Navigation to a human in motion by using points of interest

Tim Nagelkerke

27-06-2016

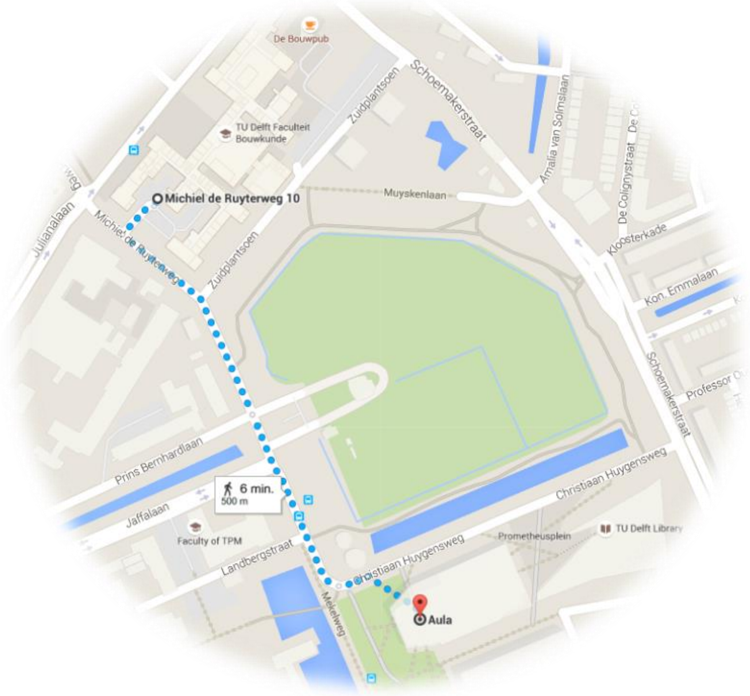
First mentor: Zlatanova, Dr.ing. S.
Second mentor: Diakite, A.A.
Co-reader: Quak, Drs.C.W.
Delegate of the board of examiners: Burg, Ir. L.P.J. van den

TU Delft

Current navigation systems: growing market

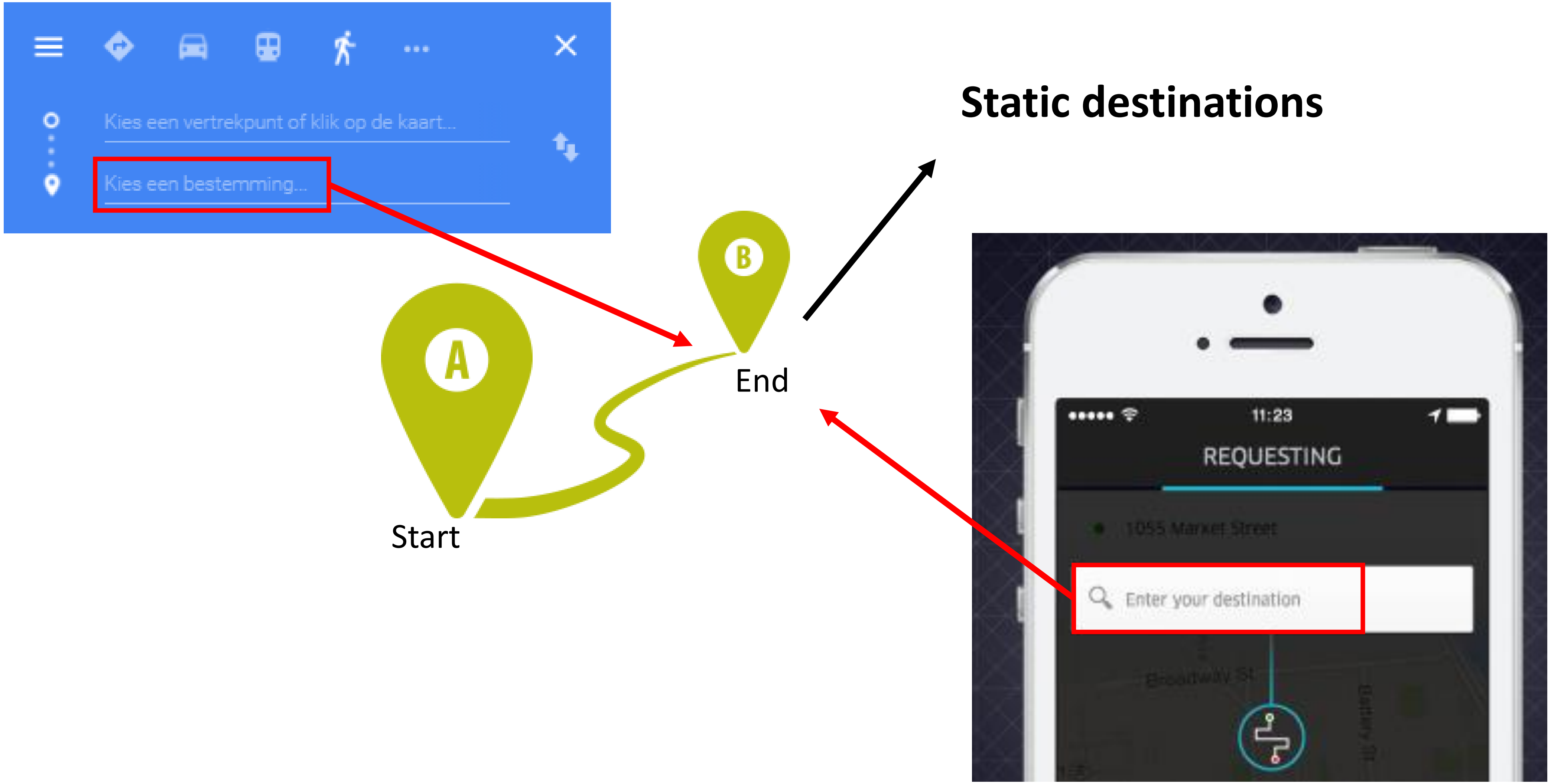
Infsoft

TomTom



Google Maps Directions

What do they have in common?



But how to find a dynamic destination?

- Find a lost child
- Find your partner in a busy environment
- Navigate to a member with dementia
- Navigate to a co-worker during an emergency

Find people that cannot or do not have to verbally communicate with the other person

Outline

Introduction:

- Problem definition
- Research objectives and question
- Research scope

Theory path planning algorithm

Conceptual framework: SEA* method

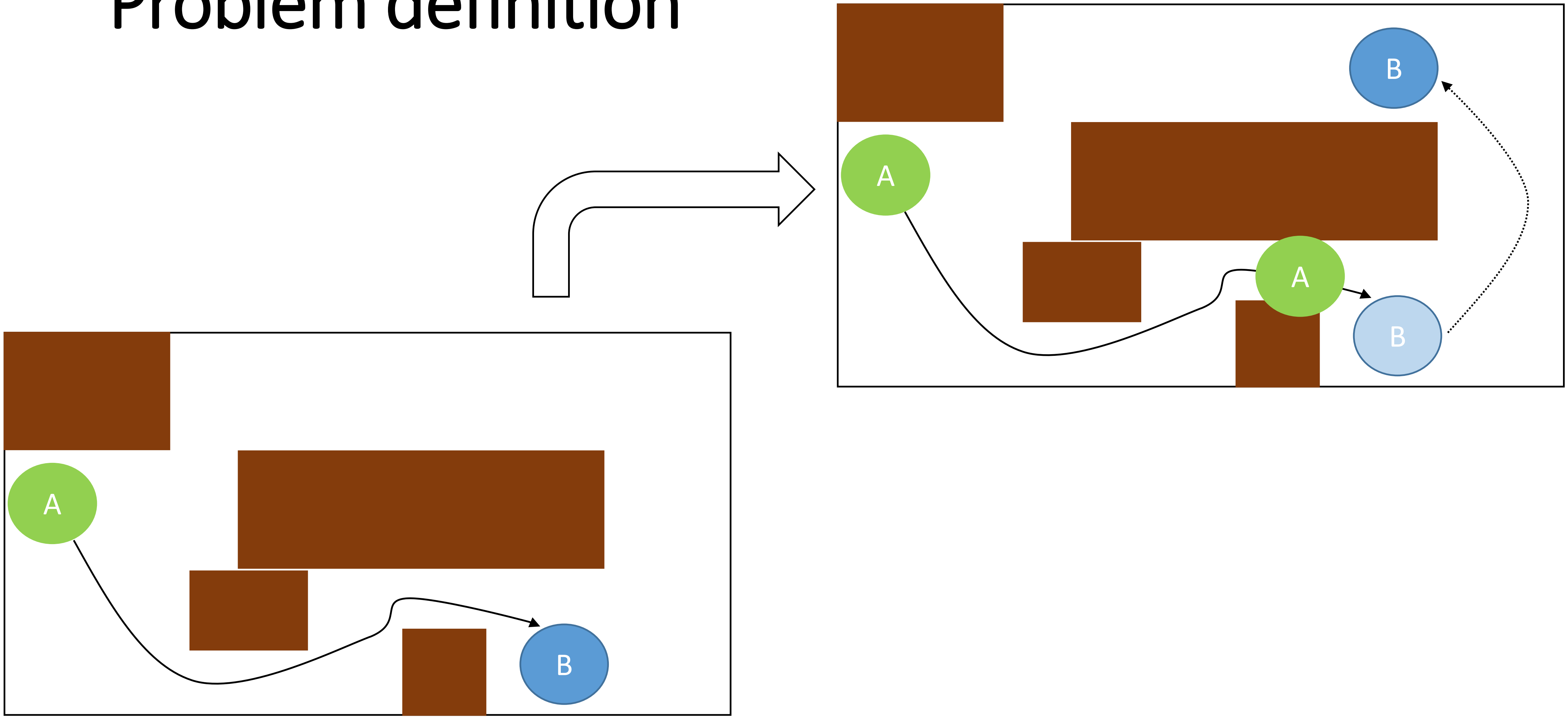
Implementation and testing:

- Indoor implementation
- Outdoor implementation

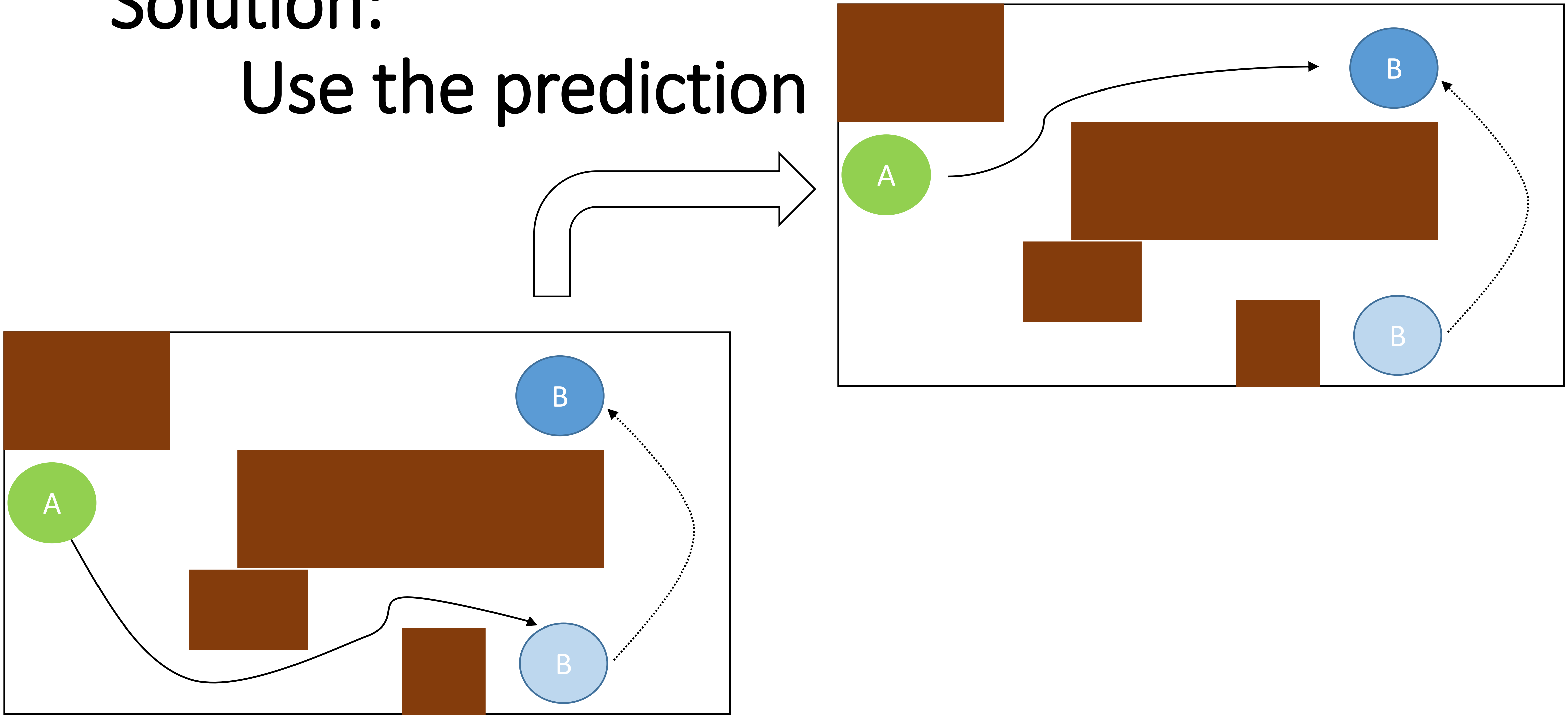
Conclusions

Future work

Problem definition



Solution:
Use the prediction



Research objective

- Investigate methods that supports navigation to a human in motion.
- Design a method to support navigation to a human in motion based on the A* algorithm, the direction of the person and semantics.
- Implement and test the proposed method: Indoor and Outdoor.

Research question

Which defined objects could be used to estimate the predicting location of a moving person to support navigation to a person in motion?

Research scope

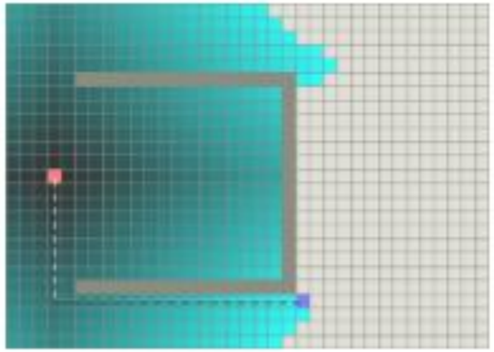
- One person to one person navigation.
- Static environment
- 2D
- No indoor positioning, but tested with GPS data.

Theory: Path planning algorithm

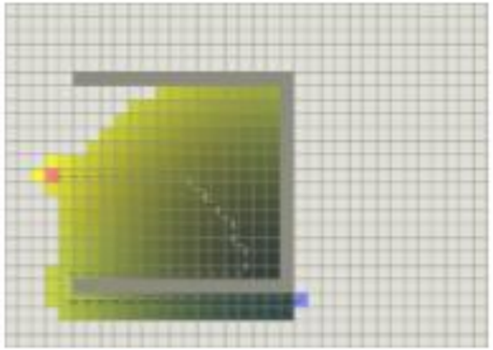
The A* algorithm is chosen because:

- Always finds a path, if there is a path
- Fast algorithm
- Determines the fastest path from A to B

