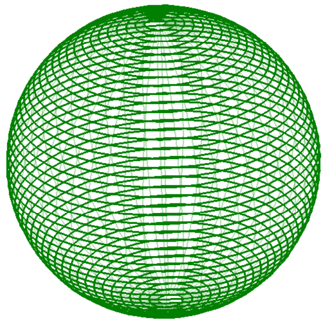
Our approach: Two stage [Carlone et al. (ICRA 2015)]

solve rotation ⟶ solve poses

- Each phase requires solving a linear system
- We use the **Gauss-Seidel algorithm as distributed linear solver**



Estimate



Optimum

Initialization Techniques for 3D SLAM: a Survey on Rotation Estimation and its Use in Pose Graph Optimization
Carlone et al. (ICRA 2015)

Centralized Two-Stage Approach

1. Solve for **rotations first**.

2. Given the rotations, **recover full poses** via a **single Gauss-Newton iteration**

Initialization Techniques for 3D SLAM: a Survey on Rotation Estimation and its Use in Pose Graph Optimization
Carlone et al. (ICRA 2015)

Centralized Two-Stage Approach

$$\min_{t_i, R_i} \sum_{(i,j)\in\mathcal{E}} \omega_t^2 \| t_j - t_i - R_i \bar{t}_{ij} \|^2 + \omega_R^2 \| R_j - R_i \overline{R}_j^i \|_F^2$$

Centralized Two-Stage Approach

$$\min_{t_i, R_i} \sum_{(i,j)\in\mathcal{E}} \boxed{\omega_t^2 \|t_j - t_i - R_i \bar{t}_{ij}\|^2} + \boxed{\omega_R^2 \|R_j - R_i \overline{R}_j^i\|_F^2}$$

translation subproblem **rotation subproblem**

Centralized Two-Stage Approach

$$\min_{R_i} \sum_{(i,j)\in\mathcal{E}}$$

$$\boxed{\omega_R^2 \|R_j - R_i \overline{R}_j^i\|_F^2}$$

rotation subproblem

Centralized Two-Stage Approach

$$\min_{\boldsymbol{R}_i \in \mathrm{SO}(3)} \sum_{(i,j)\in\mathcal{E}} \omega_R^2 \| R_j - R_i \bar{R}_i^j \|_F^2$$

Centralized Two-Stage Approach

$$\min_{\boldsymbol{R}_i \in \mathrm{SO}(3)} \sum_{(i,j)\in\mathcal{E}} \omega_R^2 \| R_j - R_i \bar{R}_i^j \|_F^2$$

$\downarrow$ quadratic relaxation

$$\min_y \|Ay - b\|^2$$

$\downarrow$ normal equation

$$\underbrace{(A^T A)}_{\text{Hessian (H)}} y = \underbrace{A^T b}_{\text{g}}$$

$\downarrow$

$$Hy = g$$

Centralized Two-Stage Approach

$$\min_{\boldsymbol{R}_i \in \mathrm{SO}(3)} \sum_{(i,j)\in\mathcal{E}} \omega_R^2 \| R_j - R_i \bar{R}_i^j \|_F^2$$

$\downarrow$ quadratic relaxation

$$\min_y \| Ay - b \|^2$$

$\downarrow$ normal equation

$$\underbrace{(A^T A)}_{\text{Hessian (H)}} y = \underbrace{A^T b}_{\text{g}}$$

$\downarrow$

$$Hy = g$$

First stage

Centralized Two-Stage Approach

$$\min_{\boldsymbol{R}_i \in \mathrm{SO}(3)} \sum_{(i,j)\in\mathcal{E}} \omega_R^2 \| R_j - R_i \bar{R}_i^j \|_F^2$$

$\downarrow$ quadratic relaxation

$$\min_y \|Ay - b\|^2$$

$\downarrow$ normal equation

$$\underbrace{\left(A^T A\right)}_{\text{Hessian (H)}} y = \underbrace{A^T b}_{\text{g}}$$

$\downarrow$

$$Hy = g$$

First stage

use the rotation estimates from the first stage as initialization

Centralized Two-Stage Approach

$$\min_{\boldsymbol{R}_i \in \mathrm{SO}(3)} \sum_{(i,j)\in\mathcal{E}} \omega_R^2 \|R_j - R_i \bar{R}_i^j\|_F^2$$

$\downarrow$ quadratic relaxation

$$\min_{y} \|Ay - b\|^2$$

$\downarrow$ normal equation

$$\underbrace{\left(A^T A\right)}_{\text{Hessian (H)}} y = \underbrace{A^T b}_{\text{g}}$$

$$\downarrow$$

$$Hy = g$$

First stage

use the rotation estimates from the first stage as initialization

$$\min_{t_i, R_i} \sum_{(i,j)\in\mathcal{E}} \omega_t^2 \|t_j - t_i - R_i \bar{t}_{ij}\|^2 + \omega_R^2 \|R_j - R_i \overline{R}_j^i\|_F^2$$

Centralized Two-Stage Approach

$$\min_{\boldsymbol{R}_i \in \mathrm{SO}(3)} \sum_{(i,j)\in\mathcal{E}} \omega_R^2 \|R_j - R_i \bar{R}_i^j\|_F^2$$

$\downarrow$ quadratic relaxation

$$\min_y \|Ay - b\|^2$$

$\downarrow$ normal equation

$$\underbrace{\left(A^T A\right)}_{\text{Hessian (H)}} y = \underbrace{A^T b}_{\text{g}}$$

$\downarrow$

$$Hy = g$$

First stage

use the rotation estimates from the first stage as initialization

$$\min_{t_i, R_i} \sum_{(i,j)\in\mathcal{E}} \omega_t^2 \|t_j - t_i - R_i \bar{t}_{ij}\|^2 + \omega_R^2 \|R_j - R_i \overline{R}_j^i\|_F^2$$

$\downarrow$ linearize

$$\min_y \|Ay - b\|^2$$

$\downarrow$ solve

$$\underbrace{\left(A^T A\right)}_{\text{Hessian (H)}} y = \underbrace{A^T b}_{\text{g}}$$

$\downarrow$

$$Hy = g$$

Centralized Two-Stage Approach

$$\min_{\boldsymbol{R}_i \in \mathrm{SO}(3)} \sum_{(i,j)\in\mathcal{E}} \omega_R^2 \|R_j - R_i \bar{R}_i^j\|_F^2$$

$\downarrow$ quadratic relaxation

$$\min_y \|Ay - b\|^2$$

$\downarrow$ normal equation

$$\underbrace{\left(A^T A\right)}_{\text{Hessian (H)}} y = \underbrace{A^T b}_{\text{g}}$$

$\downarrow$

$$Hy = g$$

First stage

use the rotation estimates from the first stage as initialization

$$\min_{t_i, R_i} \sum_{(i,j)\in\mathcal{E}} \omega_t^2 \|t_j - t_i - R_i \bar{t}_{ij}\|^2 + \omega_R^2 \|R_j - R_i \overline{R}_j^i\|_F^2$$

$\downarrow$ linearize

$$\min_y \|Ay - b\|^2$$

$\downarrow$ solve

$$\underbrace{\left(A^T A\right)}_{\text{Hessian (H)}} y = \underbrace{A^T b}_{\text{g}}$$

$\downarrow$

$$Hy = g$$

Second stage

Centralized Two-Stage Approach

$$\min_{\boldsymbol{R}_i \in \mathrm{SO}(3)} \sum_{(i,j)\in\mathcal{E}} \omega_R^2 \|R_j - R_i \bar{R}_i^j\|_F^2$$

$\downarrow$ quadratic relaxation

$$\min_y \|Ay - b\|^2$$

$\downarrow$ normal equation

$$\underbrace{\left(A^T A\right)}_{\text{Hessian (H)}} y = \underbrace{A^T b}_{\text{g}}$$

$$\boxed{Hy = g}$$

solve in a distributed manner

First stage

use the rotation estimates from the first stage as initialization

$$\min_{t_i, R_i} \sum_{(i,j)\in\mathcal{E}} \omega_t^2 \|t_j - t_i - R_i \bar{t}_{ij}\|^2 + \omega_R^2 \|R_j - R_i \overline{R}_j^i\|_F^2$$

$\downarrow$ linearize

$$\min_y \|Ay - b\|^2$$

$\downarrow$ solve

$$\underbrace{\left(A^T A\right)}_{\text{Hessian (H)}} y = \underbrace{A^T b}_{\text{g}}$$

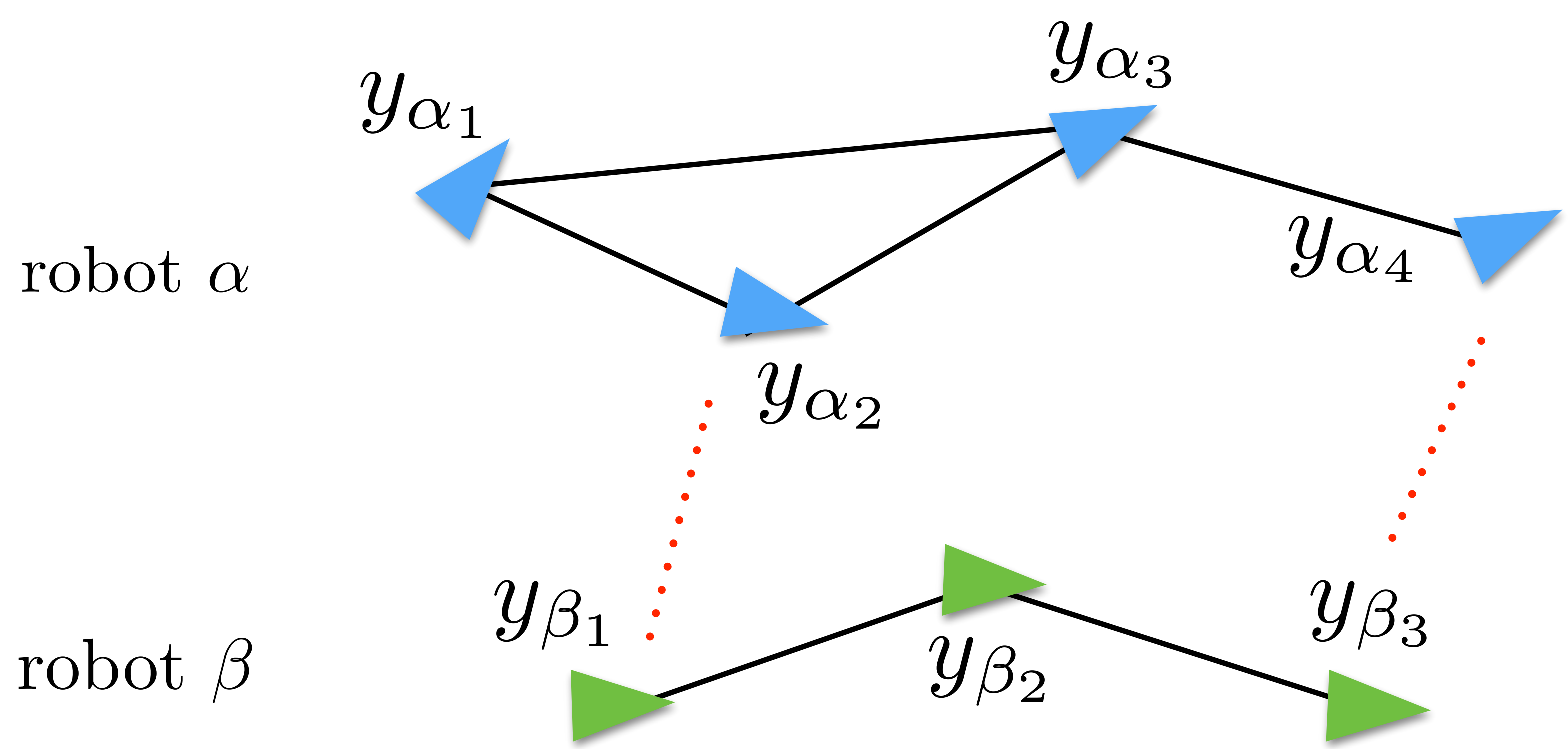
$$\boxed{Hy = g}$$

solve in a distributed manner

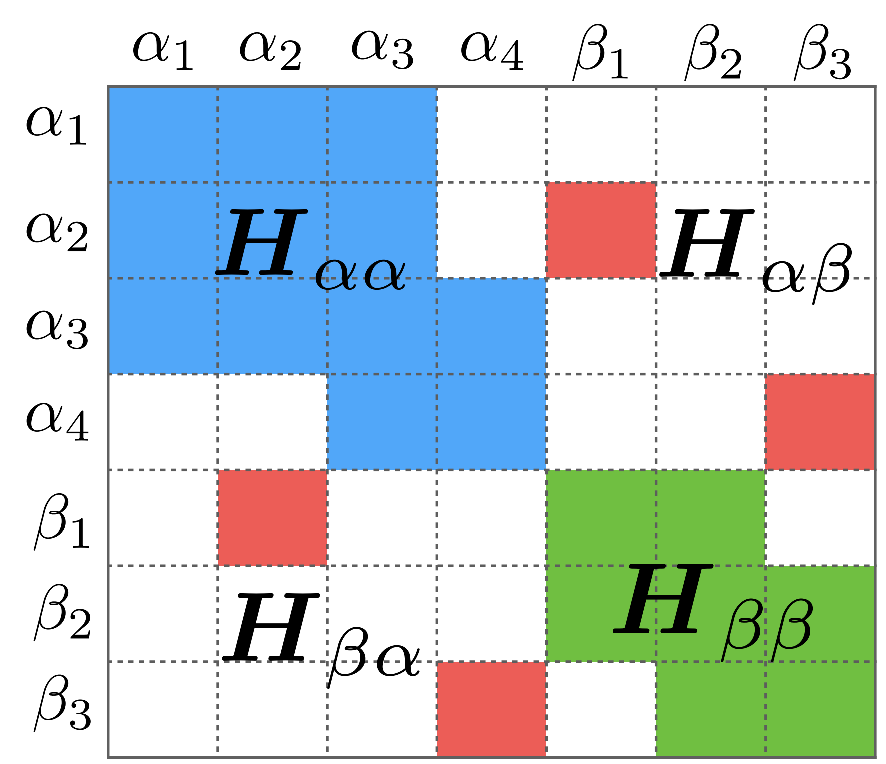
Second stage

Distributed Gauss-Seidel Approach

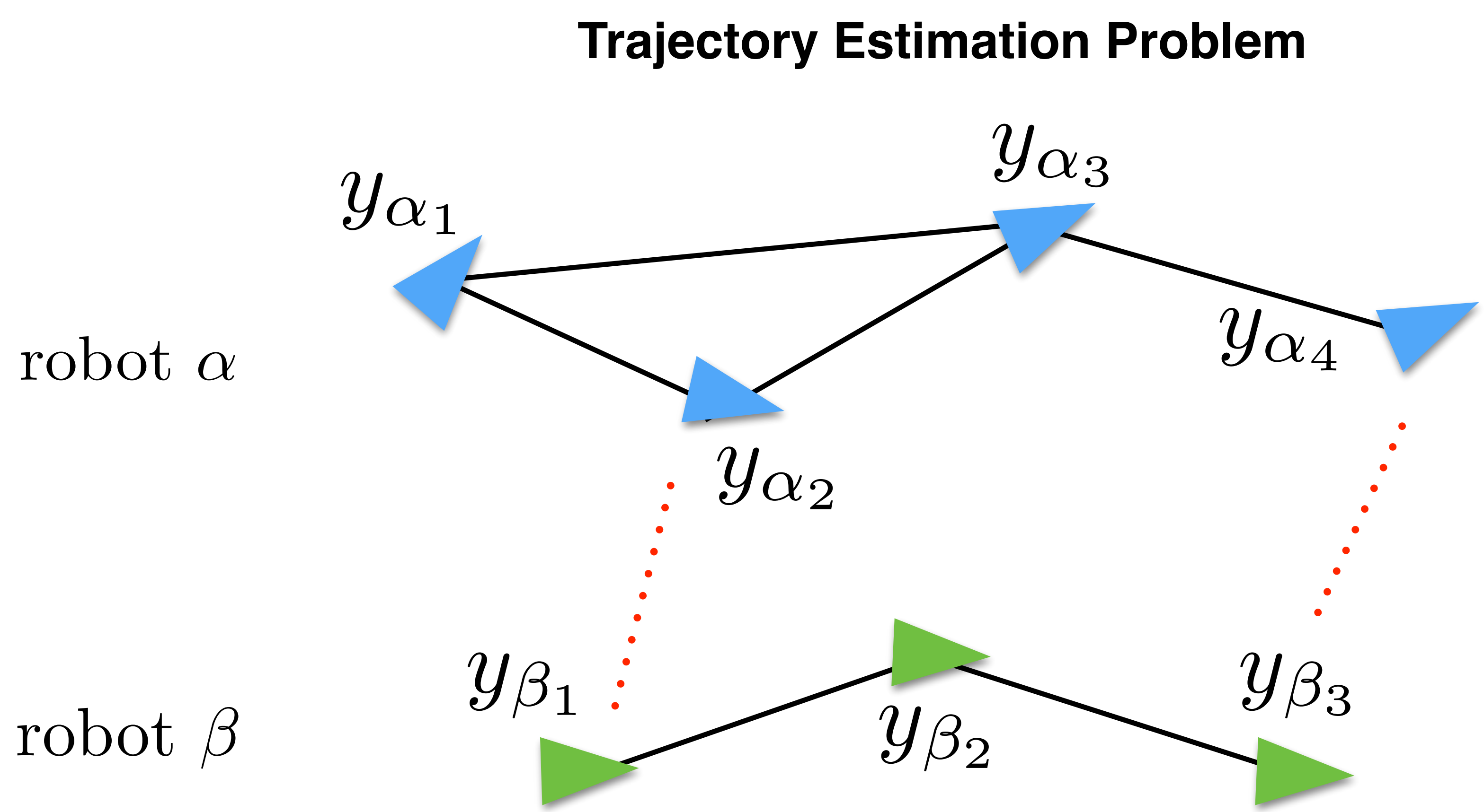
Trajectory Estimation Problem

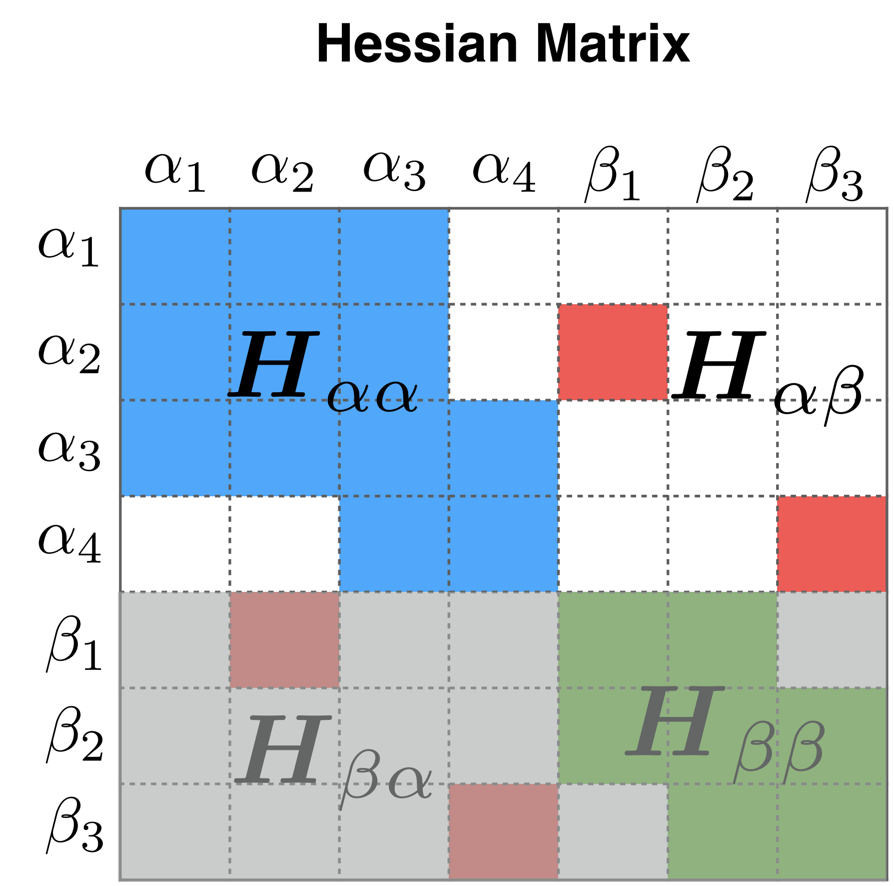


Hessian Matrix



Distributed Gauss-Seidel Approach

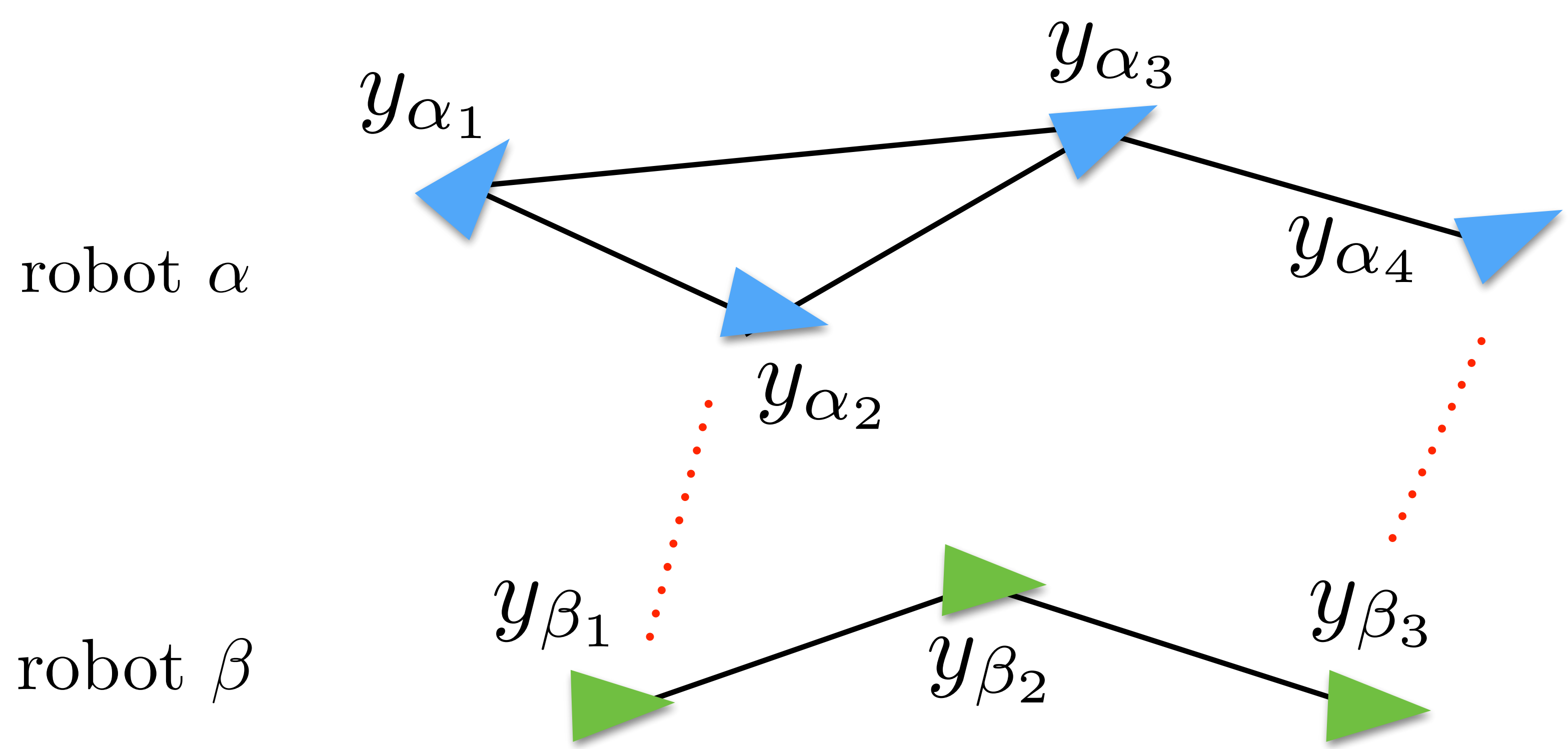




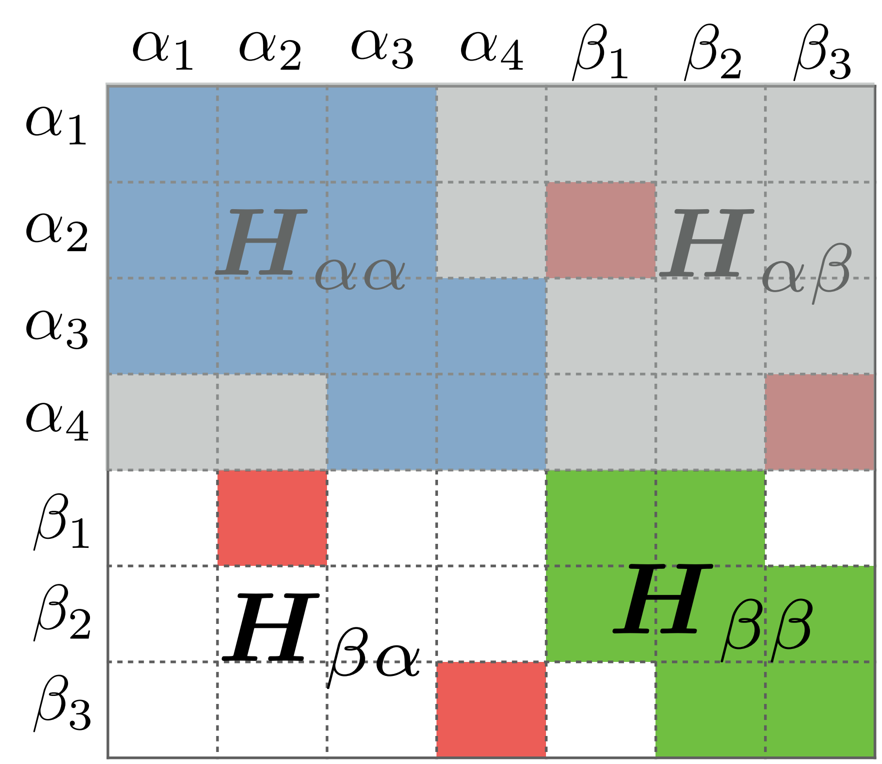
$$\boldsymbol{y}_{\alpha}^{k+1} = \boldsymbol{H}_{\alpha\alpha}^{-1}\left(\boxed{-\boldsymbol{H}_{\alpha\beta}}\,\boldsymbol{y}_{\beta}^{k} + \boldsymbol{g}_{\alpha}\right)$$

Distributed Gauss-Seidel Approach

Trajectory Estimation Problem



Hessian Matrix

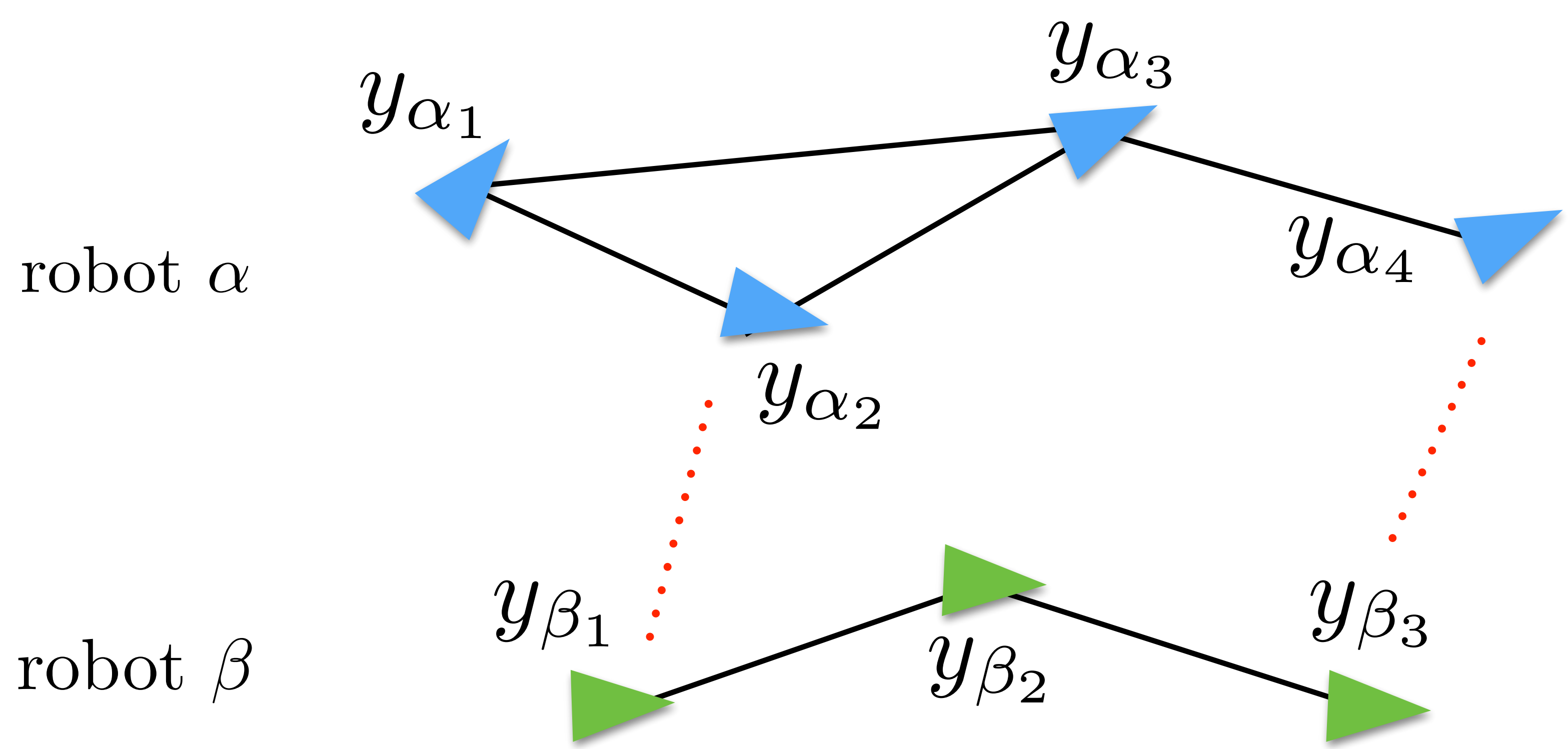


$$\boldsymbol{y}_\alpha^{k+1} = \boldsymbol{H}_{\alpha\alpha}^{-1}\left(\boxed{-\boldsymbol{H}_{\alpha\beta}}\,\boldsymbol{y}_\beta^k + \boldsymbol{g}_\alpha\right)$$

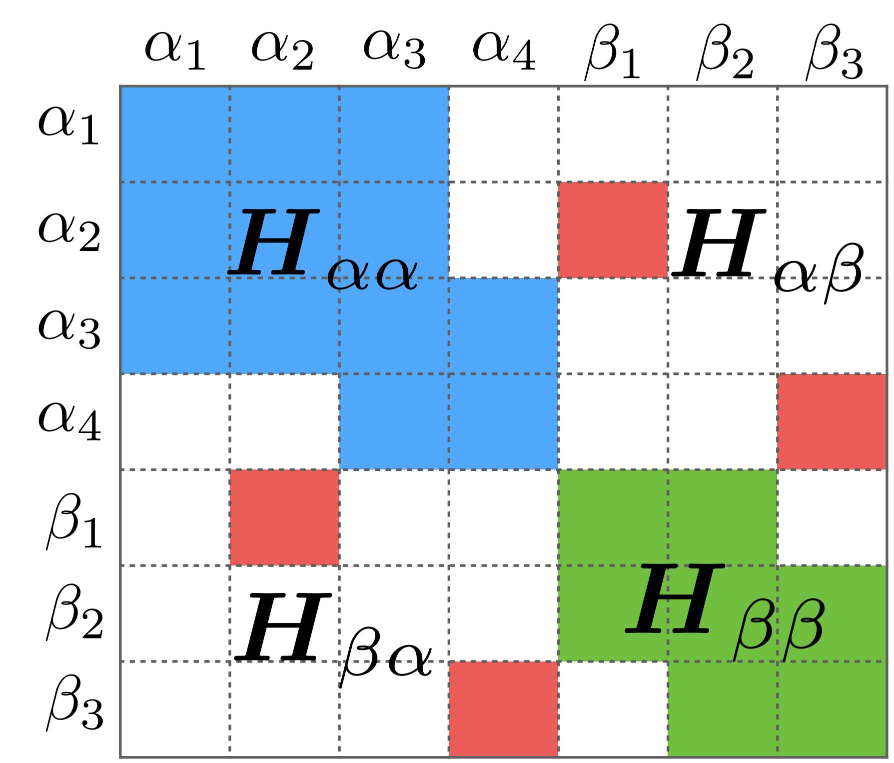
$$\boldsymbol{y}_\beta^{k+1} = \boldsymbol{H}_{\beta\beta}^{-1}\left(\boxed{-\boldsymbol{H}_{\beta\alpha}}\,\boldsymbol{y}_\alpha^k + \boldsymbol{g}_\beta\right)$$

Distributed Gauss-Seidel Approach

Trajectory Estimation Problem



Hessian Matrix



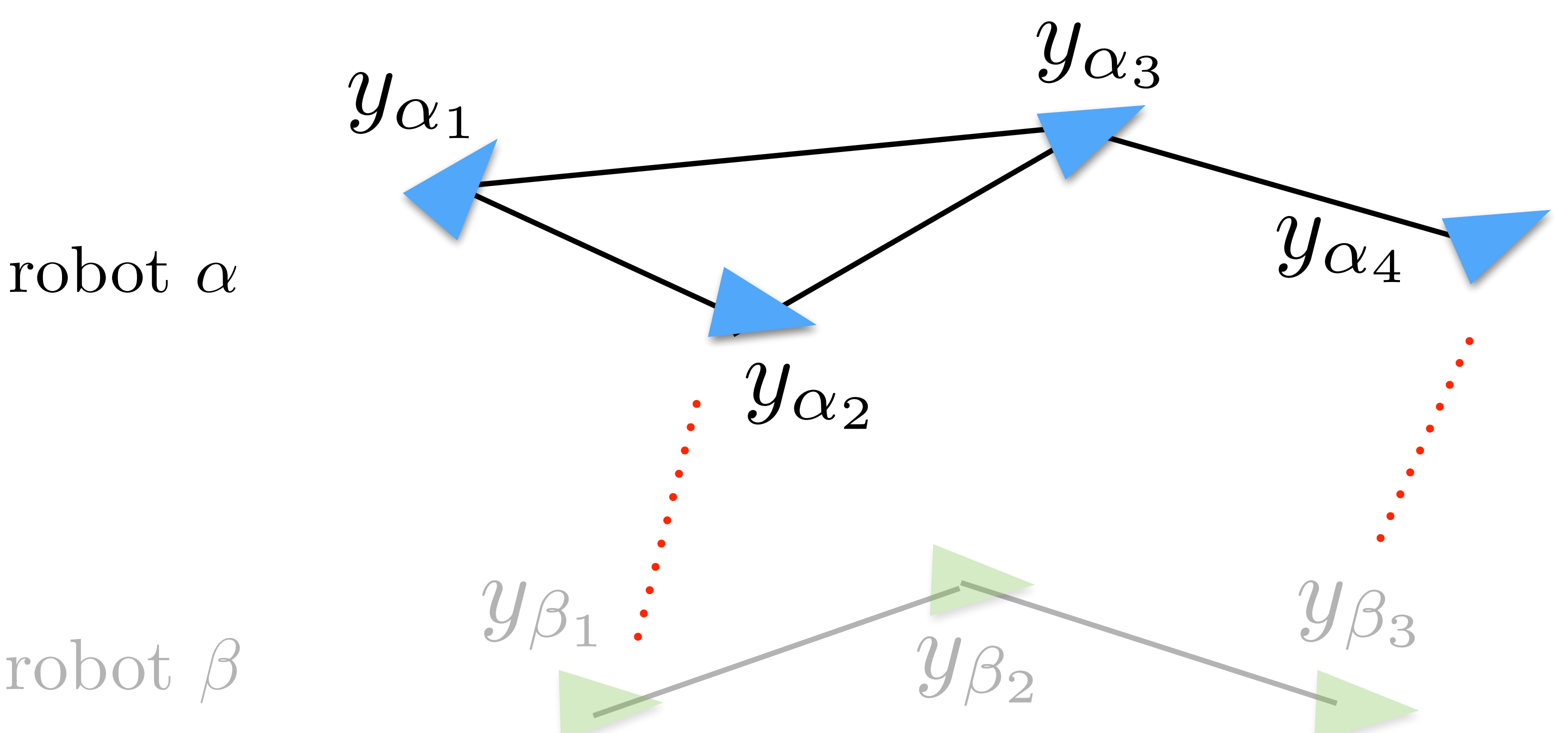
Iterate

$$\boldsymbol{y}_{\alpha}^{k+1} = \boldsymbol{H}_{\alpha\alpha}^{-1}\left(\boxed{-\boldsymbol{H}_{\alpha\beta}}\,\boldsymbol{y}_{\beta}^{k} + \boldsymbol{g}_{\alpha}\right)$$

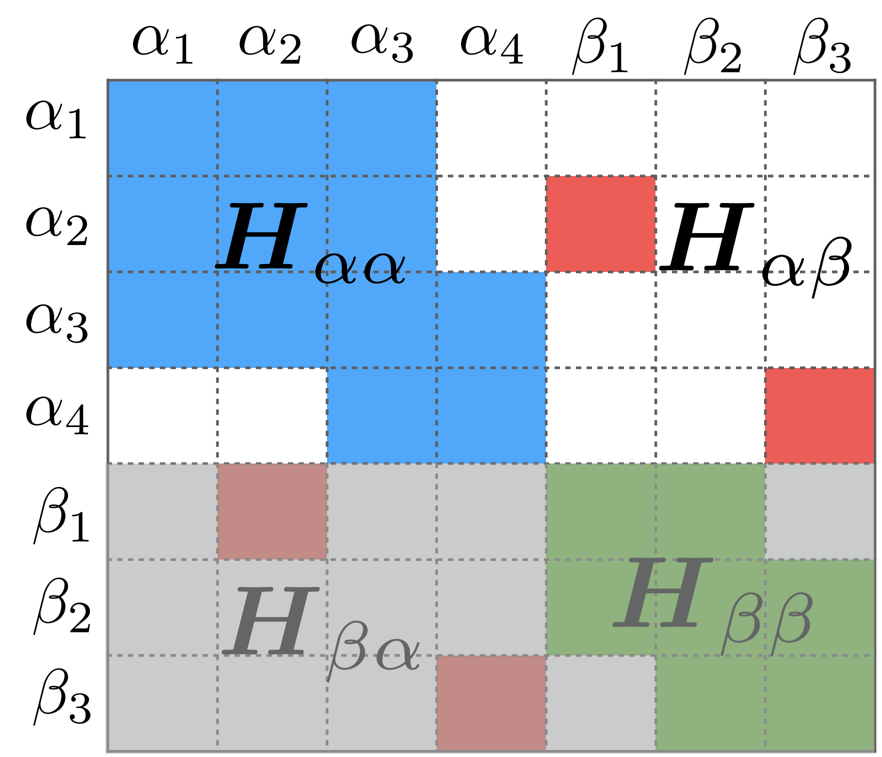
$$\boldsymbol{y}_{\beta}^{k+1} = \boldsymbol{H}_{\beta\beta}^{-1}\left(\boxed{-\boldsymbol{H}_{\beta\alpha}}\,\boldsymbol{y}_{\alpha}^{k} + \boldsymbol{g}_{\beta}\right)$$

Distributed Gauss-Seidel Approach

Trajectory Estimation Problem

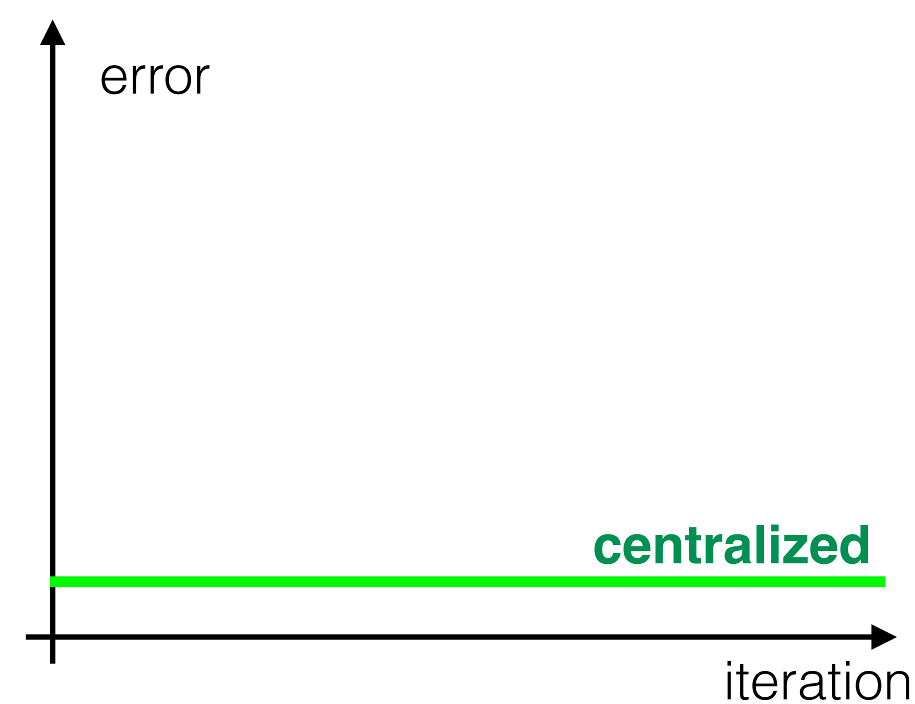


Hessian Matrix



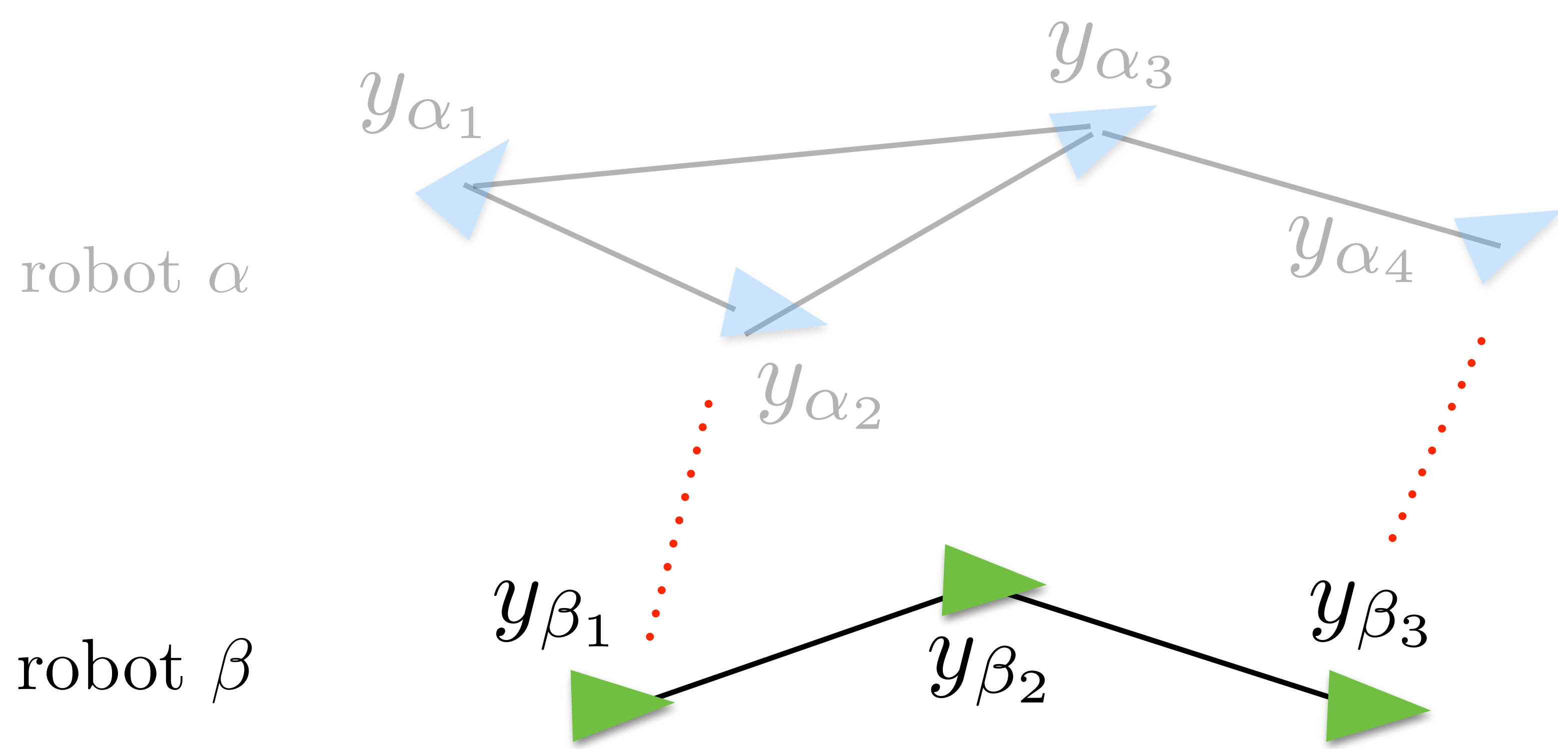
$$\boldsymbol{y}_{\alpha}^{k+1} = \boldsymbol{H}_{\alpha\alpha}^{-1}\left(\boxed{-\boldsymbol{H}_{\alpha\beta}}\,\boldsymbol{y}_{\beta}^{k} + \boldsymbol{g}_{\alpha}\right)$$

$$\boldsymbol{y}_{\beta}^{k+1} = \boldsymbol{H}_{\beta\beta}^{-1}\left(\boxed{-\boldsymbol{H}_{\beta\alpha}}\,\boldsymbol{y}_{\alpha}^{k} + \boldsymbol{g}_{\beta}\right)$$

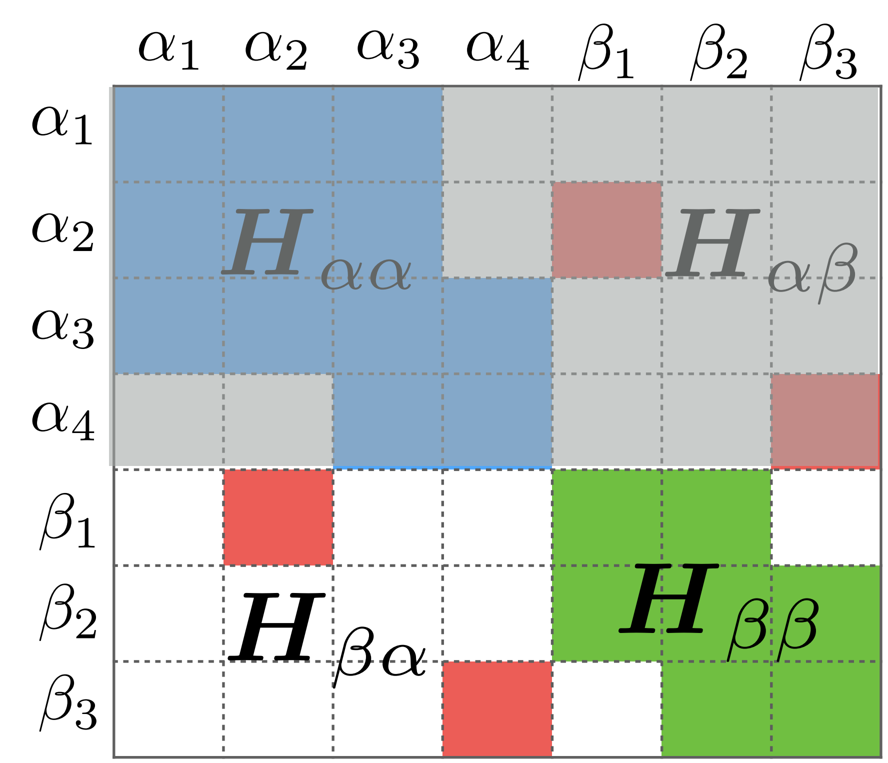


Distributed Gauss-Seidel Approach

Trajectory Estimation Problem

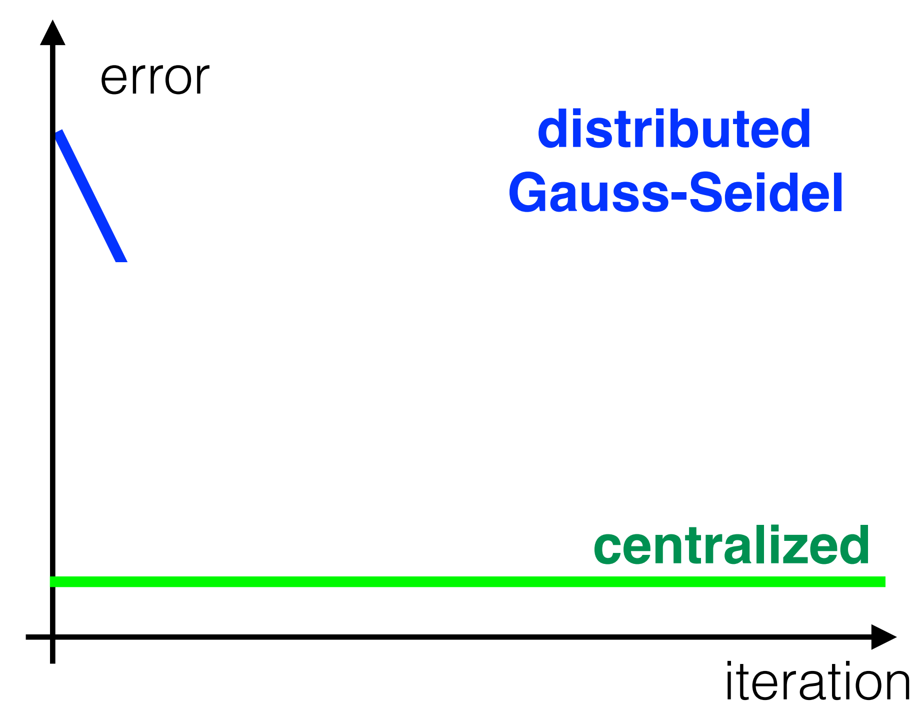


Hessian Matrix



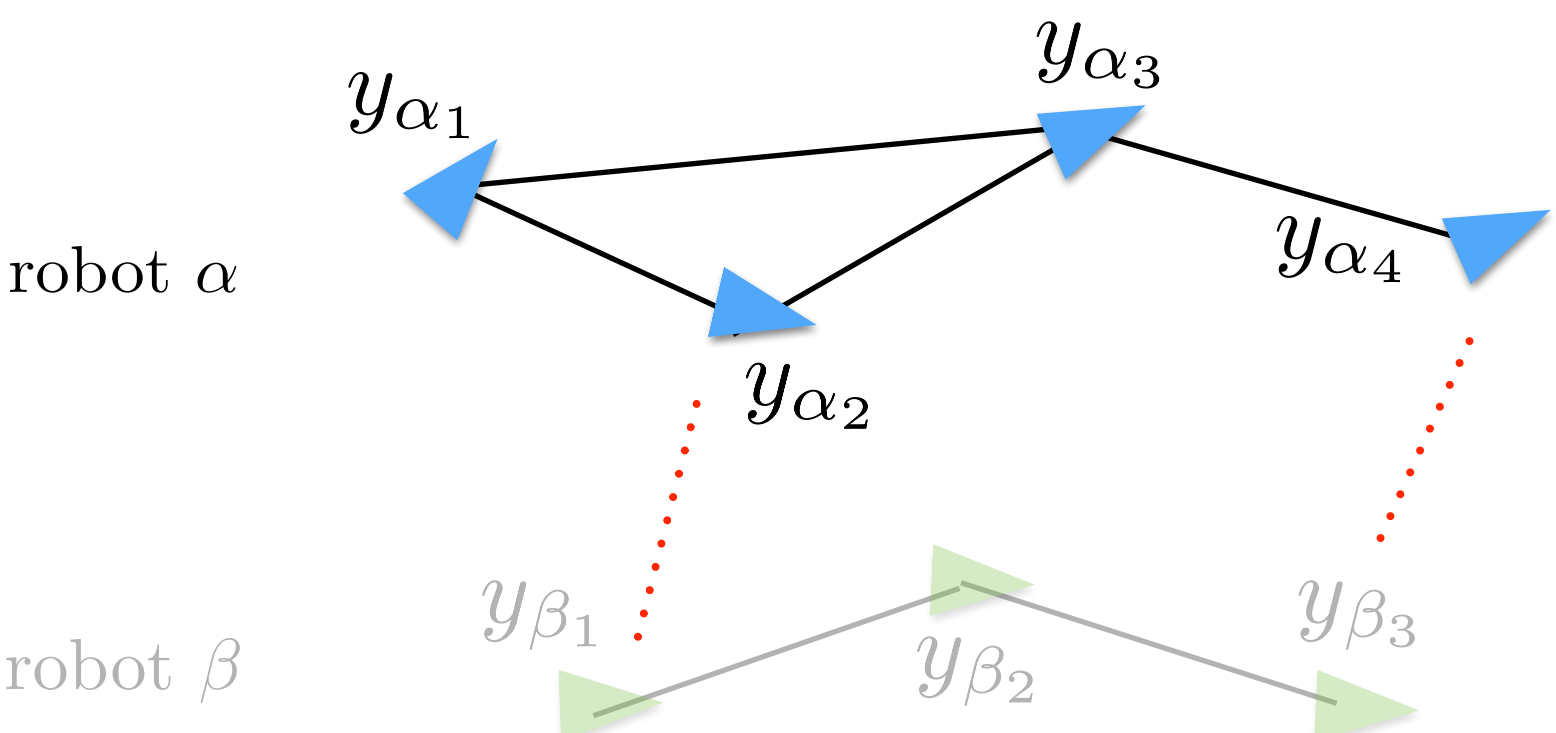
$$\boldsymbol{y}_{\alpha}^{k+1} = \boldsymbol{H}_{\alpha\alpha}^{-1}\left(\boxed{-\boldsymbol{H}_{\alpha\beta}}\,\boldsymbol{y}_{\beta}^{k} + \boldsymbol{g}_{\alpha}\right)$$

$$\boldsymbol{y}_{\beta}^{k+1} = \boldsymbol{H}_{\beta\beta}^{-1}\left(\boxed{-\boldsymbol{H}_{\beta\alpha}}\,\boldsymbol{y}_{\alpha}^{k} + \boldsymbol{g}_{\beta}\right)$$

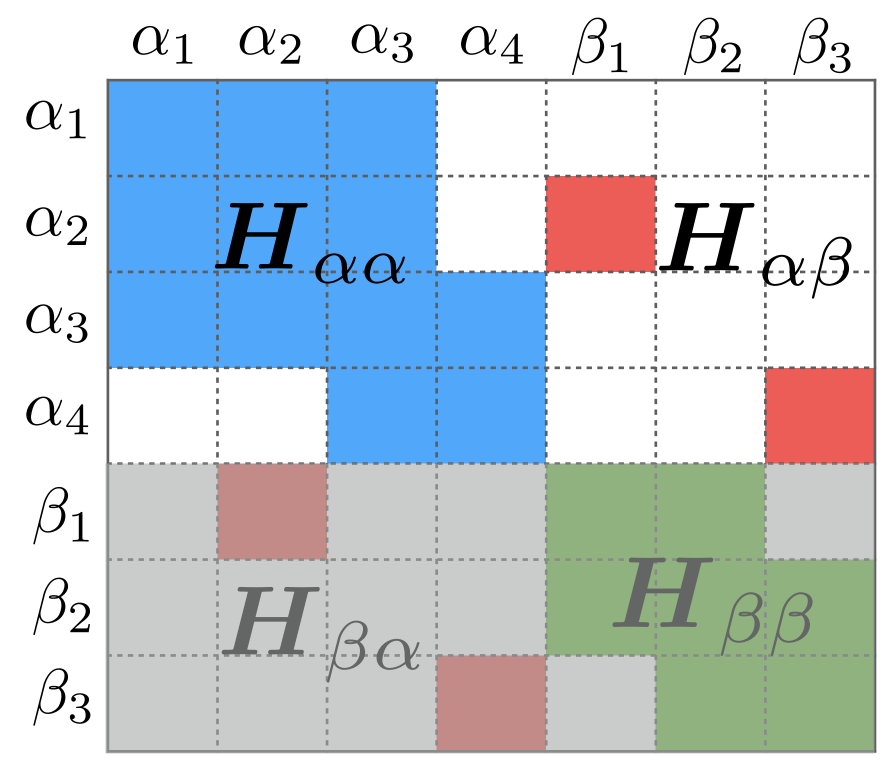


Distributed Gauss-Seidel Approach

Trajectory Estimation Problem

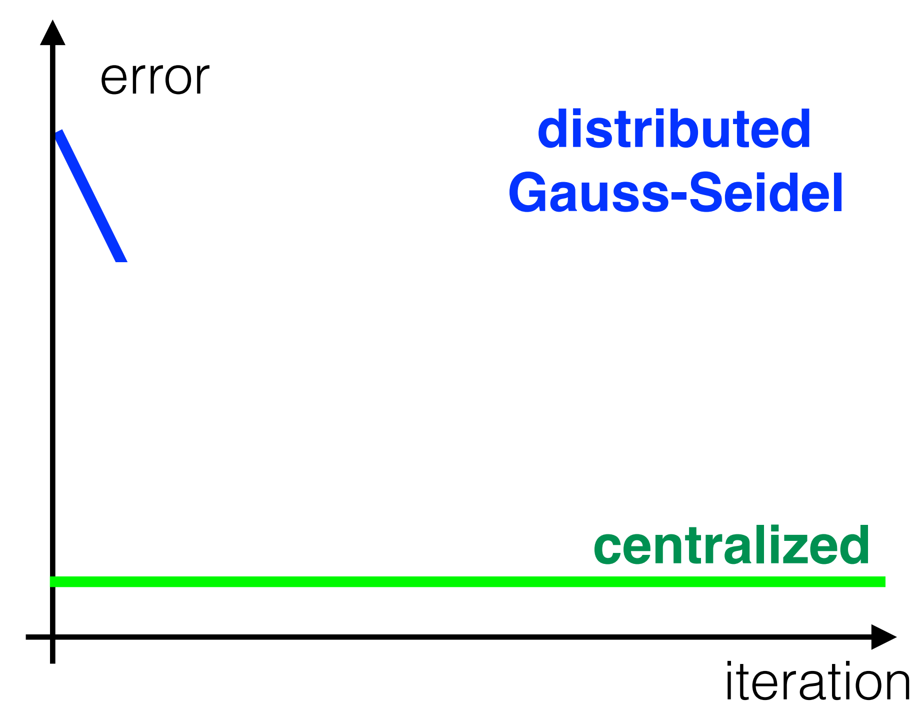


Hessian Matrix



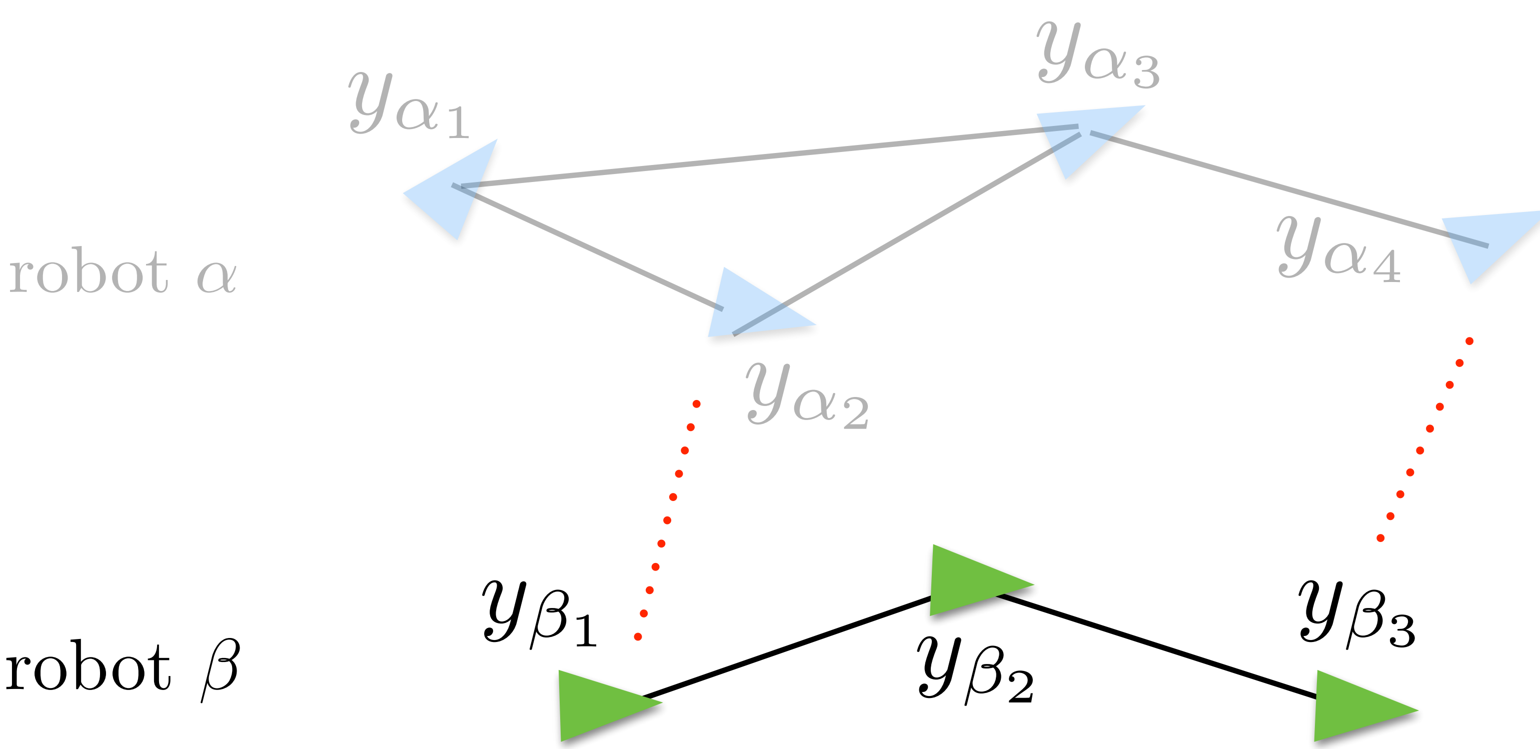
$$\boldsymbol{y}_\alpha^{k+1} = \boldsymbol{H}_{\alpha\alpha}^{-1}\left(\boxed{-\boldsymbol{H}_{\alpha\beta}}\,\boldsymbol{y}_\beta^k + \boldsymbol{g}_\alpha\right)$$

$$\boldsymbol{y}_\beta^{k+1} = \boldsymbol{H}_{\beta\beta}^{-1}\left(\boxed{-\boldsymbol{H}_{\beta\alpha}}\,\boldsymbol{y}_\alpha^k + \boldsymbol{g}_\beta\right)$$

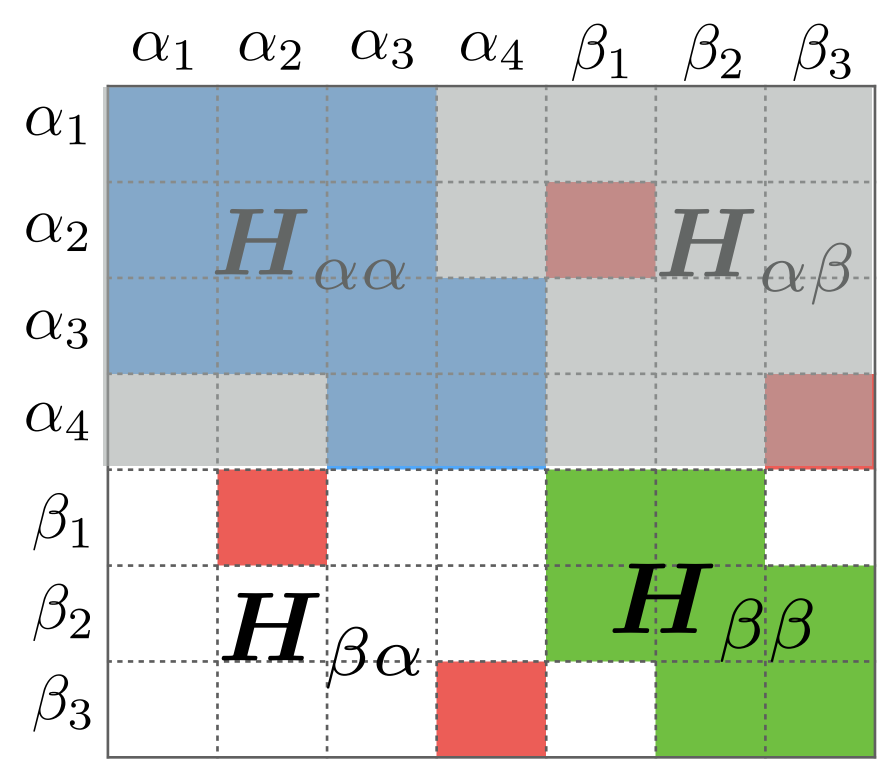


Distributed Gauss-Seidel Approach

Trajectory Estimation Problem

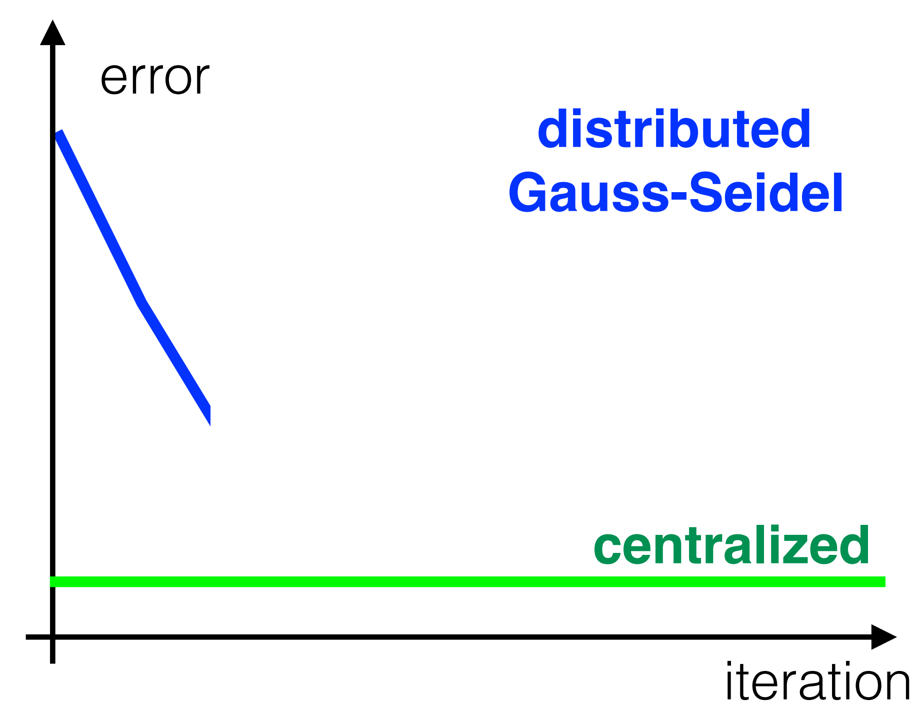


Hessian Matrix



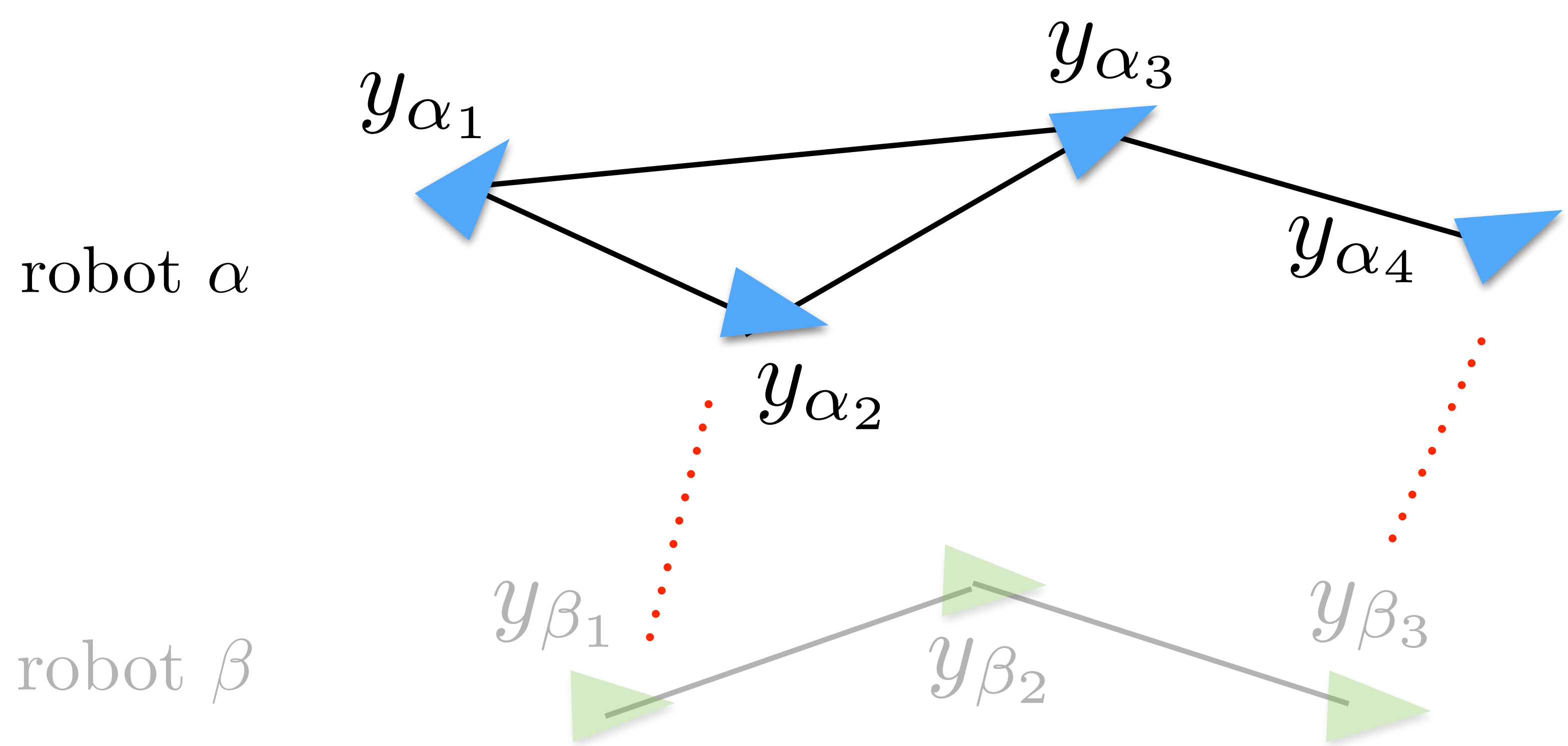
$$\boldsymbol{y}_\alpha^{k+1} = \boldsymbol{H}_{\alpha\alpha}^{-1}\left(\boxed{-\boldsymbol{H}_{\alpha\beta}}\,\boldsymbol{y}_\beta^k + \boldsymbol{g}_\alpha\right)$$

$$\boldsymbol{y}_\beta^{k+1} = \boldsymbol{H}_{\beta\beta}^{-1}\left(\boxed{-\boldsymbol{H}_{\beta\alpha}}\,\boldsymbol{y}_\alpha^k + \boldsymbol{g}_\beta\right)$$

