Sampling-based Planning 02

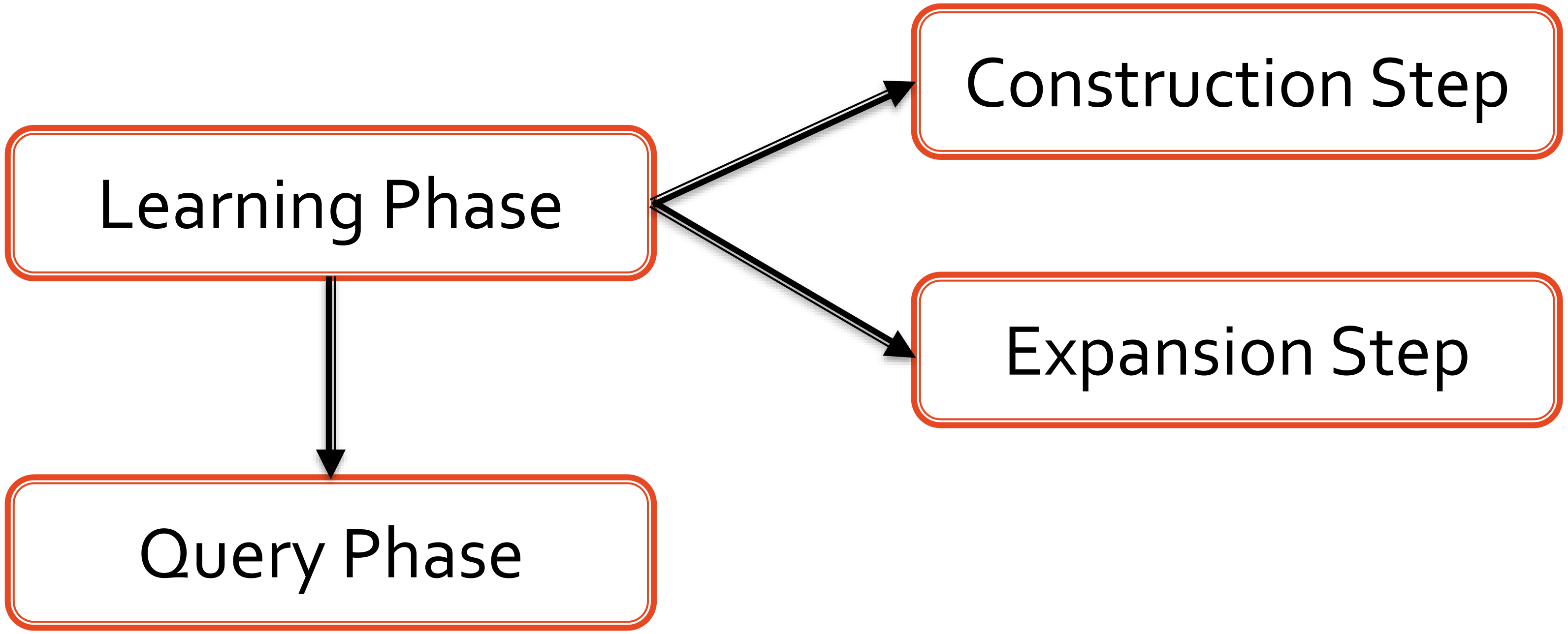
Jane Li

Assistant Professor
Mechanical Engineering Department, Robotic Engineering Program
Worcester Polytechnic Institute

Quiz (10 pts)

- (3 pts) Explain at high-level how to plan a path using PRM?
- (3 pts) What are the two popular ways to find nearest neighbor in PRM? And how to speed up your search?
- (4 pts) What heuristics can be used to guide expansion? List two, and explain why

Two-phase solution



- Environment remains unchanged
- Reuse roadmap for multi-query

Finding Nearest Neighbors (NN)

- Two popular ways to do NN in PRM
  - Find k nearest neighbors (even if they are distant)
  - Find all nearest neighbors within a certain distance

- Naive NN computation can be slow with thousands of nodes
  - use *kd-tree* to store nodes and do NN queries

Possible Heuristics

- # of Nodes nearby
  - For a node **c**, count the # of nodes **N** within a predefined distance
  - N is small → obstacle region may occupy large portion of **c**'s neighborhood
  - Use Heuristics = 1/**N** to guide random sampling

Possible Heuristics

- Distance to nearest reacheable neighbor
  - For a node **c**, find the distance **d** to the nearest connected component that doesn't contains this node
  - **d** is small → **c** lies in the region where two components fail to connect
  - Heuristics = 1/d

Possible Heuristics

- Others?

- Behavior of local planner?
  - Always fail to connect → difficult region

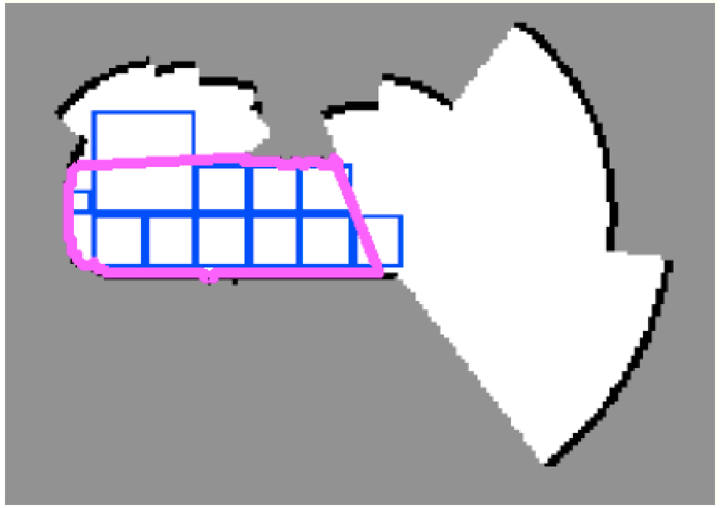
Advanced Roadmaps

Overview

- Sampling strategies
- Hierarchical roadmap

Motivating problem

- A mobile robot navigates in an unknown or partially known home environment
  - 2D navigation
  - Focus on efficient expansion and reconstruction of roadmap



How to represent the traversable areas?

- Cell decomposition (e.g., grid) – commonly used
- Pros
  - Complete coverage
  - Regular space division
  - Can search for a path as new traversable area is explored
- Cons
  - Sensitive to dimensionality

How to represent the traversable areas?

- Roadmap
  - Construct map based on samples in traversable areas
- Pros
  - Reduced computational complexity
- Cons
  - Unknown or partially known environment → Need to expand or reconstruct map
  - Random sampling → fail to cover narrow passages.

How to fix the problem?

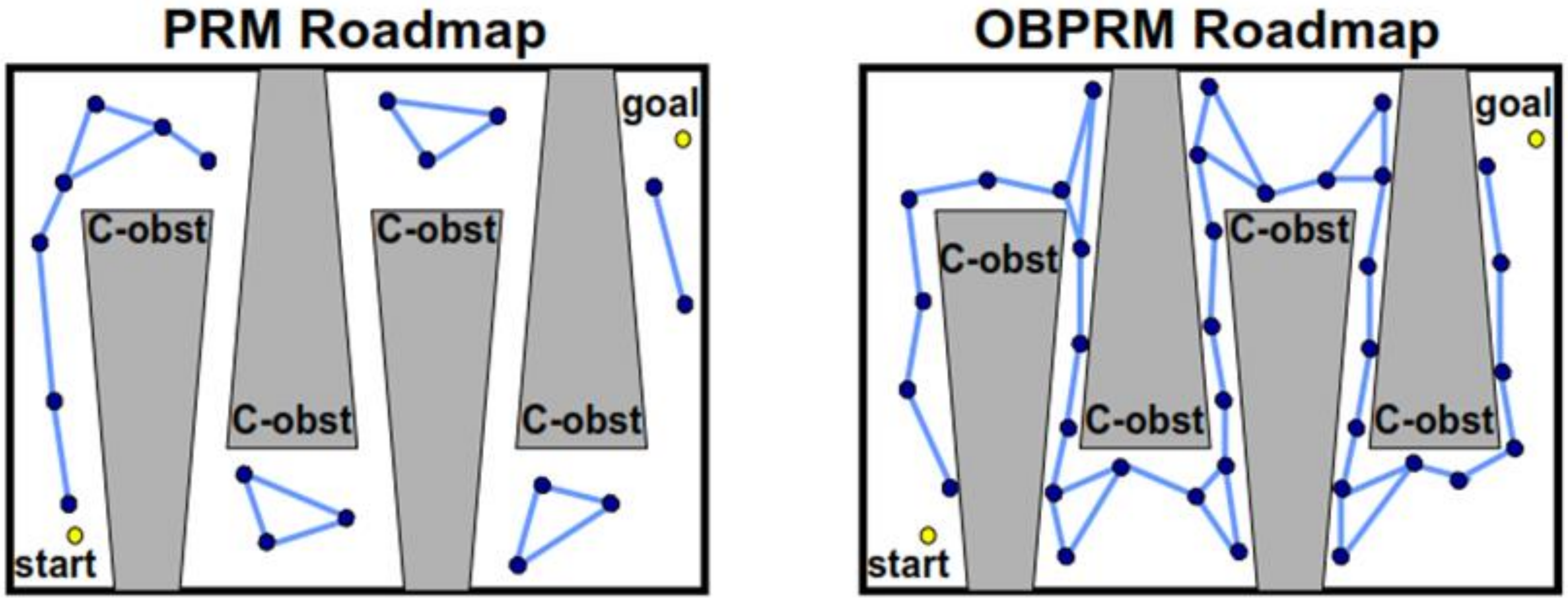
- Random sampling fails to cover the whole traversable areas?
  - Open area
  - Narrow passages

Other sampling strategies?

