Environment and Planning B: Planning and Design 2010, volume 37, pages 628 – 645

doi:10.1068/b35124

Computing walking distances within buildings using the universal circulation network

Jin-kook Lee, Charles M Eastman, Jaemin Lee, Matti Kannala¶, Yeon-suk Jeong
College of Architecture, Georgia Institute of Technology, 247 4th Street NW, Atlanta, GA 30332-0155, USA; e-mail: jklee@gatech.edu, chuck.eastman@arch.gatech.edu, jaemin@gatech.edu, matti.kannala@solibri.com, yeon-suk.jeong@gatech.edu
Received 29 September 2008; in revised form 20 July 2009

Abstract. In this paper we define a computational method for measuring walking distances within buildings based on a length-weighted graph structure for a given building model. We name it the universal circulation network (UCN) and it has been implemented as plug-in software in Solibri Model Checker using building information modeling technologies. It provides a new explicitly defined method for representing circulation paths on top of building models supporting further circulation-related analysis as a network application. We define the computing algorithms and how we realize them. We focus not only on the implementation issues, but also on other intrinsic aspects that need to be considered while dealing with pedestrian circulation within buildings. The UCN is determined mainly by the spatial topology and geometry of a given building, and it returns consistent and accurate scalar quantities. It takes into consideration people-movement patterns, reflecting that people tend to walk along the shortest, easiest, and most visible paths. In several actual-design review projects, the UCN has proved that it is of practical benefit, not only to the distance measurement but also the visualization of pedestrian circulation, especially in reviewing building circulation.

1 Introduction

Many people are interested in building circulation for various purposes before construction: architects and designers want to show their design intent using spatial programs, building owners and occupants need to foresee their circulation within the buildings, and design evaluators have to assess building circulation in terms of predefined rules regarding circulation, including some security, metric distance, and performance issues. Building circulation is formally regulated by some qualitative factors from building-design guides, as well as building codes and fire codes. Some building-design guides and building codes provide general methods to measure travel distances between two spaces. However, in the actual-design review regarding issues of building circulation, we found that it is difficult to define a specific and consistent method due to implicit and inadequate descriptions. It is not easy to acquire accurately and consistently measured distances before construction, even though growing numbers of architects design buildings supported by advanced computing environments, especially building information modeling (BIM) technologies. The introduction of automated checking for code compliance and constructability, and in particular circulation checking, using BIM before construction, is a recently adopted trend (Eastman et al, 2008). We are motivated by those varied demands.

A fundamental issue in building circulation is the specific path people take when moving from one space to another. If we observe such behavior and trace their walking on a building floor plan we will see that people do not all follow the same path. Behavior can be varied with more knowledge of the state of the environment and the person. We do not attempt to address such navigational knowledge, which is especially important for way-finding issues, or the detailed modeling of different people's paths,

¶ Also: Solibri Inc., Ilälahdenatu 21A, 00210 Helsinki, Finland.

but rather to define in a consistent way a method for defining 'good paths' that meets assumptions about efficient routes, mostly determined by a given building model. Consequently, walking distances are measured using the paths defined and visualized in the building model. In other words, for both circulation-representation and distance-measurement purposes, we aim to develop a computational method to draw the paths which represent the typical paths taken by people within a given building. We use the term *metric graph* for differentiating our graph structure from the graph which generally transfers topological relations of objects. We define a metric graph structure on top of the building model for representing building circulation and computing walking distances as an implementation and visualization method, and we name it the universal circulation network (UCN). As a BIM-enabled application, it provides a solution to cover the distance-measurement-related problems for design-assessment purposes. Our main goal is to provide an accurate and broadly acceptable computational method of measuring walking distances and representating circulation within buildings using the UCN.

Our focus is on the metric-graph structure determined by the geometry and spatial connections of the building object, rather than by observation of the free circulation of individuals. A given building model has the determining role of creating the graph, and we define a parametric method to generate the metric-graph structure on top of it. On the basis of various interests, a graph structure representing a certain type of circulation could be classified by several factor sets such as agents, environment, scale, and target spaces. We list the object classifications that we focus on.

(1) Circulation agent: people who are walking through and within buildings.

(2) Unit of traversal: a room. The term 'space' is often equivalent to a physically modeled room using a BIM tool in a building, and 'a room' generally means a fully enclosed space boundary with doors. BIM tools allow the assigning of multiple space objects within open spaces or the partitioning of different spaces within rooms defined by a bounded wall perimeter.

(3) Circulation environment: the building especially in early design phases before construction. It has several spaces and doors, and where the building has multiple stories it has vertical access objects such as stairs, elevators, and ramps. Thus the scale of graphs is equivalent to the range of scales with which building architects work.

2 Approaches

2.1 Circulation representation and graphs

The graph-based approach for representing and analyzing circulation and navigational knowledge is commonly accepted by researchers because of its simplicity and efficiency, for example (Werner et al, 2000). As a graph application, the UCN is derived from the BIM, and its length-weighted edges represent the circulation and are used for computing metric distances. The conventional distance-measurement method within spaces is a center-line-based measurement. However, it has a large tolerance value compared with real human-movement paths, and it is not easy to establish a consistent computational method for it. In real-world buildings, there are many difficulties in determining the correct center of certain spaces, such as those with free-standing columns. Kannala (2005) developed a metric-graph structure using his evacuation-graph model that is determined totally by the given building model; it was implemented in Solibri Model Checker (SMC) (http://www.solibri.com). His method provides a precise computational method with consistency for fire-egress-distance measurement based on his graph structure, but it does not reflect actual human-circulation patterns as shown in figure 1(e). We need a consistent method to determine metric distance within buildings with intuitively correct path measurement. Moreover, our goal is to develop

