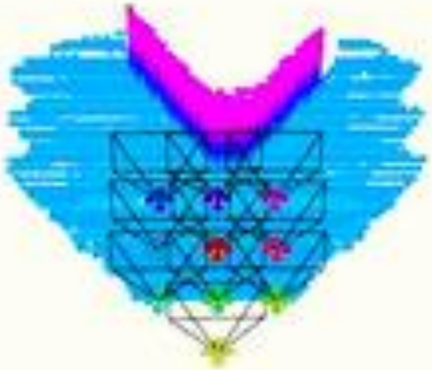
Autonomous Exploration

Tobias Lübbe

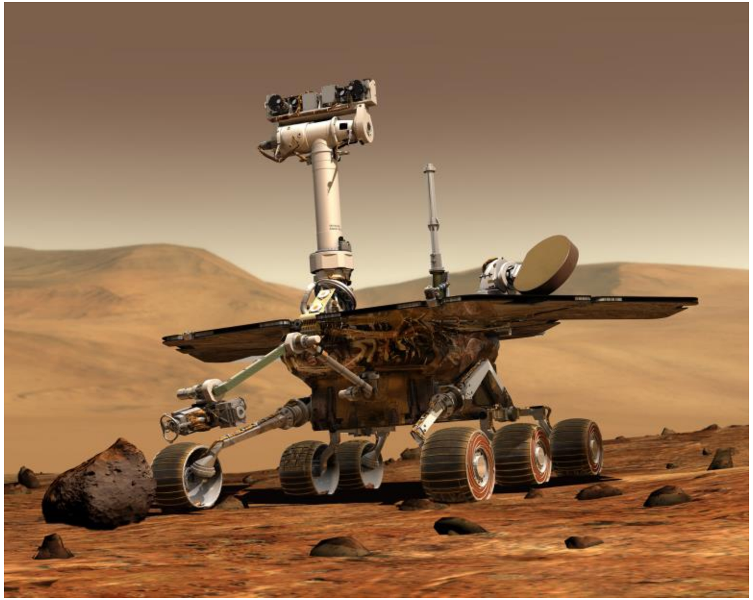
Seminar: Robot Perception & Intelligence

Munich, 02. December 2024



Motivation

Space Exploration [14]

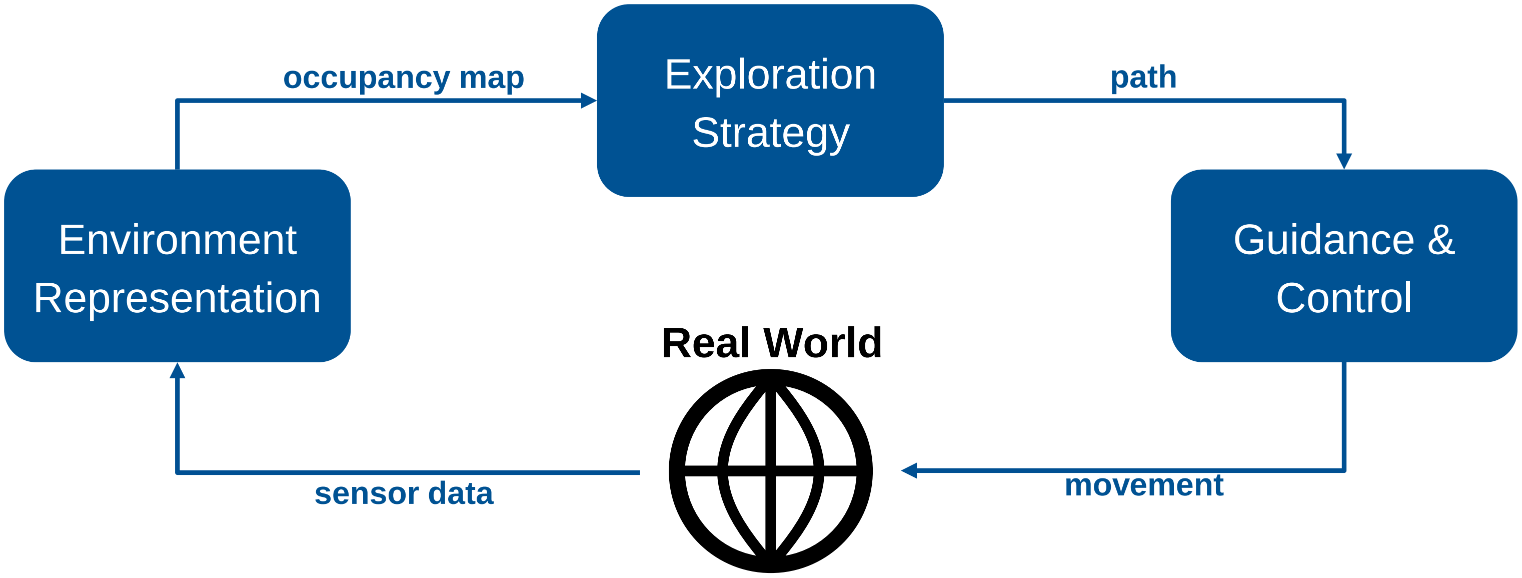


Search and Rescue Missions [15]

