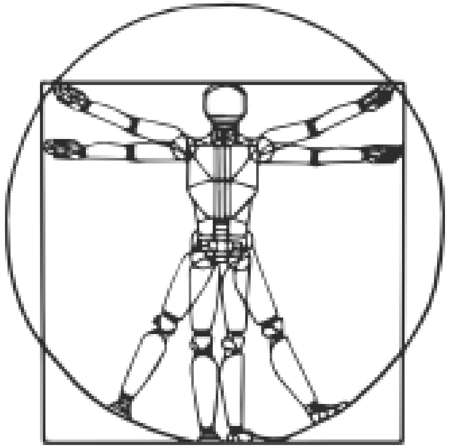
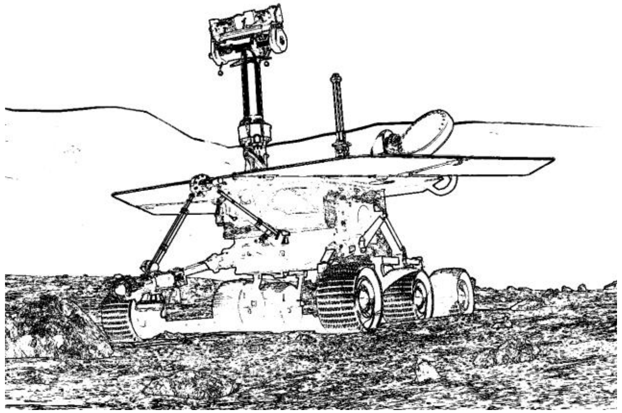




COMP417
Introduction to Robotics and Intelligent Systems

Lecture 10: Map Representations and Alignment

Florian Shkurti

Computer Science Ph.D. student

florian@cim.mcgill.ca

McGill

MRL Mobile Robotics Lab at McGill University

Today’s agenda

- A2 is posted
- Tomorrow’s tutorial: coding Dijkstra in Python
- Correction on state representation for robot arms
- Data structures for representing maps
- Map alignment

Categories of maps

- Metric
  - Map accurately represents lengths and angles
- Topological
  - Map is reduced to a graph representation of the structure of free space
- Topometric
  - Atlas: a combination of local metric maps (nodes) connected via edges
- Sequence of raw time-series observations (e.g. video)
  - No metric or topological information directly represented by the map

Typical operations on maps

- Distance and direction to closest obstacle
- Collision detection: is a given robot configuration in free space?
- Map merging / alignment
- Occupancy updates
- Raytracing

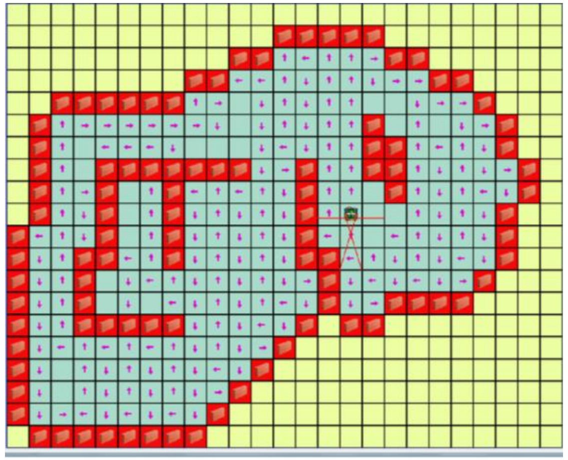
Typical operations on maps

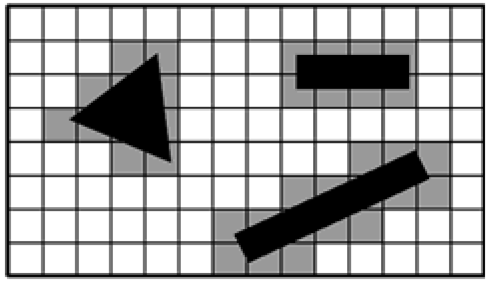
- Distance and direction to closest obstacle
- Collision detection: is a given robot configuration in free space?
- Map merging / alignment
- Occupancy updates
- Raytracing

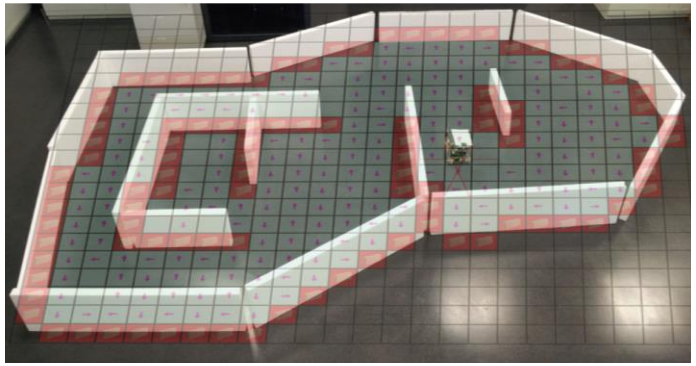
Common operations in computer graphics

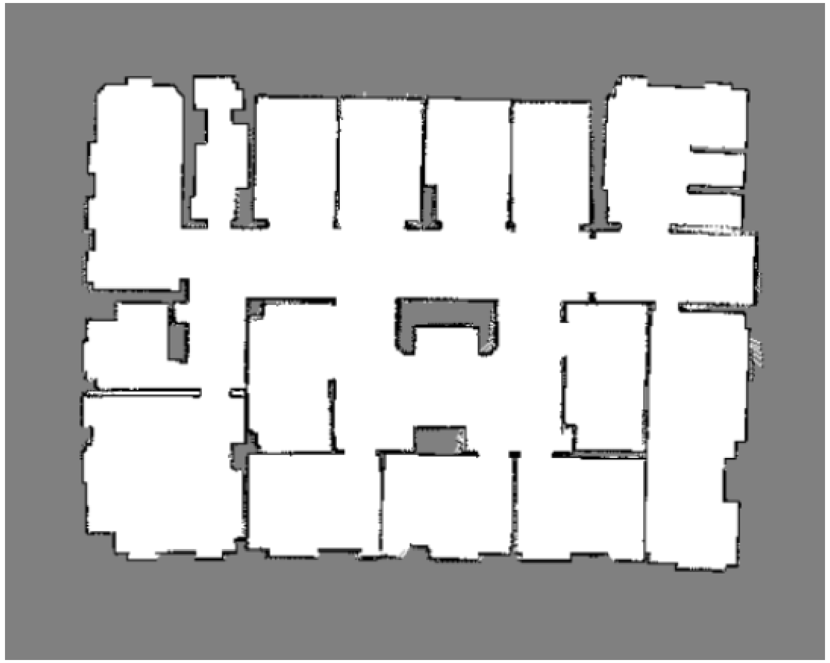
Metric Maps

Occupancy Grids



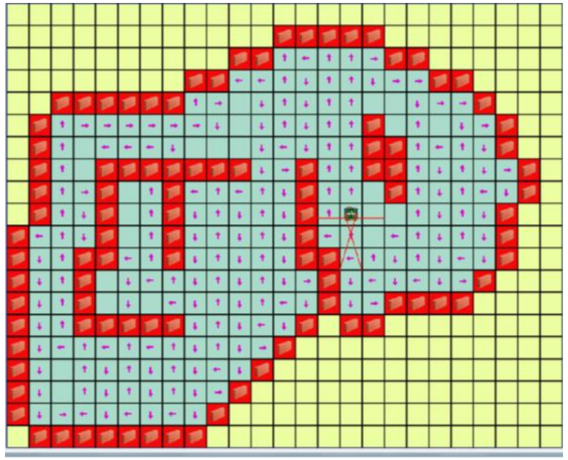


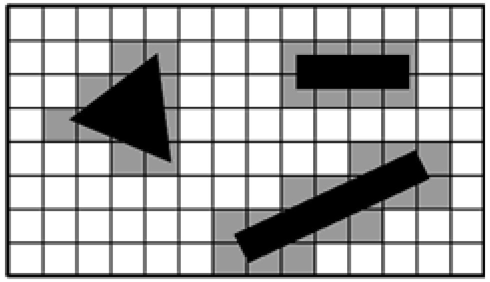




Each cell contains either:
- unknown/unexplored (grey)
- probability of occupation