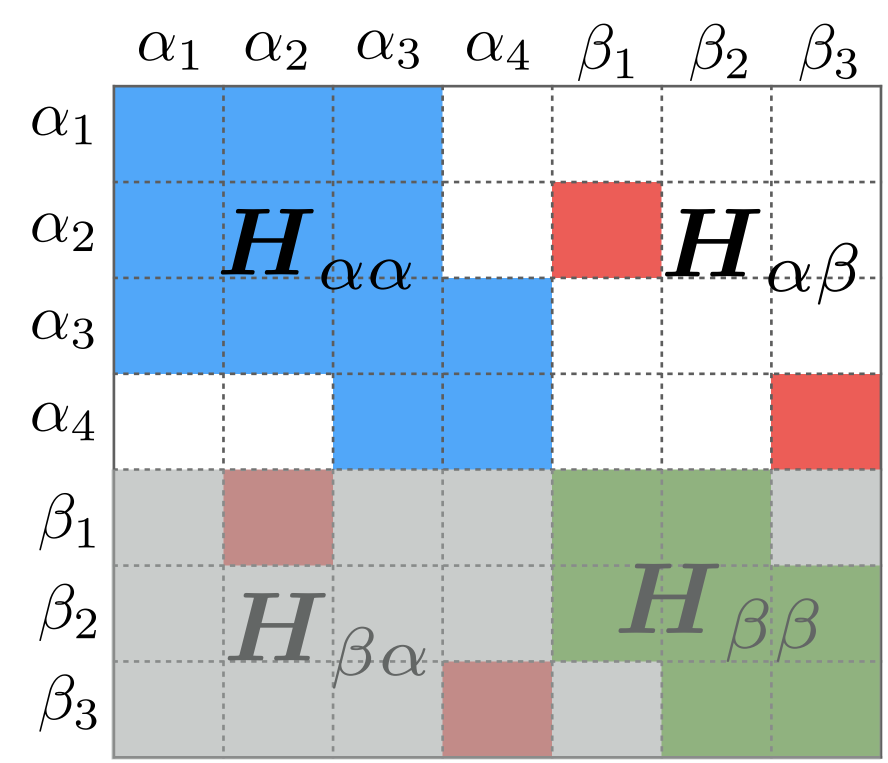


Distributed Gauss-Seidel Approach

Trajectory Estimation Problem

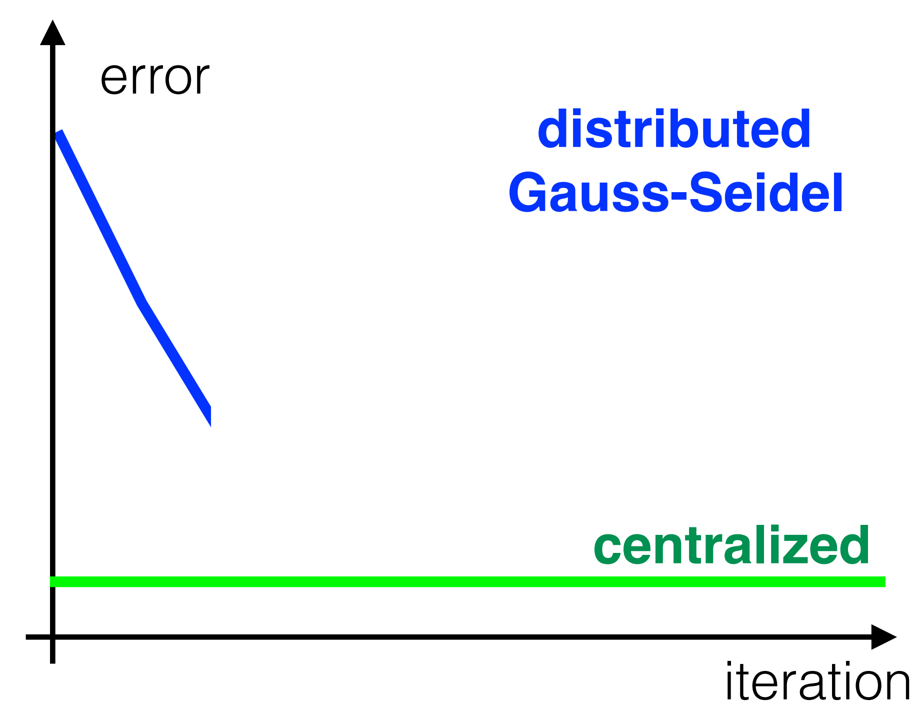


Hessian Matrix



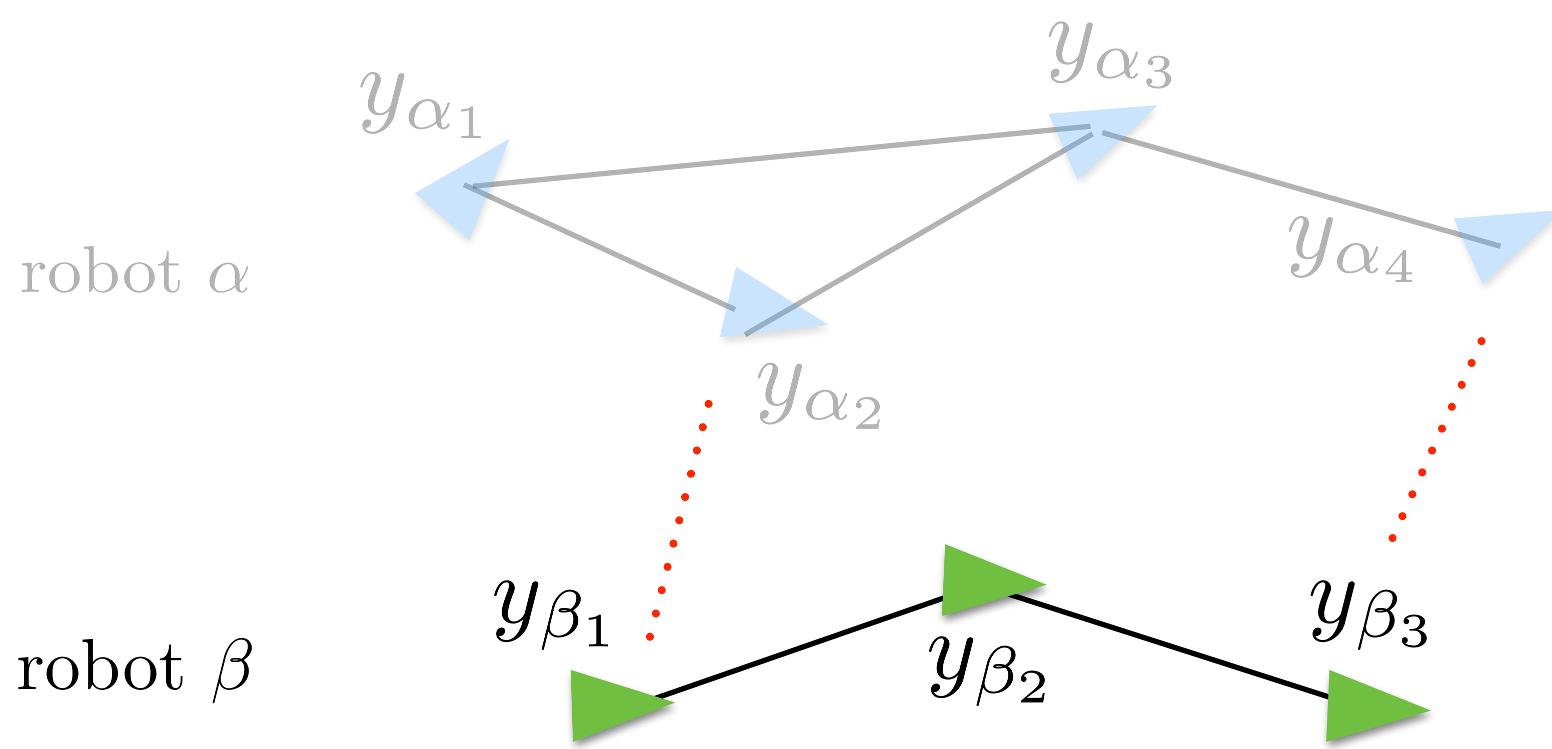
$$\boldsymbol{y}_{\alpha}^{k+1} = \boldsymbol{H}_{\alpha\alpha}^{-1} \left(\boxed{-\boldsymbol{H}_{\alpha\beta}} \boldsymbol{y}_{\beta}^{k} + \boldsymbol{g}_{\alpha} \right)$$

$$\boldsymbol{y}_{\beta}^{k+1} = \boldsymbol{H}_{\beta\beta}^{-1} \left(\boxed{-\boldsymbol{H}_{\beta\alpha}} \boldsymbol{y}_{\alpha}^{k} + \boldsymbol{g}_{\beta} \right)$$

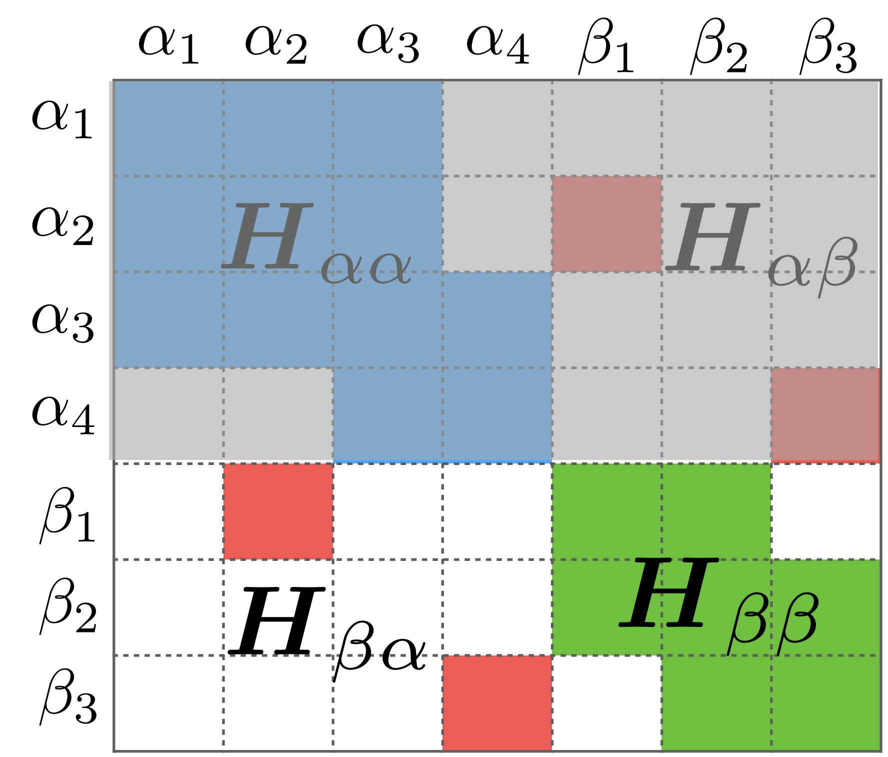


Distributed Gauss-Seidel Approach

Trajectory Estimation Problem

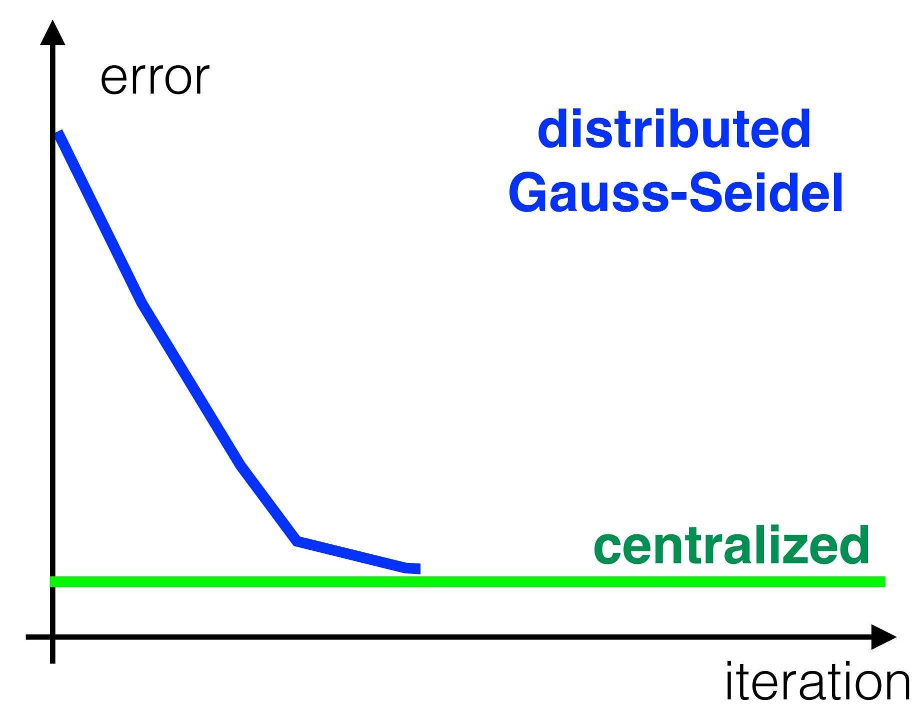


Hessian Matrix



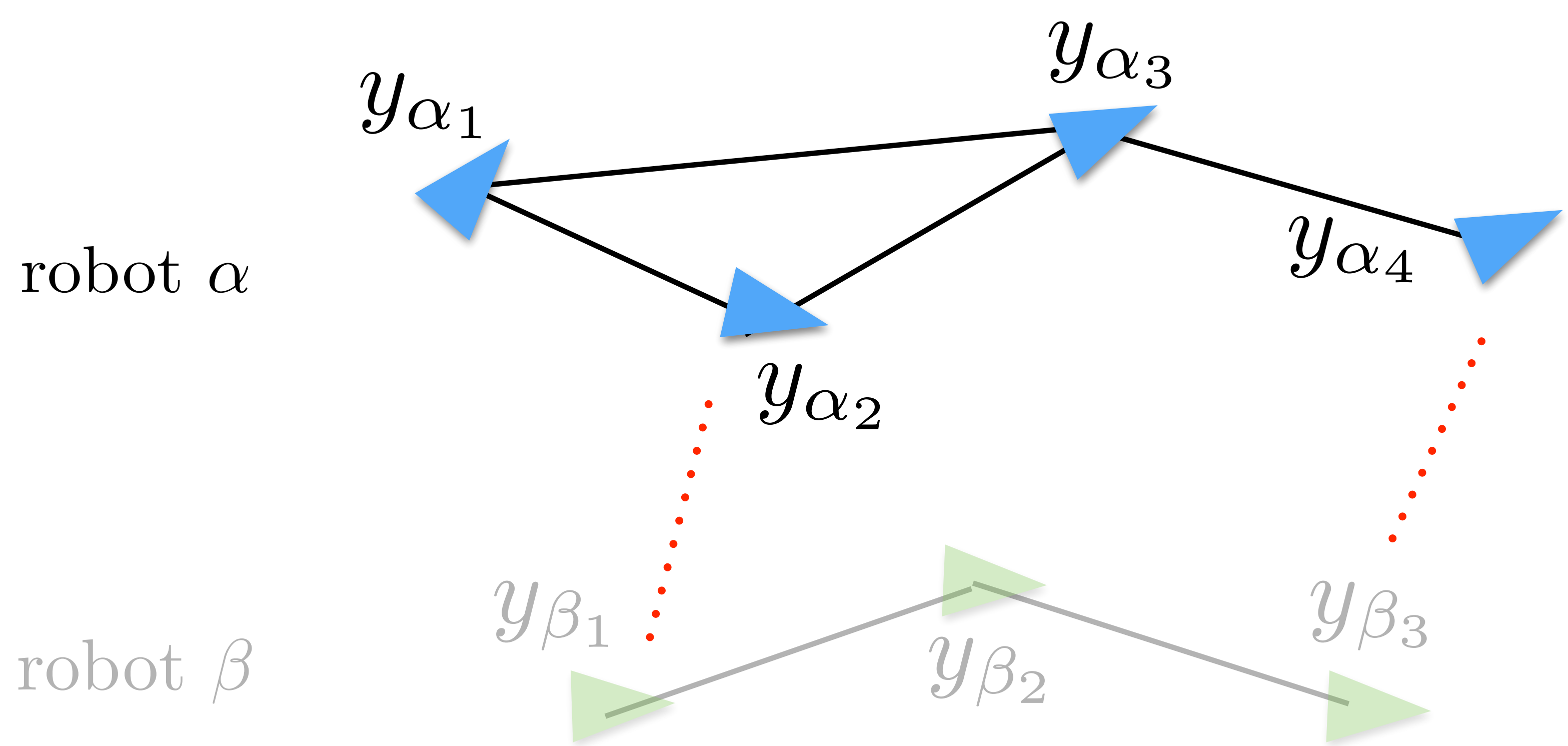
$$\boldsymbol{y}_\alpha^{k+1} = \boldsymbol{H}_{\alpha\alpha}^{-1}\left(\boxed{-\boldsymbol{H}_{\alpha\beta}}\,\boldsymbol{y}_\beta^k + \boldsymbol{g}_\alpha\right)$$

$$\boldsymbol{y}_\beta^{k+1} = \boldsymbol{H}_{\beta\beta}^{-1}\left(\boxed{-\boldsymbol{H}_{\beta\alpha}}\,\boldsymbol{y}_\alpha^k + \boldsymbol{g}_\beta\right)$$

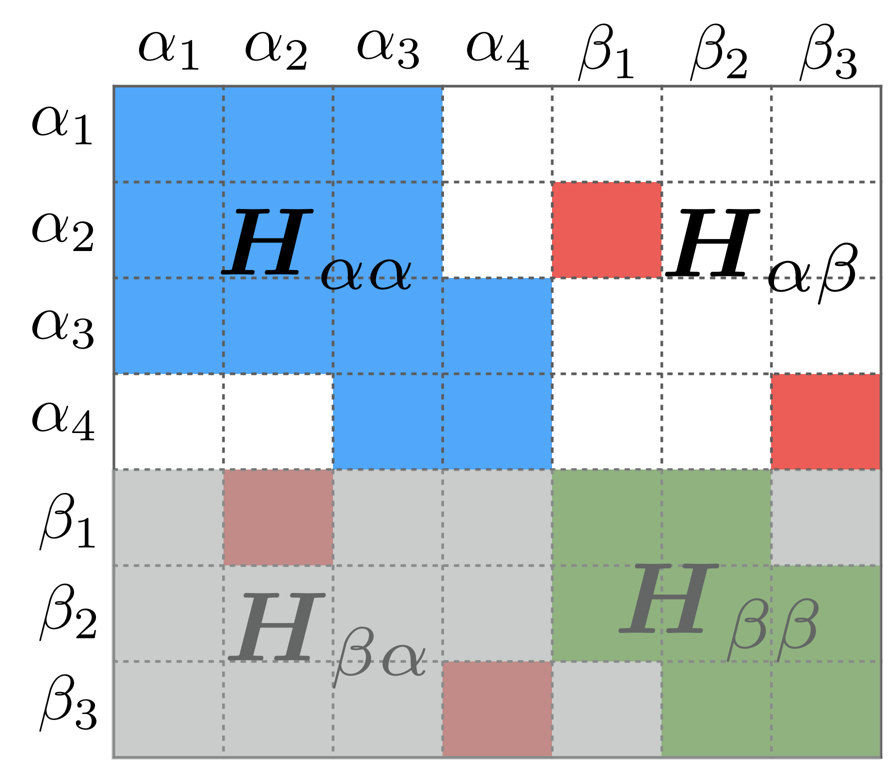


Distributed Gauss-Seidel Approach

Trajectory Estimation Problem

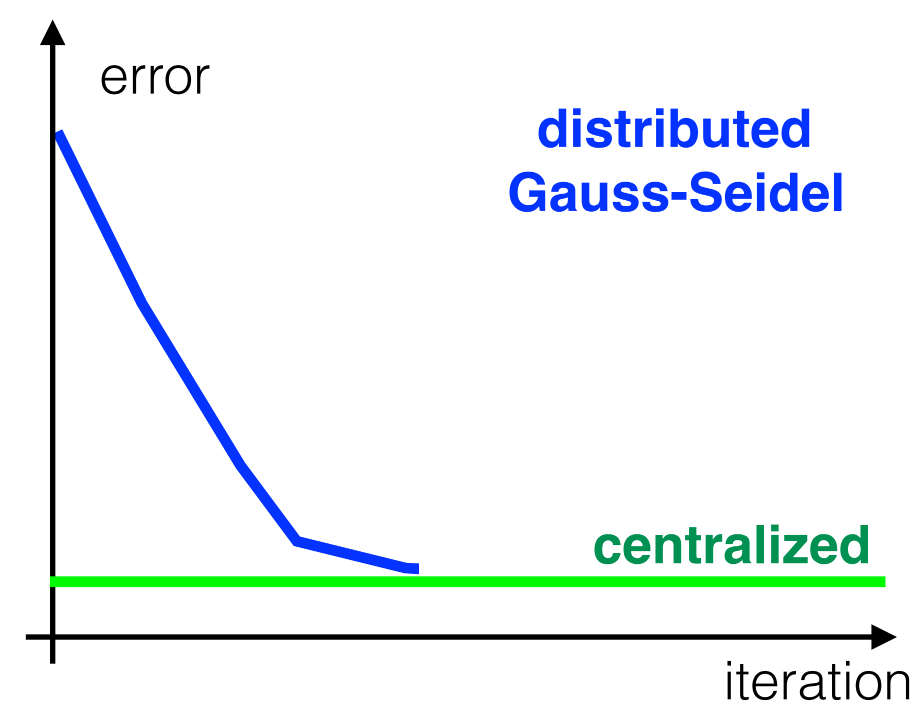


Hessian Matrix



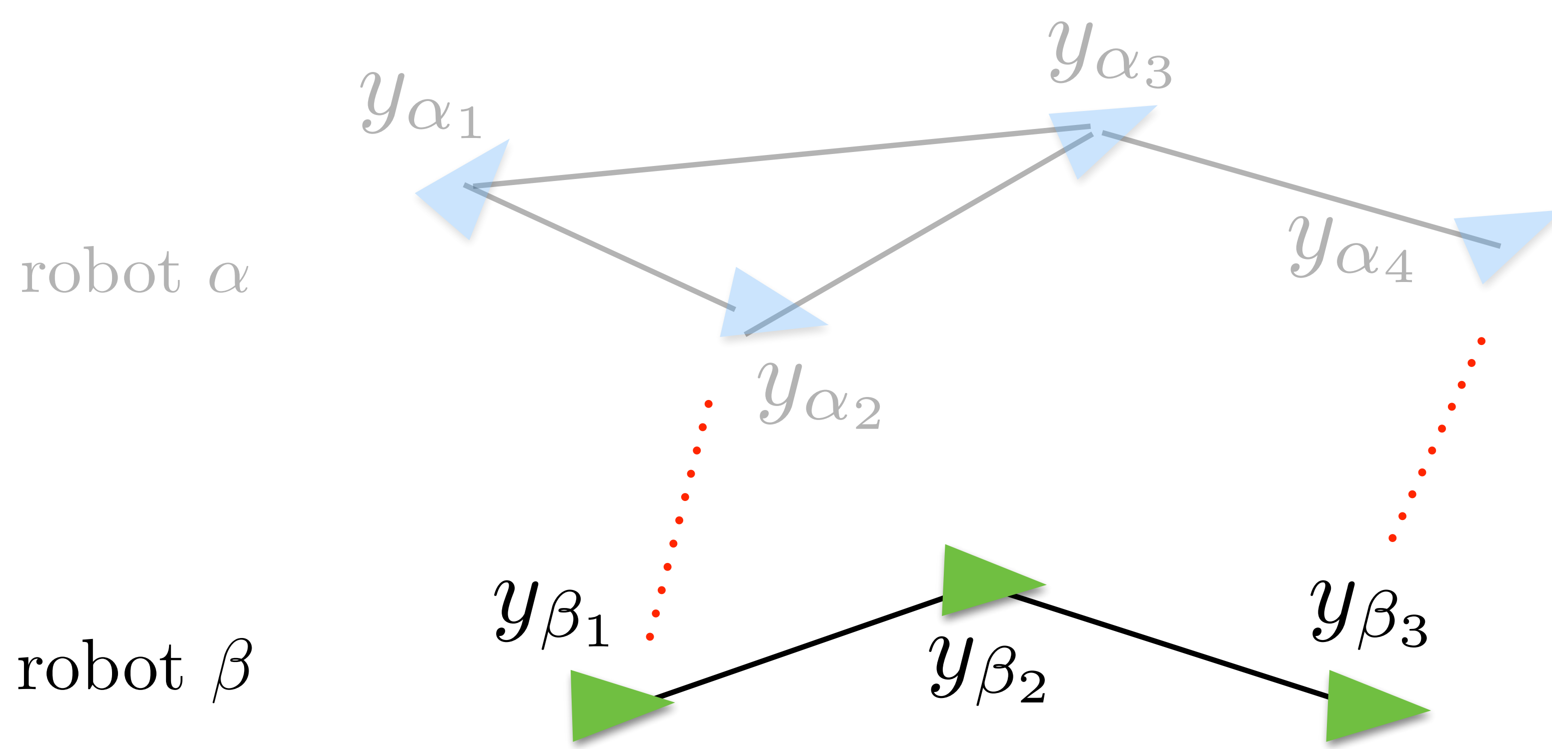
$$\boldsymbol{y}_{\alpha}^{k+1} = \boldsymbol{H}_{\alpha\alpha}^{-1}\left(\boxed{-\boldsymbol{H}_{\alpha\beta}}\,\boldsymbol{y}_{\beta}^{k} + \boldsymbol{g}_{\alpha}\right)$$

$$\boldsymbol{y}_{\beta}^{k+1} = \boldsymbol{H}_{\beta\beta}^{-1}\left(\boxed{-\boldsymbol{H}_{\beta\alpha}}\,\boldsymbol{y}_{\alpha}^{k} + \boldsymbol{g}_{\beta}\right)$$

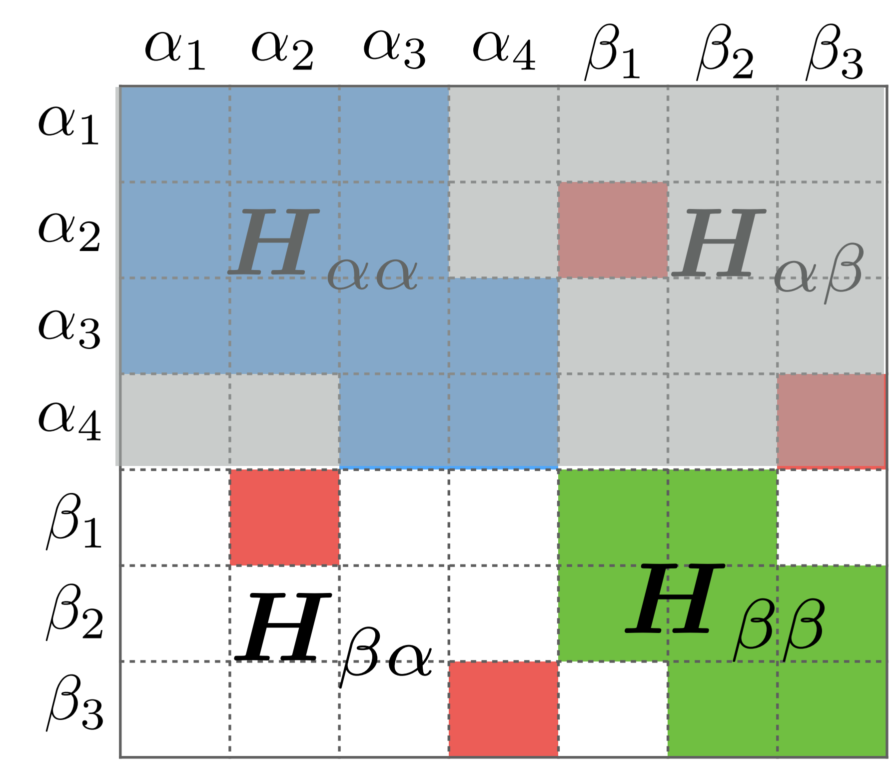


Distributed Gauss-Seidel Approach

Trajectory Estimation Problem

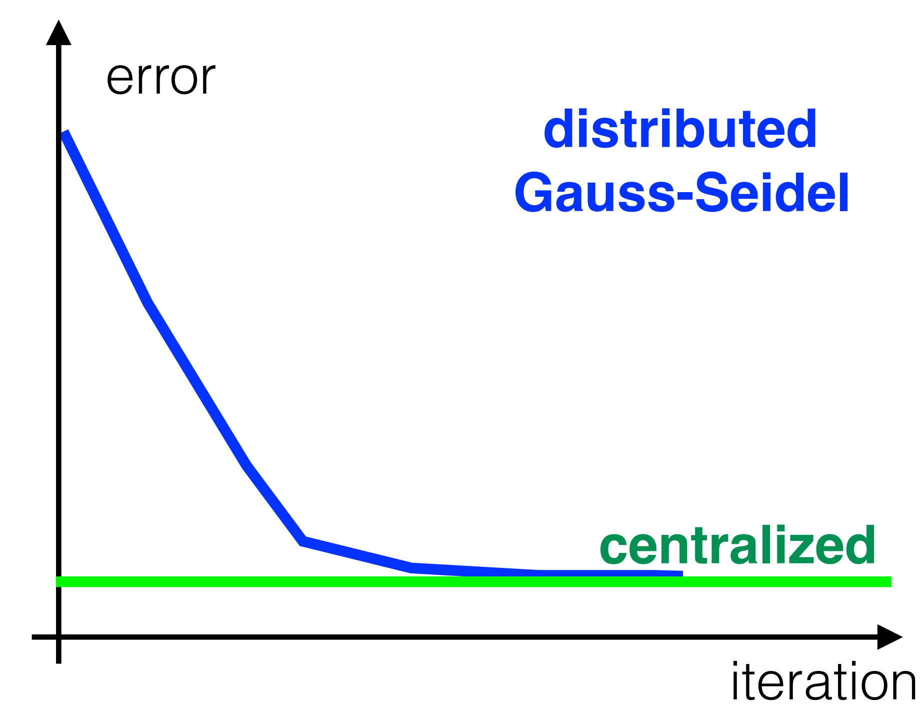


Hessian Matrix



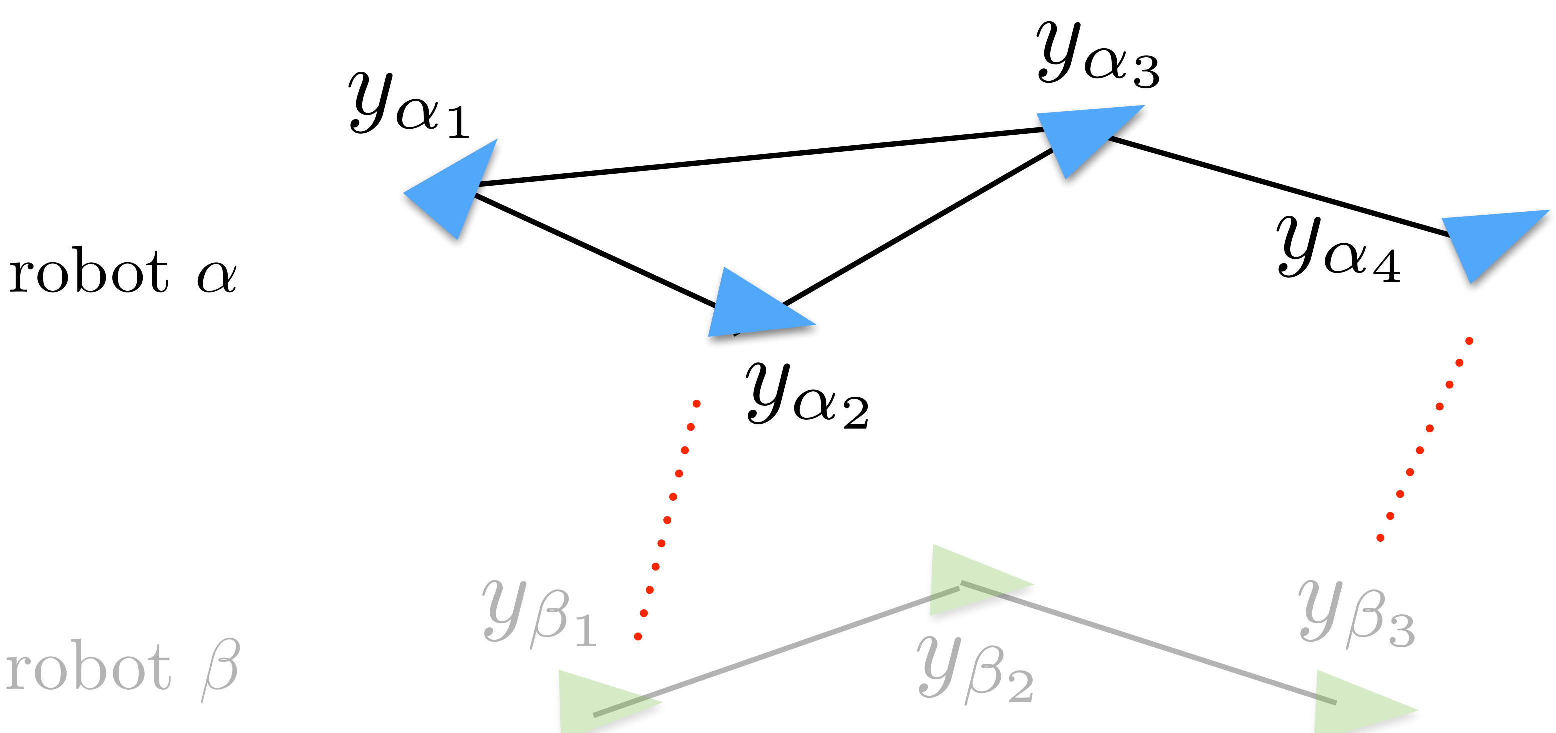
$$\boldsymbol{y}_{\alpha}^{k+1} = \boldsymbol{H}_{\alpha\alpha}^{-1}\left(\boxed{-\boldsymbol{H}_{\alpha\beta}}\,\boldsymbol{y}_{\beta}^{k} + \boldsymbol{g}_{\alpha}\right)$$

$$\boldsymbol{y}_{\beta}^{k+1} = \boldsymbol{H}_{\beta\beta}^{-1}\left(\boxed{-\boldsymbol{H}_{\beta\alpha}}\,\boldsymbol{y}_{\alpha}^{k} + \boldsymbol{g}_{\beta}\right)$$

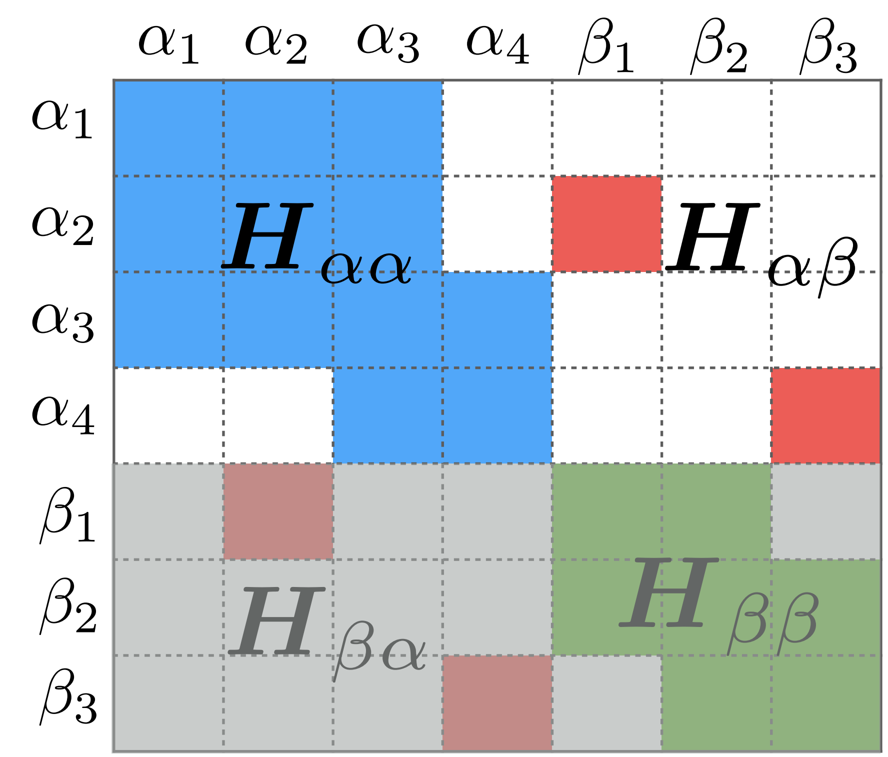


Distributed Gauss-Seidel Approach

Trajectory Estimation Problem

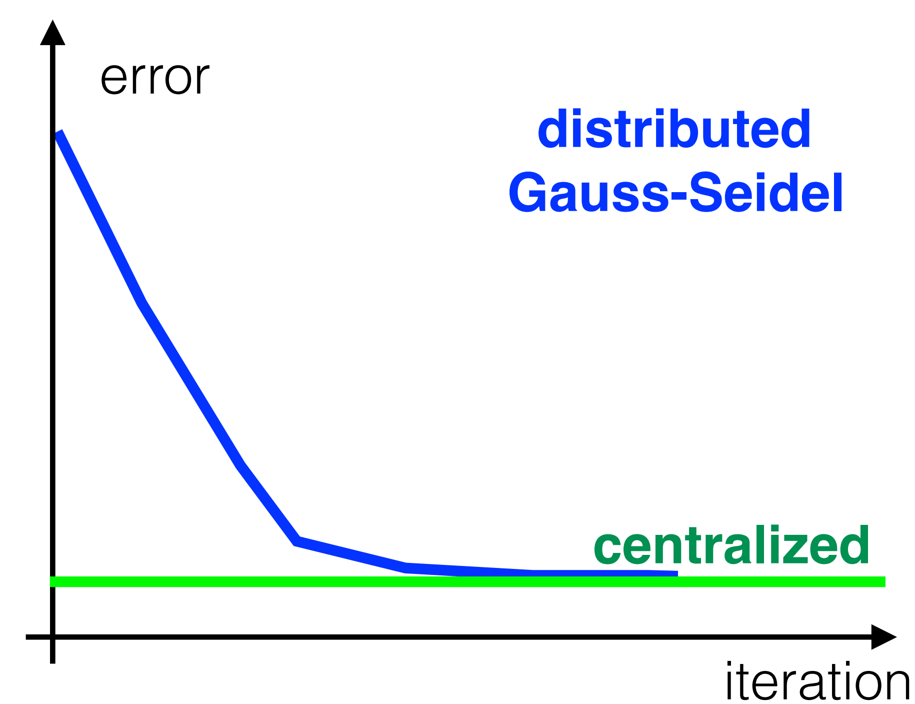


Hessian Matrix



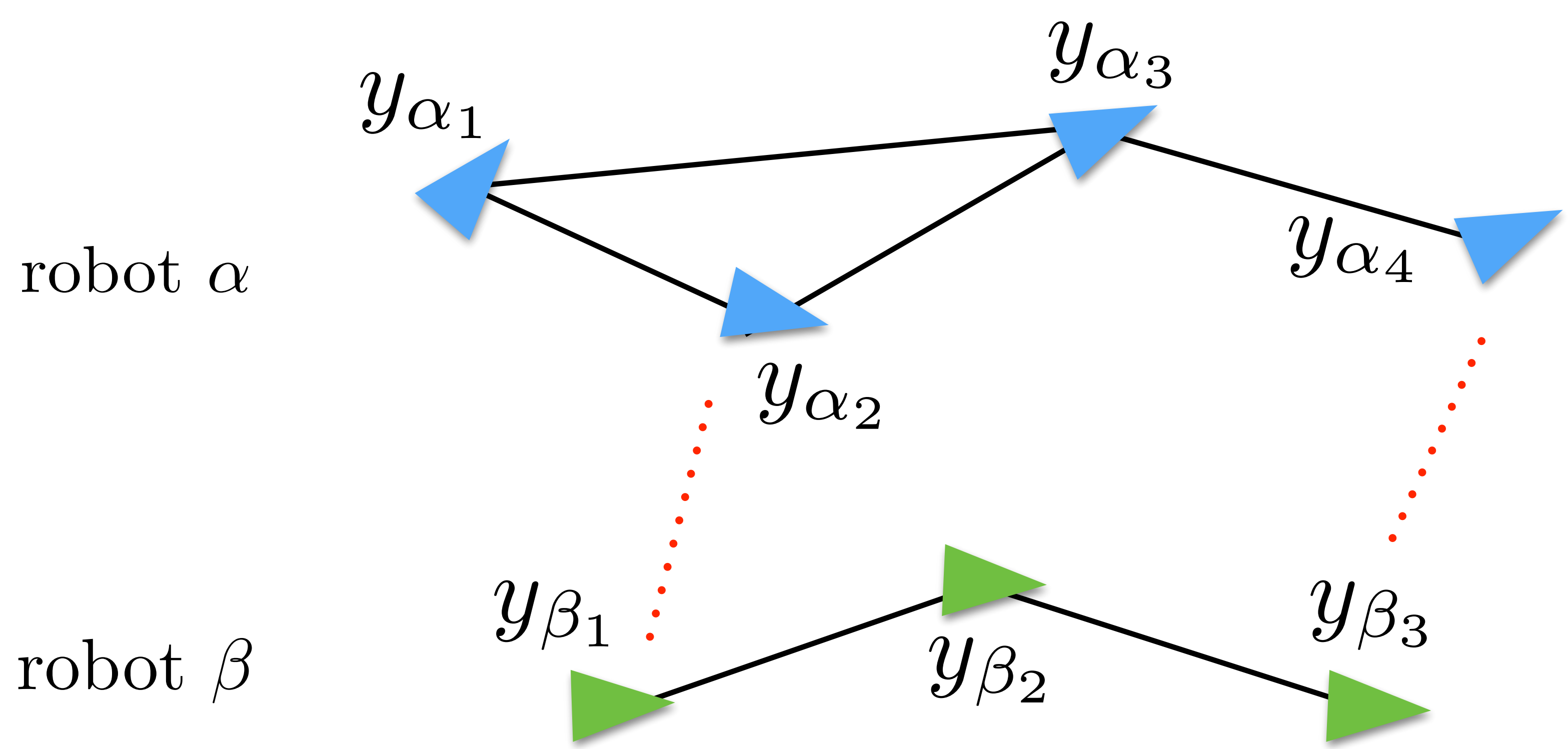
$$\boldsymbol{y}_\alpha^{k+1} = \boldsymbol{H}_{\alpha\alpha}^{-1}\left(\boxed{-\boldsymbol{H}_{\alpha\beta}}\,\boldsymbol{y}_\beta^{k} + \boldsymbol{g}_\alpha\right)$$

$$\boldsymbol{y}_\beta^{k+1} = \boldsymbol{H}_{\beta\beta}^{-1}\left(\boxed{-\boldsymbol{H}_{\beta\alpha}}\,\boldsymbol{y}_\alpha^{k} + \boldsymbol{g}_\beta\right)$$

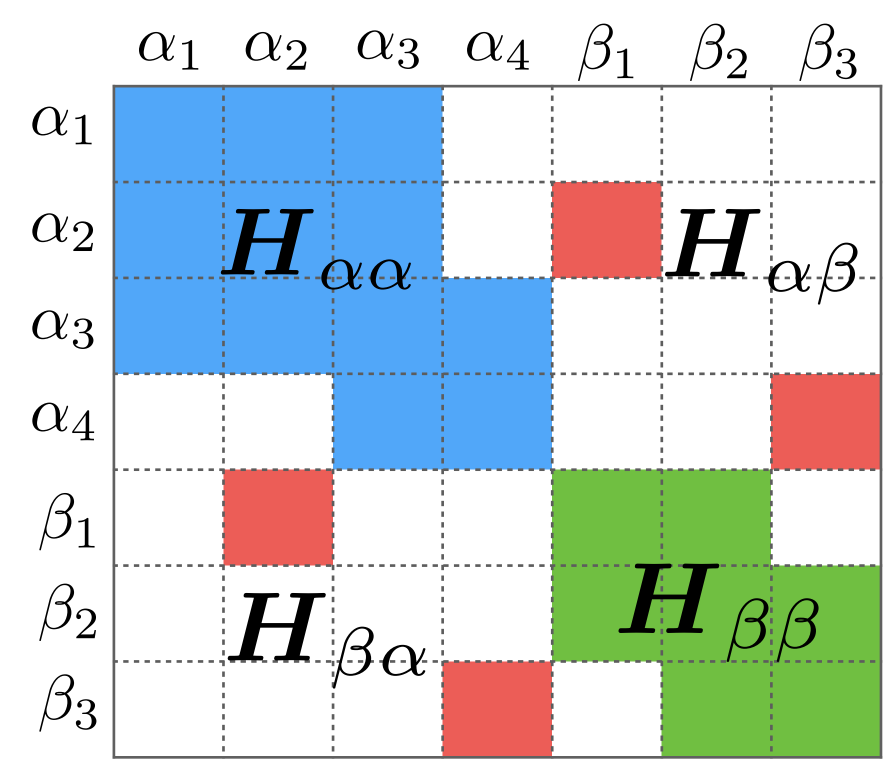


Distributed Gauss-Seidel Approach

Trajectory Estimation Problem

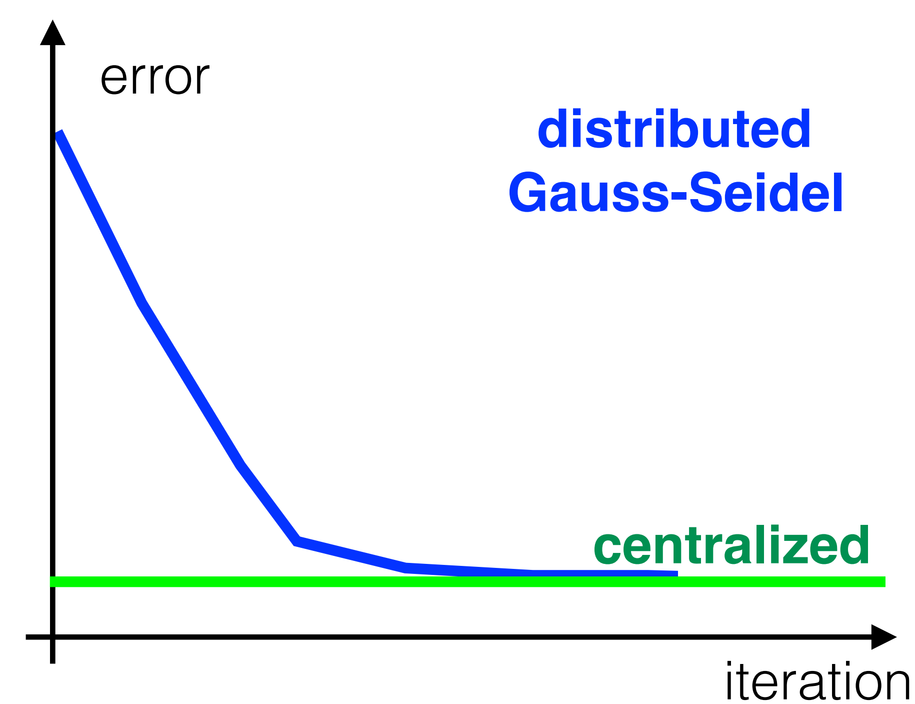


Hessian Matrix

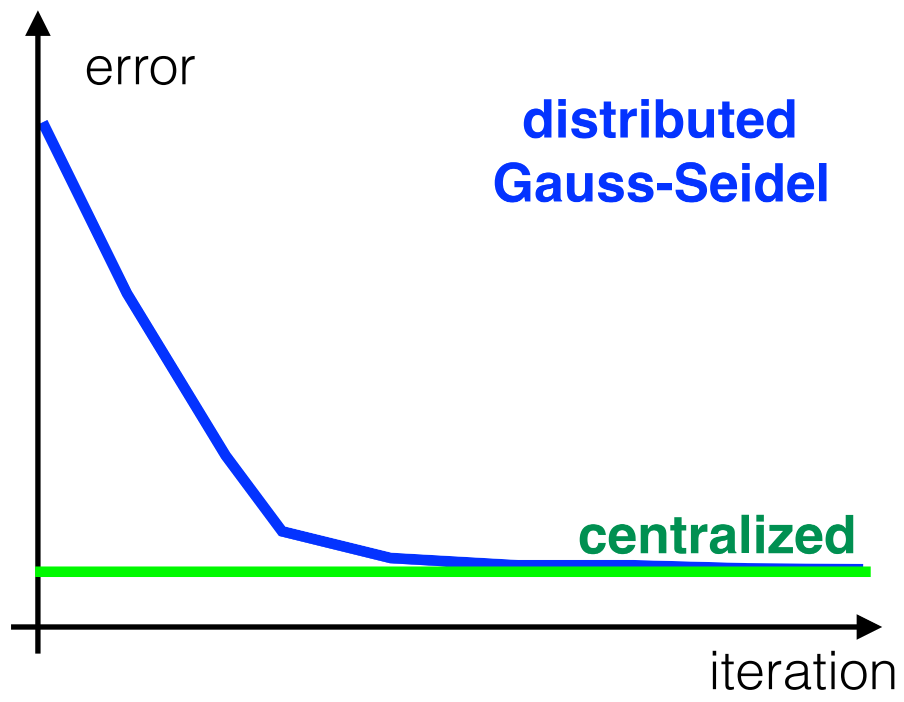


Stop the iterations if the change in the estimate is sufficiently small

$$\|y^{k+1} - y^k\| \leq threshold$$



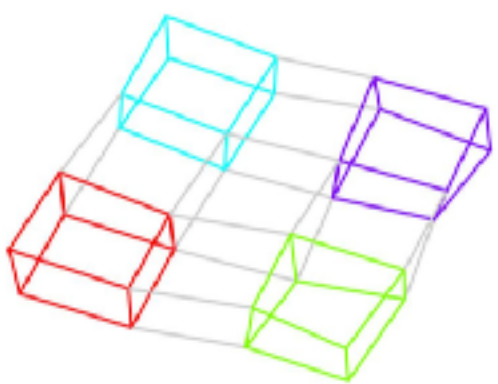
Distributed Gauss-Seidel Approach



Guaranteed Convergence:

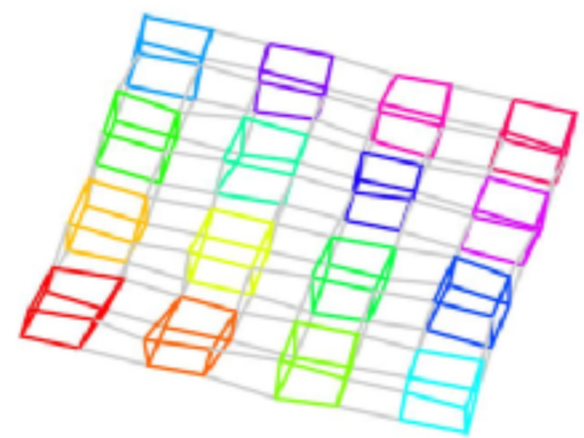
The Gauss-Seidel iterations converge to the centralized solution starting from any initial estimate

Simulation Experiments

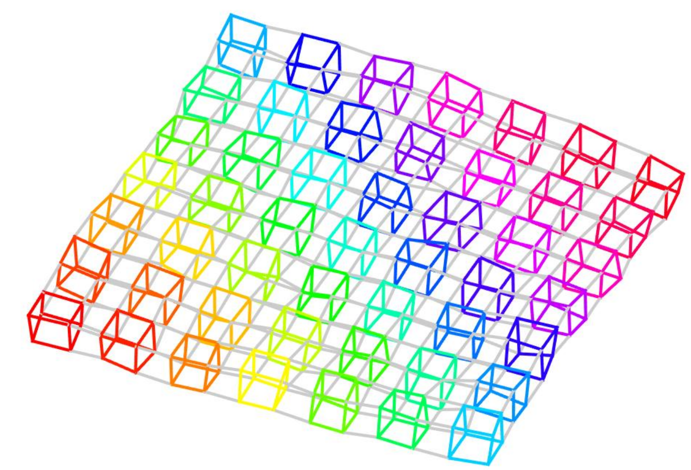


4 robots

9 robots



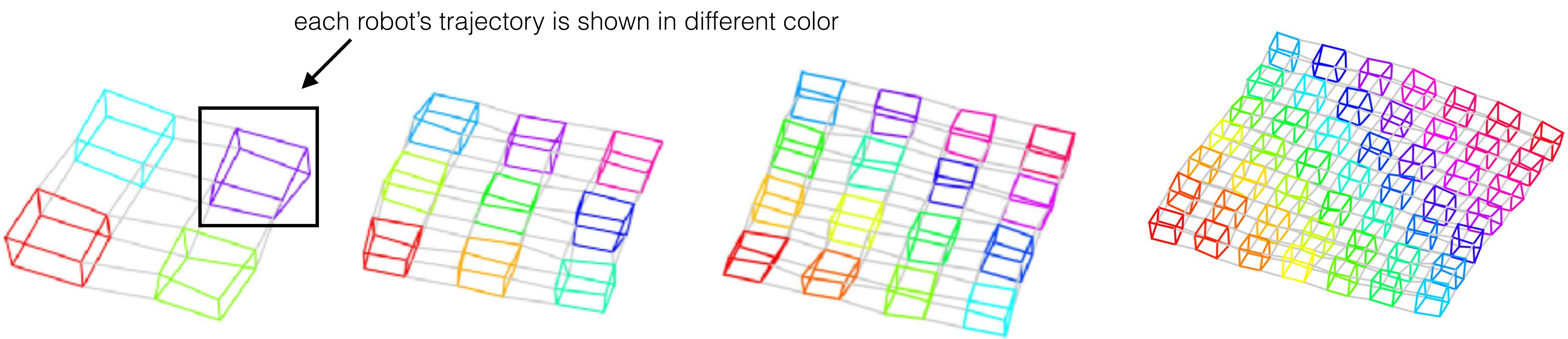
16 robots



49 robots

We simulate different problems with robots moving along a 3D grid

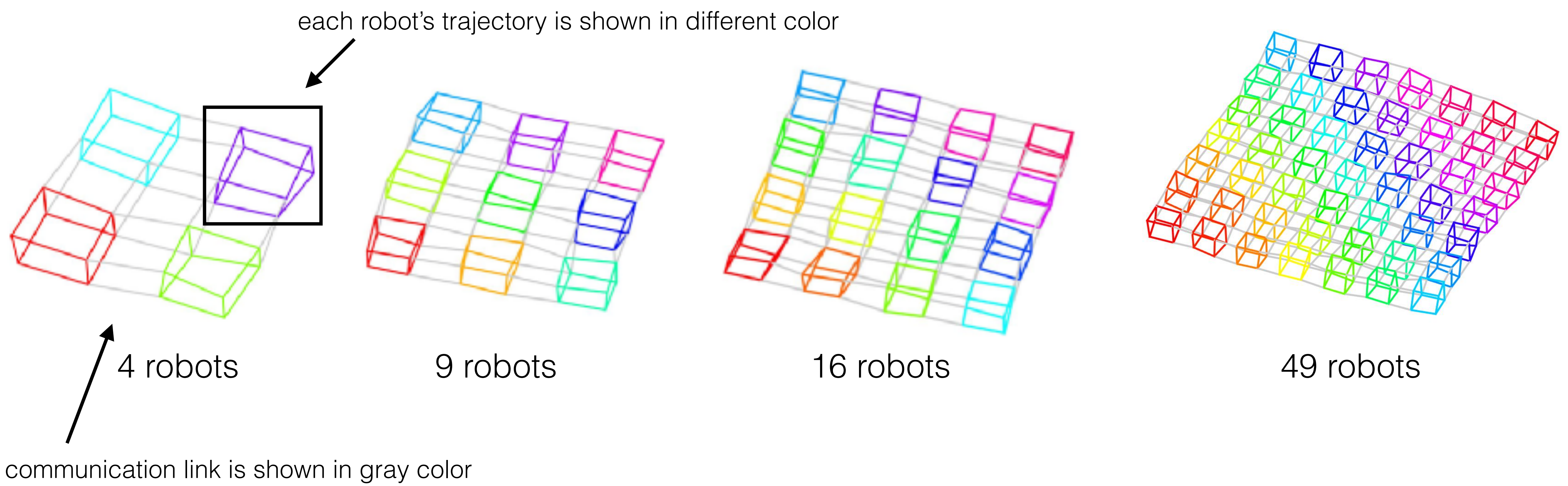
Simulation Experiments



4 robots | 9 robots | 16 robots | 49 robots

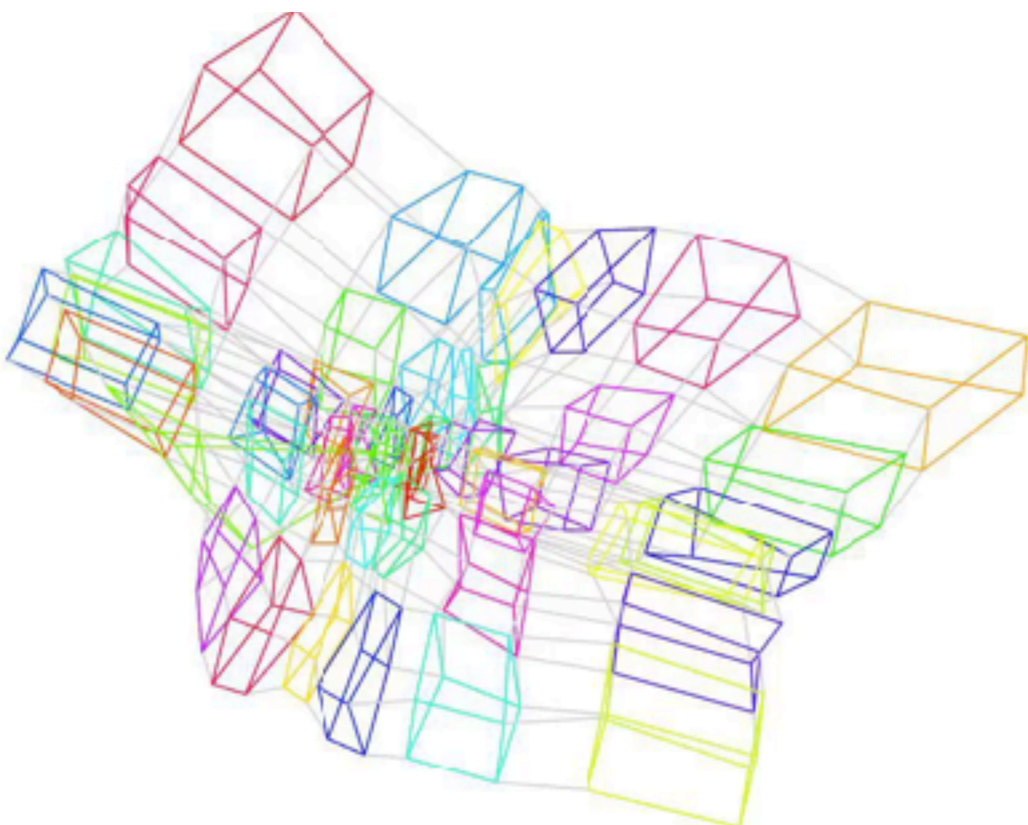
We simulate different problems with robots moving along a 3D grid

Simulation Experiments



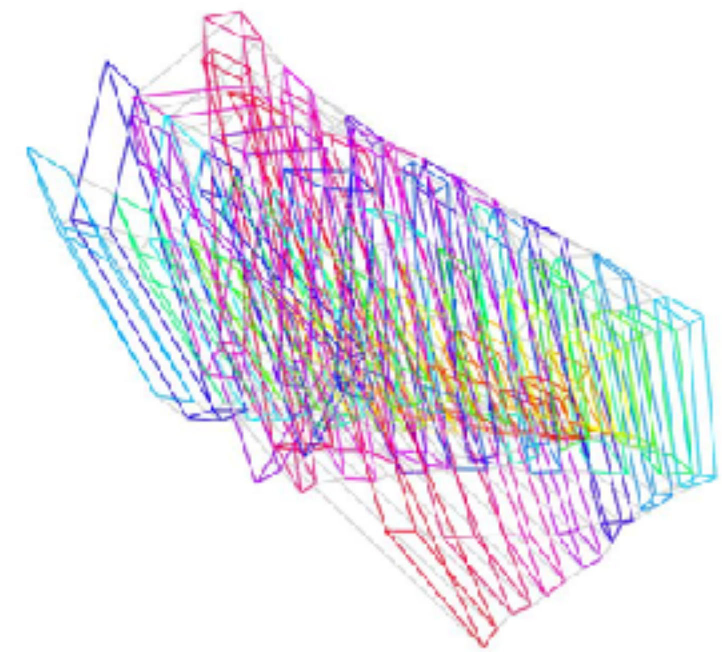
We simulate different problems with robots moving along a 3D grid

Robust to bad Initialization



Distributed Gauss-Seidel

(our algorithm)

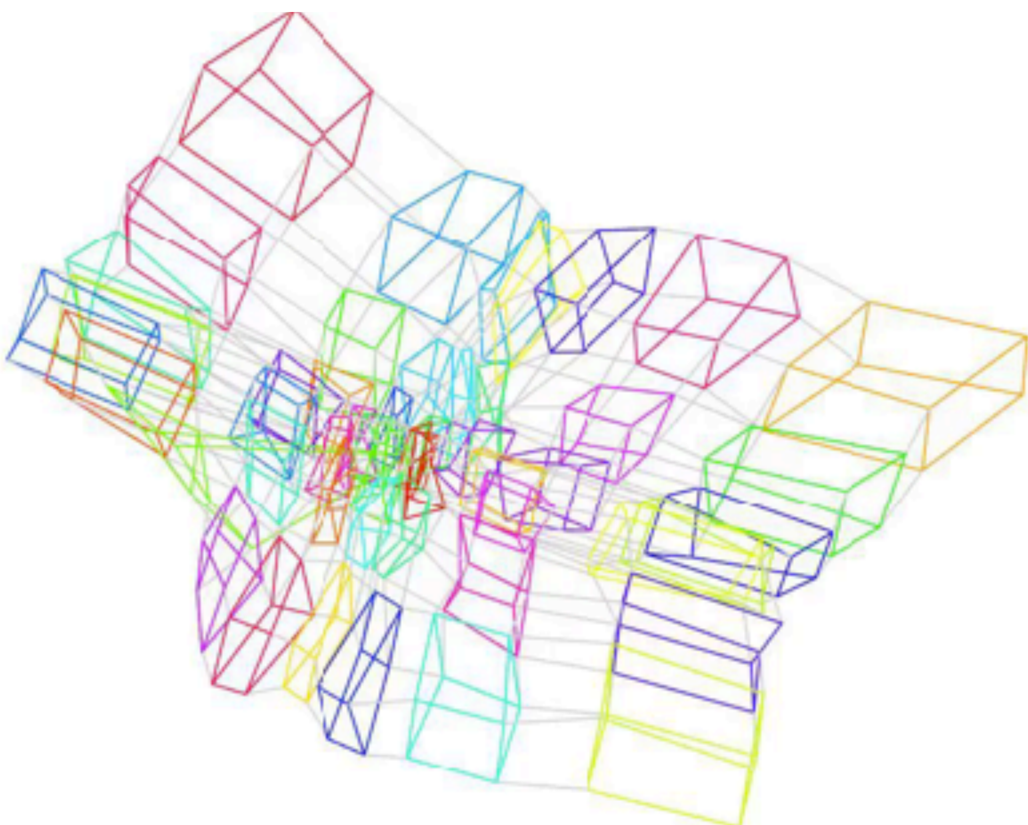


DDF-SAM

(state of art)

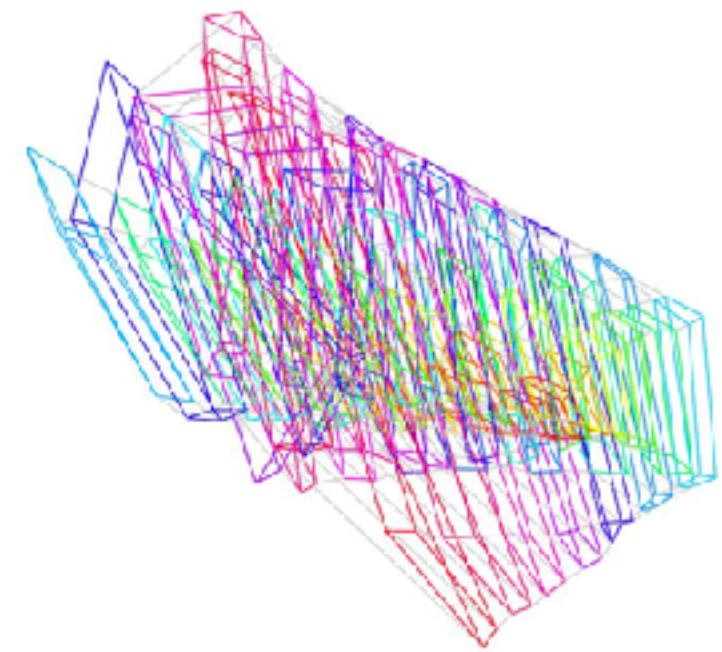
DDF-SAM: Fully distributed SLAM using constrained factor graphs
Cunningham et al. (IROS 2010)

