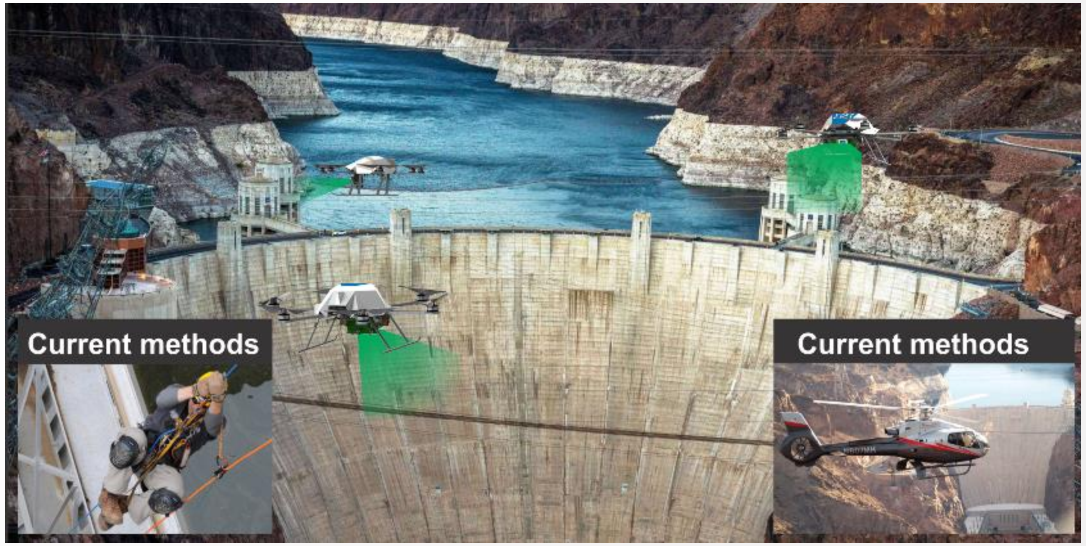
CS491/691: Introduction to Aerial Robotics

Topic: Sampling-based Inspection Path Planning

Dr. Kostas Alexis (CSE)

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.



The inspection path planning problem

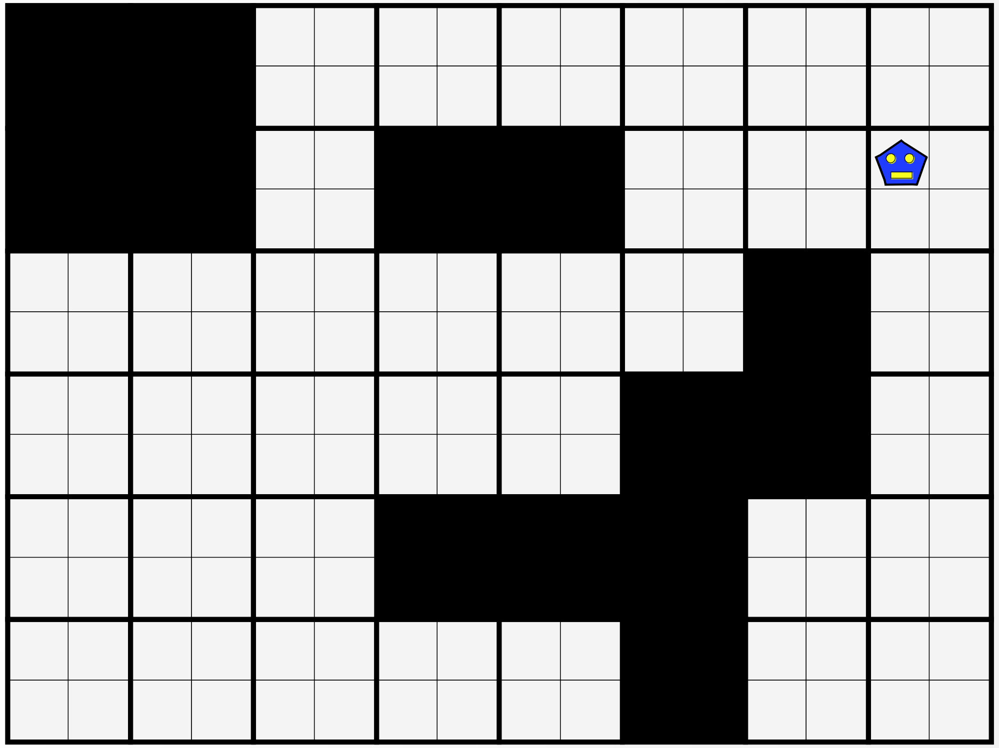
- Consider a dynamical control system defined by an ODE of the form:

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

- 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.

Old approaches to the problem

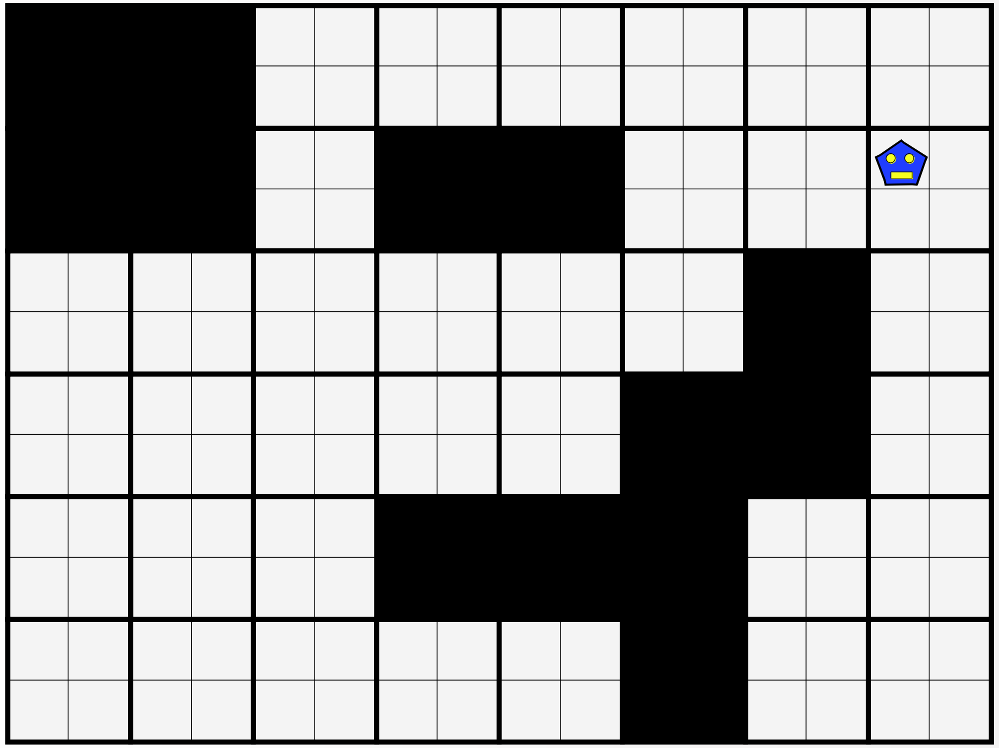
- A 2D grid of large square cells
- Some of the cells may be blocked
- Each open cell is divided to 4 small cells



We want our robot to cover all cells in the minimal possible time

Old approaches to the problem

- A 2D grid of large square cells
- Some of the cells may be blocked
- Each open cell is divided to 4 small cells



Old approaches to the problem

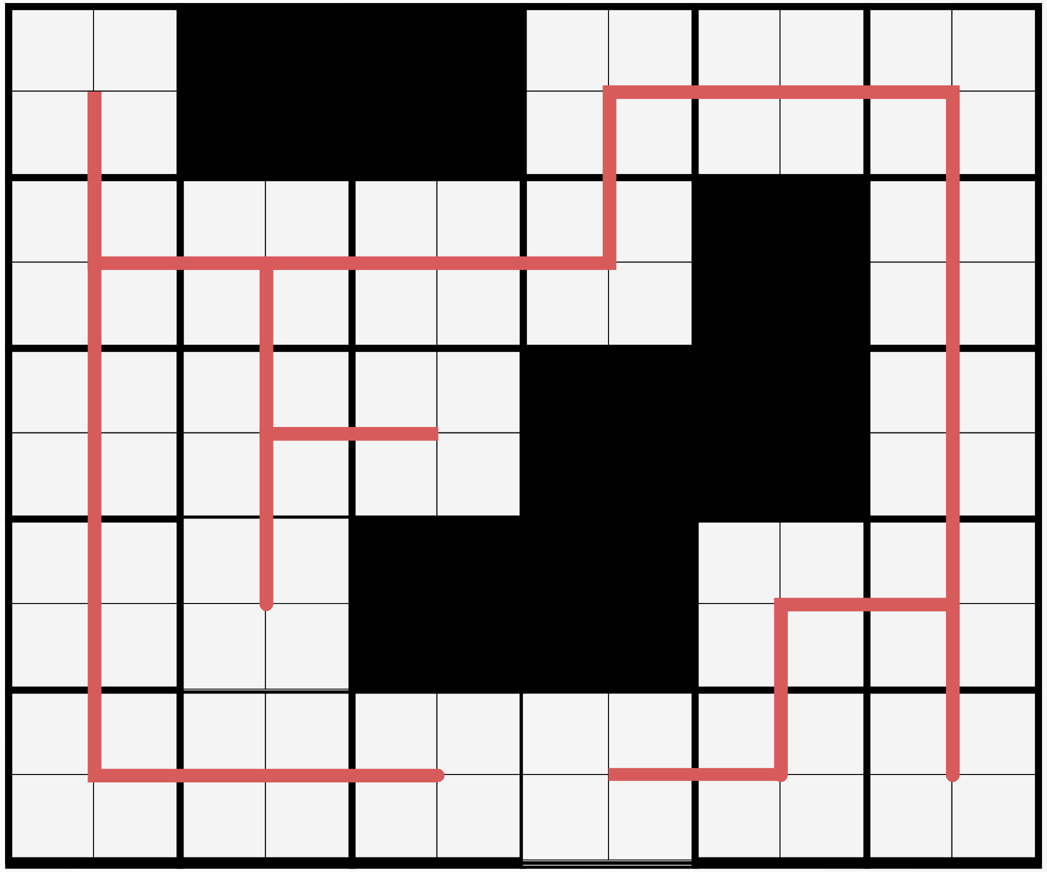
- Define a graph **G** as follows:
  - The center of each large cell is a vertex
  - There is an edge between every two adjacent cells
- Find a spanning tree **T** for the graph.
- The robot walks clockwise around the tree, stopping right before the starting point.

