第 8 回 会津大学ロボットシンポジウム

Enhanced Path Planning Method for Autonomous Service Robot

THE UNIVERSITY OF AIZU

Raihan Kabir and Yutaka Watanobe

raihan.kabir.cse@gmail.com, yutaka@u-aizu.ac.jp

Table of Contents:

Background study and Objective of this Research

- An autonomous robotic wheelchair is a type of mobile robot that is designed to help people with disabilities or mobility impairments to move around independently.
- Features of autonomous robotic wheelchairs:
  - Navigation
  - Vision system to avoid obstacles
  - Control, management and Safety
  - Accessibility
  - Independence

Robotic Wheelchair

- Our Research focus is in the area of robot **Navigation, Vision, Control,** and **management system** to automate the robotic wheelchair that can move autonomously with some simple commands.

Proposed approach for the enhancement in Robotic Path Planning

Robotic Wheelchair

Robot Navigation

A* path planning algorithm

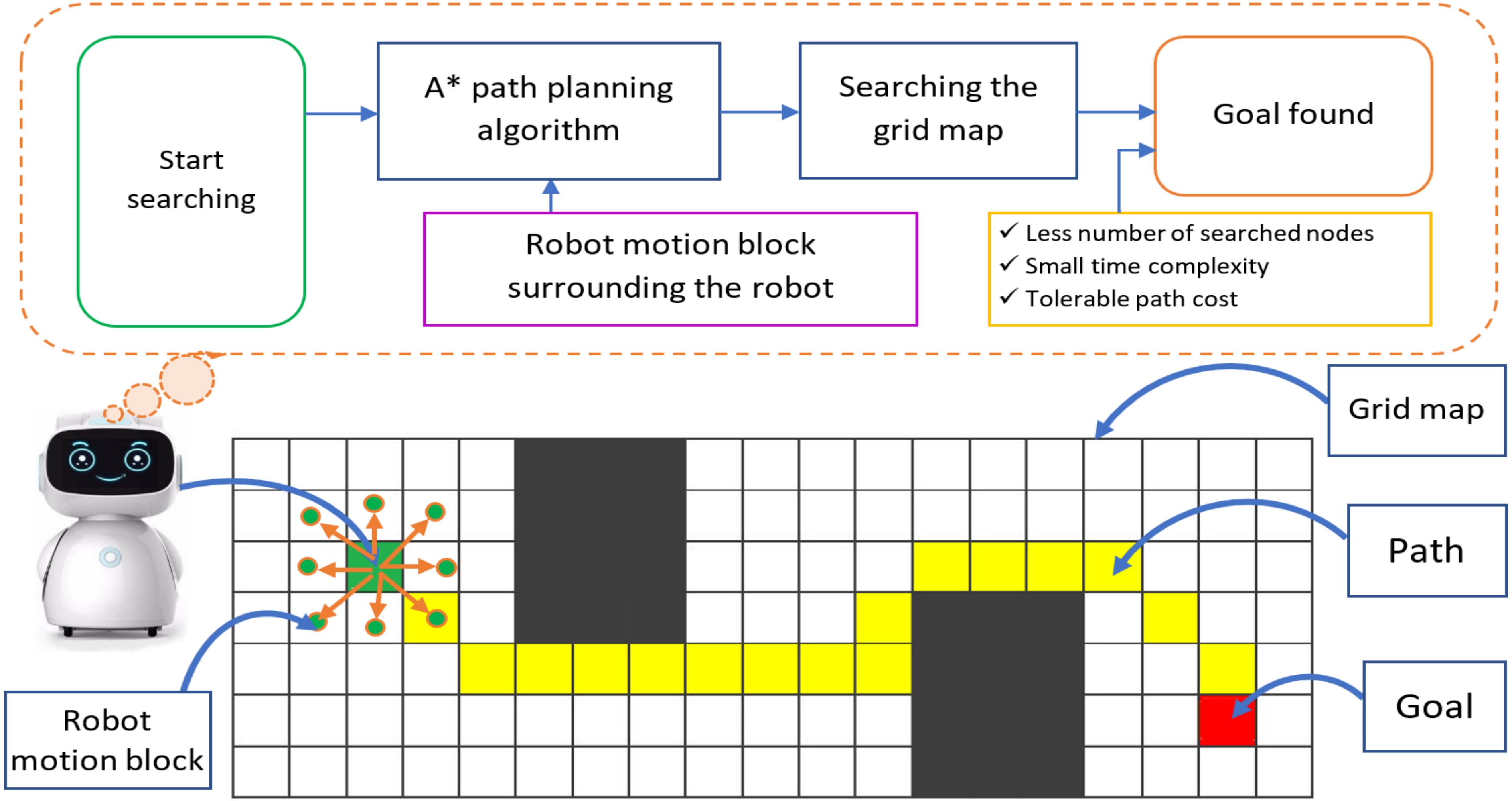
Performance enhancement using proposed Robot motion block

Publications: *(Major conference)*

- **Raihan Kabir,** Yutaka Watanobe, Keitaro Naruse, Rashedul Islam, (2022), " **Effectiveness of Robot Motion Block on A-star Algorithm for Robotic Path Planning** " In: SoMet-2022, pp: 85-96. DOI: 10.3233/FAIA220241

