

Exploration and Implementation of Camera and LiDAR-based SLAM Algorithms

Austin Zhang, Bryce Bernstein, Henry Burkhardt,
Ngelek Thayai, and Raymond Kim

Autonomous systems need to travel the world on their own.

Autonomous systems need to travel the world on their own.

What does the world look like?

Where am I?

Autonomous systems need to travel the world on their own

Simultaneous

Localization

And

Mapping

Autonomous systems need to travel the world on their own

Simultaneous

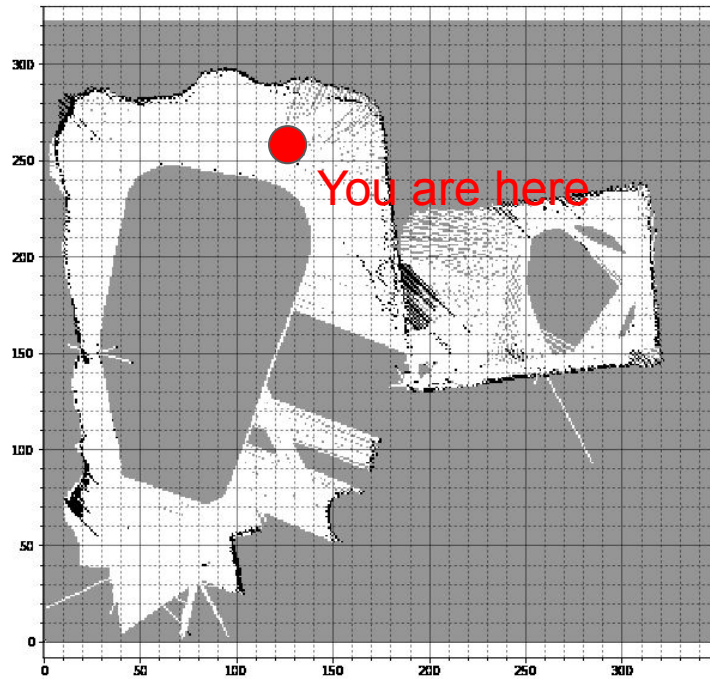
Localization

Where am I?

And

Mapping

**What does the world
look like?**



We built two map-making systems using:

LiDAR range finders (Henry and Raymond)



Cameras (Austin, Bryce & Ngelek)



Road map



Overview

SLAM Introduction

Pose-Graph SLAM

(with 2D LiDAR)

Orb SLAM

(Oriented FAST and Rotated
BRIEF)

Conclusion

SLAM

Pose-Graph SLAM (using 2D LiDAR)

some vocabulary...

Common concepts

Pose

$$(x, y, \theta)$$

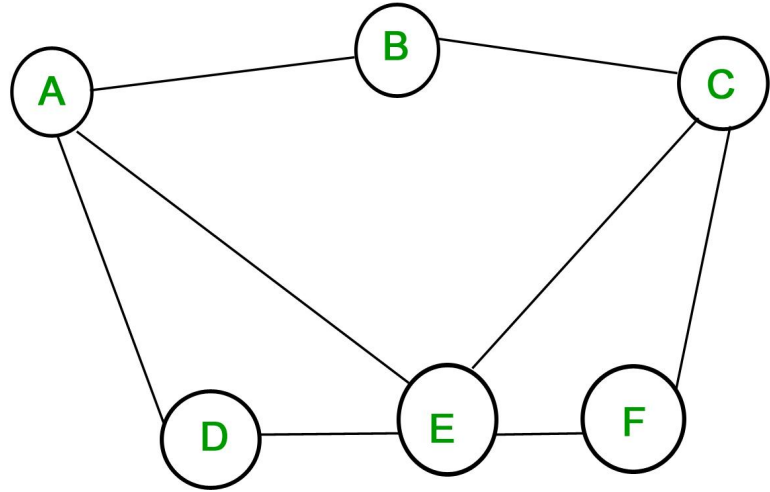
or

$$(x, y, z, \theta, \phi)$$

A robot's location and orientation in space.

Common concepts

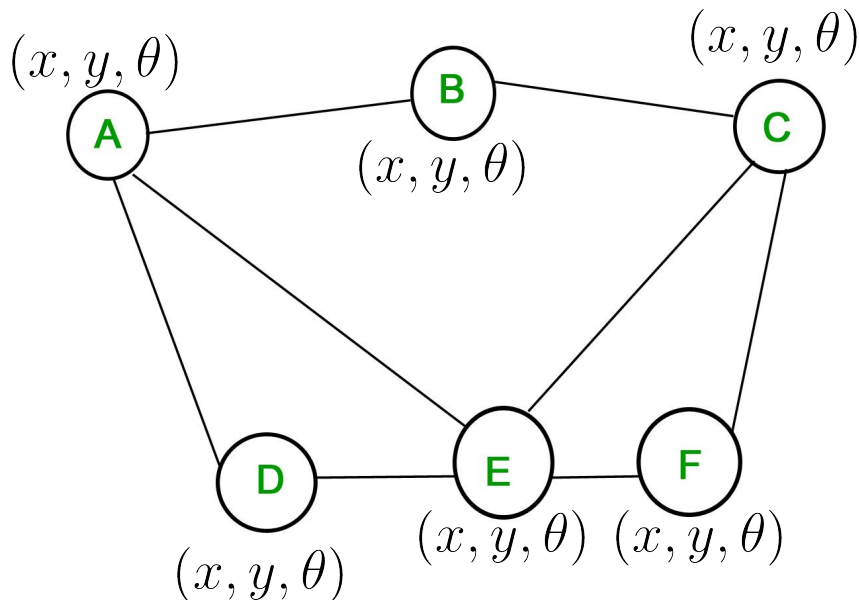
Graph



Representation using *nodes*, and *edges*.

Common concepts

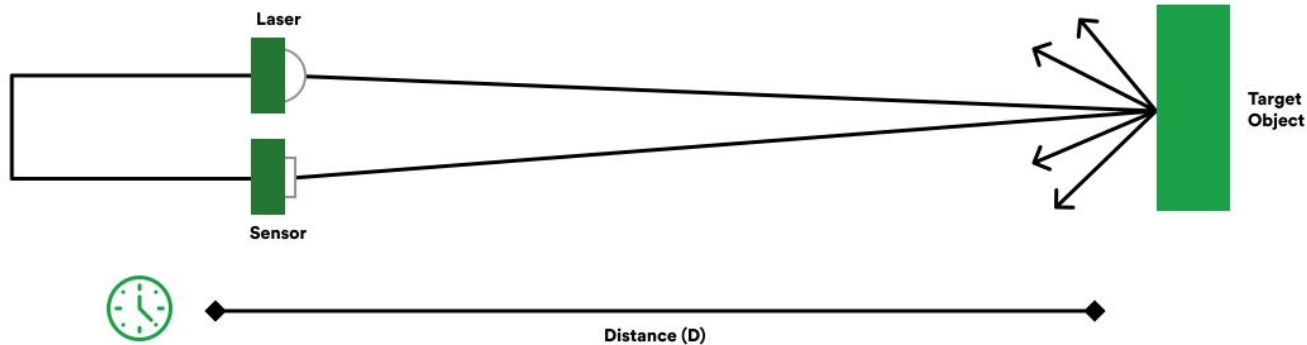
Pose-Graph



Yeah...you can guess

Common concepts

LiDAR = “Light Detection and Ranging”



Common concepts

LiDAR = “Light Detection and Ranging”



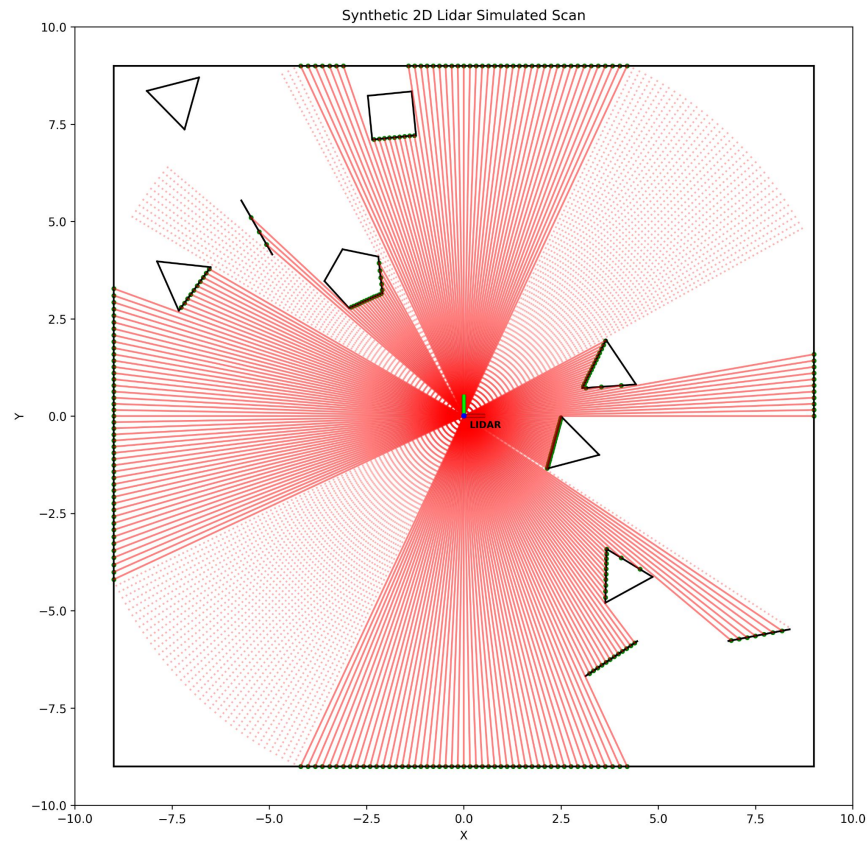
Common concepts

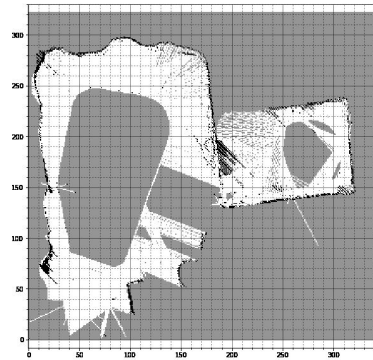
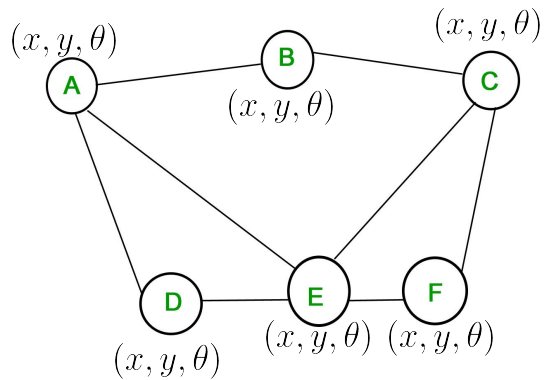
LiDAR = “Light Detection and Ranging”



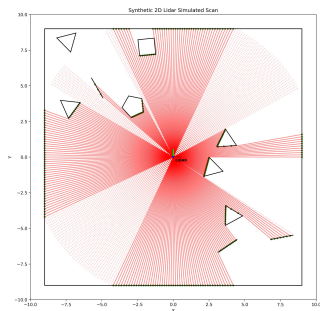
Common concepts

LiDAR

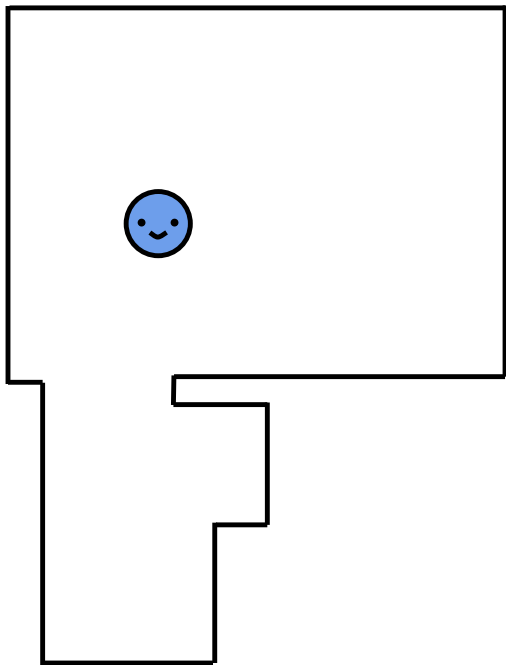




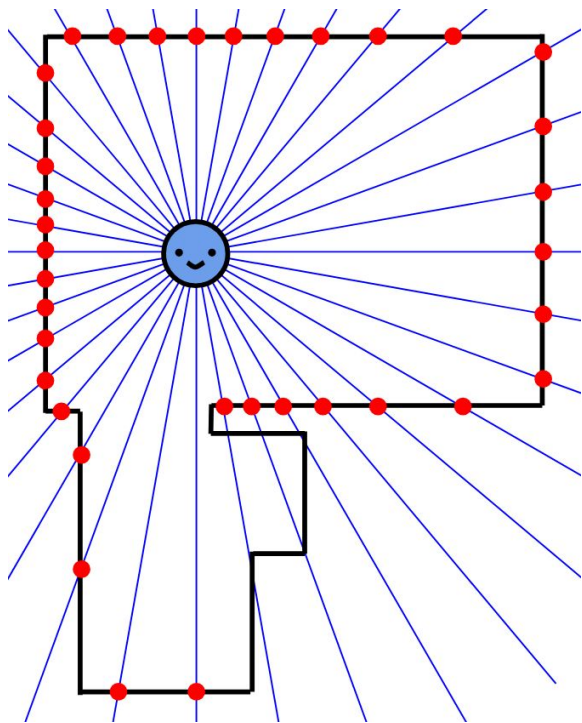
Pose-Graph SLAM (using 2D LiDAR)



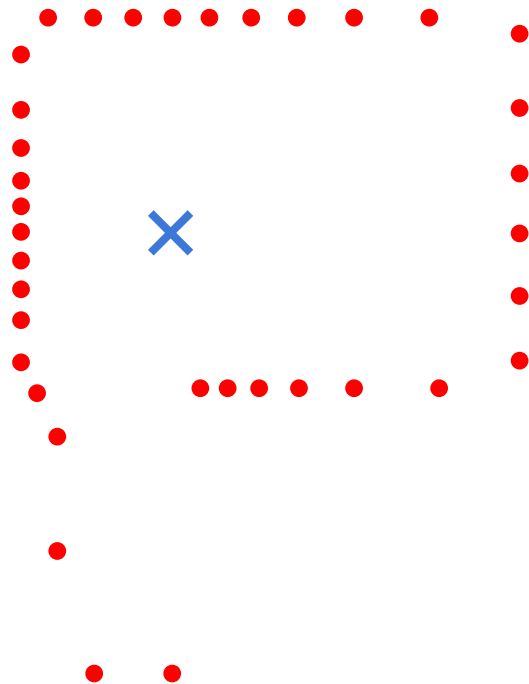
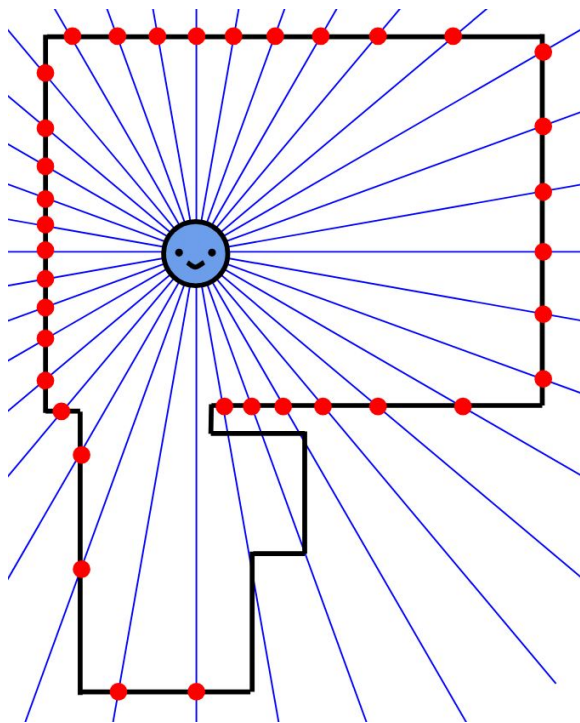
LiDAR SLAM



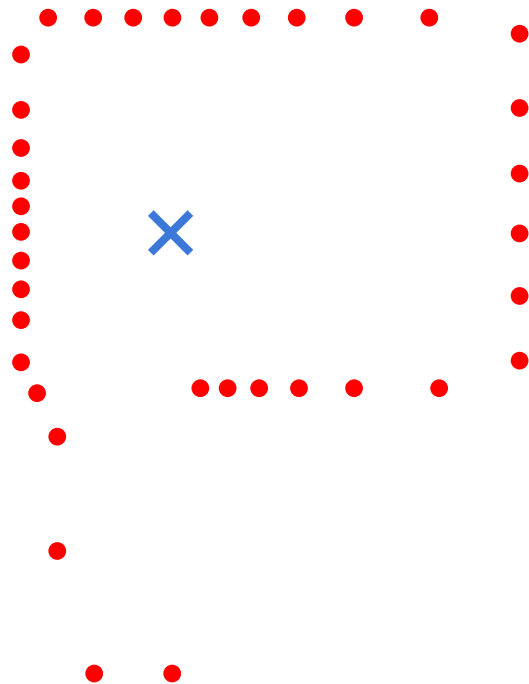
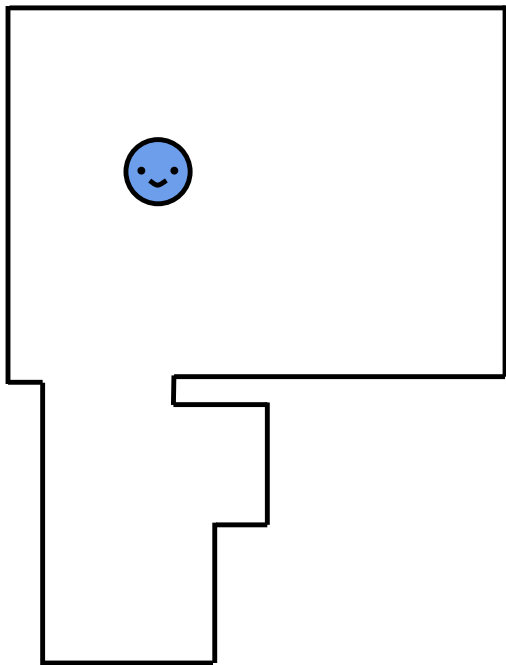
LiDAR SLAM



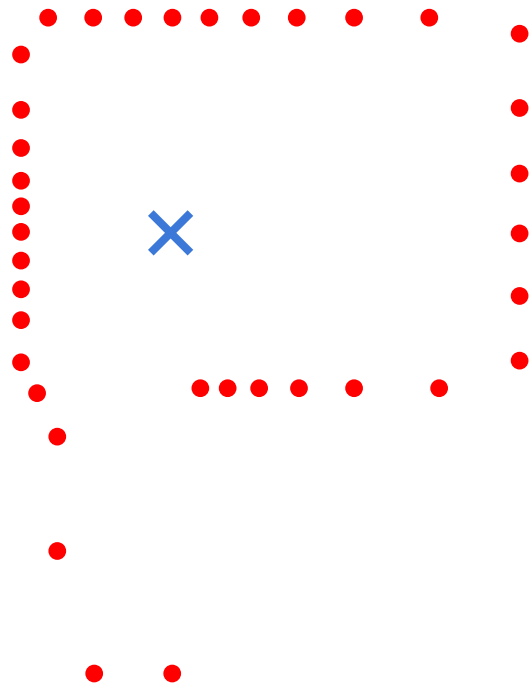
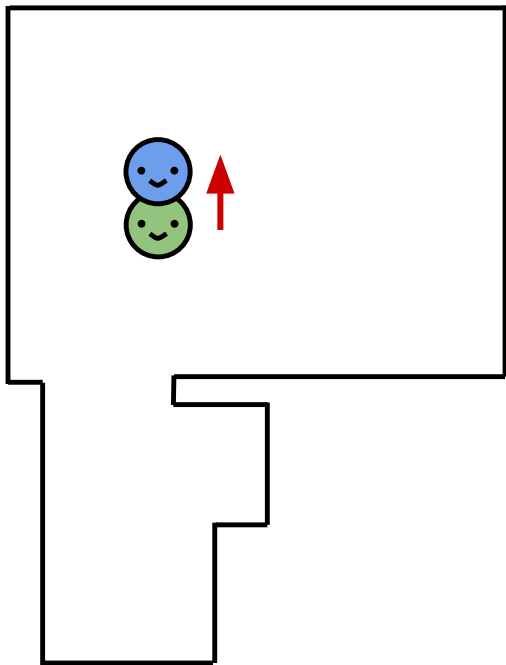
LiDAR SLAM



