Paper Title:

**Algorithms for Automated Generation of Navigation Models from
Building Information Models to Support Indoor Map-Matching**

Saurabh Taneja¹, Burcu Akinci^{*2}, James H. Garrett³, Jr. and Lucio Soibelman⁴

¹PhD, Data Scientist, SAP Labs LLC,

1251 Waterfront Pl, Suite 301

Pittsburgh, PA, 15222

Phone: 412-454-4163

Email: saurabh.taneja@sap.com

²M.ASCE, Professor, Department of Civil and Environmental Engineering,

123K Porter Hall, Carnegie Mellon University,

5000 Forbes Ave, Pittsburgh, PA, 15213

Phone: 412-268-2959

E-Mail: bakinci@andrew.cmu.edu

³P.E., F.ASCE, Dean of College of Engineering and Thomas Lord Professor of Civil and
Environmental Engineering,

110 Scaife Hall, Carnegie Mellon University,

5000 Forbes Ave, Pittsburgh, PA, 15213

Phone: 412-268-5090

E-Mail: garrett@cmu.edu

⁵ M.ASCE, Professor and Chair, Department of Civil and Environmental Engineering,

210A Kaprielian Hall, University of Southern California,

3620 S. Vermont Avenue, Los Angeles, CA 90089-2531

Phone: 213-740-0609

E-Mail: soibelman@usc.edu

* Corresponding author

Algorithms for Automated Generation of Navigation Models from Building Information Models to Support Indoor Map-Matching

Saurabh Taneja¹, Burcu Akinci², James H. Garrett³, Jr. and Lucio Soibelman⁴

Abstract

Navigation models are explicit representations of geometrical and topological information of physical environments that can be utilized for map-matching of indoor positioning data. This research paper presents algorithms for automated generation of three different types of navigation models, namely, centerline-based network, metric-based and grid-based navigation models, for map-matching of indoor positioning data. The abovementioned navigation models have been generated in an automated fashion from Industry Foundation Classes (IFC)-based building information models (BIM). Specifically, we have: 1) built on and targeted addressing limitations of existing algorithms that generate centerline-based network navigation models for polygonal shapes, 2) developed an approach to extract 2D geometry and topology from IFC-based BIM for creating metric-based navigation models, and 3) modified an existing algorithm to generate grid-based navigation models using geometry and topology extracted from BIM. The abovementioned three types of navigation models have been generated for six different testbeds with varying shape, size and density of spaces. We have validated the *generality* of the developed algorithms by evaluating the accuracy of geometrical and topological information contained within the three types of navigation models generated from testbeds with varying spatial characteristics.

Keywords: Map-matching, Indoor Positioning, Navigation Models, Industry Foundation Classes, Geometrical and Topological Representations

1. Introduction

Map-matching is the process of matching raw positioning data output from a positioning system to a map or a two dimensional (2D) navigation model depicting the surrounding physical environment [1,2]. Map-matching was first introduced to improve the overall position estimate from a Global Positioning System (GPS) by correcting erroneous raw data output by a GPS onto two dimensional (2D) road network navigation model stored in Geographic Information System (GIS) databases [2,3,4]. The existence of the road network navigation models in standard GIS vector format with nodes and edges assists greatly in map-matching of the GPS data [5,6,7]. Map-matching of indoor positioning data, on the other hand, is still not formalized as there is no standard representation of indoor environments for map-matching (akin to GIS-based representation for map-matching GPS data).

This paper addresses the issue of lack of approaches for automated generation of the navigation models identified in [8] for map-matching of indoor positioning data. Manual generation of navigation models for the purpose of indoor map-matching can be very expensive for large buildings, such as malls and hospitals. Moreover, buildings keep on changing and get renovated periodically over their lifecycle. Cotts [27] pointed out that on an average 30% of all building area is changed annually due to repairs and renovations. There is a lack of existing approaches that automate the generation of different types of navigation models required for map-matching of indoor positioning data even though there are several methods for automated generation of GIS-based road network navigation models [21,22,23,24,25], which are utilized for map-matching of GPS data.

Approaches for automated generation of indoor navigation models should utilize widely used building spatial information formats, such as .DWG [27] and .IFC [28]. For the research

described in this paper, we have selected Industry Foundation Classes (IFC)-based building information models (BIM), for automated generation of different types of navigation models required for indoor map-matching. The main reasoning behind such a selection is the fact that IFC provide standard for product modeling schema in the Architecture/Engineering/Construction domain [28].

Taneja [8] identified the requirements for map-matching of different types of indoor positioning data with varied navigation model representations and developed a domain-model to formalize the process of map-matching indoor positioning data. Requirements for map-matching identified in [8] include utilizing a) *network elements*, such as nodes and edges, that conform to the geometry and topology of the environment they represent; b) *boundary-representations* that contain connectivity information regarding the spaces they separate; and c) existence of *grids* that contain connectivity and adjacency information regarding neighboring grids. Altogether, Taneja [8] identified three broad categories of navigation models that are applicable for map-matching indoor-positioning data, namely i) network-based model, ii) metric-based model and iii) grid-based model. In this research work, we have developed algorithms for automated generation of the above-stated three broad categories of navigation models. The following paragraphs describe briefly the contribution of this research work towards developing automated generation approaches for the three categories of navigation models.

In this paper, we compare the strengths and weaknesses of existing computational geometry algorithms, such as the Medial Axis Transform [31], the Voronoi Graph [31,32], and the Straight Medial Axis Transform [30] (described in detail in Section 3.1), developed within the robotics and computational graphics domains for generating a centerline-based network navigation model similar to the GIS-based road navigation network. Due to the shortcomings of

current algorithms to generate a centerline-based network navigation model with linear elements, we developed a Modified Medial Axis Transform algorithm by modifying existing algorithms. The shortcomings of the Straight Medial Axis Transform algorithm [30] as well as the conceptual idea of the Modified Medial Axis Transform were presented in an earlier conference publication [50] and the current research work contains the results of the generated centerline-based network navigation model and validation of the Modified Medial Axis Transform.

To develop an approach for automated generation of metric-based navigation models, we have extracted *boundary-representations* of obstructions and portals existing in indoor environments from IFC-based BIM and added semantic information to each extracted boundary to determine whether the boundary is navigable (like a portal) or non-navigable (like an obstruction). Finally, for grid-based navigation models, we have modified existing algorithms, originally developed by Yuan and Schneider [29] and Li et al., [49], for extracting topologies from IFC-based BIM and automated the process of generating the *grid* occupancy map and movement probabilities required for map-matching indoor grid positioning data.

The algorithms developed for automated generation of navigation models have been verified and validated in a Java-based prototype developed with the goal of supporting the spatial queries contained within different types of map-matching algorithms. The algorithms developed in this research have been validated by generating the aforementioned three types of navigation models for six testbeds that differ in terms of i) total area of the testbed, ii) average area of the spaces in the testbed, iii) 75th percentile area of the spaces in the testbed (75th percentile area is the area of the space in the testbed that is larger than 75 percent of the spaces present in the testbed), iv) average room/space perimeter, v) average form factor (form factor is the ratio of the area of a space polygon to the area of the smallest circumscribing circle of that space polygon),

vi) node density (node density is the number of nodes per unit area in a network navigation model), and vii) average edge length in network navigation model.

The next section overviews the requirements for composition of navigation models to support the spatial queries required for map-matching of indoor positioning data. The subsequent sections describe the representations of geometrical and topological information contained in the IFC2x3 schema [28], background research on algorithms for transforming geometrical and topological information into navigation models, and the algorithms and approaches developed in this research paper for automated generation of a) centerline-based network, b) metric-based and c) grid-based navigation models from IFC-based BIM models. This paper concludes with the presentation of results, validation and discussions.

2. Requirements for composition of navigation models to support spatial queries required for indoor map-matching

Taneja [8] identified four major categories of spatial concepts; 1) free areas/spaces, 2) portals, 3) obstructions, and 4) paths; all of which are required to execute geometrical and topological queries contained within various map-matching algorithms. For example, a map-matching algorithm could query the free area/space within which the current raw indoor positioning data resides or a map-matching algorithm could query if the indoor positioning data crossed a physical obstruction. These spatial queries, which are incorporated within map-matching algorithms, are generated from physical laws of human navigation. An example of such a physical law is that humans are always contained within a certain room/space or that humans cannot pass through physical walls inside a building. To support the queries arising out of these physical laws of human navigation, existing navigation models abstract *spatial objects* contained in indoor environments, such as rooms/spaces, doors and walls, into corresponding *spatial*

concepts, such as free areas, portals and obstructions respectively. Thus one of the main requirements of map-matching is representing the above-stated spatial concepts within navigation models. For example, for GPS map-matching, roads can be represented as edges in a GIS-based network spatial model. Section 2.2 describes the different compositions of navigation models that have been utilized by different researchers to represent the above-stated spatial concepts and in turn support the execution of indoor map-matching queries. The next section presents the details of another important requirement for map-matching of indoor positioning data: the format of indoor positioning data.