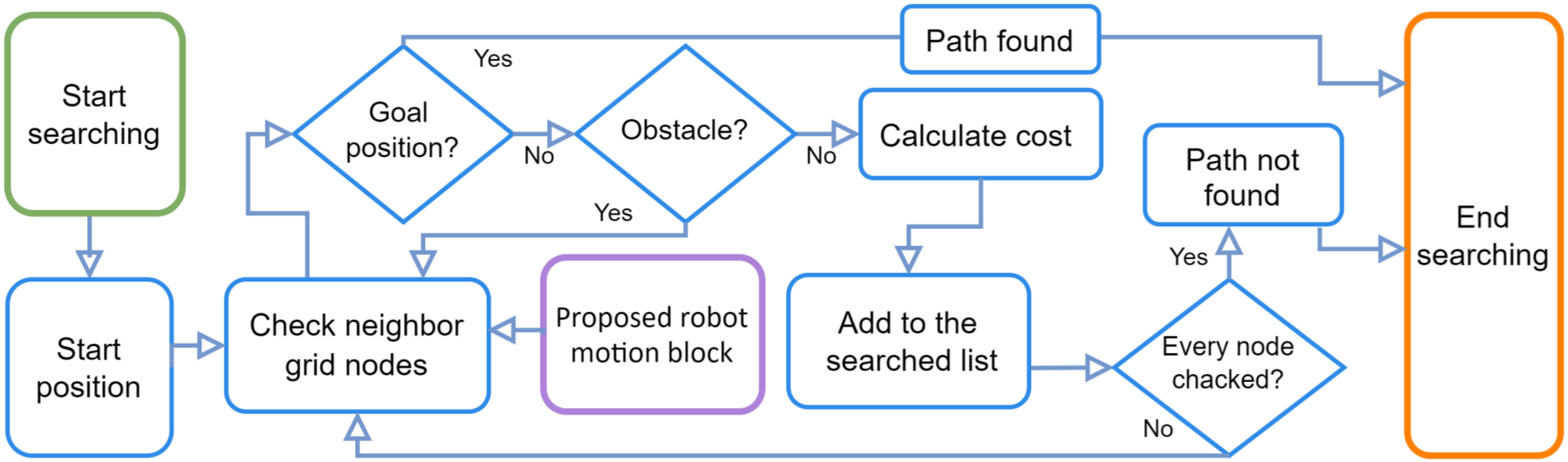
Proposed Method

To achieve the research goal, the general block diagram of our proposed model is:



A* path planning algorithm with proposed robot motion block

- A* is one of the famous path searching and graph traversal algorithm for its optimality, completeness, and optimal efficiency.
- It is an updated version of best-first search or informed search algorithm.
- The strategy is to search in a weighted graph to find the given goal node by taking a consistent heuristic function h(s) as input that uses Euclidean distance $d(p,q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2}$.
- It calculate the cost for each nodes $Cost(C) = min\{searched - list,\ \ h(S_{goal}, searched - list)\}$



Proposed robot motion block:

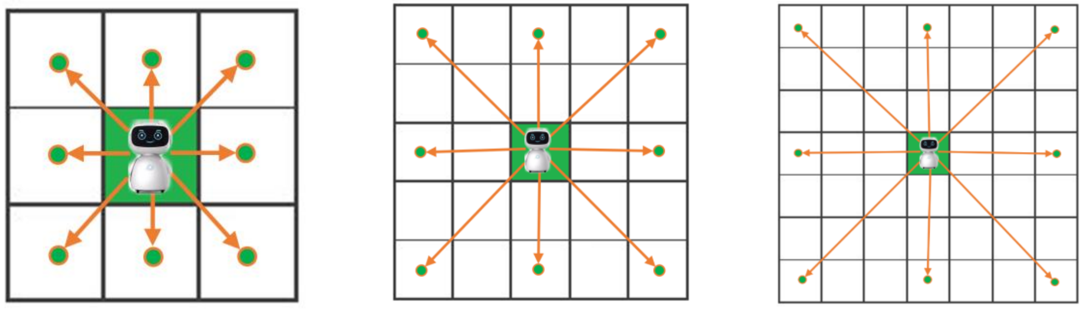
- In a grid-based search, a robot needs to search its surrounding neighbor cells optimally so that it can find the goal node by checking less amount of grid cells.
- For searching the neighbor nodes, it uses a block of cell nodes which is called a robot motion block or motion kernel.
- The first, second, and third columns represent the x-axis, y-axis, and the costs of the particular points in 2D grid space.

$$motion = \begin{bmatrix} n & 0 & C(n) \\ 0 & n & C(n) \\ -n & 0 & C(n) \\ 0 & -n & C(n) \\ -n & -n & C(\sqrt{n*2}) \\ -n & n & C(\sqrt{n*2}) \\ n & -n & C(\sqrt{n*2}) \\ n & n & C(\sqrt{n*2}) \end{bmatrix}$$

Here, $n = 1, 2, 3, \ldots, N$

C = movement cost towards the neighbor cells.

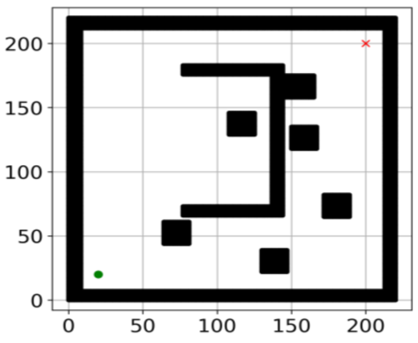
- In the proposed robot motion block matrix, n represents the distance between the robot and the searching cells.
- The proposed motion block has eight neighbor cell nodes surrounding the robot.



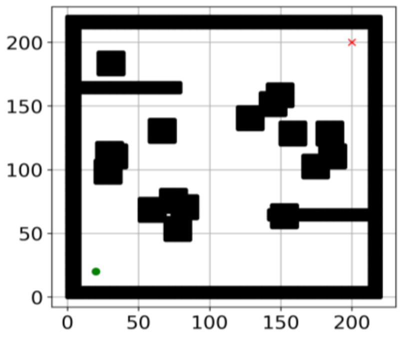
$n = 1$ $n = 2$ $n = 3$ …

Grid-map preparation using the dataset: (Dataset)

