BadgerWorks

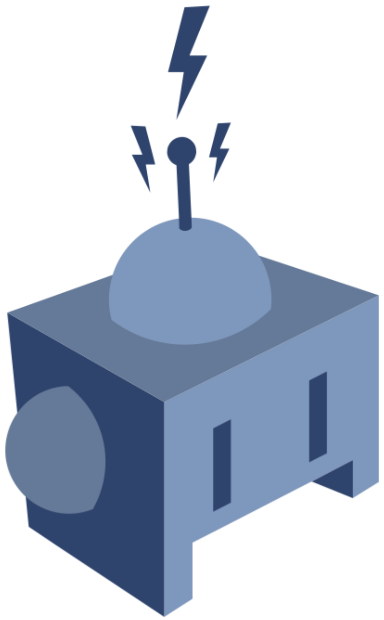
Topic: Exploration and Inspection Path Planning

Dr. Kostas Alexis (CSE)

AUTONOMOUS
ROBOTS
LAB

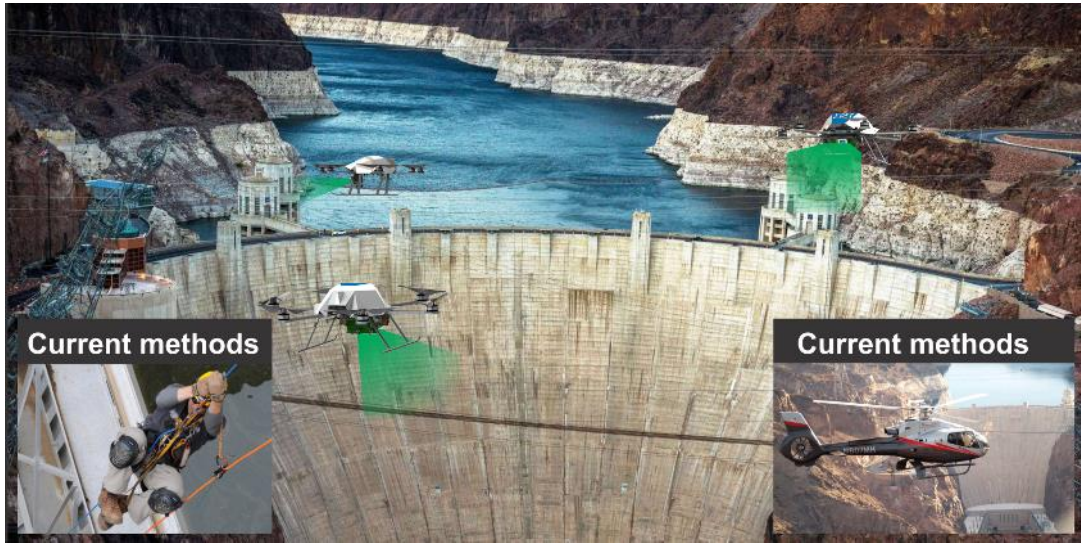
Autonomous Robot Challenges

How to explore and inspect structures and environments?



Motivation

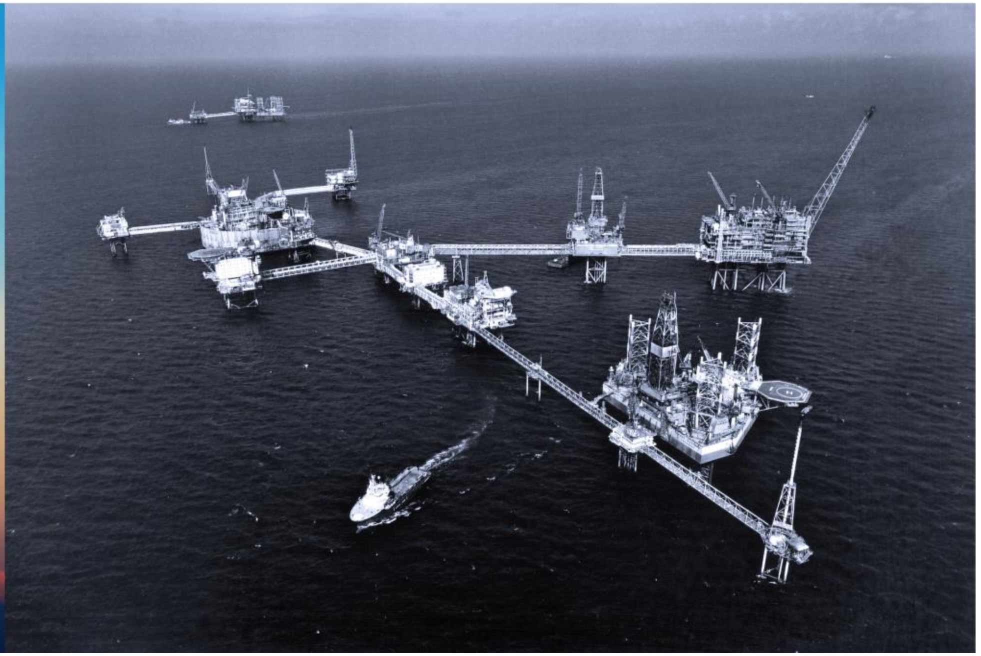
- Autonomous Exploration and Inspection of even unknown or partially known environments.
- Autonomous complete coverage 3D **structural path planning**
- Enable real-time dense reconstruction of infrastructure
- Consistent mapping and re-mapping of infrastructure to derive models and detect change
- Long-endurance mission by exploiting the ground robot battery capacity
- Aerial robots that autonomously inspect our infrastructure or fields, detect changes and risks.



Real-life is 3D, Complex, Possibly unknown

Known Model to Compute Global Inspection Path

Unknown Model – execute Autonomous Exploration



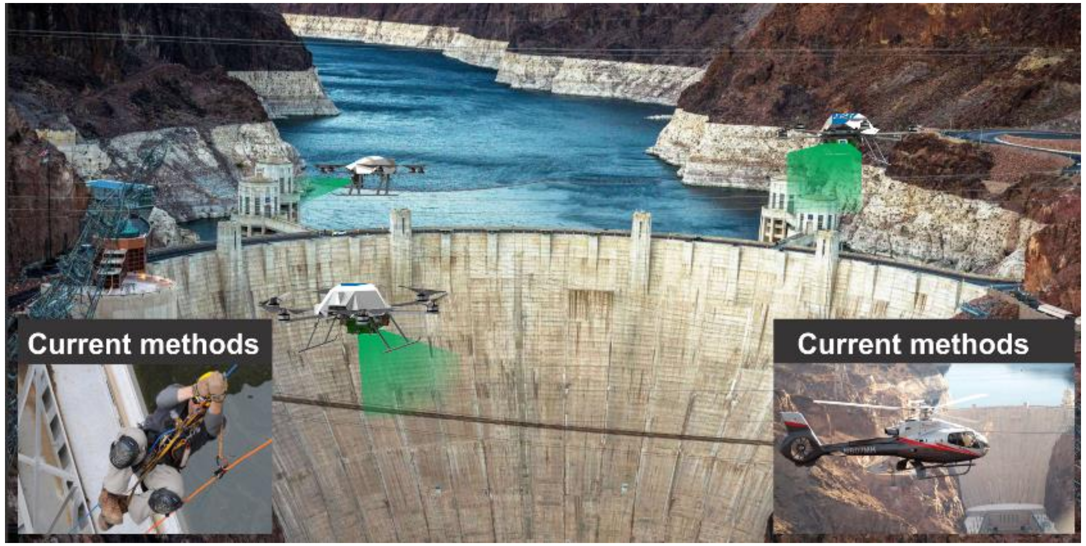
Real-life is 3D, Complex, Possibly unknown

Known Model to Compute Global Inspection Path

Unknown Model – execute Autonomous Exploration

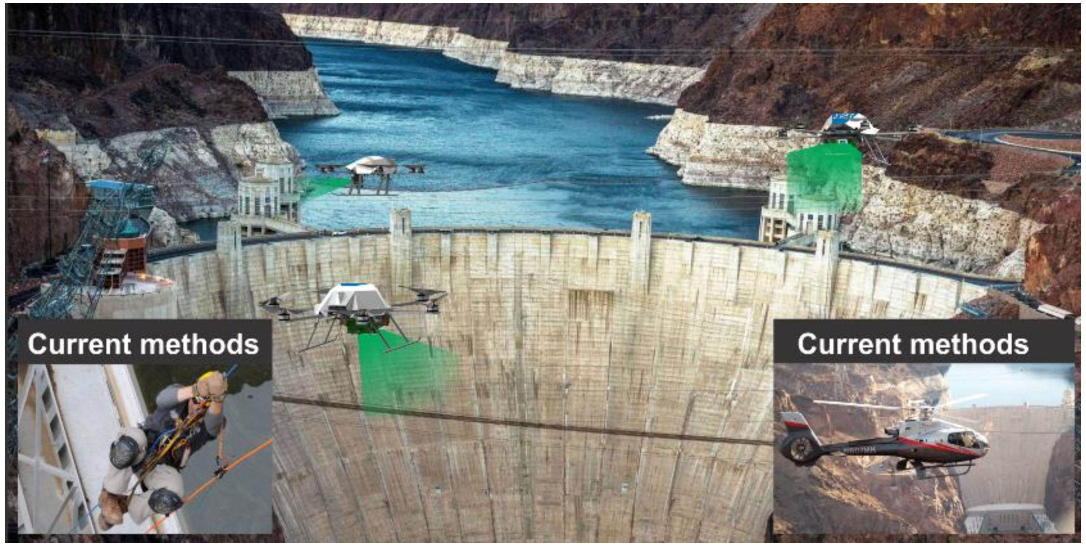
What is exploration?

- How robots map an unknown area in order to determine the conditions and characteristics of the environment (typically: to map it).



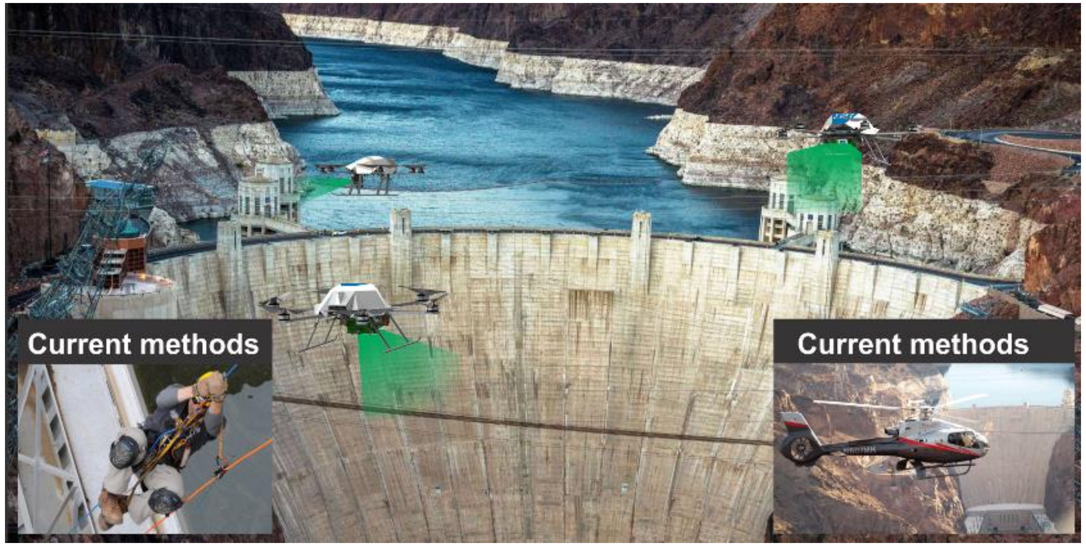
Exploration is different than Coverage

- Coverage problems assume that the map is known and the objective is to optimally cover and/or possibly identify targets of interest in it.
- Exploration problems deal with how to map a previously unknown world!



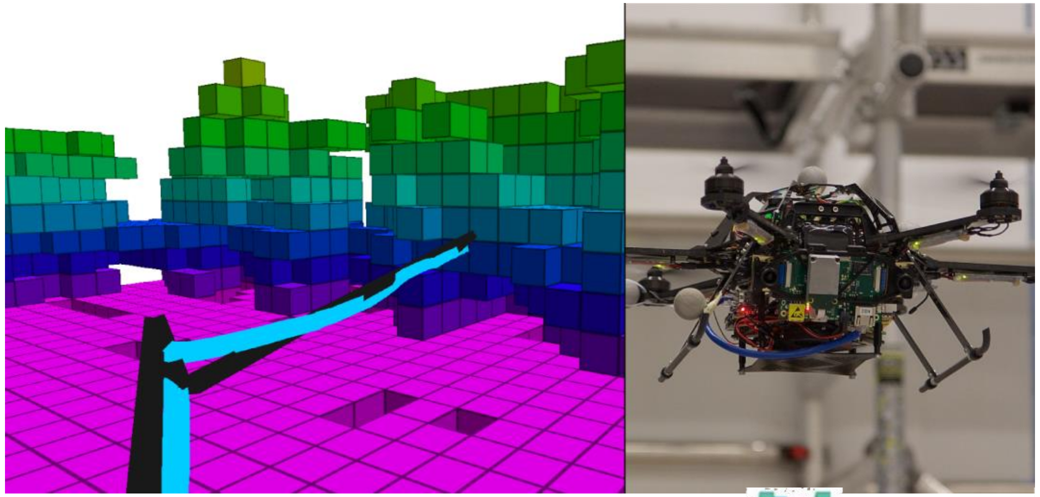
Applications of Autonomous Exploration

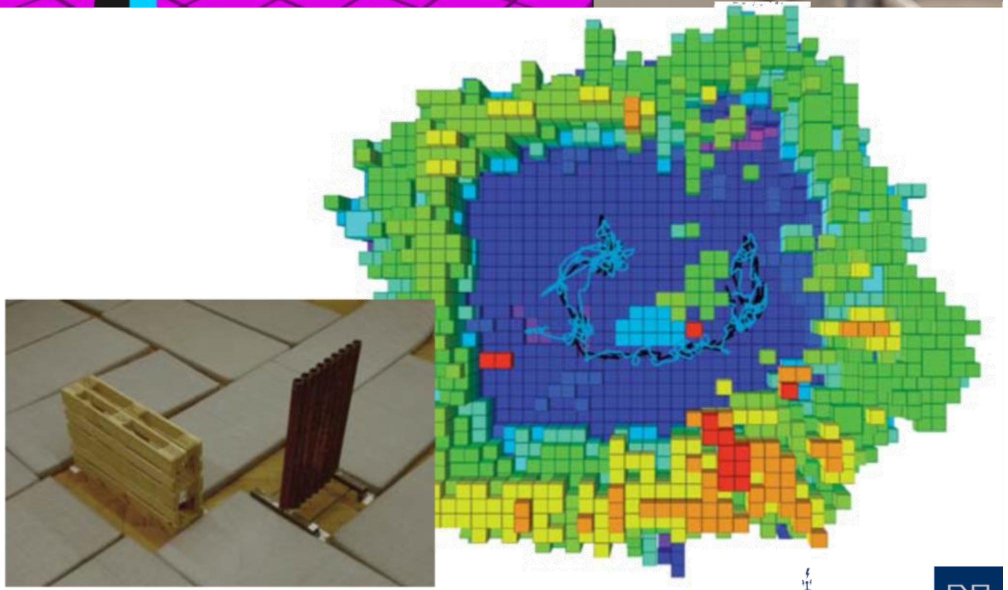
- Infrastructure monitoring and maintenance
- Rapid support of search and rescue operations
- Surveillance and reconnaissance
- Operation in any environment not suitable for human operators



Receding Horizon Next-Best-View Exploration

- Rapid exploration of unknown environments.
- Define **sequences of viewpoints based on vertices sampled using random trees**.
- Select the path with the best sequence of best views.
- Execute only the first step of this best exploration path.
- Repeat the whole process in a receding horizon fashion.





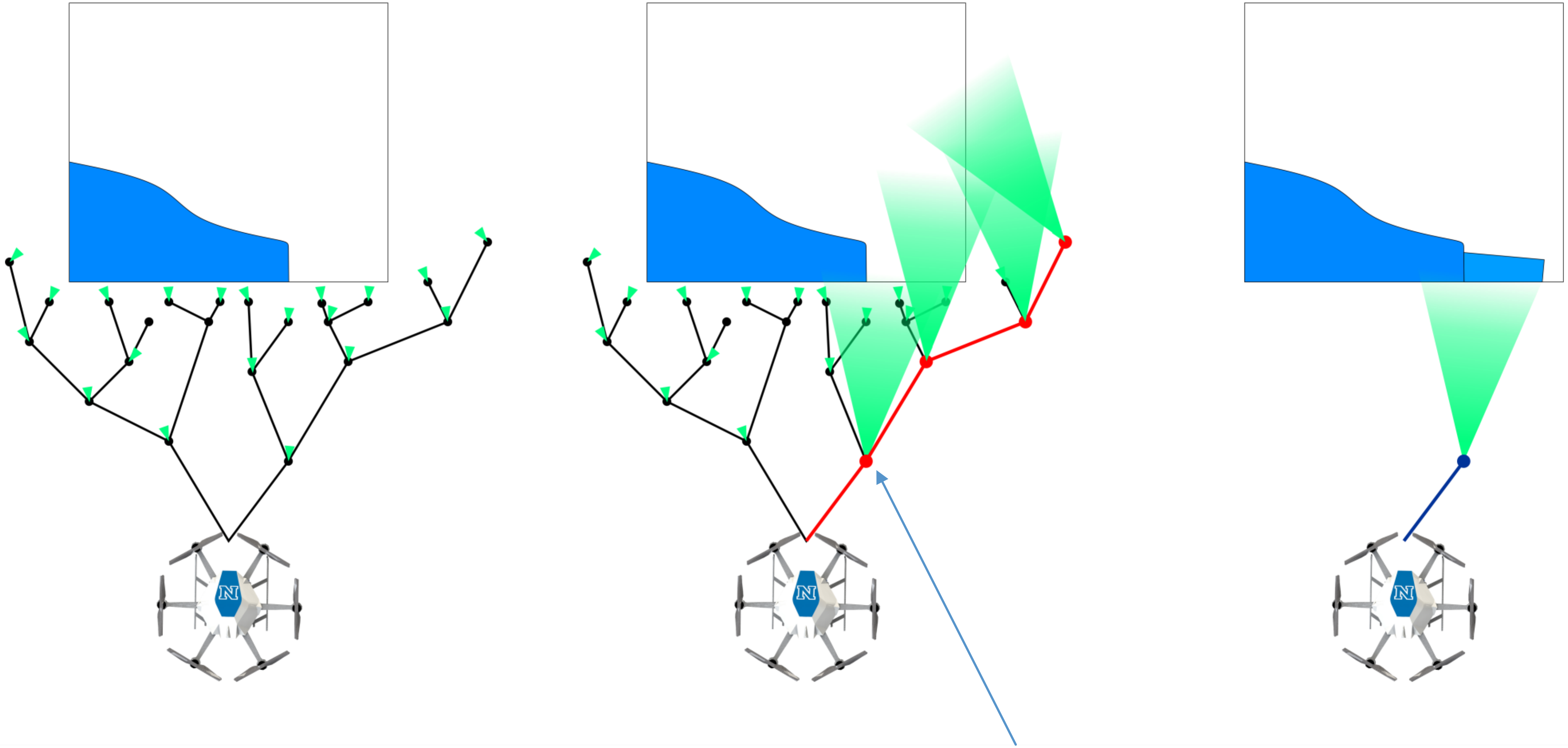
The Exploration path planning problem

Problem Definition

The exploration path planning problem consists in exploring a bounded 3D space $V \subset \mathbb{R}^3$. This is to determine which parts of the initially unmapped space $V_{unm} = V$ are free $V_{free} \subset V$ or occupied $V_{occ} \subset V$. The operation is subject to vehicle kinematic and dynamic constraints, localization uncertainty and limitations of the employed sensor system with which the space is explored.

