ICCEPM 2024 https://dx.doi.org/10.6106/ICCEPM.2024.1041
The 10th International Conference on Construction Engineering and Project Management
Jul. 29-Aug.1, 2024, Sapporo

Global Route Optimization for Autonomous Robots Using BIM

Shotaro TAKAZOE[1]*, Kazuya SHIDE[2]

[1] *Master Course, Graduate School of Science and Technology, Shibaura Institute of Technology, Japan,* E-mail address: mj23082@shibaura-it.ac.jp
[2] *Prof., School of Architecture, Shibaura Institute of Technology, Dr.Eng, Japan,* E-mail address: shide@shibaura-it.ac.jp

Abstract: In this study, we aim to enhance the navigation method for autonomous systems, including various types of robots and UAVs. Previous work has made progress in enhancing the autonomous flight capabilities of UAVs through both simulation and real-world tests, yet it lacked detailed insights into the aspect of global route planning. To address this gap, we propose a novel method to automatically generate routes for autonomous robots using BIM and 3D city models. Specifically, we create a program that utilizes geometry and attribute information extracted from BIM and 3D City Model to compute the optimal route rapidly. The program automatically generates routes that optimize the efficiency of various autonomous robots by considering the conditions on the routes.

Key words: BIM, 3D City Model, Robot, Route Planning, Indoor Navigation

1. INTRODUCTION

In recent years, the declining labor force in Japan's construction industry has become a serious problem. The Japanese government has been planning to address these issues by enhancing construction industry productivity, with a goal to improve productivity at construction sites by 20% by FY2025[1]. Toward this goal, MLIT (the Ministry of Land, Infrastructure, Transport, and Tourism) has been promoting "i-Construction," which promotes the use of ICT (Information and Communication Technology) and other technologies at construction sites since FY2016, to increase productivity beyond the decrease in the workforce[2]. As a means of achieving this, the use of autonomous robots and UAVs has been attracting attention. Although poised to automate simple tasks, the necessity for manual creation of indoor maps and route setting has limited these vehicles' labor-saving potential. Our previous research demonstrated that the use of BIM (Building Information Modeling) and reinforcement learning algorithms for local route planning can improve the indoor autonomous flight capability of UAVs[3]. However, detailed insights into global route planning were lacking. In this study, we propose a method to automatically generate optimized routes for various autonomous systems, including UAVs and robots. This method utilizes 3D geometry, attribute information, and spatial relationships from BIM and 3D city models to save labor in construction site supervision and maintenance.

2. RELATED WORK

Traditionally , in indoor route planning for robots, classical searching algorithms such as A* are used to find the shortest route connecting two points on a 2D map. Recently, however, the use of BIM for indoor route planning has gained attention, with methods for creating maps from BIM classified into a grid-based map, a network-based map, and their composites.

2.1. Grid-based Map

Chen et al. proposed a method to discretize a 3D geometry of BIM model into a cellular map, setting attribute information to each cell[4]. Although this method can accurately represent geometry by using small cells, it is difficult to use in large-scale environments with multiple buildings because the computation cost of route planning increases exponentially as the number of cells in the map increases.

2.2. Network-based Map

Karimi et al. proposed a method to extract BIM semantics from IFC data and transform it into a network-based map[5]. This method creates a graph with rooms as nodes and doors as edges, searching for the optimal path between two points by considering not only the distance but also the state of each room. This method requires far less data size than a grid-based map and lower computation cost.

Takahashi et al. utilized semantic web technology to propose a method that combines BIM and 3D city models for route planning[6]. This method entails sequential extraction of equivalent instances from BIM and 3D city models for each route planning, which results in relatively high computation costs during route planning. This could pose a challenge for real-time robot navigation applications.

2.3. Grid-Network-based Map

Zhou et al. proposed a map that combines the advantages of the above two methods[7]. In this method, a grid-based map and a network-based map are created from BIM. When the start and destination points are not on the network-based map, the routing algorithm finds the nearest network node on the grid map using the A* algorithm, then quickly finds the shortest path with Dijkstra algorithm.