2.1. Characteristics of algorithms for map-matching different types of indoor positioning data

Taneja [8] identified that different types of indoor positioning data formats, such as: i) absolute point-positioning data, ii) relative-point positioning data, and iii) grid-positioning data require different geometrical and topological queries for map-matching. Hightower and Borriello [14] classified indoor positioning data into two broad categories: 1) point positioning data, and 2) symbolic positioning data. Point positioning data is the positioning data described as a coordinate point in some frame of reference. Point positioning data can be further classified into i) absolute point positioning data, which has a global frame of reference, and ii) relative point positioning data, which has a local frame of reference [14]. Examples of absolute point positioning data include WLAN-based Ekahau system [15] that outputs data in the global coordinate system of a building and examples of relative point positioning data includes inertial sensor data, which reports only the displacement relative to the previous reported location. The second broad category of positioning data is called symbolic positioning data and it represents a position as a symbol, such as a room number or a zone number [14]. Symbolic data can also be classified into regular shaped grid data or irregular shaped room/zone data [14]. Proximity-based positioning

1 systems, such as ultrasonic-sensor-based systems [16] and bluetooth-based systems provide
2 positioning data as a room number or a zone number when a user is in proximity to a positioning
3 sensor. A number of positioning systems also output position data as grids of variable sizes [14].
4 Such data represents position as a grid number rather than a point (for example, grid number 64).

5 Map-matching of point positioning data can be performed by confirming data on to a
6 network navigation model represented by nodes and edges [11,12,17,18,19]. Various researchers
7 such as [1,2], have shown that a 2D GIS-based representation of the road network can map-
8 match erroneous positioning data from the GPS. A road network in a GIS database is composed
9 of *nodes* and *edges*. A node is a decision point and an intersection of two edges, and an edge is a
10 connection between two nodes. The explicitly stored geometry and the spatially indexed
11 topology of road networks in GIS databases assist in map-matching the GPS position data that is
12 output in NMEA 0183 format, i.e. latitude, longitude and error variances [2,19]. This is due to
13 the fact that raw positioning data represented as a point can be geometrically reasoned with
14 vector features, such as nodes and edges, to determine a) the distance of the raw data point from
15 an edge, or b) the orientation of raw data point with respect to the different edges in the road
16 network, and hence can be used for supporting spatial queries incorporated in map-matching
17 algorithms [17,18,19]. Similar to a 2D GIS representation of the road network, a network
18 navigation model of an indoor environment should contain nodes and edges that accurately
19 represent the length of navigation routes in indoor environments, as well as their topology, to
20 map-match raw positioning data from various indoor positioning systems. However, unlike road
21 networks, within which vehicles have to move only in one dimension along a roadway,
22 navigation routes in indoor environments can provide pedestrians several degrees of freedom to
23 move anywhere within a plane (Figure 1). Therefore, a network-based navigation model should

1 be constructed from a decomposition of planar geometrical representations of different spaces,
2 such as hallways and atriums into navigation routes (Figure 2).

3 Map-matching within buildings can also be performed by making sure that positioning
4 data stays confined within free areas/spaces and does not pass through obstructions represented
5 in a metric-based navigation model [10,13]. This type of map-matching reasons with raw
6 positioning with respect to the boundaries of free areas/spaces to determine: a) the room/space in
7 which the raw positioning data is confined, or b) if raw positioning data is crossing an
8 obstruction. Hence, a metric-based navigation model should contain a *boundary-representation*
9 of different spatial concepts (free areas/spaces, obstructions and portals) where boundaries have
10 semantics embedded in them to represent obstacles (walls) or portals (doors).

11 Map-matching of positioning data represented as grids requires analyzing the
12 probabilities of travelling from one grid to another grid so as to ensure that a user maintained the
13 geometrical and topological integrity of the environment [9,20]. Hence, a grid-based navigation
14 model should contain *grids* that represent navigable and non-navigable areas (Figure 3). In
15 addition to the representation of grids and their navigability, a grid-based navigation models
16 should also include connectivity using adjacency lists or adjacency matrix so that the probability
17 of movement from one grid to the other can be calculated. This is not as trivial as finding out the
18 neighboring grids of each grid and then determining which one of the neighbors is navigable and
19 which one is not navigable. Diagonal neighbors might not be topologically connected (depicted
20 by grids enclosed in the circle in Figure 3) even if both the diagonal grids are navigable. To
21 summarize, a navigation model should therefore either: a) contain vector features (either *nodes*
22 and *edges*, or *boundary-representations*) to map-match point positioning data, or b) have a *grid*
23 representation of obstructions and navigable areas for map-matching of grid positioning data.

Figure 1 Layout of a small office space generated from an IFC-based BIM model

Figure 2 Desired network navigation model of the small office space

Figure 3 Desired grid navigation model of the small office space

2.2. Composition of navigation models selected to evaluate indoor map-matching

In this research, we have focused on building on existing map-matching research in GIS-GPS domain as it has been widely adopted for vehicular map-matching. GIS domain contains both vector and raster GIS representations of road networks that can be utilized for map-matching of vehicular GPS data. Navigation routes in GIS road networks have been represented by centerlines in 2D vector representations containing nodes and edges. Similarly navigation routes in indoor environments, which are characterized by furnishing elements placed along the periphery of walkways, can also be represented by space-centerlines. As an early step towards evaluation of different types of navigation models for map-matching of indoor positioning data, in this research we have only focused on utilizing centerline-based network navigation models since these navigation models resemble closely to GIS representations of road networks that contain road centerlines. Even though there are a number of other types of existing network models that contain *nodes* and *edges*, such as metric-based topology networks, which are different than metric-based navigation models, [41] and visibility-based circulation networks [40], that have been utilized for way-finding, path-planning and code-checking in indoor environments [40, 41, 51, 52], these models are currently not in the scope of this research and will be studied in future research to evaluate their efficacy to map-match indoor positioning data.

Similarly, we have selected to build on top of existing GIS-based Shapefile representation that represents building footprints as *boundary-representations* on a 2D map. In our selected metric-based navigation model, similar to a GIS-based Shapefile representation, we have represented footprints of spaces existing in indoor environments through 2D boundary representations. We would like to clarify that the selected metric-based navigation model is different from metric-based topology network [41] mentioned in the previous paragraph. Metric-based topology networks [41], mentioned in the previous paragraph, are essentially network models composed of network elements- *nodes* and *edges*. These topology networks explicitly represent connectivity between edges and nodes. On the other hand, metric-based navigation models are composed of *boundary-representations* (or b-reps) where each boundary either represents a navigable object, such as a door, or a non-navigable object, such as a wall. These metric-based models represent connectivity implicitly, as opposed to topology networks, through meta-data stored for each boundary, such as navigable or non-navigable boundary. In this research paper, metric-based navigation models have been selected due to their similarity to GIS-based Shapefile representations that have been used for map-matching in GIS-GPS domain.

For the third navigation model, we have focused on a regular square grid representation for grid-based navigation model because 2D GIS raster representation of geographical data contains regular square-shaped pixels. There are other grid-based representations as well that contain regular triangular grids, regular hexagonal grids and irregular-shaped grids, but currently we have only focused on regular square-shaped grids as they resemble the GIS raster representation.

It is clear from the description in the last two paragraphs that we have currently focused only on 2D navigation models, which are based on 2D GIS representations used for map-

matching of vehicular GPS data. It is safe to assume that a pedestrian's feet are constrained to lie on the surface of a planar floor and we can map-match indoor positioning data generated by a pedestrian using a 2D navigation model representing the planar floor. We envision that same type of 2D navigation model can represent different planar floors and vertical connections between floors (stairs, elevators, escalators or ramps) can be represented by different navigation model elements, such as nodes and edges or boundary-representations. In our current research work, we have only focused on generating 2D navigation models and map-matching indoor positioning data on top of them; map-matching indoor positioning data on vertical connections between different floors will be evaluated in future research studies. The following section presents background research on algorithms for generating navigation models identified in our research for map-matching of indoor positioning data.

3. Background research on algorithms for generating navigation models

This section presents a brief overview of the algorithms developed by various researchers for generation of the three models selected in this research for evaluation: i) centerline-based network, ii) metric-based, and iii) grid-based navigation models.

3.1. Background research on algorithms for generating centerline-based network navigation models

Ukrainian/Russian mathematician Gregory Voronoy pioneered research in the field of generating network models in the late 19th century. He defined the well-known Voronoi diagram to represent planar polygons by their corresponding shape-skeletons. A shape-skeleton is the boundary of all the Voronoi regions inside a given planar polygon. A Voronoi region is an enclosed region formed by an element of a polygon, either an edge or a vertex, and the

corresponding bisectors. Figure 4 illustrates the various bisectors possible in a simple planar polygon. Figure 4a illustrates the creation of an angle bisector of two edges. Figure 4b illustrates the parabolic bisector of a reflex vertex and an edge. Since a parabola is a locus of all the points equidistant from a point and a line, a bisector of a reflex vertex and an edge will always be a parabola. Figure 4c depicts the case where the bisector of two reflex vertex elements of a simple planar polygon is the perpendicular bisector of these two vertices. In Figure 5, bisectors r_{12} , r_{23} and edge e_2 , and bisectors r_{45} , r_{51} and edge e_5 enclose two separate Voronoi regions. Each Voronoi region is closest to the element that encloses it, and due to this property the resulting shape-skeleton is equidistant from at least two elements of the polygon. This results in a centerline-based network model generated for any planar polygon. Lee [31] stated that the Medial Axis Transform (MAT), the algorithm for generating the Medial Axis, of a polygon is the same as the Voronoi Diagram of that polygon minus the bisectors that originate from reflex vertices. Like the Voronoi diagram, the medial axis of a polygon is the set of points internal to a polygon that are equidistant from and closest to at least two points on the boundary of the polygon.