- As for most sensors the perception stops at surfaces, hollow spaces or narrow pockets can sometimes not be explored with a given setup. This residual space is denoted as V_{res}. The problem is considered to be fully solved when $V_{free} \cup V_{occ} = V \backslash V_{res}$.
- Due to the nature of the problem, a suitable path has to be computed online and in real-time, as free space to navigate is not known prior to its exploration.

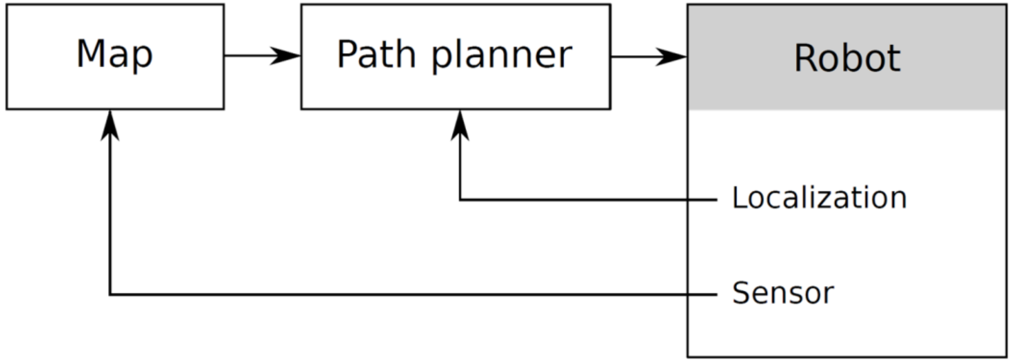
Exploration Planning (nbvplanner)



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

Exploration Planning (nbvplanner)

- **Environment representation:** Occupancy Map dividing space V into $m \in M$ cubical volumes (voxels) that can be marked either as free, occupied or unmapped.
- Array of voxels is saved in an octree structure to enable computationally efficient access and search.
- Paths are planned only within the free space V_{free} and collision-free point-to-point navigation is inherently supported.
- At each viewpoint/configuration of the environment ξ, the amount of space that is visible is computed as $Visible(M, \xi)$



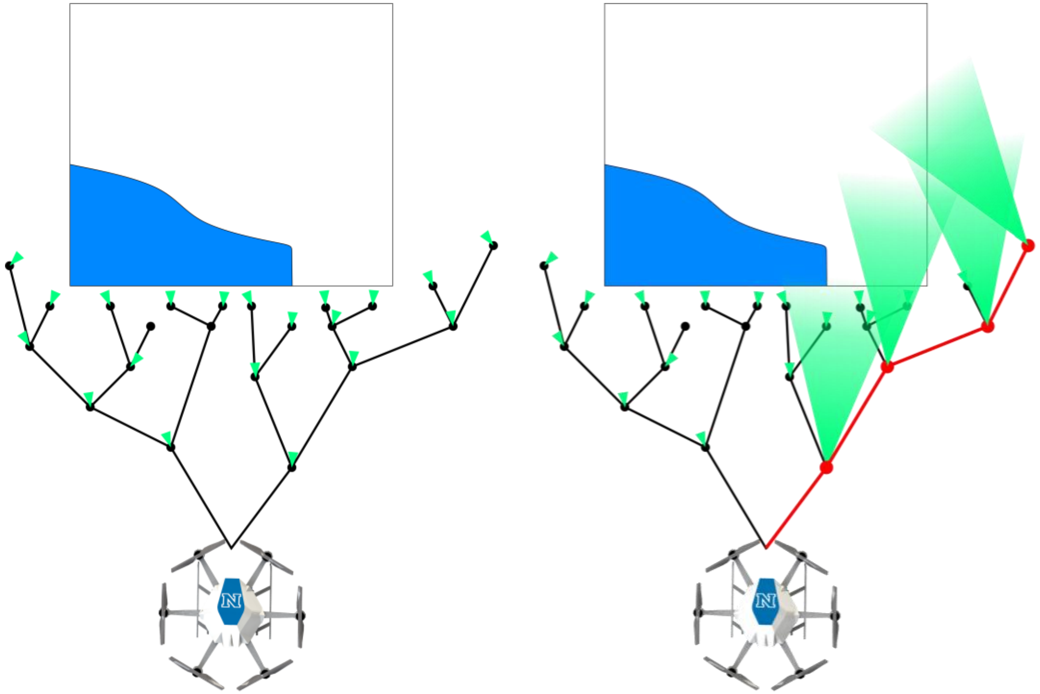
The Receding Horizon Next-Best-View Exploration Planner relies on the real-time update of the 3D map of the environment.

Exploration Planning (nbvplanner)

- **Tree-based exploration:** At every iteration, RH-NBVP spans a random tree of finite depth. Each vertex of the tree is annotated regarding the collected Information Gain – a metric of how much new space is going to be explored.

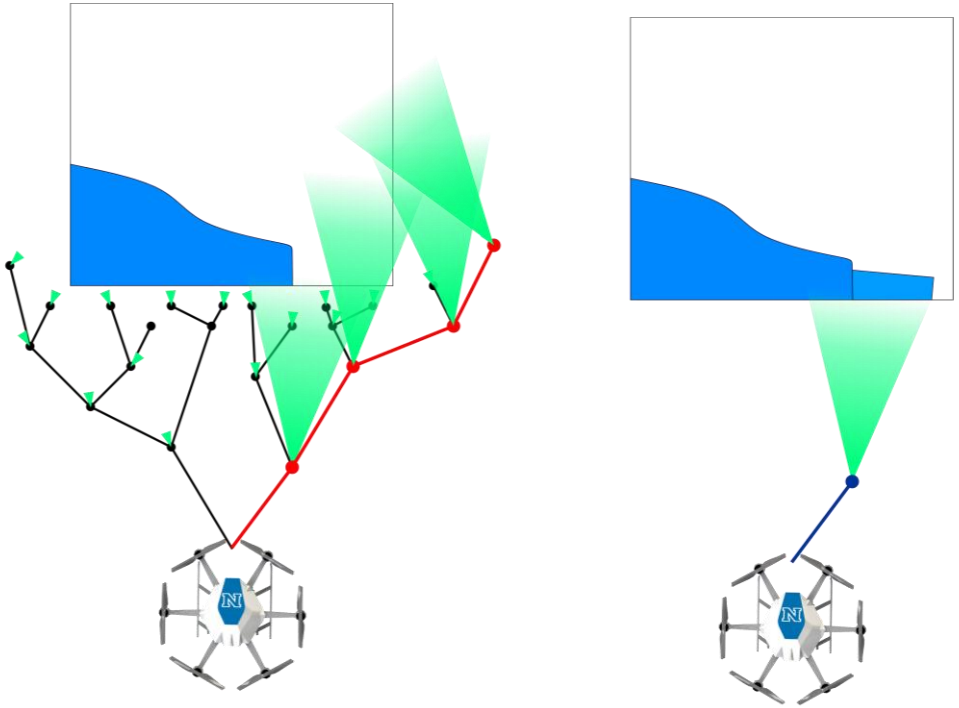
$$\mathbf{Gain}(n_k) = \mathbf{Gain}(n_{k-1}) + \mathbf{Visible}(\mathcal{M}, \xi_k) e^{-\lambda c(\sigma^k_{k-1})}$$

- Within the sampled tree, evaluation regarding the path that overall leads to the highest information gain is conducted. This corresponds to the **best path** for the given iteration. It is a sequence of next-best-views as sampled based on the vertices of the spanned random tree.



Exploration Planning (nbvplanner)

- **Receding Horizon:** For the extracted best path of viewpoints, only the first viewpoint is actually executed.
- The system moves to the first viewpoint of the path of best viewpoints.
- Subsequently, the whole process is repeated within the next iteration. This gives rise to a receding horizon operation.



nbvplanner Algorithm

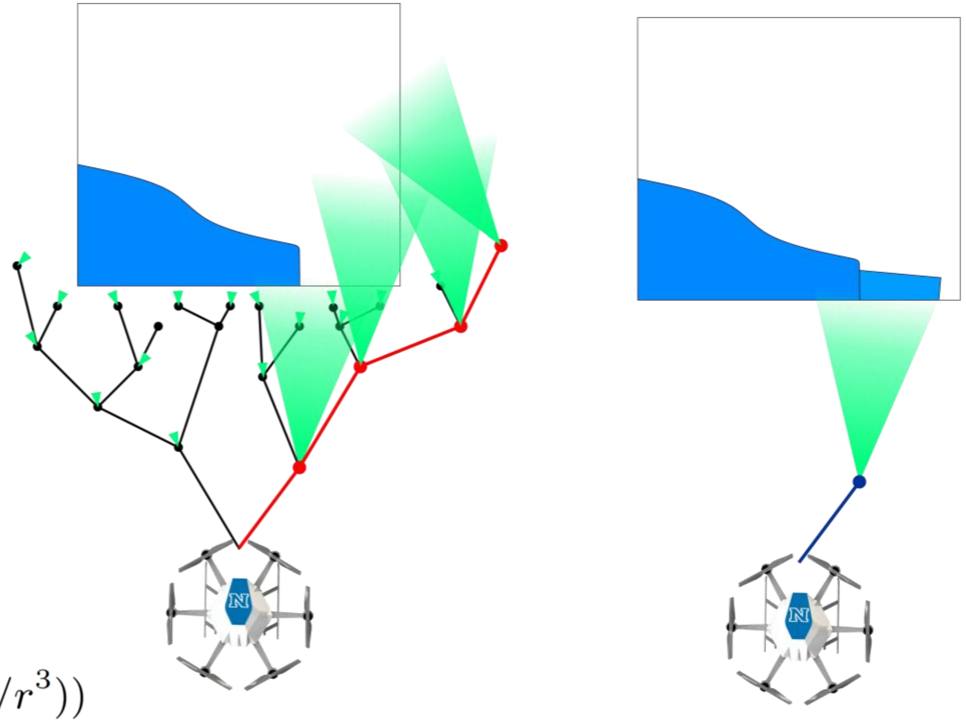
NBVP Iterative Step

- ξ_0 ←current vehicle configuration
- Initialize $\boldsymbol{T}$ with ξ_0 and, unless first planner call, also previous best branch
- $g_{best} \leftarrow 0$ // Set best gain to zero
- $n_{best} \leftarrow n_0(\xi_0)$ // Set best node to root
- N_T ←Number of nodes in $\boldsymbol{T}$
- **while** $N_T < N_{max}$ or $g_{best} == 0$ **do**
  - Incrementally build T by adding $n_{new}(\xi_{new})$
  - $N_T \leftarrow N_T + 1$
  - **if** $Gain(n_{new}) > g_{best}$ **then**
    - $n_{best} \leftarrow n_{new}$
    - $g_{best} \leftarrow Gain(n_{new})$
  - **if** $N_T > N_{TOT}$ **then**
    - Terminate exploration
- $\sigma \leftarrow \boldsymbol{ExtractBestPathSegment}(n_{best})$
- Delete $\boldsymbol{T}$
- **return** σ

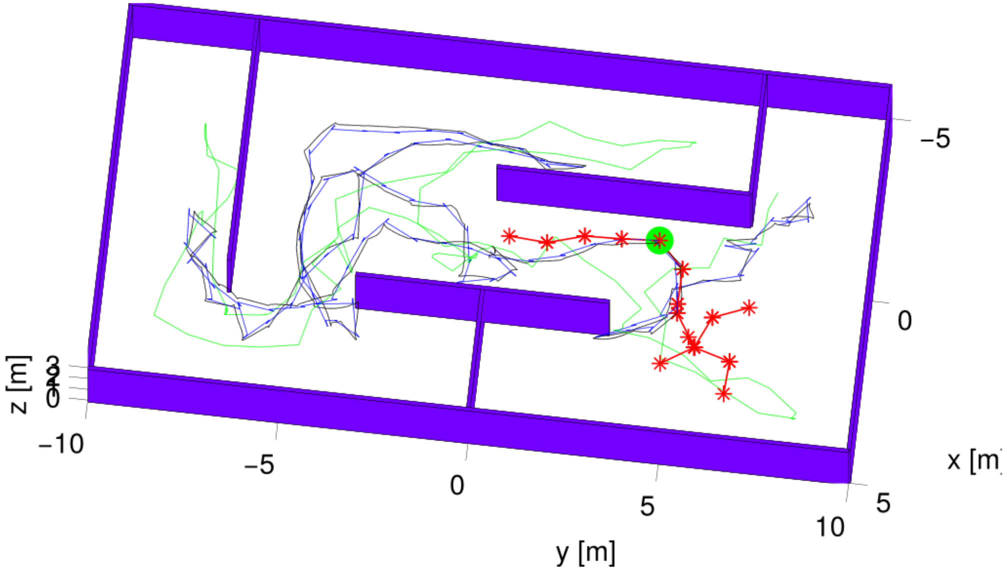
nbvplanner Remarks

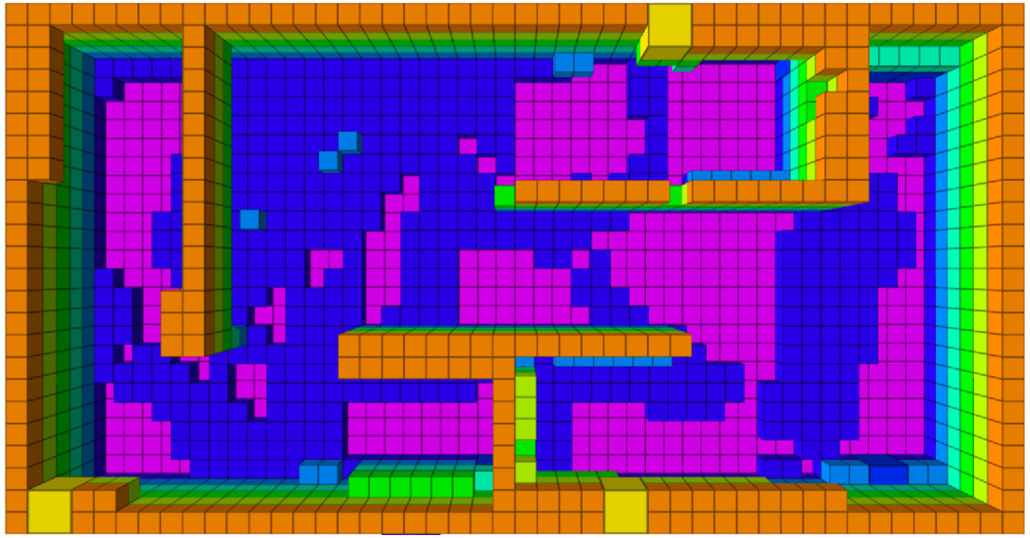
- **Inherently Collision-free:** As all paths of NBVP are selected along branches within RRT-based spanned trees, all paths are inherently collision-free.
- **Computational Cost:** NBVP has a thin structure and most of the computational cost is related with collision-checking functionalities. The formula that expresses the complexity of the algorithm takes the form:

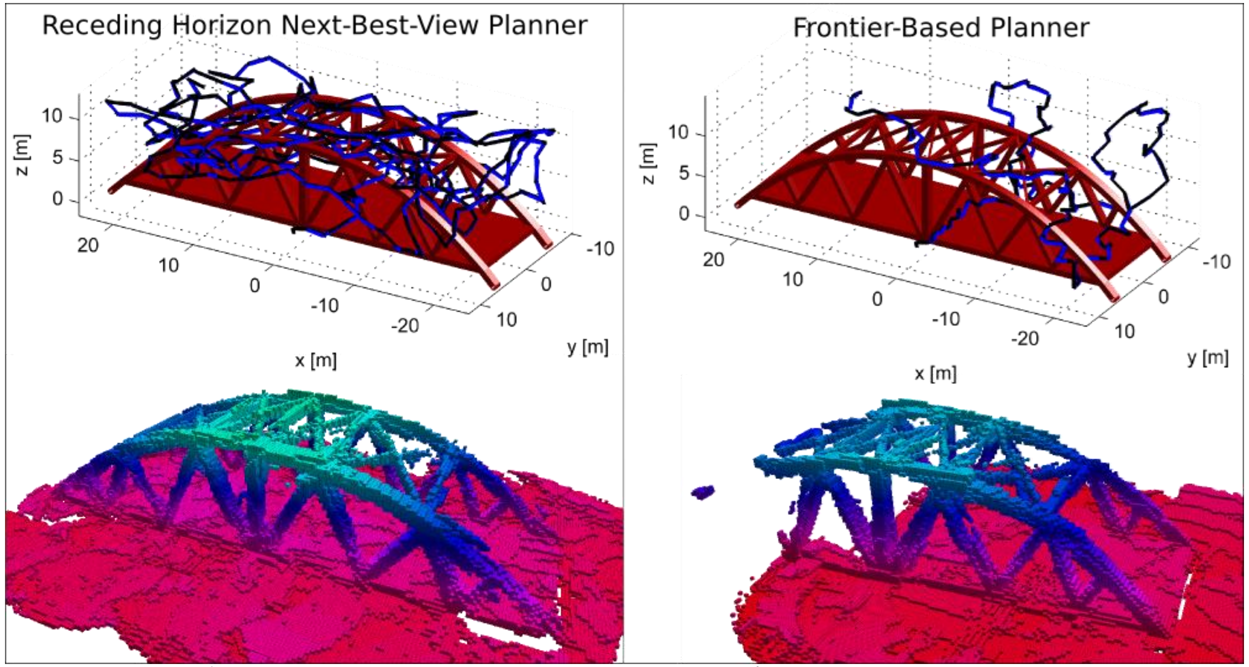
$$\mathcal{O}(N_{\mathbb{T}} \log(N_{\mathbb{T}}) + N_{\mathbb{T}}/r^3 \log(V/r^3) + N_{\mathbb{T}}(d_{\max}^{\text{planner}}/r)^4 \log(V/r^3))$$

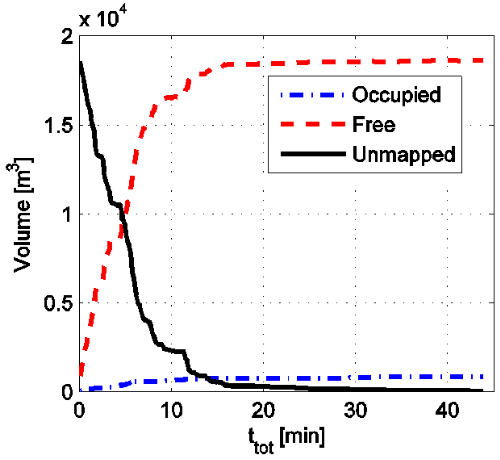


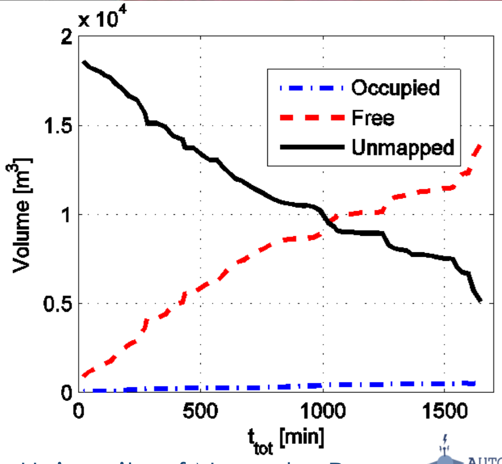
nbvplanner Evaluation (Simulation)