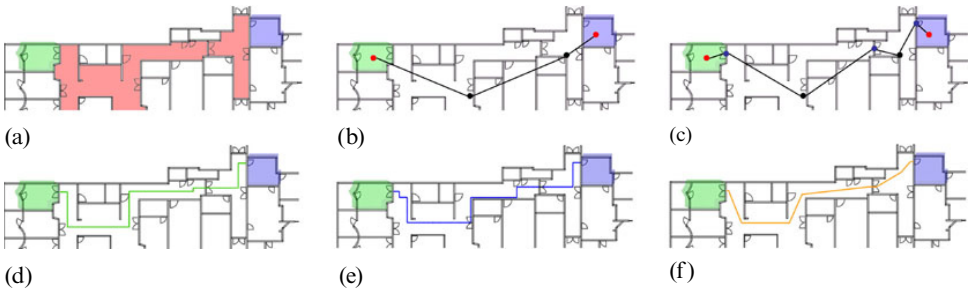


Figure 1. [In color online, see http://dx.doi.org/10.1068/b35124] Various computer-aided circulation graphs: (a) spaces without graphs; (b) topological graph; (c) topological graph with door vertices; (d) center-line-based metric graph; (e) Kannala's (2005) metric graph; and (f) metric graph as a preview of the universal circulation network.

a 'good path' representation method that reflects people's real circulation rather than animals, gaming avatars, or robots.

Figure 1 shows some graphic examples of computational methods for generating circulation between two spaces. There are shaded spaces as a start and a target space, and two corridors between them as circulation spaces. The UCN generates the most efficient trace reflecting the most frequently shown pattern. Moreover, as shown in the last three examples in figure 1, only metric graphs support the measure of metric distances between two spaces. Those three metric graphs return the distance 124′ 11″, 122′ 2″, and 103′ 8″, respectively. We describe how the UCN is generated and how it is used to compute metric distances. Distances are instantly computed by the sum of the edge lengths after generating the metric-graph set. Thus we offer rigorous principles and specifications for implementation, as described in the following sections.

The literature is extensive regarding how circulation could be represented by certain types of information graphics, especially in studies on the perception of architectural space and analysis of the built environment; for example, Benedikt's isovists (1979), linear representations of spatial configuration (Peponis et al, 1998), and visibility graphs and axial map (Turner et al, 2001; 2005). Apart from the deep meaning of analysis on such graphics, they use a certain kind of graph structure for representing both an environmental factor and a human-behavioral aspect affected by it. Similarly, an issue is how to speculate and capture the relation between the given building model (environmental factor) and the circulation-graph representation (human-behavioral aspect). A frequent and well-documented empirical premise about pedestrian circulation is that people tend to follow axial lines because they are visible paths, and they tend to walk following an easy and efficient path (Duckham and Kulik, 2003). In other words, if someone attempts to define an automated representation of pedestrian circulation on a given building model, one of the dominant walking traces will be the shortest, easiest, and most visible path. There are many more factors and conditions that affect human-behavioral aspects: for example, traffic, safety, pleasing aesthetics, knowledge of the building space, natural features, pedestrian amenities, and diversity (Brown et al, 2007). However, they are beyond the scope of this paper. Here we aim to define a computationally coherent method mainly determined by a given building model at the conceptual design level of definition. We are motivated by the visibility-based approach, to provide an answer to the matter of defining 'good circulation paths'.

We have captured two main aspects of circulation for the implementation of the UCN that differentiate our approach from other domain-specific approaches to circulation.

(1) Building-model-oriented circulation: we have developed a building-model-oriented circulation model, rather than an agent-oriented model. It facilitates the use of the

geometry of a given building model and its spatial topology rather the use of other external factors that are uncontrollable.
(2) Most efficient circulation: people generally tend to walk along the shortest, easiest, and most visible paths. Our intention is to represent the most frequently seen movement pattern, and thus the UCN aims to represent those paths. Thus the UCN is also representing reasonable paths for evacuees in emergency circumstances; basically, the UCN does not restrict the target spaces as evacuation does, but pursues a generic method to represent all combinatorial cases of the paths in a given building model.

2.2 The building codes and other related studies

Clear definitions of the graph-representation theory are needed before formal definitions in a consistent method. Our graph representation also aims to follow the rules described in binding regulations for the well-represented graph and its practical use. Some selected rules [refer to the NY Building Code (2004), 13: section 27-360 travel distance] are as follows:

> "Travel distance shall be measured along a natural and unobstructed path of travel. Where the path of travel is over an access stair, it shall be measured along an inclined straight line through the center of the outer edge of each tread.
> ... the centerline of a door from any habitable room within a dwelling unit either to the centerline of a door opening on a corridor or to the center of a door opening on an exit shall not be greater than ... ."

These statements provide not only the distance-measurement method, but also define how to draw the graph edges for some building objects. For example, the edges passing doors or accessing stairs will be represented following their center points. Simultaneously the term 'unobstructed path' gives a clue to generating circulation graphs through and within building spaces defined and affected by walls and columns. In other words, a series of visible paths from one point to another point could be an unobstructed path of travel. Centers of doors and centers of vertical-access objects such as stair treads will be the vertices of the metric graph. If two doors to different spaces are visible from each other, they are simply connected through an edge. In the case of two doors that are not visible from each other due to an L-shape corner, there will be an area that is visible from both doors, and the point in that area which is closest to both doors will be the coupling vertex between two edges from the point to the doors. This is called the 'concave corner point'.

