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, 
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*,  
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,

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*,  
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


  *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


  *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 necessarily be closed (planning an open path without returning to the depot).
- Planning for a heterogeneous robotic team (robots can have different capabilities for traversing the environment).

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

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

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

Self-Organizing Maps for the Traveling Salesman Problem
Self-Organizing Maps – Literature


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

Self-Organizing Maps for the TSP

• First approaches proposed in 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 $g$, $g \in G = \{g_1, \ldots, g_n\}$ 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 $g$

     $\nu^* = \arg\min_{\nu \in \mathcal{N}} |S(\nu, g)|$.

     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. $\nu^*$ and its neighbors are updated using the learning rule

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

     Cooperative phase.

     $\mu$ – 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

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

- $l$ is distance of $\nu$ from the winner $\nu^*$.
- $M$ is the number of neurons, $M = kN$ for $N$ goals, $k \in \langle 2, 3 \rangle$.
- $\sigma$ 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.


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,

А self-organising model for the travelling salesman problem,

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

\[
N \leftarrow \text{initialization}(\nu_1, \ldots, \nu_m) \quad // \text{set weights to form a ring around } g_1
\]
\[
\sigma \leftarrow \sigma_0 \quad // \text{set the initial value of the learning gain}
\]
\begin{algorithm}
\begin{algorithmic}
\Repeat
\State $\mathcal{I} \leftarrow \emptyset$ \hspace{1cm} // clear inhibited neurons
\State $\text{error} \leftarrow 0$ \hspace{1cm} // set the maximal error
\State $\Pi \leftarrow \text{create a random permutation of the goals}$
\ForEach{$g \in \Pi(G)$}
\State $\nu^* \leftarrow \arg\min_{\nu \in \mathcal{N}, \nu \notin \mathcal{I}} |(g, \nu)|$ \hspace{1cm} // the closest non-inhibited $\nu$ to $g$
\State $\text{error} \leftarrow \max\{\text{error}, |(g, \nu^*)|\}$ \hspace{1cm} // update error
\ForEach{$\nu_i \text{ in } \text{l neighborhood of } \nu^*$}
\State $\nu_i \leftarrow \nu_i + \mu f(\sigma, l)|(g, \nu_i)|$ \hspace{1cm} // adapt winner and its neighbors
\EndFor
\EndFor
\State $\sigma \leftarrow (1 - \alpha) \cdot \sigma$ \hspace{1cm} // decrease the learning gain
\Until{$\text{error} \leq \delta$}
\end{algorithmic}
\end{algorithm}
## Parameters of the adaption

Parameters providing good results in practice.

*Stability of the convergence and quality of solution.*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>the number of goals ($N =</td>
</tr>
<tr>
<td>$M$</td>
<td>$2.5N$ the number of neurons</td>
</tr>
<tr>
<td>$\sigma_0$</td>
<td>$12.41N + 0.06$ the initial learning gain</td>
</tr>
<tr>
<td>$\mu$</td>
<td>$0.6$ the learning rate</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>$0.1$ the gain decreasing rate</td>
</tr>
<tr>
<td>$\delta$</td>
<td>$0.001$ the minimal required distance of the winner from the goal</td>
</tr>
</tbody>
</table>

$\sigma_0$ depends on problem size (regarding quality of solution). This is a linear regression model that has been found experimentally.

$I.e.$, move $\nu$ towards $g$ about 60% of their distance at maximum.

The gain is decreased after each epoch, $\sigma = (1 - \alpha)\sigma$.
Selected Modifications for the Euclidean TSP

• Dynamical creation of neurons (duplication/deletion)

  \[ f(\sigma, l) = \beta j \mu e^{-\left(l/(\gamma_j \sigma)\right)^2}, \]

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

  B. Angéniol et al. (1988)

• Reducing topological defects using multiple scale neighborhood functions (\( \beta_j, \gamma_j \) parameters)


• 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)
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 parameters for a better convergence.