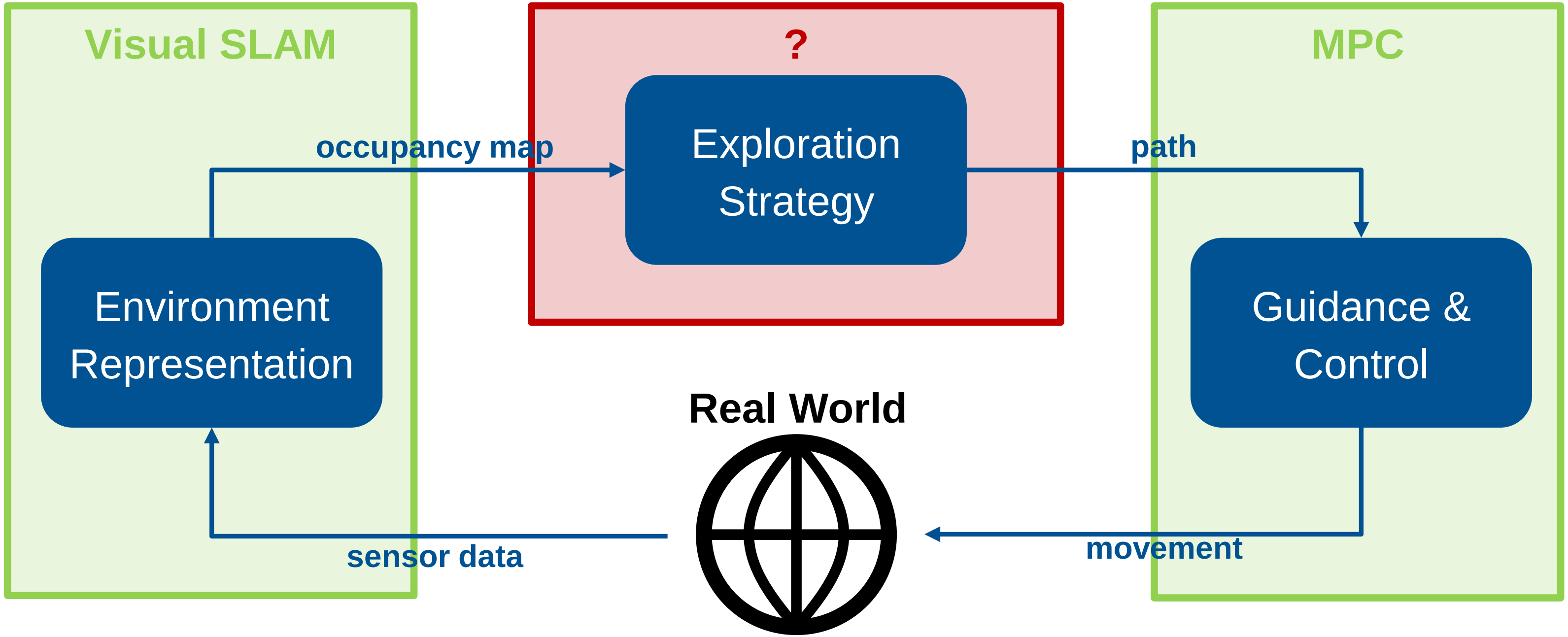
Outline

- Challenges in Autonomous Exploration
- Key Concepts of Exploration Strategies
- Existing Work & Results
  - Receding Horizon Next Best View
  - Frontier-based Information-driven Exploration
  - FUEL: Fast UAV Exploration
- Conclusions & Further Work

Challenges in Autonomous Exploration



Challenges in Autonomous Exploration



Key Concepts of Exploration Strategies

Candidate Generation: sampling-based / frontier-based

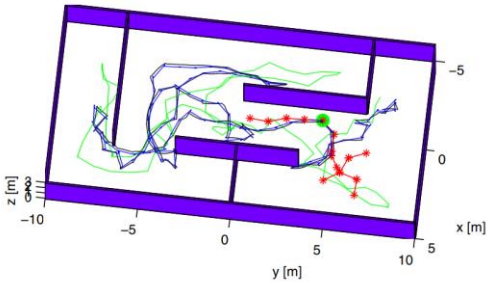
Utility Measure: volumetric / entropy-based / travel time / etc.

Optimization: global strategy / greedy strategy

Path Planning: RRT / RRT* / Dijkstra / A* / etc.

Existing Work

I

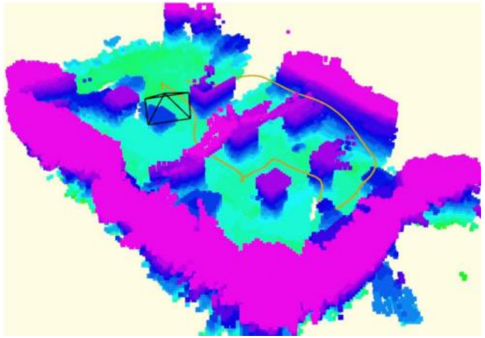


Receding Horizon Next Best View Planner (2016)

II

Fast Frontier-based Information-driven Exploration (2020)

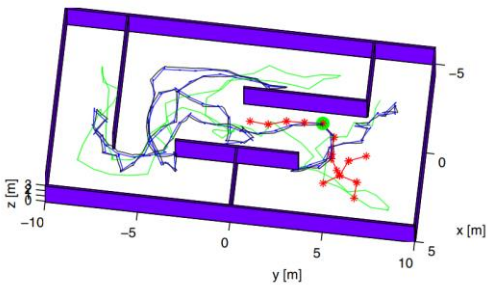
III



FUEL: Fast UAV Exploration (2021)

Existing Work

I



Receding Horizon Next Best View Planner
(2016)

II

Fast Frontier-based Information-driven Exploration
(2020)

III

FUEL: Fast UAV Exploration
(2021)

Existing Work

- sampling-based
- volumetric + travel time
- greedy
- RRT

II

Fast Frontier-based Information-driven Exploration (2020)

III

FUEL: Fast UAV Exploration (2021)

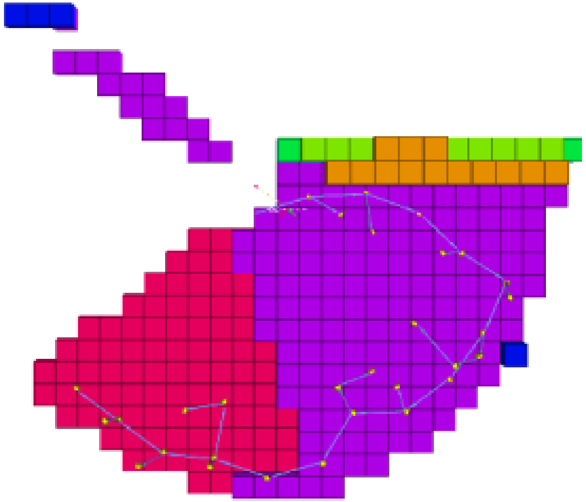
Receding Horizon Next Best View Planner (2016) [1][4]

Description

Receding Horizon Next Best View Planner (2016) [1][4]

Description

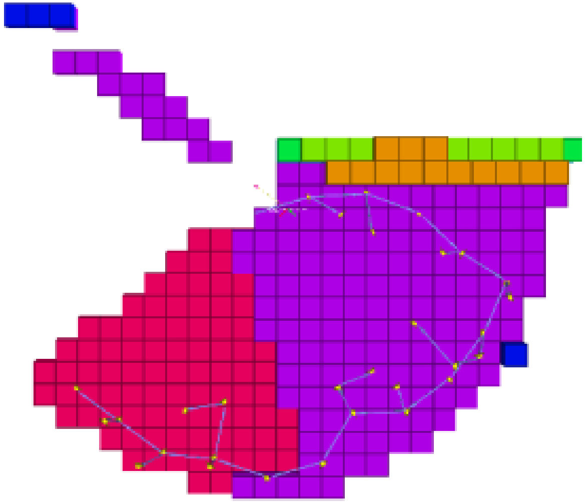
- Sample RRT (x, y, z, yaw) with at N_{min} nodes



Receding Horizon Next Best View Planner (2016) [1][4]

Description

- Sample RRT (x, y, z, yaw) with at N_{min} nodes
- Associate each node n with an information gain $g(n)$ based on number of **visible unexplored voxels** and **path costs**

