Self-Organizing Maps in Path Planning Tasks of Mobile Robotics

Machine Learning and Modelling Seminar

Jan Faigl

Agent Technology Center (ATG)

Department of Computer Science and Engineering

Czech Technical University in Prague

14/03/2013

Overview

1. Problem Motivation
2. Self-Organizing Maps for the Traveling Salesman Problem
3. Self-Organizing Maps for the Multi-Goal Path Planning
4. Unified SOM for 2D Multi-Goal Path Planning Problems
5. Multi-Goal Path Planning with Localization Uncertainty
6. Recent, Ongoing and Future Work
7. Concluding Remarks

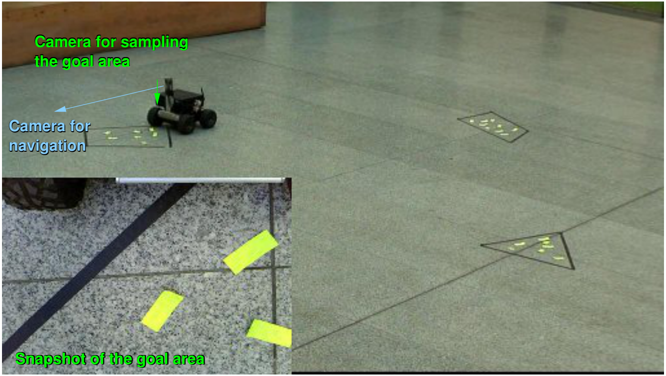
Part I

Problem Motivation

Multi-Goal Path Planning Problem Motivation

Inspection, surveillance or environment monitoring missions.

E.g., Visit goal regions to take a sample measurement at each goal



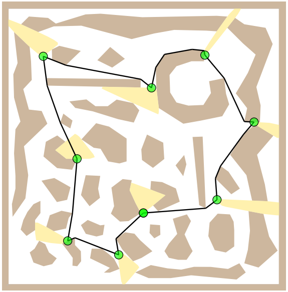
Multi-Goal Path Planning Problem Motivation

Inspection, surveillance or environment monitoring missions.

E.g., Visit goal regions to take a sample measurement at each goal

Problem Specification:

- A map of the environment
- A set of goals
- A shortest path visiting all requested goals
- Sensing and motion constraints
- Autonomous navigation capabilities



Inspection Planning

Find paths to "see" the whole environment (the Polygonal Domain $\mathcal{W}$) as quickly as possible.

Combination of sensing and motion costs

- Discrete sensing - (decoupled approach)
  1. Sensor Placement

     Art Gallery Problem (AGP) with d-visibility
  2. Multi-Goal Path Planning Problem

     Traveling Salesman Problem (TSP) in $\mathcal{W}$
- Continuous sensing
  - Watchman Route Problem (WRP)

    Goals are not explicitly prescribed
- Additional constraints

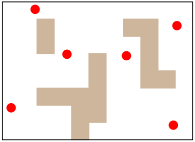
  limited sensing, motion constraints, etc

Jan Faigl,
Multi-Goal Path Planning for Cooperative Sensing,
Ph.D. thesis, CTU in Prague (2010).

Inspection Planning

Find paths to "see" the whole environment (the Polygonal Domain $\mathcal{W}$) as quickly as possible.

Combination of sensing and motion costs



- Discrete sensing - (decoupled approach)
  1. Sensor Placement

     Art Gallery Problem (AGP) with d-visibility
  2. Multi-Goal Path Planning Problem

     Traveling Salesman Problem (TSP) in $\mathcal{W}$
- Continuous sensing
  - Watchman Route Problem (WRP)

    Goals are not explicitly prescribed
- Additional constraints

  limited sensing, motion constraints, etc

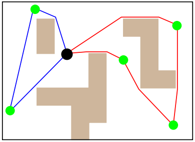
Jan Faigl, *Multi-Goal Path Planning for Cooperative Sensing*, Ph.D. thesis, CTU in Prague (2010).

Inspection Planning

Find paths to "see" the whole environment (the Polygonal Domain $\mathcal{W}$) as quickly as possible.

Combination of sensing and motion costs

- Discrete sensing - (decoupled approach)
  1. Sensor Placement
     Art Gallery Problem (AGP) with d-visibility
  2. Multi-Goal Path Planning Problem
     Traveling Salesman Problem (TSP) in $\mathcal{W}$
- Continuous sensing
  - Watchman Route Problem (WRP)
    Goals are not explicitly prescribed
- Additional constraints
  limited sensing, motion constraints, etc



Jan Faigl,
Multi-Goal Path Planning for Cooperative Sensing,
Ph.D. thesis, CTU in Prague (2010).

Inspection Planning

Find paths to "see" the whole environment (the Polygonal Domain $\mathcal{W}$) as quickly as possible.

Combination of sensing and motion costs

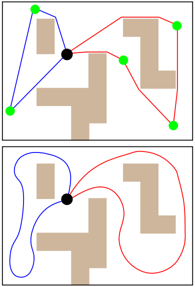
- Discrete sensing - (decoupled approach)
  1. Sensor Placement

     Art Gallery Problem (AGP) with d-visibility
  2. Multi-Goal Path Planning Problem

     Traveling Salesman Problem (TSP) in $\mathcal{W}$
- Continuous sensing
  - Watchman Route Problem (WRP)

    Goals are not explicitly prescribed
- Additional constraints

  limited sensing, motion constraints, etc



Jan Faigl,
Multi-Goal Path Planning for Cooperative Sensing,
Ph.D. thesis, CTU in Prague (2010).

Multi-Goal Path Planning Problem as the Traveling Salesman Problem

- Given a set of goals, the problem is to find a sequence of goals' visits.
- Having the paths between goals, the problem can be formulated as the traveling salesman problem.

Traveling Salesman Problem (TSP):

Given a list of cities (goals) and the distances between each pair of cities (path lengths), what is the shortest possible route that visits each city exactly once and returns to the origin city (depot).

http://www.tsp.gatech.edu

http://www.tsp.gatech.edu/history/travelling.html

- Most common problem representations:
  - Euclidean TSP – cities are 2D points in a plane.
  - TSP on a graph – cities are vertices of the graph.

Multi-Goal Path Planning Problem as the Traveling Salesman Problem

- Given a set of goals, the problem is to find a sequence of goals' visits.
- Having the paths between goals, the problem can be formulated as the traveling salesman problem.

Traveling Salesman Problem (TSP):

Given a list of cities (goals) and the distances between each pair of cities (path lengths), what is the shortest possible route that visits each city exactly once and returns to the origin city (depot).

http://www.tsp.gatech.edu

http://www.tsp.gatech.edu/history/travelling.html

- Most common problem representations:
  - Euclidean TSP – cities are 2D points in a plane.
  - TSP on a graph – cities are vertices of the graph.

Multi-Goal Path Planning for Multi-Robot Team

For a group of mobile robots, the problem becomes the *Multiple Traveling Salesman Problem* (MTSP).

Optimization criteria:

- **MinSum** – For minimization of the total sum cost the problem can be transformed to the TSP.

M. Bellmore M. and S. Hong (1974)

- **MinMax** – A variant with minimizing the maximal cost of a tour must be solved directly.
  - A more suitable for minimizing the time to visit all goals.
  - MTSP→TSP provides degenerative solutions.
  - The first attempt to solve the MTSP-MinMax was in 1995

P. M. França, M. Gendreau, G. Laporte, F. M. Müller, *The m-Traveling Salesman Problem with Minmax Objective*, Transportation Science, 29(3):267–275 (1995).

The approach is based on a Distance Constrained VRP, where a solution of the MTSP is used as a constraint.

Multi-Goal Path Planning for Multi-Robot Team

For a group of mobile robots, the problem becomes the *Multiple Traveling Salesman Problem* (MTSP).

Optimization criteria:

- **MinSum** – For minimization of the total sum cost the problem can be transformed to the TSP.

  M. Bellmore M. and S. Hong (1974)

- **MinMax** – A variant with minimizing the maximal cost of a tour must be solved directly.
  - A more suitable for minimizing the time to visit all goals.
  - MTSP→TSP provides degenerative solutions.
  - The first attempt to solve the MTSP-MinMax was in 1995

P. M. França, M. Gendreau, G. Laporte, F. M. Müller, *The m-Traveling Salesman Problem with Minmax Objective*, Transportation Science, 29(3):267–275 (1995).

The approach is based on a Distance Constrained VRP, where a solution of the MTSP is used as a constraint.

Problem Variants

- Robots start from different locations (multi-depot).
- A path does not necessary by closed (planning an open path without returning to the depot).
- Planning for a heterogenous robotic team (robots can have different capabilities for traversing environment).

- *Considering other constraints arising from robotics:*
  - *kinematic*
  - *sensing*
  - *operational*
  - *...*

Problem Variants

- Robots start from different locations (multi-depot).
- A path does not necessary by closed (planning an open path without returning to the depot).
- Planning for a heterogenous robotic team (robots can have different capabilities for traversing environment).

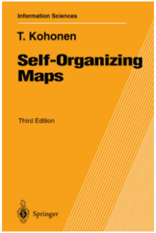
- *Considering other constraints arising from robotics:*
  - *kinematic*
  - *sensing*
  - *operational*
  - ...

Part II

Self-Organizing Maps for the Traveling Salesman Problem

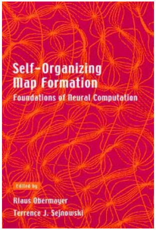
Self-Organizing Maps – Literature

- *Self-Organizing Maps, Third Edition*, Kohonen, T. and Schroeder, M. R. and Huang, T. S., Springer-Verlag New York, Inc. 2001.



http://www.cis.hut.fi/research/som-research

- *Self-Organizing Map Formation: Foundations of Neural Computation*, Edited by K. Obermayer and T. J. Sejnowski, The MIT Press, 2001.



Self-Organizing Maps for the TSP

- First approaches proposed in 1988.

  B. Angéniol et al., Neural Networks, 1(4):289–293 (1988).

  C. Fort, Biological Cybernetics, 59(1):33–40 (1988).

- In general, performance of SOM for the TSP can be considered poor regarding classical heuristic approaches.

  However, notice that Lin-Kernighan was proposed in 1973, while efficient implementation is from 2000 by Keld Helsgaun.

- SOM can be used as a constructing heuristic.

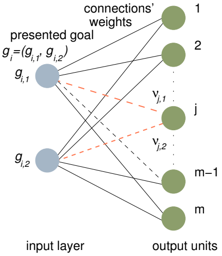
  L. I. Burke (1994)

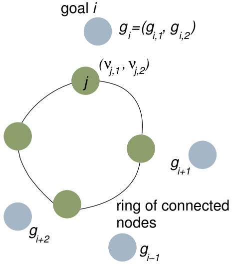
- Many variants of SOM for the TSP:
  - modification of adaptation rules
  - combinations with heuristics, genetic algorithms, memetic, or immune systems.

SOM for the TSP

- Two-layered unsupervised learning network
- Neurons' weights are nodes $\mathcal{N} = \{\nu_1, \ldots, \nu_m\})$ in a plane.

  Neurons are "fixed" and only weights are adapted.
- The output layer organizes the nodes into a ring.
- The ring evolves in the problem domain during learning.





Adaptation phases

- Network adapts to a goal g, $g \in \boldsymbol{G} = \{g_1, \dots g_n\}$ in two phases:

Goals are presented in a random order for a single epoch.

1. Winner is selected using its distance $|S(\nu, g)|$ to the goal g

$$\nu^* = \text{argmin}_{\nu \in \mathcal{N}} |S(\nu, g)|.$$

Competitive phase.

Neurons compete to be the winner, which is selected as the closest one (neuron's weights) to the goal using Euclidean distance, i.e., $|S(\nu, g)| = |(\nu, g)|$.

2. ν^* and its neighbors are updated using the learning rule

$$\nu(t+1) = \nu(t) + \mu f(\sigma, l)|S(\nu(t), g)|.$$

Cooperative phase.

