Selective Autonomous Exploration

Based on TARE Exploration Planner

Presented by Fan Ding

Selective Autonomous Exploration

New concept I come up

Autonomous Exploration

VS

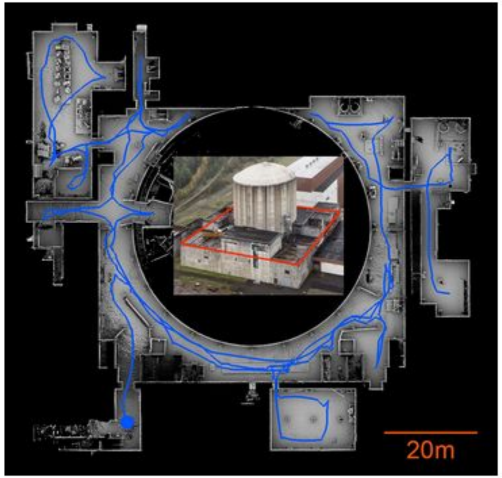
Selective Autonomous Exploration

- Whole environment
- Long time

- Partial environment
- Short time
- Efficient

Motivation

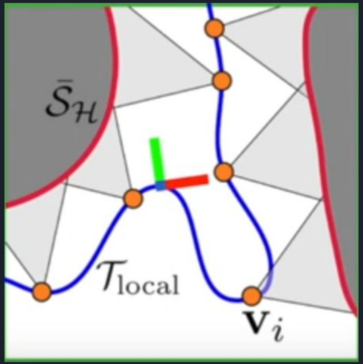
- Paper: “TARE: A Hierarchical Framework for Efficiently Exploring Complex 3D Environments”.
- Robots replace human in dangerous and inaccessible environments.



Urban circuit competition result from DARPA Subterranean Challenge in Satsop Nuclear Plant, WA. Our vehicle traveled over 886m in 1458s to explore the site, fully autonomously.

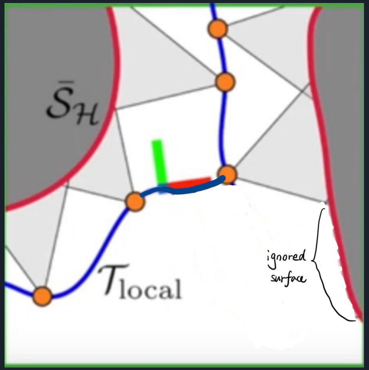
Main Idea

Shorten the travel distance and increase efficiency by ignoring some time-consuming exploration.



TARE

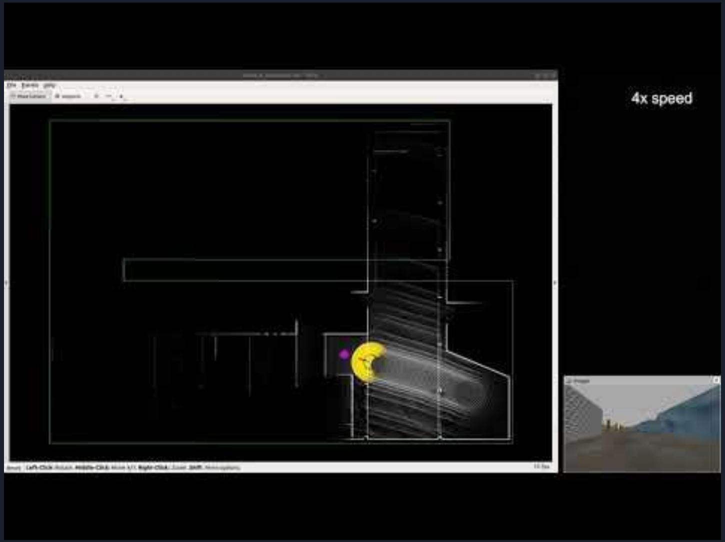
VS



SAE

Autonomous Exploration Development Environment (from CMU)