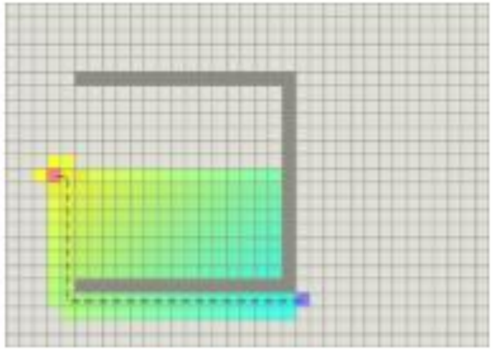
Dijkstra



Greedy best search



A*

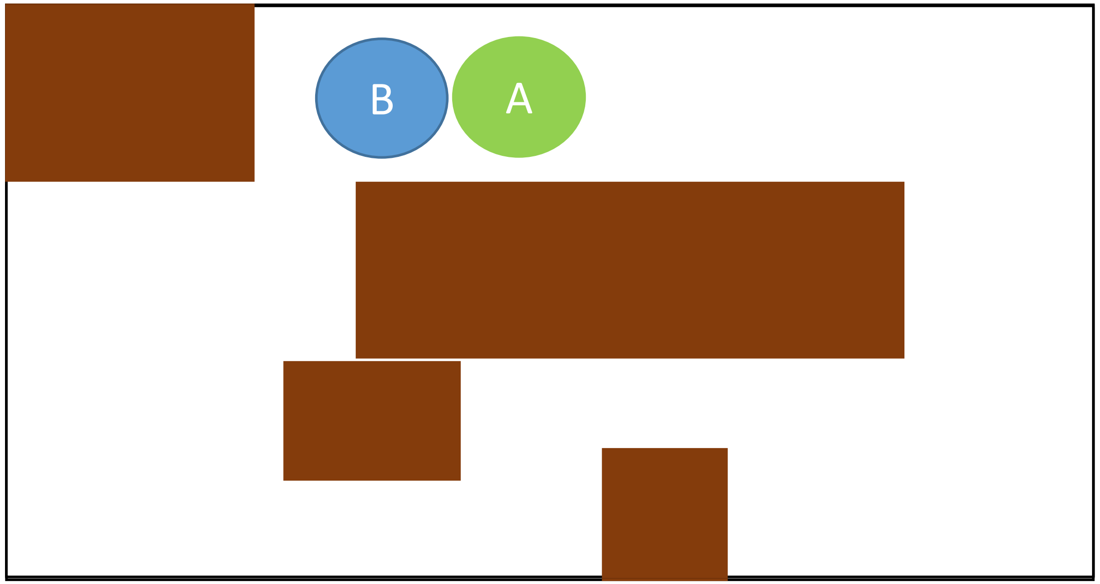


Theory: Path planning algorithm

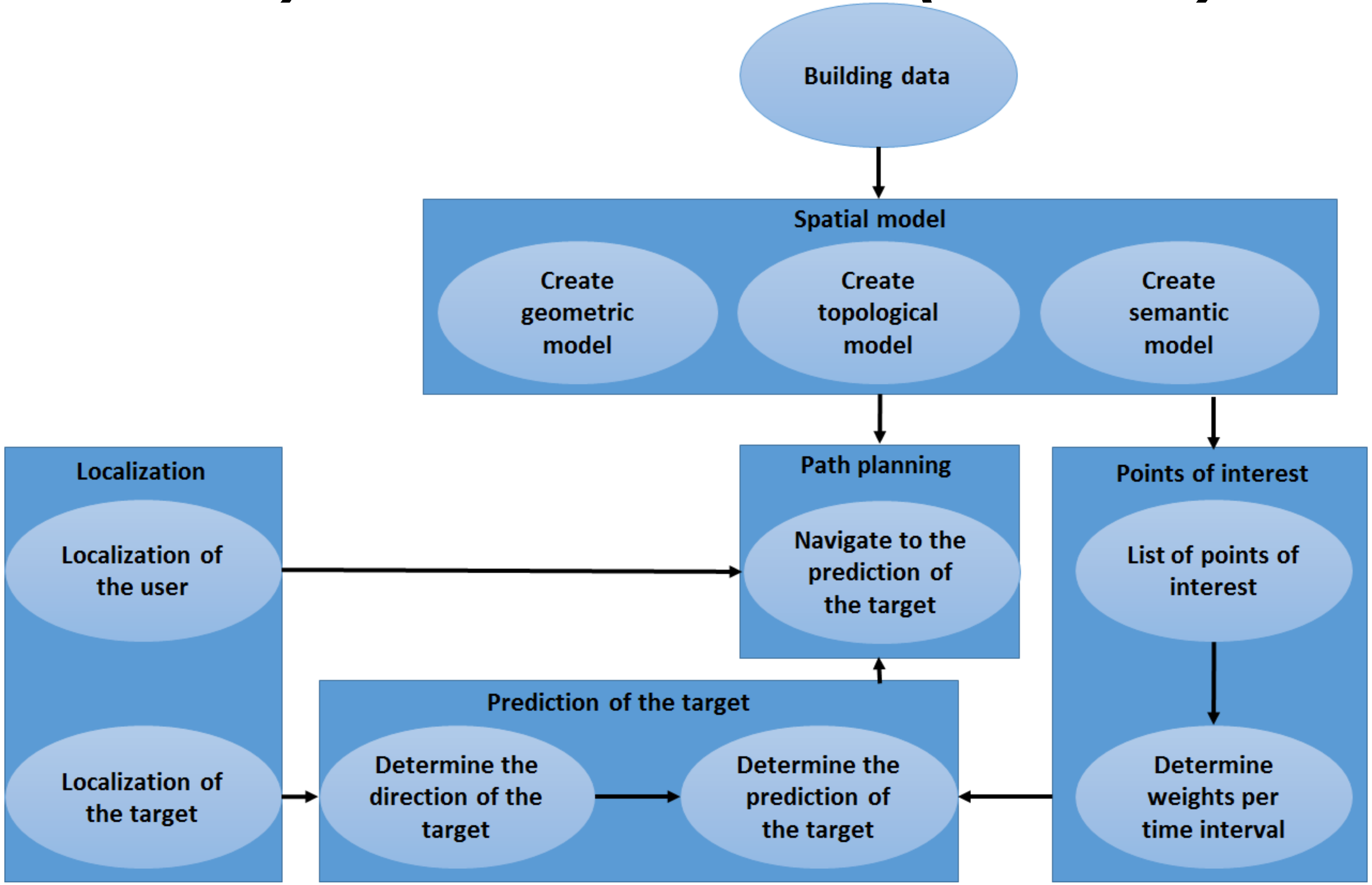
Current path planning algorithm: Incremental A* in robotics and gaming industry

Limitation:

- Following behavior
- Not always find the target, if the target continues moving

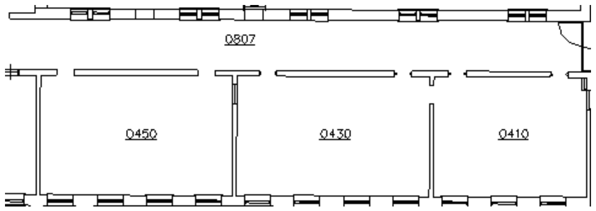


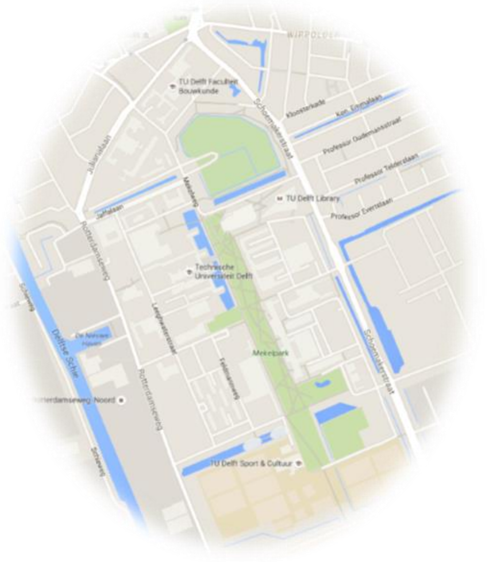
Conceptual framework:
Semantically Enriched A (SEA*) method*



Conceptual framework: Spatial model

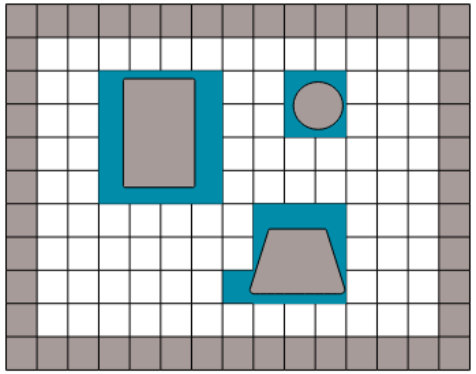
Terrain data

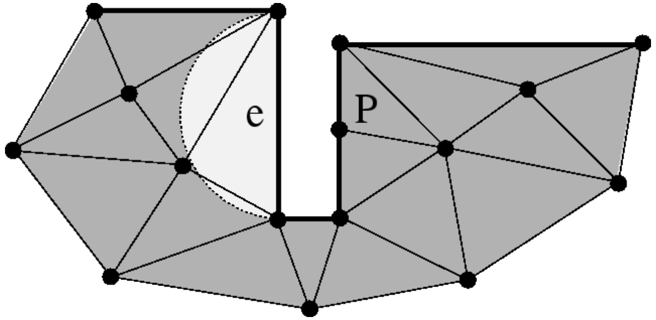




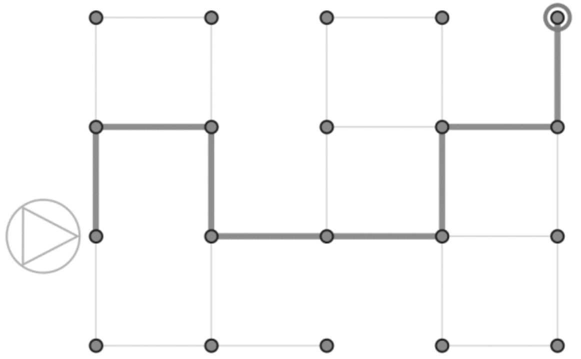
Spatial model:

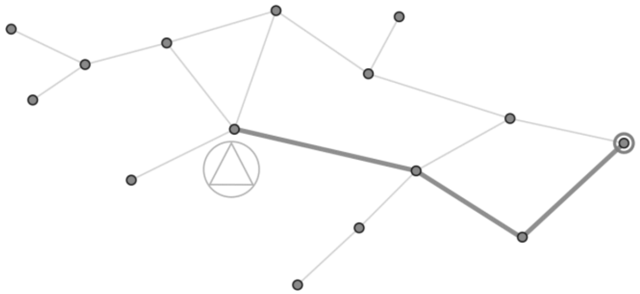
- Geometrical model → positions
- Topological model → relations
- Semantic model → meaning





Navigable graph





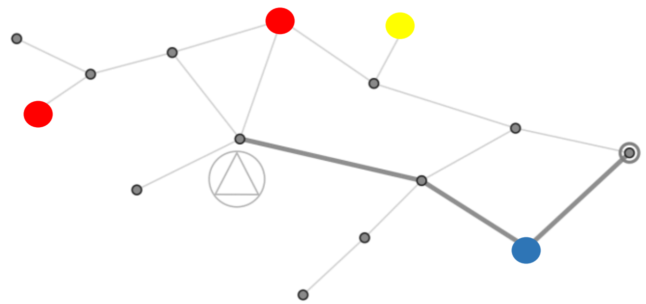
Conceptual framework: Points of interest

Terrain data → Points of interest

Represents possible predictions of the target

Examples indoor are: Rooms, exits, doors, stairs, coffee machines, information desk, …
Examples outdoor are: Buildings, meeting points, squares, park, …

Represented by the node where the person interacts with the location



Conceptual framework: Points of interest

Each point of interest gets a weight per time interval

Determined by external information:

- Time schedule
- Attended events
- Meetings
- Opening hours

Example cafeteria:

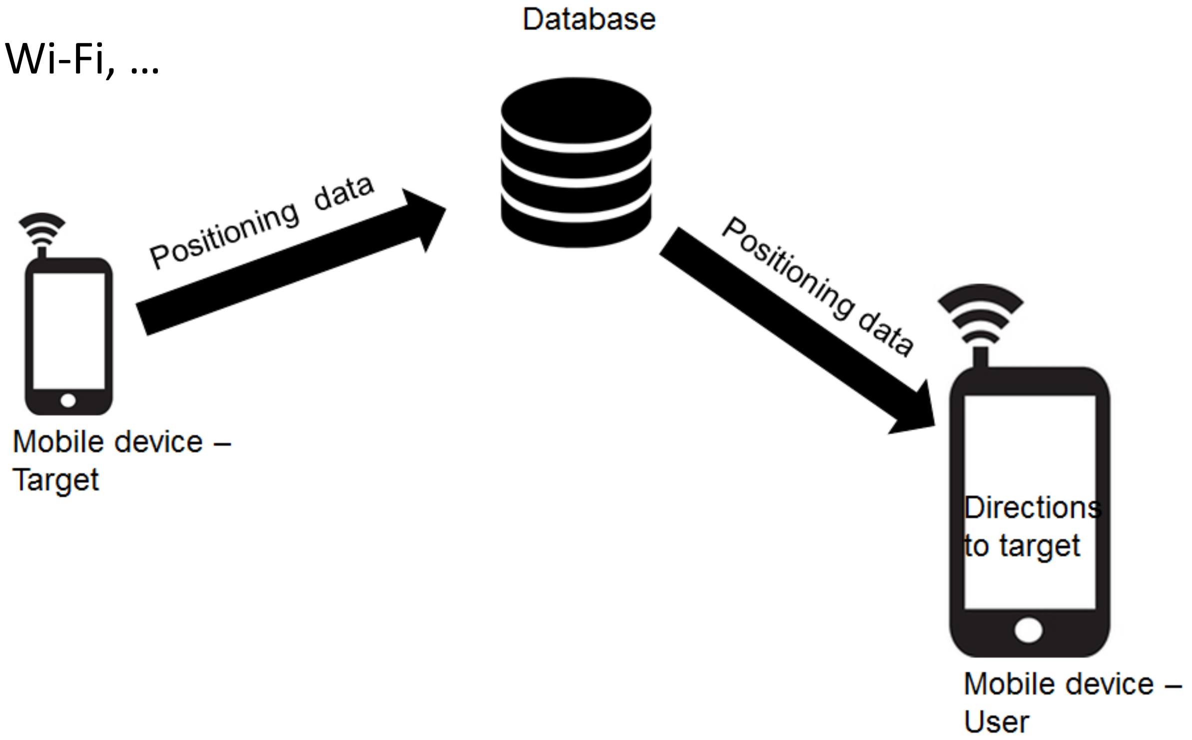
$$Weight(\Delta t) = \begin{cases} 0.1\ for\ 08{:}00 - 11{:}45 \\ 0.7\ for\ 11{:}46 - 13{:}00 \\ 0.1\ for\ 13{:}01 - 18{:}00 \end{cases}$$

Conceptual framework: Localization

Position of the user and the target in real time ⇨ Position in the spatial model

Every ‘n’ seconds

Indoor positioning: RFID, Bluetooth, Wi-Fi, …
Outdoor positioning: GPS



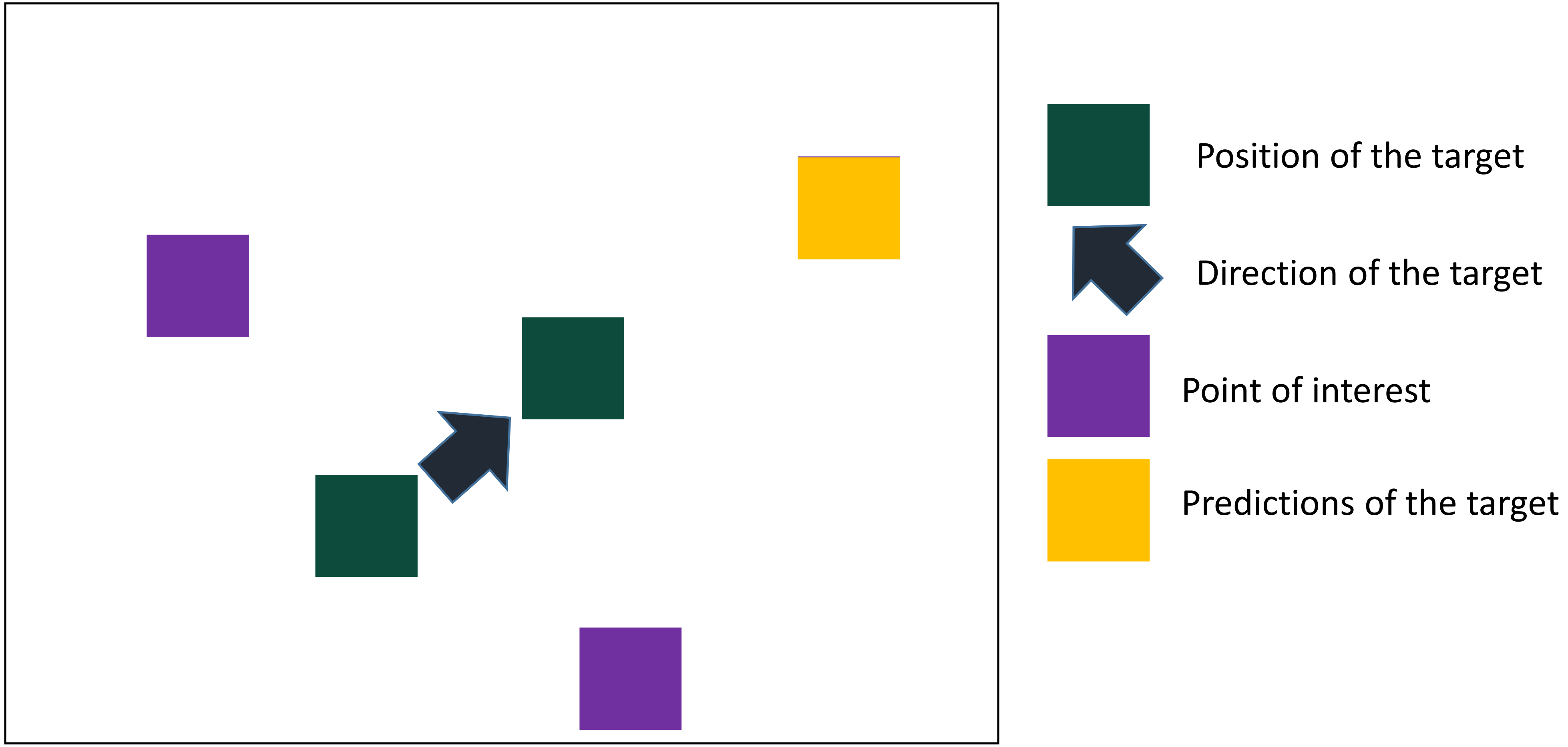
Conceptual framework: Prediction of the target