2.4. Our Approach

In this study, we extend the methods discussed in Section 2.2 and Section 2.3 to achieve large-scale map creation using BIM and 3D city models, and to expedite route calculations by leveraging the advantages of network-based mapping. This enables robot movement between buildings and more extensive path planning. In addition, using a directed multigraph[i] instead of an undirected simple graph enables optimal path selection under various conditions, not just a single shortest path.

3. METHODOLOGY FOR MAP CREATION

This chapter outlines our approach for creating network-based maps and grid-based maps from BIM and 3D city models. Our case study focuses on the first and second floors of Building A, Building B, and Building C at SIT (the Shibaura Institute of Technology) Toyosu Campus. Fig. 1 shows an overview of the map creation process. We have developed an add-in tool for the BIM software Revit to facilitate map creation.

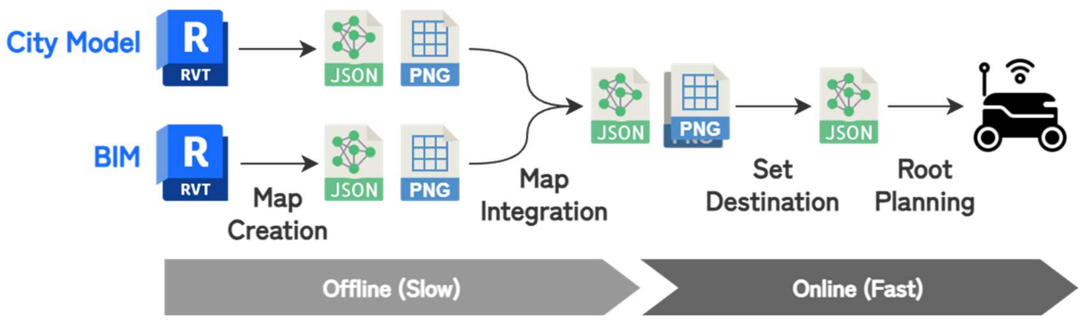


Figure 1. Overview of The Map Creation Process

3.1. Map Creation from 3D City Model

Initially, we generate a network-based map based on a 3D city model provided by the PLATEAU service of MLIT. This network-based map is a directed multigraph with nodes as doors and edges as connections between nodes. The PLATEAU model has four levels of LOD (Level of Detail), with LOD3 and LOD4 include information of openings such as doors. Regrettably, only the LOD2 model, which lacks door information, is available for the area encompassing the SIT Toyosu Campus. Therefore, we manually set door information into the LOD2 model. This process involves the following steps.

---

[i] A **graph** is defined as the pair (N, E), where N is a set of nodes and E is a set of edges. In terms of edge composition, if edge e_{ij} from node n_i to node n_j is considered distinct from edge e_{ji} from node n_j to node n_i, the graph is classified as a directed graph. If not, it is classified as an undirected graph. Furthermore, a graph where there is at most one edge from any node to another is called a simple graph, while any other graph is referred to as a multigraph.

(1) First, import the LOD2 model into Revit (Fig. 2). Following the PLATEAU guidelines, set the door locations and their respective UniqueID in Revit. These UniqueIDs are unique identifiers assigned to each door within the Revit model and are used to integrate the network-based map.

(2) Next, use the PathOfTravel[ii] function to find the shortest path between doors that avoids green spaces, then add it to the map. Also, search the shortest path that ignores green spaces, and if a shorter path is found, add it to the map with an "Undrivable" flag.

(3) Lastly, since actual movement paths are not suitable for route calculation, convert them into Edges that connect vertices. Export the converted map as a JSON file (Fig. 3).