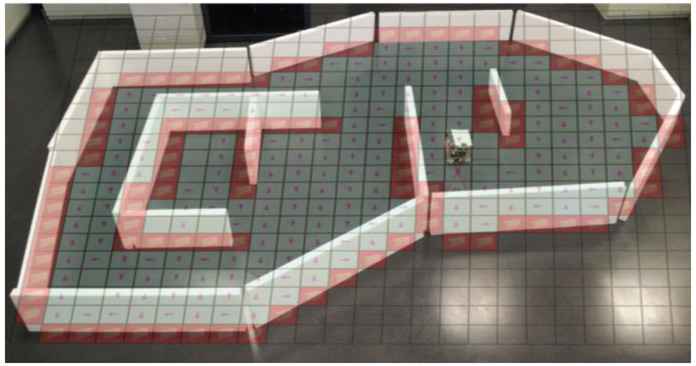
Occupancy Grids

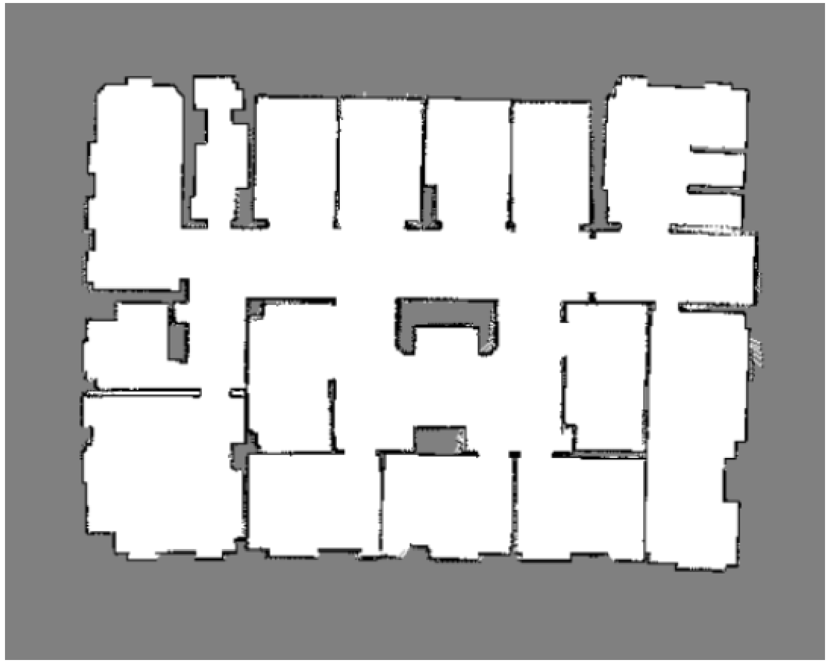




Advantages:
- O(1) occupancy lookup and update
- Supports image operations

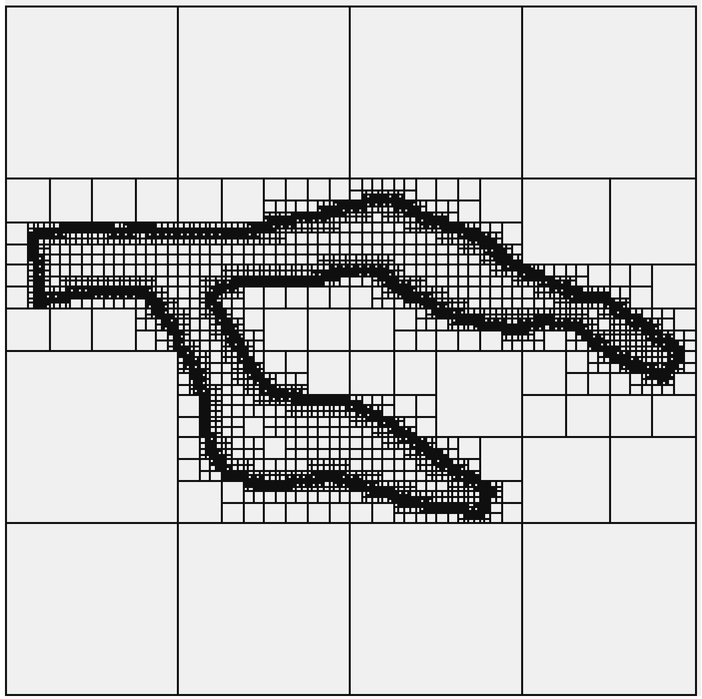
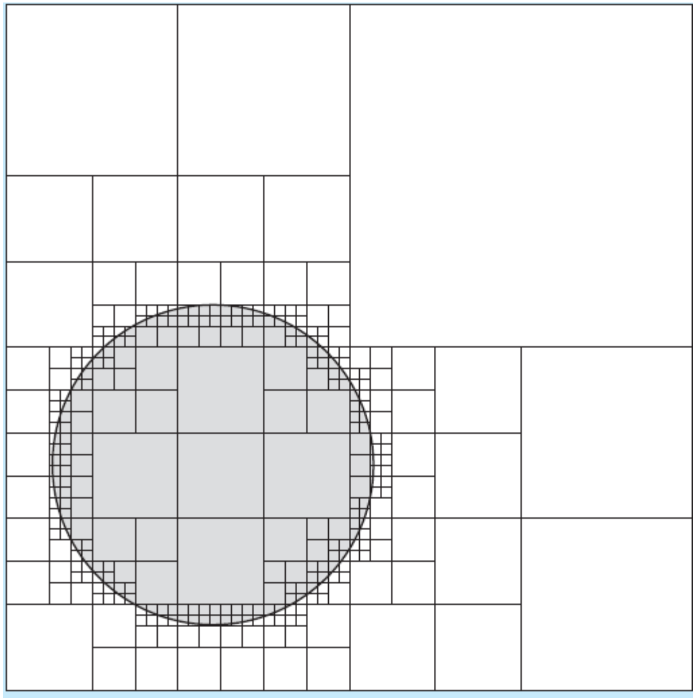
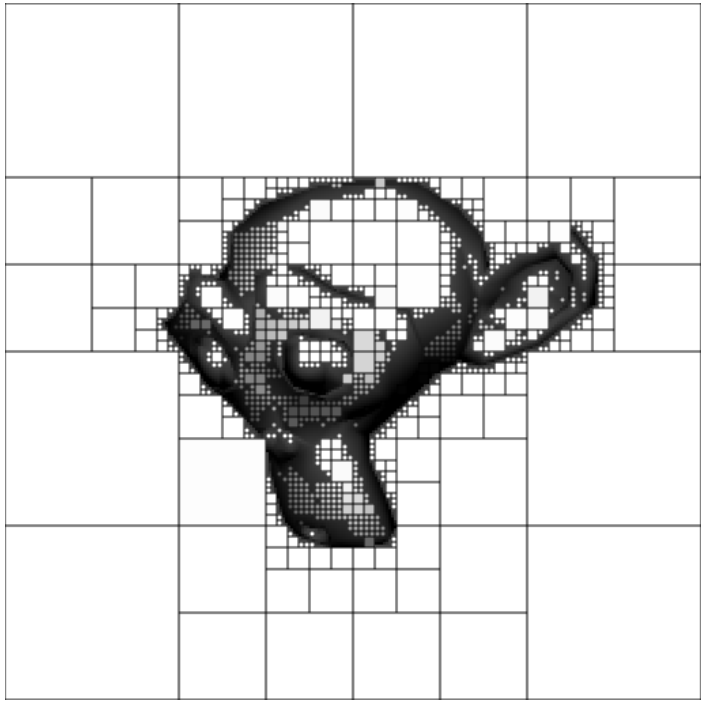
Disadvantages:
- Doesn't scale well in higher dimensions





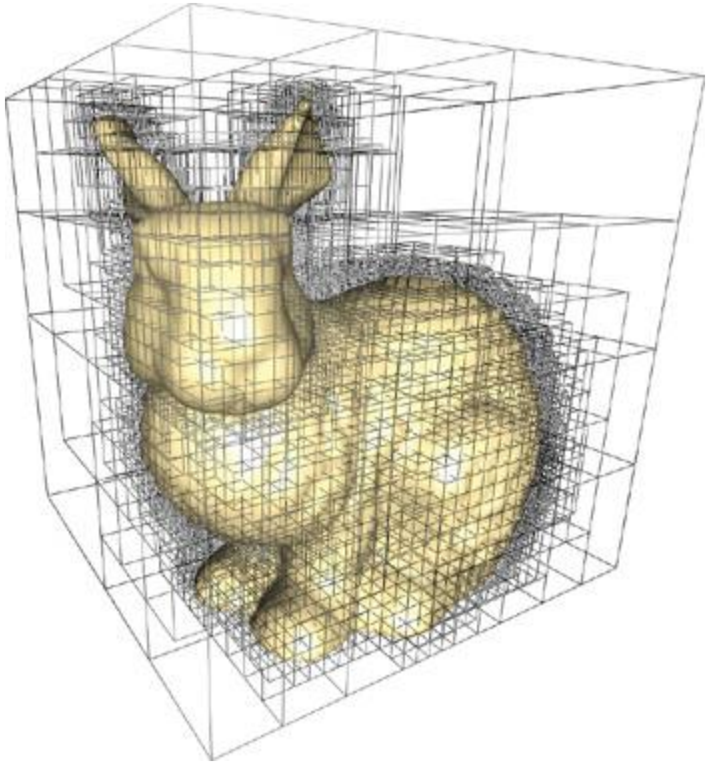
Each cell contains either:
- unknown/unexplored (grey)
- probability of occupation

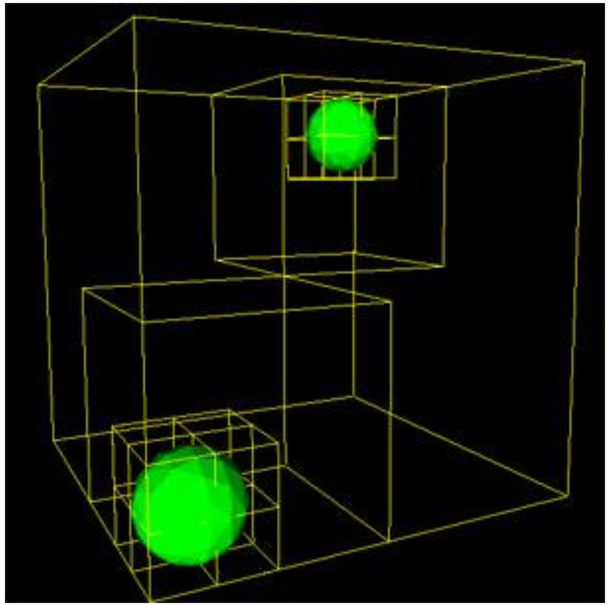
Quadtrees

Each node represents a square. If the node is fully empty or fully occupied it has no children.
If it is partially occupied it has four children. Subdivision stops after some minimal square size.

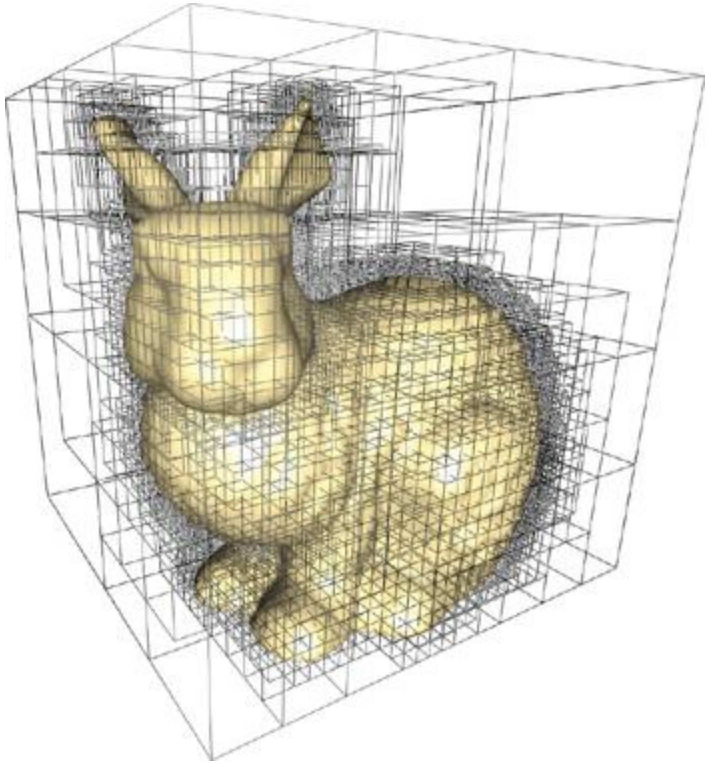
Octrees



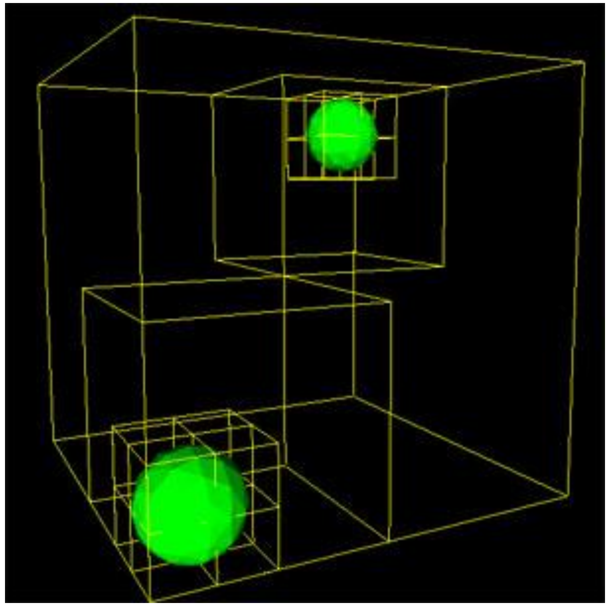


Each node represents a cube. If the node is fully empty or fully occupied it has no children. If it is partially occupied it has eight children. Subdivision stops after some minimal cube size.

Octrees

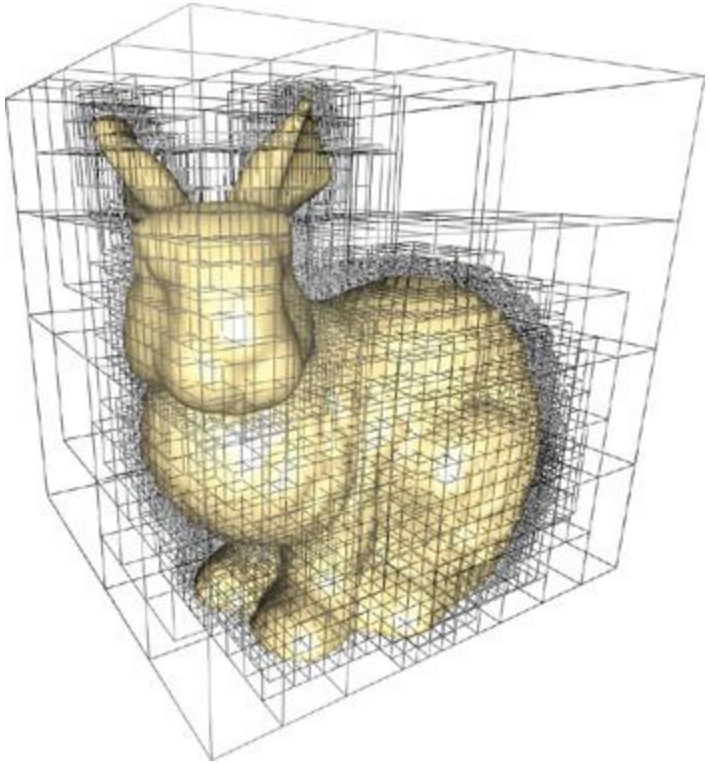


<u>Problem 1</u>: quadtrees and octrees are not balanced trees. So, in the worst case an occupancy query could be O(n) in the number of nodes.



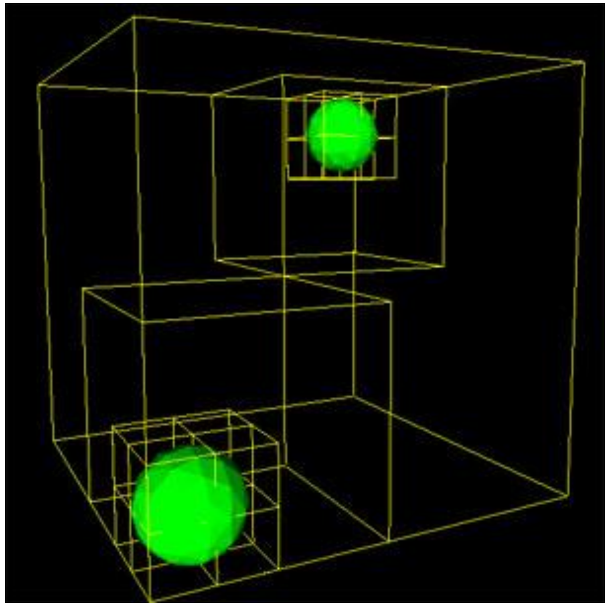
Each node represents a cube. If the node is fully empty or fully occupied it has no children. If it is partially occupied it has eight children. Subdivision stops after some minimal cube size.

