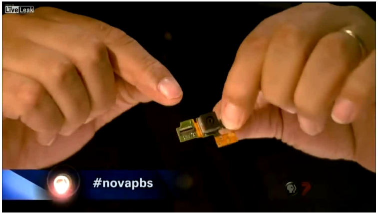
Task-driven Navigation and Mapping with Resource Constraints

Beipeng Mu

Thesis Committee: Jonathan P. How
John J. Leonard
John W. Fisher
Sertac Karaman

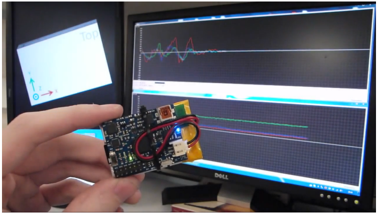
Massachusetts Institute of Technology

New sensing and processing technologies

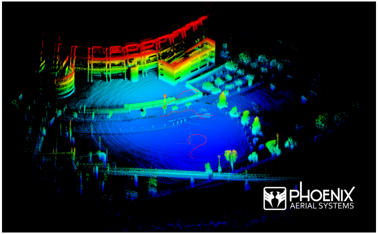


High resolution camera

RGB-D camera



High speed IMU

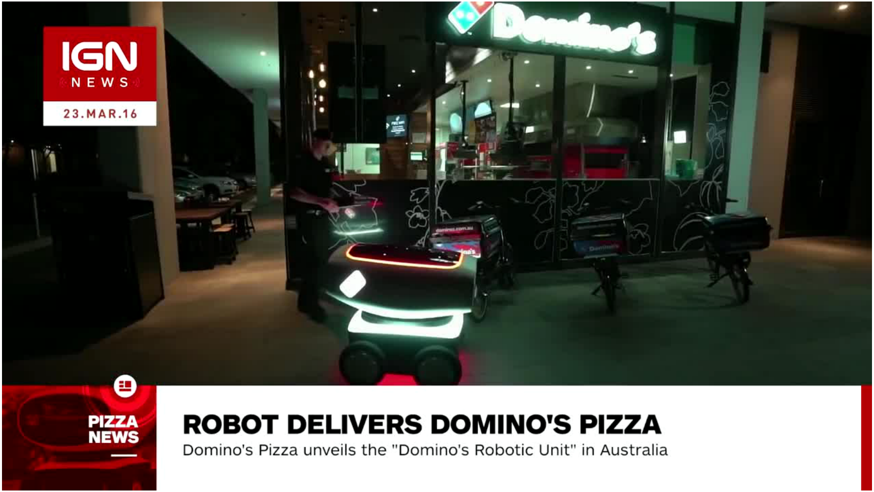


velodyne laser scanner

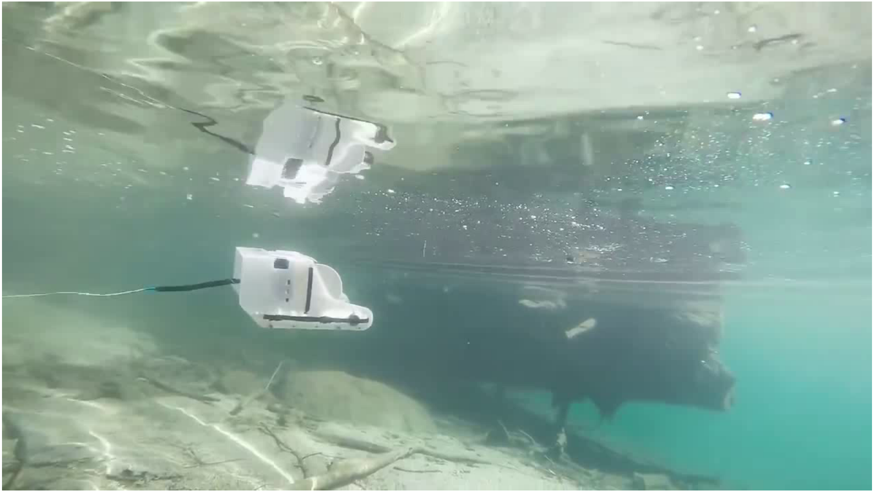
Application: autonomous driving

Application: exploring remote cave

Application: pizza delivery



Application: underwater exploration



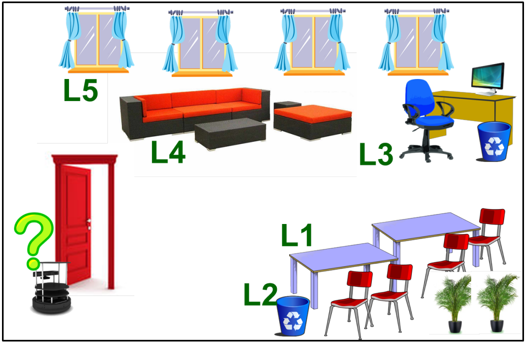
Localization and Mapping

- **Problem**: know where the robot is and where everything else is

- Given features $\{L_1, L_2, \cdots\}$, infer where robot is $\Rightarrow$ localization
- Given robot localization X_t, infer where features are $\Rightarrow$ mapping
- Given neither $\Rightarrow$ **simultaneous localization and mapping (SLAM)**

Localization and Mapping

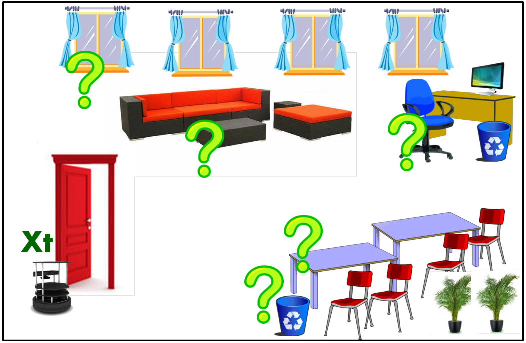
- **Problem**: know where the robot is and where everything else is



- Given features $\{L_1, L_2, \cdots\}$, infer where robot is $\Rightarrow$ localization
- Given robot localization X_t, infer where features are $\Rightarrow$ mapping
- Given neither $\Rightarrow$ **simultaneous localization and mapping (SLAM)**

Localization and Mapping

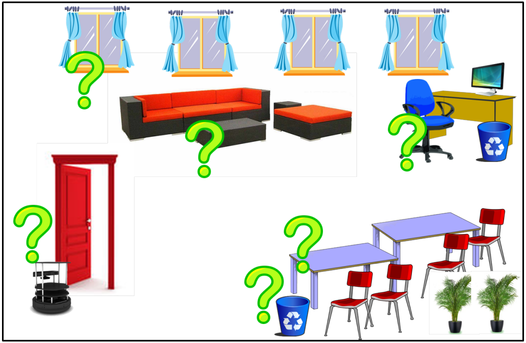
▶ **Problem**: know where the robot is and where everything else is



- Given features $\{L_1, L_2, \cdots\}$, infer where robot is $\Rightarrow$ localization
- Given robot localization X_t, infer where features are $\Rightarrow$ mapping
- Given neither $\Rightarrow$ **simultaneous localization and mapping (SLAM)**

Localization and Mapping

- **Problem**: know where the robot is and where everything else is



- Given features $\{L_1, L_2, \cdots\}$, infer where robot is $\Rightarrow$ localization
- Given robot localization X_t, infer where features are $\Rightarrow$ mapping
- Given neither $\Rightarrow$ **simultaneous localization and mapping (SLAM)**

