Real Path Planning based on Genetic Algorithm and Voronoi Diagrams

Facundo Benavides

MINA - Instituto de Computación
Facultad de Ingeniería - Universidad de la República
Montevideo - Uruguay

IX LARS, 2011
Overview

1. Motivation
   - Background
   - Introduction
   - The path planning problem
   - Previous work

2. Our proposal
   - Implementation
   - Results
1 Motivation
   - Background
   - Introduction
   - The path planning problem
   - Previous work

2 Our proposal
   - Implementation
   - Results
Teleoperation
The man has the control.

- A solution for a history moment.

- After that ...
  - Human dependence.
  - Cognitive fatigue.
  - Communication delay.
  - Broken pipe.
  - Others.
Teleoperation
The man has the control.

- A solution for a history moment.
- After that ...
  - Human dependence.
  - Cognitive fatigue.
  - Communication delay.
  - Broken pipe.
  - Others.
Overview

1 Motivation
   - Background
   - Introduction
   - The path planning problem
   - Previous work

2 Our proposal
   - Implementation
   - Results
Autonomy and Intelligence
Can the machines take control?

- **Autonomy**
  - “A robotic agent ability to perform a set of tasks in a given environment without the need of human intervention to achieve their objectives”. [1]
  - Depending on the environment nature the agent may or may not require "intelligence".

- **Intelligence**
  - “Capability that allows an entity or agent to function properly in a given environment”. [1]
  - A high degree of intelligence would allow the agent to adapt to a wider range of environments.
  - It is important to remember that the physical constraints limit the ability to adapt.
Autonomy and Intelligence
Can the machines take control?

- Autonomy
  - “A robotic agent ability to perform a set of tasks in a given environment without the need of human intervention to achieve their objectives”. [1]
  - Depending on the environment nature the agent may or may not require "intelligence".

- Intelligence
  - “Capability that allows an entity or agent to function properly in a given environment”. [1]
  - A high degree of intelligence would allow the agent to adapt to a wider range of environments.
  - It is important to remember that the physical constraints limit the ability to adapt.
Motivation

Our proposal

Summary

Background

Introduction

Motion Planning

Previous work

Autonomy and Intelligence

Can the machines take control?

- Autonomy
  - “A robotic agent ability to perform a set of tasks in a given environment without the need of human intervention to achieve their objectives”. [1]
  - Depending on the environment nature the agent may or may not require "intelligence".

- Intelligence
  - “Capability that allows an entity or agent to function properly in a given environment”. [1]
  - A high degree of intelligence would allow the agent to adapt to a wider range of environments.
  - It is important to remember that the physical constraints limit the ability to adapt.
Autonomy and Intelligence
Can the machines take control?

- **Autonomy**
  - “A robotic agent ability to perform a set of tasks in a given environment without the need of human intervention to achieve their objectives”. [1]
  - Depending on the environment nature the agent may or may not require "intelligence".

- **Intelligence**
  - “Capability that allows an entity or agent to function properly in a given environment”. [1]
  - A high degree of intelligence would allow the agent to adapt to a wider range of environments.
  - It is important to remember that the physical constraints limit the ability to adapt.
Autonomy and Intelligence
Can the machines take control?

- Autonomy
  - “A robotic agent ability to perform a set of tasks in a given environment without the need of human intervention to achieve their objectives”. [1]
  - Depending on the environment nature the agent may or may not require "intelligence".

- Intelligence
  - “Capability that allows an entity or agent to function properly in a given environment”. [1]
  - A high degree of intelligence would allow the agent to adapt to a wider range of environments.
  - It is important to remember that the physical constraints limit the ability to adapt.
Environments
The robot like a situated agent.

The environments can be classified according to the following dimensions or perspectives [2]:
- Completely or partially observable.
- Deterministic or stochastic.
- Sequential or episodic.
- Dynamic or static.
- Discrete or continuous.
- Mono or multi agent.
Motion planning
How to move in a dynamic and unknown world?

The least one can expect from an autonomous robot is to have the ability to plan their own movements. [3]

- How to achieve a target destination?
  - Navigation.

- Where I am?
  - Localization.

- Where have I been?
  - Mapping.
Motion planning
How to move in a dynamic and unknown world?

The least one can expect from an autonomous robot is to have the ability to plan their own movements. [3]

- How to achieve a target destination?
  - Navigation.

- Where I am?
  - Localization.

- Where have I been?
  - Mapping.
The least one can expect from an autonomous robot is to have the ability to plan their own movements. [3]

- How to achieve a target destination?
  - Navigation.
- Where I am?
  - Localization.
- Where have I been?
  - Mapping.
Overview

1. Motivation
   - Background
   - Introduction
   - The path planning problem
   - Previous work

2. Our proposal
   - Implementation
   - Results
Approaches
Resolution methods [3]

- **Roadmap**
  - Consists in capturing the connectivity of free space on a network of roads.

- **Cell Decomposition**
  - Possibly the most studied family of methods.
  - Consist of decomposing the free space into cells. It assumes that the intra-cell navigation is easy to solve and build a graph to represent inter-cell connectivity.