Figure 2. 3D City Model and Photo of the SIT Toyosu Campus


```
{
  "Edges": [ // An array of edges in the graph
    {
      "Start": { // The start node of the edge
        "Id": "cbc03751-bd4f-4a88-a5e6-14f4b0504baf-00172dbe", // UniqueID of the node
        "XYZ": [73.43277225027273, 11.855463981953339, 2.788713910761153], // 3D coordinates
        "IsOpen": false, // Indicates if the node corresponds to a door that is always open
        "IsDestination": true, // Indicates if the end node is a destination
        "DoorProps": null // Properties of the door, if any
      },
      "End": { // The end node of the edge
        "Id": "cbc03751-bd4f-4a88-a5e6-14f4b0504baf-00172c66", // UniqueID of the node
        "XYZ": [76.98920269646698, 2.9442738326889675, 2.788713910761153], // 3D coordinates
        "IsOpen": true, // Indicates if the node corresponds to a door that is always open
        "IsDestination": false, // Indicates if the end node is a destination
        "DoorProps": {"AssetIdentifier": null, "BarCode": null, ...} // Properties of the door, if any
      },
      "MidPoints": [ // An array of midpoints along the edge
        [74.57855987548828, 10.770756721496582, 2.788713910761153], // 3D coordinates
      ],
      "PathLenght": 17.5756526813751, // The length of the edge
      "IsUndrivable": false, // Indicates if the edge is undrivable
      "IsStair": false, // Indicates if the edge is created from stairs
      "IsEscalator": false, // Indicates if the path is created from an escalator
      "RoomProps": {"AssetIdentifier": null, "BarCode": null, ...} // Properties of the room, if any
    }, ...
  ],
  "Grids": [ // An array of grid-based map metadata
    {
      "DoorIds": { // An array of doors in the grid
        "cbc03751-bd4f-4a88-a5e6-14f4b0504baf-44332cca": [12, 5] // UniqueID of the door
       },
      "RoomProps": {"AssetIdentifier": null, "BarCode": null, ...} // Properties of the room, if any
    }, ...
  ]
}
```


Figure 3. Example of JSON data of a Network-based Map

[ii] **PathOfTravel** is a Revit API function to create a shortest path between two points. The plan view is considered as a grid of 20 cm in length and width, and the A* algorithm is used to search for the shortest path from the start point to the end point.

3.2. Map Creation from BIM

This section outlines the methodology for generating Network-based and Grid-based maps from the BIM models of Buildings A, B, and C (Fig. 4). The process involves several key steps:

(1) First, for each floor of the building, create door-to-door paths. In this process, exclude doors that are not necessary for map creation (e.g., bathroom door). Additionally, mark doors that can be autonomously navigated by robots—like automatic sliding doors and doors that are always open—with an "Open" flag. Automatic doors are distinguished based on the presence of the Uniclass “Pr_75_30_23_05 Automatic sliding door operators” parameter.

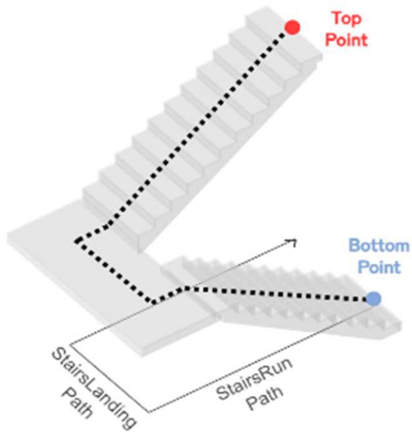
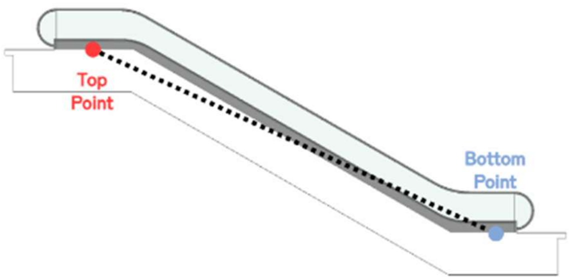
(2) Next, create paths connecting the floors, then add them to the map. These inter-floor paths typically involve stairs, ramps, escalators, and elevators. In this step, use the Revit API to create stairs paths, ramp paths, and escalator paths using the method shown in Table 1. Elevators are not included in this study.

(3) Then, convert these paths to edges, setting the path length and attributes of rooms or stairs, ramps, and escalators to the edges. Additionally, set each door's attributes and UniqueID to the nodes. Then export the transformed map as a JSON file.

(4) Lastly, create grid-based maps representing geometry of each room in a 2D grid, and then export these grid-based maps as PNG images. This map allows for the calculation of a path from the destination to the nearest node without using PathOfTravel, enabling robots to route finging independently of Revit.