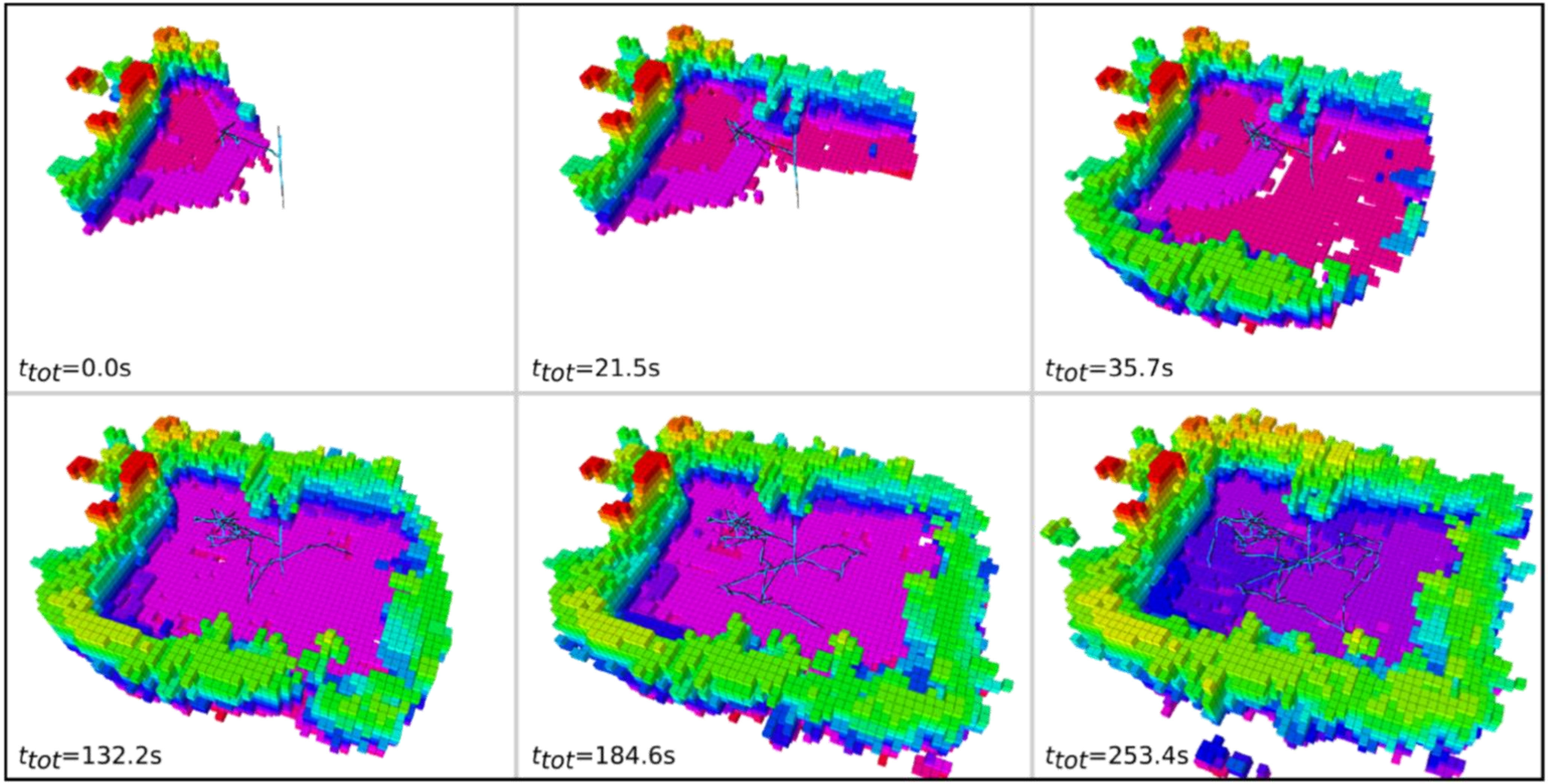




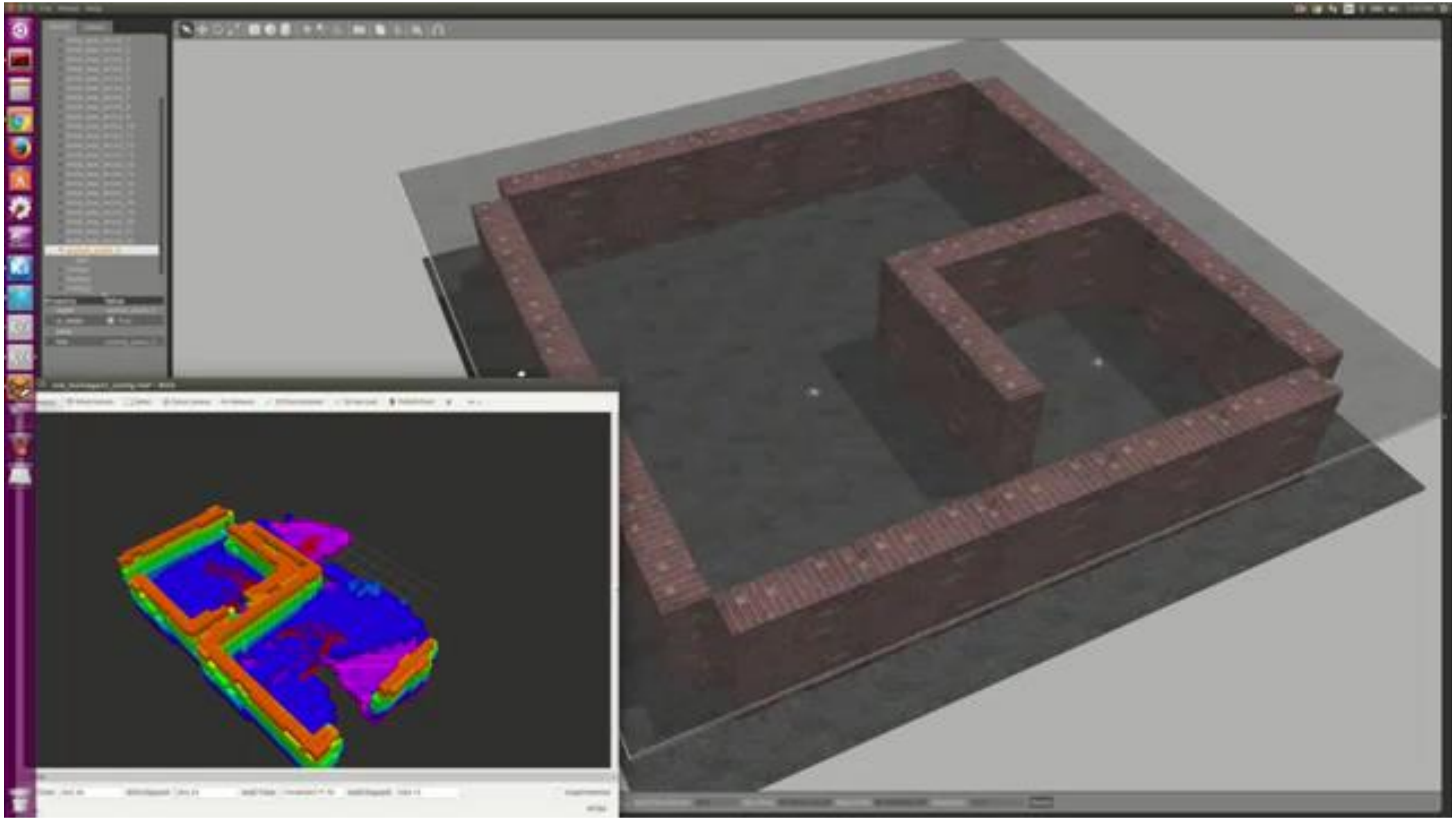




Nbvplanner Evaluation (Experiment)



Multi-Agent nbvplanner Simulation



Uncertainty-aware Exploration & Mapping

Problem Definition

The overall problem is that of exploring a bounded 3D volume $V^E \subset \mathbb{R}^3$, while aiming to minimize the localization and mapping uncertainty as evaluated through a metric over the robot pose and landmarks probabilistic belief.

Problem 1: Volumetric Exploration

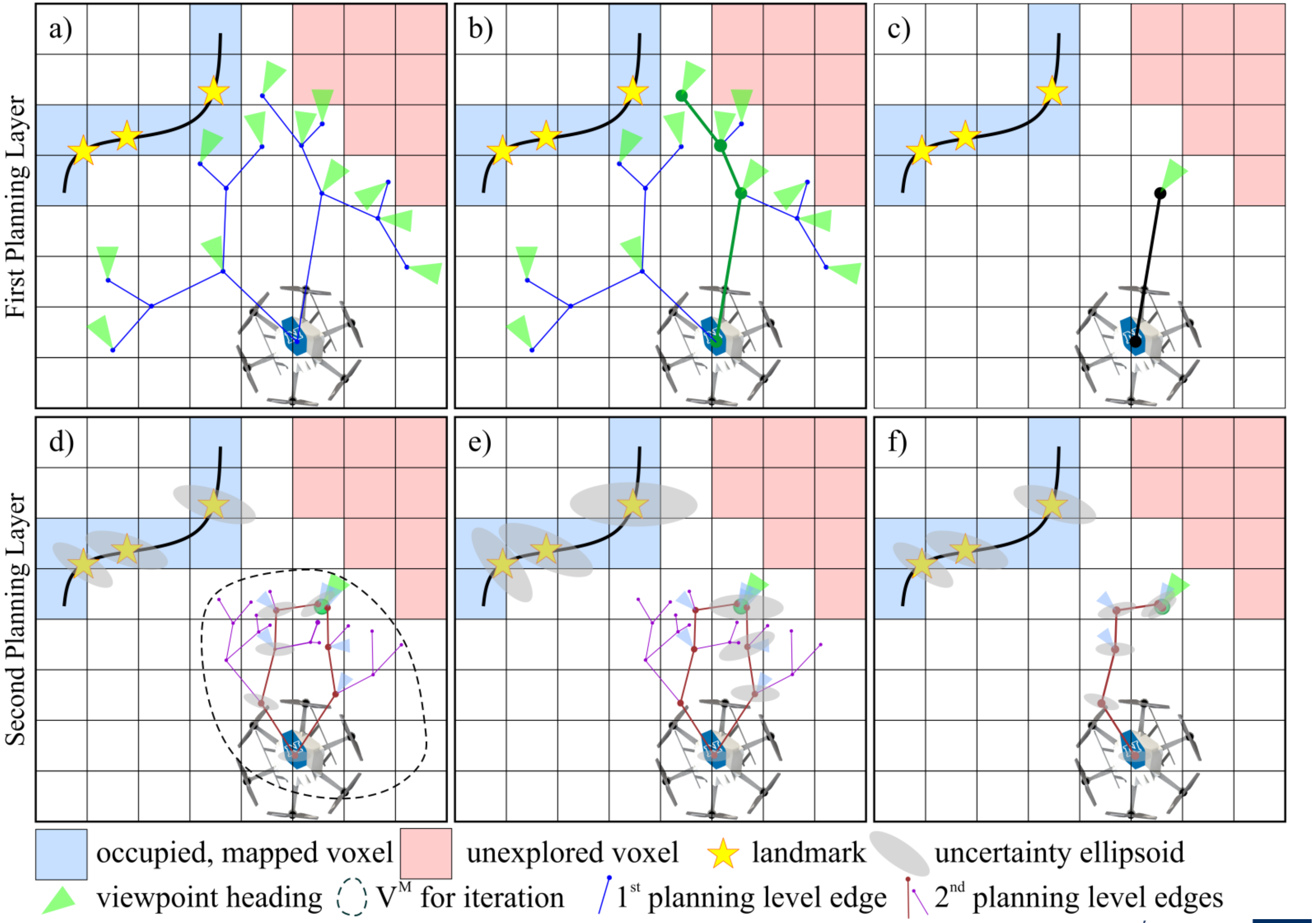
Given a bounded volume V^E, find a collision free path σ starting at an initial configuration $\xi_{init} \in \Xi$that leads to identifying the free and occupied parts V^E_{free} and V^E_{occ} when being executed, such that there does not exist any collision free configuration from which any piece of $V^E\{V^E_{free}, V^E_{occ}\}$ could be perceived.

Problem 2: Belief Uncertainty-aware planning

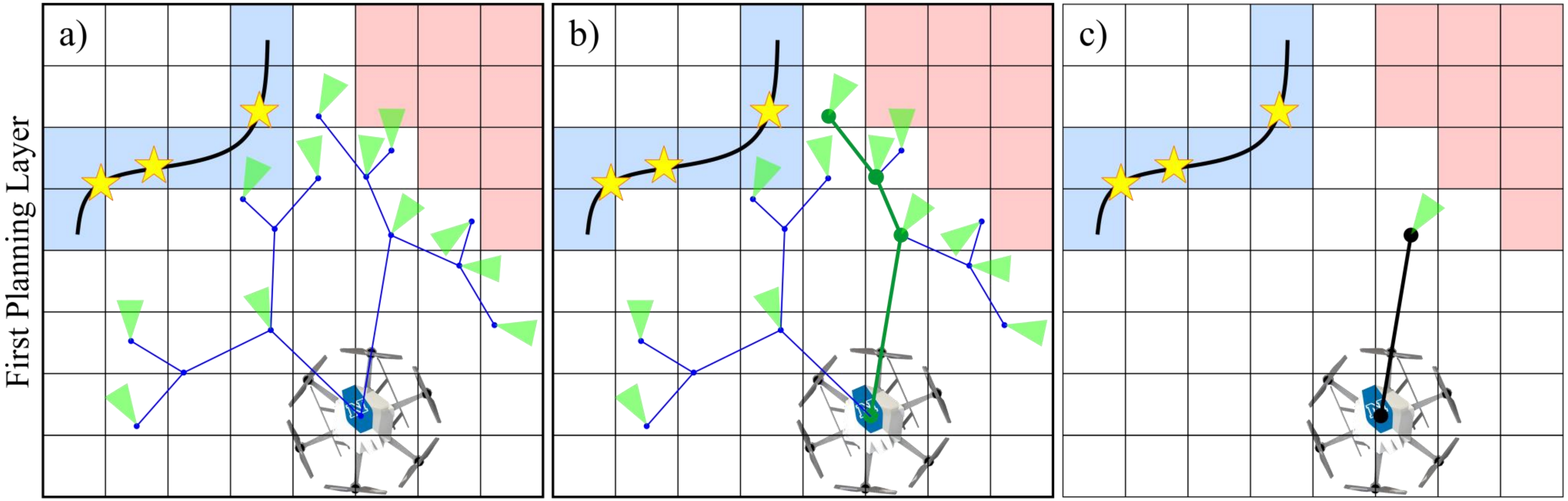
Given a $V^M \subset V^E$, find a collision free path σ^M starting at an initial configuration $\xi_0 \in \Xi$ and ending in a configuration $\xi_{final} \in \Xi$ that aims to improve the robot's localization and mapping confidence by following paths of optimized expected robot pose and tracked landmarks covariance.

Uncertainty-aware Exploration & Mapping

Receding Horizon Exploration and Mapping Planner (RHEM)



RHEM - Exploration Planning



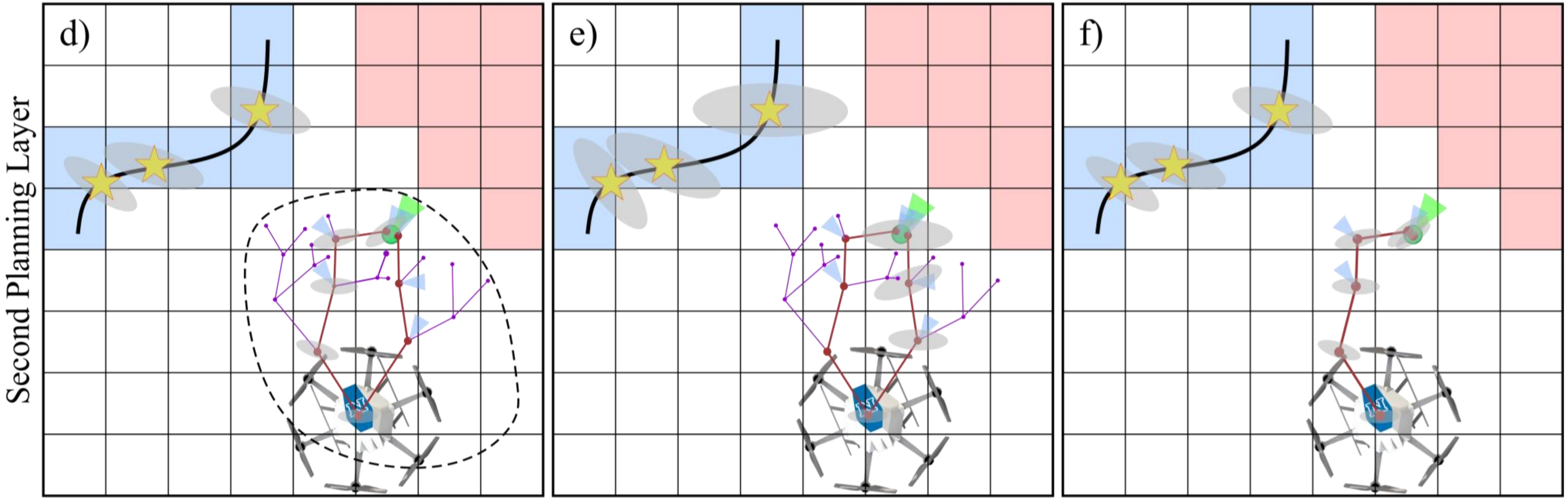
RHEM - Exploration Planning

- Exploration Gain

$$\begin{aligned}\mathbf{ExplorationGain}(n_k^E) = \mathbf{ExplorationGain}(n_{k-1}^E) + \\ \mathbf{VisibleVolume}(\mathcal{M}, \xi_k)\exp(-\lambda c(\sigma_{k-1,k}^E)) + \\ \mathbf{ReobservationGain}(\mathcal{M}, \mathcal{P}, \xi_k)\exp(-\lambda c(\sigma_{k-1,k}^E))\end{aligned}$$

  - Aiming to maximize newly explored space and reobserve space with decreased confidence of being mapped as occupied.

RHEM - Uncertainty-aware planning



RHEM - Uncertainty-aware planning

- The robot performs visual-inertial localization.
- To get an estimate about its pose, it relies on tracking features from its camera systems. The system performs odometry in an EKF-fashion and the overall state of the filter is:

$$\mathbf{x} = [\underbrace{\overbrace{\mathbf{r}\ \mathbf{q}}^{\text{pose},\ l_p}\ \boldsymbol{\upsilon}\ \mathbf{b}_f\ \mathbf{b}_\omega\ \mathbf{c}\ \mathbf{z}}_{\text{robot states},\ l_s} \mid \underbrace{\boldsymbol{\mu}_0, \cdots \boldsymbol{\mu}_J\ \rho_0 \cdots \rho_J}_{\text{features states},\ l_f}]^T$$

RHEM - Uncertainty-aware planning

- Belief Propagation: in order to identify the paths that minimize the robot uncertainty, a mechanism to propagate the robot belief about its pose and the tracked features has to be established.