Obstacle-based PRM

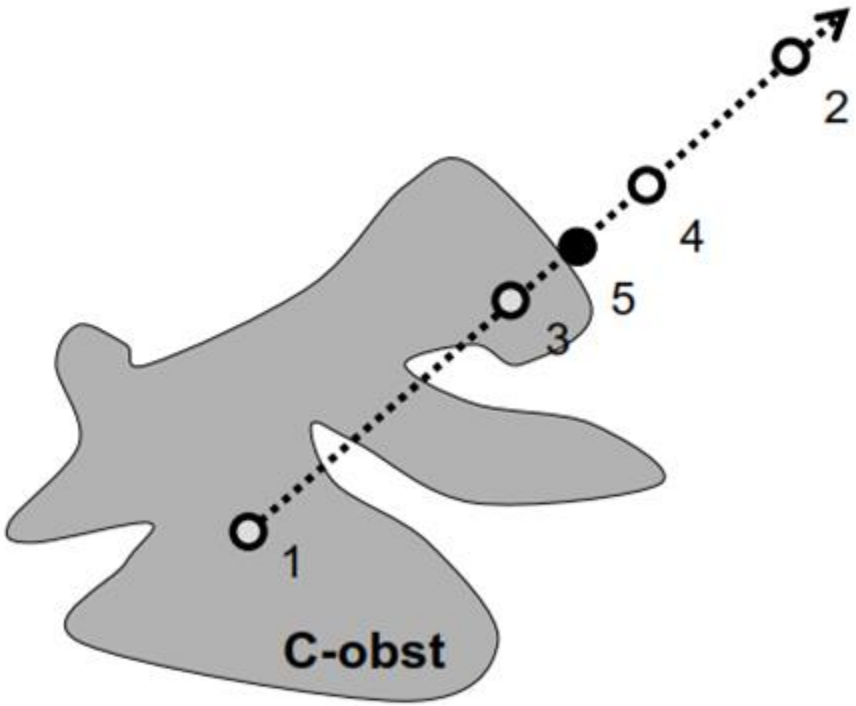
Can you explicitly construct C-obstacles?

- To navigate a narrow passage
  - Need more sample points for where planning is hard
  - Sample near C-obstacles?

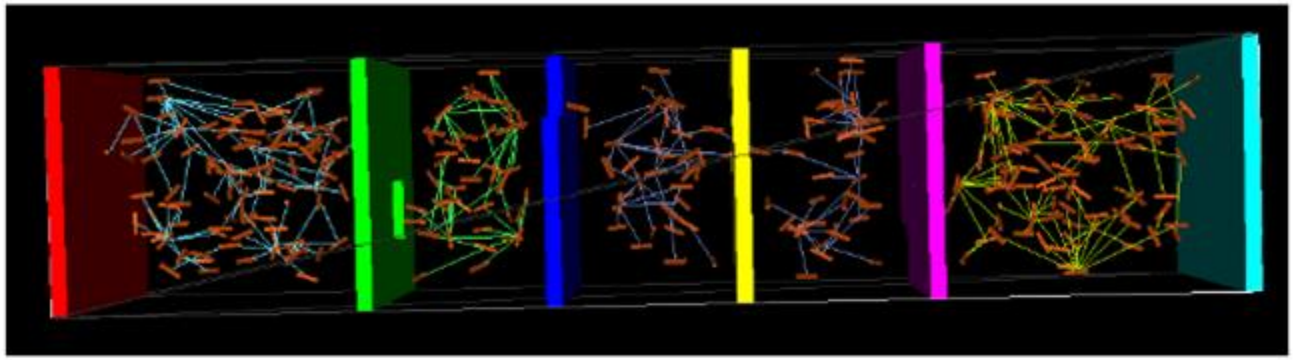


Obstacle-based PRM

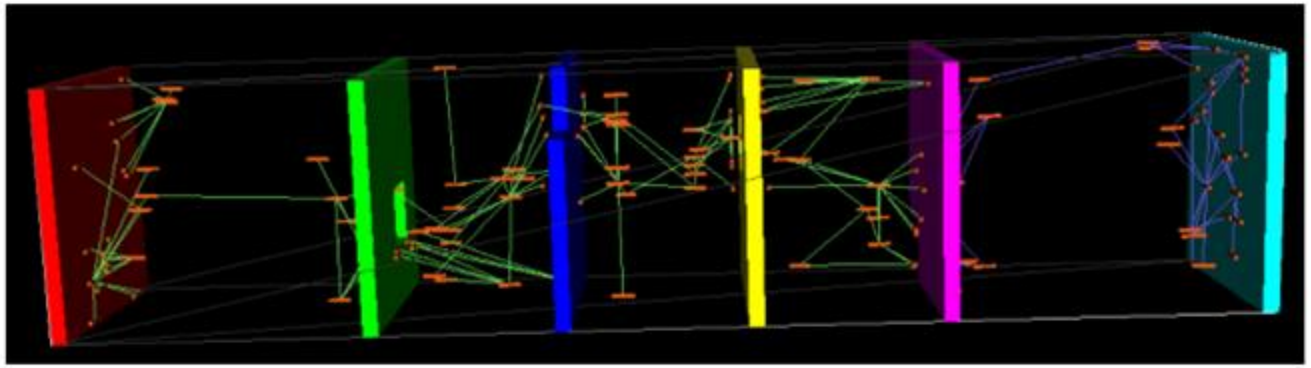
- How to find points on C-obstacles?
  - Find a point in the C-obstacles – a collision configuration
  - Select a random direction in C-space
  - Find a free point in that direction
  - Find the boundary point between then using binary search



PRM VS OBPRM



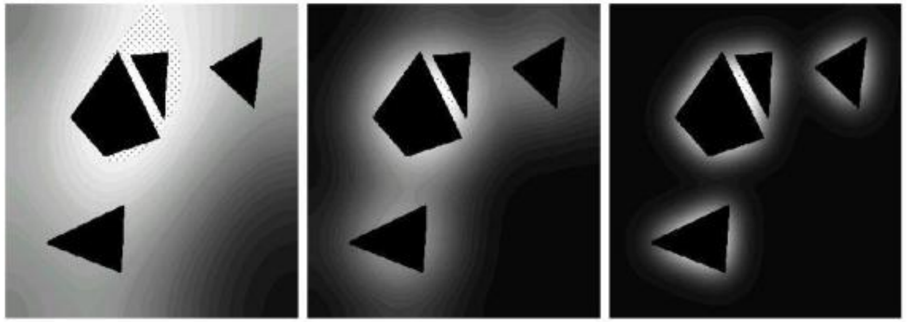
PRM
- 328 nodes
- 4 major CCs



OBPRM
- 161 nodes
- 2 major CCs

Gaussian Sampling [1]

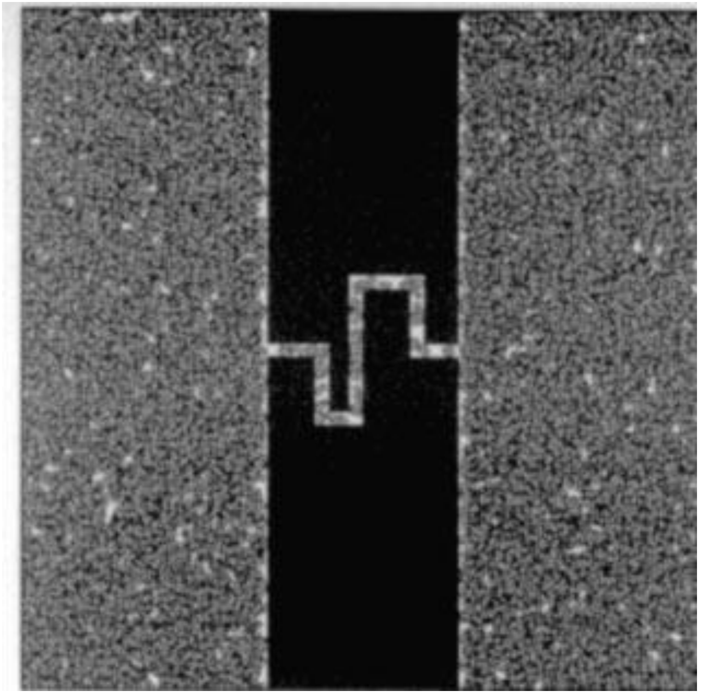
- Gaussian sampler
  - Find a q_1
  - Pick a q_2 from a Gaussian distribution centered at q_1
  - If: **both** are in collision or collision-free, discard them
  - Else: keep the collision-free one