The steps of creating a network-based map and grid-based maps for Building A are shown in Fig. 5. Following the same methodology, maps for Buildings B and C are also created. This approach ensures the generation of detailed and accurate maps for Buildings A, B, and C and facilitates efficient route planning.

Table 1. Stairs paths, Ramp paths, and Escalator Paths Creation Method

Stairs Path	Stairs instances have “StairsRun” elements and “StairsLanding” elements, and the centerline and height of these elements can be obtained using the Revit API. Using these centerlines and heights, the path of the stairs is created. Paths are also created between the top point and bottom point of the staircase and the door in the same room.	
Ramp Path	Ramp instances have a “LocationCurve” element from which the path of the ramp can be created. Also, as with stairs, a path is created between the top point and bottom point and the door.	
Escalator Path	Unlike stairs and ramps, there are no escalator components in Revit, but are created freely by the user. In the escalator family used in this study, the position is defined by top point and bottom point, so the line segment connecting these two points is used as the escalator’s path. Also, as with stairs, a path is created between the top point and bottom point and the door.	

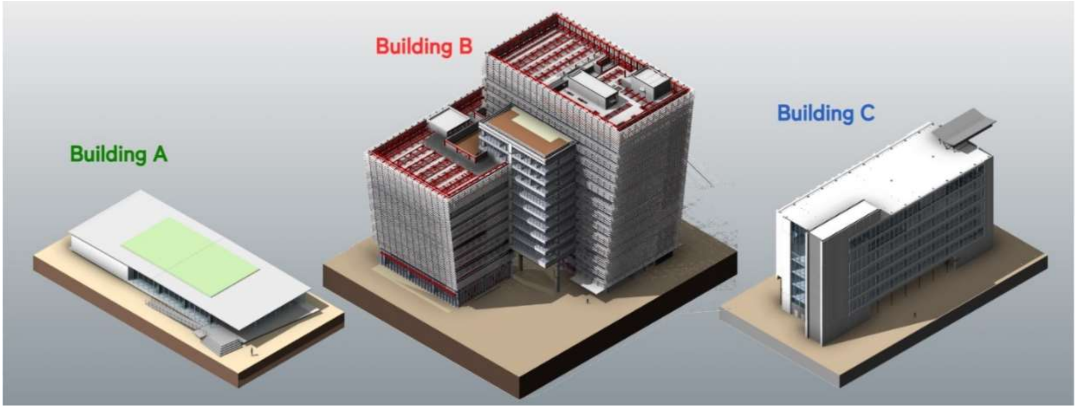


Figure 4. BIM model of Building A, Building B and Building C

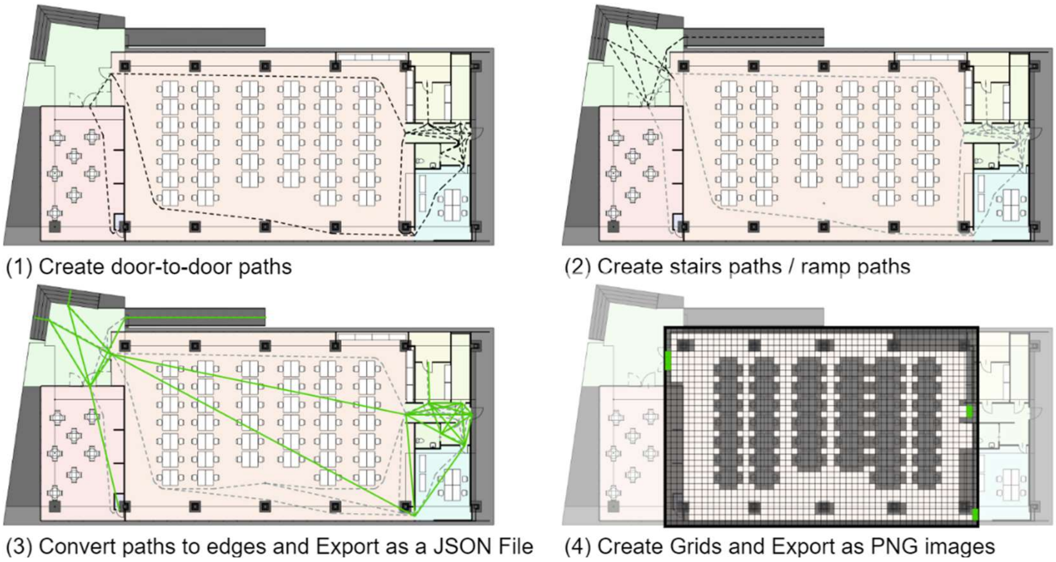


Figure 5. Network-based Map and Grid-based Map Creation (Building A)