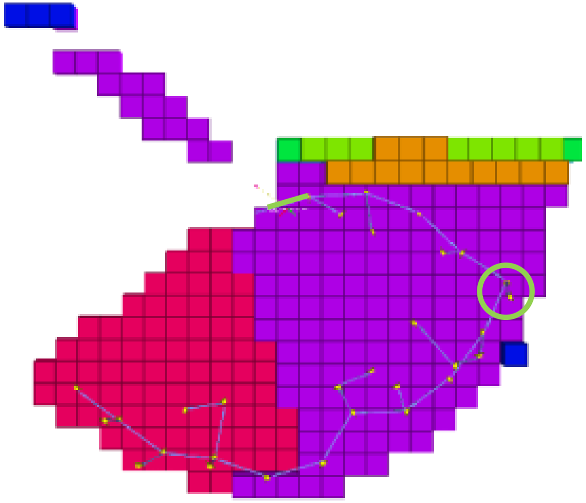


$$\mathbf{Gain}(n_k) = \mathbf{Gain}(n_{k-1}) + \mu(\mathbf{Visible_V}(\mathcal{M}, \xi_k)) e^{-\lambda c(\sigma_{k-1}^k)}$$

Receding Horizon Next Best View Planner (2016) [1][4]

Description

- Sample RRT (x, y, z, yaw) with at N_{min} nodes
- Associate each node n with an information gain $g(n)$ based on number of **visible unexplored voxels** and **path costs**
- Select node with highest $g(n)$ and execute first path segment

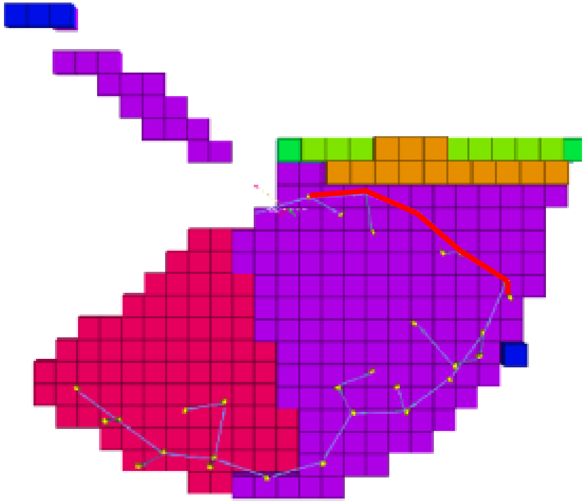


$$\mathbf{Gain}(n_k) = \mathbf{Gain}(n_{k-1}) + \mu(\mathbf{Visible_V}(\mathcal{M}, \xi_k))e^{-\lambda c(\sigma_{k-1}^k)}$$

Receding Horizon Next Best View Planner (2016) [1][4]

Description

- Sample RRT (x, y, z, yaw) with at N_{min} nodes
- Associate each node n with an information gain $g(n)$ based on number of **visible unexplored voxels** and **path costs**
- Select node with highest $g(n)$ and execute first path segment
- Use remaining path as RRT initialization for next planning iteration

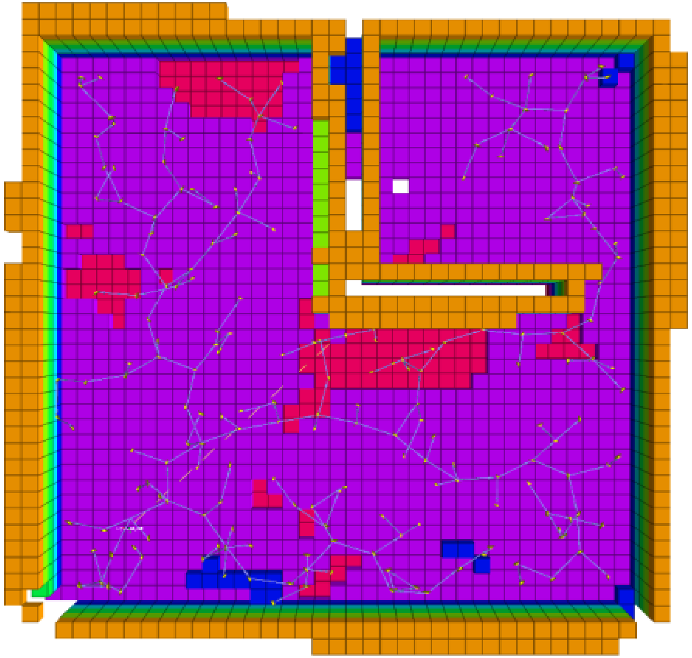


$$\mathbf{Gain}(n_k) = \mathbf{Gain}(n_{k-1}) + \mu(\mathbf{Visible_V}(\mathcal{M}, \xi_k))e^{-\lambda c(\sigma_{k-1}^k)}$$

Receding Horizon Next Best View Planner (2016) [1][4]

Description

- Sample RRT (x, y, z, yaw) with at N_{min} nodes
- Associate each node n with an information gain $g(n)$ based on number of **visible unexplored voxels** and **path costs**
- Select node with highest $g(n)$ and execute first path segment
- Use remaining path as RRT initialization for next planning iteration
- Repeat until RRT reaches N_{max} nodes with $g(n) = 0,\ \forall n$

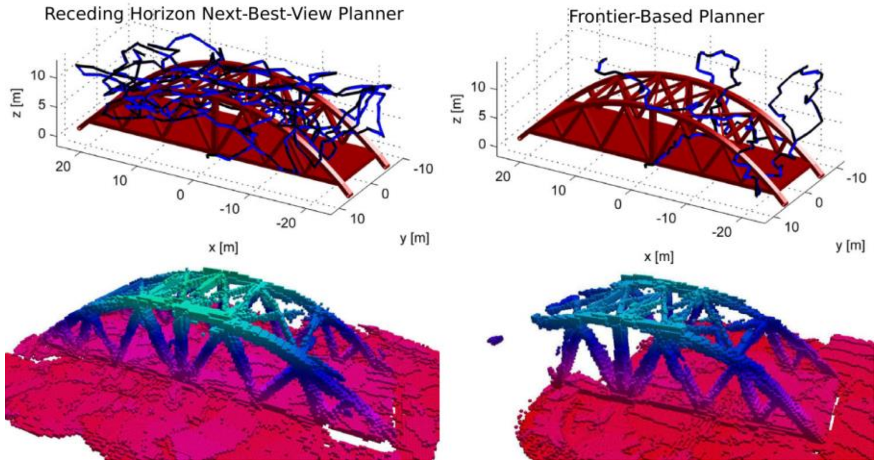


Receding Horizon Next Best View Planner (2016) [1][4]

Results

- + Faster computation leads to quicker exploration
- + Huge improvements in scalability
- \- Prone to back-and-forth movements

$$\mathcal{O}(N_{\mathbb{T}} \log(N_{\mathbb{T}}) + N_{\mathbb{T}}/r^3 \log(V/r^3) + N_{\mathbb{T}}(d_{\max}^{\text{planner}}/r)^4 \log(V/r^3))$$



terminated after 43.8 min

terminated after 1670.1 min

Existing Work

I

Receding Horizon Next Best View Planner (2016)