Challenges

- Challenges
  - Big volumes of sensing data
  - Specific tasks ⇒ mapping for navigation
  - Resource constraints
    - computation, memory, communication, battery
- **Gap**: how to convert large volume of sensor data to **sparse models** for the task of **navigation and mapping** for *resource constrained* systems

dense sensing data

sparse model for resource constrained systems

Challenges

- Challenges
  - Big volumes of sensing data
  - Specific tasks ⇒ mapping for navigation
  - Resource constraints
    - computation, memory, communication, battery
- **Gap**: how to convert large volume of sensor data to **sparse models** for the task of **navigation and mapping** for *resource constrained* systems

dense sensing data

sparse model for resource constrained systems

Challenges

- Challenges
  - Big volumes of sensing data
  - Specific tasks $\Rightarrow$ mapping for navigation
  - Resource constraints
    - computation, memory, communication, battery
- **Gap**: how to convert large volume of sensor data to **sparse models** for the task of **navigation and mapping** for *resource constrained* systems

dense sensing data

sparse model for resource constrained systems

Challenges

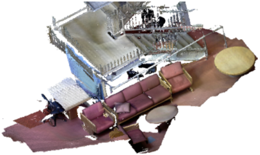
- Challenges
  - Big volumes of sensing data
  - Specific tasks ⇒ mapping for navigation
  - Resource constraints
    - computation, memory, communication, battery
- **Gap**: how to convert large volume of sensor data to **sparse models** for the task of **navigation and mapping** for *resource constrained* systems

dense sensing data

sparse model for resource constrained systems

Challenges

- Challenges
  - Big volumes of sensing data
  - Specific tasks $\Rightarrow$ mapping for navigation
  - Resource constraints
    - computation, memory, communication, battery
- **Gap**: how to convert large volume of sensor data to **sparse models** for the task of **navigation and mapping** for *resource constrained* systems



dense sensing data

sparse model for resource constrained systems

Outline

Literature Review

- Sparse mapping
  - Generic node marginalization, *Carlevaris-Bianco et al* [1]
  - Information-Based measurement removal, , *Ila et al* [2]
- Active SLAM
  - Information gain-based exploration *Stachniss, Kumar et al*[3, 4]
  - Frontier exploration *Yamauchi, Keidar et al* [5, 6]
- Object SLAM
  - Feature matching *Song, Civera, Newcombe et al*[7–9]
  - Deep learning *Pillai* [10]
- **GAP**: model built not necessary sparse or useful for tasks

Contributions

Desired capability

Active Planning

Sparse Mapping

Natural Environment

content

Task Driven

Resource Constraints

Contributions

- **Focused Mapping**
  - Evaluate information
  - compact models
- **Topological Feature Graph**
  - obstacle representations
  - active SLAM
- **Nonparametric Graph** model
  - data association
- Simulations and real-world experiments

Contributions

- **Focused Mapping**
  - Evaluate information
  - compact models
- **Topological Feature Graph**
  - obstacle representations
  - active SLAM
- **Nonparametric Graph** model
  - data association
- Simulations and real-world experiments

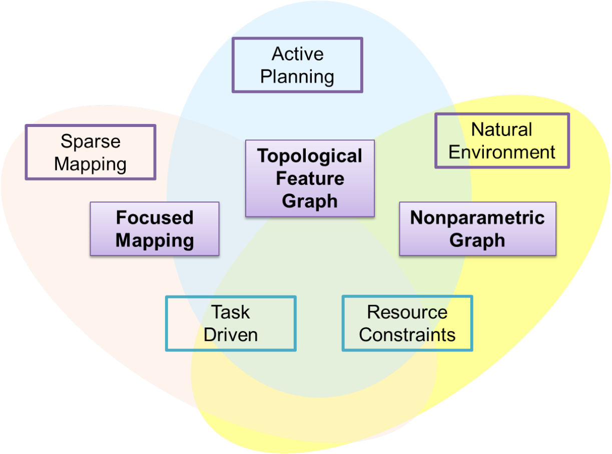
Contributions

- **Focused Mapping**
  - Evaluate information
  - compact models
- **Topological Feature Graph**
  - obstacle representations
  - active SLAM
- **Nonparametric Graph** model
  - data association
- Simulations and real-world experiments

Contributions

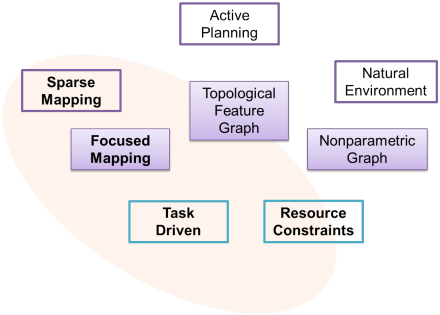
- **Focused Mapping**
  - Evaluate information
  - compact models
- **Topological Feature Graph**
  - obstacle representations
  - active SLAM
- **Nonparametric Graph** model
  - data association
- Simulations and real-world experiments

Contributions

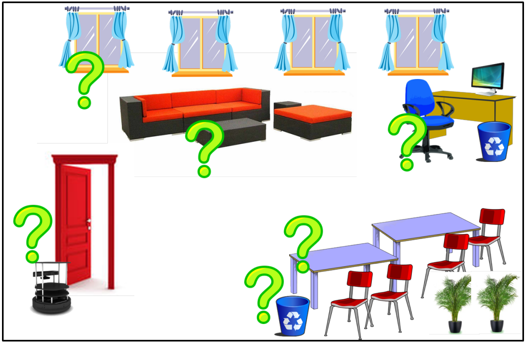


Outline

1. Motivation
2. Focused Mapping for Collision-free Navigation
3. Active Mapping
4. Object SLAM

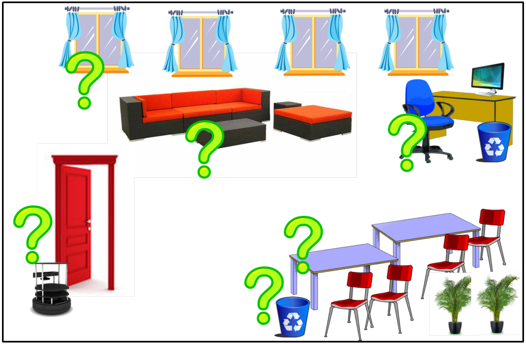


Focused Mapping for Collision-free Navigation (Mu et al [11])



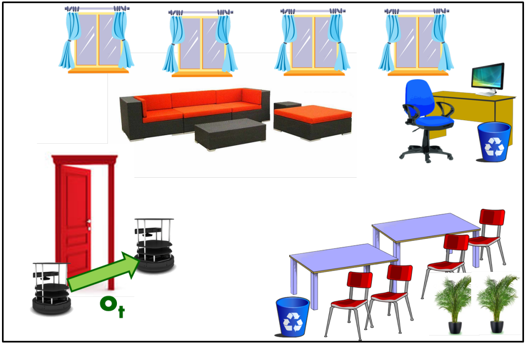
- **Problem**: given a dataset, generate a sparse map for navigation
- **Question**: what features and data to use for mapping?