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


- $\mu_k = \frac{1}{\sqrt[k]{4}}$
- $\sigma_k = (1 - 0.01k)\sigma_{k-1}$
- $\sigma_0 = 10$
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 parameters for a better convergence.

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


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,


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

E. M. Cochrane and J. E. Beasley,
*The co-adaptive neural network approach to the Euclidean travelling salesman problem,*

*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 \( \sigma \).*
- Near-tour to tour construction
  *An alternative for the inhibition mechanism.*
  *Keeping the best found tour during learning.*
SOM for the MTSP-MinMax


- A ring for each salesman
  \[ M = 2.5N/k, \ k \ \text{no. of salesmen} \]
- Common depot
- A winner from each ring is adapted towards depot.
- Then, nodes are adapted towards other goals.
- MinMax criterion
  \[ \nu^* = \arg \min_{\nu} |S(\nu, g)| \cdot \left(1 + \frac{\text{dist}_{\nu} - \text{avg}}{\text{avg}}\right), \tag{1} \]
  where \( \text{dist}_{\nu} \) is length of the ring in which \( \nu \) is, and \( \text{avg} \) is the average length of the rings.
MTSP-MinMax visualization of SOM evolution

step 36

step 41

step 45

step 51

step 63

step 74
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
    \( \mathcal{W} \)
  • 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
    \( \text{The polygonal domain} \ \mathcal{W} \)
  • 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 $\nu$ 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 $\nu$ 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 $\nu$ 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 $\nu$ 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 $\nu$ toward $g$ along $S(\nu, g)$
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 $\nu$ 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.
  \[ MTSP \rightarrow TSP \text{ provides degenerative solutions} \]
- For the TSP we need to resolve \textit{neuron–goal} distance and path queries.
  Single point queries, goals are fixed.
- For the MTSP-MinMax we need to resolve \textit{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.

\textit{This difficulty has been noted by several authors.}
SOM for a Graph Input


- 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.

Paths are provided in units of $\mu$s.

The approximation is enabling technique for applying SOM 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. 
   \textit{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.

Paths are provided in units of $\mu$s.

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

Jan Faigl, 2013
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.


Similar approximation also works for two-points path query.
SOM for the MTSP-MinMax in $\mathcal{W}$

step 17

step 30

step 40

step 46

step 52

step 55

step 60

step 79
Example of MTSP-MinMax Solutions

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.
  \[ T. \text{ Yamakawa et al. (2006)} \]
- 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.
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.  
  \textit{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.

\textit{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 \textit{d}-WRP

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

\begin{verbatim}
$\mathbf{r} \leftarrow \text{initialization}$ // create a ring of nodes
repeat
\begin{itemize}
\item $\mathbf{l} \leftarrow \emptyset$ // a set of inhibited nodes
\item $\mathbf{T}_c \leftarrow \text{triangles covered by the current ring } \mathbf{r}$
\item $\Pi(\mathbf{T}) \leftarrow \text{create a random permutation of triangles}$
\item \textbf{foreach} $\mathbf{T} \in \Pi(\mathbf{T})$ \textbf{do}
\begin{itemize}
\item \textbf{if} $\mathbf{T} \notin \mathbf{T}_c$ \textbf{then}
\begin{itemize}
\item $p_a \leftarrow \text{centroid}(\mathbf{T})$ // attraction point
\item $\nu^* \leftarrow \text{select winner node to } p_a, \nu^* \notin \mathbf{l}$
\item $\mathcal{P}_c \leftarrow \{\text{all associated convex polygons to } \mathbf{T}\}$
\item \textbf{if} $\nu^* \notin \mathcal{P}, \mathcal{P} \in \mathcal{P}_c$ \textbf{then}
\begin{itemize}
\item \text{adapt($\nu^*, p_a$)}
\end{itemize}
\item $\mathbf{T}_c \leftarrow \mathbf{T}_c \cup \{\mathbf{T} \mid \mathbf{T} \in \mathcal{P}, \mathcal{P} \in \mathcal{P}_c\}$
\item $\mathbf{l} \leftarrow \mathbf{l} \cup \{\nu^*\}$ // inhibit winner node
\end{itemize}
\end{itemize}
\end{itemize}
\item $G \leftarrow (1 - \alpha) \cdot G$ // decrease the gain
\end{verbatim}

\textbf{until} all triangles are covered by the current ring
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.