II

Fast Frontier-based Information-driven Exploration (2020)

III

FUEL: Fast UAV Exploration (2021)

Existing Work

I

Receding Horizon Next Best View Planner
(2016)

hybrid

entropy-based + travel time

greedy

informed RRT*

III

FUEL: Fast UAV Exploration
(2021)

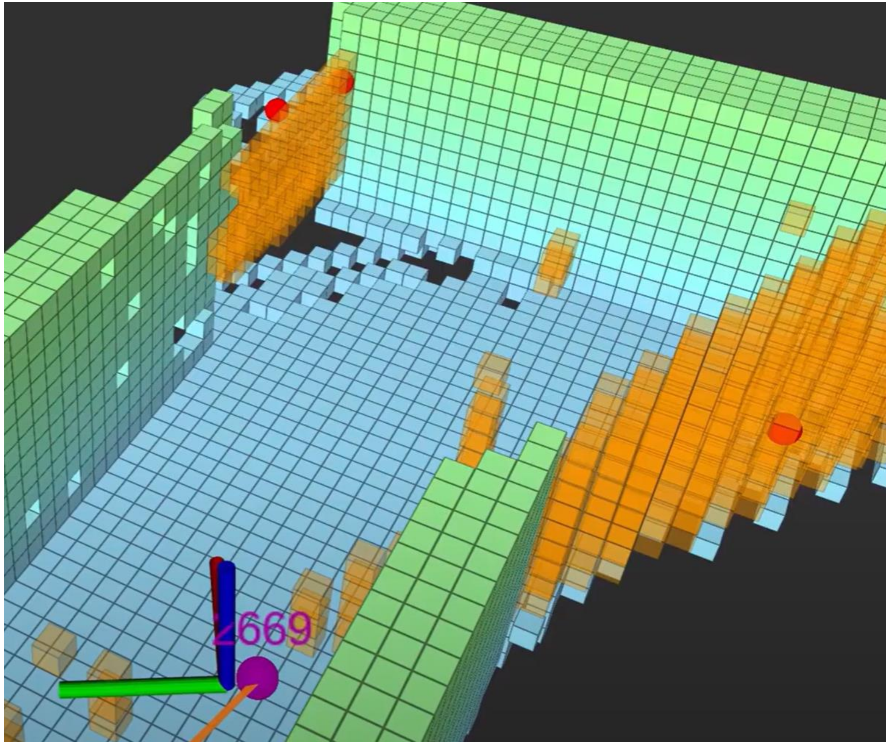
Fast Frontier-based Information-driven Exploration (2020) [2][5]

Description

Fast Frontier-based Information-driven Exploration (2020) [2][5]

Description

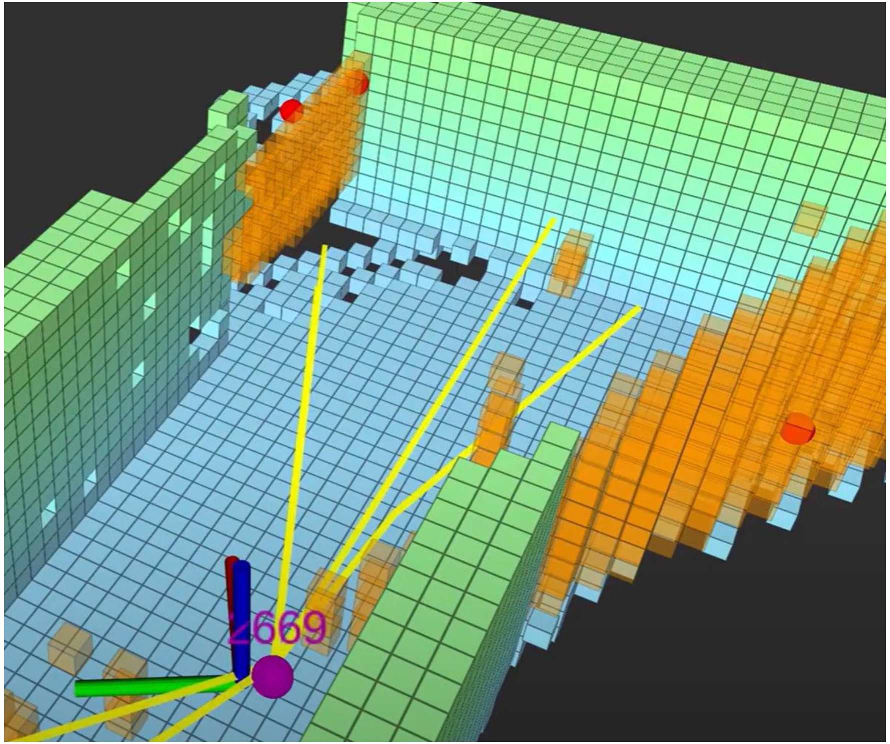
- Leverage **octree map structure** to detect frontier clusters and sample candidates (x, y, z) from frontier clusters



Fast Frontier-based Information-driven Exploration (2020) [2][5]

Description

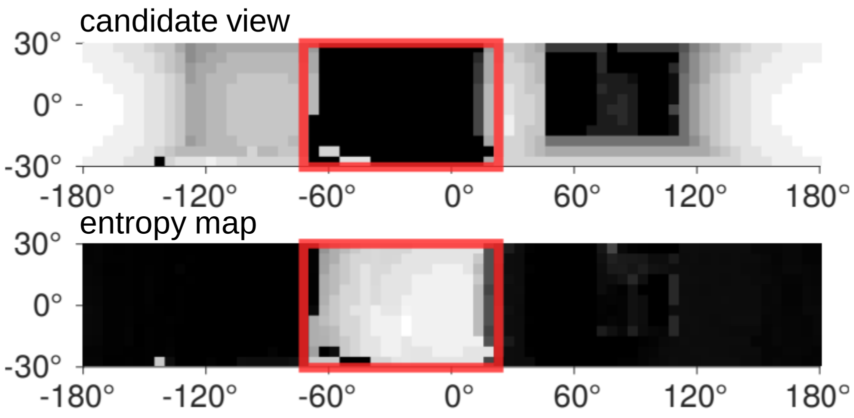
- Leverage **octree map structure** to detect frontier clusters and sample candidates (x, y, z) from frontier clusters
- Employ informed RRT* to find paths to each candidate



Fast Frontier-based Information-driven Exploration (2020) [2][5]