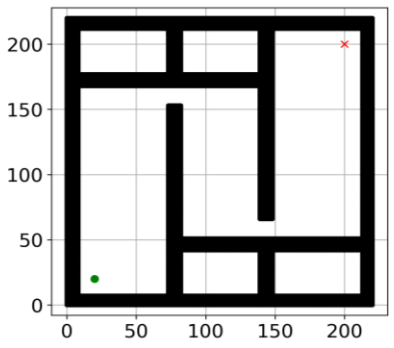
- The dataset contains eight types of grid maps which are represented as **PNG format images**. But for the experiment, **we need a matrix of grid maps.**
- **Dataset images have no border,** which is essential so that the robot cannot escape from the grid map.
- Each type of planning map environment contains **1,000 images**, and the resolution of the images is **201*201.**



bugtrap_forest

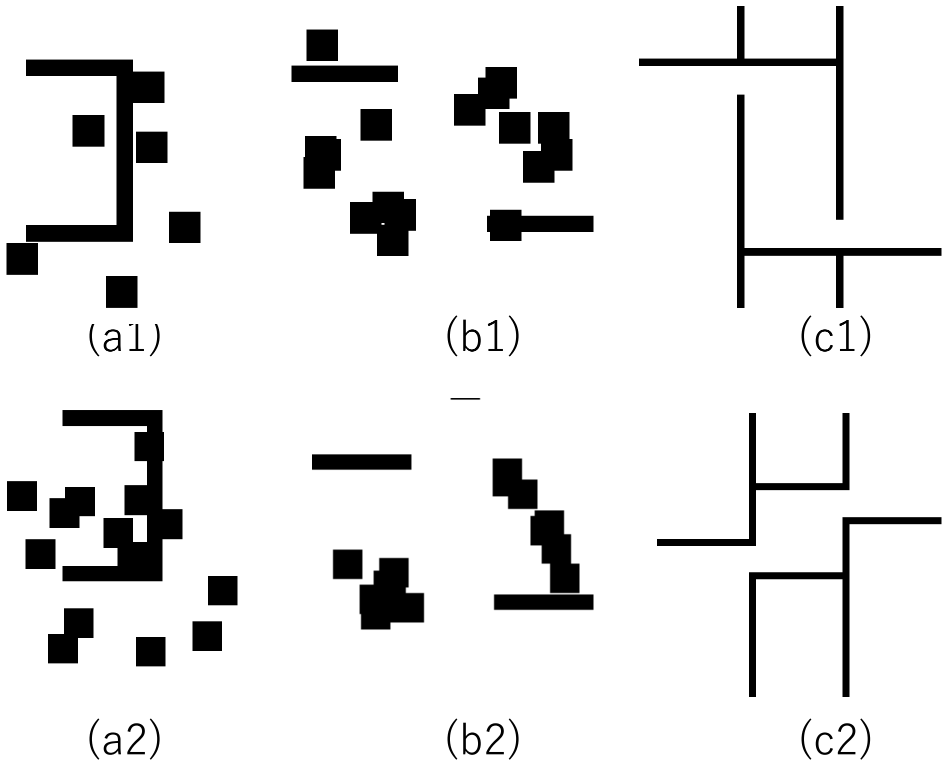


forest



maze

Pre-processed dataset



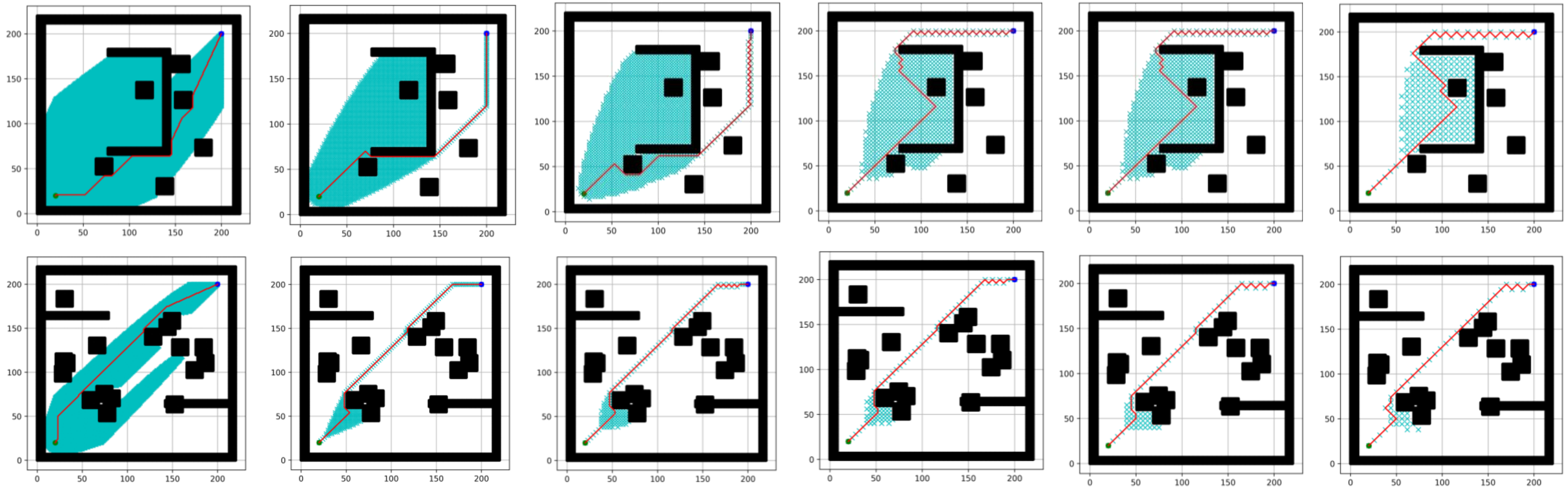
Sample map images of the dataset (a1,a2) bugtrap_forest, (b1,b2) forest, and (c1,c2) mazes.

Raw dataset

Experimental Outcomes

- The proposed model is experimented with different robot motion blocks (n = 1 to 6) for three types of maps.
- When the robot motion block size n increases, the number of searched cells drastically decreases.
- At the same time, the path costs increased a little, but it's negligible.