Robust to bad Initialization



Distributed Gauss-Seidel

(our algorithm)

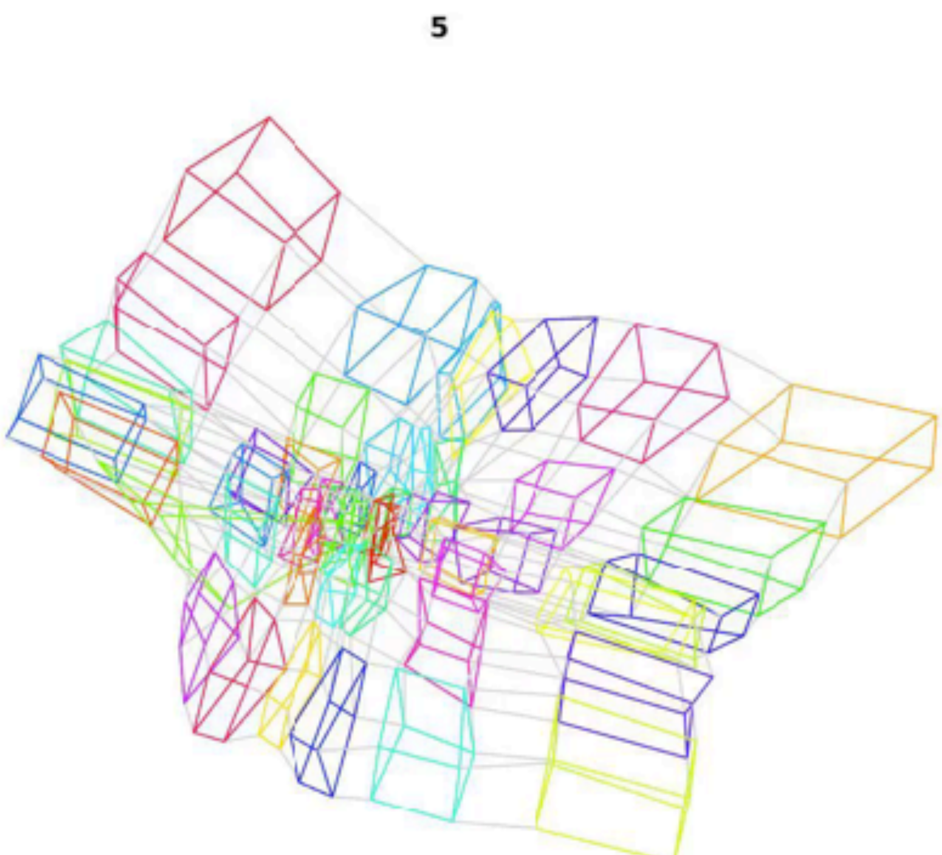


DDF-SAM

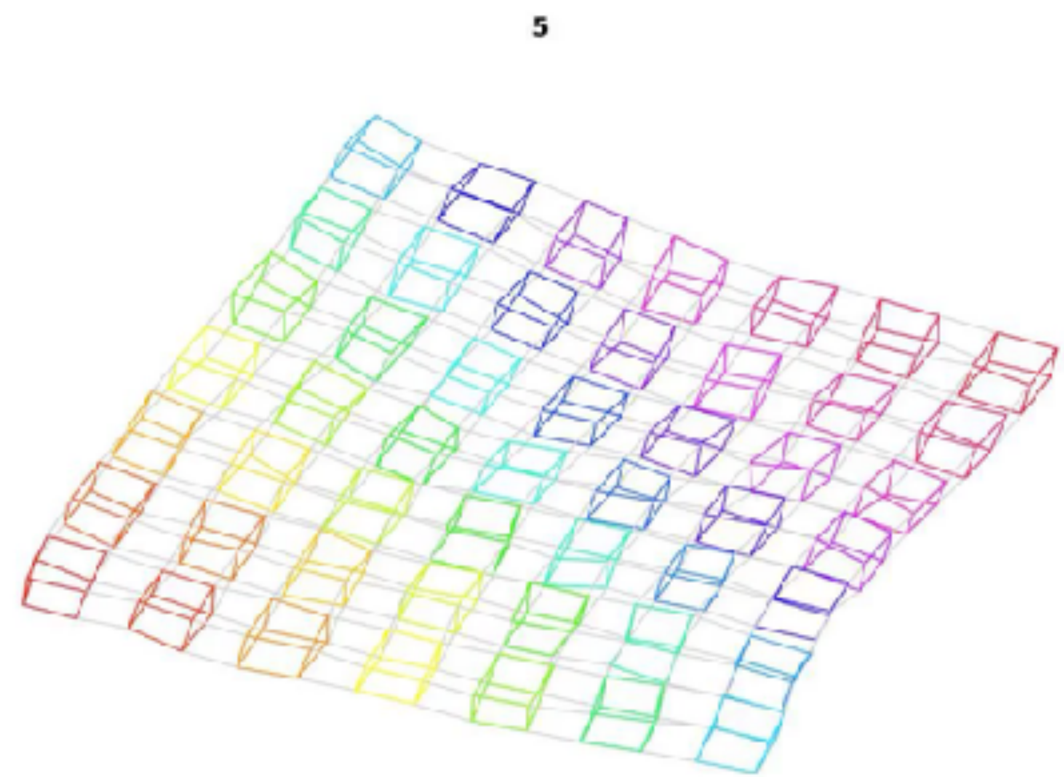
(state of art)

DDF-SAM: Fully distributed SLAM using constrained factor graphs
Cunningham et al. (IROS 2010)

Fast Convergence with Flagged Initialization

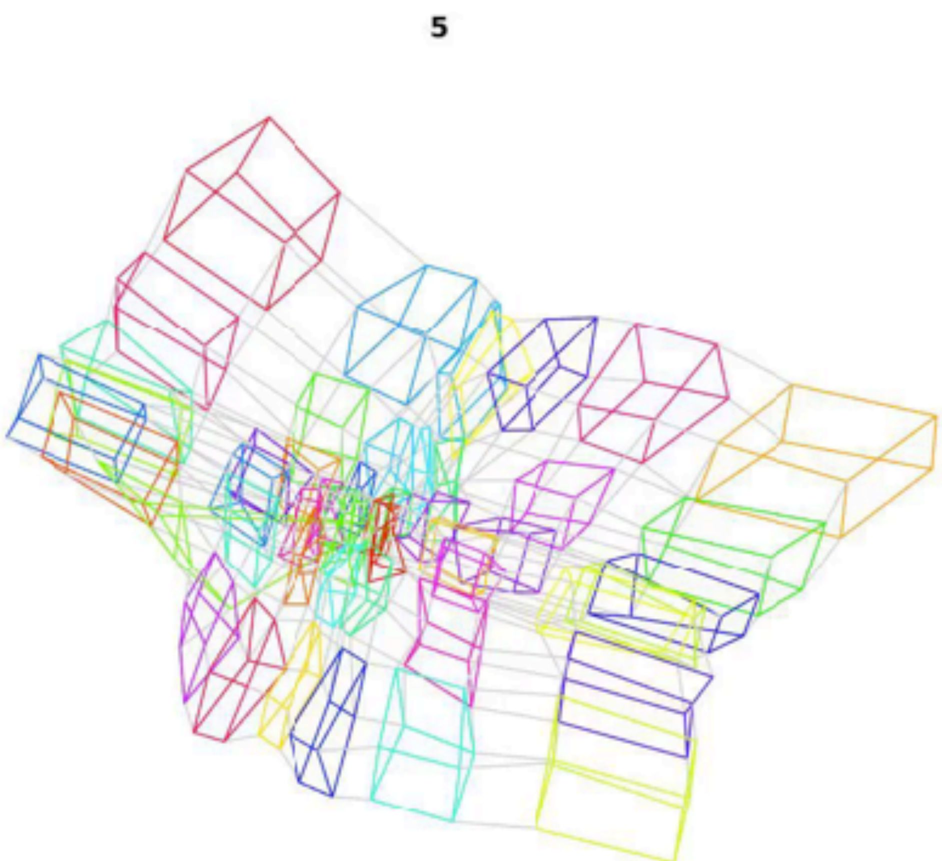


Without
Flagged Initialization

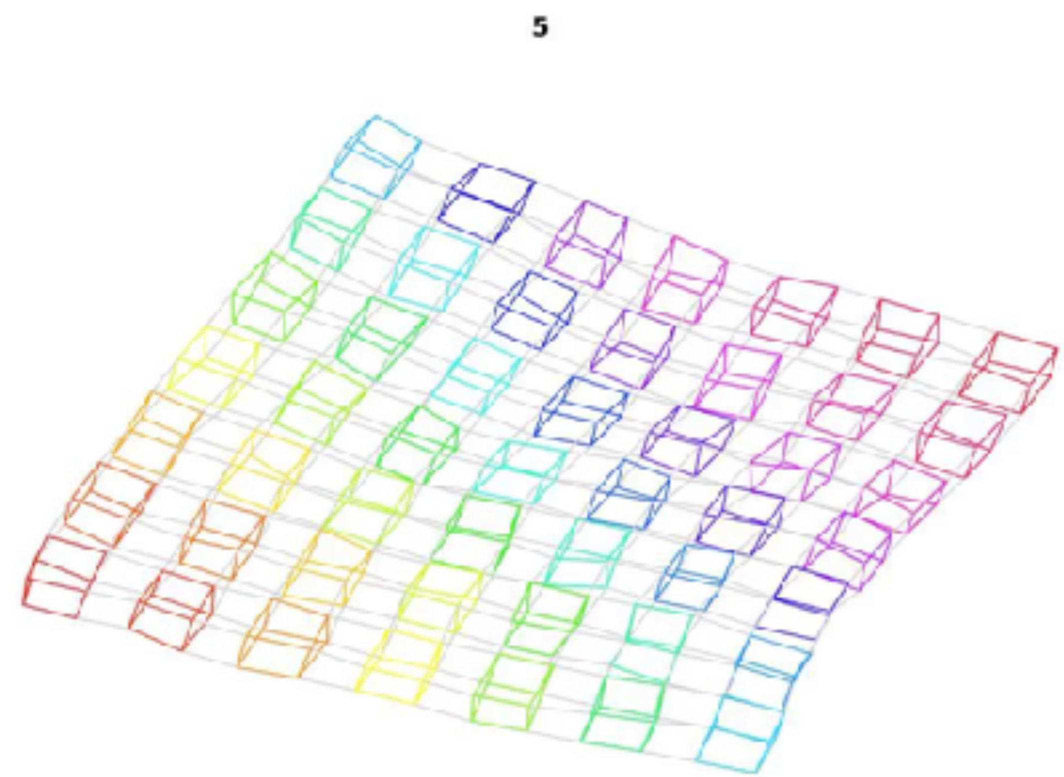


With
Flagged Initialization

Fast Convergence with Flagged Initialization

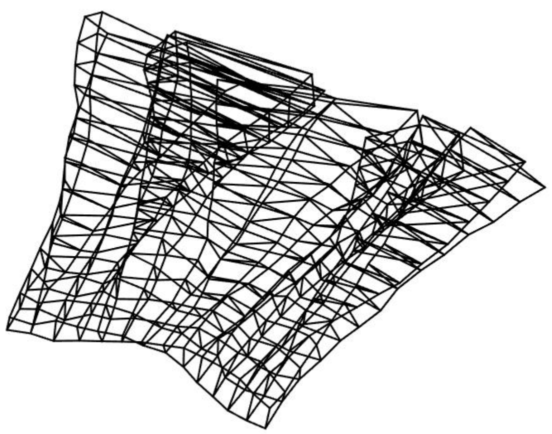


Without
Flagged Initialization

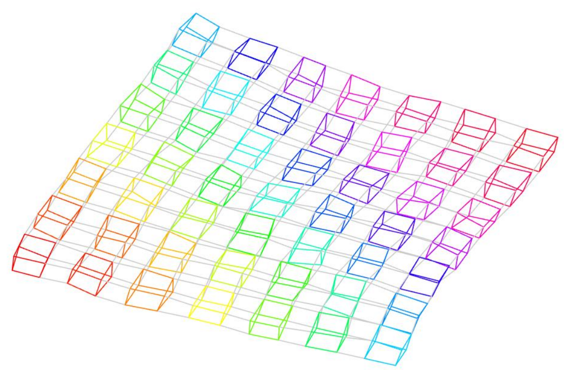


With
Flagged Initialization

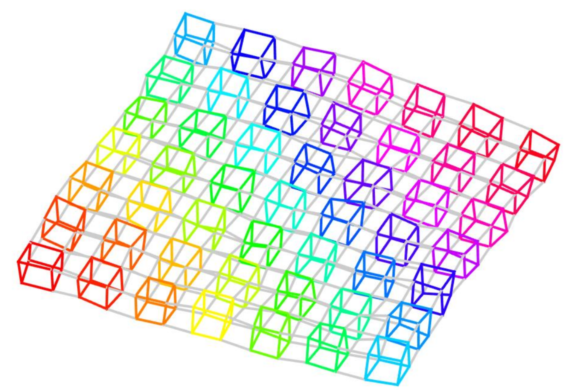
Anytime Flavor



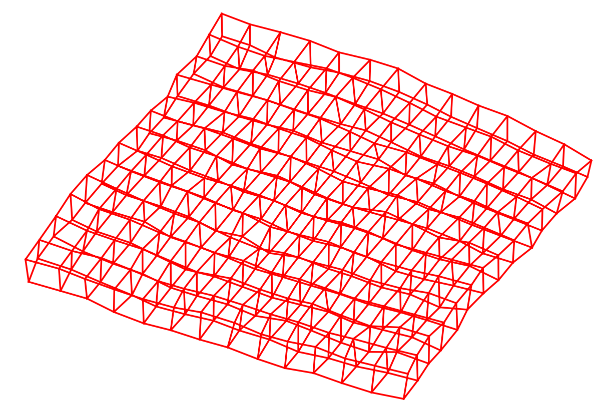
Initial



10 iterations



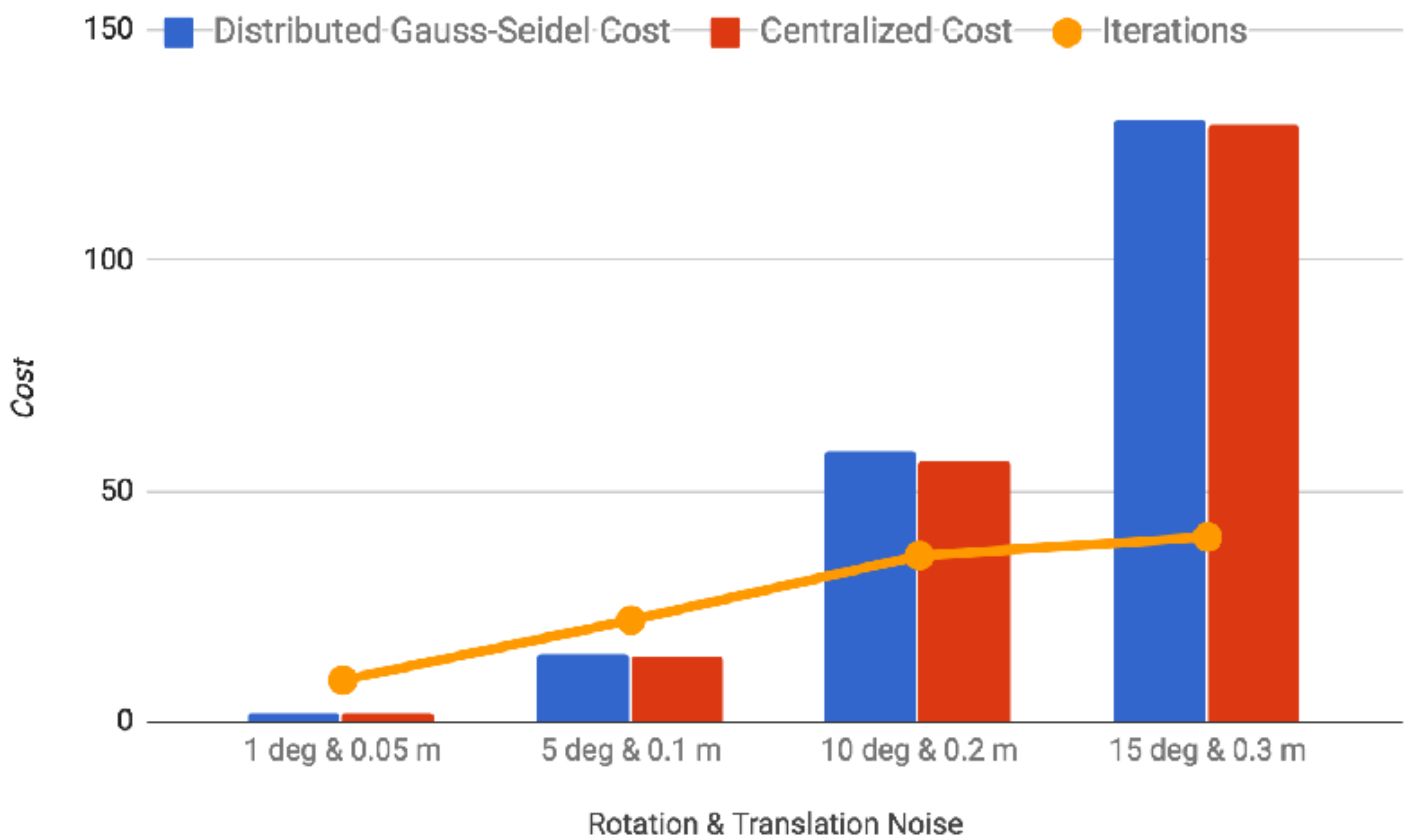
1000 iterations



Centralized

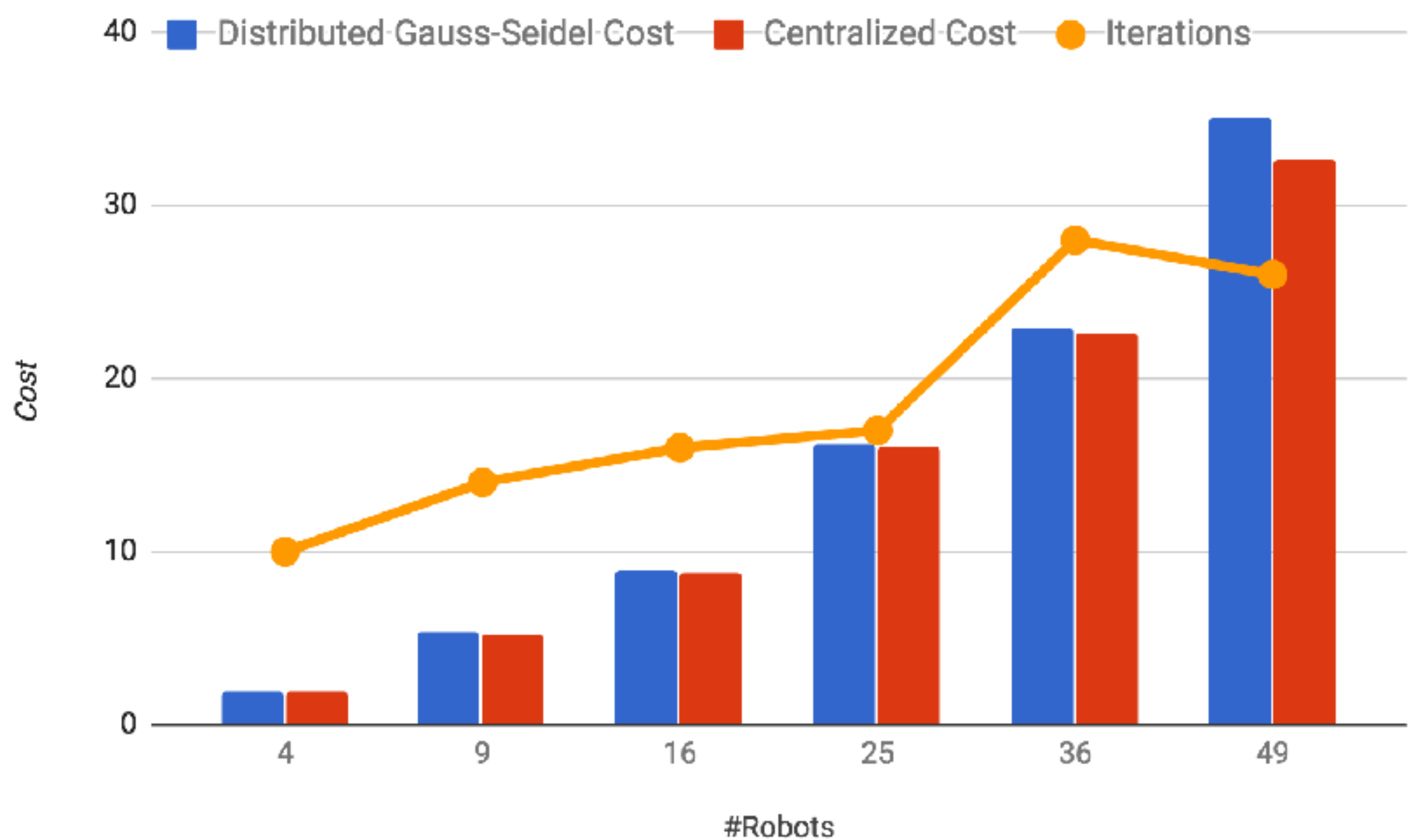
Already accurate after few iterations.

Resilient to Measurement Noise



Cost: $$\sum_{i=1}^{M} \frac{1}{2} \| \underbrace{h_i(x) - z_i}_{\text{prediction error}} \|^2_{\Sigma_i}$$

Scalable in the number of robots



Communication Bandwidth Requirements

Communication Bandwidth =

Amount of communication required to perform distributed optimization given the communication link.

+

The amount of communication required to establish that link.

Communication Bandwidth Requirements

Distributed Gauss-Seidel

Communication Bandwidth =

$\boxed{\text{Amount of communication required to perform distributed optimization given the communication link.}}$

+

The amount of communication required to establish that link.

Communication Bandwidth Requirements

Distributed Gauss-Seidel

Communication Bandwidth =

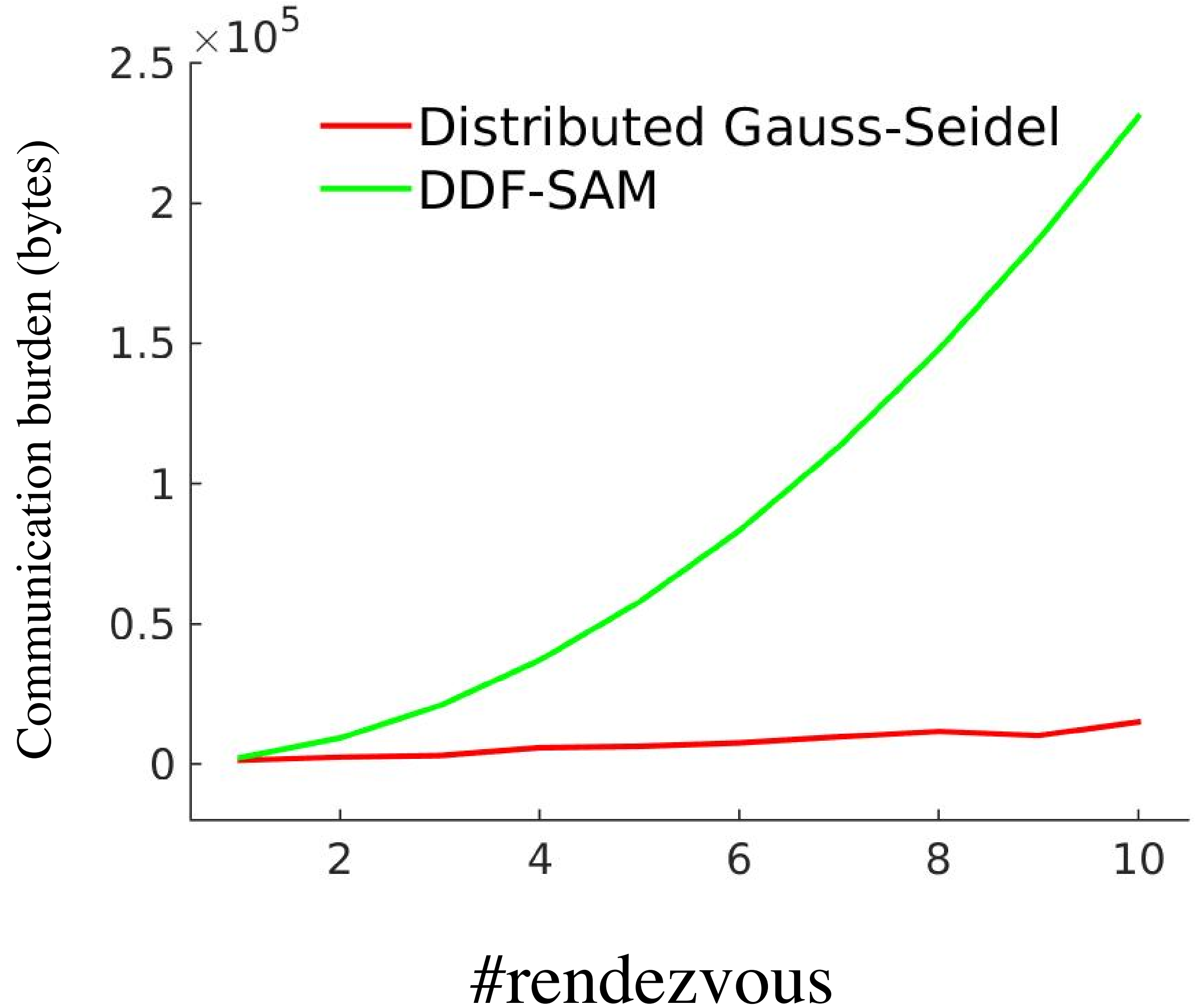
Amount of communication required to perform distributed optimization given the communication link.

+

The amount of communication required to establish that link.

Distributed Object-based SLAM

Communication is Linear



Increase in communication burden is linear with the increase in the number of communication links

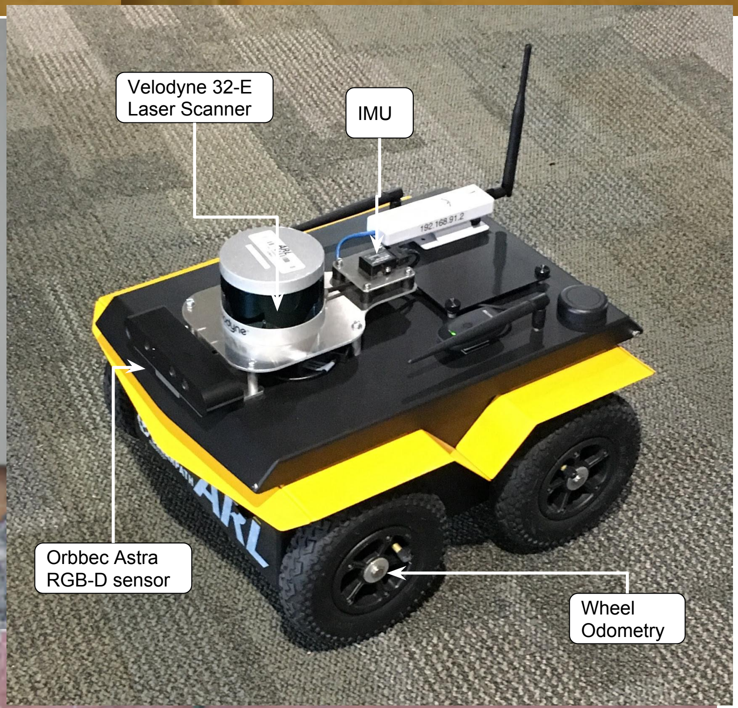
DDF-SAM: Fully distributed SLAM using constrained factor graphs
Cunningham et al. (IROS 2010)

Field Experiments









We tested the proposed approach on field data collected by two to ten Jackal robots, moving in different environments. We use the estimated trajectories to reconstruct a 3D map of the facility

Field Experiments

We tested the proposed approach on field data collected by two to ten Jackal robots, moving in different environments. We use the estimated trajectories to reconstruct a 3D map of the facility

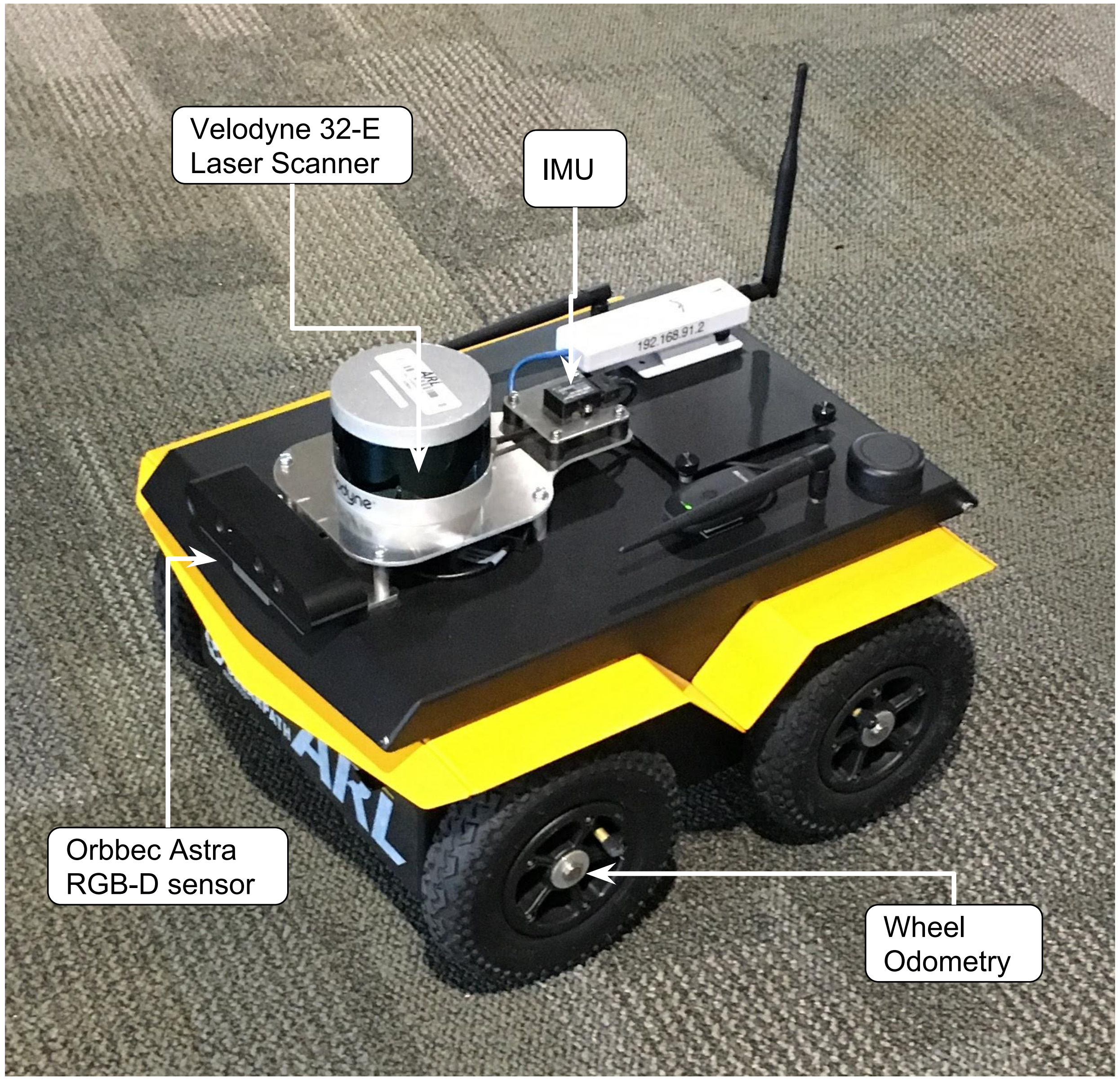
Field Experiments

RGBD Frame

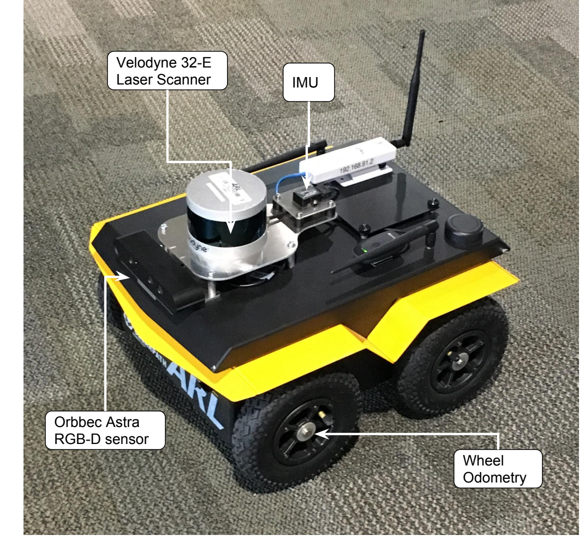
Wheel Odometry

IMU

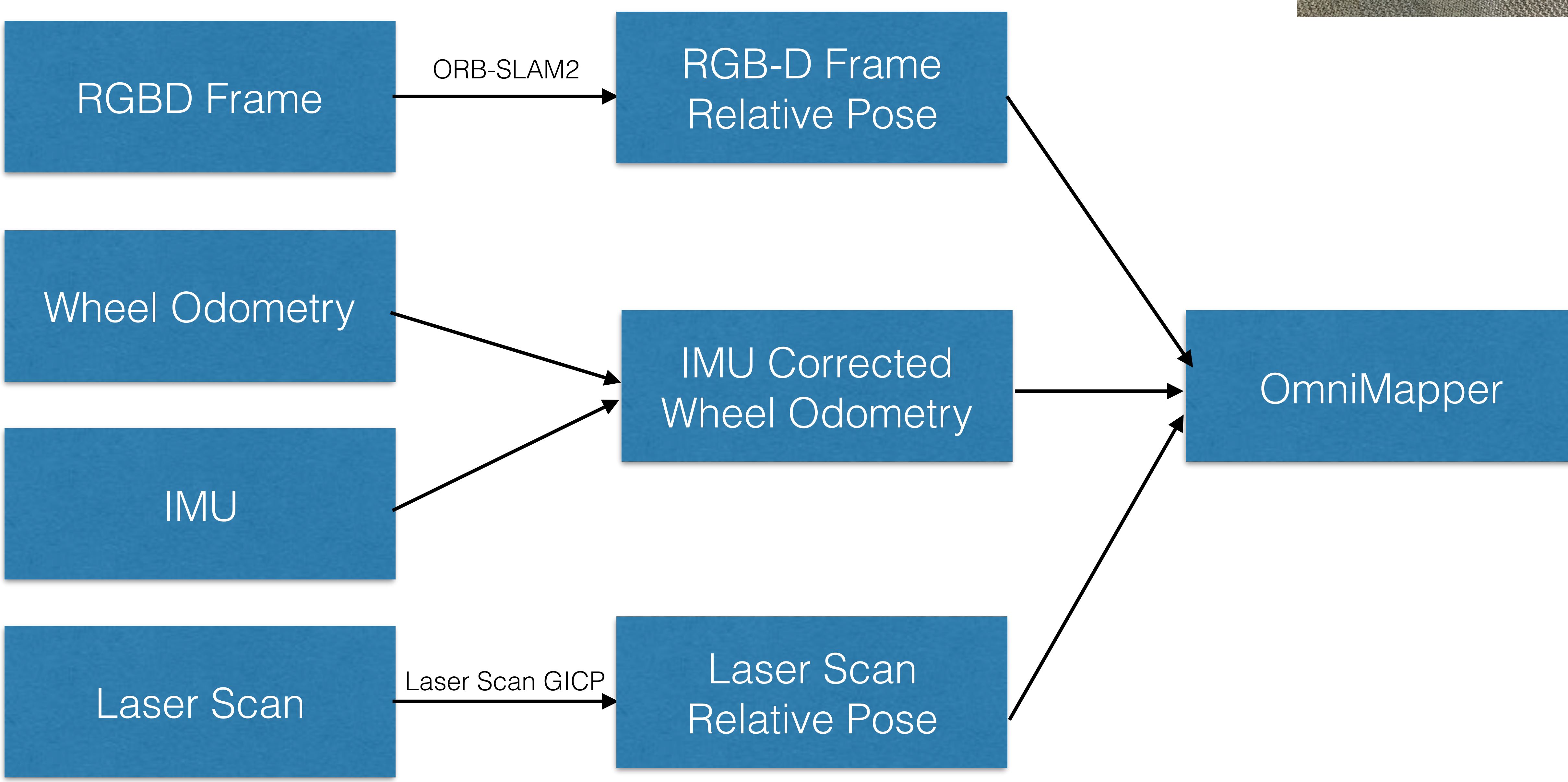
Laser Scan



Field Experiments

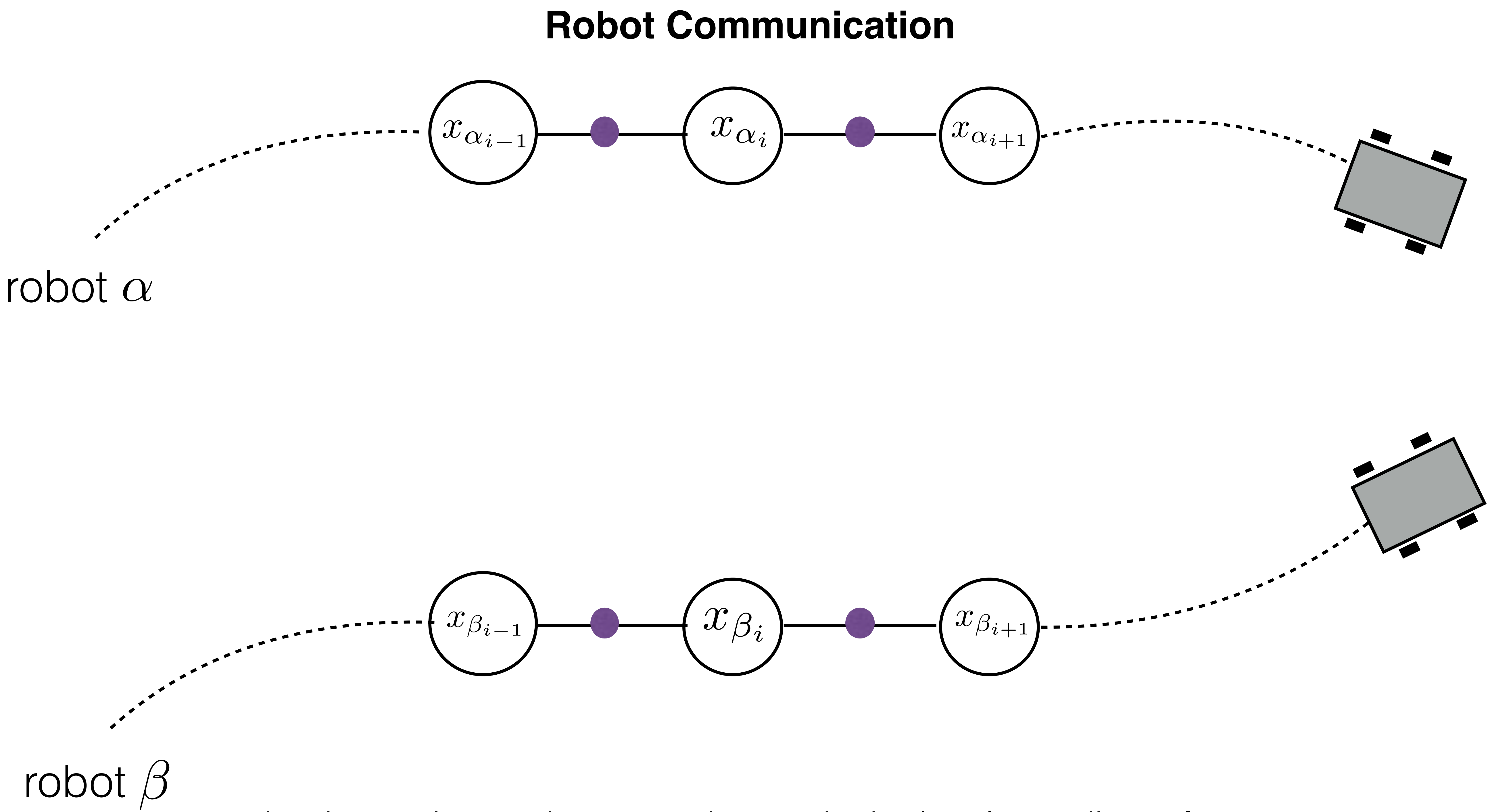


Ego-motion Estimation



OmniMapper: A multimodal mapping framework
Trevor et al. (ICRA 2014)

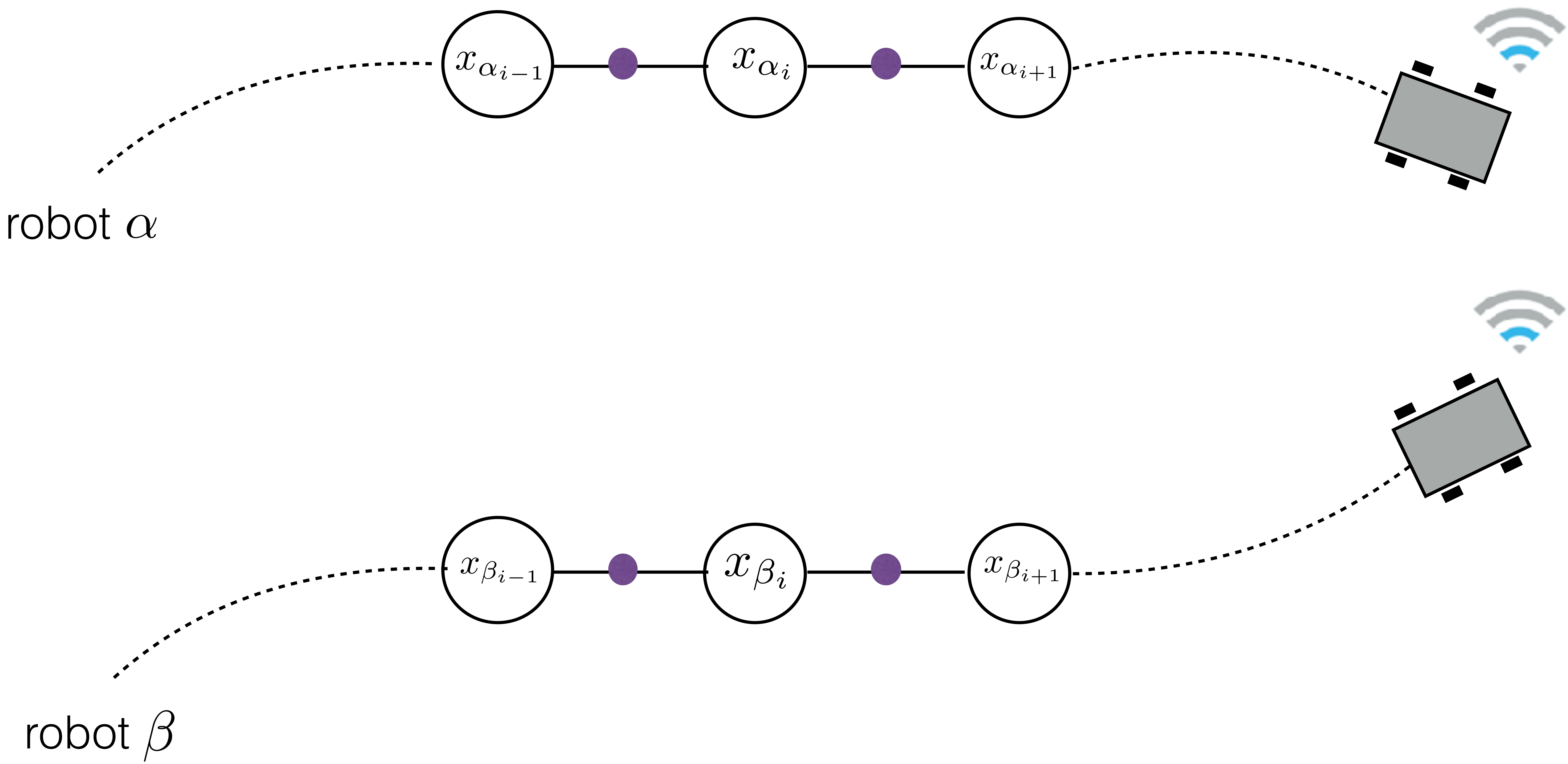
Field Experiments



each robot estimates its own trajectory in the local coordinate frame

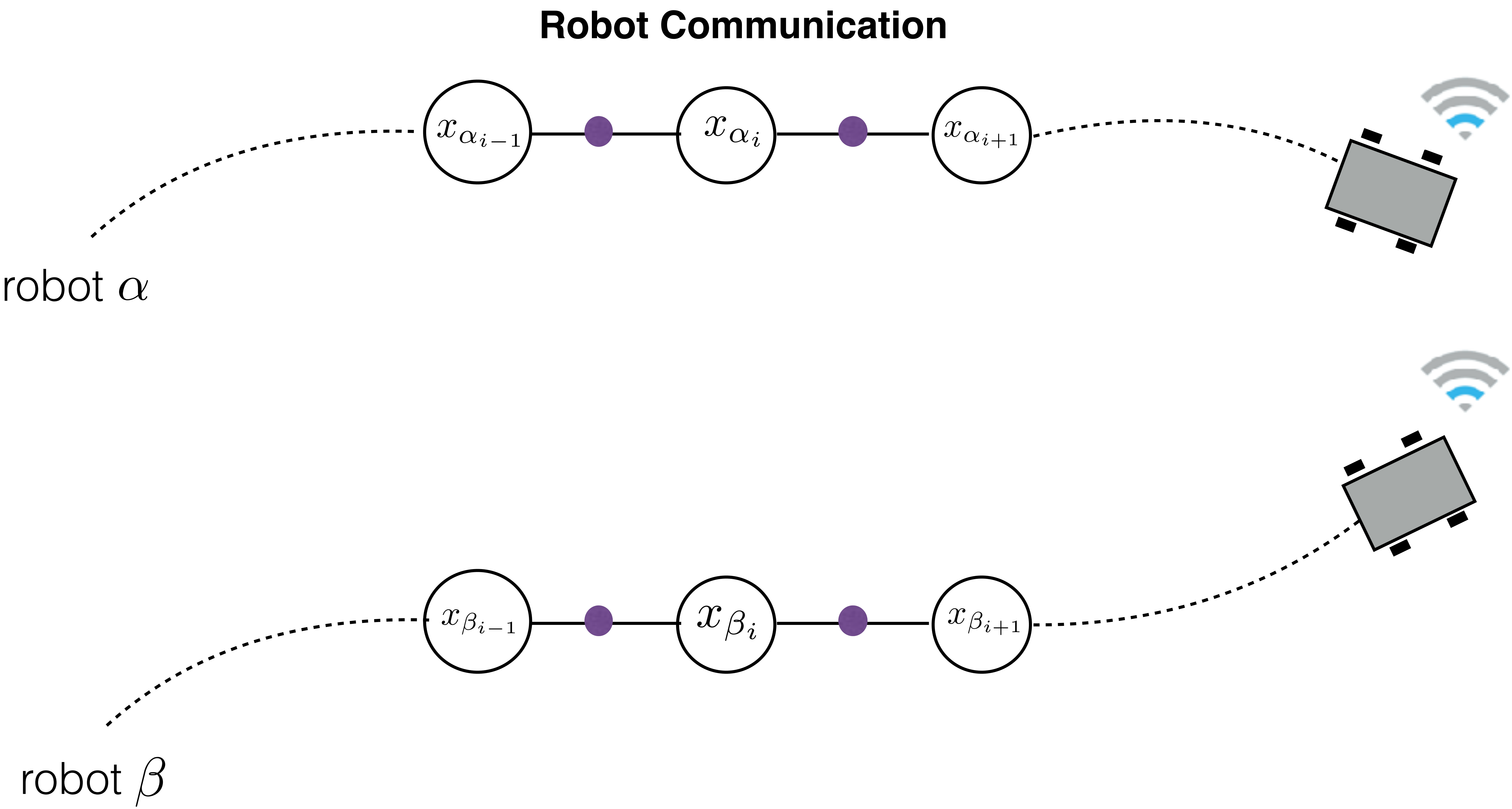
Field Experiments

Robot Communication



each robot communicates nearby keyframes and laser scans to the other robot

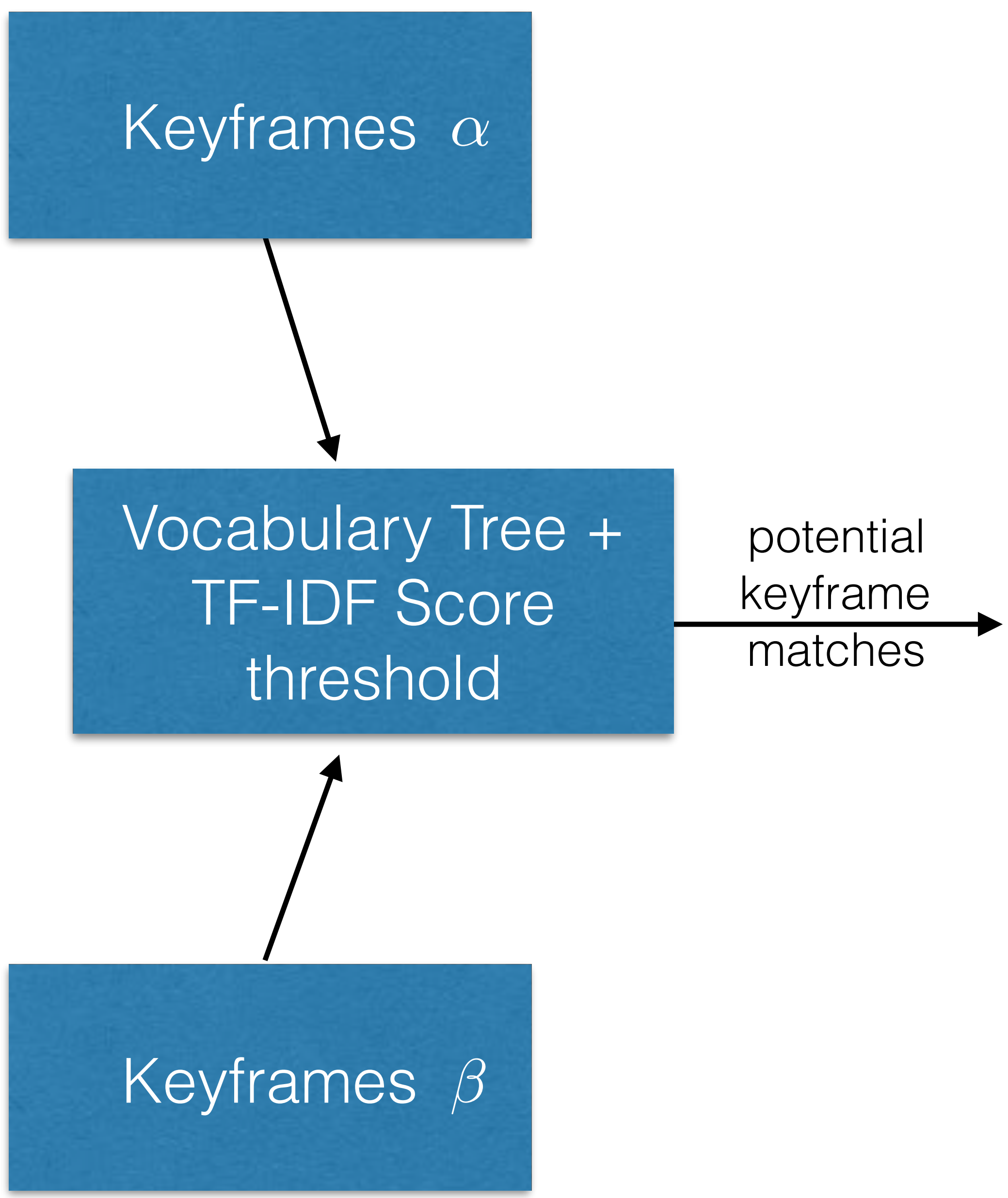
Field Experiments



each robot communicates nearby keyframes and laser scans to the other robot

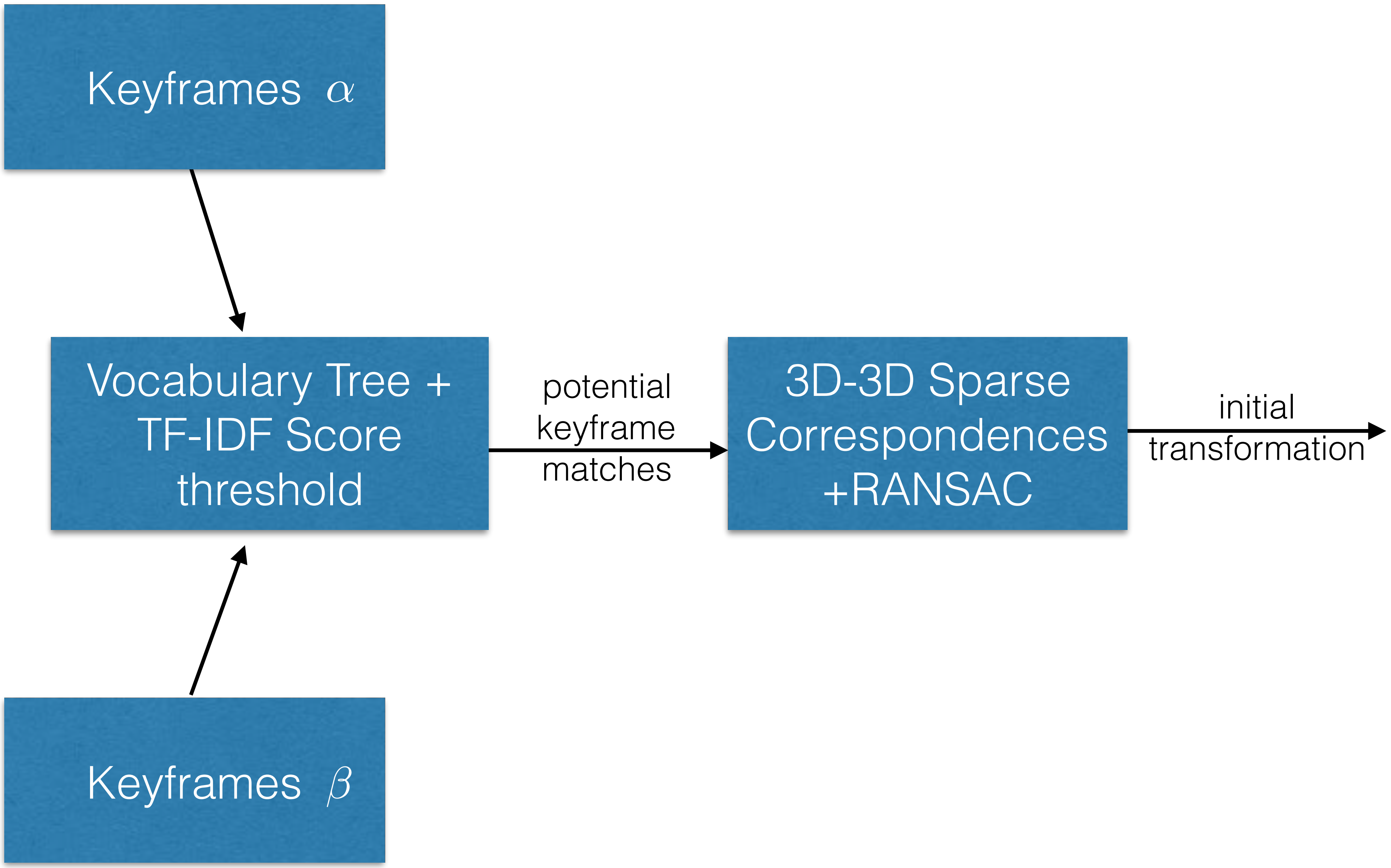
Field Experiments

Robot Communication

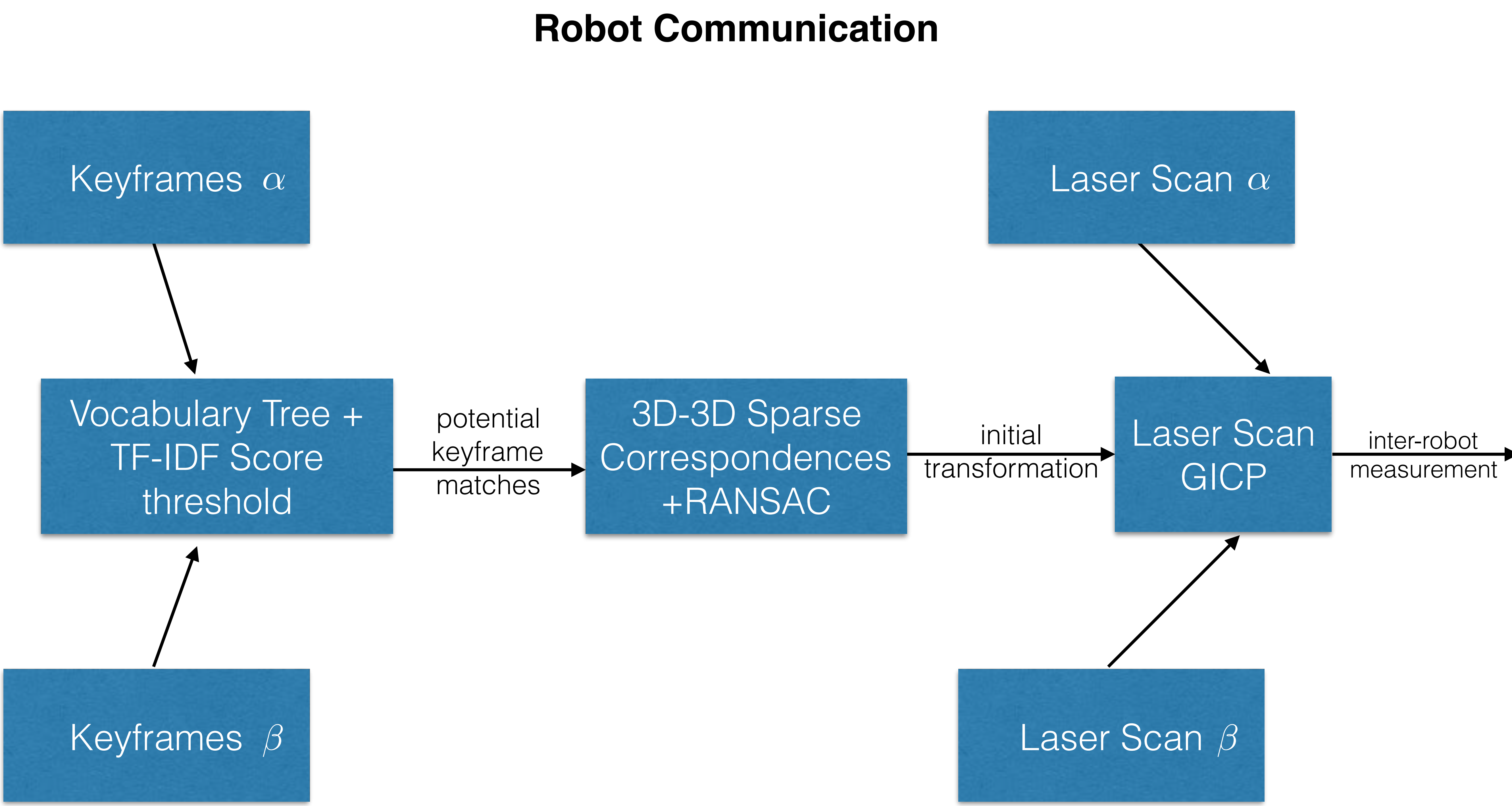


Field Experiments

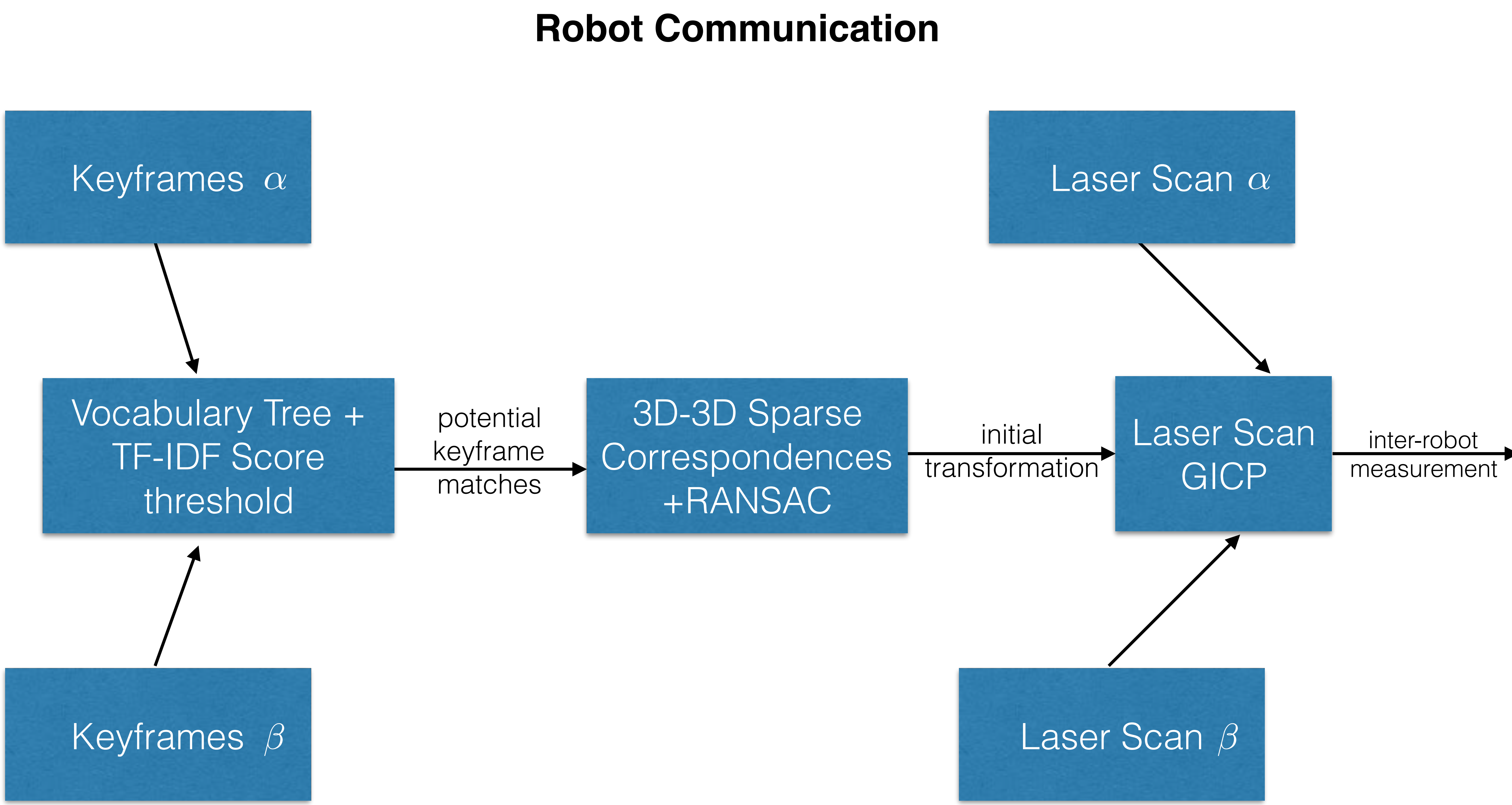
Robot Communication



Field Experiments

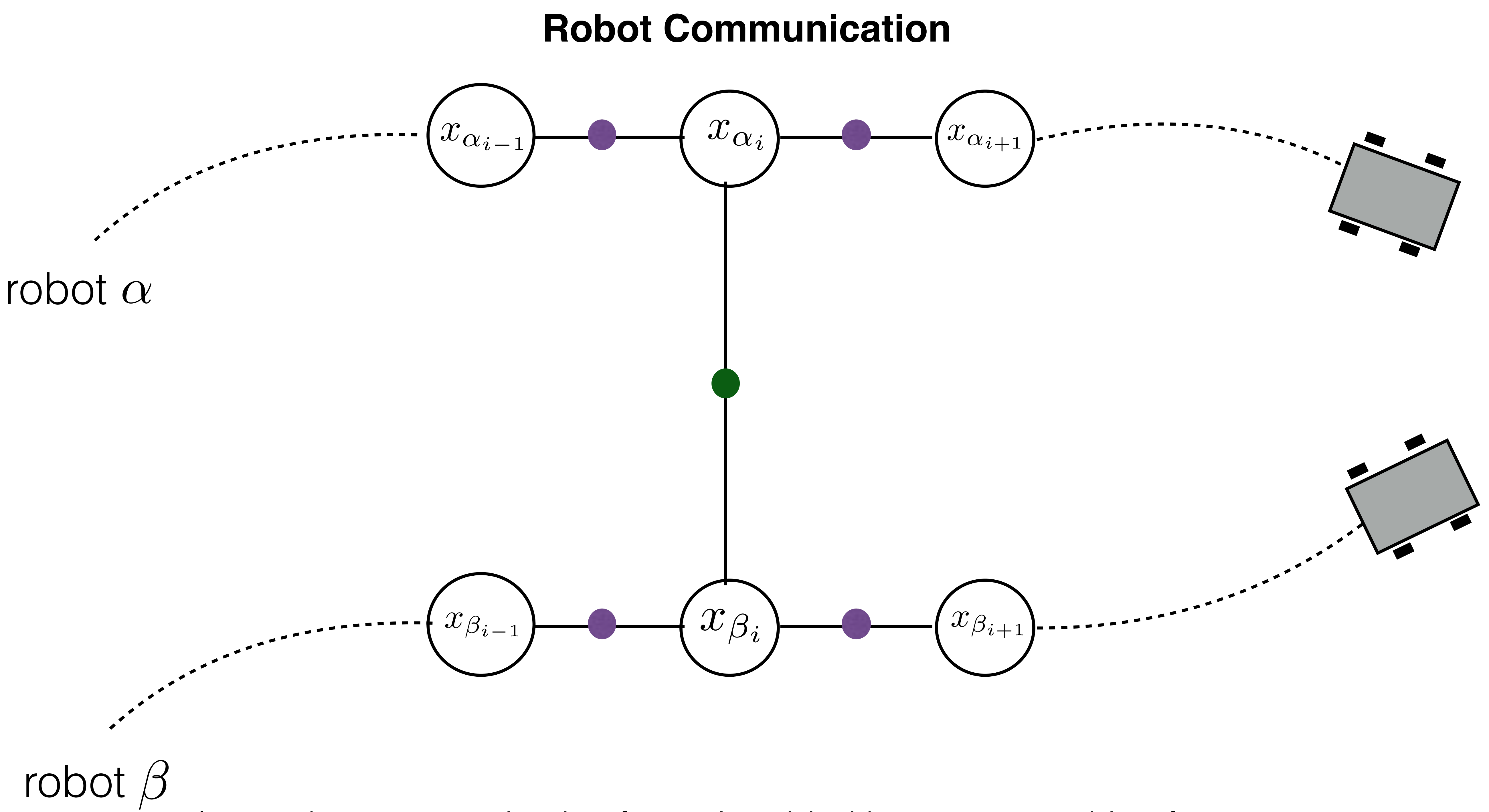


Field Experiments



Assumption: Parameters are conservatively chosen to avoid false positive matches.