Factor Graph Representation



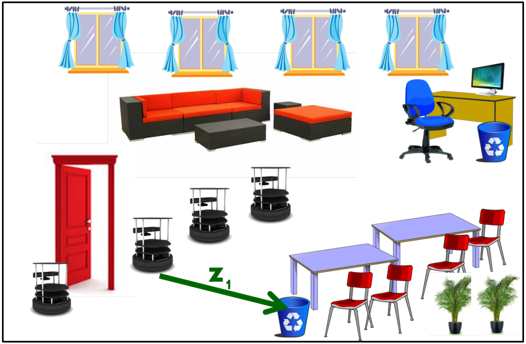
- robot poses: $X_{0:T}$; features: $L_{1:N}$

Factor Graph Representation



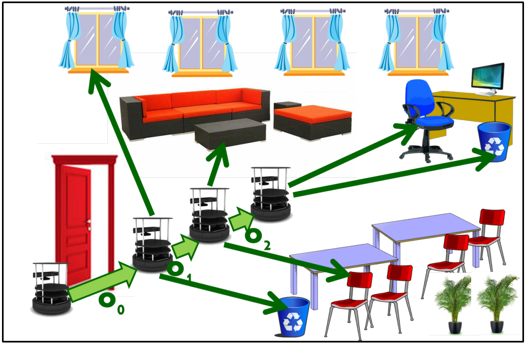
- odometry: $o_{1:T}$

Factor Graph Representation



- feature measurement: $z_{0:T}$

Factor Graph Representation



- Joint likelihood

$$P(X_{0:T}, L_{1:N}) = \prod p(o_t; X_t, X_{t-1}) \prod (z_t^k; X_t, L_k) \qquad (1)$$

Focused Mapping

- Navigation in unknown environment:
  - Long distance $\Rightarrow$ larger feature space $\Rightarrow$ memory requirement grows
  - Long duration $\Rightarrow$ more data $\Rightarrow$ computation complexity increases
- **Questions** What data/feature to retain?
  - Directly related to collision-free navigation

features in narrow corridor more important for navigation

- **Approach**: two-stage selection
  - Focused features, min $\text{Prob}_{\text{collision}}$
  - Data, max $\text{info}_{\text{features}}$
- **Contribution** a much **sparser model** that still enables good **task performance**

Focused Mapping

- Navigation in unknown environment:
  - Long distance $\Rightarrow$ larger feature space $\Rightarrow$ memory requirement grows
  - Long duration $\Rightarrow$ more data $\Rightarrow$ computation complexity increases
- **Questions** What data/feature to retain?
  - Directly related to collision-free navigation

features in narrow corridor more important for navigation

- **Approach**: two-stage selection
  - Focused features, min $\text{Prob}_{\text{collision}}$
  - Data, max $\text{info}_{\text{features}}$
- **Contribution** a much **sparser model** that still enables good **task performance**

Focused Mapping

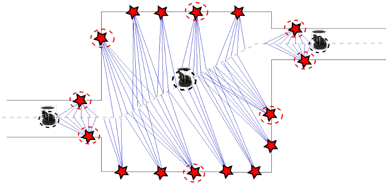
- Navigation in unknown environment:
  - Long distance $\Rightarrow$ larger feature space $\Rightarrow$ memory requirement grows
  - Long duration $\Rightarrow$ more data $\Rightarrow$ computation complexity increases
- **Questions** What data/feature to retain?
  - Directly related to collision-free navigation

features in narrow corridor more important for navigation

- **Approach**: two-stage selection
  - Focused features, min $\text{Prob}_{\text{collision}}$
  - Data, max $\text{info}_{\text{features}}$
- **Contribution** a much **sparser model** that still enables good **task performance**

Focused Mapping

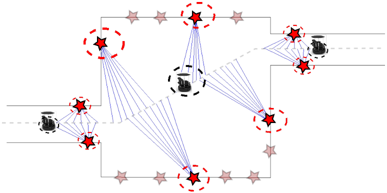
- Navigation in unknown environment:
  - Long distance $\Rightarrow$ larger feature space $\Rightarrow$ memory requirement grows
  - Long duration $\Rightarrow$ more data $\Rightarrow$ computation complexity increases
- **Questions** What data/feature to retain?
  - Directly related to collision-free navigation
- **Approach**: two-stage selection
  - Focused features, min $\text{Prob}_{\text{collision}}$
  - Data, max $\text{info}_{\text{features}}$
- **Contribution** a much **sparser model** that still enables good **task performance**



features in narrow corridor more important for navigation

Focused Mapping

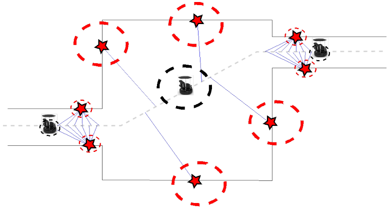
- Navigation in unknown environment:
  - Long distance ⇒ larger feature space ⇒ memory requirement grows
  - Long duration ⇒ more data ⇒ computation complexity increases
- **Questions** What data/feature to retain?
  - Directly related to collision-free navigation
- **Approach**: two-stage selection
  - Focused features, min $\text{Prob}_{\text{collision}}$
  - Data, max $\text{info}_{\text{features}}$
- **Contribution** a much **sparser model** that still enables good **task performance**



features in narrow corridor more important for navigation

Focused Mapping

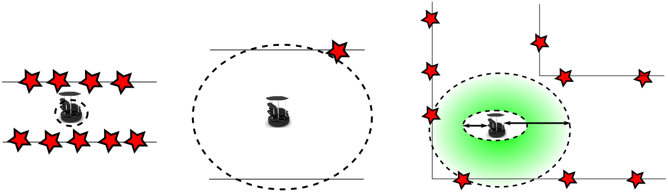
- Navigation in unknown environment:
  - Long distance ⇒ larger feature space ⇒ memory requirement grows
  - Long duration ⇒ more data ⇒ computation complexity increases
- **Questions** What data/feature to retain?
  - Directly related to collision-free navigation
- **Approach**: two-stage selection
  - Focused features, min $\text{Prob}_{\text{collision}}$
  - Data, max $\text{info}_{\text{features}}$
- **Contribution** a much **sparser model** that still enables good **task performance**



features in narrow corridor more important for navigation

Stage 1: Select focused features to minimize collision probability

- **Approximate collision probability**



- Pose uncertainty: covariance $\Sigma(x)$
- Distance to obstacle: closest obstacle point x_{obs}
- Mahalanobis dist:

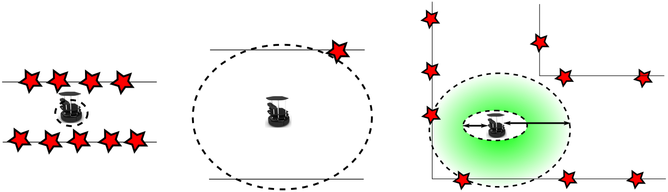
$$P(x) \sim \exp\left\{-(x - x_{obs})^T \Sigma^{-1}(x)(x - x_{obs})\right\}$$