Spanning **T**ree **C**overage

Old approaches to the problem

The graph **G**

The tree **T**



Analysis of STC

Theorem
The STC algorithm covers every small cell that is accessible from the starting cell.

Theorem
The STC algorithm is optimal, i.e. it covers every cell at most once.

Analysis of STC

Theorem
The STC algorithm covers every small cell that is accessible from the starting cell.

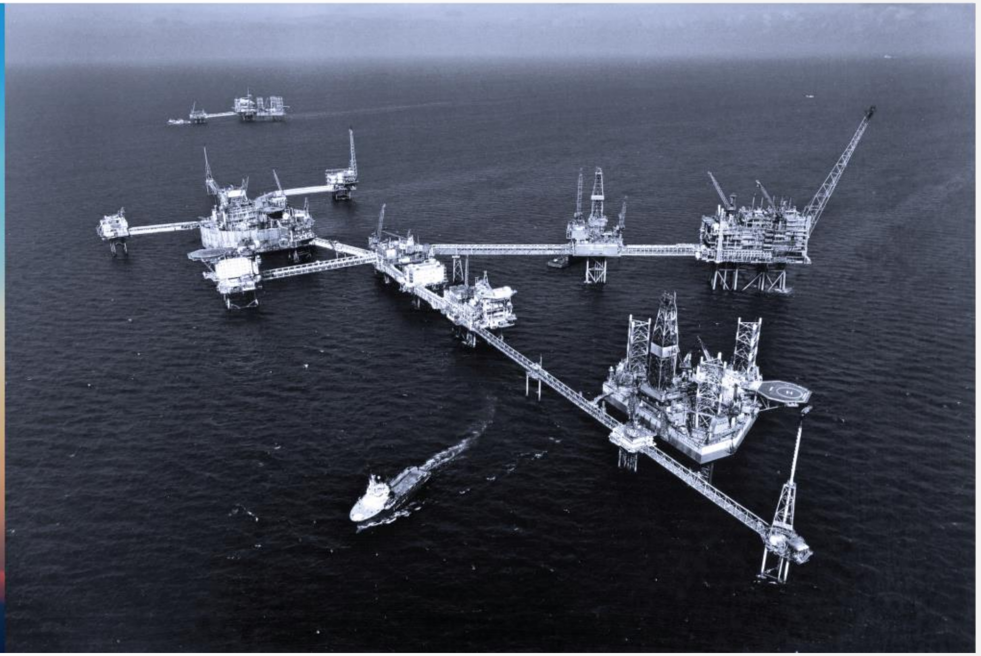
Theorem
The STC algorithm is optimal, i.e. it covers every cell at most once.

But it does not scale to three dimensions, large problems, complex sensor models, constrained dynamics

Real-life is 3D, Complex, Possibly unknown

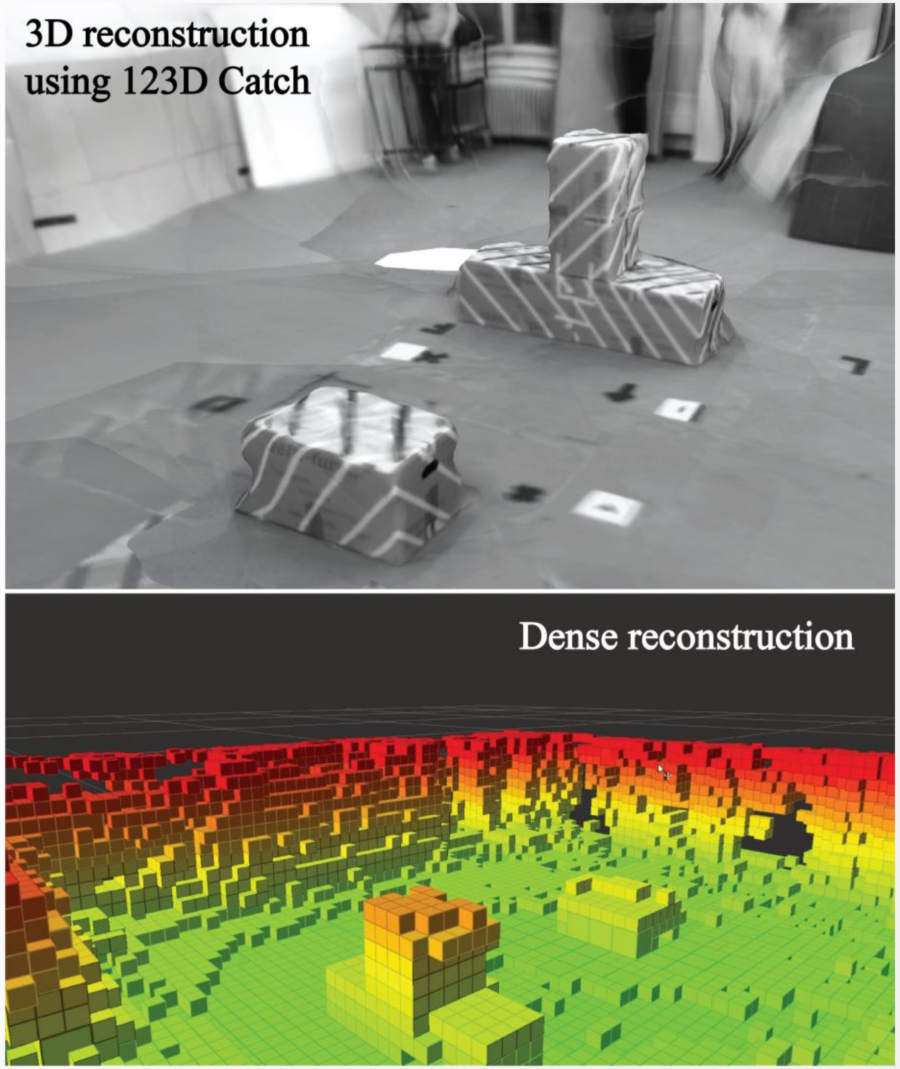
Known Model to Compute Global Inspection Path

Unknown Model – execute Autonomous Exploration



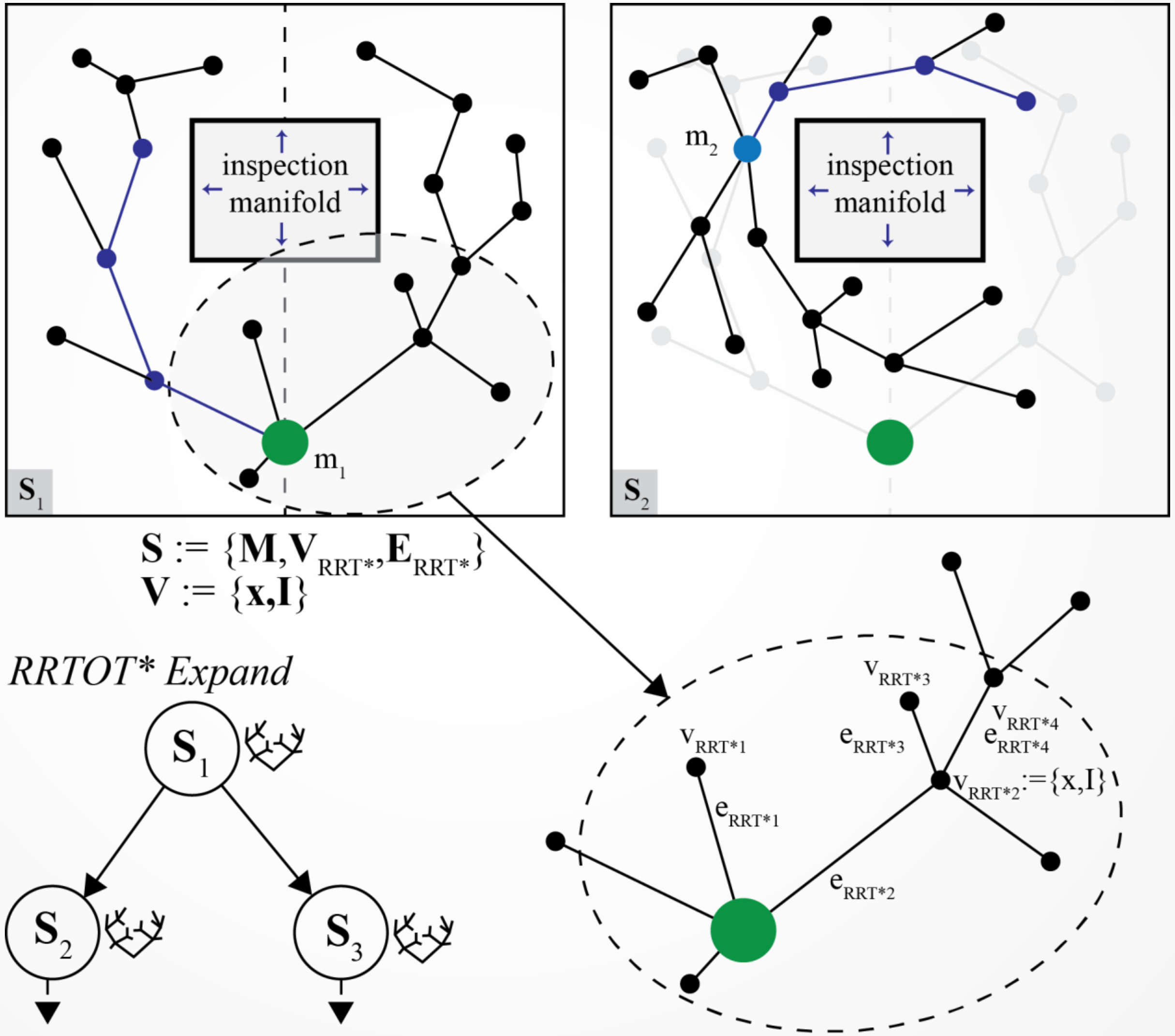
Rapidly-exploring Random Tree-Of-Trees (RRTOT)

- **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.