All points of interest ⇨ One point of interest: The prediction of the target

Use the distance of the path to each point of interest

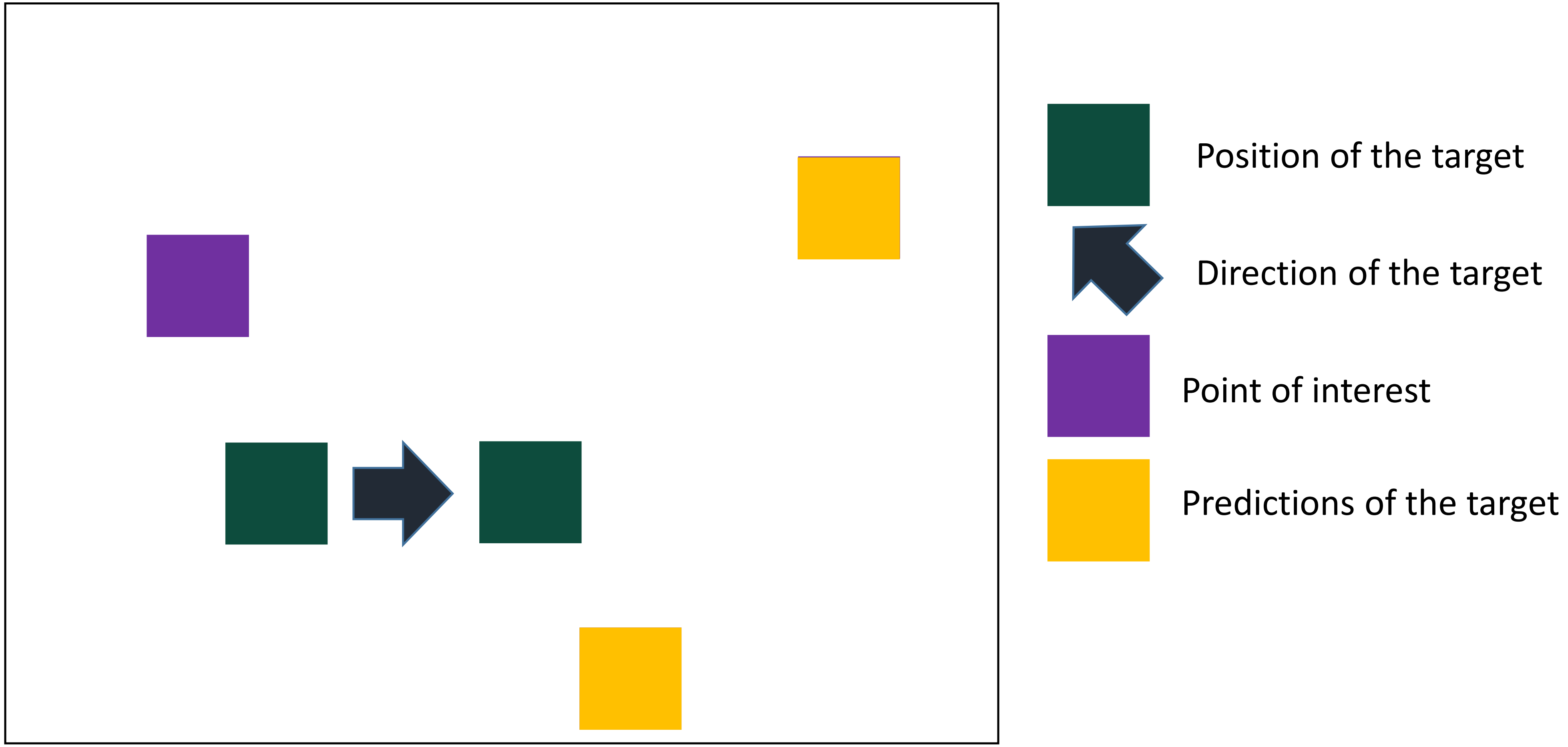
Conceptual framework: Prediction of the target

Use of the distance: one point of interest



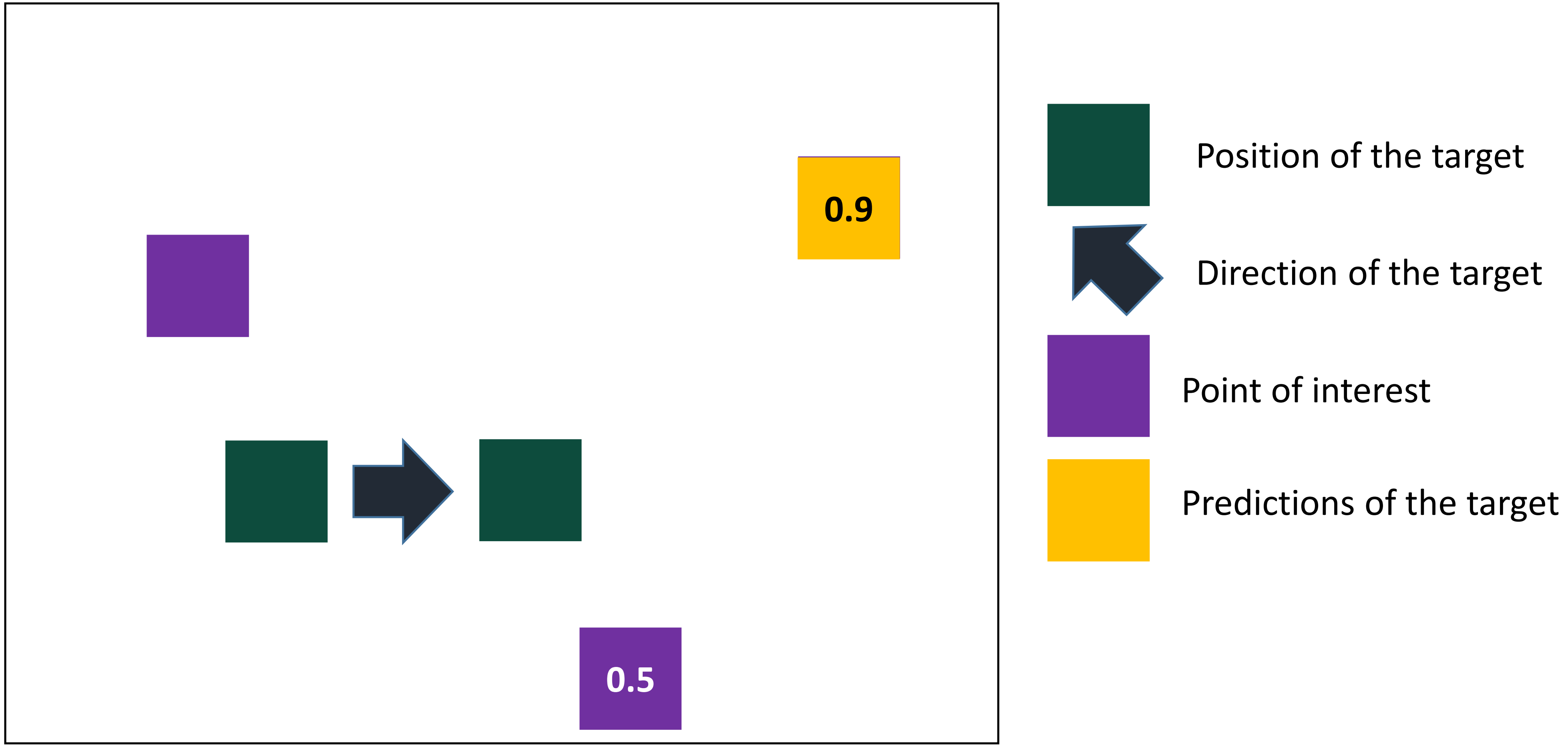
Conceptual framework: Prediction of the target

Use of the distance: multiple points of interest



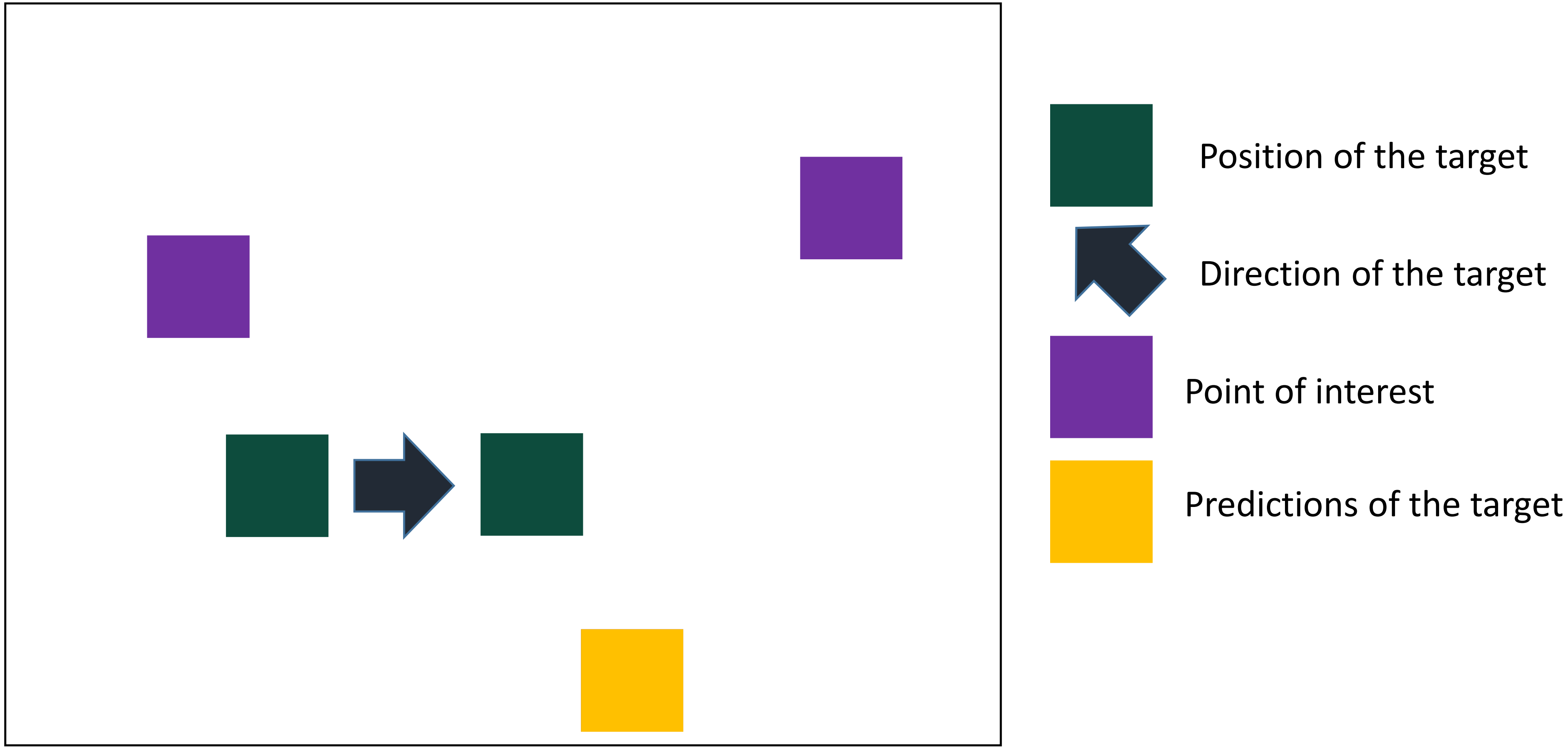
Conceptual framework: Prediction of the target

1. *Multiple points of interest: Using weights*



Conceptual framework: Prediction of the target

2. *Multiple points of interest: Using closest point*



Conceptual framework: Path planning algorithm

A* algorithm

Recalculate the method by every new position of the target.

Navigate to the target or the prediction of the target?

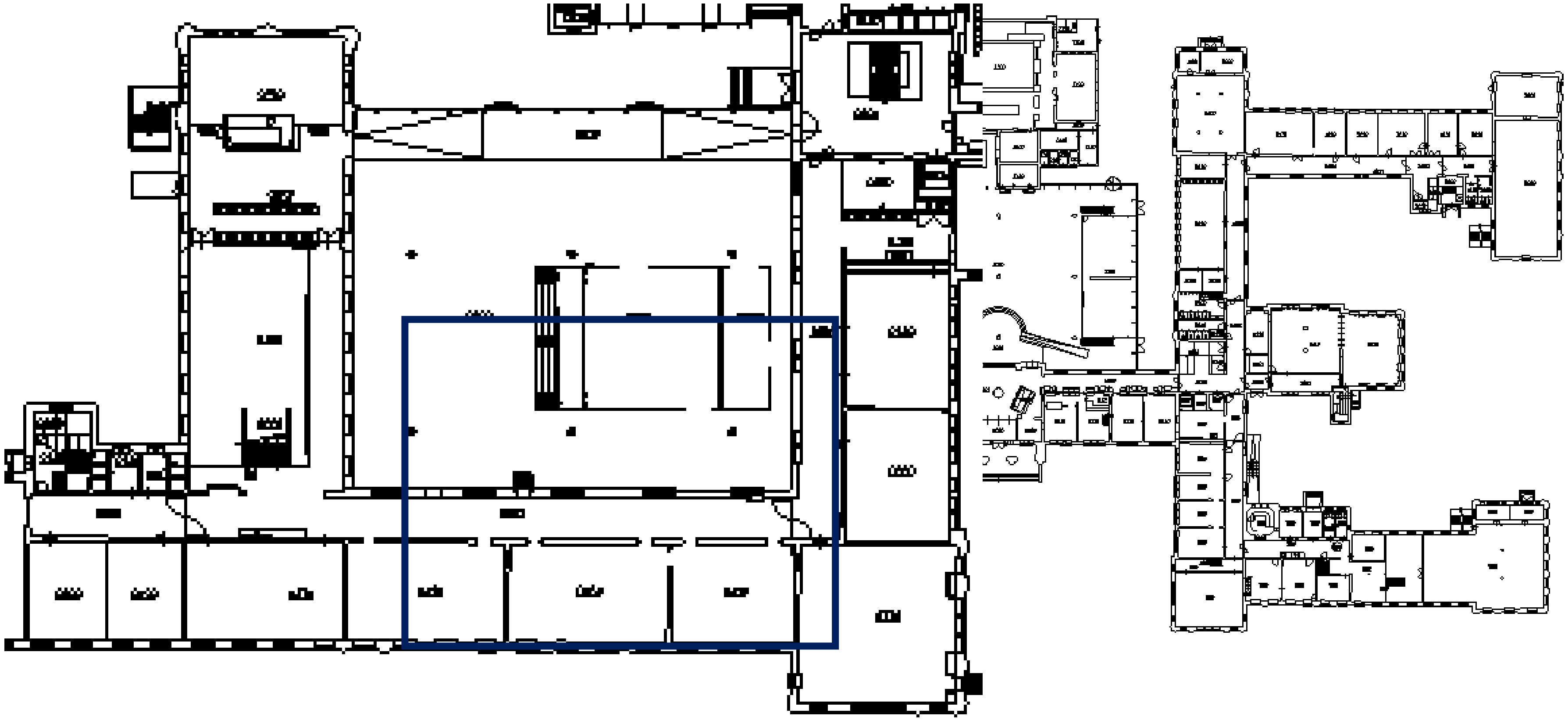
→ **6 cases**

Conceptual framework: Path planning algorithm
Navigate to: Target or Prediction?