- **Focused features**: reduce maximal collision probability

$$L_f = \arg\min_{L_f \subset L} \max_{x \in X_{0:T}} P(x)$$

Stage 1: Select focused features to minimize collision probability

- **Approximate collision probability**



  - Pose uncertainty: covariance $\Sigma(x)$
  - Distance to obstacle: closest obstacle point x_{obs}
  - Mahalanobis dist:

$$P(x) \sim \exp\left\{-(x - x_{obs})^T \Sigma^{-1}(x)(x - x_{obs})\right\}$$

- **Focused features**: reduce maximal collision probability

$$L_f = \arg\min_{L_f \subset L} \max_{x \in X_{0:T}} P(x)$$

Stage 2: Select data to maximize info gain

- Initialize with:
  - odometry, features, but no measurements
- For a set of measurements z^R, quantify information:
  - Log likelihood

$$\log P(o, z^R; X, L) \sim \sum_c \phi_c(o_c, z_c^R; X_c, L_c)$$

  - Quantify uncertainty: Laplacian approximation $P(o, z^R; X, L) \approx \mathcal{N}(\zeta, \Lambda_z)$, $\Rightarrow$ close-form entropy $H(L_f | z^R)$
- Select measurements z^R out of all measurements z greedily to max info on focused features L_f

$$\max_{z^R \subset z} \quad H(L_f \mid z^R) \text{ s.t. } \quad |z^R| \leq C$$

Stage 2: Select data to maximize info gain

- Initialize with:
  - odometry, features, but no measurements
- For a set of measurements z^R, quantify information:
  - Log likelihood

$$\log P(o, z^R; X, L) \sim \sum_c \phi_c(o_c, z^R_c; X_c, L_c)$$

  - Quantify uncertainty: Laplacian approximation $P(o, z^R; X, L) \approx \mathcal{N}(\zeta, \Lambda_z)$, $\Rightarrow$ close-form entropy $H(L_f | z^R)$
- Select measurements z^R out of all measurements z greedily to max info on focused features L_f

$$\max_{z^R \subset z} \quad H(L_f \mid z^R) \ \text{ s.t. } \ |z^R| \le C$$

Stage 2: Select data to maximize info gain

- Initialize with:
  - odometry, features, but no measurements
- For a set of measurements z^R, quantify information:
  - Log likelihood

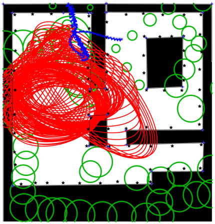
$$\log P(o, z^R; X, L) \sim \sum_c \phi_c(o_c, z_c^R; X_c, L_c)$$

  - Quantify uncertainty: Laplacian approximation
    $P(o, z^R; X, L) \approx \mathcal{N}(\zeta, \Lambda_z)$, $\Rightarrow$ close-form entropy $H(L_f | z^R)$
- Select measurements z^R out of all measurements z greedily to max info on focused features L_f

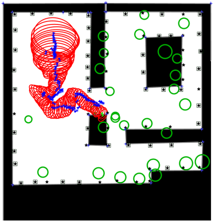
$$\max_{z^R \subset z} \quad H(L_f \mid z^R) \ \text{ s.t. } \quad |z^R| \leq C$$

Simulation

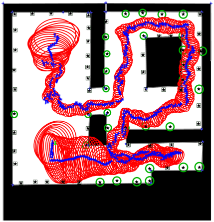
- Simulated, compared with select either feature or measurements only



measurements only



features only

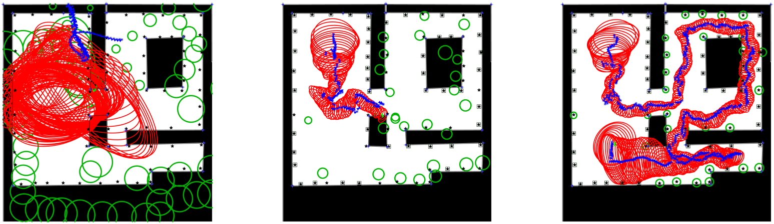


two stages

- Green circles: selected features with their size representing uncertainty
- Blue lines: robot nominal trajectories
- Red circles: robot pose uncertainty.
- Two-stage has less uncertainty in narrow passages and lower $\text{Prob}_{\text{collision}}$ compared to selecting feature or measurements separately

Simulation

- Simulated, compared with select either feature or measurements only

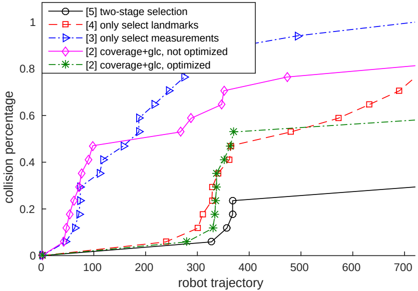


measurements only features only two stages

- Green circles: selected features with their size representing uncertainty
- Blue lines: robot nominal trajectories
- Red circles: robot pose uncertainty.
- Two-stage has less uncertainty in narrow passages and lower $\text{Prob}_{\text{collision}}$ compared to selecting feature or measurements separately

Performance Comparison

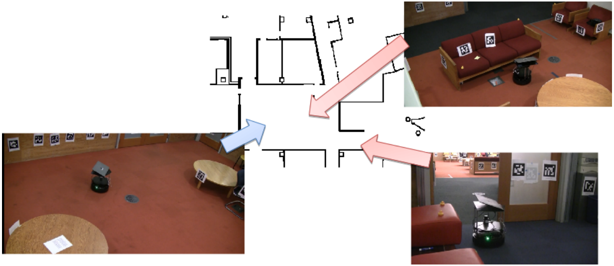
Approach	Number of Variables	Number of Edges	Mean Error on Features (m)
no reduction	2067	14753	**0.02**
GLC [12]	166	532	44.5
measurements only[2]	130	131	33.48
features only	100	143	4.88
Stages 1 and 2	**60**	**120**	**0.13**



Comparison of collision prob

Hardware

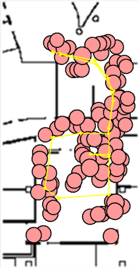
- Environment and robot



office environment, red(narrow), blue(wide)

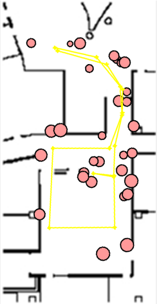
Hardware

- Measurements only



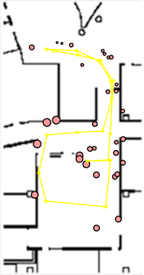
Hardware

- Features only

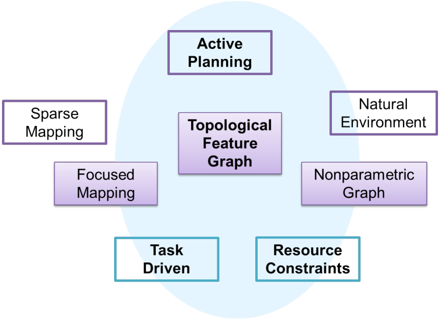


Hardware

- Both stages

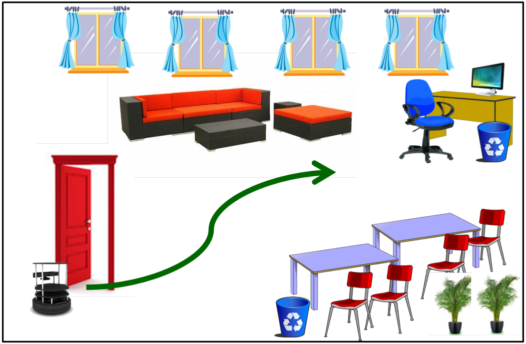


Outline



Active Planning
Sparse Mapping
Topological Feature Graph
Natural Environment
Focused Mapping
Nonparametric Graph
Task Driven
Resource Constraints

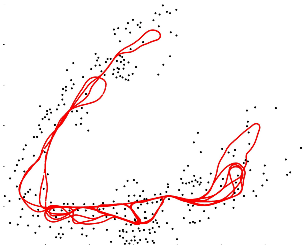
Active Localization and Mapping (Mu et al [13])



- **Question**: how can the robot plan own path while building a map?

