IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 20 November 2023, accepted 1 December 2023, date of publication 5 December 2023, date of current version 13 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3339822

RESEARCH ARTICLE

Automated Generation of Site Map Graphs for Robot-Inclusive Tactile Marker Placement

MATTHEW S. K. YEO, M. A. VIRAJ J. MUTHUGALA, (Member, IEEE), AND MOHAN RAJESH ELARA

Engineering Product Development, Singapore University of Technology and Design, Singapore 487372

Corresponding author: Matthew S. K. Yeo (matthew_yeo@mymail.sutd.edu.sg)

This research is supported by the National Robotics Programme under its National Robotics Programme (NRP) BAU, Ermine III: Deployable Reconfigurable Robots, Award No. M22NBK0054 and also supported by A*STAR under its ''RIE2025 IAF-PP Advanced ROS2-native Platform Technologies for Cross sectorial Robotics Adoption (M21K1a0104)'' programme.

ABSTRACT Navigation is a primary requirement for mobile robots as they encounter hazards during operation. To improve safety for robotic deployments, hazard alert systems could be integrated into their spatial environments to reduce robotic computational load and hardware requirements, while providing benefits to visually-impaired human users in the area. Modified tactile paving markers known as the Passive Auto-Tactile Heuristic (PATH) tiles are configured for robotic usage to aid in their navigation and hazard avoidance along with a specialized Tactile Sensor Module, while providing tactile cues to visually-impaired human users. This paper proposes a novel algorithm for generating standardized layouts of 'safe routes' from site plans of the robot's workspace as input. The creation of 'safe routes' is implemented by extracting site boundaries and outlines of interior obstacles of an input site plan, generating of the site's Medial Axis (MA) through its connectivity graph and overlaying the corresponding PATH tile types. This implementation is explored using the Rhinoceros3D CAD program, Grasshopper development platform and associated plugins for processing the site plan and visualization of the eventual routes. The algorithm is tested on 5 sites of varying spatial functions for validation.

INDEX TERMS Automated, design for robot, graph generation, robot-inclusive designs, robot navigation, tactile paving.

I. INTRODUCTION

Mobile service robots are being deployed for increasing types of menial and repetitive tasks as technology improves. They are seen as a potential supplement to manpower shortages for dull, dirty and dangerous jobs [1]. These robots aid in multiple tasks such as cleaning, area inspections, transports of goods and navigation of unknown environments, among other resource-intensive work [2], [3]. However, most environments have not been built with their operation in mind, leading to damage to robot and properties, maintenance down-times and reduced productivity for the tasks they were made to supplement the lack of human labour in. Public spaces such as hotel lobbies, hospitals, libraries and some offices contain multiple spatial hazards that lead to robotic issues of collision damage, subpar navigation or incomplete tasks [4], reducing the potential of using robotic aid to mitigate monotonous jobs and dangerous maintenance tasks.

Mobile service robots encounter multiple hazards during operation, both from the robot's end and how the robot interacts with its environment. Deficiencies on the types of sensors used, or lack of hardware or software functions can cause the robot to fail, mainly by collisions or power outages [5]. Other negative repercussions the robot can arise during their deployment within their workplaces, such as restricted access due to haphazard layouts that require human intervention, or collision with objects that the robot does not have the sensors to detect for. Problem resolution often comes under the responsibility of the robot operator to reposition or restart the robot as a means to troubleshoot robot failures,

The associate editor coordinating the review of this manuscript and approving it for publication was Chih-Yu Hsu.

and the operators should be trained to pre-empt these types of robot failures as part of their job [6].

A. BENEFITS FOR BOTH HUMANS AND ROBOTS THROUGH IMPLEMENTATION OF ROBOT-INCLUSIVE TACTILE PAVING

A large proportion of mobile service robots implement visual equipment for them to detect and avoid obstacles in their environment during operation. However, robots working in dynamic and ever-changing environments may face issues in deciphering inputs with visual clutter / noise, or areas that are crowded with moving people or traffic. Thus, crowded areas fail to benefit from having robotic deployments to perform the repetitive and drudging tasks such as cleaning or safety inspections. This paper proposes the implementation of a modified form of tactile paving which the VIB people use for navigation and wayfinding as an alternate means of hazard alert system for robots.

There is thus potential in utilizing tactile means for alerting robots to its surrounding hazards [7], [8]. The benefits of using tactile inputs for alerting robots on spatial hazards in their surroundings would mainly be: 1) offloading an extent of computational power from the robot from visual data processing onto spatial infrastructure, and 2) the potential of implementing mobile service robotic aid in crowded areas, or in zones with erratic or extreme lux values [9].

On the end of benefiting the VIB human users, there have been efforts to provide architectural changes for the VIB people based on their conditions and requirements for navigating and hazard detection cues during travel [10], [11]. Through the implementation of such robot-inclusive tactile paving, these architectural changes could allow VIB people to access these public spaces using these tactile cues for increased range of accessible spaces and perhaps improve their quality of life, while also enabling robotic hazard detection. Some of the existing methods that VIB people rely on for navigation such as smart glasses [12] which fully limited vision users are unable to utilize. Some VIB users have guide dogs and their robotic variants [13], [14], [15], but those require additional logistics to accommodate them. Other methods utilize their own hearing augmented with additional equipment [16], [17], which compromises their sense of hearing for conversation or other audio cues. Intelligent floors require substantial logistics and extensive embedding of electronic equipment that require active maintenance and resources [18]. Utilizing modified tactile paving would be a form of passive infrastructural modification where the active sensing is on the mobile robots or the visually-impaired users' white canes. These factors provide basis for relying on the VIB users' white cane as the preferred mode of navigation.

To mitigate such issues for robotic hazard detection and navigation, we can thus consider the use of tactile markers as an alternate hazard alert system. The existing tactile paving designs could be modified to include robotic detection and usage. These modified tactile markers could be then used to passively denote 'safe routes' within the robot's work environment, similar to the existing tactile paving that the Visually-Impaired or Blind (VIB) people utilize as cues to find a safe path around environmental obstacles without relying on memory or excessive electronic equipment [19], [20], [21]. This would reduce the complexity on the robot's end by modifying the environment to preemptively warn the robots of environmental obstacles, instead of utilizing computationally expensive visual detection methods to detect objects outside of their view.

B. BACKGROUND INFORMATION FOR THE EXISTING TACTILE PAVING FOR THE VISUALLY-IMPAIRED

The existing tactile paving alert system for VIB users was created by Seiichi Miyake in 1965 and implemented in Okayama, Japan in 1967 [22]. Standards for the design and implementation of the tactile paving are usually referenced from the ISO/FDIS 23599 [23] and CEN/TS 15209 [24]. Morevover, countries such as Singapore [25], [26], New Zealand (NZ) [27], United Kingdom (UK) [28], the United States [29] and Japan [30] have their own customized national standards for tactile paving [31], [32]. Without a global standard for tactile paving layouts, each country's implementation of tactile paving is different in layout and significance, and is prone to errors or oversights [33]. There is currently no precedent against modifications to tactile paving patterns. Examples of common tactile tile designs is seen in Fig. 1. A summary of the various tactile paving types and function is shown in Table. 1, based on information gathered from [31] and [34].

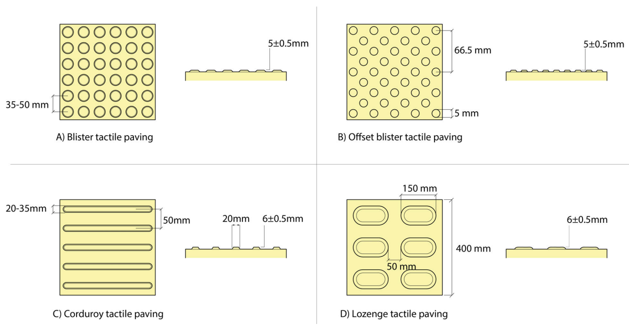

FIGURE 1. Existing types of tactile paving tiles used by the visually-impaired or blind (VIB) people.

With the above requirements in mind, a suitable route generation method is researched to determine suitable robot-inclusive tactile paving placements in public zones. This would enable VIB users and robots to utilize this type of spatial modification for navigation and hazard detection in the subsequent section II-A. This modified tactile paving is not meant to supersede other forms of existing sensor methods. Instead, this system would provide the architectural elements that mobile robots could use with the relevant tactile sensors. This would extend robotic hazard detection of its work environment in their tactile capabilities.

TABLE 1. Summary of existing tactile paving types.

Tactile Indicator type	Typical Locations	Usage / Function
Blister paving	Before crossing interfaces	Alert VIB users of intersection ahead, to proceed with care
Offset blister paving	Near level change of railway platform edge	Warn VIB users of platform edges / level drop
Corduroy hazard warning paving	Near obstacles or level changes	Warn VIB users of hazards ahead, to proceed with care
Lozenge paving	On-street Platform edges	Warn VIB users of platform edge of light transport systems
Cycleway paving	Beginning and end of cycleway and pedestrian intersections	Alert VIB users of pedestrian pathways and cycleway paths
Guidance / directional paving	Safe route around obstacles	Identify safe route to VIB users, provide directional cues and avoid obstacles

Implementing modified tactile paving for robots in public spatial infrastructure would also require a proper method in their placement in the environment to avoid sending wrong signals or damage to users of the zone. This paper thus seeks to:

1) Explain the rationale for integrating robot-inclusive modified tactile paving in spaces for robot deployments,
2) Provide a basis for arrangement/configuring the modified tactile alert system within the given zone,
3) Application of the arrangement algorithm on various site plans for validation.