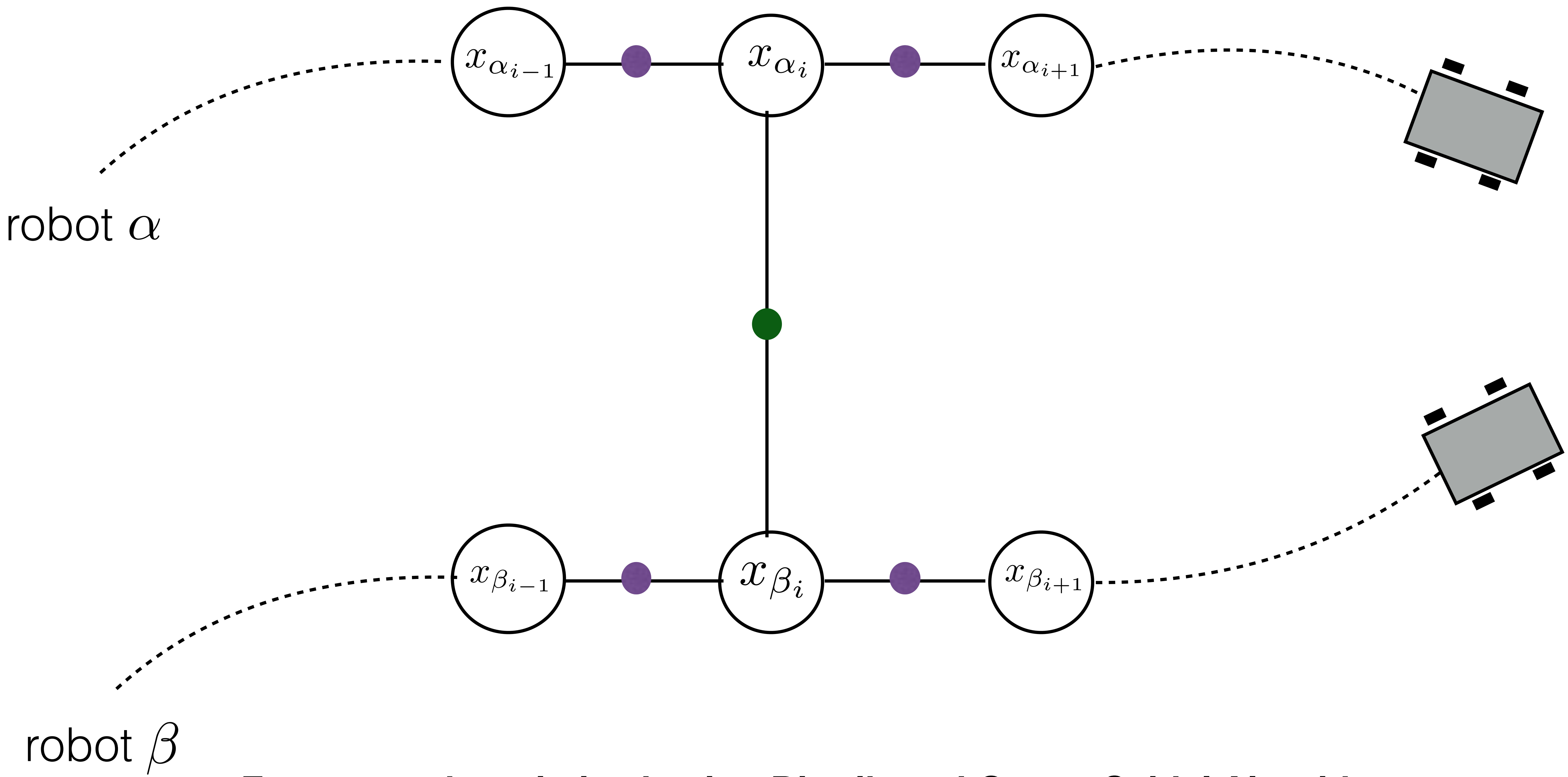
Field Experiments



Inter-robot communication factor is added between matching frames

Field Experiments

Robot Communication



Factor graph optimized using Distributed Gauss-Seidel Algorithm

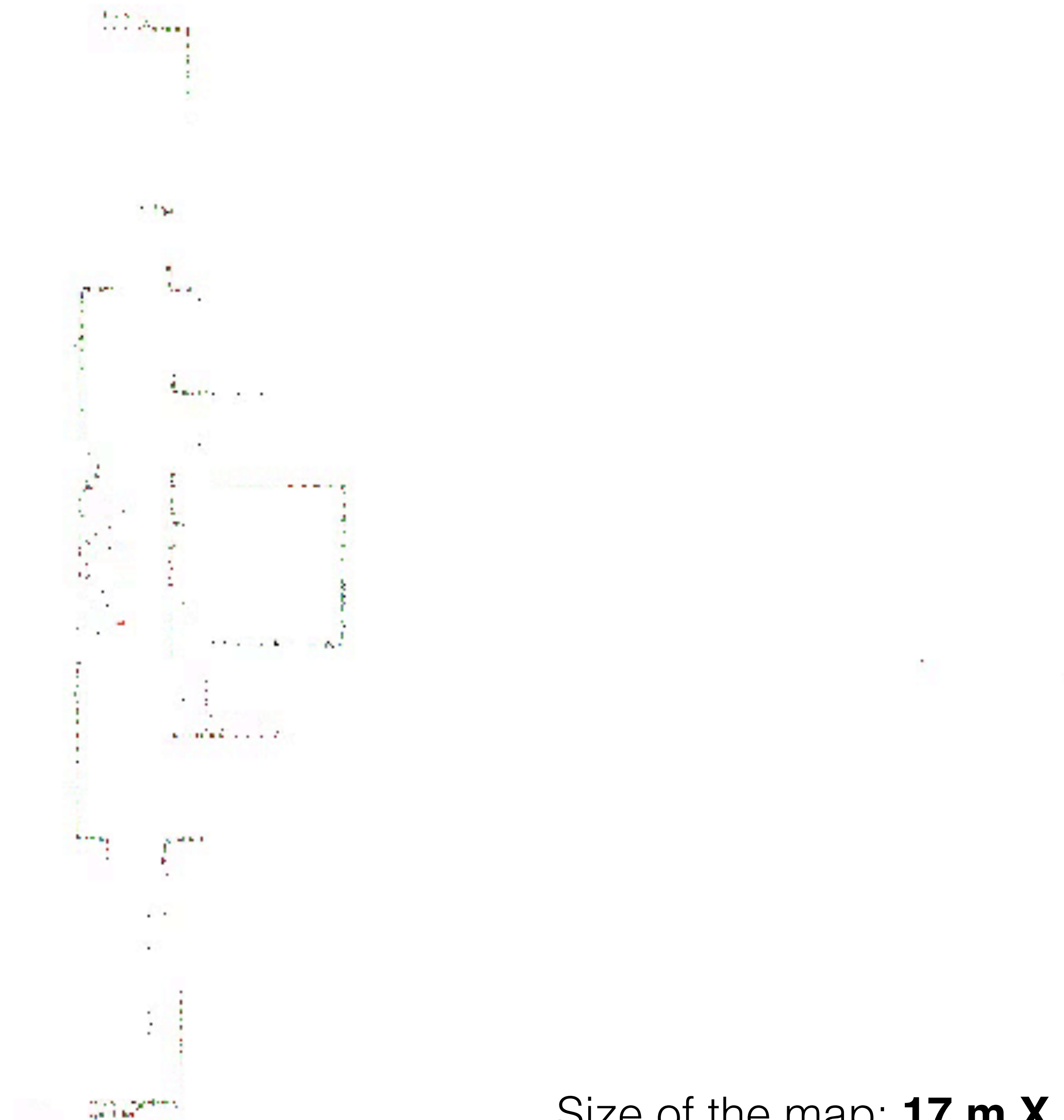
Field Experiments (4 Robots)

Military Training Facility

Size of the map: **17 m X 17 m**
Average Number of Keyframes: **264**
Average Distance travelled by each robot: **246.28 m**

Field Experiments (4 Robots)

Military Training Facility



Size of the map: **17 m X 17 m**
Average Number of Keyframes: **264**
Average Distance travelled by each robot: **246.28 m**

Field Experiments (5 Robots)

Klaus Building (3rd Floor)
Georgia Tech

Size of the map: **151 m X 118 m**
Average Number of Poses: **6835**
Average Distance travelled by each robot: **231.67 m**

Field Experiments (5 Robots)

Klaus Building (3rd Floor)
Georgia Tech

Size of the map: **151 m X 118 m**
Average Number of Poses: **6835**
Average Distance travelled by each robot: **231.67 m**

Field Experiments (10 Robots)

Military Training Facility

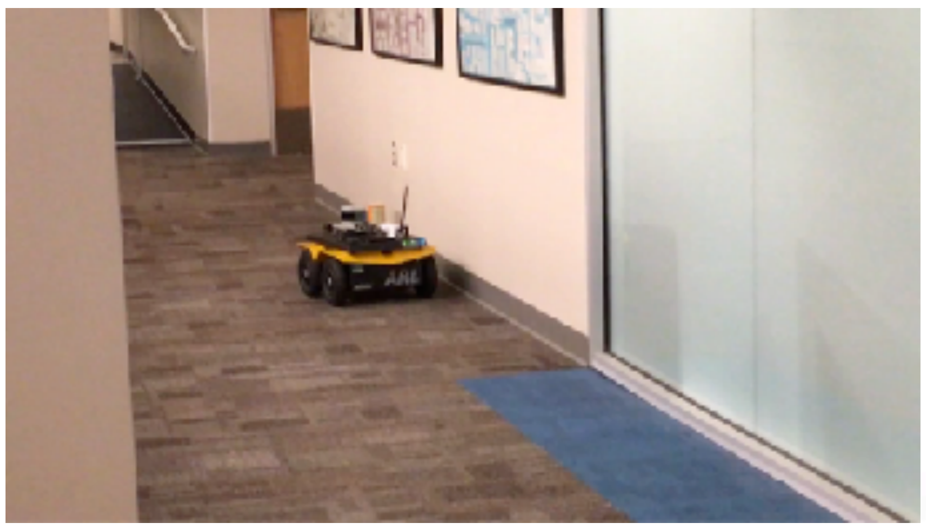
Size of the map: **15.8 m X 11.8 m**
Average Number of Poses: **4564**
Average Distance travelled by each robot: **20.1 m**

Field Experiments (10 Robots)

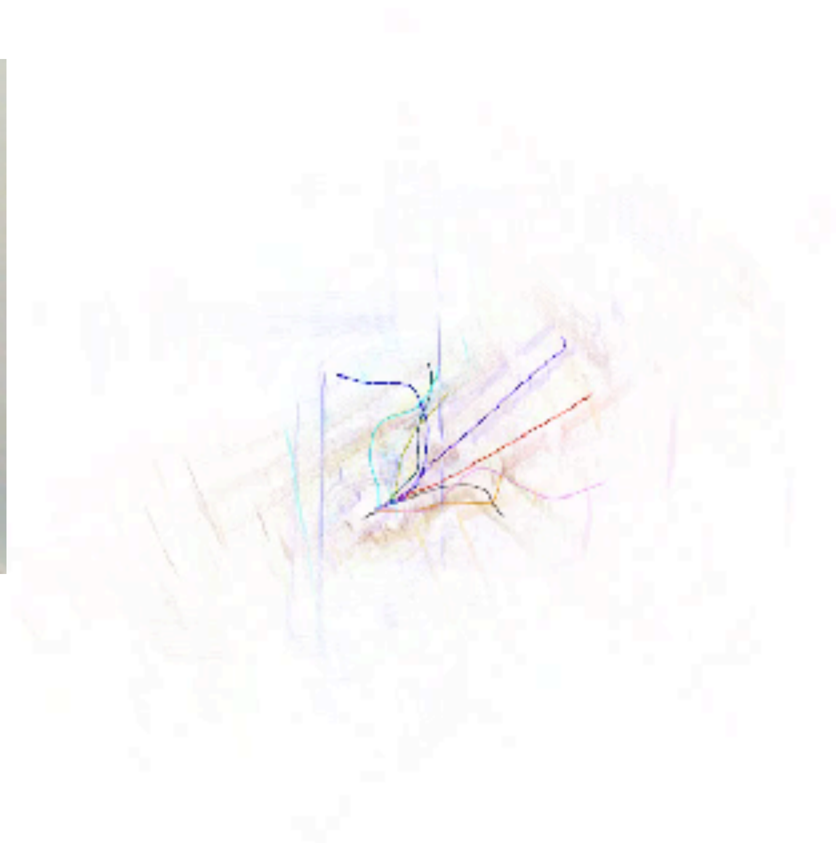
Military Training Facility

Size of the map: **15.8 m X 11.8 m**
Average Number of Poses: **4564**
Average Distance travelled by each robot: **20.1 m**

Field Experiments (11 Robots)

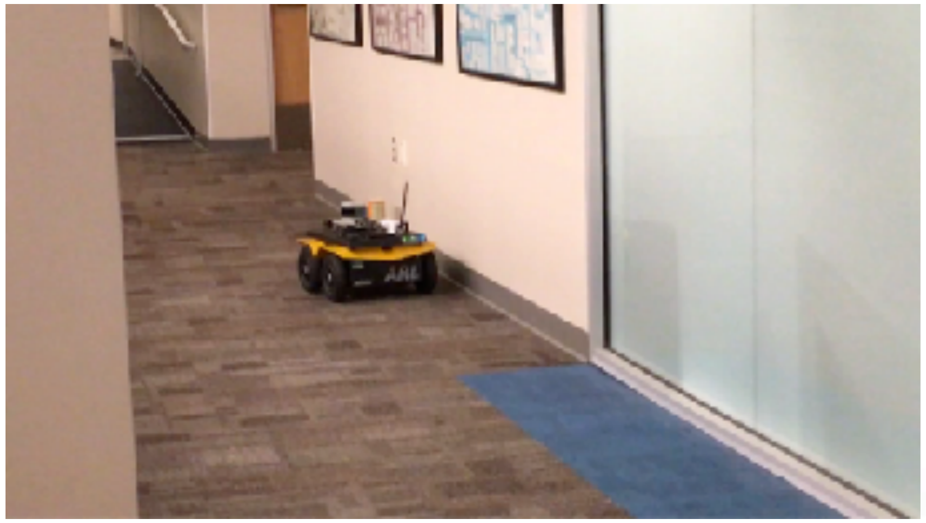


IRIM Lab
Georgia Tech



Size of the map: **55 m X 61 m**
Average Number of Poses: **3995**
Average Distance travelled by each robot: **72.4 m**

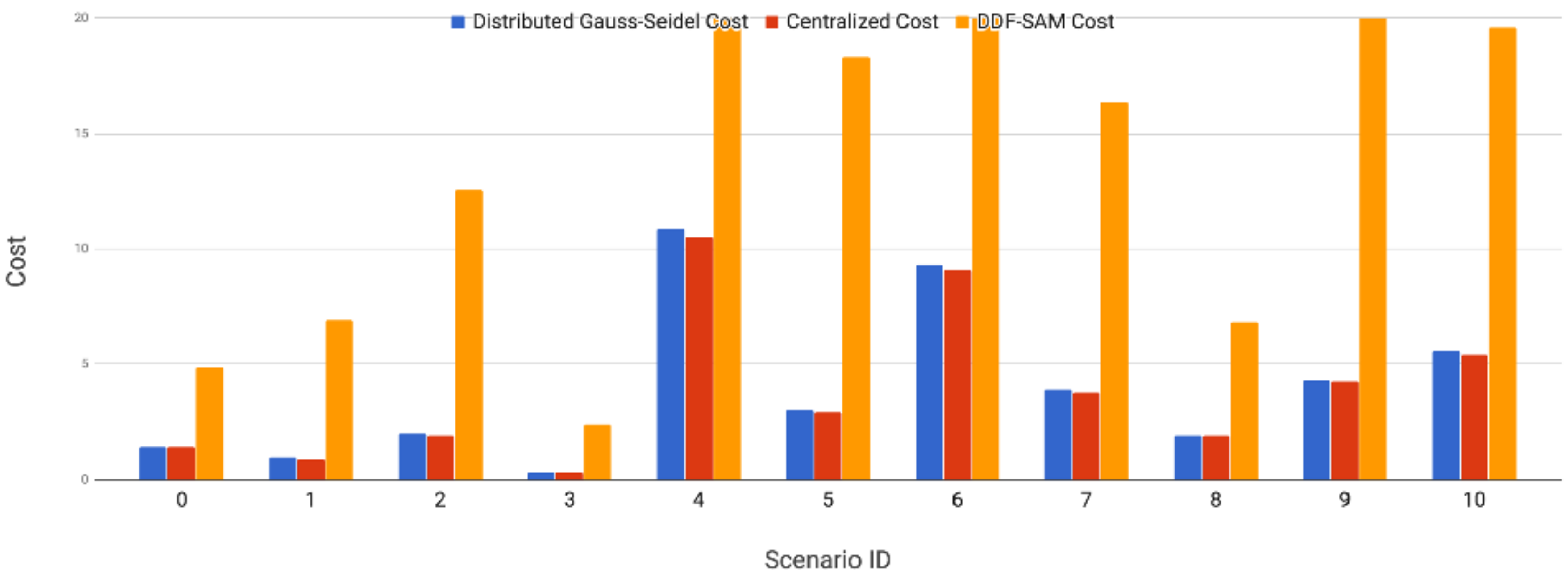
Field Experiments (11 Robots)



IRIM Lab
Georgia Tech

Size of the map: **55 m X 61 m**
Average Number of Poses: **3995**
Average Distance travelled by each robot: **72.4 m**

Performance of our approach on Field Data



So far…

state of art distributed optimizer

+

*Using **objects** as landmarks*

↓ ***distributed SLAM***

1. ***reduce** the **communication bandwidth and the memory***
2. *improves the **robustness and scalability***
3. *outputs a **human understandable map***

So far…

$\boxed{\textit{\textbf{state of art distributed optimizer}}}$ **Distributed Gauss-Seidel**

+

*Using **objects** as landmarks*

distributed SLAM

1. ***reduce** the **communication bandwidth and the memory***

2. *improves the **robustness and scalability***

3. *outputs a **human understandable map***

So far…

$\boxed{\textbf{\textit{state of art distributed optimizer}}}$

Distributed Gauss-Seidel

+

*Using **objects** as landmarks*

distributed SLAM

1. ***reduce** the $\boxed{\textbf{\textit{communication bandwidth}}}$ **and the memory***

   Linear in Communication

2. *improves the **robustness and scalability***

3. *outputs a **human understandable map***

So far…

$\boxed{\textit{\textbf{state of art distributed optimizer}}}$ — **Distributed Gauss-Seidel**

+

*Using **objects** as landmarks*

distributed SLAM

1. ***reduce** the* $\boxed{\textit{\textbf{communication bandwidth}}}$ ***and the memory*** — **Linear in Communication**
2. *improves the* $\boxed{\textit{\textbf{robustness and scalability}}}$
   - **Robust to bad initialization**
   - **Resilient to measurement noise**
   - **Scalable in the number of robots**
3. *outputs a **human understandable map***

Remaining Issues…

state of art distributed optimizer

+

*Using **objects** as landmarks*

↓ ***distributed SLAM***

1. ***reduce** the **communication bandwidth and the memory***
2. *improves the **robustness and scalability***
3. *outputs a **human understandable map***

Requires exchange of 3D point clouds to establish communication link among robots.

Each robot has to store point cloud map which increases their memory requirement

Outline

1. Distributed Gauss-Seidel Approach
   (*state of art distributed optimizer*)

2. Distributed Object-based SLAM with Known Object Models
   (*using objects as landmarks*)

3. Distributed Object-based SLAM with Joint Object Modeling and Mapping
   (*jointly model objects along with SLAM*)

4. Conclusions and Future Work

Outline

1. Distributed Gauss-Seidel Approach
   (*state of art distributed optimizer*)

2. Distributed Object-based SLAM with Known Object Models
   (*using objects as landmarks*)

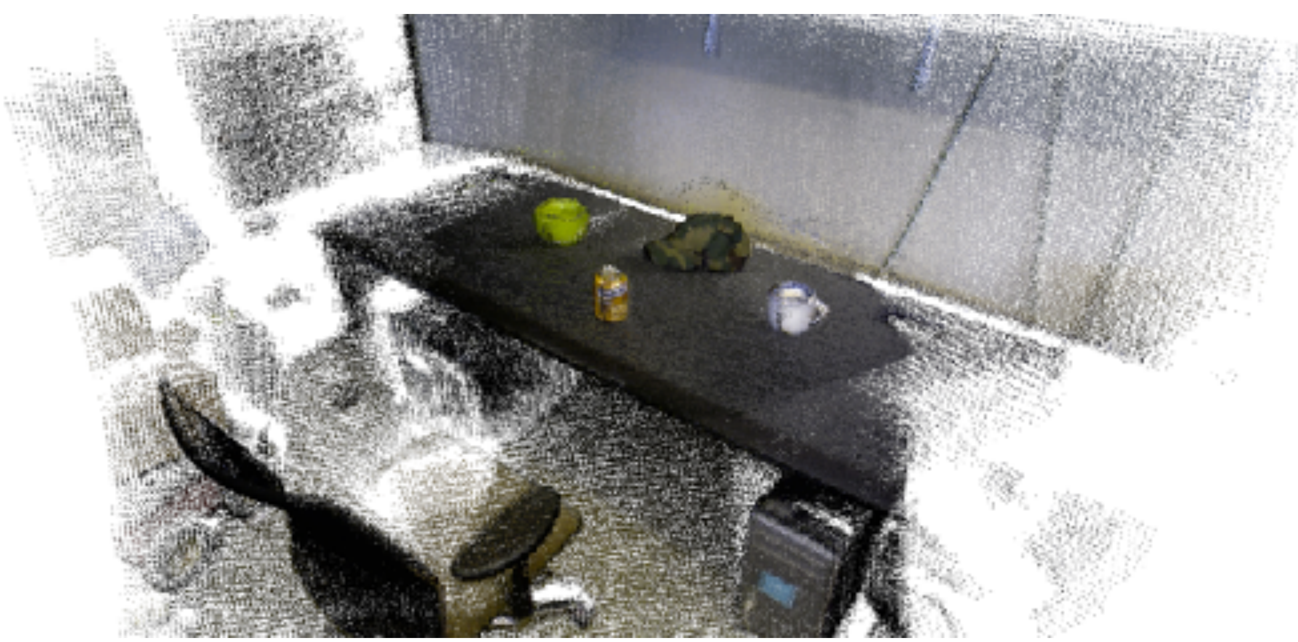
3. Distributed Object-based SLAM with Joint Object Modeling and Mapping
   (*jointly model objects along with SLAM*)

4. Conclusions and Future Work

Why should we use objects?

Why should we use objects?

1. **Reduces memory requirements** and **information exchange** among robots compared to point cloud based representation



>

~13 MB

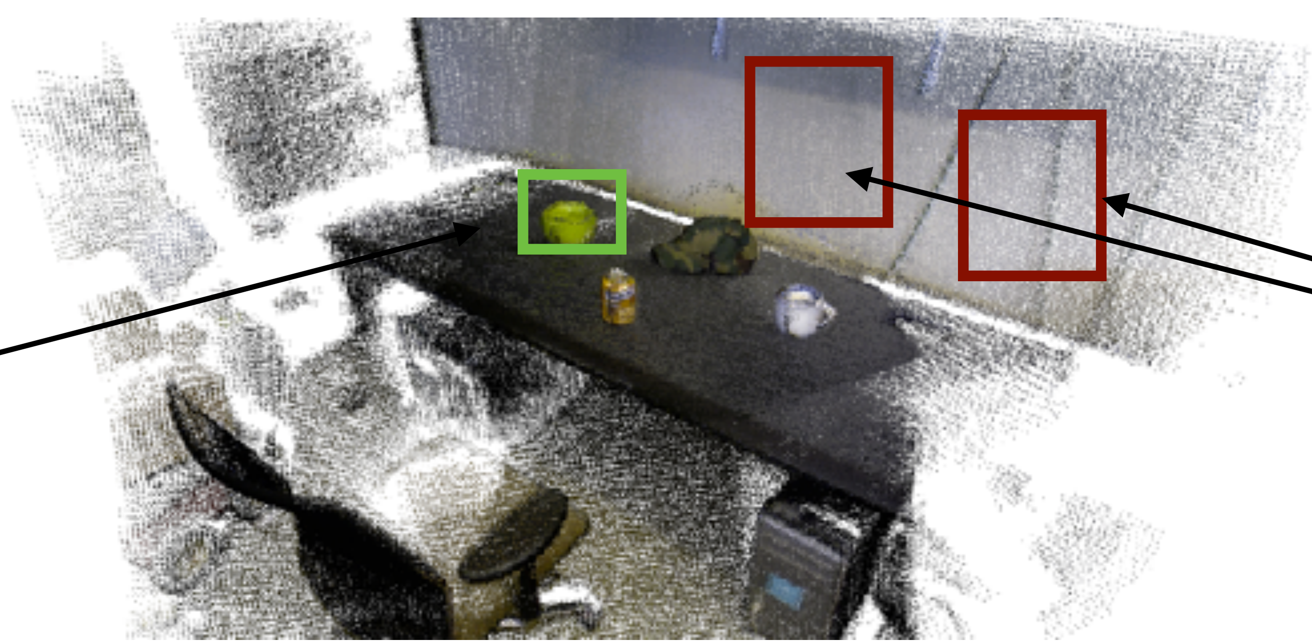
~1 KB

Why should we use objects?

1. **Reduces memory requirements** and **information exchange** among robots compared to point cloud based representation

2. Objects are **more discriminative** as compared to point clouds.

objects look dissimilar to the points in its surrounding



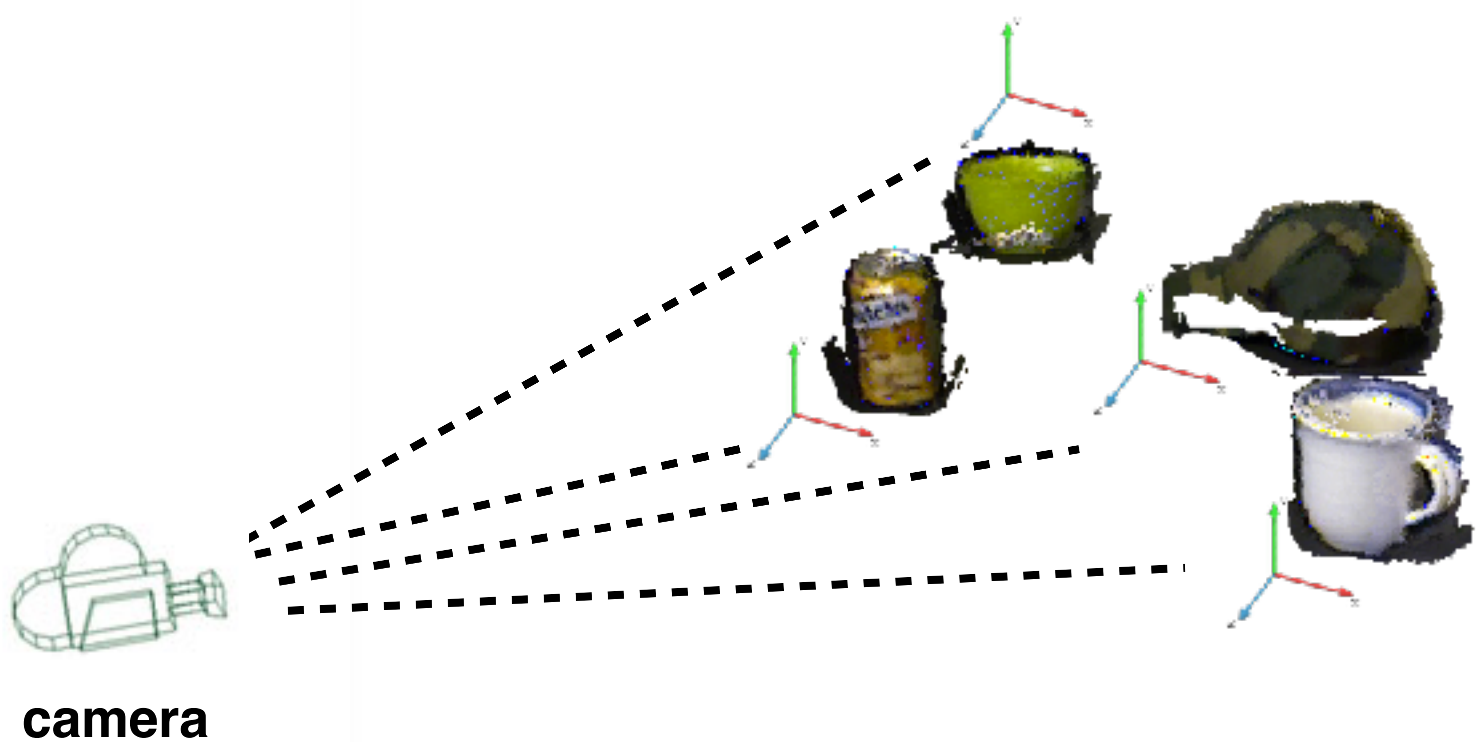
each point looks similar to the points in its surrounding

Why should we use objects?

1. **Reduces memory requirements** and **information exchange** among robots compared to point cloud based representation

2. Objects are **more discriminative** as compared to point clouds.

3. **Reduces** the **computational complexity** of SLAM.



Why should we use objects?

1. **Reduces memory requirements** and **information exchange** among robots compared to point cloud based representation

2. Objects are **more discriminative** as compared to point clouds.

3. **Reduces** the **computational complexity** of SLAM.

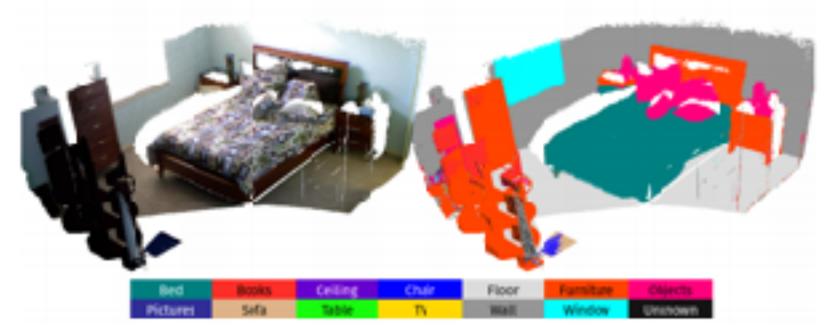
4. Results in maps that are easier for **humans** to understand.

Related Work

Related Work

- B. Kuipers (AI 2000)
- Mozos et al. (RAS 2007)
- Nieto et al. (IROS 2010)
- Koppula et al. (NIPS 2011)
- Pronobis et al. (IJRR 2009, ICRA 2012)
- Kim et al. (TOG 2012)
- Dame et al. (CVPR 2013)
- Finman et al. (ECMR 2013)
- Kundu el al. (ECCV 2014)
- Valentin et al. (TOG 2015)
- Vineet et al. (ICRA 2015)
- Miksik et al. (IROS 2015)
- McCormac et al. (arXiv 2016)

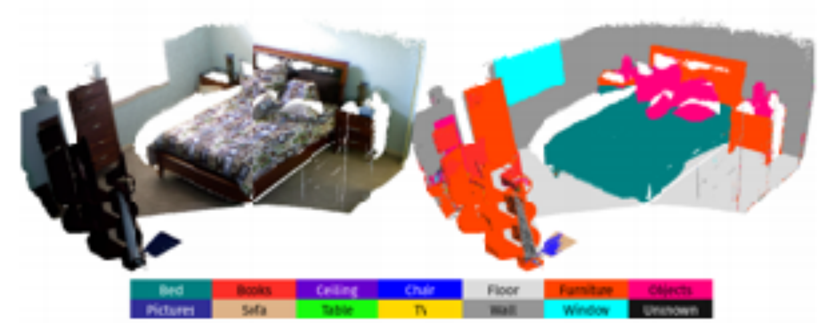
Dense Semantic Mapping



Related Work

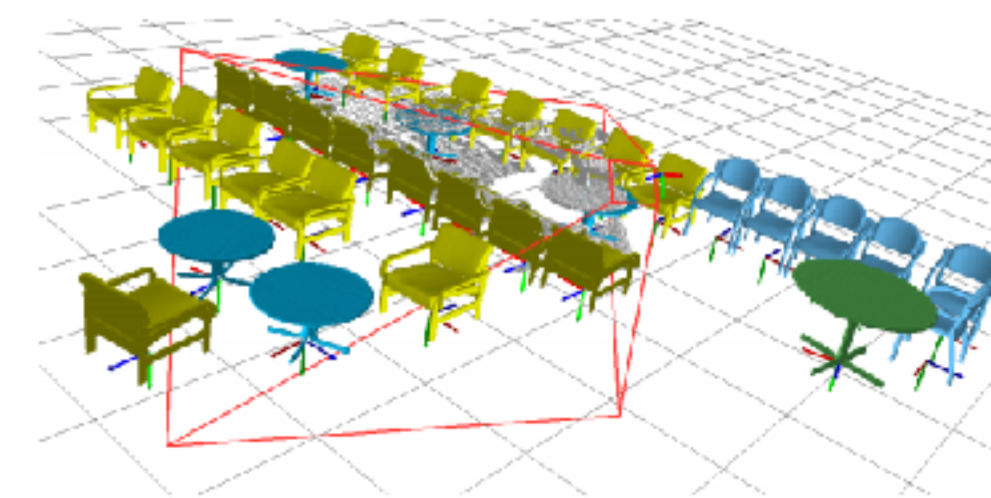
- B. Kuipers (AI 2000)
- Mozos et al. (RAS 2007)
- Nieto et al. (IROS 2010)
- Koppula et al. (NIPS 2011)
- Pronobis et al. (IJRR 2009, ICRA 2012)
- Kim et al. (TOG 2012)
- Dame et al. (CVPR 2013)
- Finman et al. (ECMR 2013)
- Kundu el al. (ECCV 2014)
- Valentin et al. (TOG 2015)
- Vineet et al. (ICRA 2015)
- Miksik et al. (IROS 2015)
- McCormac et al. (arXiv 2016)

Dense Semantic Mapping



- Castle et al. (ICRA 2007)
- Ranganthan and Dellaert (RSS 2007)
- Rogers et al. (IROS 2011)
- Civera et al. (IROS 2011)
- Trevor et al. (ICRA 2012, HRI 2013)
- Bao et al. (CVPR 2012)
- Salas-Moreno et al. (CVPR 2013)
- Song and Chandraker (CVPR 2015)
- Pillai et al. (RSS 2015)
- Singhal et al. (arXiv 2016)
- Sunderhauf et al. (arXiv 2016)

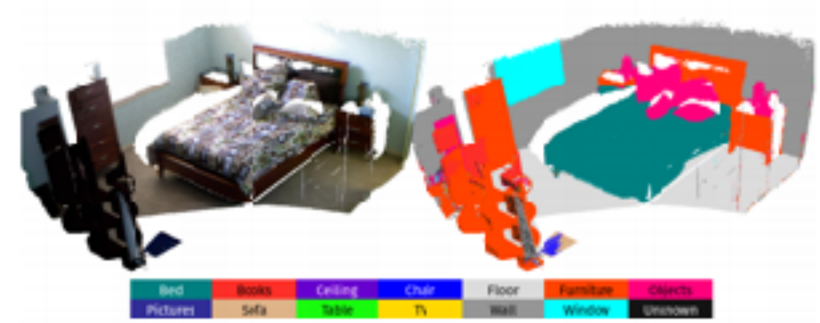
Object level Mapping



Related Work

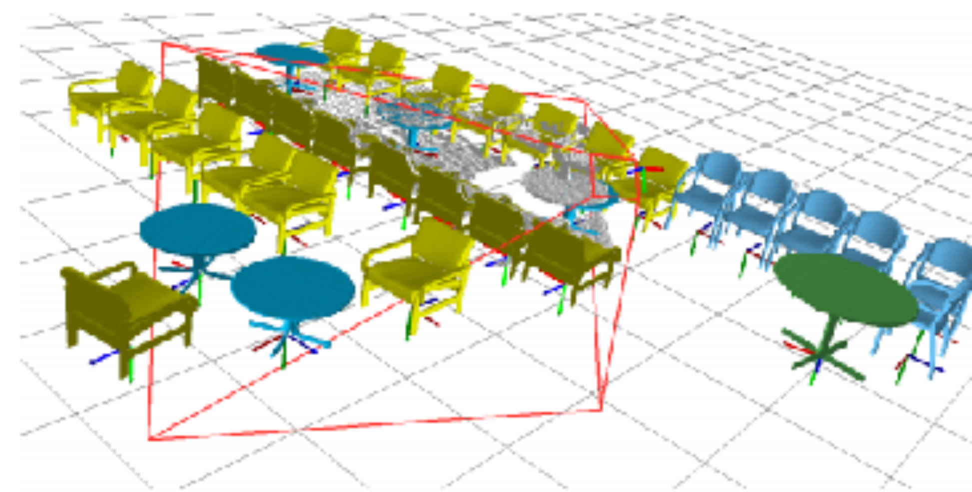
- B. Kuipers (AI 2000)
- Mozos et al. (RAS 2007)
- Nieto et al. (IROS 2010)
- Koppula et al. (NIPS 2011)
- Pronobis et al. (IJRR 2009, ICRA 2012)
- Kim et al. (TOG 2012)
- Dame et al. (CVPR 2013)
- Finman et al. (ECMR 2013)
- Kundu el al. (ECCV 2014)
- Valentin et al. (TOG 2015)
- Vineet et al. (ICRA 2015)
- Miksik et al. (IROS 2015)
- McCormac et al. (arXiv 2016)

Dense Semantic Mapping



- Castle et al. (ICRA 2007)
- Ranganthan and Dellaert (RSS 2007)
- Rogers et al. (IROS 2011)
- Civera et al. (IROS 2011)
- Trevor et al. (ICRA 2012, HRI 2013)
- Bao et al. (CVPR 2012)
- Salas-Moreno et al. (CVPR 2013)
- Song and Chandraker (CVPR 2015)
- Pillai et al. (RSS 2015)
- Singhal et al. (arXiv 2016)
- Sunderhauf et al. (arXiv 2016)

Object level Mapping

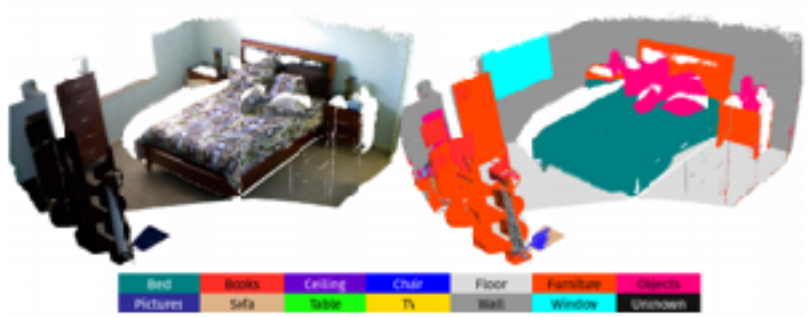


- **Fast and accurate object detection using Convolutional Neural Networks.**

Related Work

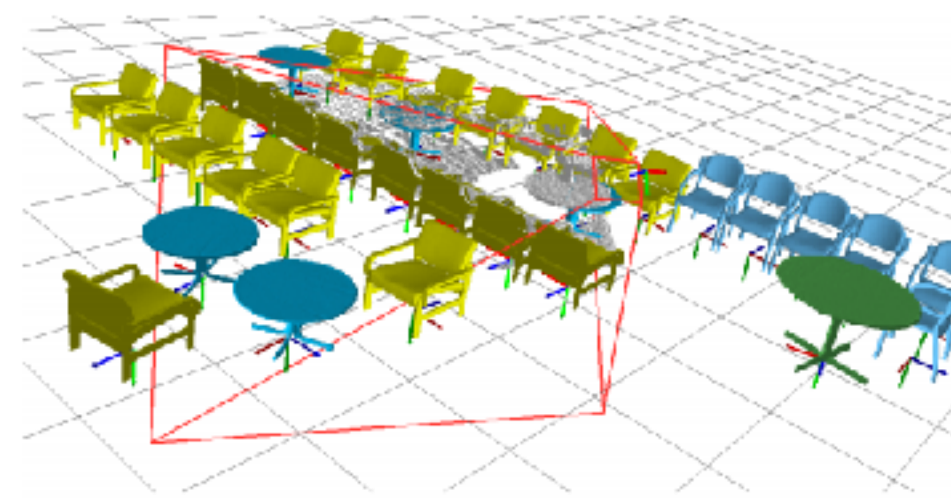
- B. Kuipers (AI 2000)
- Mozos et al. (RAS 2007)
- Nieto et al. (IROS 2010)
- Koppula et al. (NIPS 2011)
- Pronobis et al. (IJRR 2009, ICRA 2012)
- Kim et al. (TOG 2012)
- Dame et al. (CVPR 2013)
- Finman et al. (ECMR 2013)
- Kundu el al. (ECCV 2014)
- Valentin et al. (TOG 2015)
- Vineet et al. (ICRA 2015)
- Miksik et al. (IROS 2015)
- McCormac et al. (arXiv 2016)

Dense Semantic Mapping



- Castle et al. (ICRA 2007)
- Ranganthan and Dellaert (RSS 2007)
- Rogers et al. (IROS 2011)
- Civera et al. (IROS 2011)
- Trevor et al. (ICRA 2012, HRI 2013)
- Bao et al. (CVPR 2012)
- Salas-Moreno et al. (CVPR 2013)
- Song and Chandraker (CVPR 2015)
- Pillai et al. (RSS 2015)
- Singhal et al. (arXiv 2016)
- Sunderhauf et al. (arXiv 2016)

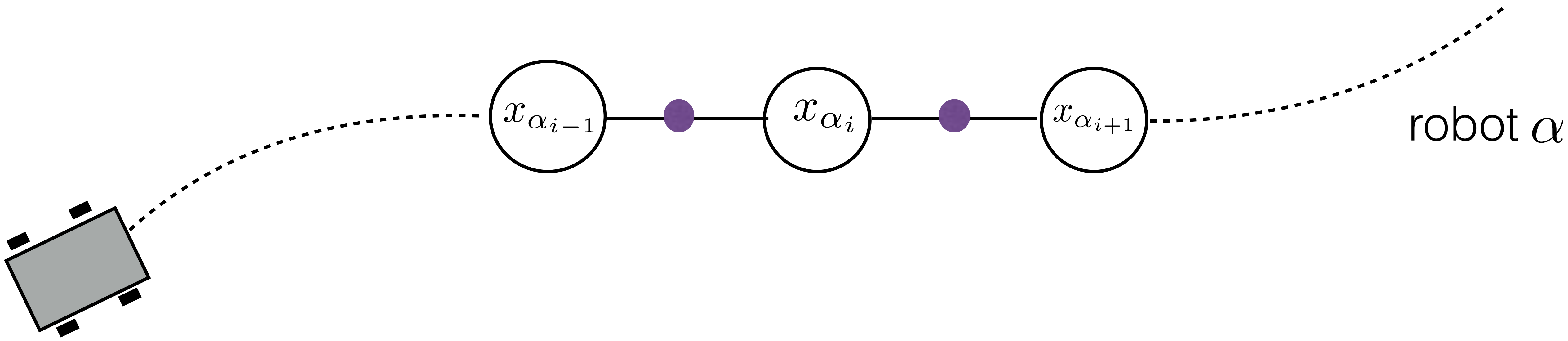
Object level Mapping



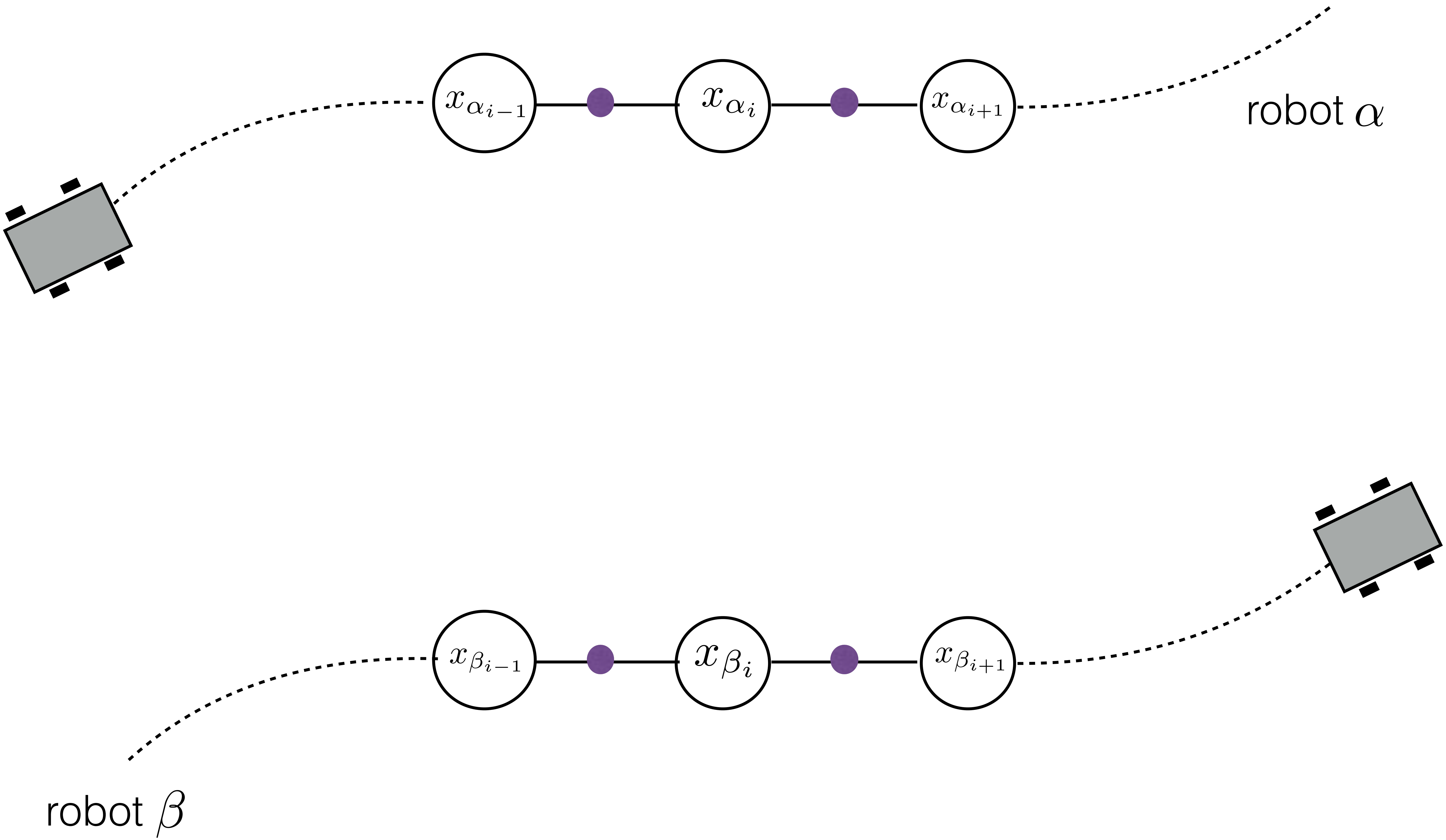
- **Fast and accurate object detection using Convolutional Neural Networks.**
- **Extend the current approaches to multi robot setting.**

Distributed Object-based SLAM

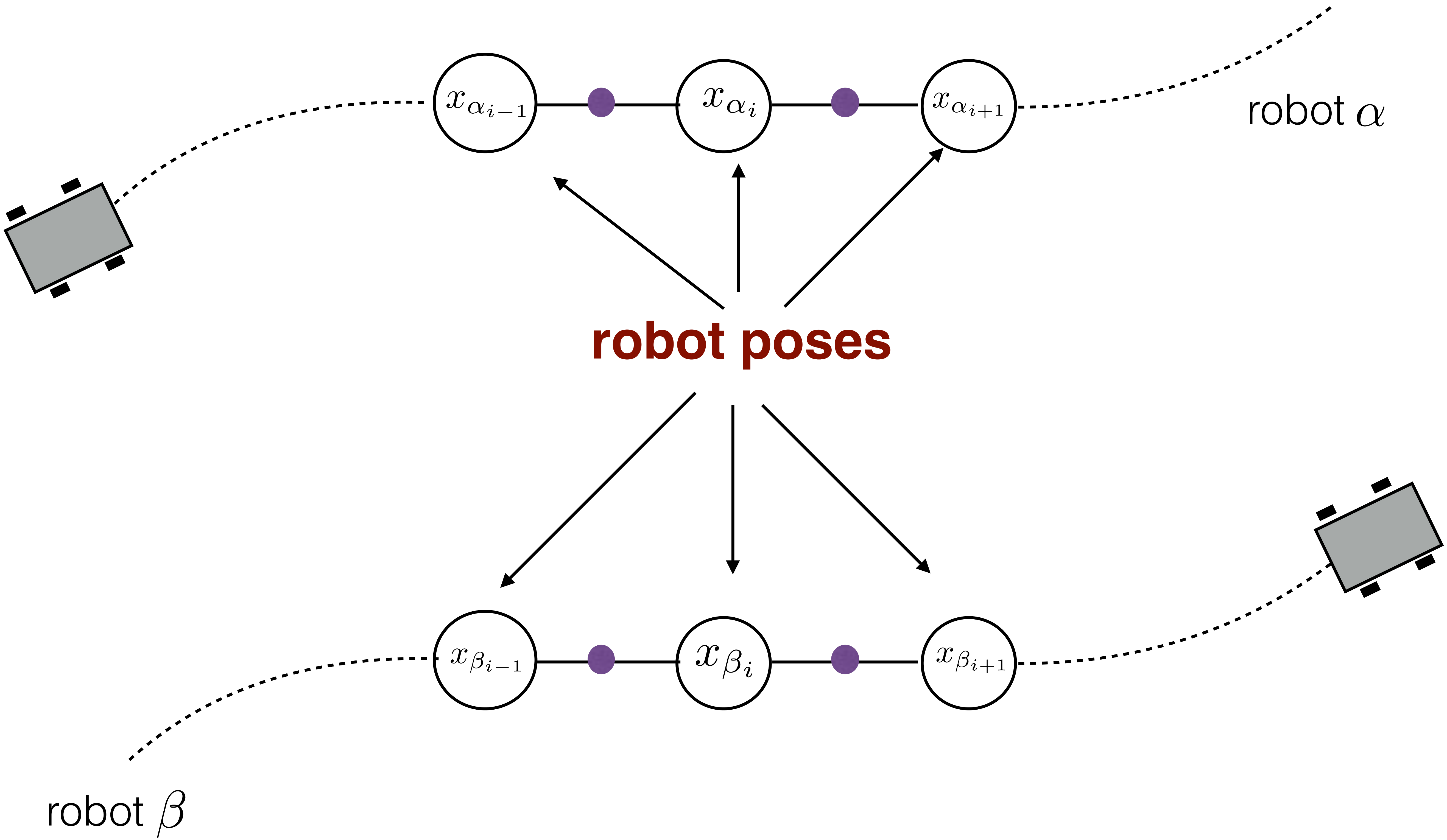
Distributed Object-based SLAM



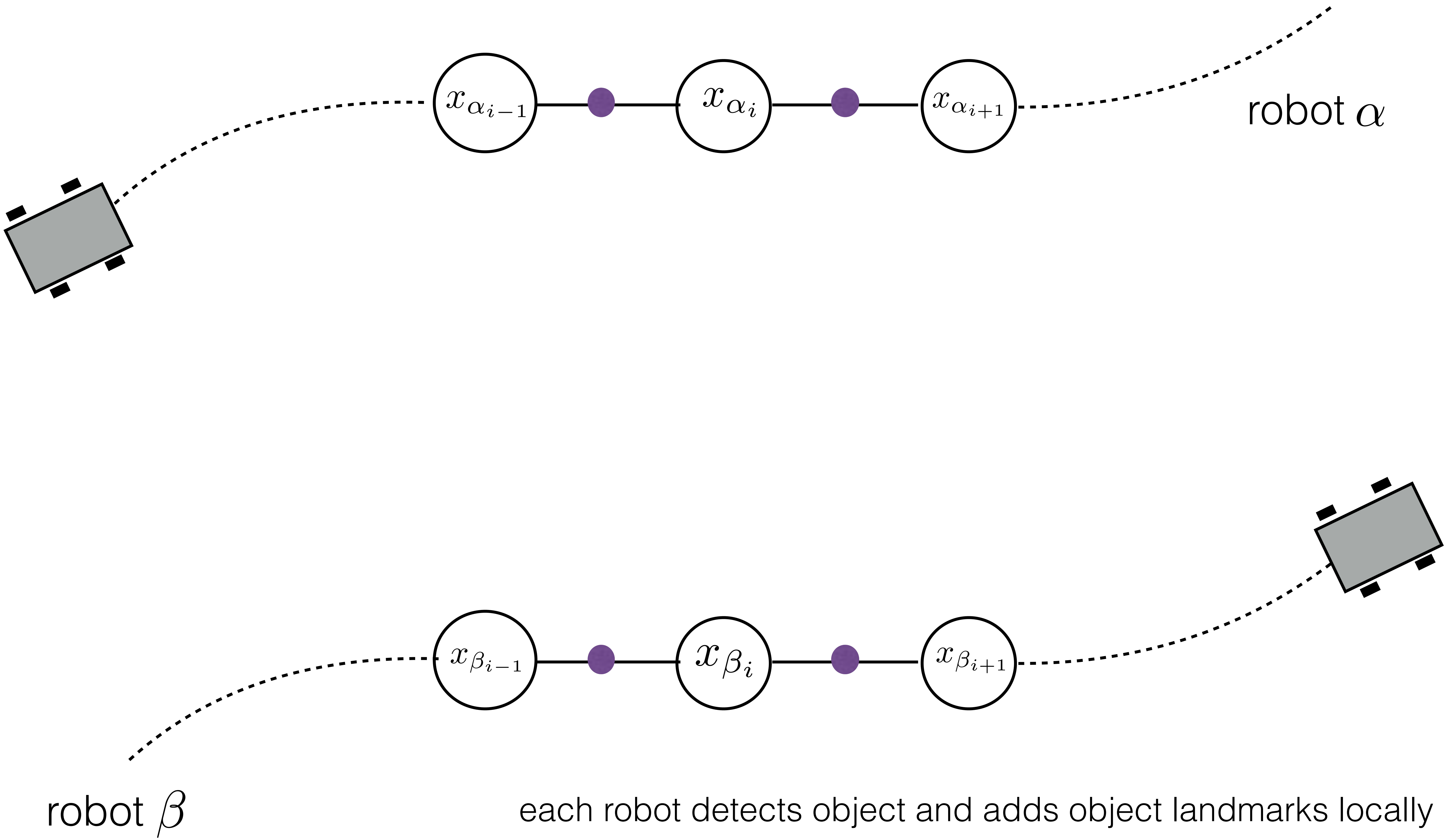
Distributed Object-based SLAM



Distributed Object-based SLAM

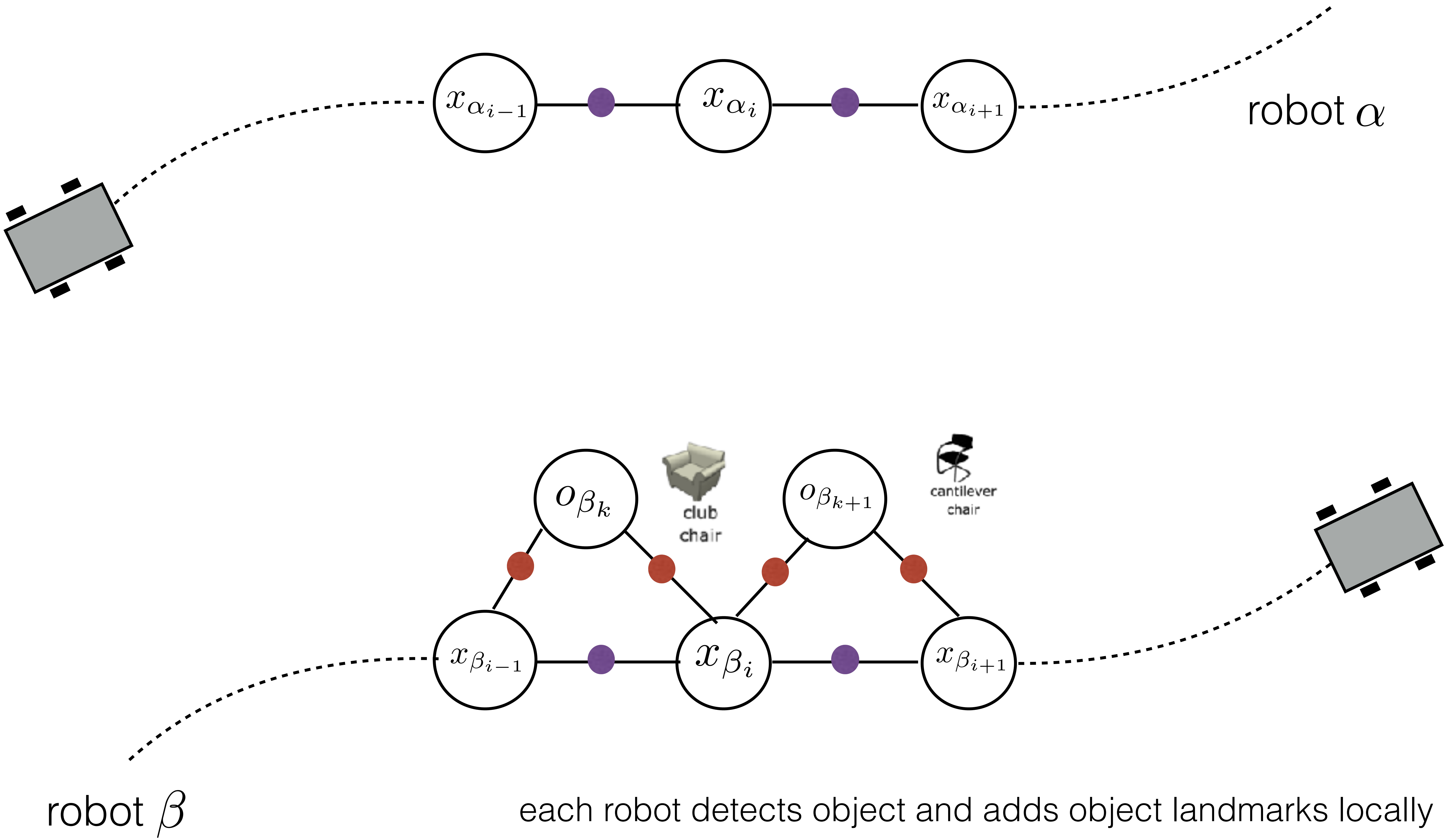


Distributed Object-based SLAM



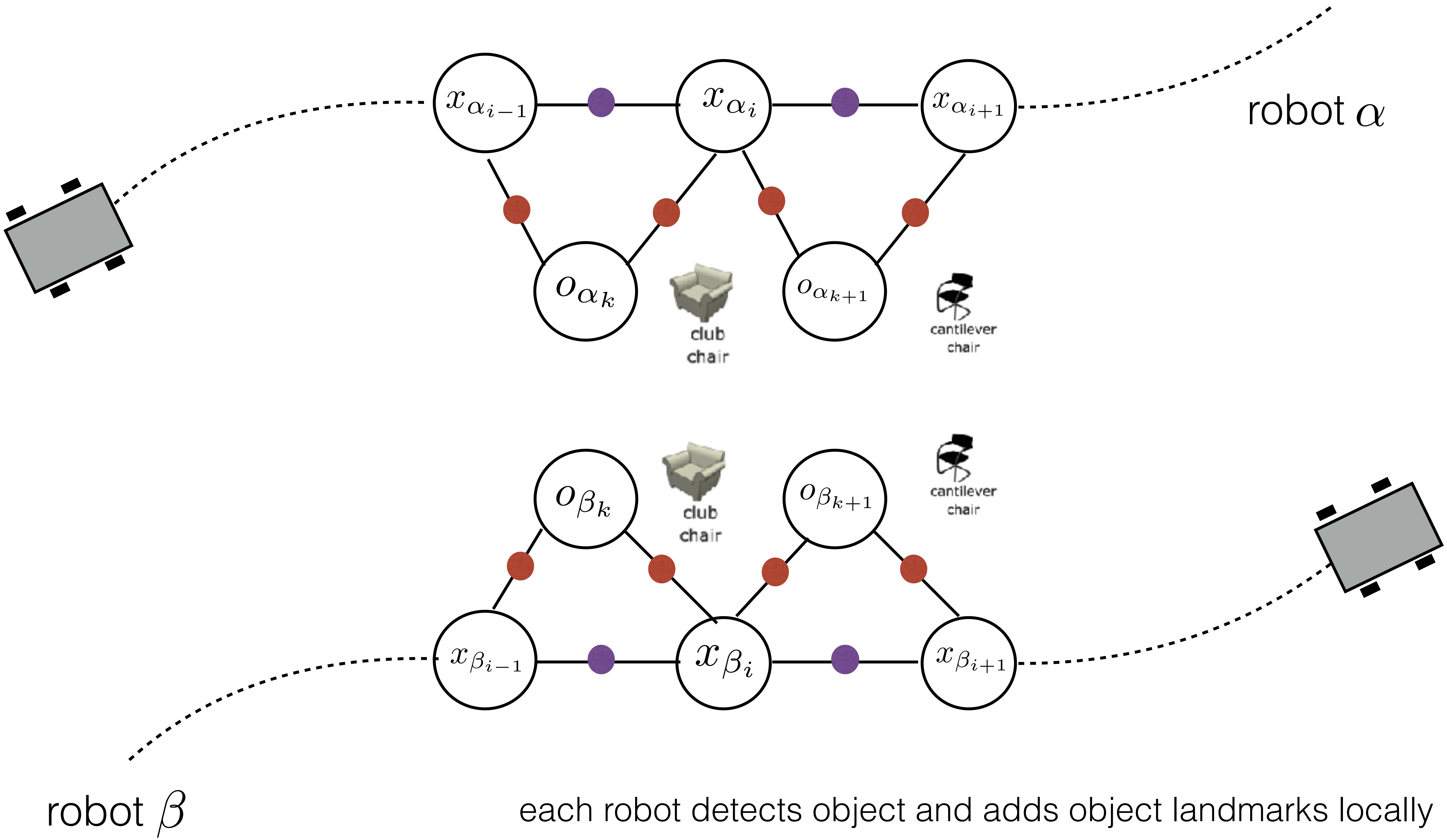
each robot detects object and adds object landmarks locally

Distributed Object-based SLAM



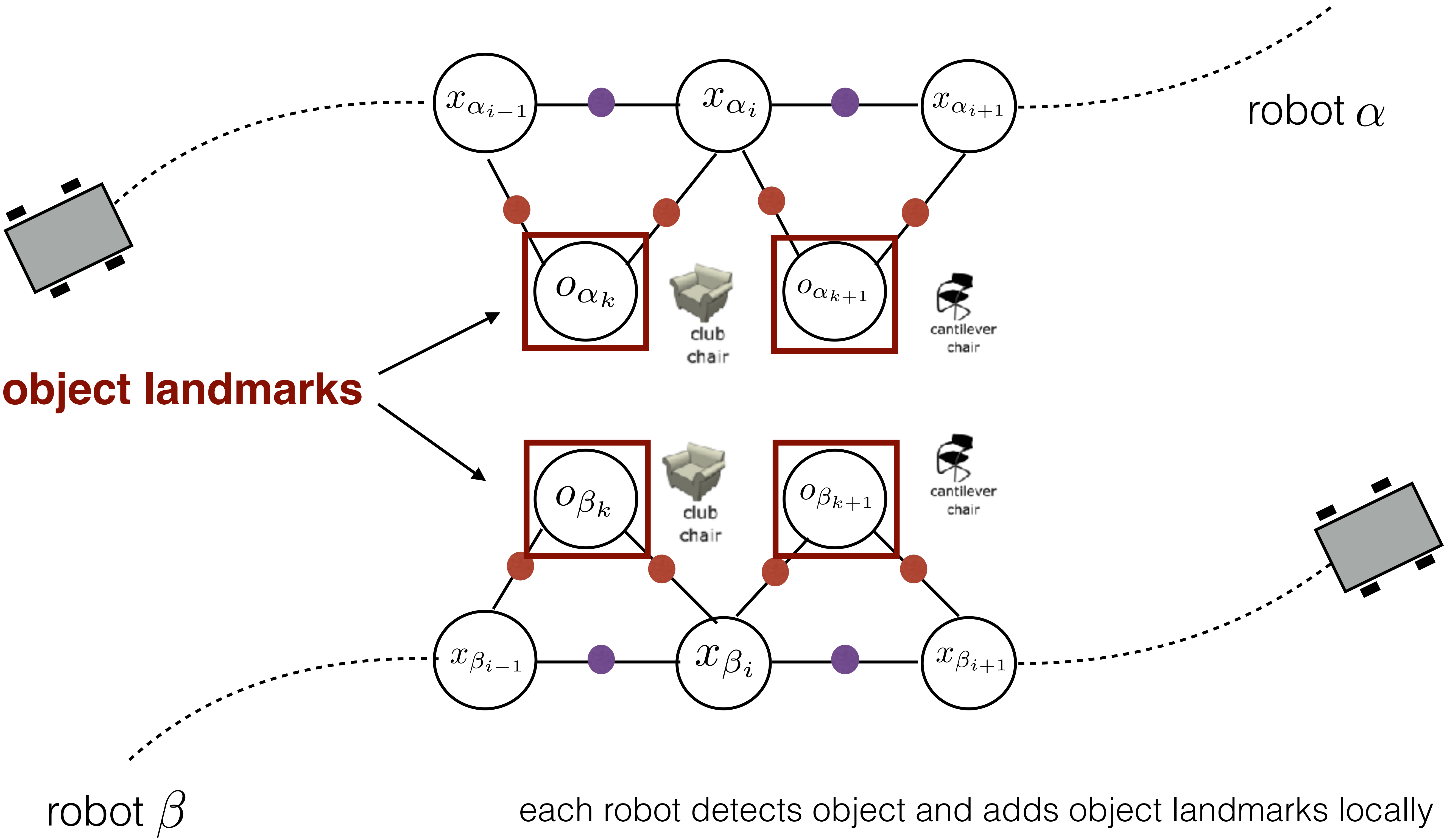
each robot detects object and adds object landmarks locally

Distributed Object-based SLAM



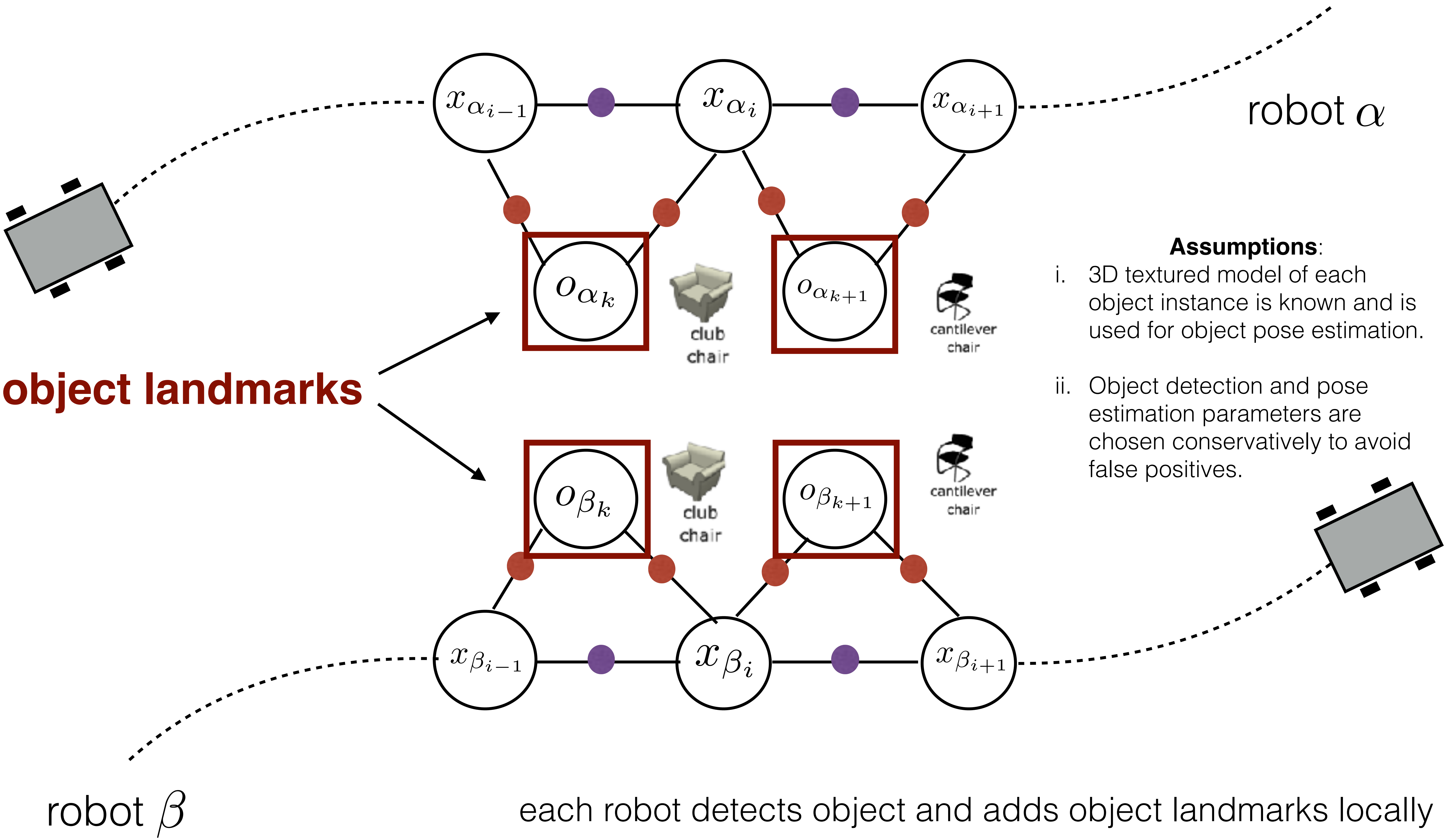
each robot detects object and adds object landmarks locally