State Propagation Step - Equations (3)

$$
\begin{aligned}
\dot{\mathbf{r}} &= -\hat{\boldsymbol{\omega}}^{\times}\mathbf{r} + \boldsymbol{\upsilon} + \mathbf{w}_r \\
\dot{\boldsymbol{\upsilon}} &= -\hat{\boldsymbol{\omega}}^{\times}\boldsymbol{\upsilon} + \hat{\mathbf{f}} + \mathbf{q}^{-1}(\mathbf{g}) \\
\dot{\mathbf{q}} &= -\mathbf{q}(\hat{\boldsymbol{\omega}}) \\
\dot{\mathbf{b}}_f &= \mathbf{w}_{bf} \\
\dot{\mathbf{b}}_\omega &= \mathbf{w}_{bw} \\
\dot{\mathbf{c}} &= \mathbf{w}_c \\
\dot{\mathbf{z}} &= \mathbf{w}_z \\
\dot{\boldsymbol{\mu}}_j &= \mathbf{N}^T(\boldsymbol{\mu}_j)\hat{\boldsymbol{\omega}}_{\mathcal{V}} - \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \mathbf{N}^T(\boldsymbol{\mu}_j)\frac{\hat{\boldsymbol{\upsilon}}_{\mathcal{V}}}{d(\rho_j)} + \mathbf{w}_{\mu,j} \\
\dot{\rho}_j &= -\boldsymbol{\mu}_j^T \hat{\boldsymbol{\upsilon}}_{\mathcal{V}}/d'(\rho_j) + w_{\rho,j}
\end{aligned}
$$

$$
\begin{aligned}
\hat{\mathbf{f}} &= \tilde{\mathbf{f}} - \mathbf{b}_f - \mathbf{w}_f \\
\hat{\boldsymbol{\omega}} &= \tilde{\boldsymbol{\omega}} - \mathbf{b}_\omega - \mathbf{w}_\omega \\
\hat{\boldsymbol{\upsilon}}_{\mathcal{V}} &= \mathbf{z}(\boldsymbol{\upsilon} + \hat{\boldsymbol{\omega}}^{\times}\mathbf{c}) \\
\hat{\boldsymbol{\omega}}_{\mathcal{V}} &= \mathbf{z}(\hat{\boldsymbol{\omega}})
\end{aligned}
$$

Filter Update Step - Equations (4)

$$
\begin{aligned}
\mathbf{y}_j &= \mathbf{b}_j(\boldsymbol{\pi}(\hat{\boldsymbol{\mu}}_j)) + \mathbf{n}_j \\
\mathbf{H}_j &= \mathbf{A}_j(\boldsymbol{\pi}(\hat{\boldsymbol{\mu}}_j))\frac{d\boldsymbol{\pi}}{d\boldsymbol{\mu}}(\hat{\boldsymbol{\mu}}_j)
\end{aligned}
$$

By stacking the above terms for all visible features, standard EKF update step is directly performed to derive the new estimate of the robot belief for its state and the tracked features.

Notation

$^{\times} \rightarrow$ skew symmetric matrix of a vector, $\mathbf{f} \rightarrow$ proper acceleration measurement, $\tilde{\boldsymbol{\omega}} \rightarrow$ rotational rate measurement, $\hat{\mathbf{f}} \rightarrow$ biased corrected acceleration, $\hat{\boldsymbol{\omega}} \rightarrow$ bias corrected rotational rate, $\mathbf{N}^T(\boldsymbol{\mu}) \rightarrow$ projection of a 3D vector onto the 2D tangent space around the bearing vector, $\mathbf{g} \rightarrow$ gravity vector, $\mathbf{w}_\star \rightarrow$ white Gaussian noise processes, $\boldsymbol{\pi}(\boldsymbol{\mu}) \rightarrow$ pixel coordinates of a feature, $\mathbf{b}_i(\boldsymbol{\pi}(\hat{\boldsymbol{\mu}}_j)) \rightarrow$ a 2D linear constraint for the j^{th} feature which is predicted to be visible in the current frame with bearing vector $\hat{\boldsymbol{\mu}}_j$

RHEM - Uncertainty-aware planning

State Propagation Step - Equations (3)

$$
\begin{aligned}
\dot{\mathbf{r}} &= -\hat{\boldsymbol{\omega}}^{\times}\mathbf{r} + \boldsymbol{\upsilon} + \mathbf{w}_r \\
\dot{\boldsymbol{\upsilon}} &= -\hat{\boldsymbol{\omega}}^{\times}\boldsymbol{\upsilon} + \hat{\mathbf{f}} + \mathbf{q}^{-1}(\mathbf{g}) \\
\dot{\mathbf{q}} &= -\mathbf{q}(\hat{\boldsymbol{\omega}}) \\
\dot{\mathbf{b}}_f &= \mathbf{w}_{bf} \\
\dot{\mathbf{b}}_\omega &= \mathbf{w}_{bw} \\
\dot{\mathbf{c}} &= \mathbf{w}_c \\
\dot{\mathbf{z}} &= \mathbf{w}_z \\
\dot{\boldsymbol{\mu}}_j &= \mathbf{N}^T(\boldsymbol{\mu}_j)\hat{\boldsymbol{\omega}}_{\mathcal{V}} - \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \mathbf{N}^T(\boldsymbol{\mu}_j)\frac{\hat{\boldsymbol{\upsilon}}_{\mathcal{V}}}{d(\rho_j)} + \mathbf{w}_{\mu,j} \\
\dot{\rho}_j &= -\boldsymbol{\mu}_j^T \hat{\boldsymbol{\upsilon}}_{\mathcal{V}} / d'(\rho_j) + w_{\rho,j}
\end{aligned}
$$

$$
\begin{aligned}
\hat{\mathbf{f}} &= \tilde{\mathbf{f}} - \mathbf{b}_f - \mathbf{w}_f \\
\hat{\boldsymbol{\omega}} &= \tilde{\boldsymbol{\omega}} - \mathbf{b}_\omega - \mathbf{w}_\omega \\
\hat{\boldsymbol{\upsilon}}_{\mathcal{V}} &= \mathbf{z}(\boldsymbol{\upsilon} + \hat{\boldsymbol{\omega}}^{\times}\mathbf{c}) \\
\hat{\boldsymbol{\omega}}_{\mathcal{V}} &= \mathbf{z}(\hat{\boldsymbol{\omega}})
\end{aligned}
$$

RHEM - Uncertainty-aware planning

Filter Update Step - Equations (4)
$\mathbf{y}_j = \mathbf{b}_j(\boldsymbol{\pi}(\hat{\boldsymbol{\mu}}_j)) + \mathbf{n}_j$ $\mathbf{H}_j = \mathbf{A}_j(\boldsymbol{\pi}(\hat{\boldsymbol{\mu}}_j))\frac{d\boldsymbol{\pi}}{d\boldsymbol{\mu}}(\hat{\boldsymbol{\mu}}_j)$ By stacking the above terms for all visible features, standard EKF update step is directly performed to derive the new estimate of the robot belief for its state and the tracked features.

- Identify which viewpoints are expected to be visible given the next pose and the known map

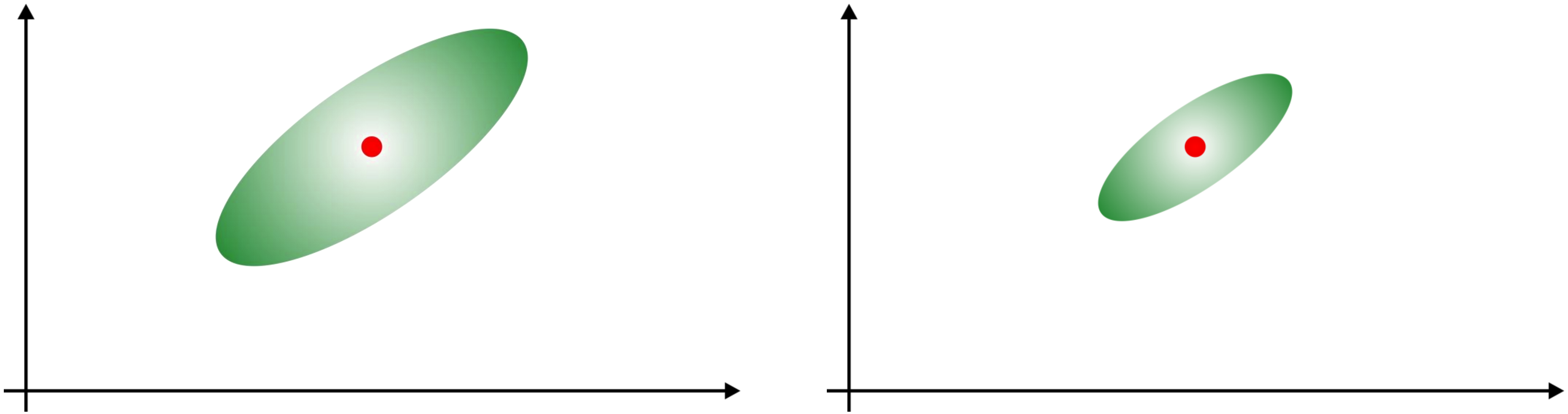
$$\begin{bmatrix} X_j \\ Y_j \\ Z_j \end{bmatrix} = \mathbf{R}_C^W \left(\mathbf{K}_I^{-1} \frac{1}{\rho_j} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \right) + \mathbf{T}_C^W$$

- Compute the propagated belief covariance matrix

$$\boldsymbol{\Sigma}_{t,l} = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t)\bar{\boldsymbol{\Sigma}}_{t,l}$$

RHEM - Uncertainty-aware planning

- Uncertainty optimization: do be able to derive which path minimizes the robot uncertainty about its pose and the tracked landmarks, a metric of how small the covariance ellipsoid is has to be defined.



RHEM - Uncertainty-aware planning

- Uncertainty optimization: do be able to derive which path minimizes the robot uncertainty about its pose and the tracked landmarks, a metric of how small the covariance ellipsoid is has to be defined.

$$D_{opt}(\sigma^M) = \exp(\log([\det(\mathbf{\Sigma}_{p,f}(\sigma^M)]^{1/(l_p+l_f)}))$$

$$\mathbf{BeliefGain}(\sigma_\alpha^M) \quad = \quad D_{opt}(\sigma_\alpha^M)$$

RHEM Algorithm

First Planning Step

- $\xi_0 \leftarrow$current vehicle configuration
- Initialize $\boldsymbol{T}^E$ with ξ_0
- $g_{best}^E \leftarrow 0$ // Set best exploration gain to zero
- $n_{best} \leftarrow n_0(\xi_0)$ // Set best best exploration node to root
- $N_T^E \leftarrow$Number of nodes in $\boldsymbol{T}^E$
- **While** $N_T^E < N_{max}^E$ or $g_{best}^E == 0$ **do**
  - Incrementally build $\boldsymbol{T}^E$ by adding $n_{new}^E(\xi_{new}^E)$
  - $N_T^E \leftarrow N_T^E + 1$
  - **if** $\mathrm{Exploration}Gain(n_{new}^E) > g_{best}^E$ **then**
    - $n_{new}^E \leftarrow n_{new}^E$
    - $g_{best}^E \leftarrow \mathrm{Exploration}Gain(n_{new}^E)$
  - **If** $N_T^E > N_{TOT}^E$ **then**
    - Terminate planning
- $\sigma_{RH}^E, n_{RH}^E, \xi_{RH} \leftarrow \boldsymbol{ExtractBestPathSegment}(n_{best})$
- $S_{\xi_{RH}} \leftarrow LocalSet(\xi_{RH})$

RHEM Algorithm

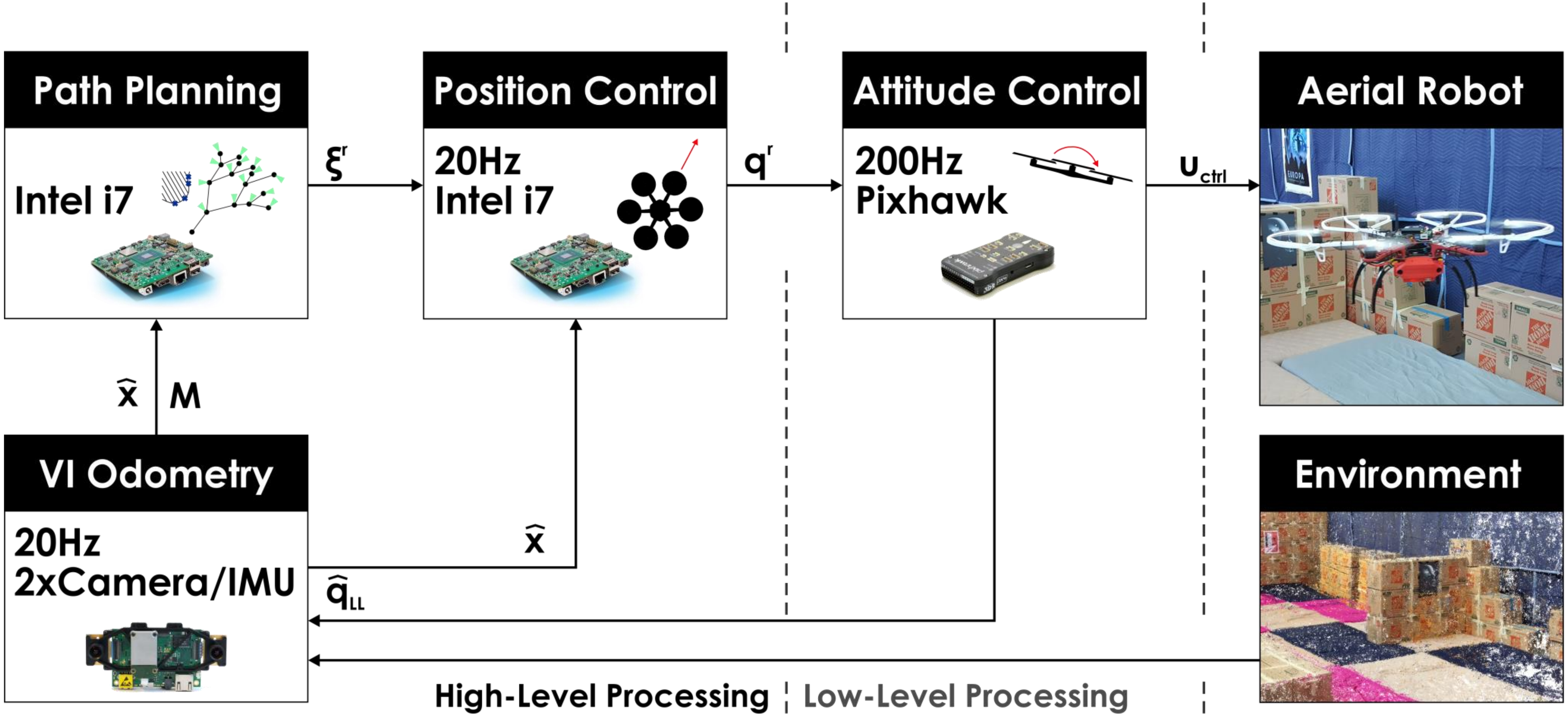
Second Planning Step

- Propagate robot belief along σ_{RH}^{E}
- $a \leftarrow 1$ // number of admissible paths
- $g_a^M \leftarrow BeliefGain(\sigma_{RH}^{E})$
- $g_{best}^M \leftarrow g_a^M$ // straight path belief gain
- $\sigma_{best}^M \leftarrow \sigma_{RH}^M$
- **While** $N_T^M < N_{max}^M$ or $V(T^M) \cap = \emptyset S_{\xi_{RH}}$ **do**
  - Incrementally build $\boldsymbol{T}^M$ by adding $n_{new}^M(\xi_{new})$
  - Propagate robot belief from current to planned vertex
  - **If** $\xi_{new} \in S_{\xi_{RH}}$ **then**
    - Add new vertex n_{new}^M at ξ_{RH} and connect
    - $a \leftarrow a + 1$
    - $\sigma_a^M \leftarrow ExtractBranch(n_{new}^M)$
    - $g_a^M \leftarrow BeliefGain(\sigma_a^M)$
    - **If** $g_a^M < g_{best}^M$ **then**
      - $\sigma^M \leftarrow \sigma_a^M$
      - $g_{best}^M \leftarrow g_a^M$
- Return σ^M

RHEM Complexity

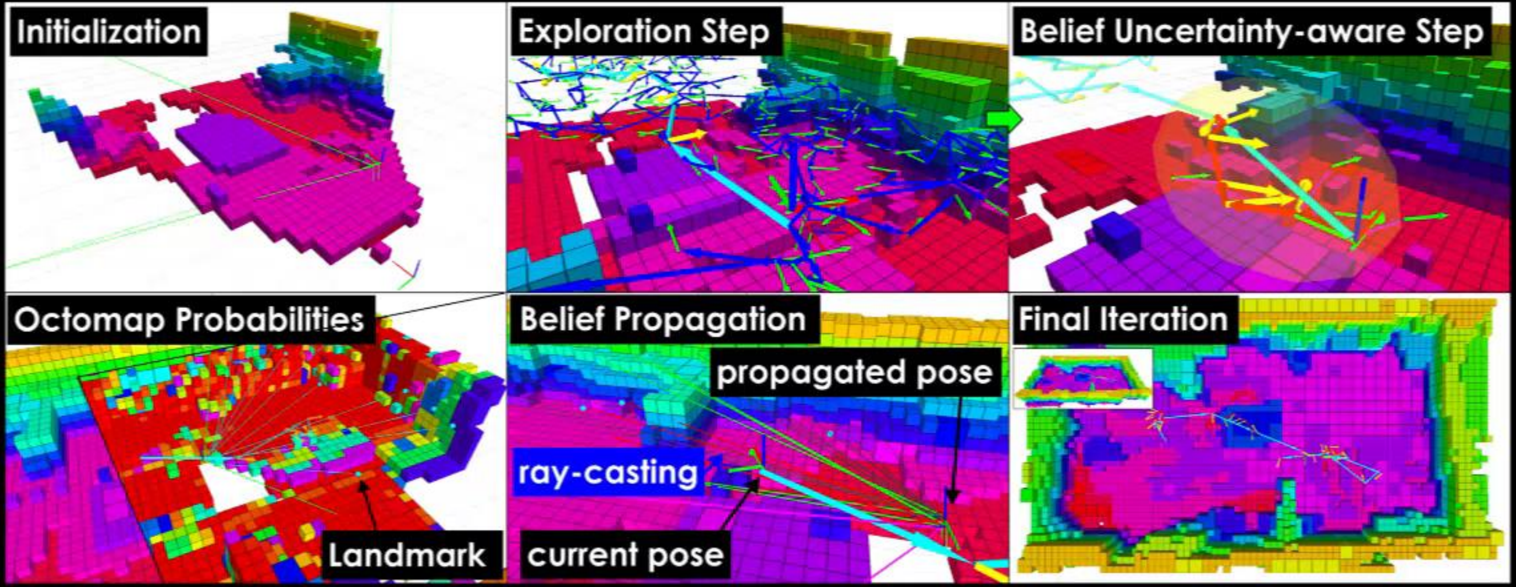
RRT construction	Collision checking
$\mathcal{O}(N_{\mathbb{T}}^{S}\log(N_{\mathbb{T}}^{S})),\ S \rightarrow E, M$	$\mathcal{O}(N_{\mathbb{T}}^{S}/r^3\log(V^E/r^3)),\ S \rightarrow E, M$
1^{st} planning level gain computation	
$\mathcal{O}(N_{\mathbb{T}}^{E}(d_{\max}^{\text{planner}}/r)^4\log(V^E/r^3))$	
2^{nd} planning level gain computation	
$\mathcal{O}(N_{\mathbb{T}}^{M}(d_{\max}^{\text{sensor}}/r)^4\log(V^E/r^3)l_f + n_M(l_s^{2.4} + l_f^2) + n_M(l_p + l_f))$	

RHEM Evaluation (Experimental)