Implementation and testing

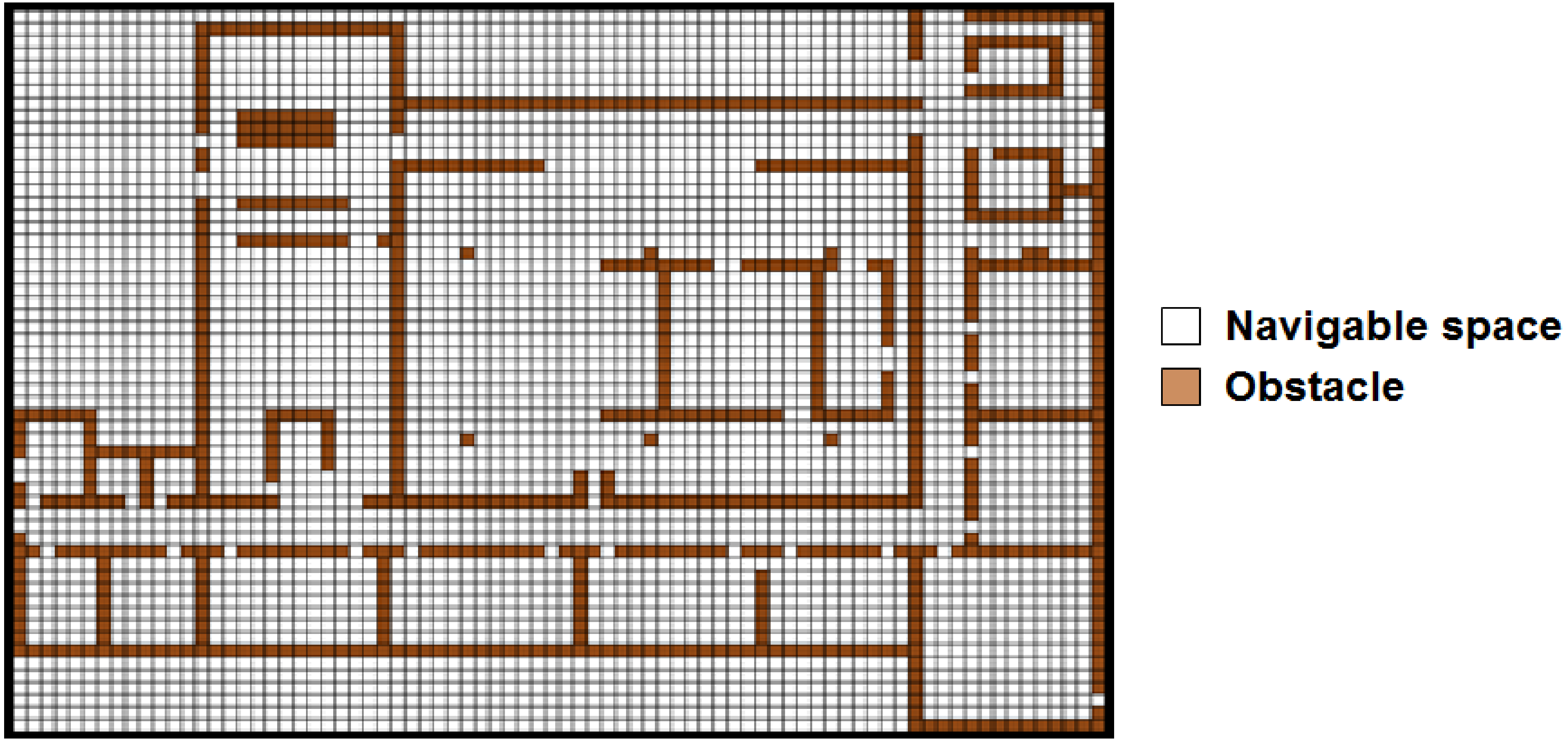
Indoor implementation

Map data of the faculty of architecture, TU Delft

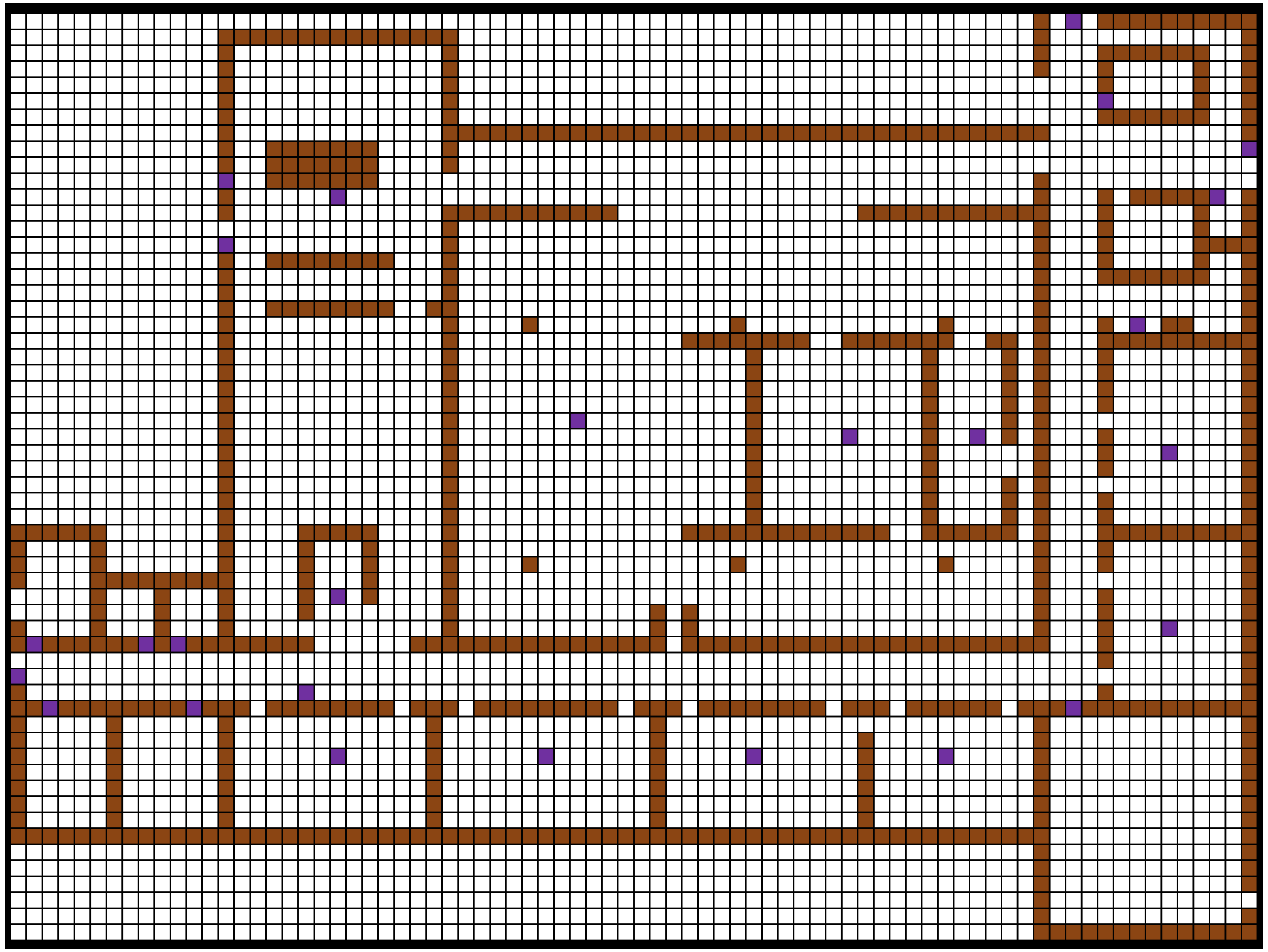


Indoor implementation: spatial model

2D square grid by rasterization



Indoor implementation: Points of interest



- Navigable space
- Obstacle
- Point of interest

Indoor implementation: Scenario analysis

Compare methods:

- SEA* method
- Iterative A* method navigating to the target

Based on:

- Different starting and end points based on real situations

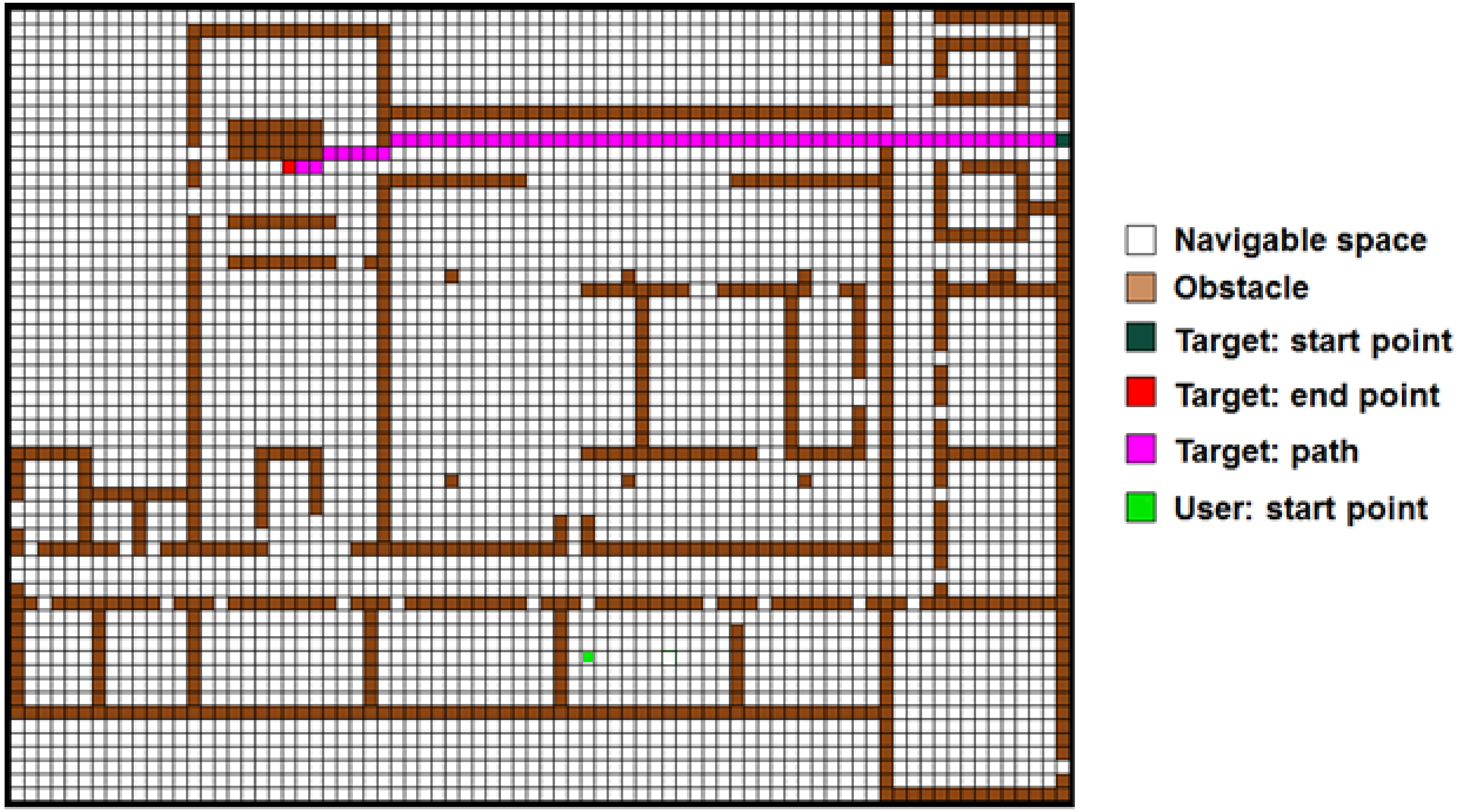
Effectiveness:

→ Is there a path?

→ Number of steps

Scenario 1: 15 minute break between a lecture

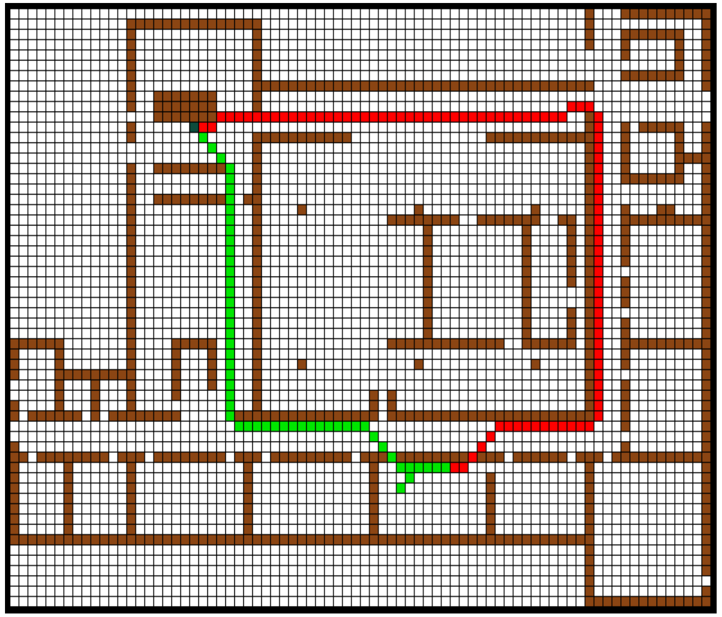
The target path and destination are only used to simulate the target, this information is not needed to use the method.



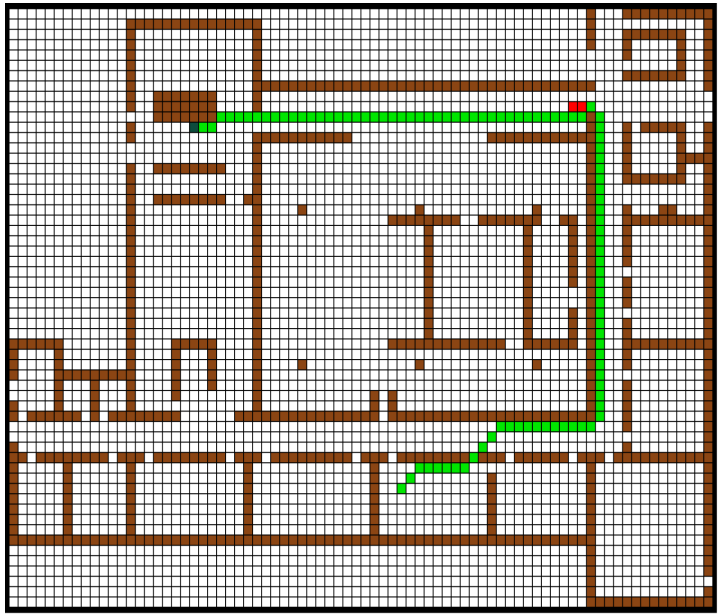
Scenario 1: 15 minute break between a lecture

SEA* Correct weights	SEA* Incorrect weights	SEA* Closest point	Iterative A*
57	96	67	96