C. ROBOT-INCLUSIVITY PRINCIPLES FOR MODIFIED TACTILE PAVING

Robot designers are to take note of the robustness, pricing, and sensor performance and system integration to improve the robots' productivity when implementing multiple sensor arrays on mobile robots. The typical way of resolving operational problems often involve improving the robot's equipment and programs to enable the robot to explore and detect its surroundings better. Examples of such improvements include deployments of advanced control methods [35], complex hardware enhancements [36], reconfigurable mechanisms [37], and algorithmic improvements [38], [39]. The robot eventually becomes more complex and costly to surpass environmental challenges during operation. The 'Design for Robot' (DfR) methodology would help mitigate such issues during robot deployments, by modifying the environment for better robot productivity.

Using the Universal Design (UD) guidelines [40] as a basis, a set of DfR principles were created for producing more favourable working zones for robots to passively enhance their tasking outputs [41], [42]. These outputs can be quantified using metrics of percentage for area covered, or time taken for waypoint travel routines. Existing approaches were looked at by Ivanov et al. [43], where they discussed design suggestions for mobile service robots like waiter robots and drones being deployed in hospitality premises such as hotels and restaurants. There are other examples of changing the architectural elements to enhance robotic productivity, mostly in the areas of robotic cleaning of floors [41], vertical garden robotic maintenance [44], false ceiling maintenance [45] and drain inspection [46] respectively.

A simplified version of the overall DfR methodology can be seen in Fig. 2. The two main approaches of DfR utilize the deductive and inductive processes. The deductive approach determines the robot's constraints uses the provided floor plans and drawings of its work environment to evaluate the robot's productivity in the zone before deploying the actual robot. The inductive approach uses physical trial runs of the robot onsite, and obtaining the insights from the results and operational feedback. Design recommendations for both the robot and the environment can then be generated using the DfR robot-inclusive principles, which are namely safety, activity, manipulability, observability and accessibility [47]. For this paper, it would focus more on the deductive stage of the DfR process.

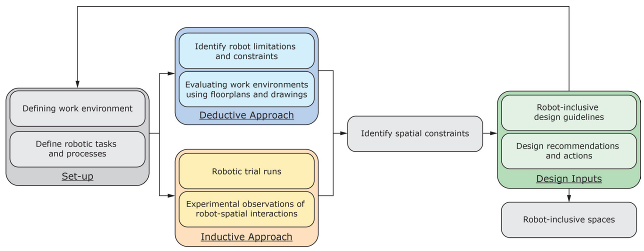

FIGURE 2. Design for Robot (DfR) methodology detailing the deductive and inductive stages and its involvement in creating robot inclusive spaces.

The 5 robot-inclusive principles are based on ensuring protection for robots, environment and the other users of the zone during operation of the robots. The Safety principle aims to cut down on hazards found in the robot's work zone and reduce incidents of robot or environmental damage during deployments. The Observability principle determines how the object layout can be considered and made more distinct for robotic navigation and hazard detection. The Activity principle strives to implement a cohesive system whereby a collaborative work environment can be achieved for both robot and human users simultaneously. Accessibility principle aims to generate a barrier-free environment for the robot to perform tasks in a given spatial zone. The Manipulability principle focuses on the methods that objects which the robots interact with during their operation can be handled more readily. Listed guidelines for each section varies for different spatial environments, and are to be updated as technology and spatial requirements evolve and adapt over time. These guidelines enabled the development of a robot-inclusive tactile paving system as an alternative hazard alert system for mobile service robots.

For the purpose of hazard detection, the modified tactile paving would focus on the Observability and Accessibility DfR principles. Under the Observability DfR principle, methods of making the environmental hazards more apparent for robots by changing qualities in their spatial aspects are considered. Some examples include, but are not limited to, addition of opaque markers onto glass panels, or utilizing Quick-Response (QR) codes as location markers for robotic usage. In this case, the modified robot inclusive tactile paving would provide robots with environmental cues on the 'safe route' within their work zone, or enable the robot to detect the hazard cues pre-emptively.

Moreover, the implementation of a modified robot inclusive tactile paving system is aligned with the Accessibility DfR principle to allow the mobile robots to access a higher proportion of the workspace and circumvent the workspace hazards easier. Implementing this system of updated tactile cues would also lead architects to enable their buildings to be more inclusive to the VIB users, and provide a more holistic usage of the architectural space.

The main information to encode upon the tactile pavings would be: 1) hazard type, and 2) distance from hazard. The tactile pavings should be made modular for easier installation and potential for combination with different tiles to convey varied hazard data types. The existing tactile paving tile system implemented by the VIB human users can be used as reference for a potential robot-inclusive tactile paving system known as Passive Auto-Tactile Heuristic (PATH) tiles, expanded in section II-B.

Section II describes the rationale and criteria for generating obstacle-free routes from site plans, the tactile paving hazard alert system known as the Passive Auto-Tactile Heuristic (PATH) tiles, its corresponding tactile sensor module and the algorithm for placement of these PATH tiles within a built environment with a provided site plan. Section III provides and discusses the results of the algorithm conducted on five existing architectural floor plans. The paper is then concluded in Section IV.

II. MATERIALS AND METHODS

A. GENERATION OF OBSTACLE-FREE ROUTES FROM PROVIDED SITE PLANS

For navigation tasks, it would be ideal for the route to be direct and compacted for faster completion of intended goals, and less energy wastage. This could be applied to mobile robotic tasks of checkpoint tracking and exploration. An idealized route for robots to follow would be one that allows them to:

- avoid all obstacles in the environment that would cause collisions or damage,
- be pre-emptively informed of hazards in its path
- reach a majority/all points of the accessible zone (if the robot is to perform area coverage tasks).

By providing the shortest access route through the space as the expedient route, the mobile robots would then be able to split off from this shortest route of the zone to navigate and access their required zones. This would involve finding the main spline that avoids all obstacles and walls of a room, and maintaining the connectivity among the rooms for this route of expediency. This type of route exists by taking the center line among all walls and objects present known as the MA [48]. This MA, also known as a 'topological skeleton', allows for preservation of a space's connectivity, linear dimensions, and its topology. This would determine a path within a site plan that avoids all obstacles and walls present, by collapsing the spatial boundaries into its topological MA. The robot's size can be overlaid afterwards to check if it can fit within the space as part of its accessible zone.

In cases where the site plan layouts lead to difficulties in generating a MA, approximations of its MA could be used instead. It is especially useful for sites with irregular or self-intersecting zones. It is fortunate that these cases are limited with physical limitations of building designs, with preference for straight walls, and radial circulation from a common entry point. Multiple algorithms exist to obtain the MA of a given shape. For our case, the site plan can be abstracted into its walls and interior furniture, to generate a bounded shape to perform MA operations upon.

MA of shapes can be generated with a myriad of algorithms, the typical methods used are MA Transform (MAT) [49], [50], Straight Skeleton (SS) [51], [52] and its related algorithms, and Voronoi Diagrams (VD) [53], [54].

MAT works by applying a distance transform on the shape. MAT calculates the distance from each point within the shape to that of its closest point on the boundary. The points which are locally maximal would lie upon the approximated MA. The MAT can be optimized using other types of transform such as the Euclidean distance transform algorithms [55], [56].

SS approximates the shape's MA by iteratively shrinking the shape boundary inwards, while maintaining the boundary's topology. The eventual collapsed line traces out the shape's 'skeleton' and is the approximation of the shape's MA. Other similar skeletonization algorithms work in a similar way of iteratively reducing the shape's boundary inwards [55], [57] without compromising on the shape's connectivity and topology and reduce a shape into its core skeleton lines to obtain the MA approximation.

VD is conducted by splitting the shape's boundary curve in intervals to obtain the points for Voronoi cell generation [58]. Only the Voronoi edges that are strictly within the shape are considered for obtaining the shape's MA. The selected Voronoi edges are joined together to create the shape's MA. The MA' degree of refinement can be adjusted by changing the intervals by which the boundary curve of the shape is split. The smaller the intervals between the points on the boundary curve, the more refined the eventual boundary curve would be. The resulting line is then smoothened to obtain the shape's MA [59], [60].

These path generation methods can be hybridized or combined with other algorithms to mitigate their individual drawbacks. However, limitations for mobile robots navigating a complex environment within public spaces still remain due to erroneous environmental inputs, or constraints in robotic hardware. Other constraints that mobile robots encounter during operation mainly involve complexities of real-time navigation, level of complexity of its work environment, accuracy in discerning obstacles in its environment, its own capabilities and energy efficiency, hardware limitations and other safety regulations [61]. These operating conditions would also affect the choice of path planning methods used by the mobile robots for their tasks [62], [63]. These conditions are also dependent on whether the robot has a pre-loaded map in its software for navigation, and whether the robot requires dynamic obstacle detection during its work.

B. PASSIVE AUTO-TACTILE HEURISTIC (PATH) TILES

The Passive Auto-Tactile Heuristic (PATH) tiles [64] are a catalogue of modified tactile paving tiles designed for conveying basic environmental hazard data to robots with a dedicated tactile sensor. The implementation of PATH tiles reflects the DfR methodology of improving robotic productivity through the use of robot-inclusive architectural design changes, rather than solely improving the robots' functionalities [41], [44], [65]. This proposed infrastructural system of PATH tiles is a novel non-intrusive architectural integration, aiming to improve mobile robots' ability in detecting and reacting to environmental hazards in their deployment sites. Implementing this modified tactile system could also provide an impetus for correcting installation errors in existing tactile paving layouts.