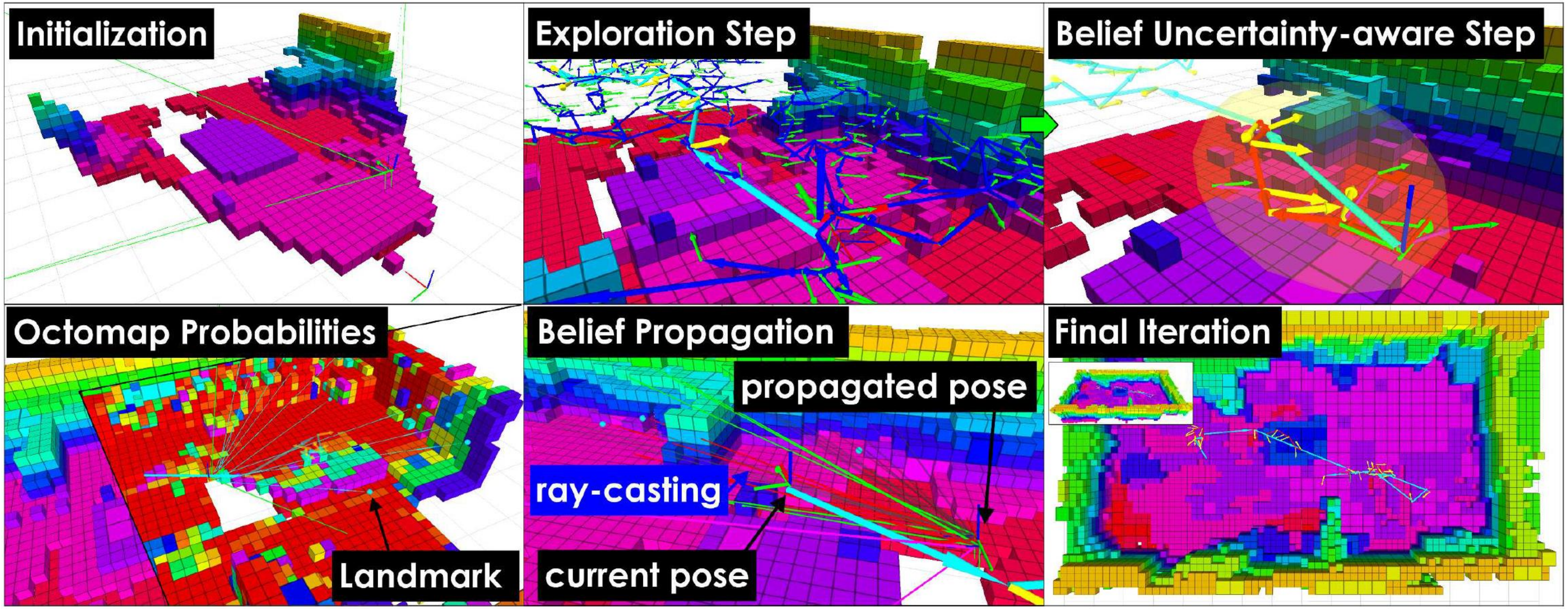


Uncertainty-aware Receding Horizon Exploration and Mapping using Aerial Robots

Christos Papachristos, Shehryar Khattak, Kostas Alexis



RHEM Evaluation (Experimental)



RHEM Evaluation (Experimental)

Real-life is 3D, Complex, Possibly unknown

Known Model to Compute Global Inspection Path

Unknown Model – execute Autonomous Exploration

The inspection path planning problem

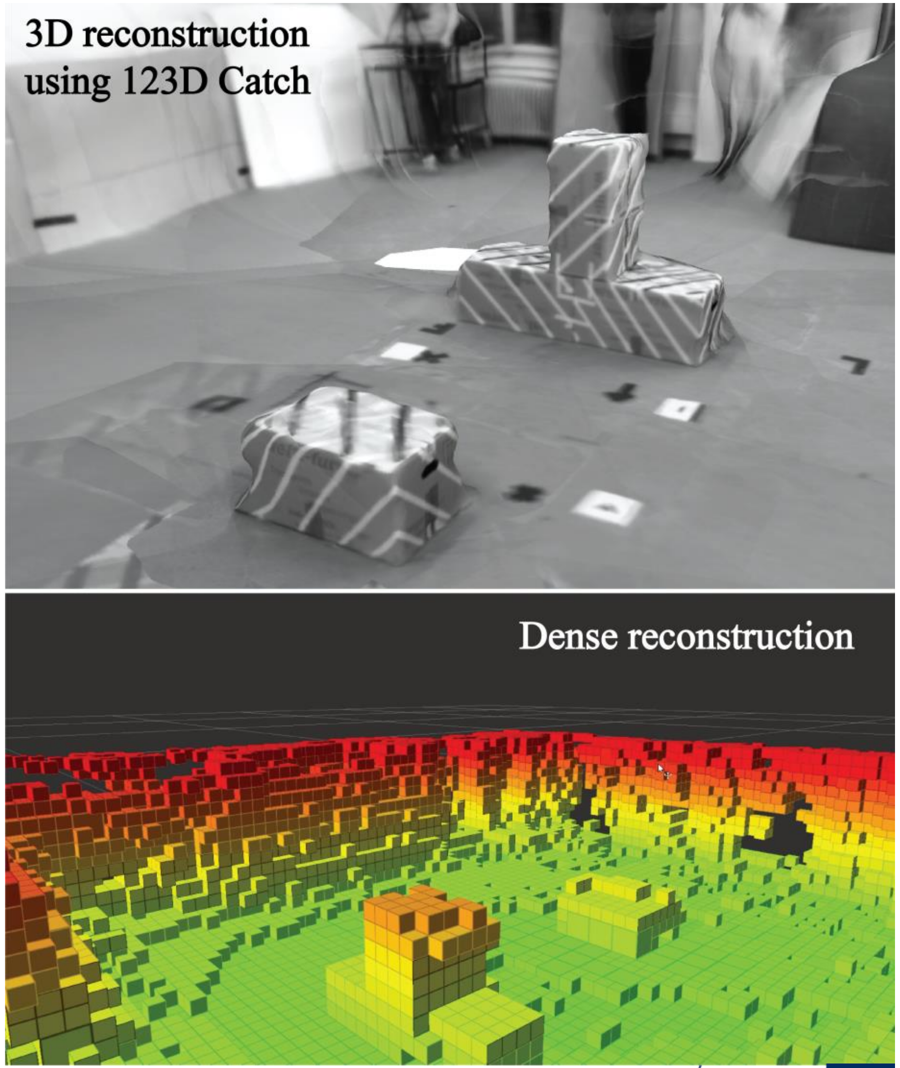
- Consider a dynamical control system defined by an ODE of the form:

$$\frac{dx}{dt} = f(x,u), x(0) = x_{init}$$

- Where is x the state, u is the control. As well as a sensor model of field of view $FOV = [F_H, F_V]$ and maximum range d.
- Given an obstacle set X_{obs}, and a inspection manifold S_I, the objective of the motion planning problem is to find, if it exists, a path $\boldsymbol{r}$ that provides the viewpoints to the sensor such that the whole surface of S_I is perceived, the vehicle dynamics are respected and the cost of the path (distance, time, etc) is minimized.

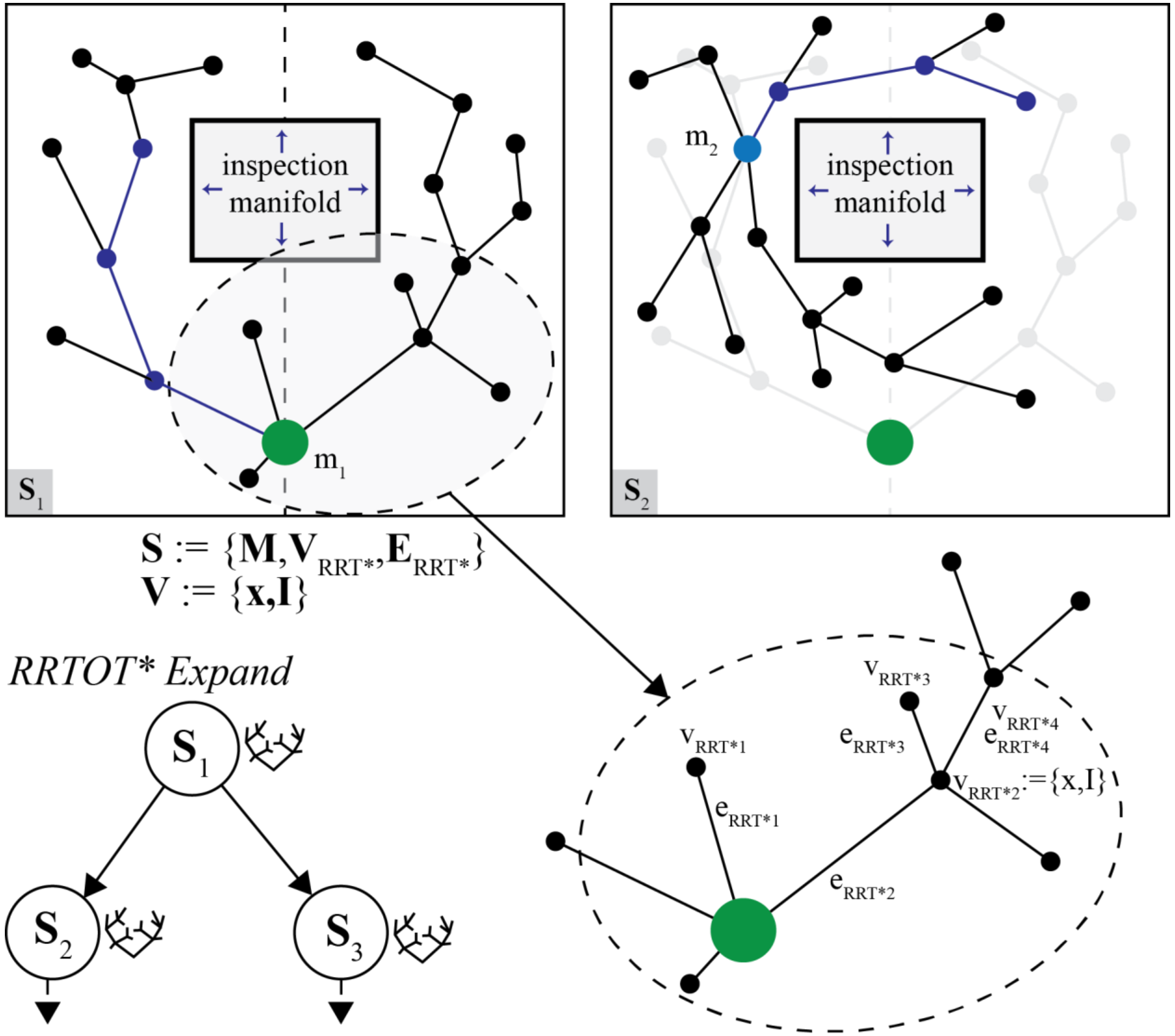
Rapidly-exploring Random Tree-Of-Trees (RRTOT)

- **Problem:** given a representation of the structure find the optimal coverage path.
- **Challenges:** can we find the optimal path? Can we converge asymptotically to that solution?
- **Goal:** Provide an algorithm that can incrementally derive the optimal solution and be able to provide admissible paths "anytime".



RRTOT: Functional Principle

Overcome the limitations of motion planners designed for navigation problems.

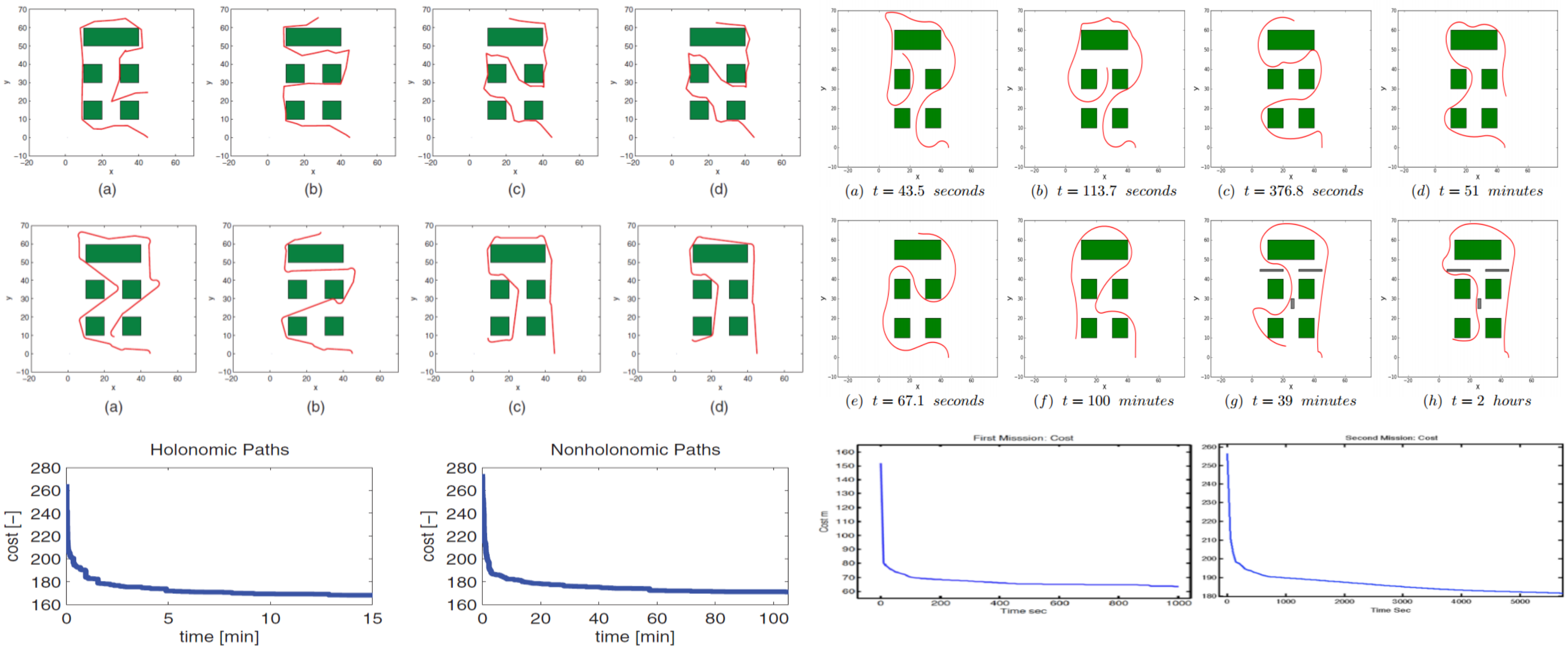


Vary the solution topology – be able to find the optimal solution. X`

Overcome the limitations of SIP but in a computationally very expensive way.

RRTOT: Functional Principle

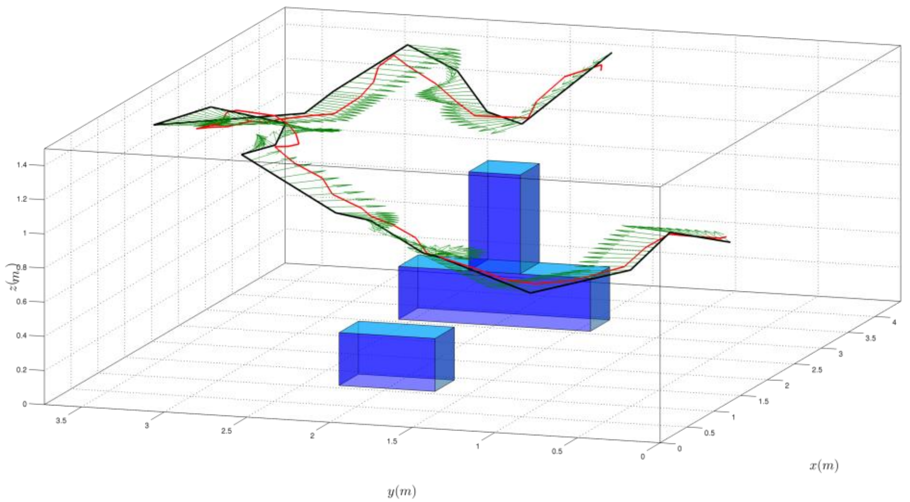
- **Comparison with the state-of-the-art:** RRTOT seems to be able to provide solutions faster.



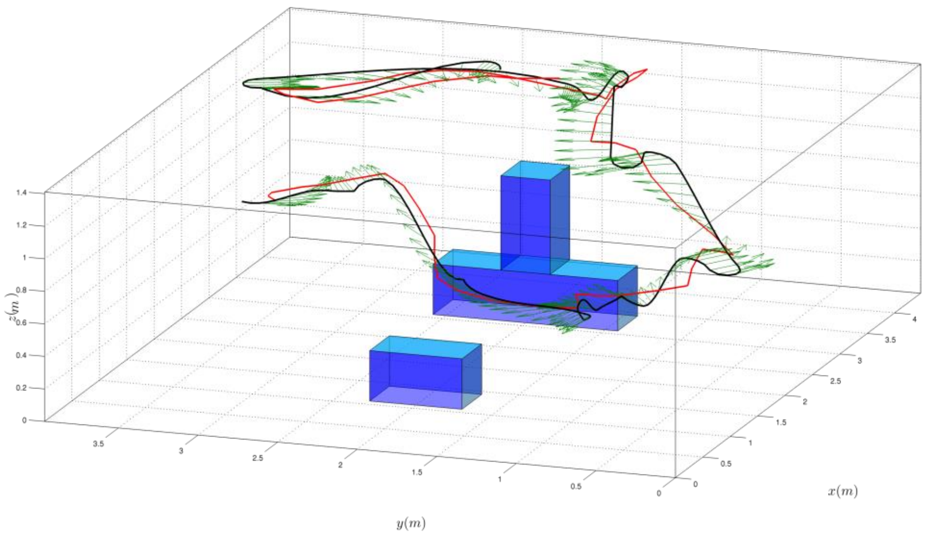
- **Comparison against:** G Papadopoulos, H Kurniawati, N Patrikalakis, "Asymptotically optimal path planning and surface reconstruction for inspection", IEEE International Conference on Robotics and Automation (ICRA) 2013.