Although there are various algorithms for generating the Voronoi diagram and the Medial Axis [31,32,33,34,35,36,37], the basics of all these algorithms are the same. Most of the algorithms involve starting the process of skeleton generation from the boundary, i.e., the vertices and the edges of the polygon and gradually heading in towards the center. Voronoi regions are first formed at the highest level and then secondary and tertiary regions are formed in the inner parts. For example, in Figure 5, bisectors r_{12} , r_{23} and edge e_2 , and bisectors r_{45} , r_{51} and edge e_5 enclose two separate Voronoi regions at the first level. Bisectors r_{12} , r_{13} , r_{41} , r_{51} and edge

1 e_1 enclose a Voronoi region at the second level. This process continues until each element of the
2 polygon, i.e. every reflex vertex and each edge has a Voronoi region associated with it.

4 **Figure 4. a) Angle bisector of two edges, b) Bisector of an edge and a reflex vertex, c) Perpendicular**
5 **bisector of two reflex vertices, d) Bisectors of a reflex vertex**

6 One of the drawbacks of the Medial Axis and the Voronoi diagram is the fact that these
7 representations include points, lines and parabolic curves. Representations containing complex
8 parabolic curves work do not work well for map-matching algorithms that utilize linear elements
9 as well as due to other limitations in various fields. Researchers in the domain of computational
10 geometry recognized these drawbacks and developed algorithms that would produce topology
11 networks that contain only linear elements. Aichholzer, et al. [38] developed the straight
12 skeleton representation that is used for calculating the polygonal roof over a layout of ground
13 walls. This representation consists only of straight elements and no parabolic arcs, and hence is
14 referred to as a straight skeleton. Lee [30] modified the algorithm developed by Yao and Rokne
15 [39] to create a 3D topology network containing only straight segments for providing geospatial
16 analysis capabilities for urban environments. Lee [30] named their algorithm the Straight Medial
17 Axis Transform (S-MAT) algorithm.

Figure 5 Network generated by the Straight Medial Axis Transform (S-MAT) algorithm developed by Lee [30]

The process of generating the S-MAT (developed by Lee [30]) of a polygon with only convex vertices is easier because it only requires generating angle bisectors of convex vertices (Figure 4a), and the resulting S-MAT, MAT and Voronoi diagrams are the same. On the other hand, if the planar polygon contains reflex or inflex vertices as well, then the Voronoi diagram constructs all the bisectors depicted in Figure 4a-d, MAT constructs all bisectors apart from the one depicted in Figure 4d, and S-MAT constructs only bisectors depicted in Figure 4a. Lee [30] defined two empirical rules for S-MAT creation in cases within which the polygon contains reflex vertices as well. The first rule is that a bisector of two edges, where one of the edge ends in a convex vertex and the other one ends in a reflex or inflex vertex, has the end point as the mid-point of the line segment joining the convex and reflex vertex [30]. This rule is illustrated in Figure 6a, where bisector r_{ij} is the bisector of vertex e_i and e_j , and e_i ends in a reflex or inflex vertex and e_j ends in a convex vertex. The second rule is that if two edges intersect in a common reflex vertex, then the two distinct bisectors of these edges intersect to form a valid node in the S-MAT diagram of the planar polygon. This rule is illustrated in Figure 6b.

Figure 6. a) Illustration of the first empirical rule of S-MAT algorithm, b) Illustration of the second empirical rule of S-MAT algorithm

Since the S-MAT algorithm involves constructing bisectors of only convex vertices, whenever there is an intersection of bisectors at a reflex or inflex vertex (such as intersection of r_{13} and r_{41}), a new bisector does not emerge from that intersection point. Unless there is another bisector that is heading from the opposite direction, such as bisector r_{67} in Figure 5, the algorithm gets stuck at that point. Figure 7 illustrates the fact that the S-MAT algorithm gets stuck after determining nodes n_1 and n_2 at reflex vertices. Similarly, as shown in Figure 6b the algorithm gets stuck at the intersection of bisectors at the common reflex vertex. This limitation arises due to the fact that S-MAT algorithm contains empirical rules, like the ones stated in the previous paragraph, and it becomes really difficult to generalize using empirical rules. Hence, in the cases of an I-shaped hallway the empirical rules formulated by Lee [30] do not work. In this research, we have built on top of S-MAT algorithm developed by Lee [30] and extended that algorithm to address the afore-mentioned limitations.

Figure 7 Limitation of S-MAT algorithm. The algorithm gets stuck after reaching nodes n_1 and n_2

As stated in the previous section, for this research we have only focused on generating

navigation models that resemble the GIS representations used for map-matching of GPS data [1,2]. Hence, we have only focused on centerline-based network models and have not looked at generating other types of network navigation models, such as the visibility-based circulation network developed by Lee et al. [40] (Figure 8b) and the metric-based topology network developed by Kannala [41] (Figure 8c). Figure 8 shows the difference between the three types of network-based models using a network connecting the two shaded rooms in the figure. A centerline-based network model runs along the medial axis of an indoor environment (Figure 8a), a metric-based topology network (depicted in Figure 8b) runs along the boundaries of navigable spaces in an indoor environment (but contains *nodes* and *edges* rather than *boundary-representations* unlike a metric-based navigation model), and a visibility-based circulation network (depicted in Figure 8c) runs along the visibility lines inside an environment (which need not be running along centerlines). Kannala [41] developed a metric-based topology network for fire-egress distance measurement, as illustrated in Figure 8b, but this network does not reflect actual navigation routes in indoor environments as the network runs along the boundaries of walls in an indoor environment. Lee et al. [40] used the IFC schema to develop a visibility-based circulation network (Figure 8c) for code-compliance checking on human circulation distances in a building. The developed circulation network includes all visibility lines within an indoor environment represented as *edges* and the end points of visibility lines are represented as *nodes*. There are other researchers who have generated visibility-based network models from BIM for path planning indoor environment [51, 52], but as stated earlier, in this research we have focused only on centerline-based network navigation models as an early step towards formalization of indoor map-matching [8].

Figure 8. a) Centerline-based network navigation model [30], b) Metric-based network navigation model [41], c) Visibility-based network navigation model [40]

3.2. Background research on algorithms for generating metric-based navigation models

Metric-based navigation models consist of explicitly represented boundaries of obstructions and portals and, as stated earlier, are different from metric-based topology networks. Object boundaries can be represented in a number of ways, such as parametric versus non-parametric models and global versus local models [47]. Parametric models describe a shape through the representation of model parameters, such as radius of a sphere and its center coordinate. Non-parametric models contain explicit shapes that combine together to describe an object, such as a triangle-mesh of the surface of an object [47]. Global models have a global frame of reference and local models have a frame of reference relative to the object being modeled [47]. Since parametric models describe a shape through a limited number of parameters, they are more compact than the non-parametric models. Examples include, linear shapes, such as lines and planes, as well as non-linear curves, such as ellipse, parabola, or hyperbola. While non-parametric models are computationally heavy to represent and store, they can represent any complex shape.

Researchers have utilized IFC-based BIM for extracting solid polygon geometry as well as surface geometry and boundary representations to generate 3D metric models for gaming, navigation assistance and code checking. Most of them [52, 53] have extracted detailed 3D

geometry and topology from BIM, whereas for map matching we require boundary representations of the 2D footprint of all the spaces contained within an environment (as described in Section 2). Along with the boundary representations, semantic information regarding whether these boundaries represent obstructions (walls) or portals (doors), is also required by the spatial queries contained within the map-matching algorithms originally developed in the GIS-GPS domain. For example, a map-matching algorithm could query if the raw positioning data crossed a physical obstruction, such as a wall, or went through a portal into a connected space/room. To address such queries, there is a need to determine the intersection of line segment formed by raw positioning data from two consecutive time points and the different boundaries in proximity of the raw positioning data. Hence, for realizing map-matching in an indoor environment there is a need to extract 2D footprint geometry along with topology and semantics about object boundaries contained in that environment.