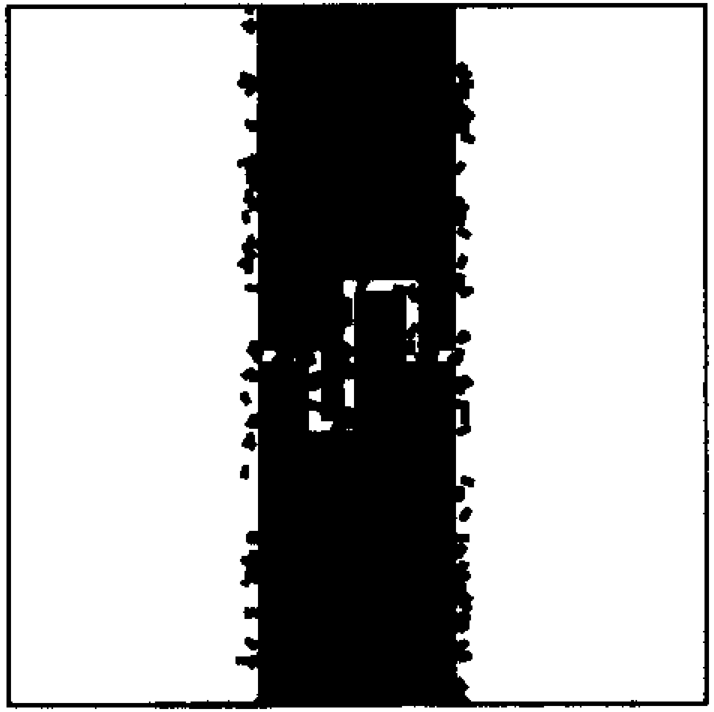
Sampling distribution for varying Gaussian width (width decreasing from left to right) [1]

Gaussian Sampling

The gain is not in sampling fewer milestones, but in connecting fewer pairs of milestones



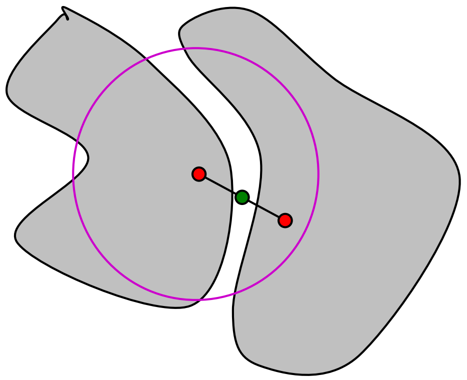
Milestones (13,000) created by uniform sampling before the narrow passage was adequately sampled



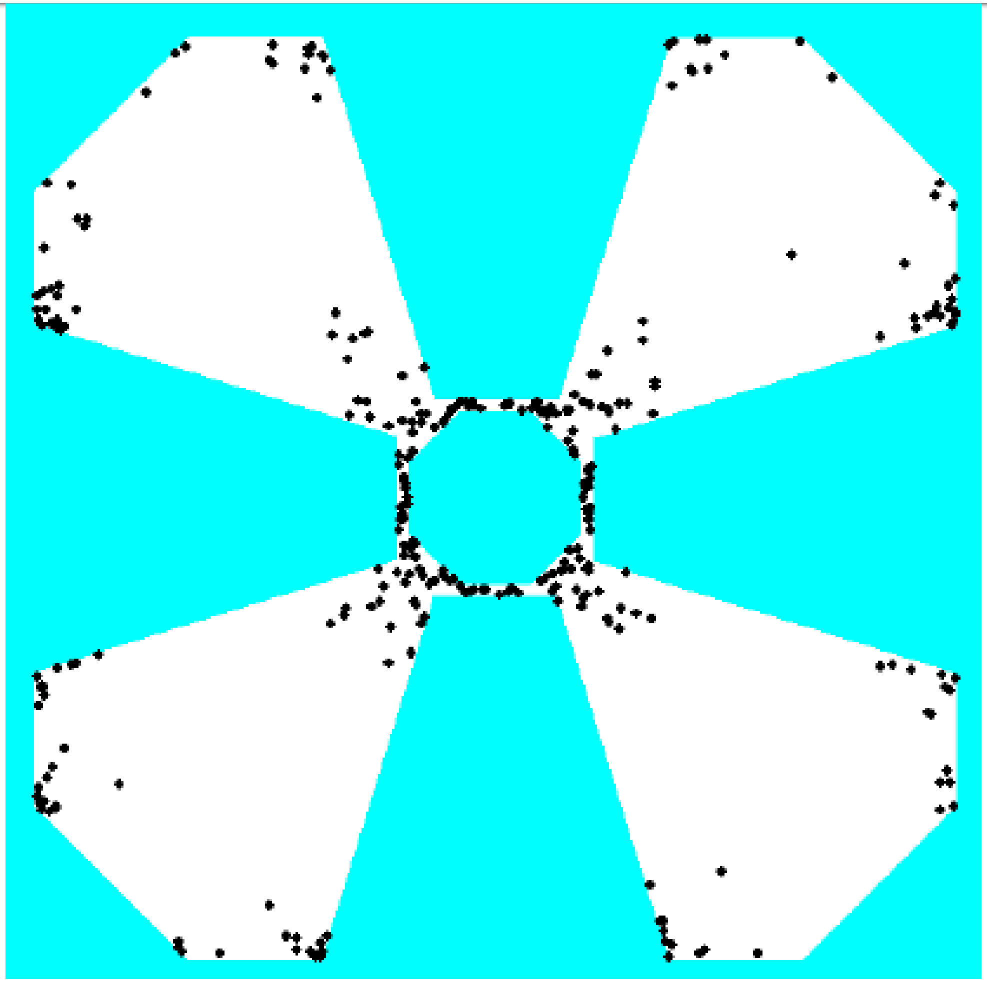
Milestones (150) created by Gaussian sampling

Bridge sampling [2]

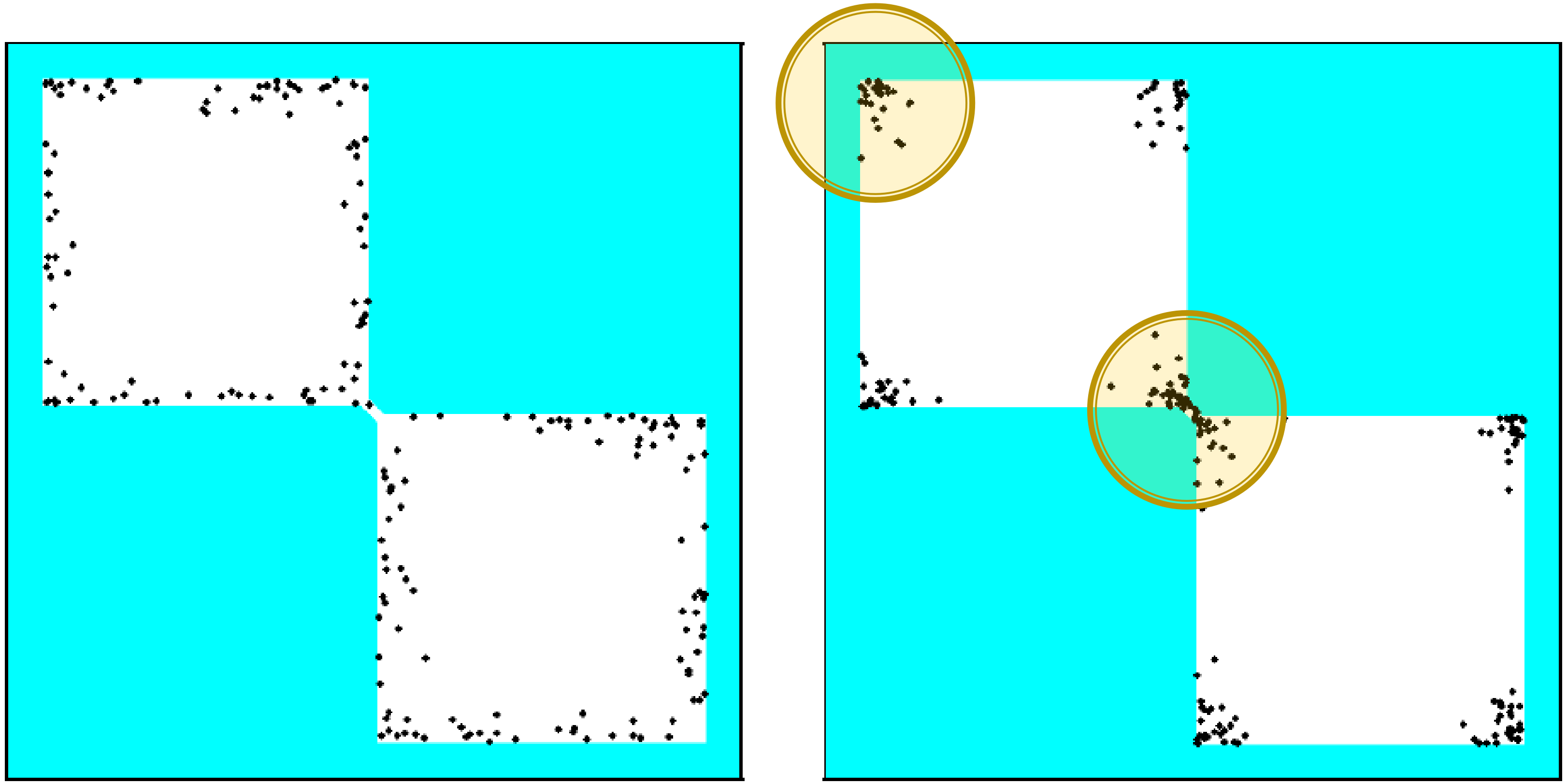
- Bridge sampler
  - Sample a q_1 that is in collision
  - Sample a q_2 in neighborhood of q_1 using some probability distribution (e.g. Gaussian)
  - If q_2 in collision, get the midpoint of (q_1, q_2)
  - Check if midpoint is in collision, if not, add it as a node