Distributed Object-based SLAM



each robot detects object and adds object landmarks locally

Distributed Object-based SLAM



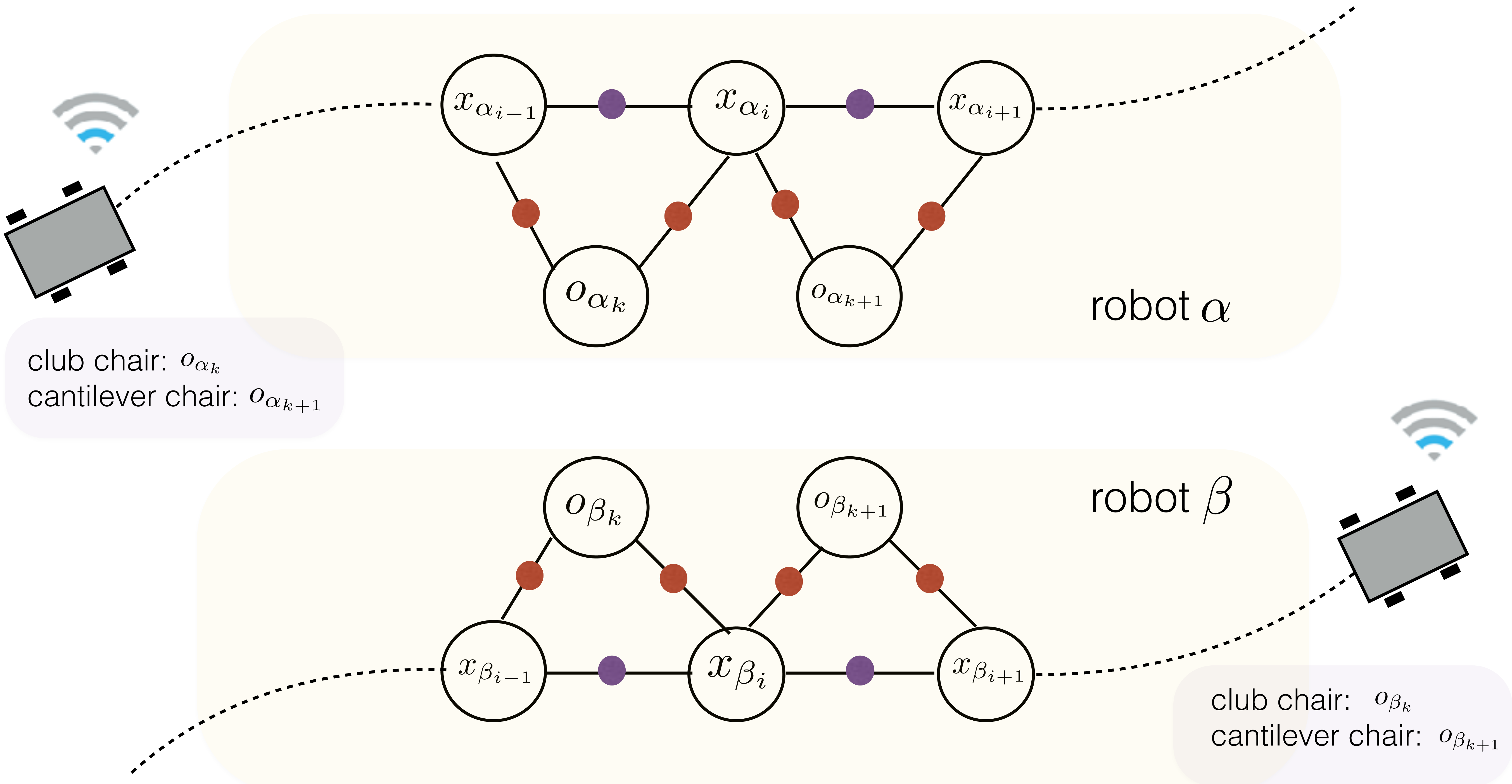
Assumptions:

i. 3D textured model of each object instance is known and is used for object pose estimation.

ii. Object detection and pose estimation parameters are chosen conservatively to avoid false positives.

each robot detects object and adds object landmarks locally

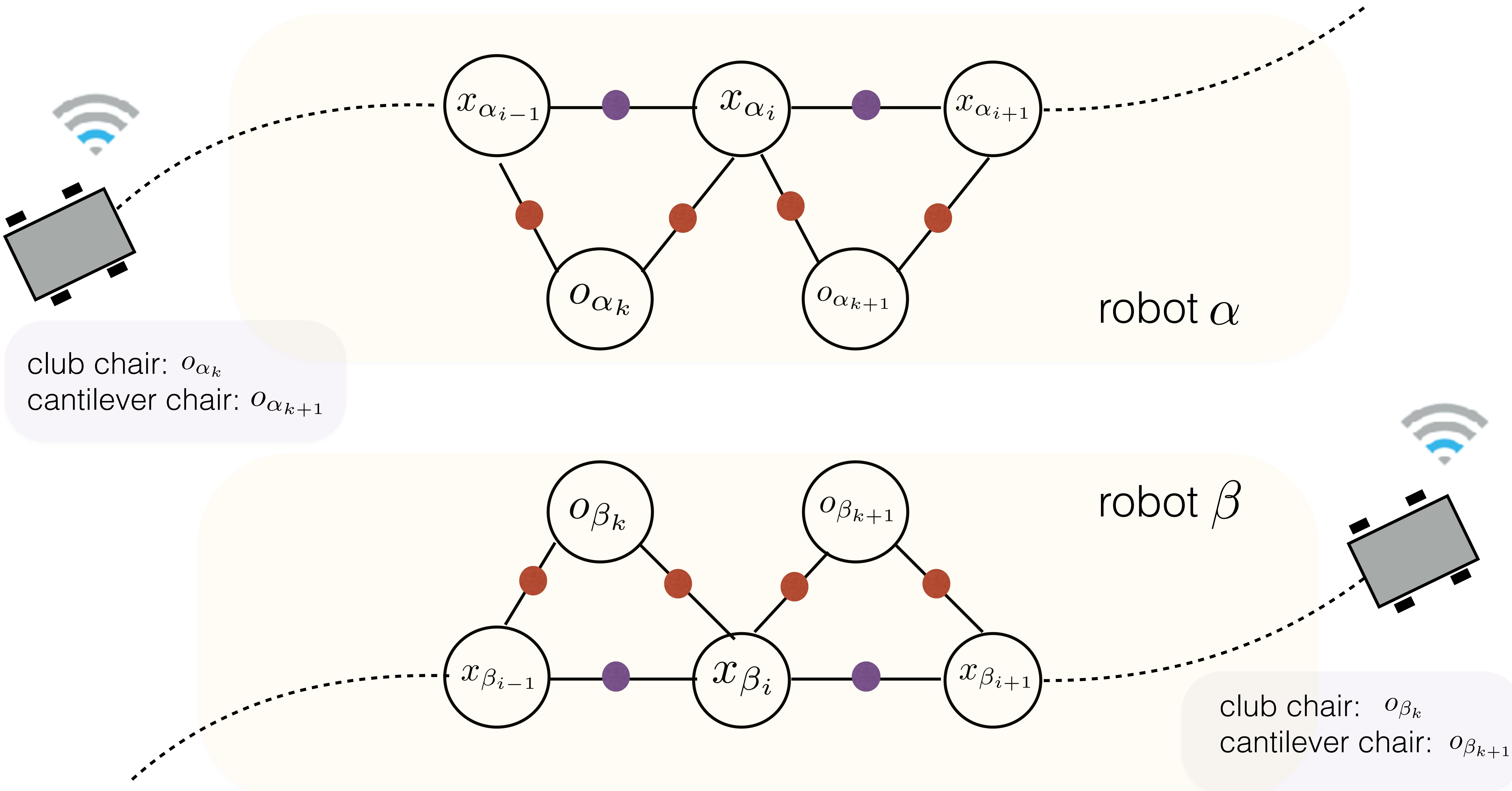
Distributed Object-based SLAM



Assumption: initial position of each robot is known to all the other robots

each robot communicates the list of objects to the other robot

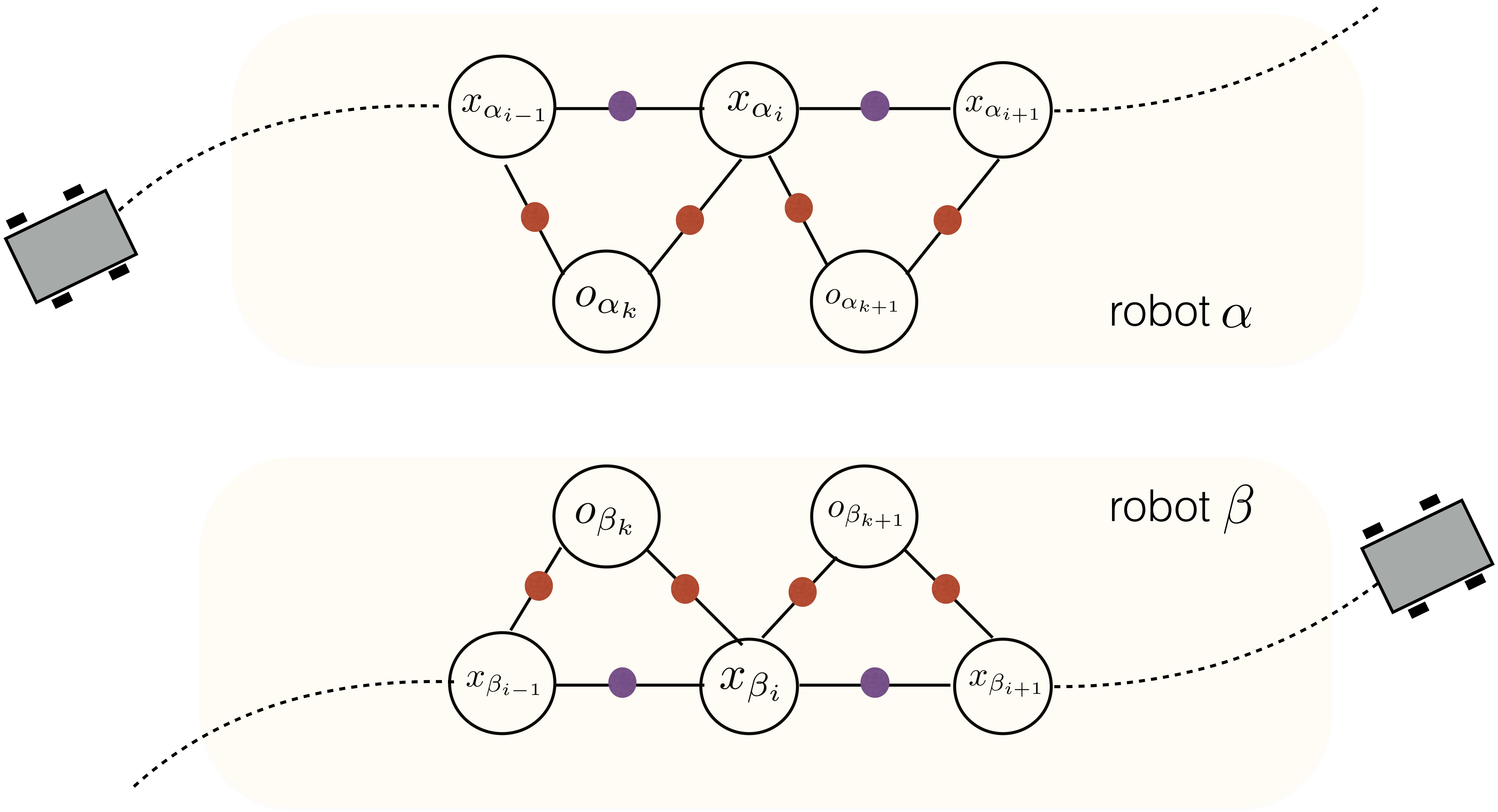
Distributed Object-based SLAM



Assumption: initial position of each robot is known to all the other robots

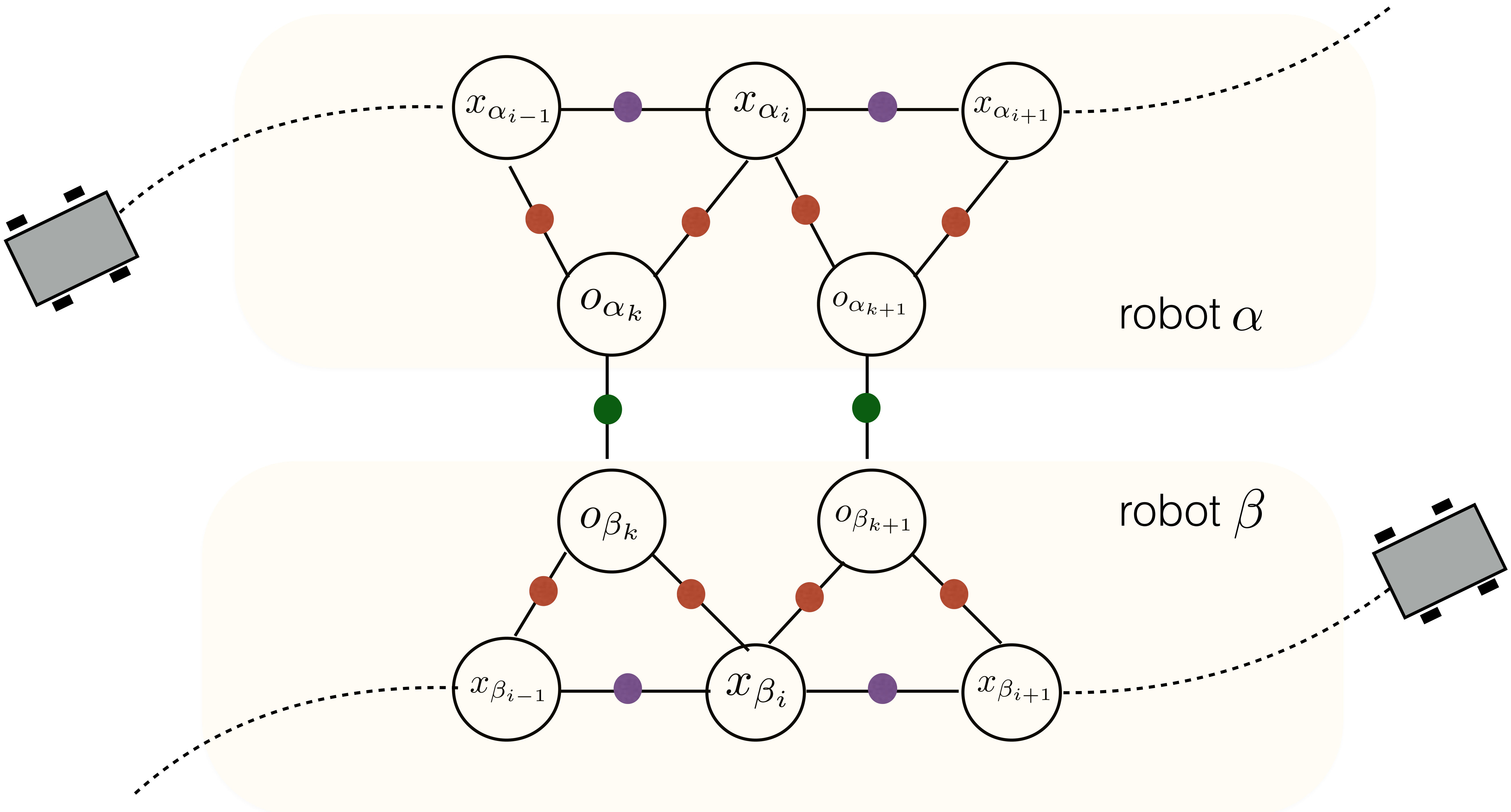
each robot communicates the list of objects to the other robot

Distributed Object-based SLAM



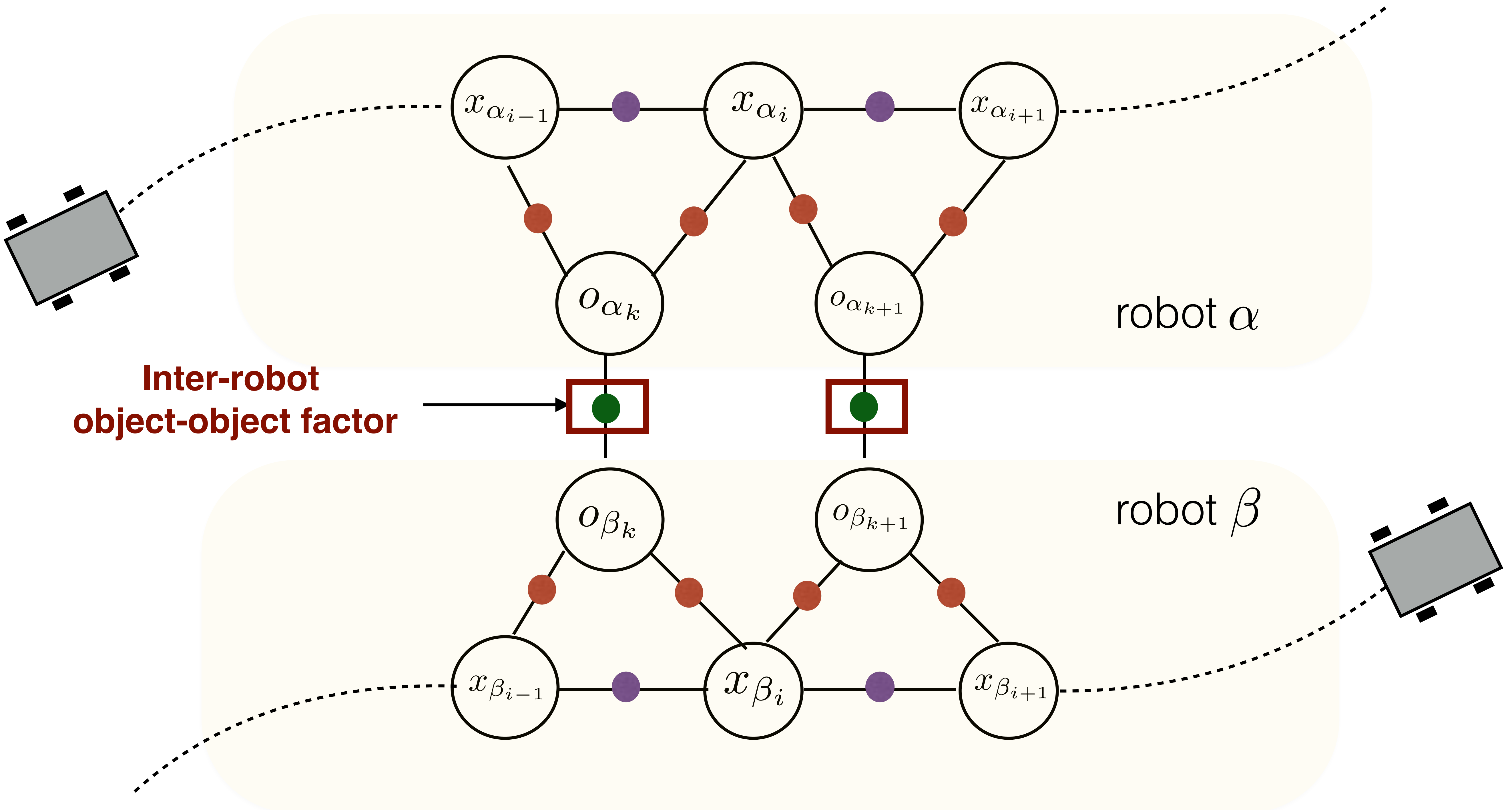
loop closure constraints are added between the common set of objects seen by both the robots

Distributed Object-based SLAM



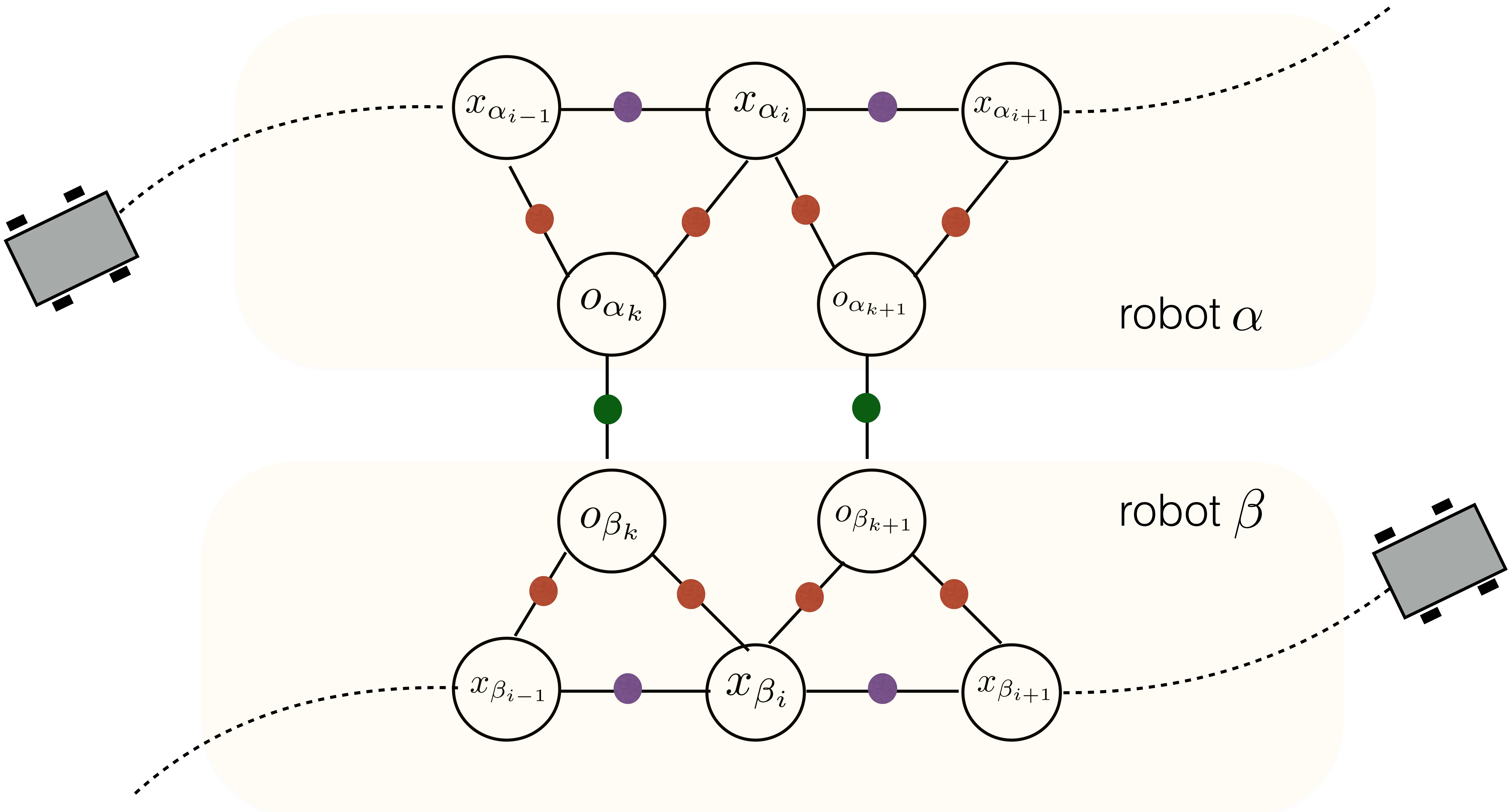
loop closure constraints are added between the common set of objects seen by both the robots

Distributed Object-based SLAM



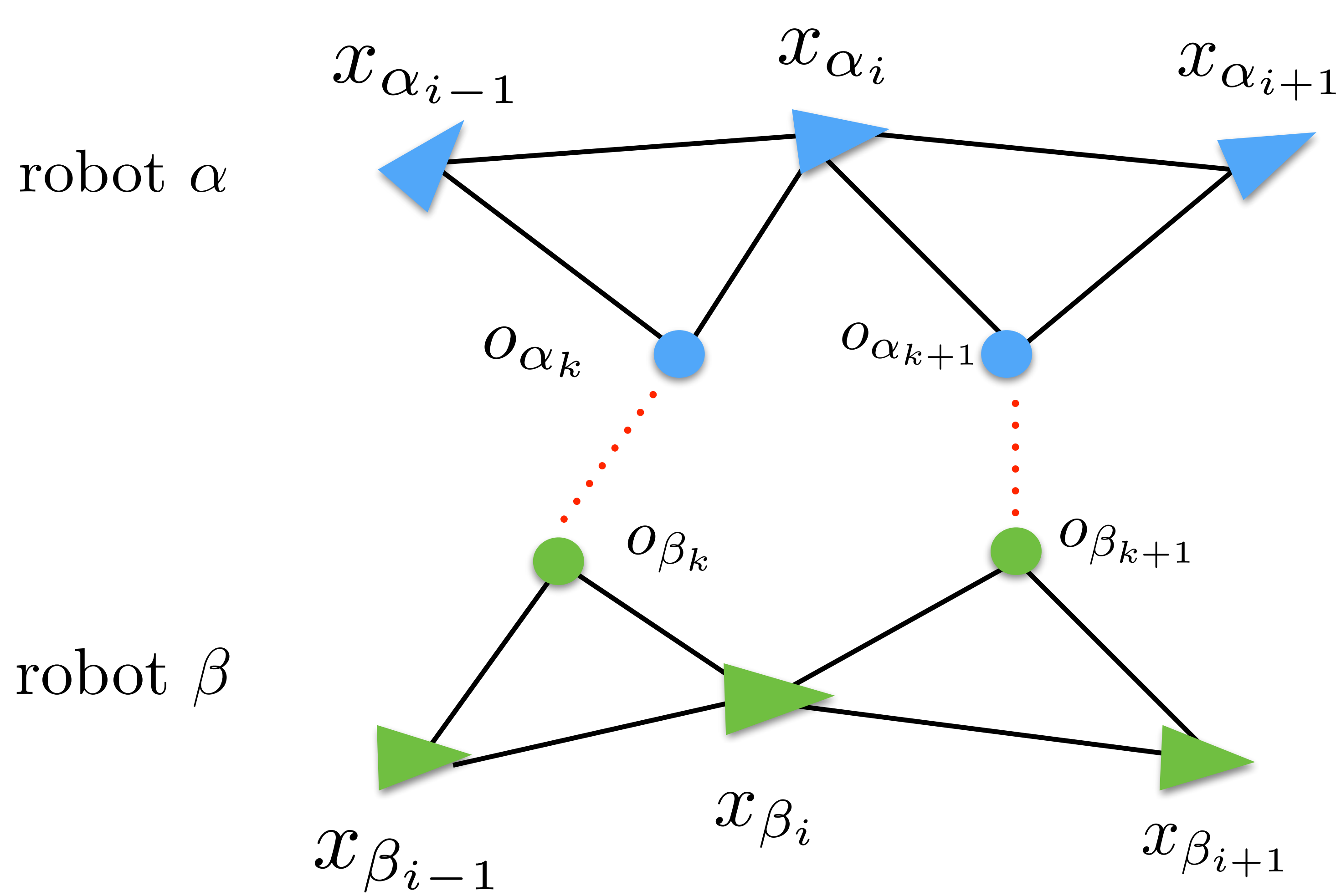
loop closure constraints are added between the common set of objects seen by both the robots

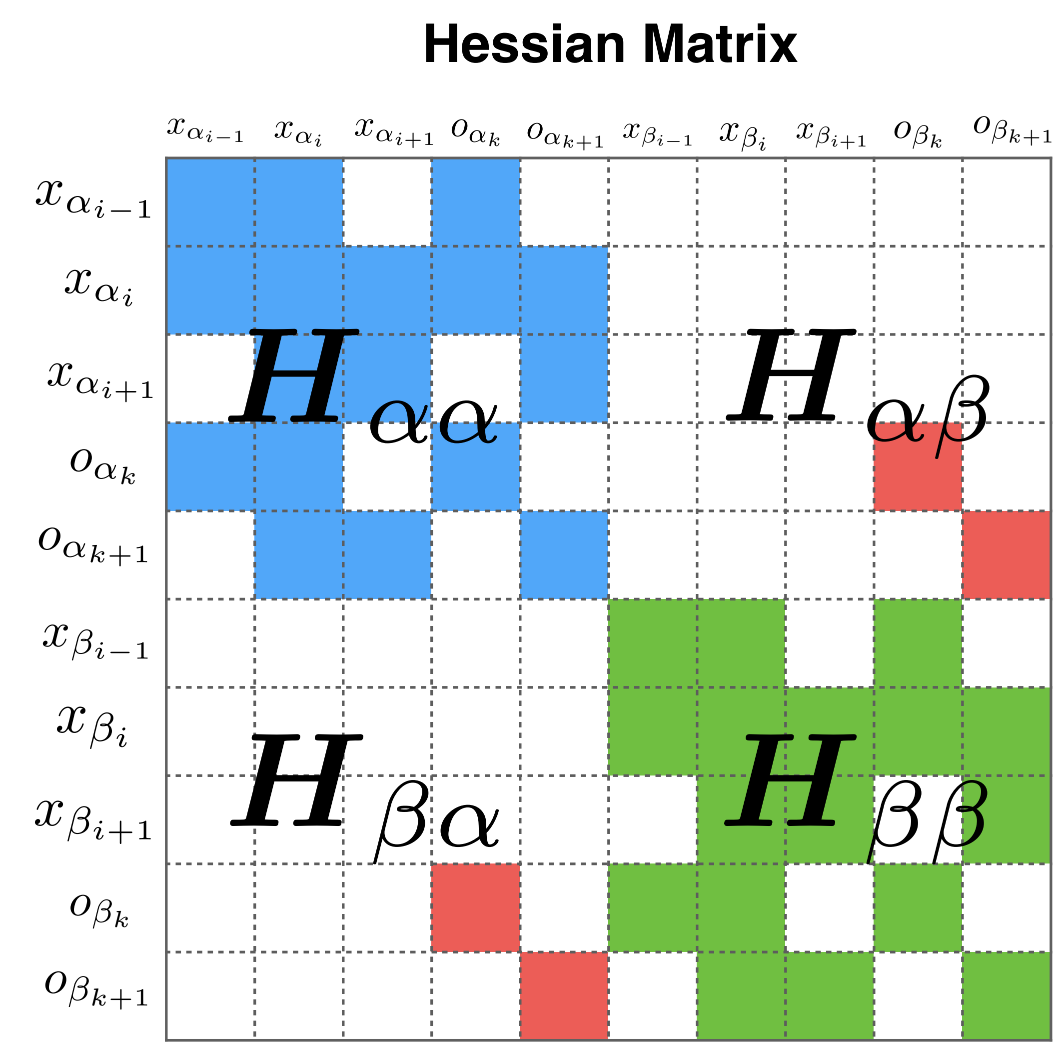
Distributed Object-based SLAM



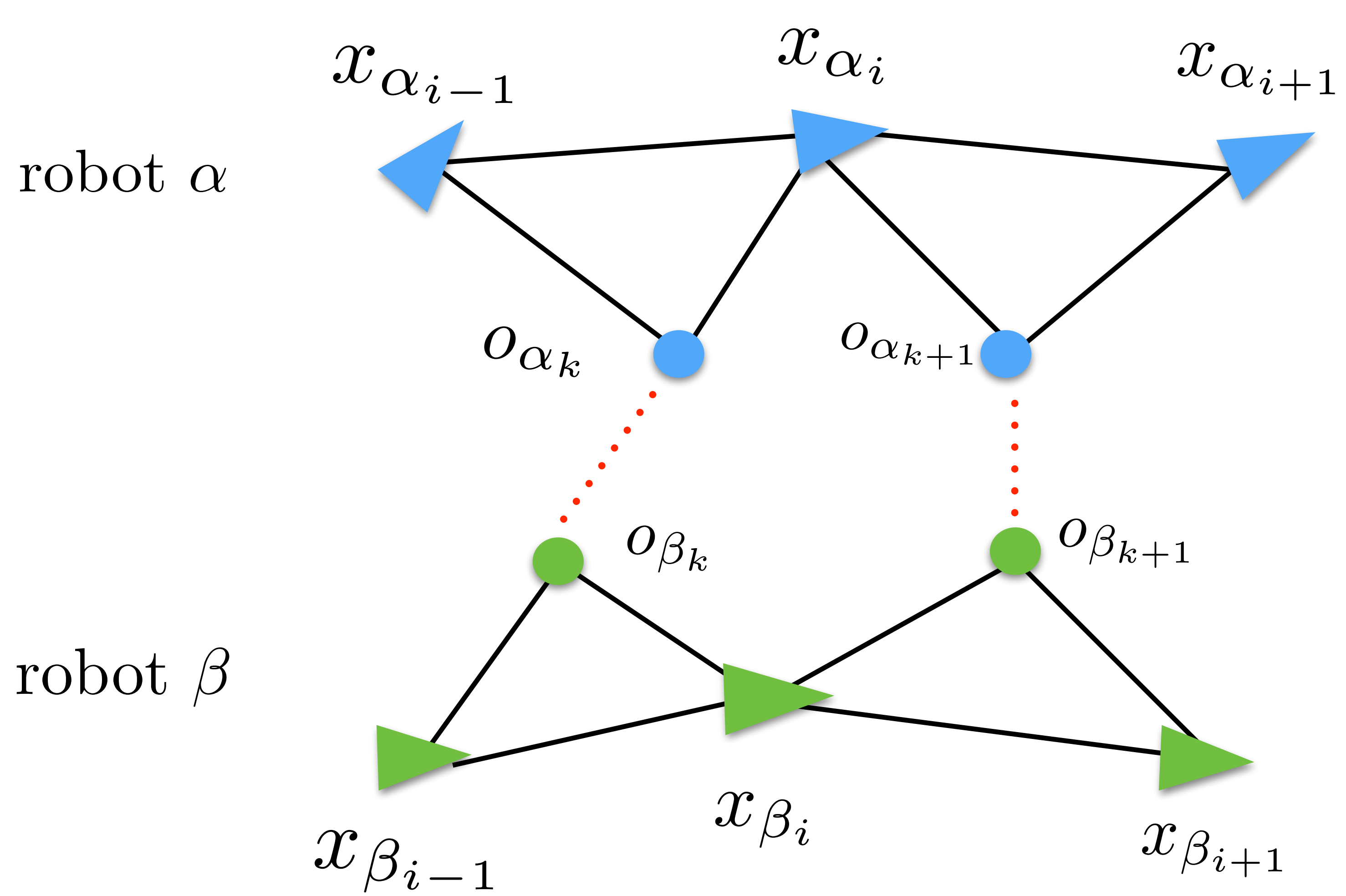
Factor graph optimized using Distributed Gauss-Seidel Algorithm

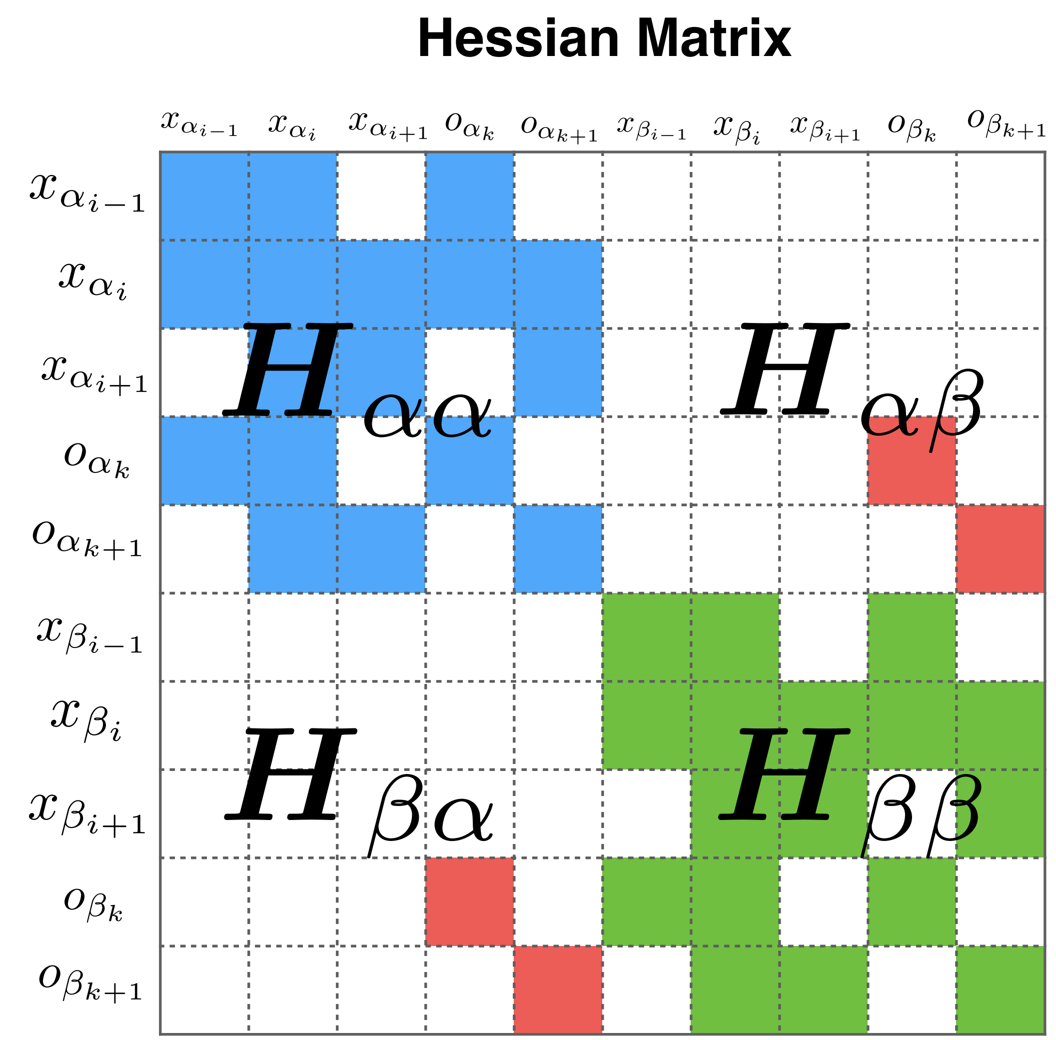
Distributed Object-based SLAM





Distributed Object-based SLAM



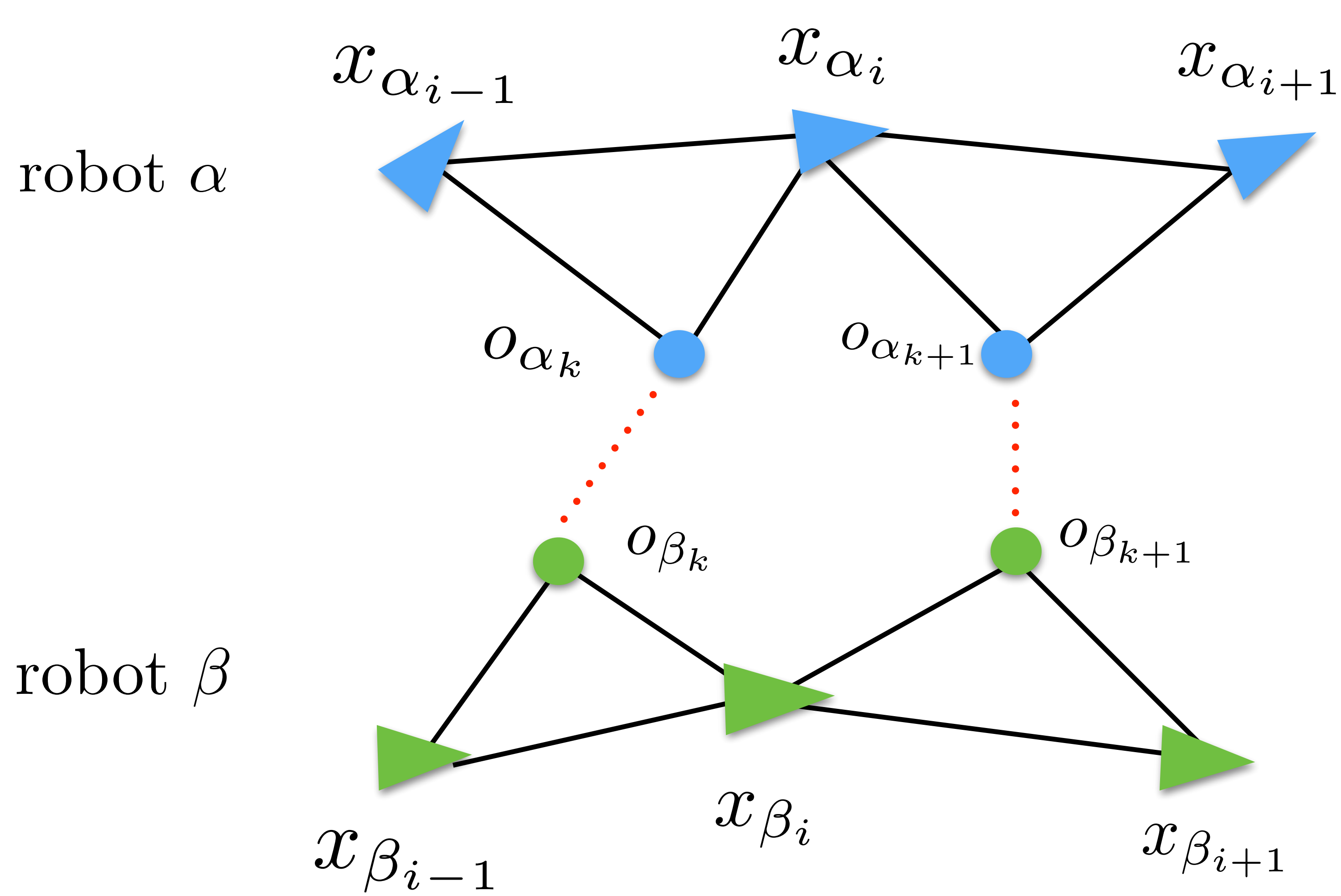


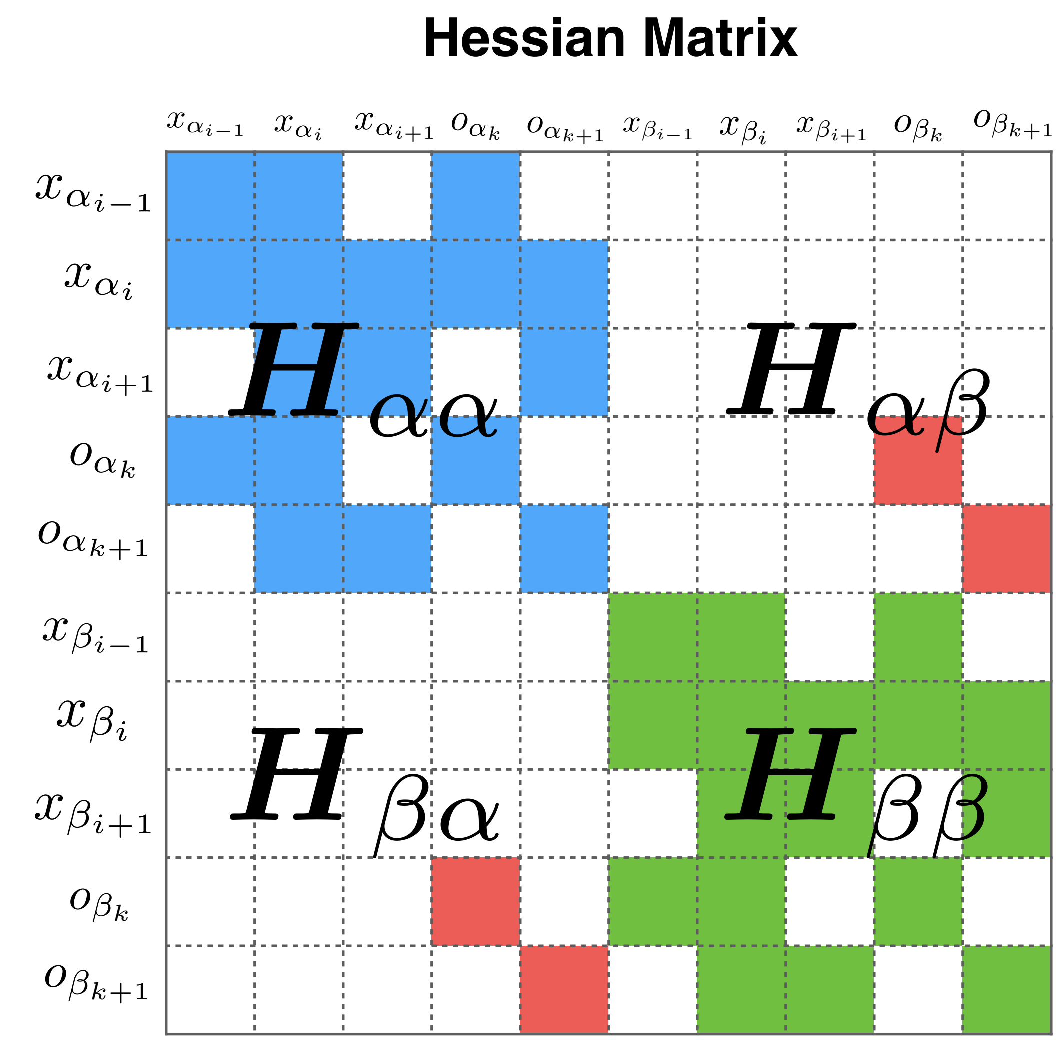
Assuming

$$y_\alpha = \{x_{\alpha_{i-1}}, x_{\alpha_i}, x_{\alpha_{i+1}}, o_{\alpha_k}, o_{\alpha_{k+1}}\}$$

$$y_\beta = \{x_{\beta_{i-1}}, x_{\beta_i}, x_{\beta_{i+1}}, o_{\beta_k}, o_{\beta_{k+1}}\}$$

Distributed Object-based SLAM





$$\boldsymbol{y}_\alpha^{k+1} = \boldsymbol{H}_{\alpha\alpha}^{-1}\left(\boxed{-\boldsymbol{H}_{\alpha\beta}}\,\boldsymbol{y}_\beta^k + \boldsymbol{g}_\alpha\right)$$

Iterate

$$\boldsymbol{y}_\beta^{k+1} = \boldsymbol{H}_{\beta\beta}^{-1}\left(\boxed{-\boldsymbol{H}_{\beta\alpha}}\,\boldsymbol{y}_\alpha^k + \boldsymbol{g}_\beta\right)$$

Object Dataset

BigBIRD Dataset

BigBIRD: A Large-Scale 3D Database of Object Instances
Singh et al. (ICRA 2014)

Object Dataset

BigBIRD Dataset

- Contains high resolution images for training the object detector.

BigBIRD: A Large-Scale 3D Database of Object Instances
Singh et al. (ICRA 2014)

Object Dataset

BigBIRD Dataset

- Contains high resolution images for training the object detector.
- Contains 3D textured object models for object pose estimation.

BigBIRD: A Large-Scale 3D Database of Object Instances
Singh et al. (ICRA 2014)

Object Dataset

BigBIRD Dataset

360 turntable video

Object Dataset

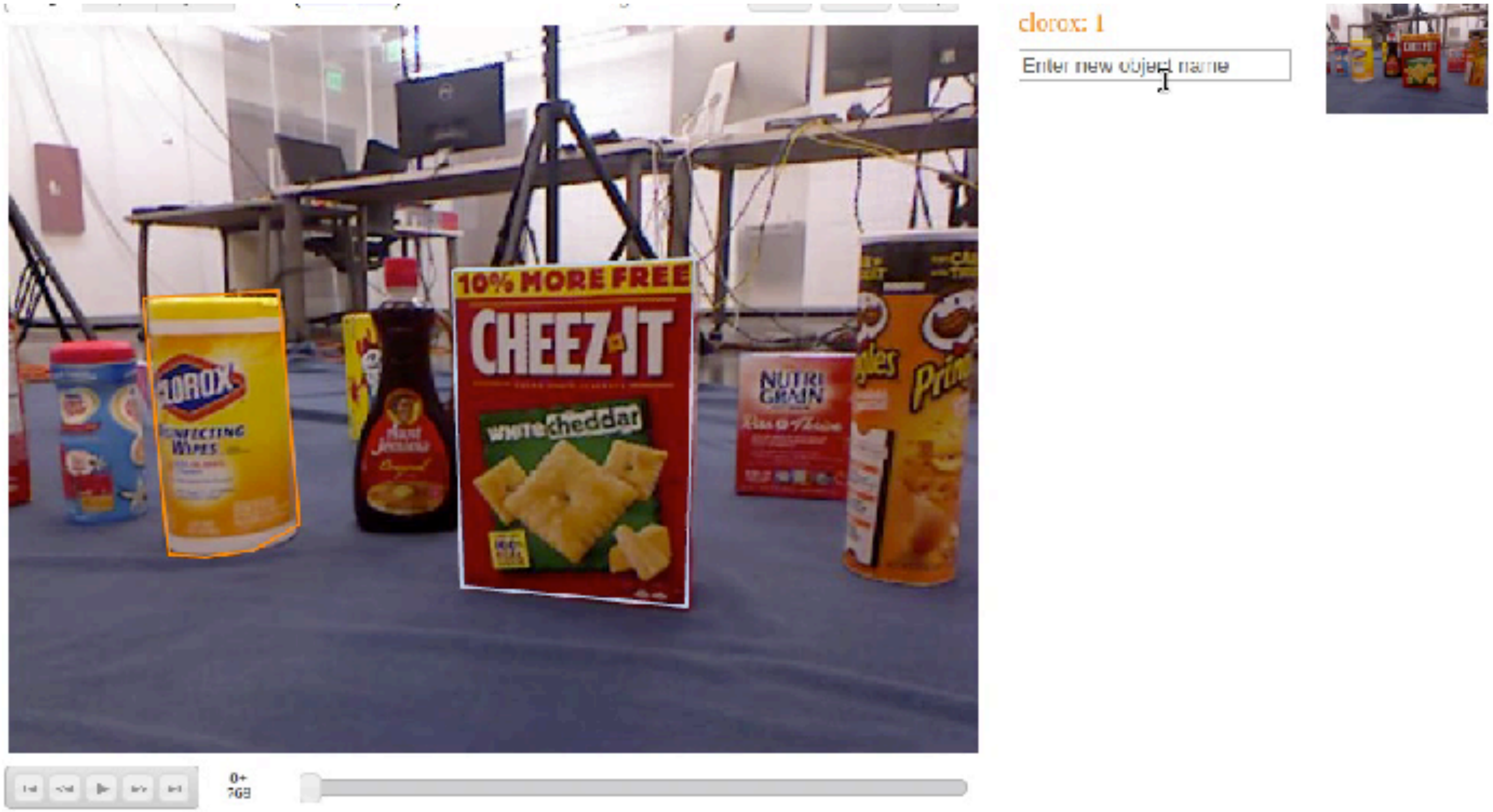
BigBIRD Dataset

360 turntable video

Object Dataset

BigBIRD Dataset

Motion capture data annotated using Sun3D annotator



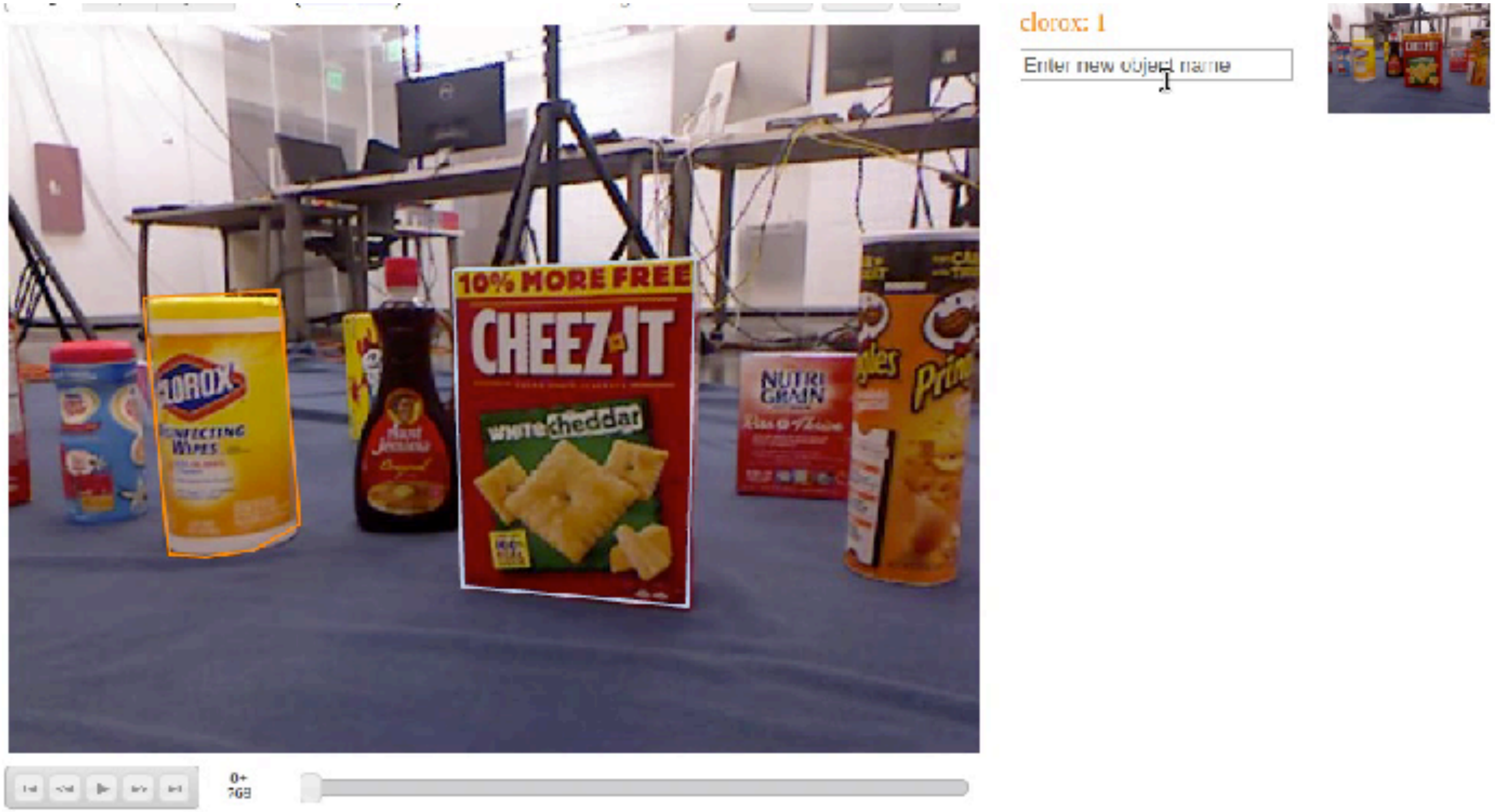
Pose estimates from Motion Capture data is used by Sun3D annotator to automatically propagate annotations.

SUN3D: A Database of Big Spaces Reconstructed using SfM and Object Labels
Xiao et al. (ICCV 2013)

Object Dataset

BigBIRD Dataset

Motion capture data annotated using Sun3D annotator



Pose estimates from Motion Capture data is used by Sun3D annotator to automatically propagate annotations.

SUN3D: A Database of Big Spaces Reconstructed using SfM and Object Labels
Xiao et al. (ICCV 2013)

Object Dataset

BigBIRD Dataset

SLAM data annotated using Sun3D annotator

Pose estimates from SLAM data is used by Sun3D annotator to automatically propagate annotations.

SUN3D: A Database of Big Spaces Reconstructed using SfM and Object Labels
Xiao et al. (ICCV 2013)

Object Dataset

BigBIRD Dataset

SLAM data annotated using Sun3D annotator

Pose estimates from SLAM data is used by Sun3D annotator to automatically propagate annotations.

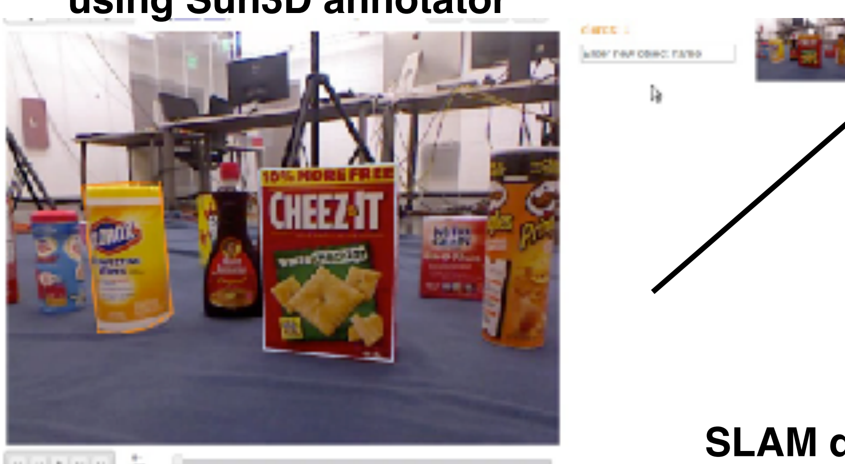
SUN3D: A Database of Big Spaces Reconstructed using SfM and Object Labels
Xiao et al. (ICCV 2013)

Object Detection

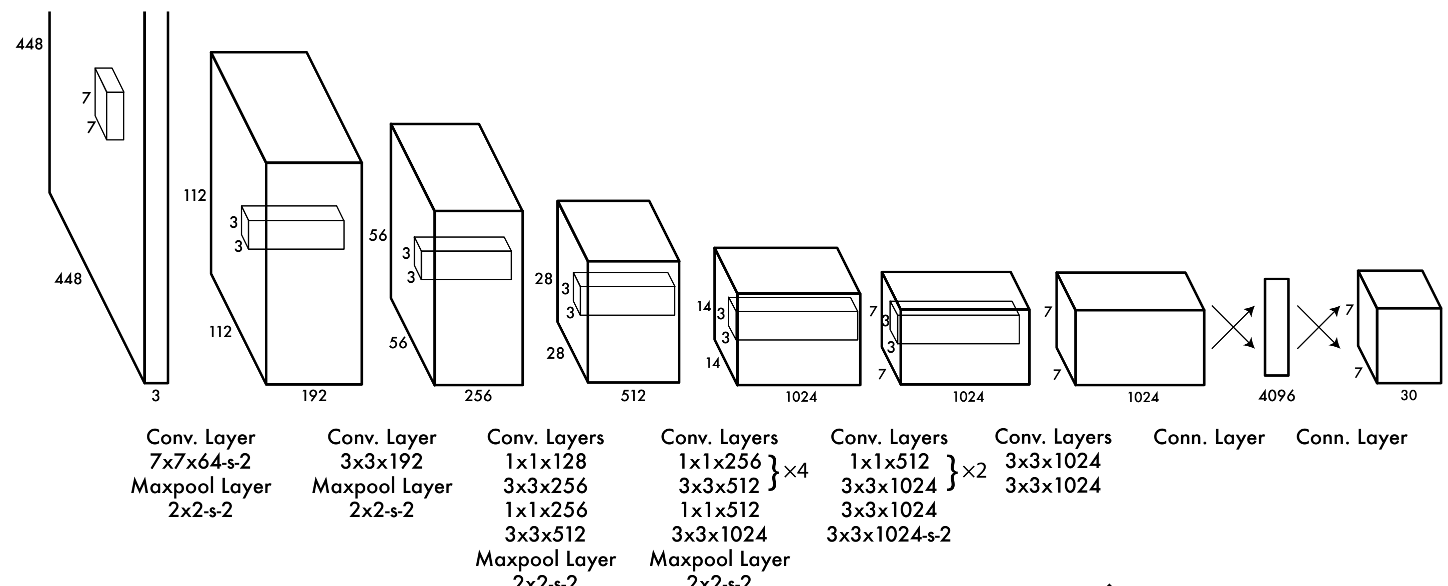
BigBIRD Dataset

360 turntable video

Motion capture data annotated using Sun3D annotator



YOLO Detector



SLAM data annotated using Sun3D annotator

Trained Network

Detects objects at 45fps

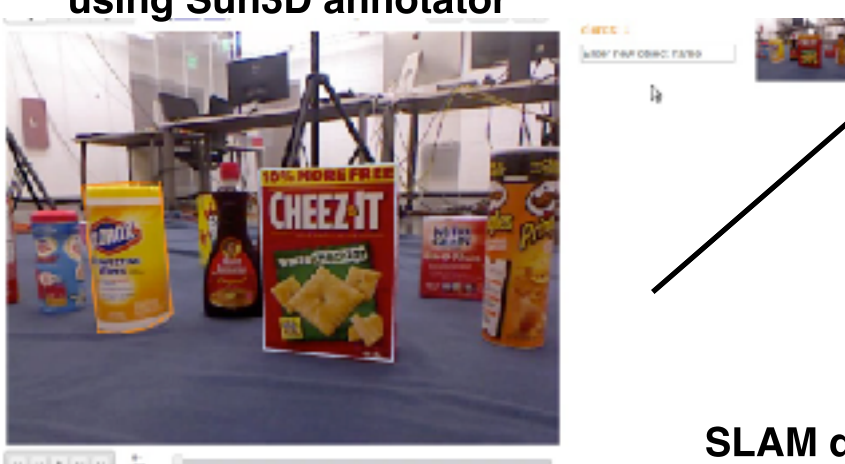
You Only Look Once: Unified, Real-Time Object Detection
Redmon et al. (CVPR 2016)

Object Detection

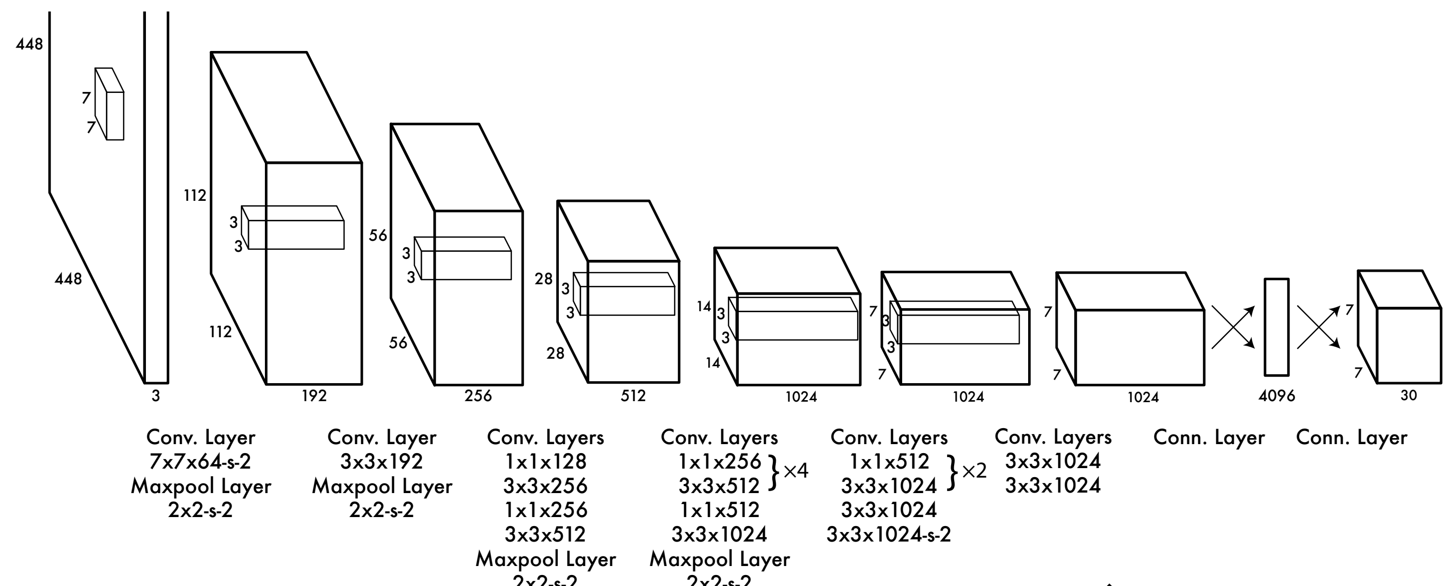
BigBIRD Dataset

360 turntable video

Motion capture data annotated using Sun3D annotator



YOLO Detector



SLAM data annotated using Sun3D annotator

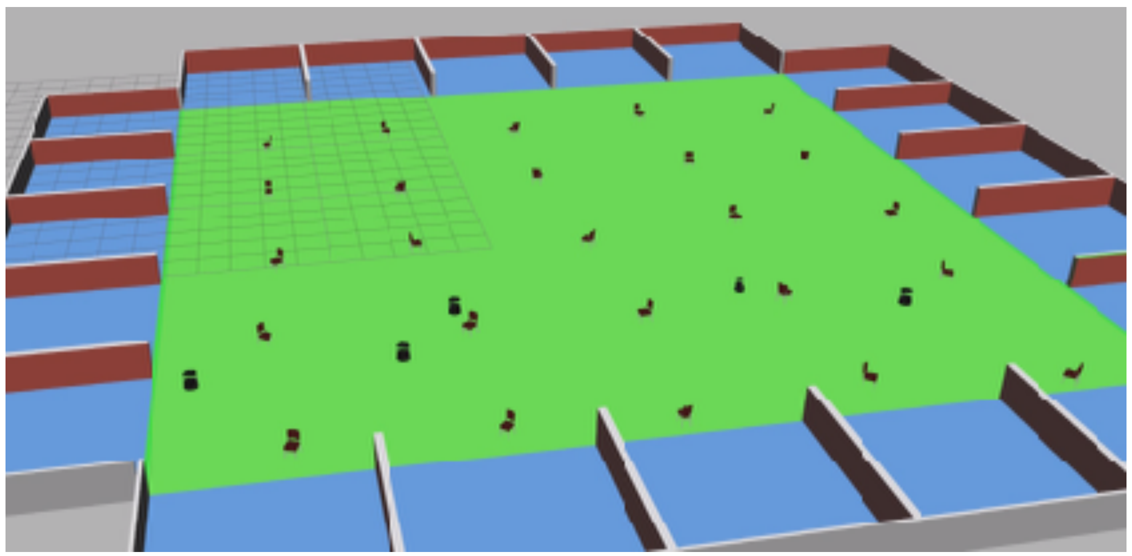
Trained Network

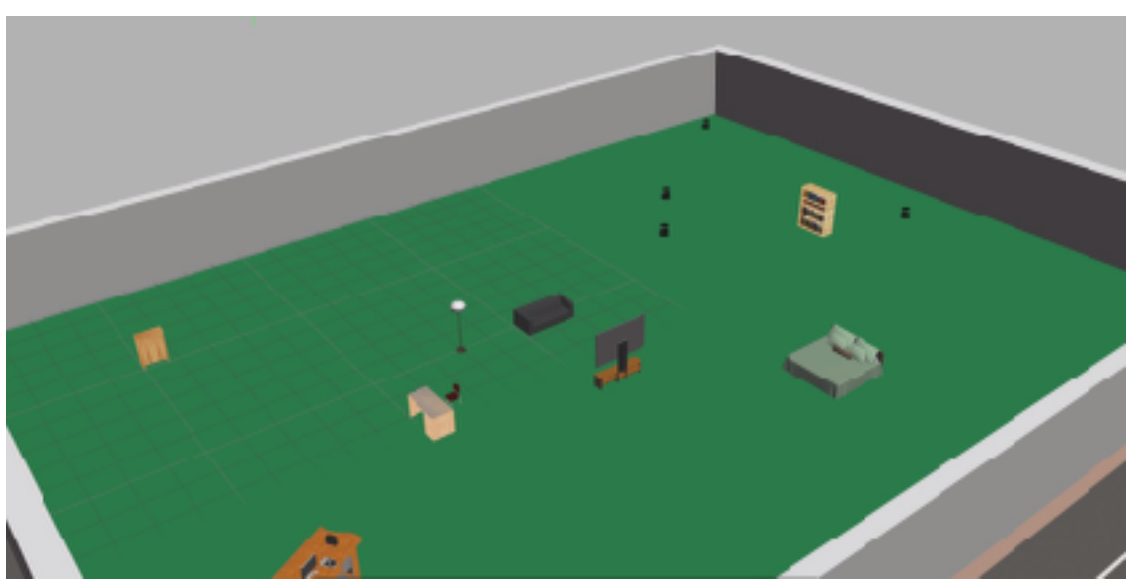
Detects objects at 45fps

You Only Look Once: Unified, Real-Time Object Detection
Redmon et al. (CVPR 2016)

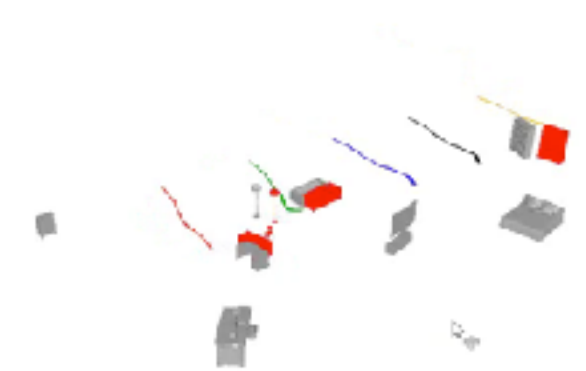
Simulation Experiments

Tested the approach in simulation on two scenarios over 10 Monte Carlo runs.