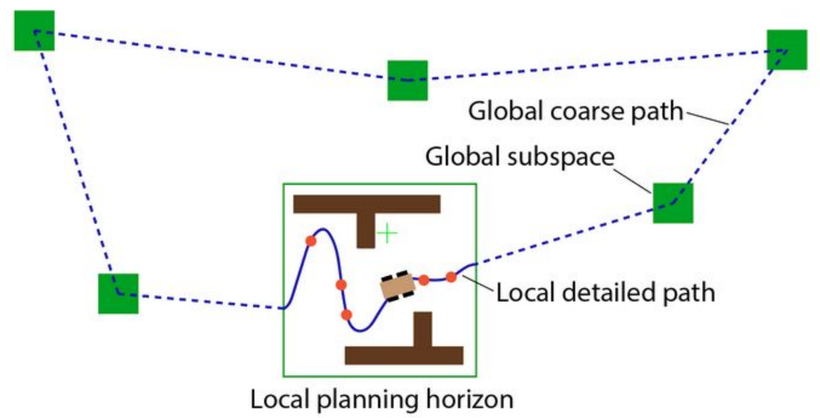
- Simulation environments
- Autonomous navigation modules
  - Collision avoidance
  - Terrain traversability analysis
  - Waypoints following



TARE Exploration Planner (from CMU)

Local: dense data

Global: sparse data

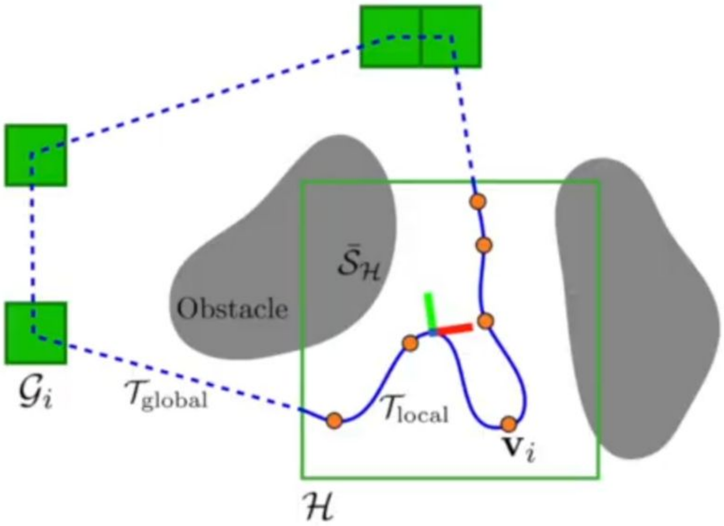


TARE hierarchical exploration framework. Inside the local planning horizon, data is densely maintained and a local detailed path (dark-blue) is computed. At the global scale, data is sparsely maintained in the distant subspaces and a global coarse path (light-blue) is computed. The local path and global path are connected on the boundary of the local planning horizon to form the exploration path.

Reference: https://www.cmu-exploration.com/tare-planner

Our Approach - Global Planning

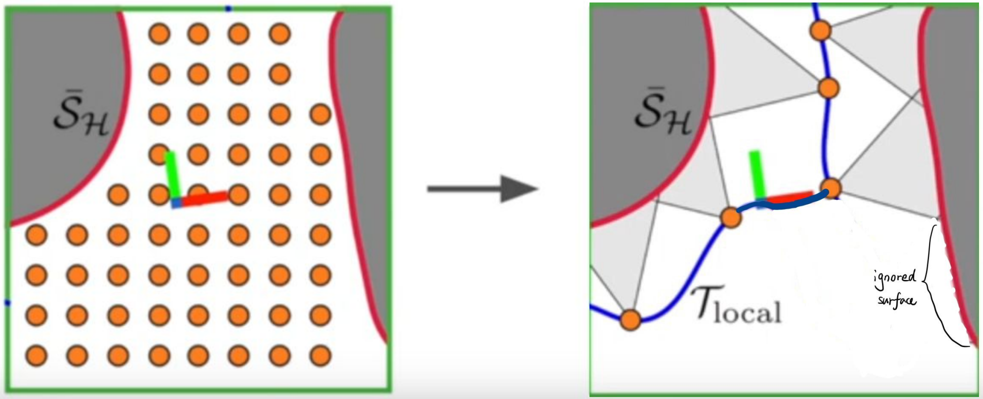
- Maintain sparse information outside the local area
  a. Divide the space outside the local area into subspaces
  b. Compute a TSP tour visiting all unexplored subspaces
  c. Concatenate the global and local TSP tours