Motion block's size: $n = 1$ $n = 2$ $n = 3$ $n = 4$ $n = 5$ $n = 6$



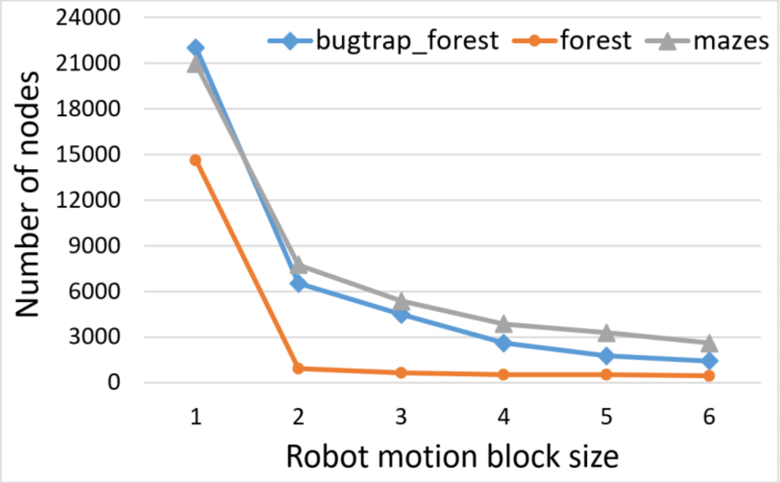
Experimental Outcomes: (cont.)

- Compared to the conventional method the experimental results shows a 92 % to 99 % reduction for time complexity.
- 63 % to 93 % reduction for the number of searched nodes compared to the conventional method.
- The path costs increased a little 0.2 % to 12 %.

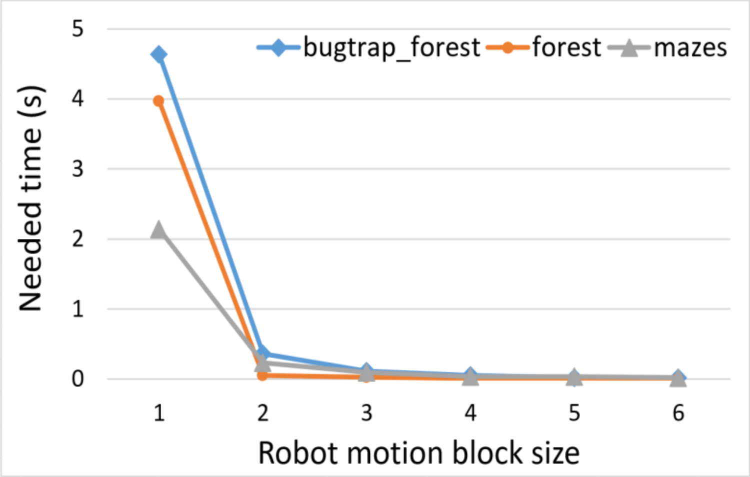
Map type	Robot motion block	n = 1	n = 2	n = 3	n = 4	n = 5	n = 6
bugtrap_forest	Cell searched	22011	6528	4473	2596	1760	1422
	Path cost	261	262	264	292	290	294
	Time needed (s)	4.640	0.359	0.109	0.046	0.015	0.009
forest	Cell searched	14596	904	639	524	517	448
	Path cost	323	326	328	329	334	333
	Time needed (s)	3.968	0.047	0.023	0.009	0.008	0.002
mazes	Cell searched	20950	7736	5355	3852	3275	2604
	Path cost	377	378	376	384	385	390
	Time needed (s)	2.140	0.234	0.093	0.031	0.030	0.015

Experimental Outcomes: (cont.)