The PATH tiles are categorized into two types, one for hazard type (hazard PATH tiles), the other for conveying distance from PATH tiles to an obstacle (distance PATH tiles) seen in Fig. 3 for differentiated responses to different hazard placements [64]. This divergence between hazard and distance PATH tile designs enables customization between the various safety distances and hazard types based on the object layouts for different areas. For the distance PATH tiles, it presumes that the robot takes its current location on the PATH tile as the corresponding distance from the hazard.

A PATH tile is sized 300mm by 300mm, and has reflectional symmetry in the axis of travel. Counterparts to existing tactile indicators have been designed, as seen in the bump design of the guidance tile, moving obstacle, and vertical level change PATH tiles, to be analogous to the existing guidance tile, blister tile, and offset blister tile designs respectively. Truncated blisters would indicate upcoming hazards, while raised bars or stripes aligned to the direction of travel are used to indicate safe routes on which to travel in the updated tactile paving system to reduce the learning curve for human VIB users. The blister PATH tiles would be located near the hazards to alert users in advance, while the guidance PATH tiles would be located on a route that minimizes obstruction and lead the users to places or objects of interest within the zone. For the blister and guidance tiles, they can be arranged in a similar manner to convey the same type of information on which path to follow, being analogues of existing tactile paving. Implementation of the newly modified PATH tile layout would be shown in Section III of this paper.

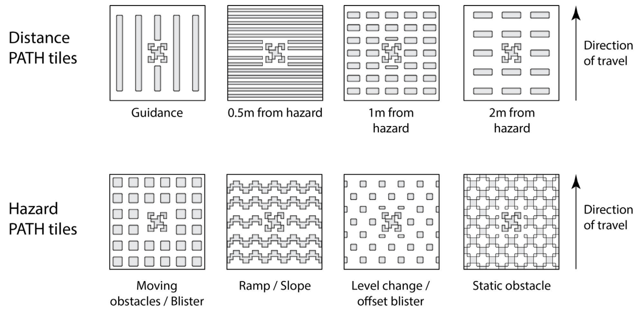

FIGURE 3. **Novel robot-inclusive Passive Auto-Tactile Heuristic (PATH) tiles - Top: Guidance, Distance PATH tiles; Bottom: Hazard PATH tiles.**

As tactile paving layout implementation and regulations differ among countries, there would be a need to create an algorithm for the placement for PATH tiles to preemptively avoid similar problems, especially since current mobile service robots have less leeway in dynamic decision-making during deployment. The algorithm would take in the inputs of the mapped site, and the robot's detection range based on its sensor array to provide a PATH layout for future robots to avoid hazards and travel to its required way-points more efficiently. This consistency of signals conveyed would aid VIB users in interpreting the updated tactile paving. This was reflected in the design choice where truncated blisters are still being used to mainly indicate hazards, and raised bars or stripes are being used to indicate paths of travel in the updated tactile paving system [29]. The PATH tile placement algorithm is explained in Section II-D.

C. TACTILE SENSOR MODULE

In order to enable mobile robots to detect and ascertain tactile cues from the PATH tiles, a dedicated tactile sensor module (TSM) is required. This TSM would function similar to the comb of a music box for surveying tactile cues. [64]. In Yeo et. al. [64], a TSM as seen in Fig. 4 was created using multiple contact limit switches arranged on a linear frame, and to be mounted on a mobile robot for PATH tile detection. A tuned Chebyshev Graph Neural Network (CGNN) model [66], [67] was implemented for parsing tactile cues from the PATH tiles for interpreting them for avoiding the hazards ahead. However, that paper only focused on the TSM and CGNN implementation rather than the method of placement within public zones.

D. GRAPHING OF SITE PLAN

It is ideal to generate the PATH tile layout arrangements through automation rather than manual methods. However, rules for arranging furniture to improve robotic navigation and wayfinding safety are still needed for a proper and robot-inclusive configuration for the given zone. By considering the robot's range of detection by its sensors, we can determine the effective detection and movement field for it to capture data from its environment within a specified period

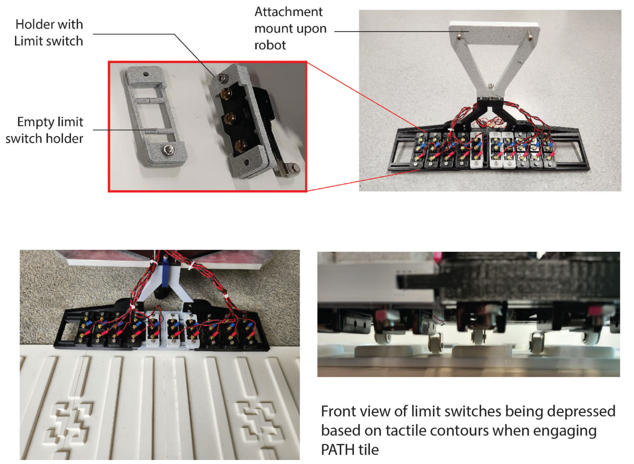

FIGURE 4. Tactile Sensor Module for detecting PATH tiles; source: [64].

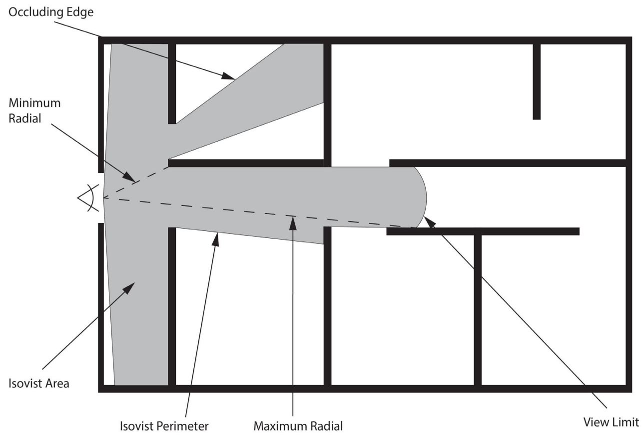

FIGURE 5. Terminology of isovist graphs.

of its run. To this end, graphing the site into an abstract form would help reduce complexity for the robot to parse and travel through the site. The site can be converted into a graph that shows the connections between multiple points of interest within the site, a network of path possibilities [68]. Two zones are connected topologically if they share a common wall or doorway, whereby it is described as 'mutually adjacent' and 'mutually accessible' respectively in Dawes et. al [69]. Graphing a site into its simplified diagram allows for easier computation of an eventual optimal path methodology of robots, that could utilize the methods described in [70] for determining passages or routes in a site with multiple location nodes. This algorithm is created for the purpose of laying out robot-inclusive PATH tiles for hazard alerts for both visually-impaired/limited vision human users and robots in a standardized manner.

The use of isovist graphs would aid in separating the space for centroid node identification [71]. Isovist graphs are 2-dimensional planar representations of the visible area viewed from a singular view point within a space, limited by occluding boundaries and view limits from the current position. The terminology of isovist graphs is seen in Fig. 5.

The eventual graph for our scenario would take the cumulative isovists and room separation lines to obtain a form of spatial separation between different zones within the site for further analysis and graphing, to place location nodes and determine the placement of the PATH tiles as a form of hazard alert system for mobile robots. It is assumed that if the robot can travel between 2 points without interruption or obstacles (e.g. furniture or walls), the entire space can be collapsed into a single location node at its weighted centroid for the simplified graph for the given site plan.

The summarized method of graphing an input site plan would be:

1) Splitting the overall site into its constituent rooms
2) Assigning graph nodes to the centroids of each discrete zone,
3) Connecting the nodes together to generate a connectivity graph, and obtaining refined graph and determining connectivity for each node using wall/obstacle boundary outlines for MA generation,
4) Determine standardized placement and layout of PATH tiles for the specific site plan.

These steps are further explained in the subsequent subsections II-D1 to II-D4.

1) DETERMINING CONSTITUENT ROOMS FOR GIVEN SITE

For step 1, the overall zone is to split to its constituent rooms if it spans an area with doors and openings. The robot's view is modelled as a point source initially positioned at the entrance of the site. The use of isovist graphs would aid in separating the space for node identification [71], [72], [73], by consideration of the room boundaries and obstacle bounding boxes as occluding bounds to the robot's sensor view. Isovist graphs are beneficial in providing a logic in creating convex spaces on which further analysis can be conducted in terms of visual connectivity and distance to other points based on line-of-sight connections [74].

Open zones that are much larger than the robot's footprint would be assigned a node at the zone's weighted centroid. If obstacles are present within the zone, the unobstructed area would be subdivided into quadrilaterals wherever able to make it simpler for the centroids of the subdivided spaces to be obtained. Objects are determined to be obstacles if they occlude the robot's view ahead (perceptional obstacles), or if they are physical obstructions (navigational obstacles).

The robot's view is modelled as a point source initially positioned at the entrance of the site. The use of isovist graphs would aid in separating the space for node identification [71], [72], [73], by consideration of the room boundaries and obstacle bounding boxes as occluding bounds to the robot's sensor view. Isovist graphs are beneficial in providing a logic in creating convex spaces on which further analysis can be conducted in terms of visual connectivity and distance to other points based on line-of-sight connections [74].