Bridge sampling

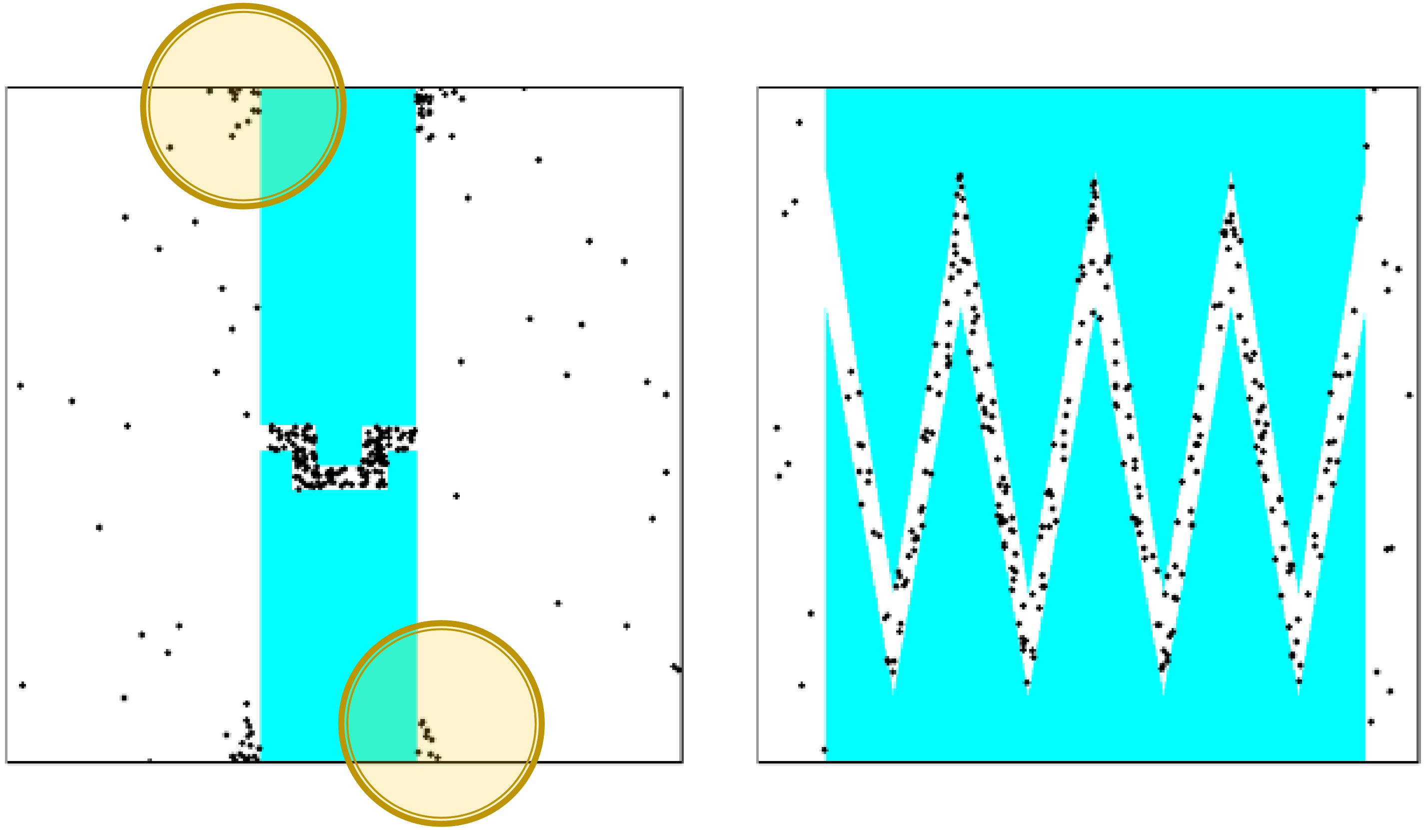


Bridge vs Gaussian



Gaussian | Bridge test

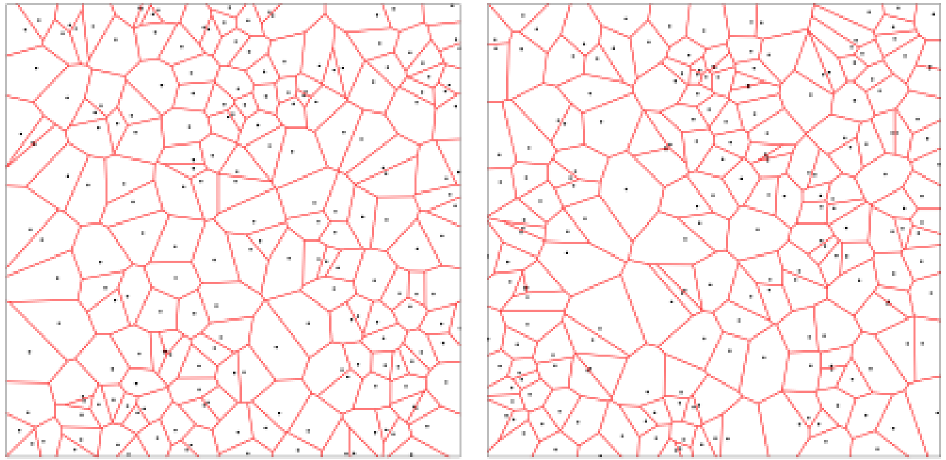
Bridge sampling



Bridge Sampling performs well in narrow passages

Deterministic Sampling

- Random sampling (biased or not) can be unpredictable and irregular
  - Each time your run your algorithm you get a **different sequence of samples,** so **performance varies**
  - In the limit, space will be sampled well, but in **finite time result may be irregular**



(a) 196 pseudorandom samples (b) 196 pseudorandom samples

Figure 5.3: Irregularity in a collection of (pseudo)random samples can be nicely observed with Voronoi diagrams.