It is difficult to maintain consistency if we assign a specific node to a space as both a start point and a target point. A centroid is a potential node for representing the geometry of any given space, but sometimes the centroid is located outside the space boundary due to the geometry of that space [see figures 1(b) and (c)]. Following some rules from the NY Building Code (2004) and NFPA 101 (2006), the center of a door object is the end node of a circulation path. This is significant for distance measurement of fire egress, but is also applicable to general-purpose distance measurement in most cases. Also building codes and fire codes indicate that in a space the point that is most remote from the door should be a mapping node. For example, travel distances from a certain space to the outdoor exit should be measured from the start space to the point most remote from the door of that space. However, our focus is to define these door-to-door mappings of the circulation graph. If we acquire the distance of the most remote point from a certain start or target space, it could be added to the door-to-door distance between them, and the sum is the distance defined in the codes above. Thus, we are mainly dealing with door-to-door distance measurement.

Dym et al (1988) suggested travel-distance algorithms using shortest paths of corridors based on his method for defining travel distance. Theoretically, among several

edges generated by vertices in a U-shaped corridor, the shortest path between two end points follows a line overlapping some of the corridor boundary edges. However, his method addresses only limited cases and does not reflect human behavior realistically. At least half the width of a person's shoulder from the wall is required for representing metric-graph centerline edges as a well-represented circulation path. We call this the 'minimum buffer distance'.

The fields of computer science and robotics have studied path generation within spaces. Lozano-Perez and Wesley (1979) provided a well-known algorithm for finding collision-free paths against obstacles using visibility graphs. Kuipers et al (2003) presented preliminary computational and empirical tests for their hypothesis on navigational knowledge and pathfinding. The related studies capture overlapped features even though their circulation agent, unit of traverse, and environmental scale are different and limited, as was mentioned in section 1.

2.3 Implementation principles

Synthetically, we list some critical features established for the implementation of UCN. We consider the following to be key specifications for comprehending how the UCN can be actualized.

(1) Buffer from a wall: buffer distance is the minimum distance between a wall and the top-center point of an agent. The default distance is half the width of a person's shoulders. Taking into consideration various situations (for example, a woman with a child, a crowd, a single wheelchair accessing or turning, or two wheelchairs passing) this buffer distance should be adjustable. If there is a space polygon, a buffer polygon could be generated by shrinking the space polygon using a given buffer distance.

(2) Door points: these include the center of a door and two buffer-distance points from the center of the door, perpendicular to its frame, and in opposite directions. It is the most reasonable path for passing through the door, as a spatial-connection method.

(3) Concave points: if there is an L-shaped corridor, we can find one concave point and five convex points along the space-boundary polygon. Concave points are essential for the metric graph because they are intermediate 'coupling' vertices between door points in computing the shortest distance between two doors.

(4) A series of visible vertices generates edges for representing the circulation path: graph edges are generated from the points set described above. Concave points and door points are located on a buffer polygon of a single space object, and door points in the buffer polygon should be connected when they are simply visible from each other. When they are visually obstructed one or more concave points are generated, and the closest concave points will be the intermediate vertices between two doors in a space. In this way, space objects have metric-graph edges if the space has two or more doors and is naturally regarded as a circulation space.

(5) Shortest path: among the edges, and referring to the series of topological space sets, the shortest path is chosen through the shortest-path-finding algorithm for representing the single most efficient path among all those possible between two spaces.

3 BIM-enabled circulation graph

A BIM model can support multiple views of the data contained within a 2D or 3D drawing set, and it can be described by its building objects or its attributes rather than mere building geometry (Eastman et al, 2008). As a BIM-enabled application, the UCN can be generated on top of the BIM model based on space objects, door objects, and vertical-access objects (such as stairs, ramps, and elevators) as a spatial-connection method. The BIM-enabled features facilitate generation of the metric graph, especially for retrieving the geometries of building objects and their topological connections.

In design criteria, BIM represents building-design elements as generic objects or product-specific ones, solid shapes or void-space oriented, that carry their geometry and attributes. BIM provides many advantages that we adopt, making use of the industry foundation classes (IFCs) (IAI, 2008), so that we acquire adequate information for circulation representation and analysis via not only the geometric information but also the spatial topology of the building. Figure 2 shows the hierarchy of some selected IFC entities for describing how we acquire the building-object information and spatial topologies. For example, if we retrieve and track back IfcRelSpaceBoundary entities and their associated IfcSpace and IfcDoor entities, we acquire the spatial-topology information without consideration of the geometric features. IfcSpace, IfcDoor, IfcStair, and IfcRamp provide the main building objects and their geometries needed to represent the metric graph, and the IfcRelSpaceBoundary entity delivers spatial topologies. Here we list the key entities of the IFC view for generating the metric graph:
(1) IfcSpace and its geometry;
(2) IfcDoor and its geometry;
(3) IfcRelSpaceBoundary for acquiring spatial topology information;
(4) vertical-access objects: IfcStair and IfcRamp, for example;
(5) predefined agreement on the naming of objects, to distinguish circulation space and private versus public space; basically, property sets dealing with those data mostly associated with the IfcSpace entity.

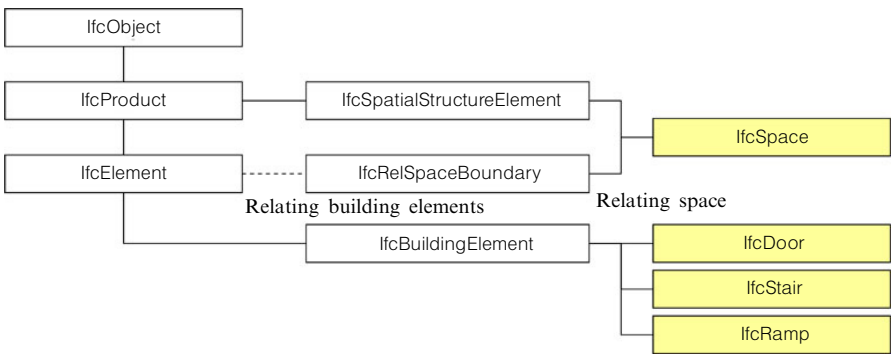


Figure 2. [In color online.] A selected industry foundation class (IFC) entities hierarchy. Shaded objects are main entities for the circulation-graph generation. This also describes how the IFC has spatial topology: IfcRelSpaceBoundary contains 1-to-2 relations between a door and spaces.