To obtain the centroid of a given discrete zone, we would model robot's Point of View (PoV) as 360° point source to conduct isovist operation. Discrete or unconnected obstacles would have a bounding box to remove irregular outlines and for easier processing of the site plan. Radial lines from the initial robot's view are extended until it encounters the room corners or an obstacle's bounding box. The centroid for each discrete zone is obtained by finding the PoV point where its isovist ray lengths to the room corners or obstacles are made equal to form convex polygons. Thus, for given room corners C_i, for $i = 1$ to n, we are to find the point $P(x_p, y_p)$ where all distances between P and the corners are equal, as given in (1). Here, (X_{C_i}, Y_{C_i}) are the coordinates of the i^{th} corner.

$$\text{Find } P(x, y) \text{ such that } d(P, C_i)\ \forall\ i = 1 \text{ to } n \text{ is equal}$$
$$\text{where,}\quad d(P, C_i) = \sqrt{(x_p - x_{c_i})^2 + (y_p - y_{c_i})^2} \tag{1}$$

Unconnected obstacles would also split the zone into multiple quadrilaterals and their corresponding centroid nodes. An example can be seen in Fig. 6.

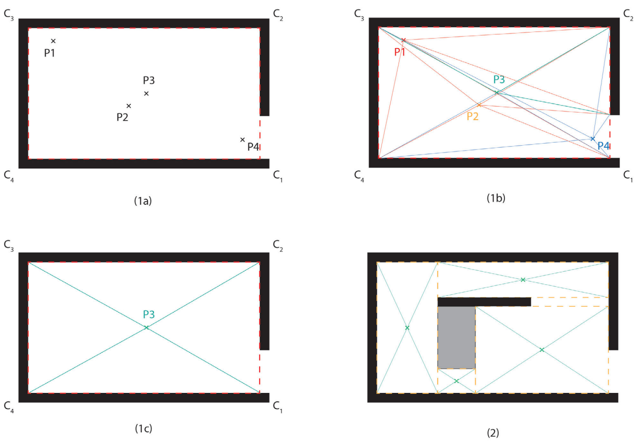

FIGURE 6. Process for obtaining discrete zones and their centroids; 1a) For a given room boundary of 4 corners (C1-C4) indicated by dotted red line, model robot's point of view (PoV) as 360deg point source to conduct isovist operation, 1b) Assume isovist ray length = infinity for rays to hit room corners / obstacles, 1c) Move robot's modelled PoV point such that isovist ray length are equal for isovist rays that connect to room corners to obtain discrete zone's centroid (in this case P3); 2) Multiple discrete zones within single room will be generated due to presence of obstacle (modelled here as grey rectangle), or interior walls.

Separated rooms and occluded edges are then marked out using dotted site separation lines. In this way, a site can be split into multiple zones, each specific to the view of the robot and how it would view or assess the space with its sensor array, limited by its detection range and field of view. Occluded zones are split from the main space using dashed lines. Corridors and narrowed spaces would have their own separation lines.

A discrete zone is defined as the largest non-occluded isovist perimeter within a room boundary. The weighted node of each discrete zone would be located at the centroid of its isovist rays, or simply the center of a discrete zone if the room itself has no obstacles within it. This process can be scaled up for site plans with multiple rooms and corridors. Corridors or passageways that connect multiple rooms would also be tagged as a separate 'room', shown in Fig. 7. For our scenario, obstacle bounding boxes would be added onto the input map even if they are out of the robot's planar isovist view but still present as a navigational obstacle.

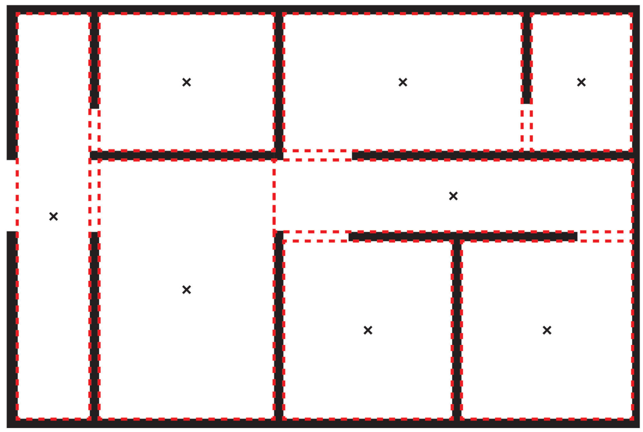

FIGURE 7. Generated centroids of discrete zones, corridors would be tagged with their own centroids.

2) NUMBERING NODES BY PROXIMITY AND CONNECTIVITY

For step 2, the individual rooms would then be numbered by their degree of connectivity from the ingress point of the site. This connectivity degree is determined by its proximity from the ingress point, and how many distinct rooms a robot has to pass through from entering the site in order to reach the room of interest. An example is seen in Fig. 8, with example site's ingress point located on the left. With reference to Fig. 8, the rooms are labelled based on the number of steps needed for the radial connectivity isovist (RCI) to terminate (room 2a compared to room 2b; rooms 4a and 4b compared to room 4c), and by their proximity to ingress point (room 4a versus room 4b).

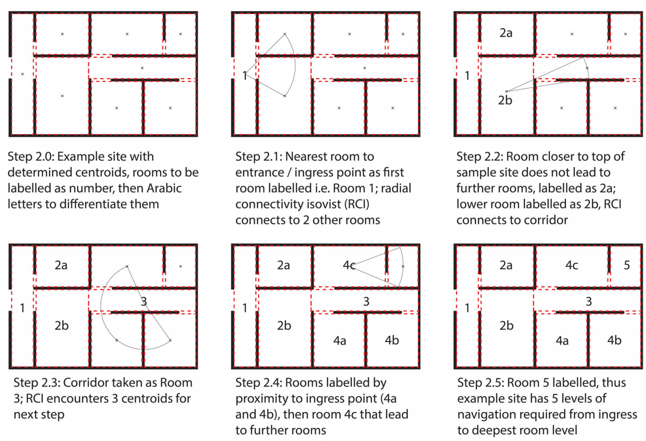

FIGURE 8. Rooms numbered based on their proximity and connection to preceding rooms.

Graph vertices for the connectivity graph would mainly be the Euclidean distance between the nodes to its neighbouring nodes in other rooms.

3) DETERMINING PATH TILE PLACEMENT

For step 3, after the zone has been sub-divided using the rules and guidelines for segregating the zone detailed in the previous section, we obtain a site's connectivity graph. This connectivity graph can then be used to assign PATH tiles at the crucial locations where the robot might have problems determining the next actions to take.

The generated MA is thus dependent on the resolution and regularity of the initial boundary curve, which can be mitigated by pre-processing the input site boundary curves to make it more regular and defined wherever able. Disconnected or un-linked branch lines may be generated due to the boundary curve irregularities or artifacts introduced during the Voronoi cell generation and their subsequent intersection lines. Post-processing steps of branch pruning, or skeleton simplification may be needed to cull the unwanted vertices and branch lines, and to touch up the obtained MA curves.

For complicated shapes (or site plans in this scenario), the zone could be sub-divided beforehand to split up the zones for easier VD computation and MA generation. Complex or irregularly shaped zones may lose some detail when generating their MA using VD. The polyline curves generated by VD could give sub-optimal results from the presence of narrow passageways/corridors, or site plans that have multiple self-intersections that could cause similar self-intersections in the generated MA that have to be resolved manually in the post-processing step.

These methods of MAT, SS and VD each have their advantages and drawbacks, mainly in terms of the types of shapes the algorithm can process, computational efficiency and the required level of accuracy for the generated MA. However, VD would be preferred here for creating the MA for our case as it can handle shapes with concavities, and ability to preserve topological features of the original shape. VD would also provide information on the shape's connectivity and ease of access from other zones within the same shape.

For this paper, Blister PATH tiles would be placed at main entry points and path bifurcations. Guidance PATH tiles would be placed along the vertices of the MA lines between the nodes. This would provide an initial layout of the medial axes on which further designs of PATH tiles could be added upon. If multiple robots are being deployed within the same area, their positions within the site could be modelled as a Voronoi seed to see how it would affect the generated route for the other robots as well.

4) IMPLEMENTATION OF PLACEMENT ALGORITHM USING RHINO3D GRASSHOPPER PLUGIN

Step 4 of the alogrithm involves the placement of the PATH tiles described in Section II-B. To test the placement algorithm, images of 2D site plan were input into Rhinoceros 3D CAD program (Rhino3D) using the Grasshopper plugin for visualization and image processing to generate the node placement and planned PATH tile placement. The benefit of using these programs is for the ease of importing various file formats for site plan images or their CAD files, along with access to parametric adjustment of the values used for the algorithm. Sub-plugins of 'Manhattan', 'Clipper', 'Topologizer' and 'Pufferfish' components are also required within Grasshopper libraries for the algorithm to work. At this stage, the code would implement the blister and guidance PATH tile patterns upon the input site plans.

The initial block of Grasshopper code is used for image processing. It passes the extracted data to the next block for extracting the centroids of each discrete zone, which are then connected using Manhattan distance metrics. The output vertices are then tagged with the corresponding hazard PATH tiles, and the relevant distance PATH tiles are added afterwards, to fit with local architectural regulations.

Each component of the Grasshopper code represents a function applied on the preceding inputs. The entire process is split into 4 main blocks of code. The various stages of the algorithm are documented in Fig. 9 while examples of the executed code is shown in Section III.

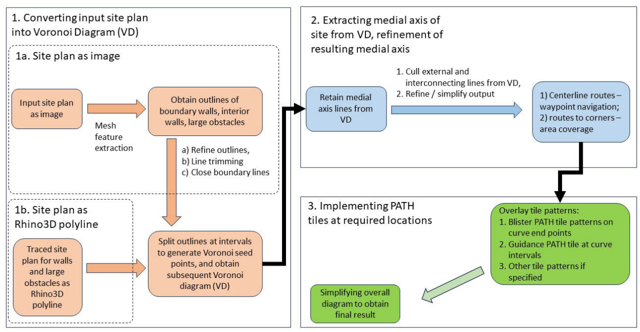

FIGURE 9. Summary of PATH implementation method on input floorplans.