3.3. Background research on algorithms for generating grid-based navigation models

Researchers have worked on generating two different types of grid-based navigation models primarily for route planning and shortest path determination [29,42]. The first category of grid-based models is generated from meshing techniques within which grids of different shapes and sizes compose a given space. This category of grid-models requires methods, such as triangulation of polygons representing spaces in physical environments, to determine the obstacles in the space and to construct the grid-mesh for a given space [43]. The generated mesh in this case usually contains grids of irregular shapes and sizes determined by the shape of the polygon for which triangulation is performed using the Delaunay Triangulation algorithm. The second major category of grid-based navigation models consists of models that contain uniform shape and size grids (usually rectangular or square) [42,44,45,46,49,54]. The advantage of this

category of grid-based navigation model is that they are easy to generate and store, but the disadvantage is that combining all the individual grids together might not generate the exact shape since the uniform size and shape grids merely approximate the exact geometry of the space/environment. One of the most common usage of uniform grid generation method is to determine whether each grid is occupied or empty by determining if a majority of the area of the grid overlaps with some obstacle in the given space or not [45]. This approach can be extrapolated to 3D by specifying whether each grid is surmountable or insurmountable by a human for navigation purposes [29].

Bandi and Thalman [45], Yuan and Schneider [29], Li et al. [49] and Song et al. [54], performed geometrical analysis on the generated grid-based navigation models to determine the topology of the physical environment for which grid-based navigation model is being generated to find out if a user can navigate from one grid to the other. When the grids are diagonal neighbors, the process of topology identification based on geometrical reasoning can introduce topological errors (depicted by grids enclosed in the circle in Figure 3) within which geometrical reasoning can classify the two grids to be connected although they are not connected in the physical environment. We have utilized IFC-based BIM to extract both geometrical and topological information to generate occupancy grid and movement probabilities in our grid-based navigation model for map-matching grid positioning data. Utilizing topological information from IFC-based BIM for determining connected grids eliminates the scenario where two diagonally opposite unoccupied grids (circled grids in Figure 3) are wrongly classified as topologically connected. Such an error might occur when geometrical data is used for deducing topological connectivity information, an approach employed in some of the previous research works [29, 45, 49, 54].

To summarize, the key limitations of the existing approaches include: a) the algorithm for generating centerline-based network navigation model (called the S-MAT algorithm) getting stuck in the case of an I-shaped hallway, b) diagonally opposite located, unoccupied but unconnected grids (circled grids in Figure 3) getting misclassified as connected grids when utilizing geometrical data for inferring topological connections in grid-based navigation models, and c) lack of topological information contained within metric-based navigation models generated only from geometrical data. The next section describes the algorithms that address the above-stated key limitations for generating the three types of navigation models that are developed in the research described in this paper.

4. Algorithms developed for generating navigation models from IFC-based building information models

This section presents the algorithms and approaches developed and modified in this research for automated generation of three afore-mentioned types of navigation models required for map-matching of indoor positioning data.

4.1. Transforming IFC-based building information models into navigation models

Automated generation of navigation models from IFC-based BIM requires extraction of geometrical and topological information related to four types of spatial objects/concepts, namely 1) free areas/spaces, 2) portals/doors, 3) obstructions/walls, and 4) paths [8]. We have used an IFC-to-Java parser created by OpenIFCTools [48] for parsing the IFC objects represented in the EXPRESS schema into Java objects and developed our own Java-based prototype to read the parsed IFC objects. Once the IFC objects are available in Java, all the instances of

1 *IFCRelContainedInSpatialStructure* and *IFCRelAggregates* classes can be reasoned with to
2 determine the number of floors in a building and all the spaces contained in each floor. This step
3 helps to create the navigation model for all the free areas/spaces contained in each floor of a
4 building in our Java-based prototype.

5 We have extracted the geometry of the required spatial concepts represented in an IFC-
6 based BIM, particularly the geometry of a) *IFCSpace* (free areas), b) *IFCDoor* (portals), and c)
7 *IFCWallStandardCase* (obstructions). Every building element or spatial entity described in IFC-
8 based BIM contains an *IfcLocalPlacement* and an *IfcProductDefinitionShape* attribute that
9 describe the placement of the coordinate system and the geometric representation of the building
10 element or spatial entity, respectively. Building elements, such as walls, doors, and columns, and
11 spatial entities, such as rooms, have a two-dimensional planar representation as well as a three-
12 dimensional solid representation. The two-dimensional planar representations include description
13 of the layout of the boundary of these spatial concepts through *IfcPolyline*, *IfcCompositeCurve*,
14 *IfcArbitraryClosedProfileDef*, *IfcArbitraryOpenProfileDef*, *IfcRectangleProfileDef* or other
15 representations of the boundaries.

16 Apart from the boundary geometry, navigation models for map-matching also require
17 topological data, such as connections between spaces and doors, and between spaces and other
18 spaces. Movement probabilities to connected grids also need to be generated and represented in
19 the grid-based navigation model. The IFC schema provides a number of relationship classes,
20 such as the *IfcRelSpaceBoundary* that relates a boundary with the corresponding space. An
21 *IfcRelSpaceBoundary* class consists of an instance of *RelatingSpace*, *RelatedBuildingElement*
22 and an *IfcConnectionSurfaceGeometry*. The values of the attributes *RelatingSpace* and
23 *RelatedBuildingElement* assist in identification of the specific building element that is

1 topologically related to a specific building space. The building elements that are topologically
2 related to two different spaces then provide a means of topological connection between the
3 concerned two spaces. For example, if a door with unique ID 418 is related to space number
4 PH121 and is also related space number PH123, then spaces PH121 and PH123 are topologically
5 connected to each other through the door with unique ID 418.

6 Figure 9 describes the hierarchy of the IFC objects that have been utilized in this
7 research to extract geometrical representation and spatial topologies required to generate the
8 aforementioned navigation models. The IFC entities depicted in the shadowed boxes are the ones
9 that are utilized for navigation model generation and the attributes, stated besides the dashed
10 lines, have been used to identify topological connections between building elements and
11 corresponding spaces. The logic for parsing IFC-based BIM into object instances in Java,
12 generating the aforementioned types of navigation models from the object instances created from
13 BIM and map-matching of raw indoor positioning data with generated navigation models have
14 been carried out in our Java-based prototype developed using Eclipse as the development
15 environment. All the spatial and topological queries regarding BIM that are required for map-
16 matching and validating the algorithms developed in this research have been incorporated into
17 the abovementioned Java-based prototype.

Figure 9 Hierarchy of selected IFC entities utilized in this research

4.2. Centerline-based network navigation model generation algorithm

As stated in Section 3.1, we have selected to build upon the Straight Medial Axis Transform (S-MAT) algorithm originally modified by Lee [30] from Yao and Rokne [39] for the generation of a centerline-based network navigation model. We presented the conceptual logic of the M-MAT algorithm along with the shortcomings of the S-MAT algorithm in an earlier conference publication [50] and this research presents the details of the developed algorithm as well as the results and validation in the next two sections. The Modified Medial Axis Transform (M-MAT) involves constructing all but one type of bisector of the elements of a planar polygon (the bisector depicted in Figure 4c). The M-MAT algorithm involves constructing the two perpendicular bisectors of reflex vertices to determine the nodes of the medial axis. Two perpendicular bisectors of a reflex vertex are shown in Figure 10b as dotted lines. The nodes, n_1 , n_2 and n_3 , of the medial axis are determined by intersecting these perpendicular bisectors with other angle bisectors. The original algorithm for generating Medial Axis (MAT) or Voronoi diagrams involves constructing the parabolic bisector, p_1 or p_2 , of a reflex vertex and determining

1 the nodes, n_1 , n_2 and n_3 , of the medial axis by intersecting the parabolic bisector with other
2 bisectors as shown in Figure 10a. A major difference between the MAT and the M-MAT
3 algorithms is the fact that in the MAT algorithm the parabolic bisector of a reflex vertex is a part
4 of the medial axis of the polygon (Figure 10a), whereas in the M-MAT algorithm the two
5 perpendicular bisectors of a reflex vertex (the dotted lines in Figure 10b) only assist in
6 determining the nodes of a medial axis. To complete the modified medial axis in the M-MAT
7 algorithm, line segments, l_1 and l_2 are drawn between nodes n_1 and n_3 , and n_2 and n_3 of the
8 medial axis, as shown in Figure 10b, that would originally contain a parabolic bisector.

Figure 10. a) Voronoi diagram of a polygon, b) Modified-medial axis of the same polygon

Figure 11 Modified-medial axis transform (M-MAT) algorithm for generating centerline-based network navigation models from IFC-based BIM models

Figure 11 presents the M-MAT algorithm modified in this research from Yao and Rokne [39] and Lee [5]. The shaded box in the flowchart shows our modifications, within which Yao and Rokne [39] construct a parabolic bisector for a reflex vertex (Figure 4b) and Lee [5] does not draw the bisector of a reflex vertex, but our algorithm constructs two perpendicular bisectors (Figure 4d). Figure 12 illustrates the construction of the modified-medial axis for an L-shaped hallway using the M-MAT algorithm. In Figure 12a, the first level angle bisectors, namely 1, 2, 3, and 4, are generated from all the convex vertices. These angle bisectors intersect to produce second level angle bisectors 5 and 6. In the meantime, the first level angle bisectors 7 and 8 of

the reflex vertex are also generated. Due to the asymmetry of the L-shaped hallway (the horizontal arm of the hallway is narrower than the vertical arm), bisector 7 intersects bisector 5 before bisector 8 can intersect bisector 6. Hence, we get a third level angle bisector, bisector 9. Again due to asymmetry, third level angle bisector, bisector 9, intersects with the first level angle bisector, bisector 8, before bisector 8 can intersect with the second level angle bisector, bisector 6. Therefore, we get a fourth level angle bisector, bisector 10. This fourth level angle bisector intersects with the second level angle bisector, bisector 6, to complete the network navigation model. This process of generating angle bisectors with the corresponding elements of the L-shaped hallway makes sure that a complete modified-medial axis is generated, although due to the incidence angle of the bisectors (45 degrees in this case because of bisecting two perpendicular lines), the shape of the modified-medial axis deviates from the medial axis or the voronoi diagram.