Octrees



<u>Problem 1</u>: quadtrees and octrees are not balanced trees. So, in the worst case an occupancy query could be O(n) in the number of nodes.

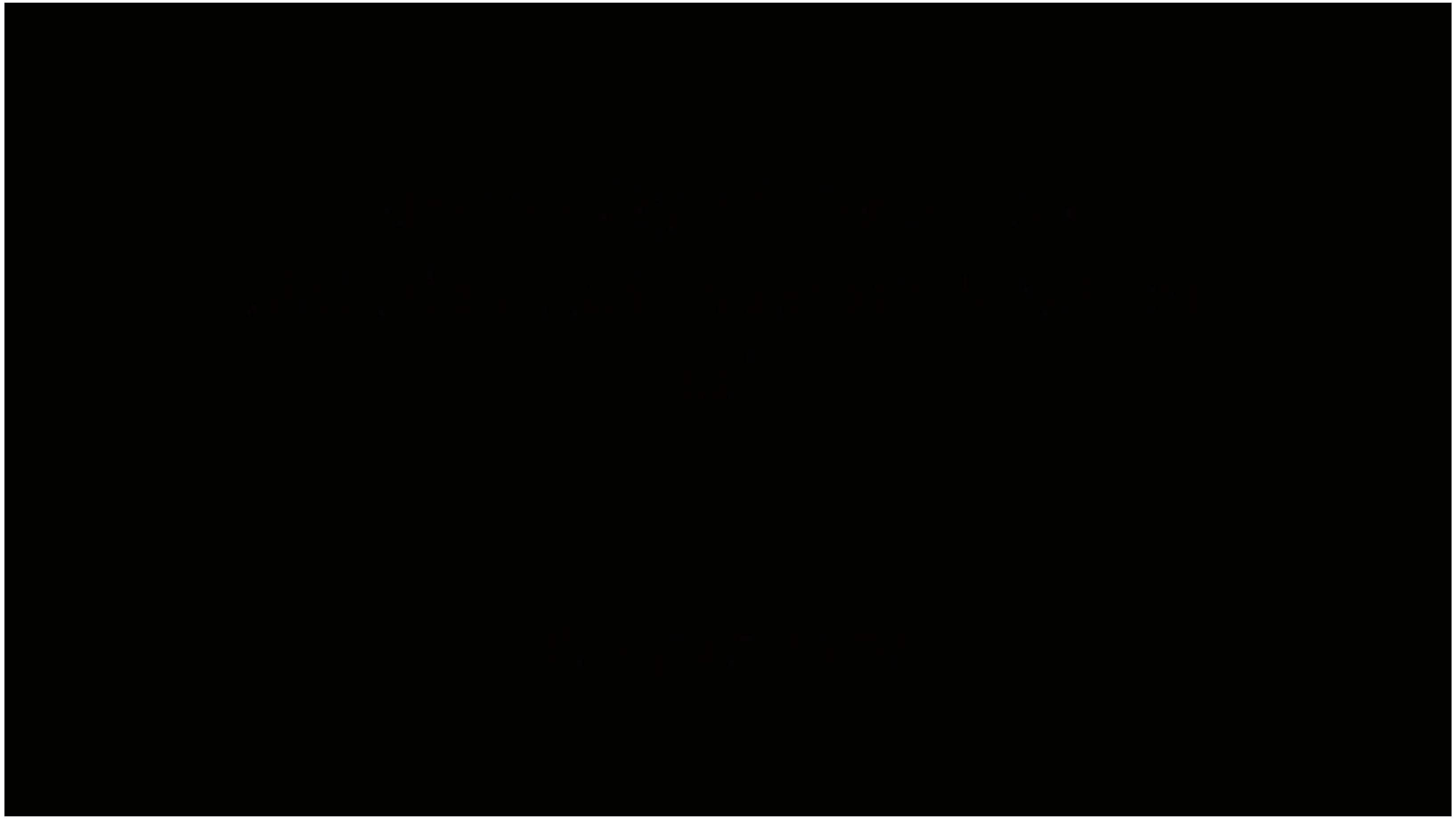
<u>Problem 2</u>: quadtrees and octrees are sensitive to small changes in the location of obstacles.



Each node represents a cube. If the node is fully empty or fully occupied it has no children. If it is partially occupied it has eight children. Subdivision stops after some minimal cube size.

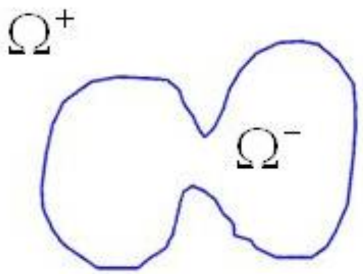
Octree Example: Octomap

Open source
as a ROS package

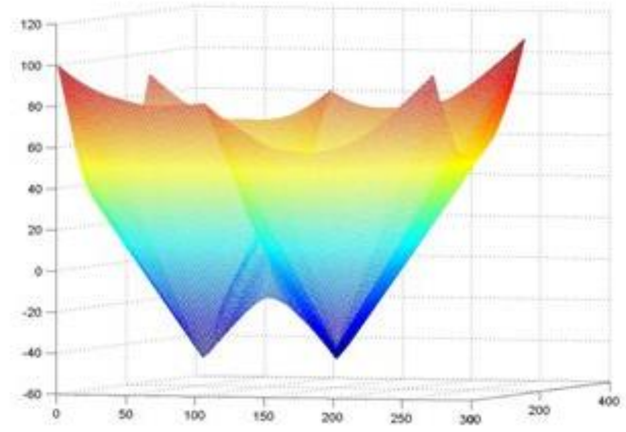


Implicit Surface Definitions: Signed Distance Function

Contour C



Signed distance function

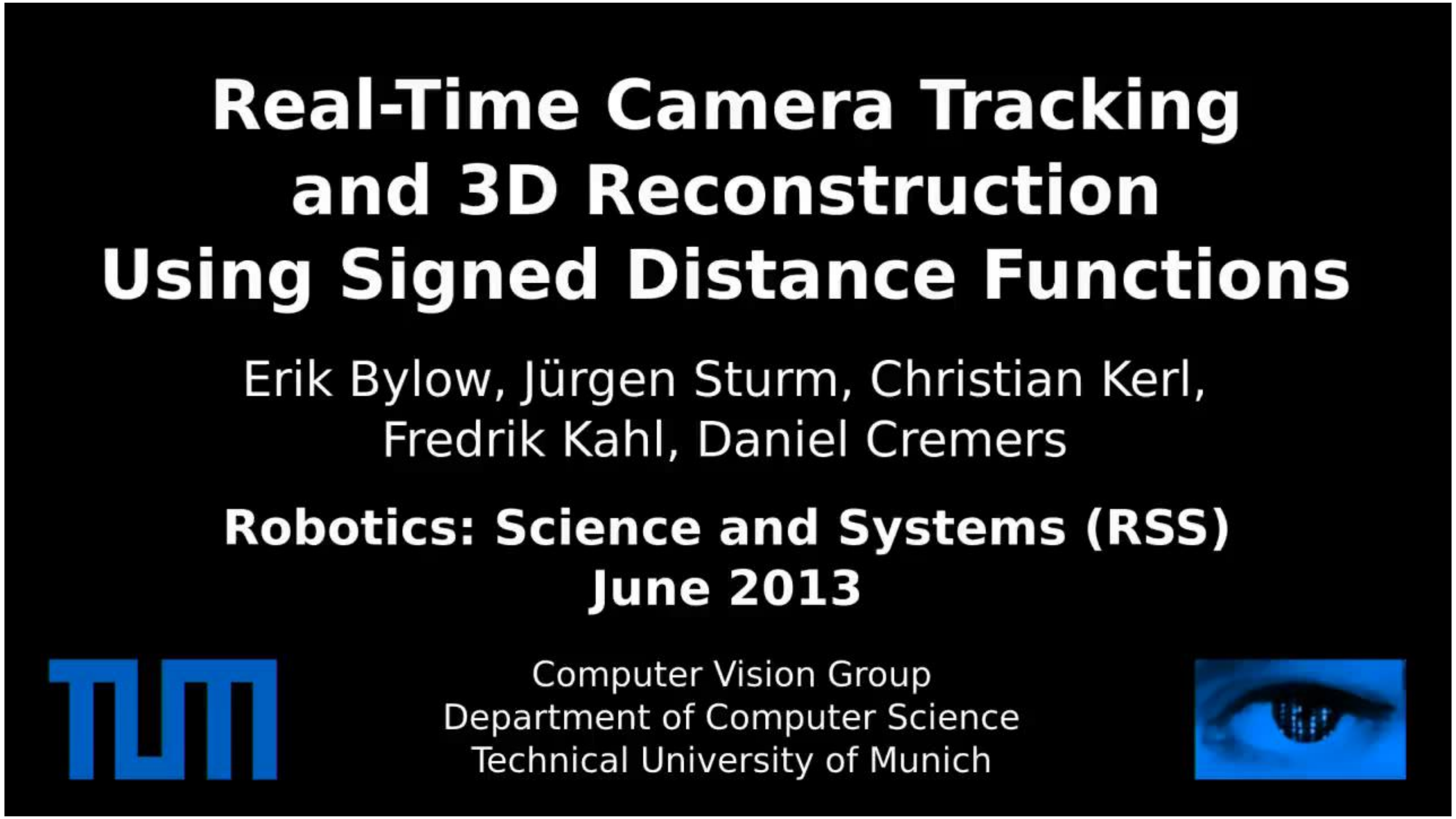


This distance function is defined over any point in 3D space.

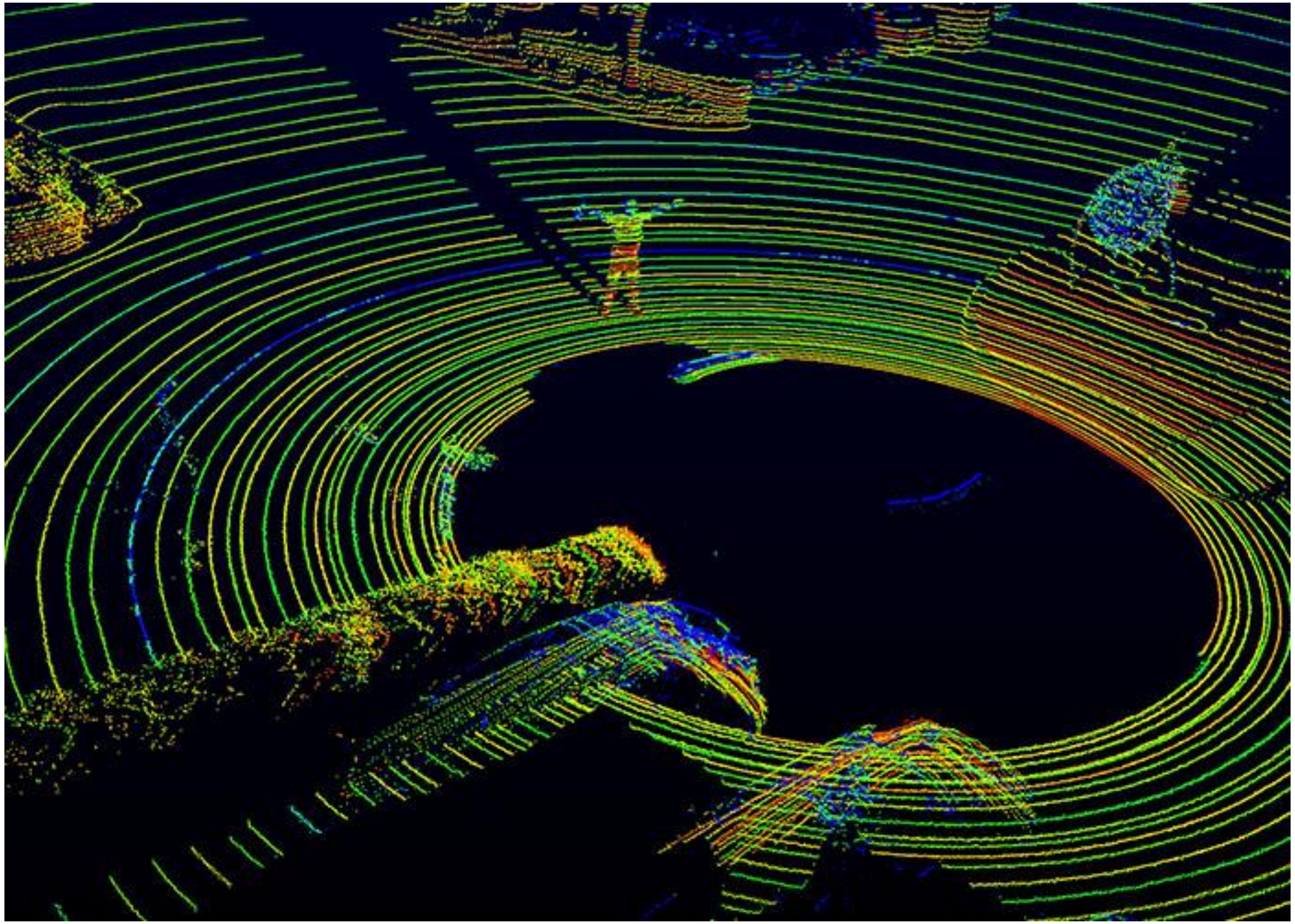
A signed distance function is defined by :

$$\phi(\mathrm{x}) = \begin{cases} dist(\mathrm{x}, C) & \text{if x is outside } C \\ 0 & \mathrm{x} \in C \\ -dist(\mathrm{x}, C) & \text{if x is inside } C \end{cases}$$