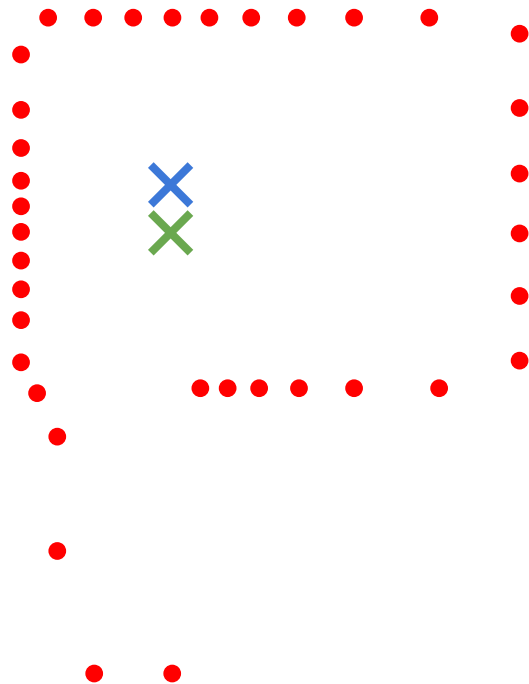
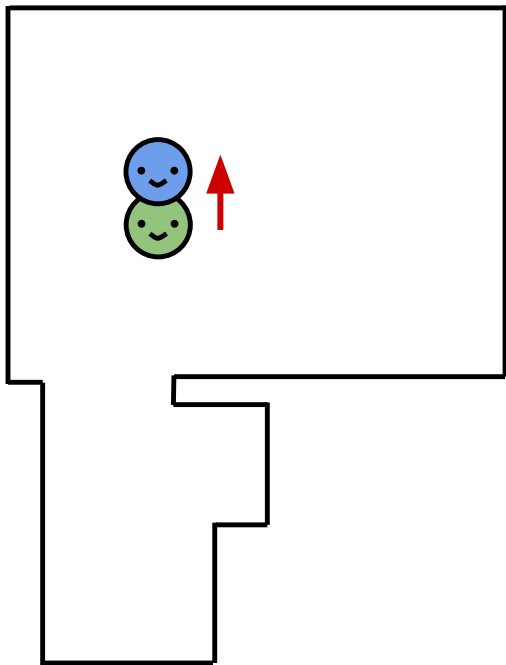
LiDAR SLAM



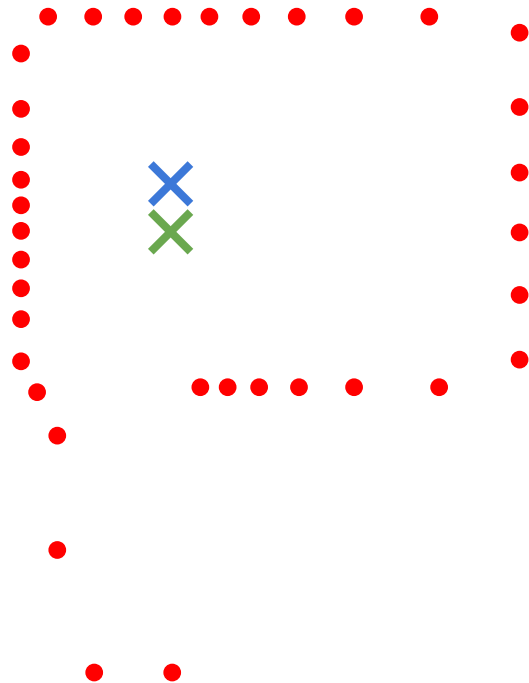
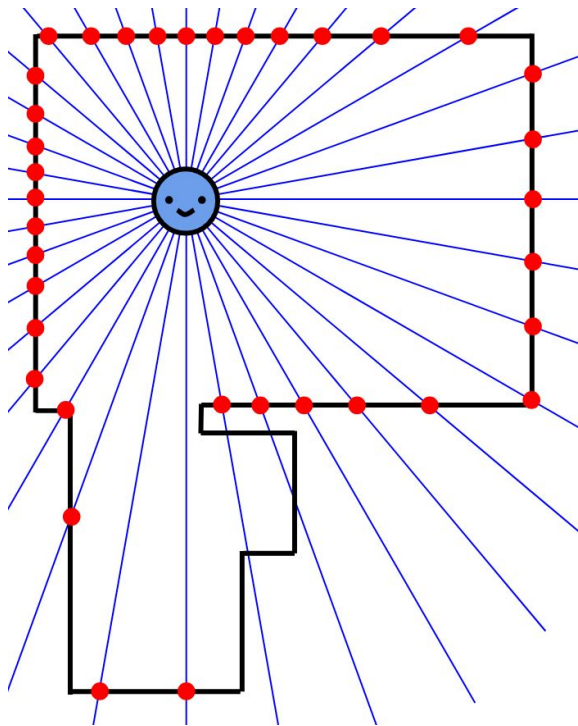
LiDAR SLAM



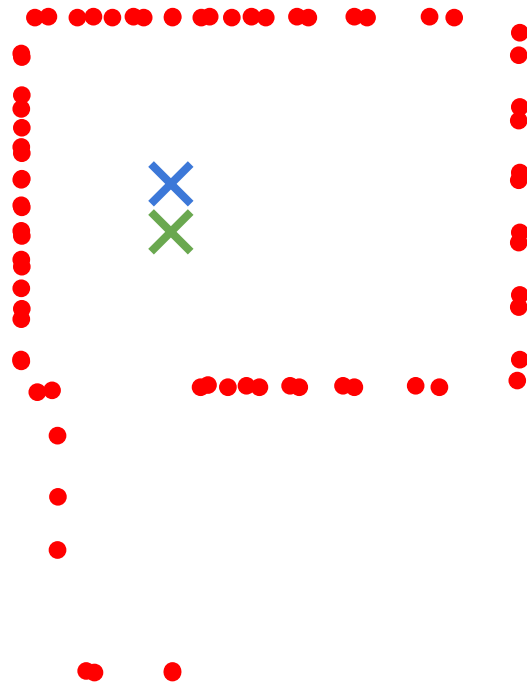
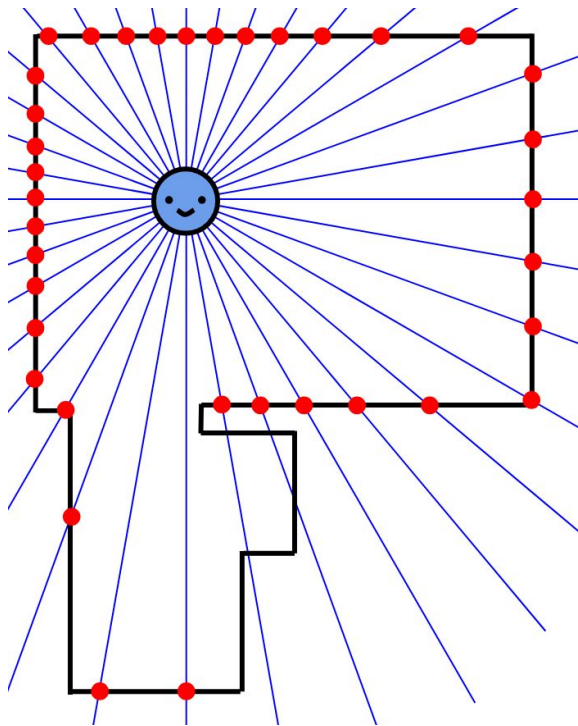
LiDAR SLAM



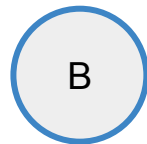
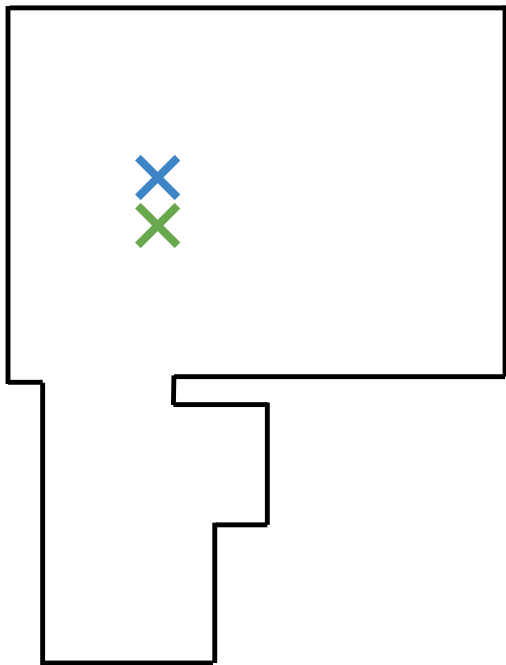
LiDAR SLAM



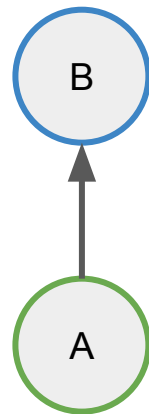
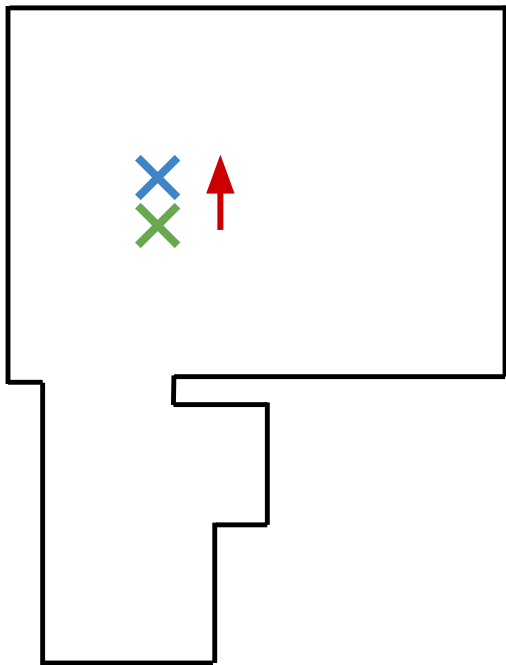
LiDAR SLAM



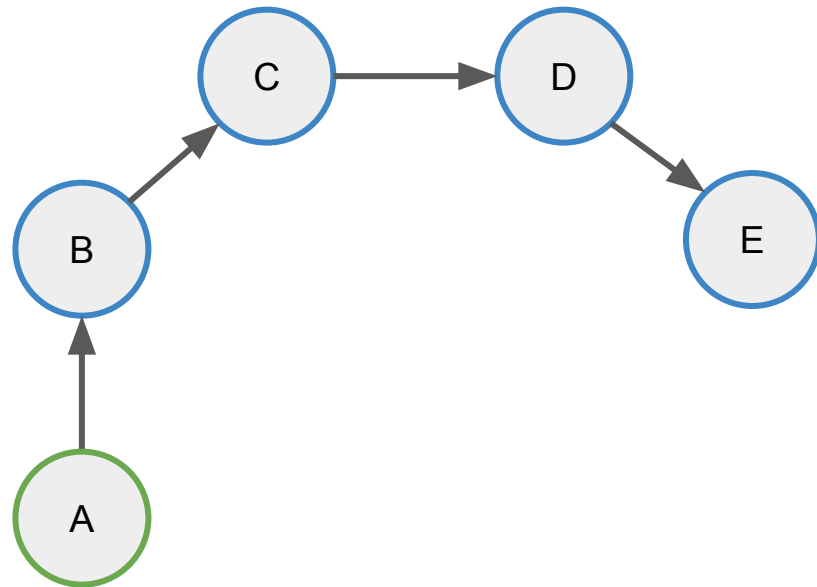
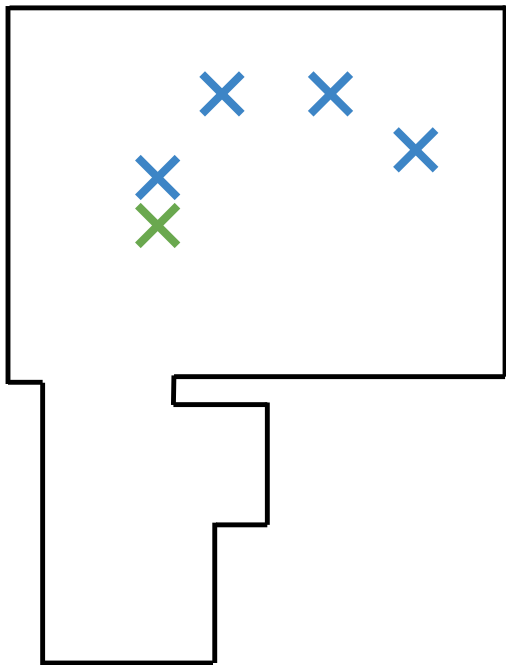
LiDAR SLAM

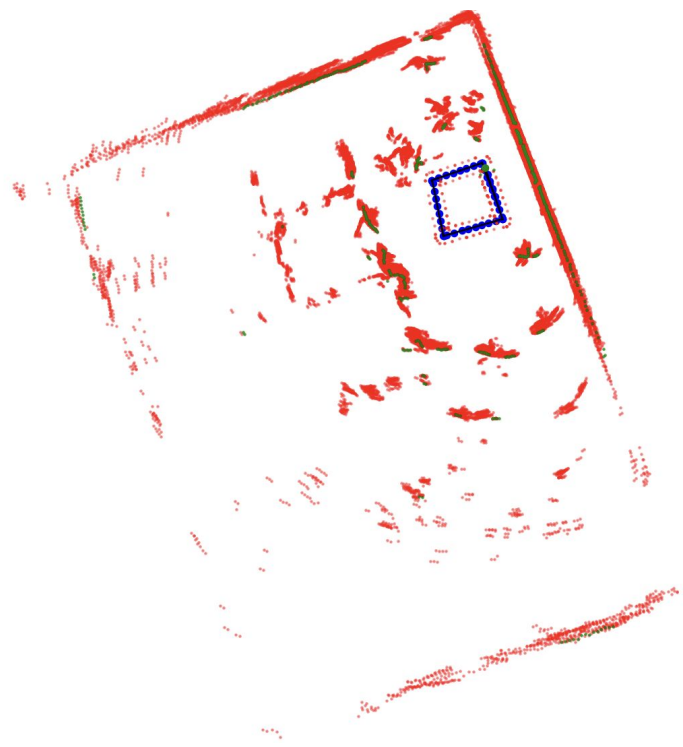


LiDAR SLAM

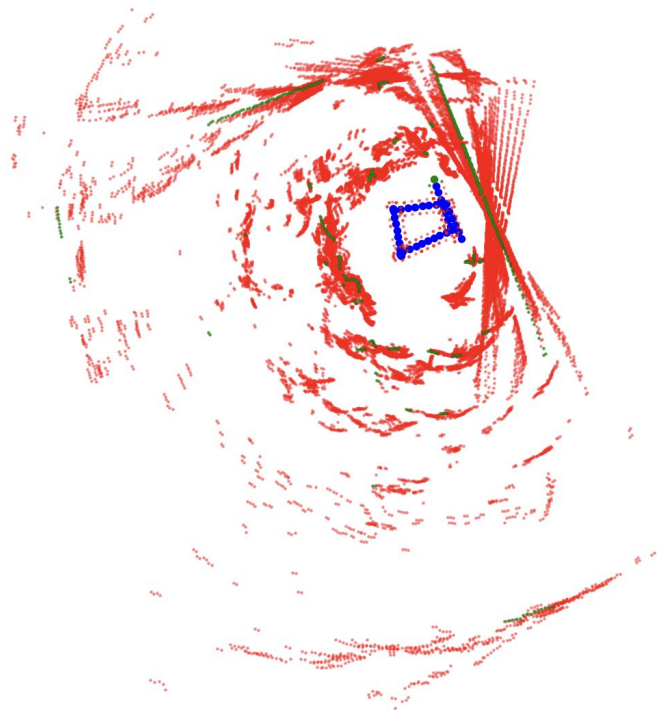
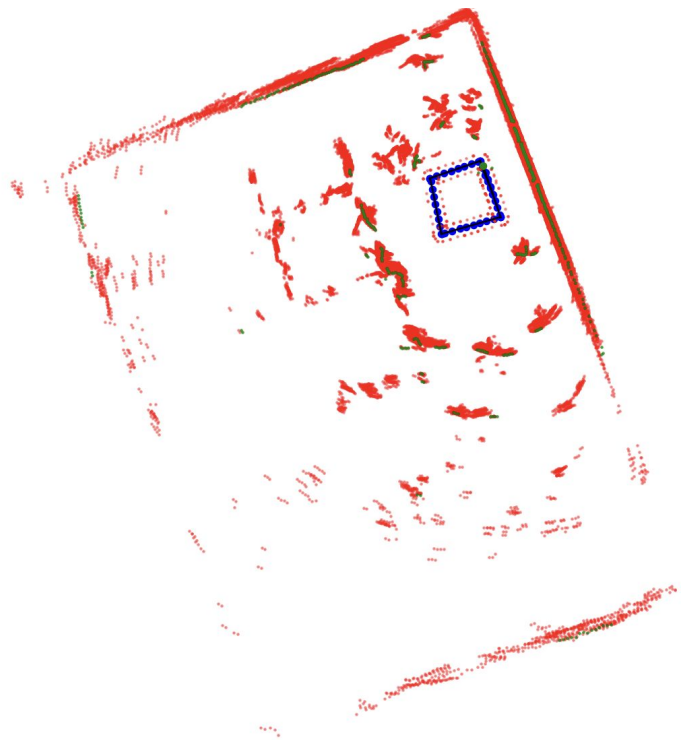


LiDAR SLAM



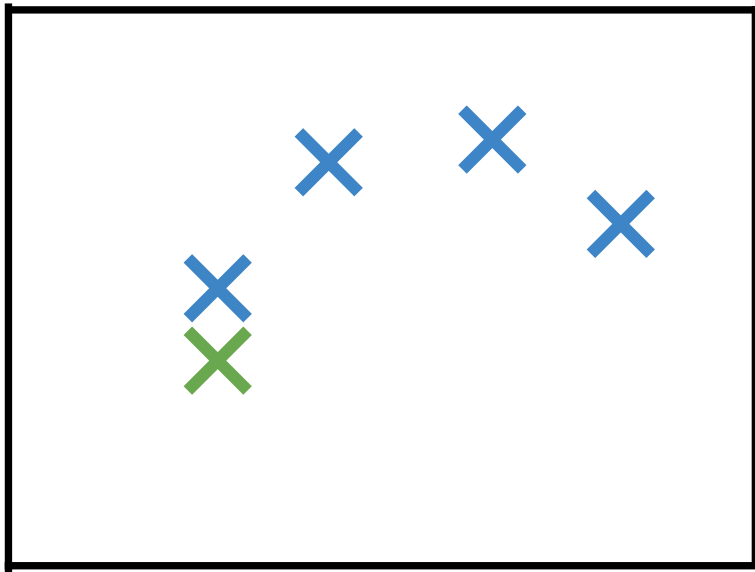


Drift!