Figure 12 Elaboration of M-MAT generation for an L-shaped hallway

The M-MAT algorithm has the advantage of being generally applicable to any shape or layout of the indoor environment, whereas the S-MAT algorithm breaks if there is an I-shaped hallway, a commonly occurring architectural configuration, especially in office spaces, where cubicles are placed regularly around hallways. On the other hand, the resulting medial axis generated from the M-MAT algorithm presents certain challenges to map-matching. For instance, in Figure 10b, if a user has to walk straight through the hallway crossing nodes n_1 and

n_2 , then the node n_3 does not lie on the user's path, although the medial axis generated from the M-MAT algorithm will take the user through node n_3 , as there is no direct connection between nodes n_1 and n_2 . The straight medial axis resulting from the S-MAT algorithm, shown in Figure 5, does not suffer from this limitation. Figure 13 depicts a network navigation model generated from an IFC-based BIM of an academic building.

Figure 13 Centerline-based network navigation model generated from a BIM model of an academic building

4.3. Metric-based navigation model generation approach

The developed approach for generating metric-based navigation models from IFC-based BIM involves constructing boundary representation of required spatial concepts by reading all the instances of *IFCRelSpaceBoundary* class at each level of a building in a BIM. This class has three attributes, namely *RelatedBuildingElement*, *RelatingSpace* and *IfcConnectionSurfaceGeometry*. *IfcConnectionSurfaceGeometry* describes the surface of the boundary between a space and related building element, such as a wall, door, slab or window. The surface boundary is represented through *IfcCurveBoundedPlane*, *IfcSurfaceOrFaceSurface*

1 or other surface representations. Geometrical and topological information is extracted through
2 the process described in Section 4.1. Upon parsing every *IFCRelSpaceBoundary* contained in an
3 IFC-based BIM and determining whether the corresponding *RelatedBuildingElement* is a
4 *IFCWallStandardCase* or an *IFCDoor*, each *IFCRelSpaceBoundary* can be represented as a b-
5 rep with accurate geometry, topology (as described in Section 4.1) and semantics of whether it is
6 a non-navigable obstruction or a navigable portal respectively. Figure 14 presents the approach
7 developed in this paper for transforming IFC-based spatial information into metric-based
8 navigation models. Once the geometry and topology of all boundaries within a level have been
9 determined, then all the generated boundaries can be co-located to form the metric-based
10 navigation model as depicted in Figure 15. Figure 15 illustrates the metric-based navigation
11 model generated for the same academic building whose centerline-based network navigation
12 model was presented in Figure 13. The darker boundaries depict the obstructions that represent
13 the walls and the lighter boundaries depict the portals that represent doors as well as virtual zone
14 separations.

Figure 14 Approach for generating metric-based navigation model from IFC-based BIM models

Figure 15 Metric-based navigation model generated from the BIM model of an academic building with black boundaries representing obstructions and grey boundaries portals

4.4. Grid-based navigation model generation approach

We have built on the algorithm developed by Yuan and Schneider [29] for generating the grid-based navigation model of a building from IFC-based BIM. Yuan and Schneider [29] developed a LEGO-based representation for 3-D route planning inside buildings for pedestrians and wheelchairs. Their representation consists of 3D LEGO blocks of uniform footprint, but varying height, depending upon the height of the space at the location of LEGO block. Each LEGO block has a property of obstacle or non-obstacle and surmountable or non-surmountable. Upon approximation of obstacles, Yuan and Schneider [29] group together the non-obstacle blocks to determine the width of the non-obstacle area so as to calculate whether a wheelchair can navigate through it or not. We have adopted their algorithm, but instead of grouping the non-obstacle blocks/grids for route planning, we generated an adjacency list with movement probabilities for every grid to its adjacent grids. This information is required by map-matching algorithms to calculate the likelihood of a positioning data moving from one grid to another in consecutive time points.

Figure 16 describes the approach that we have adopted for generation of grid-based navigation models. The approach involves determining the range of points that represent building elements and spatial entities in each floor of the input BIM model. Once the point range is determined for every floor, an empty grid map is created with uniform square grids of size 0.5m. Grid size of 0.5m is chosen to evaluate positioning accuracy at the granularity of distinguishing the position of a person from one room to the other or to determine the exact door a person passes through in the case of two adjacent doors. For every space in a floor, all the grids, whose center point lies within the space boundary, are assigned the attribute of non-obstacle or navigable. For every wall in a floor, if the thickness of the wall is less than grid size

1 of 0.5m then the algorithm assigns all the grids that contain the center line of the wall as an
2 obstacle or non-navigable. If the thickness of the wall is greater than 0.5m, then the algorithm
3 assigns all the grids whose center point lies within the wall polygon as an obstacle. For every
4 door on a floor, the algorithm assigns all the grids that contain the centerline of the door as non-
5 obstacle or navigable. Figure 17 depicts grid-based navigation model generated from the IFC-
6 based BIM of the same academic building whose centerline-based network navigation model is
7 depicted in Figure 13. The blacked-out grids in Figure 17 represent the non-navigable grids,
8 whereas the white grids represent navigable grids. As stated earlier, it is not necessary that
9 navigable grids, which are diagonal neighbors, be topologically connected to each other.
10 Topological connectivity is determined by the two attributes of *IFCRelSpaceBoundary* class,
11 namely *RelatingSpace* and *RelatedBuildingElement* as described in Section 4.1.

Figure 16 Algorithm for generating grid-based navigation model from IFC-based BIM models

1

2

3

4

Figure 17 Grid-based navigation model generated from an IFC-based BIM model of an academic building

5

6

7

8

9

10

11

12

As is clear from Figure 13, Figure 15 and Figure 17, the centreline-based network model contains a representation of the paths and junctions inside a building, the metric-based model represents boundaries of free areas that can be penetrated (portals) and that cannot be penetrated (obstructions) and the grid-based model contains representations of navigable and non-navigable grids as well as a representation of topological connections between the grids, and. All these representations are used by different fusion algorithms for map-matching of indoor positioning data. The next section presents results on the qualitative comparison of navigation models generated by different algorithms for different shapes and layouts.

13

5. Results and Discussions

14

15

16

We generated three different types of navigation models for six different testbeds that differ in: a) size of spaces, b) density of spaces, and c) shape of spaces that they contain. One of the testbeds is located in an academic building, and Figure 13, Figure 15 and Figure 17, represent

the centerline-based network, the metric-based and the grid-based navigation model respectively of that testbed. The remaining five testbeds are located in a recreational center that contains spaces for multiple types of usages. Four of these testbeds are located in the lower level of the recreational center and the 5th testbed is located on the ground level. Figures 18 through 25 represent the centerline-based network and grid-based navigation models for the four testbeds located in the lower level of the recreational center. We selected these six testbeds to validate the generality of the developed and implemented navigation model generation approaches. All selected testbeds differ in the shape of the main hallway(s) and size and density of spaces surrounding the main hallway(s). The next section provides greater details on the similarities and differences in the spatial characteristics of the testbeds. The rest of this section presents discussion of impact of indoor layouts and shapes on the performance of algorithms developed in this research for automated generation of navigation models and the identification of cases where these algorithms perform poorly.

Figure 18 Network navigation model of Testbed_1 in the lower level of an activities and recreational center

Figure 19 Grid navigation model of Testbed_1 in the lower level of an activities and recreational center

Figure 20 Network navigation model of Testbed_2

Figure 21 Grid navigation model of Testbed_3

Figure 22 Network navigation model of Testbed_3

Figure 23 Grid navigation model of Testbed_4

Figure 24 Network navigation model of Testbed_5 in the upper level of an activities and recreational center

Figure 25 Grid navigation model of Testbed_5 in the upper level of an activities and recreational center

Figure 26 depicts the Modified Medial Axis Transform (M-MAT) and corresponding Medial Axis Transform of common hallway shapes. It is clear from Figure 26a and Figure 26c that although M-MAT algorithm is able to generate the medial axis of the common hallway shapes, the modified medial axis does not truly represent Voronoi regions in the hallway shapes (Figure 26b and Figure 26d). For example, in Figure 26a, near the bend in the L-shaped hallway, because a straight bisector replaces a parabolic bisector, the intersection of the bisectors does not lie on the Voronoi diagram or the middle of the hallway as it does in Figure 26b. A detailed discussion on why there is a curl in the modified medial axis around the bend of the L-shaped hallway (Figure 26a) was provided in Section 4.2. The modified medial axis of the I-shaped hallway, one of the limitations of the S-MAT algorithm (as stated earlier), closely follows the medial axis or the Voronoi diagram, as depicted in Figure 26c and Figure 26d.