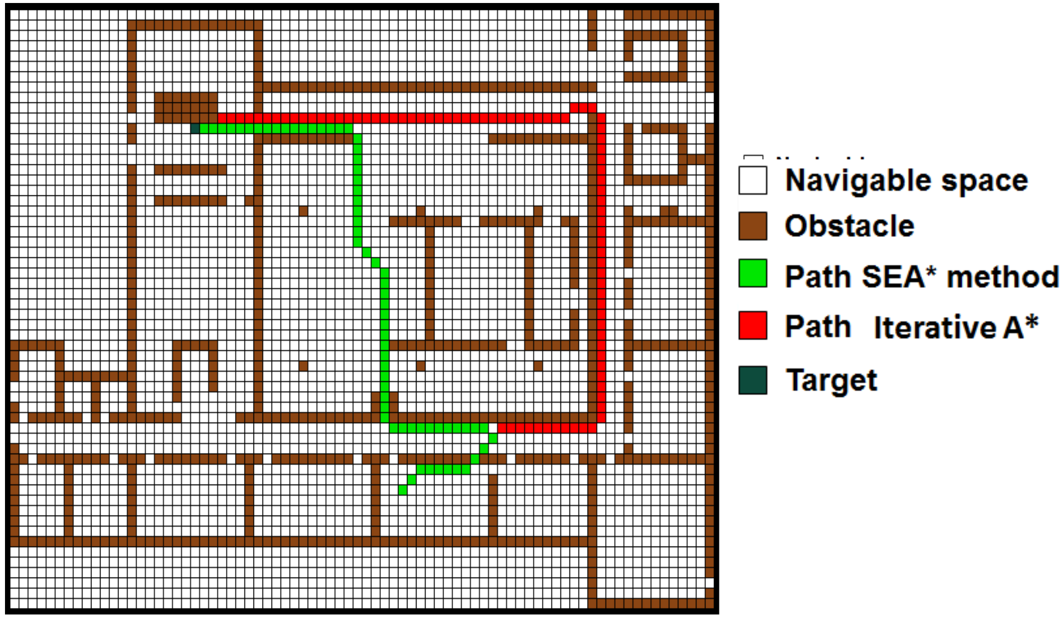
Using weights: **correct**



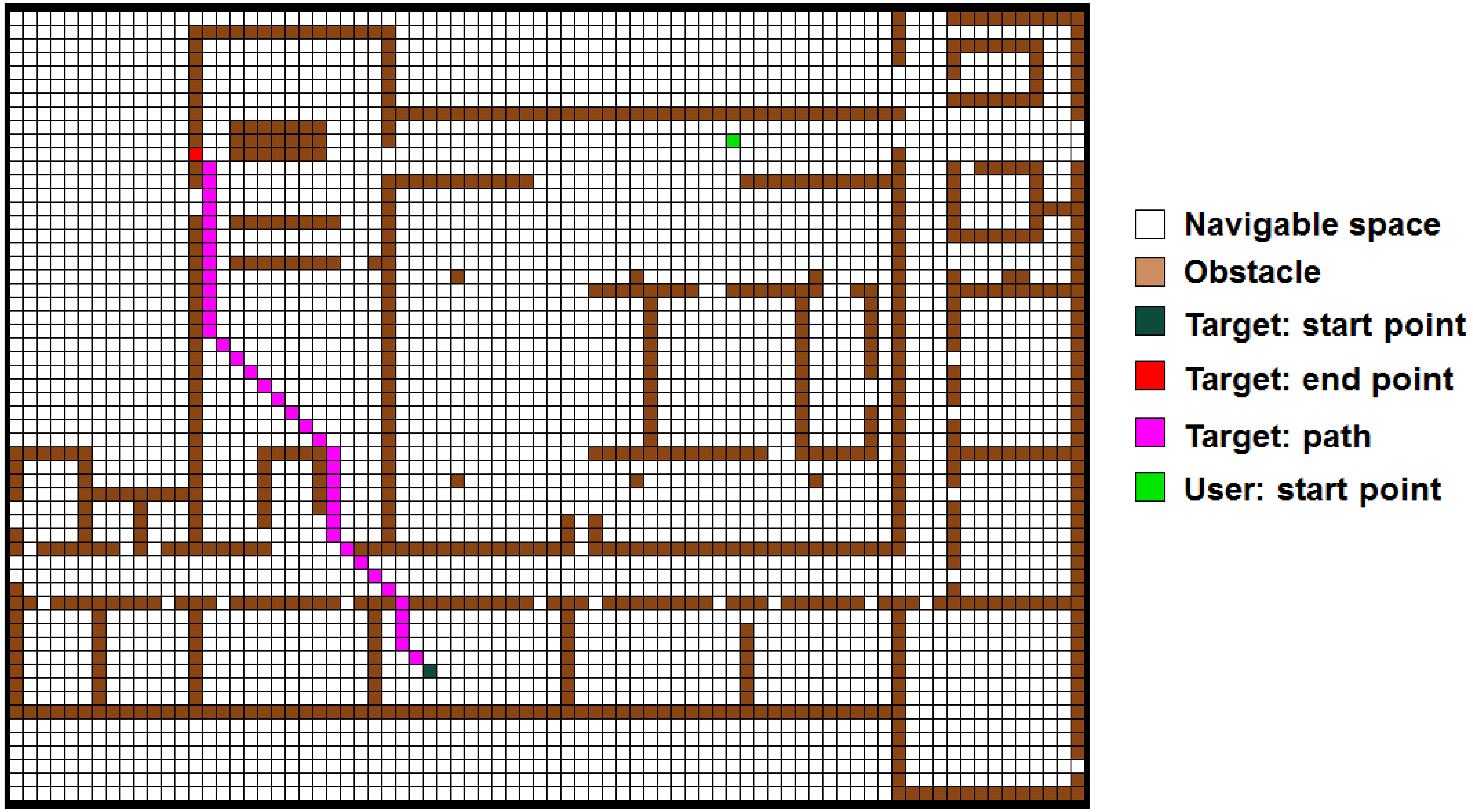
Using weights: **incorrect**



Using the **closest point** of interest



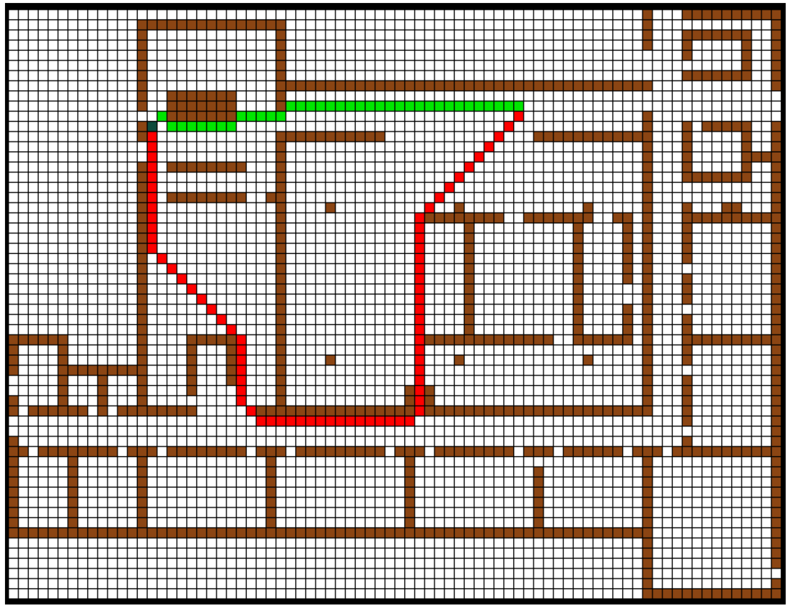
Scenario 2: Building closes



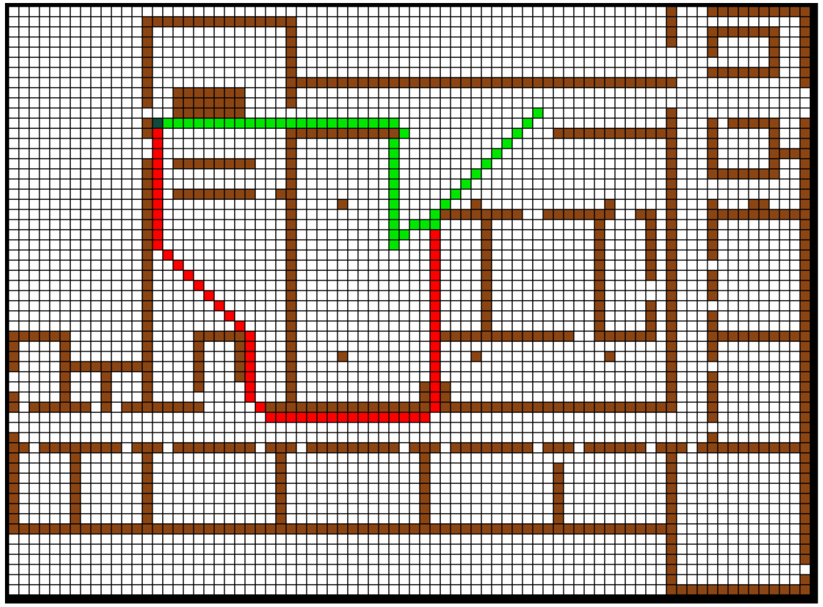
Scenario 2: Building closes

SEA* Correct weights	SEA* Closest point	Iterative A*
37	50	74

Using weights: **correct**

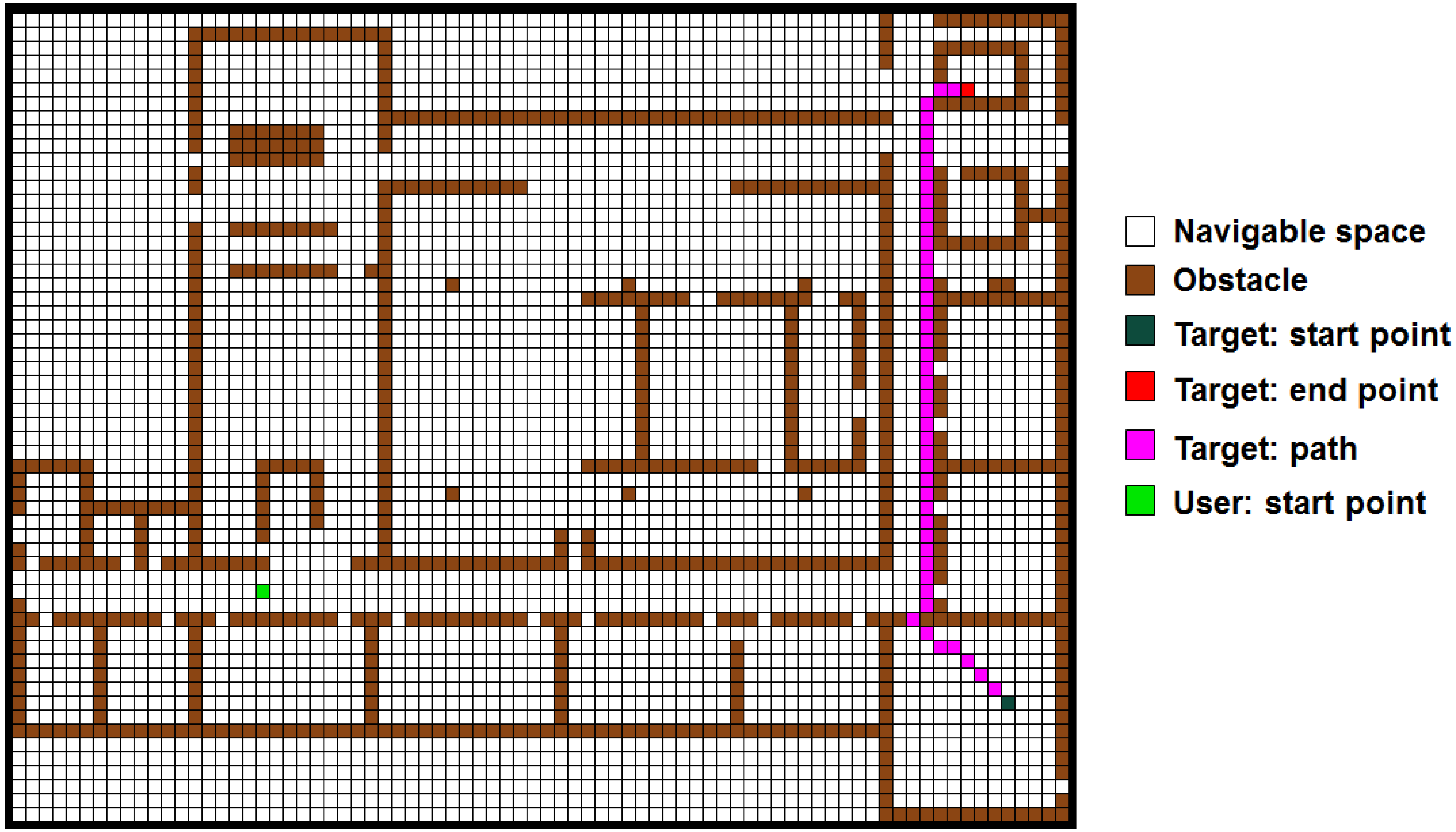


Using **the closest point** of interest



- Navigable space
- Obstacle
- Path SEA* method
- Path Iterative A*
- Target

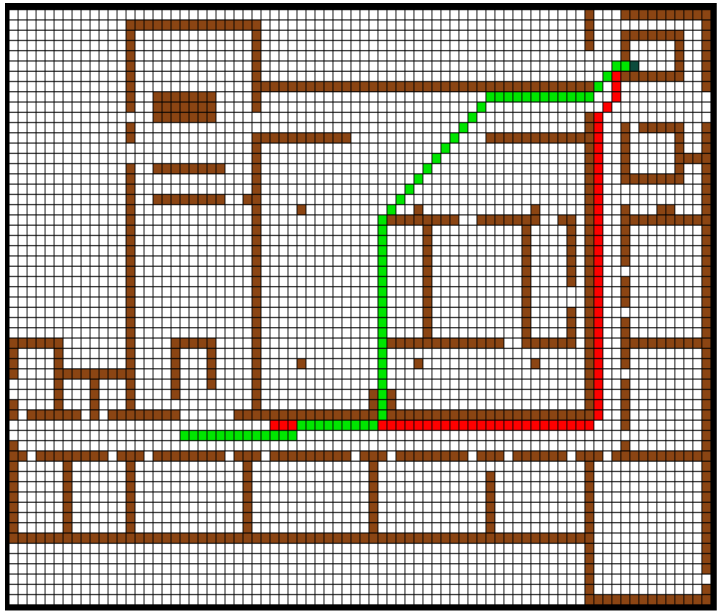
Scenario 3: Lecture on the first floor



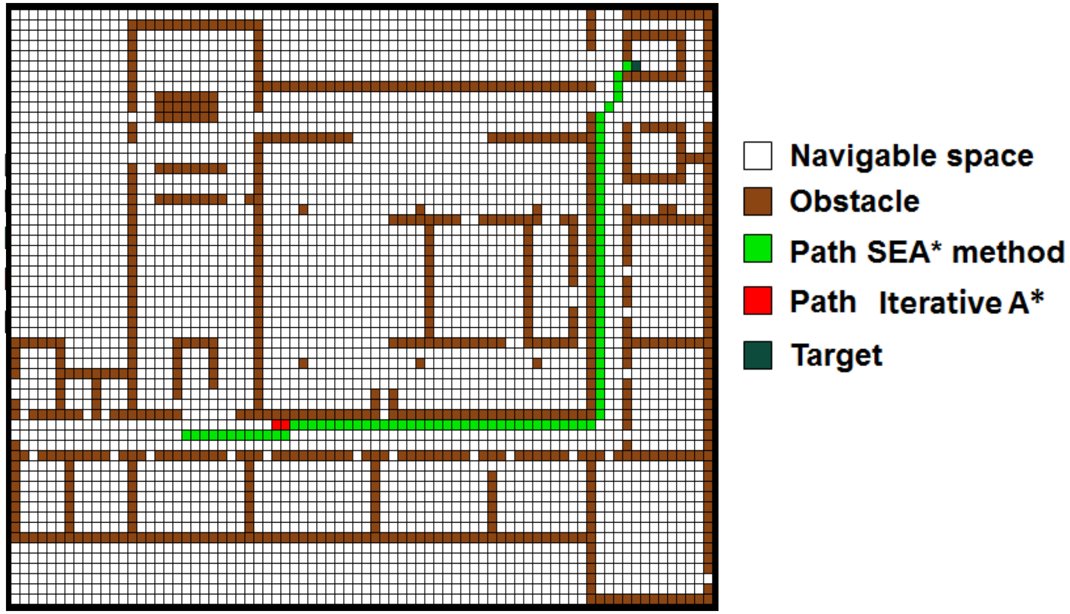
Scenario 3: Lecture on the first floor

SEA* Correct weights	SEA* Closest point	Iterative A*
69	81	81

Using weights: **correct**



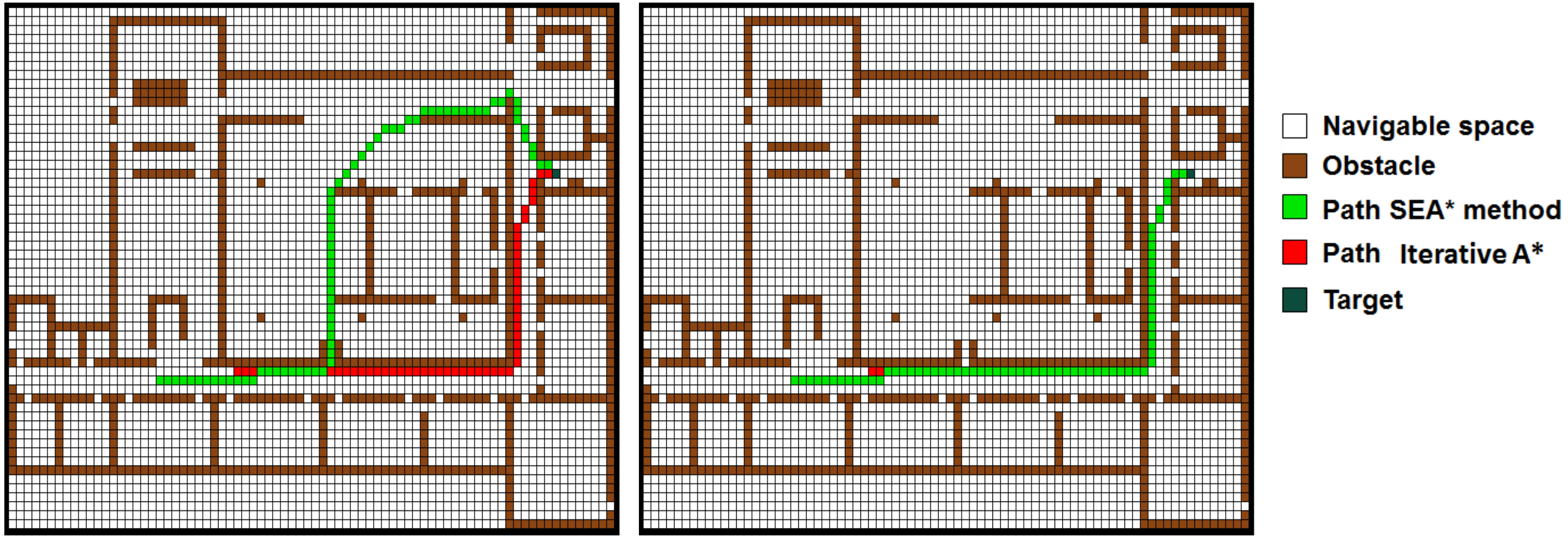
Using the **closest point** of interest



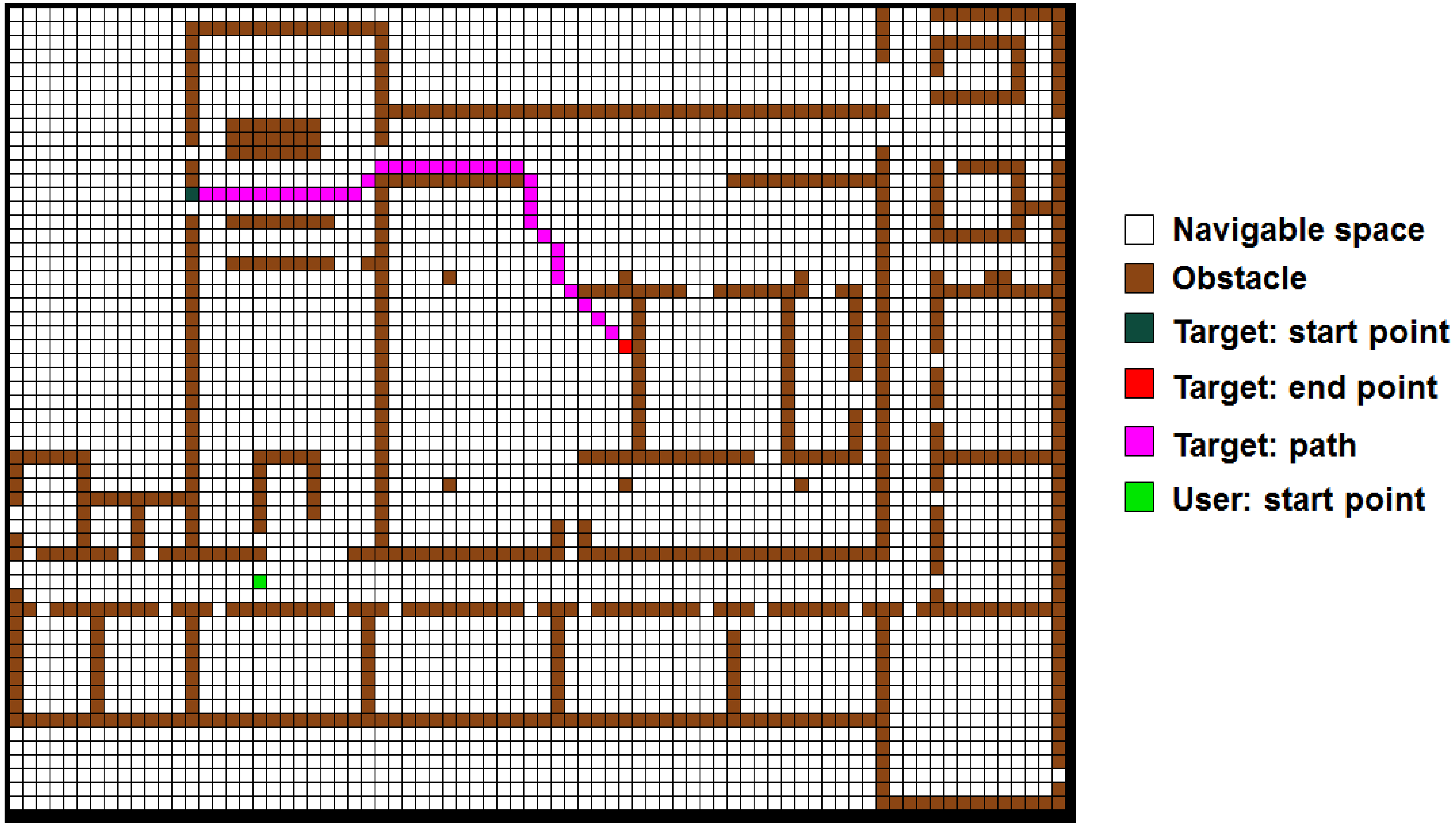
Scenario 3: Lecture on the first floor, but first a coffee

SEA* Incorrect weights	SEA* Closest point	Iterative A*
74	69	69

Using weights: **incorrect** Using the **closest point** of interest



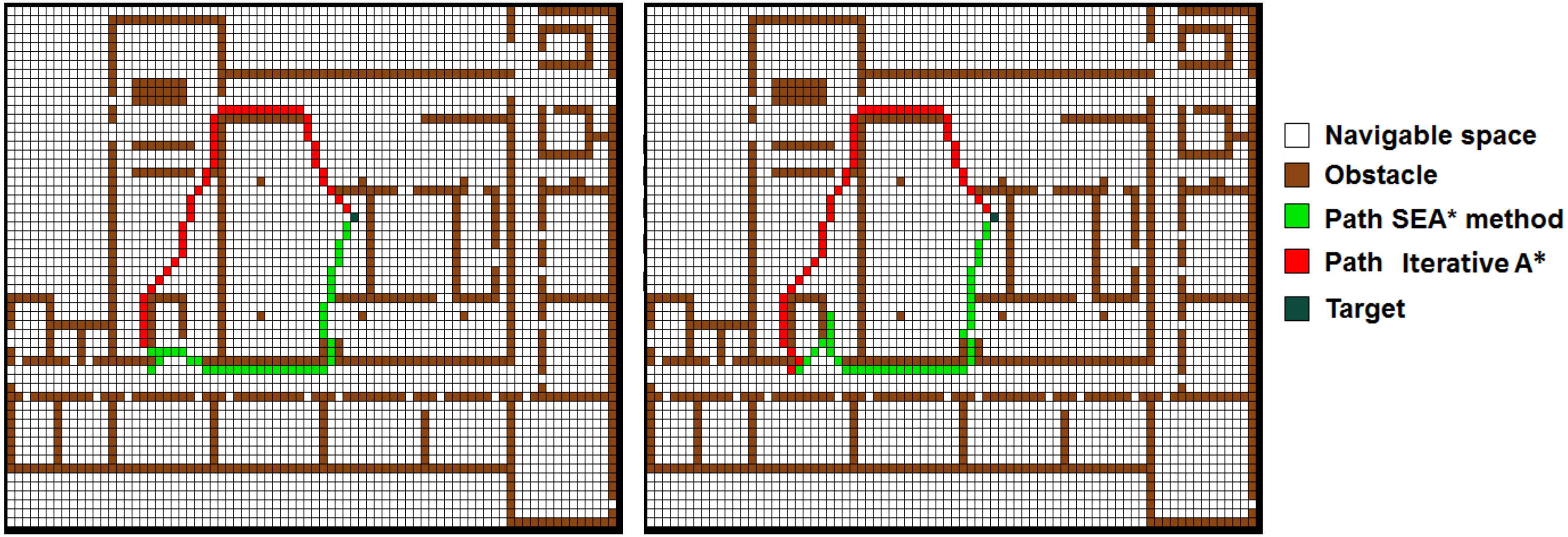
Scenario 4: Visitor visits an event



Scenario 4: Visitor visits an event

SEA* Correct weights	SEA* Closest point	Iterative A*
44	45	51

Using weights: **correct** Using the **closest point** of interest



Indoor implementation: Validation

Scenario	Determine the prediction	SEA* method	Iterative A* method
1. Break	Correct weight	57	96
1. Break	Incorrect weight	96	96
1. Break	Closest point	67	96
2. Closing the building	Correct weight	37	74
2. Closing the building	Closest point	50	74
2. Closing the building, in between	Correct weight	11	11
2. Closing the building, approaching left	Correct weight	39	40
2. Closing the building, approaching right	Correct weight	53	53
3. lecture on first floor	Correct weight	69	81
3. lecture on first floor	Closest point	81	81
3. lecture on first floor, first coffee	Incorrect weight	74	69
3. lecture on first floor, first coffee	Closest point	69	69
4. Event	Correct weight	44	51
4. Event	All equal	45	51