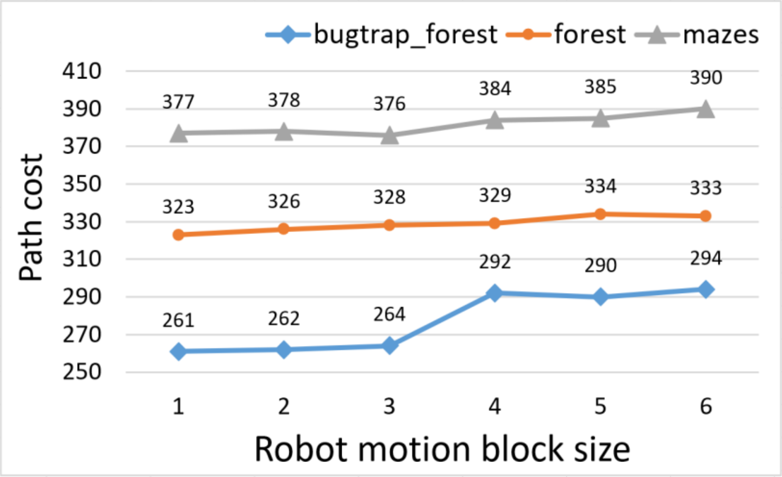
Number of searched nodes



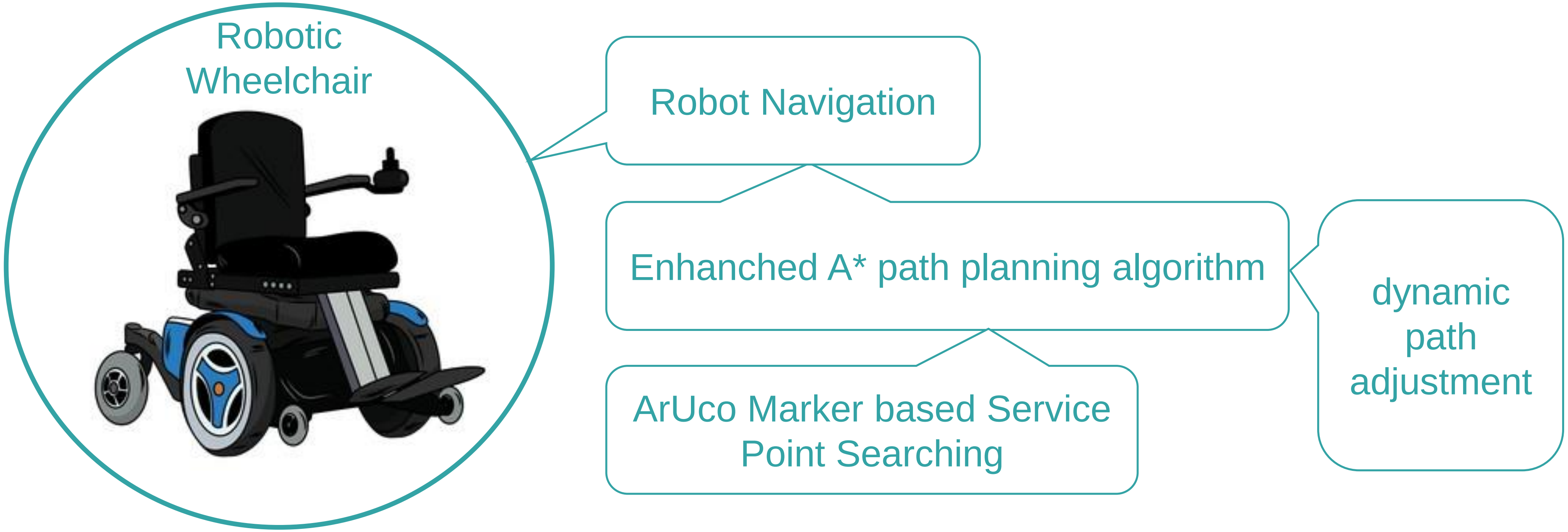
Time needed for searching the goal



Number of nodes for path cost



ArUco Marker based Service Point Searching and dynamic path adjustment

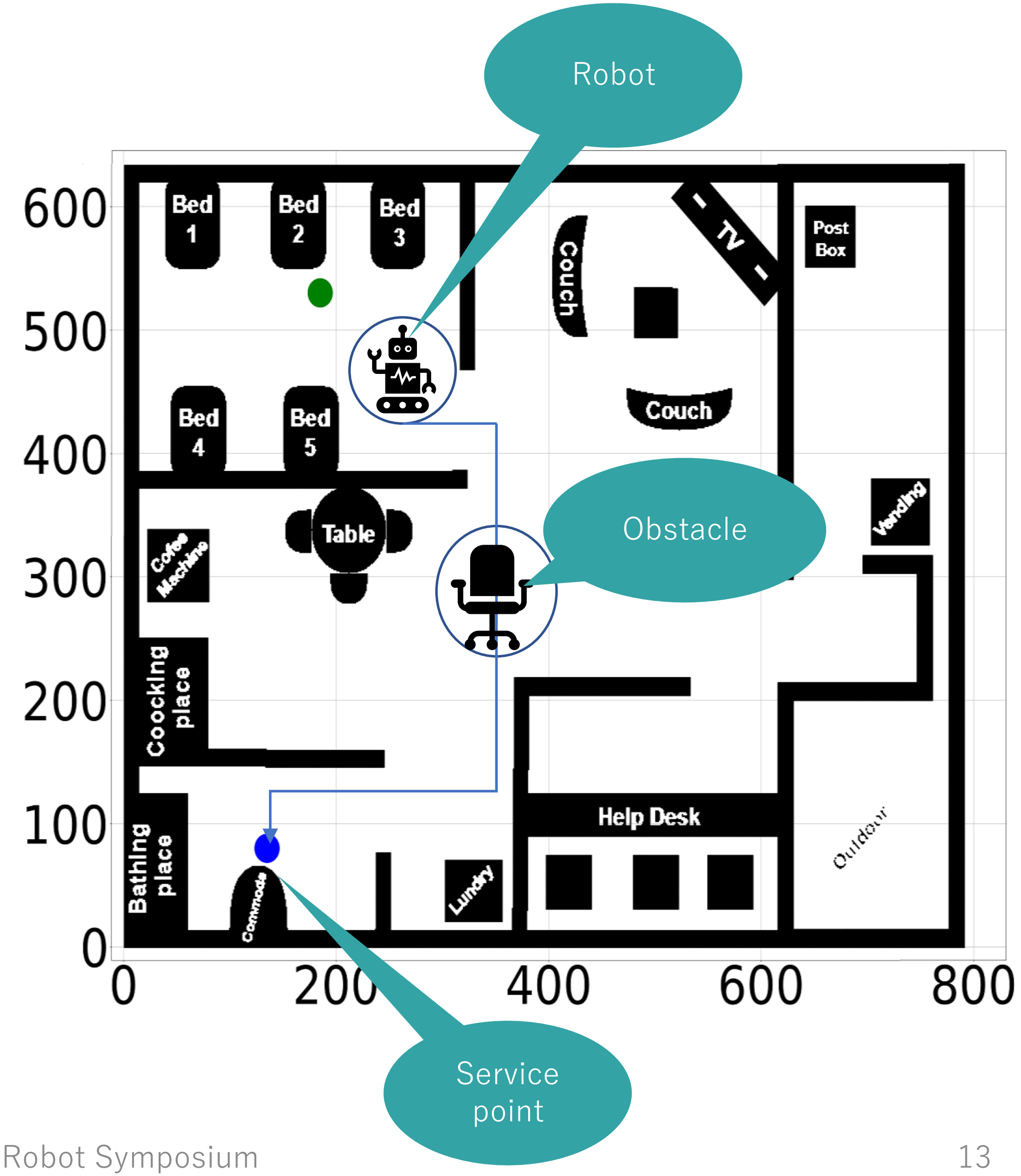


__Publications:__ *(Major conference)*

- **Kabir, Raihan**, Yutaka Watanobe, Md Rashedul Islam, Keitaro Naruse, (2022), "**Service Point Searching for Disabled People using Wheelchair based Robotic Path Planning and ArUco Markers**" In: IEEE 8th World Forum on Internet of Things (WF-IoT-2022), Yokohama, Japan.