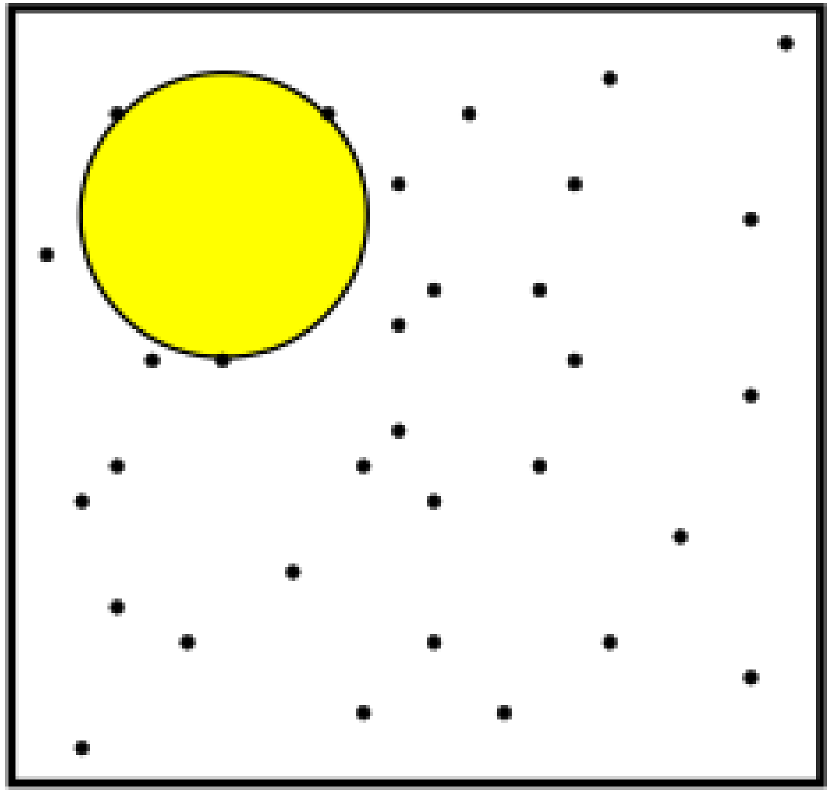
Deterministic Sampling

- What do we care about?
  - Dispersion

$$\delta(P) = \sup_{x \in X} \left\{ \min_{p \in P} \{\rho(x,p)\} \right\}.$$

P is a finite set of points, (X, ρ) is a metric space (ρ is a distance metric), which is the radius of the largest empty ball

  - What does it mean?
    - Intuitively, the dispersion quantifies <u>how well a space is covered by a set of points S in terms of the largest open Euclidean ball that touches none of the points.</u>

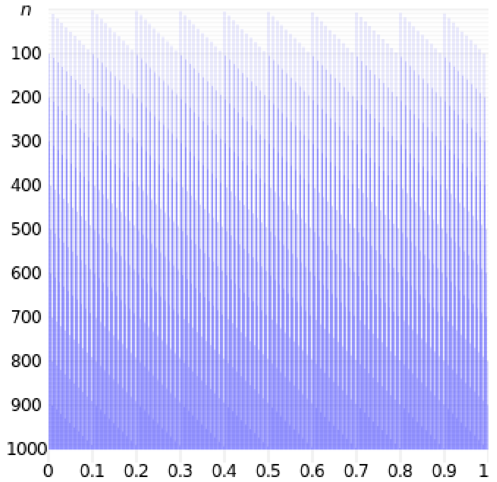


Quasi-random sampling

- Use **quasi-random** to replace random sampling
  - Deterministic sequence of **equivalent in dispersion**
  - Consistent performance
  - E.g., Van der Corput sequence (for base = 10)

$$\left\{\frac{1}{10}, \frac{2}{10}, \frac{3}{10}, \frac{4}{10}, \frac{5}{10}, \frac{6}{10}, \frac{7}{10}, \frac{8}{10}, \frac{9}{10}, \frac{1}{100}, \frac{11}{100}, \frac{21}{100}, \frac{31}{100}, \frac{41}{100}, \frac{51}{100}, \frac{61}{100}, \frac{71}{100}, \frac{81}{100}, \frac{91}{100}, \frac{2}{100}, \frac{12}{100}, \frac{22}{100}, \frac{32}{100}, \ldots\right\},$$

i	Naive Sequence	Binary	Reverse Binary	Van der Corput	Points in $[0,1]/\sim$
1	0	.0000	.0000	0	
2	1/16	.0001	.1000	1/2	
3	1/8	.0010	.0100	1/4	
4	3/16	.0011	.1100	3/4	
5	1/4	.0100	.0010	1/8	
6	5/16	.0101	.1010	5/8	
7	3/8	.0110	.0110	3/8	
8	7/16	.0111	.1110	7/8	
9	1/2	.1000	.0001	1/16	
10	9/16	.1001	.1001	9/16	
11	5/8	.1010	.0101	5/16	
12	11/16	.1011	.1101	13/16	
13	3/4	.1100	.0011	3/16	
14	13/16	.1101	.1011	11/16	
15	7/8	.1110	.0111	7/16	
16	15/16	.1111	.1111	15/16	

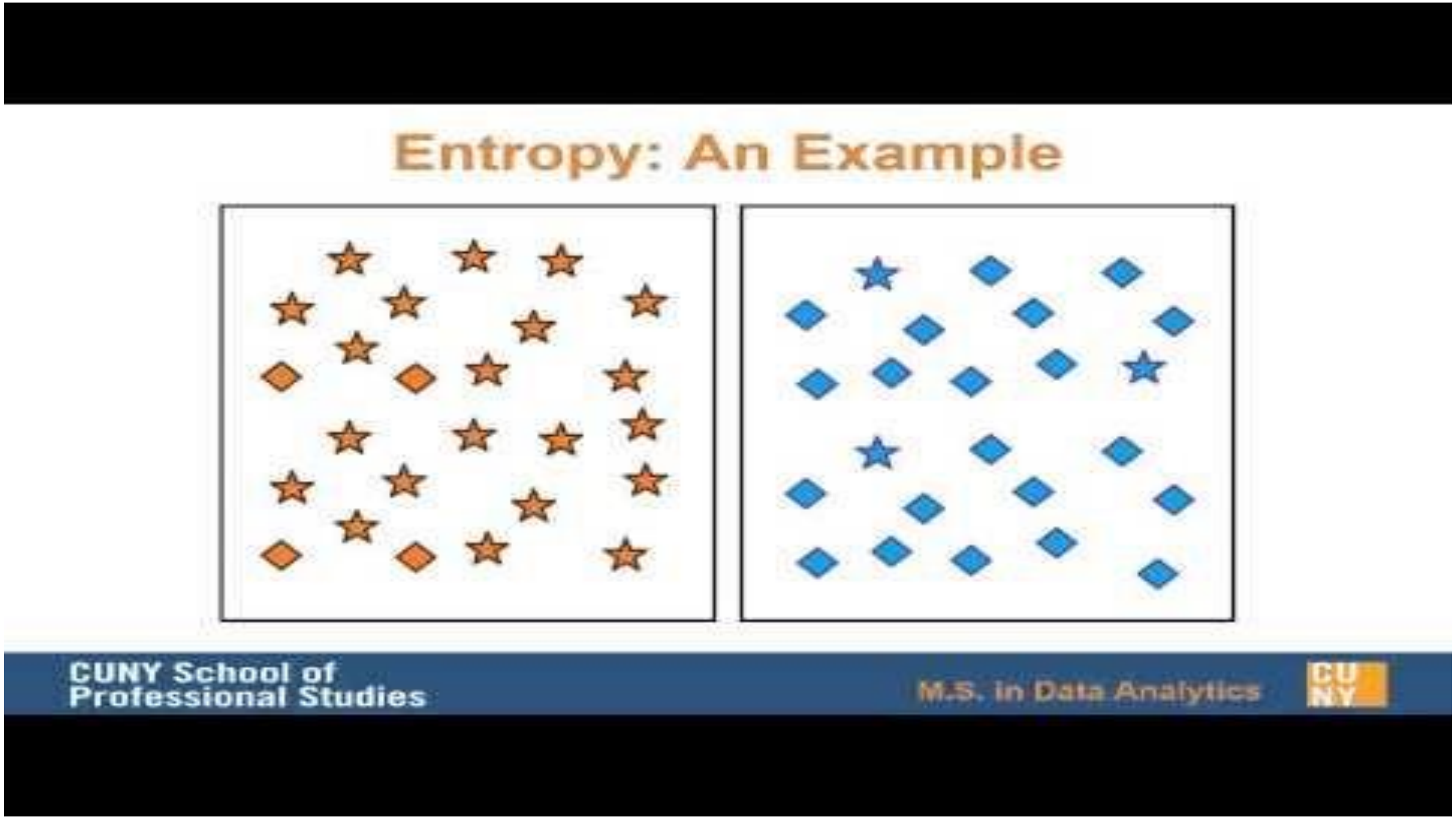


$$\boxed{g_b(n) = \sum_{k=0}^{L-1} d_k(n) b^{-k-1}}$$

Adaptive sampling [3]