- **Potential Field**
  - It consists of discretizing the entire configuration space using a regular grid of small cells.
  - Then move the robot assuming that in each cell it is a particle under different artificial potential fields.
Approaches

Resolution methods [3]

- **Roadmap**
  - Consists in capturing the connectivity of free space on a network of roads.

- **Cell Decomposition**
  - Possibly the most studied family of methods.
  - Consist of decomposing the free space into cells. It assumes that the intra-cell navigation is easy to solve and build a graph to represent inter-cell connectivity.

- **Potential Field**
  - It consists of discretizing the entire configuration space using a regular grid of small cells.
  - Then move the robot assuming that in each cell it is a particle under different artificial potential fields.
**Approaches**

**Resolution methods [3]**

- **Roadmap**
  - Consists in capturing the connectivity of free space on a network of roads.

- **Cell Decomposition**
  - Possibly the most studied family of methods.
  - Consist of decomposing the free space into cells. It assumes that the intra-cell navigation is easy to solve and build a graph to represent inter-cell connectivity.

- **Potential Field**
  - It consists of discretizing the entire configuration space using a regular grid of small cells.
  - Then move the robot assuming that in each cell it is a particle under different artificial potential fields.
Path planning
Three parts of a single problem.

- Path planning.
  - Solve a free path between the initial state in which the robot is and a destination state under certain evaluation criteria.
  - The most used are:
    - Distance, Security, Energy consumption, Softness.

- Trajectory planning.
  - Takes into account the time variable.
  - Low level control.
  - Resolves monitoring and following of predetermined paths in the previous steps.
Path planning
Three parts of a single problem.

- **Path planning.**
  - Solve a free path between the initial state in which the robot is and a destination state under certain evaluation criteria.
  - The most used are:
    - Distance, Security, Energy consumption, Softness.

- **Trajectory planning.**
  - Takes into account the time variable.
  - Low level control.
    - Resolves monitoring and following of predetermined paths in the previous steps.
Overview

1 Motivation
   - Background
   - Introduction
   - The path planning problem
   - Previous work

2 Our proposal
   - Implementation
   - Results
Approaches


- Path Planning based on AG: From a representation of the environment called Digital Potential Fields, apply a GA to improve the length, safety and smoothness of the paths. [5]
Approaches


- Path Planning based on AG: From a representation of the environment called Digital Potential Fields, apply a GA to improve the length, safety and smoothness of the paths. [5]
Two properties of interest in the context of path planning for mobile robots:

1. Every Voronoi edge belongs to the bisector of the line segment formed by the two generating points of the regions that determine the edge.

2. Every Voronoi vertex is located exactly in the center of the circle whose perimeter generating points are the regions that determine the vertex.
Overview

1 Motivation
   - Background
   - Introduction
   - The path planning problem
   - Previous work

2 Our proposal
   - Implementation
   - Results
World model

- Completely observable, static, deterministic, sequential, discrete and single-agent.
- Regular grid, free cells or obstacles.
- Voronoi diagram generated by the set of obstacles.
- Particle like Robot. [6]
World model

- Completely observable, static, deterministic, sequential, discrete and single-agent.

- Regular grid, free cells or obstacles.

- Voronoi diagram generated by the set of obstacles.

- Particle like Robot. [6]
World model

- Completely observable, static, deterministic, sequential, discrete and single-agent.
- Regular grid, free cells or obstacles.
- Voronoi diagram generated by the set of obstacles.
- Particle like Robot. [6]
World model

- Completely observable, static, deterministic, sequential, discrete and single-agent.
- Regular grid, free cells or obstacles.
- Voronoi diagram generated by the set of obstacles.
- Particle like Robot. [6]
Chromosomes: each path is encoded as a sequence of free cells of variable length.

Initial population:
- Generate the DV from obstacles in the environment.
- Identify the regions that contain the start and end position.
- Dijkstra for all combinations of points of departure and arrival.
- Randomly selection of paths and cells.

Fitness function:
- Feasible, length, smoothness and security.
- Infeasible: length, rate and degree of infeasibility.
Motivation

Our proposal

Summary

Our proposal

Results

GA application I

- Chromosomes: each path is encoded as a sequence of free cells of variable length.

- Initial population:
  - Generate the DV from obstacles in the environment.
  - Identify the regions that contain the start and end position.
  - Dijkstra for all combinations of points of departure and arrival.
  - Randomly selection of paths and cells.

- Fitness function:
  - Feasible, length, smoothness and security.
  - Infeasible: length, rate and degree of infeasibility.
Chromosomes: each path is encoded as a sequence of free cells of variable length.

Initial population:
- Generate the DV from obstacles in the environment.
- Identify the regions that contain the start and end position.
- Dijkstra for all combinations of points of departure and arrival.
- Randomly selection of paths and cells.