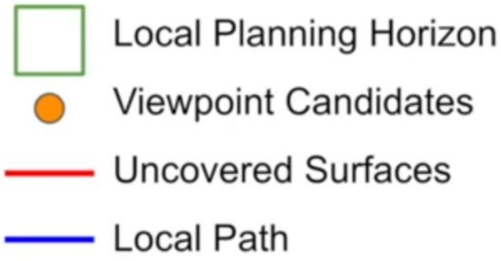


Reference:
https://www.youtube.com/watch?v=v9dRqpp5QXQ

Our Approach - Local Planning

- Uniformly generate viewpoint candidates in a local planning horizon H
- Solve for a shortest path that gives a full coverage of H:
  a. Iterate K times:
     - Select a min-cardinality set of viewpoints that gives a full coverage inside H
     - Compute a TSP tour visiting all selected viewpoints that aligns with the global guidance (next slide)
     - Keep track of the shortest TSP tour found so far





Reference: https://www.youtube.com/watch?v=v9dRqpp5QXQ

Algorithm 1: Compute Local Path

input : traversable C-space $\mathcal{C}^{\mathcal{H}}_{\text{trav}}$, uncovered surfaces $\bar{\mathcal{S}}_{\mathcal{H}}$, current viewpoint $\mathbf{v}_{\text{current}}$, boundary viewpoints $\mathbf{v}^1_{\text{boundary}}$ and $\mathbf{v}^2_{\text{boundary}}$
output: local path $\mathcal{T}_{\text{local}}$

Generate a set of viewpoint candidates $\mathcal{V}$ in $\mathcal{C}^{\mathcal{H}}_{\text{trav}}$;
Initialize priority queue Q;
For every $\mathbf{v} \in \mathcal{V}$, estimate coverage $\bar{\mathcal{S}}_{\mathbf{v}}$, then push $\mathbf{v}$ into Q with the priority set to its reward $A_{\mathbf{v}}$;
$\mathcal{T}_{\text{local}} \leftarrow \emptyset$, $c_{best} \leftarrow +\infty$;
for $i := 1$ to K **do**
  $Q' \leftarrow Q$, $\mathcal{V}' \leftarrow \{\mathbf{v}_{\text{current}}, \mathbf{v}^1_{\text{boundary}}, \mathbf{v}^2_{\text{boundary}}\}$;
  while $Q' \neq \emptyset$ and Q' contains at least one non-zero priority **do**
    Probabilistically pick viewpoint $\mathbf{v}'$ in Q', then remove $\mathbf{v}'$ from Q';
    $\mathcal{V}' \leftarrow \mathcal{V}' \cup \mathbf{v}'$;
    Update priorities for all viewpoints in Q';
  end
  Compute smooth path $\mathcal{T}'_{\text{smooth}}$ and cost c'_{smooth} using Algorithm 2;
  if $c'_{\text{smooth}} < c_{\text{best}}$ **then**
    $\mathcal{T}_{\text{local}} \leftarrow \mathcal{T}'_{\text{smooth}}$, $c_{\text{best}} \leftarrow c'_{\text{smooth}}$;
  end
end
return $\mathcal{T}_{\text{local}}$;

Algorithm 2: Selective Autonomous Exploration Compute Local Path

input : traversable C-space $\mathcal{C}^{\mathcal{H}}_{\text{trav}}$, uncovered surfaces $\bar{\mathcal{S}}_{\mathcal{H}}$, current viewpoint $\mathbf{v}_{\text{current}}$, boundary viewpoints $\mathbf{v}^1_{\text{boundary}}$ and $\mathbf{v}^2_{\text{boundary}}$
output: local path $\mathcal{T}_{\text{local}}$