Coverage from the ring is determined during the adaptation.

Jan Faigl,

*Approximate Solution of the Multiple Watchman Routes Problem with Restricted Visibility Range*,

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 \( W \) 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 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 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$^{\text{ext}}$ – Multi-Goal Motion Planner
  A feasibility study to find patrolling trajectories using SOM with the RRT–Path$^{\text{ext}}$

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).
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:

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 $g$.
   - 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 \( g \).
   - 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:

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 $g$.
   - 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:

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 \( g \).
   - 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 between two segments*

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

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

  $\text{nodes and map vertices}$

- 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 between 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 between two segments

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

- Goal - segments $\{s^g_1, \ldots, s^g_k\}$
- Ring - segments $\{s'_1, \ldots, s'_l\}$

Winner Selection
1. Determine a pair $(s'_i, s^g_j)$ with minimal distance
   two resulting points $p_r \in s'_i$, $p_g \in s^g_j$
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 between two segments**

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

- Goal - segments \( \{s^g_1, \ldots, s^g_k\} \)
- Ring - segments \( \{s^r_1, \ldots, s^r_l\} \)

  nodes and map vertices

- Winner Selection
  1. Determine a pair \((s^r_i, s^g_j)\) with minimal distance
     two resulting points \(p_r \in s^r_i, p_g \in s^g_j\)
  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 $(s^r_i, s^g_j)$
   two resulting points $p_r \in s^r_i$, $p_g \in s^g_j$
   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
   
   - Euclidean pre-selection of $(s_{r_i}^r, s_{g_j}^g)$.
   - Two resulting points $p_r \in s_{r_i}^r$, $p_g \in s_{g_j}^g$
   
   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.

Jan Faigl, 2013
Proposed Adaptation Schema

1. **For each goal** $g$

   **Winner selection**

   regarding $|S(p_r, p_g)|$

   Euclidean pre-selection of $(s^r_i, s^g_j)$.

   *two resulting points* $p_r \in s^r_i$, $p_g \in s^g_j$

   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 \((s^r_i, s^g_j)\).
   
   *two resulting points* \( p_r \in s^r_i, p_g \in s^g_j \)

   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 \((s^r_i, s^g_j)\),
   - Two resulting points \( p_r \in s^r_i, p_g \in s^g_j \)
   - 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.

Jan Faigl, 2013
Proposed Adaptation Schema

1. **For each goal** $g$
   - **Winner selection** regarding $|S(p_r, p_g)|$
     - Euclidean pre-selection of $(s^r_i, s^g_j)$.
     - Two resulting points $p_r \in s^r_i, p_g \in s^g_j$
   - 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.

Jan Faigl, 2013
Proposed Adaptation Schema

1. **For each goal** $g$
   
   **Winner selection**
   
   Winner selection regarding $|S(p_r, p_g)|$
   
   Euclidean pre-selection of $(s^r_i, s^g_j)$.
   
   Two resulting points $p_r \in s^r_i$, $p_g \in s^g_j$
   
   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

- 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

- 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

- 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

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

J. Faigl, L. Přeučil

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
  \textit{Multi-Goal Path Planning} \sim \textit{TSP}

• Precise navigation to the goals

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.