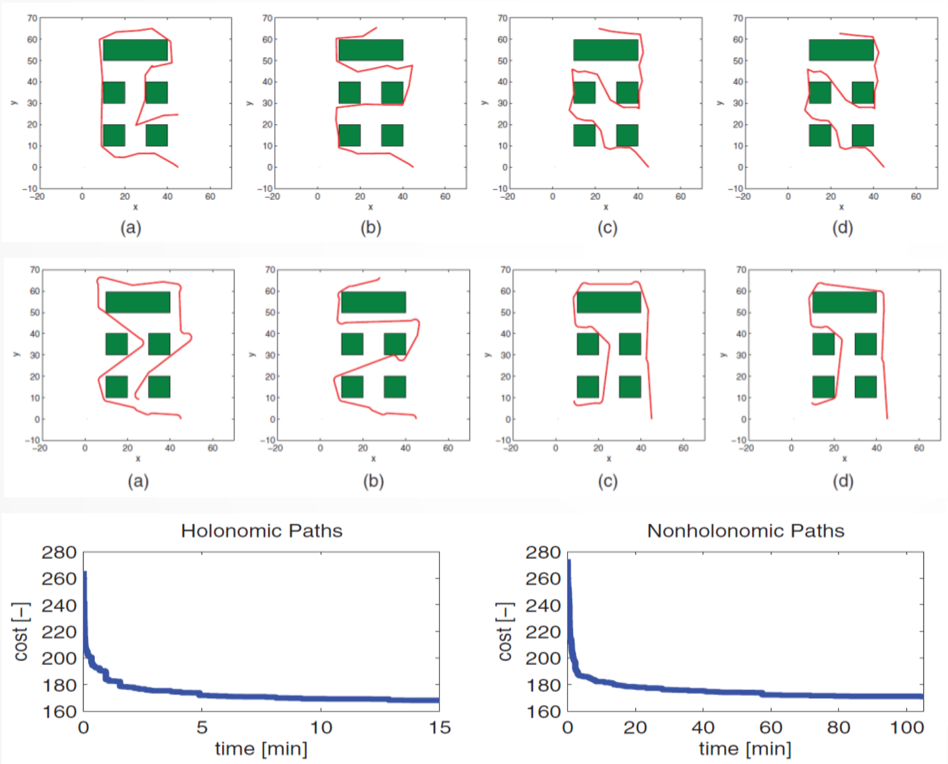


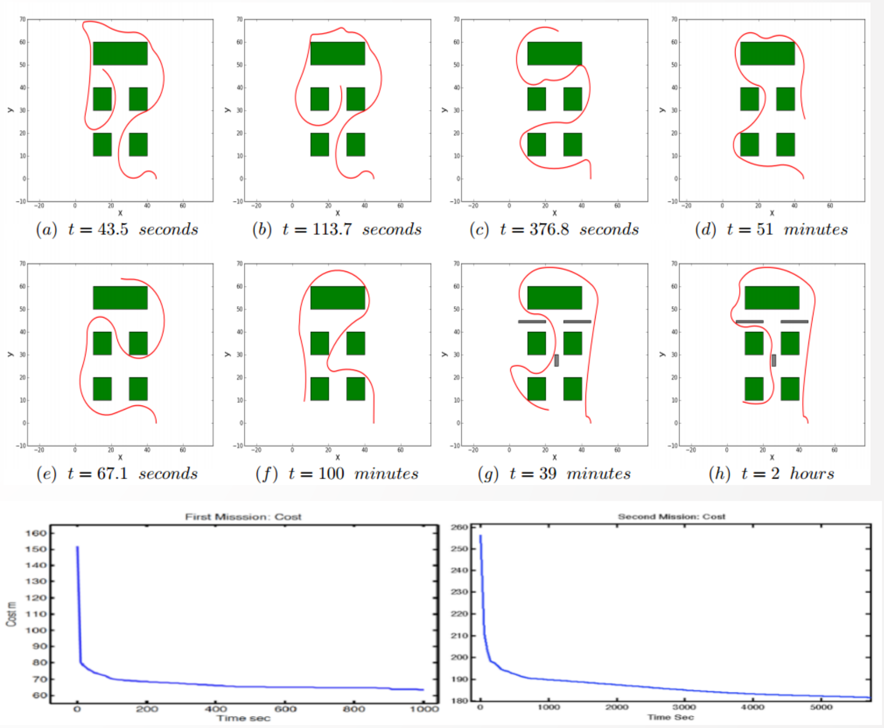
Vary the solution topology – be able to find the optimal solution. X`

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.

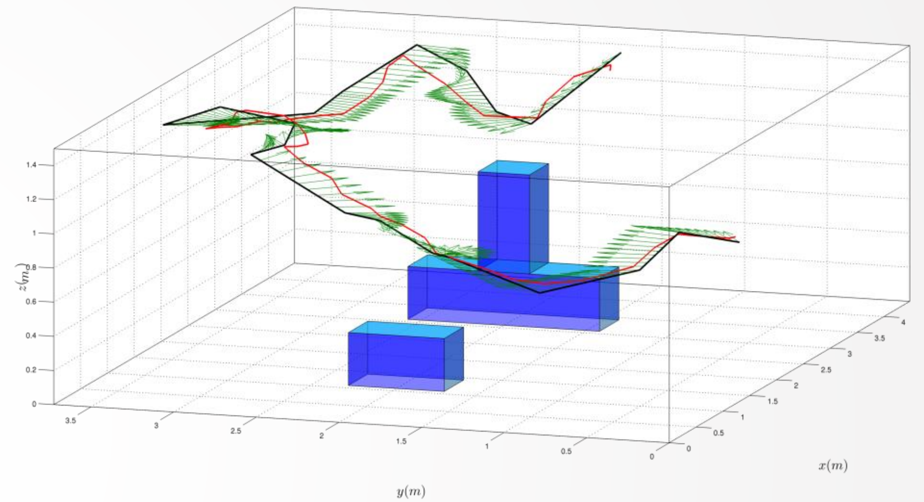




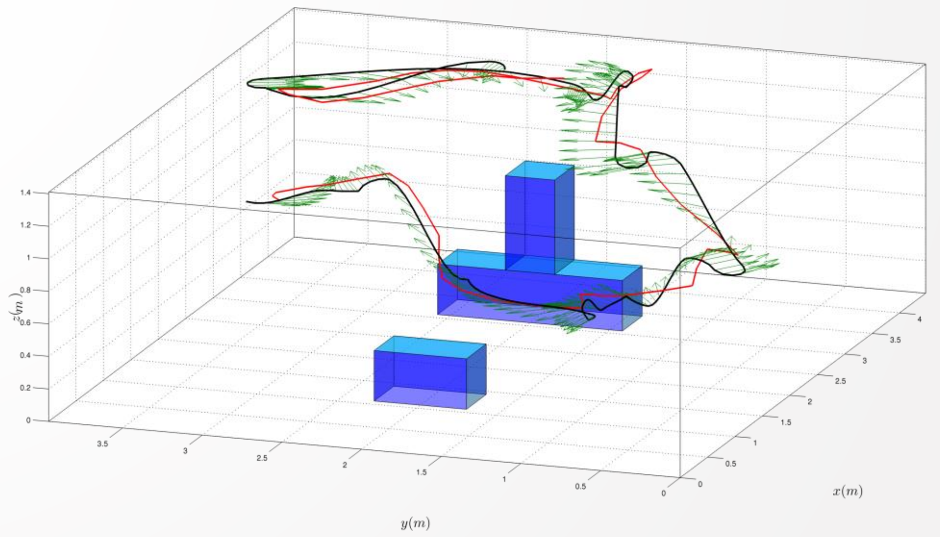
- **Comparison against:** G Papadopoulos, H Kurniawati, N Patrikalakis, "Asymptotically optimal path planning and surface reconstruction for inspection", IEEE International Conference on Robotics and Automation (ICRA) 2013.