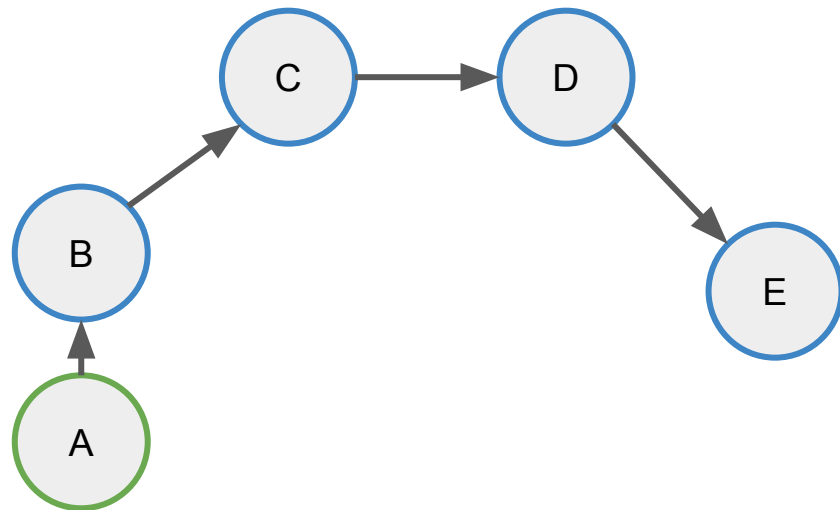


Loop closure

Real world

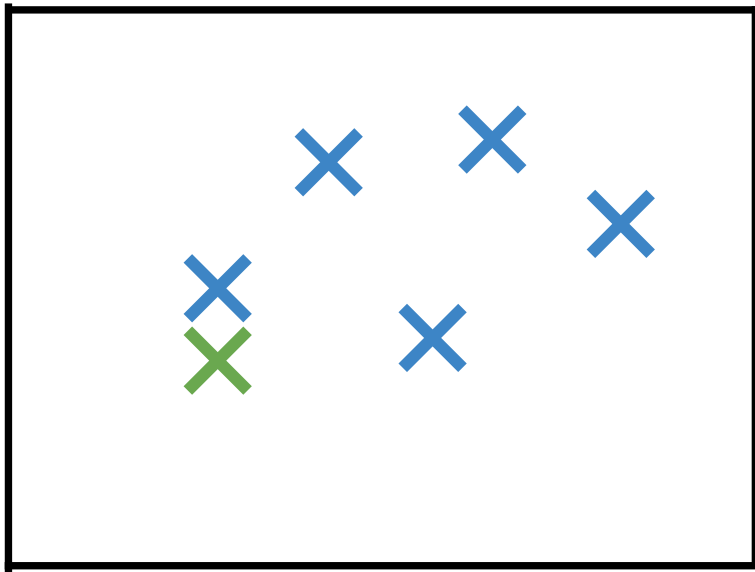


Robot's brain

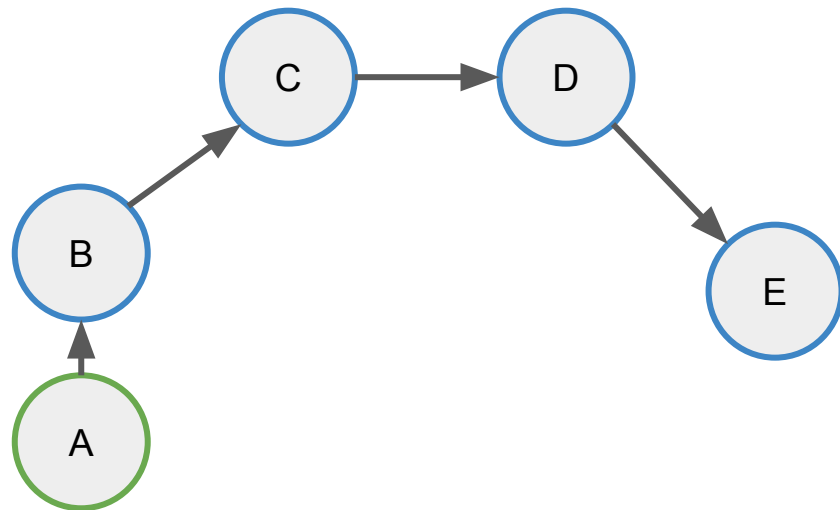


Loop closure

Real world

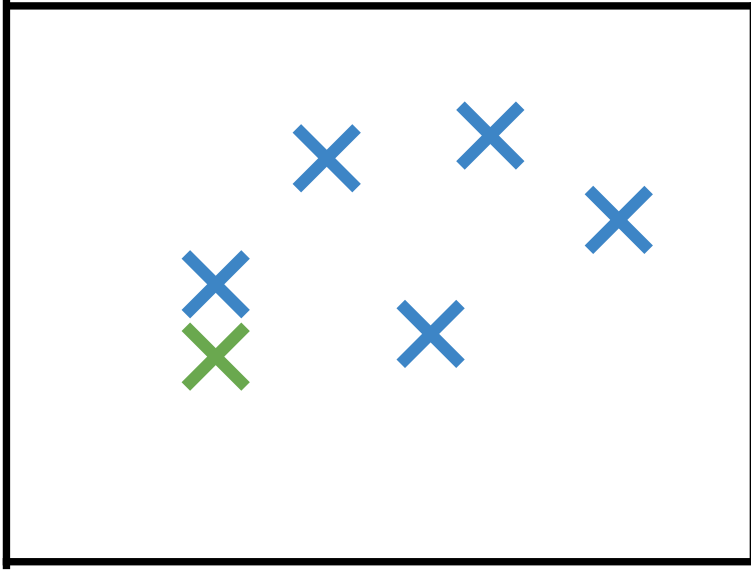


Robot's brain

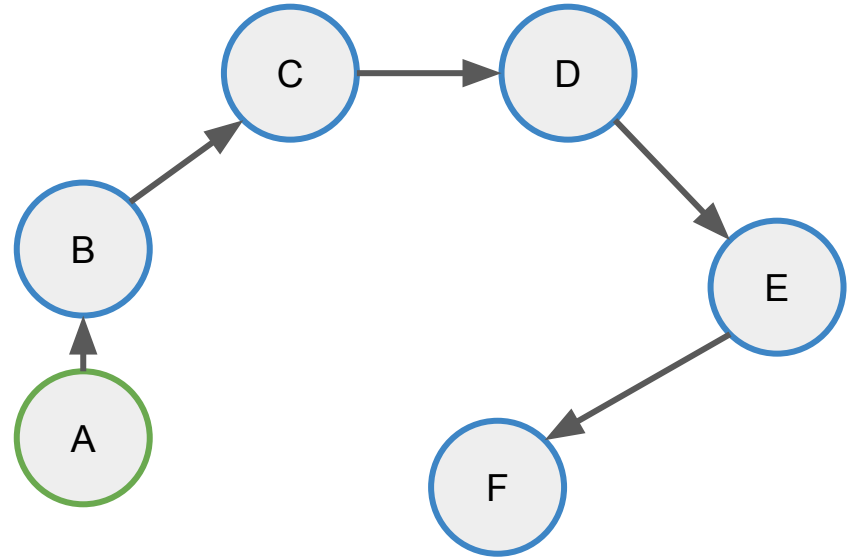


Loop closure

Real world

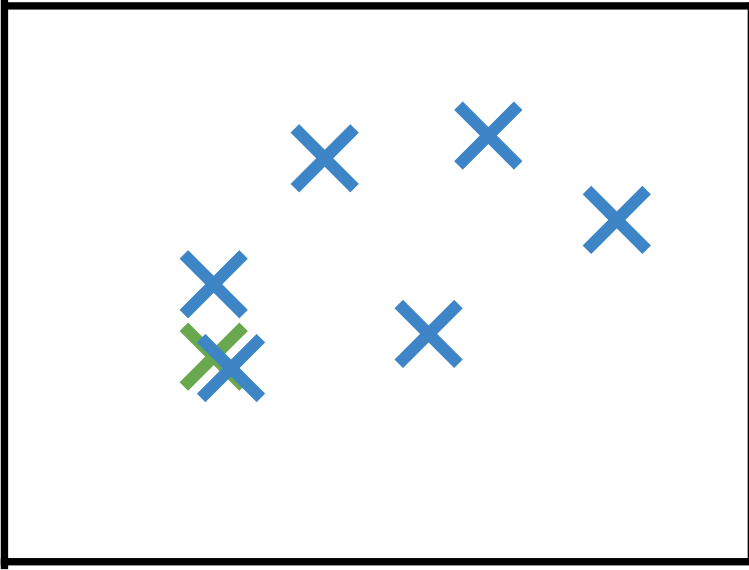


Robot's brain

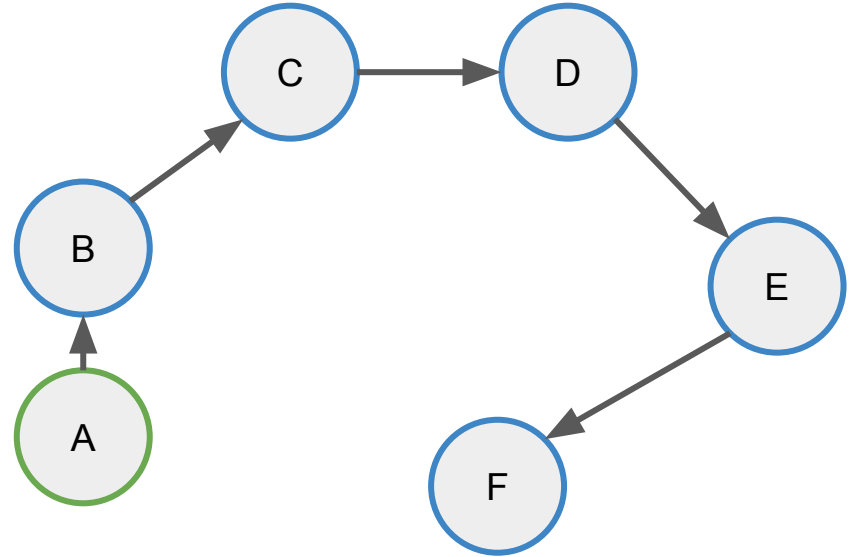


Loop closure

Real world

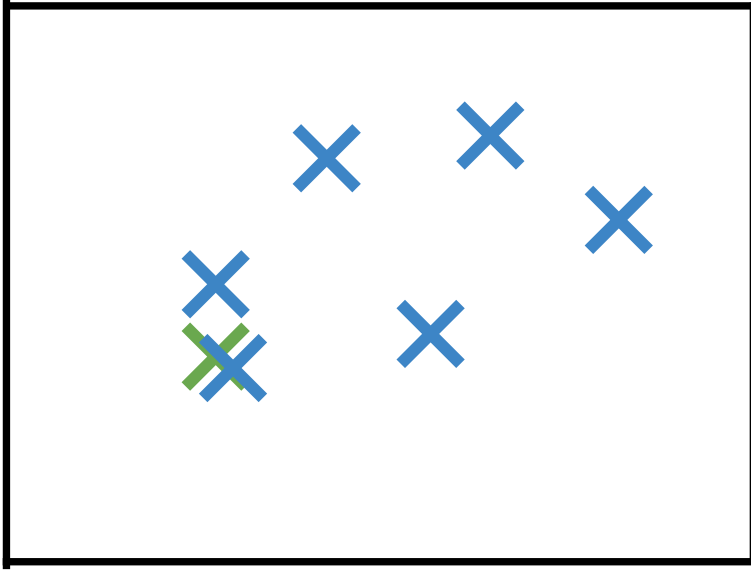


Robot's brain

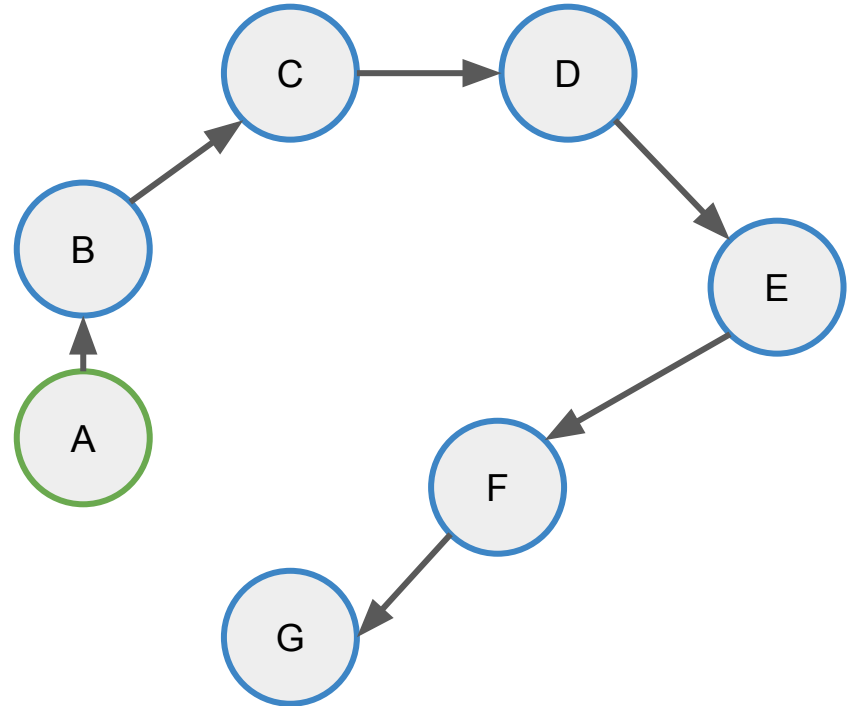


Loop closure

Real world

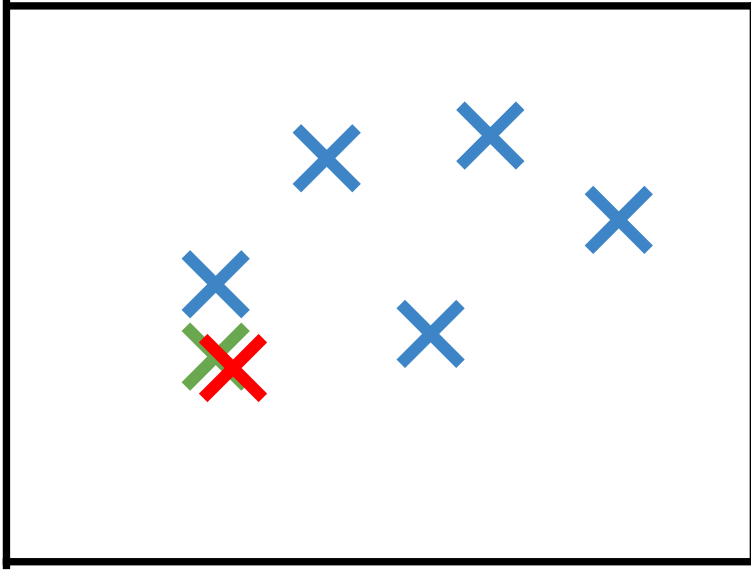


Robot's brain

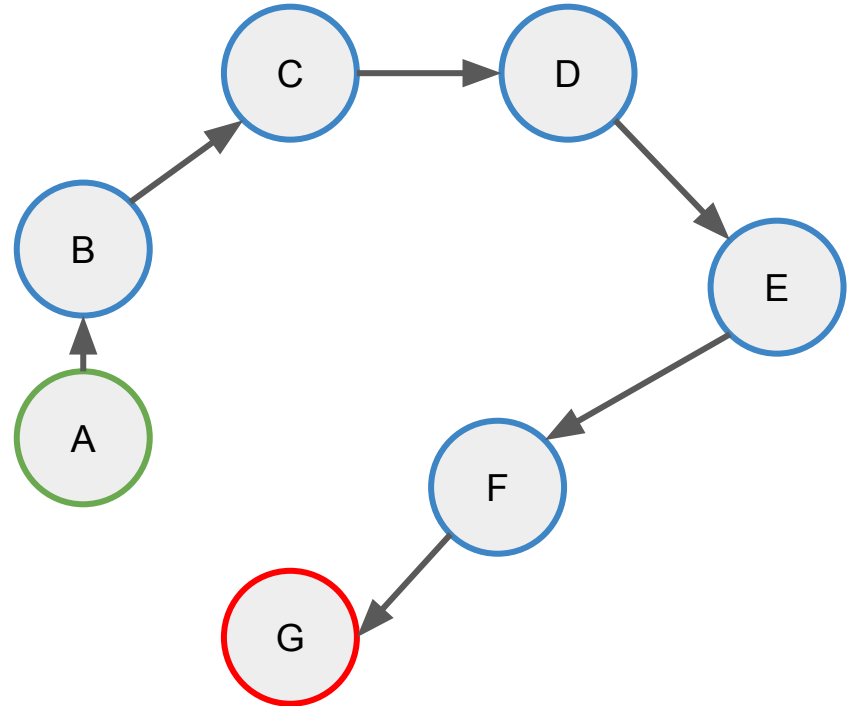


Loop closure

Real world

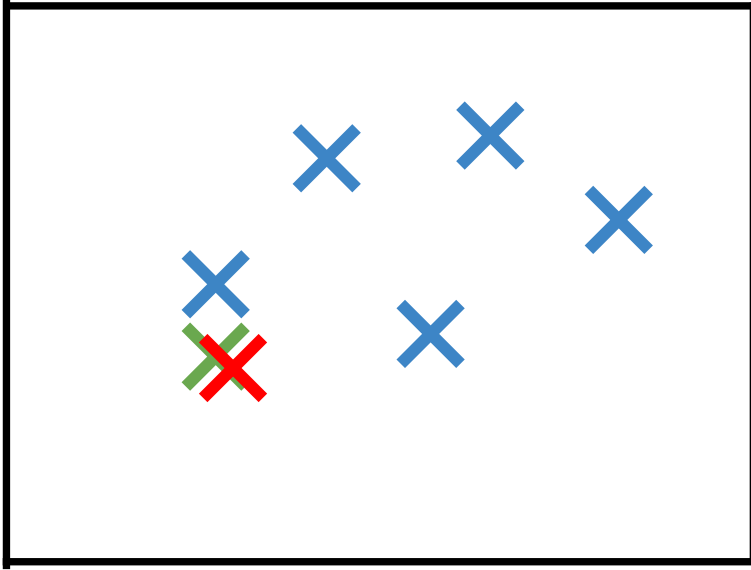


Robot's brain

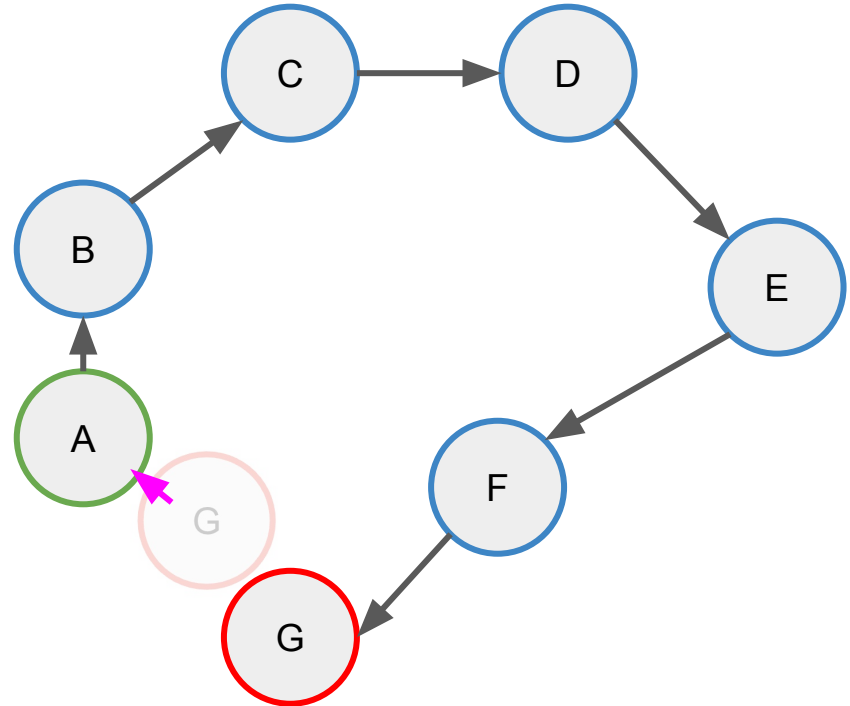


Loop closure

Real world

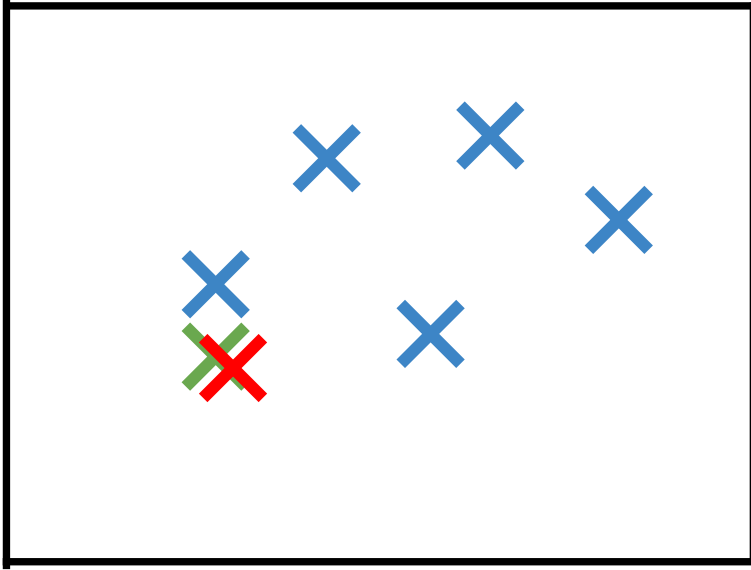