3.3. Map Integration

By utilizing the UniqueID as a key, we link the nodes of the network-based maps from Sections 3.1 and 3.2, thereby integrating these maps into a unified network-based map. This map forms a directed multigraph, with nodes representing doors and edges delineating connections between these nodes.

3.4. Destination Setting

We set destinations D1, D2, D3, and D4 as shown in Fig. 6, and create paths connecting doors within the same room using the grid-based map from Section 3.2. Like the door-to-door paths, these paths are converted to edges and added to the network-based map.

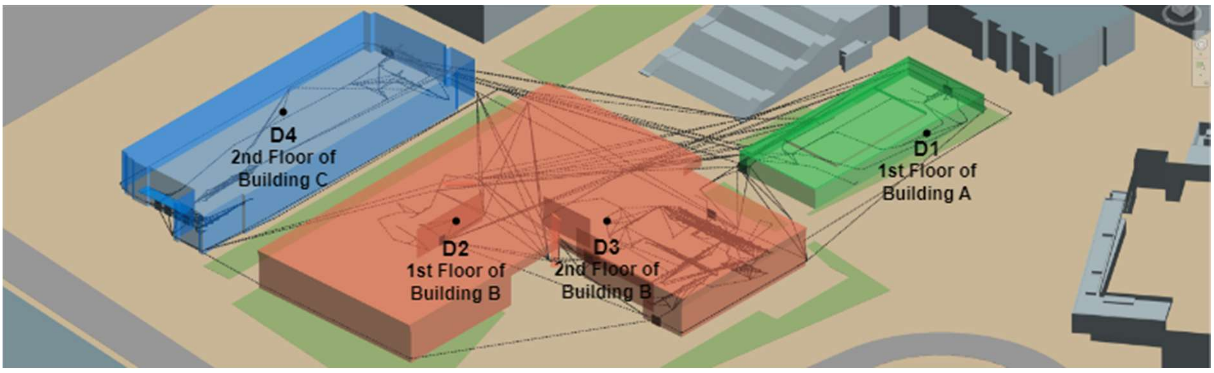


Figure 6. Integrated network-based map and Destinations

4. ROUTE PLANNING

In this chapter, we assume the movement of Wheeled Robots, Humanoid Robots, and UAVs along routes, and experiment with the maps created in Chapter 3 to investigate whether optimal routes can be found for each. Table 2 details the specific conditions under which these robots can pass through the nodes and edges.

Table 2. Conditions for passable Edge and Node

		Wheeled Robot	Humanoid Robot	UAV
Node	Manual door	×	○	×
	Automatic doors / Permanently open doors	○	○	○
Edge	Stairs / escalators	×	○	○
	Ramps	○	○	○
	Undrivable areas (e.g., where bushes are planted)	×	×	○

4.1. Shortest Route Between Start and Destination

In the first experiment, with D1 set as the starting point and D2 as the destination, we determine the shortest route connecting these points. This problem can be solved as SPP (Shortest Path Problem): after removing edges from the graph that do not satisfy the conditions of Table 2, the shortest route in the remaining graph is searched using Dijkstra method.

The obtained route is shown in Fig. 7, which shows that the Humanoid Robot and the UAV choose the route through the stairs, while the Wheeled Robot chooses the longer route through the ramp.