Distributed estimate converges to the centralized estimate

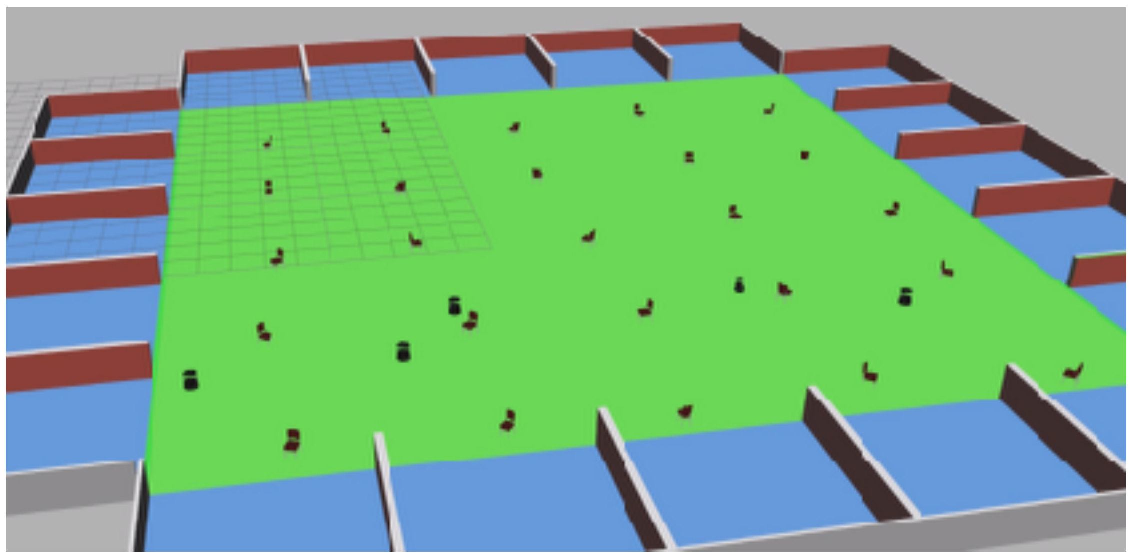


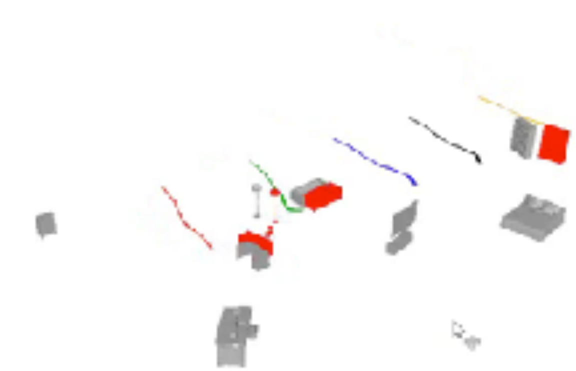
Gray = Centralized Estimate

Red = Distributed Estimate

Simulation Experiments

Tested the approach in simulation on two scenarios over 10 Monte Carlo runs.



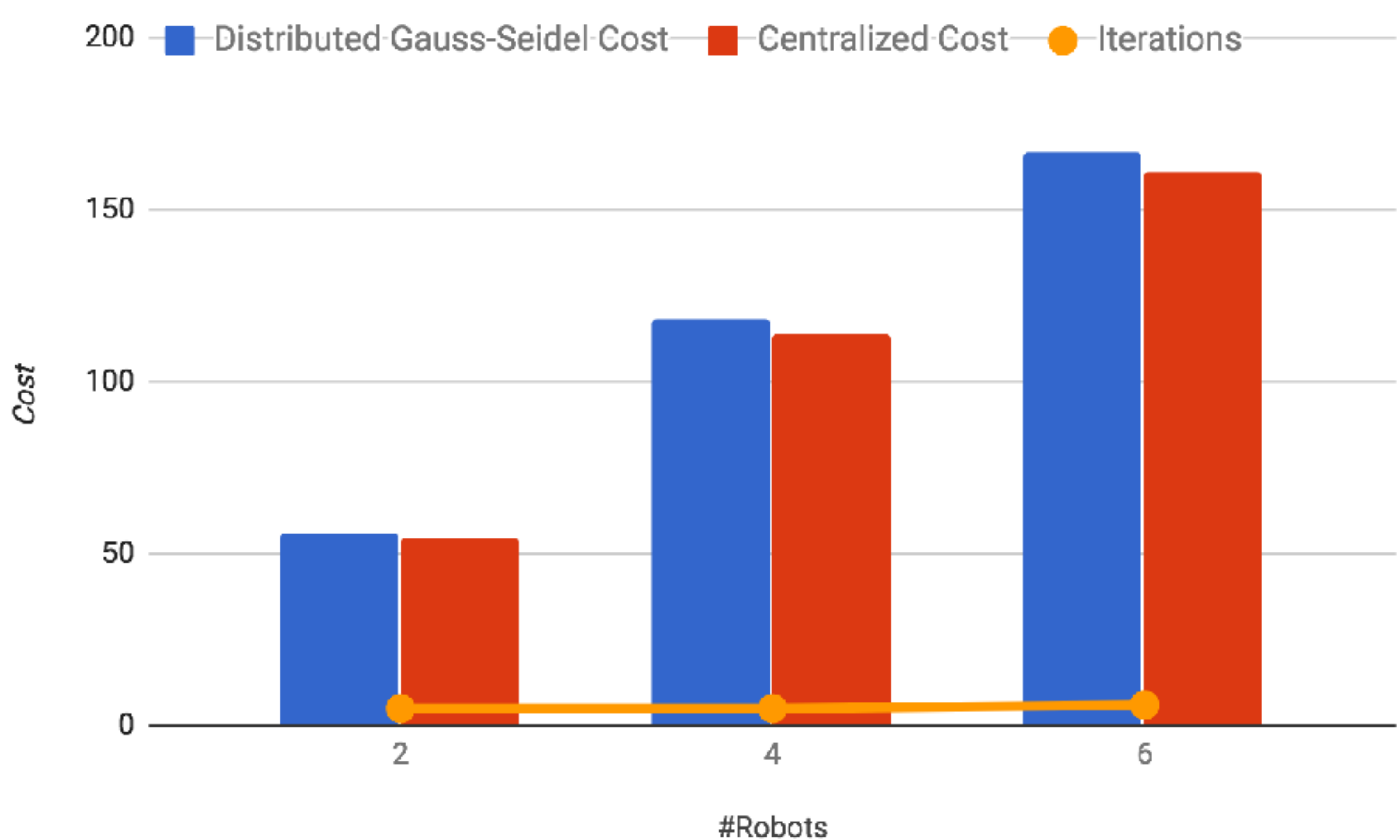


Distributed estimate converges to the centralized estimate

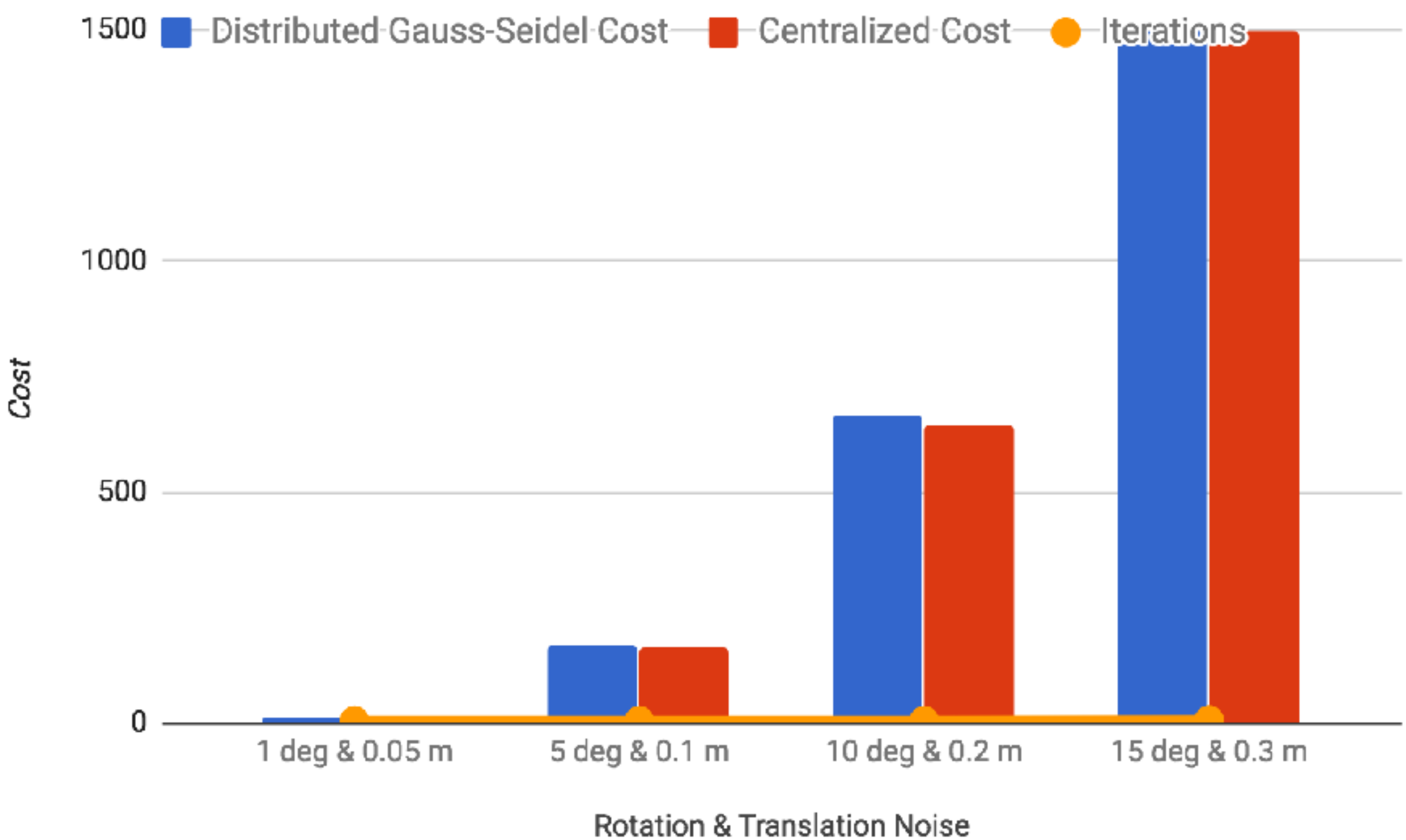
Gray = Centralized Estimate

Red = Distributed Estimate

Scalable in the number of robots

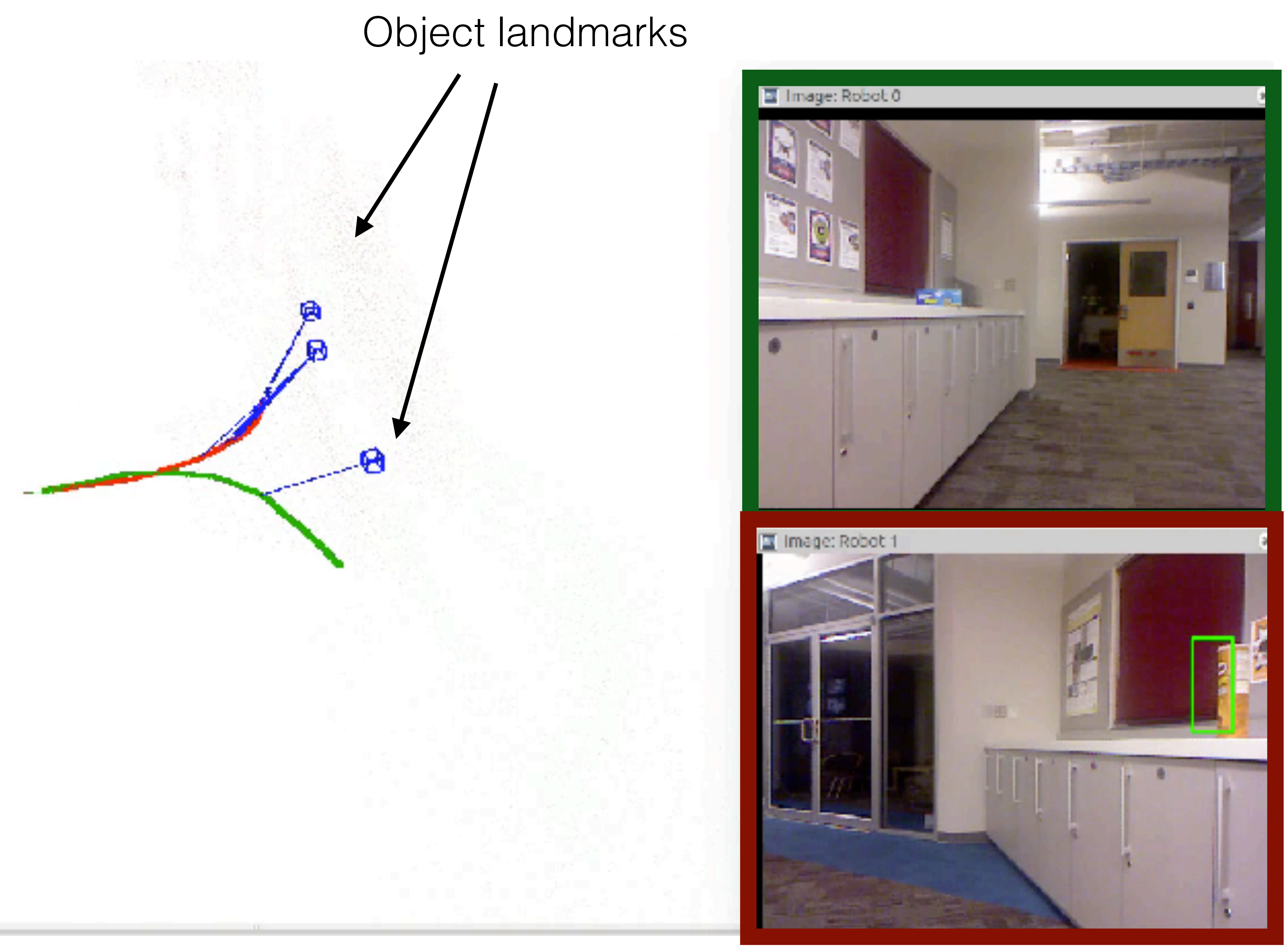


Resilient to Noise



Field Experiments

Object detection and SLAM



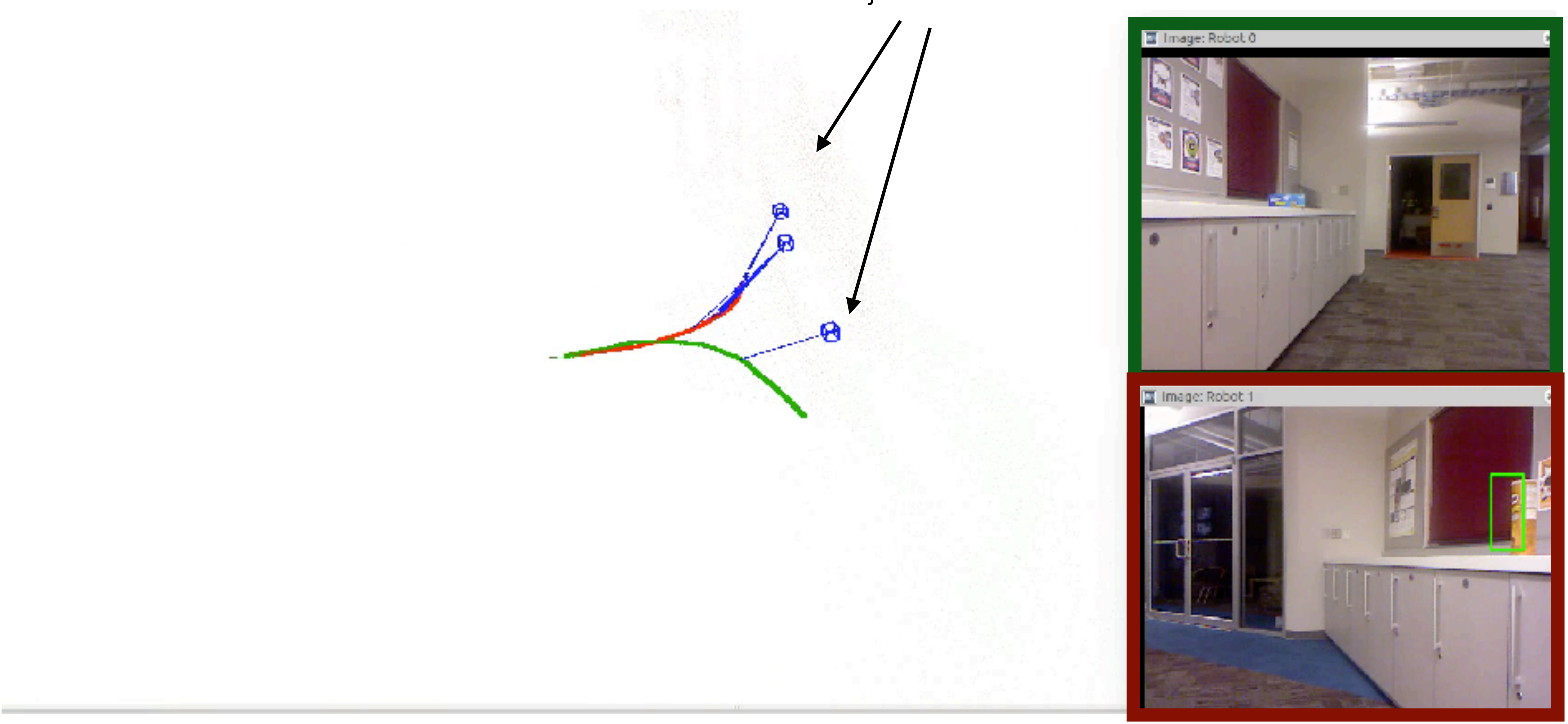
Detected objects are added as landmarks in the map

Green = Robot 0 **Red = Robot 1**

Field Experiments

Object detection and SLAM

Object landmarks

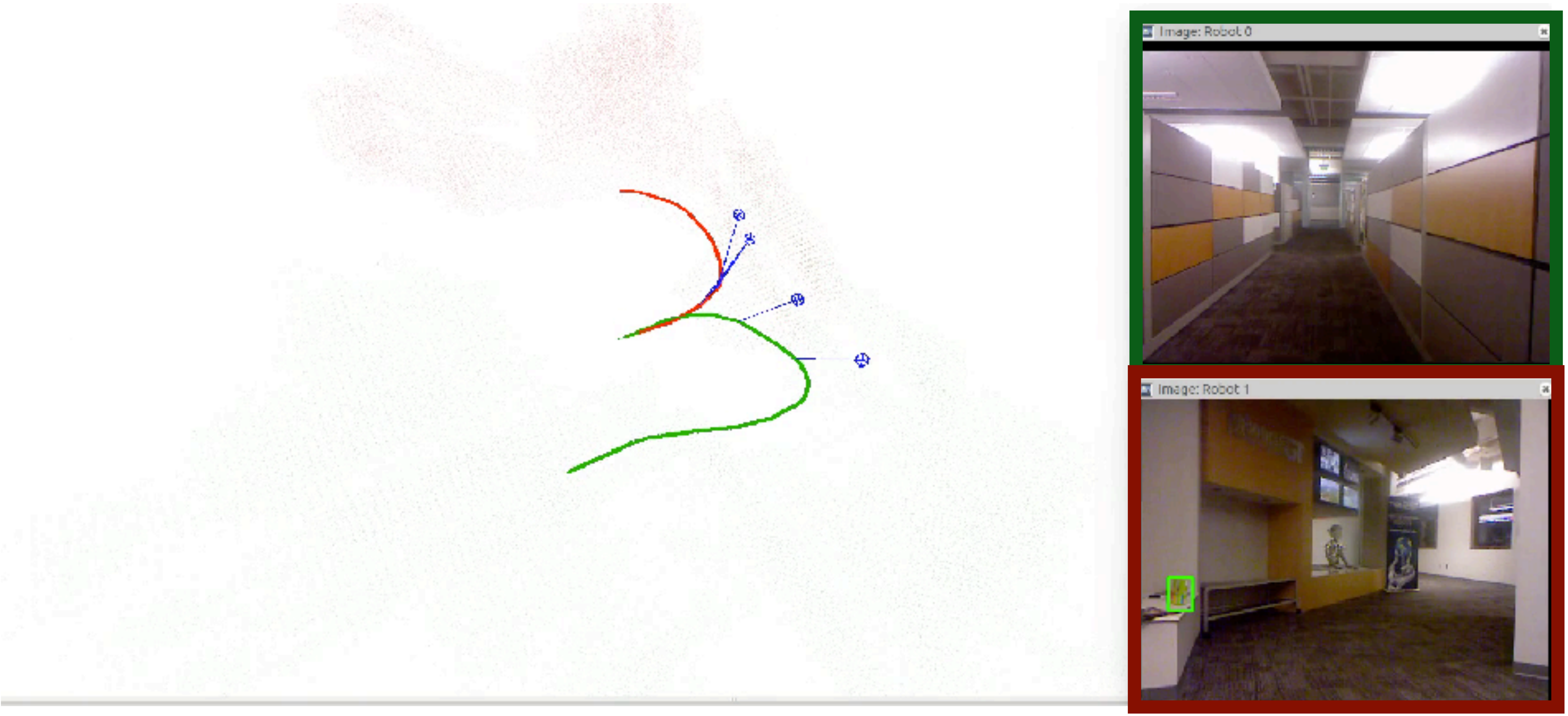


Detected objects are added as landmarks in the map

Green = Robot 0 **Red = Robot 1**

Field Experiments

Inter-Robot Loop Closure

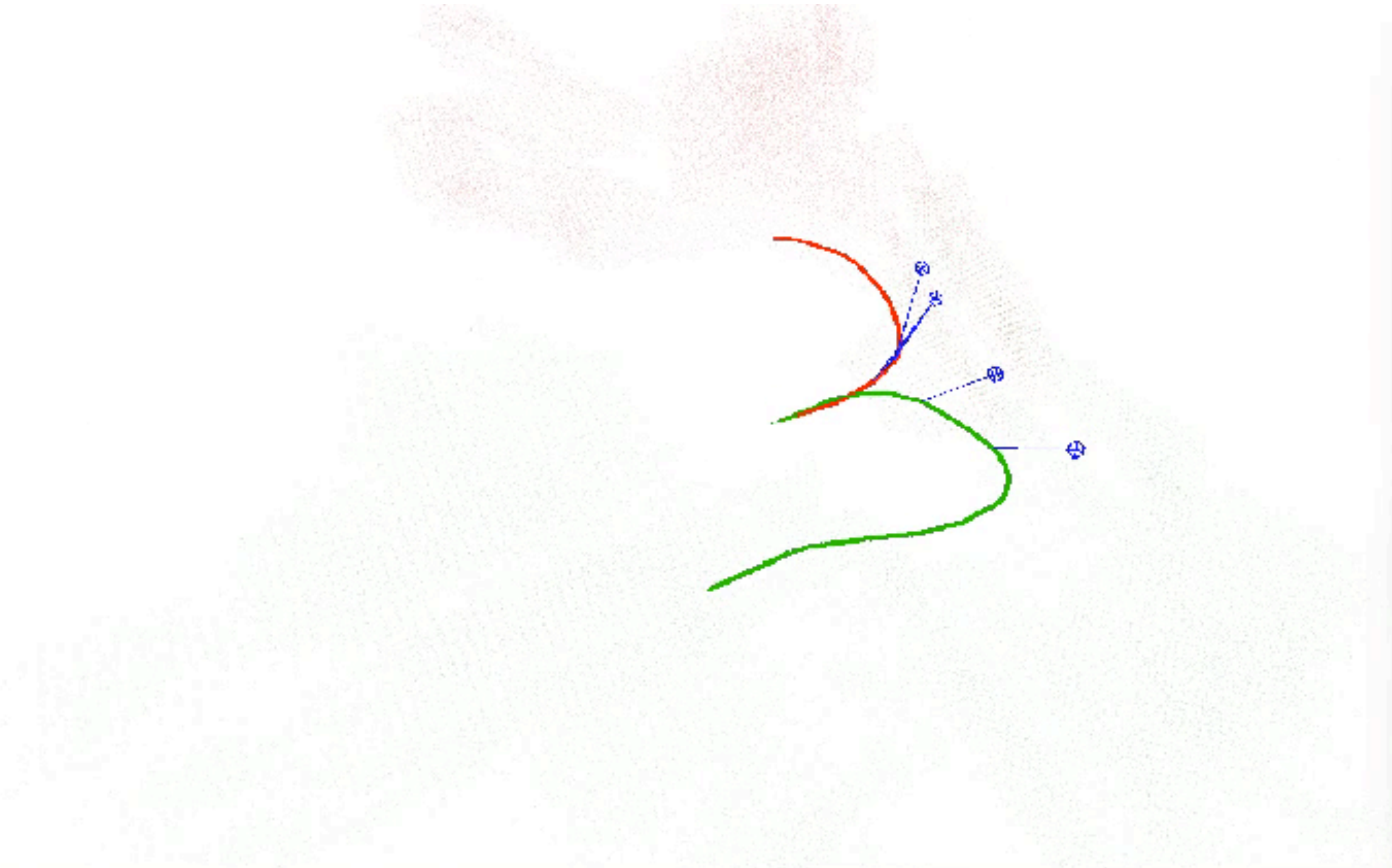


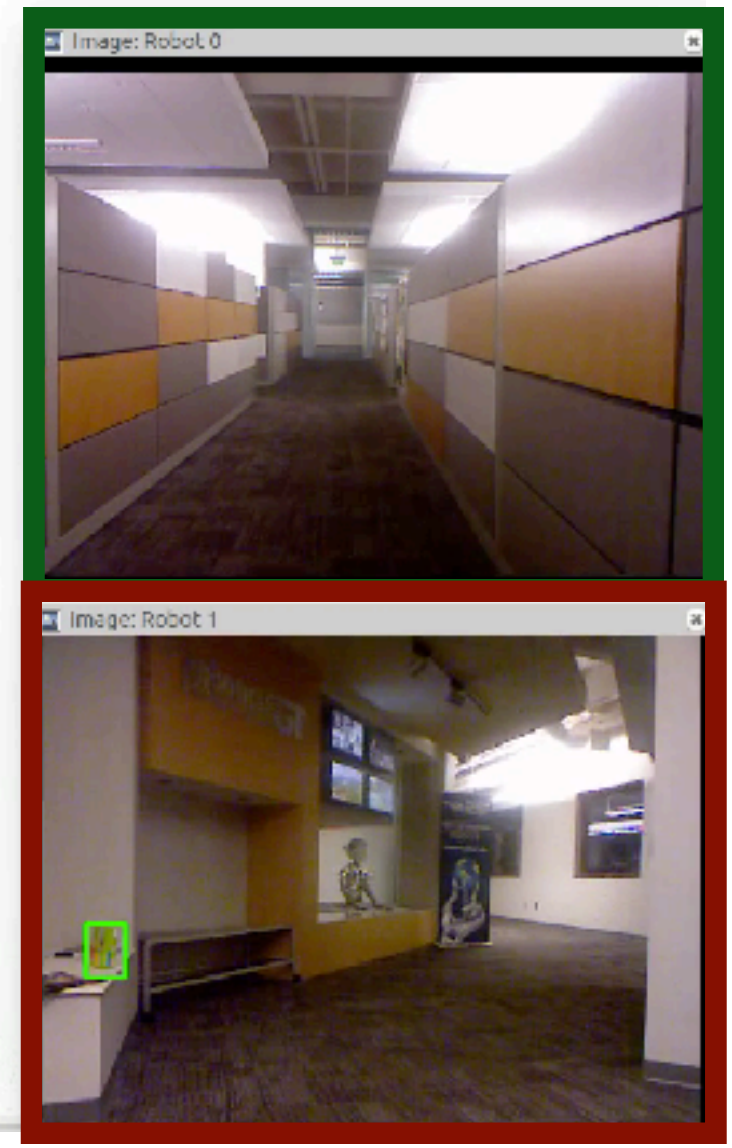
Loop closure factor is added when two robots see the same object

Green = Robot 0 **Red = Robot 1**

Field Experiments

Inter-Robot Loop Closure



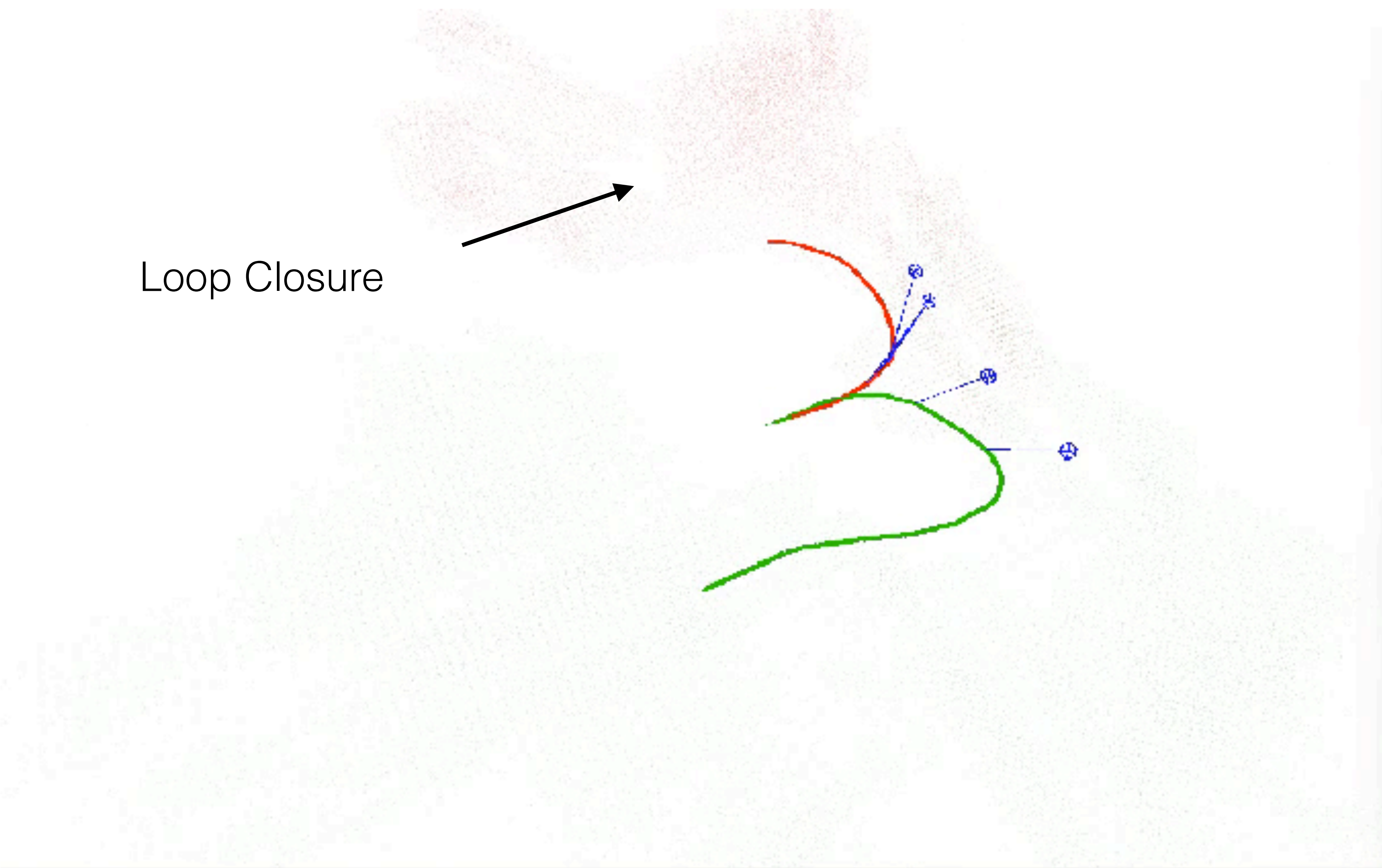


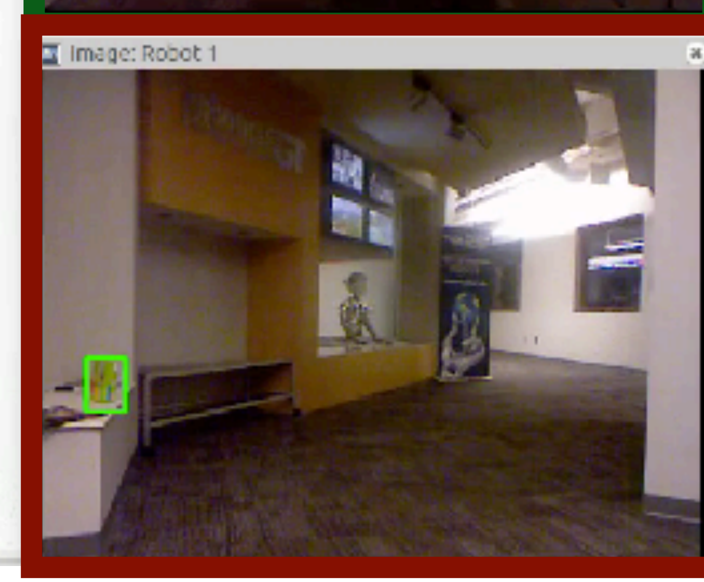
Loop closure factor is added when two robots see the same object

Green = Robot 0 **Red = Robot 1**

Field Experiments

Inter-Robot Loop Closure



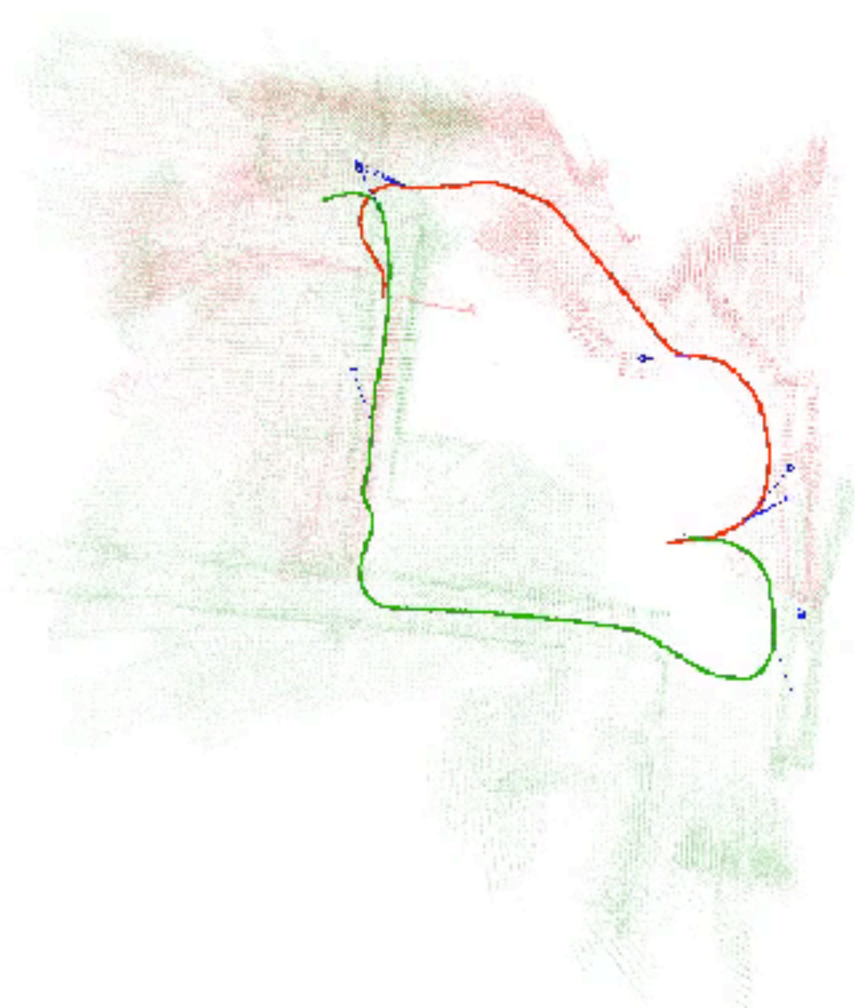


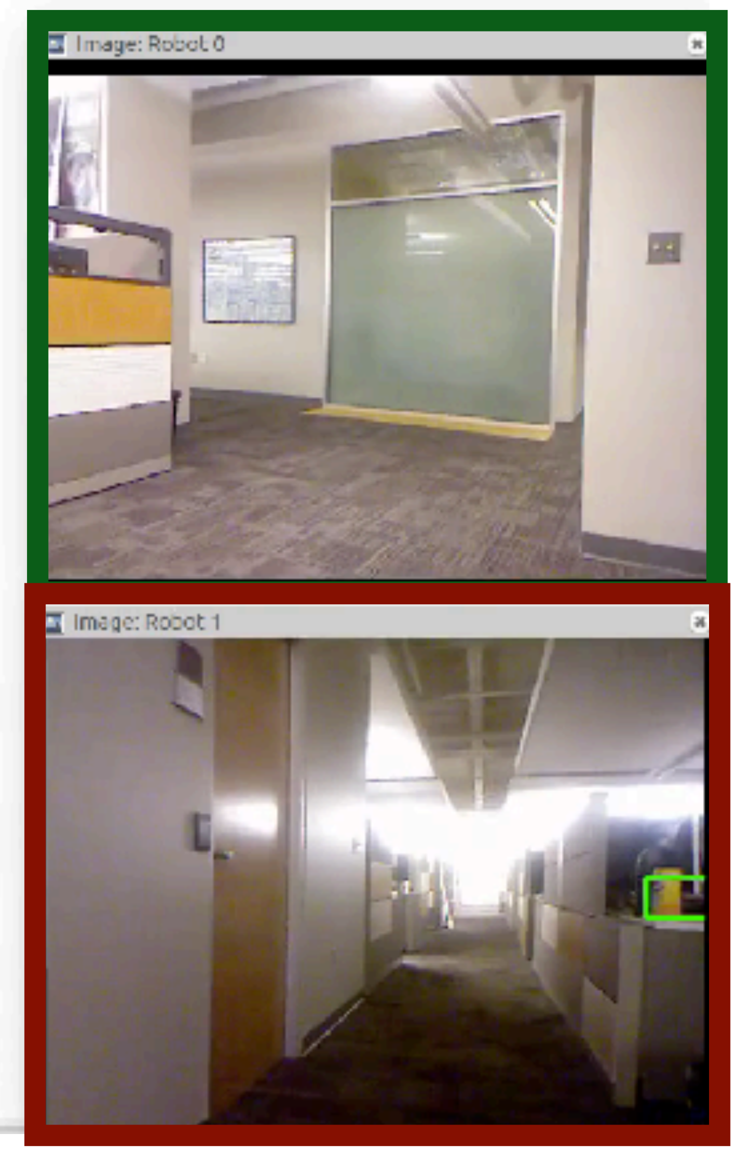
Loop closure factor is added when two robots see the same object

Green = Robot 0 **Red = Robot 1**

Field Experiments

Experimented with 18 objects in a large scale environment.

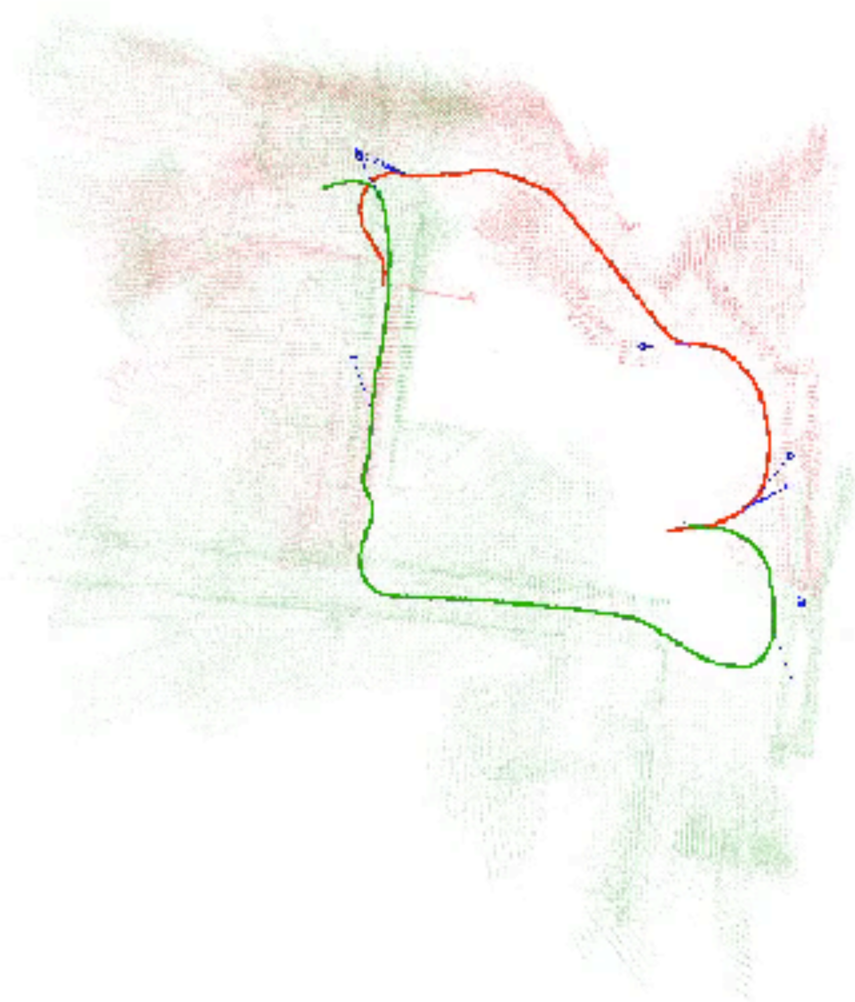




Green = Robot 0 **Red = Robot 1**

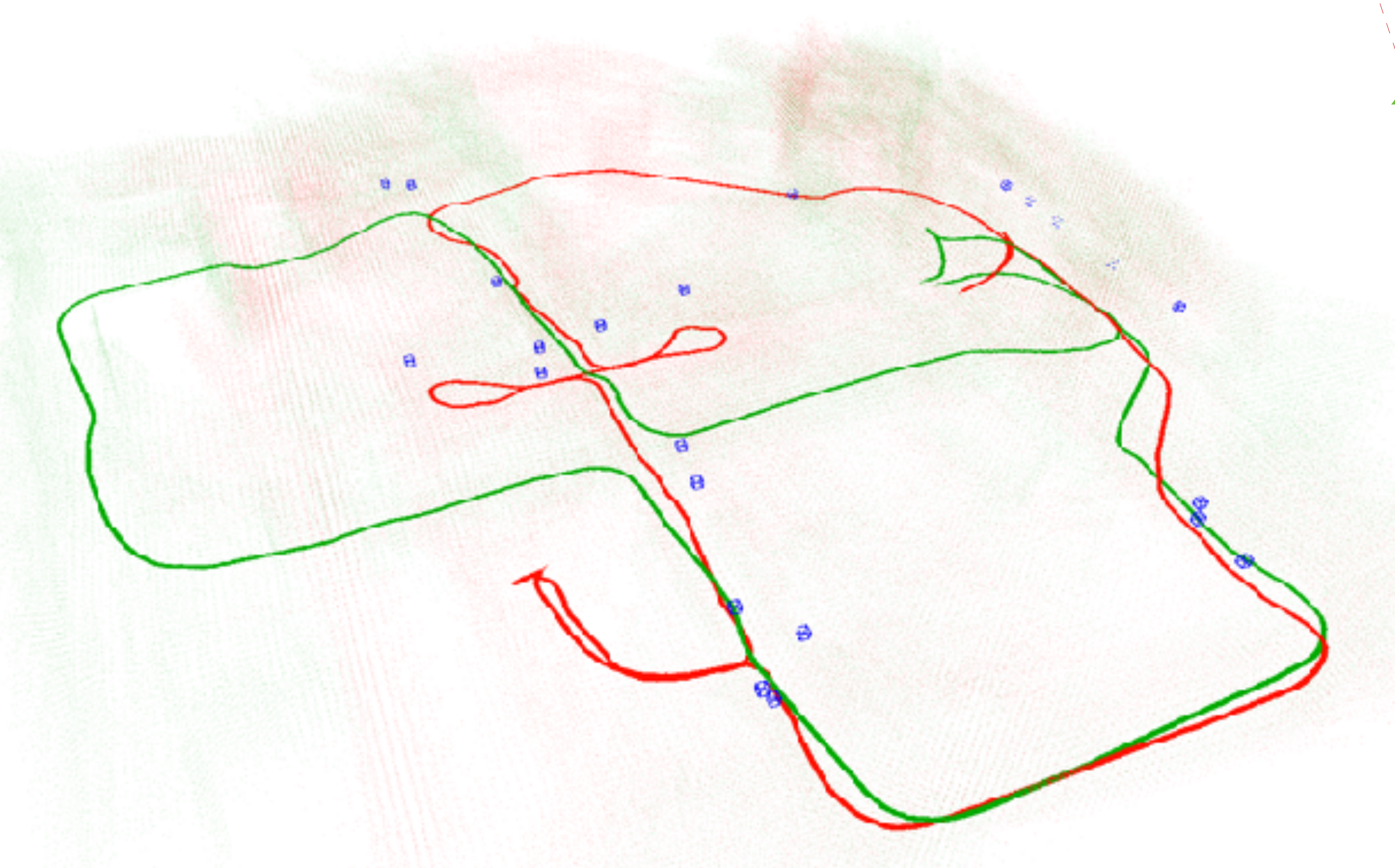
Field Experiments

Experimented with 18 objects in a large scale environment.

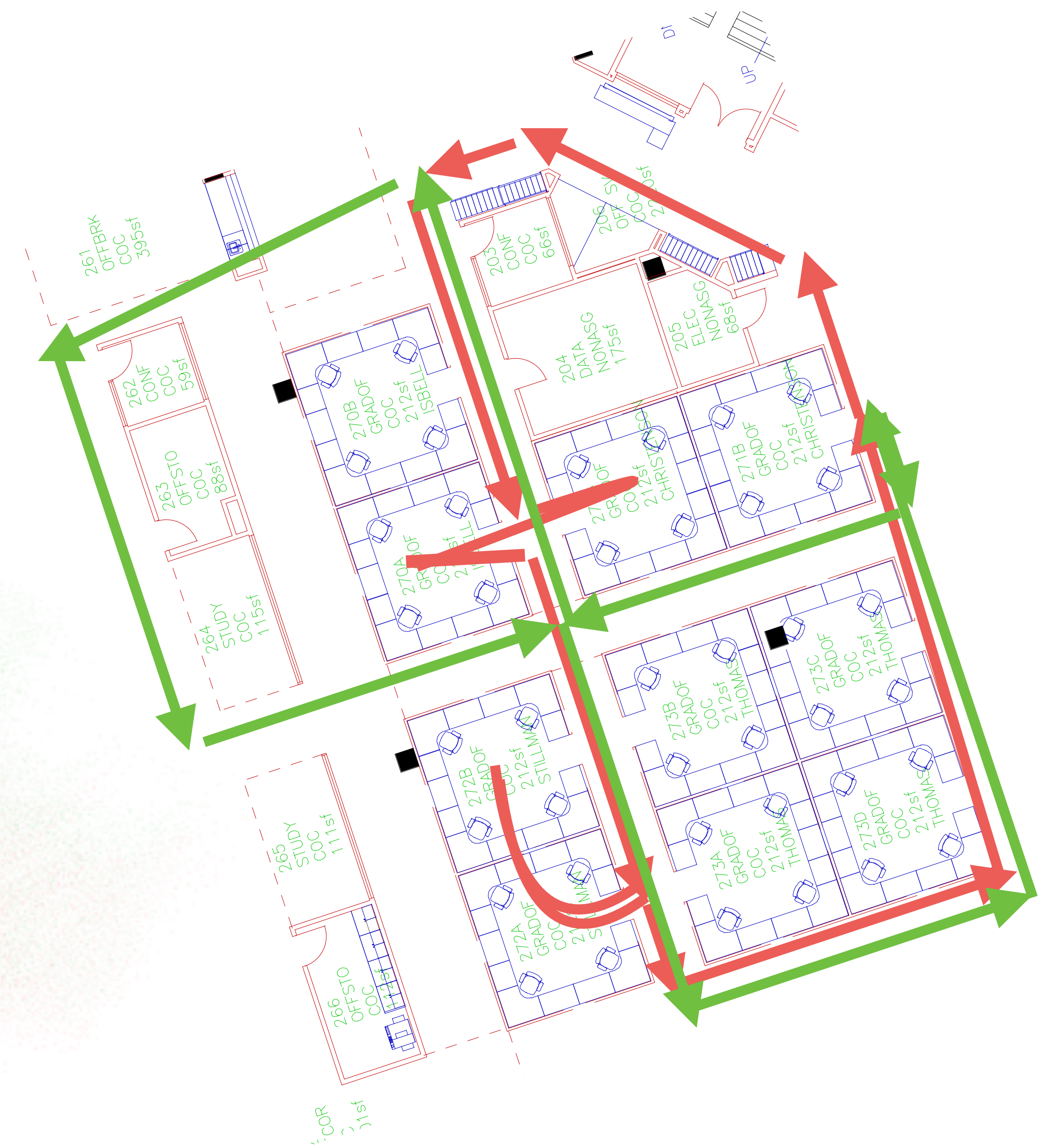


Green = Robot 0 **Red = Robot 1**

Field Experiments

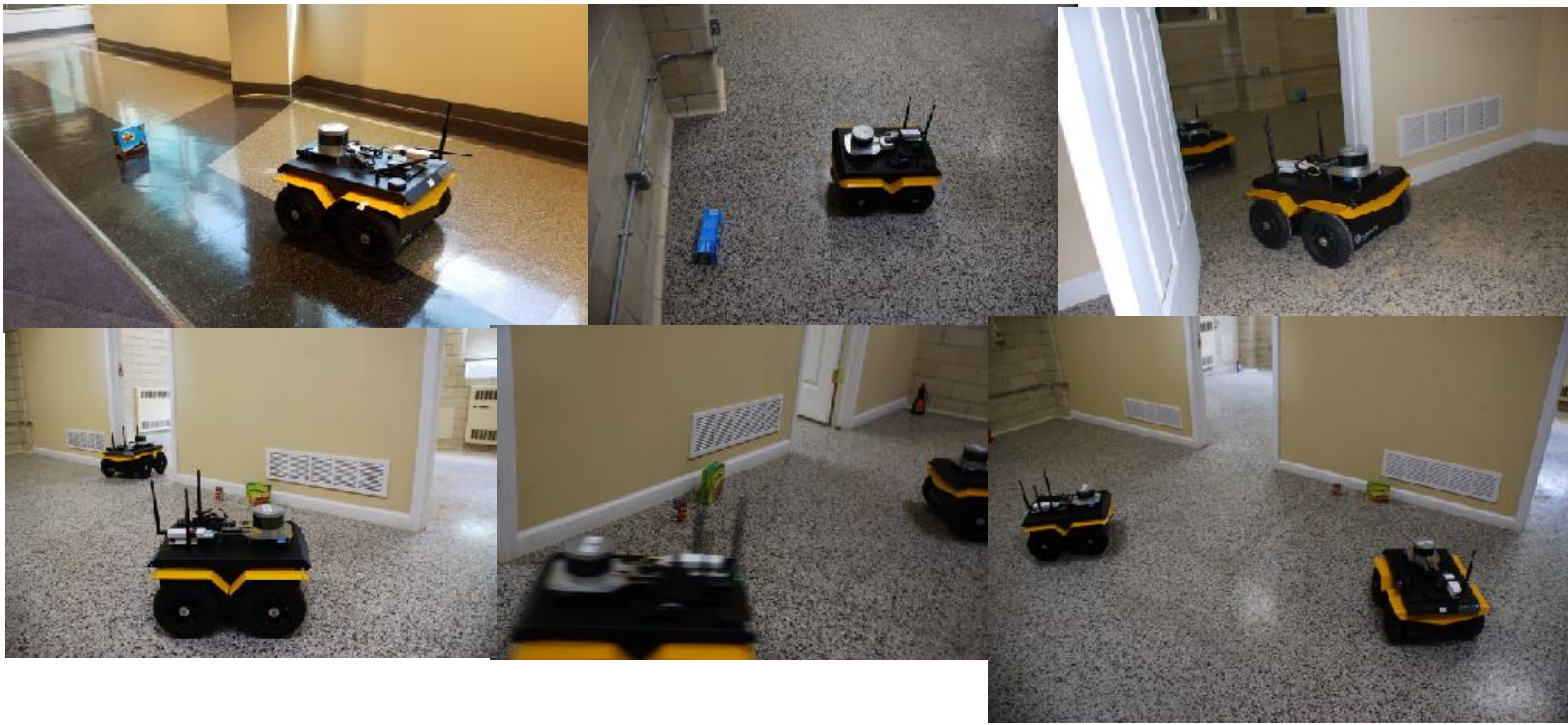


Estimated trajectory



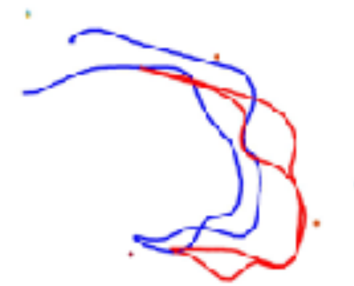
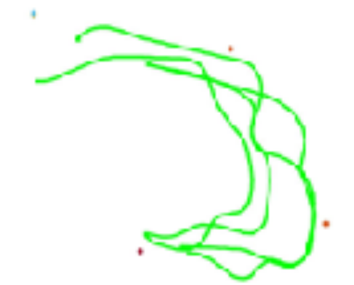
Approximate trajectory

Field Experiments



We tested the proposed approach on field data collected by two Jackal robots, moving in different indoor settings.

Accuracy

	Distributed	**Centralized**
Stadium-1		
Stadium-2		
House		

Distributed estimate converges to the centralized estimate

Memory Requirements

Scenario	Average Per Robot Memory Requirement	
	Point Cloud Map	Object level Map
Stadium-1	1.2 GB	≈ 1.9 MB
Stadium-2	1.4 GB	≈1.9 MB
House	2.1 GB	≈1.9 MB

+ Memory required to store object models

Memory Requirements

Scenario	Average Per Robot Memory Requirement	
	Point Cloud Map	Object level Map
Stadium-1	1.2 GB	≈ 1.9 MB
Stadium-2	1.4 GB	≈1.9 MB
House	2.1 GB	≈1.9 MB

\+ Memory required to store object models

Object level map requires three orders of magnitude less memory as compared to Point cloud map

Communication Bandwidth Requirements

Distributed Gauss-Seidel

Amount of communication required to perform distributed optimization given the communication link.

Communication Bandwidth =

+

The amount of communication required to establish that link.

Distributed Object-based SLAM

Communication Bandwidth Requirements

Scenario	Average Communication Bandwidth Requirement	
	Point Cloud Map	Object level Map
Stadium-1	19 MB	0.16 KB
Stadium-2	14 MB	0.12 KB
House	16 MB	0.22 KB

Communication Bandwidth is the amount of communication required to establish to the link.

Communication Bandwidth Requirements

Scenario	Average Communication Bandwidth Requirement	
	Point Cloud Map	Object level Map
Stadium-1	19 MB	0.16 KB
Stadium-2	14 MB	0.12 KB
House	16 MB	0.22 KB

Object level map requires four orders of magnitude less communication bandwidth as compared to Point cloud map

So far…

state of art distributed optimizer

+

*Using **objects** as landmarks*

↓ ***distributed SLAM***

1. ***reduce** the **communication bandwidth and the memory***

2. *improves the **robustness and scalability***

3. *outputs a **human understandable map***

So far…

state of art distributed optimizer

+

*Using **objects** as landmarks*

Distributed Object based SLAM

↓ ***distributed SLAM***

1. ***reduce** the **communication bandwidth and the memory***
2. *improves the **robustness and scalability***
3. *outputs a **human understandable map***

So far…

state of art distributed optimizer

+

*Using **objects** as landmarks*

Distributed Object based SLAM

↓ ***distributed SLAM***

1. ***reduce** the **communication bandwidth** and the **memory***
2. *improves the **robustness and scalability***
3. *outputs a **human understandable map***

Using objects reduces the communication bandwidth and the memory requirements

So far…

state of art distributed optimizer

+

$\boxed{\textit{Using } \textbf{\textit{objects}} \textit{ as landmarks}}$

Distributed Object based SLAM

↓ ***distributed SLAM***

1. ***reduce*** *the* ***communication bandwidth*** ***and the memory***

Using objects reduces the communication bandwidth and the memory requirements

2. *improves the* ***robustness and scalability***

- **Robust to bad initialization**
- **Resilient to measurement noise**
- **Scalable in the number of robots**

3. *outputs a* ***human understandable map***

So far…

state of art distributed optimizer

+

*Using **objects** as landmarks*

Distributed Object based SLAM

$\downarrow$ ***distributed SLAM***

1. ***reduce** the **communication bandwidth** and the **memory***

Using objects reduces the communication bandwidth and the memory requirements

2. *improves the* ***robustness and scalability***

- **Robust to bad initialization**
- **Resilient to measurement noise**
- **Scalable in the number of robots**

3. *outputs a* ***human understandable map***

Resulting object level map can be used for manipulation tasks like pick and place

So far…

state of art distributed optimizer — **Distributed Gauss-Seidel**

+

Using ***objects*** *as landmarks* — **Distributed Object based SLAM**

↓ ***distributed SLAM***

1. ***reduce*** *the* ***communication bandwidth and the memory***

   Using objects reduces the communication bandwidth and the memory requirements

2. *improves the* ***robustness and scalability***

   - **Robust to bad initialization**
   - **Resilient to measurement noise**
   - **Scalable in the number of robots**

3. *outputs a* ***human understandable map***

   Resulting object level map can be used for manipulation tasks like pick and place

Remaining Issues…

state of art distributed optimizer

+

$\boxed{\text{Using \textbf{objects} as landmarks}}$

Won’t generalize to unseen object instances

↓ ***distributed SLAM***

1. ***reduce** the **communication bandwidth and the memory***
2. *improves the **robustness and scalability***
3. *outputs a **human understandable map***

Outline

1. Distributed Gauss-Seidel Approach
(*state of art distributed optimizer*)

2. Distributed Object-based SLAM with Known Object Models
(*using objects as landmarks*)

3. Distributed Object-based SLAM with Joint Object Modeling and Mapping
(*jointly model objects along with SLAM*)

4. Conclusions and Future Work

Outline

1. Distributed Gauss-Seidel Approach
(*state of art distributed optimizer*)

2. Distributed Object-based SLAM with Known Object Models
(*using objects as landmarks*)

3. Distributed Object-based SLAM with Joint Object Modeling and Mapping
(*jointly model objects along with SLAM*)

4. Conclusions and Future Work

Outline

1. Distributed Gauss-Seidel Approach
(*state of art distributed optimizer*)

2. Distributed Object-based SLAM with Known Object Models
(*using objects as landmarks*)

3. Distributed Object-based SLAM with Joint Object Modeling and Mapping
(*jointly model objects along with SLAM*)

4. Conclusions and Future Work

Motivation

- Can be **challenging to store a model** of all the object instances due to large intra-class variation.

Google 3D warehouse

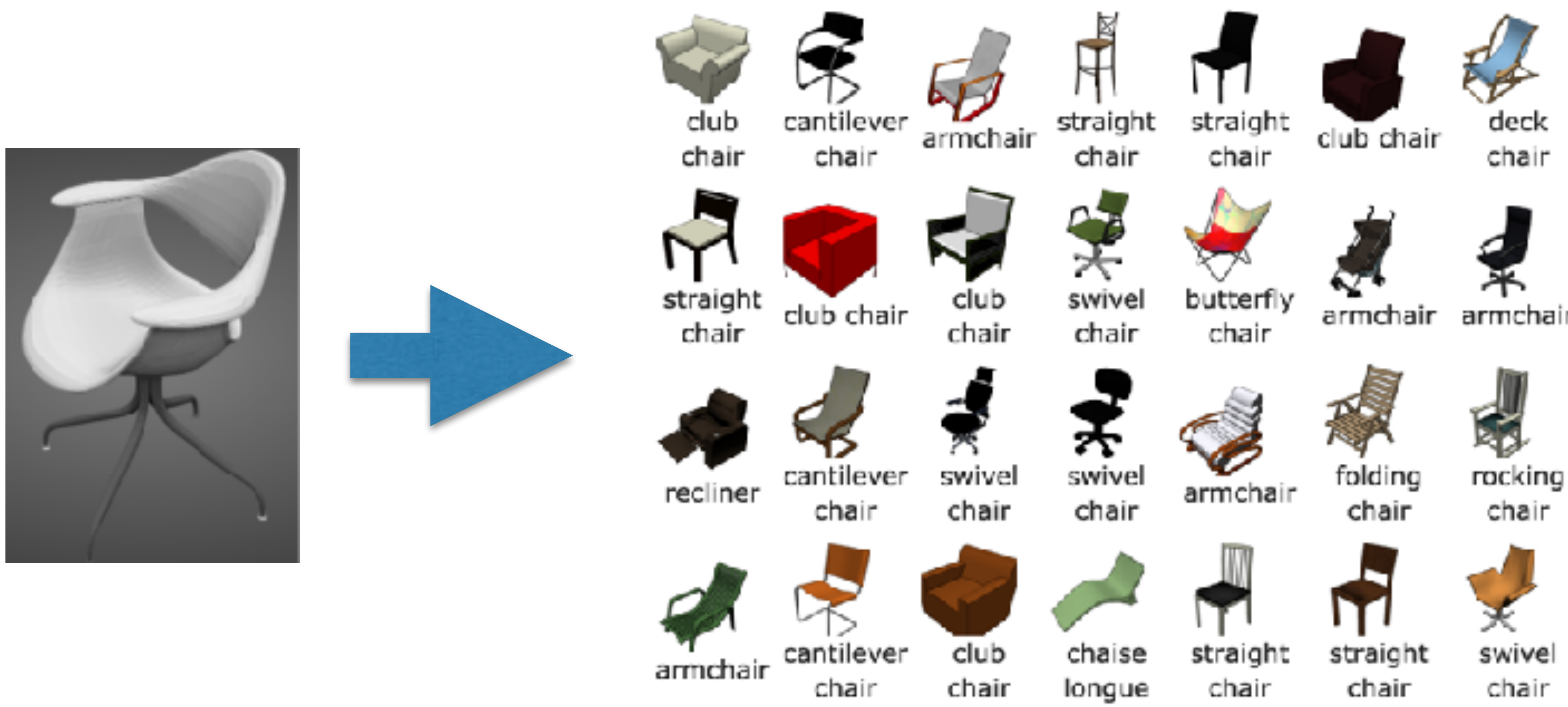
(about 2.5 million models)

c.f. Changhyun Choi et al.

Motivation

- Can be **challenging to store a model** of all the object instances due to large intra-class variation.

- **Searching** through all the object models for object pose estimation can be **computationally demanding**.

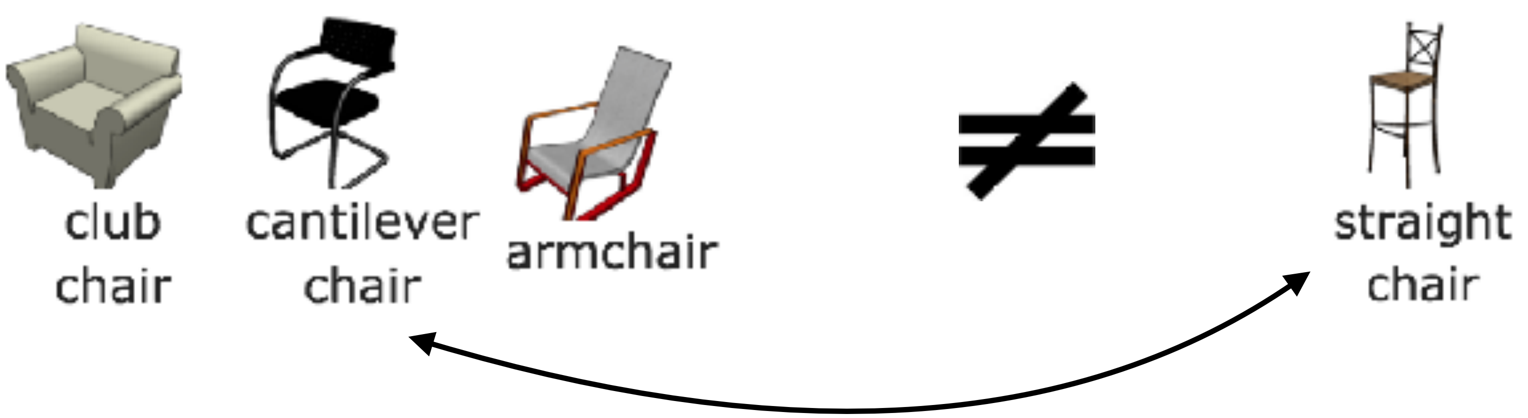


Motivation

- Can be **challenging to store a model** of all the object instances due to large intra-class variation.

- **Searching** through all the object models for object pose estimation can be **computationally demanding**.

- It **won't generalize to unseen instances** of the same object category as well.



object pose estimation will fail

Distributed Object-based SLAM with Joint Object Modeling and Mapping

Propose to extend the previous work to the case where **object models are previously unknown** and are **modeled jointly** with Distributed Object based SLAM.