Active Localization and Mapping (Active-SLAM)

- **Factor graphs**
  - Scales well, robust to long-term drift
  - Easy to incorporate visual data, semantic labels, sensor cheap
  - Hard to check path feasibility, thus typically not used for path planning



feature graph

- **Occupancy grid map**[3, 4, 14–16]
  - Easy to check path feasibility
  - Does not scale well, not robust, computation heavy, expensive sensor

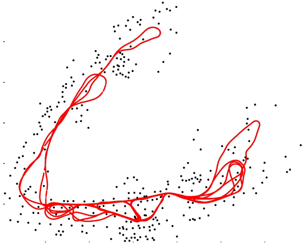
occupancy grid map

- **Gap**: Possible to maintain advantages of factor graphs, but how enable fast path feasibility check?

Active Localization and Mapping (Active-SLAM)

▶ Factor graphs

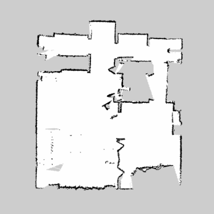
- Scales well, robust to long-term drift
- Easy to incorporate visual data, semantic labels, sensor cheap
- Hard to check path feasibility, thus typically not used for path planning



feature graph

▶ Occupancy grid map[3, 4, 14–16]

- Easy to check path feasibility
- Does not scale well, not robust, computation heavy, expensive sensor

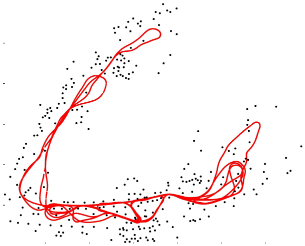


occupancy grid map

▶ **Gap**: Possible to maintain advantages of factor graphs, but how enable fast path feasibility check?

Active Localization and Mapping (Active-SLAM)

- **Factor graphs**
  - Scales well, robust to long-term drift
  - Easy to incorporate visual data, semantic labels, sensor cheap
  - Hard to check path feasibility, thus typically not used for path planning
- **Occupancy grid map**[3, 4, 14–16]
  - Easy to check path feasibility
  - Does not scale well, not robust, computation heavy, expensive sensor
- **Gap**: Possible to maintain advantages of factor graphs, but how enable fast path feasibility check?



feature graph

occupancy grid map

Topological Feature Graph (TFG)

- **Observations**
  - Features usually lie on obstacle surfaces
  - Can connect features to represent geometry information
- **Topological Feature graphs**

$$\mathcal{G}(V, E)$$

  - V, vertices, represent features
  - E, edges, represent obstacle surfaces
- **Approach**
  - RGB image $\Rightarrow$ features
  - Depth image $\Rightarrow$ obstacle surfaces $\Rightarrow$ edges between two features

Topological Feature Graph (TFG)

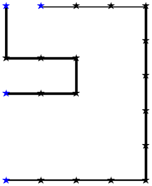
▶ Observations

- Features usually lie on obstacle surfaces
- Can connect features to represent geometry information

▶ Topological Feature graphs

$$\mathcal{G}(V, E)$$

- V, vertices, represent features
- E, edges, represent obstacle surfaces



Topological Feature Graph

▶ Approach

- RGB image $\Rightarrow$ features
- Depth image $\Rightarrow$ obstacle surfaces $\Rightarrow$ edges between two features

Topological Feature Graph (TFG)

Observations

- Features usually lie on obstacle surfaces
- Can connect features to represent geometry information

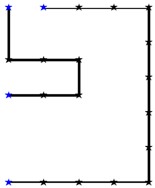
Topological Feature graphs

$$\mathcal{G}(V, E)$$

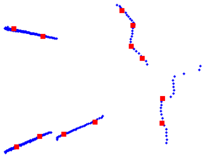
- V, vertices, represent features
- E, edges, represent obstacle surfaces

Approach

- RGB image $\Rightarrow$ features
- Depth image $\Rightarrow$ obstacle surfaces $\Rightarrow$ edges between two features



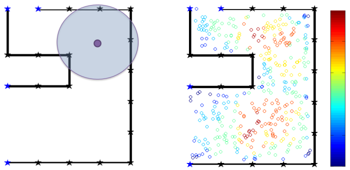
Topological Feature Graph



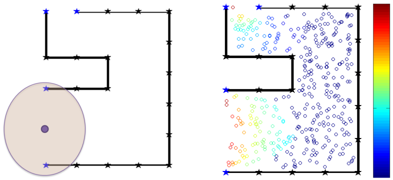
Segment depth

Path Planning for Information Gathering

- **Problem**: given a partial map, where should the robot go next to gather more information?



Exploit

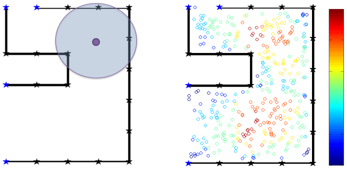


Explore

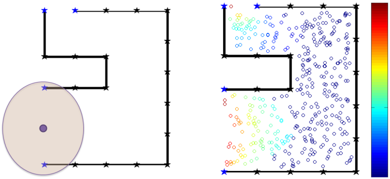
- **Path Planning** sample space and plan paths
  - $\text{Info}_{\text{total}} = \text{Info}_{\text{explore}} + \text{Info}_{\text{exploit}}$
  - **Goal point**: max $\text{Info}_{\text{total}}$
  - **Planner**: Probabilistic Roadmap (PRM)[17]
  - **Controller**: pure-pursuit

Path Planning for Information Gathering

- **Problem**: given a partial map, where should the robot go next to gather more information?

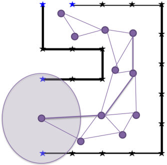


Exploit



Explore

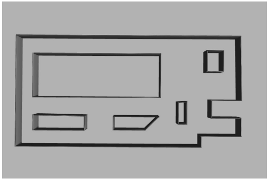
- **Path Planning**
  - $\text{Info}_{\text{total}} = \text{Info}_{\text{explore}} + \text{Info}_{\text{exploit}}$
  - **Goal point**: max $\text{Info}_{\text{total}}$
  - **Planner**: Probabilistic Roadmap (PRM)[17]
  - **Controller**: pure-pursuit



sample space and plan paths

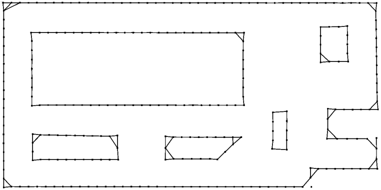
Simulations – Gazebo

- Simulation environment: 46m×22m
  - Gazebo, Turtlebot
  - Depth camera ⇒ laser; Fake AprilTag measurements
- Mapping Results