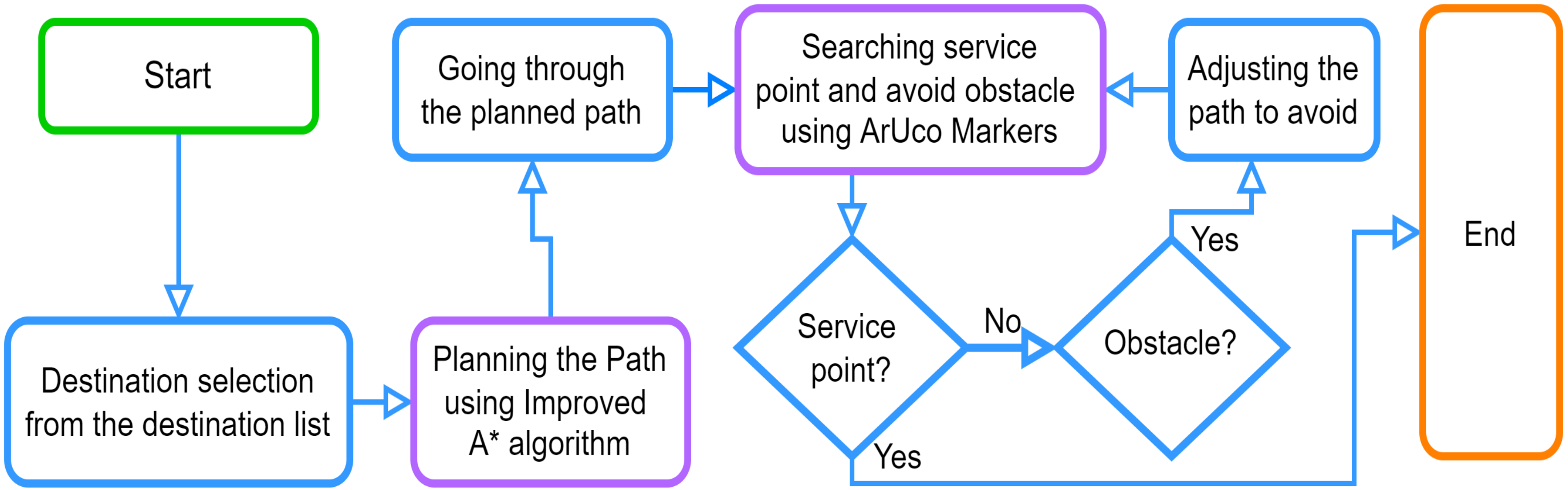
Research objective

- Moving towards to the service point movable obstacles rises a problem of a collision.
- Dynamic path adjustment can be a solve of that problem.
- There are several object detection methods but those are highly computational systems.
- On the other hand, ArUco marker can be detected without any computational overhead. (**light, detection in any direction, and axis of the marker**)



Proposed Method:

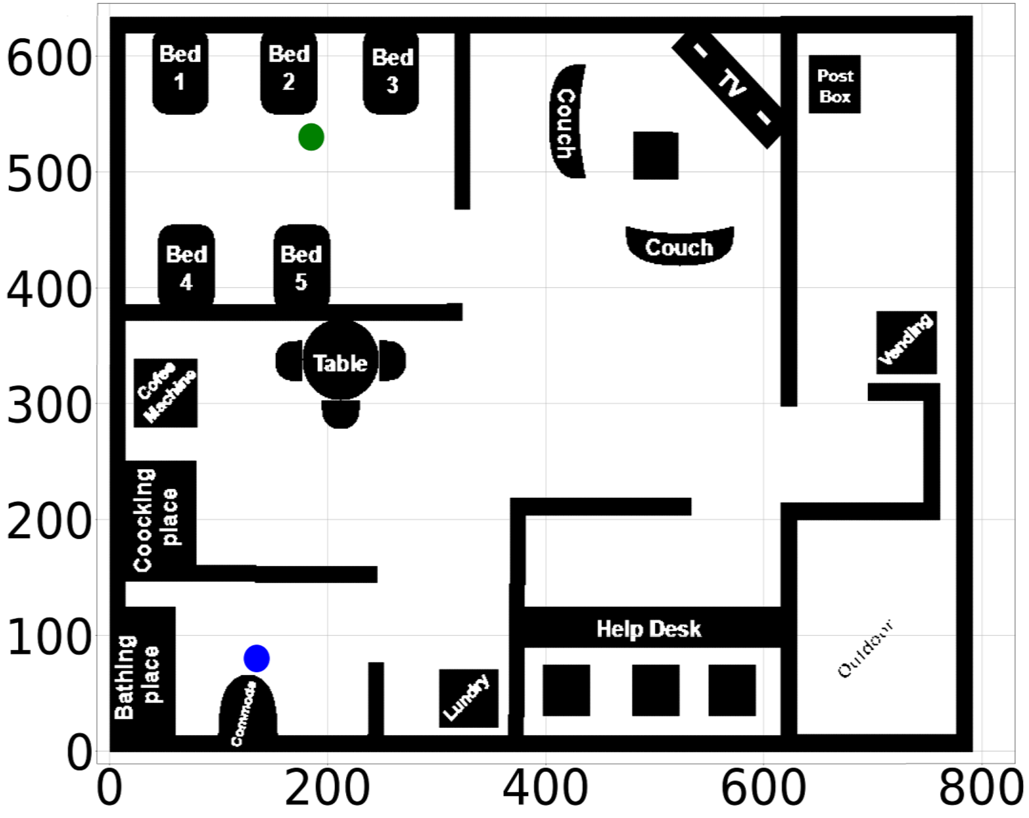
To achieve the research goal, the general block diagram of our proposed model is:



Indoor Grid-map preparation:

- To test the improved A* algorithm and other parts of the proposed model, a grid map of an indoor environment is prepared and had a **dimension of 792 × 635**.

- In this map, the number of **obstacle cells** was **130,743** among the **total number** of cells **(502,920).**



Sample indoor grid map of an elderly care center.
where black pixels = obstacles, white pixels = free space, green pixels = start, and blue pixels = goal.

Service points in an indoor environment:

- Different **colored push button** for different Service points in an indoor environment to go to desired place.