Robot's brain

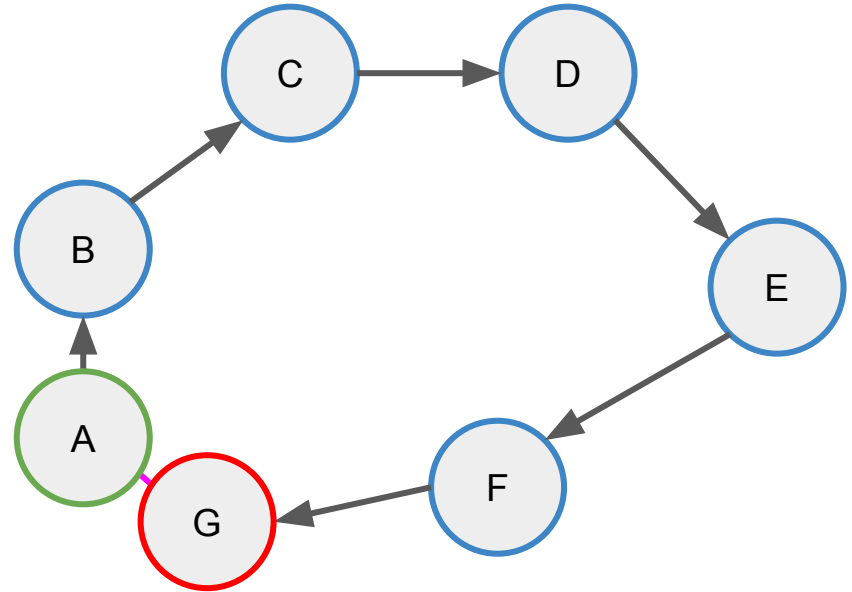


Loop closure

Real world

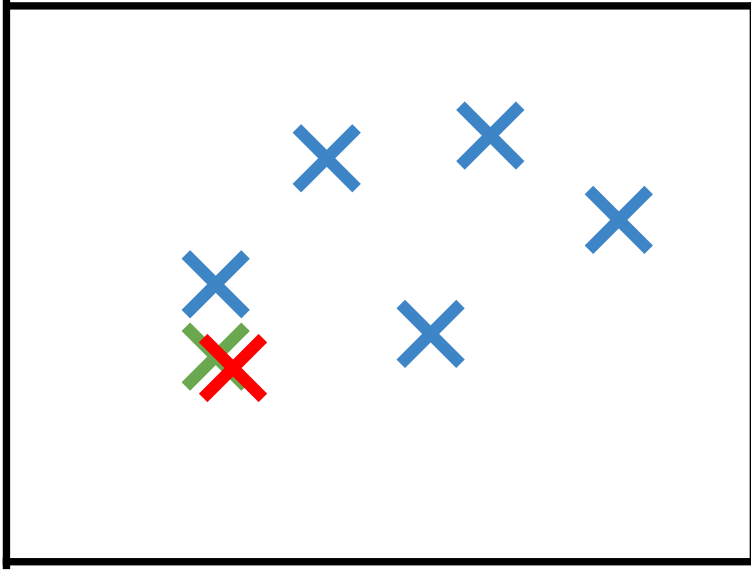


Robot's brain

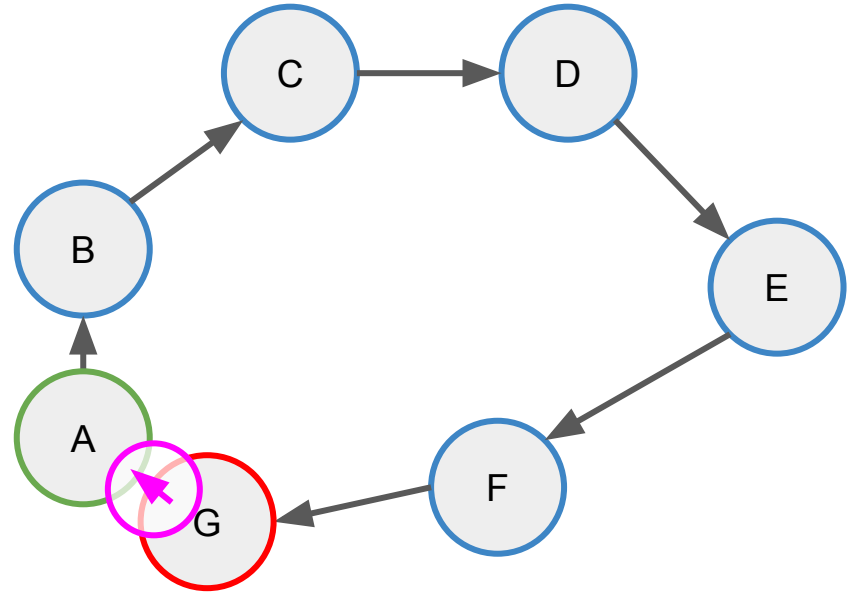


Loop closure

Real world

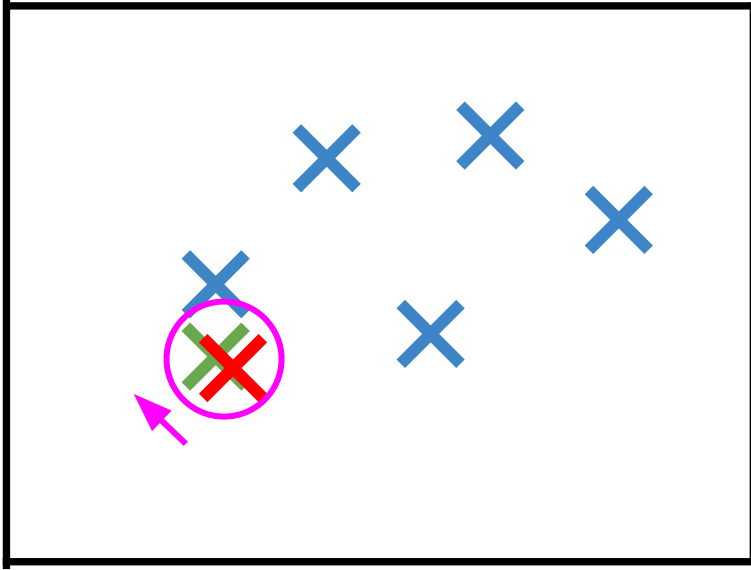


Robot's brain

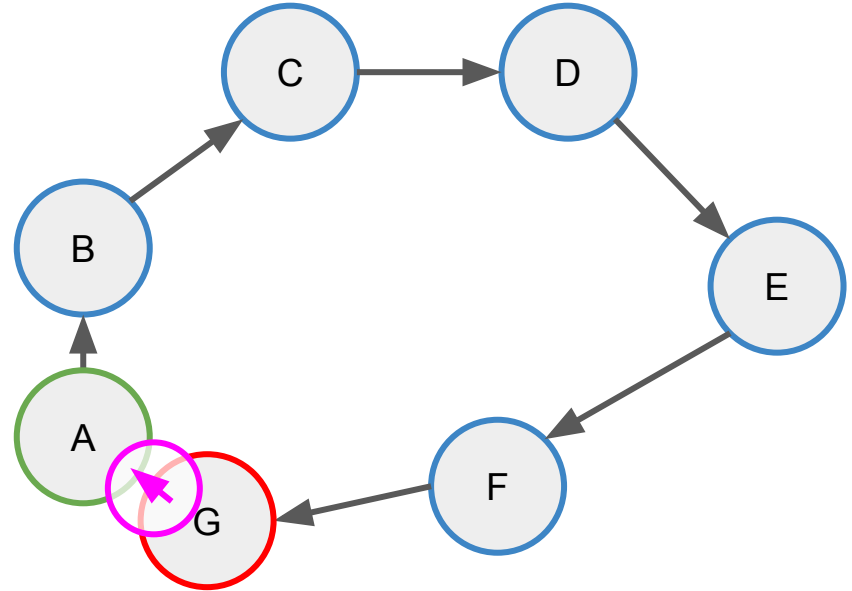


Loop closure

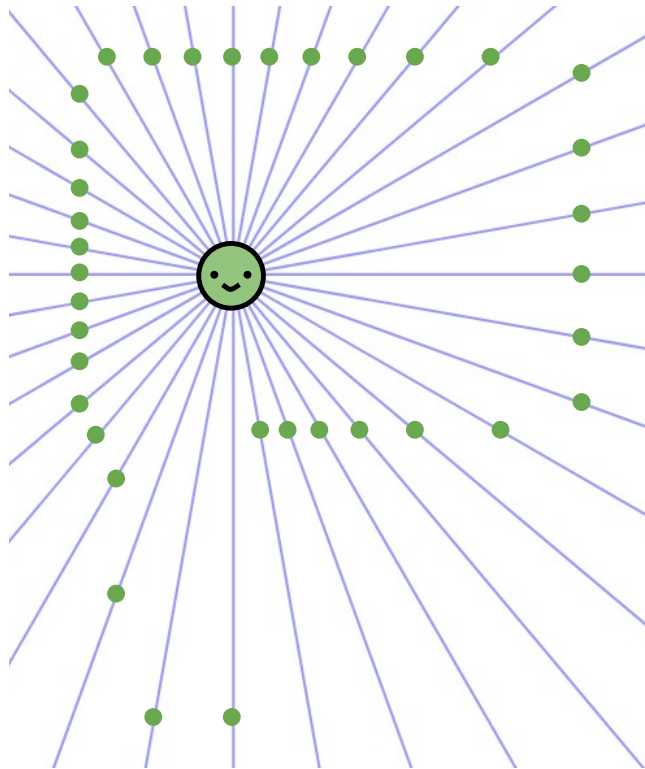
Real world



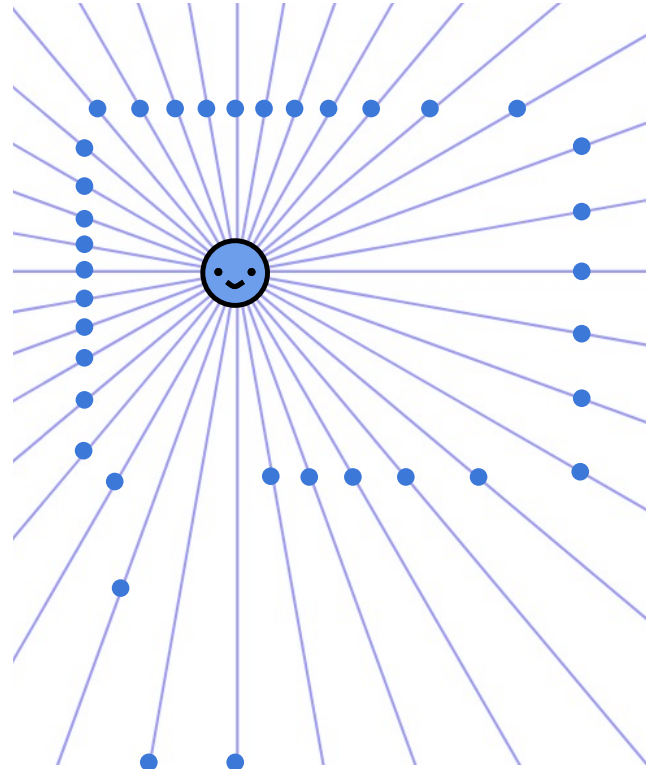
Robot's brain

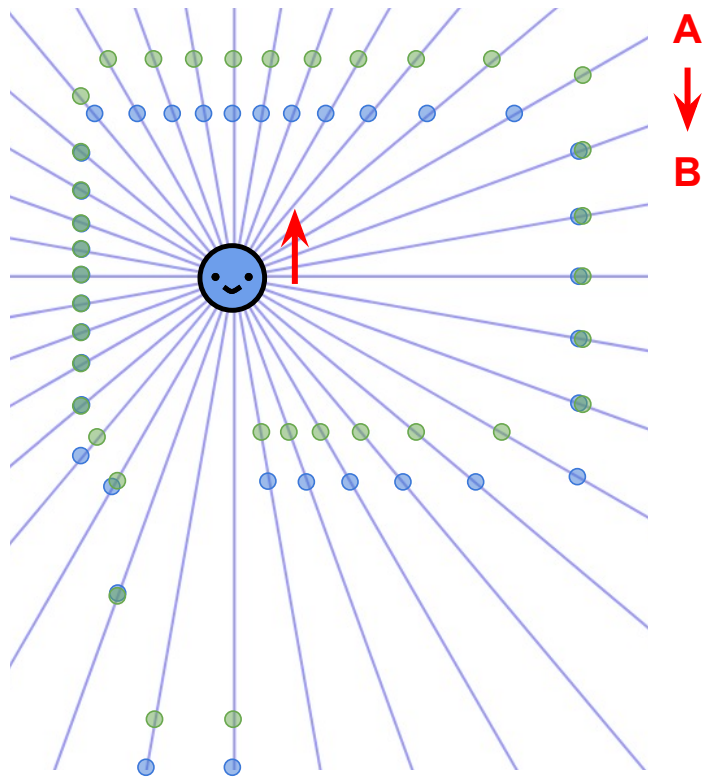


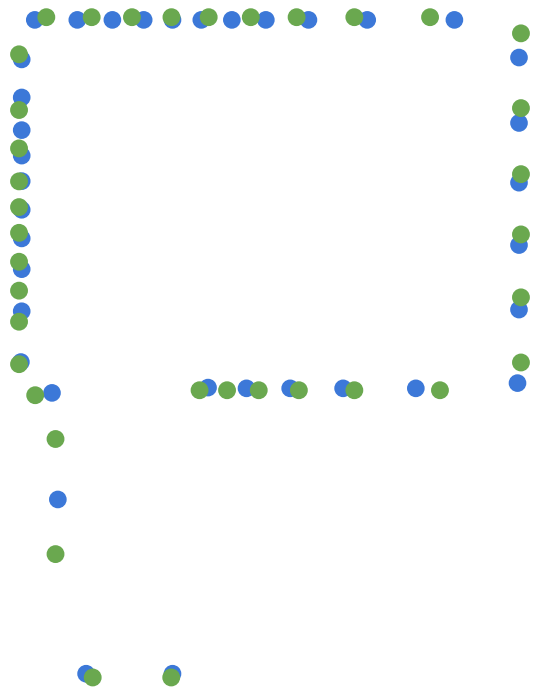
A



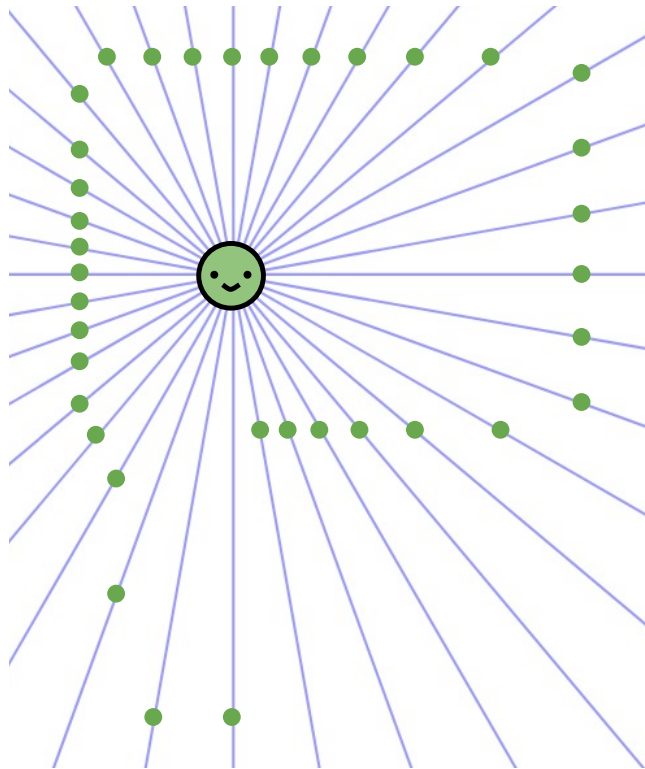
B



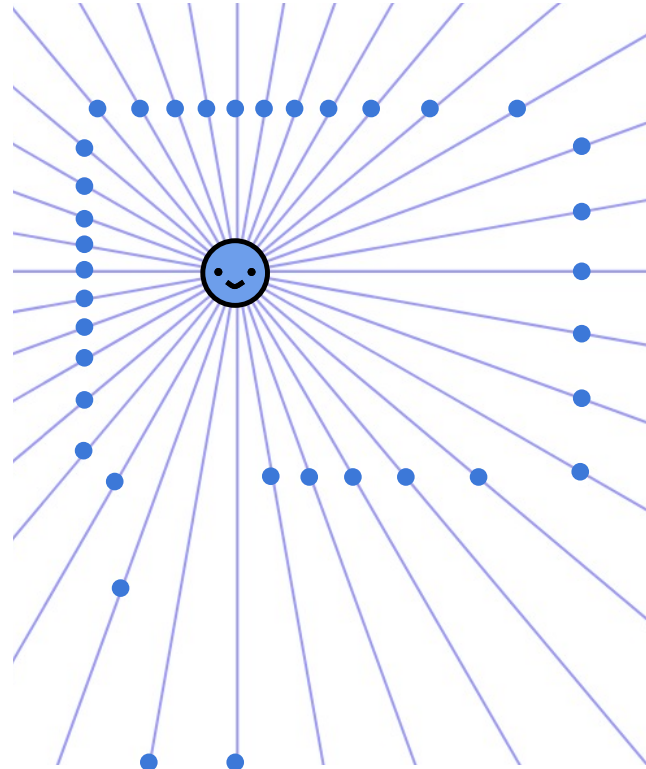




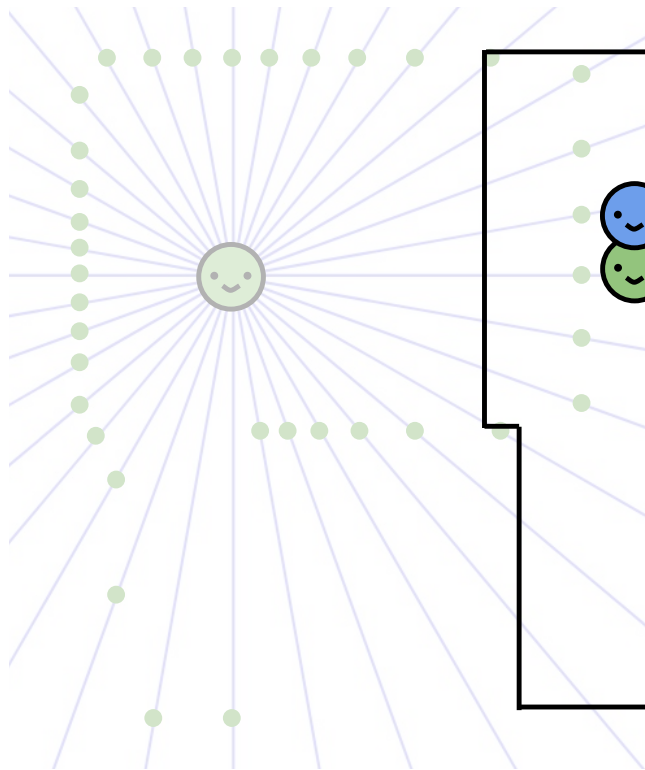
A



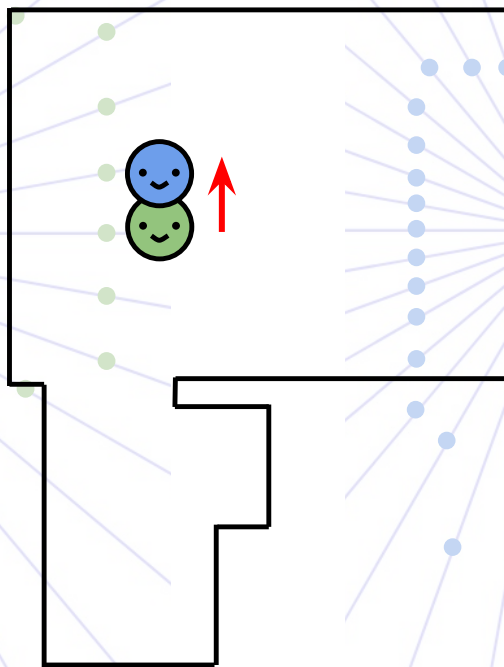
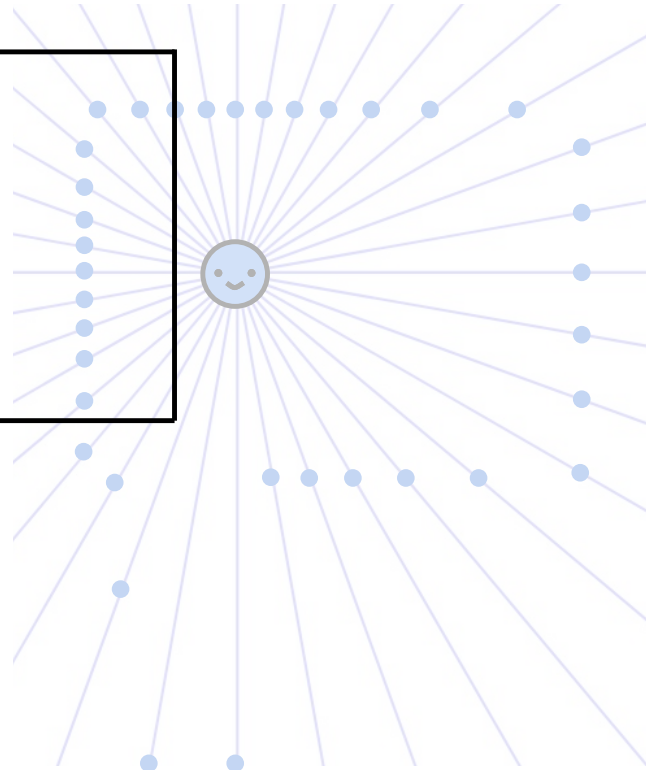
B



A



B



Point Cloud Registration (PCR)

Point Cloud Registration

How can we map one point cloud to another?