Description

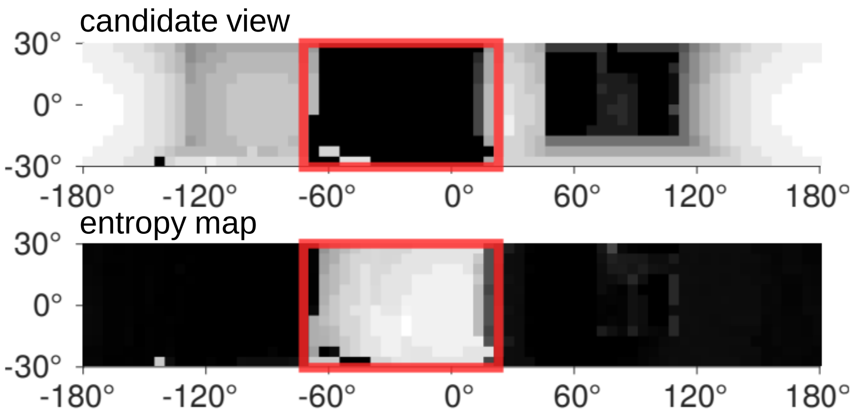
- Leverage **octree map structure** to detect frontier clusters and sample candidates (x, y, z) from frontier clusters
- Employ informed RRT* to find paths to each candidate
- Estimate entropy map around each candidate to and perform **yaw angle optimization** for maximum information gain



Fast Frontier-based Information-driven Exploration (2020) [2][5]

Description

- Leverage **octree map structure** to detect frontier clusters and sample candidates (x, y, z) from frontier clusters
- Employ informed RRT* to find paths to each candidate
- Estimate entropy map around each candidate to and perform **yaw angle optimization** for maximum information gain
- Compute utility values for each candidate based on **entropy** and **path cost** and select best candidate as exploration goal

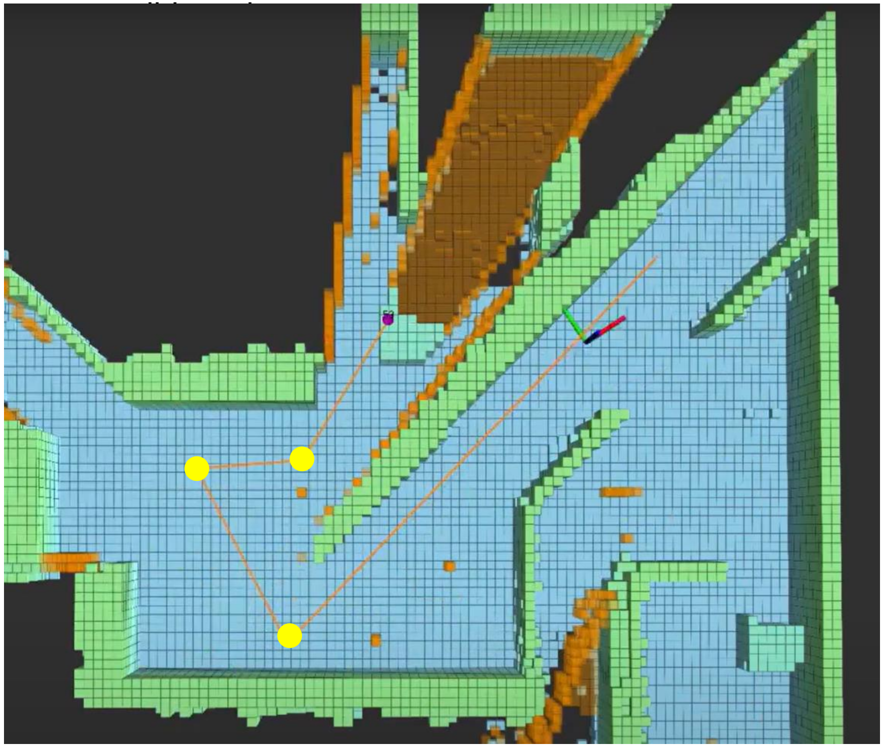


$$u(\mathbf{x}_i, \hat{W}_i) = \frac{\mathbb{H}(\mathbf{x}_i)}{T(\hat{W}_i)}$$

Fast Frontier-based Information-driven Exploration (2020) [2][5]

Description

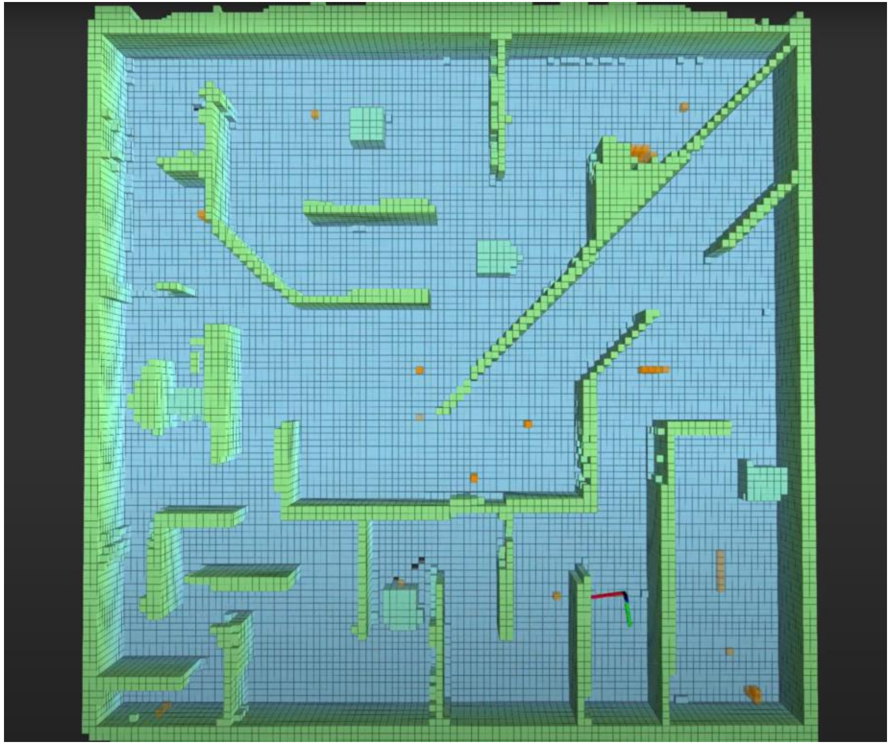
- Leverage **octree map structure** to detect frontier clusters and sample candidates (x, y, z) from frontier clusters
- Employ informed RRT* to find paths to each candidate
- Estimate entropy map around each candidate to and perform **yaw angle optimization** for maximum information gain
- Compute utility values for each candidate based on **entropy** and **path cost** and select best candidate as exploration goal
- Perform **yaw angle optimization** for intermediate waypoints and execute path



Fast Frontier-based Information-driven Exploration (2020) [2][5]

Description

- Leverage **octree map structure** to detect frontier clusters and sample candidates (x, y, z) from frontier clusters
- Employ informed RRT* to find paths to each candidate
- Estimate entropy map around each candidate to and perform **yaw angle optimization** for maximum information gain
- Compute utility values for each candidate based on **entropy** and **path cost** and select best candidate as exploration goal
- Perform **yaw angle optimization** for intermediate waypoints and execute path
- Terminate when all frontier clusters are explored



