An Efficient Mobile Robot Path Planning Using **Hierarchical Roadmap** Representation in Indoor Environment

Denim Patel

Why we even need this! ??

- For path planning, we need discretized world
- Create a Roadmap (Generally PRM)
- And then we do the optimal path finding

BUT,

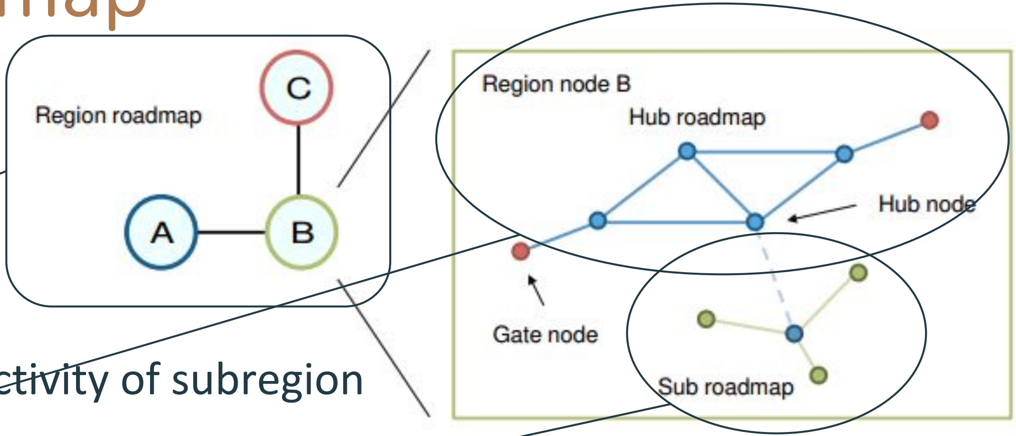
- For finding the optimal path from the created PRM, we need to find the optimal path using all the nodes and edges of the roadmap
- Even when we don’t know if it will find a way eventually!
- What if we do the process in Hierarchy/steps?
  - Means we first find a vague way/roadmap to reach to the goal
  - Once we have a path we will optimize it for minimum cost
  - Node traversal can be saved!

Advantages/benefits of using hierarchy?

- **Node traversal** can be reduced
  - Search Space size
- **Computation expenses** can be reduced
  - Useful for online system
- Roadmaps is created as the robot travels
  - No burden any particular stage of the process
- Unlike the PRM this Roadmap **fully covers the traversable environment**
  - No issue of narrow gaps
- This approach **efficiently represents traversable area** in hierarchical fashion
  - Multilayer graph structure
- This representation is similar to the **environmental perception of human beings**

Hierarchical Roadmap

- **3 layers**
  - **Region Roadmap**
    - Abstracts the connectivity of subregion
  - **Hub Roadmap**
    - Abstracts the traversal within subregion
  - **Sub Roadmap**
    - Abstracts the traversal around hub node



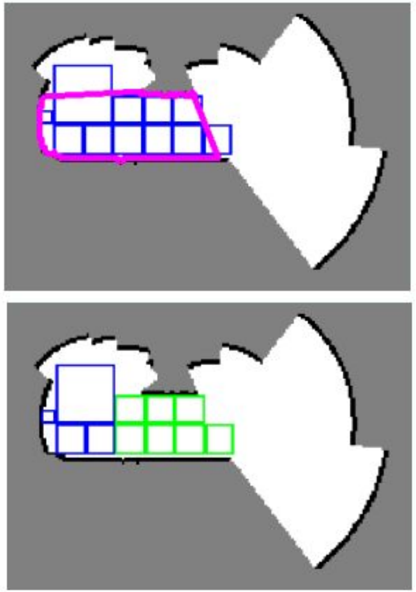
Step 1: Region Roadmap Construction

Goal: Divide the the full environment into subregions

Procedure:

Overview only

- Obtain reliable region
- Apply cell decomposition
- Acquire two tentative subregions
- Extract new subregion



Reliable Region Extraction

- Confidence Cost of each grid cell is calculated

$$\boxed{Conf(x,y) = \sum_{s \in S_{occ}(x,y)} P_s(r,\theta)}$$

- Calculate cost of each node
- Find the mean value of the confidence value using all the cells of the region
- Use this mean as a threshold value for Reliability check
  - If confidence of a particular cell > mean
    - Keep it!
  - Else
    - Discard It!