Point Cloud Registration

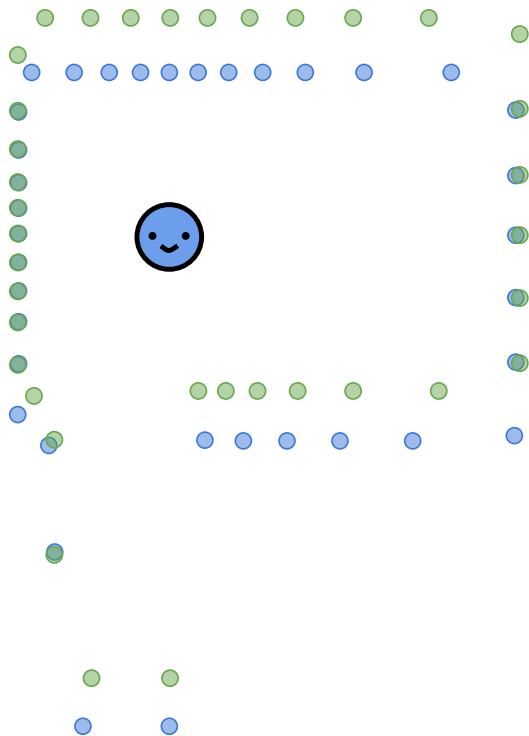
How can we map one point cloud to another?

How should the robot pose change?

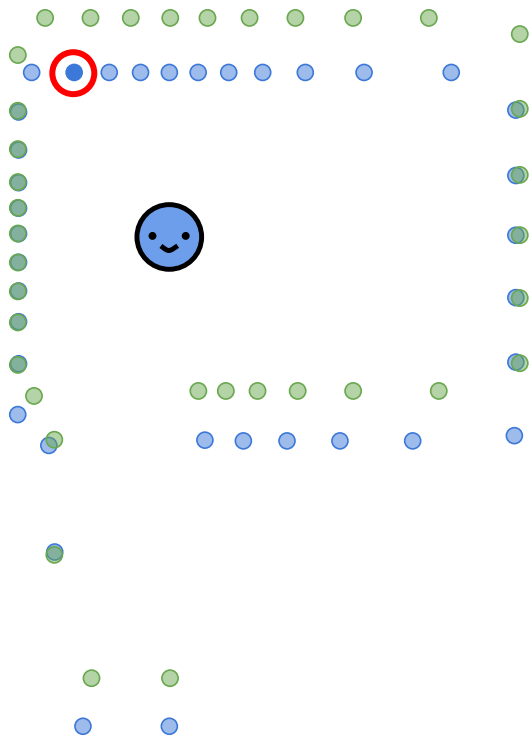
Point Cloud Registration

Iterative Closest Point (ICP)

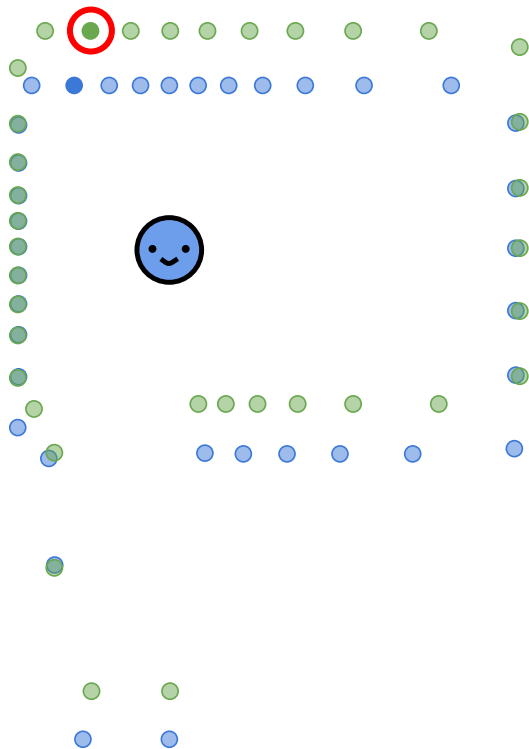
Iterative Closest Point



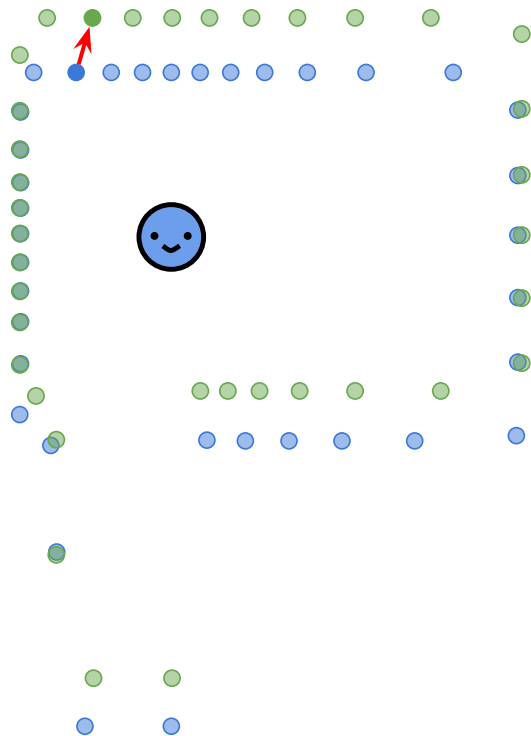
Iterative Closest Point



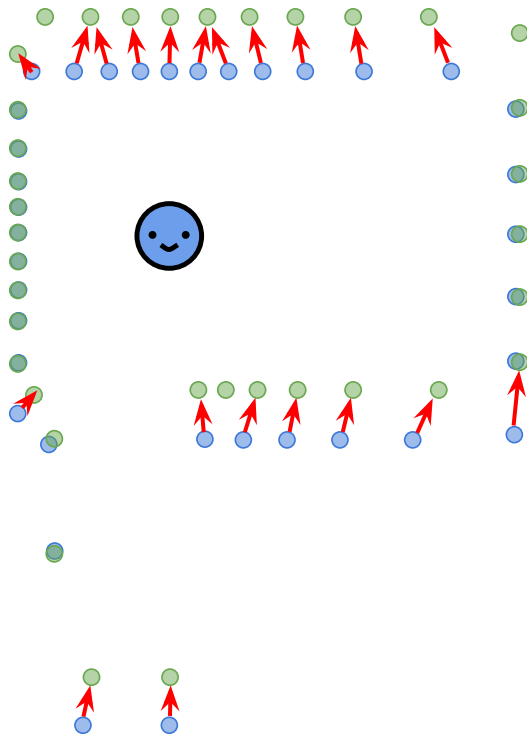
Iterative Closest Point



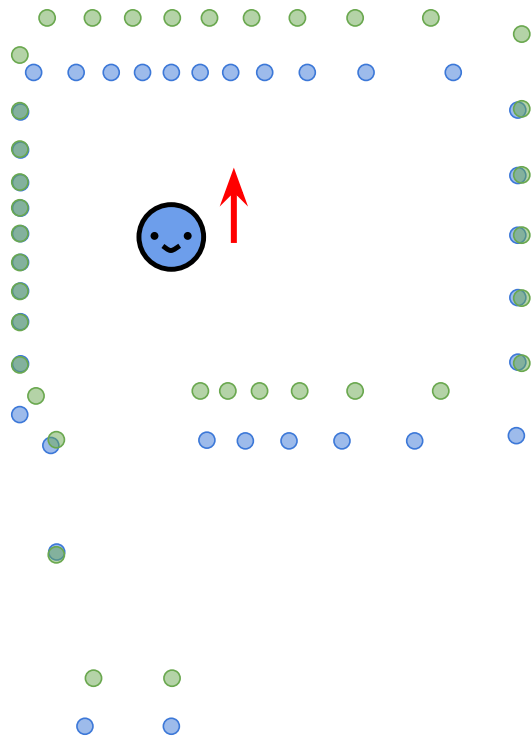
Iterative Closest Point



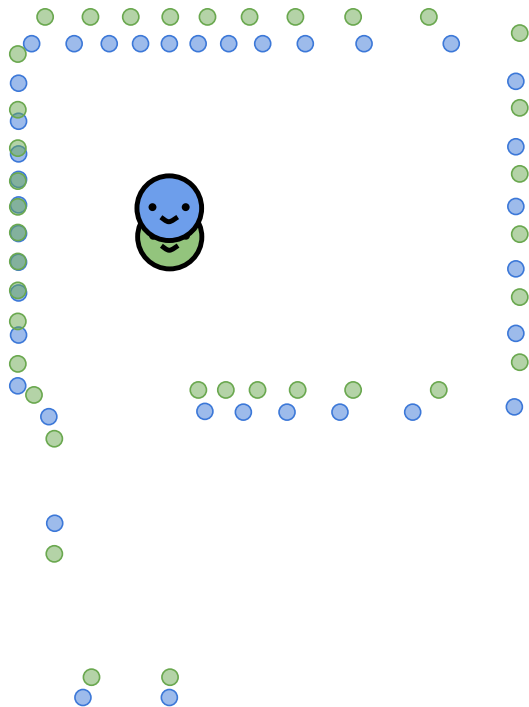
Iterative Closest Point



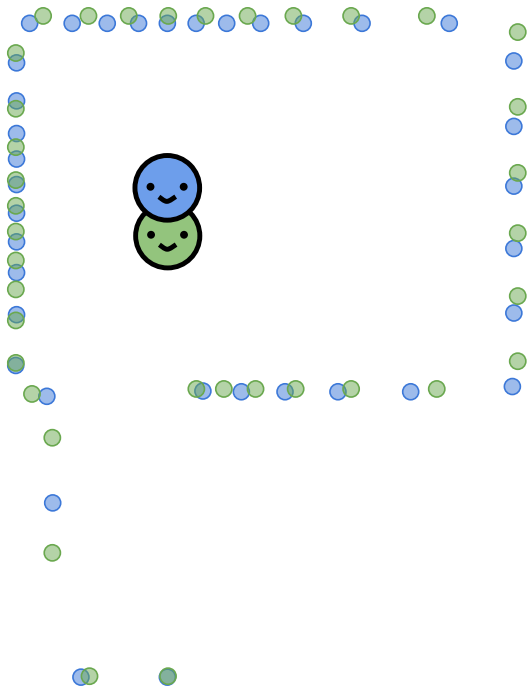
Iterative Closest Point



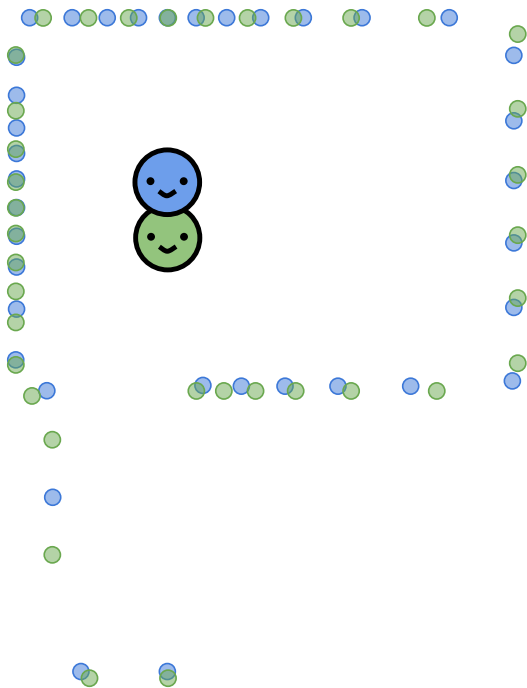
Iterative Closest Point



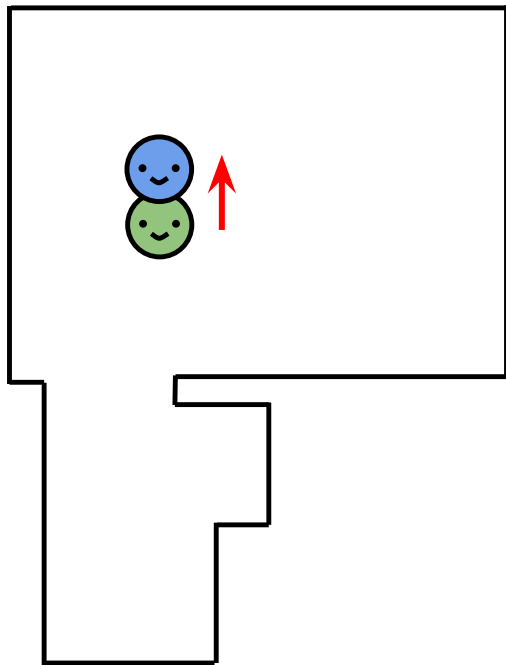
Iterative Closest Point



Iterative Closest Point



Iterative Closest Point



Our System

1. Map & Localize

2. Detect Errors

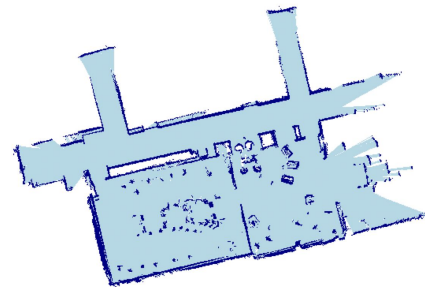
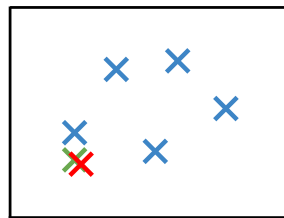
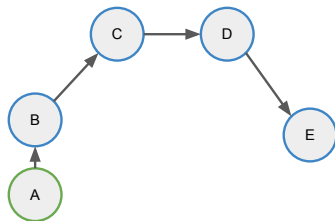
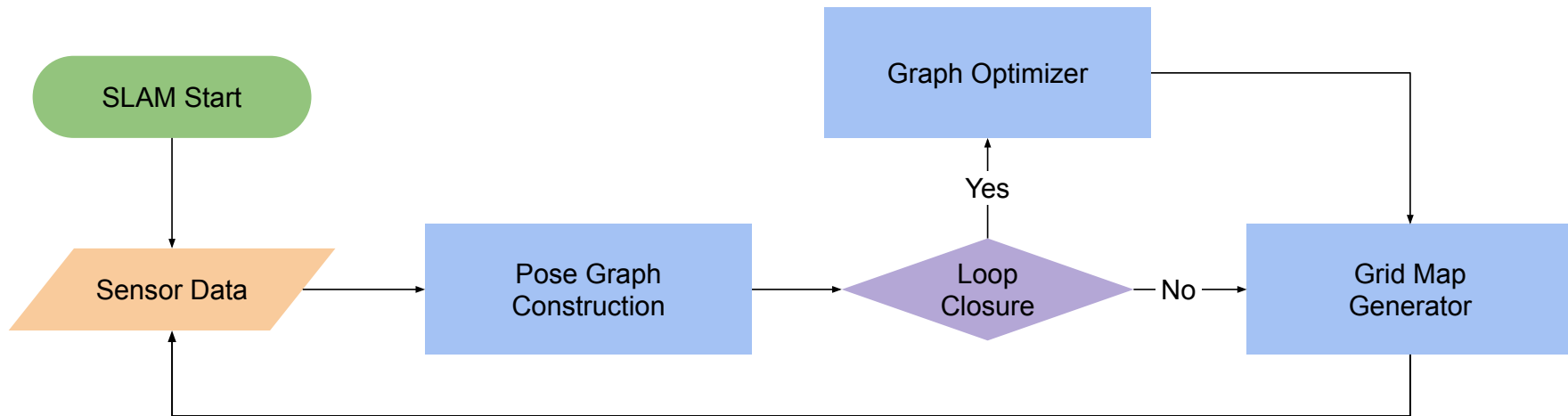
3. Fix Errors