RRTOT: Indicative Solutions

- **Holonomic**

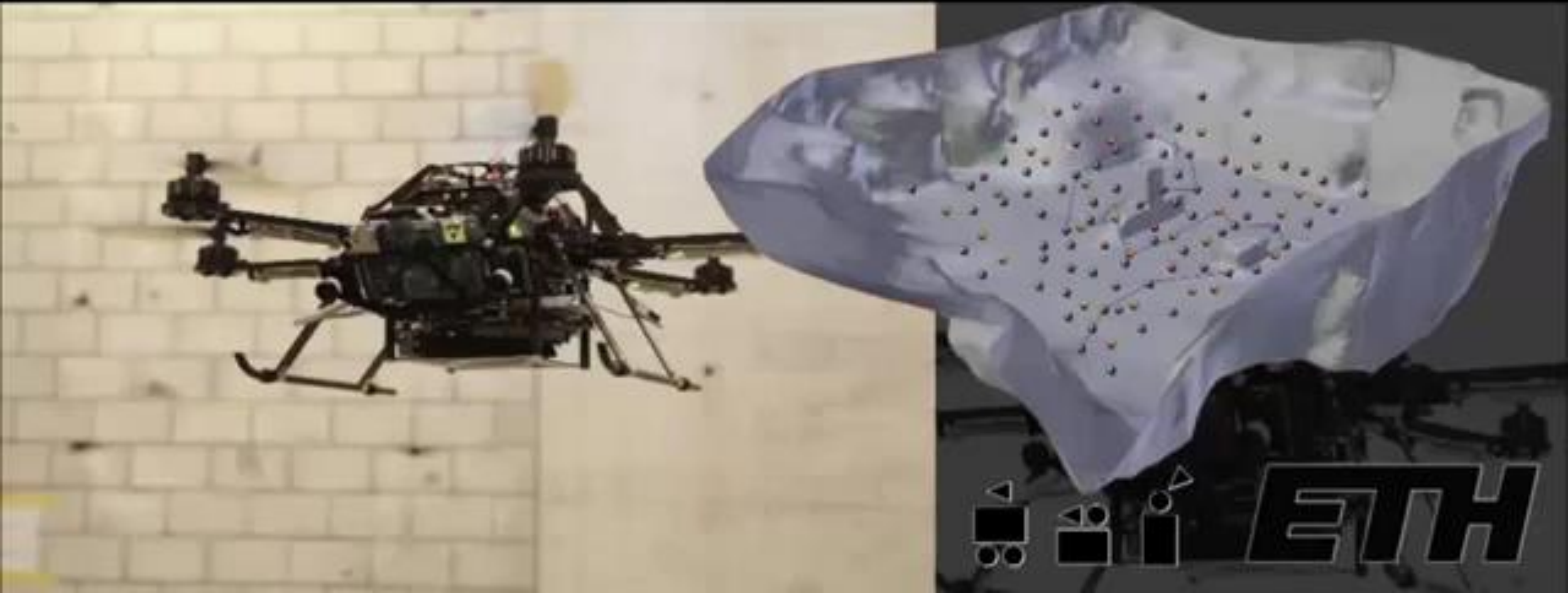


- **Nonholonomic**



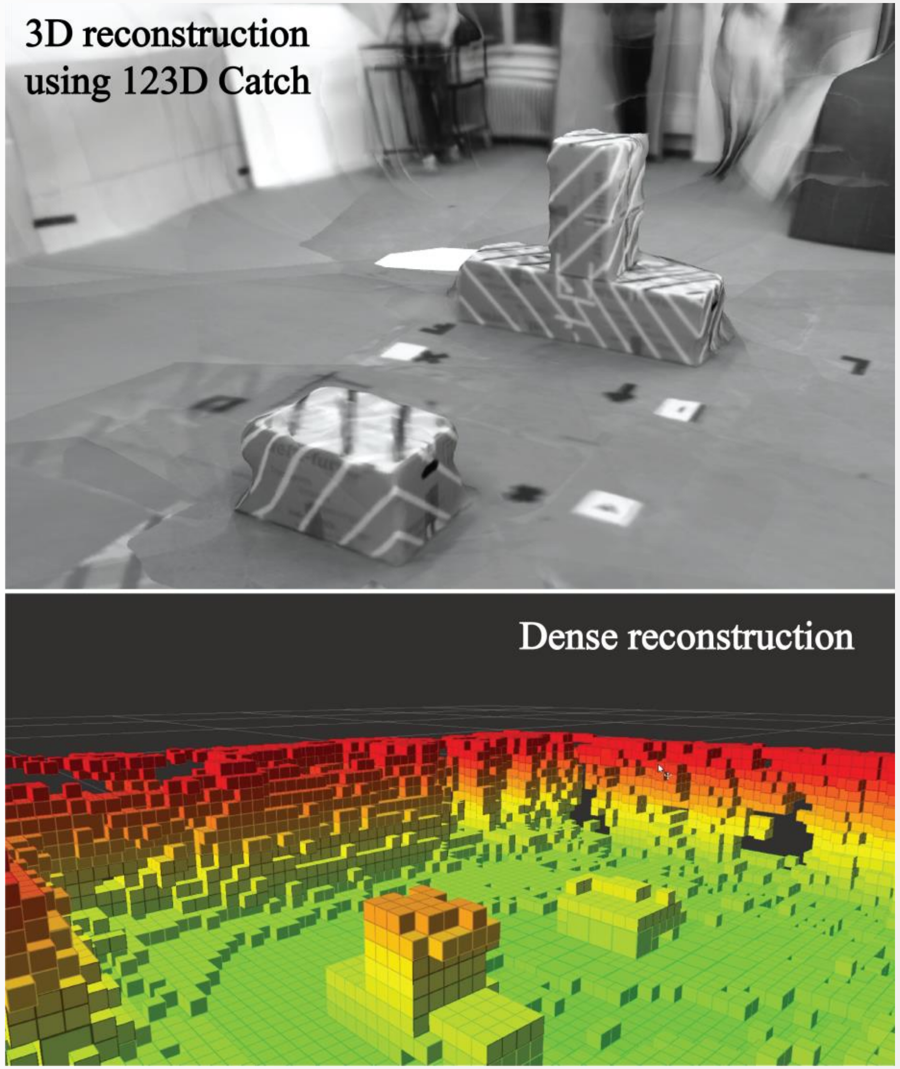
An Incremental Sampling-based approach to Inspection Planning: the Rapidly-exploring Random Tree Of Trees

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



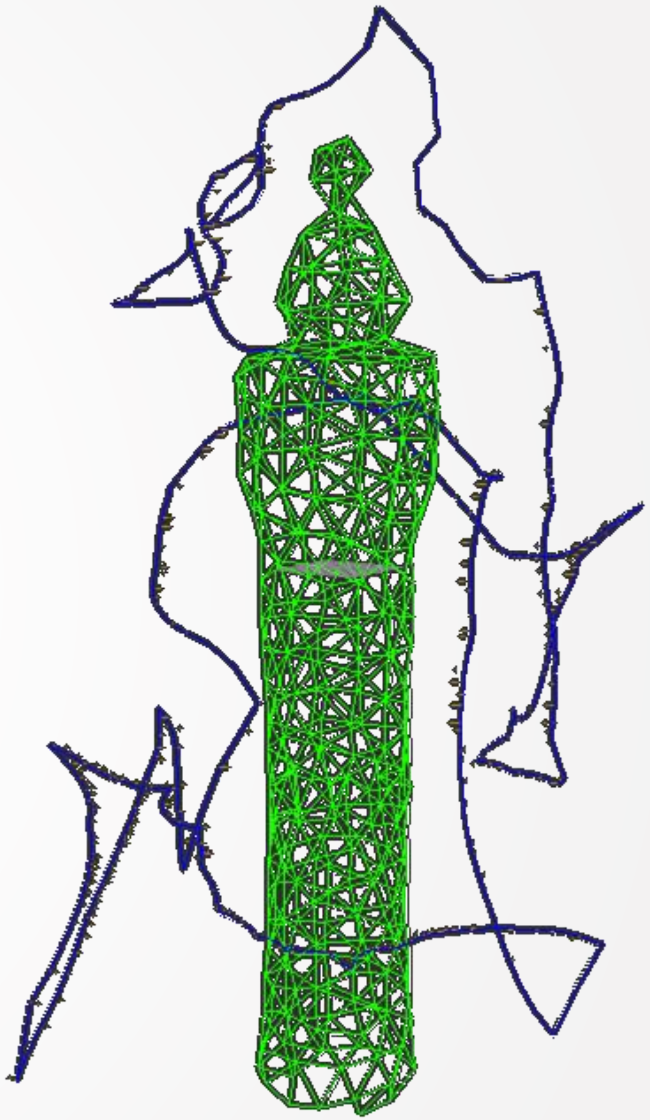
Benefits and Disadvantages

- **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?**

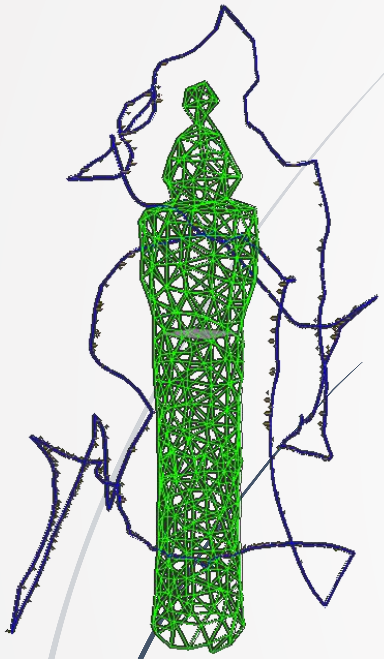


Basic Concepts of the Inspection Planner

- **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

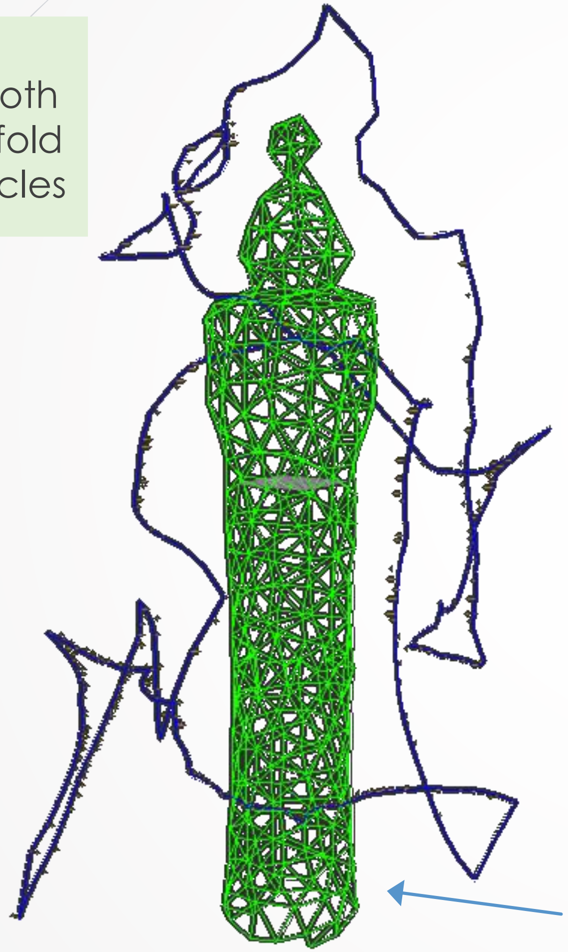
- Load the mesh model
- k = 0
- Sample Initial Viewpoint Configurations (Viewpoint Sampler)
- Find a Collision-free path for all possible viewpoint combinations (BVS, RRT*)
- Populate the Cost Matrix and Solve the Traveling Salesman Problem (LKH)
- **while** running
  - Re-sample Viewpoint Configurations (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***