μ – learning rate, $f(\sigma, l)$ – neighboring (or activation) function

Neighbouring function $f(\sigma, l)$

- The neurons cooperate in the adaptation of the weights:

$$\nu(t+1) = \nu(t) + \mu f(\sigma, l)|(\nu(t), g)|.$$

- $f(\sigma, l)$ must possess two important characteristics:
  1. It should decrease for farther neighbors
  2. Its pervasiveness should decrease during learning

$$f(\sigma, l) = \begin{cases} e^{-\frac{l^2}{\sigma^2}} & l < 0.2M \\ 0 & \textit{otherwise} \end{cases},$$

  - l is distance of ν from the winner ν^*.
  - M is the number of neurons, $M = kN$ for N goals, $k \in \langle 2, 3 \rangle$.
  - σ is called learning gain and it is decreased after each learning epoch (presentation of all goals to the network).

General adaptation schema

1. Initialization, e.g., randomize weights or create a small ring around a goal.
2. Present all goals to the network and adapt the network
3. If all winners are sufficiently close to the goals ($|(\nu, g)| \leq \delta$) stop the adaptation, otherwise go to Step 2.

Alternatively, stop adaptation after n steps.

Determination of the final path

- After the adaptation, the final sequence of goals' visit is determined by traversing ring.

  Each goal should have a distinct winner.

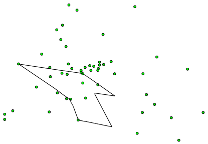
- A solution can be retrieved after each learning epoch.
- An inhibited mechanism can be used to guarantee distinct winners

  A neuron can be a winner only once in a single epoch.

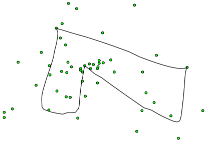
A visualization of the learning process

SOM evolution for the problem *berlin52* from the TSPLIB.

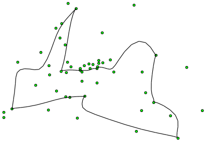
G. Reinelt, TSPLIB - A Traveling Salesman Problem Library (1991).



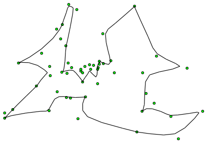
step 27



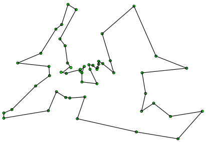
step 36



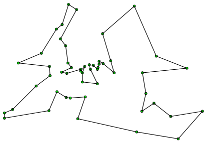
step 45



step 51



step 63



step 72

SOM algorithm for the TSP

- Selected algorithm providing relatively good solutions.

  Samerkae Somhom, Abdolhamid Modares, Takao Enkawa, *A self-organising model for the travelling salesman problem*, Journal of the Operational Research Society, 919–928 (1997).

Input: $\boldsymbol{G} = \{g_1, \ldots, g_n\}$ - given set of goals
Input: δ - maximal allowable error
Output: $(\nu_1, \ldots, \nu_M)$ - a sequence of neurons' weights representing the goal tour.

```
N ← initialization(ν_1, ..., ν_m)          // set weights to form a ring around g_1
σ ← σ_0                          // set the initial value of the learning gain
repeat
    I ← ∅                                              // clear inhibited neurons
    error ← 0                                           // set the maximal error
    Π ← create a random permutation of the goals
    foreach g ∈ Π(G) do
        ν* ← argmin_{ν∈N, ν∉I} |(g, ν)|      // the closest non-inhibited ν to g
        error ← max{error, |(g, ν*)|}                          // update error
        foreach ν_i in l neighborhood of ν* do
            ν_i ← ν_i + μf(σ, l)|(g, ν_i)|       // adapt winner and its neighbors
    σ ← (1 − α) · σ                              // decrease the learning gain
until error ≤ δ
```

Parameters of the adaption

Parameters providing good results in practice.

Stability of the convergence and quality of solution.

N		the number of goals ($N = \|\boldsymbol{G}\|$)	
M	=	$2.5N$	the number of neurons
σ_0	=	$12.41N + 0.06$	the initial learning gain *σ_0 depends of problem size (regarding quality of solution). This is a linear regression model that has been found experimentally.*
μ	=	0.6	the learning rate *I.e, move ν towards g about 60% of their distance at maximum.*
α	=	0.1	the gain decreasing rate *The gain is decreased after each epoch,* $\sigma = (1-\alpha)\sigma$
δ	=	0.001	the minimal required distance of the winner from the goal

S. Somhom et al. (1997)

Selected Modifications for the Euclidean TSP

- Dynamical creation of neurons (duplication/deletion)

B. Angéniol et al. (1988)

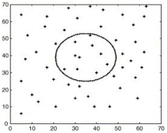
- Reducing topological defects using multiple scale neighborhood functions (β_j, γ_j parameters)

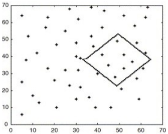
$$f(\sigma, l) = \beta_j \mu e^{-(l/(\gamma_j \sigma))^2},$$

where $\beta_j \in \{0.25, 0.5, 1, 0.5, 0.25, 0.125\}$ *and* $\gamma_j \in \{0.25, 0.5, 1, 2, 4, 8\}$

K. Murakoshi and Y. Sato (2006)

- Initialization of the network (rhombic frame)





W. D. Zhang et al. (2006)

Considering geometrical properties of the ring and topology of the cities

- Kohonen Network Incorporating Explicit Statistics (KNIES)
  - The winner neuron and its neighboring neurons are adapted towards the presented goal
  - Other neurons are dispersed to keep properties unchanged (the mean of neurons coincides with the mean of the cities).

N. Aras et al. (1999)

- Considering distance to the segment joining two neurons (points)

A. Plebe (2002)

- Convex-hull expanding property

K.-S. Leung et al. (2004)

H. Yang and H. Yang (2005)

Variants of Adaptation Parameters

Let k be the current number of the learning epoch.

- $\nu' = \nu + \mu f(\sigma, l)|(\nu(t), g)|$, $\sigma_k = (1-\alpha)\sigma_{k-1}$
- $\mu_k = \mu_0 e^{-\frac{k}{\tau_1}}$, $\sigma_k = \sigma_0 e^{-\frac{k}{\tau_2}}$

Kohonen's exponential evolution of the paramters for a better convergence.

- Decreasing the learning rate ($\alpha = 0.998\alpha$)

A. Zhu and S. X. Yang (2003)

Wendong Zhang and Yanping Bai and Hong Ping Hu,
The incorporation of an efficient initialization method and parameter adaptation using self-organizing maps to solve the TSP,
Applied Mathematics and Computation, 172(1):603–623 (2006).

- $\mu_k = \frac{1}{\sqrt[4]{k}}$
- $\sigma_k = (1 - 0.01k)\sigma_{k-1}$
- $\sigma_0 = 10$

Variants of Adaptation Parameters

Let k be the current number of the learning epoch.

- $\nu' = \nu + \mu f(\sigma, l)|(\nu(t), g)|$, $\sigma_k = (1-\alpha)\sigma_{k-1}$
- $\mu_k = \mu_0 e^{-\frac{k}{\tau_1}}$, $\sigma_k = \sigma_0 e^{-\frac{k}{\tau_2}}$

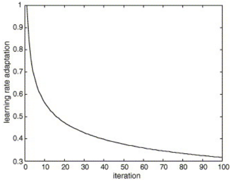
Kohonen's exponential evolution of the paramters for a better convergence.

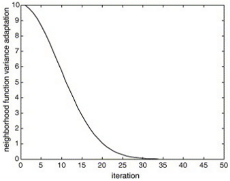
- Decreasing the learning rate ($\alpha = 0.998\alpha$)

A. Zhu and S. X. Yang (2003)

Wendong Zhang and Yanping Bai and Hong Ping Hu,
The incorporation of an efficient initialization method and parameter adaptation using self-organizing maps to solve the TSP,
Applied Mathematics and Computation, 172(1):603–623 (2006).

- $\mu_k = \frac{1}{\sqrt[4]{k}}$
- $\sigma_k = (1 - 0.01k)\sigma_{k-1}$
- $\sigma_0 = 10$





Co-Adaptive Net

E. M. Cochrane and J. E. Beasley,
The co-adaptive neural network approach to the Euclidean travelling salesman problem,
Neural Networks, 16(10):1499–1525 (2003).

It includes a comprehensive overview of previous approaches.

- One of the most complex SOM for the TSP
- A stronger co-operation between neurons
  Adaptation of neighbors without moving the winner.
- Neuron-specific gain
  $\sigma_j = \sigma(1 - |(\nu_j, g)|/\sqrt(2))$
- Adaptive neuron neighborhood
  The size of the activation bubble is changed and it also depends on σ.
- Near-tour to tour construction
  An alternative for the inhibition mechanism.
  Keeping the best found tour during learning.

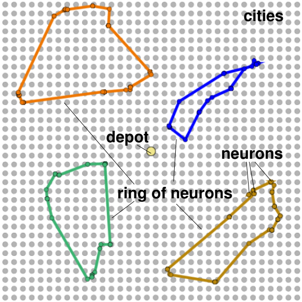
SOM for the MTSP-MinMax

Samerkae Somhom and Abdolhamid Modares and Takao Enkawa, *Competition-based neural network for the multiple travelling salesmen problem with minmax objective*, Computers and Operations Research, 26(4):395–407 (1999).

- A ring for each salesman

  $M = 2.5N/k$, *k no. of salesmen*

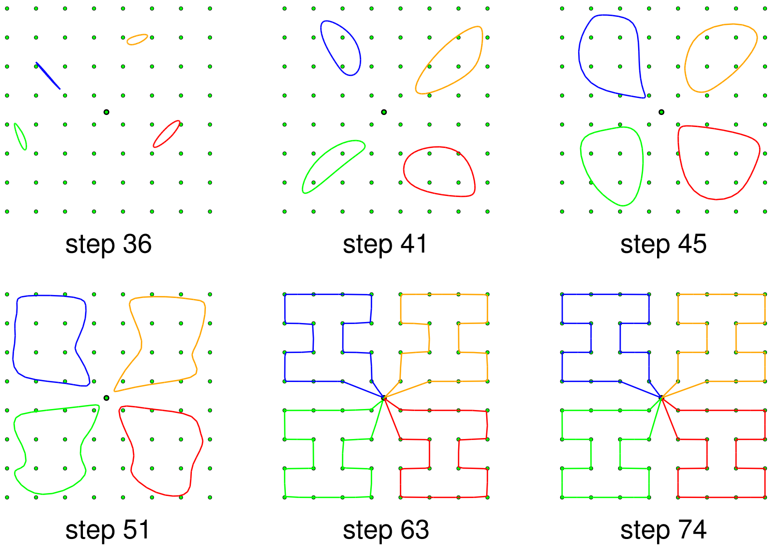
- Common depot
- A winner from each ring is adapted towards depot.
- Then, nodes are adapted towards other goals.
- MinMax criterion



$$\nu^\star = \text{argmin}_\nu \, |S(\nu, g)| \cdot \left(1 + \frac{dist_\nu - avg}{avg}\right), \qquad (1)$$

where $dist_\nu$ is length of the ring in which ν is, and avg is the average length of the rings.

MTSP-MinMax visualization of SOM evolution



Part III

Self-Organizing Maps for the Multi-Goal Path Planning

Multi-Goal Path Planning Problem

Find shortest path connecting given set of goals.

Specification:

- Input:
  - A map of the environment

    The polygonal domain $\mathcal{W}$
  - A set of goals
- Output:
  - A shortest path visiting all requested goals

- Paths connecting obstacles must respect obstacles.

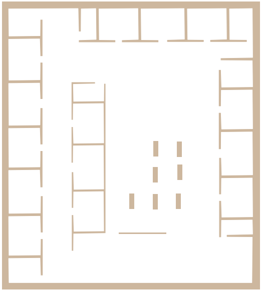
Additional motion constraints can be considered.

Multi-Goal Path Planning Problem

Find shortest path connecting given set of goals.

Specification:

- Input:
  - A map of the environment
    The polygonal domain $\mathcal{W}$
  - A set of goals
- Output:
  - A shortest path visiting all requested goals



- Paths connecting obstacles must respect obstacles.