Our System



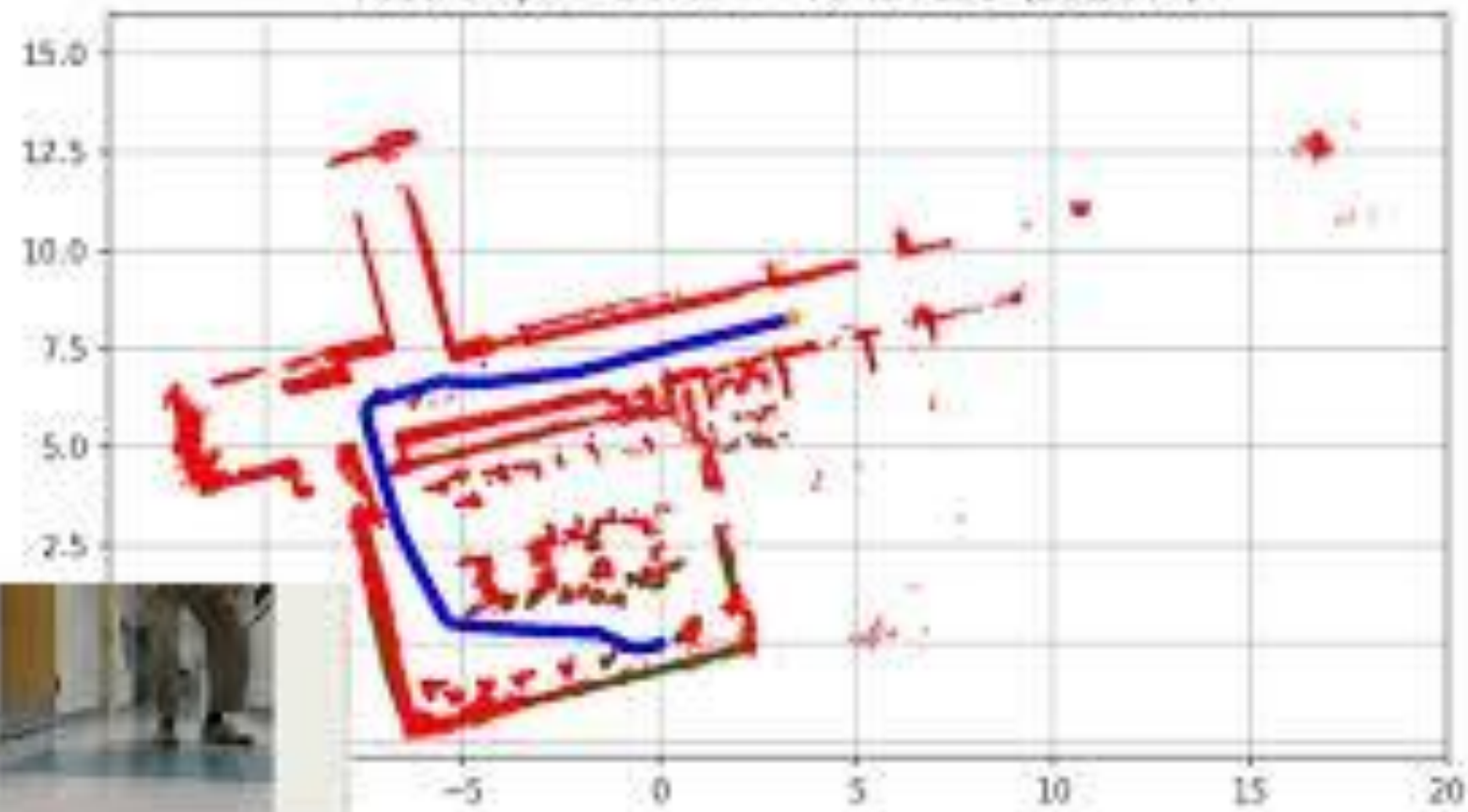
ROS2

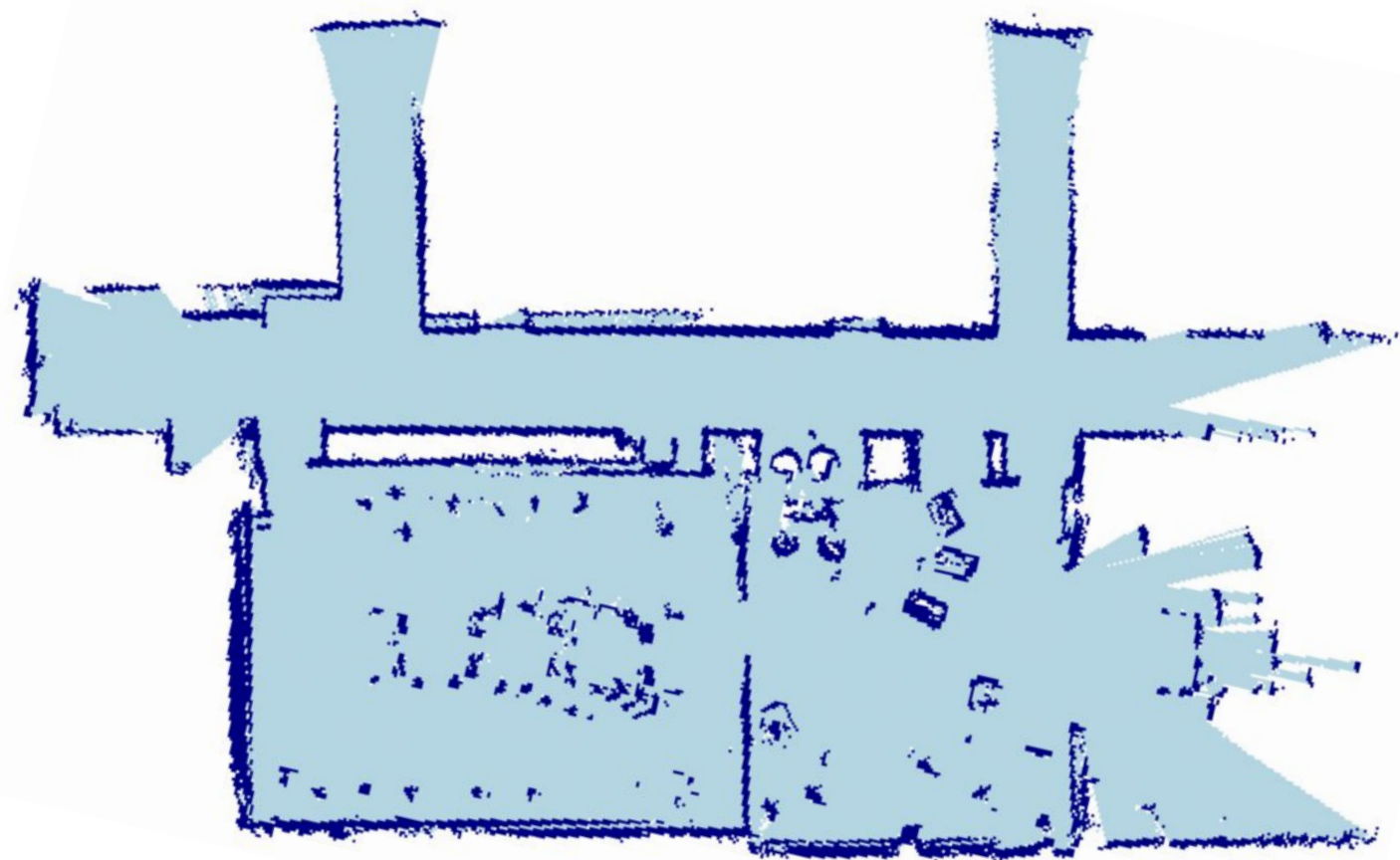
Our Final SLAM System

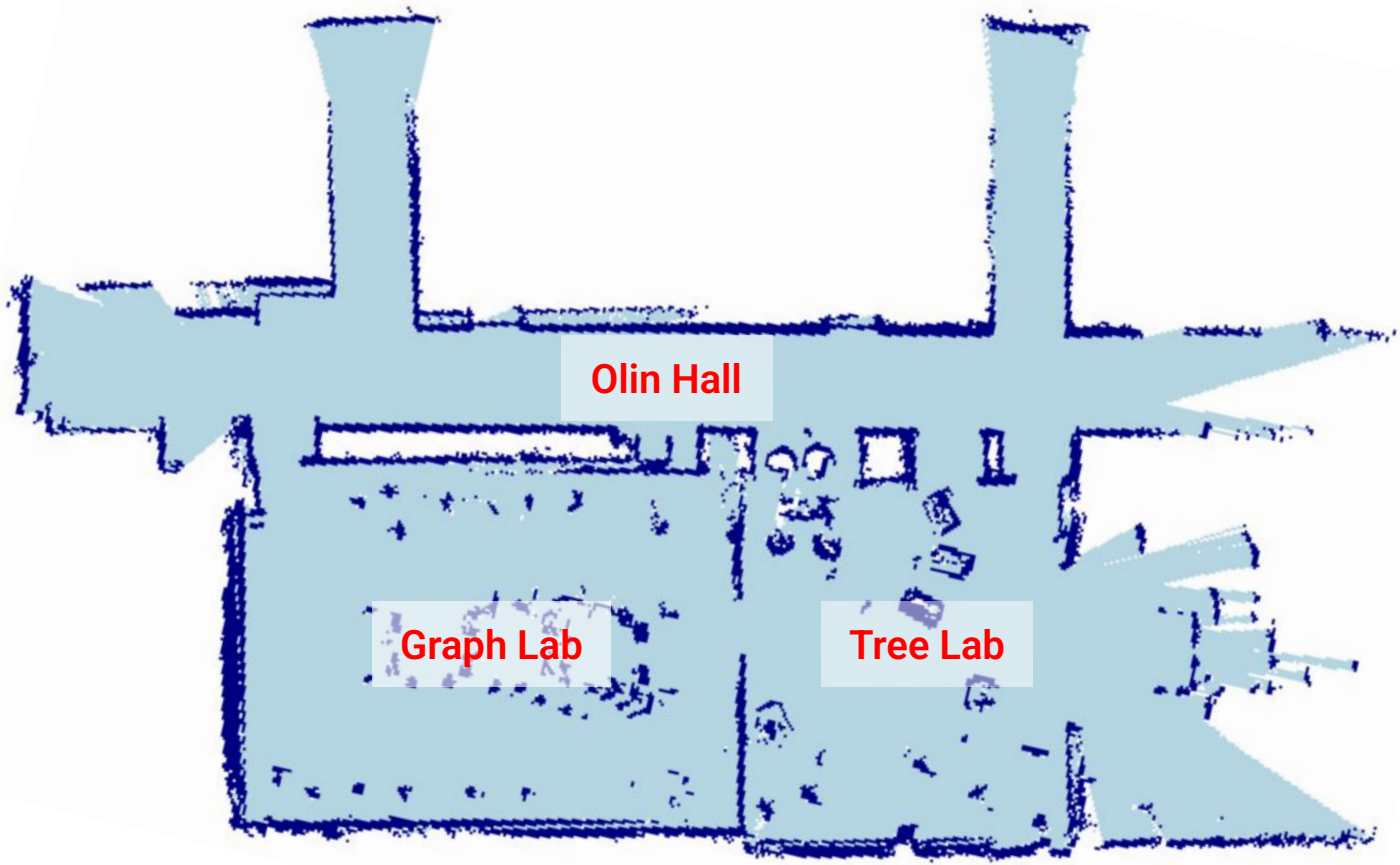


Results!

Pose Graph + LiDAR — vertex 211 (211/364)







Road map



Overview

SLAM Introduction

Pose-Graph SLAM

(with 2D LiDAR)

Orb SLAM

(Oriented FAST and Rotated
BRIEF)

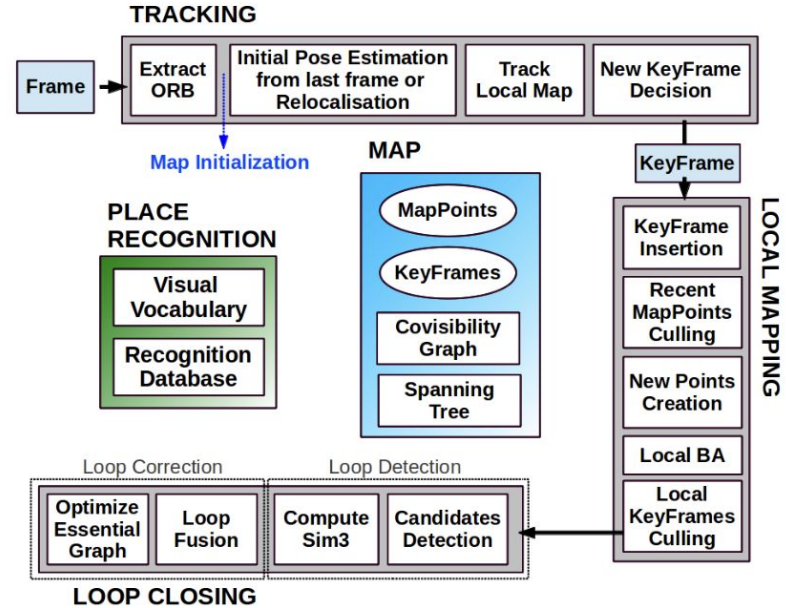
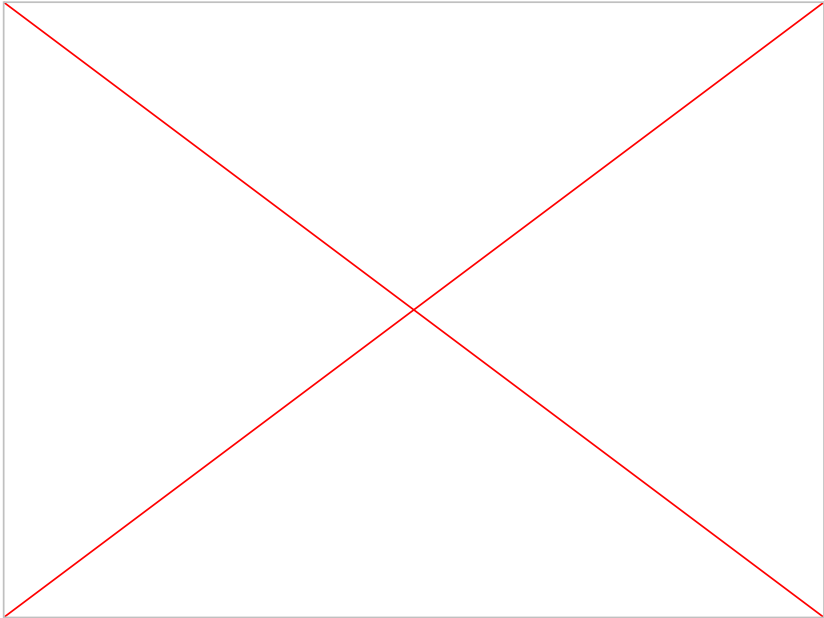
Conclusion

SLAM

ORB-SLAM

What is ORB-SLAM?

- Visual SLAM
- Variant of Graph SLAM

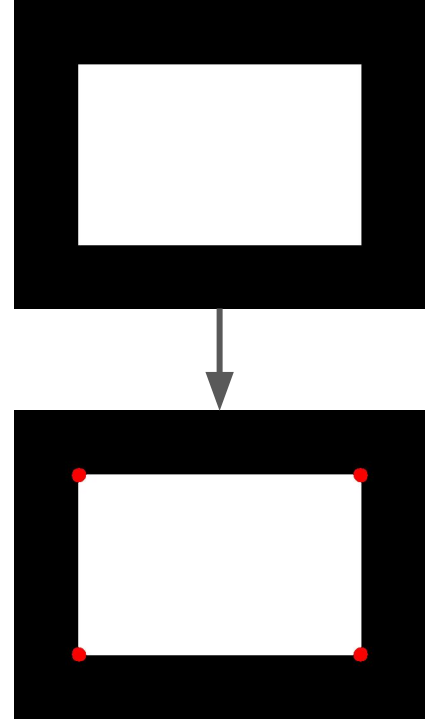
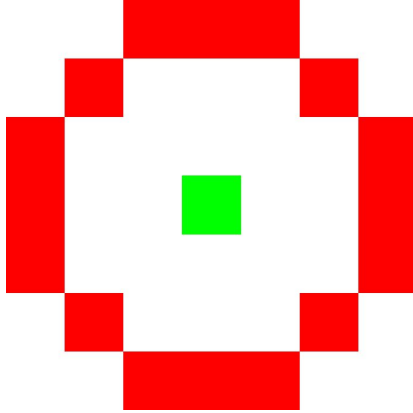


[Mur-Artal *et al.*, 2015]

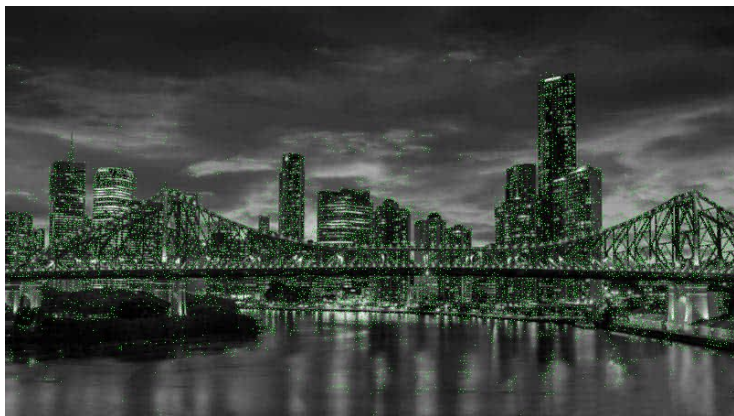
ORB - Oriented FAST and Rotated BRIEF

FAST (Features from Accelerated Segment Test)

- Detect corners of image
- Check surrounding pixels for brighter or darker



FAST on real examples



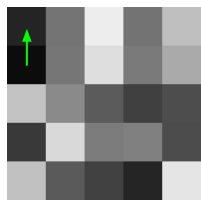
That's a lot of points!

- Take derivative of image
- Sort points by strongest response

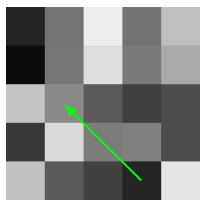


BRIEF (Binary Robust Independent Elementary Features)

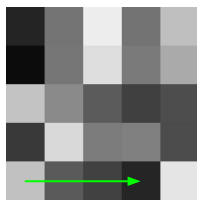
- Create binary number from tests between points
 - 1 if second point is brighter than reference point
 - 0 if it is darker or the same
- This image would be described as 110011



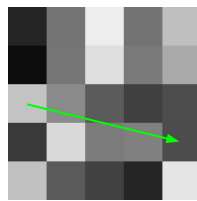
1



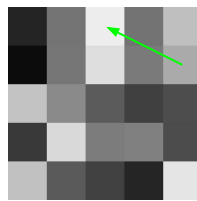
1



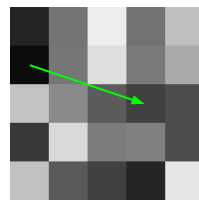
0



0



1



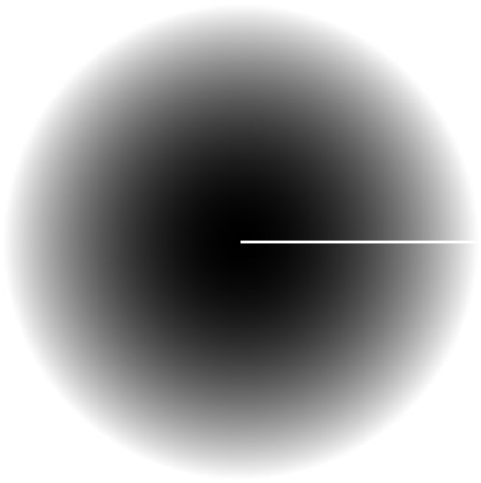
1

Oriented FAST and Rotated BRIEF

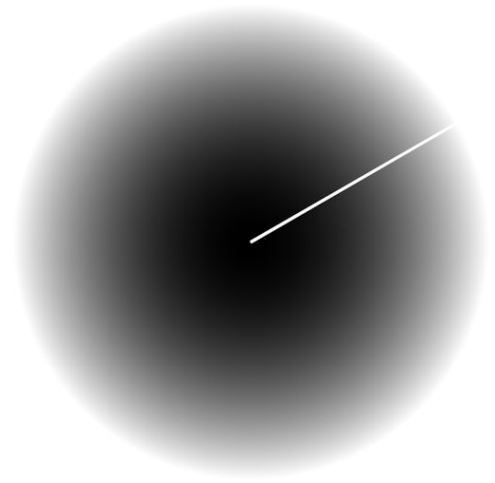
- We calculate orientation for each keypoint
- Calculate direction of strongest growth
- Rotate BRIEF tests to go with keypoint orientation



Angle = undefined



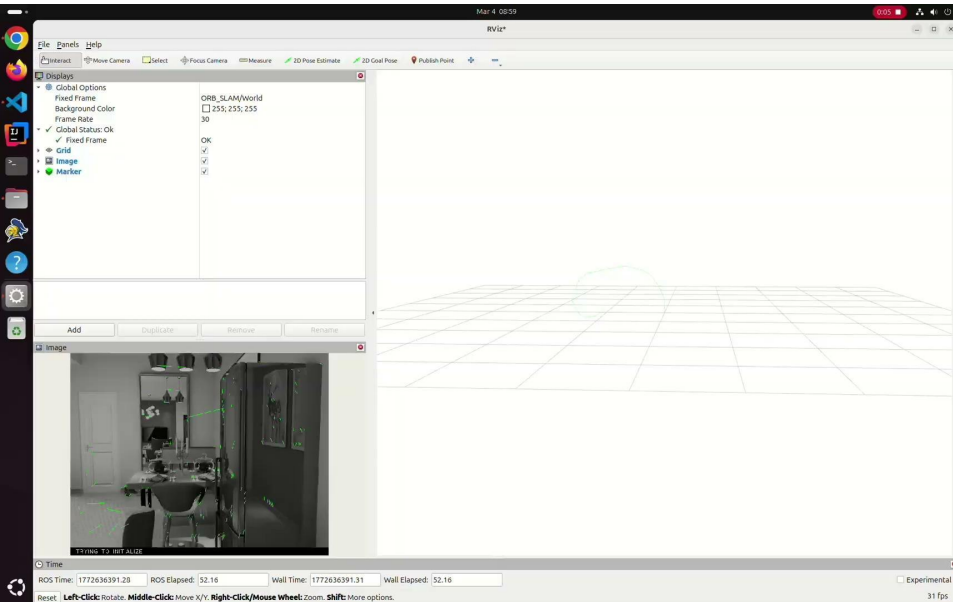
Angle = 0



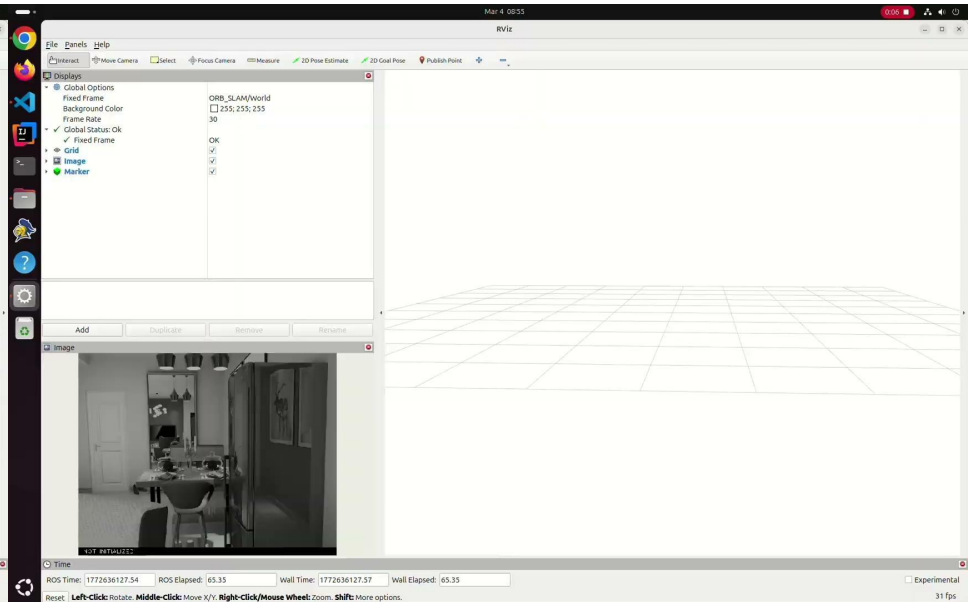
Angle = 329.94

Results

Original



My ORB Implementation



ORB Results on Demo

	Total Trials	Loop Closure Success	Sudden Movement Success	Rotated Image Success
Original rBRIEF	25	16	16	14
New rBRIEF	25	9	10	10