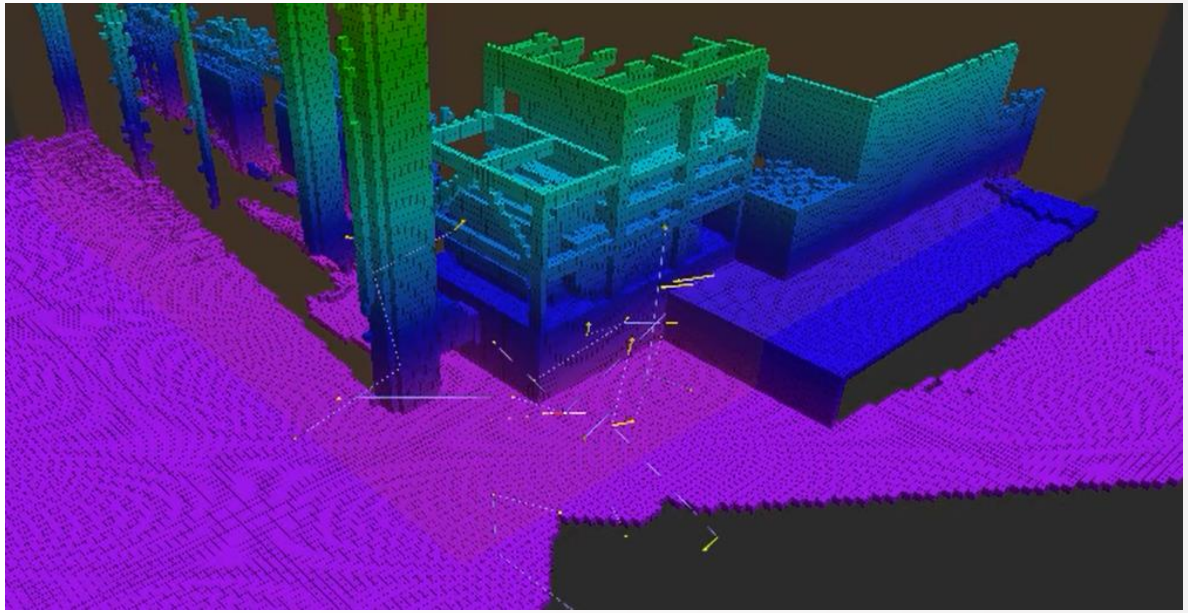
Available Time

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

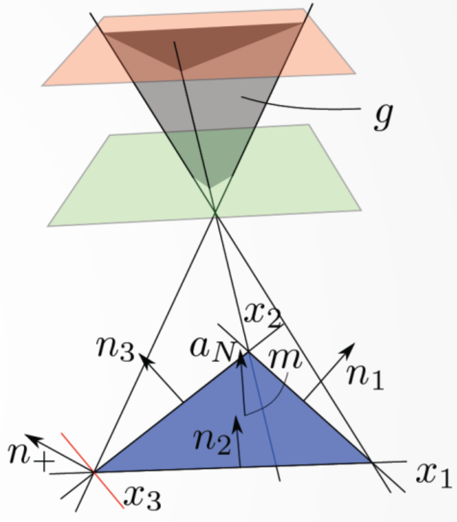
SIP: Viewpoint Sampler

- **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:

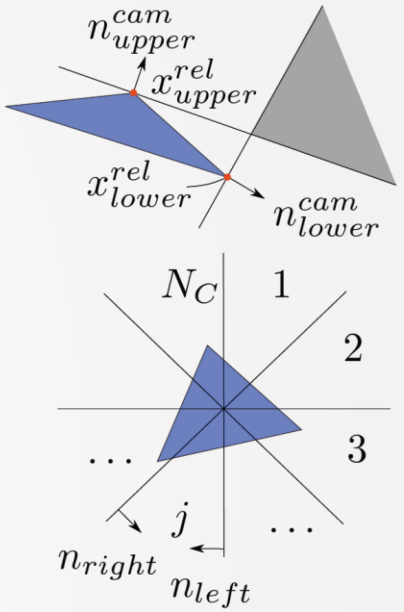
$$\begin{bmatrix} (g - x_i)^T n_i \\ (g - x_1)^T a_N \\ -(g - x_1)^T a_N \end{bmatrix} \succeq \begin{bmatrix} 0 \\ d_{min} \\ -d_{max} \end{bmatrix}, i = \{1, 2, 3\}$$

  - 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.

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



Incidence angle constraints on a triangular surface



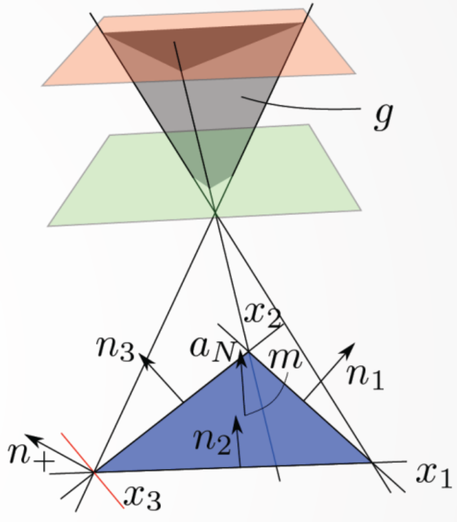
Camera constraints and convexification

SIP: Viewpoint Sampler

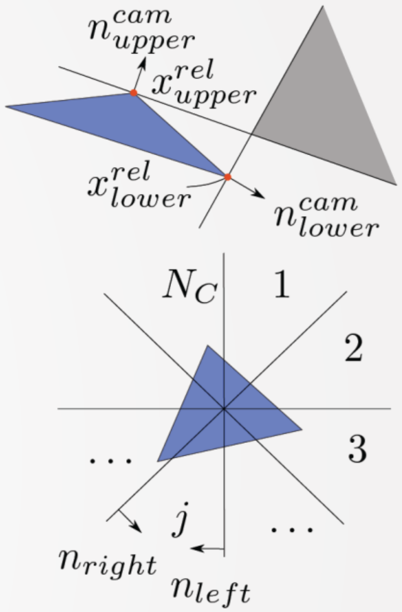
- **Sample 1 Viewpoint/Triangular face**
  - Minimize the sum of squared distances to the preceding viewpoint g_p^{k-1}, the subsequent viewpoint g_s^{k-1} and the current viewpoint in the old tour g^{k-1}.

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

QP + Linear Constraints



Incidence angle constraints on a triangular surface



Camera constraints and convexification

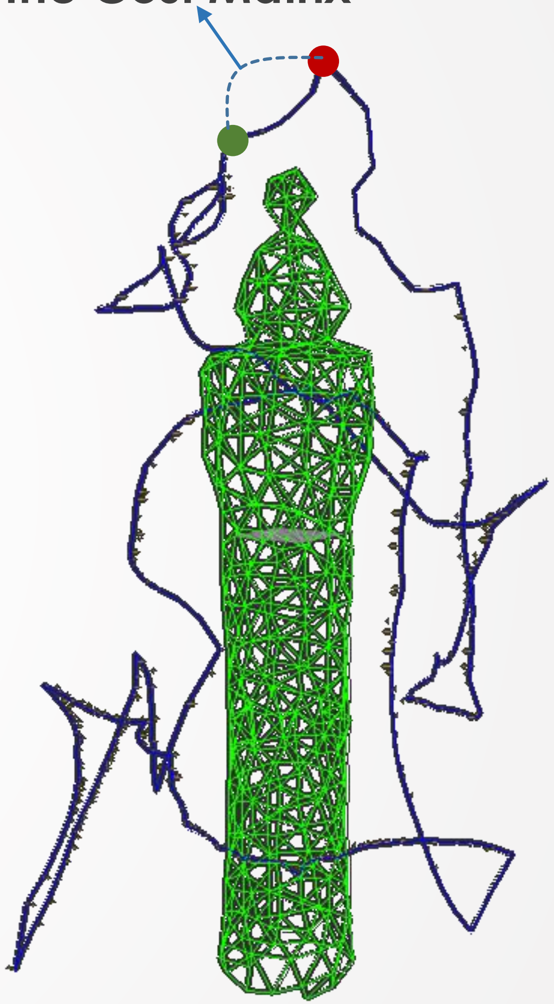
- The heading is determined according to:

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

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

SIP: Point-to-Point Paths

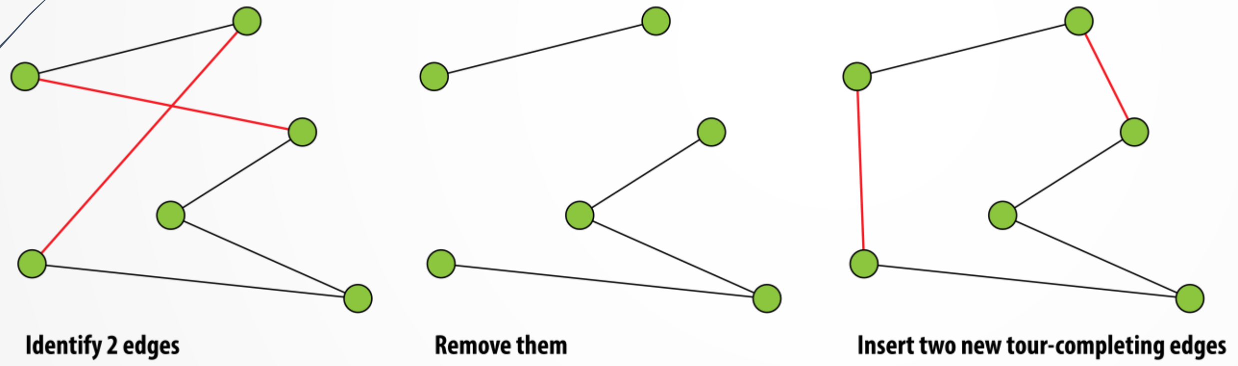
- Compute RRT* Path
- Extract the t_{ex} of the RRT* Path
- Populate the Cost Matrix



