Strict Theta*
Shorter Motion Planning
Using Taut Paths

Shunhao Oh
ohoh@u.nus.edu

Hon Wai Leong
leonghw@comp.nus.edu.sg

Department of Computer Science
National University of Singapore
Any-Angle Path Planning

Problem Definition
Any-Angle Path Planning

Problem Definition
Any-Angle Path Planning

Problem Definition

Grid
Any-Angle Path Planning

Problem Definition

Blocked Tile
# Any-Angle Path Planning

## Problem Definition

![Diagram](image-url)

- **Unblocked Tile**

The diagram illustrates the concept of any-angle path planning, where the goal is to find a path between any two points on a grid, considering the orientation of the path. The unblocked tiles represent the areas where the path can be directed, indicating flexibility in the angle of the path compared to traditional path planning methods that often require paths to follow specific angles or orientations.
Any-Angle Path Planning

Problem Definition
Any-Angle Path Planning

Problem Definition

Start
Any-Angle Path Planning

Problem Definition
8-Directional Path

Total Length: \(4 + 3\sqrt{2} \approx 8.243\)
Any-Angle Path

Total Length: $2 + \sqrt{13} + \sqrt{5} \approx 7.842$
A* on 8-directional Grids

8-directional A*
A* on 8-directional Grids
A* on 8-directional Grids

8-directional A*
A* on 8-directional Grids

8-directional A*
A* on 8-directional Grids
A* on 8-directional Grids
A* on 8-directional Grids

8-directional path!
Basic Theta*

(Daniel, Nash, Koenig, Felner, 2007)
Theta* Algorithm
Theta* Algorithm

Basic Theta*
Theta* Algorithm

Basic Theta*
Theta* Algorithm

Basic Theta*
Theta* Algorithm

PATH 2

Basic Theta*
Theta* Algorithm

PATH 2 ≤ PATH 1
Theta* Algorithm

Basic Theta*
8-directional A*

Theta*

Basic Theta*
Basic Theta*
Basic Theta*
Basic Theta*
Basic Theta*
Basic Theta*
Basic Theta*
Non-Taut Path
Taut Path

Strict Theta*
Taut Path

Heading change
Taut Path

Strict Theta*
Strict Theta*
Shorter Motion Planning
Using Taut Paths
Strict Theta*
Strict Theta*

Easy to implement
Much shorter paths
Low runtime overhead
Idea:
Restrict the search to Taut Paths
Idea:

“Restrict” the search to Taut Paths

Penalise non-taut paths
Strict Theta*
Is Taut

\[ p \quad u \quad v \]
Is Taut

$p' \xrightarrow{} p\xrightarrow{} u \xrightarrow{} v$
Not Taut

\[ p', p, u, v \]
Not Taut
Add Penalties

$p'$

$p$

$u$

$+1$ $+1$ $+0$

$p'$

$p$

$u$

$+1$ $+1$ $+0$
Tautness checks
Tautness Checks

We need only check one tile.

<table>
<thead>
<tr>
<th>X</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Tautness Checks
We need only check one tile.
Tautness Checks
We need only check one tile.

Cannot be taut!
The Advantage of Taut Path Restriction
Basic Theta*
Basic Theta*
Basic Theta*
Basic Theta*
Basic Theta*
Basic Theta*
Basic Theta*
Basic Theta*
Basic Theta*
Strict Theta*
Strict Theta*
Strict Theta* 

Relaxed with penalty
Strict Theta*
Strict Theta*
Strict Theta*
Strict Theta*
Strict Theta*
Basic Theta*

Strict Theta*
Basic Theta*
Strict Theta*
Recursive Strict Theta*
Recursive Strict Theta*
Recursive Strict Theta*
Recursive Strict Theta*
Recursive Strict Theta*
Path Length
(As a ratio to the optimal)
Path Length
(As a ratio to the optimal)
Path Length

(As a ratio to the optimal)

Theta*
Path Length

(As a ratio to the optimal)

Strict Theta*
Path Length
(As a ratio to the optimal)