Gazebo environment

Frontier grid map



TFG

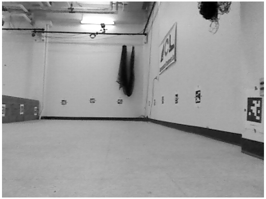
Simulations – Gazebo

- Comparison

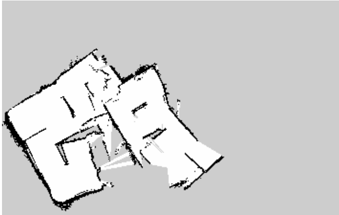
Simulation Performance Comparison

	TFG Active SLAM	grid map frontier
No. of variables	274x6	800000
CPU idle time	75%	0%
time (s)	2433 ± 546	2293 ± 375
position error (m)	0.147 ± 0.115	5.26 ± 3.53
orientation error (rad)	0.0217 ± 0.016	0.0213 ± 0.0165

Hardware Experiments – Indoors



▶ Result comparison

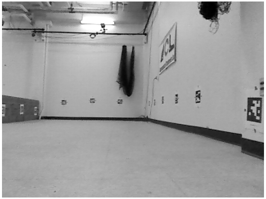


Grid Map

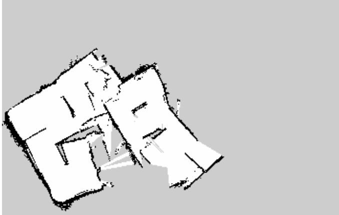
Topological Feature Graph

Hardware Experiments – Indoors

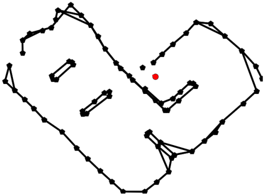




▶ Result comparison

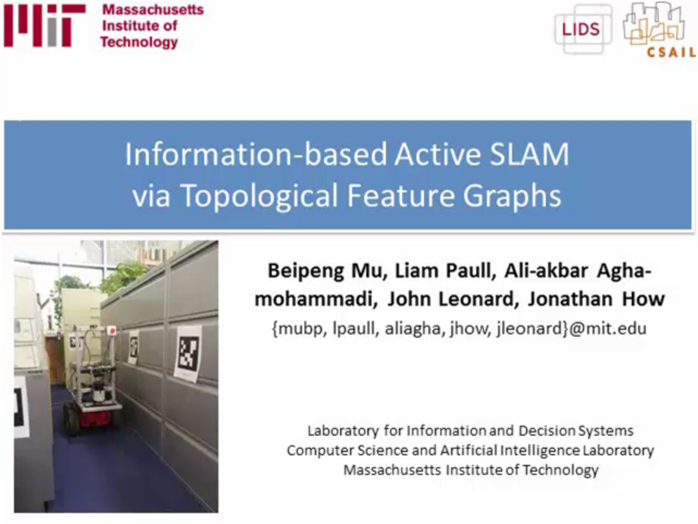


Grid Map



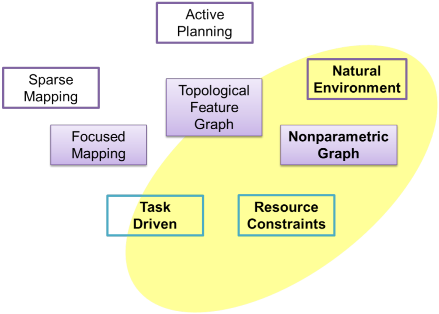
Topological Feature Graph

Hardware Experiments



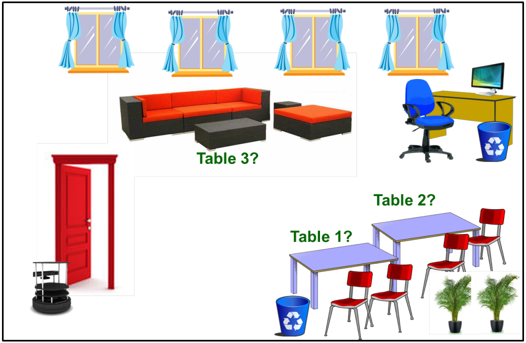
Outline

1. Motivation
2. Focused Mapping for Collision-free Navigation
3. Active Mapping
4. Object SLAM



Object SLAM with Nonparametric Pose Graph (Mu et al [18])

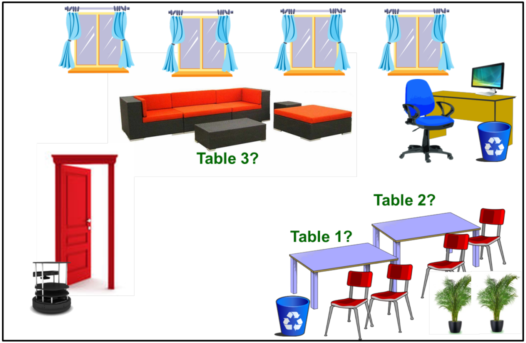
- **Interact with natual environment**:



- **Challenge: perception**
  - Data association
  - False positives

Object SLAM with Nonparametric Pose Graph (Mu et al [18])

- **Interact with natual environment**:



- **Challenge: perception**
  - Data association
  - False positives

Object SLAM with Nonparametric Pose Graph

- **What perception system to use?**

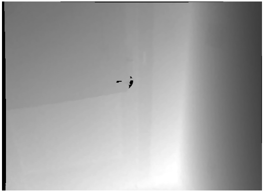
- **Perception**:

- **Object** Abundance; Real-time possibility; Semantic meanings

- **Object Detection**: SIFT matching, HOG, CNN

Object SLAM with Nonparametric Pose Graph

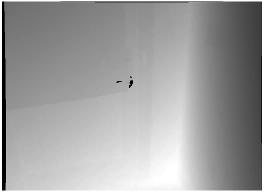
- **What perception system to use?**

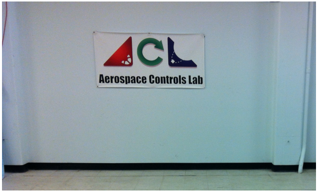


- **Perception**:
- **Object** Abundance; Real-time possibility; Semantic meanings
- **Object Detection**: SIFT matching, HOG, CNN

Object SLAM with Nonparametric Pose Graph

- **What perception system to use?**





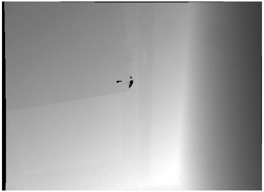
- **Perception**:

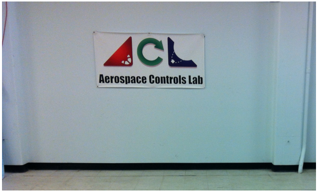
- **Object** Abundance; Real-time possibility; Semantic meanings

- **Object Detection**: SIFT matching, HOG, CNN

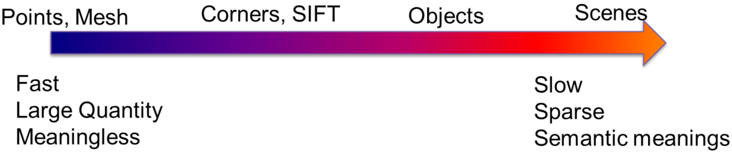
Object SLAM with Nonparametric Pose Graph