Generate a set of viewpoint candidates $\mathcal{V}$ in $\mathcal{C}^{\mathcal{H}}_{\text{trav}}$;
Initialize priority queue Q;
For every $\mathbf{v} \in \mathcal{V}$, estimate coverage $\bar{\mathcal{S}}_{\mathbf{v}}$, then push $\mathbf{v}$ into Q with the priority set to its reward $A_{\mathbf{v}}$;
$\mathcal{T}_{\text{local}} \leftarrow \emptyset$, $c_{best} \leftarrow +\infty$;
for $i := 1$ to K **do**
  $Q' \leftarrow Q$, $\mathcal{V}' \leftarrow \{\mathbf{v}_{\text{current}}, \mathbf{v}^1_{\text{boundary}}, \mathbf{v}^2_{\text{boundary}}\}$;
  while $Q' \neq \emptyset$ and Q'.top().Av > maxIgnore do
    Probabilistically pick viewpoint $\mathbf{v}'$ in Q', then remove $\mathbf{v}'$ from Q';
    $\mathcal{V}' \leftarrow \mathcal{V}' \cup \mathbf{v}'$;
    Update priorities for all viewpoints in Q';
  end
  Compute smooth path $\mathcal{T}'_{\text{smooth}}$ and cost c'_{smooth} using Algorithm 2;
  if $c'_{\text{smooth}} < c_{\text{best}}$ **then**
    $\mathcal{T}_{\text{local}} \leftarrow \mathcal{T}'_{\text{smooth}}$, $c_{\text{best}} \leftarrow c'_{\text{smooth}}$;
  end
end
return $\mathcal{T}_{\text{local}}$;

Algorithm 3: Compute Exploration Path

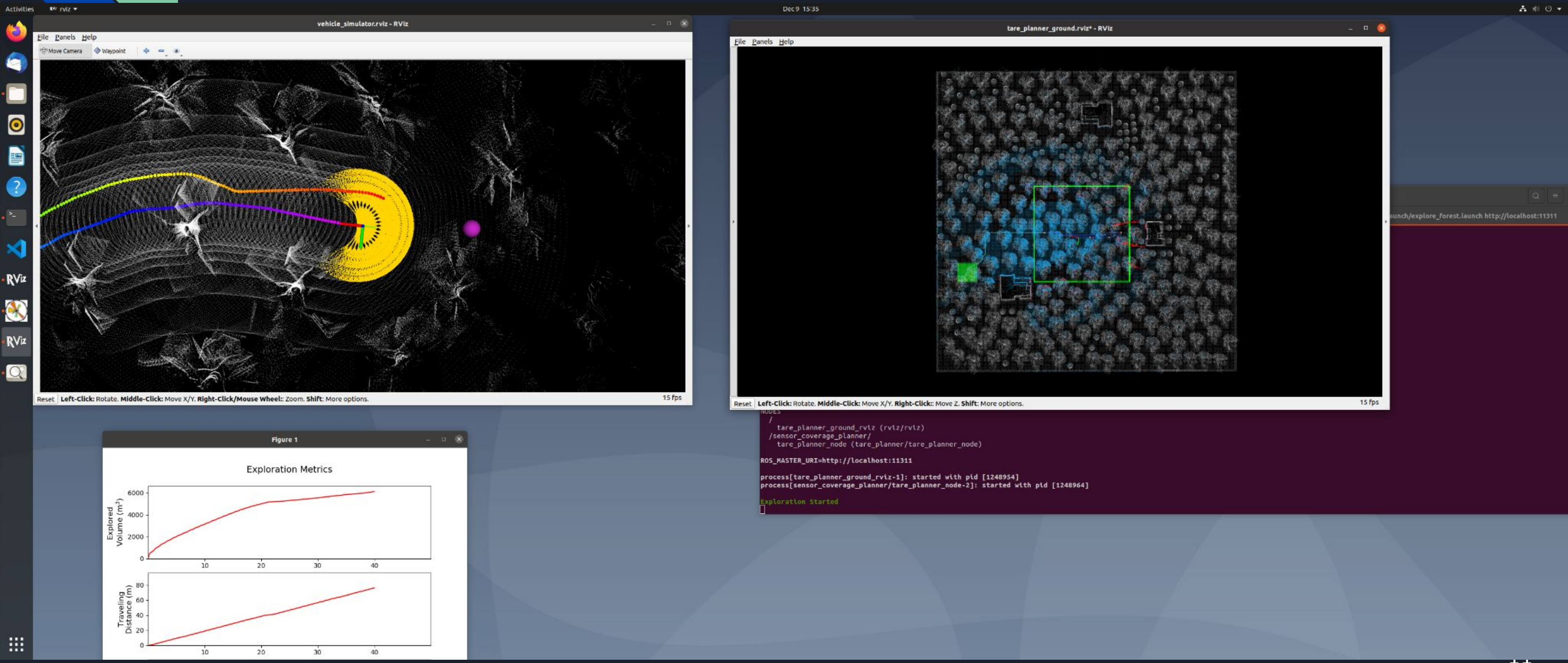
input : local planning horizon $\mathcal{H}$, traversable C-space $\mathcal{C}^{\mathcal{H}}_{\text{trav}}$, uncovered surfaces $\bar{\mathcal{S}}_{\mathcal{H}}$, exploring subspaces $\hat{\mathcal{G}}$, current viewpoint $\mathbf{v}_{\text{current}}$
output: exploration path $\mathcal{T}$