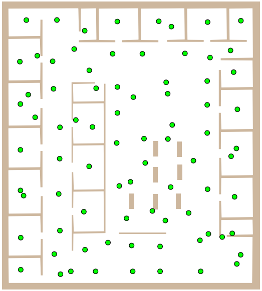
Additional motion constraints can be considered.

Multi-Goal Path Planning Problem

Find shortest path connecting given set of goals.

Specification:

- Input:
  - A map of the environment
    The polygonal domain $\mathcal{W}$
  - A set of goals
- Output:
  - A shortest path visiting all requested goals



- Paths connecting obstacles must respect obstacles.

Additional motion constraints can be considered.

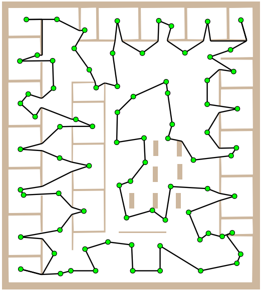
Multi-Goal Path Planning Problem

Find shortest path connecting given set of goals.

Specification:

- Input:
  - A map of the environment

    The polygonal domain $\mathcal{W}$
  - A set of goals
- Output:
  - A shortest path visiting all requested goals



- Paths connecting obstacles must respect obstacles.

  Additional motion constraints can be considered.

SOM for the Multi-Goal Path Planning Problem (MTP)

- Can be based on SOM for the TSP.
- How to compute $|S(\nu, g)|$?

A distance metric for the input vector and neuron's weights.

1. Euclidean distance

provides poor solutions

2. $|S(\nu, g)|$ has to respect obstacles
   - $S(\nu, g)$ - the shortest path among obstacles.
   - Adaptation - a movement of ν toward g along $S(\nu, g)$

Paths between nodes are only for a visualization!

SOM for the Multi-Goal Path Planning Problem (MTP)

- Can be based on SOM for the TSP.
- How to compute $|S(\nu, g)|$?

A distance metric for the input vector and neuron's weights.

1. Euclidean distance

   provides poor solutions

2. $|S(\nu, g)|$ has to respect obstacles
   - $S(\nu, g)$ - the shortest path among obstacles.
   - Adaptation - a movement of ν toward g along $S(\nu, g)$

Paths between nodes are only for a visualization!

SOM for the Multi-Goal Path Planning Problem (MTP)

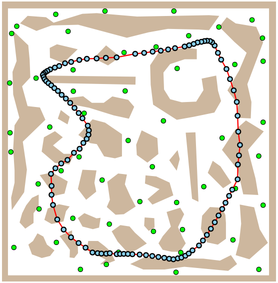
- Can be based on SOM for the TSP.
- How to compute $|S(\nu, g)|$?

A distance metric for the input vector and neuron's weights.

1. Euclidean distance

   provides poor solutions

2. $|S(\nu, g)|$ has to respect obstacles
   - $S(\nu, g)$ - the shortest path among obstacles.
   - Adaptation - a movement of ν toward g along $S(\nu, g)$



Paths between nodes are only for a visualization!

SOM for the Multi-Goal Path Planning Problem (MTP)

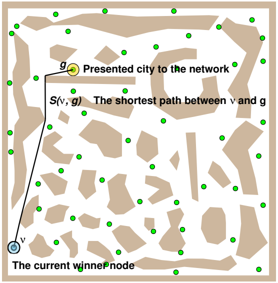
- Can be based on SOM for the TSP.
- How to compute $|S(\nu, g)|$?

A distance metric for the input vector and neuron's weights.

1. Euclidean distance

   provides poor solutions

2. $|S(\nu, g)|$ has to respect obstacles
   - $S(\nu, g)$ - the shortest path among obstacles.
   - Adaptation - a movement of ν toward g along $S(\nu, g)$



Paths between nodes are only for a visualization!

SOM for the Multi-Goal Path Planning Problem (MTP)

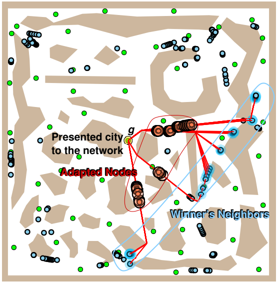
- Can be based on SOM for the TSP.
- How to compute $|S(\nu, g)|$?

A distance metric for the input vector and neuron's weights.

1. Euclidean distance

   provides poor solutions

2. $|S(\nu, g)|$ has to respect obstacles
   - $S(\nu, g)$ - the shortest path among obstacles.
   - Adaptation - a movement of ν toward g along $S(\nu, g)$



Paths between nodes are only for a visualization!

SOM for the Multi-Goal Path Planning Problem (MTP)

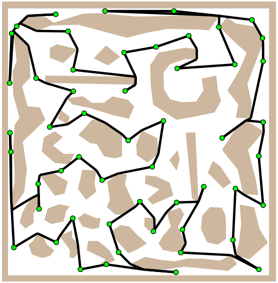
- Can be based on SOM for the TSP.
- How to compute $|S(\nu, g)|$?

A distance metric for the input vector and neuron's weights.

1. Euclidean distance

   provides poor solutions

2. $|S(\nu, g)|$ has to respect obstacles
   - $S(\nu, g)$ - the shortest path among obstacles.
   - Adaptation - a movement of ν toward g along $S(\nu, g)$

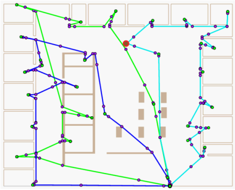


Paths between nodes are only for a visualization!

Multi-Robot Multi-Goal Path Planning

- Multiple Traveling Salesman Problem with MinMax in $\mathcal{W}$
- The MTSP-MinMax must be solved directly

MTSP→TSP provides degenerative solutions



- For the TSP we need to resolve *neuron–goal* distance and path queries.

Single point queries, goals are fixed.

- For the MTSP-MinMax we need to resolve *neuron–neuron* distance queries.

Two points queries.

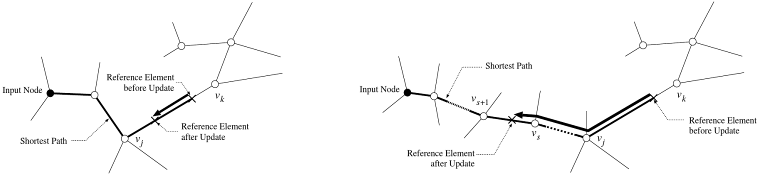
Naïve approach works, but it is too computational demanding to for practical scenarios such as search and rescue missions.

This difficulty has been noted by several authors.

SOM for a Graph Input

Takeshi Yamakawa, Keiichi Horio, Masaharu Hoshino,
Self-Organizing Map with Input Data Represented as Graph,
Neural Information Processing, Lecture Notes in Computer Science,
4232:907–914 (2006).

- Neurons movements are restricted to the graph edges.
- During adaptation neurons moved along shortest path in the graph.



It seems as a suitable solution; however, using the visibility graph of the environment and goals provides poor solutions.

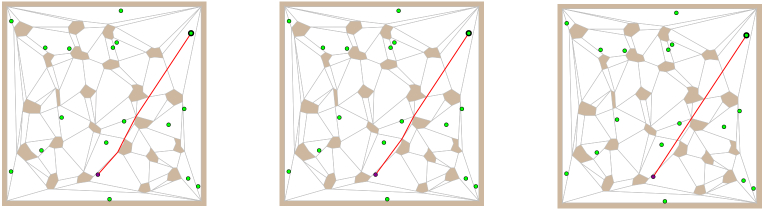
Dealing with Shortest Paths in $\mathcal{W}$

Approximate shortest path to the goal using precomputed visibility graph, convex partitioning and ray-shooting technique.

1. A node is always in some convex cell.

   All cells are formed from the map vertices.

2. A rough path goes over a map vertex.
3. A path is refined using ray-shooting technique.



path over v_0 path over v_1 full refinement

Paths are provided in units of μs.

The approximation is enabling technique for applying SOM in $\mathcal{W}$.

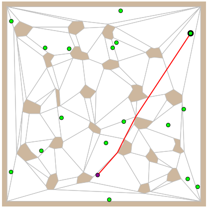
Dealing with Shortest Paths in $\mathcal{W}$

Approximate shortest path to the goal using precomputed visibility graph, convex partitioning and ray-shooting technique.

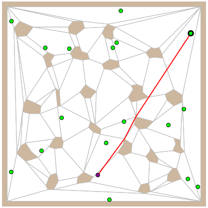
1. A node is always in some convex cell.

   All cells are formed from the map vertices.

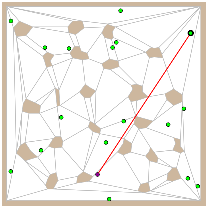
2. A rough path goes over a map vertex.
3. A path is refined using ray-shooting technique.



path over v_0



path over v_1



full refinement

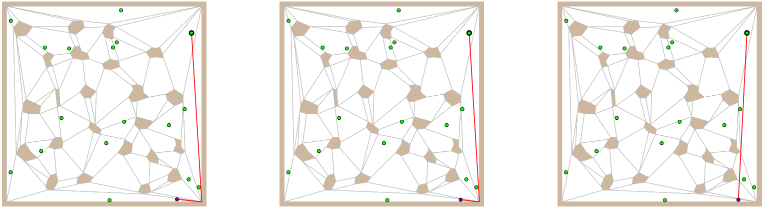
Paths are provided in units of μs.

The approximation is enabling technique for applying SOM in $\mathcal{W}$.

SOM with Approximate Shortest Path in $\mathcal{W}$

Approximate shortest path is sufficient.

A full path refinement is not necessary.

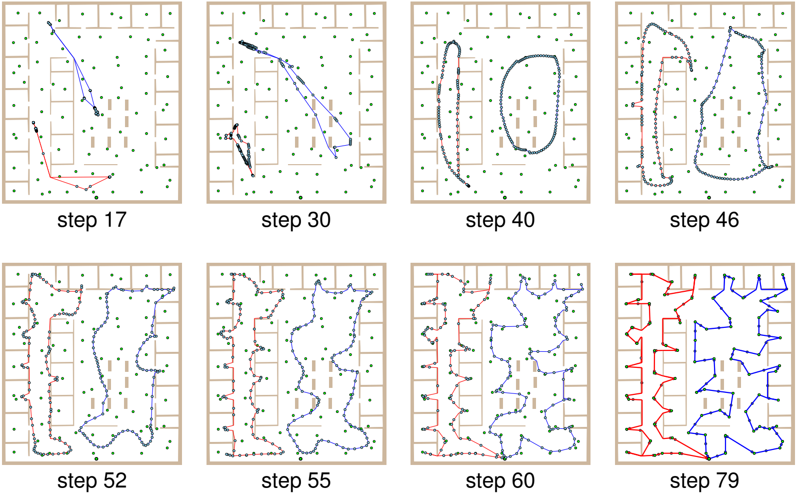


The approximation becomes a more precise as a node moves towards the goal.

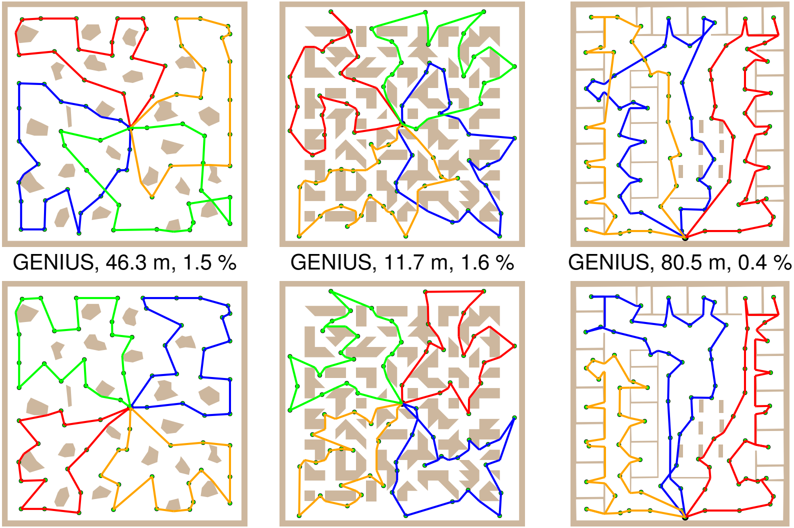
J. Faigl, M. Kulich, V. Vonásek, L. Přeučil,
An Application of Self-Organizing Map in the non-Euclidean Traveling Salesman Problem,
Neurocomputing, 74(5): 671–679 (2011).