SDF Example


Real-Time Camera Tracking
and 3D Reconstruction
Using Signed Distance Functions
Erik Bylow, Jürgen Sturm, Christian Kerl,
Fredrik Kahl, Daniel Cremers
Robotics: Science and Systems (RSS)
June 2013
Computer Vision Group
Department of Computer Science
Technical University of Munich

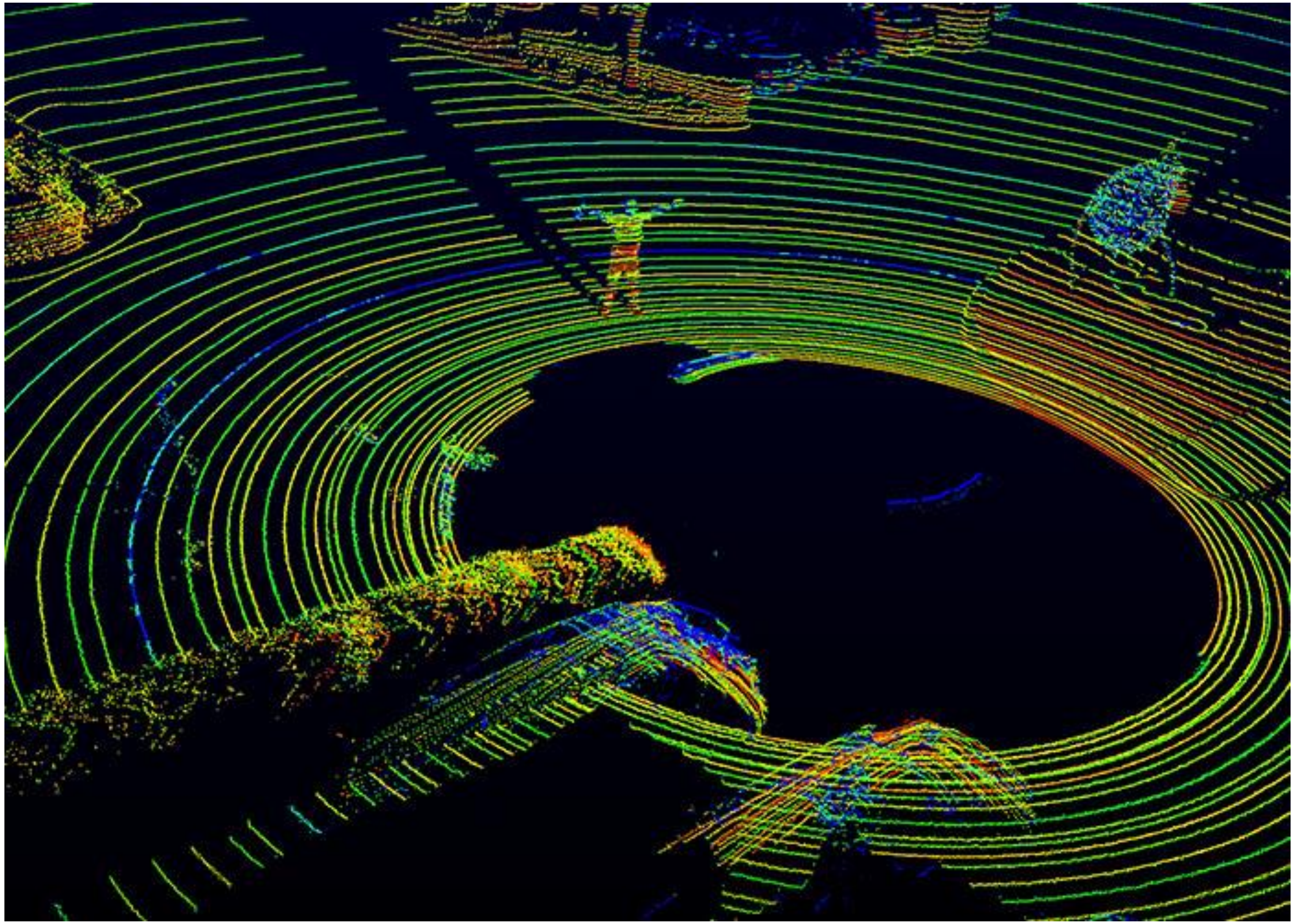
Pointclouds



Pointclouds

Advantages:
- can make local changes to the map without affecting the pointcloud globally
- can align pointclouds
- nearest neighbor queries are easy with kd-trees or locality-sensitive hashing

Disadvantages:
- need to segment objects in the map
- raytracing is approximate and nontrivial



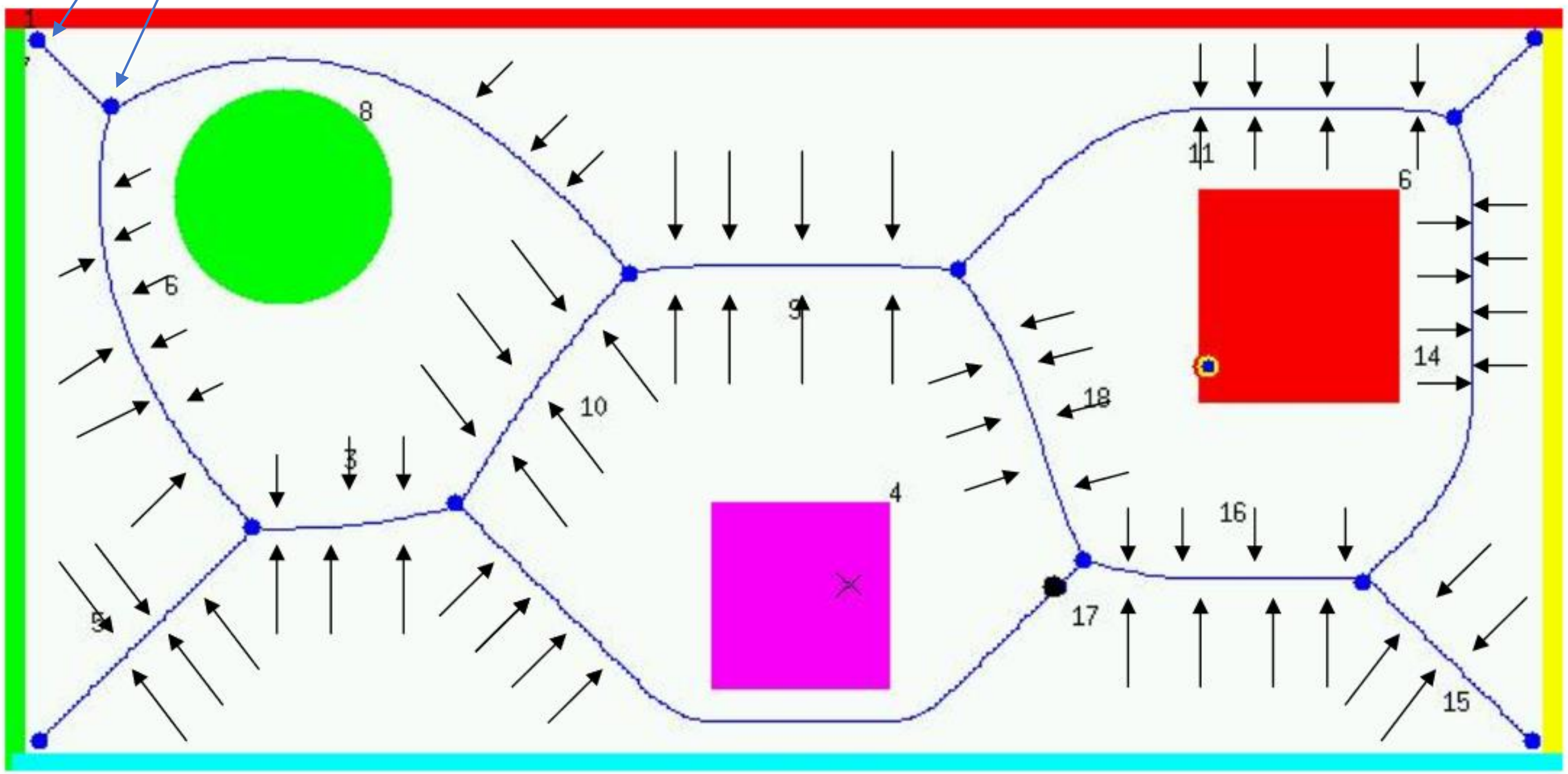
Topological Maps

Topology: study of spatial properties that are preserved under continuous deformations of the space.

Generalized Voronoi Graph (GVG)

Deformation Retraction: GVG in Plane

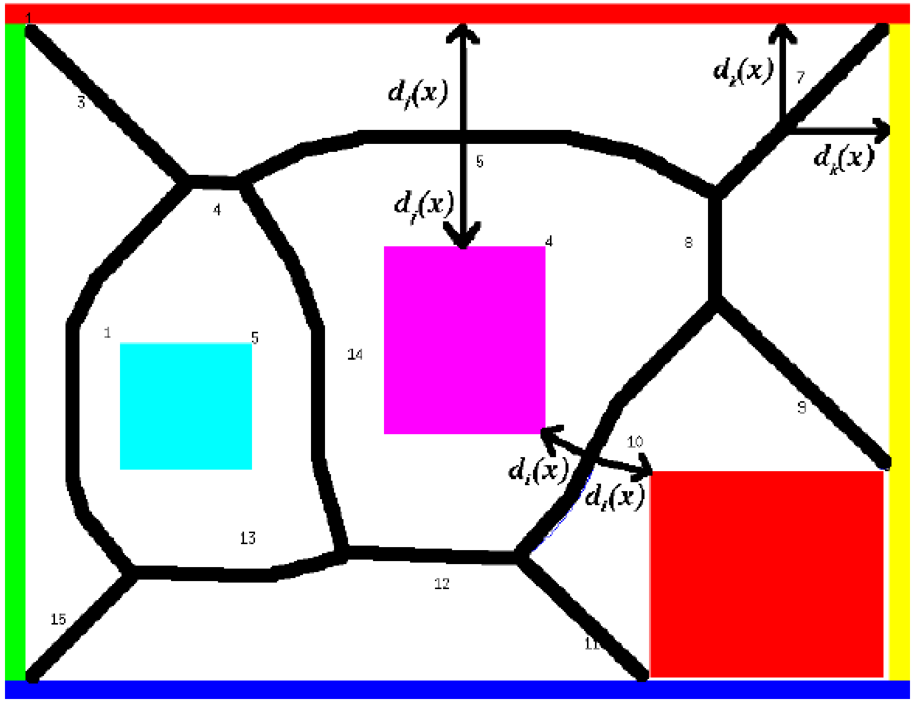
GVG nodes: points that are equidistant to 3 or more obstacle points



Retractions are also called roadmaps.