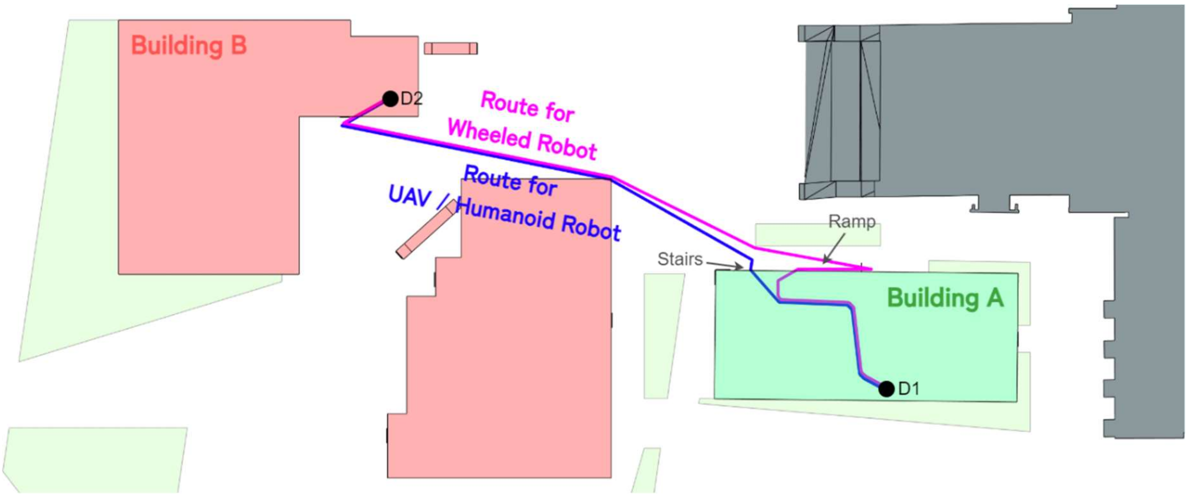


Figure 7. Shortest Route Between D1 and D2

4.2. Shortest Circuit through Multiple Destinations

In the next experiment, with D1 set as the starting point and D2, D3, and D4 as the destination, we identify the shortest travel route connecting these destinations. This problem can be solved as TSP (Traveling Salesman Problem)[iii]: after removing edges from the graph that do not satisfy the conditions of Table 2, the shortest travel route in the remaining graph is found for using an annealing method.

The obtained route is shown in Fig. 8, which shows that the UAV can move in both directions because it flies over the escalator, but the Humanoid Robot chooses a longer route because it is restricted to move only one direction on the escalator. Note that circuit for the Wheeled Robot cannot be found, because it must use the stairs or escalator to access D3 and D4.

[iii] **TSP** and **MTSP** are optimization problems focusing on finding the most efficient routes for visiting cities. TSP addresses the challenge for one salesman to visit all cities and return to the start, while MTSP involves multiple salesmen, aiming to minimize total travel distance and coordinate multiple routes efficiently. As cities increases, both problems see a significant rise in the difficulty of obtaining exact solutions, generally necessitating the use of approximation algorithms.