Compute shortest paths between centroids in $\hat{\mathcal{G}}$ and $\mathbf{v}_{\text{current}}$, then create distance matrix $\mathbf{D}$;
Compute global path $\mathcal{T}_{\text{global}}$ by solving TSP using $\mathbf{D}$;
Extract $\mathbf{v}^1_{\text{boundary}}$ and $\mathbf{v}^2_{\text{boundary}}$ as the intersections between $\mathcal{T}_{\text{global}}$ and the boundary of $\mathcal{H}$;
Compute local path $\mathcal{T}_{\text{local}}$ using Algorithm 1;
Concatenate $\mathcal{T}_{\text{local}}$ and $\mathcal{T}_{\text{global}}$ to generate $\mathcal{T}$;
return $\mathcal{T}$;

Experiment

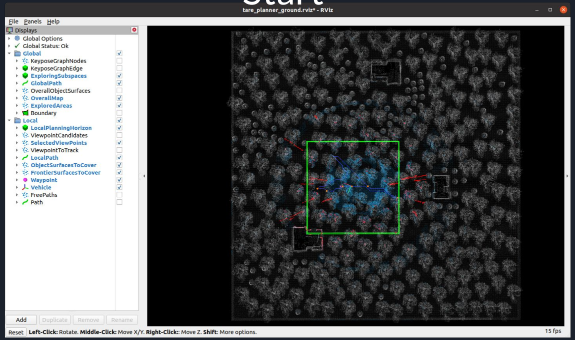
- Compare SAE with the original TARE
- Metrics:
  - exploration_volume (m^3)
  - travel_distance (m)
  - exploration_time (second)
  - coverage_Rate = exploration_volumn/exploration_time (m^3/s)

Experiment (bases on Autonomous exploration development environment and TARE)