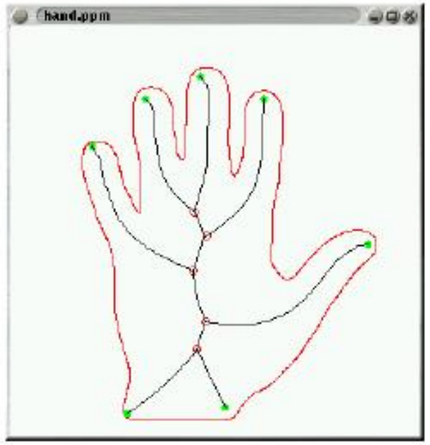
Roadmap: Voronoi diagrams

- GVG is formed by paths equidistant from the two closest objects
- maximizing the clearance between the obstacles.



- This generates a very safe roadmap which avoids obstacles as much as possible

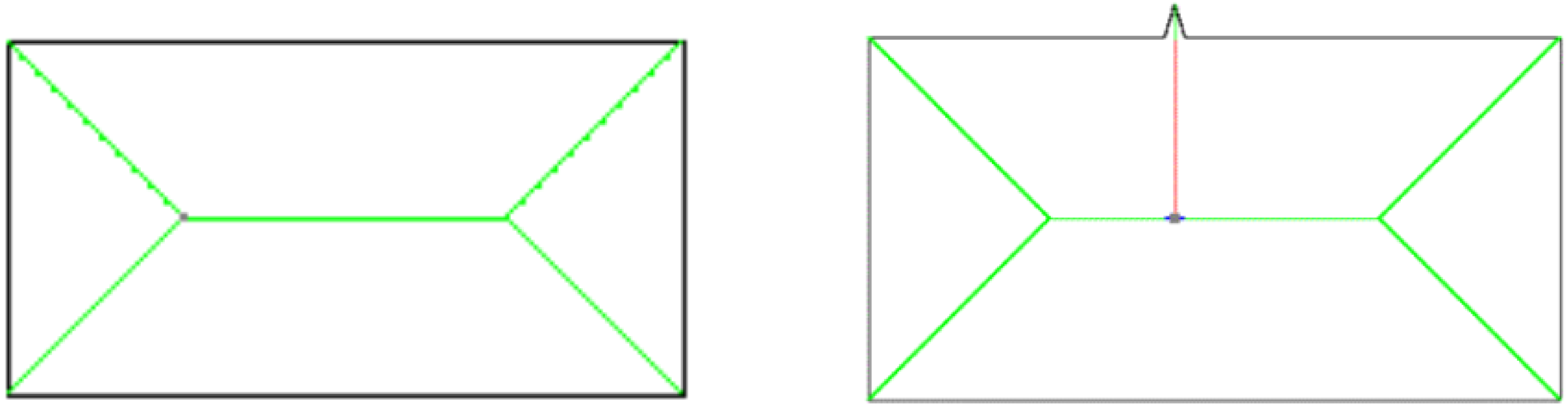
Generalized Voronoi Graphs (GVG)



in 2d, it's called
a *medial axis*

Turns comparison between pixels to comparison between graphs.

GVG: sensitivity



The skeleton is sensitive to small changes in the object's boundary.

GVG: advantages

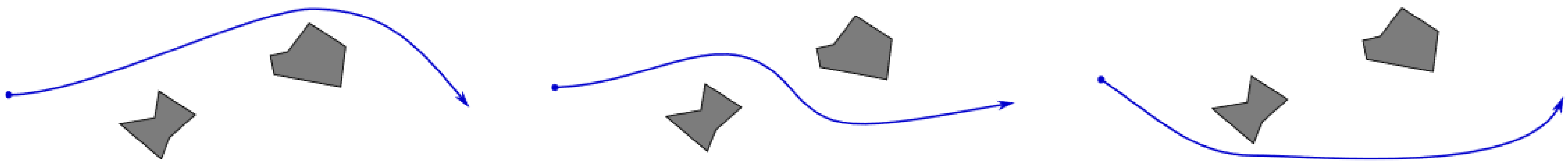
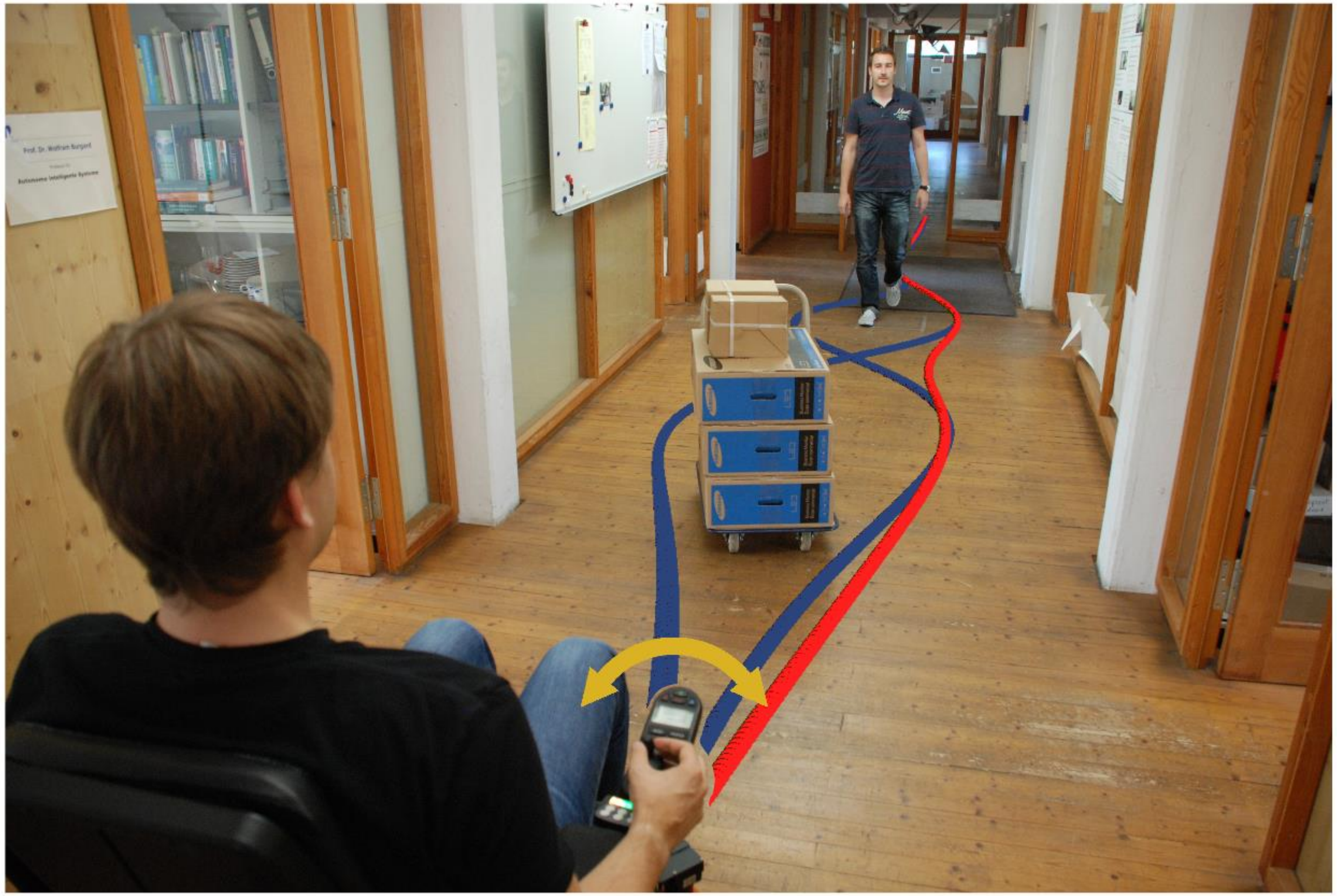


Figure 4.1: Example of three homotopically distinct composite trajectories in the same situation. The agent depicted in blue passes the solid obstacles on different sides. With fixed start- and end points these trajectories cannot be continuously transformed into each other without colliding with obstacles.

Can specify whether we pass on the "left" or "right" of each obstacle on our way to the goal.

GVG: advantages



How a curve winds around an obstacle

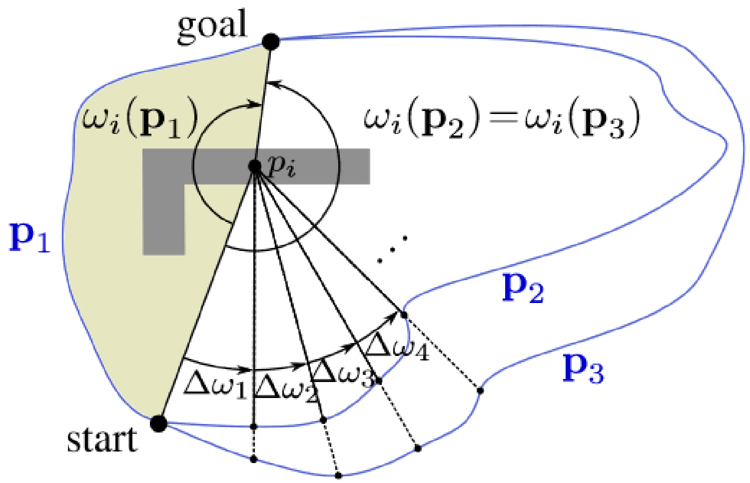


Figure 4.7: Computation of the winding numbers with respect to obstacles. The figure shows three paths $\mathbf{p}_1, \mathbf{p}_2$ and $\mathbf{p}_3$ that bypass the obstacle i. The infinitesimal angles $\Delta\omega$ sum up to the winding number ω_i around the representative point p_i of the obstacle. The two paths on the right yield the same winding number $\omega_i(\mathbf{p}_2) = \omega_i(\mathbf{p}_3)$ in contrast to the path on the left.

How a curve winds around an obstacle

Note: winding angle of a path can be more than 360 degrees

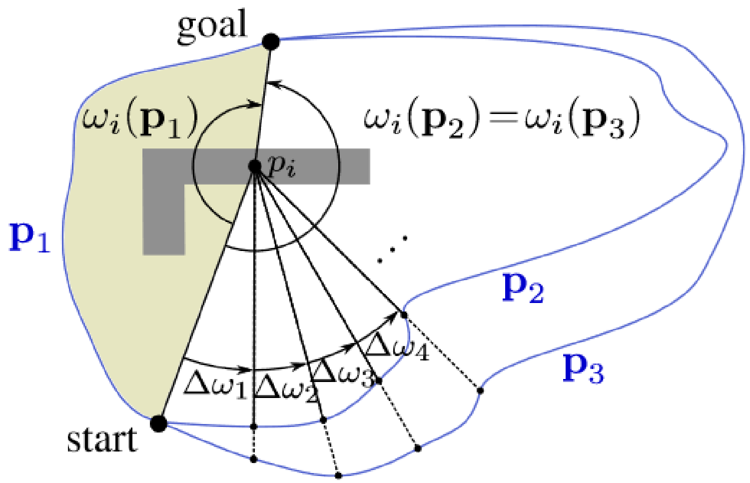


Figure 4.7: Computation of the winding numbers with respect to obstacles. The figure shows three paths $\mathbf{p}_1, \mathbf{p}_2$ and $\mathbf{p}_3$ that bypass the obstacle i. The infinitesimal angles $\Delta\omega$ sum up to the winding number ω_i around the representative point p_i of the obstacle. The two paths on the right yield the same winding number $\omega_i(\mathbf{p}_2) = \omega_i(\mathbf{p}_3)$ in contrast to the path on the left.