**Fitness function:**
- Feasible, length, smoothness and security.
- Infeasible: length, rate and degree of infeasibility.
GA application II

- Genetic Operators:
<table>
<thead>
<tr>
<th>Platform I</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulated</td>
</tr>
<tr>
<td><strong>CPU</strong></td>
</tr>
<tr>
<td><strong>RAM</strong></td>
</tr>
<tr>
<td><strong>Swap</strong></td>
</tr>
<tr>
<td><strong>Disk</strong></td>
</tr>
<tr>
<td>Real</td>
</tr>
<tr>
<td><strong>Arena</strong></td>
</tr>
<tr>
<td><strong>Vision</strong></td>
</tr>
<tr>
<td><strong>Robot</strong></td>
</tr>
</tbody>
</table>

**Cuadro:** Hardware platform.
### Platform II

<table>
<thead>
<tr>
<th>OS</th>
<th>GNU/Linux - Xubuntu 2.6.31-22</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDE</td>
<td>Eclipse Helios 3.6</td>
</tr>
<tr>
<td>Language</td>
<td>Java 1.6</td>
</tr>
<tr>
<td>AG Library</td>
<td>JGAP 3.3.3</td>
</tr>
<tr>
<td>Graph Library</td>
<td>JGrapht 0.8.1</td>
</tr>
<tr>
<td>Voronoi Library</td>
<td>Quickhull3d 1.4</td>
</tr>
<tr>
<td>Viewer</td>
<td>Processing 1.0.7</td>
</tr>
</tbody>
</table>

**Cuadro:** Software platform.
Overview

1 Motivation
   - Background
   - Introduction
   - The path planning problem
   - Previous work

2 Our proposal
   - Implementation
   - Results
Test environment
Facundo Benavides

Real Path Planning: A GA & VD based aproach.
Standard deviation
Generations

![Graph showing generations vs. envs](image-url)
Facundo Benavides
Real Path Planning: A GA & VD based approach.
## Numerical comparison

<table>
<thead>
<tr>
<th></th>
<th>GA over DPF</th>
<th>GA over VD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environment generation (ms)</td>
<td>1632</td>
<td>1217</td>
</tr>
<tr>
<td>GA Evolution time (ms)</td>
<td>275.79</td>
<td>1299</td>
</tr>
<tr>
<td>Average number of generations</td>
<td>12.51</td>
<td>17.09</td>
</tr>
<tr>
<td>Average Fitness</td>
<td>0.82990</td>
<td>0.84669</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>0.01067</td>
<td>0.00567</td>
</tr>
<tr>
<td>Best Fitness</td>
<td>0.84583</td>
<td>0.85459</td>
</tr>
<tr>
<td>Deviation from the fittest</td>
<td>0.01267</td>
<td>0.00790</td>
</tr>
<tr>
<td>Average length</td>
<td>656.27</td>
<td>657.63</td>
</tr>
<tr>
<td>Average safety</td>
<td>253.58</td>
<td>280.46</td>
</tr>
<tr>
<td>Average smoothness</td>
<td>0.29111</td>
<td>0.21318</td>
</tr>
</tbody>
</table>
### Numerical comparison II

<table>
<thead>
<tr>
<th></th>
<th>AG over DV</th>
<th>Parameters of DPF</th>
<th>Parameters ad-hoc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environment generation (ms)</td>
<td>1305</td>
<td>1136</td>
<td></td>
</tr>
<tr>
<td>GA Evolution time (ms)</td>
<td>223</td>
<td>265</td>
<td></td>
</tr>
<tr>
<td>Average number of generations</td>
<td>15</td>
<td>15.45</td>
<td></td>
</tr>
<tr>
<td><strong>Average Fitness</strong></td>
<td>0.83437</td>
<td>0.83720</td>
<td></td>
</tr>
<tr>
<td>Standard deviation</td>
<td>0.01376</td>
<td>0.01169</td>
<td></td>
</tr>
<tr>
<td><strong>Best Fitness</strong></td>
<td>0.85433</td>
<td>0.85433</td>
<td></td>
</tr>
<tr>
<td>Deviation from the fittest</td>
<td>0.01996</td>
<td>0.01714</td>
<td></td>
</tr>
<tr>
<td>Average length</td>
<td>662.44</td>
<td>661.07</td>
<td></td>
</tr>
<tr>
<td>Average safety</td>
<td>291.19</td>
<td>287.55</td>
<td></td>
</tr>
<tr>
<td>Average smoothness</td>
<td>0.29058</td>
<td>0.27583</td>
<td></td>
</tr>
</tbody>
</table>
Visual comparison
Multiple tasks execution time

![Graph showing multiple tasks execution time](image_url)
Conclusion

- Feasibility from the methodological point of view and in practical applications with real robots.
- Significant improvement in performance maintaining the quality of the solutions reached.
Conclusion

- Feasibility from the methodological point of view and in practical applications with real robots.
- Significant improvement in performance maintaining the quality of the solutions reached.
Future works

- Generalize VD library to support the modeling of two-dimensional bodies.
- Consider the dynamic aspect.
Future works

- Generalize VD library to support the modeling of two-dimensional bodies.
- Consider the dynamic aspect.
Acknowledgment

- Co-Authors
  - Gonzalo Tejera
  - Martín Pedemonte
  - Serrana Casella

- Colleagues
  - Eduardo Grampín
  - Federico Andrade