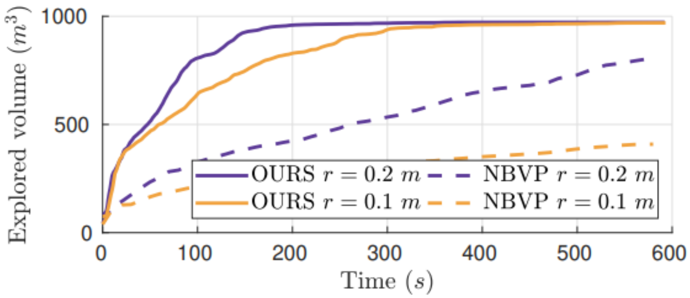
Fast Frontier-based Information-driven Exploration (2020) [2][5]

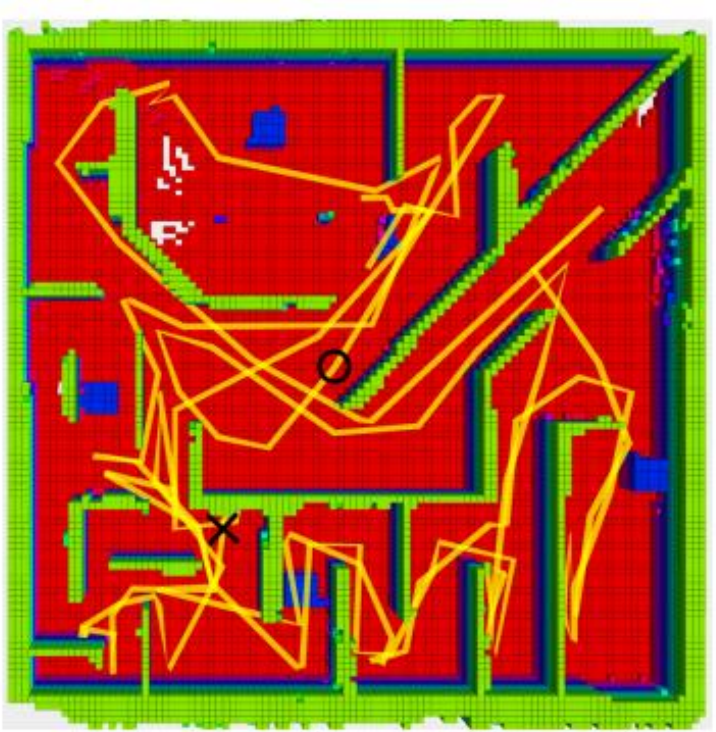
Results

+ Faster and more robust exploration than RH-NBVP due to:
  - guided path planning
  - no expensive frontier clustering

+ Entropy based information gain helps to increase map quality

- No real-time path adaption possible





Existing Work

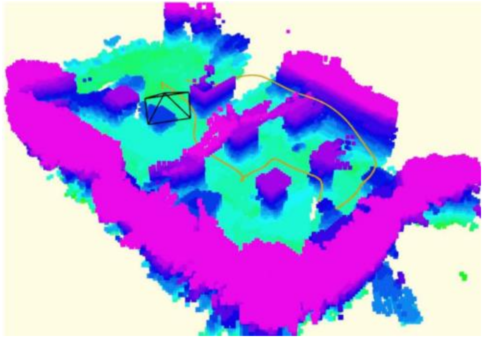
I

Receding Horizon Next Best View Planner (2016)

II

Fast Frontier-based Information-driven Exploration (2020)

III



FUEL: Fast UAV Exploration (2021)

Existing Work

I

Receding Horizon Next Best View Planner (2016)

II

Fast Frontier-based Information-driven Exploration (2020)

- Frontier-based
- volumetric + travel time + vehicle dynamics
- global
- A*

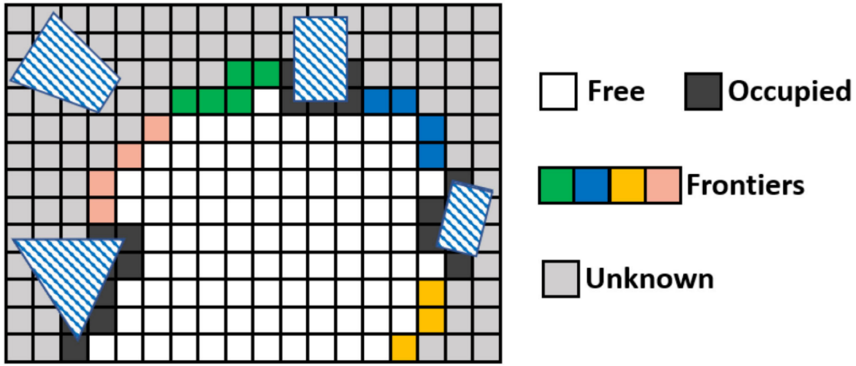
FUEL: Fast UAV Exploration (2021) [3][6]

Description

FUEL: Fast UAV Exploration (2021) [3][6]

Description

- Initial detection of frontier clusters and store them as **FIS**

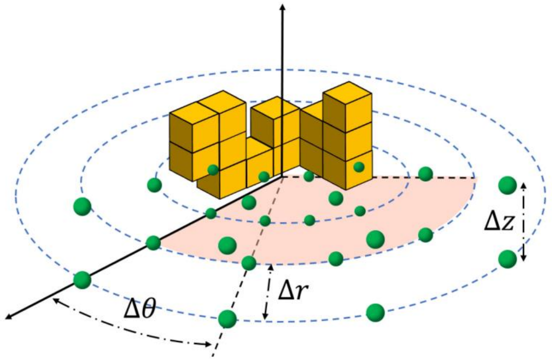


Data	Explanation
C_i	Frontier cells that belong to the cluster
$\mathbf{p}_{\mathrm{avg},i}$	Average position of C_i
B_i	Axis-aligned bounding box of C_i
VP_i	Viewpoints covering the cluster
$L_{\mathrm{cost},i}$	Doubly linked list of connection costs to all other clusters

FUEL: Fast UAV Exploration (2021) [3][6]

Description

- Initial detection of frontier clusters and store them as **FIS**
- Uniformly sample views around cluster center. **Store views with high cluster coverage** in FIS