Homotopy classes

Two paths with the same endpoints are homotopic or belong to the same homotopy class iff one can be deformed continuously (without hitting obstacles) into the other. Formally, the paths:

$$\tau_1 : [0,T] \to \mathbb{R}^2$$
$$\tau_2 : [0,T] \to \mathbb{R}^2$$

with

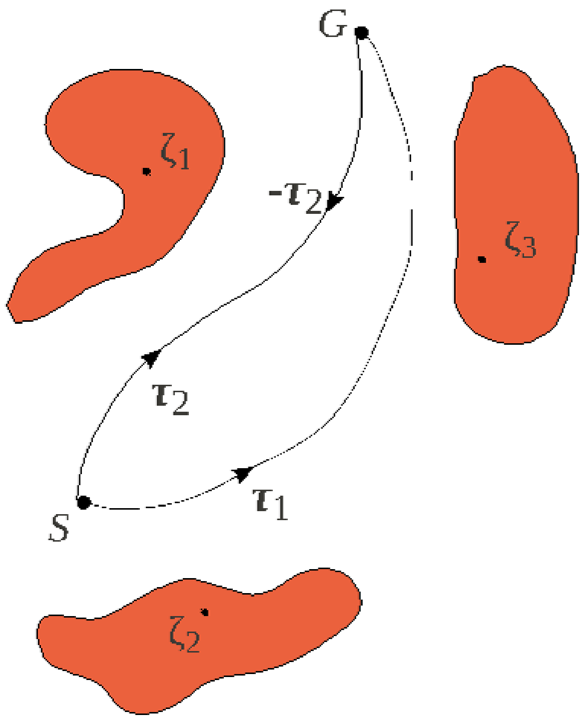
$$\tau_1(0) = \tau_2(0)$$
$$\tau_1(T) = \tau_2(T)$$

are homotopic iff there exists a continuous function

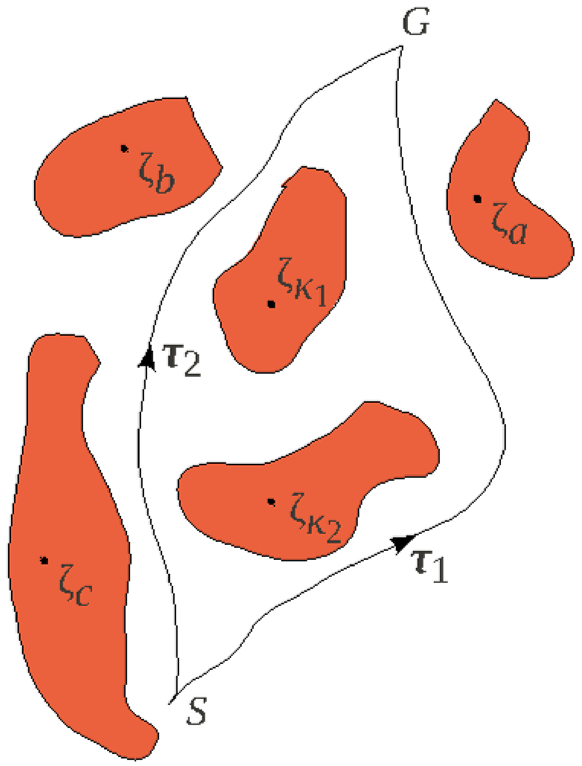
$$H : [0,1] \times [0,T] \to \mathbb{R}^2$$

such that for any time t:

$$H(0,t) = \tau_1(t)$$
$$H(1,t) = \tau_2(t)$$



(a) In same Homotopy class, forming a closed contour.



(b) In different Homotopy classes, enclosing obstacles.

Homotopy functions for deformations

Two paths with the same endpoints are homotopic or belong to the same homotopy class iff one can be deformed continuously (without hitting obstacles) into the other. Formally, the paths:

$$\tau_1 : [0,T] \to \mathbb{R}^2$$
$$\tau_2 : [0,T] \to \mathbb{R}^2$$

with

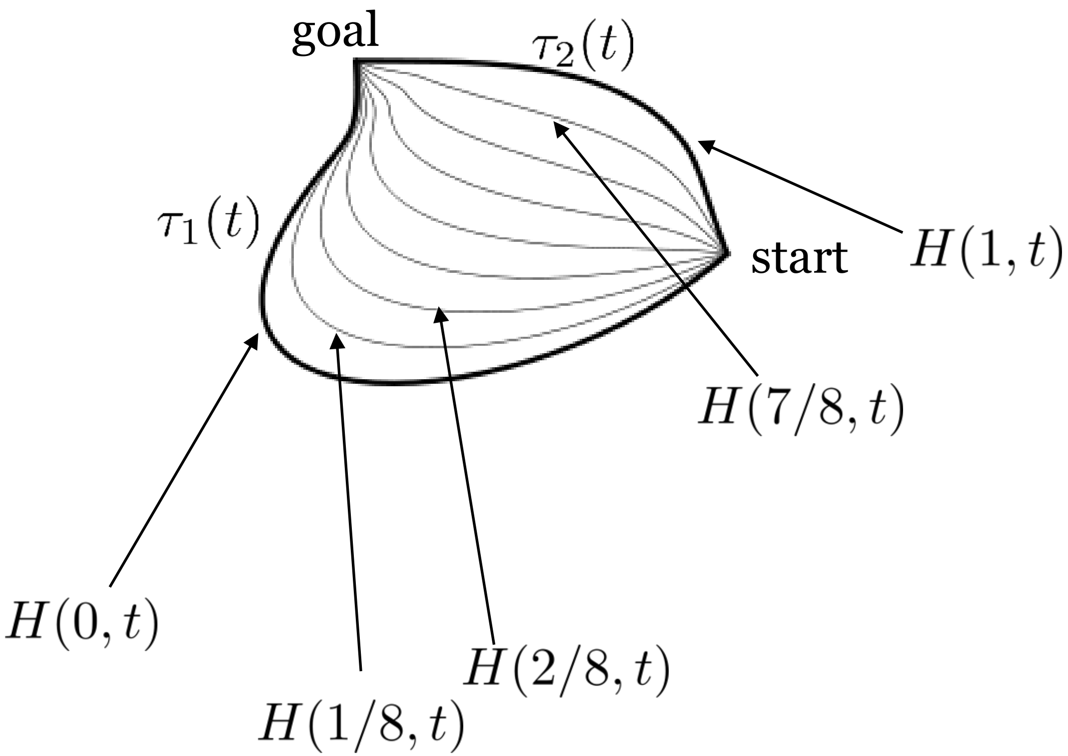
$$\tau_1(0) = \tau_2(0)$$
$$\tau_1(T) = \tau_2(T)$$

are homotopic iff there exists a continuous function

$$H : [0,1] \times [0,T] \to \mathbb{R}^2$$

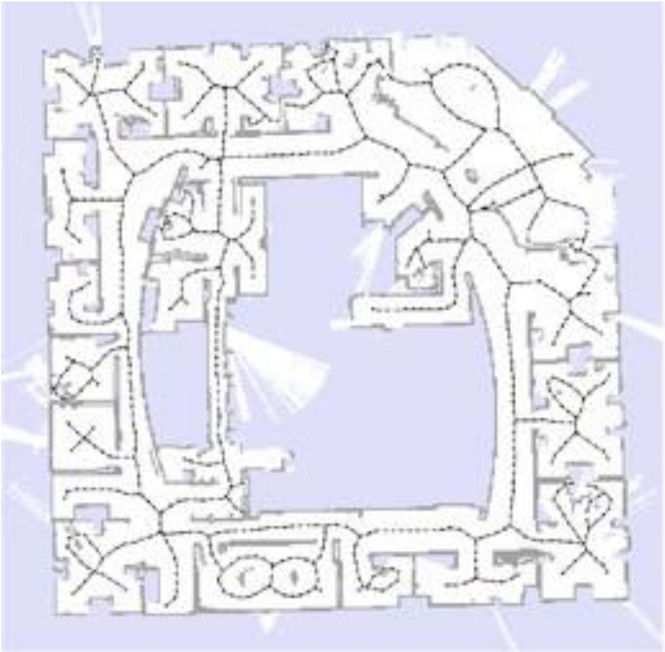
such that for any time t:

$$H(0,t) = \tau_1(t)$$
$$H(1,t) = \tau_2(t)$$

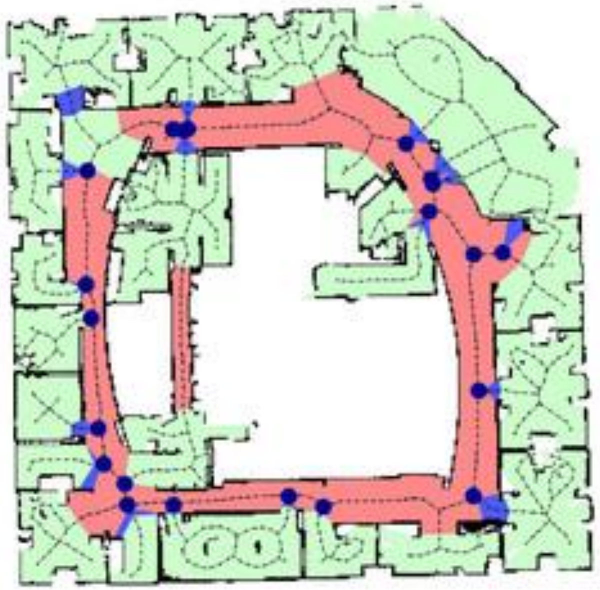


Topometric Maps

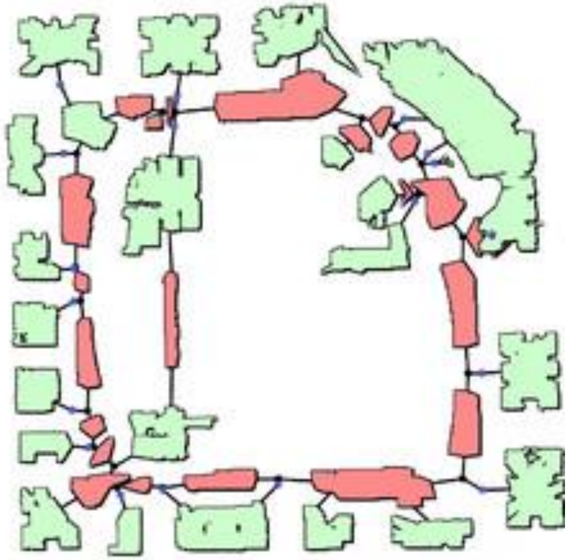
Topometric maps



Occupancy grid



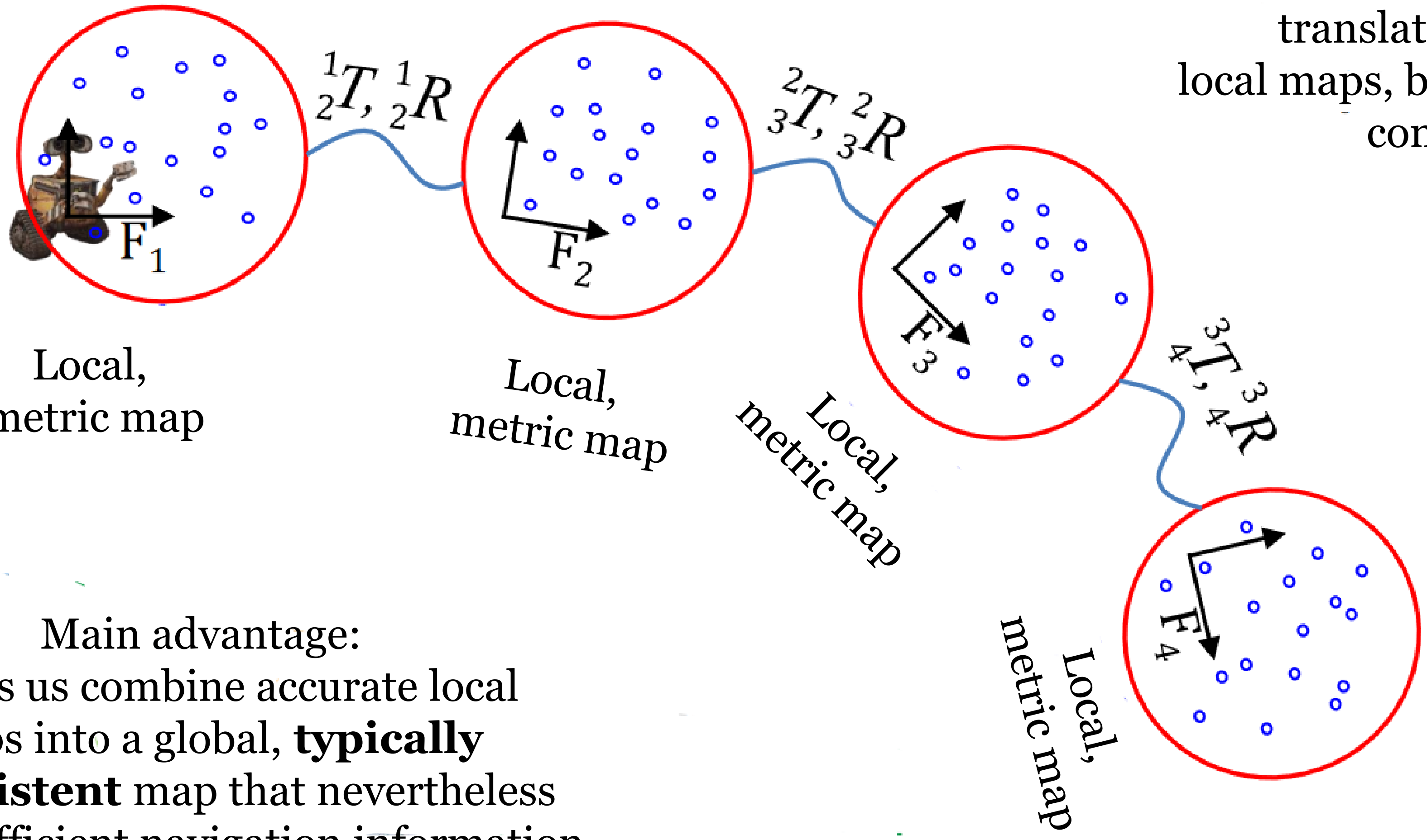
Topological map



Topometric map

Topometric maps

Edges: rotations and translations between local maps, but also topological connectivity



Main advantage:
allows us combine accurate local maps into a global, **typically inconsistent** map that nevertheless provides sufficient navigation information.