Similar approximation also works for two-points path query.

SOM for the MTSP-MinMax in $\mathcal{W}$



Example of MTSP-MinMax Solutions



GENIUS, 46.3 m, 1.5 % — GENIUS, 11.7 m, 1.6 % — GENIUS, 80.5 m, 0.4 %

SOM, 46.1 m, 6.0 % — SOM, 11.6 m, 3.4 % — SOM, 81.7 m, 5.1 %

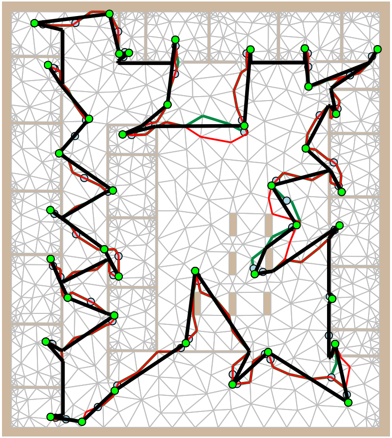
SOM provides competitive solutions, while it prefers non-crossing paths.

Graph Representation of Freespace of $\mathcal{W}$

Triangular mesh with a sufficient density.

- SOM for the graph input.

  T. Yamakawa et al. (2006)
- Nodes movement (weights' changes) are restricted to be on graph edges.
- Paths found in the graph, e.g., by Dijkstra's algorithm.
- Sequence of goals' visits is determined from the ring
- Final path is found using visibility graph of the goals.



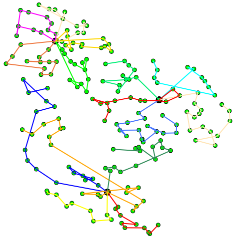
The approach is usable; however, it does not provide better solutions or lower computational requirements than the approximate shortest path.

Generalization of the Graph Based Approach

- Extended variant of the Multi-Depot MTSP-MinMax on a graph for logistic planning

- Cost is associated not only to edges, but also to vehicles
- Triangle inequality does not hold
- Vehicles can have initial added cost

E.g., a travel cost from a garage to a starting location.

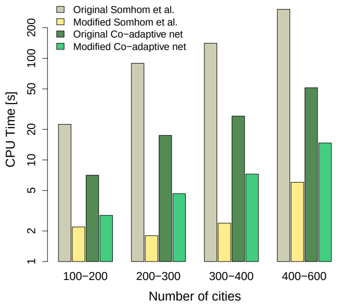


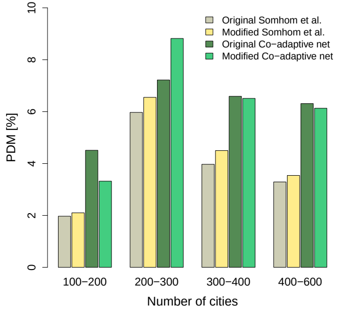
- *The problem is to determine the number of particular vehicles to visit given set of cities within a given time constraint.*

One of the SOM feature is that solutions found look acceptable for the operators.

Performance of SOM for the TSP in $\mathcal{W}$

- Combination of the new and already published modifications with implementation optimizations.





Initialization (including computation of the distance matrix) is sometimes more computationally demanding than the adaption.

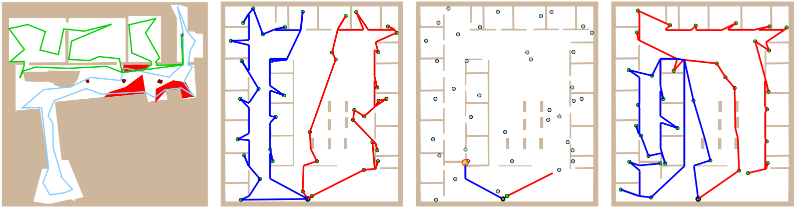
Jan Faigl
On the Performance of Self-Organizing Maps for the non-Euclidean Traveling Salesman Problem in the Polygonal Domain
Information Sciences, 181(19):4214–4229, (2011)

SOM's Features in $\mathcal{W}$

- The geometric interpretation of the adaptation procedure.

  Straightforward extensions for variants of multi-goal path planning.

- Any path planning method may eventually be used for determining $|S(\nu, g)|$.
- In MTSP, solutions with mutually non-crossing tours are preferred in the SOM adaptation.
- Flexible to addressed heterogeneous robots and re-planning.



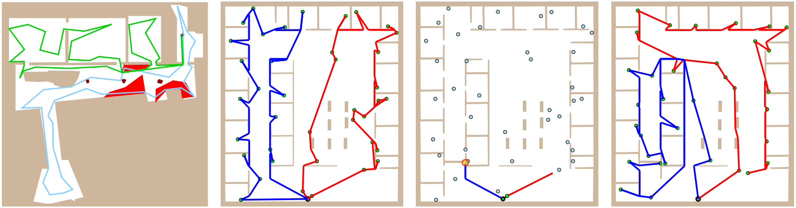
An evolution of the ring in the polygonal domain is inspiring for other problems.

SOM's Features in $\mathcal{W}$

- The geometric interpretation of the adaptation procedure.

  Straightforward extensions for variants of multi-goal path planning.

- Any path planning method may eventually be used for determining $|S(\nu, g)|$.
- In MTSP, solutions with mutually non-crossing tours are preferred in the SOM adaptation.
- Flexible to addressed heterogeneous robots and re-planning.

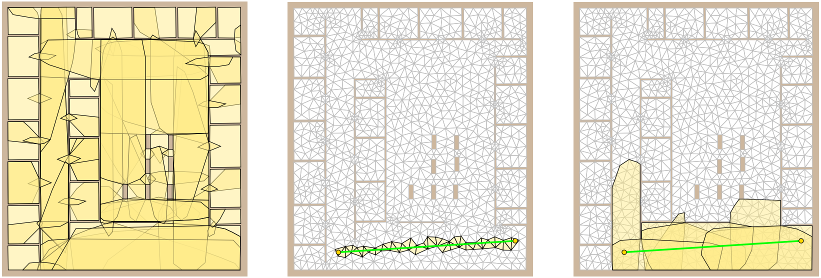


An evolution of the ring in the polygonal domain is inspiring for other problems.

Watchman Route Problem (WRP)

Compute coverage considering d-visibility of $\mathcal{W}$ from the ring of nodes and adapt nodes towards uncovered parts of $\mathcal{W}$.

- Convex cover set of $\mathcal{W}$ created on top of a triangular mesh
- Incident convex polygons with a straight line segment are found by walking in a triangular mesh technique.

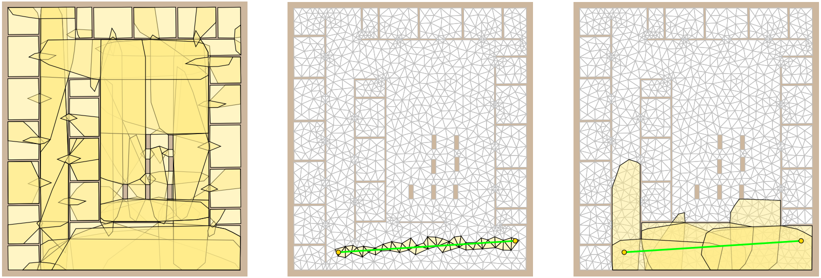


Having this supporting structure, we can consider coverage of the ring and adaptation towards not yet covered parts of $\mathcal{W}$.

Watchman Route Problem (WRP)

Compute ***coverage*** *considering d-visibility of $\mathcal{W}$ from the* ***ring*** *of nodes and* ***adapt*** *nodes* ***towards uncovered parts*** *of $\mathcal{W}$.*

- Convex cover set of $\mathcal{W}$ created on top of a triangular mesh
- Incident convex polygons with a straight line segment are found by walking in a triangular mesh technique.



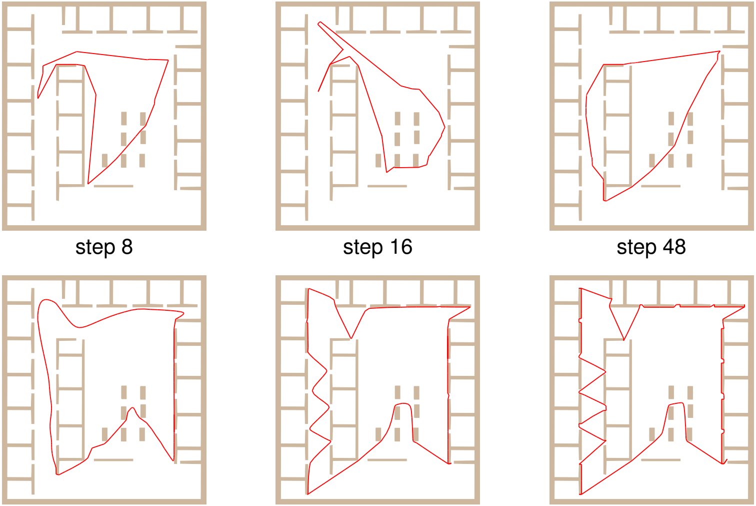
Having this supporting structure, we can consider coverage of the ring and adaptation towards not yet covered parts of $\mathcal{W}$.

Algorithm for the *d*-WRP

Input: $\mathcal{T} = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{T})$ – a triangular mesh of $\mathcal{W}$
Input: $\boldsymbol{P}$ – a set of convex polygons associated to $\boldsymbol{T}$
Output: $(\nu_1, \ldots, \nu_m)$ - nodes representing a route

```
r ← initialization                                   // create a ring of nodes
repeat
    I ← ∅                                            // a set of inhibited nodes
    T_c ← triangles covered by the current ring r
    Π(T) ← create a random permutation of triangles
    foreach T ∈ Π(T) do
        if T ∉ T_c then
            p_a ← centroid(T)                        // attraction point
            ν* ← select winner node to p_a, ν* ∉ I
            P_c ← {all associated convex polygons to T}
            if ν* ∉ P, P ∈ P_c then
                adapt(ν*, p_a)
            T_c ← T_c ∪ {T | T ∈ P, P ∈ P_c}
            I ← I ∪ {ν*}                             // inhibit winner node
    G ← (1 − α) · G                                  // decrease the gain
until all triangles are covered by the current ring
```

Evolution of SOM for the WRP



step 8 | step 16 | step 48

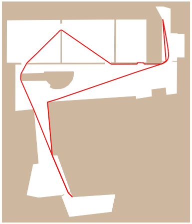
step 73 | step 81 | step 111

Supporting triangular mesh with 1417 triangles and 100 convex polygons.

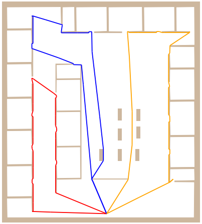
SOM based WRP with *d*-visibility

Ring of nodes represents watchman route and nodes are adapted towards uncovered parts of $\mathcal{W}$.

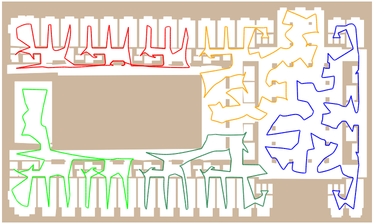
Representatives of uncovered parts are used as attraction points towards them the nodes are adapted.



d-WRP



d-WRP MinMax



Patrolling routes

Coverage from the ring is determined during the adaptation.

Jan Faigl,
Approximate Solution of the Multiple Watchman Routes Problem with Restricted Visibility Range,
IEEE Transactions on Neural Networks, 21(10):1668–1679 (2010).

Multi-Goal Planning with Trajectory Generation

- The distance metric can be computed by various approaches.

  It does not affect the main principle of SOM.

- A real computational requirements of the metric evaluation is crucial.

  Many distance / path queries have to be resolved during the SOM learning.