Push Button No.	Service point name	Color	Push Button No.	Service point name	Color
1	Help_desk	G	5	Laundry	Y
2	Vending_machine	O	6	Commode	P
3	Cofee_machine	R	7	Post_box	W
4	Bed	B			

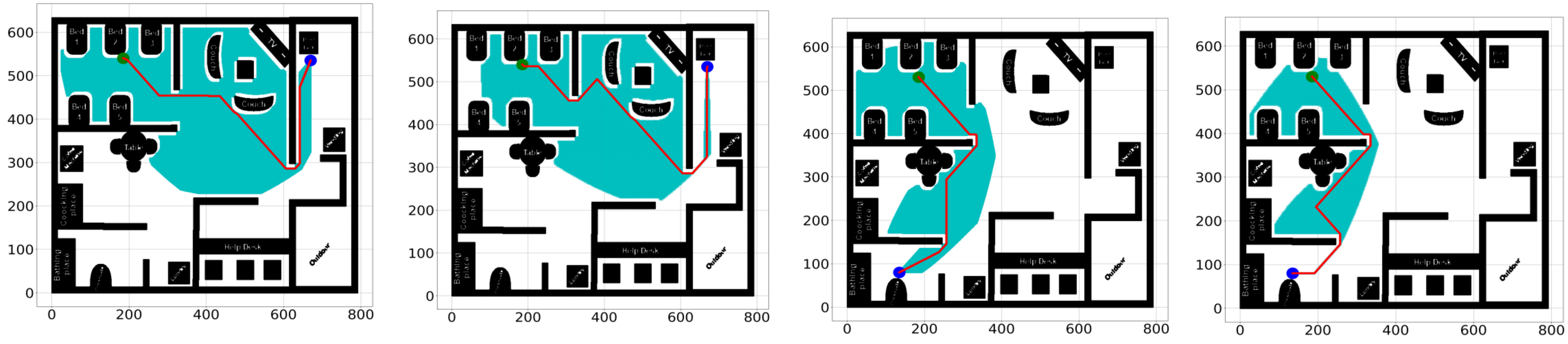
Path planning using improved A* algorithm:

- Compared to the conventional method the experimental results show around an 89% reduction in time complexity.
- 55 % to 65 % reduction for the number of searched nodes compared to the conventional method.
- The path costs increased a little around 0.2 %

Method	Start	End	#Searched cell	Time complexity (s)	Path cost
Conventional A* method	Bed	Post_box	127,934	42.09	691
	Bed	Commode	75,121	24.89	520
Proposed A* method	Bed	Post_box	**58,152**	**5.81**	**696**
	Bed	Commode	**26,210**	**2.76**	**527**

Path planning using improved A* algorithm: (cont.)

- The proposed model has experimented with different start and target positions (service points).
- These are the graphical view of the outcome results.

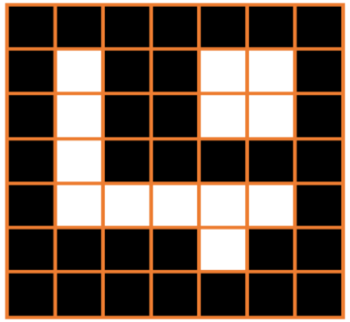


(a1) (a2) (b1) (b2)

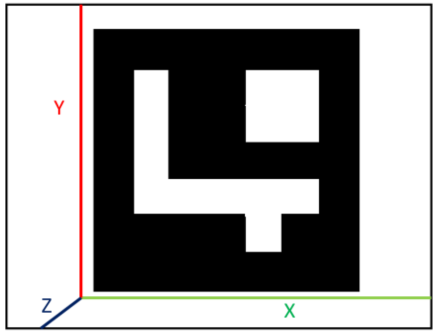
The outcomes of the improved A* algorithm, (a1, b1) conventional A* method, and (a2, b2) proposed improved A* method.

ArUco marker-based service point and obstacle searching:

- In a 3D space, detecting service points and obstacles is challenging and requires high computational power. Also, it is **difficult to accurately detect** every obstacle and service point **under different lighting conditions** from **different angles**.
- However, these challenges can be handled easily using ArUco markers because they meet all the requirements in terms of **versatility, robustness,** and **reliability**.
- A geometrical square 7 × 7 matrix is used to represent the ArUco marker and up to 1024 different marker generation is possible.
- Also, it is possible to detect the axis of the marker.



0	0	0	0	0	0	0
0	1	0	0	1	1	0
0	1	0	0	1	1	0
0	1	0	0	0	0	0
0	1	1	1	1	1	0
0	0	0	0	1		0
0	0	0	0	0	0	0



ArUco marker-based service point and obstacle searching: (results)

- The robotic wheelchair follows the prepared path from one place to the target service points.
- On its path, it also searches for the obstacles and the target service point using the ArUco markers.

(a)

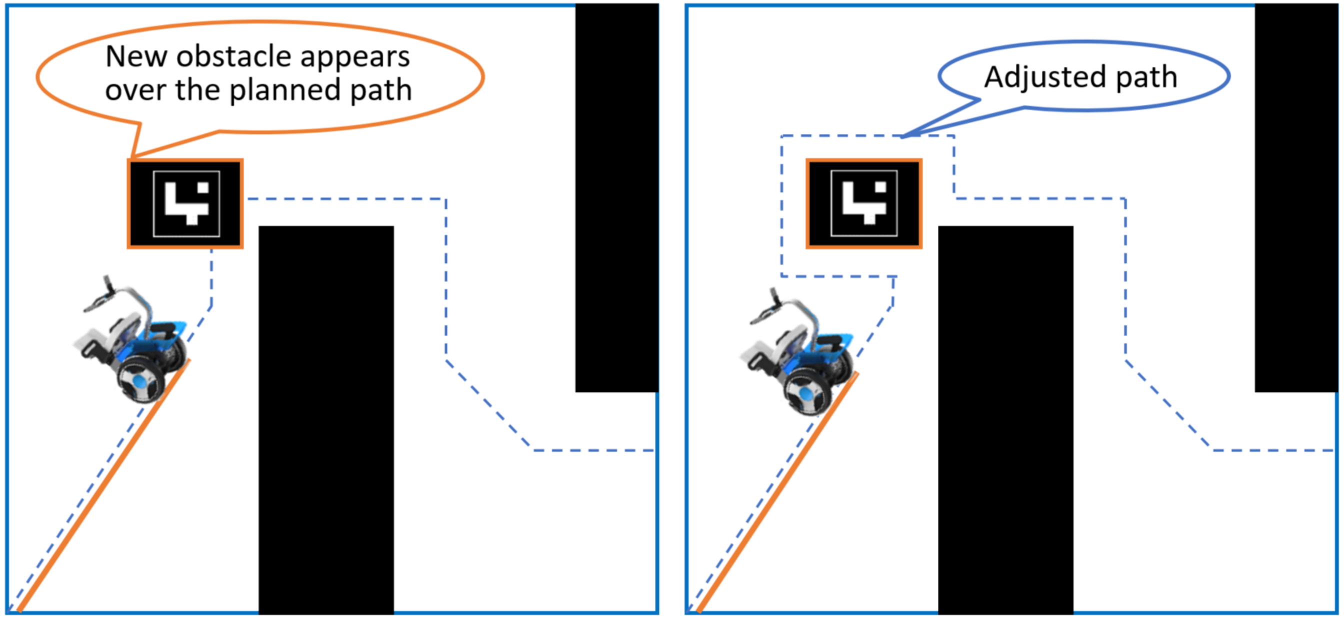
(b)