Two methods are used for extracting the wall boundaries, and hazard outlines in the algorithm: 1) extracting from image, or 2) extracting from a traced input polyline in the Rhino3D software. The first method is reliant on the image quality, whereas the second method requires additional time to trace out the outlines. However, using the second method would reduce post-processing work required on the extracted site boundaries and the resulting VD centerlines.

The first method for extracting wall and spatial hazard boundary data takes in a monochrome site plan and extracts the spatial features of the walls and interior furniture of the site (Step 1a of Fig. 9, the feature extraction stage using monochrome image). The walls and main obstacles are extracted from the input site plan by using a monochrome colour filter, with walls and obstacles being coloured in black, and area permitted for robotic access in white. The data noise is filtered out using a bounded-size filter to remove

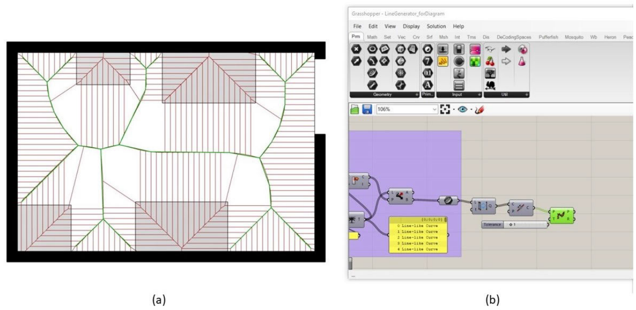

FIGURE 10. a) Obtaining center lines of accessible zones after post-processing VD lines from sample site plan (generated lines coloured in red, while remaining center lines are coloured in green); b) Diagram of Grasshopper console and partial code.

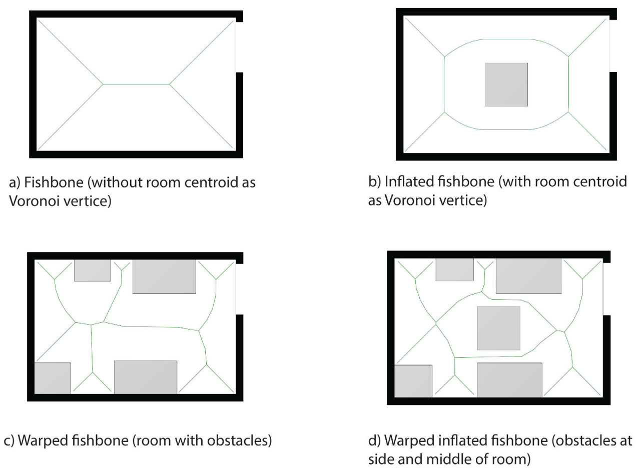

FIGURE 11. Types of MA generated: a) fishbone; b) inflated fishbone (if central node included); c) warped fishbone (obstacles at side of room); d) warped inflated fishbone (if obstacles are present at side and middle of room); generated paths in green.

unwanted lines or small spots that would arise from error signals of the object placements. The eventual lines and borders are then joined to form boundary curves for further processing.

The second method (Step 1b of Fig. 9, feature extraction stage using traced polyline) extracts the room boundaries using the boundary curve output after the site plan was processed, along with obtaining the footprint of any obstacles in the robot's path. A Rhino polyline traced from the site plan and the interior furniture footprints could also be used as input as well, which would then remove the step of boundary line processing filtering step. Outlines for the structural pillars that protrude from the walls may be omitted in this stage to minimize additional artifacts created during the VD centerline / MA generation stage.

The subsequent block of code processed the accessible area to generate two outputs from the site plan, 1) room centroids and 2) MA of the site. This would generate the path with least obstacles for the robot to traverse/ cover. The boundary curve of the accessible area would then be split for Voronoi cell generation to obtain the the center line of the traversable space, as seen in Fig. 10. The area considered for Voronoi cell generation is also an analogue for the bounded isovists that is clear of local obstacles. The room centroids are determined by subdividing the accessible zone by the doorways or corridor entrances/exits, and obtaining the centroid of the subdivided area. This points would then serve as the nodes for the graph of the site plan, as well as the center point of the Voronoi cell generation in each room to obtain a loop route for increased area coverage within the room, as compared to a linear route.

Algorithm 1 Summary of Site Graphing Process

Require: Site plan
Extract interior boundaries of walls of site plan
Extract boundaries of objects within site
for $i = 1 : No\_of\_boundaries$ **do**
  Segmenting *boundary*[*i*] for Voronoi Diagram (VD) generation
  Create VD cell using segment endpoints as the center
  if Line intersects/contacts with *boundary*[*i*] OR self-intersect **then**
    Remove line
  else if Line is aligned to main spline **then**
    Join at ends or generated segment intersections to form continuous trimmed line
  end if
  MA is considered trimmed if no fragments / small lines split off or overlap on main MA without corresponding topological branch
  Join remaining trimmed lines to obtain approximate MA
end for
Ensure: Valid and trimmed MA is generated
for MA[i] **do**
  Divide MA into segments of 30cm between them (size of PATH tile dimension)
  Obtain endpoints of segments and overall trimmed MA line
  if Point = endpoint **then**
    Overlay blister PATH tile pattern
    for blister nodes intersecting / outside of boundary[i] **do**
      Remove blister node
    end for
  else if Point located along line length **then**
    Align guidance PATH tile pattern along line direction
    Overlay guidance PATH tile pattern centered on points
    if guidance pattern overlap/intersect **then**
      Concatenate overlapping portions of guidance pattern to form closed linear bumps
    end if
  end if
end for

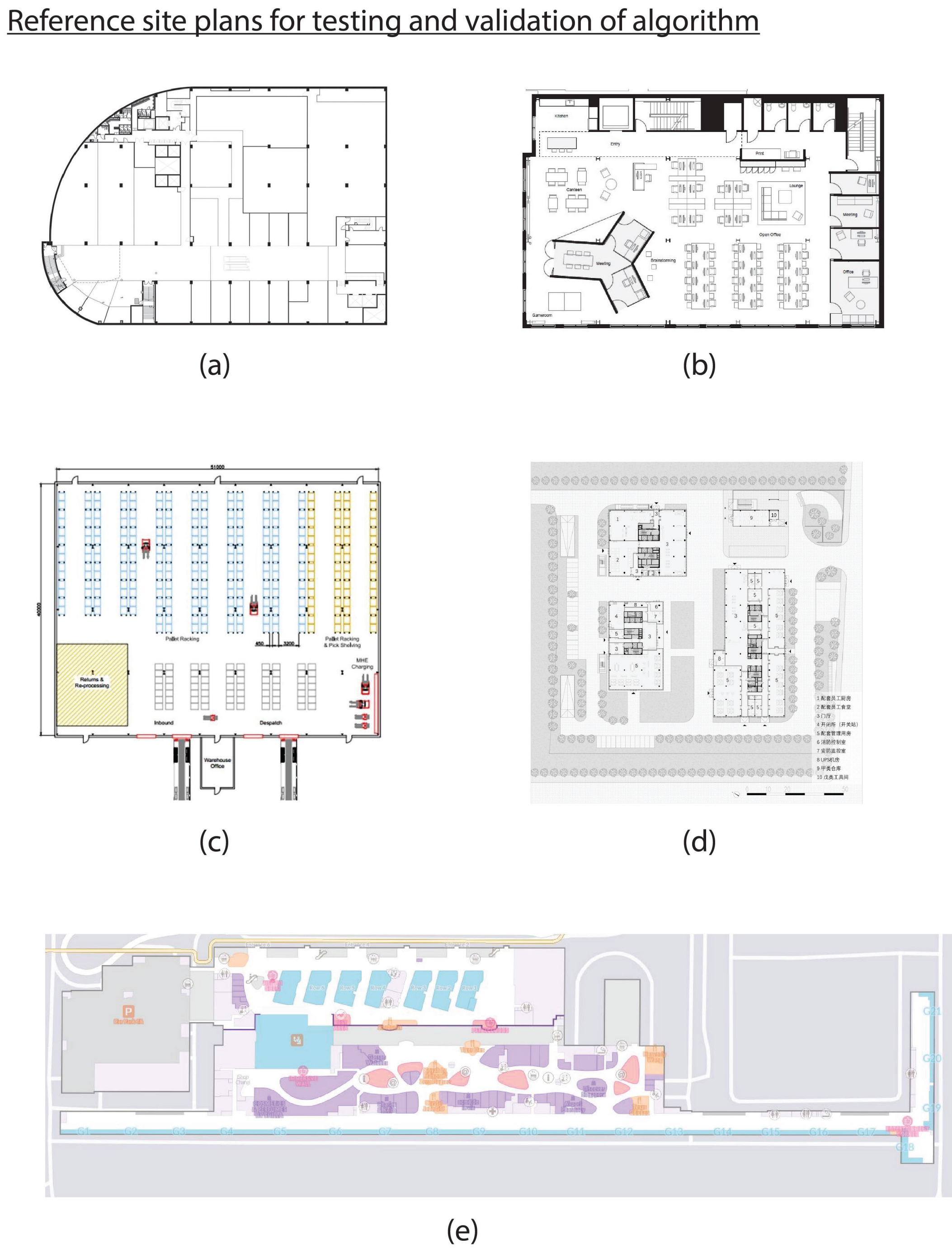

FIGURE 12. Reference sites used for validation - Site (a): Retail zone - the basement level of GRID shopping mall located in Singapore [75], Site (b): office - FINE Design Group [76], Site (c) - Industrial warehouse design [77], Site (d): Mixed use building - Smart New World Innovation Center, a mixed use building [78], and Site (e): Airport - Level 2 of Singapore's Changi Airport Terminal 4 [79], images not to scale to each other.