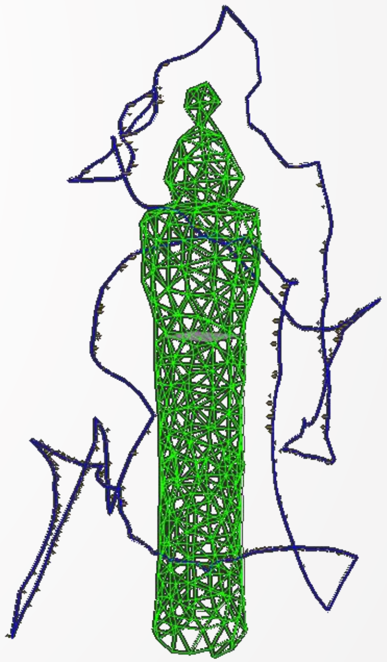
- State-Space Sampling – *extension to Control-Space sampling possible*
- Employ **B**oundary **V**alue **S**olvers for
  - Holonomic (with Yaw-rate constraints) or
  - Nonholonomic Aerial Robots *(fixed-wing UAVs – 2.5D approx., Dubins Airplane approx.)*
- Derive Collision-free paths that connect all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

SIP: TSP Solution

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

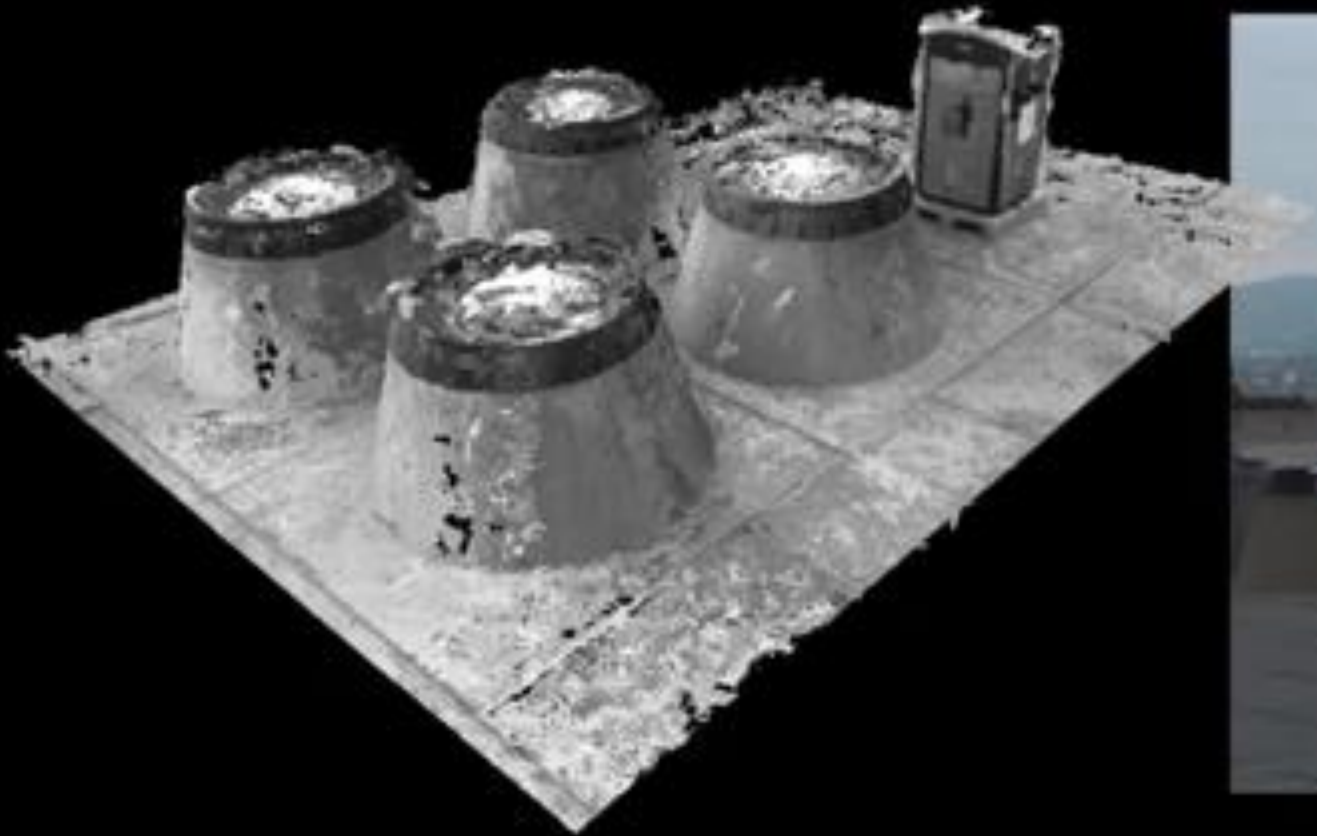


$O(N^{2.2})$, N the number of viewpoints



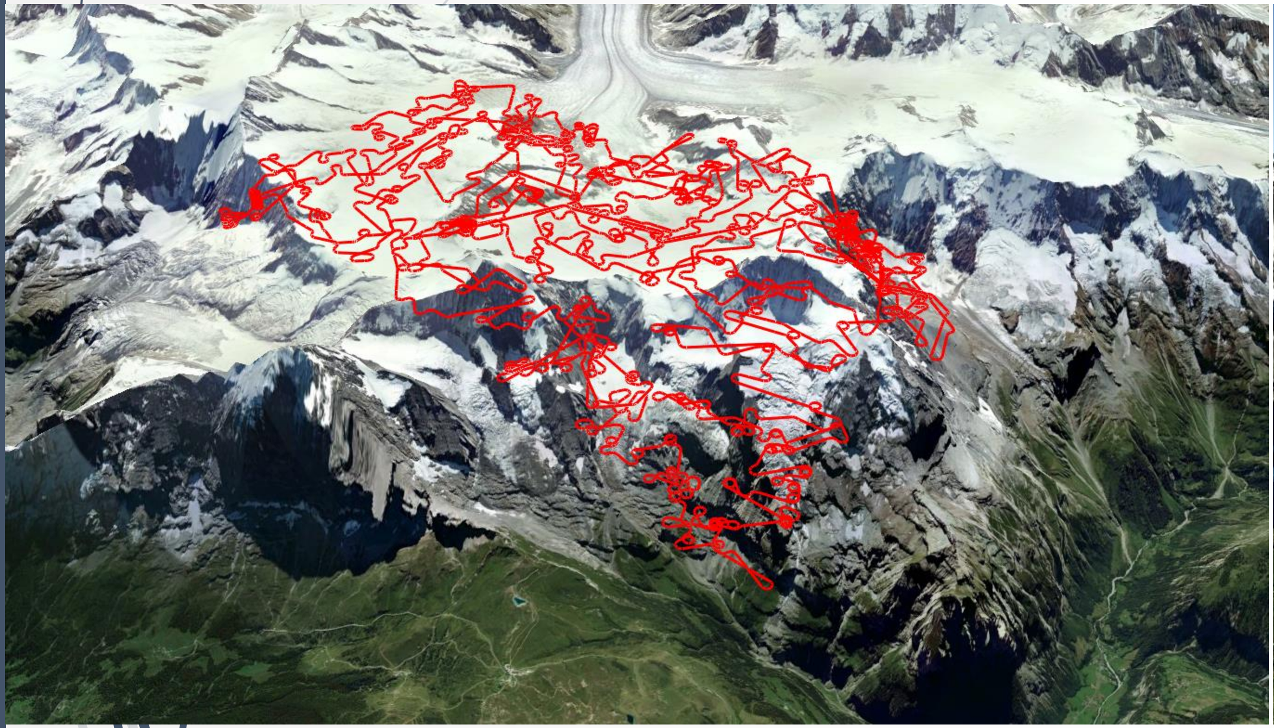
Three-dimensional Coverage Path Planning via Viewpoint Resampling and Tour Optimization using Aerial Robots