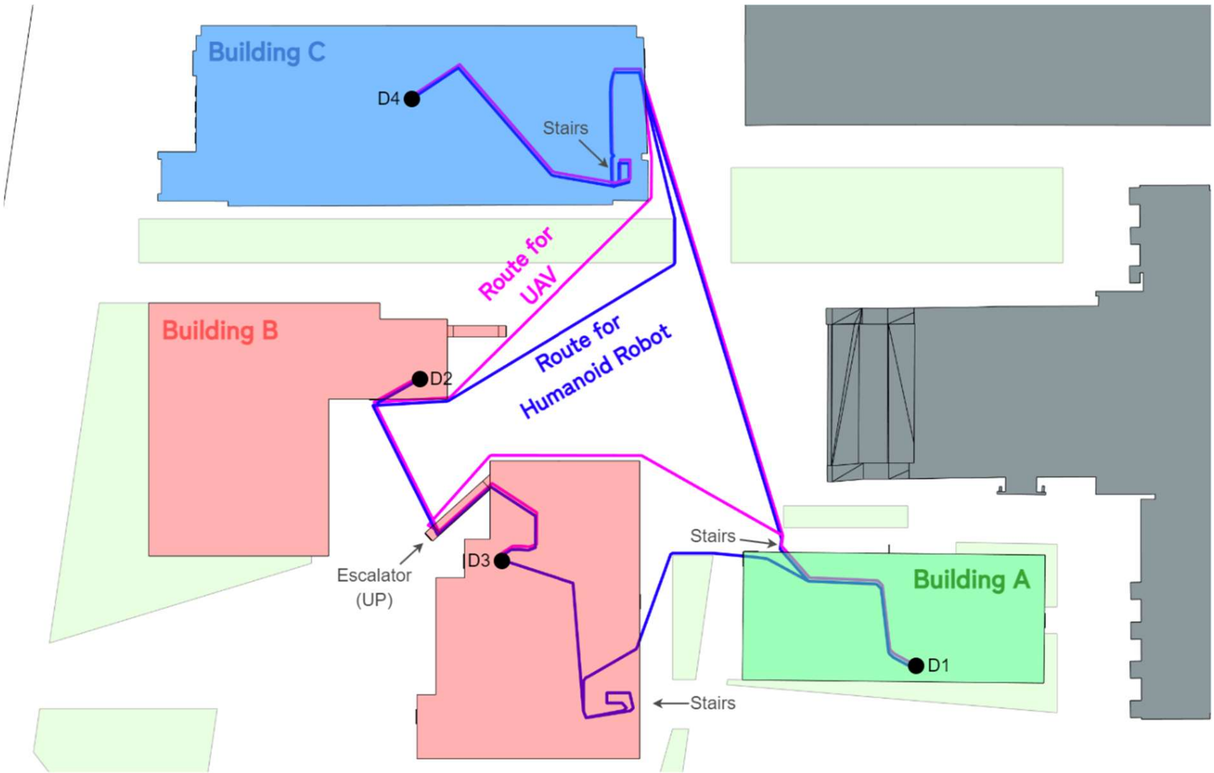


Figure 8. Shortest circuit through D1, D2, D3, and D4

4.3. Route Sharing and Optimization with Multiple UAVs

In the final experiment, employing two UAVs under conditions analogous to those in Section 4.2, with the goal of sharing the destination points and reducing the average flight distance. This problem can be regarded as MTPS (Multiple Traveling Salesmen Problem). We utilize ACO (Ant Colony Optimization) to find the circuits for UAV-1 and UAV-2[8].

The obtained routes are shown in Fig. 9, where UAV-1 visits D4, UAV-2 visits D2 and D3, and returns to D1, showing that the flight distance is reduced compared to Section 4.2.