- **What perception system to use?**





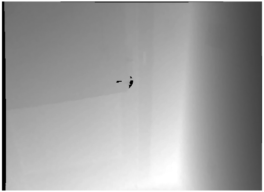
- **Perception**:

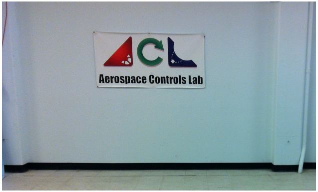


- **Object** Abundance; Real-time possibility; Semantic meanings
- **Object Detection**: SIFT matching, HOG, CNN

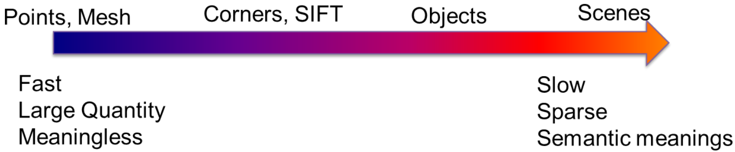
Object SLAM with Nonparametric Pose Graph

- **What perception system to use?**





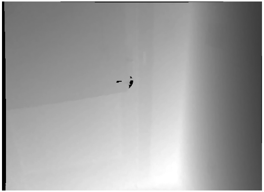
- **Perception**:

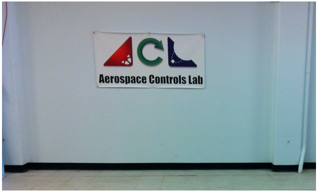


- **Object** Abundance; Real-time possibility; Semantic meanings
- **Object Detection**: SIFT matching, HOG, CNN

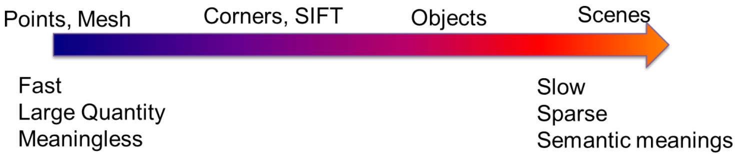
Object SLAM with Nonparametric Pose Graph

- **What perception system to use?**





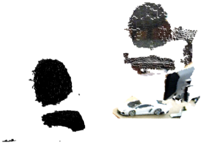
- **Perception**:



- **Object** Abundance; Real-time possibility; Semantic meanings
- **Object Detection**: SIFT matching, HOG, CNN

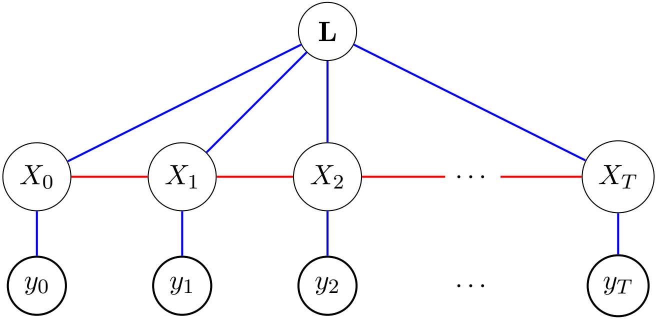
State-of-art Object Detection

- **Deep Learning**:
  - Data availability; high accuracy; generalize well to different object classes
- **Faster RCNN** [19], train region-proposals at the same time
- 3D Object Measurement
  - Detect object in RGB ⇒ crop bounding box in depth ⇒ 3D center



Factor Graph with Imperfect Data Association

- Factor Graph
  - Variables: poses X objects L
  - Measurement: $z_t = L_{y_t} \ominus X_t + w, \quad w \sim \mathcal{N}(0, R)$
- Imperfect data association
  - Data association y_t unknown
  - Number of objects unknown



Nonparametric Pose Graph

- Objects of different classes
  - $L = \{L_1, \cdots, L_M\}$, M objects
  - $\pi_i = \{\pi_i(0), \pi_i(1), \ldots, \pi_i(N)\}$, probability object i in $\{1, \ldots, N\}$ classes
  - $\pi_i(0)$ represents false positive

- **Measurement Model**
  - $\{z_t^k, u_t^k\}$, 3D measurement and observed object class
  - $z_t = L_{y_t} \ominus X_t + w, \quad w \sim \mathcal{N}(0, R)$
  - $u_t^k \sim \pi_i$ follows a categorical distribution

- **Data association**
  - Number of objects unknown ahead of time
  - y_t prior: Dirichlet process

$$p(y_t = m) \propto \begin{cases} \text{Number}(y_{0:t-1} = m) & m \leq M \\ \alpha & m = M + 1 \end{cases} \tag{2}$$

Inference Algorithms

- Initialization
  - Based on odometry

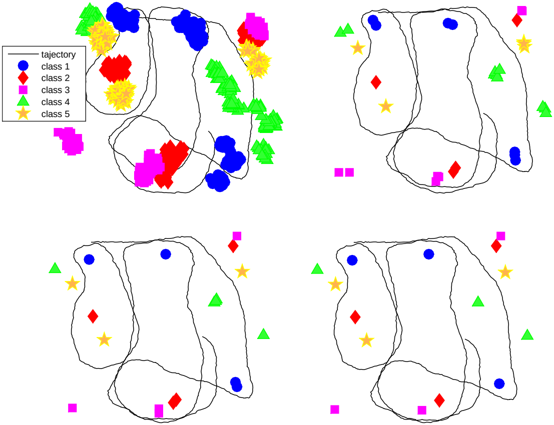
(E) **Expectation** Fix association y, optimize poses X, L
  - Standard SLAM

(M) **Maximization** Fix poses X, L, optimize association y

- Repeat E and M until stop criteria met

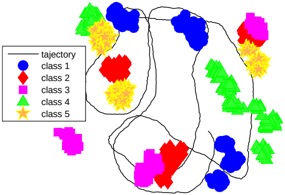
Simulation

- Results

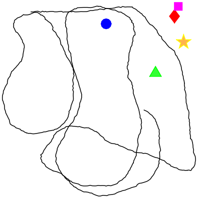


Result of nonparametric pose graph at iterations 0, 1, 2, 3

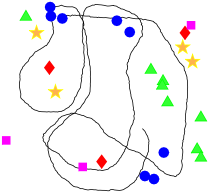
Simulation: comparison



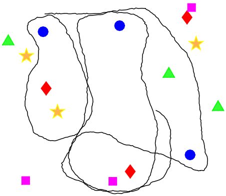
Frame by Frame



Robust SLAM (R-SLAM) [20]

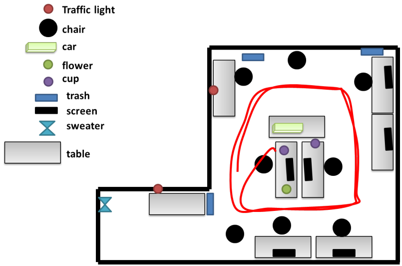


Open-loop Object Detection (OL) [10]

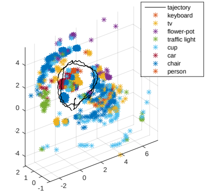


Nonparametric Pose Graph

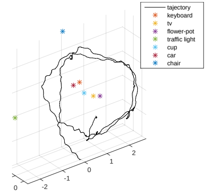
Office Dataset



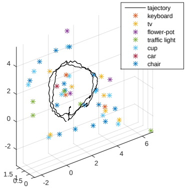
Floor Map



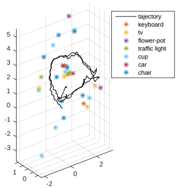
Frame by Frame



Robust SLAM (R-SLAM)