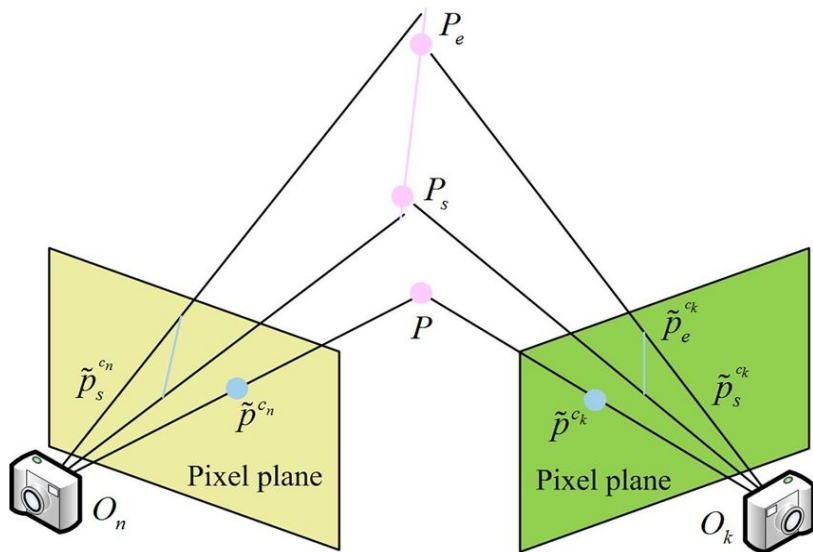
Challenges

- Needs to be **very** fast
- Does not adapt well to low texture environments
- Does not handle repeated objects well

Map Point Selection

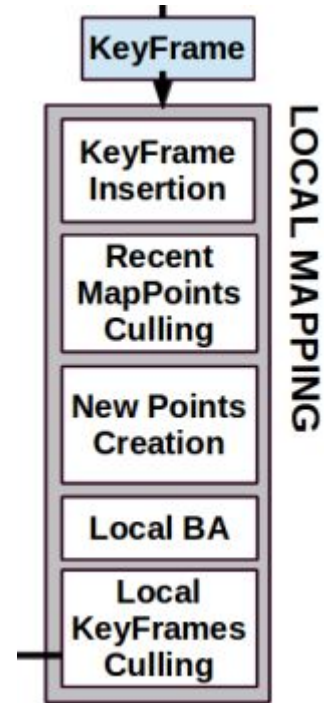
Map Point Creation

- “Generous in spawning”
- Triangulation:



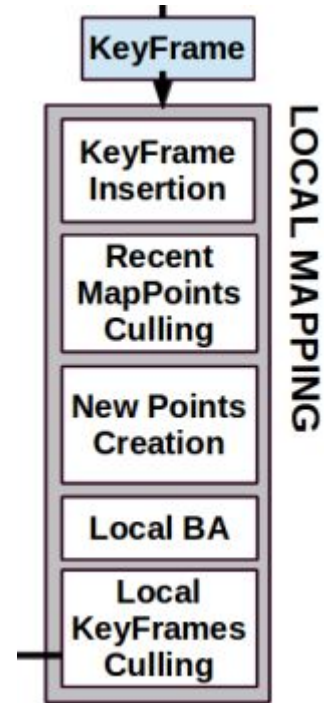
Selection in Local Mapping

- “Survival of the Fittest”
- Optimization methods
 - Map Point Culling
 - Local BA
 - Key Frame Culling
- Goal: keep the map small but informative



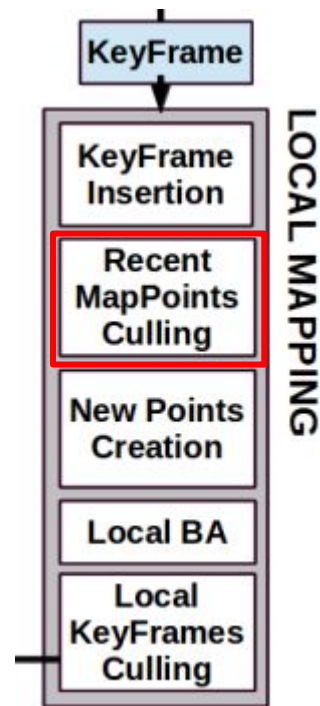
Selection in Local Mapping

- “Survival of the Fittest”
- Optimization methods
 - Map Point Culling
 - Local BA
 - Key Frame Culling
- Goal: keep the map small but informative



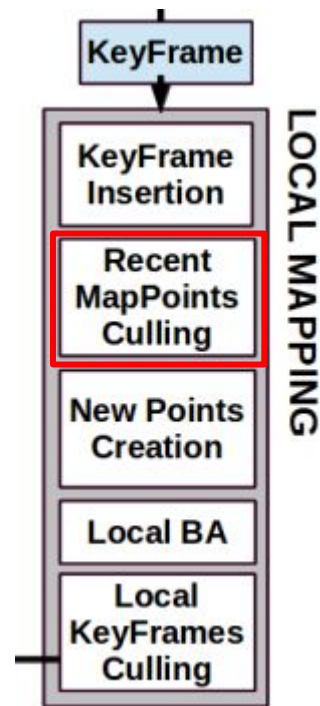
Original Map Point Culling

- “Restrictive in culling”
- A new Map Point must pass two tests
- Goal: remove poorly-triangulated points



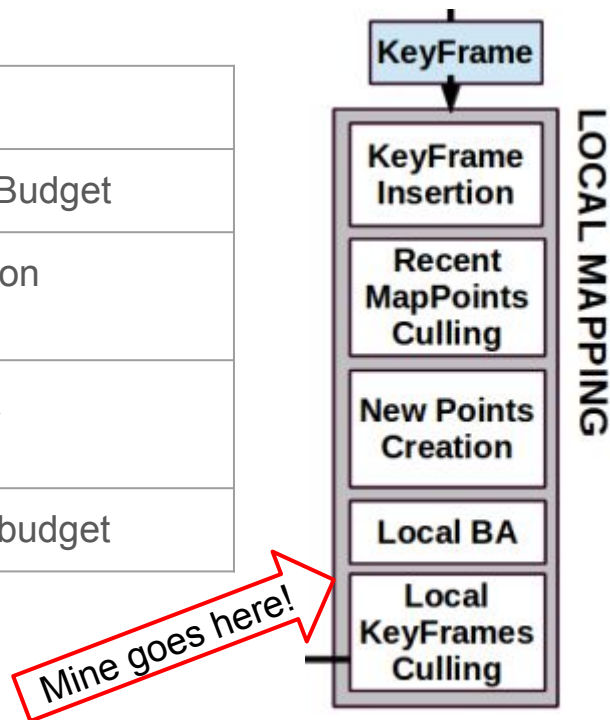
Original Map Point Culling

- “Restrictive in culling”
- A new Map Point must pass two tests
- Goal: remove poorly-triangulated points
- Limitation: only check quality but not relevance



Comparison

	Original Culling	My Selection
Filter type	Quality	Relevance + Budget
Criterion	Tracking ratio Observation count	Info contribution
When it runs	First 3 Key Frames after point creation	After local BA
Map size	Unbounded	Configurable budget

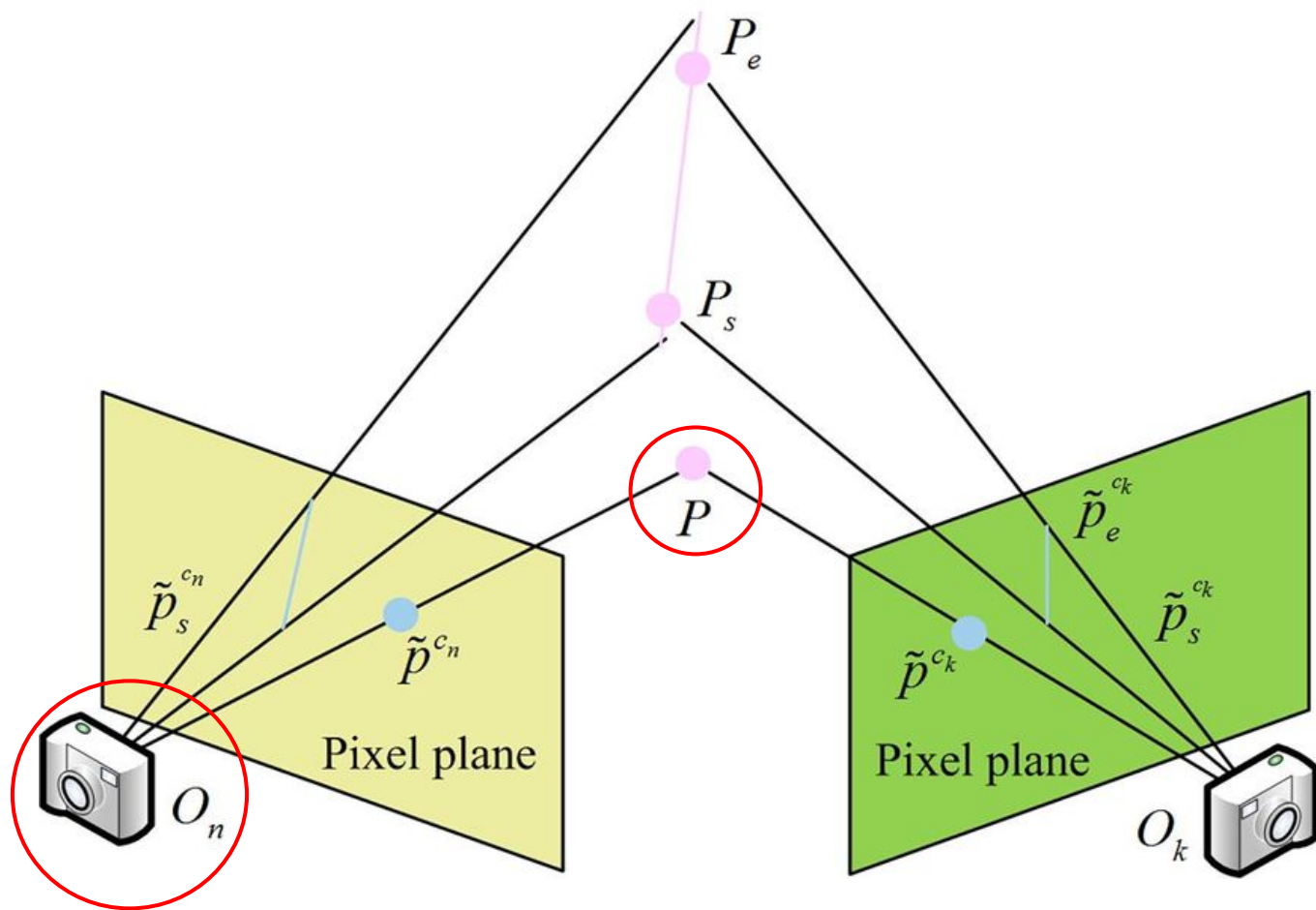


How Our Selection Works

- “Darwinian contest of information”
- Core idea: each Map Point contributes differently to camera pose estimation

How Our Selection Works

- “Darwinian contest of information”
- Core idea: each Map Point contributes differently to camera pose estimation
- Key concepts:
 - Observation Information



How Our Selection Works

- “Darwinian contest of information”
- Core idea: each Map Point contributes differently to camera pose estimation
- Key concepts:
 - Observation Information
 - Localisation Utility

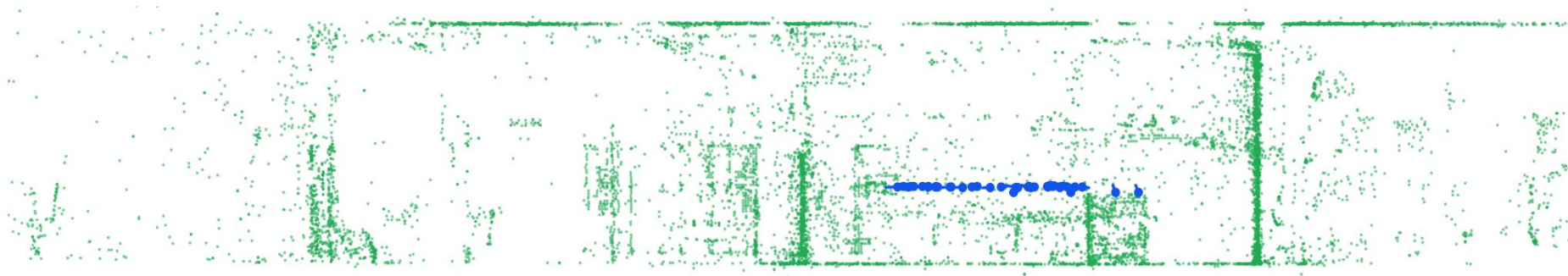
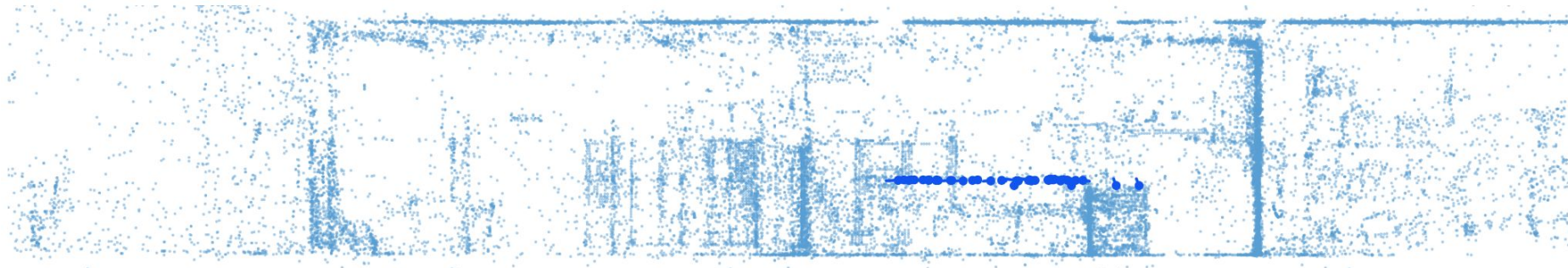
How Our Selection Works

- “Darwinian contest of information”
- Core idea: each Map Point contributes differently to camera pose estimation
- Key concepts:
 - Observation Information
 - Localisation Utility
 - Marginal Gain

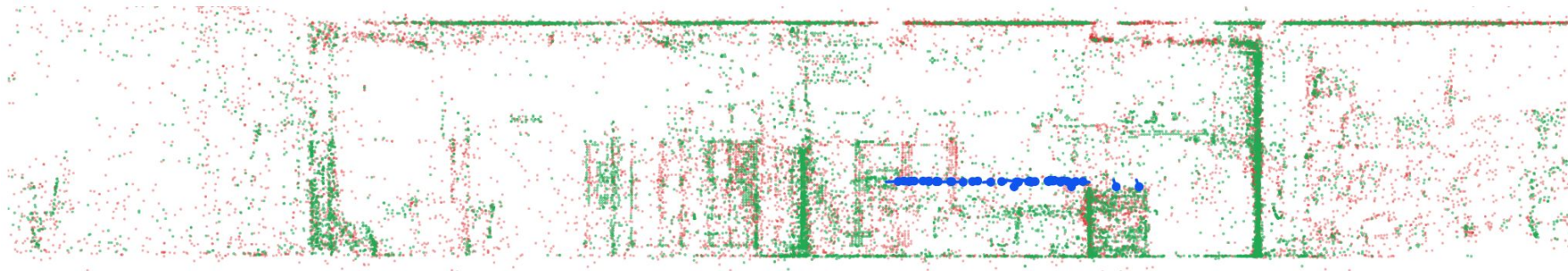
How Our Selection Works

- “Darwinian contest of information”
- Core idea: each Map Point contributes differently to camera pose estimation
- Key concepts:
 - Observation Information
 - Localisation Utility
 - Marginal Gain
- Selection Algorithm

Results(on an ETH3D Dataset)



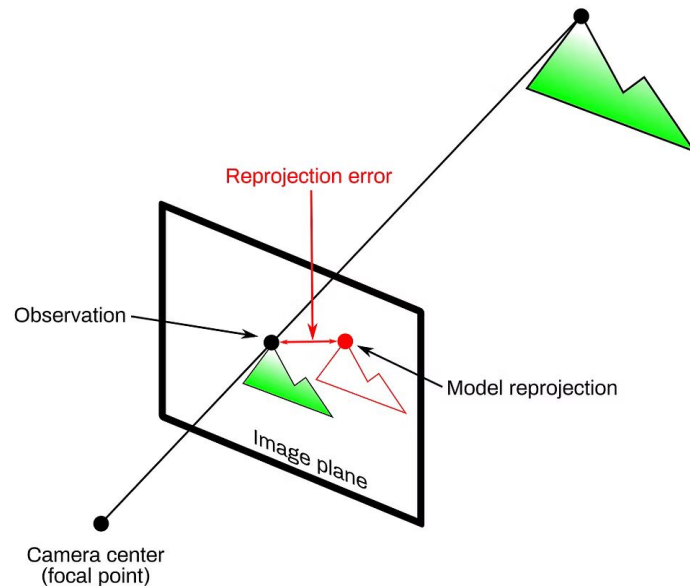
Results(on an ETH3D Dataset)



Bundle Adjustment (BA)

Bundle Adjustment

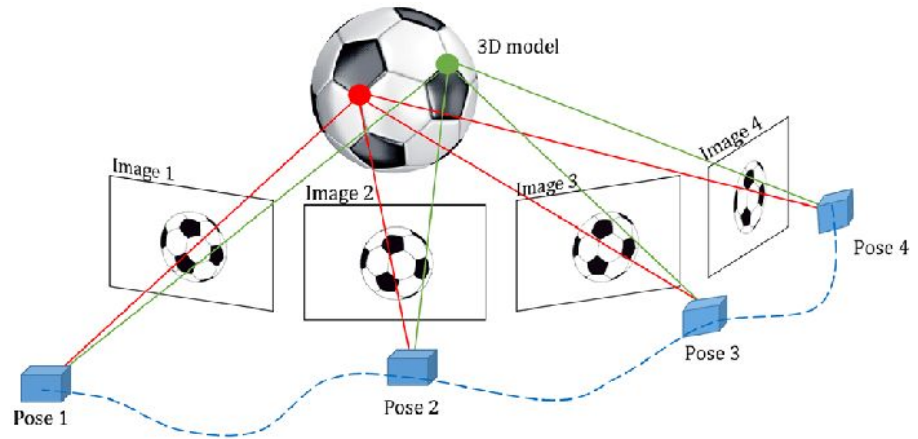
- Reduce the reprojection error
- Optimization of map points and poses
- Graph where
 - Nodes = map points and camera poses
 - Edges = visual observations



[Camcalib, 2022]

The Reprojection Error

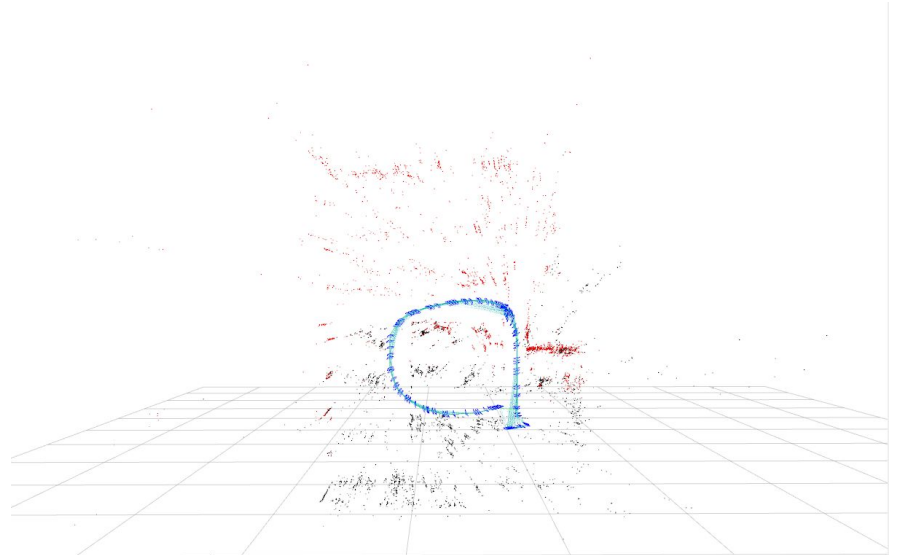
- Project 3D points to 2D coordinates
- There will always a difference between the reprojection and actual observation
- That gap is the reprojection error



[Azzam *et-al.*, 2020]

Minimizing the Error

- Shift camera poses and points
- Iteratively evaluates the error until a minimum is found
- Results in an accurate map!