A wall object, especially defined by the IfcWall entity within an IFC, is one of the fundamental building objects, but the IfcSpace entity provides adequate spatial information for generating the graph even without walls. Most BIM-authoring tools strongly support parametric design functionalities as delimiters of space boundaries (Eastman, 2008), and most spaces are automatically and parametrically derived from their bounding wall objects and the objects excluded from the space object such as columns. Therefore if we acquire IfcSpace objects adequately, we also gain all geometric conditions affected by walls, columns, or any other obstacles in the space object, as well as its combined additional attributes. Besides, a space object could be functionally divided into two or more spaces for some design purposes, but those subspace objects generated by so-called virtual walls in the virtual spaces do not affect the metric-graph formation. BIM enables focusing on the issues of handling circulation using graph-theory approaches based on its rich building information, and we call the UCN the BIM-enabled circulation graph.

4 Algorithms

4.1 Overview

In this section we provide a formal definition of how the UCN is derived from a given building model, and how it is visualized by the metric-graph structure. A graph structure is generally described in various mathematical definitions, but it is commonly accepted to describe its elements, vertices, and edges using set theory. A graph G is generally described in an ordered binary set: $G = \{V, E\}$ where V is a set of vertices and E is a set of edges each connecting two vertices. We focus on describing their derivation according to logical rules from given building objects using the set-theory approach to the metric graph. The aim of the algorithm is: (1) to define related building objects, (2) to define the metric-graph-generation operators from the building objects, (3) to realize the metric graph from the gathered vertex and edge sets and, after graph generation, (4) we describe how to retrieve the circulation graph as a subgraph of the entire metric-graph structure for the circulation visualization and walking-distance measurements between two spaces. These processes are combined in multiple iterated functions and performed simultaneously in actual implementation, but we explain them separately for clarity. Before constructing our target metric graph, let G_B be our goal of the entire building metric graph, so that G_B is a collection of the floor metric-graph set G_F and vertical-access metric-graph set G_W. G_F covers all metric graphs for all floor levels, and G_W is composed of all metric graphs generated from all vertical-access objects in the building. The floor metric-graph set G_F is composed of space metric-graph set G_S which is generated for every single space classified by the floor F, and they connect to each other by spatial adjacency through the door objects. Therefore the union of G_S and G_W returns G_B. We aim to describe how metric graphs are generated in terms of the building-object-oriented approach, and all metric graphs will be on top of the building model in a coherent way. All BIM authoring tools now explicitly define space polyhedra. The bottom polygon of the space volume is accepted as the space plan boundary (GSA, 2008). The UCN is realized on each floor level with the associated space boundaries sitting on it, and in 3D spaces floors will be connected via the graphs of the vertical-circulation objects as the BIM defines.

4.2 Building object definition

A building contains diverse types of objects in multiple hierarchical structures, and there have been several attempts and well-organized building product models to represent a building digitally for various purposes (Eastman, 1999). We are interested in the objects relevant to circulation, thus our building B is defined as a subset of the building model as follows.

$$B = \{F, W\}, \tag{1}$$

where F is a set of floors in the building and W is a set of vertical-access objects. F and W could be defined in terms of the number of each object as follows:

$$F = \{f_i | 1 \leqslant i \leqslant n;\ n = \text{number of floors in the building } B\}, \tag{2}$$

$$W = \{w_i | 1 \leqslant i \leqslant n;\ n = \text{number of vertical-access objects in the building } B\}. \tag{3}$$

In terms of pedestrian circulation, F carries mainly the circulation spaces of the building, and W settles its interfloor connections. Similar to this, a floor is made up of a set of spaces S_{f_i} on a certain floor f_i, and they make connections through the doors D_{f_i} on floor f_i: $f_i = \{S_{f_i}, D_{f_i}\}$. Thus, B could be represented in an IFC view by the building objects involved in the graph derivation: spaces, doors, and vertical-access objects,

as we described previously.

$$B = \{S, D, W\},$$

where S is the set of space objects, D is the set of door objects, and W is the set of vertical-access objects.

We focus on the fact that the geometry of these objects allows direct derivation of the metric graph as a representation of circulation, and also directly affects the spatial topology. Most of all, space-boundary polygons are very fundamental elements for the metric-graph formation. S and D are defined as follows:

$$S = \{s_i | 1 \leqslant i \leqslant n; n = \text{number of spaces in the building } B\}, \quad (5)$$

$$D = \{d_i | 1 \leqslant i \leqslant n; n = \text{number of door objects in the building } B\}. \quad (6)$$

S denotes a set of space-boundary polygons equal to the space objects, and basically polygons are composed of a finite sequence of line segments. Therefore S, and other building objects also, can be represented by a graph structure and we use an ordered binary set $\{V, E\}$ structure. Space-boundary polygons provide corner points (vertices) and line segments (edges) as a given geometry, and an example of a space-boundary polygon s_i could be represented in terms of the graph structure, as follows:

$$s_i = \{(V_{s_i})_m, (E_{s_i})_n\}, \quad (7)$$

where V_{s_i} and E_{s_i} denote a vertex set and an edge set of the space-boundary polygon s_i, and they have m and n elements, respectively.