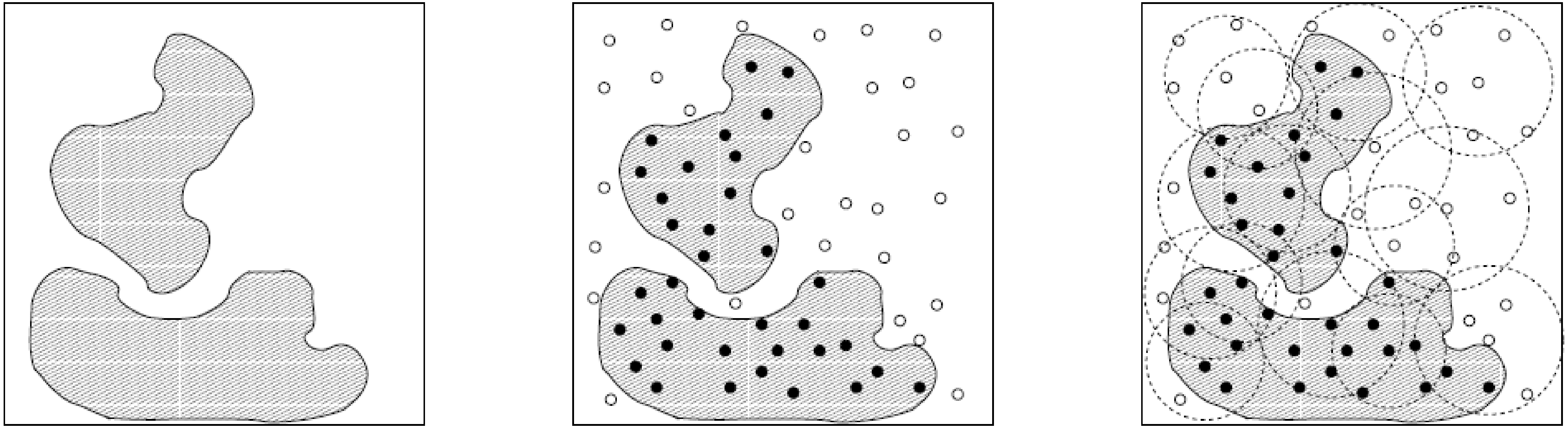
- Region-Sensitive Adaptive Motion Planner (RESAMPL)
- Main idea
  - Classify regions based on the ***entropy*** of the samples in it
  - Uses the classification to further refine the sampling

A brief introdEntropy



Region Construction

- Region construction



(a) C-space (b) Initial Sampling (c) Region Construction

Region construction

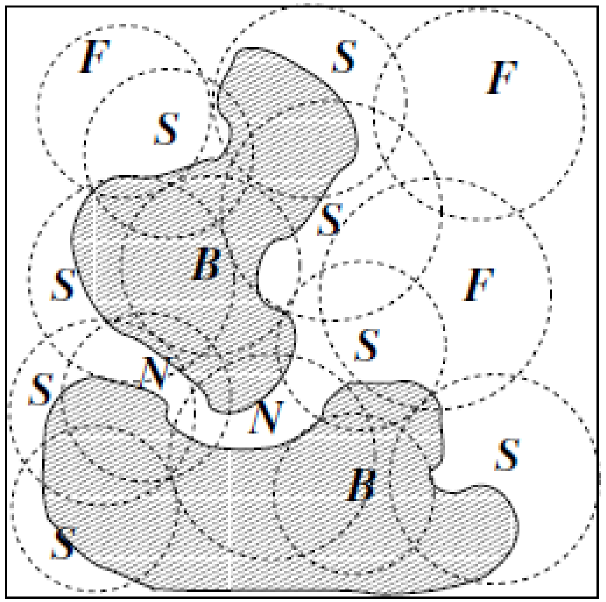
Algorithm 3.1 Region Construction

Require: Model $\mathcal{M}$, initial samples S, and k.

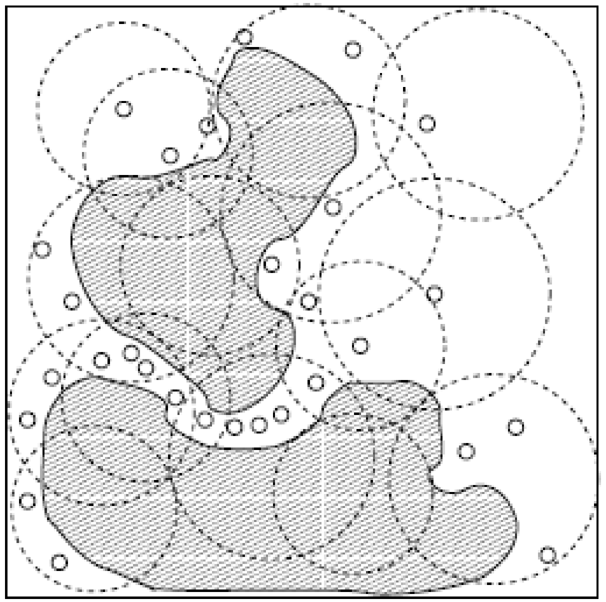
```
while there exists an unmarked sample in S do
    Let c be a randomly selected unmarked sample ∈ S.
    Set N = {k nearest neighbors to c}.
    Set R = a new region with center c and neighbors N.
    Add R to M.
    Flag c and N as marked.
end while
return M
```

Adaptive sampling

- Re-sampling based on region classification
  - Use entropy to measure the percentage of free/blocked pointed

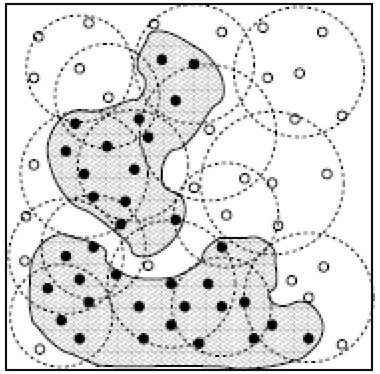


(d) Region Classification



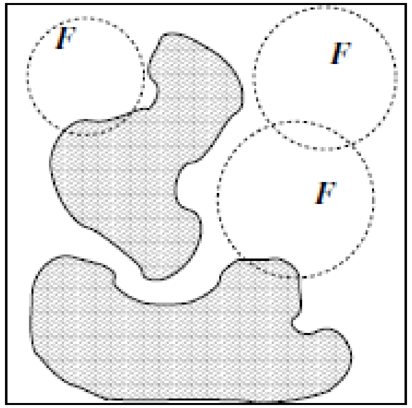
(e) Resulting Samples

Region classification

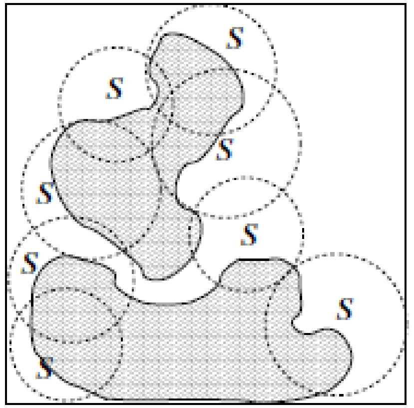


(a) Region Construction

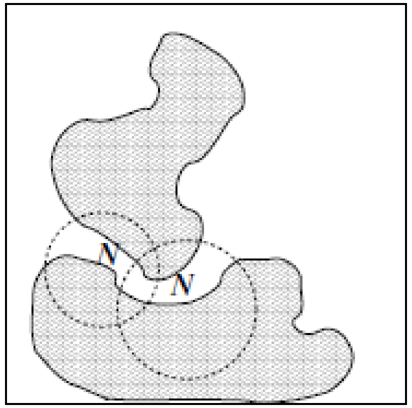
$$H(X) = -\sum_{i=1}^{n} P(x_i)\log_b P(x_i)$$



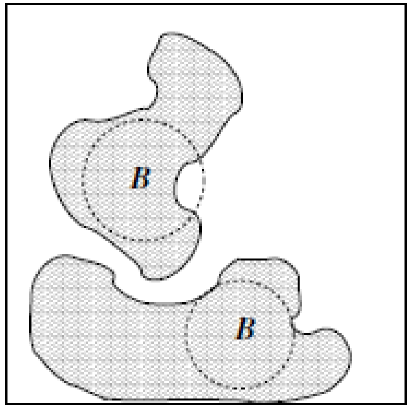
(b) Free



(c) Surface



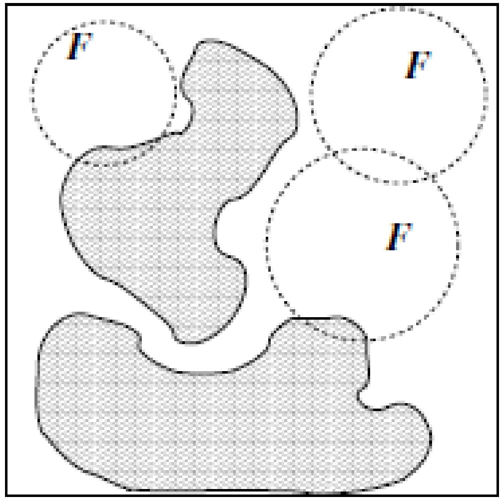
(d) Narrow



(e) Blocked

Region classification

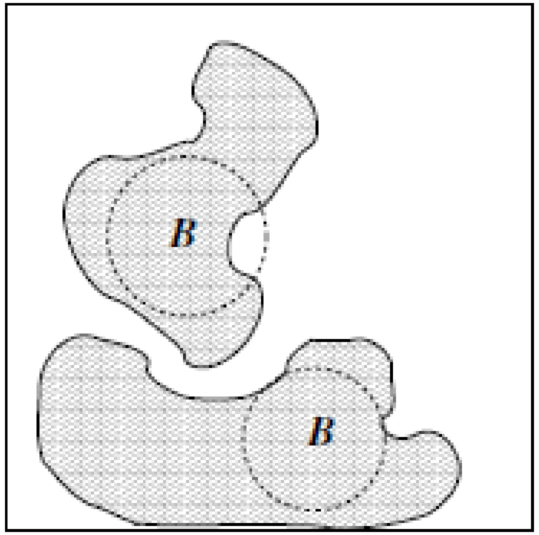
- Free region
  - Percentage (or entropy) of blocked sample is low enough