Maps of Raw Observations

Main Idea

- Map = entire (unprocessed) sequence of observations, e.g. video.
- Do not try to support distance, collision, and raytracing queries.
- Instead, provide only a similarity/nearest neighbors query
  - “Find the image in the video that is most similar to the one I’m seeing now.”
- History of observations determines a (set of) location(s) in the map

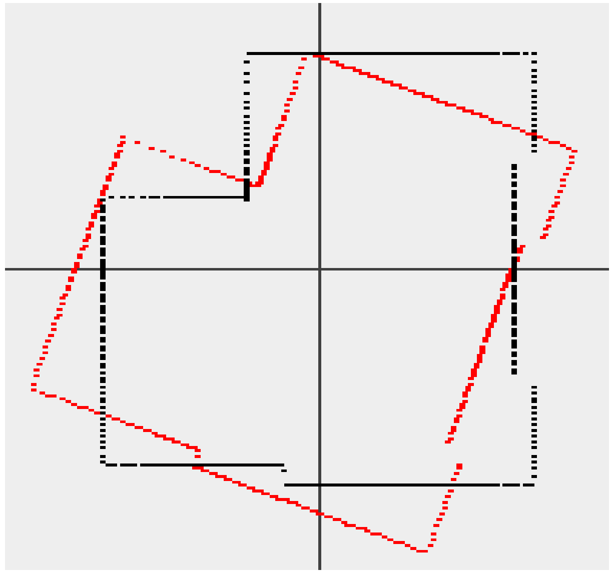
Metric Map Alignment

a.k.a. scan matching, a.k.a. iterative closest point (ICP), a.k.a. registration

Problem definition

- Given
  - two pointclouds or
  - a (local) laser scan and a pointcloud (global map) or
  - two maps

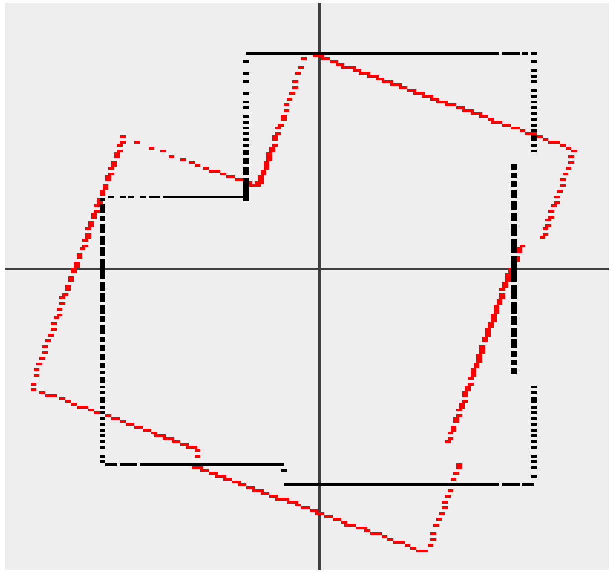
  find the rotation and translation that aligns them



Problem definition

- Given
  - two pointclouds or
  - a (local) laser scan and a pointcloud (global map) or
  - two maps

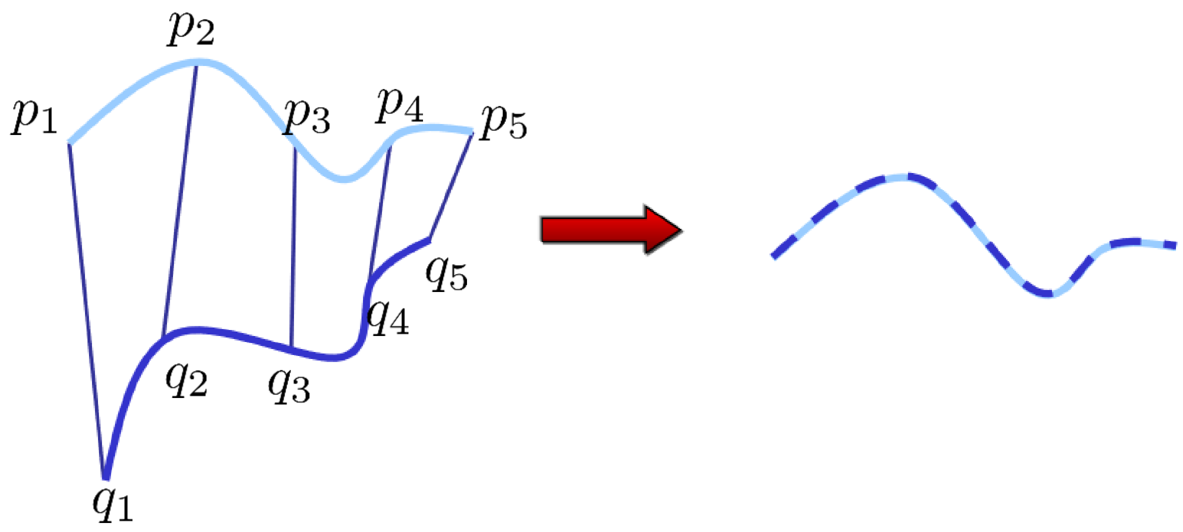
  find the rotation and translation that aligns them



- Assumption: We are assuming in these slides for simplicity that that rigid-body transformations are sufficient to align the scans. They might not be. We might need to also model scaling, non-uniform stretching and other nonlinear transformations.

Scan alignment with known correspondences

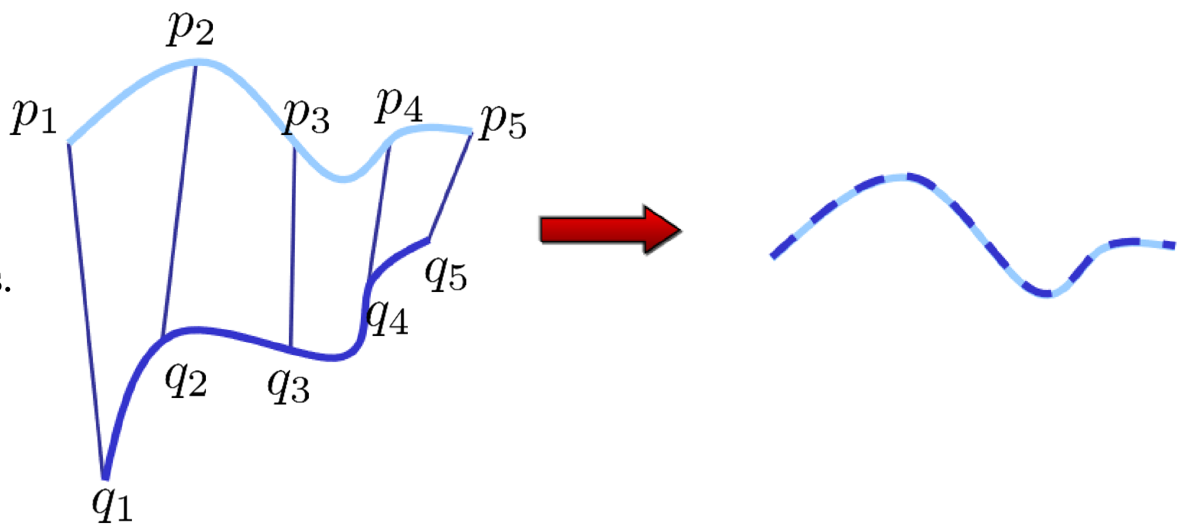
If the correct correspondences are known, the correct relative rotation/translation can be calculated in closed form.



Scan alignment with known correspondences

If the correct correspondences are known, the correct relative rotation/translation can be calculated in closed form.

When correct correspondences are known we say that **data association** is known/unambiguous.



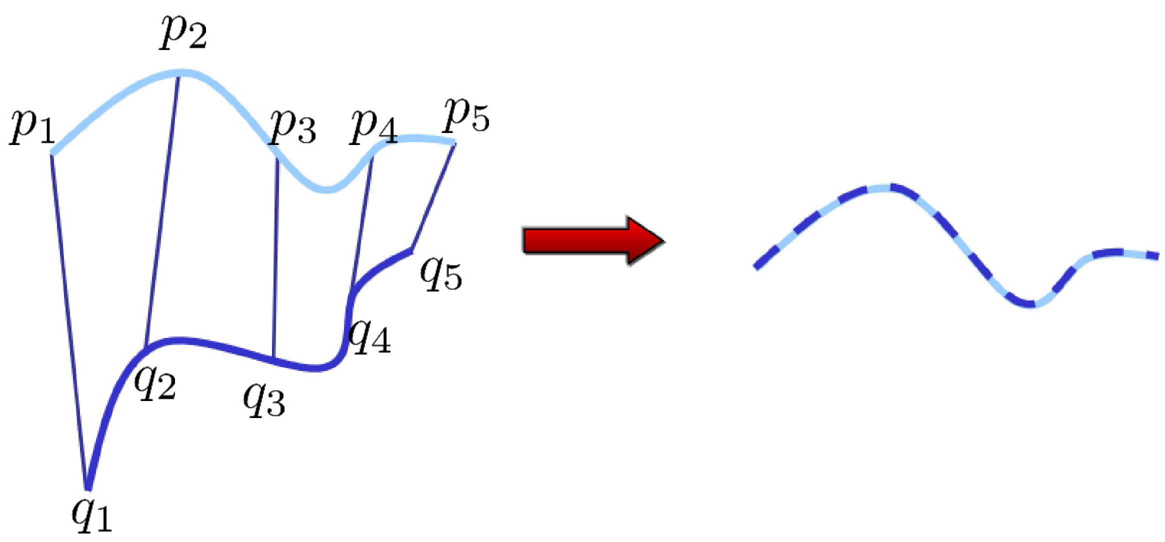
In general, data association is a real and hairy problem in robotics.

Scan alignment with known correspondences

Find the 3D rotation matrix R and the 3D translation vector t that will best align the corresponding points

$$\mathrm{error}(R,t) = \frac{1}{N}\sum_{i=1}^{N} ||p_i - (Rq_i + t)||^2$$

$$R^*, t^* = \underset{R,t}{\mathrm{argmin}}\ \mathrm{error}(R,t)$$

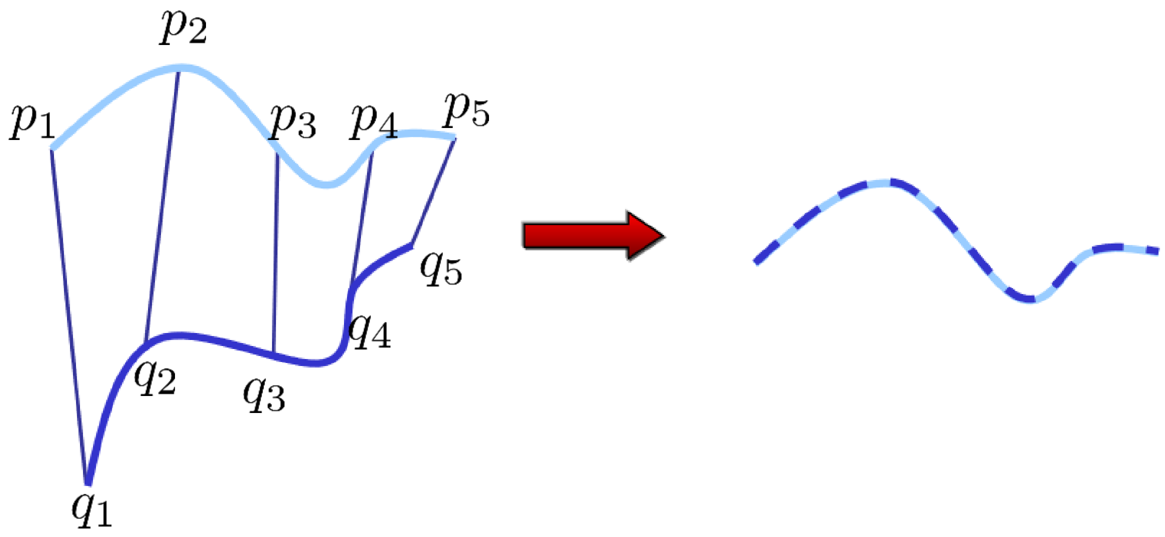


Scan alignment with known correspondences

Find the 3D rotation matrix R and the 3D translation vector t that will best align the corresponding points

$$\text{error}(R, t) = \frac{1}{N} \sum_{i=1}^{N} ||p_i - (Rq_i + t)||^2$$

$$R^*, t^* = \underset{R,t}{\text{argmin}}\ \text{error}(R, t)$$



Q: How do we minimize this error?
A: Turns out it has a closed-form solution.