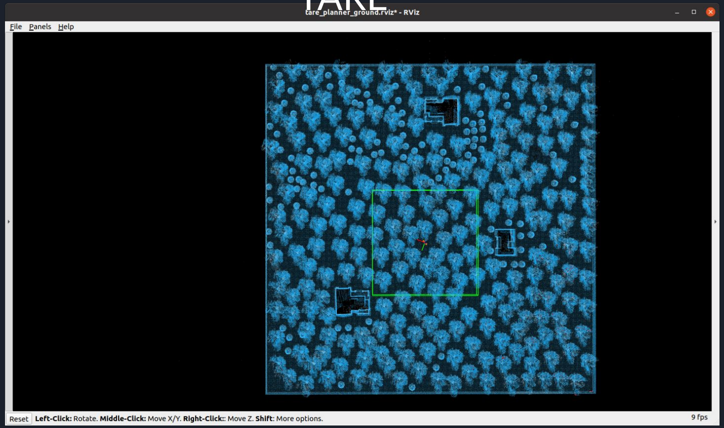


Results

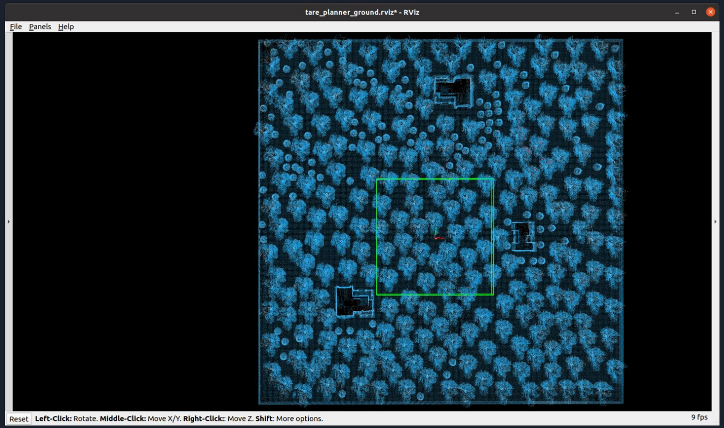
Start



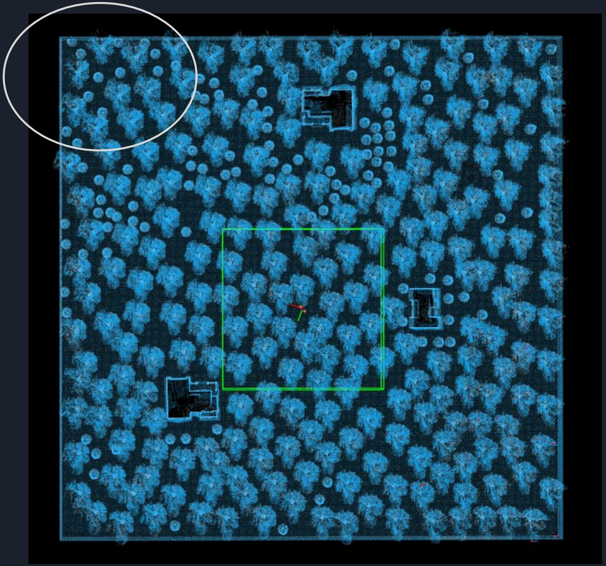
TARE



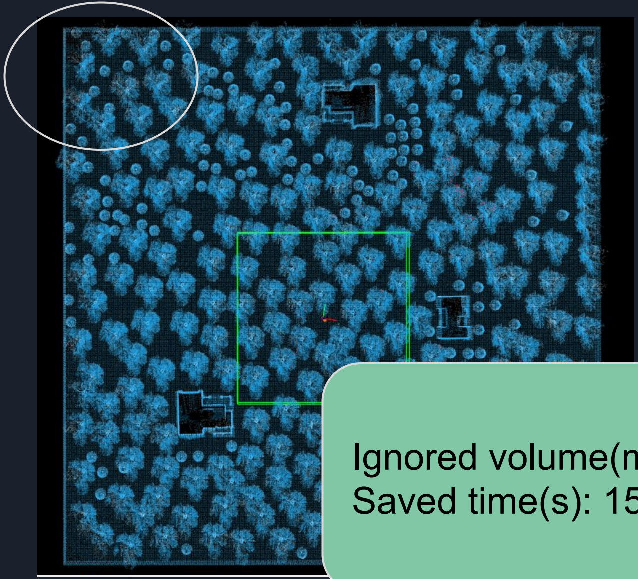
SAE



Results



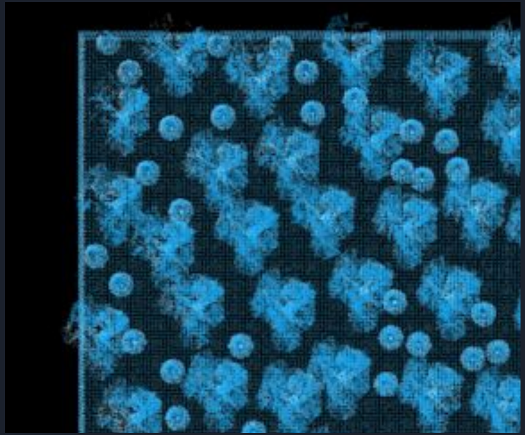
TARE:
Volumn(m^3): 39688
Distance(m): 1368
Time(s): 818
Coverage rate(m^3/s): 48.52