The resulting MA lines would look like the fishbone diagram lines, made by the centerlines of the open space (which can be utilized for waypoint travelling tasks), and lines that lead to the room corners (which would aid in area coverage tasks). Comparisons between the routes can be seen in Fig. 11, made adjustable by the choice of including / excluding the room centroid point in the Voronoi cell generation step, as well as whether obstacles are present within the room. An empty room generates a MA diagram akin to a line diagram of a fish's tail, hence a terminology of 'fishbone' is used. Variations of such 'fishbone diagrams' are created through the MA line generation process, the main variants are listed as:

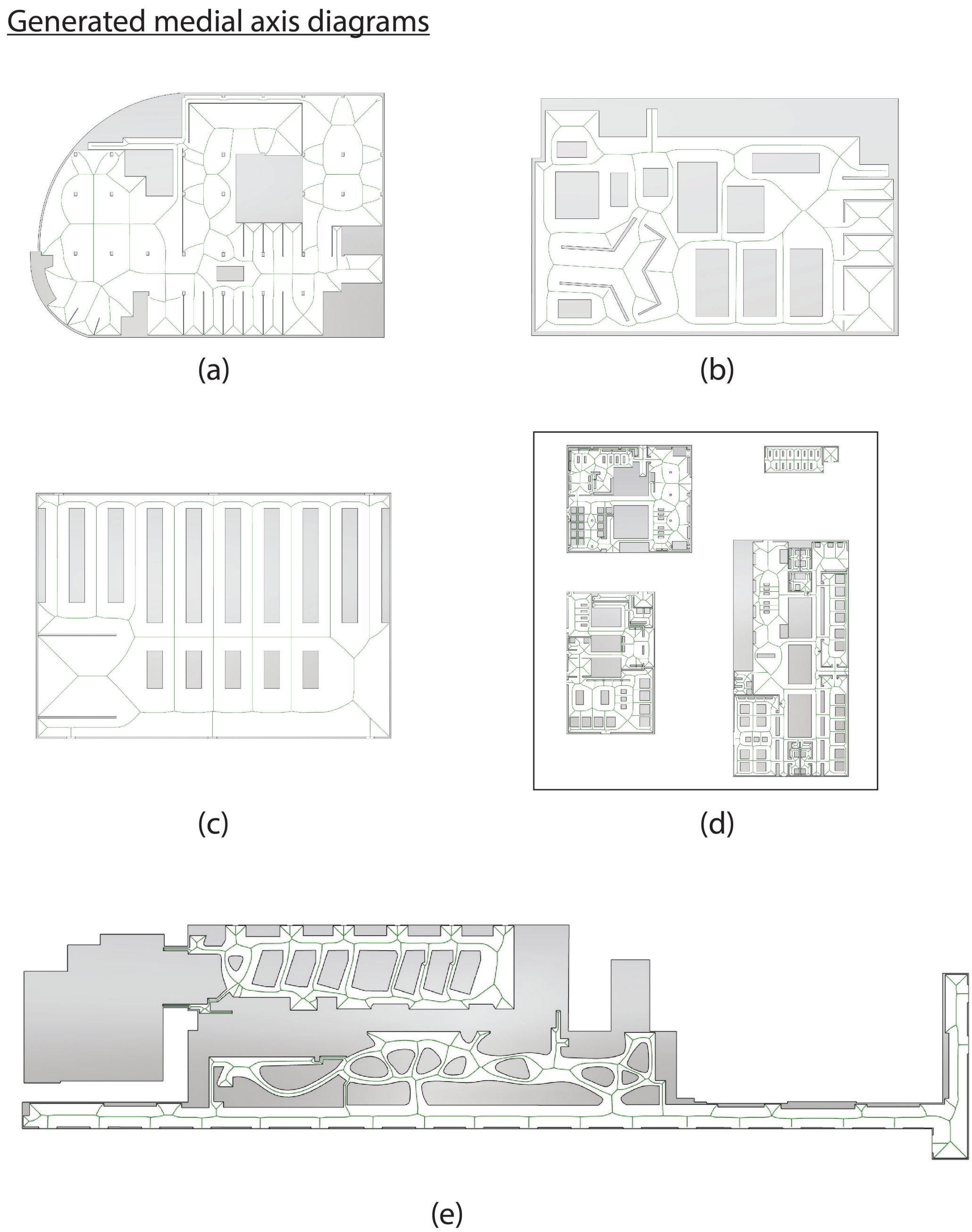

FIGURE 13. Medial Axis (MA) routes generated for sites of varying functions: a) retail; b) office; c) industrial, d) mixed use building, e) transport hub - airport; generated routes in green, images are not to scale relative to each other.

1) fishbone
2) inflated fishbone (if central node included / obstacle at room center)
3) warped fishbone (obstacles at side of room)
4) warped inflated fishbone (obstacles at side and middle of room)

The last block of the code implements the guidance and blister PATH tiles over the different points of the generated route, based on their location on the planned routes, with blister PATH tiles located at the endpoints of the generated route, and guidance PATH tiles at regular intervals along the lines of the generated routes. Overlapping tile patterns were culled or further processed to provide a cohesive tile layout within the chosen site plans.

In summary, the method of graphing a site plan in pseudocode form is given in Algorithm 1.

III. RESULTS AND DISCUSSION

A. RESULTS

5 site plans of existing buildings in various locations were input into the Grasshopper code for generating the

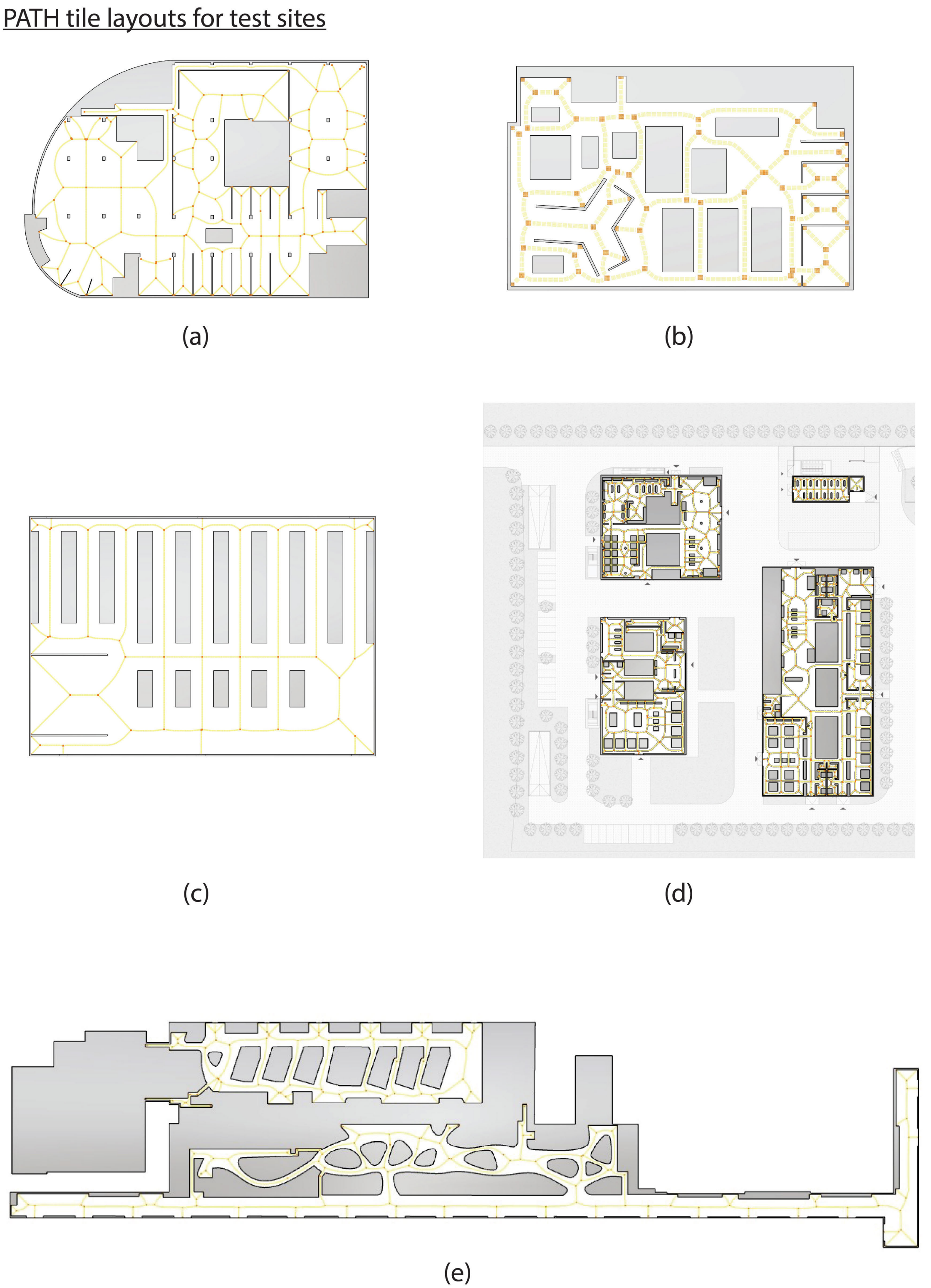

FIGURE 14. **Results - Final PATH routes generated for sites of varying functions: a) retail; b) office; c) industrial; d) mixed use; 5) transport hub; generated guidance tile layouts in light yellow, blister tiles in orange, images are not to scale relative to each other.**

'safe route' layouts. These sites were chosen for their varied functions, and to see how the tile layouts would be implemented on the different sites based on the building's footprint and interior layouts. Site 1 is a site plan depicting a retail zone - the basement level of GRID shopping mall located in Singapore [75], Site 2 represents a office site - FINE Design Group [76], Site 3 shows one for an industrial warehouse design [77], Site 4 shows the ground level floor plan from the Smart New World Innovation Center, a mixed use building [78], and Site 5 depicts a section of an airport, Level 2 of Singapore's Changi Airport Terminal 4 [79]. The referenced site plans are seen in Fig. 12. The generated MA routes are seen in Fig. 13, while their respective finalized PATH layouts are seen in Fig. 14.