Outdoor implementation

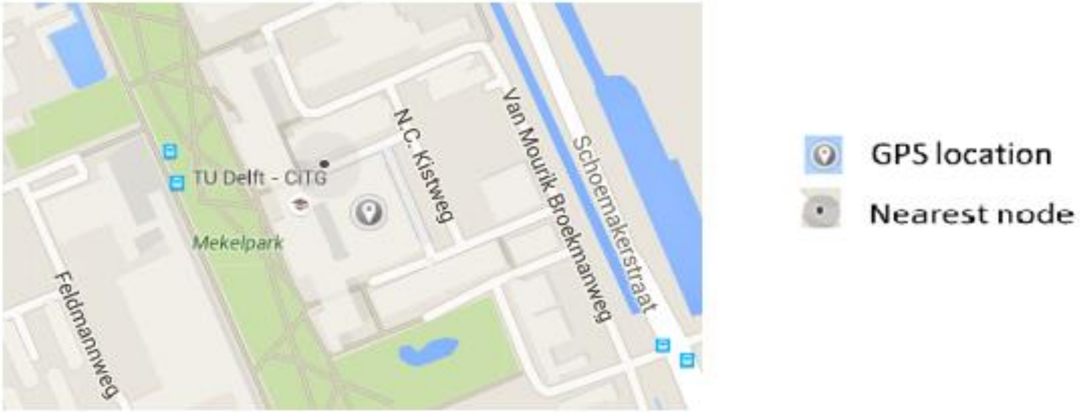
Using the Google Maps API V3: (**proprietary data**)

- Google Maps directions service → Navigation
- Google Maps distance matrix → calculate the distance to each point of interest
- Possibility to add points of interest
- The API supports routing for walking, cycling, driving and public transport.

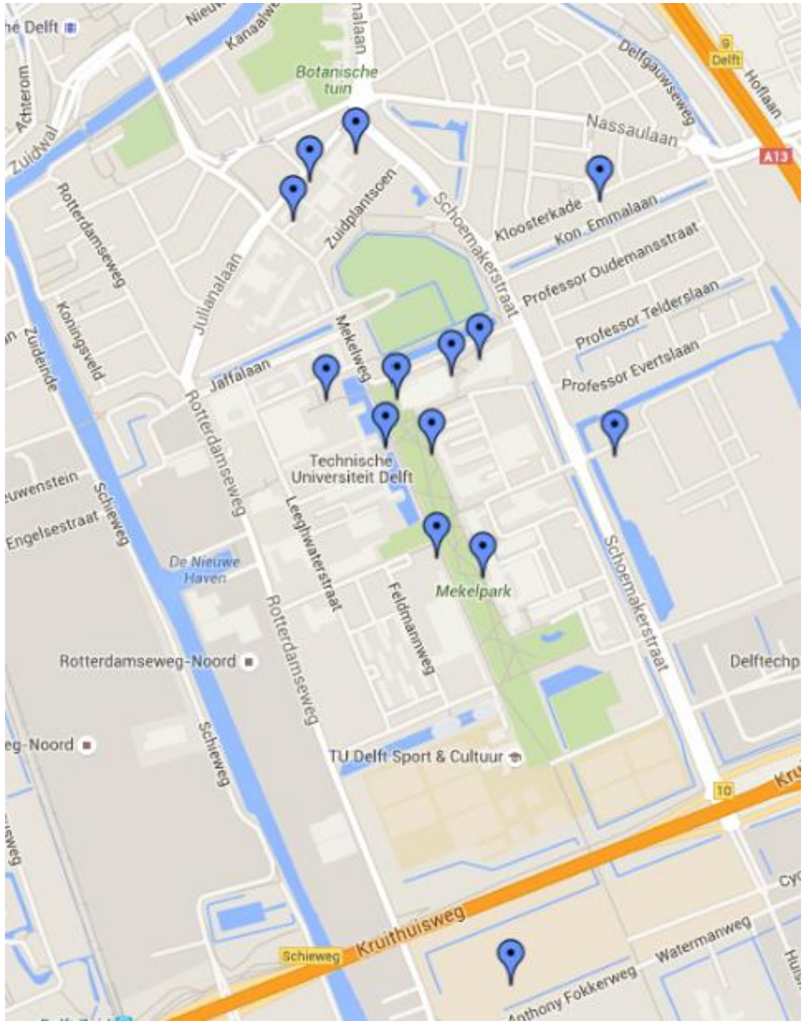
Test area: The campus of the TU Delft

Test data: GPS log

Outdoor implementation: closest point in the model

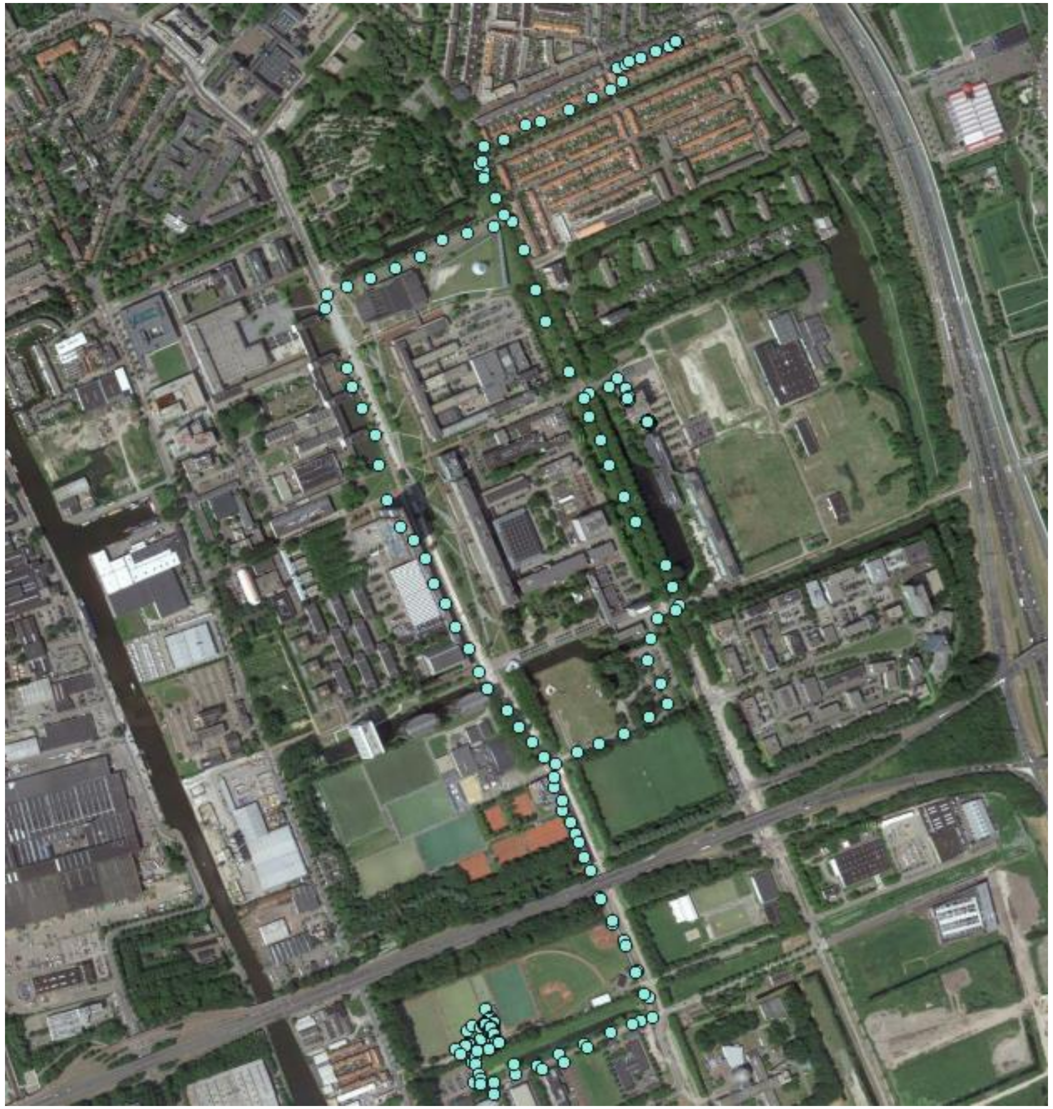


Outdoor implementation: Points of interest



Outdoor implementation: GPS log (Wilko)

GPS log reduced to every 10 seconds



Outdoor implementation: Scenario analysis

Compare methods:

- SEA* method with the closest point
- Iterative A* method

Based on:

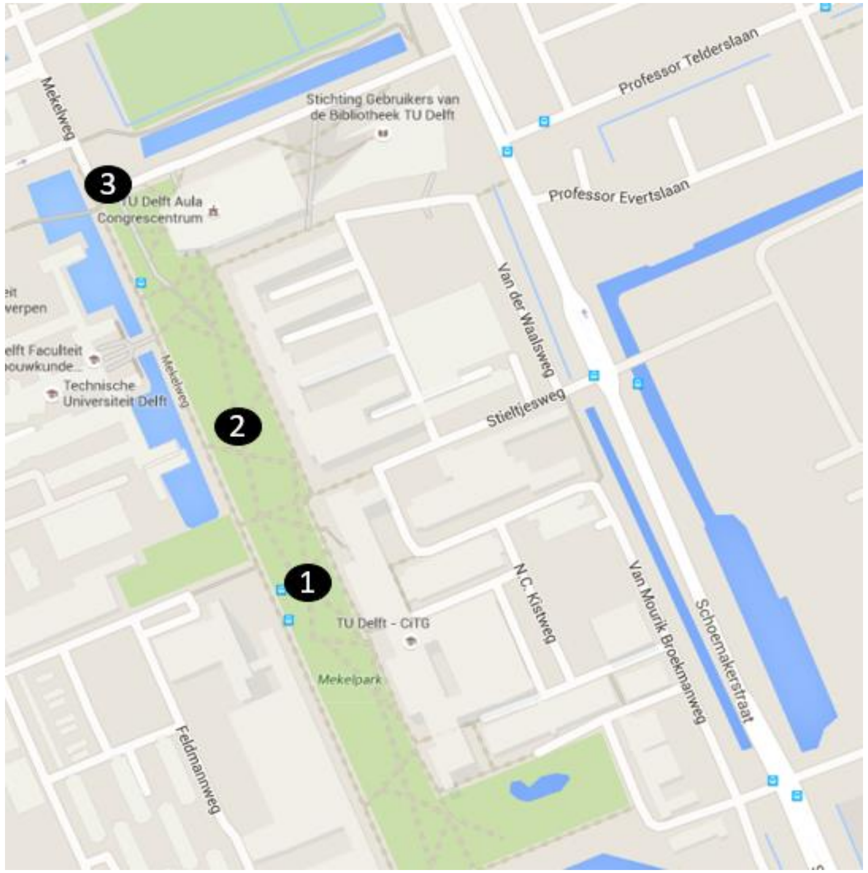
- GPS data
- Different starting points of the user

Effectiveness:

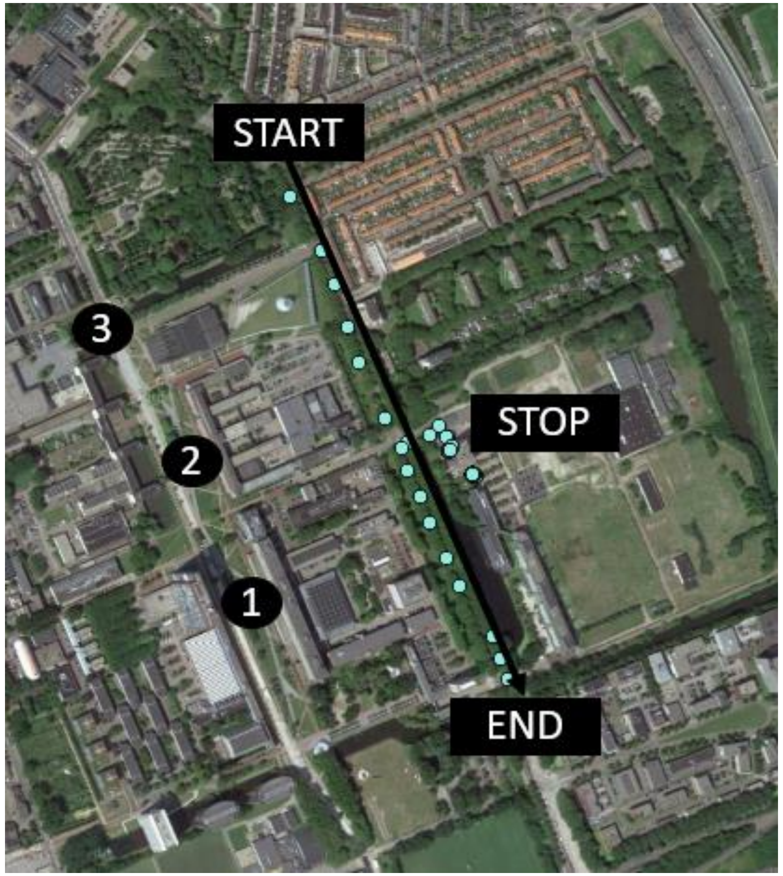
→ Is there a path?

→ Duration of the path / distance of the path

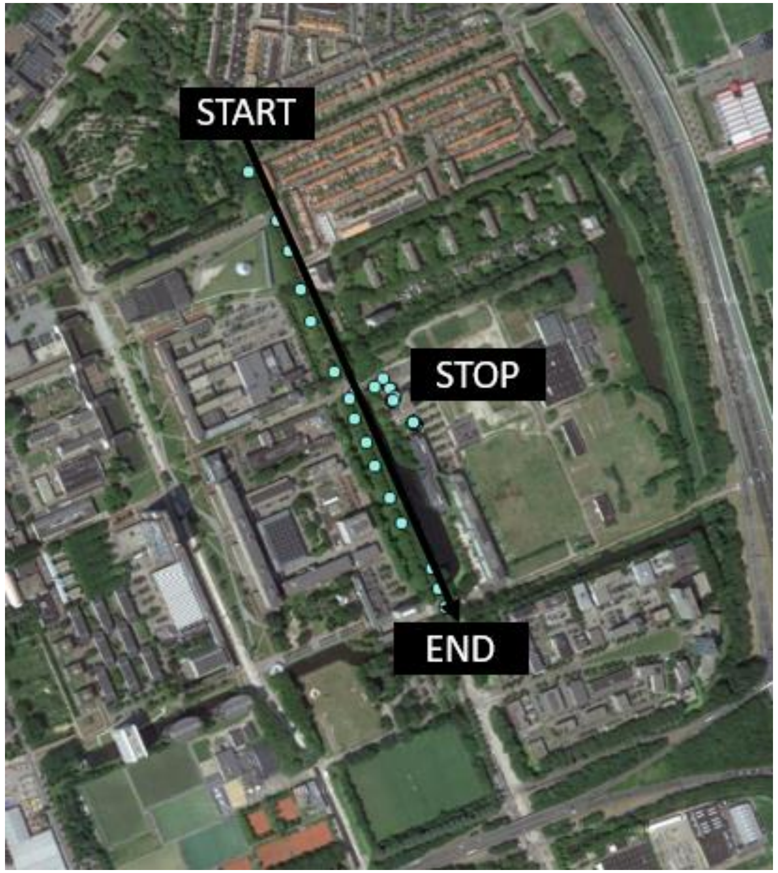
Scenario 1: Target is moving, stops in between