A floor set F is made up of S and D, thus all space-boundary polygons s_i can be grouped by a specific floor. Similarly, door objects and vertical-access objects can be classified in the same way. For instance, S_{f_j} can be defined as a set of space-boundary polygons on the floor f_j. In other words, S_{f_j} is satisfied by: $\forall s_i | s_i \in S_{f_j}$. Doors make topological connections between two adjacent space objects; for instance the door d_i always has a binary space object set as a coupling vertex between the spaces. It is the way to define spatial adjacency in IFC, especially by the entity IfcRelSpace-Boundary. Similar to S, door objects d_i could be grouped by each floor. Moreover, d_i can be classified by specific space-object elements for describing spatial adjacency, based on the notational convention: for example, S_{d_i} contains two space instances s_1 and s_2 which are adjacent through door d_i, and D_{s_i} contains at least one door in s_i. Vertical-access objects such as stairs, ramps, and elevators make connections between different floors, similar to the way in which spaces are connected by doors. They could be a part of another space object, but in most cases they are separate spaces. The detailed description for the vertical-access object and its metric graph will be provided in section 4.5.

Basically, an edge is defined by a binary set of vertices. Here we describe the metric-graph structure in the graph definition form using its vertex and edge sets. The following sections explain how these graph elements can be obtained and visualized using the definition scheme of derived objects: for example, spaces, buffered space-boundary polygons, doors, vertical-access objects, and concave points.

4.3 Buffered space-boundary polygons

We use buffered boundary polygons populated from the space objects' boundary polygons for generating buffered graph edges from the wall. Buffer distance is the distance between a wall and the top-center point of a person, and in this paper we assign a default value as half the width of human shoulders; $1'0''$ is acceptable for the default value in terms of human scale. It is not a realistic circulation edge if there is no buffer distance, because sometimes the walking path would be overlapped by walls.

It should be an adjustable value for considering various purposes or agents' conditions, but we use a $1'0''$ default value.

The buffer-distance approach is implemented by using a shrinking polygon from a space-boundary polygon s_i using a given buffer distance b, and it is denoted by $s_{b,i}$. We declare an operator Buffer( ) to compute a shrinking space-boundary polygon from the space-boundary polygon.

$$\text{Buffer}\ (s_i, b) \rightarrow s_{b,i}\ . \tag{8}$$

Input: space-boundary polygon s_i, and given buffer distance value b.

Output: buffered space-boundary polygon $s_{b,i}$ which is shrunk by the length of b.

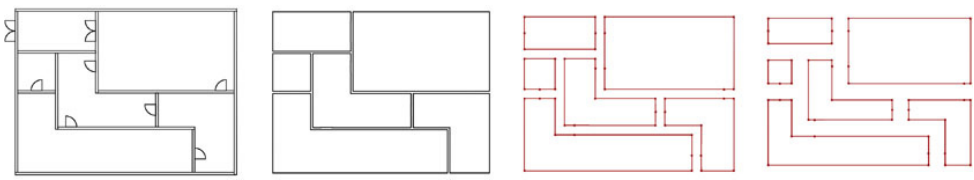
Various geometry libraries provide details of this operation: for example, CGAL (2006).

The metric graph should be generated not on the vertices of s_i, but on the vertices of the buffered boundary polygon. Therefore we define S_b that is derived from S as a set of buffered space-boundary polygons. The instances $s_{b,i}$ are directly derived from s_i. Thus definitions are represented similarly.

$$S_b = \{s_{b,i} | 1 \leqslant i \leqslant n;\ n = \text{number of spaces in the building } B\}\,, \tag{9}$$

$$S_{b,f_j} = \{s_{b,i} | 1 \leqslant i \leqslant n;\ n = \text{number of spaces in the floor } f_j, \text{ and } \forall s_{b,i} \in s_{b,f_j}\}\,. \tag{10}$$

Figure 3 shows how Buffer( ) operator generates S_b. The sample test model has six space objects and the same number of space-boundary polygons s_i, and Buffer( ) computes six buffered space-boundary polygons $s_{b,i}$ using two different buffer-distance values.



(a) (b) (c) (d)

Figure 3. [In color online.] (a) A sample test model; (b) its space-boundary polygons; (c) its buffered space-boundary polygons at $1'0''$ buffer distance; (d) at $2'6''$ buffer distance.

4.4 Space metric graph

We declare a collection of space metric graphs as G_S, generated from the gathered buffered space-boundary polygons, door objects, and vertical-access objects. We define SGraph( ) for space metric-graph generation. The operator SGraph( ) iterates $s_{b,i}$ until obtaining the last space metric graph g_{s_n}, where the building has n spaces, and then G_S will be obtained.

$$\text{SGraph}(s_{b,i},\ d_j, w_k) \rightarrow G_{s_i}\ . \tag{11}$$

Input: buffered space-boundary polygon $s_{b,i}$, door objects d_j, and vertical-access objects w_k in the space object s_i.

Output: a metric-graph set G_{s_i} generated within s_i.

4.4.1 *Doors and visible points*

An instance of space object s_i has at least one door object d_j, where j is the number of doors in s_i, and possibly has vertical-access objects w_k, where k is the number of vertical-access objects in s_i. The edge of the metric graph is generated between the doors in s_i. Considering the fundamental design intent, all spaces which have two or

more doors could be circulation spaces. Here, we disregard the consideration that a space which has multiple doors could have attributes defining it as a private space rather than a circulation space such as a corridor. Such consideration could be handled with additional solutions in the implementation using relevant properties. We add a condition SGraph() as follows:

$$\mathsf{SGraph}(s_{b,i},\ d_j,\ w_k)|j \geqslant 1, \text{ and } k \geqslant 0 \ \rightarrow\ G_{s_i}\ . \tag{12}$$

In particular, if s_i has only one door object, it is not a space for circulation, and G_{s_i} is null. Therefore, s_i requires at least two doors for generating G_{s_i} on it.

$$\mathsf{sGraph}(s_{b,i},\ d_j,\ w_k)|\ j\ =\ 1\ \rightarrow\ G_{s_i}\ =\ \phi\,. \tag{13}$$

A vertical-access object makes a connection to a different floor, but its facing edge to the current s_i should be considered to be another door-like connection vertex. If s_i has an interior vertical-access object w_k, in terms of circulation paths inside s_i w_k is regarded as an obstacle and a door-like object which makes a connection to the outside of s_i. In these comparatively rare cases, a vertical-access object is part of another bigger space object, for example, an interior stairwell in a space. Therefore we need to elaborate on the geometry of s_i with a Boolean difference using s_i minus w_k: $s_i - w_k \rightarrow s_i$, when s_i contains one or more interior vertical-access objects w_k. Figure 4(d) shows this example.