Figure 26 Modified-medial axis of common hallway shapes depicted in a) and c) with corresponding medial axis depicted in b) and d)

M-MAT algorithm is generic for all shapes due to its adaptation from the universally accepted Voronoi diagram as well as the fact that it generates angle bisectors (which form the basis of Voronoi diagrams and medial axis) for all the vertices in a polygon (unlike [31]). Although M-MAT algorithm is generic like the Voronoi diagram, it has some drawbacks of generating non-smooth networks (Figure 26), especially for

asymmetric shapes that contain larger number of reflex vertices (Figure 26a and Figure 27c). Figure 27 depicts the Modified Medial Axis Transform and corresponding Medial Axis Transform of more complex shapes that contain larger number of reflex vertices. It is clear from, Figure 26c-d as well as Figure 27a-b that if the reflex vertices are symmetrically aligned opposite to each other, then the modified-medial axis and the medial axis will closely resemble each other. In cases where, the reflex vertices are not symmetrically aligned with other reflex vertices or where there are odd number of reflex vertices, then the approximation of a parabolic bisector by a straight bisector can cause some deformations in the medial axis. This fact is clear from Figure 27c-d as well as Figure 26a-b (a curl near the bend in the L-shape hallway). Hence, the prevalence and arrangement of reflex vertices in space polygons determines the deviation of the generated M-MAT from the Voronoi diagram or the medial axis.

Figure 27 Modified-medial axis of complex hallway shapes depicted in a) and c) with corresponding medial axis depicted in b) and d)

The approach developed for the generation of grid-based navigation models from IFC-based BIM performs well for modelling topology of the environment, but introduces some errors in classifying a small percentage of navigable area as occupied grids. Figure 28 illustrates both of these cases. In certain cases, an error in the geometry of the generated grid (depicted within the circle in Figure 28) does not result in a topological error because the two rooms are not topologically connected to each other in the BIM

(Figure 28b). In other cases, some of the walls depicted as grids in Figure 28a have more thickness than the actual walls due to errors introduced by keeping a uniform grid size of 0.5m for generating the grid-based navigation model. The latter case of classifying a small percentage of navigable area as occupied grids is common to all the approaches that utilize a uniform grid size, such as a square-shaped grid, because of the limitation of the level-of-detail that can be modelled with uniform grid size. Our approach performs well on the former case depicted in the circle in Figure 28a. This is not the case for the model generated by Bandi and Thalman [45] and Yuan and Schneider [29] since their algorithms construct a direct topological connection between two rooms due to the opening depicted by the circle in Figure 28a. Our algorithm for grid-based navigation model generation fails when extracting topology in the scenario illustrated in Figure 3 within which two diagonal grids, which have been classified as navigable, are topologically connected to each other although they are not directly physically connected. Hence, our algorithm misclassifies (validation results presented in next section) some percentage of navigable area as obstacle/non-navigable and vice-versa. There are also cases where physically non-connected grids get topologically connected in the grid-based navigation model (depicted in Figure 3).

The algorithm developed for generation of metric-based navigation models extracts boundary representations from parametric and non-parametric geometric representations within IFC-based BIM models, such as *IfcPolyline*, *IfcCompositeCurve*, *IfcRectangleProfileDef*, *IfcArbitraryClosedProfileDef* and *IfcArbitraryOpenProfileDef*. IFC schema provides details on these geometric representations, and the extraction of geometry and topology is straightforward as explained in Section 4.1. A problem that we

encountered in this research was the fact that in some cases there were missing instances of *IfcRelSpaceBoundary*, which limited our ability to generate the boundaries. Figure 28b depicts missing boundaries in a generated metric-based navigation model. Longer ovals depict missing obstructions and shorter oval depicts missing portal boundary. Despite of these missing boundaries, due to errors in export of IFC-based BIM from authoring software, the topological information within the generated metric model is accurate. This is because of the fact that missing topological information can be easily added in an IFC-based BIM by adding *IfcRelSpaceBoundary* class in the text-based .IFC file for missing boundaries. Hence, metric-based navigation models only suffer from the limitation of missing instances of *IfcRelSpaceBoundary* in IFC, which can be overcome by adding missing information. The following sections present the validation of the developed algorithms and the discussion on performance of the developed algorithms in different testbeds.

Figure 28 a) Topologically disconnected grids (contained in the circle), b) missing obstruction boundaries (represented by long ovals) and missing portal boundary (represented by smaller oval).

6. Validation

Section 2 presented the requirements for composition of navigation models needed for

map-matching of indoor positioning data originally identified by [8]. Taneja [8] also validated that centerline-based network navigation model, metric-based navigation model and grid-based navigation model satisfy the requirements of spatial queries contained in most of the indoor map-matching algorithms. In this section, we discuss the results of our attempts to validate the *generality* of the developed algorithms for generating the three types of navigation models for indoor environments containing varied spatial features.

We characterize the generality of the developed algorithms by utilizing seven different metrics to represent three different spatial properties of each of the testbeds: a) size, b) density, and c) shape of spaces within each testbed. These metrics are; i) total area of the testbed, ii) average area of the spaces in the testbed, iii) 75th percentile area of the spaces in the testbed, iv) average room perimeter, v) average form factor, vi) node density, and vii) average edge length. The first four metrics are related to the size of the testbed and the size of the spaces contained in the testbed. The fifth metric is related to the shape of the spaces contained in the testbed and the last two metrics represent the density of spaces in the testbed. Most of the metrics are self-explanatory except for ‘form factor’, which is the ratio of the area of a room polygon to the area of the smallest circumscribing circle of the room polygon. This metric measures the deviation of the shape of a room polygon from a square, the lower the form factor, the larger the deviation from a square shape. Longer, narrower hallways have a lower form factor as compared to larger atriums, which have a higher form factor.

Table 1 presents the results for comparison of the abovementioned metrics for the five different testbeds illustrated in Figure 18 through Figure 25. Table 1 depicts the fact that Testbeds 1 and 3 are denser with smaller regular shaped spaces, higher node

density, smaller average edge length, and few spaces with lower form factor. Thus, these two testbeds have regular shaped smaller spaces. On the other hand, Testbed 5 has a large area, very low density of nodes, higher average edge length and higher form factor. These numbers indicate that Testbed 5 has a more open layout with big open spaces and almost non-existent narrow hallways. Figure 29 presents the distribution of form factor for different spaces in the Testbeds 1-5. Testbed 5 has the lowest form factor for almost 100% of spaces contained in it among all the testbeds whereas Testbed 1 has the highest form factor for almost 90% of the spaces contained in it as compared to other testbeds. It is clear from Table 1 and Figure 29 that the five testbeds selected in our evaluations have different spatial characteristics and these varied spatial properties of the five testbeds have been used to validate the *generality* of the algorithms developed in this research paper for automated generation of the three afore-mentioned navigation models.

Table 1 Metrics to compare a) size, b) density, c) shape of spaces within different testbeds

Metric	Testbed_1	Testbed_2	Testbed_3	Tetbed_4	Testbed_5
Total Area (Sq ft)	13854	24655	17333	22127	31613
Avg Room Area (Sq ft)	201	357	247	375	718
75 %ile Room Area (Sq ft)	176	453	220	321	609
Avg Room Perimeter	60	85	69	79	97
Avg Form Factor	0.53	0.5	0.52	0.51	0.5
Node Density (Nodes/Sq ft)	0.02	0.013	0.019	0.014	0.006
Avg Edge Length (ft)	5.26	6.46	5.4	6.25	10.69

Figure 29 Distribution of form factors for different spaces in the five different testbeds

Section 5 presented the factors that affect the performance of the algorithms developed in this research paper for automated generation of navigation models. As stated in Section 5, the prevalence and arrangement of reflex vertices in space polygon, affect the quality of the generated centerline-based network navigation model. Similarly, the limitations of modeling indoor environments using regular square-shaped grids introduces some error in the generated grid-based navigation model. Table 2 presents the effects of these factors on the generated navigation models for Testbeds 1-5. Table 2 illustrates that the percentage of spaces that have larger number of reflex vertices in indoor environments is comparatively very less. Only around 11-16% spaces in all these testbeds have at least one reflex vertex and the other spaces do not contain any reflex vertex. In fact, a large percentage of spaces are regular rectangles and the network model generated for these figures is the same as the Voronoi diagram or the medial axis. There are only a couple spaces, such as the I-shaped hallways that have more than two reflex vertices, and whose network model can deviate from the Voronoi diagram. Testbed 5 does not have any spaces with more than two reflex vertices because it is a testbed with an open layout and does not contain an I-shaped hallway. Hence, the algorithm for

1 generation of centerline-based network navigation model performs well for all the
2 testbeds and different spatial configurations.