(c)

Here, (a) = Service point, (b, c) = Obstacle, (c) = Obstacle detection with three axis x, y, and z.

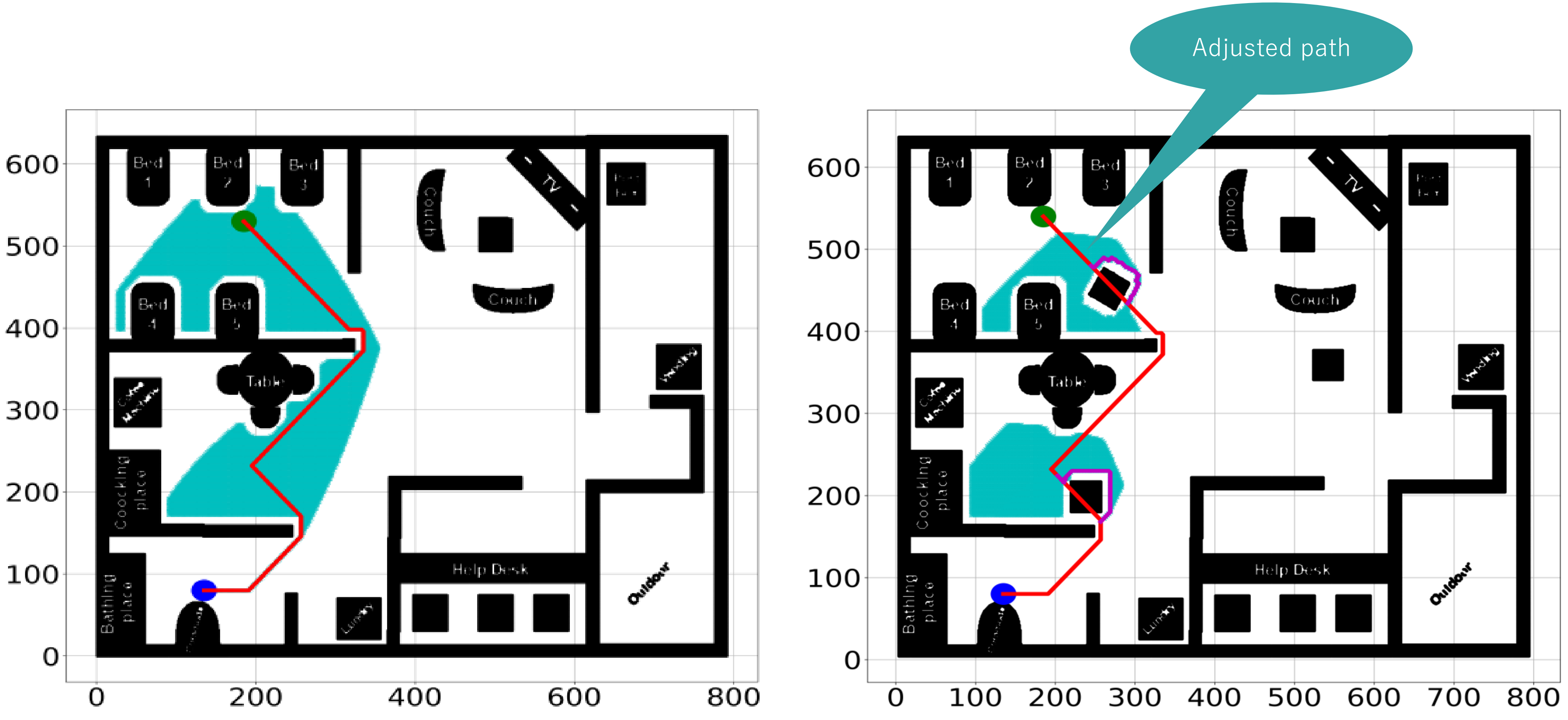
Dynamic path adjustment to avoid obstacles:

- Planning the shortest path is a challenging task, and dynamically adjusting the planned path to avoid obstacles is another challenging task.
- It is very important for a real-time robotic wheelchair to avoid uncertain collisions.
- The proposed approach makes an adjustment in the initially planned path to avoid any collision and makes a new adjusted path dynamically.



Dynamic path adjustment to avoid obstacles: (results)

- Outcome results of dynamic path adjustment method.



Conclusion:

- An enhanced A* algorithm with different **robot motion blocks** is proposed to present a **faster robotic path planning algorithm.**
- The proposed **ArUco marker-based searching method** can **quickly** search the service point and obstacles **without any computational overhead**. Also, **it can detect the axis of the obstacle** which helps the obstacle avoidance system
- The proposed **dynamic path adjustment system** is **helpful** for the automated robot **for collusion-free movement**.
- Comparison of **performances with state-of-the-art methods** shows the effectiveness of our proposed model.
- It is our hope that this proposed path planning method will improve the performance of automated wheelchair’s safety operation.
- However, still there are some lickings in our proposed methods which we will investigate in our future works.

Thank you for your attention