NOT EXAMINABLE

Closed form solution of scan alignment with known correspondences

Find the 3D rotation matrix R and the 3D translation vector t that will best align the corresponding points

$$\text{error}(R, t) = \frac{1}{N} \sum_{i=1}^{N} ||p_i - (Rq_i + t)||^2$$

$$R^*, t^* = \underset{R,t}{\text{argmin}}\ \text{error}(R, t)$$

Step 1: compute the means of the two scans

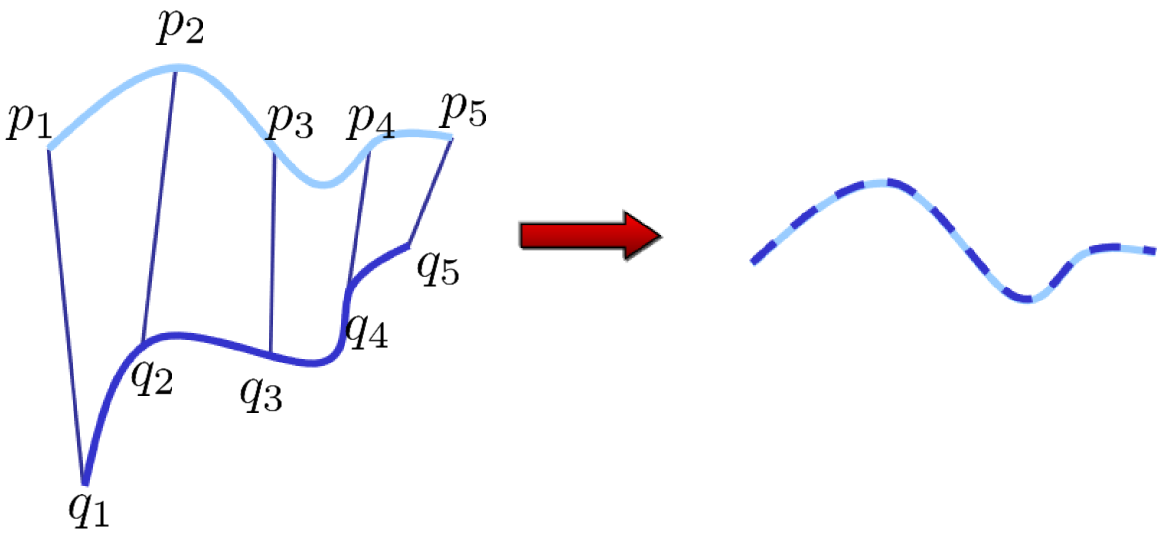
$$\mu_p = \frac{1}{N} \sum_{i=1}^{N} p_i \qquad \mu_q = \frac{1}{N} \sum_{i=1}^{N} q_i$$

Step 2: subtract the means from the scans

$$\bar{p}_i = p_i - \mu_p \qquad \bar{q}_i = q_i - \mu_q$$

Step 3: form the matrix

$$W = \sum_{i=1}^{N} \bar{p}\bar{q}^T$$



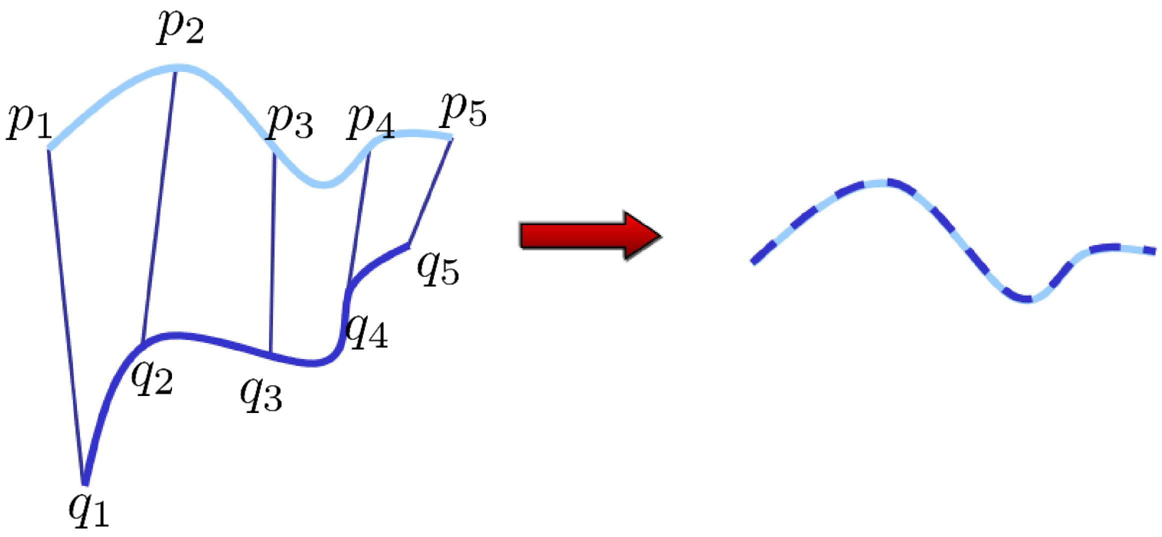
NOT EXAMINABLE

Closed form solution of scan alignment with known correspondences

Find the 3D rotation matrix R and the 3D translation vector t that will best align the corresponding points

$$\text{error}(R,t) = \frac{1}{N}\sum_{i=1}^{N} ||p_i - (Rq_i + t)||^2$$

$$R^*, t^* = \underset{R,t}{\text{argmin}}\ \text{error}(R,t)$$



Step 1: compute the means of the two scans

$$\mu_p = \frac{1}{N}\sum_{i=1}^{N} p_i \qquad \mu_q = \frac{1}{N}\sum_{i=1}^{N} q_i$$

Step 2: subtract the means from the scans

$$\bar{p}_i = p_i - \mu_p \qquad \bar{q}_i = q_i - \mu_q$$

Step 3: form the matrix

$$W = \sum_{i=1}^{N} \bar{p}\bar{q}^T$$

Step 4: compute the SVD of the matrix W

$$W = U\Sigma V^T$$

Step 5: if rank(W)=3, optimal solution is unique:

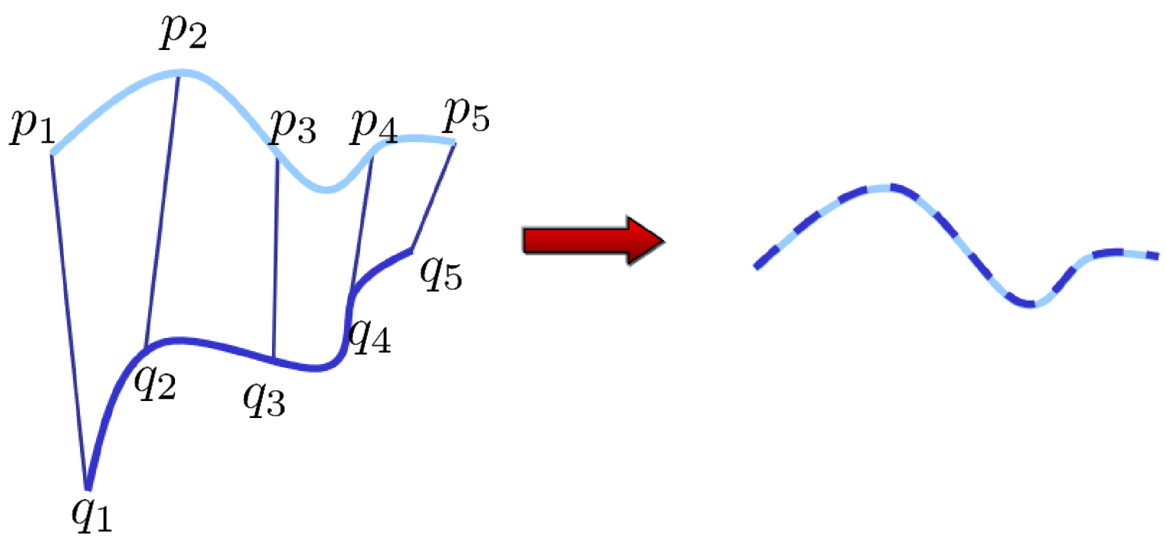
$$R^* = UV^T \qquad t^* = \mu_p - R^*\mu_q$$

Closed form solution of scan alignment with known correspondences

Find the 3D rotation matrix R and the 3D translation vector t that will best align the corresponding points

$$\mathrm{error}(R,t) = \frac{1}{N}\sum_{i=1}^{N} ||p_i - (Rq_i + t)||^2$$

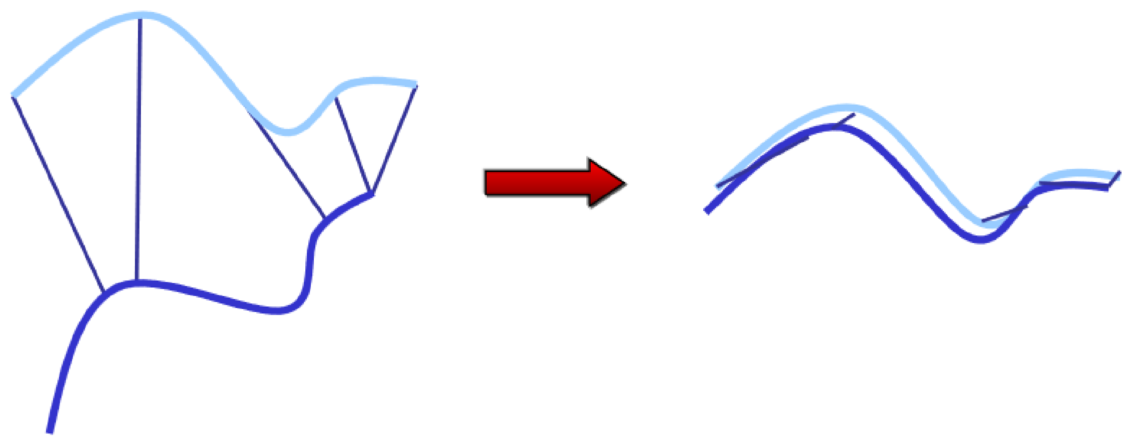
$$R^*, t^* = \underset{R,t}{\mathrm{argmin}}\ \mathrm{error}(R,t)$$



If you're interested, the proof of the closed-form solution can be found in:
K. S. Arun, T. S. Huang, and S. D. Blostein. Least square fitting of two 3-d point sets. IEEE Transactions on Pattern Analysis and Machine Intelligence, 9(5):698 – 700, 1987

Scan alignment with unknown correspondences

- If correct correspondences are not known, it is generally impossible to determine the optimal relative rotation/ translation in one step



Scan alignment with unknown correspondences

ICP-Algorithm

- Idea: iterate to find alignment
- Iterated Closest Points (ICP) [Besl & McKay 92]
- Converges if starting positions are "close enough"

Main idea for data association:

- associate each point in the source scan to its nearest neighbor in the target scan

Find optimal rotation and translation for this correspondence.

Repeat until no significant drop in error.

libpointmatcher

STUDY CASE: **Mapping a Campus**

Velodyne HDL-64e

François Pomerleau
Tyler Daoust
Tim Barfoot

Feb 9th, 2016