- Two approaches have been studied:
  - Artificial Potential Field (APF)

    Faigl J., Mačák J., ESANN, 2011

  - Rapidly-Exploring Random Tree (RRT)

    Vonásek et al., RoMoCo, 2009

    Vonásek et al., ECMR, 2011

Artificial Potential Field (APF) – Navigation Function

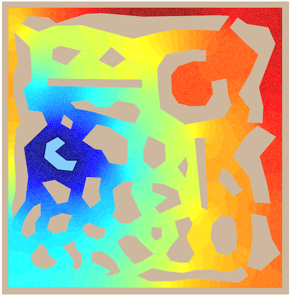
Navigation function f provides a path to the goal for an arbitrary point in the environment, i.e., -$\nabla f(q)$ points to the goal.

- Harmonic functions have only one extreme

$$\nabla^2 f(g) = 0$$

Dirichlet condition for the goal boundary
Neuman condition for obstacles boundary

- Finite Element Method
- Solution can be found for a goal with an arbitrary shape.
- Segment goals (guards).



During the SOM evolution in $\mathcal{W}$ particular points from the segment goals are selected and the final inspection path is found.

Artificial Potential Field (APF) – Navigation Function

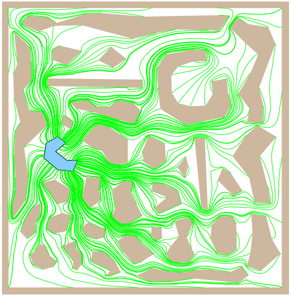
Navigation function f provides a path to the goal for an arbitrary point in the environment, i.e., $-\nabla f(q)$ points to the goal.

- Harmonic functions have only one extreme

$$\nabla^2 f(g) = 0$$

Dirichlet condition for the goal boundary
Neuman condition for obstacles boundary

- Finite Element Method
- Solution can be found for a goal with an arbitrary shape.
- Segment goals (guards).



During the SOM evolution in $\mathcal{W}$ particular points from the segment goals are selected and the final inspection path is found.

Artificial Potential Field (APF) – Navigation Function

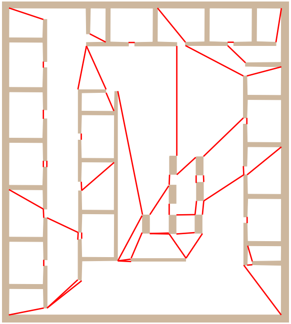
Navigation function f provides a path to the goal for an arbitrary point in the environment, i.e., $-\nabla f(q)$ points to the goal.

- Harmonic functions have only one extreme

$$\nabla^2 f(g) = 0$$

Dirichlet condition for the goal boundary
Neuman condition for obstacles boundary

- Finite Element Method
- Solution can be found for a goal with an arbitrary shape.
- Segment goals (guards).



During the SOM evolution in $\mathcal{W}$ particular points from the segment goals are selected and the final inspection path is found.

Artificial Potential Field (APF) – Navigation Function

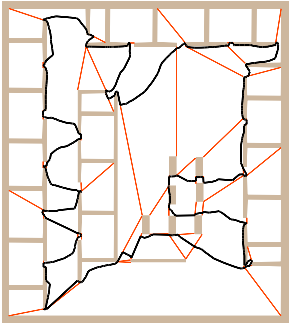
Navigation function f provides a path to the goal for an arbitrary point in the environment, i.e., $-\nabla f(q)$ points to the goal.

- Harmonic functions have only one extreme

$$\nabla^2 f(g) = 0$$

Dirichlet condition for the goal boundary
Neuman condition for obstacles boundary

- Finite Element Method
- Solution can be found for a goal with an arbitrary shape.
- Segment goals (guards).



During the SOM evolution in $\mathcal{W}$ particular points from the segment goals are selected and the final inspection path is found.

SOM with RRT

Rapidly-Exploring Random Tree - (RRT)

kinodynamic constraints

- Standard RRT approaches have poor performance

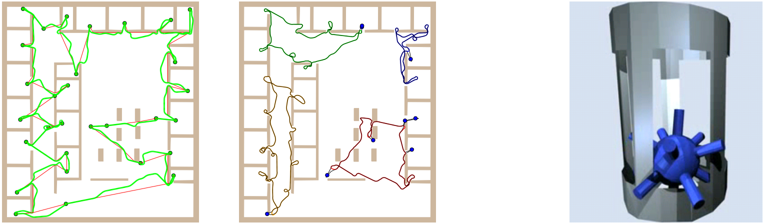
Especially in narrow passages

- RRT–Path – an improved RRT for narrow passages

Vonásek V. et al., RoMoCo, 2009

- RRT–Path^{ext} – Multi-Goal Motion Planner

A feasibility study to find patrolling trajectories using SOM with the RRT–Path^{ext}



Vonásek et al., ECMR'11

Part IV

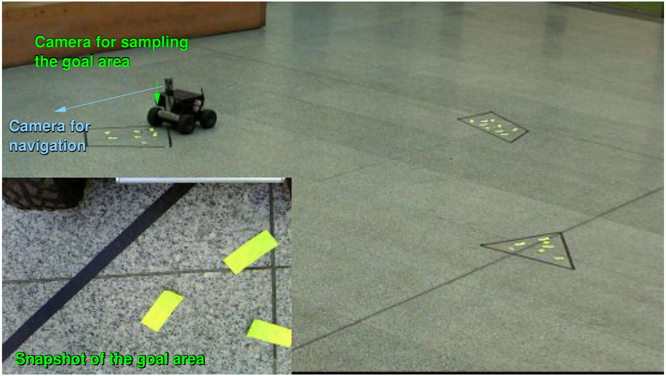
Unified Self-Organizing Maps for 2D Multi-Goal Path Planning Problems

Multi-Goal Path Planning with Polygonal Goals

Motivation:

Visit a given set of polygonal goals.

E.g., to take a sample measurement at each goal



The problem is a variant of the
Traveling Salesman Problem with Neighborhoods (TSPN).

NP-hard, APX-hard

Multi-Goal Path Planning with Polygonal Goals

Motivation:

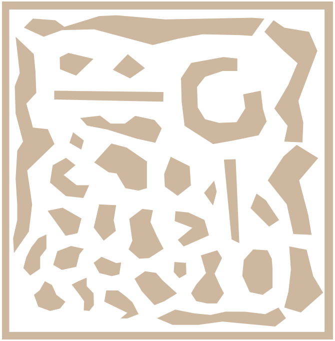
Visit a given set of polygonal goals.

E.g., to take a sample measurement at each goal

Specification:

- Input:
  - A map of the environment

    The polygonal domain $\mathcal{W}$
  - A set of goals
- Output:
  - A shortest path visiting all requested goals



The problem is a variant of the
Traveling Salesman Problem with Neighborhoods (TSPN).

NP-hard, APX-hard

Multi-Goal Path Planning with Polygonal Goals

Motivation:

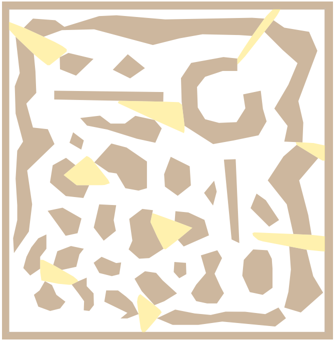
Visit a given set of polygonal goals.

E.g., to take a sample measurement at each goal

Specification:

- Input:
  - A map of the environment

    The polygonal domain $\mathcal{W}$
  - A set of goals
- Output:
  - A shortest path visiting all requested goals



The problem is a variant of the
Traveling Salesman Problem with Neighborhoods (TSPN).

NP-hard, APX-hard

Multi-Goal Path Planning with Polygonal Goals

Motivation:

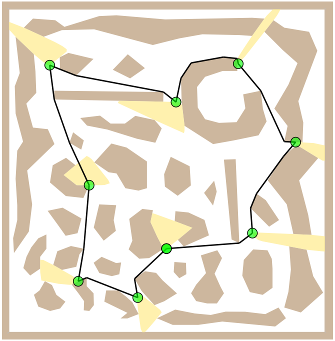
Visit a given set of polygonal goals.

E.g., to take a sample measurement at each goal

Specification:

- Input:
  - A map of the environment

    The polygonal domain $\mathcal{W}$
  - A set of goals
- Output:
  - A shortest path visiting all requested goals



The problem is a variant of the Traveling Salesman Problem with Neighborhoods (TSPN).

NP-hard, APX-hard

Multi-Goal Path Planning with Polygonal Goals

Motivation:

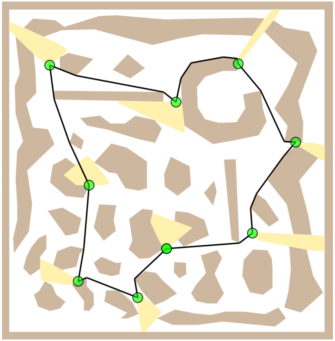
Visit a given set of polygonal goals.

E.g., to take a sample measurement at each goal

Specification:

- Input:
  - A map of the environment

    The polygonal domain $\mathcal{W}$
  - A set of goals
- Output:
  - A shortest path visiting all requested goals



The problem is a variant of the
Traveling Salesman Problem with Neighborhoods (TSPN).

NP-hard, APX-hard

Dealing with Polygonal Goals

Goals are simple (convex) polygons.

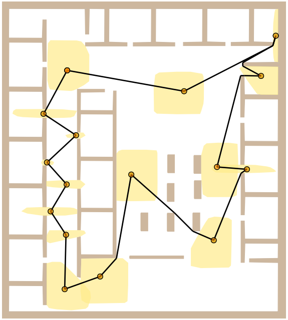
A polygonal goal g can be represented by its centroid $c(g)$.

Centroids can be used as point goals.

Straightforward extensions can provide better solutions:

Based on geometrical interpretation.

1. Interior of the goal
   - Use $c(g)$ of the goal g as a point goal.
   - Do not adapt nodes inside the goal.
2. Attraction point
   - Select winner using $c(g)$
   - Adapt neurons towards intersection point of $S(\nu, c(g))$ and g.
3. Alternate goal
   - Select winner using border of the goal g.
     g as a set of segments in $\mathcal{W}$
   - Adapt neurons towards the point at the border of g.



The alternate goal approach does not require convex goals.

Dealing with Polygonal Goals

Goals are simple (convex) polygons.

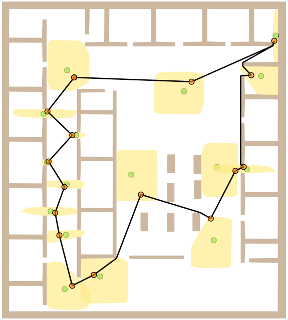
A polygonal goal g can be represented by its centroid $c(g)$.

Centroids can be used as point goals.

Straightforward extensions can provide better solutions:

Based on geometrical interpretation.

1. Interior of the goal
   - Use $c(g)$ of the goal g as a point goal.
   - Do not adapt nodes inside the goal.
2. Attraction point
   - Select winner using $c(g)$
   - Adapt neurons towards intersection point of $S(\nu, c(g))$ and g.
3. Alternate goal
   - Select winner using border of the goal g.
     g as a set of segments in $\mathcal{W}$
   - Adapt neurons towards the point at the border of g.



The alternate goal approach does not require convex goals.

Dealing with Polygonal Goals

Goals are simple (convex) polygons.

A polygonal goal g can be represented by its centroid $c(g)$.

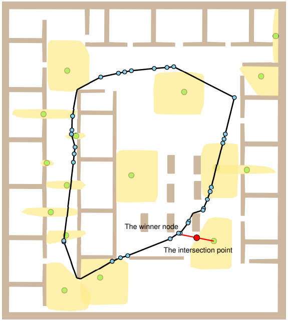
Centroids can be used as point goals.

Straightforward extensions can provide better solutions:

Based on geometrical interpretation.

1. Interior of the goal
   - Use $c(g)$ of the goal g as a point goal.
   - Do not adapt nodes inside the goal.
2. Attraction point
   - Select winner using $c(g)$
   - Adapt neurons towards intersection point of $S(\nu, c(g))$ and g.
3. Alternate goal
   - Select winner using border of the goal g. *g as a set of segments in $\mathcal{W}$*
   - Adapt neurons towards the point at the border of g.

The alternate goal approach does not require convex goals.