Cell Decomposition

Note:

Generated grid cells contains both free and occupied grid cells
For generation the roadmap we need cells with only fee/empty grid cells

Goal: Decompose the cells into smaller cells until the free region can described satisfactorily

Procedure:

Recursively divide cell into four subcell if it contains occupied grid cell
Until all the cells only contains free cells

Now,
The Decomposed cell becomes the node
Edges between the nodes are determined using the geometric adjacency

Normalized Graph Cut

Why we need it?

- If the cell decomposed whole region cannot be regarded as one subregion,
- It should be divided into two subregion
- To cut the graph into two
  - Use the concept of Normalized Graph Cut

$$Ncut = \frac{\sum_{i \in C_1, j \in C_2} w_{ij}}{\sum_{i \in C_1, j \in V} w_{ij}} + \frac{\sum_{i \in C_1, j \in C_2} w_{ij}}{\sum_{i \in C_2, j \in V} w_{ij}}$$

  - Algorithm cuts the graph into two clusters which minimizes the similarity between two clusters

Extracting a new subregion

Goal: To determine if a reliable region can be regarded as one subregion or it is better to convert it into two subregions

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

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

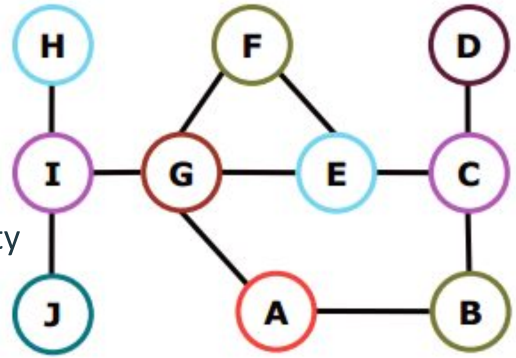
- Convexity criterion is used
    - Convexity considering region as one cluster
    - Convexity considering region as two cluster

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

- It the above conditions are met, new subregion is extracted
- Otherwise go forward and accumulate more data

Region Roadmap

- Extracted subregions become region nodes
  - Environment can be represented as an abstract form
- Nodes
  - Define node at a centre of the region
- Edges
  - Define the edge based on the geometrical connectivity
- Gate Nodes (GN)
  - For each node define Gate node
    - That works as an entry/exit point for each RN
    - The role of GN is to provide path to neighbour RN
  - GN is defined as the midpoint of the boundary between two RNs

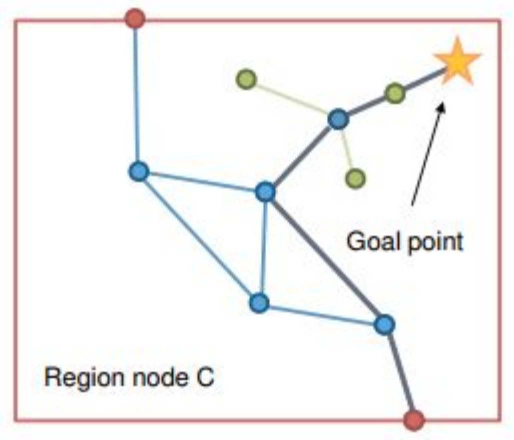


Local Roadmap Construction

Note: A local roadmap is constructed for each Region Nodes (RN)

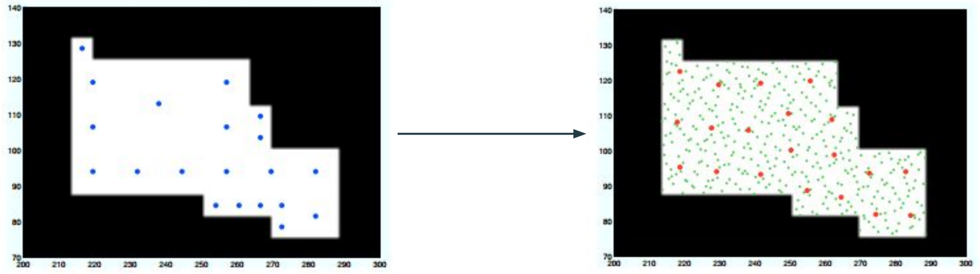
Procedure:

- Generate and rearrange initial nodes to cover and divide RN
- Classify these nodes into
  - HNs (Hub nodes)
  - SNs (Sub nodes)



- Construct Hub roadmap and subroadmap connecting HNs and SNs resp.