RRTOT: Indicative Solutions

- **Holonomic**

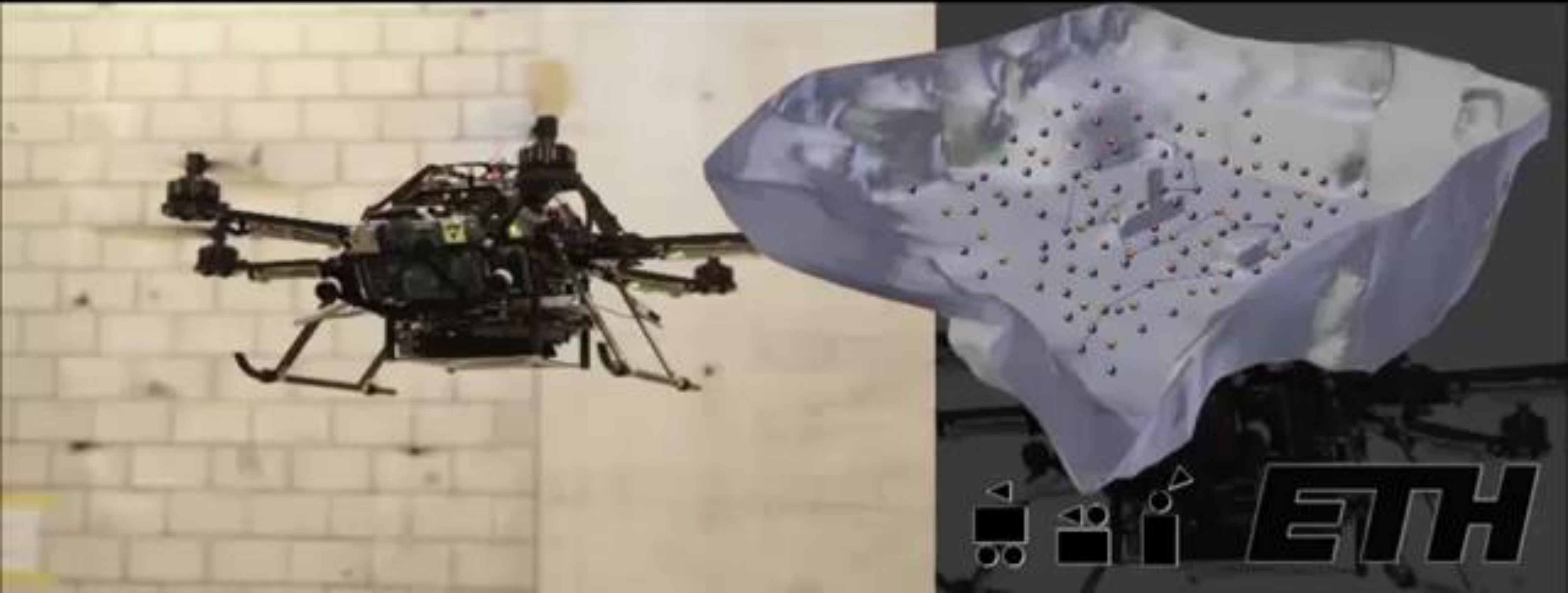


- **Nonholonomic**



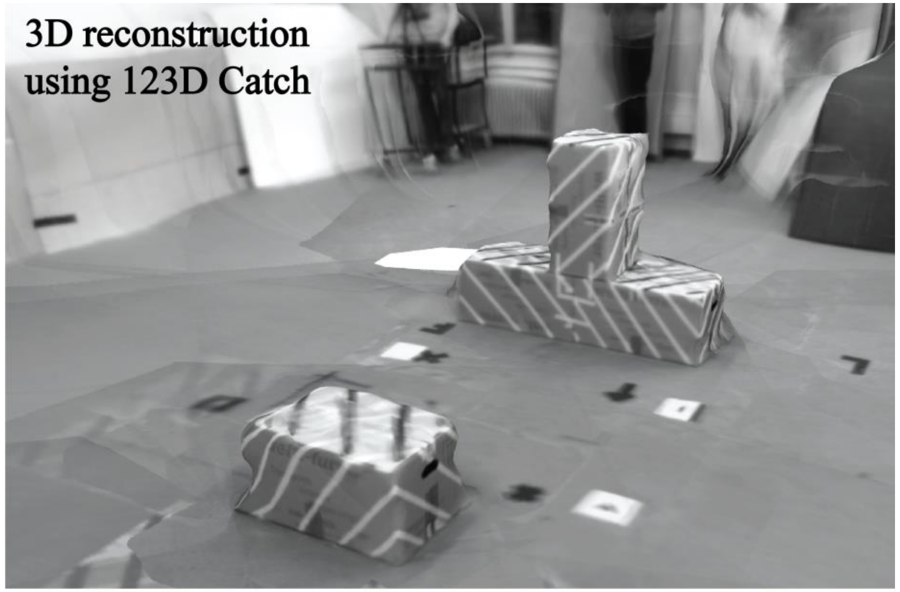
An Incremental Sampling-based approach to Inspection Planning: the Rapidly-exploring Random Tree Of Trees

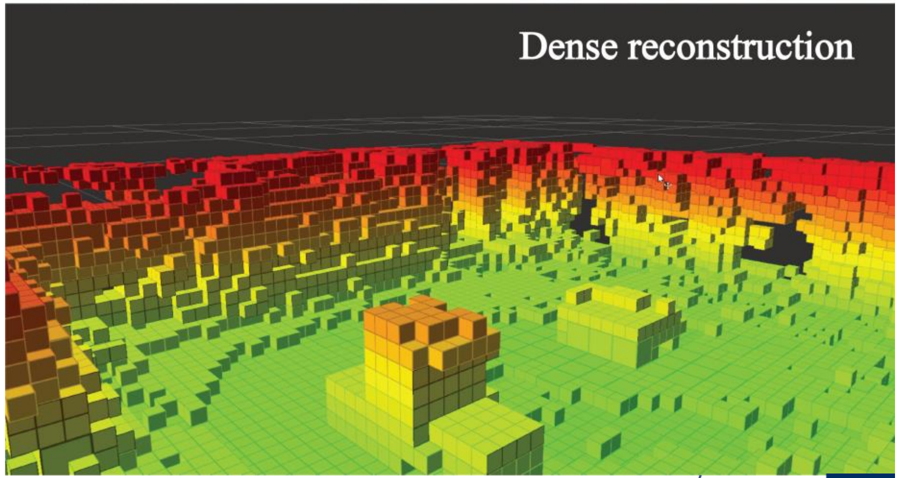
Andreas Bircher, Kostas Alexis, Ulrich Schwesinger, Sammy Omari, Michael Burri and Roland Siegwart



Benefits and Disadvantages

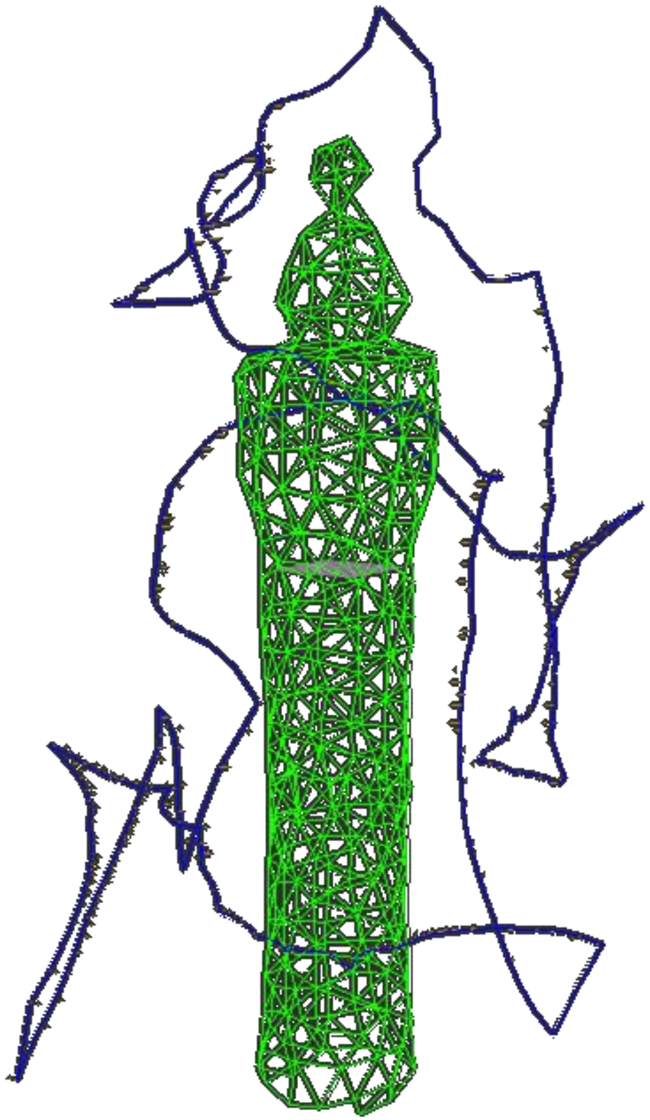
- **Quality of the Solution:** Proven to provide asymptotically optimal solution.
- **Complexity:** Practically intractable for large scale problems
- **Purpose:** More of a "theoretical tool" to compare other algorithms.





Alternative Solution

- **Can we find a “good enough” solution but compute very fast?**

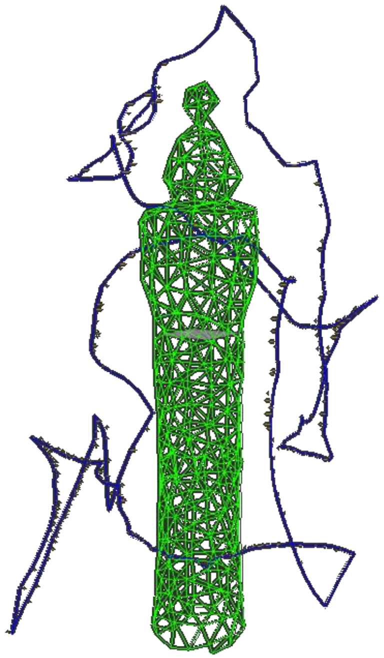


Basic Concepts of the Inspection Planner

- **Main classes of existing 3D methods:**
  - Separated Approach (AGP + TSP or Control)
    - Prone to be suboptimal
    - In specific cases lead to infeasible paths (nonholonomic vehicles)
  - First attempts for optimal solutions via a unified cycle
    - In specific cases can lead to the optimal solution
    - Very high CPU and Memory Requirements & Time

- **Structural Inspection Planner (SIP):**
  - Driven by the idea that *with a continuously sensing sensor, the number of viewpoints is not necessarily important but **mostly their configuration in space***.
  - Not a minimal set of viewpoints but a set of full coverage viewpoints positioned such that the overall path gets minimized.
  - 2-step paradigm with viewpoint alternation
  - Guaranteed feasible paths for both holonomic and nonholonomic vehicles

Structural Inspection Planner (SIP)



First solution

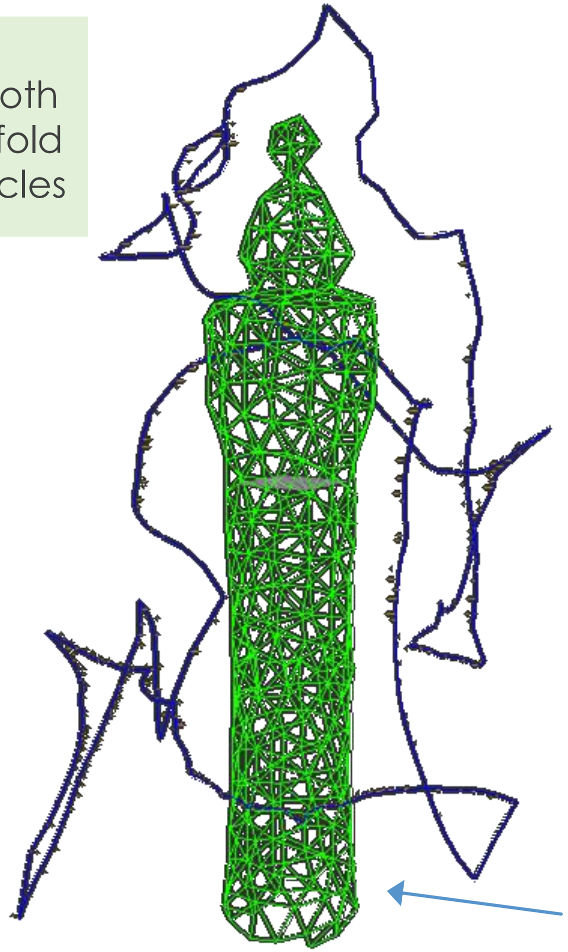
- Load the mesh model
- k = 0
- Sample Initial Viewpoint Configurations (Viewpoint Sampler)
- Find a Collision-free path for all possible viewpoint combinations (BVS, RRT*)
- Populate the Cost Matrix and Solve the Traveling Salesman Problem (LKH)
- **while** running
  - Re-sample Viewpoint Configurations (Viewpoint Sampler)
  - Re-compute the Collision-free paths (BVS, RRT*)
  - Re-populate the Cost Matrix and solve the new Traveling Salesman Problem to update the current best inspection tour (LKH)
  - k = k + 1
- **end while**
- **Return *BestTour, CostBestTour***

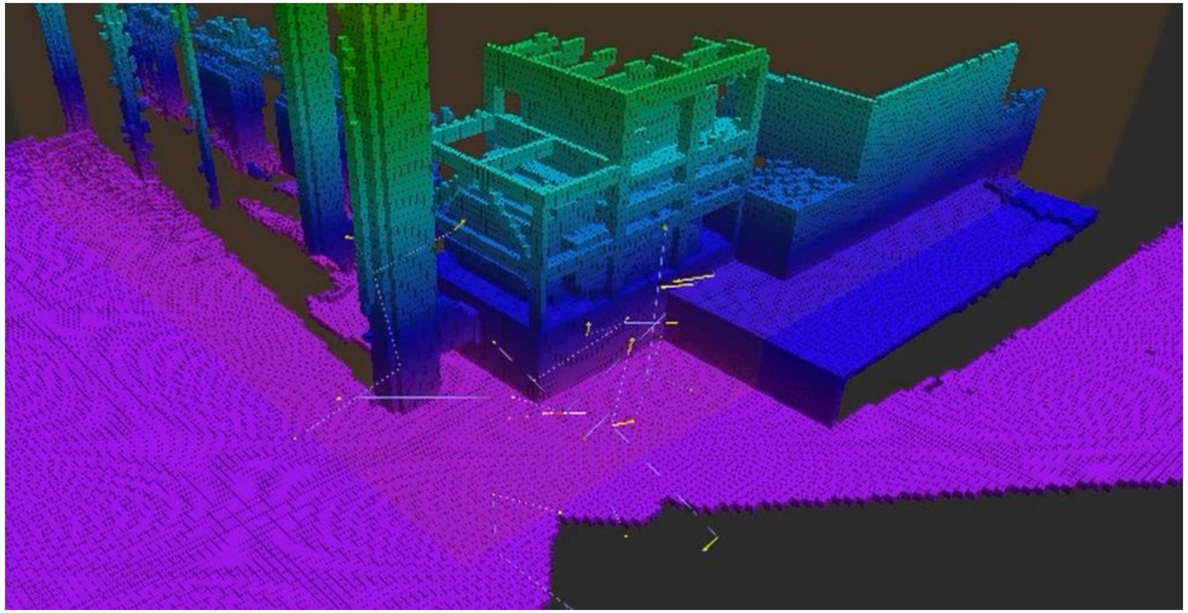
Available Time

Optimized solutions

SIP: Supported World Representations

Same type of representation for both the inspection manifold as well as any obstacles





Octomap [possibly enlarged voxels]

Not currently open-sourced

Meshes [possibly downsampled]

Supported in the open-sourced SIP

Sampling-based and Collision-checking implemented

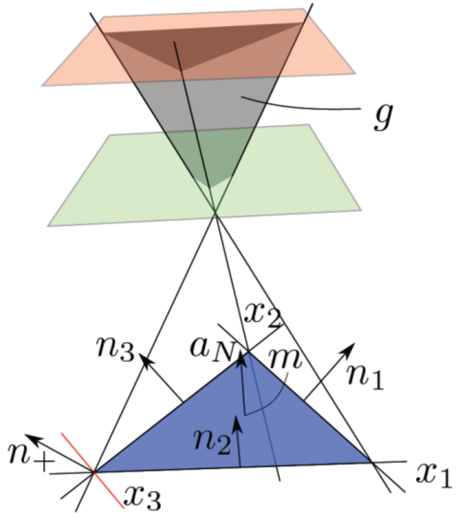
SIP: Viewpoint Sampler

- **Optimize Viewpoint Configurations**
  - Admissible viewpoints are optimized for distance to the neighboring viewpoints
  - To guarantee admissible viewpoints, the position solution $g = [x,y,z]$ is constrained to allow finding an orientation for which the triangular face is visible:

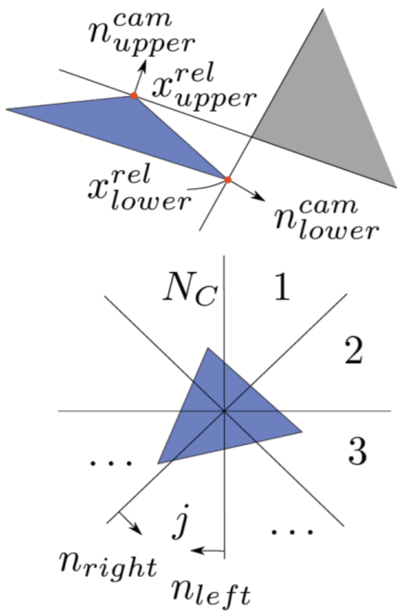
$$\begin{bmatrix} (g - x_i)^T n_i \\ (g - x_1)^T a_N \\ -(g - x_1)^T a_N \end{bmatrix} \succeq \begin{bmatrix} 0 \\ d_{min} \\ -d_{max} \end{bmatrix}, \; i = \{1,2,3\}$$

  - Account for limited **F**ield **o**f **V**iew by imposing a revoluted 2D-cone constraint. This is a nonconvex problem which is then convexified by dividing the problem into N_c equal convex pieces.

$$\begin{bmatrix} (g - x^{rel}_{lower})^T n^{cam}_{lower} \\ (g - x^{rel}_{upper})^T n^{cam}_{upper} \\ (g - m)^T n_{right} \\ (g - m)^T n_{left} \end{bmatrix} \succeq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



Incidence angle constraints on a triangular surface



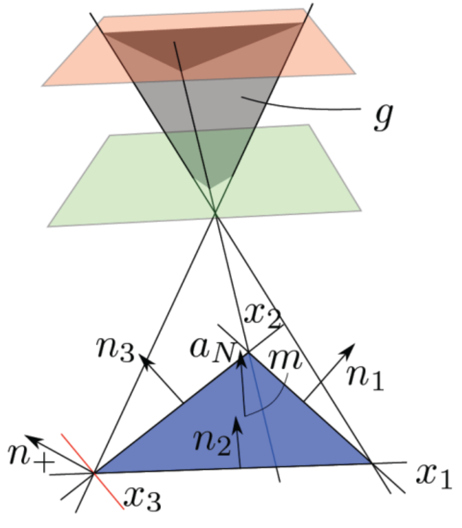
Camera constraints and convexification

SIP: Viewpoint Sampler

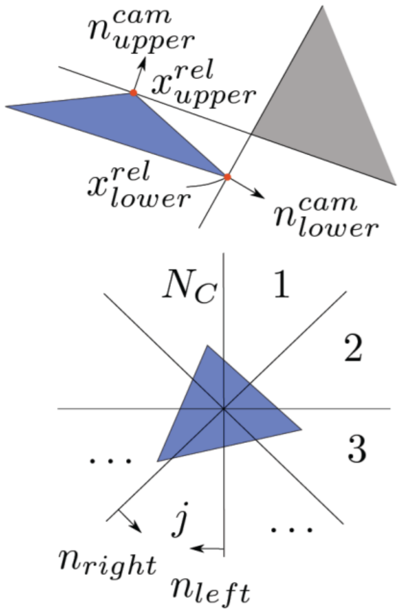
- **Sample 1 Viewpoint/Triangular face**
  - Minimize the sum of squared distances to the preceding viewpoint $\boldsymbol{g_p}^{k-1}$, the subsequent viewpoint $\boldsymbol{g_s}^{k-1}$ and the current viewpoint in the old tour $\boldsymbol{g}^{k-1}$.

$$
\begin{aligned}
\min_{g^k} \quad & (g^k - g_p^{k-1})^T (g^k - g_p^{k-1}) + \\
& + (g^k - g_s^{k-1})^T (g^k - g_s^{k-1}) + (g^k - g^{k-1})^T (g^k - g^{k-1}) \\
\text{s.t.} \quad & \begin{bmatrix} n_1^T \\ n_2^T \\ n_3^T \\ a_N^T \\ -a_N^T \\ n_{lower}^{cam\ T} \\ n_{upper}^{cam\ T} \\ n_{right}^T \\ n_{left}^T \end{bmatrix} g^k \succeq \begin{bmatrix} n_1^T x_1 \\ n_2^T x_2 \\ n_3^T x_3 \\ a_N^T x_1 + d_{min} \\ -a_N^T x_1 - d_{max} \\ n_{lower}^{cam\ T} x_{lower}^{rel} \\ n_{upper}^{cam\ T} x_{upper}^{rel} \\ n_{right}^T m \\ n_{left}^T m \end{bmatrix}
\end{aligned}
$$