Dealing with Polygonal Goals

Goals are simple (convex) polygons.

A polygonal goal g can be represented by its centroid $c(g)$.

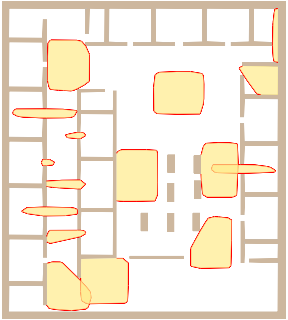
Centroids can be used as point goals.

Straightforward extensions can provide better solutions:

Based on geometrical interpretation.

1. Interior of the goal
   - Use $c(g)$ of the goal g as a point goal.
   - Do not adapt nodes inside the goal.
2. Attraction point
   - Select winner using $c(g)$
   - Adapt neurons towards intersection point of $S(\nu, c(g))$ and g.
3. Alternate goal
   - Select winner using border of the goal g.

     g as a set of segments in $\mathcal{W}$
   - Adapt neurons towards the point at the border of g.



The alternate goal approach does not require convex goals.

Dealing with Polygonal Goals

Goals are simple (convex) polygons.

A polygonal goal g can be represented by its centroid $c(g)$.

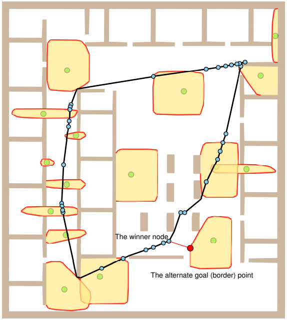
Centroids can be used as point goals.

Straightforward extensions can provide better solutions:

Based on geometrical interpretation.

1. Interior of the goal
   - Use $c(g)$ of the goal g as a point goal.
   - Do not adapt nodes inside the goal.
2. Attraction point
   - Select winner using $c(g)$
   - Adapt neurons towards intersection point of $S(\nu, c(g))$ and g.
3. Alternate goal
   - Select winner using border of the goal g.

     g as a set of segments in $\mathcal{W}$
   - Adapt neurons towards the point at the border of g.



The alternate goal approach does not require convex goals.

SOM for the MTP with Polygonal Goals

- SOM for the TSP in $\mathcal{W}$
- Distance metric - the shortest path bettwen two segments

Approximate path is used – Faigl et al. (2011)

- Goal - segments $\{s_1^g, \ldots, s_k^g\}$
- Ring - segments $\{s_1^r, \ldots, s_l^r\}$

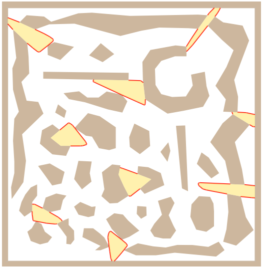
nodes and map vertices

- Winner Selection
  1. Determine a pair (s_i^r, s_j^g) with minimal distance

     two resulting points $p_r \in s_i^r$, $p_g \in s_j^g$
  2. The winner is at p_r

     a new neuron may be created at p_r
- Adapt the winner toward p_g

using adaptation for point goals



SOM for the MTP with Polygonal Goals

- SOM for the TSP in $\mathcal{W}$
- Distance metric - the shortest path bettwen two segments

Approximate path is used – Faigl et al. (2011)

- Goal - segments $\{s_1^g, \ldots, s_k^g\}$
- Ring - segments $\{s_1^r, \ldots, s_l^r\}$

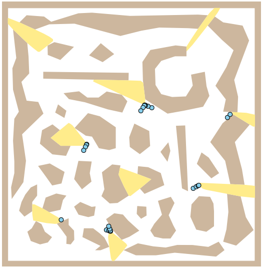
nodes and map vertices

- Winner Selection
  1. Determine a pair (s_i^r, s_j^g) with minimal distance

     two resulting points $p_r \in s_i^r$, $p_g \in s_j^g$
  2. The winner is at p_r

     a new neuron may be created at p_r
- Adapt the winner toward p_g

using adaptation for point goals



SOM for the MTP with Polygonal Goals

- SOM for the TSP in $\mathcal{W}$
- Distance metric - the shortest path bettwen two segments

Approximate path is used – Faigl et al. (2011)

- Goal - segments $\{s_1^g, \ldots, s_k^g\}$
- Ring - segments $\{s_1^r, \ldots, s_l^r\}$

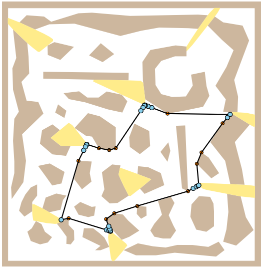
nodes and map vertices

- Winner Selection
  1. Determine a pair (s_i^r, s_j^g) with minimal distance

     two resulting points $p_r \in s_i^r$, $p_g \in s_j^g$
  2. The winner is at p_r

     a new neuron may be created at p_r
- Adapt the winner toward p_g

using adaptation for point goals



SOM for the MTP with Polygonal Goals

- SOM for the TSP in $\mathcal{W}$
- Distance metric - the shortest path bettwen two segments

Approximate path is used – Faigl et al. (2011)

- Goal - segments $\{s_1^g, \ldots, s_k^g\}$
- Ring - segments $\{s_1^r, \ldots, s_l^r\}$

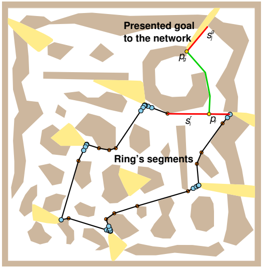
nodes and map vertices

- Winner Selection
  1. Determine a pair (s_i^r, s_j^g) with minimal distance

     two resulting points $p_r \in s_i^r$, $p_g \in s_j^g$
  2. The winner is at p_r

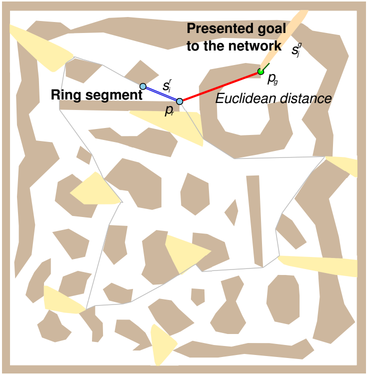
     a new neuron may be created at p_r
- Adapt the winner toward p_g

using adaptation for point goals



Proposed Adaptation Schema

1. **For each goal g**
   Winner selection
   regarding $|S(p_r, p_g)|$
   Euclidean pre-selection of (s_i^r, s_j^g).
   two resulting points $p_r \in s_i^r$, $p_g \in s_j^g$
   Approx. shortest path $S(p_r, p_g)$.
   Adapt toward the point goal p_g
2. **Regenerate ring**
   - Preserve winners
   - Connect winners
   - Add vertices as additional neurons
3. **Termination condition**
   "All goals contain a distinct winner."



The final path is constructed from the last winners.
using approx. shortest path

Proposed Adaptation Schema

1. **For each goal g**

   Winner selection

   regarding $|S(p_r, p_g)|$

   Euclidean pre-selection of (s_i^r, s_j^g).

   two resulting points $p_r \in s_i^r$, $p_g \in s_j^g$

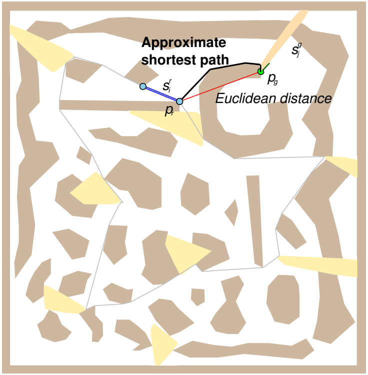
   Approx. shortest path $S(p_r, p_g)$.

   Adapt toward the point goal p_g

2. **Regenerate ring**
   - Preserve winners
   - Connect winners
   - Add vertices as additional neurons

3. **Termination condition**

   "All goals contain a distinct winner."

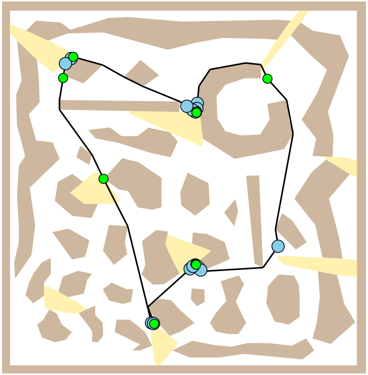


The final path is constructed from the last winners.

using approx. shortest path

Proposed Adaptation Schema

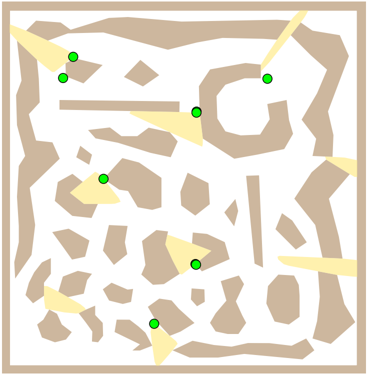
1. **For each goal g**
   Winner selection
   regarding $|S(p_r, p_g)|$
   Euclidean pre-selection of (s_i^r, s_j^g).
   two resulting points $p_r \in s_i^r$, $p_g \in s_j^g$
   Approx. shortest path $S(p_r, p_g)$.
   Adapt toward the point goal p_g
2. **Regenerate ring**
   - Preserve winners
   - Connect winners
   - Add vertices as additional neurons
3. **Termination condition**
   "All goals contain a distinct winner."



The final path is constructed from the last winners.
using approx. shortest path

Proposed Adaptation Schema

1. **For each goal g**
   Winner selection
   regarding $|S(p_r, p_g)|$
   Euclidean pre-selection of (s_i^r, s_j^g).
   two resulting points $p_r \in s_i^r$, $p_g \in s_j^g$
   Approx. shortest path $S(p_r, p_g)$.
   Adapt toward the point goal p_g
2. **Regenerate ring**
   - Preserve winners
   - Connect winners
   - Add vertices as additional neurons
3. **Termination condition**
   "All goals contain a distinct winner."



The final path is constructed from the last winners.
using approx. shortest path

Proposed Adaptation Schema

1. **For each goal g**

   Winner selection

   regarding $|S(p_r, p_g)|$

   Euclidean pre-selection of (s_i^r, s_j^g).

   two resulting points $p_r \in s_i^r$, $p_g \in s_j^g$

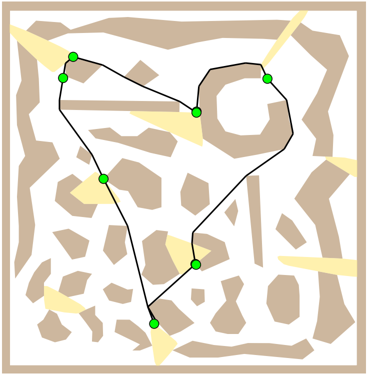
   Approx. shortest path $S(p_r, p_g)$.

   Adapt toward the point goal p_g

2. **Regenerate ring**
   - Preserve winners
   - Connect winners
   - Add vertices as additional neurons

3. **Termination condition**

   "All goals contain a distinct winner."



The final path is constructed from the last winners.

using approx. shortest path

Proposed Adaptation Schema

1. **For each goal g**

   Winner selection

   regarding $|S(p_r, p_g)|$

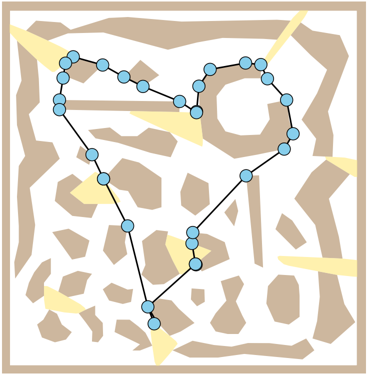
   Euclidean pre-selection of (s_i^r, s_j^g).

   two resulting points $p_r \in s_i^r$, $p_g \in s_j^g$

   Approx. shortest path $S(p_r, p_g)$.