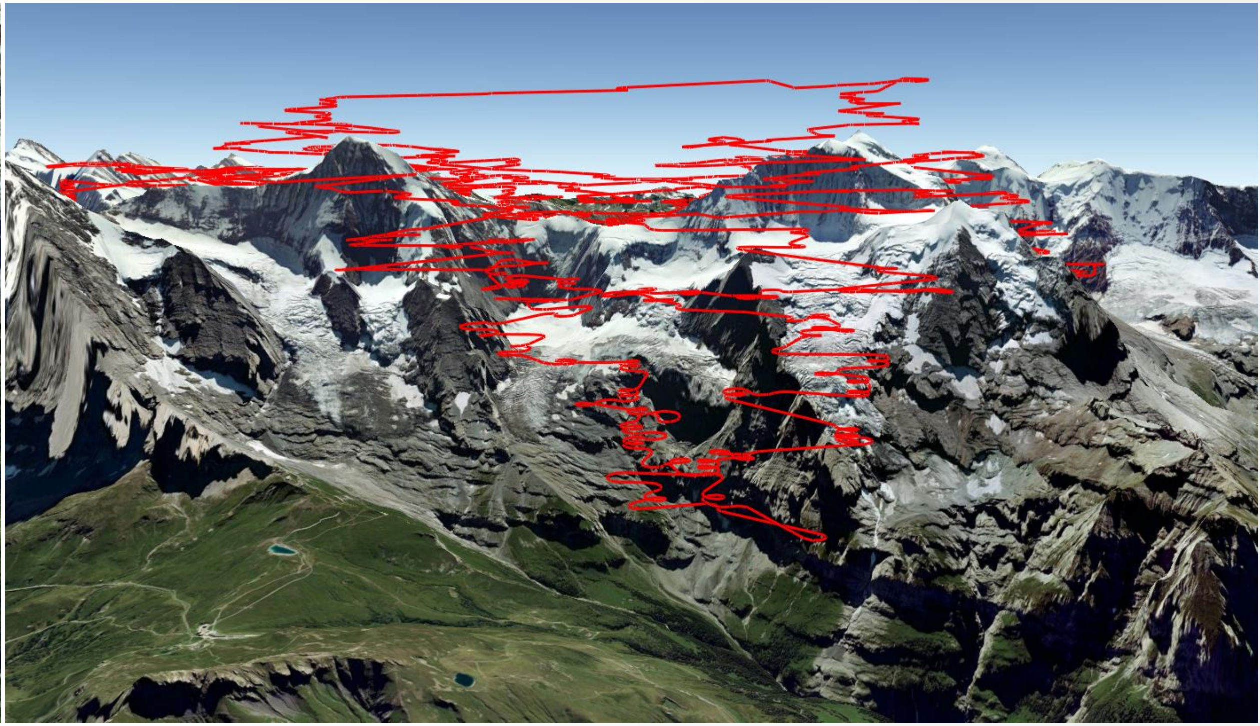
A. Bircher, K. Alexis, M. Kamel, M. Burri, P. Oettershagen, S. Omari, T. Mantel, R. Siegwart




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
ETH

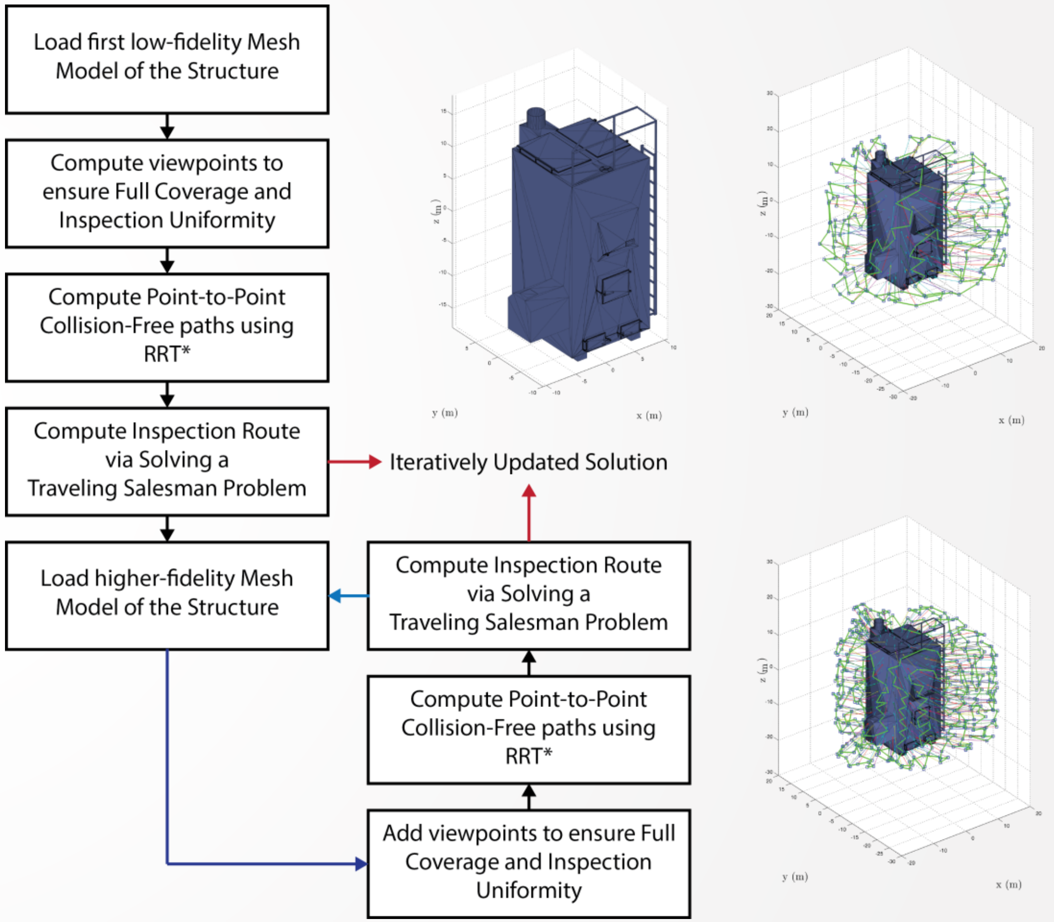
Large Scale Planning: Inspection of the JungFrau mountain (Simulation)



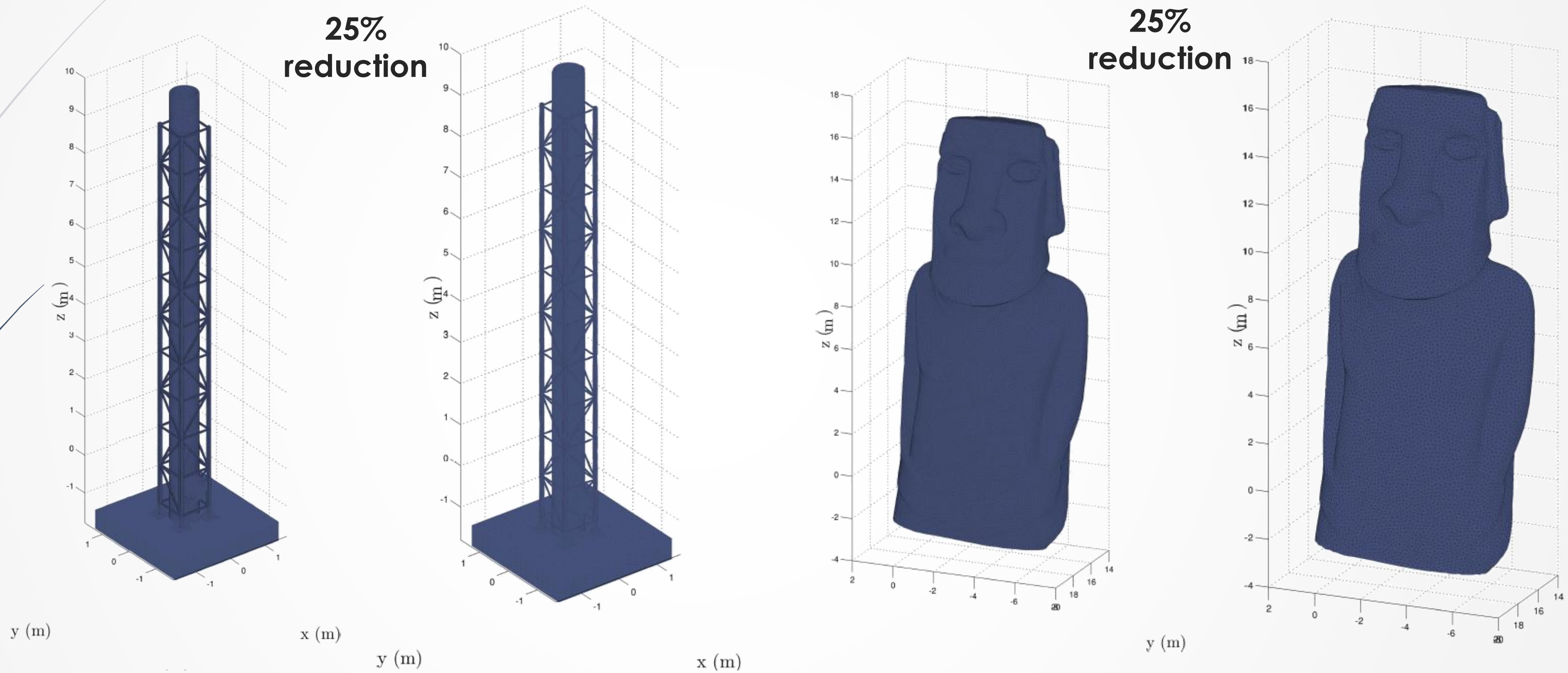


Uniform Coverage Inspection Path-Planning (UC3D)

- **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



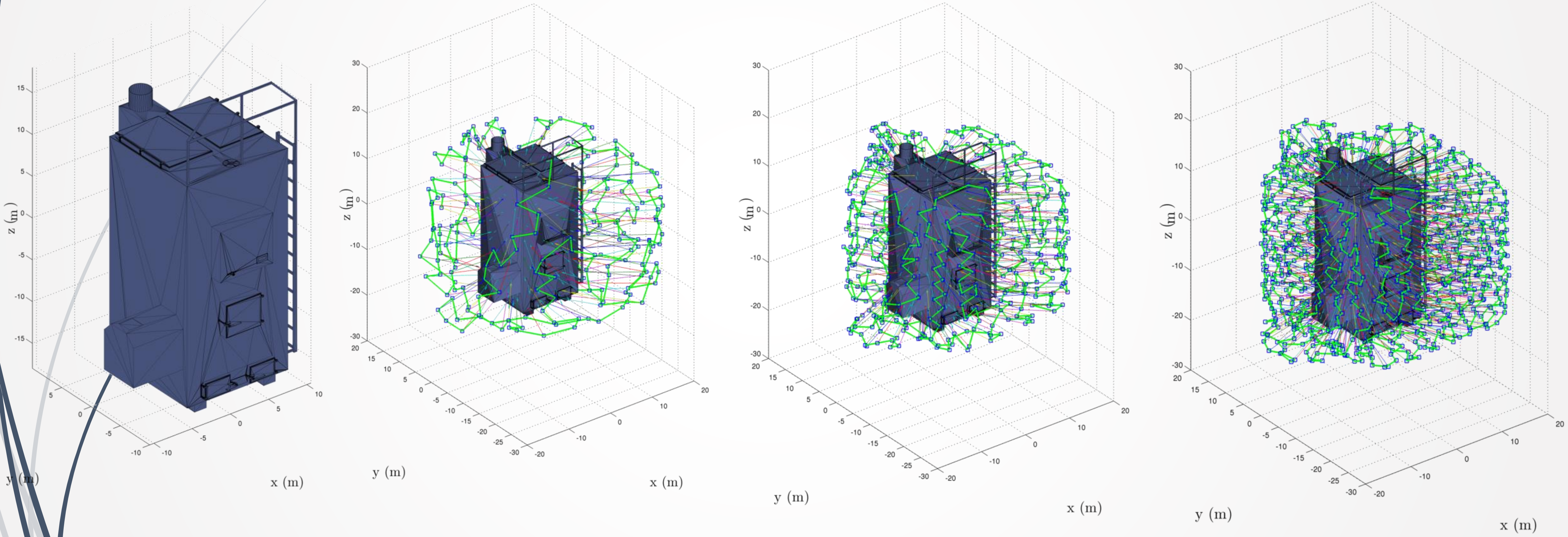
- **Voronoi-based remeshing techniques allow for uniform downsampling of the mesh with minimal structural loss**