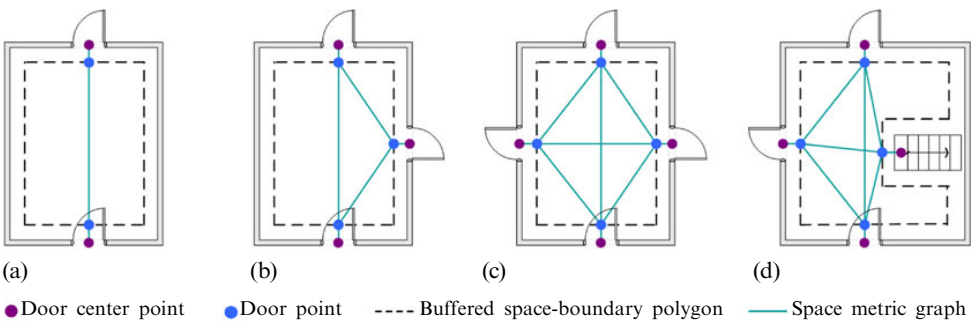


Figure 4. [In color online.] Space metric-graph examples in terms of the number of doors and vertical-access objects. In these cases, all end vertices are visible, and simply connected to each other. (a) 2-door objects; (b) 3-door objects; (c) 4-door objects; (d) 3-door objects and 1 vertical-access object.

Its center point and door-swing orientation are provided by d_j, then door points are gathered by well-known geometric operators such as 2D line-to-line intersection as described in equation (16). Those door center points and door points define simply generated edge pairs, and they are the starting point of space metric-graph generation. Similarly, w_k generates a vertical-access object point and end point as shown in figure 4(d). In figure 4, all door points are simply connected to each other because they are visible from each other. The term 'visible' could be implemented by the method as follows:

$$\begin{aligned}\mathsf{visible}(v_{d_j}, v_{d_k})\ =\ \mathsf{TRUE}\,|\,&\text{no line-to-line intersection between}\\ &e\ =\ \{v_{d_j}, v_{d_k}\}\ \text{and}\ s_{b_i}\ ,\end{aligned} \tag{14}$$

where $\mathsf{visible}(v_{d_j}, v_{d_k})$ returns TRUE or FALSE based on a line-to-line intersection method using two door points v_{d_j} and v_{d_k}. Two lines are an edge $e\ =\ \{v_{d_j}, v_{d_k}\}$ and $s_{b,i}$.

When a vertex set V_{s_i} contains all vertices for an instance of space metric graph G_{s_i}, door center points and door points are simply gathered from s_i.

$$V_{d,c_j} = \{\forall v_{d,c_j} | v_{d,c_j} = \text{door center point from } D_{s_i}\} \rightarrow v_{d,c_j} \in V_{g,s_i}, \quad (15)$$

$$V_{d_j} = \{\forall v_{d_j} | v_{d_j} = \text{intersection point between } e_{d,c_j} \text{ and } s_{b_i}\} \rightarrow v_{d_j} \in V_{g,s_i}, \quad (16)$$

where v_{d,c_j} is an instance of the door center point of door d_j in s_i, v_{d_k} is a door point; and e_{d,c_j} is an edge generated by a projection normal to the closest s_{b_i} from v_{d,c_j}; g_{s_i} is a path instance of G_{s_i}.

For the interior vertical-access objects in s_i as shown in figure 4(d), vertical-access points could be gathered in the same way as door points.

$$V_{w,c_k} = \{\forall v_{w,c_k} | v_{w,c_k} = \text{end center point of } w_{s_i}\} \rightarrow v_{w,c_k} \in V_{g,s_i}, \quad (17)$$

$$V_{w_k} = \{\forall v_{w_k} | v_{w_k} = \text{intersection point between } e_{w,c_k} \text{ and } s_{b_i}\} \rightarrow v_{w_k} \in V_{g,s_i}, \quad (18)$$

where v_{w_k} is a center point of the vertical-access object w_k in s_i, v_{w_k} is a vertical-access point; and e_{w,c_k} is an edge generated by a projection normal to the closest s_{b_i} from v_{w,c_k}.

If we simply map v_{w,c_k} to v_{d,c_j} and v_{w_k} to v_{d_j} because they work in same way, we now have two types of vertex: door center-point set V_{d,c_j} and door point set V_{d_j}, and they are always a binary set for an edge. All are used for the space metric-graph set: $G_{s_i} = \{V_{g,s_i}, E_{g,s_i}\}$. Therefore we have a vertex set V_{g,s_i} and edge set E_{g,s_i} for generating g_{s_i}, but this covers the cases when doors are visible from each other. All paths between two doors simply have three edge instances as follows:

$$V_{g,s_i} = \{v_{d,c_j}, v_{d_j}\}, \quad (19)$$

$$E_{g,s_i} = \{(v_{d,c_j}, v_{d_j}), (v_{d_j}, v_{d_{j'}}), (v_{d_{j'}}, v_{d,c_{j'}})\}, \quad (20)$$

where j is an index over the number of doors in s_i, and j' simply denotes another instance of j.