   Adapt toward the point goal p_g

2. **Regenerate ring**
   - Preserve winners
   - Connect winners
   - Add vertices as additional neurons



3. **Termination condition**

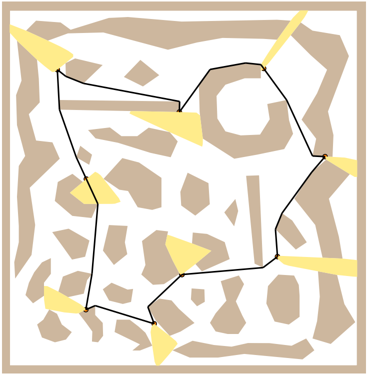
   "All goals contain a distinct winner."

The final path is constructed from the last winners.

using approx. shortest path

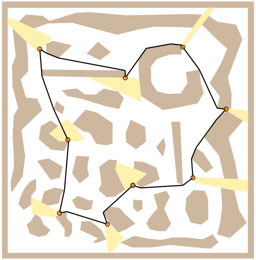
Proposed Adaptation Schema

1. **For each goal g**
   Winner selection
   regarding $|S(p_r, p_g)|$
   Euclidean pre-selection of (s_i^r, s_j^g).
   two resulting points $p_r \in s_i^r$, $p_g \in s_j^g$
   Approx. shortest path $S(p_r, p_g)$.
   Adapt toward the point goal p_g
2. **Regenerate ring**
   - Preserve winners
   - Connect winners
   - Add vertices as additional neurons
3. **Termination condition**
   "All goals contain a distinct winner."



The final path is constructed from the last winners.
using approx. shortest path

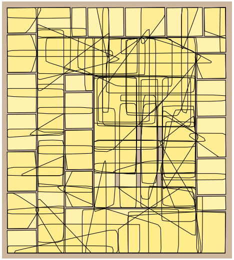
Problem Variants



Polygonal Goals

Safari Route Problem

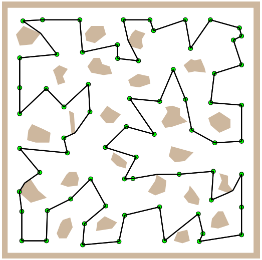
n=9, T= 0.34 s



Convex Cover Set

Watchman Route Problem

n=106, T=2.66 s



Point Goals

Traveling Salesman Problem

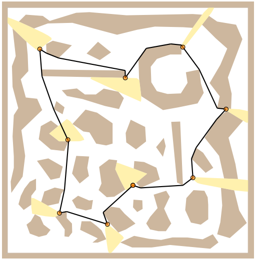
n=68, T=0.35 s

- Solutions of the MTP with polygonal goals
- Improved quality of solutions for the Watchman Route Problem
- Scales better for problems with more goals

SOM provides a unified approach to solve various problems in $\mathcal{W}$.

SOM benefit over other approximating or optimal approaches.

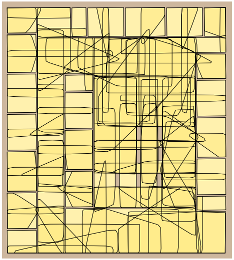
Problem Variants



Polygonal Goals

Safari Route Problem

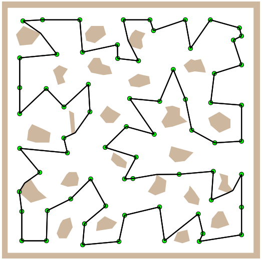
n=9, T= 0.34 s



Convex Cover Set

Watchman Route Problem

n=106, T=2.66 s



Point Goals

Traveling Salesman Problem

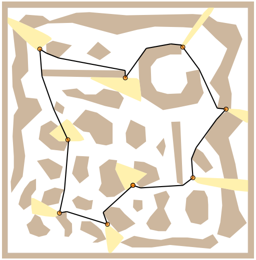
n=68, T=0.35 s

- Solutions of the MTP with polygonal goals
- Improved quality of solutions for the Watchman Route Problem
- Scales better for problems with more goals

SOM provides a unified approach to solve various problems in $\mathcal{W}$.

SOM benefit over other approximating or optimal approaches.

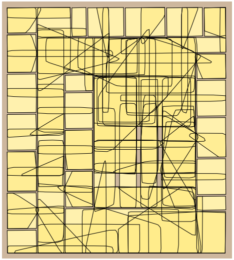
Problem Variants



Polygonal Goals

Safari Route Problem

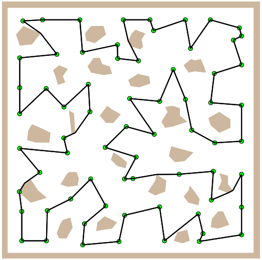
n=9, T= 0.34 s



Convex Cover Set

Watchman Route Problem

n=106, T=2.66 s



Point Goals

Traveling Salesman Problem

n=68, T=0.35 s

- Solutions of the MTP with polygonal goals
- Improved quality of solutions for the Watchman Route Problem
- Scales better for problems with more goals

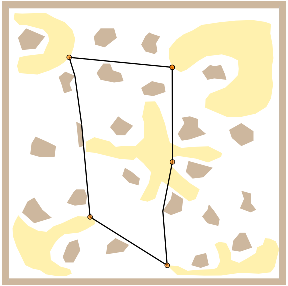
SOM provides a unified approach to solve various problems in $\mathcal{W}$.

SOM benefit over other approximating or optimal approaches.

Features of the Proposed Adaptation Schema

- Self-adjustment of the number of neurons
- It Seems to be independent on adaptation parameters
  - $\sigma_0 = 10$
  - $\alpha = 0.001$
  - $\mu = 1/\sqrt[4]{k}$; $\mu_0 = 1$

practically parameter less



Non-Convex Goals, T=0.1 s

SOM forms a framework for relatively simple algorithms providing high quality solutions of routing problems in $\mathcal{W}$.

J. Faigl, L. Přeučil
Self-Organizing Map for the Multi-Goal Path Planning with Polygonal Goals,
ICANN, 85–92 (2011).

Part V

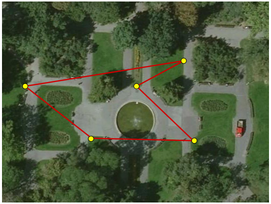
Multi-Goal Path Planning with Localization Uncertainty

Autonomous Inspection / Surveillance

The problem is to maximize the frequency of goals' visits.

It can be achieved by:

- The shortest (fastest) path connecting the goals
  Multi-Goal Path Planning $\sim$ *TSP*
- Precise navigation to the goals



Multi-criteria optimization

The idea is to consider a model of the localization uncertainty during the planning to find a path to increase robustness and reliability of the autonomous navigation to the goals.

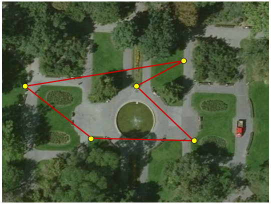
We need a realistic model of the localization uncertainty evolution.

Autonomous Inspection / Surveillance

The problem is to maximize the frequency of goals' visits.

It can be achieved by:

- The shortest (fastest) path connecting the goals
  Multi-Goal Path Planning $\sim$ *TSP*
- Precise navigation to the goals



Multi-criteria optimization

The idea is to consider a model of the localization uncertainty during the planning to find a path to increase robustness and reliability of the autonomous navigation to the goals.

We need a realistic model of the localization uncertainty evolution.

Autonomous Navigation
SURFNav - Simple and Stable Navigational Method

- Map and Replay Technique

  The map is a sequence of learned segments

- Detection of Salient Objects

  Speeded Up Robust Features (SURF)

- Navigation – a sequence of segments
  - Bearing-Only Correction
  - Dead-Reckoning for switching segments

- Model of the navigation
  - Covariance matrix of the robot position

    end of the segment $(i, i+1)$

$$\boldsymbol{A}_{i+1} = \boldsymbol{R}_i^T \boldsymbol{M}_i \boldsymbol{R}_i \boldsymbol{A}_i \boldsymbol{R}_i^T \boldsymbol{M}_i^T \boldsymbol{R}_i + \boldsymbol{R}_i^T \boldsymbol{S}_i \boldsymbol{R}_i,$$

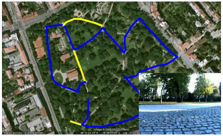
where

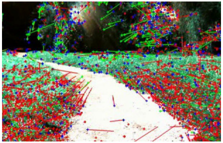
$$\boldsymbol{M}_i = \begin{bmatrix} 1 & 0 \\ 0 & m(a_i, a_{i+1}, \mathcal{M}) \end{bmatrix}, \boldsymbol{S}_i = \begin{bmatrix} s_i\eta^2 & 0 \\ 0 & \tau^2 \end{bmatrix}$$

$m(a_i, a_{i+1}, \mathcal{M})$ - model of the visible landmarks

$\eta, \tau \sim$ *"odometry and heading error"* (variances)

$s_i = |(a_i, a_{i+1})|$ - the segment length





Autonomous Navigation
SURFNav - Simple and Stable Navigational Method

- Map and Replay Technique

  The map is a sequence of learned segments

- Detection of Salient Objects

  Speeded Up Robust Features (SURF)

- Navigation – a sequence of segments
  - Bearing-Only Correction
  - Dead-Reckoning for switching segments
- Model of the navigation
  - Covariance matrix of the robot position

    end of the segment $(i, i+1)$

    $$\boldsymbol{A}_{i+1} = \boldsymbol{R}_i^T \boldsymbol{M}_i \boldsymbol{R}_i \boldsymbol{A}_i \boldsymbol{R}_i^T \boldsymbol{M}_i^T \boldsymbol{R}_i + \boldsymbol{R}_i^T \boldsymbol{S}_i \boldsymbol{R}_i,$$

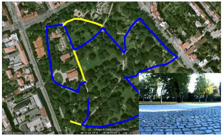
    where

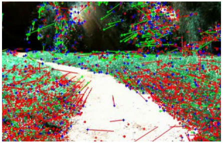
    $$\boldsymbol{M}_i = \begin{bmatrix} 1 & 0 \\ 0 & m(a_i, a_{i+1}, \mathcal{M}) \end{bmatrix}, \boldsymbol{S}_i = \begin{bmatrix} s_i\eta^2 & 0 \\ 0 & \tau^2 \end{bmatrix}$$

    $m(a_i, a_{i+1}, \mathcal{M})$ - model of the visible landmarks

    $\eta, \tau \sim$ *"odometry and heading error"* (variances)

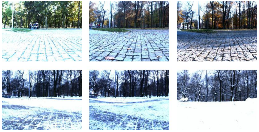
    $s_i = |(a_i, a_{i+1})|$ - the segment length



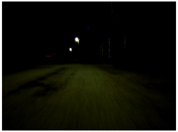


Reliability of the Navigation

One-day navigation – changing lighting conditions



Long-term reliability – seasonal changes



Night navigation

Autonomous navigation for a low-cost UAV platform

Processing time (1024x768)
CPU (2x2 GHz) $\sim$ 1 FPS
GPU

- NVS 320 $\sim$ 25 FPS
- ION $\sim$ 15 FPS

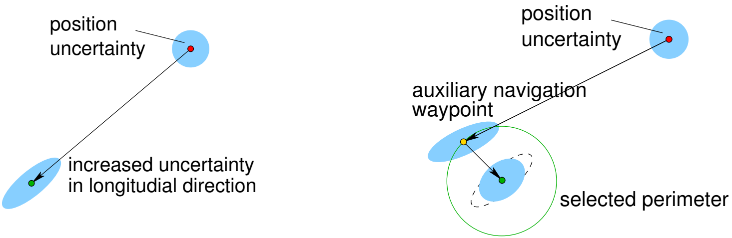
FPGA (Virtex 5) $\sim$ 10 FPS, 9 W

Stability theoretically and experimentally proven

Krajník T., Faigl J., Vonásek V., Košnar K., Kulich M., Přeučil L.,
Simple yet stable bearing-only navigation,
Journal of Field Robotics,
27(5):511–533, 2010.