Rec. Strict Theta*
Path Length
(As a ratio to the optimal)
Path Length

(As a ratio to the optimal)

1 = Optimal

- Theta*
- StrictTheta*
- R.StrictTheta*
Path Length

(As a ratio to the optimal)

1.0025*Optimal
Path Length
(As a ratio to the optimal)
Path Length

(As a ratio to the optimal)
Path Length

(As a ratio to the optimal)
Path Length
(As a ratio to the optimal)
Running Time
(Averaged, in milliseconds)
Running Time
(in milliseconds)

- Rand 6%
- Rand 20%
- Rand 40%
- Obstacles
- Mazes
- Game Maps

- Theta*
- StrictTheta*
- R.StrictTheta*
Results

Using large randomly generated maps of sizes:

500x500
1000x1000
2000x2000
3000x3000
4000x4000
5000x5000
Path Length
6% Blocked Tiles
Path Length
6% Blocked Tiles

500x500
Path Length
6% Blocked Tiles

Comparison of path lengths for different tile sizes:
- **500x500**
- **5000x5000**

Graph shows the path length compared to different tile sizes and blocking percentages.
Path Length
6% Blocked Tiles
Path Length

20% Blocked Tiles

![Graph showing path length with various tile sizes and blocking percentages.](image)
Path Length
40% Blocked Tiles
Running Time
6% Blocked Tiles
Running Time

20% Blocked Tiles

[Bar chart showing running time for different tile sizes: 500x500, 1000x1000, 2000x2000, 3000x3000, 4000x4000, 5000x5000. The chart includes bars for 'Theta', 'Strict', and 'RecStrict'.]
Running Time

40% Blocked Tiles

![Bar chart showing running time for different tile sizes and blocking strategies. The chart compares Theta, Strict, and RecStrict methods.]
Percentage Optimal / Taut
(Note: Optimal Paths are Always Taut)
Percentage Optimal / Taut

Basic Theta*
Percentage Optimal / Taut

6.7% Optimal
Percentage Optimal / Taut

12% Taut
(optimal paths are taut)
Neither Optimal nor Taut
Percentage Optimal / Taut

Basic Theta*

 kommer världen
Percentage Optimal / Taut

Strict Theta*
Percentage Optimal / Taut

Recursive Strict Theta*
Strict Theta*

Easy to implement
Much shorter paths
Low runtime overhead
Implementation

github.com/Ohohcakester/Any-Angle-Pathfinding
Implementation

github.com/Ohohcakester/Any-Angle-Pathfinding

Google for "Any Angle Pathfinding"
Strict Theta*
Shorter Motion Planning
Using Taut Paths

Shunhao Oh
ohoh@u.nus.edu

Hon Wai Leong
leonghw@comp.nus.edu.sg

Department of Computer Science
National University of Singapore

Implementation:
github.com/Ohohcakester/Any-Angle-Pathfinding
Path Length

[Bar chart showing path length comparisons with different algorithms and blocking levels.]
Running Time
Running Time
6% Blocked Tiles
Running Time
20% Blocked Tiles
Running Time
40% Blocked Tiles
Path Length vs Penalty Value

Penalty Value
% Optimal/Taut vs Penalty Value
% Optimal/Taut vs Penalty Value

- % Taut (Game)
- % Opt (Game)
- % Taut (Gen.)
- % Opt (Gen.)
Let's analyze the diagram step by step. The text in the image contains the word "obstacle." The diagram shows a grid with points labeled as $u$, $v$, and $w$. The distance $L$ is marked on the grid. The point $v$ is located outside the obstacle area, which is shaded in gray. The path from $u$ to $v$ avoids the obstacle, indicating a successful path planning.

The key takeaway is that the path chosen by the algorithm successfully navigates around the obstacle, ensuring a clear route from $u$ to $v$.
Strict Theta*
Strict Theta*
ACKNOWLEDGEMENT

This presentation benefitted from

PowerPointLabs
a PowerPoint plugin for creating
better presentations with less effort.

PowerPointLabs
is available for free at
http://PowerPointLabs.info