Node generation



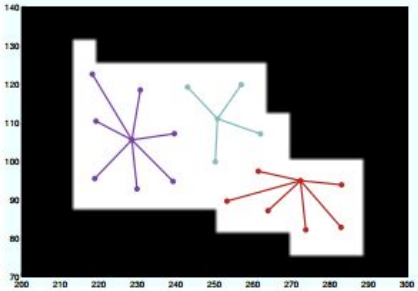
Goal:

- Generate nodes in the region which are **regularly distributed and cover** the traversable area
- In general, Sampling is the option
  - The problem is the non-uniformity of nodes
- We can use the data of cell decomposition (We have already calculated!)
  - Then arrange the nodes to improve the arrangement
  - Initially, No of nodes and position = the centroid of cells
- To improve the position iterative technique called Centroidal Voronoi tessellation(CVT)

Node Classification

Goal: classify nodes into several subsets and then identify representative nodes

How? APC(Affinity Propagation Clustering) is used for the purpose



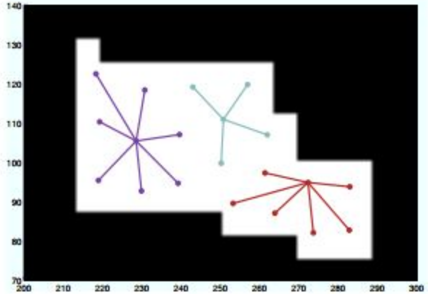
- APC identifies nodes into several subsets
- Representative node becomes the HN of the subset
- Remaining nodes are regarded as SNs
- Technique:
    - Each node recursively transmits real valued message
    - Each node receives real valued messages from another node
    - Type of messages
        - Responsibility
            - Message sent from a node to a representative node candidate
        - Availability
            - Message sent from a representative node to other nodes

Construct Local Roadmap

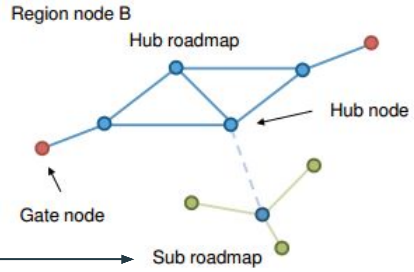
Goal: **Construct Hub roadmap and Sub roadmap**

Procedure:

- Positions of HNs as a centroid of corresponding SNs (to have minimum distance from most of the SNs



- Gate nodes are connected to Hub nodes in RN
- Each HN is connected to SNs for sub roadmap



Hierarchical Path Planning

- In **Unilayered graph traversal, complexity increases with number of nodes**
- Hierarchical Path Planning helps in reducing number of node traversal
- **DNS (Divide and Search)** approach
  - **Step-1.** A topological path is searched based on Region Roadmap.
    - Topological path consists of RNs and start and end node
  - **Step-2.** A local metric path is searched in each RNs on the topological path
    - GNs helps in this step!
  - **Step-3.** A global path is obtained using topological and metric path

RN contains starting and ending node considered as start RN and goal RN

Topological path:

Edges = euclidean distance between centroids of RNs

Hierarchical Path Planning

Things to remember/know about HPP

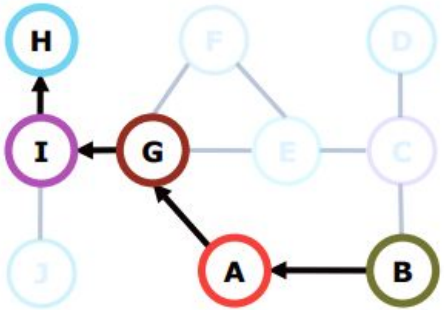
- **Topological path:**

Edges = euclidean distance between centroids of RNs

- A local metric path is created using Hub and Sub roadmap

DNC is used for solving the Hierarchical path planning queries

- **HPP Approach is more efficient than any single layered path planning technique A* or Dijkstra's**



Experimental Experiment

Pioneer 3DX Robot

Path planning comparison in Home env.

Hierarchical roadmap was compared with Dijkstra’s algorithm

NOTE: Author should have used A* for better competition

Because Dijkstra is the slowest

Hierarchical Path planner took 5.32ms

Dijkstra’s took 188.31ms

	Avg. Planning Time	Avg. Travel Cost
Single-Layered	188.31 ms	13.41 m
Hierarchical	5.32 ms	13.06 m

Thank You!