Starting positions of the user



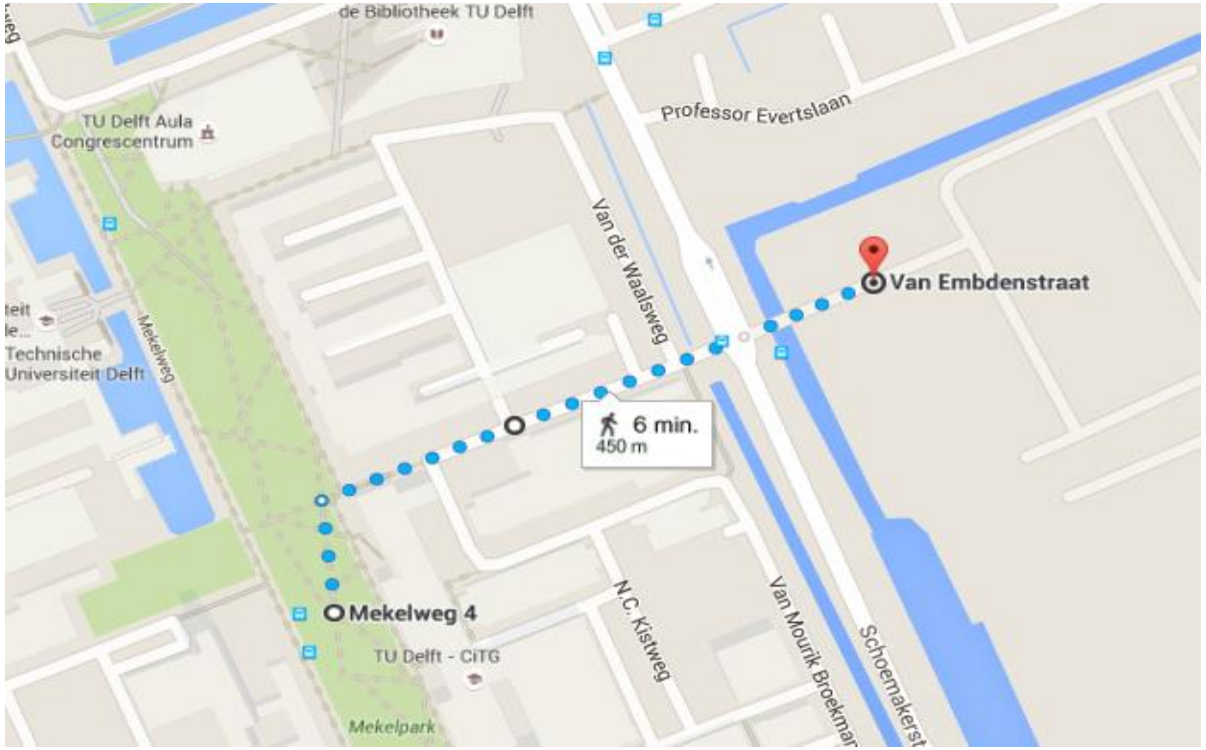
Combined



GPS data

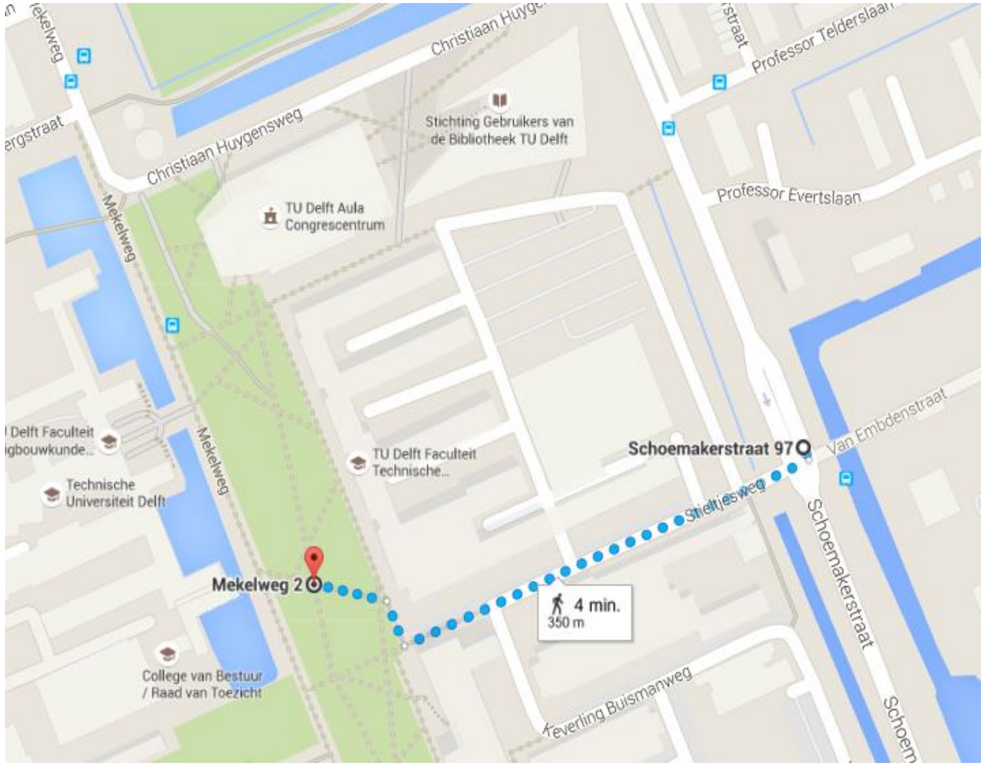
Scenario 1: Target is moving, stops in between
Start position 1

Path for the SEA method and the iterative A* method, reaching the target in 6 minutes (450 meter).*

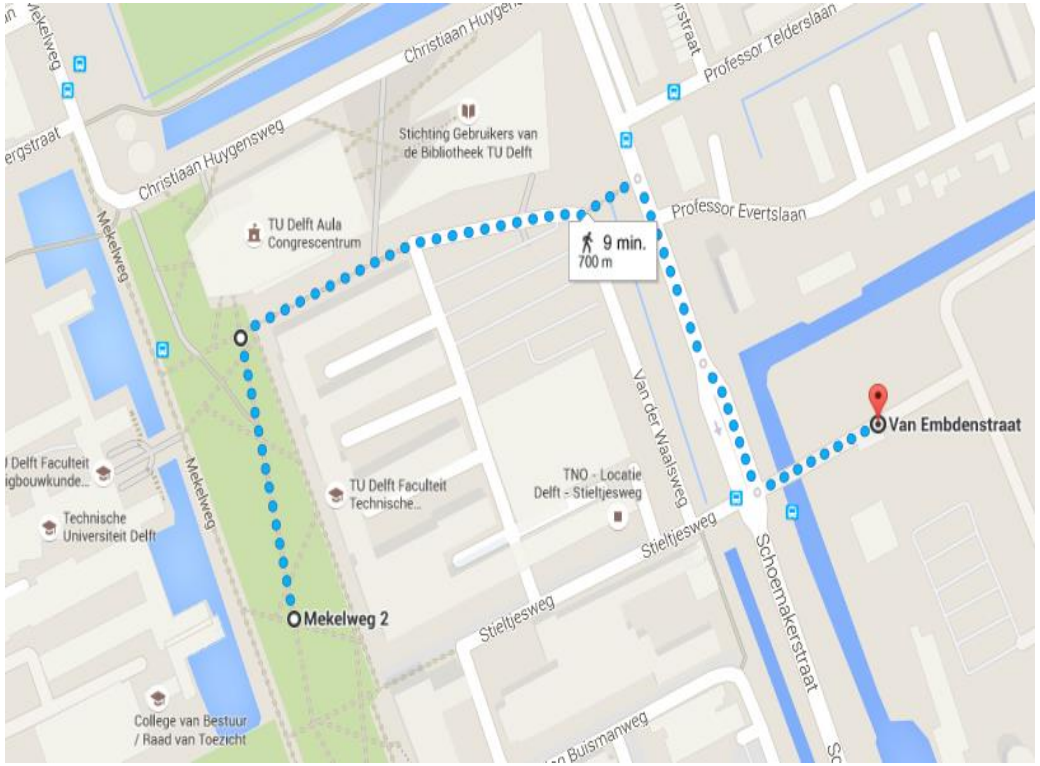


Scenario 1: Target is moving, stops in between *Start position 2*

SEA method,*
reaching the target in 4 minutes (350 meter).

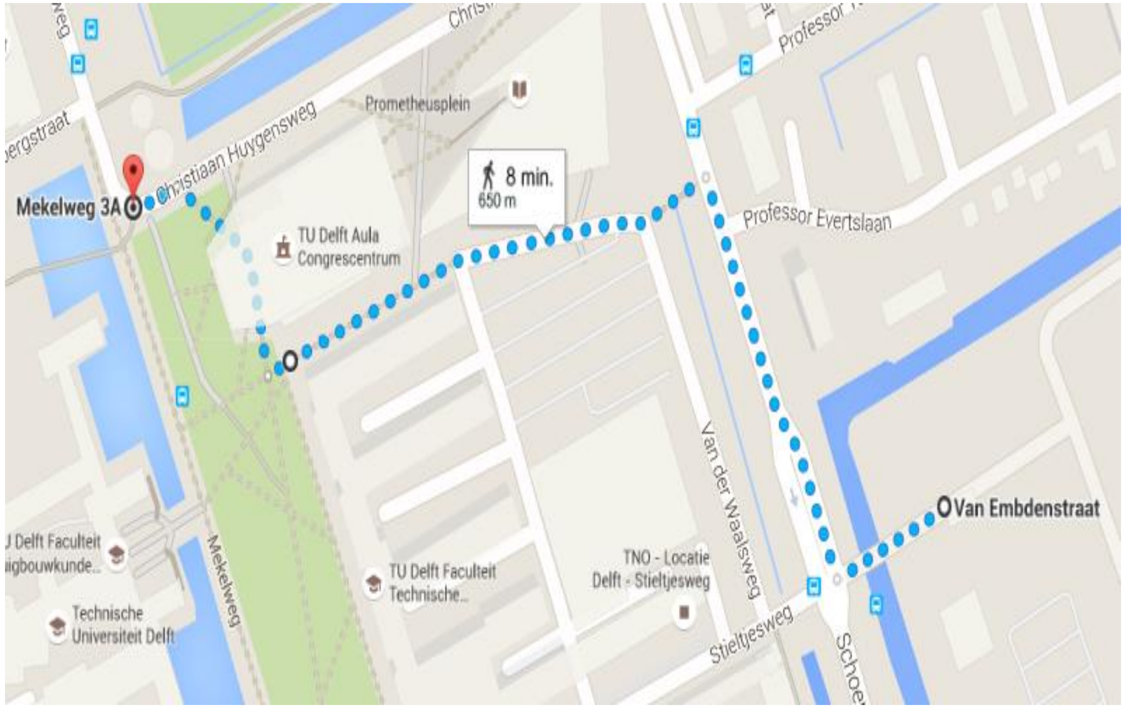


Iterative A method,*
reaching the target in 9 minutes (700 meter).

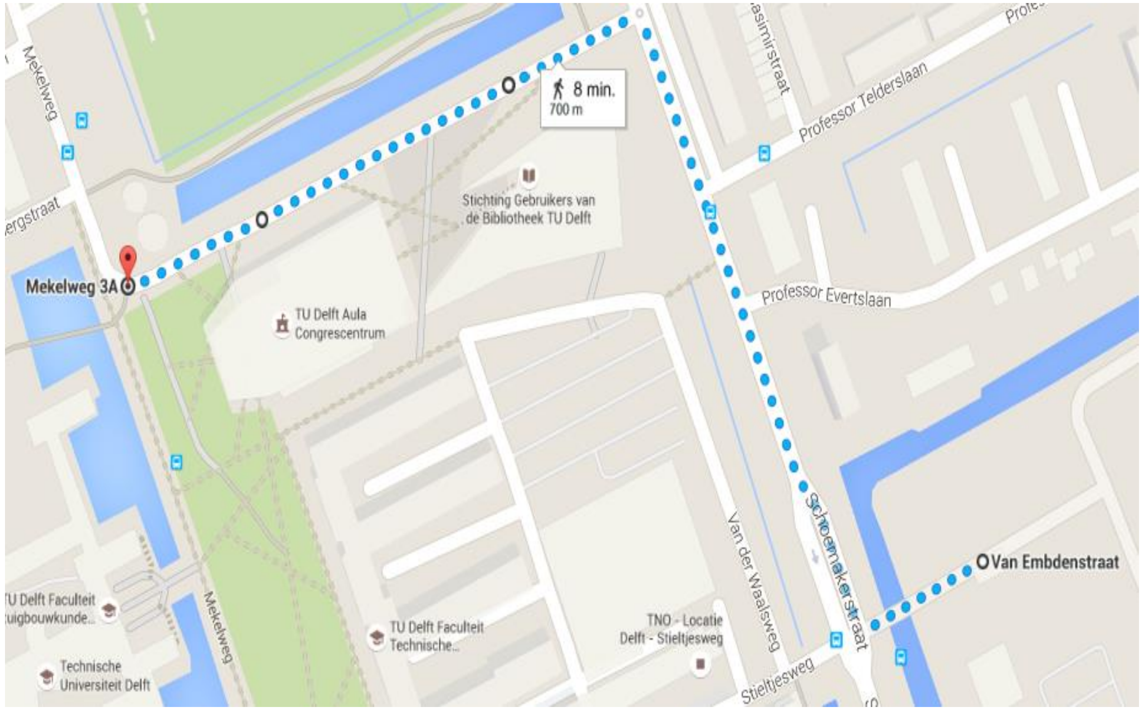


Scenario 1: Target is moving, stops in between
Start position 3

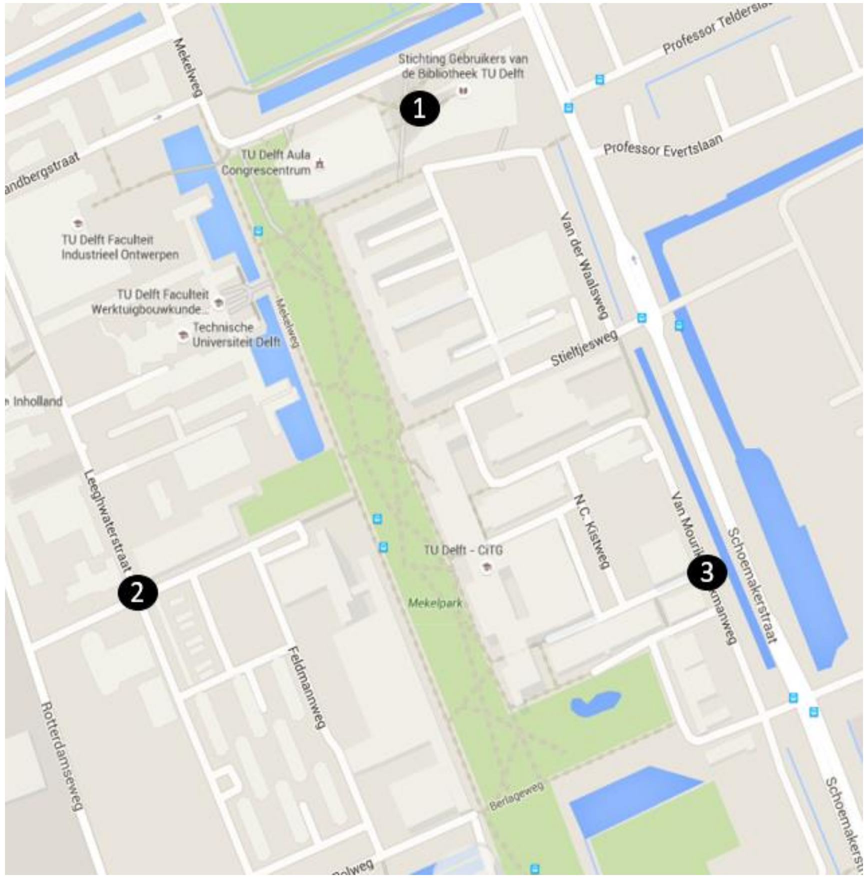
SEA method,*
reaching the target in 8 minutes (650 meter).



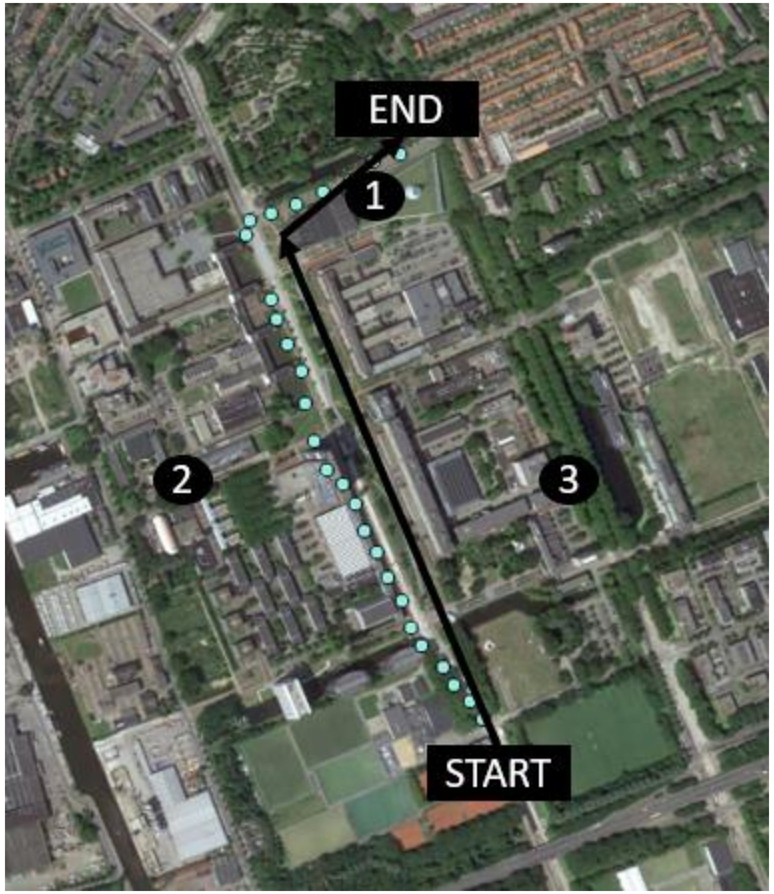
Iterative A method,*
reaching the target in 8 minutes (700 meter).