3 Table 2 also presents quantitative results on the misclassification of navigable
4 area as obstacles/non-navigable grids in grid-based navigation models. As stated in
5 Section 5, some percentage of the total navigable area is classified as non-navigable due
6 to level-of-detail limitations arising out of using a uniform grid size. Testbed1 has the
7 highest percentage of navigable area misclassified as non-navigable because this testbed
8 has the largest number of spaces, and consequently, the largest number of walls. The
9 more the number of walls, the more the probability of misclassifying navigable area
10 around the walls as non-navigable is. It should be noted here that percentage of navigable
11 area correctly classified as navigable is the complement of the percentage value depicted
12 in the table.

13 Table 2 also presents the number of cases within which diagonal grids, which are
14 physically detached, have been topologically connected in the generated grid-based
15 navigation model. As is clear from the table that the number of these cases is low due to
16 the fact that two conditions have to be true for this error to occur: 1) both the diagonal
17 grids should be navigable while horizontal and vertical grids are non-navigable (as
18 depicted in Figure 3), and 2) the two diagonal grids should be contained in two different
19 spaces that are topologically connected to each other in an IFC-based BIM (as is the case
20 in Figure 3). Overall, the navigation models generated in an automated fashion in this
21 research have a high accuracy of geometrical and topological information contained in
22 the models and the algorithms developed in this research generate navigation models for
23 varied spatial configurations.

1 **Table 2 Quality of navigation models generated in an automated fashion for five testbeds**

	Network Navigation Model			Grid Navigation Model		
	% spaces with deviations from medial axis near reflex vertices			% area inaccurately classified navigable	% area inaccurately classified non-navigable	# inaccurately connected grids
	One deviation	Two deviation	> Two deviations			
Testbed_1	11%	2%	1%	0.2%	4.3%	2
Testbed_2	7%	3.5%	1.2%	0.8%	2.6%	5
Testbed_3	16.3%	1%	0.8%	0.6%	2.2%	2
Testbed_4	15.9%	2.3%	1.1%	0.2%	2.5%	1
Testbed_5	7.7%	0%	0%	0.1%	1.2%	1

2

3 **7. Conclusions**

4 In this research paper, we developed algorithms for automated generation of three
5 different categories of navigation models from IFC-based BIM models that can support
6 map-matching of indoor positioning data. We have modified existing algorithms for
7 generating a modified medial axis, M-MAT, of general polygonal shapes that does not
8 contain any parabolic elements originally present in the Medial axis. The developed M-
9 MAT algorithm improves on an existing algorithm, the S-MAT, which fails for the cases
10 such as an I-shaped hallway. We have also modified existing algorithms for generating
11 metric-based and grid-based navigation models to directly extract geometrical and
12 topological information from IFC-based BIM models thereby reducing errors in
13 deduction of topology from geometrical information contained in other spatial formats.

14 The experimental results of automated generation of navigation models
15 demonstrate that the M-MAT algorithm generates a centerline-based network that is
16 exactly the same as the Voronoi diagram or the Medial axis when the indoor environment
17 only contains convex vertices. In situations where an even number of reflex vertices are
18 present in the indoor environment and these reflex vertices exist symmetrically opposite

1 to each other, the generated centerline-based network closely resembles the Voronoi
2 diagram or the Medial axis; the main difference being the replacement of parabolic
3 elements in Voronoi diagram with linear elements in the generated centerline-based
4 network. The centerline-based network generated from M-MAT algorithm does deviate
5 from the Voronoi diagram or Medial axis in the cases where there are odd number of
6 reflex vertices present or when the reflex vertices do not exist symmetrically in the indoor
7 environment (as illustrated in Section 5). For the grid-based navigation model generation,
8 the developed algorithm misclassifies a small percentage of navigable area as non-
9 navigable and vice versa but as the topological information is extracted directly from
10 BIM models, the number of grids misclassified as topologically connected is negligible.
11 Similarly, the generated metric-based navigation models contain boundary representation
12 with topological information, as penetrable (non-obstruction) or non-penetrable
13 (obstruction) boundaries, which is extracted directly from BIM models.

14 The *generality* of the developed algorithms for generating navigation models is
15 evaluated through the accuracy of the models generated in different indoor environments
16 with varying spatial characteristics, such as size, shape and density of spaces present in
17 indoor environments. Validation results demonstrate that the developed algorithms are
18 able to generate navigation models for dense testbeds with regular rectangular spaces as
19 well as for open testbeds with irregular and large spaces. Taneja [8] already validated the
20 fact that centerline-based network, metric-based and grid-based navigation model satisfy
21 the requirements for spatial queries contained within most of the map-matching
22 algorithms. Hence, the above-stated three types of navigation models that are generated

1 in an automated fashion from the algorithms developed in this research can be utilized to
2 map-match indoor positioning data using most of the existing map-matching algorithms.

3 Future work includes evaluating other types of navigation models, such as metric-
4 based network and visibility-based network (described in detail in Section 3.1) as well as
5 grid-based models with triangular and hexagonal grids, for map-matching indoor
6 positioning data. There is a need to compare the performance of metric-based and
7 visibility-based network navigation models, which are mainly used in indoor
8 environments for code-checking and wayfinding applications [40,41], against centerline-
9 based network models, which have been widely utilized for map-matching in the GPS-
10 GIS domain [1,2,3,4].

11 **Acknowledgements**

12 This project was funded by the Engineer Research and Development Center (ERDC) of
13 the U.S. Army Corps of Engineers (USACE) and by the Pennsylvania Infrastructure
14 Technology Alliance (PITA).

15 **References**

- 16 [1] C.A. Scott, C.R. Drane, An optimal map-aided position estimator for tracking motor
17 vehicles, Proc of 6th Intl Conf on Vehicle Navigation and Information Systems,
18 Seattle (1995), pp. 360-7.
- 19 [2] G. Taylor, G. Blewitt, D. Steup, S. Corbett, A. Car, Road Reduction Filtering for
20 GPS-GIS Navigation, Trans in GIS, 5(3) (2001), pp. 193-207.

- [3] C. Scott, Improving GPS positioning for motor vehicle through map matching. In Proceedings of ION GPS-94, The Seventh International Technical Meeting of the Satellite Division of the Institute of Navigation, Salt Lake City, Utah (1994), pp. 1391-1410.
- [4] G. Taylor, G. Blewitt, Intelligent Positioning: GIS-GPS Unification, John Wiley & Sons, Ltd., 2006.
- [5] X. Zhang, Q. Wang, W. Wan, The Relationship Among Vehicle Positioning Performance, Map Quality and Sensitivities and Feasibilities of Map-Matching Algorithms. IEEE Proc Intelligent Vehicles Symposium (2003), pp. 468-473 doi: 10.1109/IVS.2003.1212956
- [6] M.A. Quddus, R.B. Noland, W.Y. Ochieng, The effects of navigation sensors and spatial road network data quality on the performance of map matching algorithms. GeoInformatica, 13(1) (2009), pp. 85-108.
- [7] H.A. Karimi, T. Conahan, D. Roongpiboonsopit, A Methodology for Predicting Performances of Map-Matching Algorithms, In J.D. Carswell, T. Tezuka, editors. The 6th International Symposium on Web and Wireless Geographical Information Systems, Lecture Notes in Computer Science, Berlin: Springer (2006), pp. 202-213.
- [8] S. Taneja, A framework for evaluation and generation of navigation models for map-matching of indoor positioning data. Ph.D. Thesis, Carnegie Mellon University (2013). Accessible online at <http://gradworks.umi.com/35/73/3573508.html>
- [9] D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello, Bayesian filtering for location estimation, IEEE Pervasive Computing 2 (3) (2003), pp. 24-33.

- 1 [10] G. Glanzer, T. Bernoulli, T. Wiessflecker, U. Walder, Semi-autonomous indoor
2 positioning using MEMS-based inertial measurement units and building information,
3 IEEE Proc Positioning, Navigation and Communication, Hannover, Germany (2009),
4 pp. 135-9, doi: 10.1109/WPNC.2009.4907816.
- 5 [11] L. Liao, D. Fox, J. Hightower, H. Kautz, D. Schulz, Voronoi tracking: Location
6 estimation using sparse and noisy sensor data, IEEE Proc of Intl Conf on Intelligent
7 Robots and Systems, Las Vegas (2003), pp. 723-8.
- 8 [12] I. Spassov, Algorithms for Map-Aided Autonomous Indoor Pedestrian Positioning
9 and Navigation. Ph.D. Thesis, Ecole polytechnique fédérale de Lausanne (EPFL),
10 2007.
- 11 [13] O. Woodman, R. Harle, Pedestrian localisation for indoor environments. ACM Proc
12 UbiComp. (2008) DOI=10.1145/1409635.1409651
- 13 [14] J. Hightower, and G. Borriello, Location systems for ubiquitous computing, IEEE
14 Computer, 34 (8) (2001), pp. 57-66.
- 15 [15] Ekahau, Wi-Fi Based Real-time Tracking and Site Survey Solutions — available at:
16 <http://www.ekahau.com>. (Last accessed on May 6th, 2013).
- 17 [16] R. Want, A. Hopper, V. Falc, J. Gibbons, The Active Badge Location System, ACM
18 Trans. Inf. Syst. 10, 1 (1992), pp. 91-102. DOI=10.1145/128756.128759
19 <http://doi.acm.org/10.1145/128756.128759>
- 20 [17] J. S. Greenfeld, Matching GPS observations to locations on a digital map, Proc of
21 the 81st Annual Meeting of the Trans Res Board, (2002), available at -
22 <http://trid.trb.org/view.aspx?id=711583>.