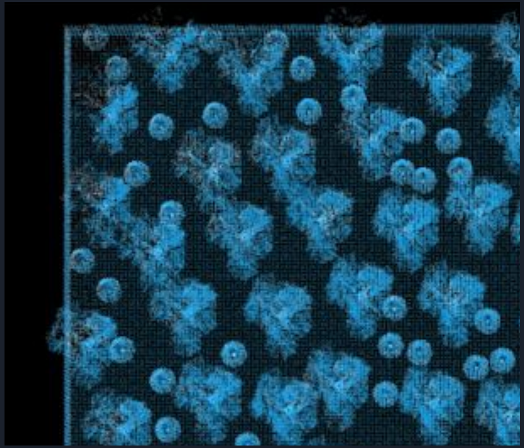


Ignored volume(m^3): 698
Saved time(s): 157

SAE:
Volumn(m^3): 38990
Distance(m): 1215
Time(s): 661
Coverage rate(m^3/s): 58.99

Results

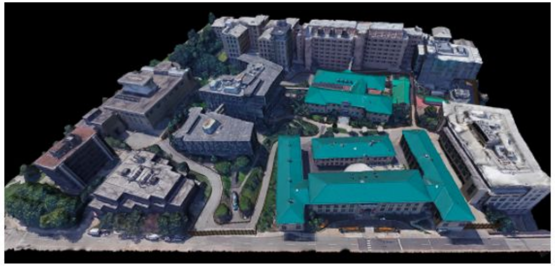




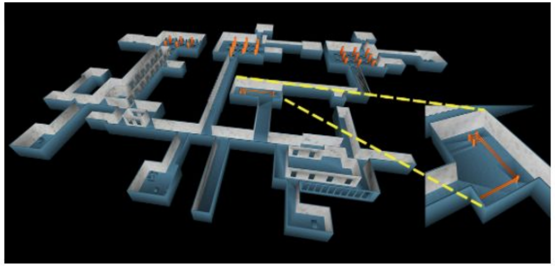
Blue

Light blue = Blue + White

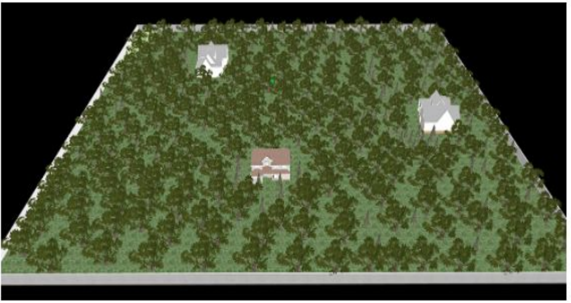
5 Environments



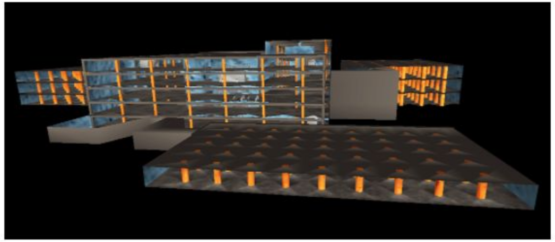
Campus (340m x 340m)



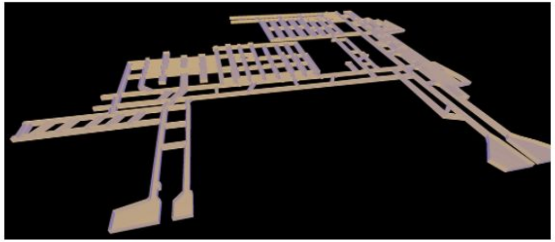
Indoor Corridors (130m x 100m)



Forest (150m x 150m)



Multi-storage Garage (140m x 130m, 5 floors)



Tunnel Network (330m x 250m)

Results on 5 Environments

Env / CR (m^3/s)	Forest	Indoor	Tunnel	Campus	Garage
TARE	48.52	9.01	7.68	34.83	19.58
SAE	58.99	9.38	8.29	39.69	20.51
ImproveRate	21.6%	4.1%	7.9%	14.0%	4.4%

Coverage Rate(CR) = exploration_volumn/exploration_time
ImproveRate = (SAE_CR-TARE_CR)/TARE_CR * 100%

Future Work

- Real-world experiments.
- Multiple vehicles explore the environment at the same time.

Thank you!