4.4.2 *Concave points*

In real buildings, most cases are different from the case in figure 4. Figure 5 illustrates a sample floor plan for describing the relation between buffered space-boundary polygons and vertex sets: door center points, door points, and concave points. In contrast to the concave points, there is a convex-point set, but it is not relevant in generating the door-to-door metric graph we are interested in. Convex points are applicable for finding the most remote points that are defined in fire codes for obtaining egress distances.

In figure 6 two door points are not visible; hence we need a different point set on $s_{b,i}$ for connecting two different doors, and we use concave points. In the examples, we find one or more concave points along s_{b_i}, and concave points are the intermediate 'coupling' vertices between two different door points. This is described as follows in terms of a series of visible edges:

$$\text{Visible}(v_{d_j}, v_{c_m}) = \text{TRUE} \wedge \text{Visible}(v_{c_m}, v_{d_{j'}}) = \text{TRUE}, \quad (21)$$

where v_{c_m} is the concave point between two different door points v_{d_j} and $v_{d_{j'}}$.

This is the case where v_{c_m} is the only concave point between v_{d_j} and v_{d_k}, as shown in figure 6(c). If there are several v_{c_m} similar to the other cases in figure 6, the shortest path between v_{d_j} and v_{d_k} will be retrieved from the collection of v_{c_m}. There could be more edges (v_{c_m}, c_{m+1}), $(v_{c_{m+1}}, v_{c_{m+2}})$, and $(v_{c_{m+n-1}}, v_{c_{m+n}})$ for g_{s_i}, for g_{s_i}, where n is the number of concave points used. Coupling points are always one or more concave points when the two door points are not visible from each other. Figure 6 shows some simple concave-point examples: an obstructed path due to a square or round

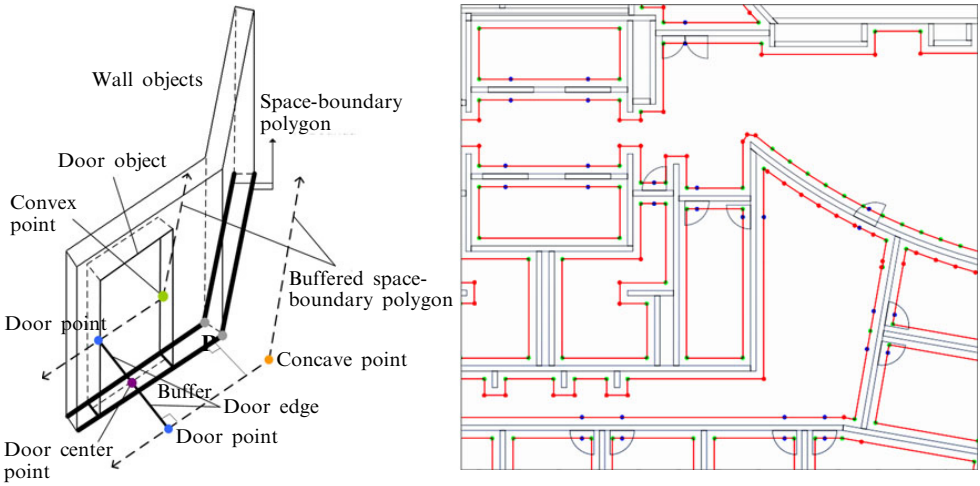


Figure 5. [In color online.] Door points, convex points, and concave points determined by the buffer polygons.

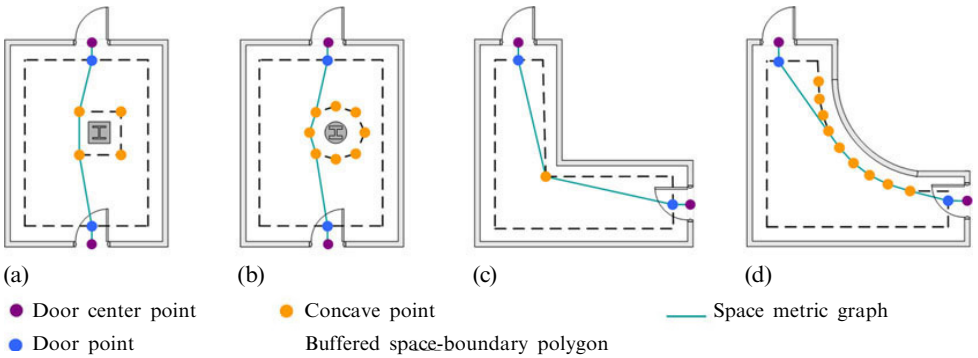


Figure 6. [In color online.] Space metric graph examples based on the concave points between two doors. (a) square column obstruction; (b) round column obstruction; (c) L-shaped corner; (d) rounded corner.

column, and an L-shaped corner and a rounded corner and their concave-point segments. In short, the geometric complexity of the path between two doors is determined by the number of concave points derived from the geometry of the given space, as well as their physical distance apart.

Now we obtain more vertices for V_{s_i}, concave vertices v_{c_m}:

$$V_{c_m} = \{\forall v_{c_m} | v_{c_m} = \text{concave points on } s_{b_i}\} \rightarrow v_{c_m} \in V_{g,s_i}. \tag{22}$$

Based on the parameter set s_{b_i}, d_j, w_k, the final space metric graph set G_{s_i} of the space instance s_i can now be generated in terms of the structure $G_{s_i} = \{V_{g,s_i}, E_{g,s_i}\}$, as follows:

$$V_{g,s_i} = \{v_{d,c_j}, v_{d_j}, v_{c_m}\}, \tag{23}$$

$$E_{g,s_i} = \{(v_{d,c_j}, v_{d_j}), (v_{d_j}, v_{c_m}), (v_{c_m}, v_{c_{m'}}), \ldots, (v_{c_{m+n}}, v_{d_{j'}}), (v_{d_{j'}}, v_{d,c_{j'}})\}, \tag{24}$$

where j denotes the number of doors in s_i, and j' simply denotes another instance of j. Similarly m denotes the number of concave points in s_i, m denotes the number of concave points in s_i, and n denotes the number of concave points for the path instances.