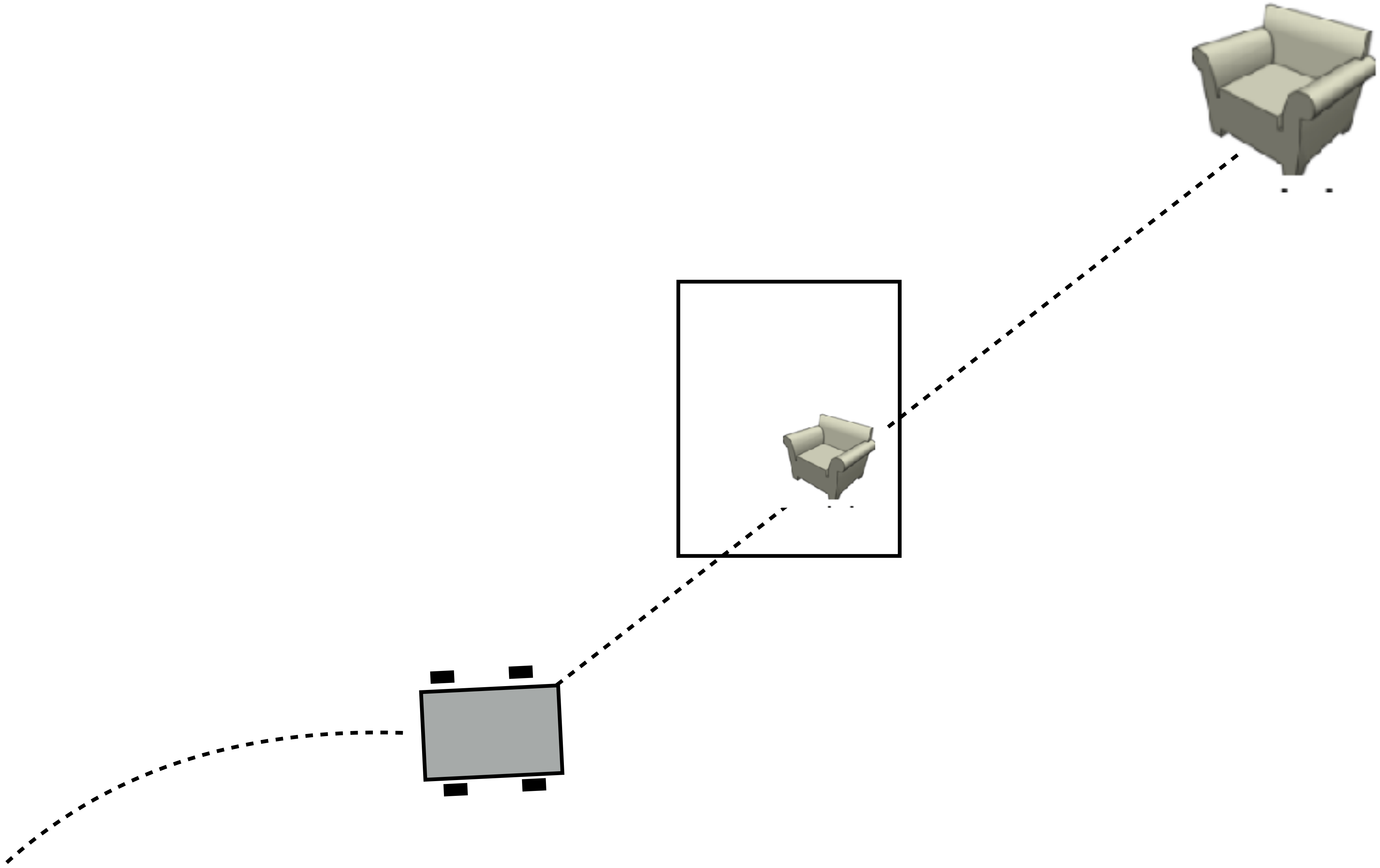
SLAM with Object Discovery, Modeling and Mapping
Choudhary et al. (IROS 2014)

Distributed Object-based SLAM with Joint Object Modeling and Mapping

Propose to extend the previous work to the case where **object models are previously unknown** and are **modeled jointly** with Distributed Object based SLAM.

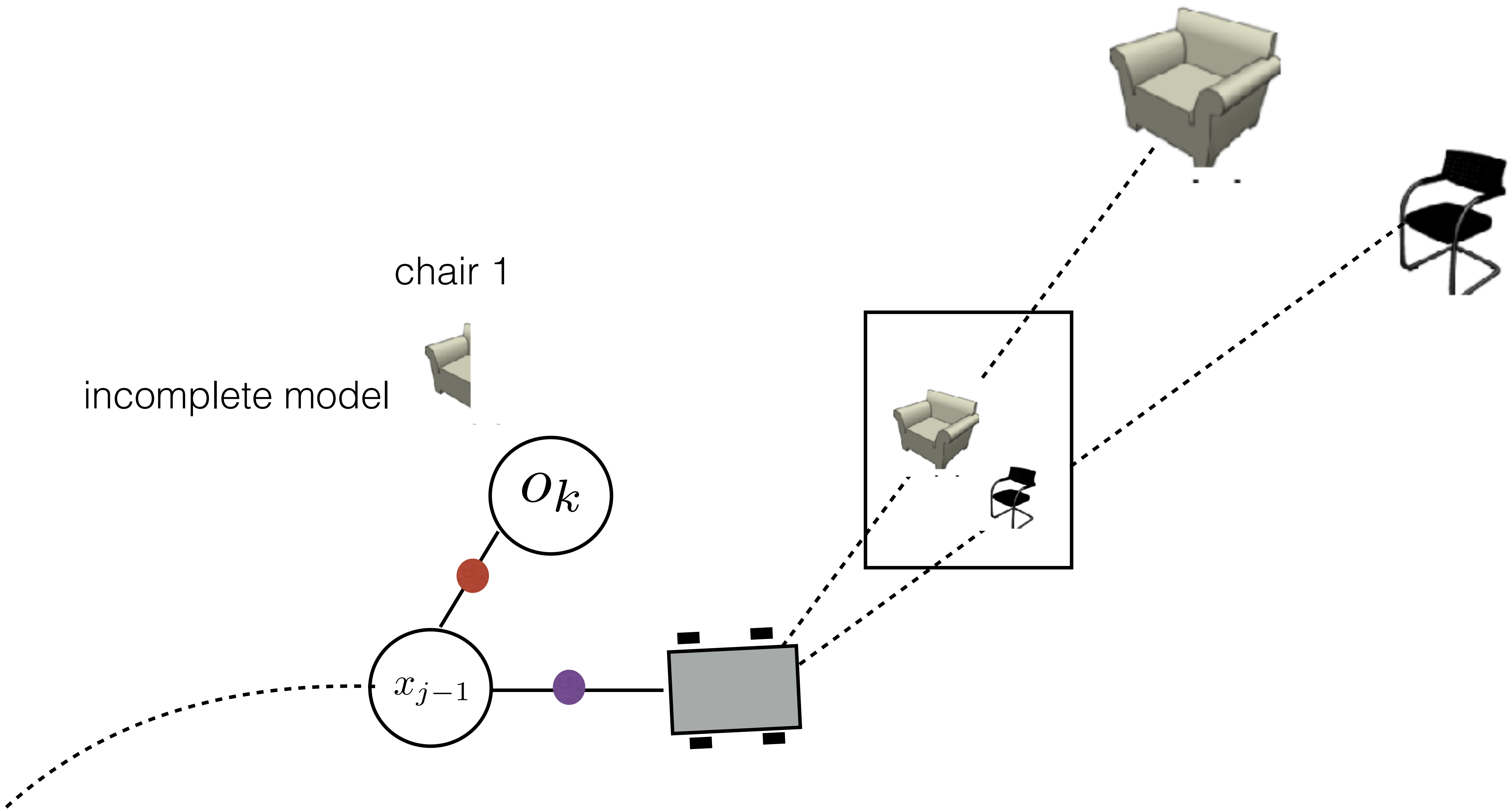
SLAM with Object Discovery, Modeling and Mapping
Choudhary et al. (IROS 2014)

Object-based SLAM
with Joint Object Modeling and Mapping

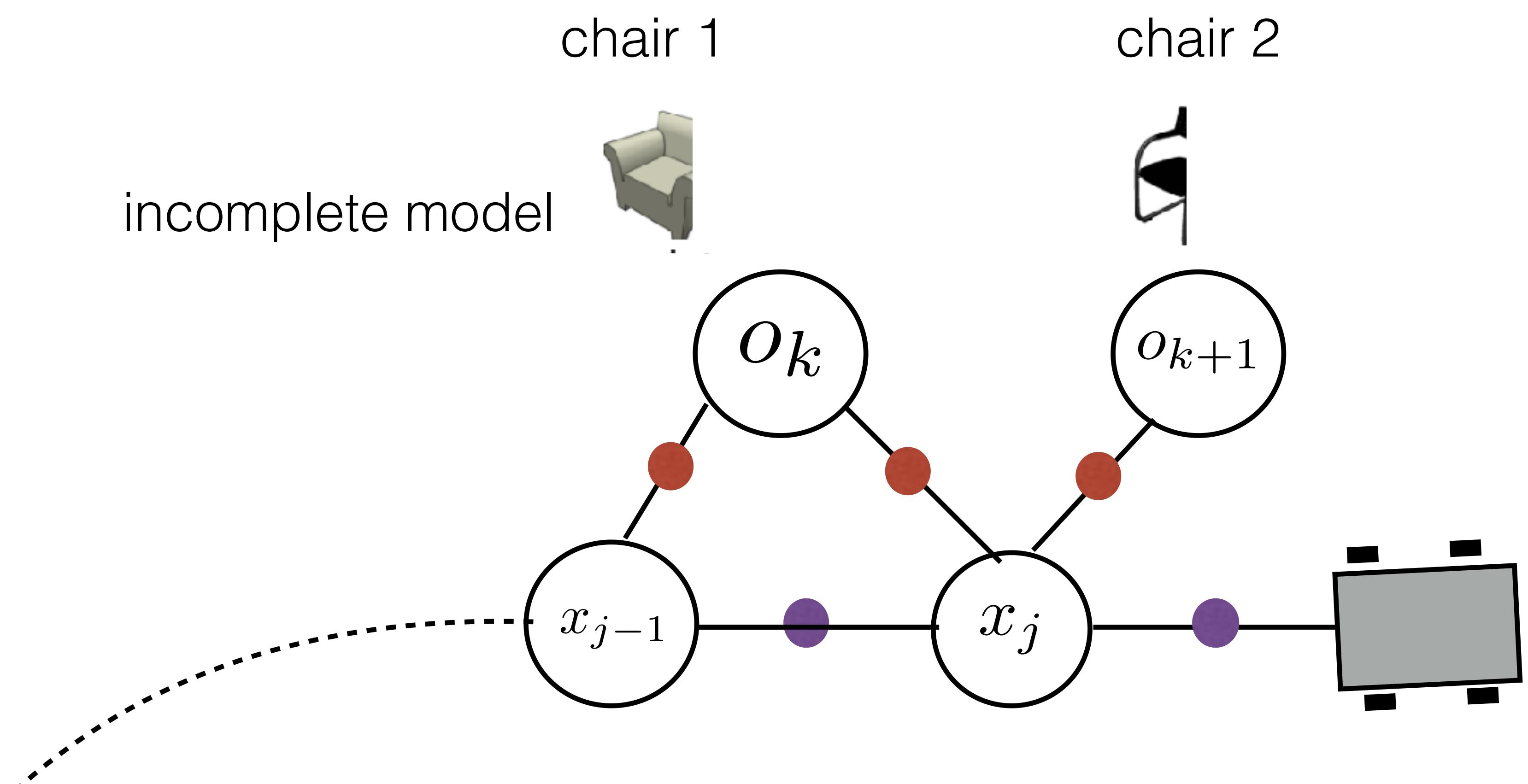


each robot detects object category and models them at the instance level in the local coordinate frame.

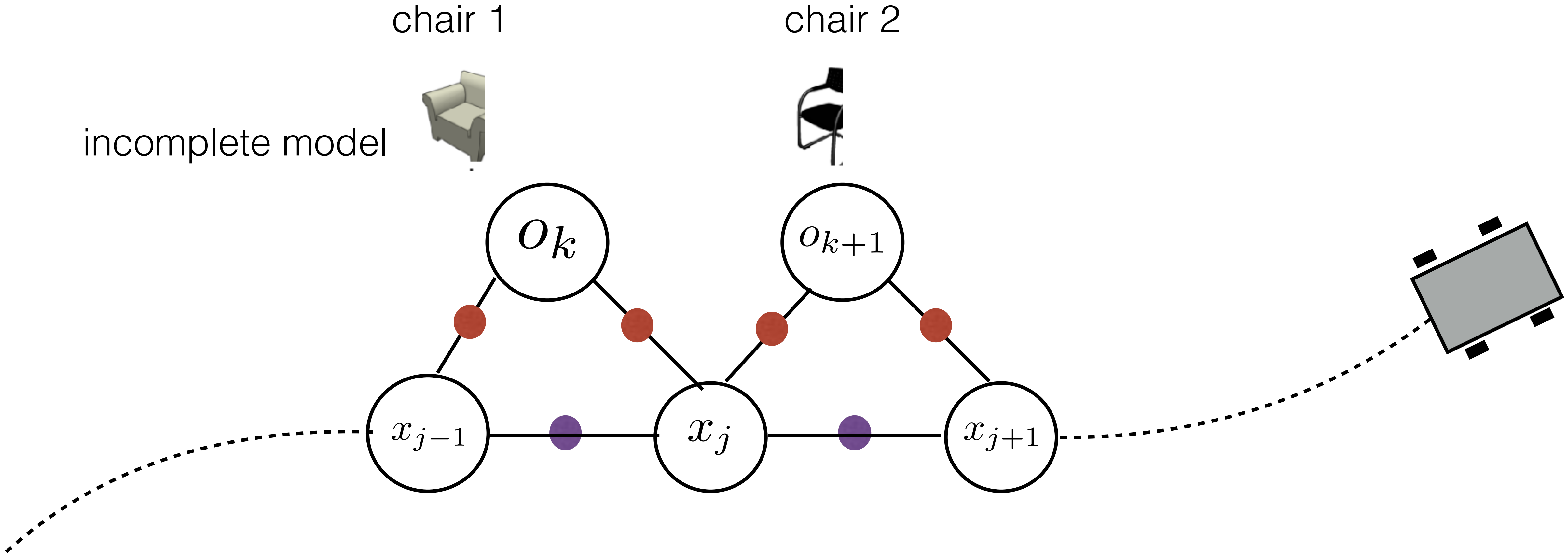
Object-based SLAM with Joint Object Modeling and Mapping



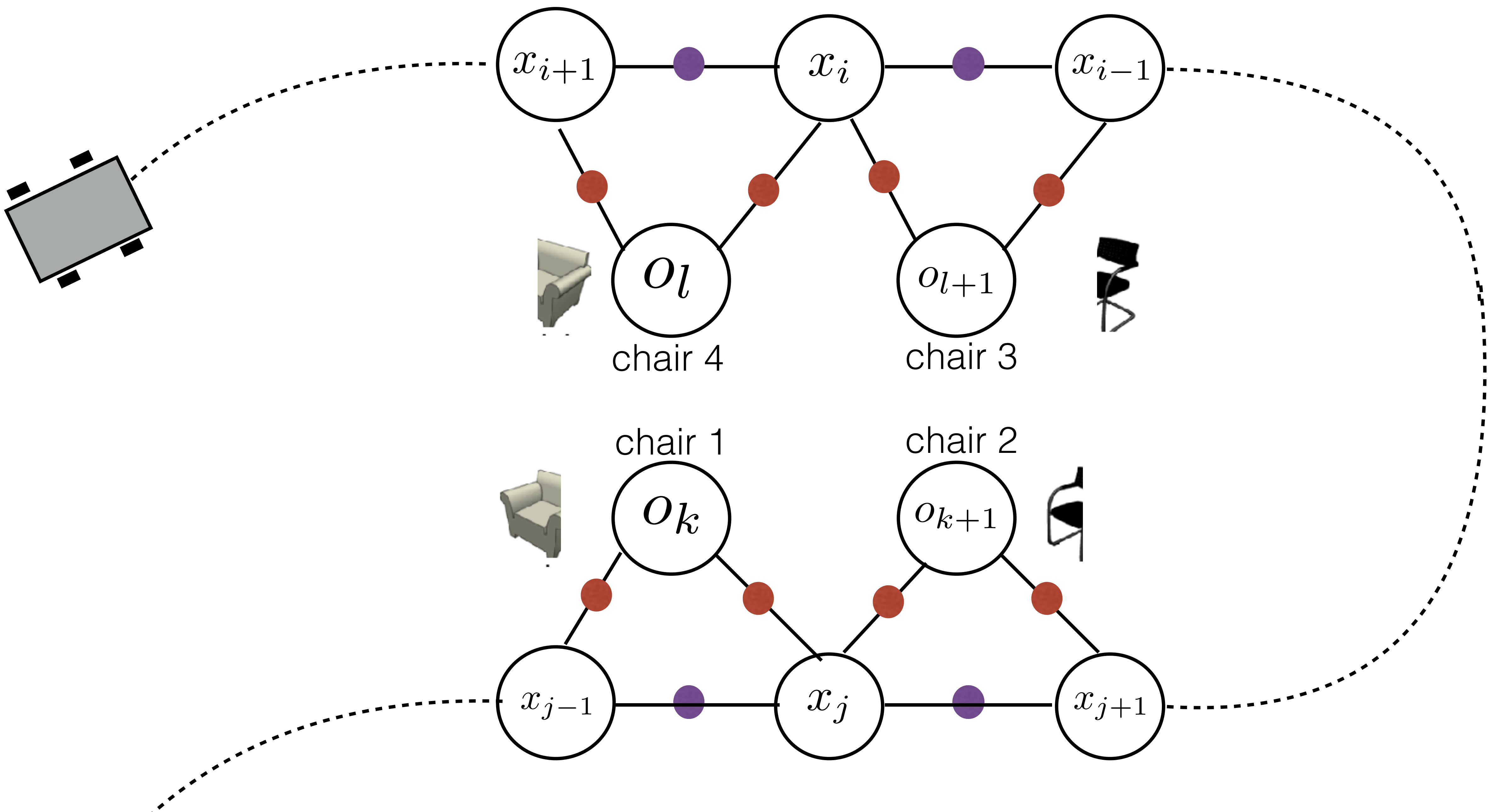
Object-based SLAM with Joint Object Modeling and Mapping



Object-based SLAM with Joint Object Modeling and Mapping

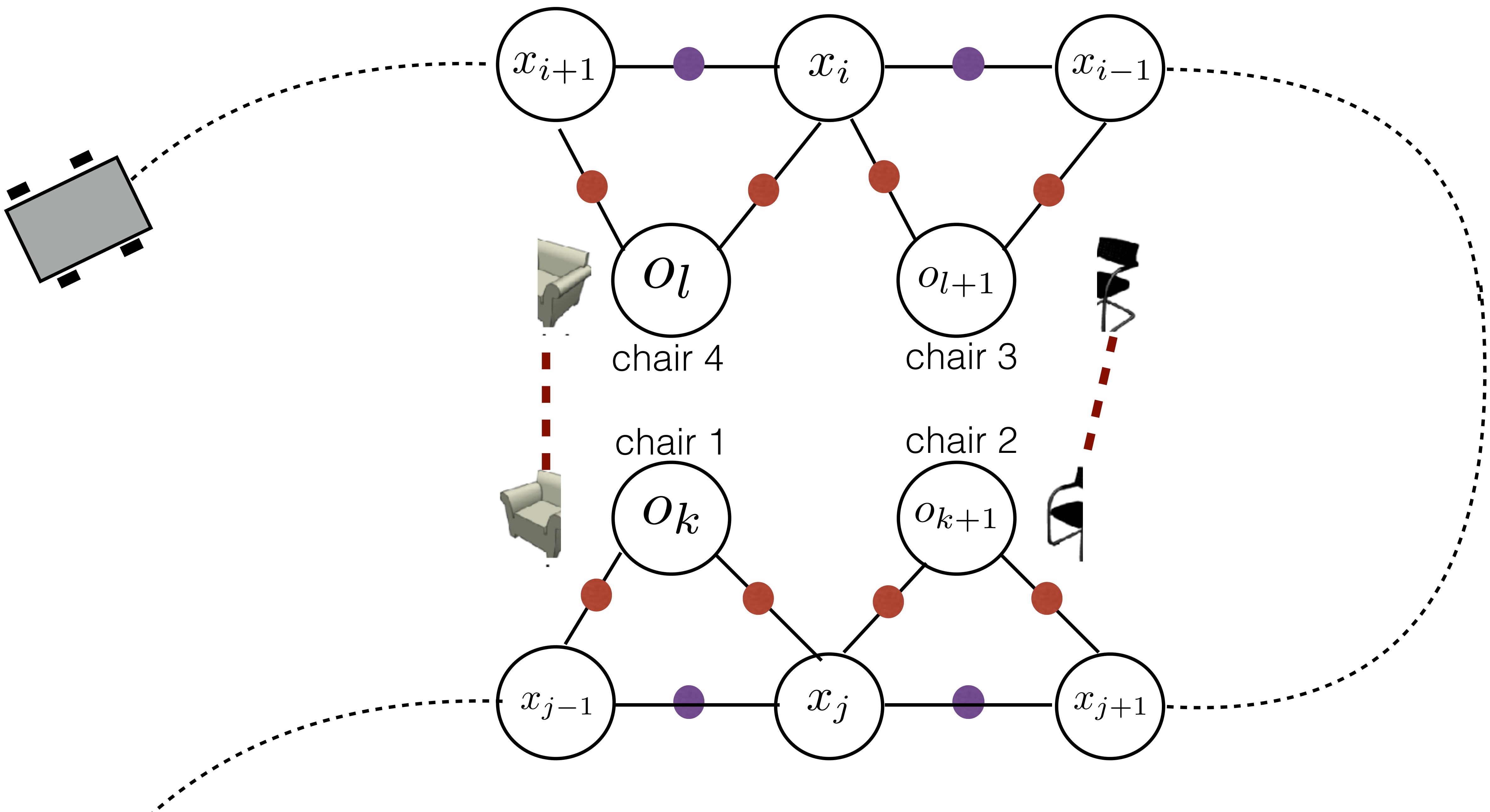


Object-based SLAM
with Joint Object Modeling and Mapping



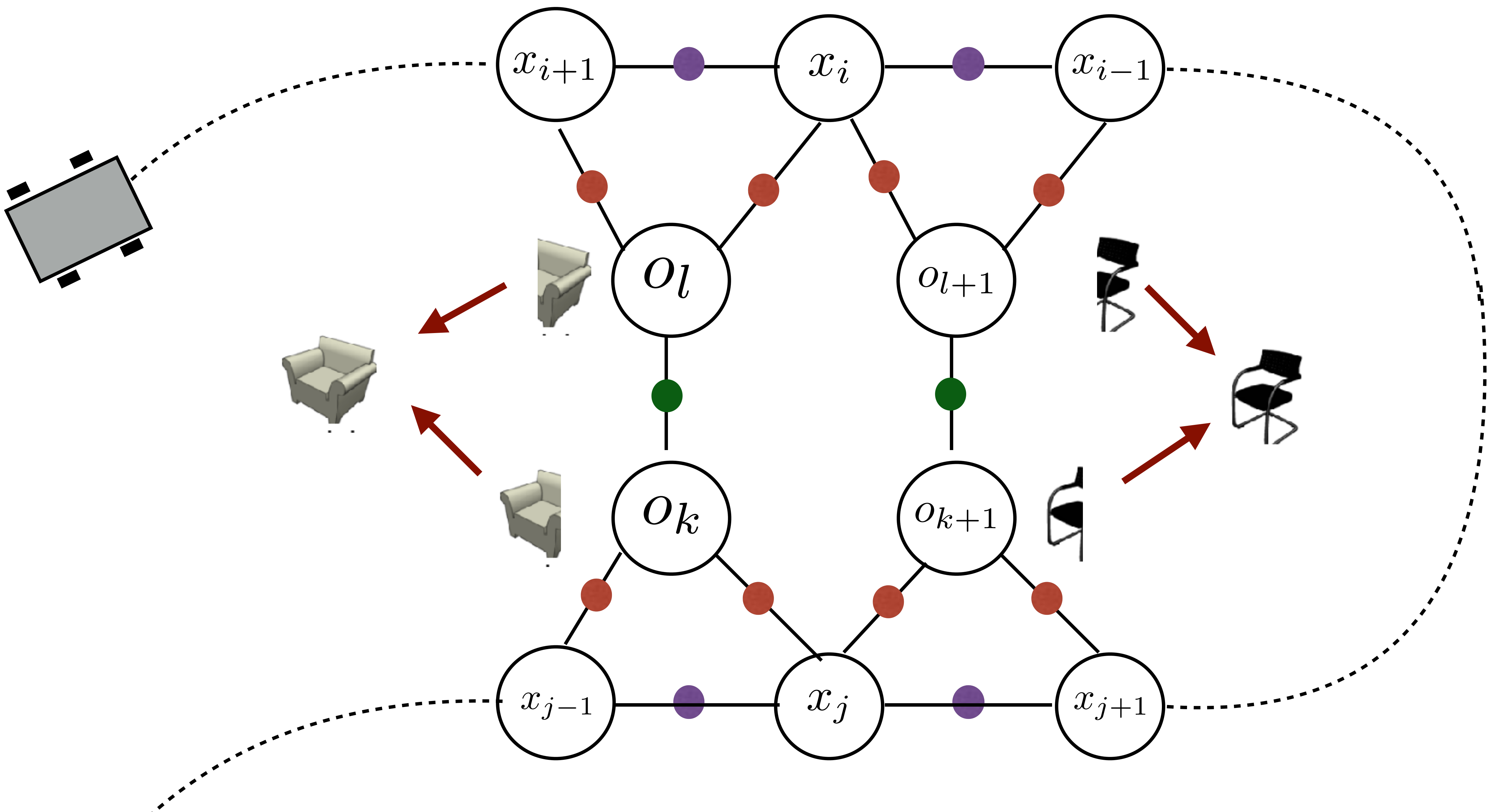
When the robot is back to the same area after a loop, it again models the objects in the same area

Object-based SLAM with Joint Object Modeling and Mapping



The object-object loop closure thread runs in parallel and matches the modeled objects

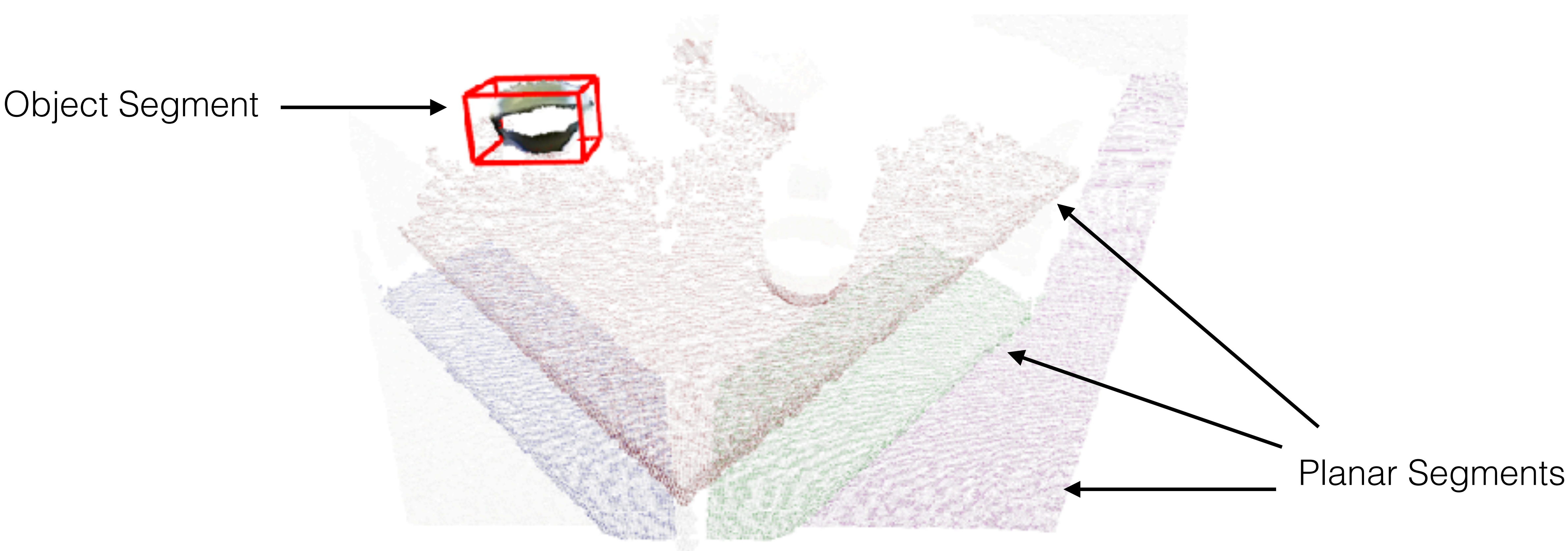
Object-based SLAM with Joint Object Modeling and Mapping



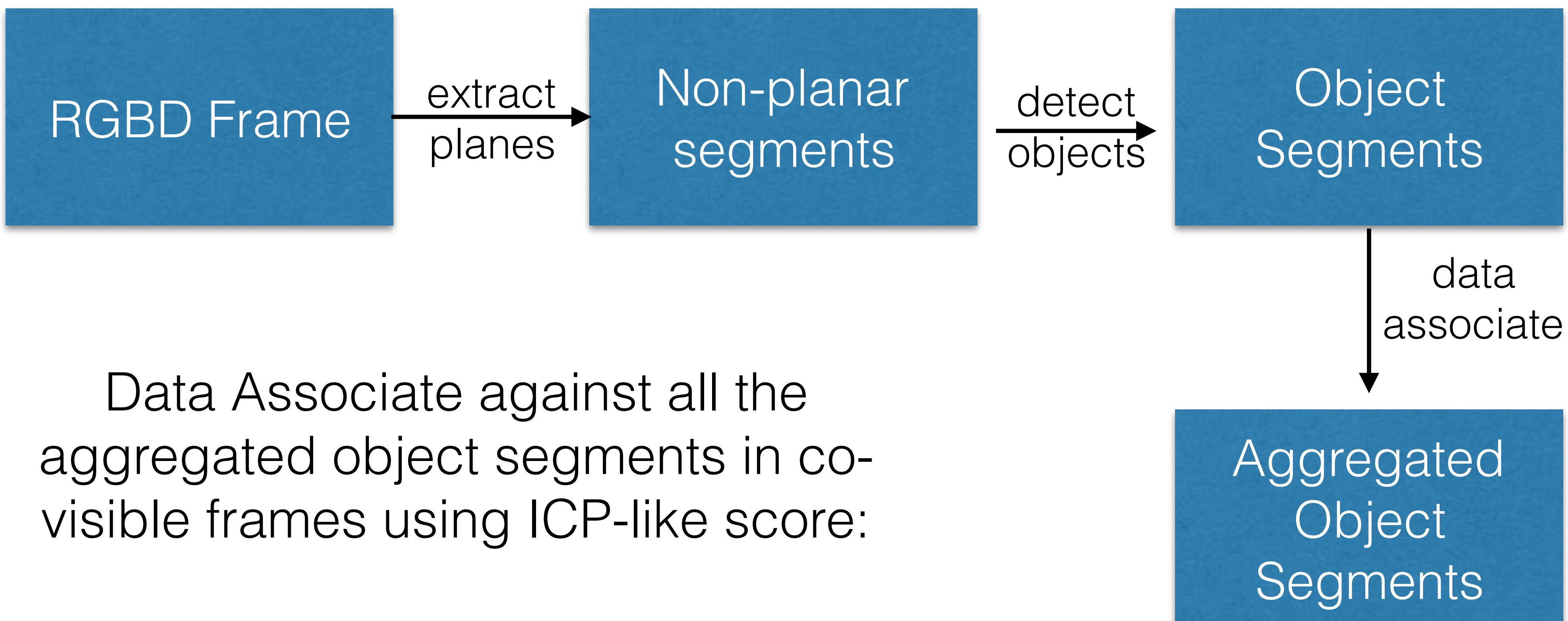
A factor between the matching object landmarks is added which is then optimized using Gauss-Newton method.

Object-based SLAM with Joint Object Modeling and Mapping

RGBD Frame $\xrightarrow{\text{extract planes}}$ Non-planar segments $\xrightarrow{\text{detect objects}}$ Object Segments



Object-based SLAM with Joint Object Modeling and Mapping



Data Associate against all the aggregated object segments in co-visible frames using ICP-like score:

if 50% of the points in an object segment have a distance of 2 cm or less to the nearest aggregated object segment, it is considered a match

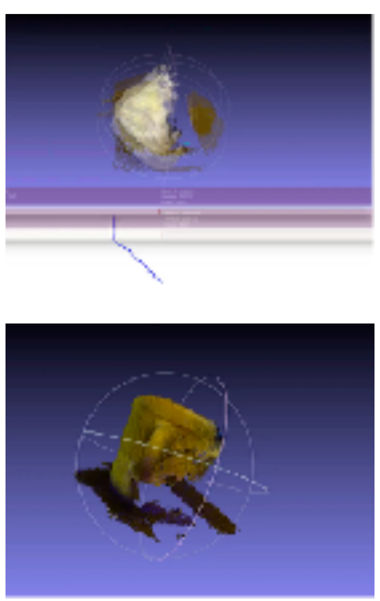
Object-based SLAM with Joint Object Modeling and Mapping

RGBD Frame $\xrightarrow{\text{extract planes}}$ Non-planar segments $\xrightarrow{\text{detect objects}}$ Object Segments

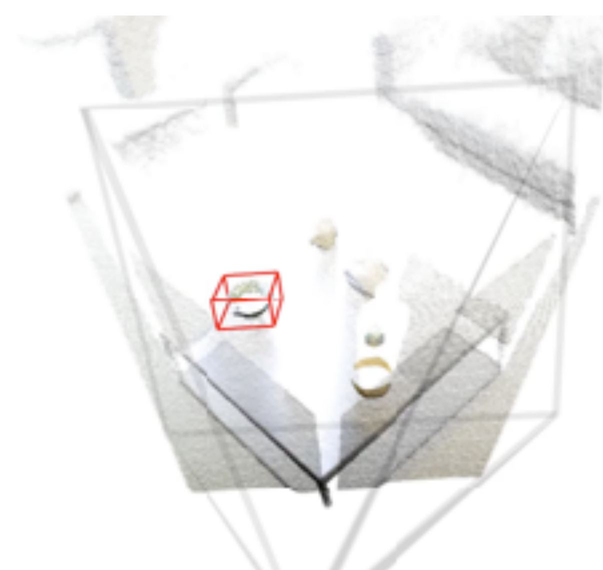
Object Segments $\xrightarrow{\text{data associate}}$ Aggregated Object Segments

Aggregated Object Segments $\xrightarrow{\text{if visible in more than K frames and aggregated max category probability is > threshold}}$ Add Object as Landmark

Add Object as Landmark → Object-Object loop closure using 3DMatch



sample
object models



3DMatch: Learning Local Geometric Descriptors from 3D Reconstructions
Zeng et al. (CVPR 2017)

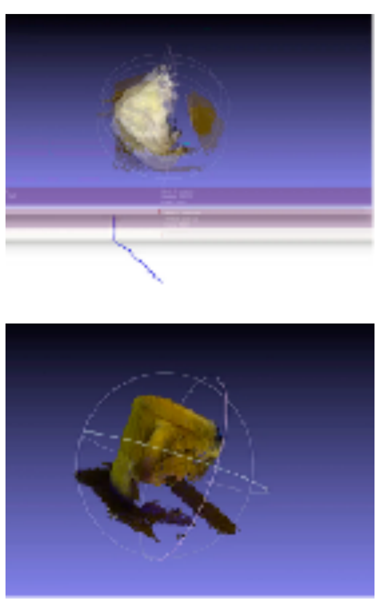
Object-based SLAM with Joint Object Modeling and Mapping

RGBD Frame → extract planes → Non-planar segments → detect objects → Object Segments

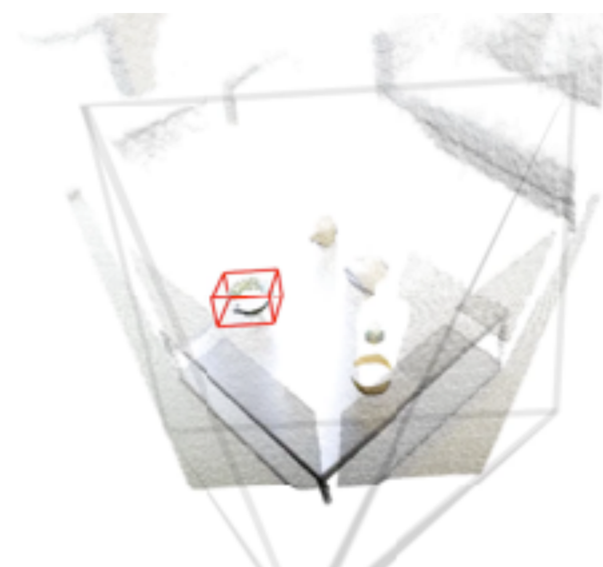
Object Segments → data associate → Aggregated Object Segments

Aggregated Object Segments → if visible in more than K frames and aggregated max category probability is > threshold → Add Object as Landmark

Add Object as Landmark → Object-Object loop closure using 3DMatch



sample
object models



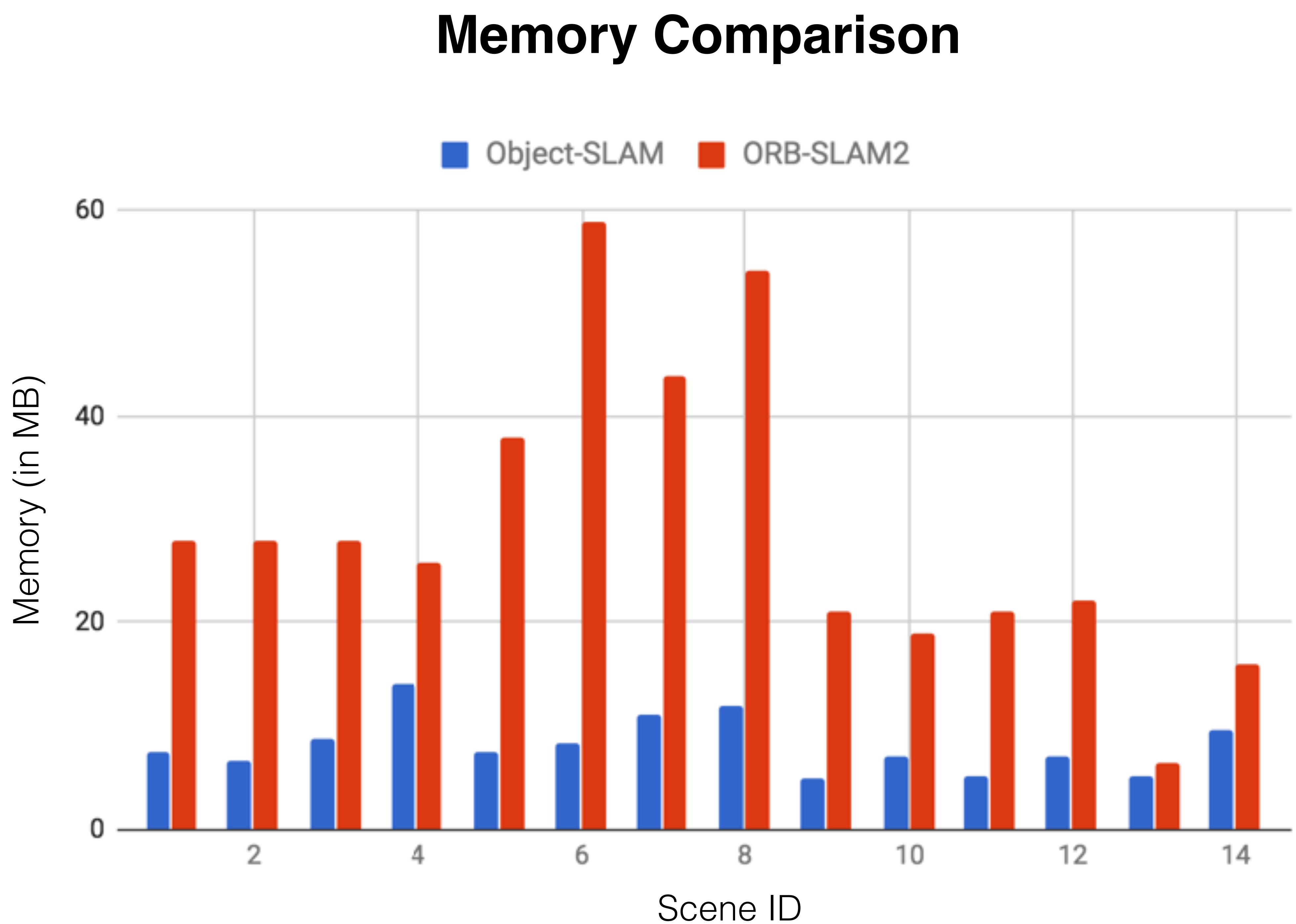
3DMatch: Learning Local Geometric Descriptors from 3D Reconstructions
Zeng et al. (CVPR 2017)

Experiments
(UW RGB-D Scenes v2 dataset)

Our approach is as accurate as ORB-SLAM2

Object detector is fine-tuned on UW Scenes v1 and UW Object dataset

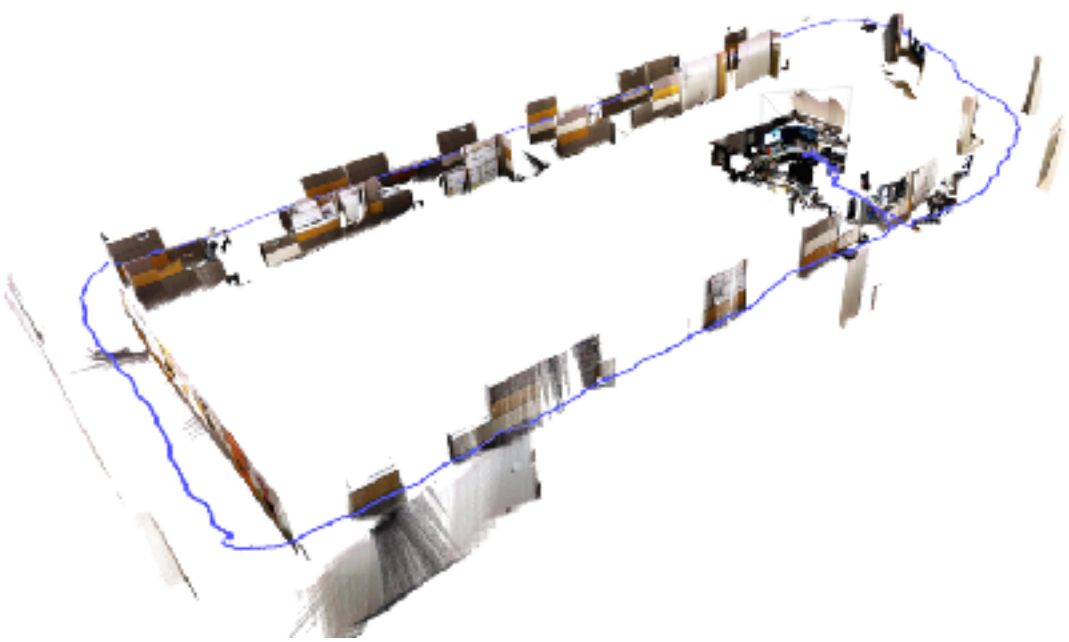
Experiments
(UW RGB-D Scenes v2 dataset)



Our approach requires much less memory than ORB-SLAM2

Experiments
(Large-Scale handheld dataset)

Object-SLAM



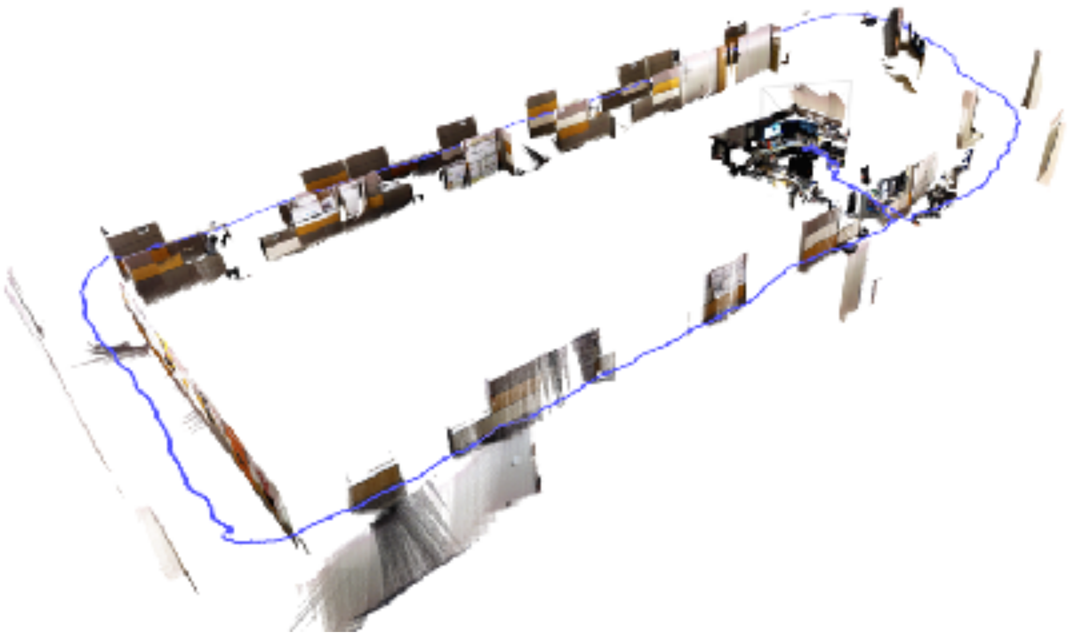
ORB-SLAM2

Size of the map: **21 m X 21 m**
Number of Poses: **5284**
Distance travelled: **93.09 m**

Pre-trained COCO Object detector is used

Experiments (Large-Scale handheld dataset)

Object-SLAM



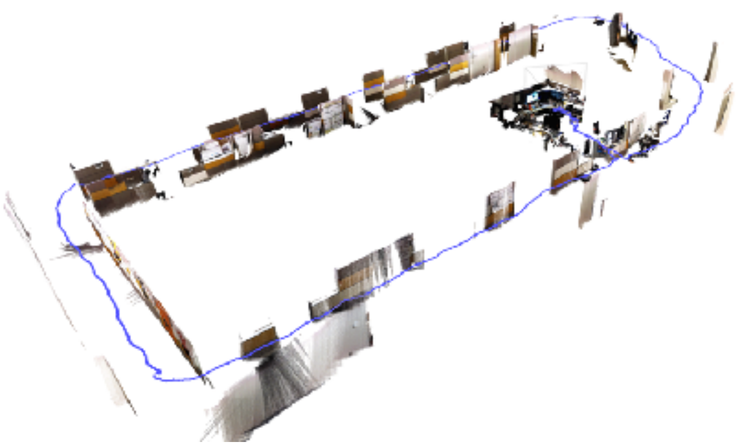
ORB-SLAM2

Size of the map: **21 m X 21 m**
Number of Poses: **5284**
Distance travelled: **93.09 m**

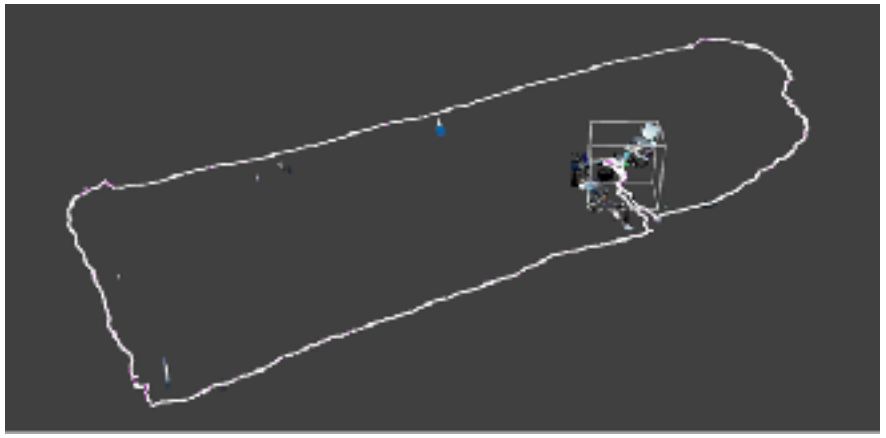
Pre-trained COCO Object detector is used

Experiments (Large-Scale handheld dataset)

Object-SLAM



ORB-SLAM2



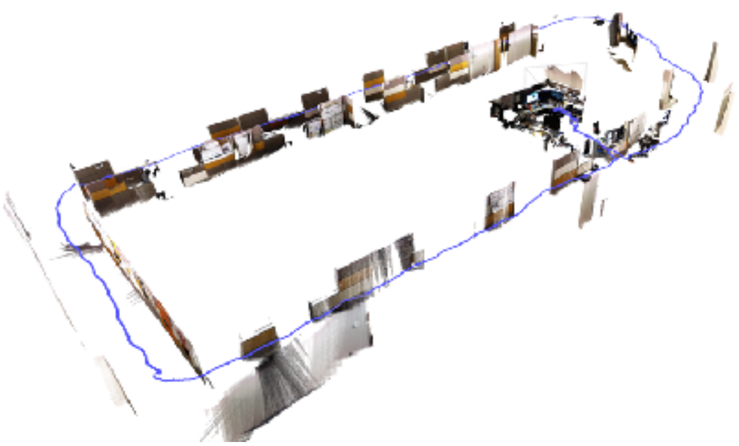
Kintinuous

Size of the map: **21 m X 21 m**
Number of Poses: **5284**
Distance travelled: **93.09 m**

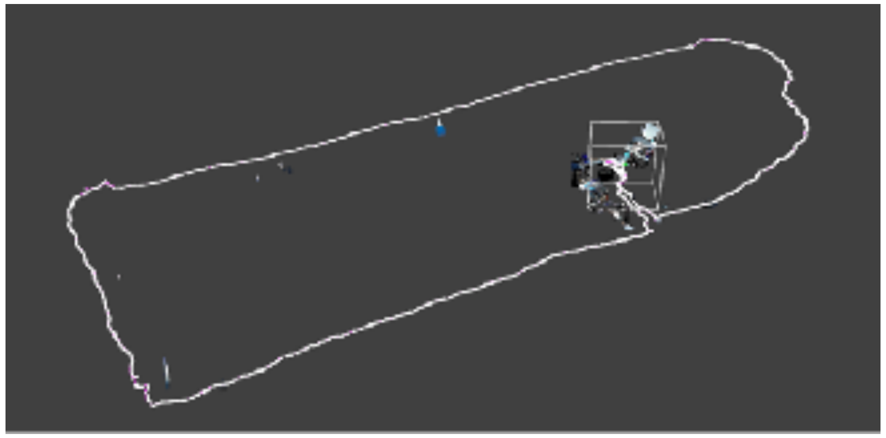
Pre-trained COCO Object detector is used

Experiments
(Large-Scale handheld dataset)

Object-SLAM



ORB-SLAM2



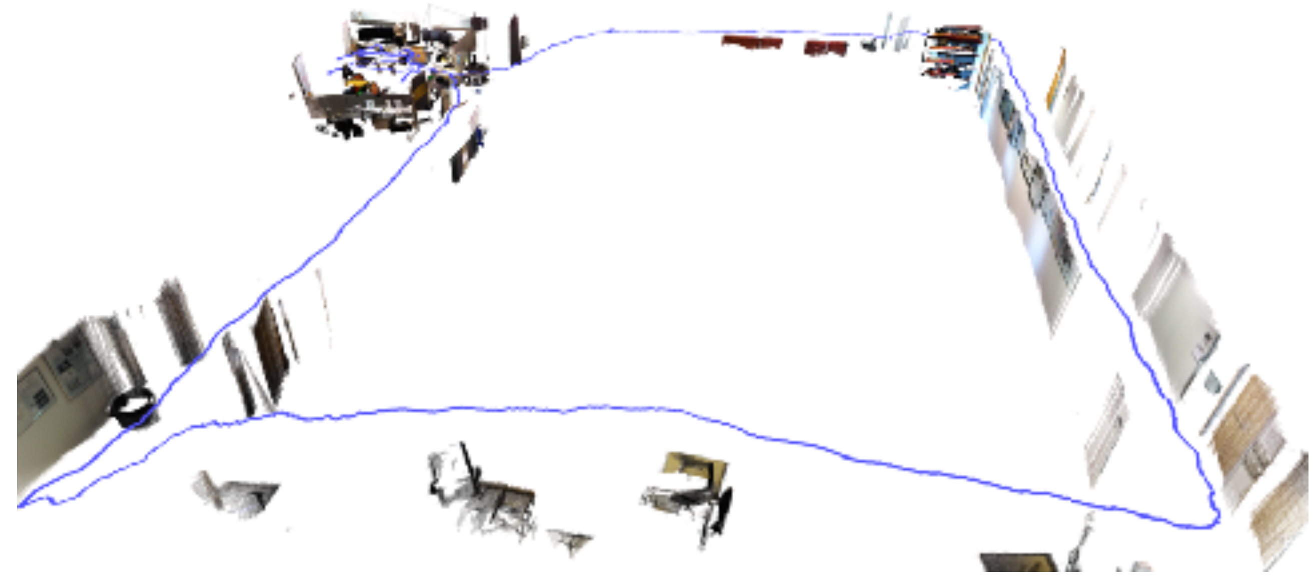
Kintinuous

Size of the map: **21 m X 21 m**
Number of Poses: **5284**
Distance travelled: **93.09 m**

Pre-trained COCO Object detector is used

Experiments
(Large-Scale handheld dataset)

Object-SLAM



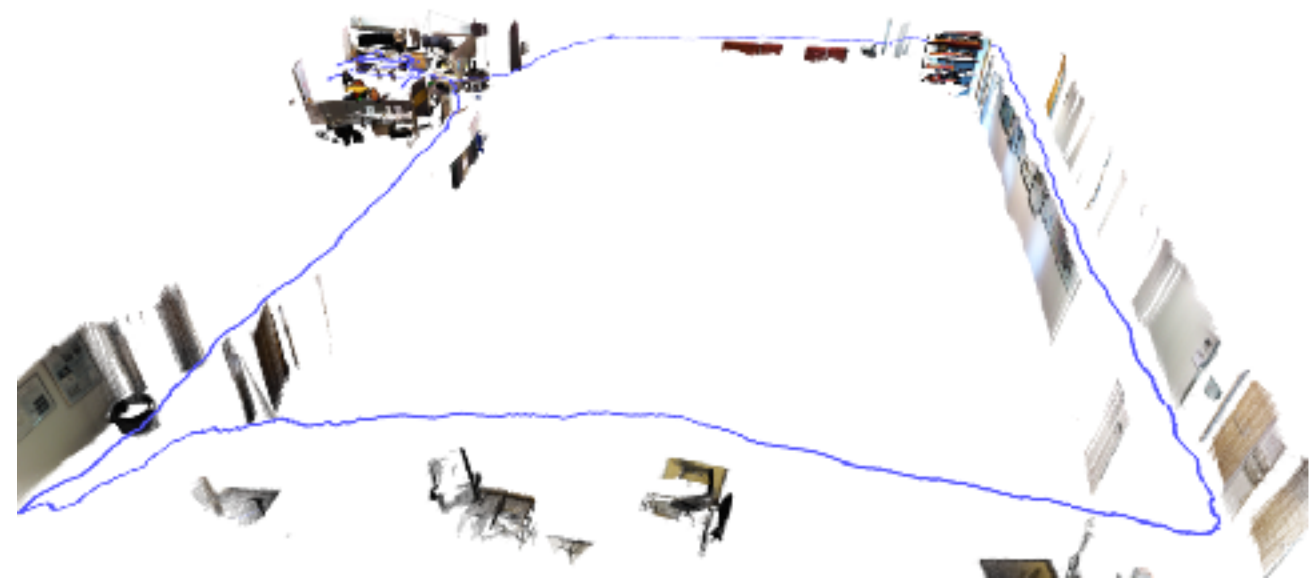
ORB-SLAM2

Size of the map: **19 m X 18 m**
Number of Poses: **6382**
Distance travelled: **102.84 m**

Pre-trained COCO Object detector is used

Experiments
(Large-Scale handheld dataset)

Object-SLAM



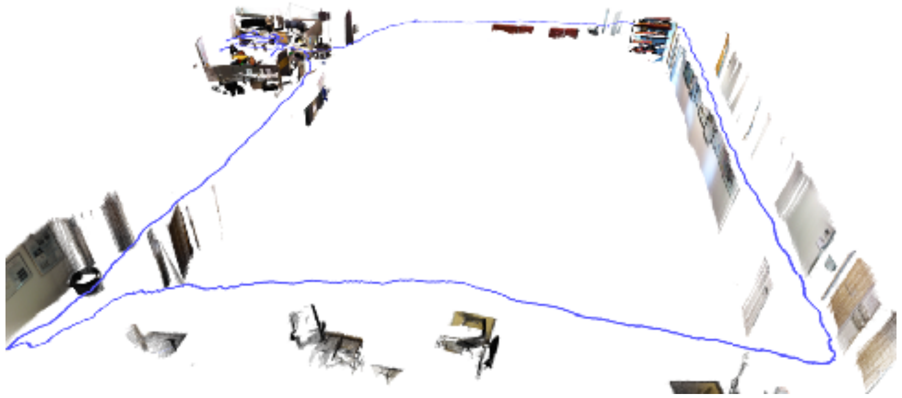
ORB-SLAM2

Size of the map: **19 m X 18 m**
Number of Poses: **6382**
Distance travelled: **102.84 m**

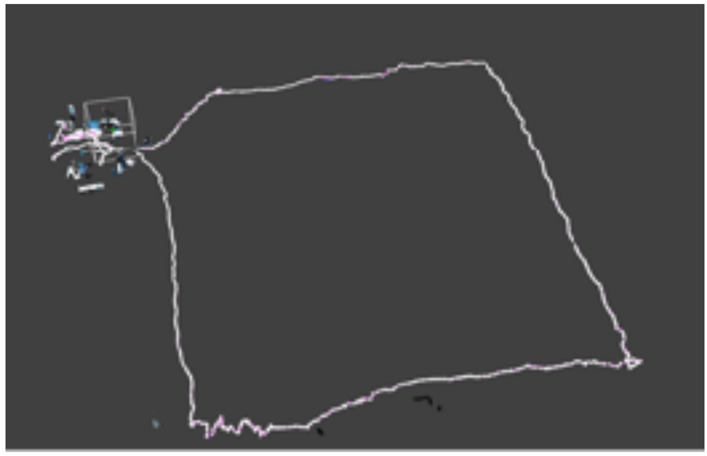
Pre-trained COCO Object detector is used

Experiments (Large-Scale handheld dataset)

Object-SLAM



ORB-SLAM2

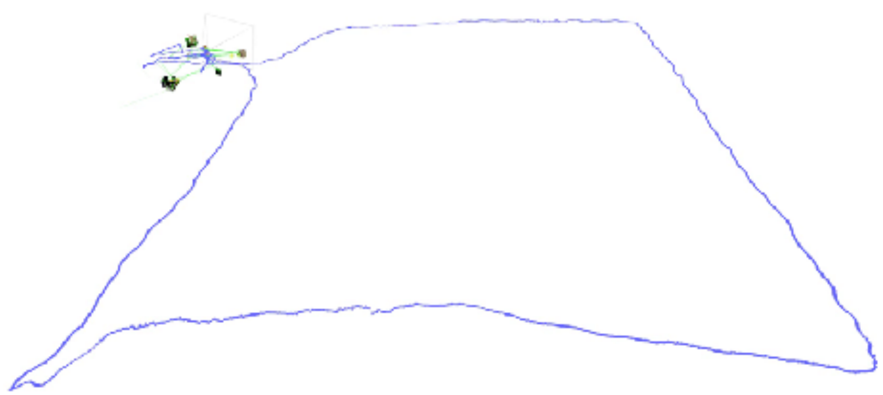


Kintinuous

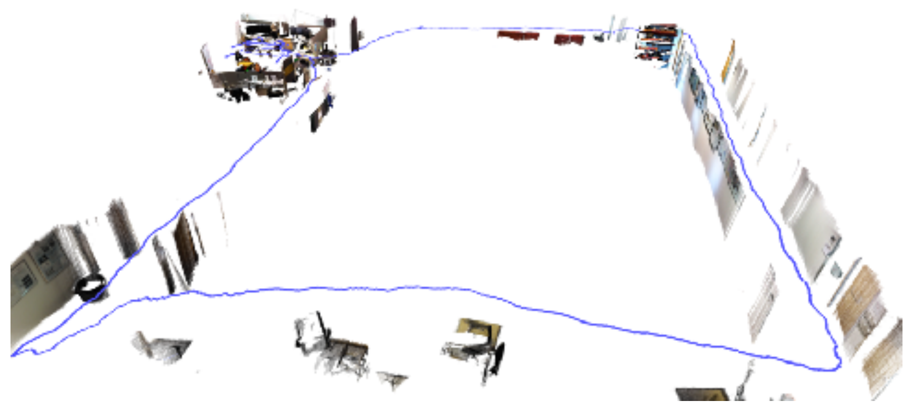
Size of the map: **19 m X 18 m**
Number of Poses: **6382**
Distance travelled: **102.84 m**

Pre-trained COCO Object detector is used

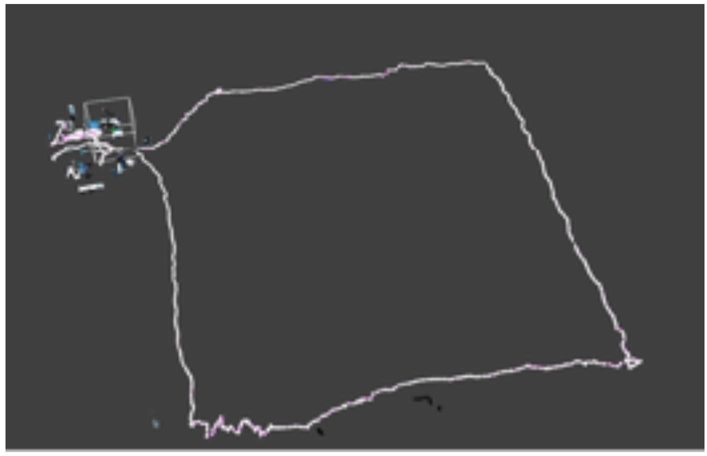
Experiments
(Large-Scale handheld dataset)



Object-SLAM



ORB-SLAM2



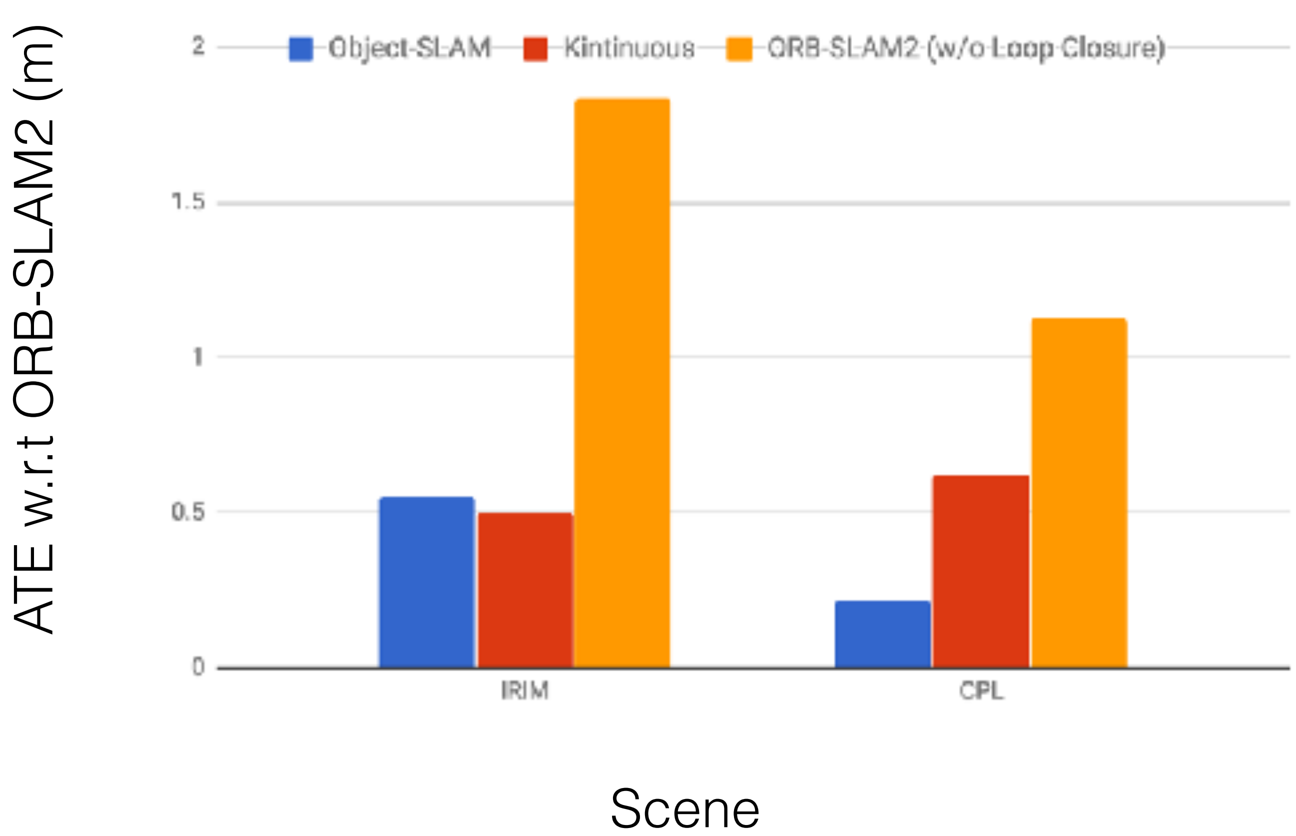
Kintinuous

Size of the map: **19 m X 18 m**
Number of Poses: **6382**
Distance travelled: **102.84 m**

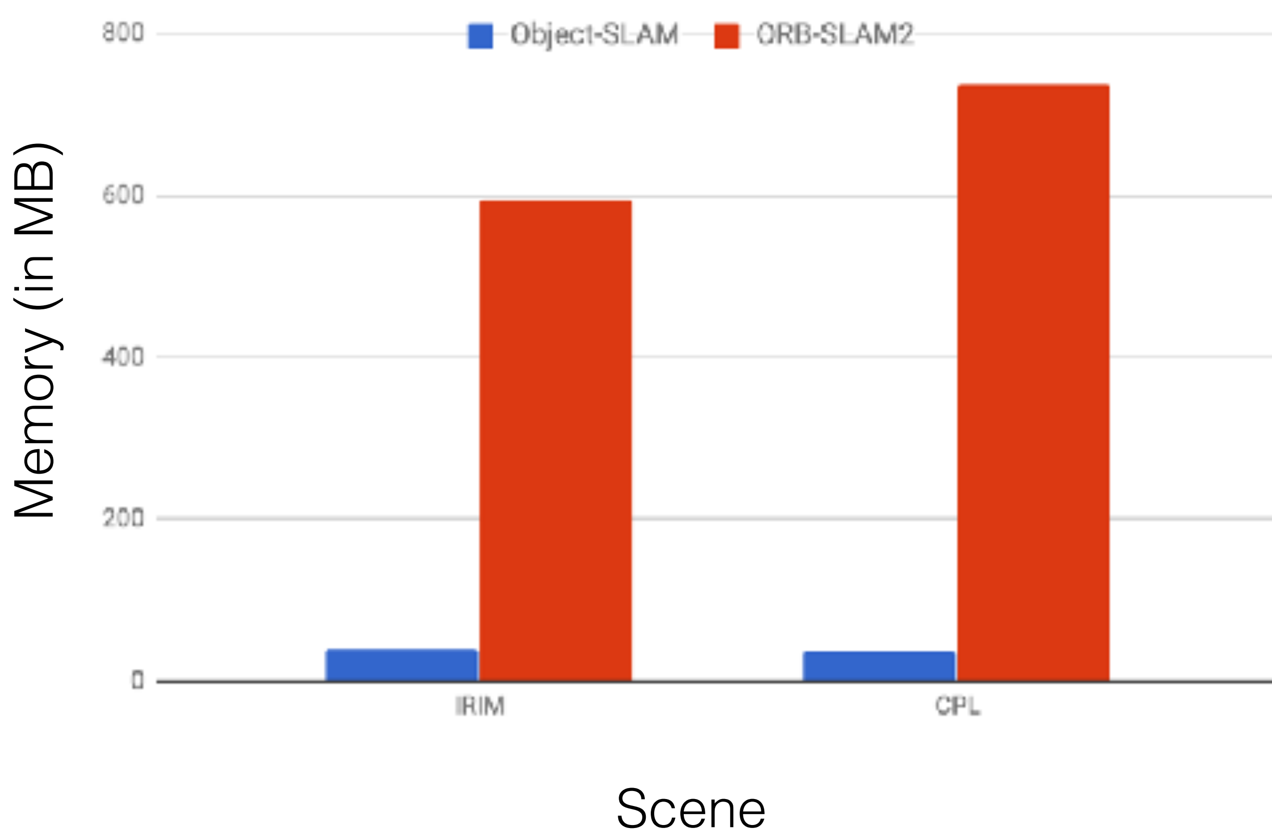
Pre-trained COCO Object detector is used