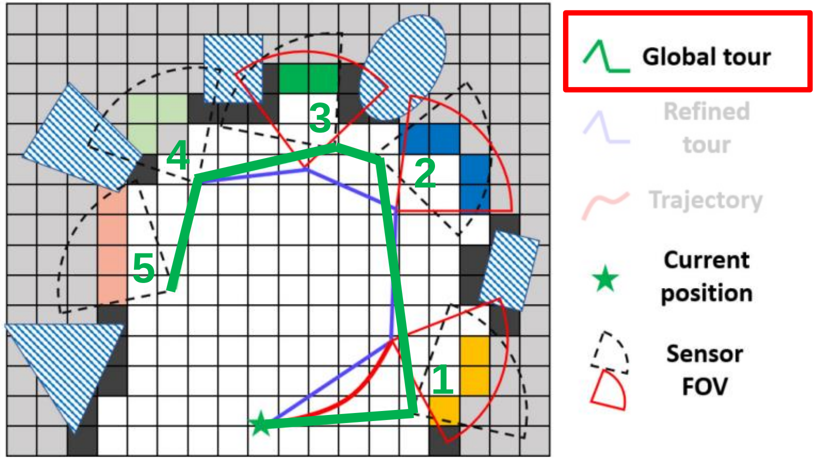


Data	Explanation
C_i	Frontier cells that belong to the cluster
$\mathbf{p}_{\text{avg},i}$	Average position of C_i
B_i	Axis-aligned bounding box of C_i
VP_i	Viewpoints covering the cluster
$L_{\text{cost},i}$	Doubly linked list of connection costs to all other clusters

FUEL: Fast UAV Exploration (2021) [3][6]

Description

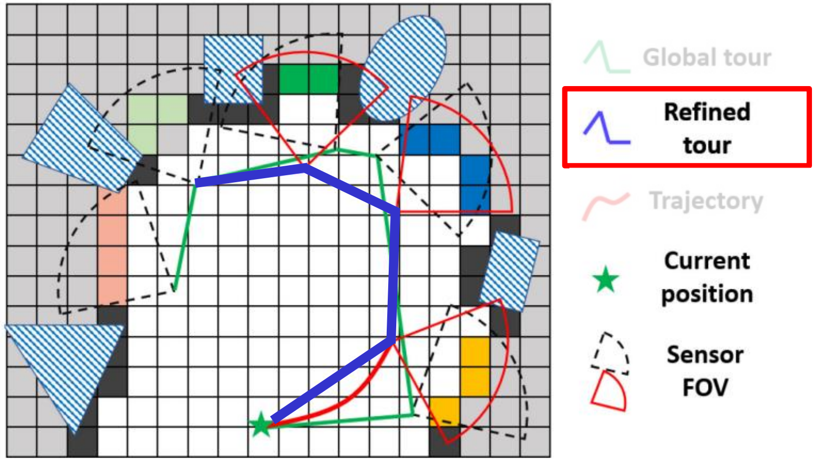
- Initial detection of frontier clusters and store them as **FIS**
- Uniformly sample views around cluster center. **Store views with high cluster coverage** in FIS
- **Global optimization:** Solve TSMP to find optimal exploration order of frontiers



FUEL: Fast UAV Exploration (2021) [3][6]

Description

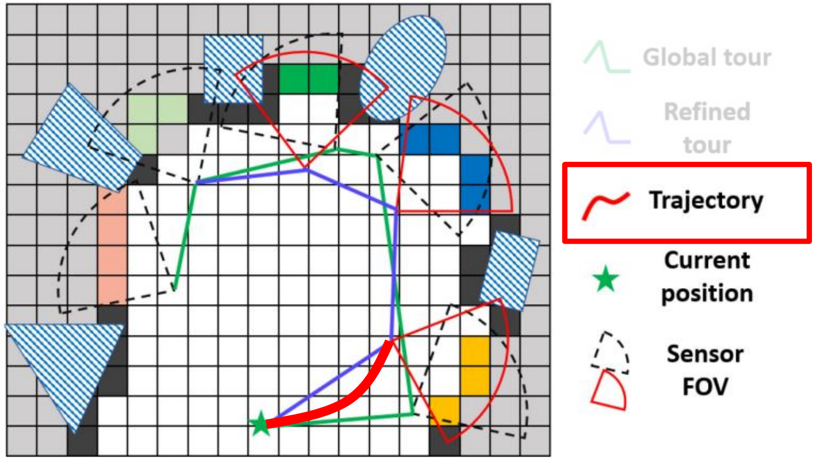
- Initial detection of frontier clusters and store them as **FIS**
- Uniformly sample views around cluster center. **Store views with high cluster coverage** in FIS
- **Global optimization:** Solve TSMP to find optimal exploration order of frontiers
- **Refine local path segment:** Build directed acyclic graph of richer viewpoint sets and find cheapest path (Dijkstra)



FUEL: Fast UAV Exploration (2021) [3][6]

Description

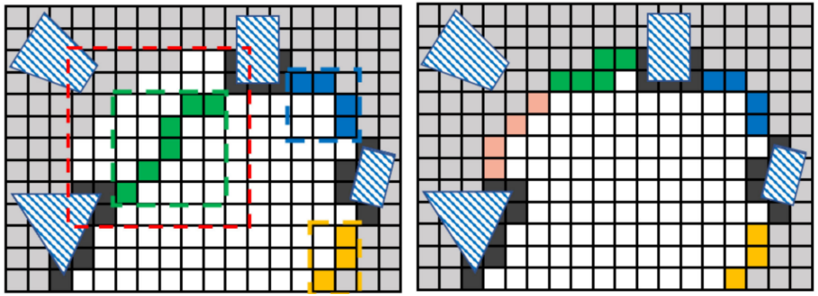
- Initial detection of frontier clusters and store them as **FIS**
- Uniformly sample views around cluster center. **Store views with high cluster coverage** in FIS
- **Global optimization:** Solve TSMP to find optimal exploration order of frontiers
- **Refine local path segment:** Build directed acyclic graph of richer viewpoint sets and find cheapest path (Dijkstra)
- **B-Spline Trajectory:** Compute Splines between next waypoints to allow for smoother and faster flying



FUEL: Fast UAV Exploration (2021) [3][6]

Description

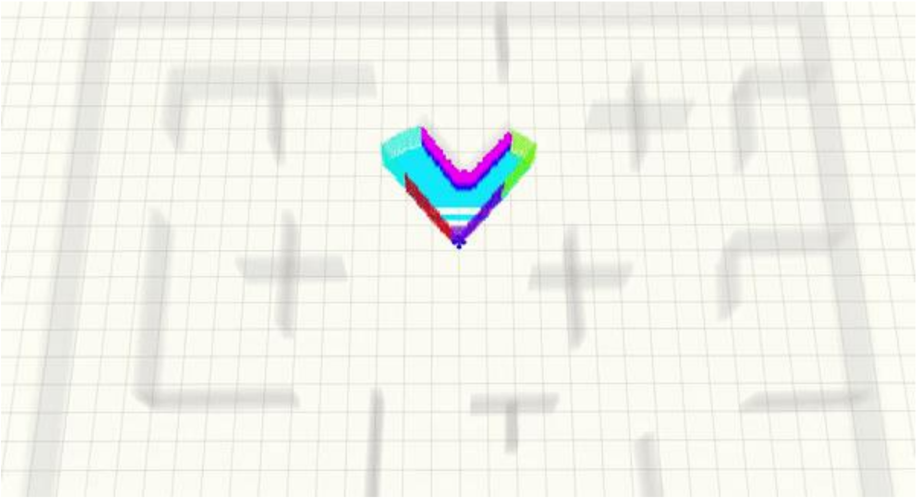
- Initial detection of frontier clusters and store them as **FIS**
- Uniformly sample views around cluster center. **Store views with high cluster coverage** in FIS
- **Global optimization:** Solve TSMP to find optimal exploration order of frontiers
- **Refine local path segment:** Build directed acyclic graph of richer viewpoint sets and find cheapest path (Dijkstra)
- **B-Spline Trajectory:** Compute Splines between next waypoints to allow for smoother and faster flying
- At each map update, **incrementally** update all FISs in that area and repeat planning process