Region classification

- Blocked region
  - Percentage of free samples in the region (or entropy) is low enough

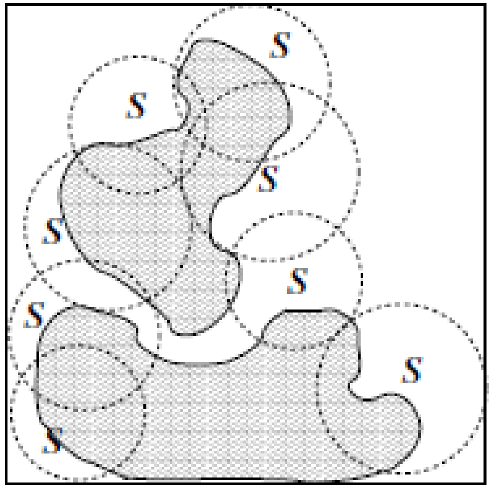
- Attention!
  - Free nodes are discovered during the classification
  - Do not classify a region as blocked until several attempts have been made to classify and add additional samples



Region classification

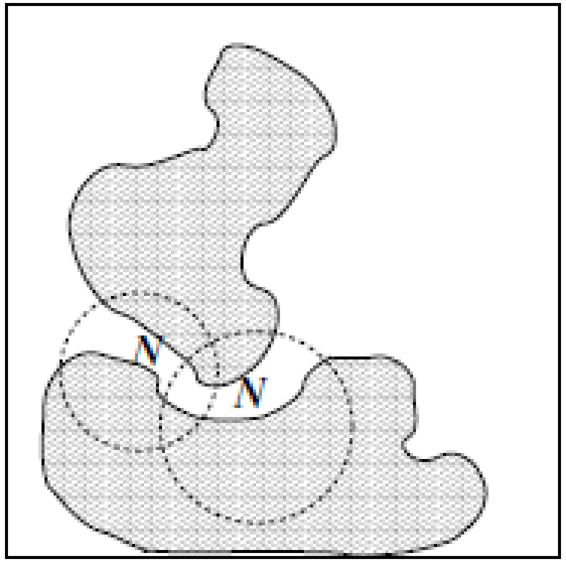
- Surface region
  - High entropy region that can be partition to two low-entropy regions

- Process
  - Divide the region to 2 sub-region
  - Determine centroid of free/blocked space
  - If both sub-region has low entropy → surface region



Region classification

- Narrow region
  - High entropy region that cannot be partition to two low-entropy regions
- Attention
  - Difficult to classify
  - Do not attempt to classify a region as narrow until several attempts have been made to classify and add additional samples



Region classification

Algorithm 3.2 Region Classification

Require: A region R, threshold e_{low}, threshold e_{high}, number of attempts to classify t, and number of samples to add in each classification attempt k.

for t attempts to classify R **do**
    Let e_R be the entropy of R (% of blocked samples in R).
    if $e_R < e_{low}$ **then**
        return *free*
    end if
    Add k additional samples to R and recompute e_R.
    Partition R into two subregions, R_{free} and $R_{blocked}$.
    Let e_{free} be the entropy of R_{free} (% of blocked samples in R_{free}).
    Let $e_{blocked}$ be the entropy of $R_{blocked}$ (% of free samples in $R_{blocked}$).
    if $e_{free} < e_{low}$ and $e_{blocked} < e_{low}$ **then**
        return *surface*
    end if
end for
if $e_R == 1$ **then**
    return *blocked*
end if
if $e_R > e_{high}$ **then**
    return *narrow*
end if
return *surface*

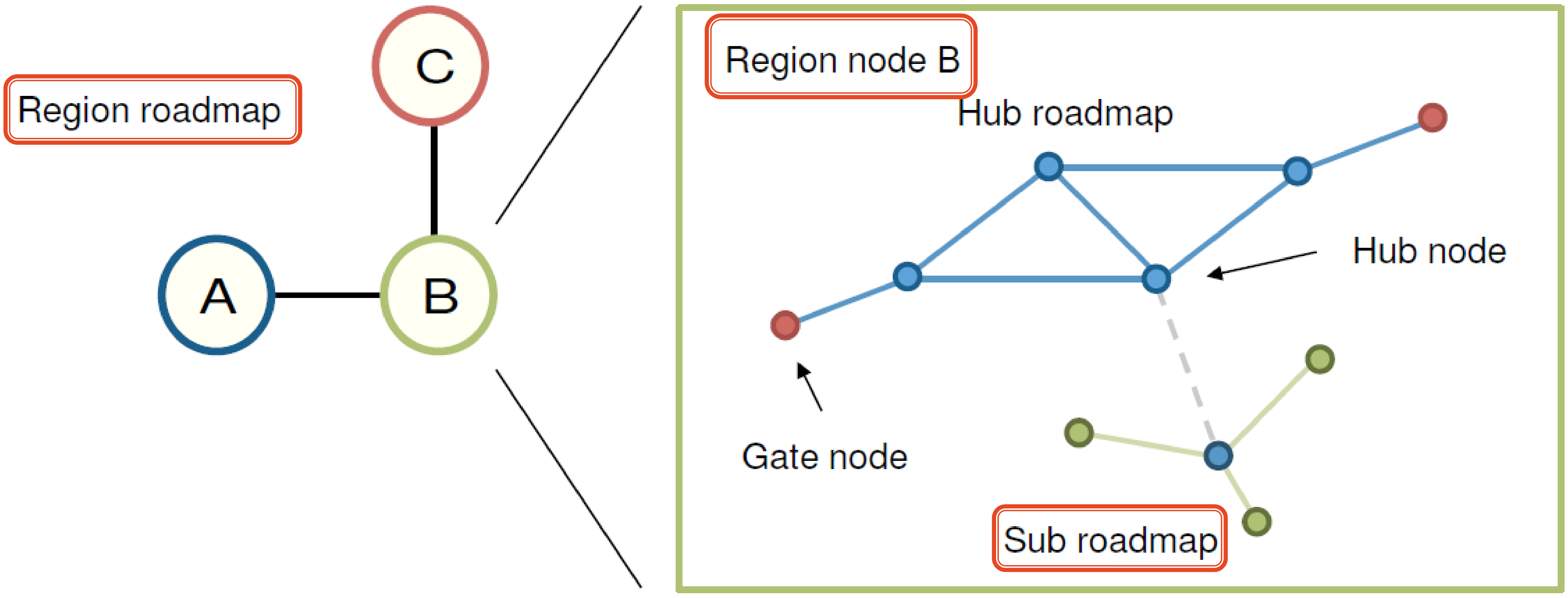
Change the sampling strategy

- Generates additional nodes in less open area
- Improved coverage, yet the roadmap still has **irregularly** distributed nodes
- Navigates in unexplored area – **unsolved**

Hierarchical roadmap

- Main idea
  - Incrementally constructs a **hierarchical roadmap** using low cost sonar sensors
- Hierarchical roadmap
  - A multi-layered structure that abstract the traversable areas using the adequate number of nodes and edges
  - Nodes in the hierarchical roadmap are distributed regularly to cover
  - and divide the traversable areas

Hierarchical roadmap



Motion planning

- Step 1
  - Search a topological path on the region roadmap
- Step 2
  - Search metric local paths in the sub-regions

Region extraction

- Region roadmap is constructed by dividing the entire environment into several sub-regions
  - E.g., rooms in a house
- How to divide?