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 $\text{IsCoveredUniformly}(\mathbf{p}_{k,i}, \mathcal{V}^{i-1}) ==$ **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 $\mathbf{r}_i$

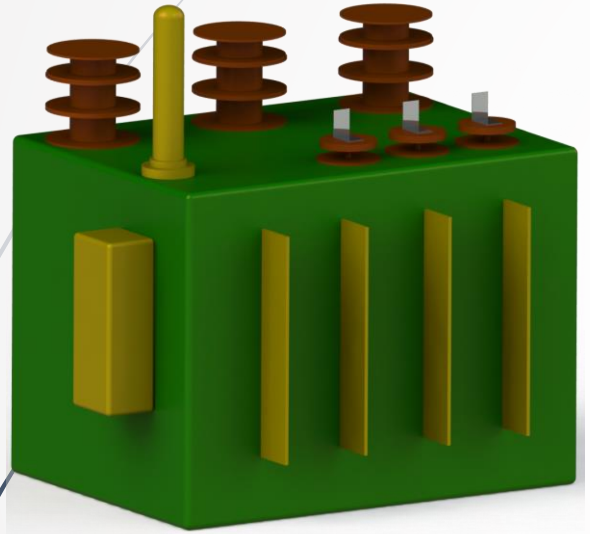
- **Difference of Iterative version:**
  - For each higher quality mesh, instead of computing a whole new set of viewpoints, only some additional are added to re-ensure uniform coverage.

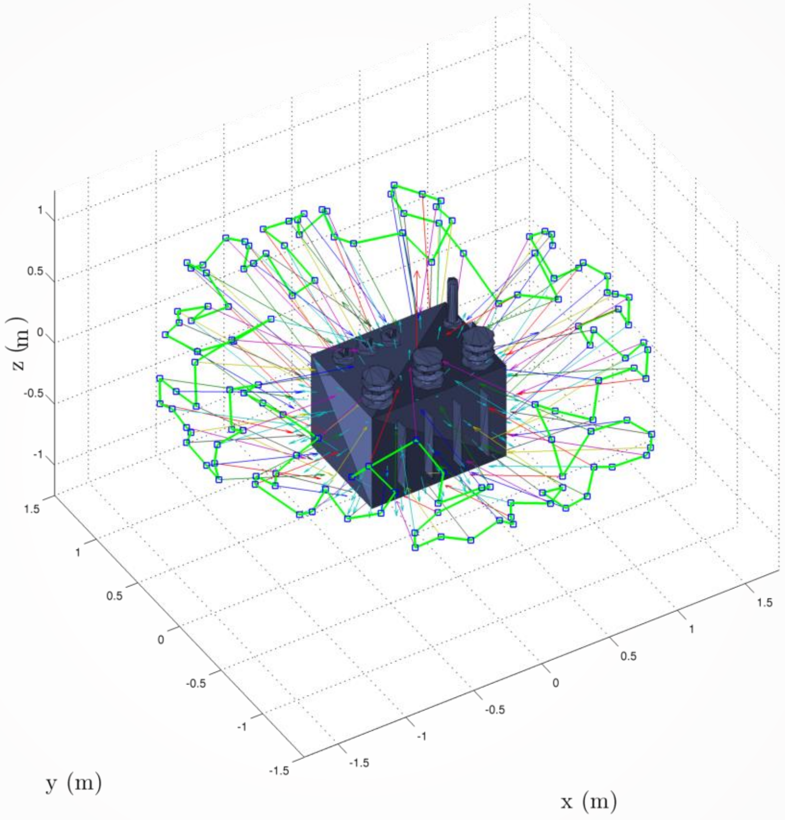
UC3D: Basic UC3D-IPP Result



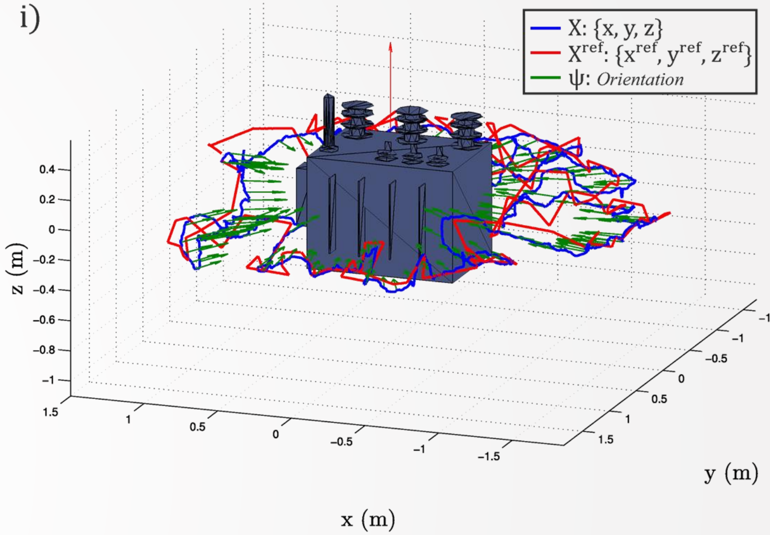
Sequential execution of the basic UC3D-IPP algorithm

UC3D: Experimental study on a Power Transforer MockUp

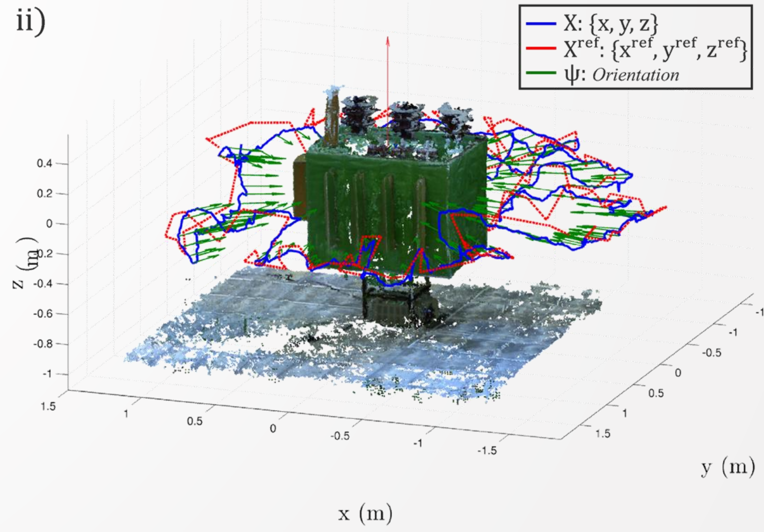




i)

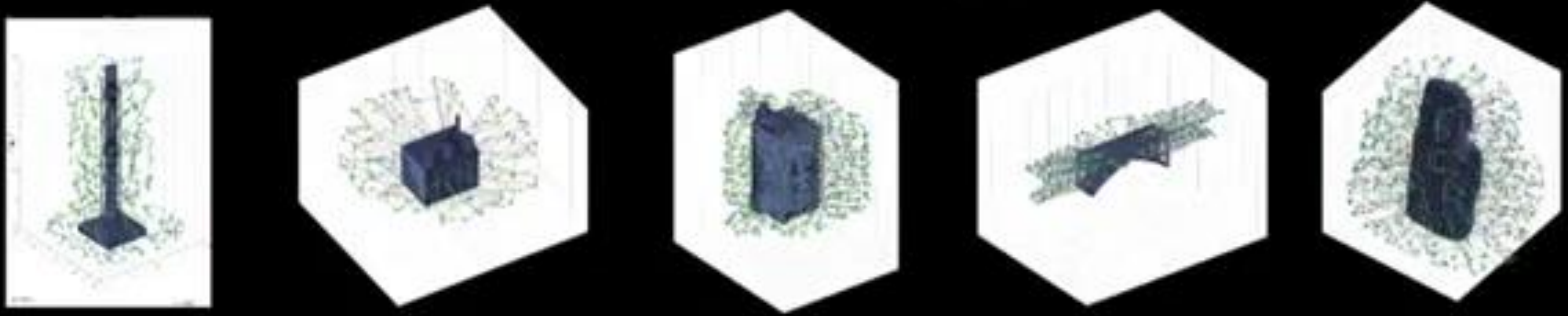


ii)



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

Find out more

- http://www.kostasalexis.com/autonomous-navigation-and-exploration.html
- http://www.kostasalexis.com/holonomic-vehicle-bvs.html
- http://www.kostasalexis.com/dubins-airplane.html
- http://www.kostasalexis.com/collision-free-navigation.html
- http://www.kostasalexis.com/structural-inspection-path-planning.html

- http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-410-principles-of-autonomy-and-decision-making-fall-2010/lecture-notes/
- http://ompl.kavrakilab.org/
- http://moveit.ros.org/
- http://planning.cs.uiuc.edu/

References

- 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!