Experiments
(Large-Scale handheld dataset)

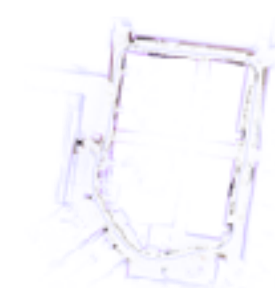
Trajectory Error (w.r.t ORB-SLAM2)



Memory Requirements



Experiments (Large-Scale Robot dataset)

	Object-SLAM	ORB-SLAM2
Klaus		
Military Training Facility		
IRIM		
CPL		

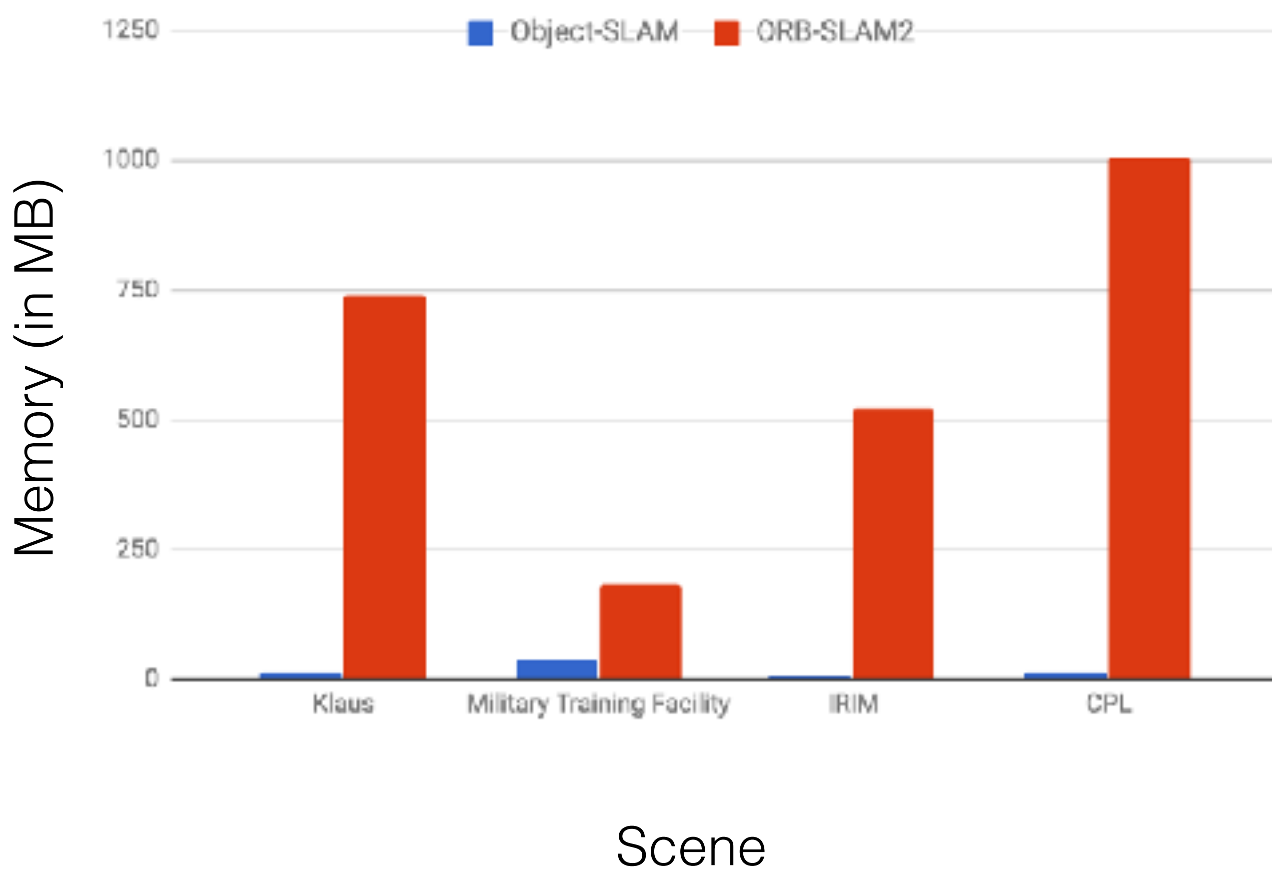
Object detector is fine-tuned on BigBird dataset

Experiments (Large-Scale Robot dataset)

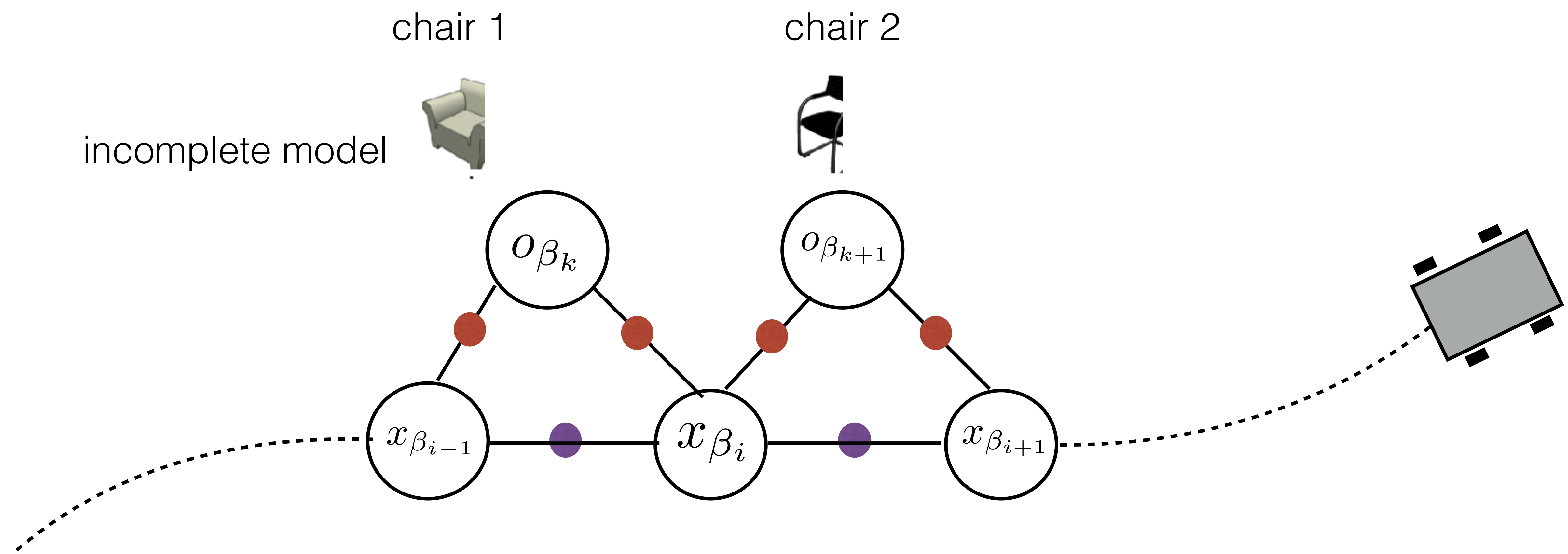
Trajectory Error (w.r.t ORB-SLAM2)

Scene	ATE (m)
Klaus	0.28
Military Facility	0.15
IRIM	0.10
CPL	0.32

Memory Requirements

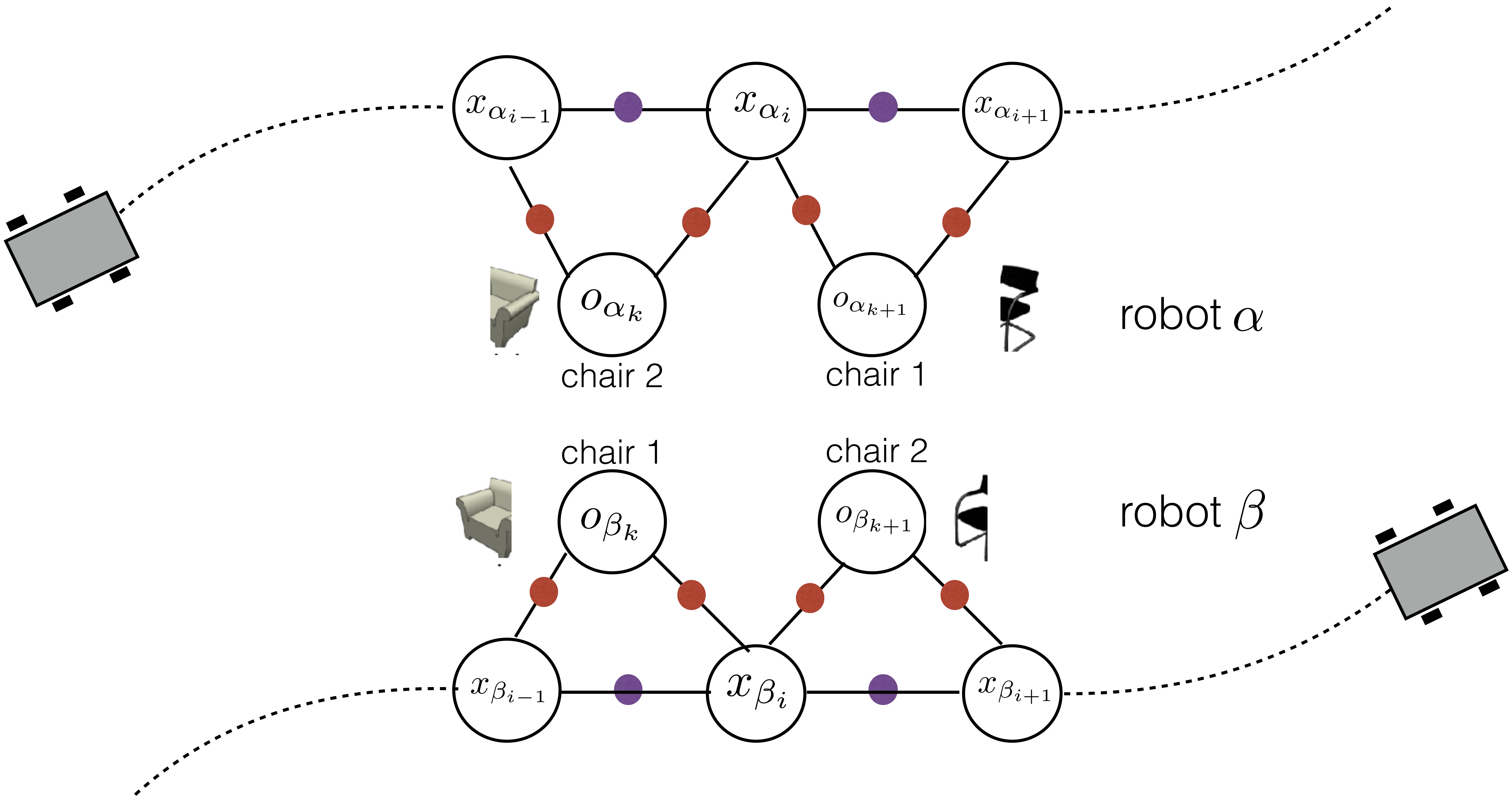


Distributed Object-based SLAM with Joint Object Modeling and Mapping



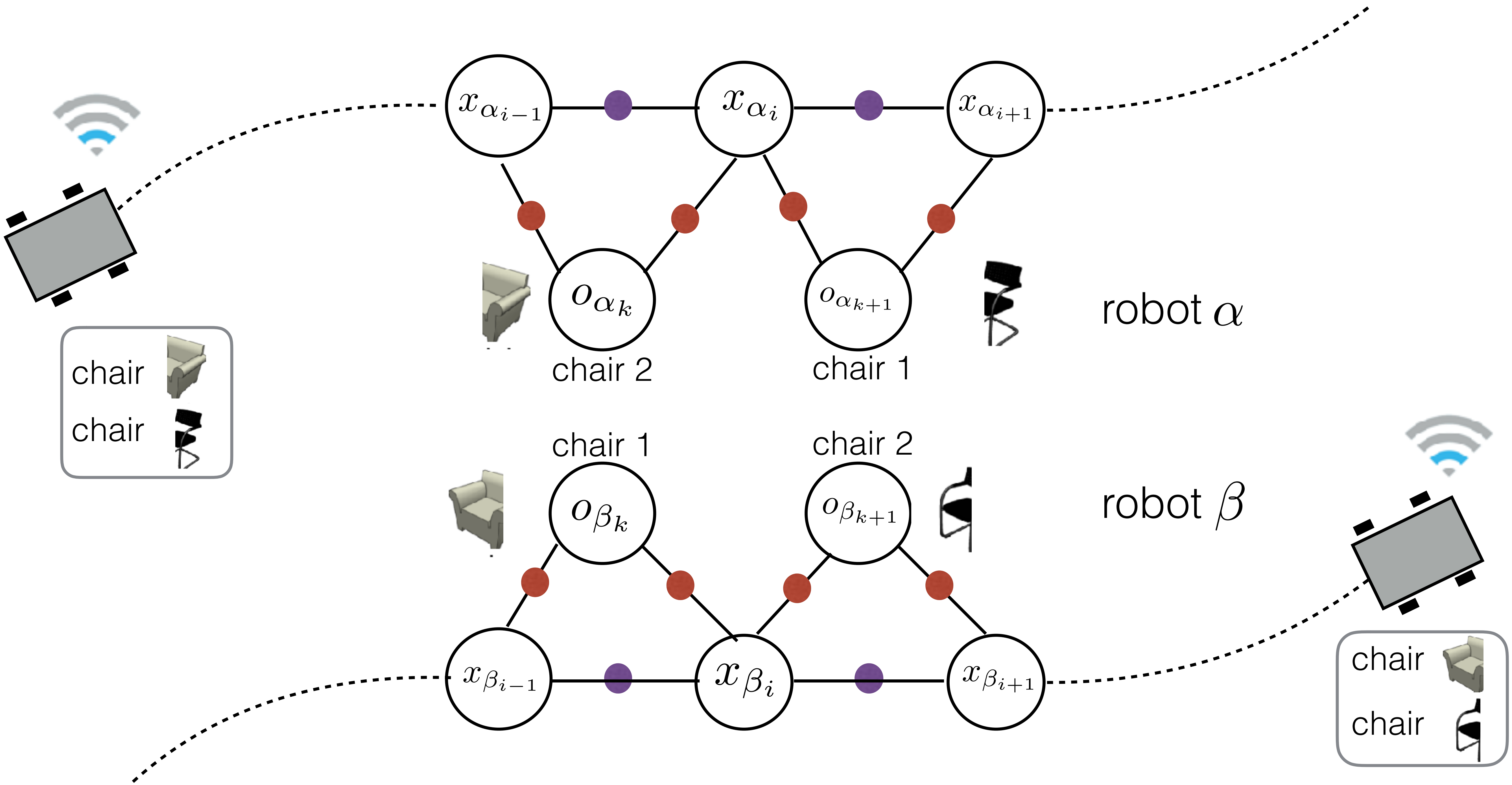
each robot will perform object slam with joint object modeling and mapping

Distributed Object-based SLAM with Joint Object Modeling and Mapping



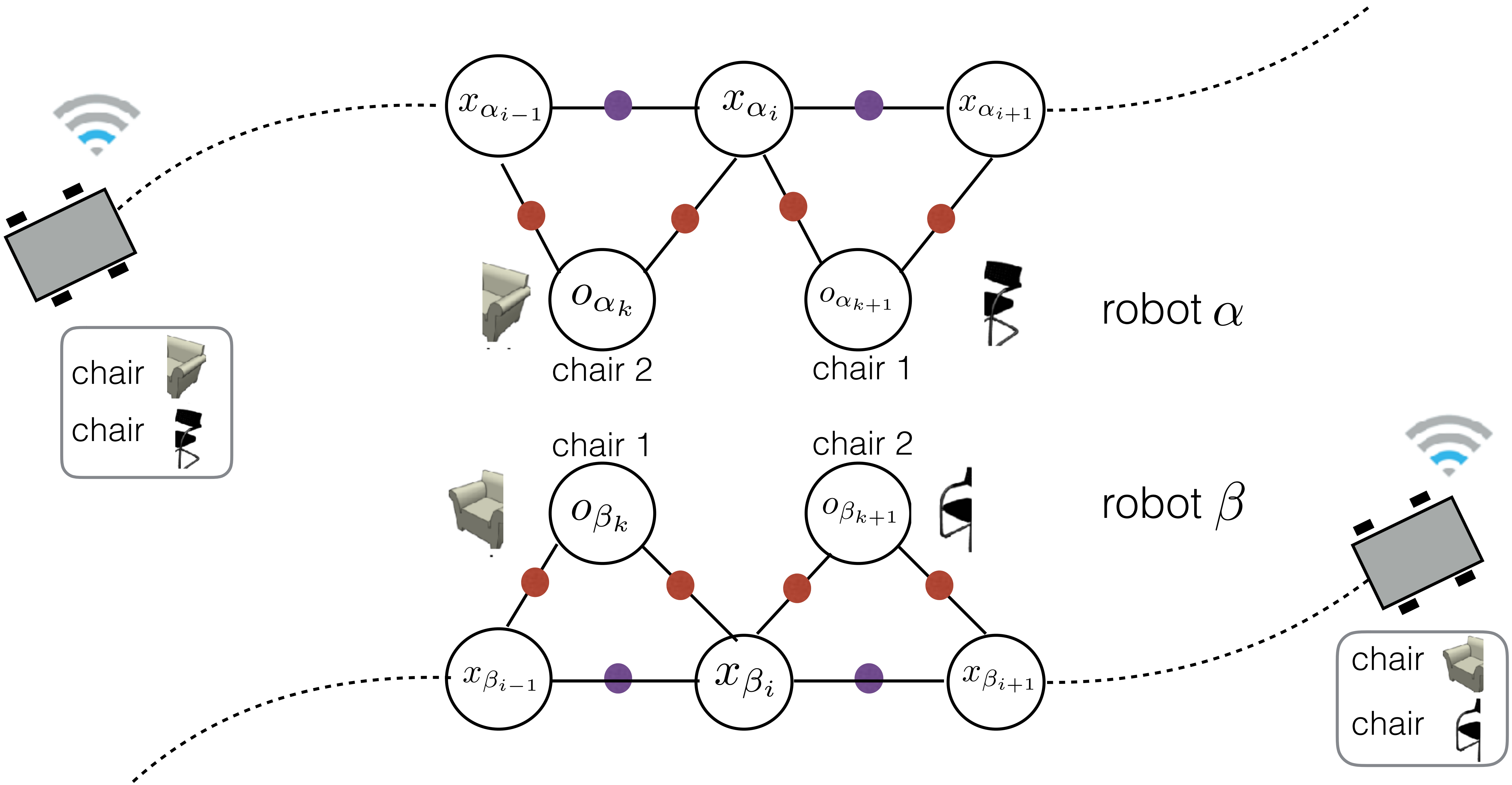
each robot will perform object slam with joint object modeling and mapping

Distributed Object-based SLAM with Joint Object Modeling and Mapping



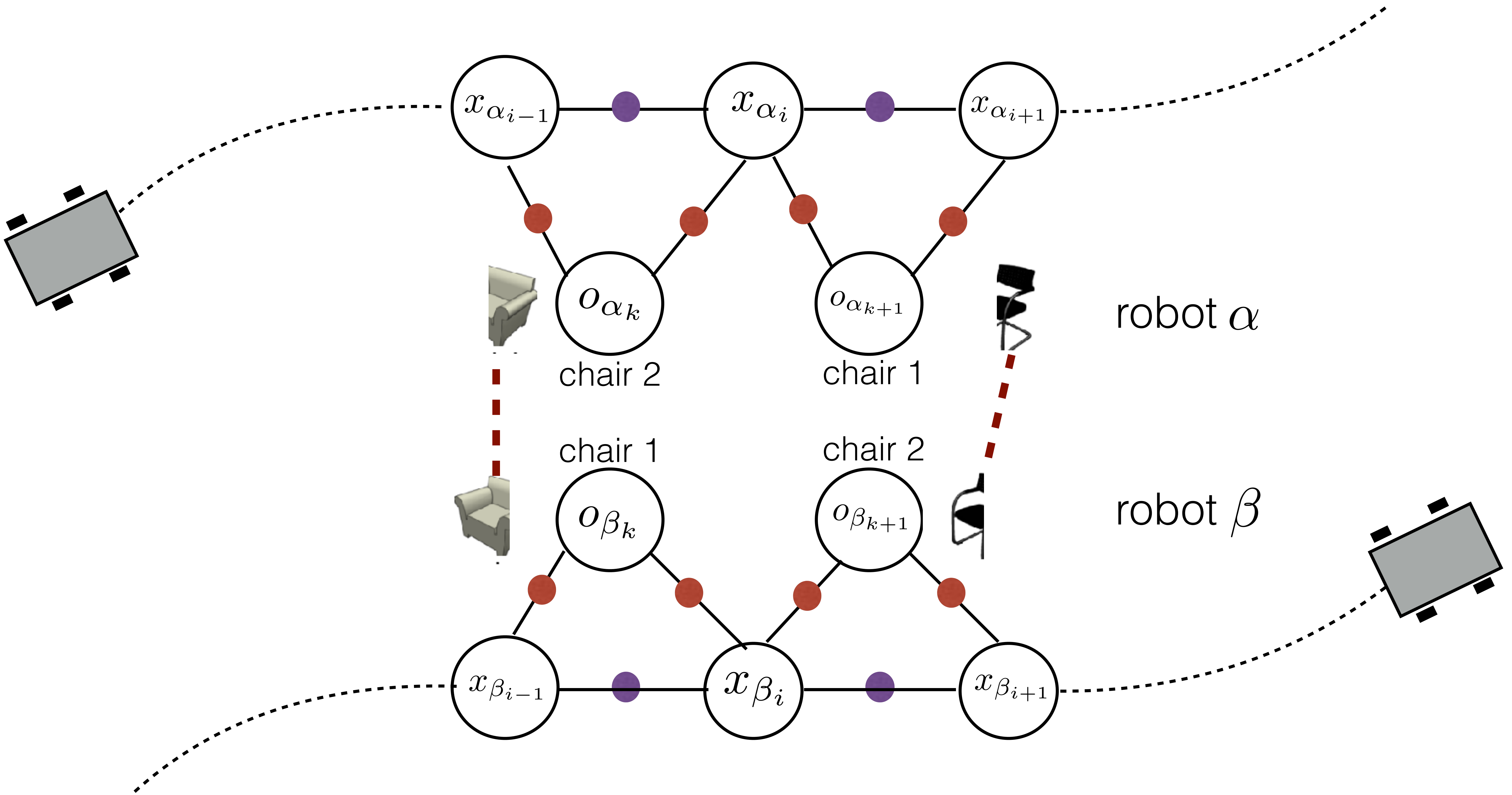
each robot communicates object category and the corresponding models to the other robot

Distributed Object-based SLAM with Joint Object Modeling and Mapping



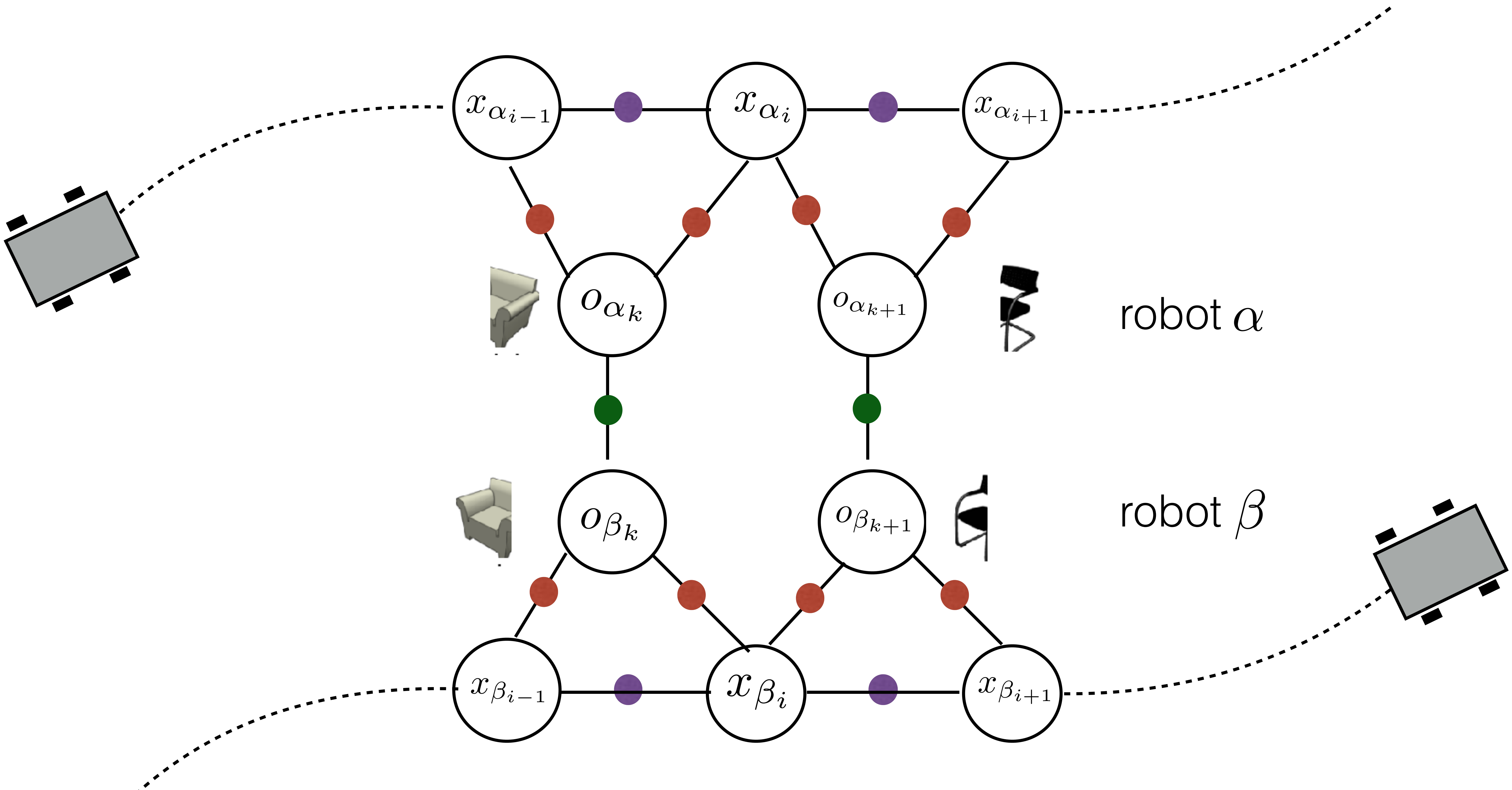
each robot communicates object category and the corresponding models to the other robot

Distributed Object-based SLAM with Joint Object Modeling and Mapping



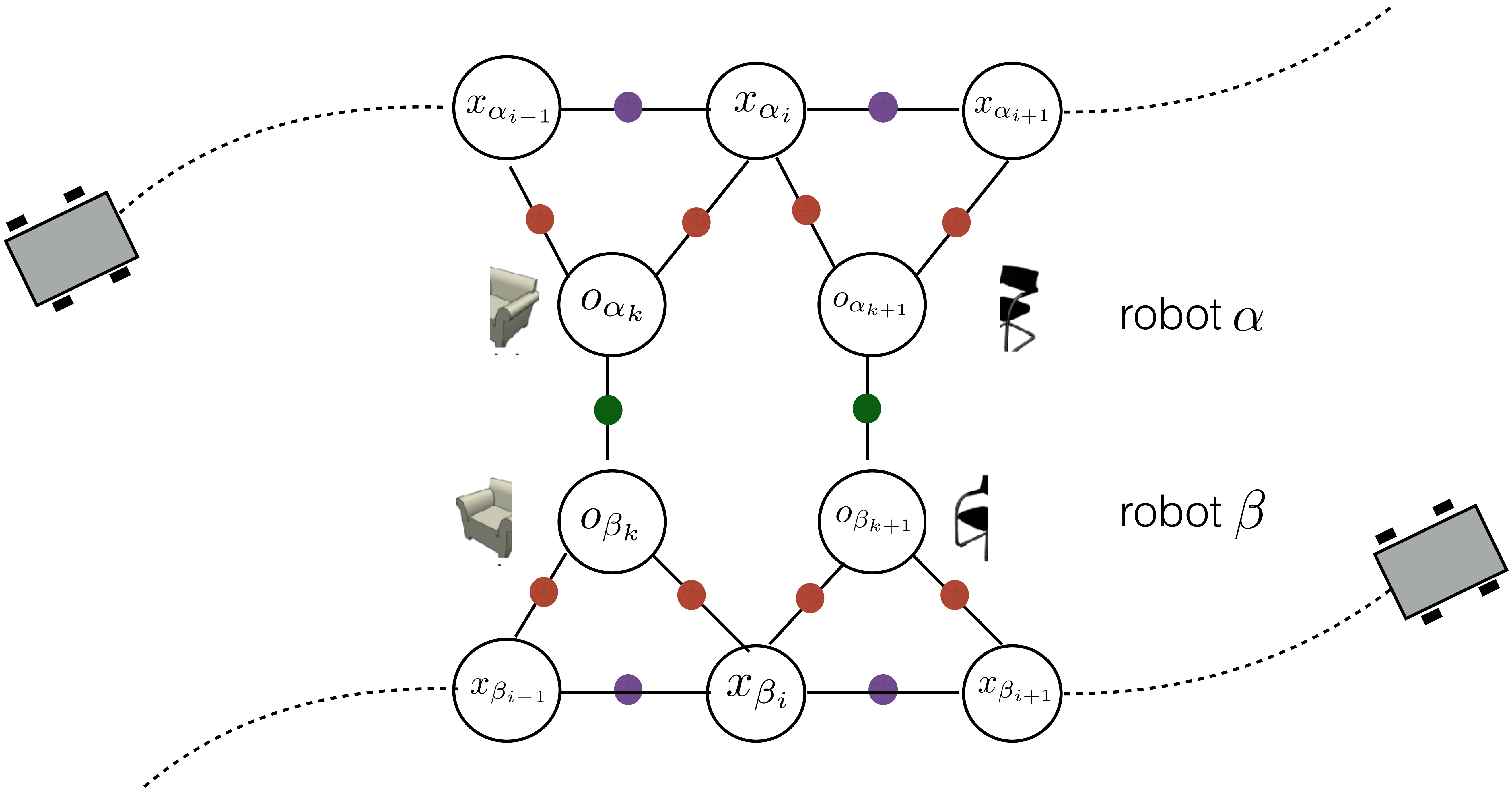
modeled objects in one robot are matched to the corresponding models from the other robot

Distributed Object-based SLAM with Joint Object Modeling and Mapping



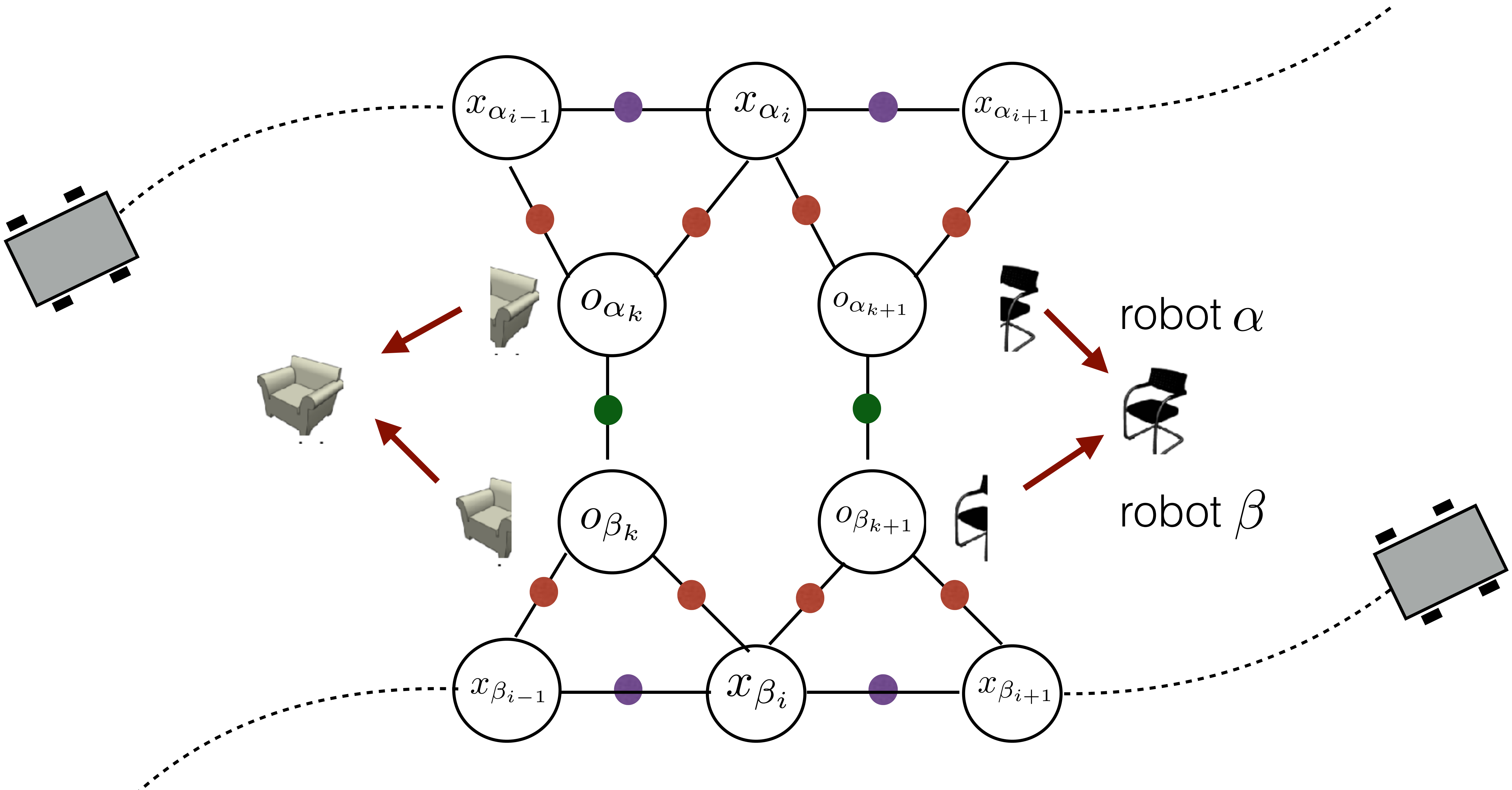
Relative pose estimated using 3DMatch is used as the measurement in inter-robot object-object factor

Distributed Object-based SLAM with Joint Object Modeling and Mapping



Factor graph optimized using Distributed Gauss-Seidel Algorithm

Distributed Object-based SLAM with Joint Object Modeling and Mapping



Optimized estimates can be used to produce fused object models

Distributed Object-based SLAM with Joint Object Modeling and Mapping

Trade-off

Distributed Object-based SLAM with Joint Object Modeling and Mapping

Trade-off

Memory Requirements

No need to store object models for each object instance

Distributed Object-based SLAM with Joint Object Modeling and Mapping

Trade-off

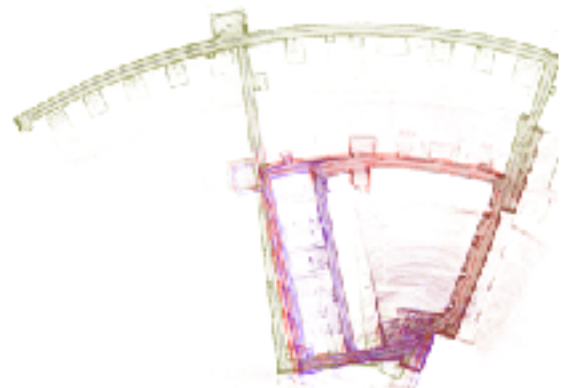
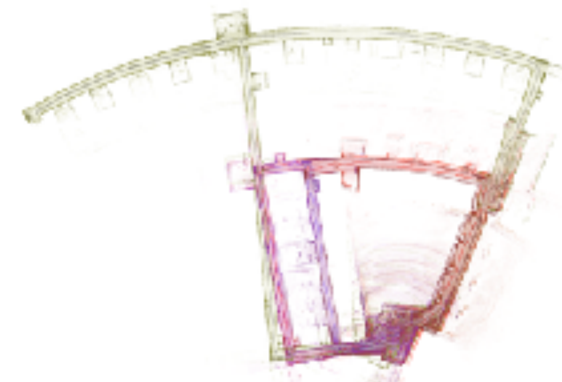
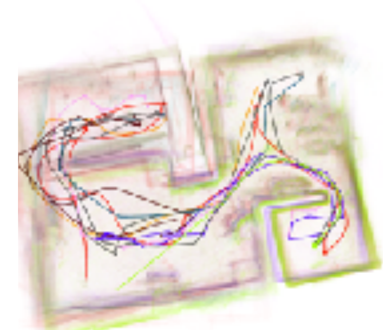
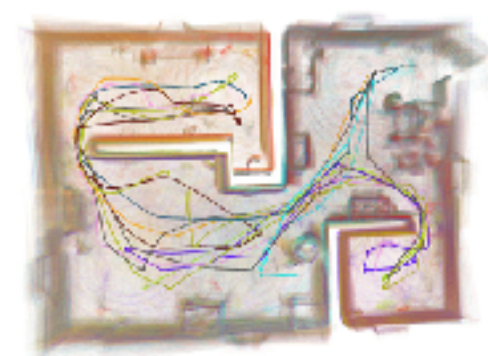
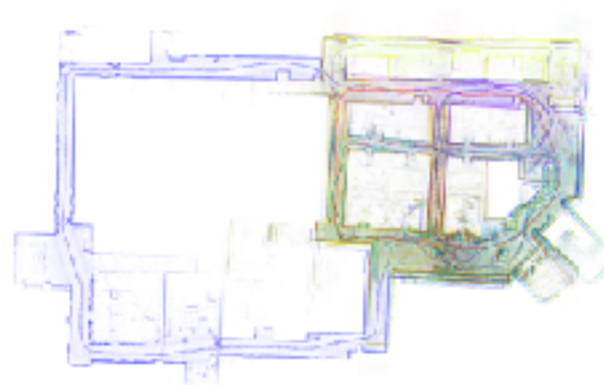
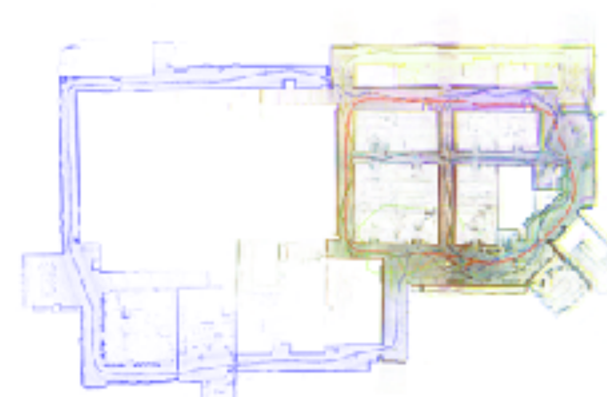
Memory Requirements	↓	No need to store object models for each object instance
Communication Bandwidth	↑	Increases because object models are communicated as well instead of communicating object labels

Distributed Object-based SLAM with Joint Object Modeling and Mapping

Trade-off

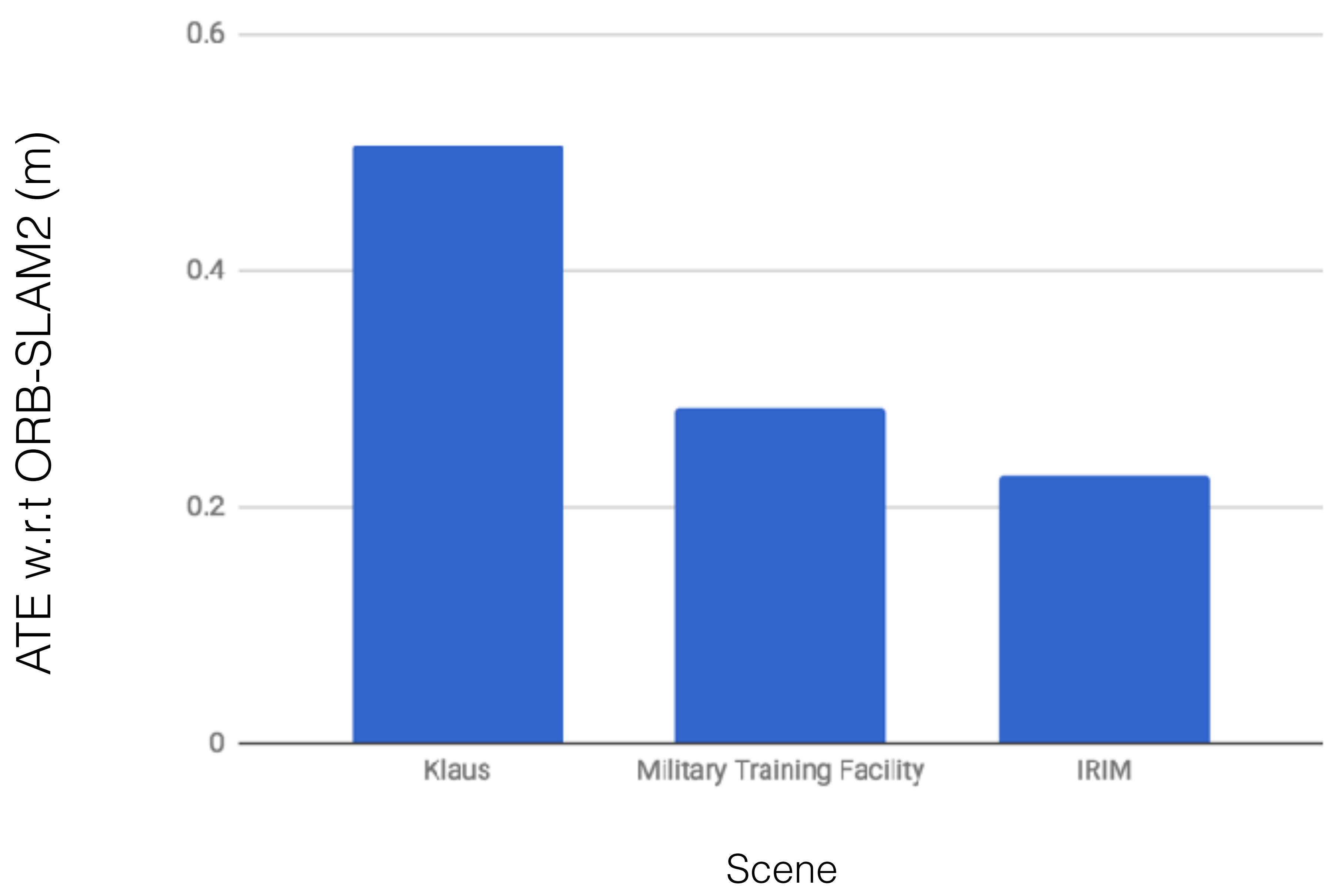
Memory Requirements	↓	No need to store object models for each object instance
Communication Bandwidth	↑	Increases because object models are communicated as well instead of communicating object labels
Generalizability	↑	Generalizes to unknown object instances

Experiments (Large-Scale Robot dataset)

	Distributed Object-SLAM	Distributed ORB-SLAM2
Klaus (5 robots)		
Military Training Facility (10 robots)		
IRIM (11 robots)		

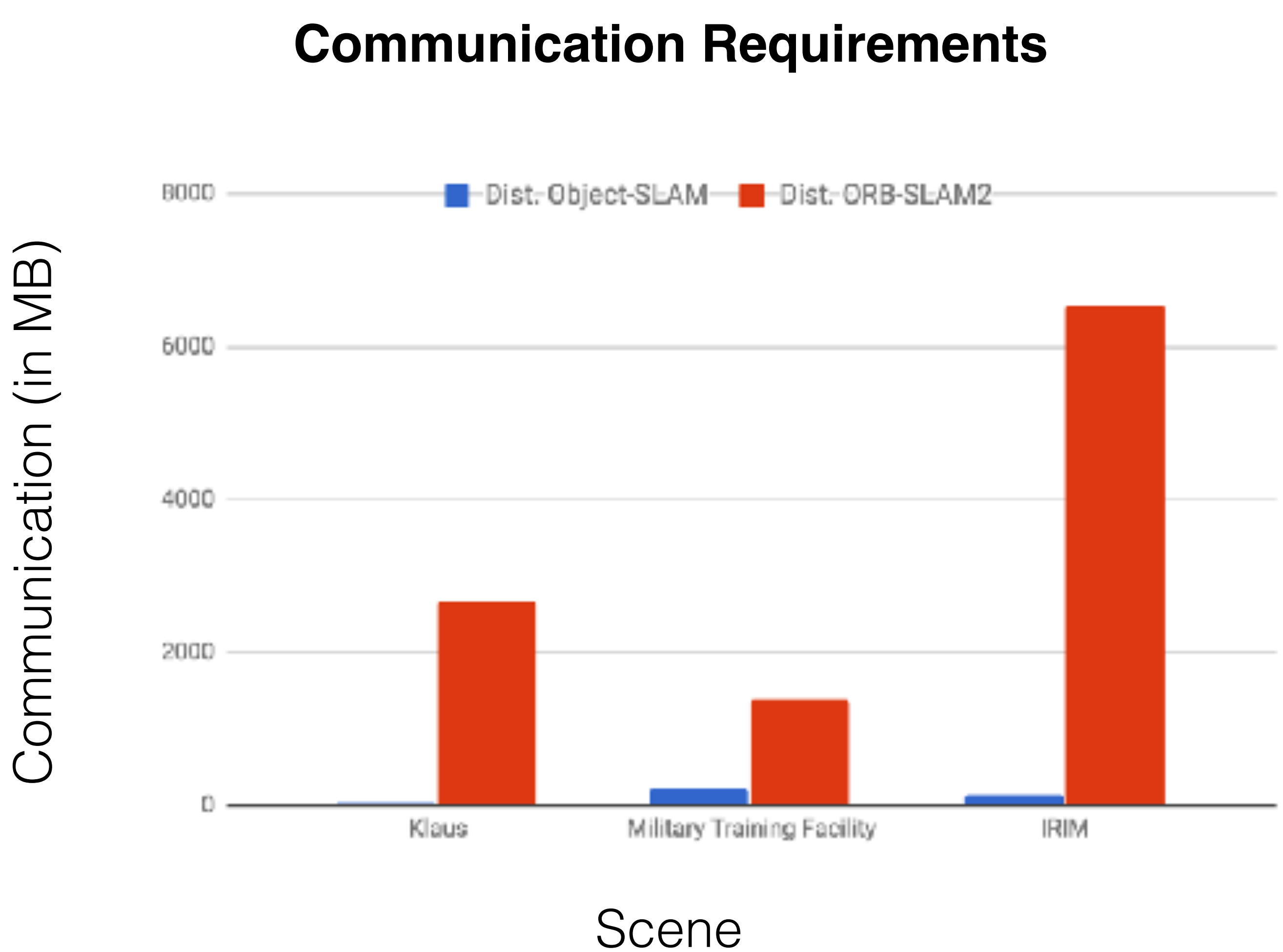
Object detector is fine-tuned on BigBird dataset

Experiments (Large-Scale Robot dataset)



Our approach is nearly as accurate as distributed keyframe based approach.

Experiments (Large-Scale Robot dataset)



Our approach has much less memory and communication requirement than distributed keyframe based approach.

Experiments
(Large-Scale handheld dataset)

Distributed Object-SLAM

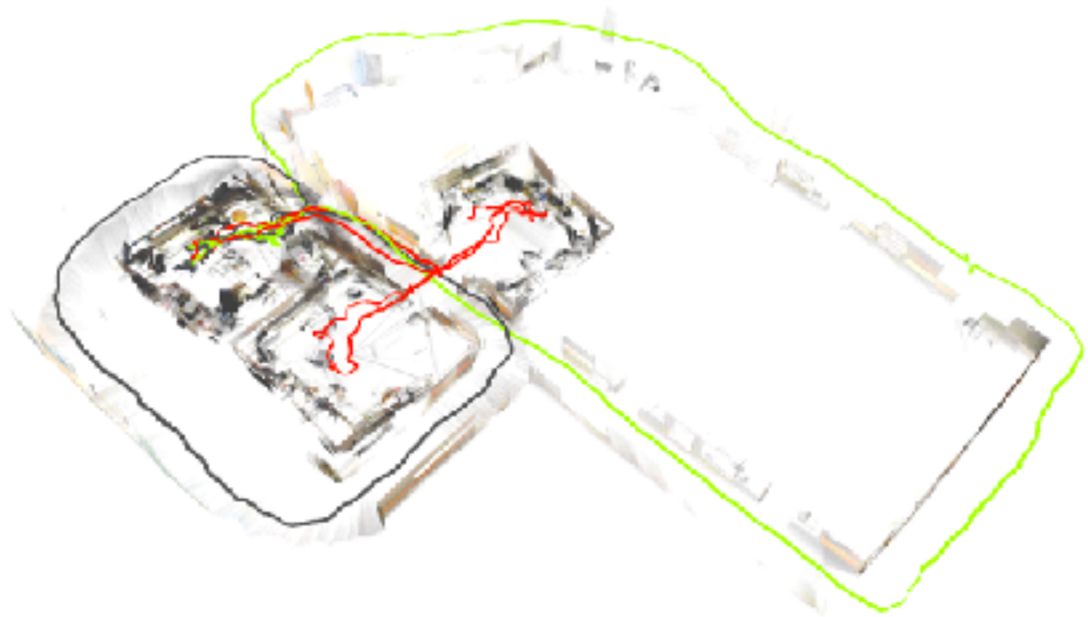
Distributed ORB-SLAM2

Size of the map: **27 m X 20 m**
Number of Poses: **4562**
Distance travelled: **84.48 m**

Pre-trained COCO Object detector is used

Experiments
(Large-Scale handheld dataset)

Distributed Object-SLAM

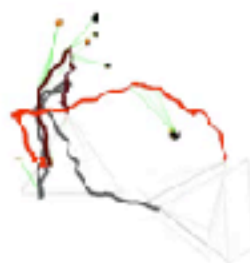


Distributed ORB-SLAM2

Size of the map: **27 m X 20 m**
Number of Poses: **4562**
Distance travelled: **84.48 m**

Pre-trained COCO Object detector is used

Experiments
(Large-Scale handheld dataset)



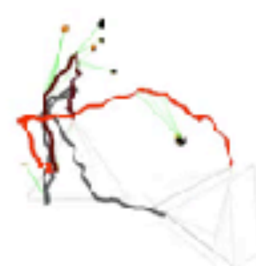
Distributed Object-SLAM

Distributed ORB-SLAM2

Size of the map: **11 m X 5 m**
Number of Poses: **4042**
Distance travelled: **65.53 m**

Pre-trained COCO Object detector is used

Experiments
(Large-Scale handheld dataset)



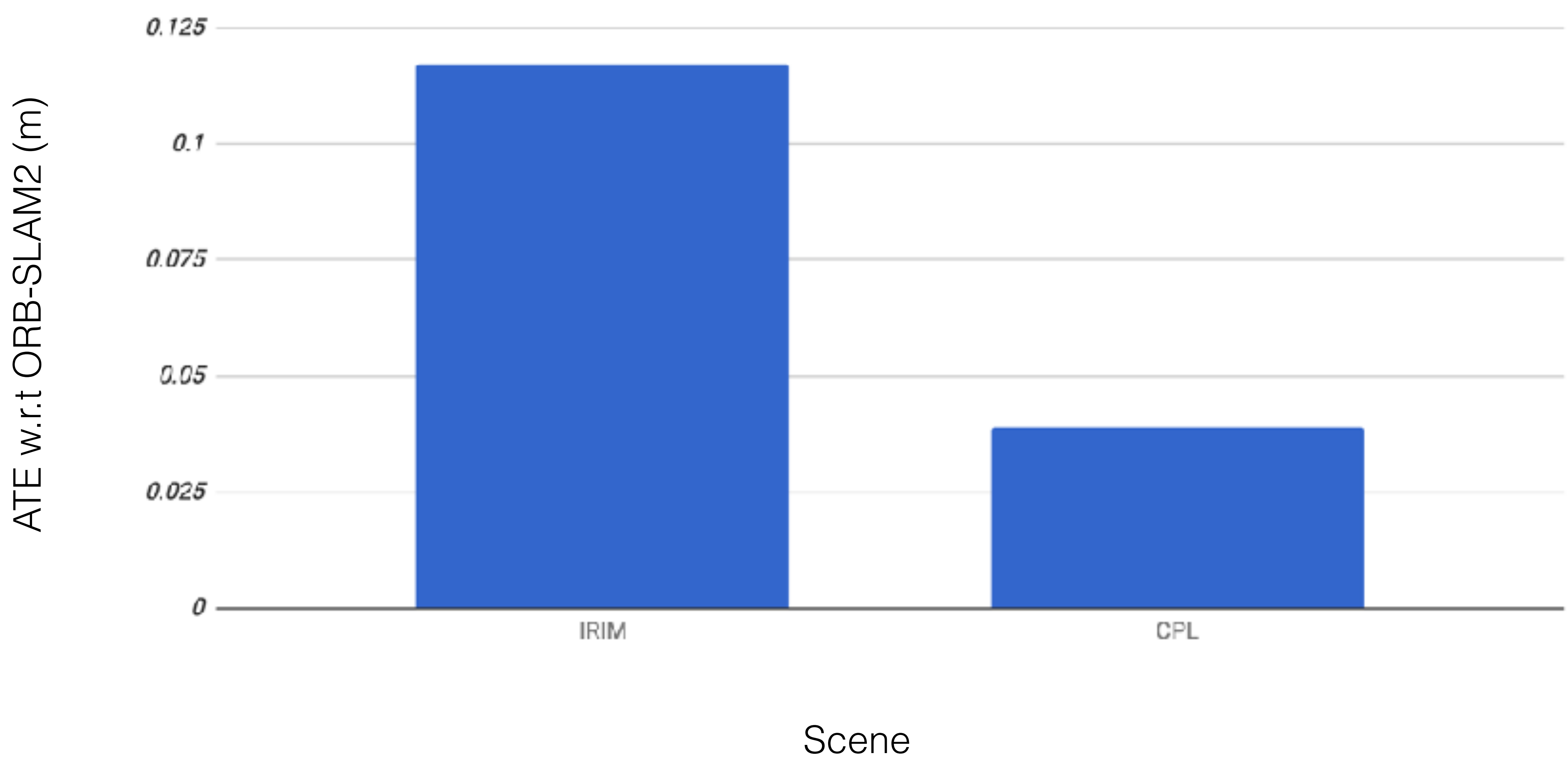
Distributed Object-SLAM

Distributed ORB-SLAM2

Size of the map: **11 m X 5 m**
Number of Poses: **4042**
Distance travelled: **65.53 m**

Pre-trained COCO Object detector is used

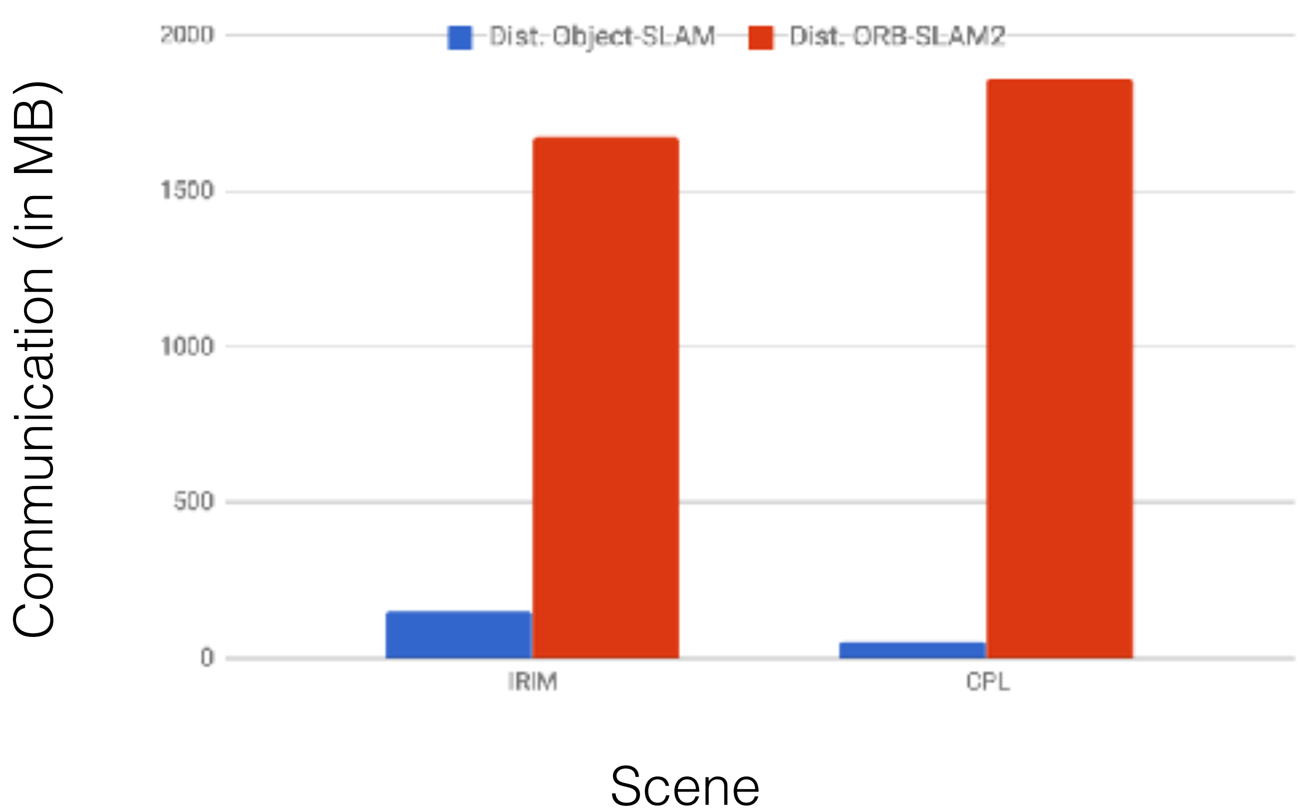
Experiments
(Large-Scale handheld dataset)



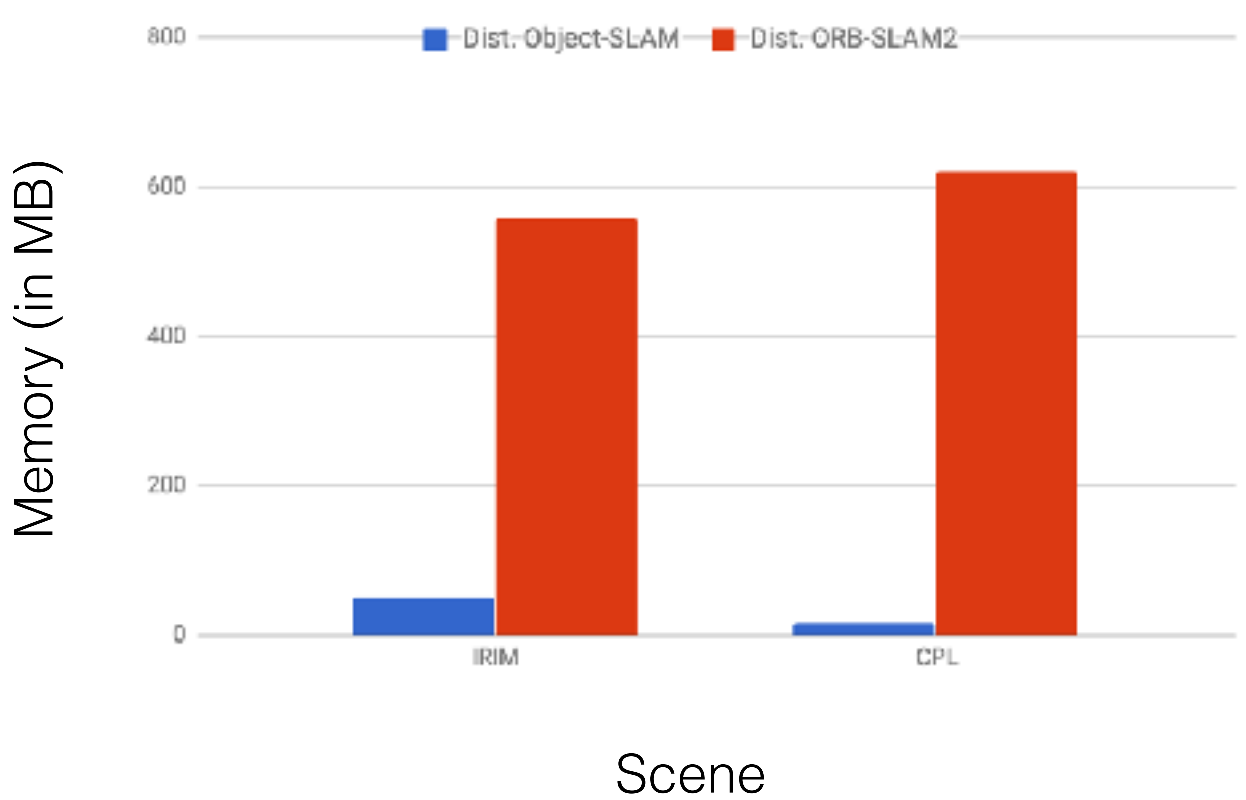
Our approach is nearly as accurate as distributed keyframe based approach.

Experiments
(Large-Scale handheld dataset)

Communication Requirements



Memory Requirements



Our approach has much less memory and communication requirement than distributed keyframe based approach.

Outline

1. Distributed Gauss-Seidel Approach
   (*state of art distributed optimizer*)

2. Distributed Object-based SLAM with Known Object Models
   (*using objects as landmarks*)

3. Distributed Object-based SLAM with Joint Object Modeling and Mapping
   (*jointly model objects along with SLAM*)

4. Conclusions and Future Work

Outline

1. Distributed Gauss-Seidel Approach
   (*state of art distributed optimizer*)

2. Distributed Object-based SLAM with Known Object Models
   (*using objects as landmarks*)

3. Distributed Object-based SLAM with Joint Object Modeling and Mapping
   (*jointly model objects along with SLAM*)

4. Conclusions and Future Work

Contributions

Contributions

- An **object-based distributed algorithm** for cooperative trajectory estimation

Contributions

- An **object-based distributed algorithm** for cooperative trajectory estimation
  - **Reduces** the **memory requirements and information exchange** among robots.

Contributions

- An **object-based distributed algorithm** for cooperative trajectory estimation
    - **Reduces** the **memory requirements and information exchange** among robots.
    - It is as **accurate** as the **centralized estimate**.

Contributions

- An **object-based distributed algorithm** for cooperative trajectory estimation
  - **Reduces** the **memory requirements and information exchange** among robots.
  - It is as **accurate** as the **centralized estimate**.
  - It **scales** well to **large number** of robots and is **resilient to noise**.

Contributions

- An **object-based distributed algorithm** for cooperative trajectory estimation
  - **Reduces** the **memory requirements and information exchange** among robots.
  - It is as **accurate** as the **centralized estimate**.
  - It **scales** well to **large number** of robots and is **resilient to noise**.

Contributions

- An **object-based distributed algorithm** for cooperative trajectory estimation
    - **Reduces** the **memory requirements and information exchange** among robots.
    - It is as **accurate** as the **centralized estimate**.
    - It **scales** well to **large number** of robots and is **resilient to noise**.
    - Extended the approach to the case where **object models** are not previously known and are **jointly optimized** within our SLAM framework.

Publications

Siddharth Choudhary, Luca Carlone, Carlos Nieto, John Rogers, Zhen Liu, Henrik I. Christensen, Frank Dellaert
Multi Robot Object-based SLAM
ISER 2016

Siddharth Choudhary, Luca Carlone, Carlos Nieto, John Rogers, Henrik I. Christensen, Frank Dellaert
Distributed Trajectory Estimation with Privacy and Communication Constraints: a Two-Stage Distributed Gauss-Seidel Approach
ICRA 2016

Siddharth Choudhary, Alexander J.B. Trevor, Henrik I. Christensen, Frank Dellaert
SLAM with Object Discovery, Modeling and Mapping
IROS 2014

Siddharth Choudhary, Luca Carlone, Carlos Nieto, John Rogers, Henrik I. Christensen, Frank Dellaert
Distributed Pose Graph Optimization with Privacy and Communication Constraints: Lightweight Algorithms and Object-based Models
IJRR 2017 (accepted)

https://cognitiverobotics.github.io/distributed-mapper/

Publications

Siddharth Choudhary, Luca Carlone, Carlos Nieto, John Rogers, Zhen Liu, Henrik I. Christensen, Frank Dellaert
Multi Robot Object-based SLAM
ISER 2016

Siddharth Choudhary, Luca Carlone, Carlos Nieto, John Rogers, Henrik I. Christensen, Frank Dellaert
Distributed Trajectory Estimation with Privacy and Communication Constraints: a Two-Stage Distributed Gauss-Seidel Approach
ICRA 2016

Siddharth Choudhary, Luca Carlone, Henrik I. Christensen, Frank Dellaert
Exactly Sparse Memory Efficient SLAM using the Multi-Block Alternating Direction Method of Multipliers
IROS 2015

Siddharth Choudhary, Vadim Indelman, Henrik I. Christensen, Frank Dellaert
Information based Reduced Landmark SLAM
ICRA 2015

Siddharth Choudhary, Alexander J.B. Trevor, Henrik I. Christensen, Frank Dellaert
SLAM with Object Discovery, Modeling and Mapping
IROS 2014

Siddharth Choudhary, Luca Carlone, Carlos Nieto, John Rogers, Henrik I. Christensen, Frank Dellaert
Distributed Pose Graph Optimization with Privacy and Communication Constraints: Lightweight Algorithms and Object-based Models
IJRR 2017 (accepted)

Distributed Outlier Rejection (Future Work)

Investigate **distributed implementation of outlier rejection methods***, so that they can be applied in a multi robot system **without** requiring all robots to **exchange all measurements**.

Distributed Outlier Rejection (Future Work)

Investigate **distributed implementation of outlier rejection methods***, so that they can be applied in a multi robot system **without** requiring all robots to **exchange all measurements**.

******Selecting good measurements via l1 relaxation: a convex approach for robust estimation over graphs, Carlone et al. (IROS 2014)*

Multi Robot Exploration and Mapping (Future Work)

Autonomous exploration and mapping using a heterogeneous team of robots.

Multi Robot Exploration and Mapping (Future Work)

Autonomous exploration and mapping using a heterogeneous team of robots.

Contributions

- An **object-based distributed algorithm** for cooperative trajectory estimation
    - **Reduces** the **memory requirements and information exchange** among robots.
    - It is as **accurate** as the **centralized estimate**.
    - It **scales** well to **large number** of robots and is **resilient to noise.**
    - Extended the approach to the case where **object models** are not previously known and are **jointly optimized** within our SLAM framework.

Thank you!

Contributions

- An **object-based distributed algorithm** for cooperative trajectory estimation
  - **Reduces** the **memory requirements and information exchange** among robots.
  - It is as **accurate** as the **centralized estimate**.
  - It **scales** well to **large number** of robots and is **resilient to noise.**
  - Extended the approach to the case where **object models** are not previously known and are **jointly optimized** within our SLAM framework.

Thank you!