Region extraction

- Step 1: Obtaining reliable region in the grid map
  - High confidence of grid cell occupancy → practical issue of sonar
- Step 2: Cell Decomposition for traversable area
  - Recursively dividing area until every cell has only empty grid cells

Region extraction

- Step 3: Normalized Graph Cut
  - Tentatively divided into two sub-regions using normalized graph cut
- Step 4: Extracting a New Sub-region
  - Use a convexity criterion determine whether the reliable region can be regarded as one sub-region

$$\mathcal{C}_{1\text{cluster}} = \frac{\sharp \text{ of } occ. \text{ grids} \in CH1}{\sum \text{size of Cell}}$$

$$\mathcal{C}_{2\text{clusters}} = \frac{\sum_{i=1}^{2} \sharp \text{ of } occ. \text{ grids} \in CH2(i)}{\sum \text{size of Cell}}$$

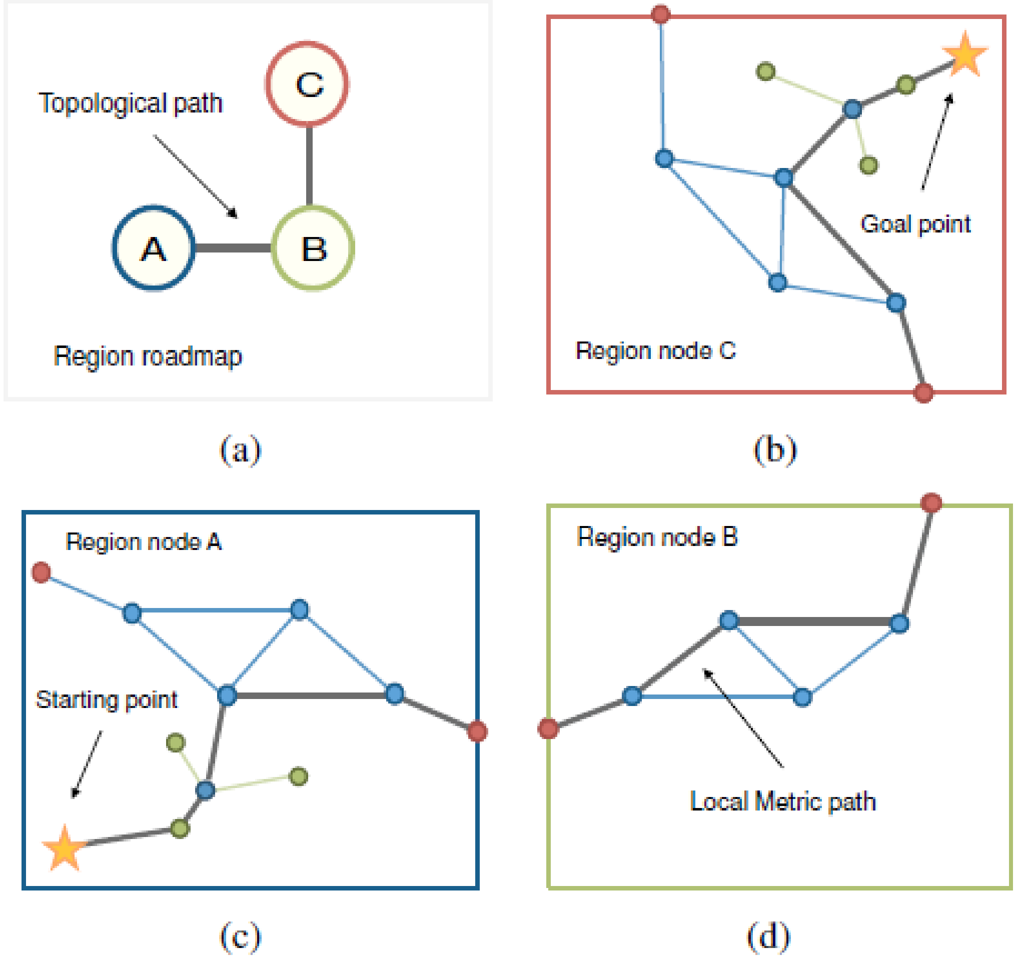
New region extracted!

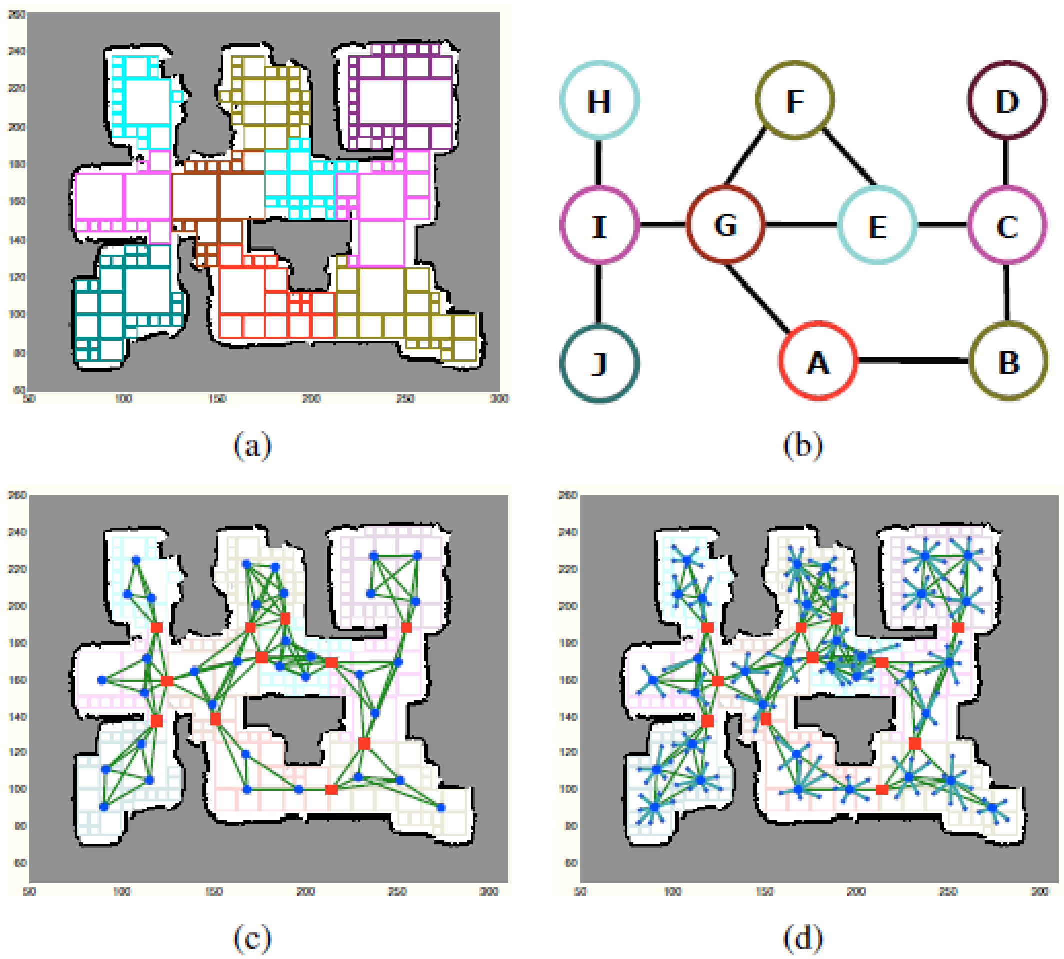
$$\mathcal{C}_{1\text{cluster}} > c_t \quad \& \quad \mathcal{C}_{2\text{clusters}} < 0.5 \times \mathcal{C}_{1\text{cluster}}$$

Region node and gate node

- Region nodes
  - The extracted sub-regions become region nodes (RNs)
  - Generate edges based on the adjacency between two sub-regions
- Gate nodes
  - Providing paths to the neighbor RNs
  - Defined as the midpoint of the boundary between two RNs

Hierarchical roadmap in the home environment





Reference

- [1] Boor, Valérie, Mark H. Overmars, and A. Frank Van Der Stappen. "The Gaussian sampling strategy for probabilistic roadmap planners." In *IEEE international conference on Robotics and automation*, vol. 2, pp. 1018-1023. IEEE, 1999.
- [2] Hsu, David, Tingting Jiang, John Reif, and Zheng Sun. "The bridge test for sampling narrow passages with probabilistic roadmap planners." In *IEEE International Conference on Robotics and Automation*, vol. 3, pp. 4420-4426. IEEE, 2003.
- [3] Rodriguez, Samuel, Shawna Thomas, Roger Pearce, and Nancy M. Amato. "RESAMPL: A region-sensitive adaptive motion planner." In *Algorithmic Foundation of Robotics VII*, pp. 285-300. Springer, Berlin, Heidelberg, 2008.
- [4] Park, B., Choi, J., & Chung, W. K. (2012, May). An efficient mobile robot path planning using hierarchical roadmap representation in indoor environment. In IEEE International Conference on Robotics and Automation (ICRA), 2012 (pp. 180-186).

Student talk

End