QP + Linear Constraints



Incidence angle constraints on a triangular surface



Camera constraints and convexification

  - The heading is determined according to:

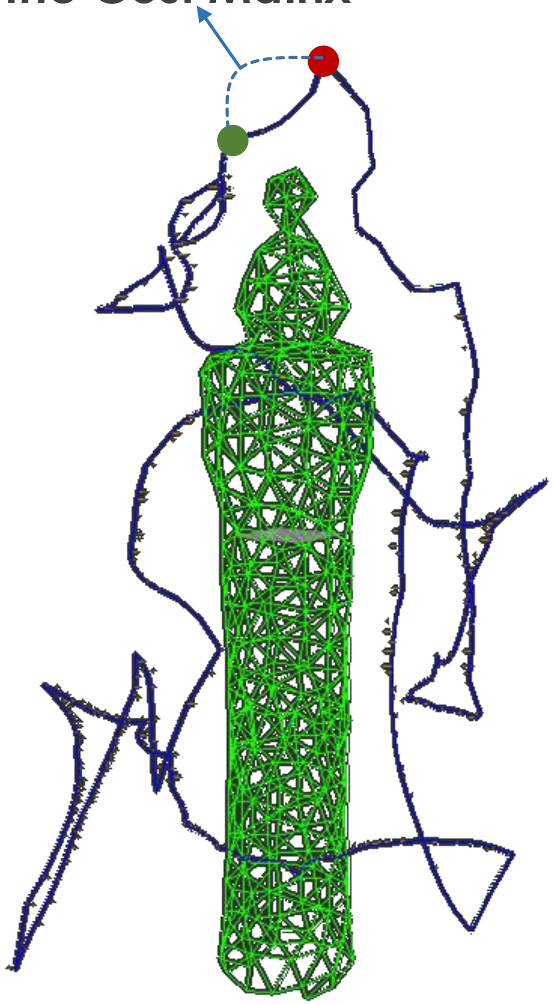
$$
\min_{\psi^k} = \left(\psi_p^{k-1} - \psi^k\right)^2 / d_p + \left(\psi_s^{k-1} - \psi^k\right)^2 / d_s, \qquad \text{s.t.} \qquad \mathbf{Visible}(g^k, \psi^k)
$$

While ensuring visibility, try to align the vehicle heading over a path

SIP: Point-to-Point Paths

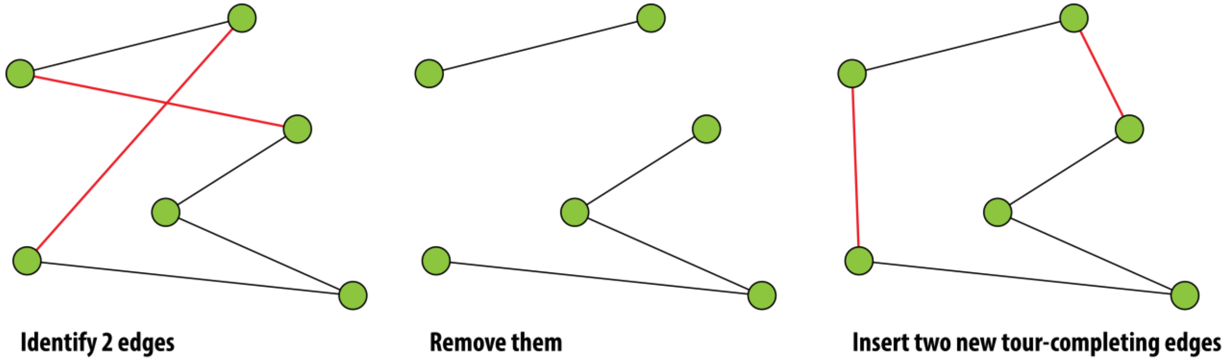
- State-Space Sampling – *extension to Control-Space sampling possible*
- Employ **B**oundary **V**alue **S**olvers for
  - Holonomic (with Yaw-rate constraints) or
  - Nonholonomic Aerial Robots *(fixed-wing UAVs – 2.5D approx., Dubins Airplane approx.)*
- Derive Collision-free paths that connect all viewpoint configurations by invoking ***RRT****
- Assemble the Traveling Salesman Problem Cost Matrix using the path execution times t_{ex}

- **Compute RRT* Path**
- **Extract the t_{ex} of the RRT* Path**
- **Populate the Cost Matrix**

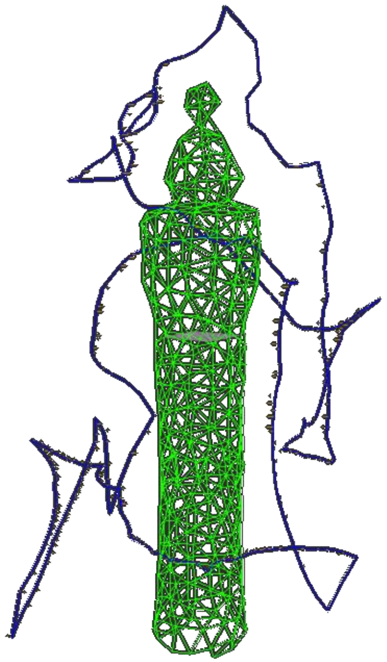


SIP: TSP Solution

- Solve the (possibly asymmetric) TSP problem using the Lin-Kernighan-Helsgaun heuristic
- ***Extract the Optimized Inspection Tour***

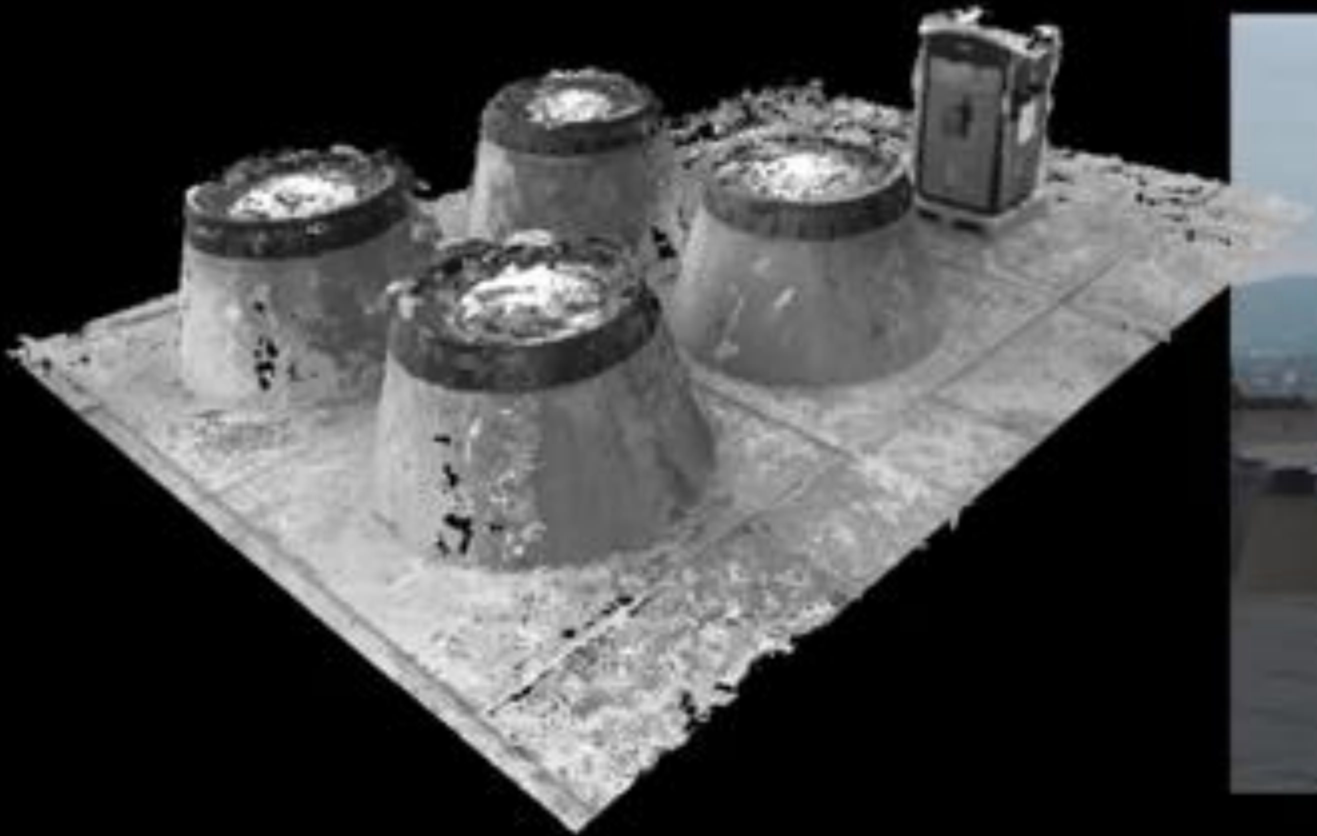


$O(N^{2.2})$, N the number of viewpoints



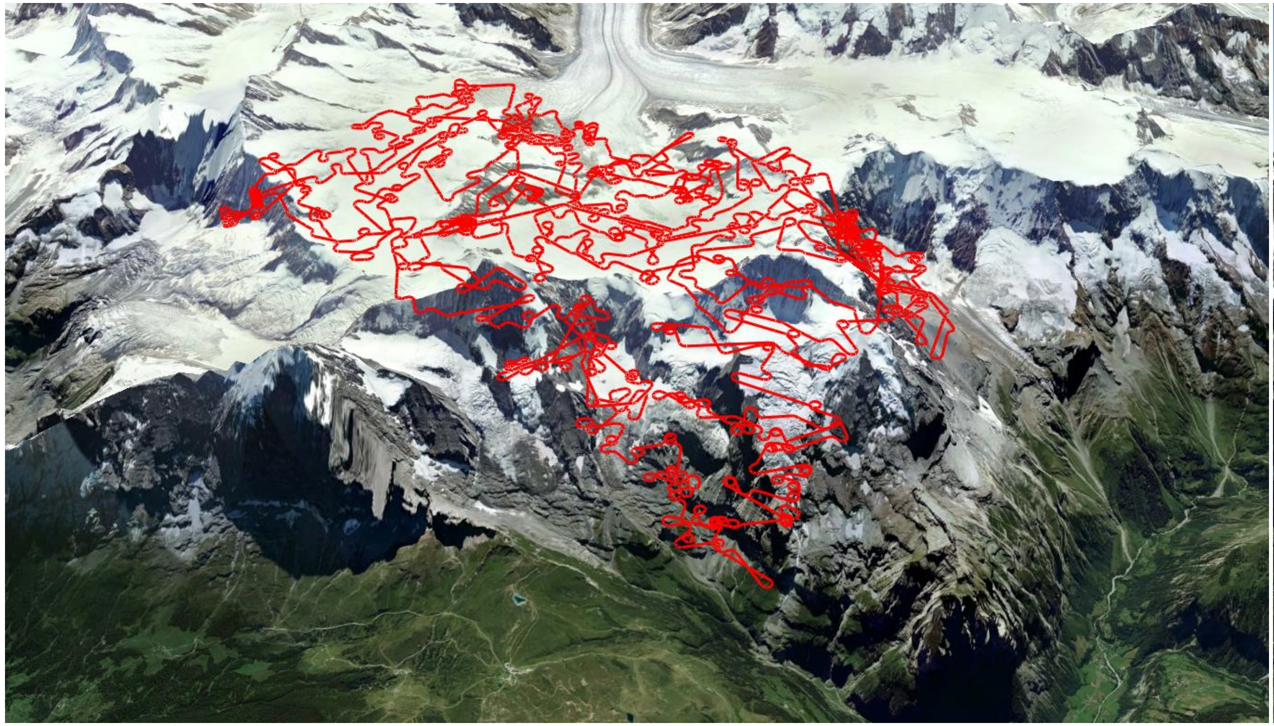
Three-dimensional Coverage Path Planning via Viewpoint Resampling and Tour Optimization using Aerial Robots

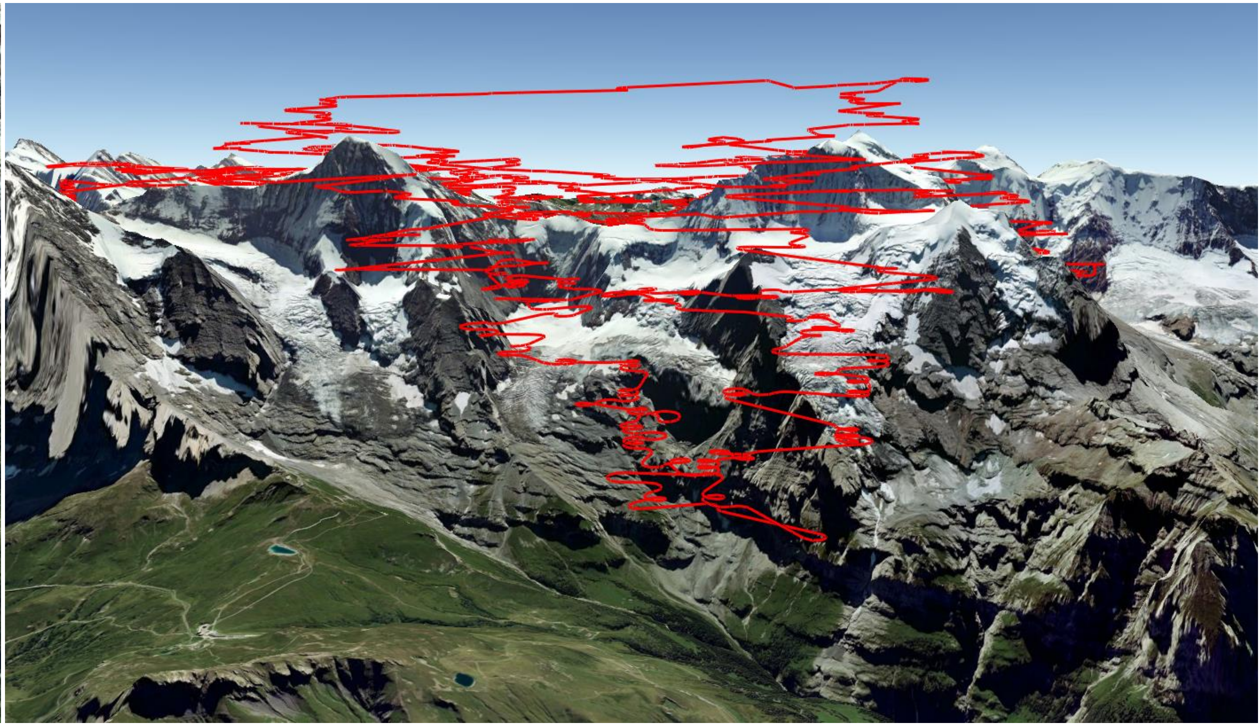
A. Bircher, K. Alexis, M. Kamel, M. Burri, P. Oettershagen, S. Omari, T. Mantel, R. Siegwart




Structural Inspection Path Planning via Iterative Viewpoint Resampling
with Application to Aerial Robotics
Andreas Bircher, Kostas Alexis, Michael Burri, Phlipp Oettershagen, Sammy Omari, Thomas Mantel and Roland Siegwart

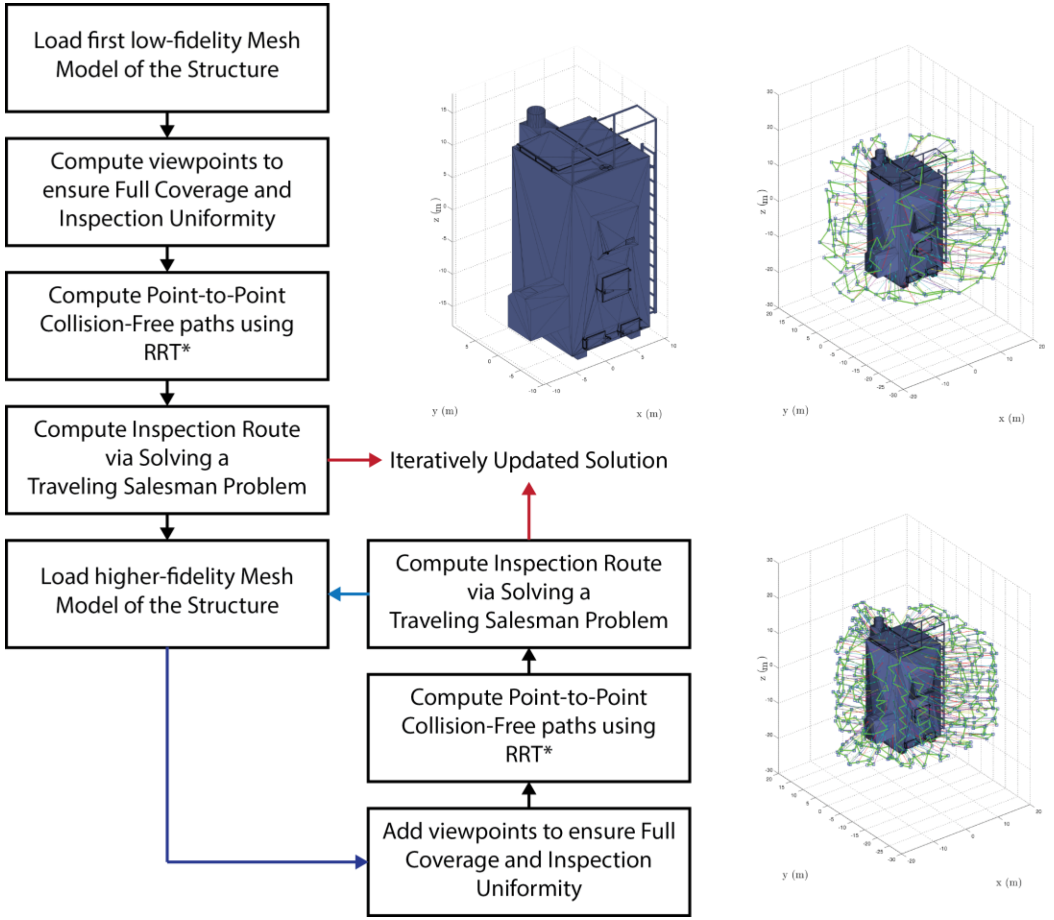
Large Scale Planning: Inspection of the JungFrau mountain (Simulation)