FUEL: Fast UAV Exploration (2021) [3][6]

Description

- Initial detection of frontier clusters and store them as **FIS**
- Uniformly sample views around cluster center. **Store views with high cluster coverage** in FIS
- **Global optimization:** Solve TSMP to find optimal exploration order of frontiers
- **Refine local path segment:** Build directed acyclic graph of richer viewpoint sets and find cheapest path (Dijkstra)
- **B-Spline Trajectory:** Compute Splines between next waypoints to allow for smoother and faster flying
- At each map update, **incrementally** update all FISs in that area and repeat planning process
- Terminate when no frontiers are left



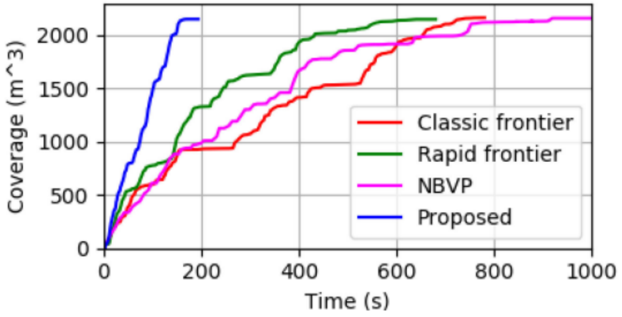
FUEL: Fast UAV Exploration (2021) [3][6]

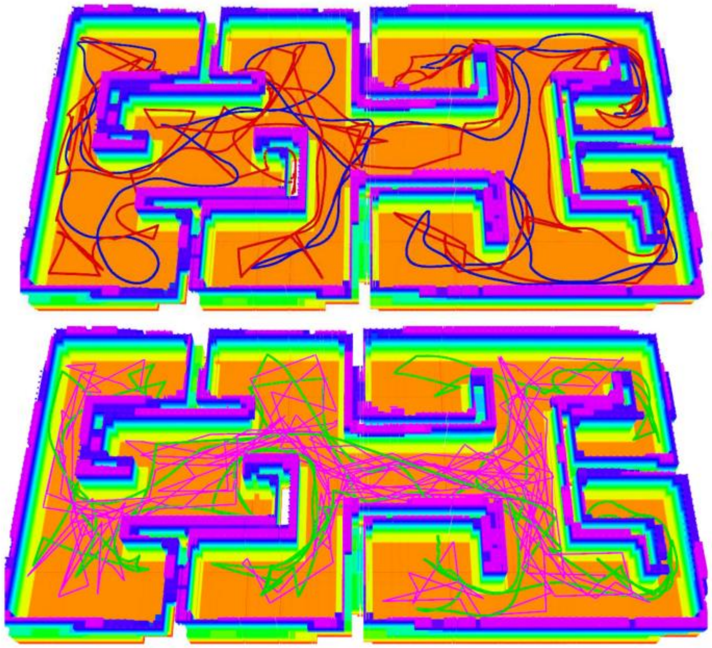
Results

+ Hierarchical Planning:
  - smoother exploration path
  - continuous path updates (adaptive)

+ Global Optimization:
  - shorter exploration path

+ Incremental Frontier Structure:
  - reduced computational overhead





Conclusions

- Frontier detection is important to guide the exploration process to unexplored space
- Sampling based methods can increase computational efficiency
- Hybrid approaches are getting more and more popular
- Global optimization leads to better path efficiency
- More elaborate information gain can be used to tune exploration

  (e.g. taking vehicle dynamics into account speeds up exploration significantly)

Further Work

- Multi Resolution Mapping [7]
- Exploit larger scale experiments [8]
- More elaborate information gains (e.g. include object semantics) [10]
- Distributed multi agent exploration [13]
- Incorporate Deep Leaning Approaches
  - Leverage Scene Completion Networks for more informed exploration [11]
  - Reinforcement Learning approaches [12]

Questions ?

References

[1] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, “Receding horizon path planning for 3D exploration and surface inspection,” Autonomous Robots, vol. 42, no. 2, pp. 291–306, Feb. 2018.
[2] A. Dai, S. Papatheodorou, N. Funk, D. Tzoumanikas, S. Leutenegger, „Fast frontier-based information-driven autonomous exploration with an MAV” , 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 9570–9576, Feb. 2020.
[3] B. Zhou, Y. Zhang, X. Chen, and S. Shen, “FUEL: Fast UAV exploration using incremental frontier structure and hierarchical planning,” IEEE Robot. Autom. Lett., vol. 6, no. 2, pp. 779–786, Apr. 2021.
[4] https://github.com/HKUST-Aerial-Robotics/FUEL
[5] https://github.com/ethz-asl/nbvplanner
[6] https://www.youtube.com/watch?v=tH2VkVony38
[7] E. Vespa, N. Funk, P. H. Kelly, and S. Leutenegger, “Adaptiveresolution octree-based volumetric SLAM,” in 2019 International Conference on 3D Vision (3DV). IEEE, Sep 2019, pp. 654–662
[8] J. Huang *et al*., "FAEL: Fast Autonomous Exploration for Large-scale Environments With a Mobile Robot," in *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1667-1674, March 2023.
[9] X. Yu, C. Chen, „Semantic-aware Next-Best-View for Multi-DoFs Mobile System in Search-and-Acquisition based Visual Perception”, 2024.
[10] L. Schmid, M. N. Cheema, V. Reijgwart, R. Siegwart, F. Tombari, and C. Cadena, “SC-explorer: Incremental 3D scene completion for safe and efficient exploration mapping and planning,” 2022
[11] Y. Cao, R. Zhao, Y. Wang, B. Xiang, and G. Sartoretti, “Deep reinforcement learning-based large-scale robot exploration,” IEEE Robotics and Automation Letters, 2024
[12] B. Zhou, H. Xu, and S. Shen, “Racer: Rapid collaborative exploration with a decentralized multi-uav system,” IEEE Transactions on Robotics, 2023
[13] https://www.eenewseurope.com/en/drones-with-ears-to-aid-in-natural-disasters/
[14] https://de.wikipedia.org/wiki/Mars-Rover