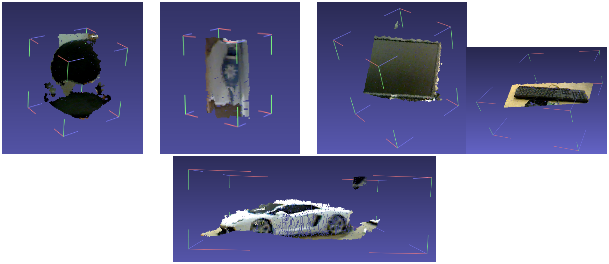


Open-loop



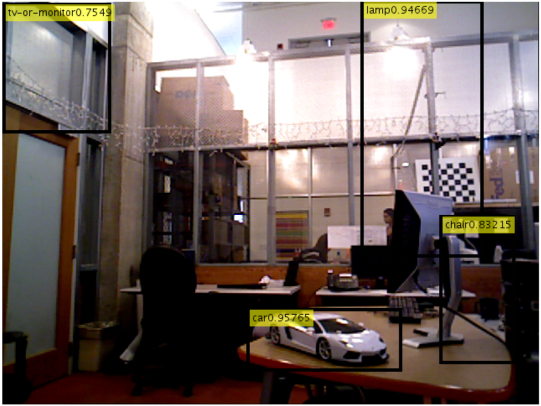
Nonparametric Pose Graph

Office Dataset



Example of detected objects. From left to right, top to down are chair, sweater in the corner, screen, keyboard and toy car.

Office Dataset

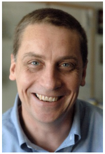


Contributions

- **Evaluate information** in sensing data by their relevance to the task of navigation, develop a **focused mapping technique** that can build compact models.
- Proposed **Topological Feature Graph** that exploits sparsity but also incorporate obstacle representations.
  - Resulting **active mapping** algorithm autonomously plan paths to build maps of unknown environments.
- Created **nonparametric graphical model**/inference algorithm to associate ambiguous object detections and localize detected objects at the same time.
- Extensive simulations and real-world experiments
  - Results show proposed algorithms have **good task performance** with **significant less resources** than prior SOA.

Acknowledgments

- Committee



Jonathan P. How

John J. Leonard

Sertac Karaman

John W. Fisher

- Collaborators
  - Liam Paull, Ali Apha, Shih-Yuan Liu, Matthew Graham, Matthew Giamou
- Aerospace Controls Lab., Marine Robotics Group
- LIDS, CSAIL
- Friends and families

References I

[1] N. Carlevaris-Bianco and R. Eustice. Conservative edge sparsification for graph SLAM node removal. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 854–860, May 2014.

[2] V. Ila, J.M. Porta, and J. Andrade-Cetto. Information-based compact pose SLAM. *Robotics, IEEE Transactions on*, 26(1):78–93, Feb 2010.

[3] C. Stachniss, G. Grisetti, and W. Burgard. Information gain-based exploration using rao-blackwellized particle filters. In *Proc. of Robotics: Science and Systems (RSS)*, Cambridge, MA, USA, 2005.

[4] Benjamin Charrow, Gregory Kahn, Sachin Patil, Sikang Liu, Ken Goldberg, Pieter Abbeel, Nathan Michael, and Vijay Kumar. Information-theoretic planning with trajectory optimization for dense 3d mapping. In *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015.

[5] B. Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, CIRA '97, pages 146–, Washington, DC, USA, 1997. IEEE Computer Society.

[6] Matan Keidar, Eran Sadeh-Or, and Gal A Kaminka. Fast frontier detection for robot exploration. In *Advanced Agent Technology*, pages 281–294. Springer, 2011.

[7] Shuran Song, Linguang Zhang, and Jianxiong Xiao. Robot in a room: Toward perfect object recognition in closed environments. *CoRR*, abs/1507.02703, 2015.

[8] J. Civera, D. Galvez-Lopez, L. Riazuelo, J.D. Tardos, and J.M.M. Montiel. Towards semantic SLAM using a monocular camera. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 1277–1284, Sept 2011.

[9] R.F. Salas-Moreno, R.A. Newcombe, H. Strasdat, P.H.J. Kelly, and A.J. Davison. SLAM++: Simultaneous localisation and mapping at the level of objects. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1352–1359, June 2013.

[10] Sudeep Pillai and John Leonard. Monocular SLAM supported object recognition. In *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015.

[11] B. Mu, A. Agha-mohammadi, L. Paull, M. Graham, J. How, and J. Leonard. Two-stage focused inference for resource-constrained collision-free navigation. In *Robotics Science and Systems*, Rome, Italy, July 2015.

[12] N. Carlevaris-Bianco and R.M. Eustice. Generic factor-based node marginalization and edge sparsification for pose-graph SLAM. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 5748–5755, May 2013.

References II

[13] Beipeng Mu, Matthew Giamou, Liam Paull, John Leonard, and Jonathan P. How. Information-based active slam via topological feature graphs. In *IEEE Conference on Decision and Control (CDC)*, December 2016.

[14] H. Carrillo, I. Reid, and J.A. Castellanos. On the comparison of uncertainty criteria for active SLAM. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 2080–2087, May 2012.

[15] L. Carlone, Jingjing Du, M.K. Ng, B. Bona, and M. Indri. An application of Kullback-Leibler divergence to active SLAM and exploration with particle filters. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 287–293, Oct 2010.

[16] J. Vallv and J. Andrade-Cetto. Active Pose SLAM with RRT*. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, May 2015.

[17] J. C. Latombe L. E. Kavraki, P. Svestka and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, Jun. 1996.

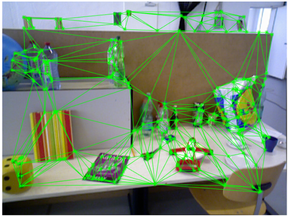
[18] Beipeng Mu, Shih-Yuan Liu, Liam Paull, John Leonard, and Jonathan P. How. Slam with objects using a nonparametric pose graph. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, Oct 2016(submitted).

[19] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

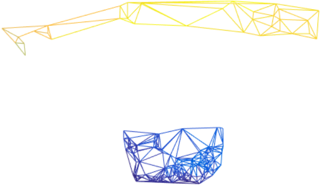
[20] M.C. Graham, J.P. How, and D.E. Gustafson. Robust incremental SLAM with consistency-checking. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 117–124, Sept 2015.

Future Work – Sparse 3D mapping

- Challenges
  - Real environments
  - Accurate state estimation
  - 3D obstacle representation
  - Cheap sensor and sparse models
- Visual inertial
  - IMU ⇒ odometry
  - Camera ⇒ visual features
- 3D topological feature graph
  - Vertices represent features
  - Triangulated surfaces represent obstacle

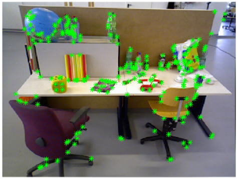


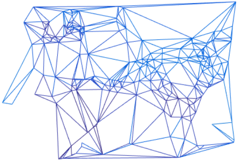
triangulated features



3D topological feature graph

Sparse 3D mapping





3D topological feature graph