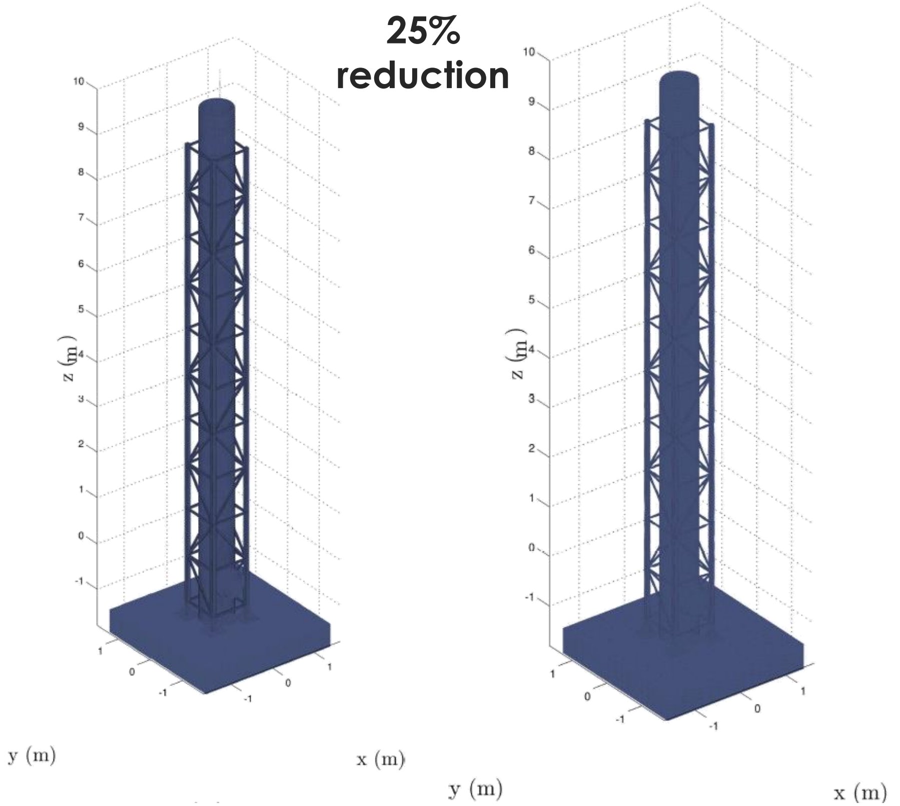


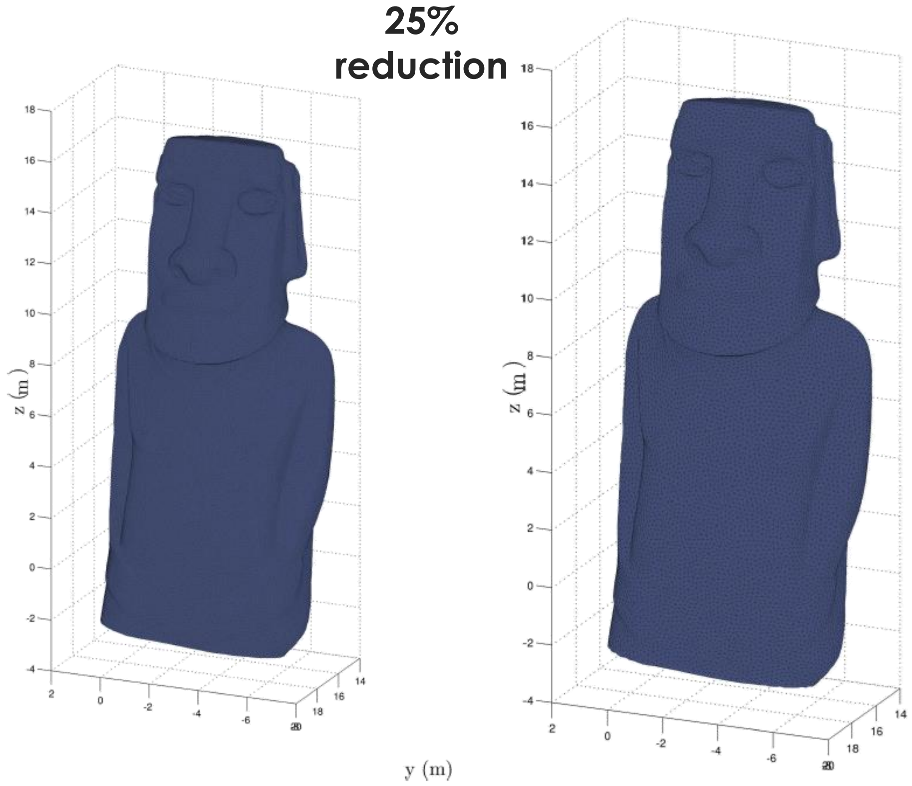
Uniform Coverage Inspection Path-Planning (UC3D)

- **Problem:** given a representation of the structure, compute a full coverage path that provides uniform focus on the details.
- **Challenge:** provide a good solution at "anytime".
- **Goal:** an efficient "anytime" inspection path planning algorithm with uniformity guarantees.
- Key for the solution: Voronoi-based remeshing techniques and a combination of viewpoint computation algorithms, collision-free planners and efficient TSP solvers.



UC3D: Remeshing techniques play a key role





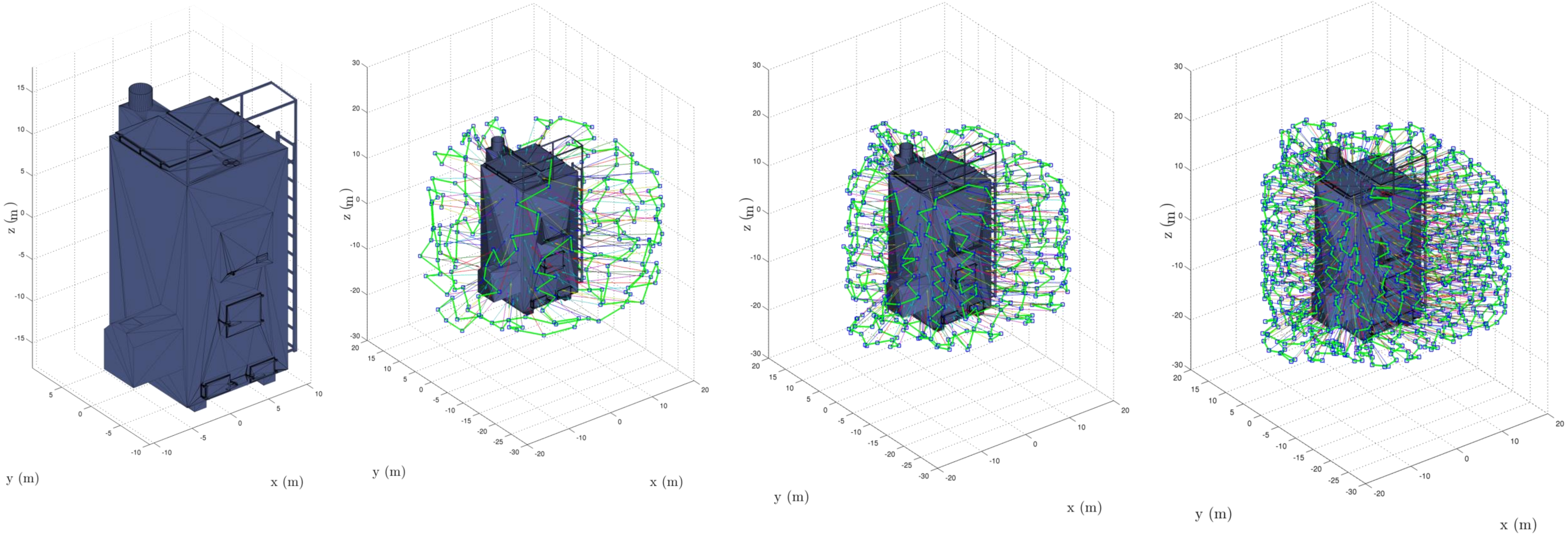
- **Voronoi-based remeshing techniques allow for uniform downsampling of the mesh with minimal structural loss**

UC3D: Iterative UC3D-IPP

$\mathcal{V}^{i-1} \leftarrow \mathcal{V}^{basic}$
$\mathcal{V}^{i} \leftarrow \mathcal{V}^{i-1}$
$\mathcal{P}_i \leftarrow \text{ExtractPolygons}(\mathcal{G}_i, \mathcal{F}_i)$
for all $\mathbf{p}_{k,i} \in \mathcal{P}_i$ **do**
  if IsCoveredUniformly$(\mathbf{p}_{k,i}, \mathcal{V}^{i-1})$ == **FALSE then**
    $\mathbf{v}_{k,i} \leftarrow \text{ComputeViewpoint}(\mathbf{p}_{k,i})$
    $\mathcal{V}^{i} \leftarrow \mathcal{V}^{i} \cup \mathbf{v}_{k,i}$
for all $\mathbf{v}_n \in \mathcal{V}^i$ **do**
  for all $\mathbf{v}_m \in \mathcal{V}^i$ **do**
    $\mathbf{C}(n,m) \leftarrow \text{ConnectionDistance}(\mathbf{v}_n, \mathbf{v}_m)$
$\mathbf{r}_i \leftarrow \text{ComputeViewpointsRoute}(\mathbf{C}(n,m))$
return $\mathbf{r}_i$

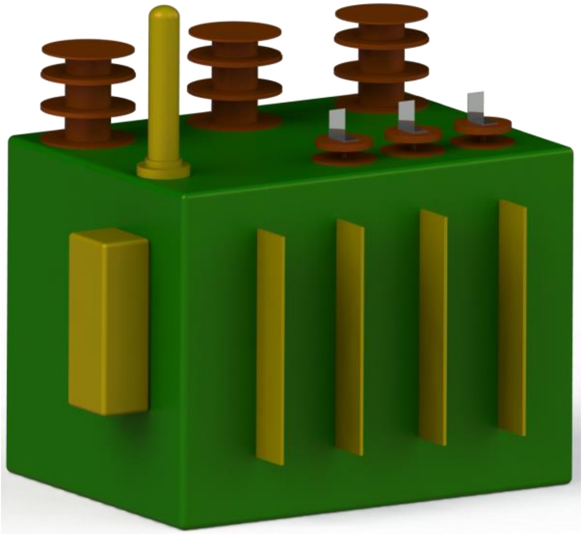
- **Difference of Iterative version:**
  - For each higher quality mesh, instead of computing a whole new set of viewpoints, only some additional are added to re-ensure uniform coverage.

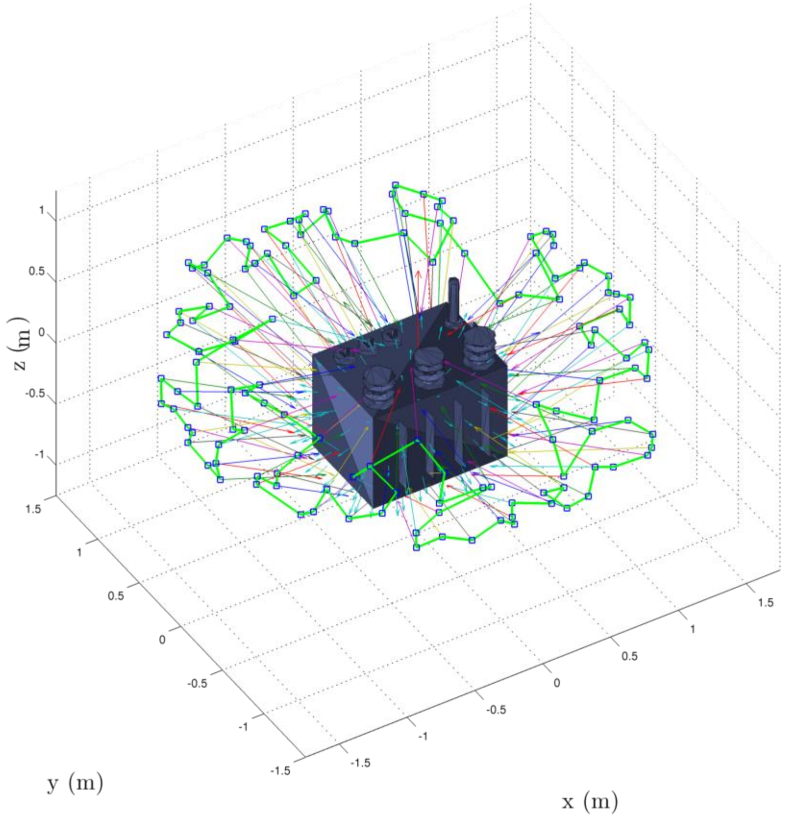
UC3D: Basic UC3D-IPP Result

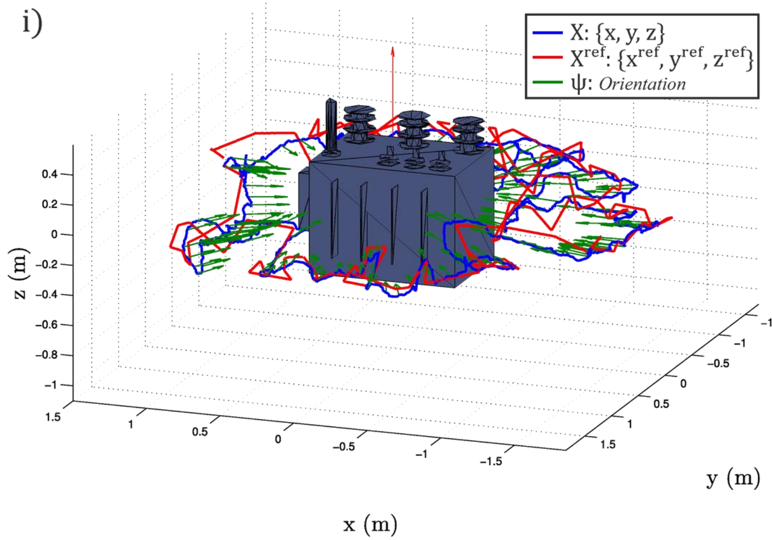


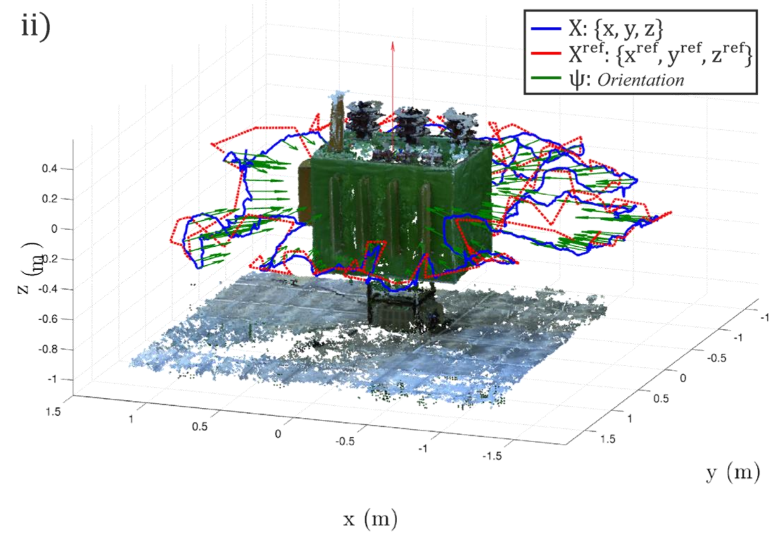
Sequential execution of the basic UC3D-IPP algorithm

UC3D: Experimental study on a Power Transforer MockUp



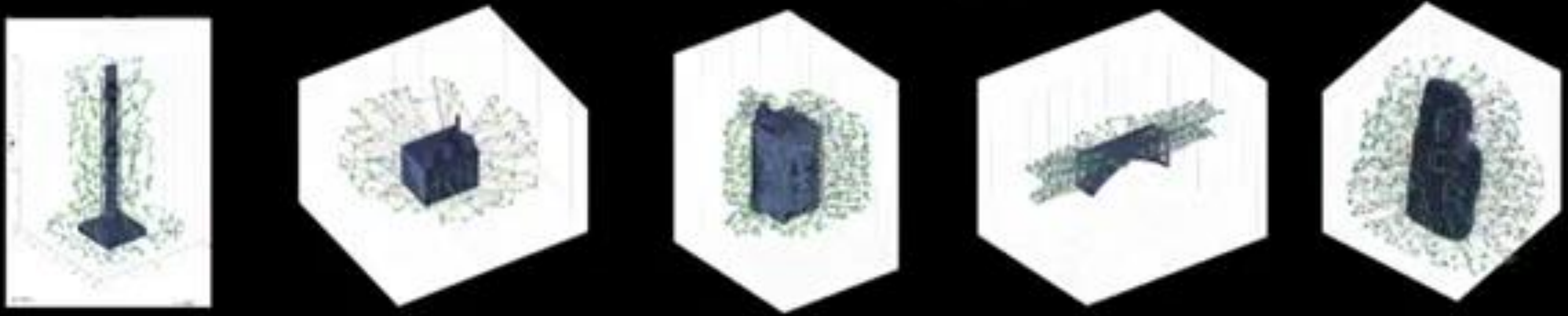




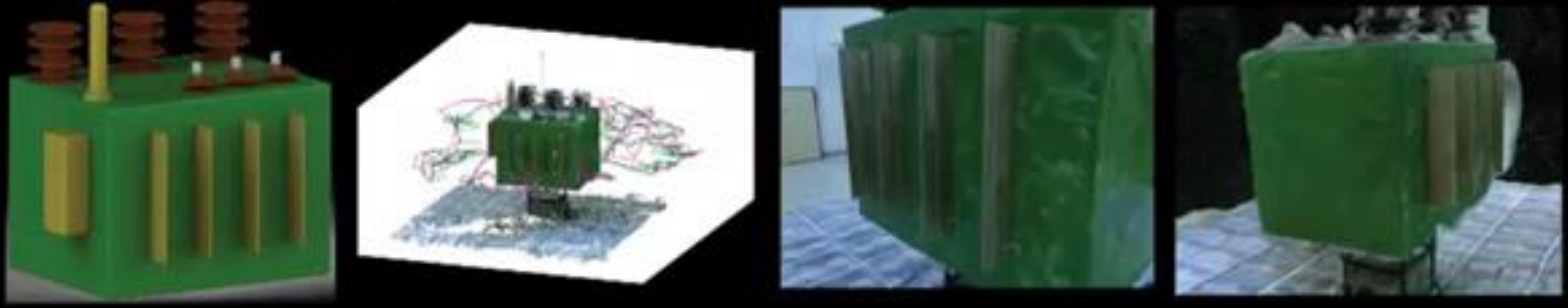


Uniform Coverage Structural Inspection Path-Planning for Micro Aerial Vehicles

K. Alexis, C. Papachristos, R. Siegwart, A. Tzes



Mesh Model   Inspection Path   Raw Camera Frames   Reconstructed Model



Code Examples and Tasks

- https://github.com/unr-arl/autonomous_mobile_robot_design_course/tree/master/ROS/path-planning/autonomous-exploration
- https://github.com/unr-arl/autonomous_mobile_robot_design_course/tree/master/ROS/path-planning/structural-inspection

Find out more

- http://www.autonomousrobotslab.com/autonomous-navigation-and-exploration.html
- A. Bircher, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel, R. Siegwart, "Structural Inspection Path Planning via Iterative Viewpoint Resampling with Application to Aerial Robotics", IEEE International Conference on Robotics & Automation, May 26-30, 2015 (ICRA 2015), Seattle, Washington, USA
- Kostas Alexis, Christos Papachristos, Roland Siegwart, Anthony Tzes, "Uniform Coverage Structural Inspection Path-Planning for Micro Aerial Vehicles", Multiconference on Systems and Control (MSC), 2015, Novotel Sydney Manly Pacific, Sydney Australia. 21-23 September, 2015
- K. Alexis, G. Darivianakis, M. Burri, and R. Siegwart, "Aerial robotic contact-based inspection: planning and control", Autonomous Robots, Springer US, DOI: 10.1007/s10514-015-9485-5, ISSN: 0929-5593, http://dx.doi.org/10.1007/s10514-015-9485-5
- A. Bircher, K. Alexis, U. Schwesinger, S. Omari, M. Burri and R. Siegwart "An Incremental Sampling–based approach to Inspection Planning: the Rapidly–exploring Random Tree Of Trees", accepted at the Robotica Journal (awaiting publication)
- A. Bircher, M. Kamel, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel, R. Siegwart, "Three-dimensional Coverage Path Planning via Viewpoint Resampling and Tour Optimization for Aerial Robots", Autonomous Robots, Springer US, DOI: 10.1007/s10514-015-9517-1, ISSN: 1573-7527
- A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, R. Siegwart, "Receding Horizon "Next-Best-View" Planner for 3D Exploration", IEEE International Conference on Robotics and Automation 2016 (ICRA 2016), Stockholm, Sweden (Accepted - to be presented)


Thank you!
Please ask your question!