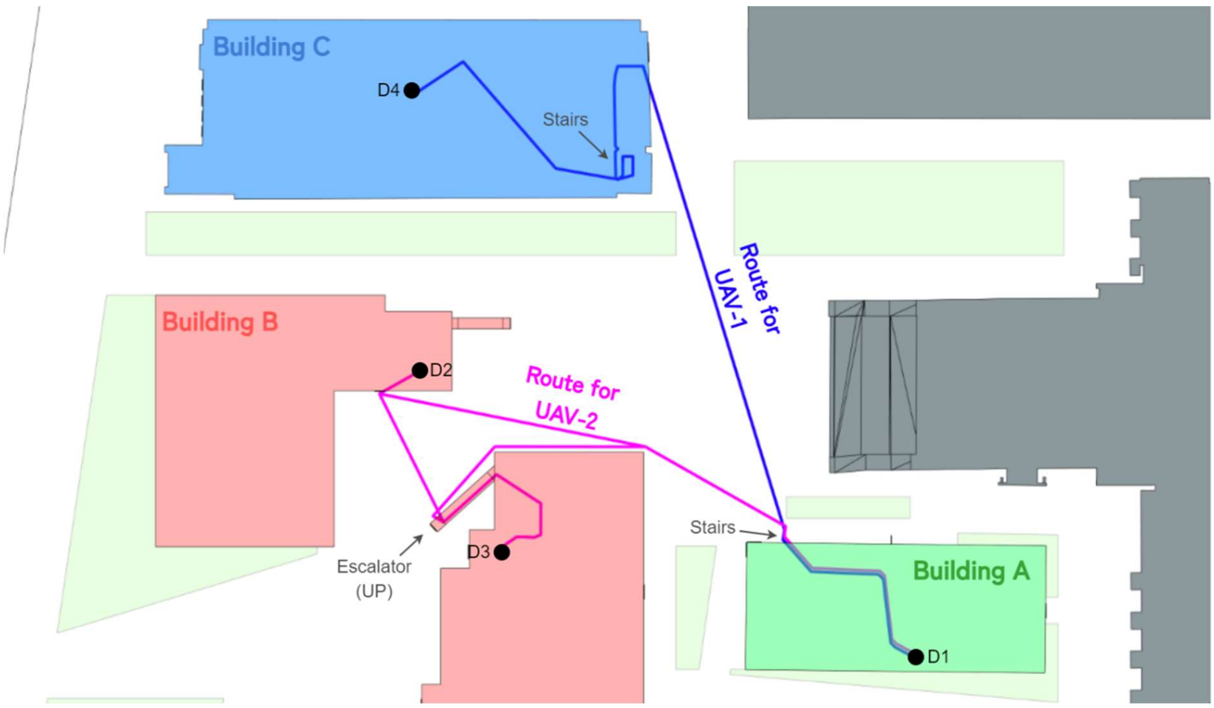


Figure 9. Shortest circuit through D1, D2, D3, and D4 with two UAVs

In this chapter, we use the method we propose to verify the feasibility of route planing for different types of robots. The results show that it is possible to efficiently search for the optimal route under certain conditions.

5. CONCLUSIONS

Aiming to promote the use of ICT and improve productivity at construction sites, this study proposes a method to automatically generate robot routes by utilizing geometry and attribute information of BIM and 3D city models. Using a network-based map with low computational cost, we show that the method can rapidly search not only a route between two points, but also a circuit of multiple points and a route assuming multiple robots. The rapid computational ability suggests that in the event of unforeseen obstacles during robot navigation, alternative routes can be quickly recalculated. In addition to route search, we believe that this method can also be used to investigate the range of possible robot movements from an arbitrary point.

The following points are left as future problems. Firstly, a method to incorporate elevators into the route planning process needs to be explored. Firstly, Elevators are an essential element for movement in multi-story buildings, and incorporating their use into route planning is expected to make route generation even more practical. Next, enriching the graph with information collected by robots during navigation could further refine the optimization of routes. For example, robots could assess and append the congestion level of rooms they traverse, allowing others sharing the graph to select less crowded routes. Lastly, exploring Semantic Web technologies could offer a more rational and efficient method for map integration, beyond the current use of door UniqueID. Such technological advances are expected to further improve the accuracy and computational speed of route planning in complex environments.

REFERENCES

[1] CAO, Government of Japan, ”日本再興戦略2016”, 2016

[2] MLIT, Government of Japan, “I-CONSTRUCTIONの推進”, 2022

[3] Shotaro Takazoe, Kazuya Shide, “Autonomous Drone Navigation Using Reinforcement Learning and BIM”, Proceedings of the 46th Symposium on Computer Technology of Information, Systems and Applications, R85, 2023

[4] Qingxiang Chen, Jing Chen, Wumeng Huang, “Pathfinding Method for An Indoor Drone Based on a BIM-Semantic Model”, Advanced Engineering Informatics, vol. 53, pp. 101686, 2022

[5] Sina Karimi, Rafael Gomes Braga, Ivanka Iordanova, David St-Onge, “Semantic Optimal Robot Navigation Using Building Information on Construction Sites”, 38th Proceedings of the International Symposium on Automation and Robotics in Construction, pp. 57-64, 2021

[6] Ko Takahashi, Kazuya Shide, “セマンティックwebによる建物データ運用のための研究 −ジオメトリデータを対象として−”, 芝浦工業大学大学院理工学研究科建設工学専攻修士論文, unpublished, 2023

[7] Zhou X., Xie Q., Guo M., Zhao J., Wang J., “Accurate and Efficient Indoor Pathfinding Based on Building Information Modeling Data”, IEEE Trans. Ind. Informt., vol. 16, no. 12, pp. 7459-7468, 2020

[8] Pan Junjie, Wang Dingwei, “An Ant Colony Optimization Algorithm for Multiple Travelling Salesman Problem”, First International Conference on Innovative Computing Information and Control, vol. 1, pp. 210-213, IEEE, 2006[9] GitHub, “ganyariya/MTSP”, https://github.com/ganyariya/MTSP_ACO, (accessed 2024.02.02)

[10] William J. Cook, “In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation”, Princeton University Press, 2012