Principle of Localization Uncertainty Decreasing

The stability of the navigation is based on bearing corrections

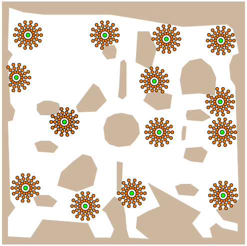


Heading corrections are more precise than odometry

The localization uncertainty can be decreased by auxiliary navigation waypoints

Visit an auxiliary navigation waypoint prior visiting the goal

Multi-Goal Path Planning with Auxiliary Waypoints



- Map of the environment
- A set of the point goals

  Traveling Salesman Problem (TSP)

- Auxiliary navigation waypoints

  A variant of the TSPN

- Selection of the most suitable auxiliary navigation waypoint

$$w = \operatorname{argmin}_{w_i \in W} ||\boldsymbol{A}_{w_i,g}||$$

SOM proposes auxiliary navigation waypoints that decrease the localization uncertainty at the goals.

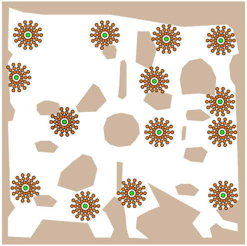
Multi-Goal Path Planning with Auxiliary Waypoints

- Map of the environment
- A set of the point goals

Traveling Salesman Problem (TSP)

- Auxiliary navigation waypoints

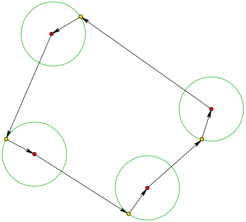
A variant of the TSPN



- Selection of the most suitable auxiliary navigation waypoint

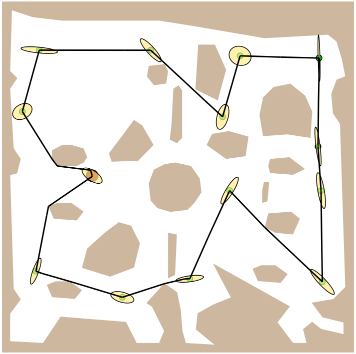
$$w = \mathrm{argmin}_{w_i \in W} ||\boldsymbol{A}_{w_i,g}||$$

SOM proposes auxiliary navigation waypoints that decrease the localization uncertainty at the goals.

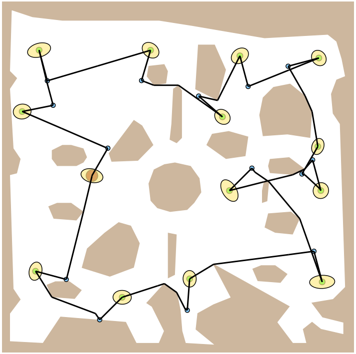


Simulation Results

- SOM selects auxiliary navigation waypoints that decrease the localization uncertainty at the goals.



L=416 m, E_{max}=1.23 m



L=425 m, E_{max}=0.7 m

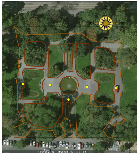
E_{max} - expected error at the goal

Experimental Results (1/2) - Outdoor Environment

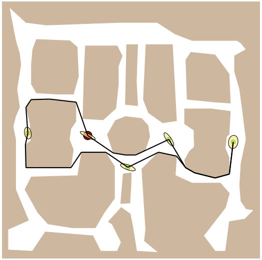
- P3AT robot
- City park - traveling on pathways

  several runs

- Random pedestrians

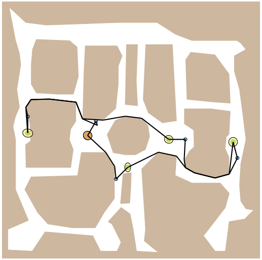


Simple TSP



L=184 m, E_{avg}=0.57, E_{max}=0.63

Proposed approach



L=202 m, E_{avg}=0.35, E_{max}=0.37

- Real overall error at the goals decreased from 0.89 m $\rightarrow$ 0.58 m

  (improvement about 35%)

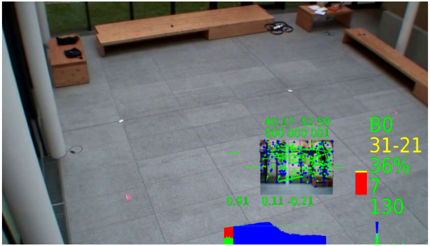
Experimental Results (2/2) - Indoor Environment

Small low-cost platforms

Small UGV - MMP5

Overall error at the goals decreased from 16.6 cm → 12.8 cm

Small UAV - Parrot AR.Drone



Improvement of the success of the goals' visits 83%→95%

Faigl et al., ICR'10

Faigl J., Krajník T., Vonásek V., Přeučil L.,
On Localization Uncertainty in an Autonomous Inspection,
ICRA, 1119–1124 (2012).

Part VI

Recent, Ongoing and Future Work

Surveillance of Objects of Interest in 3D

Problem: *Find the shortest closed inspection path $\boldsymbol{I}$ such that all objects of interest $\boldsymbol{M}$ will be seen from $\boldsymbol{I}$ by the sensor with the visibility range ρ.*

The idea is based on SOM for the WRP (2D):

- SOM evolves on a graph G_{PRM}
- Objects of interest are represented as a set of triangles
- Objects can be covered from covering spaces
- Adaptation towards covering spaces of each $m \in \boldsymbol{M}$

Fast visibility queries are the key issue

P. Janoušek, Master's Thesis, 2013 (to be defended)

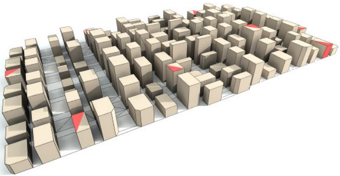
P. Janoušek, J. Faigl, Speeding Up Coverage Queries in 3D Multi-Goal Path Planning, ICRA 2013.

Surveillance of Objects of Interest in 3D

Problem: *Find the shortest closed inspection path **I** such that all objects of interest **M** will be seen from **I** by the sensor with the visibility range ρ.*

The idea is based on SOM for the WRP (2D):

- SOM evolves on a graph G_{PRM}
- Objects of interest are represented as a set of triangles
- Objects can be covered from covering spaces
- Adaptation towards covering spaces of each $m \in \boldsymbol{M}$



Fast visibility queries are the key issue

P. Janoušek, Master's Thesis, 2013 (to be defended)

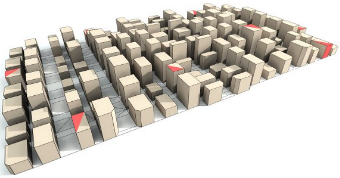
P. Janoušek, J. Faigl, Speeding Up Coverage Queries in 3D Multi-Goal Path Planning, ICRA 2013.

Surveillance of Objects of Interest in 3D

Problem: *Find the shortest closed inspection path **I** such that all objects of interest **M** will be seen from **I** by the sensor with the visibility range ρ.*

The idea is based on SOM for the WRP (2D):

- SOM evolves on a graph G_{PRM}
- Objects of interest are represented as a set of triangles
- Objects can be covered from covering spaces
- Adaptation towards covering spaces of each $m \in \boldsymbol{M}$



Fast visibility queries are the key issue

P. Janoušek, Master's Thesis, 2013 (to be defended)

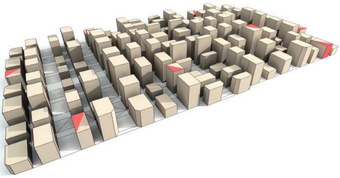
P. Janoušek, J. Faigl, Speeding Up Coverage Queries in 3D Multi-Goal Path Planning, ICRA 2013.

Surveillance of Objects of Interest in 3D

Problem: *Find the shortest closed inspection path **I** such that all objects of interest **M** will be seen from **I** by the sensor with the visibility range ρ.*

The idea is based on SOM for the WRP (2D):

- SOM evolves on a graph G_{PRM}
- Objects of interest are represented as a set of triangles
- Objects can be covered from covering spaces
- Adaptation towards covering spaces of each $m \in \boldsymbol{M}$



Fast visibility queries are the key issue

P. Janoušek, Master's Thesis, 2013 (to be defended)

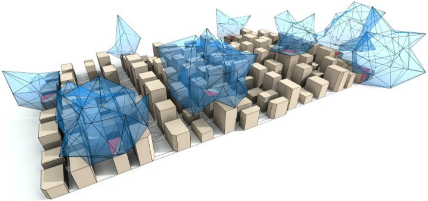
P. Janoušek, J. Faigl, Speeding Up Coverage Queries in 3D Multi-Goal Path Planning, ICRA 2013.

Surveillance of Objects of Interest in 3D

Problem: *Find the shortest closed inspection path **I** such that all objects of interest **M** will be seen from **I** by the sensor with the visibility range ρ.*

The idea is based on SOM for the WRP (2D):

- SOM evolves on a graph G_{PRM}
- Objects of interest are represented as a set of triangles
- Objects can be covered from covering spaces
- Adaptation towards covering spaces of each $m \in \boldsymbol{M}$



Fast visibility queries are the key issue

P. Janoušek, Master's Thesis, 2013 (to be defended)

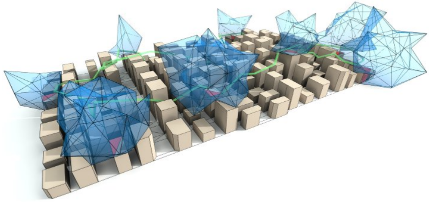
P. Janoušek, J. Faigl, Speeding Up Coverage Queries in 3D Multi-Goal Path Planning, ICRA 2013.

Surveillance of Objects of Interest in 3D

Problem: *Find the shortest closed inspection path $\boldsymbol{I}$ such that all objects of interest $\boldsymbol{M}$ will be seen from $\boldsymbol{I}$ by the sensor with the visibility range ρ.*

The idea is based on SOM for the WRP (2D):

- SOM evolves on a graph G_{PRM}
- Objects of interest are represented as a set of triangles
- Objects can be covered from covering spaces
- Adaptation towards covering spaces of each $m \in \boldsymbol{M}$



Fast visibility queries are the key issue

P. Janoušek, Master's Thesis, 2013 (to be defended)

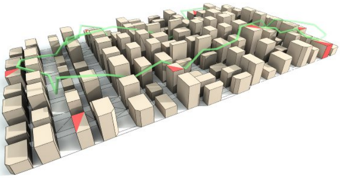
P. Janoušek, J. Faigl, Speeding Up Coverage Queries in 3D Multi-Goal Path Planning, ICRA 2013.

Surveillance of Objects of Interest in 3D

Problem: *Find the shortest closed inspection path **I** such that all objects of interest **M** will be seen from **I** by the sensor with the visibility range ρ.*

The idea is based on SOM for the WRP (2D):

- SOM evolves on a graph G_{PRM}
- Objects of interest are represented as a set of triangles
- Objects can be covered from covering spaces
- Adaptation towards covering spaces of each $m \in \boldsymbol{M}$



Fast visibility queries are the key issue

P. Janoušek, Master's Thesis, 2013 (to be defended)

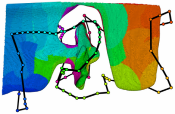
P. Janoušek, J. Faigl, Speeding Up Coverage Queries in 3D Multi-Goal Path Planning, ICRA 2013.

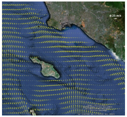
Self-Organizing Maps for Multi-Goal Path Planning

- Problems:
  - Optimal sampling design and motion constraints
  - High-dimensional configuration spaces
  - Models of sources of uncertainties

    planning in belief space
  - Planning a short and low risk path for autonomous underwater vehicles (AUVs)
  - Planning in Spatio-Temporal Spaces

    Considering ocean currents affecting the navigation.
- Framework for Planning Robotic Missions:
  - Inspection, Coverage, Surveillance
  - Environment Monitoring and Data Collections





Part VII

Concluding Remarks

Concluding Remarks

- Self-organizing map for multi-goal path planning problems in 2D environments.
- The main idea of the planning is based on the SOM principle augmented by supporting structures.
- Combining simple approximations provide quality solutions.
- Intuitive extensions based on geometric interpretation of the learning process.

- Further challenges
  - High dimensional configuration spaces

    kinematic or kinodynamic constraints
  - Considering time domain

    spatio-temporal spaces
  - Considering autonomous navigation (sources of localization uncertainties)

    belief/probability spaces

Questions and Discussion

Looking for motivated students for bachelor, master or doctoral theses. Contact me via faiglj@fel.cvut.cz

http://agents.fel.cvut.cz/~faigl