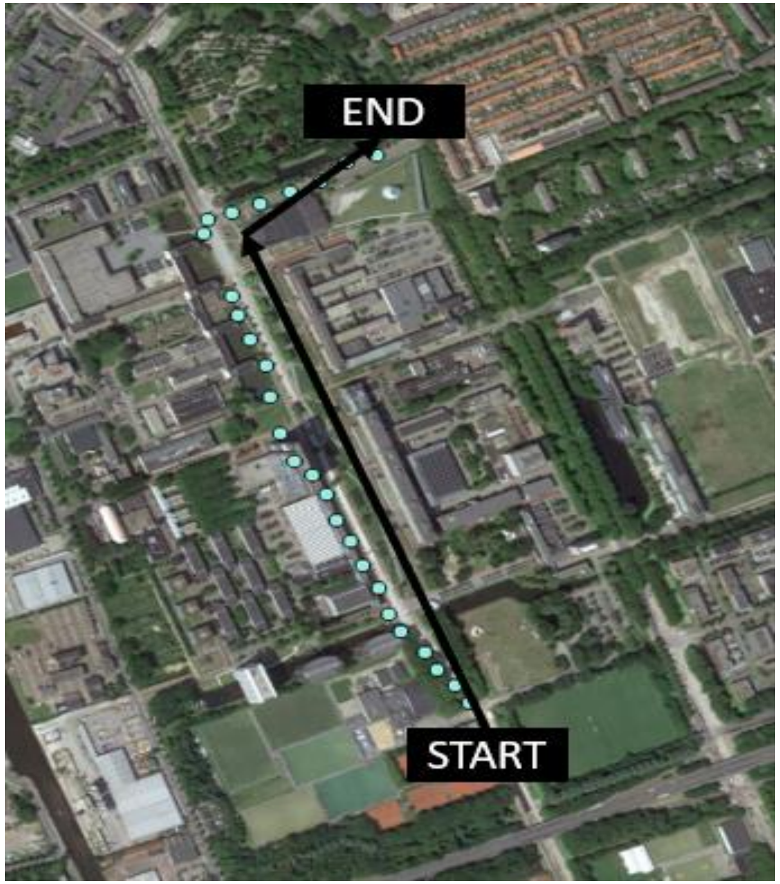
Scenario 2: Target is moving, without stopping



Starting positions of the user



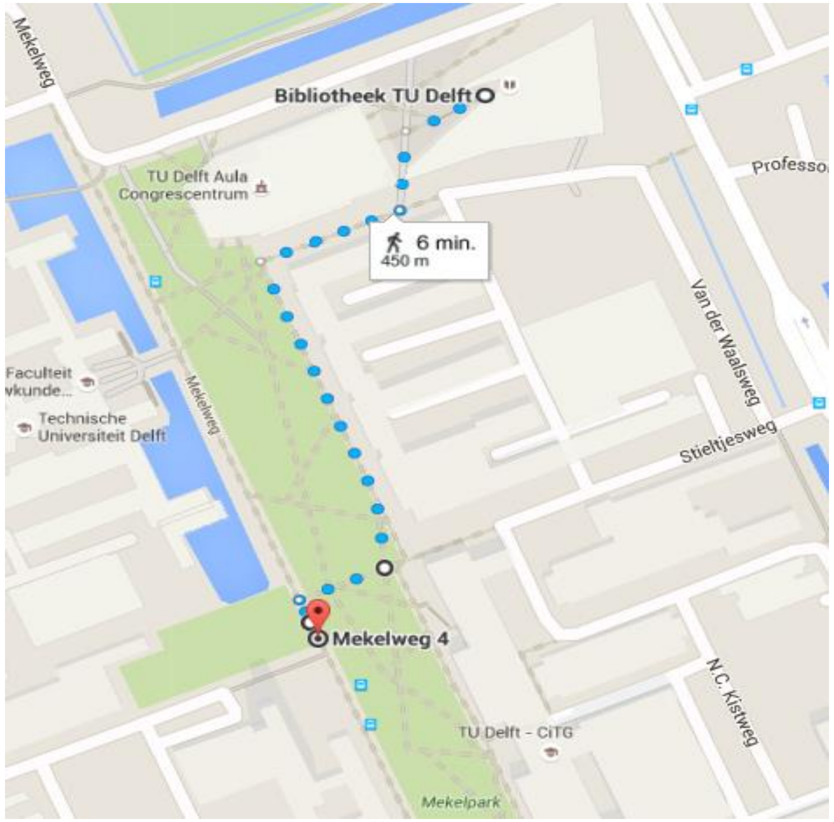
Combined



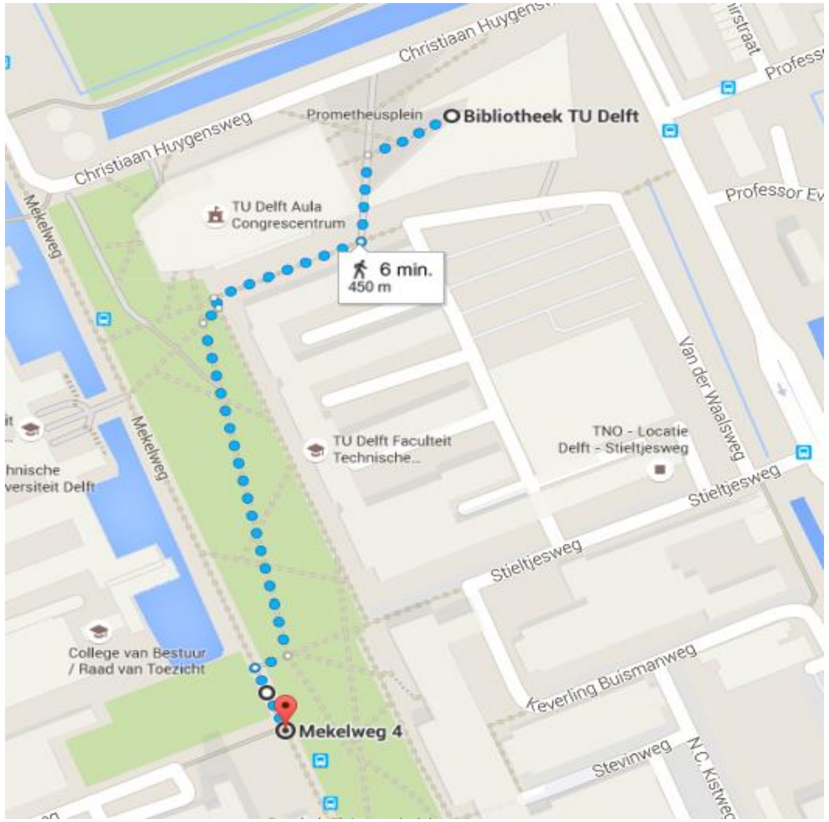
GPS data

Scenario 2: Target is moving, without stopping
Start position 1

SEA method,*
reaching the target in 6 minutes (450 meter).

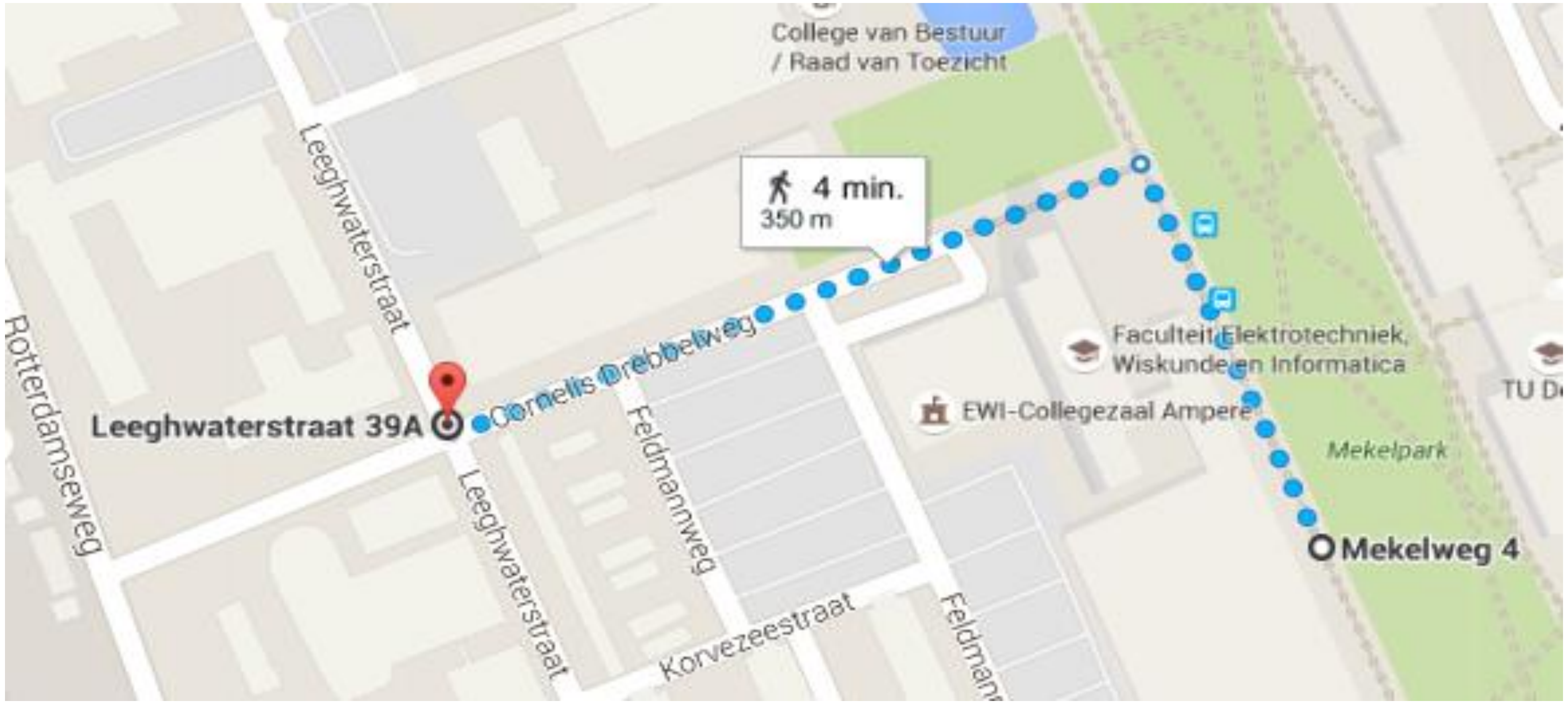


Iterative A method,*
reaching the target in 6 minutes (450 meter).



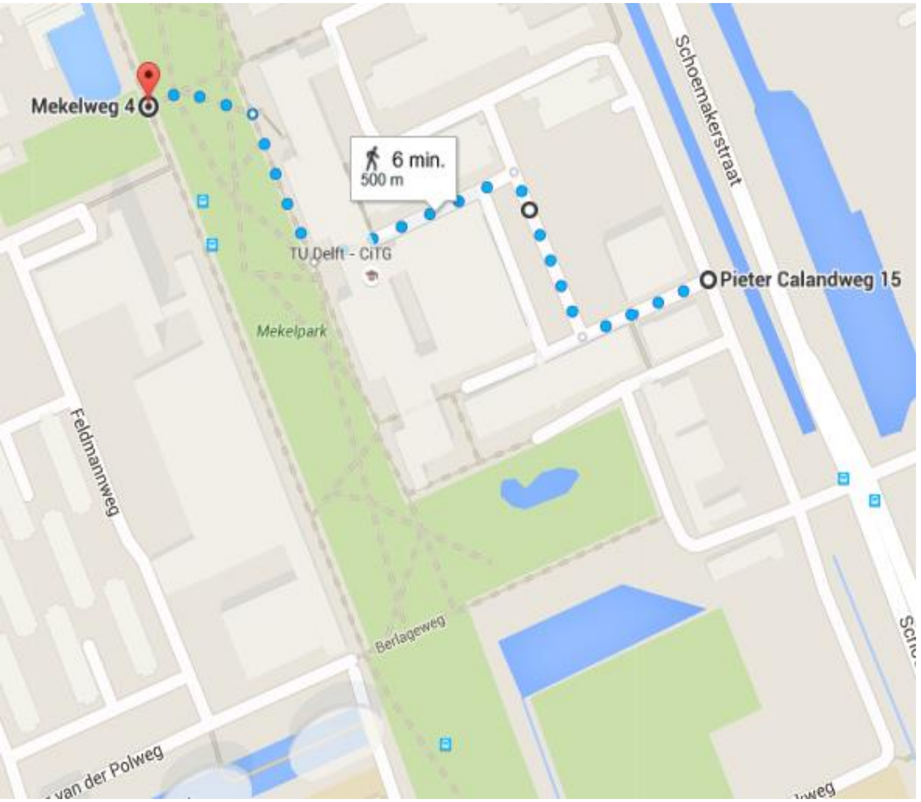
Scenario 2: Target is moving, without stopping *Start position 2*

Path for the SEA method and the iterative A* method, reaching the target in 4 minutes (350 meter).*

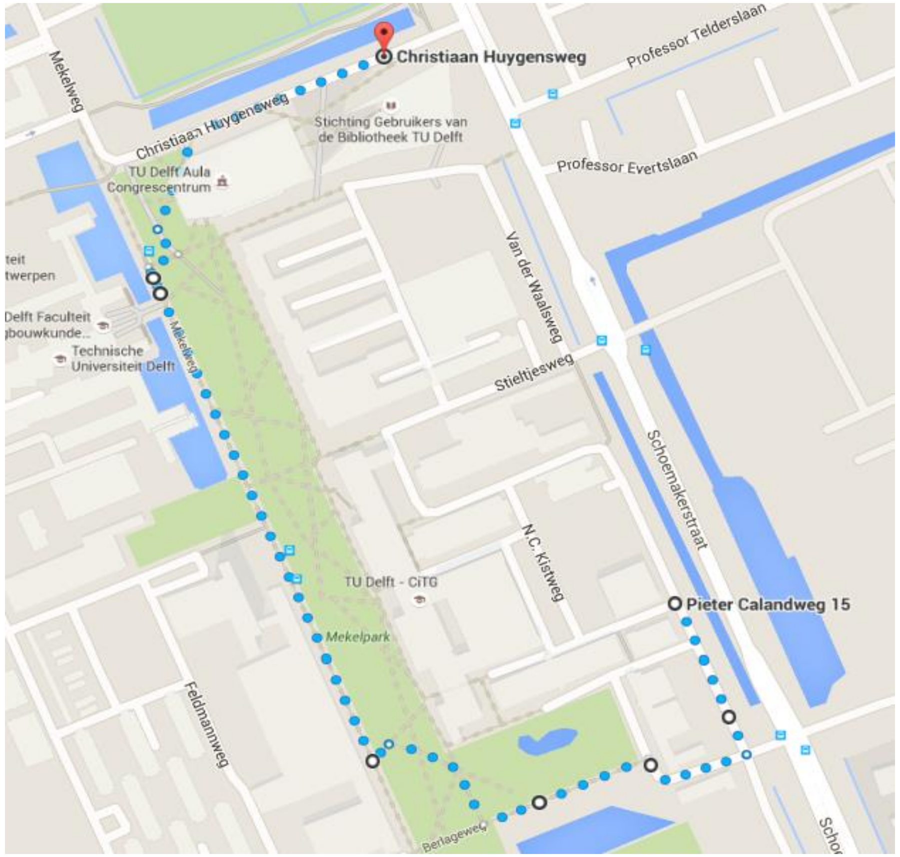


Scenario 2: Target is moving, without stopping
Start position 3

SEA method,*
reaching the target in 6 minutes (500 meter).



Iterative A method,*
not reaching the target.



Outdoor implementation: Validation

Scenario	Starting point	SEA* method	Iterative A* method
1	1	6 minutes (450 meter): Optimal	6 minutes (450 meter): Optimal
1	2	4 minutes (350 meter): Optimal	9 minutes (700 meter): Not optimal
1	3	8 minutes (650 meter): Optimal	8 minutes (700 meter): Sub-Optimal
2	1	6 minutes (450 meter): Optimal	6 minutes (450 meter): Optimal
2	2	4 minutes (350 meter): Optimal	4 minutes (350 meter): Optimal
2	3	6 minutes (500 meter): Optimal	- (target not reached)

Conclusions

Which defined objects could be used to estimate the predicting location of a moving person to support navigation to a person in motion?

Points of interest have been proven to be useful to navigate a person to another person in motion, both indoor as outdoor, by using the movement of the target.

Points of interest are defined as static locations:
landmarks, entrances of buildings or doors within buildings.

Manually selected where persons interact with the point of interest → re-use

The points of interest have not to be target dependent.
Indoor environment: points of interest are target independent
Outdoor environment: some points of interest are target dependent

Therefore target independent points of interest are predefined, which would be done preferably automatic.
- Use a semantic model or the Google Places API

On the fly addition of target specific points of interest enhances the system.

Conclusions

The SEA* method

There are two ways to determine the prediction of the target:

Certain where the target is going: use weights

Otherwise, use the closest point of interest the target is approaching

A limitation of the SEA* method is that it is slower if the user takes a shortcut, but the target is suddenly heading back.

SEA* method behaves as the iterative A* method when:

- Target is standing still (A* fastest path)
- No defined points of interest in the direction of the target

The method only uses the current positions of the target.

Conclusions

Using the SEA* method

The framework could be implemented by using different spatial models, as long as it is possible to create a navigable graph and:

- The A* algorithm is supported
- Possible to add points of interest to the model
- There is a positioning technique

Navigating multiple persons:

- When two persons want to navigate to each other, use the A* algorithm.
- It is possible to navigate multiple people to one person.
- The iterative A* and SEA* methods do not support navigation of three persons towards each other.

SEA* method uses the positive components of the iterative A* method, predefined points of interest and the direction of the target to predict where the target is going to, to successfully reach the target.

Future work

Real-time application	→ Use the GPS sensor of a mobile phone. → Use indoor positioning techniques.
3D application	Use an irregular grid, voxel representation or an Octree.
Extend the prediction	Speed of the target. Previous taken path of the target.
Extract points of interest	Automate the extraction of points of interest. Use the Google Places API.
When navigate to the target or the prediction	Determine the influence of obstacles.
Using a visibility analysis	Notify the user when the target is in vision range

Navigation to a human in motion by using points of interest

Tim Nagelkerke

27-06-2016

First mentor:	Zlatanova, Dr.ing. S.
Second mentor:	Diakite, A.A.
Co-reader:	Quak, Drs.C.W.
Delegate of the board of examiners:	Burg, Ir. L.P.J. van den

TU Delft