Challenges of Bundle Adjustment

- Computationally expensive
- Outlier points can ruin the map

ORB SLAM Overview

Bryce - Find points in image, which generate map points

Austin - Filter out useless map points

Ngelek - Optimize map point location

Conclusion

Pose-Graph SLAM (using 2D LiDAR)

- Implemented basic Graph-SLAM, ICP, and loop closure
- Explored NDT odometry
- Utilized G2O graph optimization package
- Deployed SLAM components on TurtleBot4 for real-time mapping

ORB-SLAM (using camera)

- Developed port for ROS2 implementation of ORB-SLAM
- Implemented the Oriented FAST and Rotated BRIEF
- Extended Map Point selection algorithm
- Implemented optimizer for Bundle Adjustment

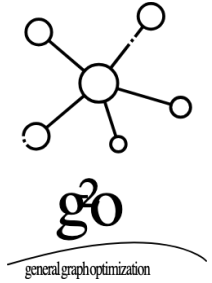
Works Cited

- Azzam, R., Taha, T., Huang, S., & Zweiri, Y. (2020). Feature-based visual simultaneous localization and mapping: A survey. *SN Applied Sciences*, 2(2). <https://doi.org/10.1007/s42452-020-2001-3>
- Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010). BRIEF: Binary Robust Independent Elementary Features. In K. Daniilidis, P. Maragos, & N. Paragios (Eds.), *Computer Vision – ECCV 2010* (pp. 778–792). Springer. https://doi.org/10.1007/978-3-642-15561-1_56
- Camcalib. (2022, April 11). *The reprojection error?* <https://www.camcalib.io/post/what-is-the-reprojection-error>
- Harris, C., Stephens, M. (1988). A Combined Corner and Edge Detector. *Plessey Research Roke Manor*. <https://www.bmva-archive.org.uk/bmvc/1988/avc-88-023.pdf>
- Müller, C. J., & van Daalen, C. E. (2023). Map point selection for visual SLAM. arXiv. <https://doi.org/10.48550/arXiv.2306.12901>
- Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5), 1147–1163. <https://doi.org/10.1109/TRO.2015.2463671>
- Rosten, E., & Drummond, T. (2006). Machine Learning for High-Speed Corner Detection. In A. Leonardis, H. Bischof, & A. Pinz (Eds.), *Computer Vision – ECCV 2006* (pp. 430–443). Springer. https://doi.org/10.1007/11744023_34
- Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. *2011 International Conference on Computer Vision*, 2564–2571. <https://doi.org/10.1109/ICCV.2011.6126544>
- Schöps, T., Schönberger, J. L., Galliani, S., Sattler, T., Schindler, K., Pollefeys, M., & Geiger, A. (2017). A multi-view stereo benchmark with high-resolution images and multi-camera videos. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3260–3269.

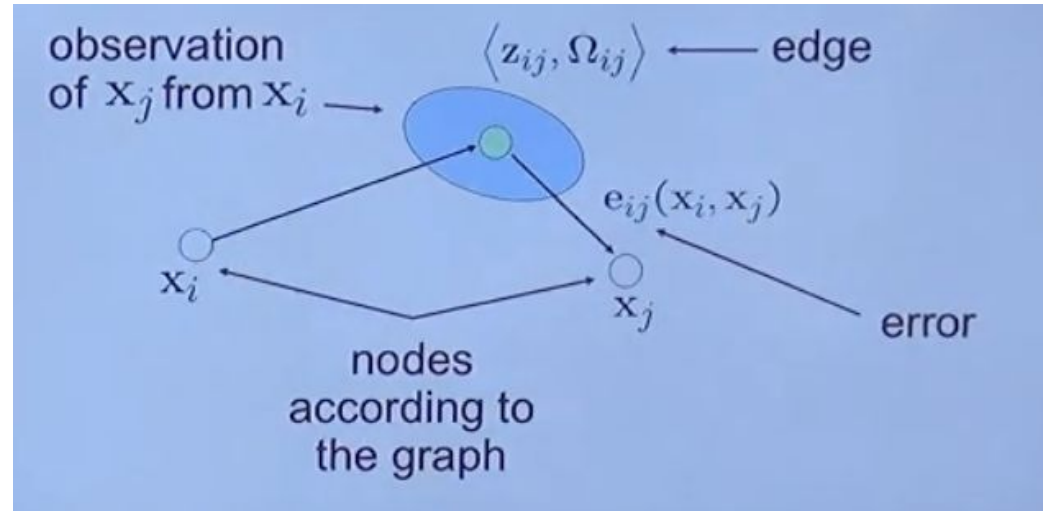
Questions?

Gauss–Newton algorithm

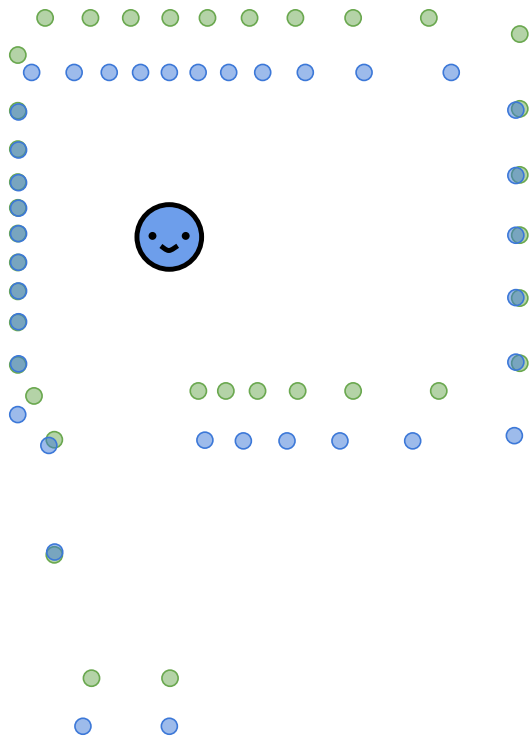
Minimizes the sum of squared function values.



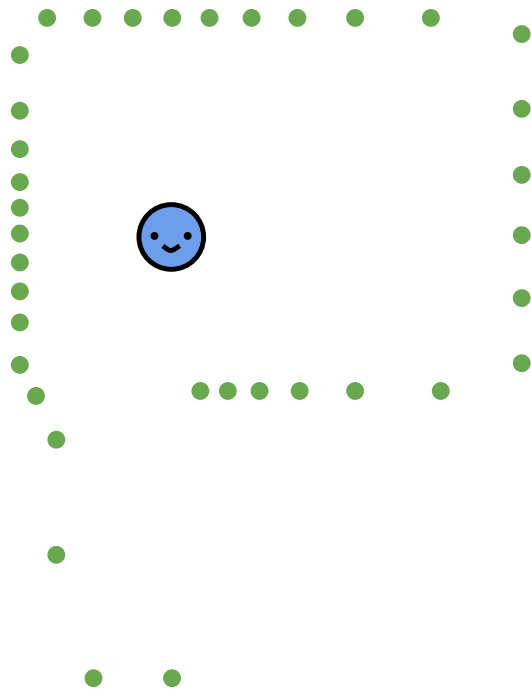
$$\sum_k \mathbf{e}_k^T(\mathbf{x}) \Omega_k \mathbf{e}_k(\mathbf{x})$$



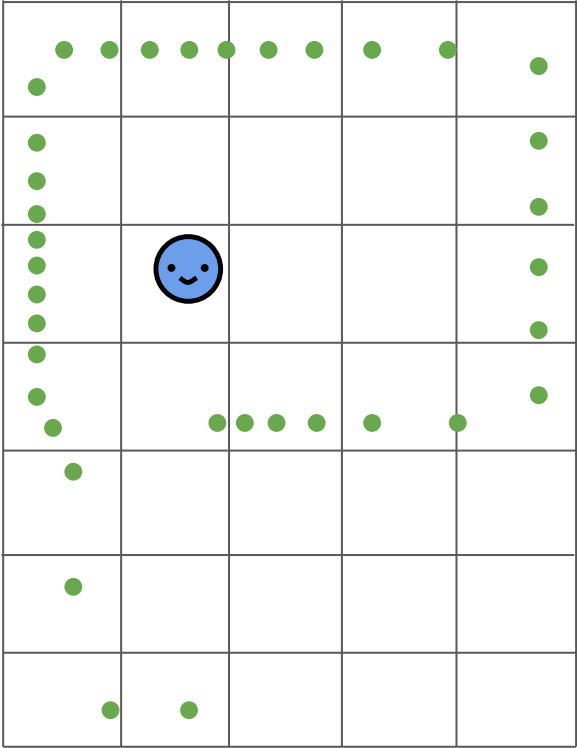
Normal Distributions Transform



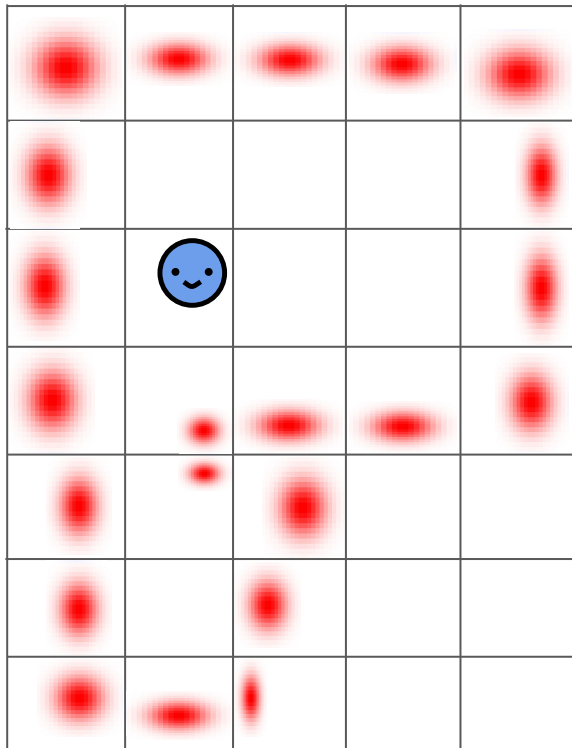
Normal Distributions Transform



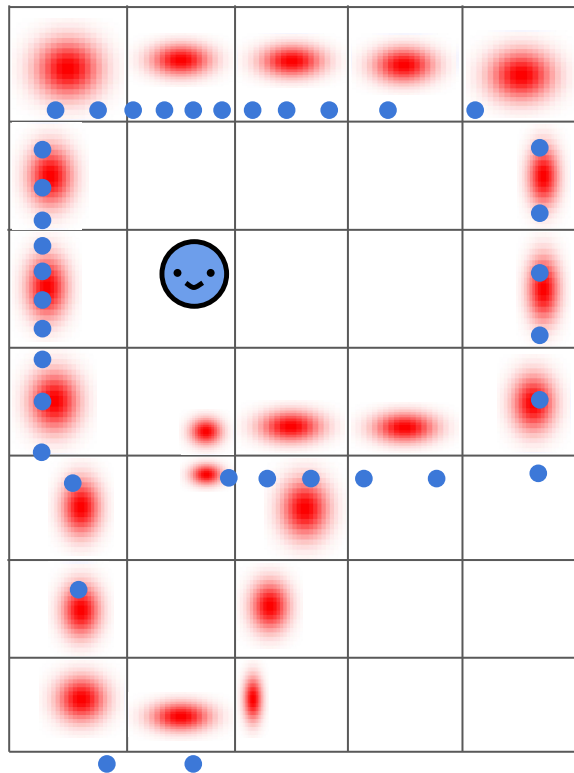
Normal Distributions Transform



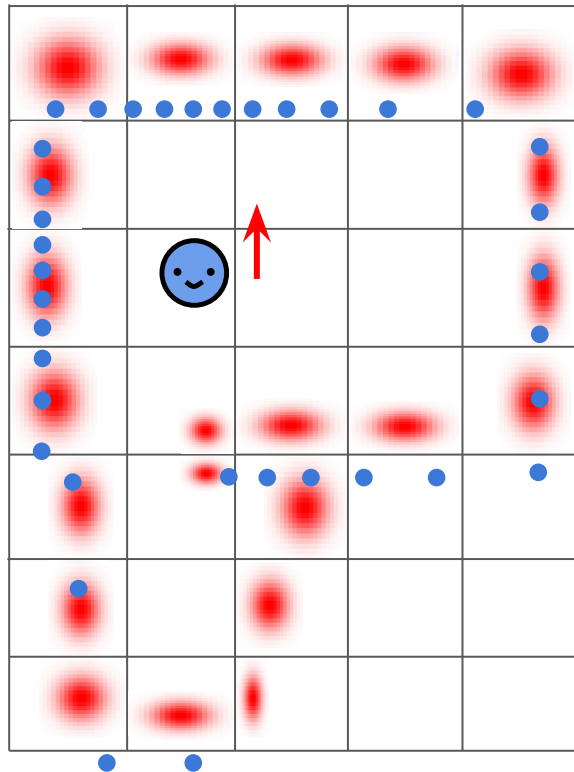
Normal Distributions Transform



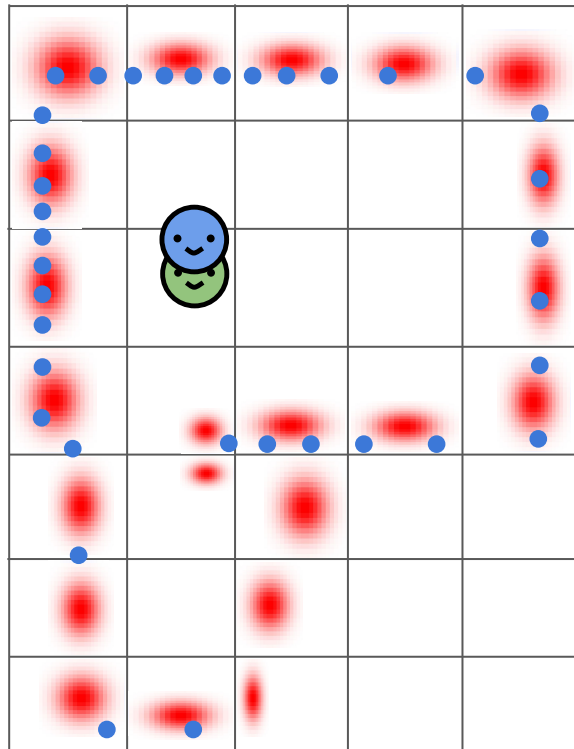
Normal Distributions Transform



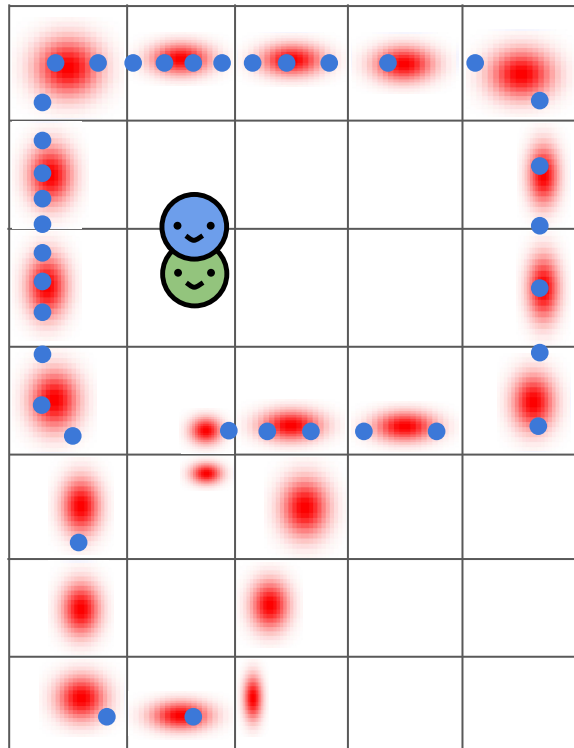
Normal Distributions Transform



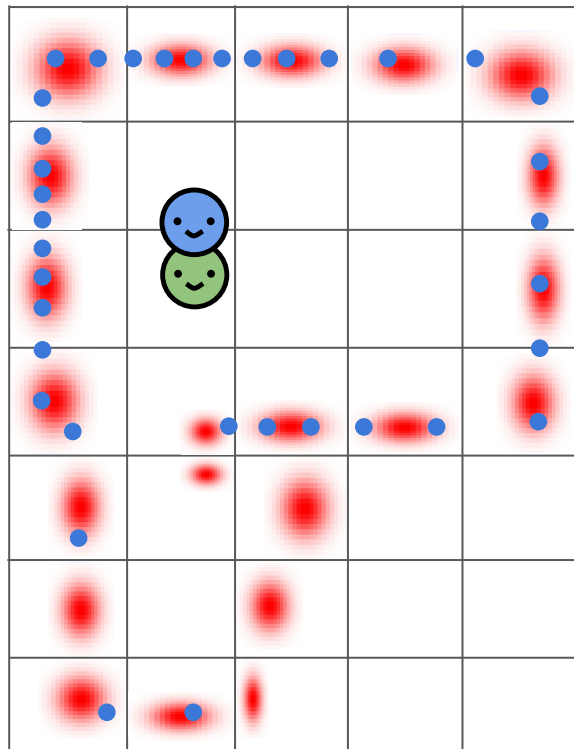
Normal Distributions Transform



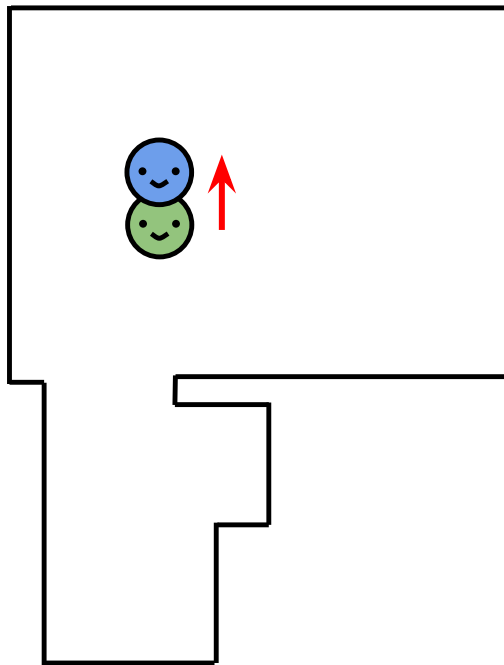
Normal Distributions Transform



Normal Distributions Transform



Normal Distributions Transform



Point Cloud Registration

(Method)	ICP	NDT
(Advantages)	Accurate	Less Accurate
(Disadvantages)	Slow Convergence	Fast Convergence

Point Cloud Registration

(Method)	ICP	NDT
(Advantages)	Accurate	Less Accurate
(Disadvantages)	Slow Convergence	Fast Convergence

NDT-ICP

Point Cloud Registration

(Method)	ICP	NDT
(Advantages)	Accurate	Less Accurate
(Disadvantages)	Slow Convergence	Fast Convergence

~~NDT-ICP~~