\textit{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

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
    \[ A_{i+1} = R_i^T M_i R_i A_i R_i^T M_i R_i + R_i^T S_i R_i, \]
    where
    \[ M_i = \begin{bmatrix} 1 & 0 \\ 0 & m(a_i, a_{i+1}, \mathcal{M}) \end{bmatrix}, ~ S_i = \begin{bmatrix} s_i \eta^2 & 0 \\ 0 & \tau^2 \end{bmatrix} \]
    \[ m(a_i, a_{i+1}, \mathcal{M}) \text{ - model of the visible landmarks} \]
    \[ \eta, \tau \sim \text{“odometry and heading error” (variances)} \]
    \[ s_i = |(a_i, a_{i+1})| \text{ - the segment length} \]

Jan Faigl, 2013
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

\[
A_{i+1} = R_i^T M_i R_i A_i R_i^T M_i R_i + R_i^T S_i R_i,
\]

where

\[
M_i = \begin{bmatrix}
1 & 0 \\
0 & m(a_i, a_{i+1}, \mathcal{M})
\end{bmatrix},
S_i = \begin{bmatrix}
{s_i \eta^2} & 0 \\
0 & {\tau^2}
\end{bmatrix}
\]

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

\[
\eta, \tau \sim \text{“odometry and heading error” (variances)}
\]

\[
s_i = |(a_i, a_{i+1})| - \text{the segment length}
\]
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) \(\sim 1 \text{ FPS}\)
GPU
- NVS 320 \(\sim 25 \text{ FPS}\)
- ION \(\sim 15 \text{ FPS}\)
FPGA (Virtex 5) \(\sim 10 \text{ FPS}, 9 \text{ W}\)

Stability theoretically and experimentally proven

Krajník T., Faigl J., Vonásek V., Košnar K., Kulich M., Přeučil L.,
*Simple yet stable bearing-only navigation*,
Principle of Localization Uncertainty Decreasing

The stability of the navigation is based on bearing corrections.

- Increased uncertainty in longitudinal direction
- Position uncertainty

- Position uncertainty
- Auxiliary navigation waypoint
- Selected perimeter

*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

\[ w = \arg\min_{w_i \in W} \| A_{w_i,g} \| \]

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

\[ w = \arg\min_{w_i \in W} \| A_{w_i}g \| \]

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.

$L=416\text{ m}, \ E_{\text{max}}=1.23\text{ m}$

$L=425\text{ m}, \ E_{\text{max}}=0.7\text{ m}$

$E_{\text{max}}$ - expected error at the goal
Experimental Results (1/2) - Outdoor Environment

- P3AT robot
- City park - traveling on pathways
  
  *several runs*

- Random pedestrians

**Simple TSP**

\[
L = 184 \, \text{m}, \quad E_{\text{avg}} = 0.57, \quad E_{\text{max}} = 0.63
\]

**Proposed approach**

\[
L = 202 \, \text{m}, \quad E_{\text{avg}} = 0.35, \quad E_{\text{max}} = 0.37
\]

- Real overall error at the goals decreased from 0.89 m $\rightarrow$ 0.58 m
  
  *(improvement about 35%)*

Jan Faigl, 2013
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.,
Part VI

Recent, Ongoing and Future Work
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 $\rho$.

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 $m \in M$

Fast visibility queries are the key issue

P. Janoušek, Master’s Thesis, 2013 (to be defended)
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* $\rho$.

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 $m \in M$

Fast visibility queries are the key issue

P. Janoušek, Master’s Thesis, 2013 (to be defended)
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 $\rho$.

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

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

Fast visibility queries are the key issue

P. Janoušek, Master’s Thesis, 2013 (to be defended)
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 $\rho$.

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 $m \in M$

Fast visibility queries are the key issue

P. Janoušek, Master’s Thesis, 2013 (to be defended)
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* $\rho$.

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 $m \in M$

Fast visibility queries are the key issue

P. Janoušek, Master’s Thesis, 2013 (to be defended)
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* \( \rho \).*

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 \( m \in M \)

Fast visibility queries are the key issue

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

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 \( \rho \).

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 \( m \in M \)

Fast visibility queries are the key issue

P. Janoušek, Master’s Thesis, 2013 (to be defended)
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
  • Considering time domain
  • Considering autonomous navigation (sources of localization uncertainties)

kinematic or kinodynamic constraints

spatio-temporal spaces

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