The results show successful implementation of the algorithm on the 5 test sites of varying complexity, scale and function. PATH tile layouts were generated for all 5 sites as functional tile layouts i.e. the layouts do not intersect the existing walls or obstacles, and provide continuous paths from entrance points to other areas of interest within the buildings which the mobile service robots can now use.

B. DISCUSSION

The algorithm worked on generating the PATH tile layouts for all 5 existing sites. However, their individual codes required customized fine-tuning due to stray lines generated in the VD creation stage, or boundary issues that resulted in tiles intersecting the wall lines. These were solved by culling the tile curves that were located in the invalid areas. Future works would include applying the remaining PATH tile patterns on their corresponding hazards on the site plans, along with implementing PATH tile layouts on a test site in real life to test the efficacy of the PATH tiles for mobile robot deployment safety. As no precedents has been found for this application and use cases, the automating of the tile layout process would help architects and civil engineers to visualize and generate the routes for robots to avoid the spatial obstacles in their work environment, and provide further feedback on better installation of the robot-inclusive tactile marker placements.

IV. CONCLUSION

This paper has shown the automated process of determining the placement of robot-inclusive tactile paving for indoor environments as a means of alternate advance hazard detection means for enabling mobile robot deployments. Discussion on 'Design for Robot' framework and the need for an alternate way for robots to detect hazards within their environment using the robot-inclusive Passive Auto-Tactile Heuristic (PATH) tactile paving system was conducted. The process for optimizing the automated placement of these robot-inclusive tactile pavings was elaborated, with experiments conducted on 5 different existing site plans to obtain customized PATH tile layouts for the zone.

REFERENCES

[1] M. Shukla and A. N. Shukla, "Growth of robotics industry early in 21st century," *Int. J. Comput. Eng. Res.*, vol. 2, no. 5, pp. 1554–1558, 2012.

[2] M. O. Qureshi and R. S. Syed, "The impact of robotics on employment and motivation of employees in the service sector, with special reference to health care," *Saf. Health Work*, vol. 5, no. 4, pp. 198–202, Dec. 2014.

[3] D. R. Poduval and P. Rajalakshmy, "A review paper on autonomous mobile robots," in *Proc. AIP Conf.*, 2022, Art. no. 030005.

[4] J. Carlson, R. R. Murphy, and A. Nelson, "Follow-up analysis of mobile robot failures," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2004, pp. 4987–4994.

[5] B. S. Dhillon, *Robot Reliability and Safety*. Berlin, Germany: Springer, 2012.

[6] A. Fallatah, J. Urann, and H. Knight, "The robot show must go on: Effective responses to robot failures," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 325–332.

[7] J. Dargahi and S. Najarian, "Advances in tactile sensors design/manufacturing and its impact on robotics applications—A review," *Ind. Robot, Int. J.*, vol. 32, no. 3, pp. 268–281, Jun. 2005.

[8] M. H. Lee, "Tactile sensing: New directions, new challenges," *Int. J. Robot. Res.*, vol. 19, no. 7, pp. 636–643, Jul. 2000.

[9] R. S. Dahiya and M. Valle, "Tactile sensing for robotic applications," in *Sensors, Focus on Tactile, Force and Stress Sensor*. Vienna, Austria: InTech, 2008, pp. 298–304.

[10] M. A. Williams, B. Dubin, C. Amaefule, L. Nguyen, A. Abdolrahmani, C. Galbraith, A. Hurst, and S. Kane, "Better supporting blind pedestrians and blind navigation technologies through accessible architecture," in *Designing Around People*. Berlin, Germany: Springer, 2016, pp. 237–246.

[11] J. Parkin and N. Smithies, "Accounting for the needs of blind and visually impaired people in public realm design," *J. Urban Des.*, vol. 17, no. 1, pp. 135–149, Feb. 2012.

[12] O. Younis, W. Al-Nuaimy, M. H., and F. Rowe, "A hazard detection and tracking system for people with peripheral vision loss using smart glasses and augmented reality," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 2, pp. 1–9, 2019.

[13] B. W. Knol, C. Roozendaal, L. van den Bogaard, and J. Bouw, "The suitability of dogs as guide dogs for the blind: Criteria and testing procedures," *Veterinary Quart.*, vol. 10, no. 3, pp. 198–204, Jul. 1988.

[14] C. K. Lakde and P. S. Prasad, "Navigation system for visually impaired people," in *Proc. Int. Conf. Comput. Power, Energy, Inf. Commun. (ICCPEIC)*, Apr. 2015, pp. 0093–0098.

[15] J. Lloyd, C. Budge, S. La Grow, and K. Stafford, "An investigation of the complexities of successful and unsuccessful guide dog matching and partnerships," *Frontiers Veterinary Sci.*, vol. 3, p. 114, Dec. 2016.

[16] A. Rodríguez, J. J. Yebes, P. Alcantarilla, L. Bergasa, J. Almazán, and A. Cela, "Assisting the visually impaired: Obstacle detection and warning system by acoustic feedback," *Sensors*, vol. 12, no. 12, pp. 17476–17496, Dec. 2012.

[17] R. Wall Emerson, D. S. Kim, K. Naghshineh, J. Pliskow, and K. Myers, "Detection of quiet vehicles by blind pedestrians," *J. Transp. Eng.*, vol. 139, no. 1, pp. 50–56, Jan. 2013.

[18] J. Yang and H. Peng, "Decision support to the application of intelligent building technologies," *Renew. Energy*, vol. 22, nos. 1–3, pp. 67–77, Jan. 2001.

[19] J. Herssens, "Designing architecture for more: A framework of haptic design parameters with the experience of people born blind," Hasselt Univ., Dept. Archit., Hasselt, Belgium, Tech. Rep., 2011.

[20] J. Gual, M. P. Cazorla, and J. Lloveras, "Universal design and visual impairment: Tactile products for heritage access," *Des. Soc.*, vol. 5, no. ICED 11, pp. 155–164, 2011.

[21] K. W. M. Siu, "Public toilets for visually impaired persons: Application of the principles of universal design," *Int. J. Interdiscipl. Social Sci., Annu. Rev.*, vol. 6, no. 2, pp. 117–126, 2011.

[22] R. Sakaguchi, S. Takasu, and T. Akiyama, "Study concerning the colours of tactile blocks for the visually handicapped-visibility for the visually handicapped and scenic congruence for those with ordinary sight and vision," in *Proc. JIPEA World Congr.*, 2000, pp. 453–462.

[23] ISO, *Assistive Products for Blind and Vision-Impaired Persons*, Mar. 2012.

[24] *2008 Tactile Paving Surface Indicators Produced From Concrete, Clay and Stone*, document Cen/TS 15209, Dec. 2013.

[25] Development & Building Control Division, *Tactile Poster V0.6*, Land Transp. Authority, Singapore, 2015.

[26] S. K. Song, "Standard details for tactile tiles on footpath ramps," Develop. Building Control, Singapore, Sep. 2006.

[27] Standards Australia, Standards New Zealand, *Australian/New Zealand Standard*, Australia, Nov. 2002.

[28] *BSI*, Bs 7997, Brit. Stands Inst., U.K., May 2003.

[29] R. W. Emerson, "Tactile walking surface indicators in the United States and internationally," Nat. Inst. Disability, Independ. Living Rehabil. Res. (NIDILRR), USA, Tech. Rep., 2021.

[30] S. Yamauch, "Tactile ground surface indicators for blind persons," Nat. Rehabil. Center Disabled Japan, Japan, Tech. Rep. 13, Dec. 2003.

[31] J. Lu, K. W. M. Siu, and P. Xu, "A comparative study of tactile paving design standards in different countries," in *Proc. 9th Int. Conf. Comput.-Aided Ind. Design Conceptual Design*, Nov. 2008, pp. 753–758.

[32] L. R. Øvstedal, T. Lindland, and I. M. Lid, "On our way establishing national guidelines on tactile surface indicators," in *Proc. Int. Congr. Ser.*, vol. 1282. Amsterdam, The Netherlands: Elsevier, 2005, pp. 1046–1050.

[33] T. Mizuno, A. Nishidate, K. Tokuda, and K. Arai, "Installation errors and corrections in tactile ground surface indicators in Europe, America, Oceania and Asia," *IATSS Res.*, vol. 32, no. 2, pp. 68–80, 2008.