- 1 [18] M.A. Quddus, R.B. Noland, W. Y. Ochieng, Validation of Map Matching
2 Algorithms using High Precision Positioning with GPS, *J of Nav*, 58(2) (2005), pp.
3 257-271 doi:10.1017/S0373463305003231
- 4 [19] C. White, Some map matching algorithms for personal navigation assistants. *Trans*
5 *Res Part C: Emerg Techn* 8 (1-6) (2000), pp. 91-108.
- 6 [20] J. Bohn, H. Vogt, Robust probabilistic positioning based on High-Level Sensor-
7 Fusion and map knowledge. Technical report No. 421, Institute of Pervasive
8 Computing, ETH Zurich (2003). Available at -
9 <http://www.inf.ethz.ch/research/disstechreps/techreports/show?lang=en&serial=421>
- 10 [21] E.P. Baltsavias, A. Gruen, L.V. Gool, Automatic Extraction of Man-Made Objects
11 from Aerial and Space Images (III), Lisse: A.A. Balkema Publishers (Eds.), 2001.
- 12 [22] P. Doucette, P. Agouris, A. Stefanidis, M. Musavi, Self-organised clustering for road
13 extraction in classified imagery, *ISPRS J. Photogrammet. Remote Sensing* 55 (2001),
14 pp. 347– 358.
- 15 [23] C. Heipke, K. Pakzad, B.M. Straub, Image analysis for GIS data acquisition,
16 *Photogrammet. Rec.* 16 (96) (2000), 963– 985.
- 17 [24] O. Hellwich, C. Wiedemann, Fusion of SAR/INSAR Data and Optical Imagery for
18 Landuse Classification, *Frequenz.* 55(3-4) (2001), pp. 73-136. Retrieved 7 May. 2013,
19 from doi:10.1515/FREQ.2001.55.3-4.129.
- 20 [25] T. Schenk, B. Csatho, Fusion of LIDAR data and aerial imagery for a more complete
21 surface description, *ISPRS. Photogrammetr. Comput. Vision*, Graz, Austria (2002),
22 pp. A-310.
- 23 [26] D.G. Cotts, *The Facility Management Handbook*. 2nd Ed., AMACOM,

1 Broadway, New York, NY 1999.

2 [27] Autodesk, Autodesk DWG – available at
 3 <http://usa.autodesk.com/adsk/servlet/pc/index?id=6703438&siteID=123112> (Last
 4 accessed May 6th, 2013)

5 [28] IFC, Industry Foundation Classes (IFC) Data Model – available at
 6 <http://www.buildingsmart.org/standards/ifc> (Last accessed May 6th, 2013)

7 [29] W. Yuan, M. Schneider, Supporting 3D route planning in indoor space based on the
 8 LEGO representation, ACM Proc SIGSPATIAL, New York (2010), pp. 16-23.

9 [30] J. Lee, A spatial access-oriented implementation of a 3-d gis topological data model
 10 for urban entities, GeoInformatica, 8 (3) (2004), pp. 237-264.

11 [31] D.T. Lee, Medial axis transformation of a planar shape, IEEE Trans. Pattern
 12 Analysis and Machine Intelligence, 4 (1982), pp. 363-369.

13 [32] H. Blum, A Transformation for Extracting New Description of Shape, Symp.
 14 Models for Perception of Speech and Visual Forms, Cambridge, MA: MIT Press
 15 (1967), pp. 362-380.

16 [33] D. Dutta, C.M. Hoffmann, On the Skeleton of Simple CSG Objects, J of Mech
 17 Design, ASME Transactions, 115(1) (1993), pp. 87–94.

18 [34] T. Culver, J. Keyser, D. Manocha, Accurate Computation of the Medial Axis of a
 19 Polyhedron, Proc. of the Fifth Symp. on Solid Modeling and Applications, 1999.

20 [35] K. Hoff, T. Culver, J. Keyser, M. Lin, D. Manocha, Fast Computation of
 21 Generalized Voronoi Diagrams Using Graphics Hardware, Technical Report TR99-
 22 011, Dept. of Comp. Sci., University of North Carolina at Chapel Hill, 1999.

- 1 [36] V. Milenkovic, Robust Construction of the Voronoi Diagram of a Polyhedron, Proc.
2 5th Canadian. Conference on Comp. Geom. (1993), pp. 473-8.
- 3 [37] K. Sugihara, M. Iri, A Robust Topology-oriented Incremental Algorithm for
4 Voronoi Diagrams, Intl J of Comp Geom Appl., 4 (1994), pp.179–228.
- 5 [38] O. Aichholzer, F. Aurenhammer, D. Alberts, B. Gartner, A novel type of skeleton
6 for polygons, J of Univ Comp Sci, 1 (1995), pp. 752-761.
- 7 [39] C. Yao, J. Rokne, A Straightforward Algorithm for Computing the Medial Axis of a
8 Simple Polygon, Intl. J. Computer Mathematics, 39 (1991), pp. 51-60.
- 9 [40] J.-K. Lee, C.M. Eastman, J. Lee, M. Kannala, Y.-S. Jeong, Computing walking
10 distances within buildings using the universal circulation network, Environ Plann B
11 Plann Des, 37 (4) (2010), pp. 628-645.
- 12 [41] M. Kannala, Escape route analysis based on building information models: design
13 and implementation, MSc thesis, Department of Computer Science and Engineering,
14 Helsinki University of Technology, Helsinki, 2005.
- 15 [42] K. Ryall, J. Marks, M. Mazer, S. Shieber, Semi-Automatic Delineation of Regions in
16 Floor Plans, Proc of Third Intl Conf on Document Analysis and Recognition, Montreal
17 (Canada) (1995).
- 18 [43] F. Lamarche, S. Donikian, Crowd of virtual humans: a new approach for real time
19 navigation in complex and structured environments, Computer Graphics Forum 23, 3
20 (2004), pp. 509-518.
- 21 [44] A. Bandera, C. Urdiales, F. Sandoval, A hierarchical approach to grid-based and
22 topological maps integration for autonomous indoor navigation, IEEE Proc Intl Conf
23 on Intell Robots and Sys, 2 (2001), pp. 883-8.

- 1 [45] S. Bandi, D. Thalmann, An Adaptive Spatial Subdivision of the Object Space for
2 Fast Collision Detection of Animated Rigid Bodies, Proc Comput Graph Forum, 14
3 (1995), pp. 259-270.
- 4 [46] S. Thrun, A. Bucken, Integrating grid-based and topological maps for mobile robot
5 navigation, Proc of the thirteenth national conference on Artificial intelligence - Vol.
6 AAAI Press, 2 (1996), pp. 944-950
- 7 [47] R. Campbell, P. Flynn, A survey of free-form object representation and recognition
8 techniques, Computer Vision and Image Understanding (CVIU) 81 (2) (2001), pp.
9 166–210.
- 10 [48] OpenIFCTools (2013) – http://www.openifctools.org/Open_IFC_Tools/Home.html
11 (last accessed May 6th, 2013)
- 12 [49] X. Li, C. Claramunt, C. Ray, A grid graph-based model for the analysis of 2D indoor
13 spaces, Computers, Environment and Urban Systems, Vol. 34 (6) (2010), pp. 532-540
- 14 [50] S. Taneja, B. Akinci, J. Garrett, L. Soibelman, B. East, Transforming IFC-Based
15 Building Layout Information into a Geometric Topology Network for Indoor
16 Navigation Assistance, Computing in Civil Engineering (2011), pp. 315-322
- 17 [51] Y.-H. Lin, Y.-S. Liu, G. Gao, X.-G. Han, C.-Y. Lai, M. Gu, The IFC-based path
18 planning for 3D indoor spaces, Advanced Engineering Informatics, Volume 27 (2)
19 (2013), pp. 189-205
- 20 [52] W. Yan, C. Culp, R. Graf, Integrating BIM and gaming for real-time interactive
21 architectural visualization, Automation in Construction, Volume 20 (4) (2011), pp.
22 446-458

- 1 [53] U. Isikdag, S. Zlatanova, J. Underwood, A BIM-Oriented Model for supporting
2 indoor navigation requirements, Computers, Environment and Urban Systems,
3 Volume 41 (2013), pp. 112-123, ISSN 0198-9715
- 4 [54] Y. Song, J. Gong, L. Niu, Y. Li, Y. Jiang, W. Zhang, T. Cui, A grid-based spatial
5 data model for the simulation and analysis of individual behaviours in micro-spatial
6 environments, Simulation Modelling Practice and Theory, Volume 38 (2013), pp. 58-68,
7 ISSN 1569-190X