[34] *Transport Department of the Environment and the Regions*, Dept. Environ., Transp. Regions, U.K., Apr. 2000.

[35] M. A. V. J. Muthugala, M. Vega-Heredia, R. E. Mohan, and S. R. Vishaal, "Design and control of a wall cleaning robot with adhesion-awareness," *Symmetry*, vol. 12, no. 1, p. 122, Jan. 2020.

[36] N. A. B. H. Yahya, N. Ashrafi, and A. H. Humod, "Development and adaptability of in-pipe inspection robots," *IOSR J. Mech. Civil Eng.*, vol. 11, no. 4, pp. 1–8, 2014.

[37] S. M. B. P. Samarakoon, M. A. V. J. Muthugala, M. R. Elara, and S. Kumaran, "Toward pleomorphic reconfigurable robots for optimum coverage," *Complexity*, vol. 2021, pp. 1–10, Jan. 2021.

[38] J. Wang, S. Elfwing, and E. Uchibe, "Modular deep reinforcement learning from reward and punishment for robot navigation," *Neural Netw.*, vol. 135, pp. 115–126, Mar. 2021.

[39] S. X. Yang and C. Luo, "A neural network approach to complete coverage path planning," *IEEE Trans. Syst., Man Cybern., B*, vol. 34, no. 1, pp. 718–724, Feb. 2004.

[40] Betsy Case, *Universal design*, Pearson Education, San Antonio, TX, USA, 2008.

[41] M. R. Elara, N. Rojas, and A. Chua, "Design principles for robot inclusive spaces: A case study with roomba," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 5593–5599.

[42] P. R. Naraharisetti, M. A. Saliba, and S. G. Fabri, "Towards the quantification of robot-inclusiveness of a space and the implications on robot complexity," in *Proc. 8th Int. Conf. Autom., Robot. Appl. (ICARA)*, Feb. 2022, pp. 39–43.

[43] S. H. Ivanov and C. Webster, "Designing robot-friendly hospitality facilities," in *Proc. Sci. Conf. 'Tourism. Innov. Strategies*, 2017, pp. 13–14.

[44] M. S. Yeo, S. B. P. Samarakoon, Q. B. Ng, M. V. J. Muthugala, and M. R. Elara, "Design of robot-inclusive vertical green landscape," *Buildings*, vol. 11, no. 5, p. 203, May 2021.

[45] M. Yeo, S. Samarakoon, Q. Ng, Y. Ng, M. Muthugala, M. Elara, and R. Yeong, "Robot-inclusive false ceiling design guidelines," *Buildings*, vol. 11, no. 12, p. 600, Dec. 2021.

[46] M. S. Kiat Yeo, A. Q. Boon Ng, T. Y. Jin Ng, S. Mudiyanselage, B. P. Samarakoon, M. A. V. J. Muthugala, R. E. Mohan, and D. T. Ng, "Robot-inclusive guidelines for drain inspection," in *Proc. 8th Int. Conf. Inf. Technol., Comput. Electr. Eng. (ICITACEE)*, Sep. 2021, pp. 7–12.

[47] R. E. Mohan, N. Rojas, S. Seah, and R. Sosa, "Design principles for robot inclusive spaces," in *Proc. 19th Int. Conf. Eng. Design (ICED)*, 2013, pp. 1–8.

[48] E. Lewandowicz and P. Flisek, "A method for generating the centerline of an elongated polygon on the example of a watercourse," *ISPRS Int. J. Geo-Inf.*, vol. 9, no. 5, p. 304, May 2020.

[49] D. T. Lee, "Medial axis transformation of a planar shape," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-4, no. 4, pp. 363–369, Jul. 1982.

[50] R. Tam and W. Heidrich, "Shape simplification based on the medial axis transform," in *Proc. IEEE Visualizat.*, Oct. 2003, pp. 481–488.

[51] M. Tanase and R. C. Veltkamp, "A straight skeleton approximating the medial axis," in *Proc. Eur. Symp. Algorithms*. Cham, Switzerland: Springer, 2004, pp. 809–821.

[52] J.-H. Haunert and M. Sester, "Area collapse and road centerlines based on straight skeletons," *GeoInformatica*, vol. 12, no. 2, pp. 169–191, Jun. 2008.

[53] F. Aurenhammer and R. Klein, "Voronoi diagrams," in *Handbook of Computational Geometry*, vol. 5, no. 10. Amsterdam, The Netherlands: Elsevier, 2000, pp. 201–290.

[54] S. Fortune, "Voronoi diagrams and Delaunay triangulations," in *Computing in Euclidean Geometry*. Singapore: World Scientific, 1995, pp. 225–265.

[55] E. Lewandowicz, P. Lisowski, and P. Flisek, "A modified methodology for generating indoor navigation models," *ISPRS Int. J. Geo-Inf.*, vol. 8, no. 2, p. 60, Jan. 2019.

[56] F. Mortari, E. Clementini, S. Zlatanova, and L. Liu, "An indoor navigation model and its network extraction," *Appl. Geomatics*, vol. 11, no. 4, pp. 413–427, Dec. 2019.

[57] A. Ravankar, A. Ravankar, Y. Kobayashi, Y. Hoshino, and C.-C. Peng, "Path smoothing techniques in robot navigation: State-of-the-art, current and future challenges," *Sensors*, vol. 18, no. 9, p. 3170, Sep. 2018.

[58] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Hoboken, NJ, USA: Wiley, 2009.

[59] S.-K. Huang, W.-J. Wang, and C.-H. Sun, "A path planning strategy for multi-robot moving with path-priority order based on a generalized Voronoi diagram," *Appl. Sci.*, vol. 11, no. 20, p. 9650, Oct. 2021.

[60] Y. Qi, R. Wang, B. He, F. Lu, and Y. Xu, "Compact and efficient topological mapping for large-scale environment with pruned Voronoi diagram," *Drones*, vol. 6, no. 7, p. 183, 2022.

[61] Z. M. Al-Zubaidi, A. Y. Serdar, and A. K. Mohanned, "A comparative study of various path planning algorithms for pick-and-place robots," *Chin. J. Mech. Eng.*, pp. 1–27, May 2023.

[62] H. Qin, S. Shao, T. Wang, X. Yu, Y. Jiang, and Z. Cao, "Review of autonomous path planning algorithms for mobile robots," *Drones*, vol. 7, no. 3, p. 211, Mar. 2023.

[63] B. K. Jogeshwar and K. Lochan, "Algorithms for path planning on mobile robots," *IFAC-PapersOnLine*, vol. 55, no. 1, pp. 94–100, 2022.

[64] M. S. Yeo, J. J. Pey, and M. R. Elara, "Passive auto-tactile heuristic (PATH) tiles: Novel robot-inclusive tactile paving hazard alert system," *Buildings*, vol. 13, no. 10, p. 2504, 2023.

[65] N. Tan, R. E. Mohan, and A. Watanabe, "Toward a framework for robot-inclusive environments," *Autom. Construct.*, vol. 69, pp. 68–78, Sep. 2016.

[66] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," 2016, *arXiv:1606.09375*.

[67] B. Yan, G. Wang, J. Yu, X. Jin, and H. Zhang, "Spatial-temporal Chebyshev graph neural network for traffic flow prediction in IoT-based ITS," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9266–9279, Jun. 2022.

[68] J. M. Wiener, G. Franz, N. Rossmanith, A. Reichelt, H. A. Mallot, and H. H. Bülthoff, "Isovist analysis captures properties of space relevant for locomotion and experience," *Perception*, vol. 36, no. 7, pp. 1066–1083, Jul. 2007.

[69] M. Dawes and M. J. Ostwald, "Precise locations in space: An alternative approach to space syntax analysis using intersection points," *Archit. Res.*, vol. 3, no. 1, pp. 1–11, 2013.

[70] N. Napong, "The graph geometry for architectural planning," *J. Asian Archit. Building Eng.*, vol. 3, no. 1, pp. 157–164, May 2004.

[71] M. J. Ostwald, M. J. Dawes, M. J. Ostwald, and M. J. Dawes, "Isovist analysis, theories and methods," in *The Mathematics of the Modernist Villa: Architectural Analysis Using Space Syntax and Isovists*. Springer, 2018, pp. 95–124.

[72] M. J. Ostwald and M. Dawes, "Using isovists to analyse architecture: Methodological considerations and new approaches," *Int. J. Constructed Environ.*, vol. 3, no. 1, pp. 85–106, 2013.

[73] L. S. Davis and M. L. Benedikt, "Computational models of space: Isovists and isovist fields," *Comput. Graph. Image Process.*, vol. 11, no. 1, pp. 49–72, Sep. 1979.

[74] P. Osmond, "The convex space as the'atom'of urban analysis," *J. Space Syntax*, vol. 2, no. 1, pp. 97–114, 2011.

[75] S. Han, "Gallery of grid/spark-64," Archdaily, Santiago, Chile, Tech. Rep., May 2022.

[76] J. Alarc, "Gallery of fine/bora architects-12," Archdaily, Santiago, Chile, Tech. Rep., Apr. 2013.

[77] C. Pham, "Pin by Chuong Pham on warehouse: Warehouse design, warehouse layout, warehouse floor plan," Archdaily, Santiago, Chile, Tech. Rep., Sep. 2018.

[78] S. Han, "Gallery of smart new world innovation center/gad-38," Archdaily, Santiago, Chile, Tech. Rep., Jul. 2023.

[79] I. Tan, "Indoor map: Getting around Singapore Changi airport," Archdaily, Santiago, Chile, Tech. Rep., Feb. 2000.