4.4.3 *Generating space metric-graph edges*

For generating G_{s_i}, all end vertices should be v_{d,c_i}. Door center points v_{d,c_j} are located outside s_i, but all other vertices, such as v_{d_j} and v_{c_m}, are located along s_{b_i}. If V_{g,s_i} has one or more concave points v_{c_m}, that means possibly some v_{d_j} are not visible from each other; thus v_{d_j} should be connected through one or more v_{c_m} using a well-known shortest-path-finding method such as Dijkstra's algorithm (Gross and Yellen, 1998).

In this level of space metric graph, the total number of paths $n(\mathsf{Path}(g_{s_i}))$ can be calculated as follows, derived from the equation of combination without repetitions. This is not relevant to the metric-graph generation, but applicable to verify that the operator sGraph( ) generates the correct number of circulation paths in s_i.

$$n(\mathsf{Path}(g_{s_i})) = \frac{1}{2}\,[n(v_{d,c_j})]\,[n(v_{d,c_j}) - 1]\,, \tag{25}$$

where $n(\mathsf{Path}(g_{s_i}))$ is the number of shortest metric-graph paths between doors in s_i, $n(v_{d,c_j})$ is the number of door center points. As we described previously, if s_i has an interior vertical-access object, v_{w,c_k} is mapped with v_{d,c_i}.

An example space shown in figure 7 has seven doors; thus it has twenty-one paths between doors ($0.5 \times 7 \times 6$). The edge-generation process will be initiated by forming door edges, and then computing visible paths. As shown in figure 7, all paths will be generated and the process is concluded using the shortest-path-finding method. The space graph set G_S can be derived using the SGraph( ) operator for all space objects in a given building. If all vertex and edge instances of g_{s_i} are gathered by iteration for all doors d_j in s_i, they are a collection of paths between different doors in s_i, and the space metric graph set G_{s_i} is obtained. Finally, the iteration over all space instances s_i for obtaining G_{s_i} will generate the entire space metric-graph set G_S.

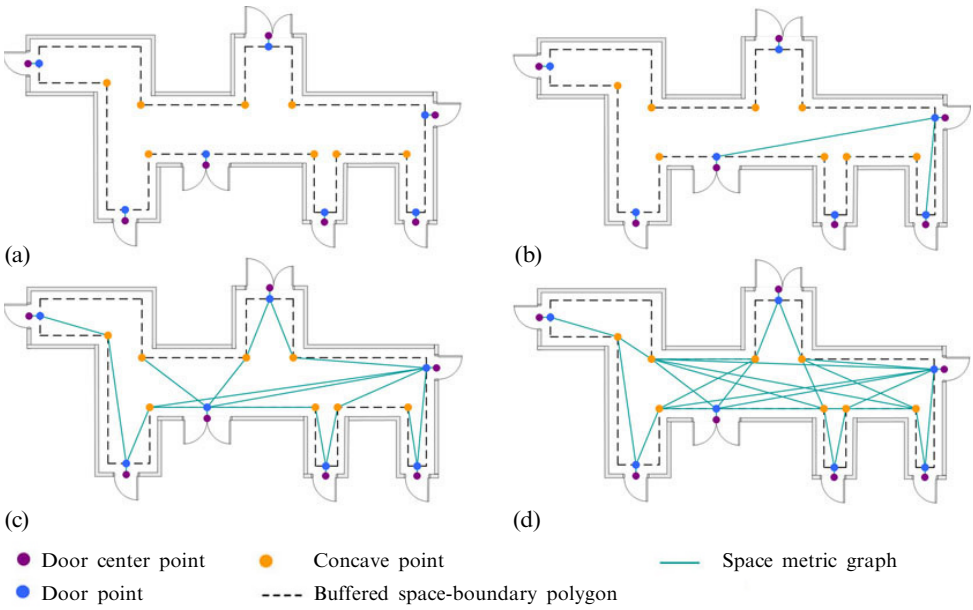


Figure 7. [In color online.] An example of the generation of a space metric graph G_{s_i} on a corridor space: (a) door edges; (b) visible door edges; (c) shortest and visible edge generation from door points to concave points; and (d) generating all shortest edges between door points using concave points.

4.5 Vertical-access object metric graph

Vertical-access objects are stairs, ramps, and elevators, and they have the role of coupling graph edges between space metric graphs of different floors. We define an operator WGraph( ) for generating g_{w_k} as follows.

$$\mathsf{WGraph}(s_{b_i}, d_j, w_k) \rightarrow g_{w_k} . \tag{26}$$

Input: buffered space-boundary polygon s_{b_i}, which is directly derived from s_i using the Buffer( ) operator, and s_i contains vertical-access objects w_k and door objects d_j.
Output: an instance of metric graph g_{w_k} generated within w_k and s_i.

As shown in figure 8, the metric graph on vertical-access objects G_W is generated in a slightly different way. We apply an object center-point-based graph generation instead of using the concave-point-based shortest paths we used in the space metric-graph generation, for considering access – safety issues rather than access efficiency. Moreover, most fire codes define the method for measuring egress paths using the center-line-based measurement for vertical access. A stairwell object in figure 8 shows that all vertices are located on the center of risers or landings. Especially on the landing which has a connection to another space object through a door, a buffered space-boundary polygon similar to the enclosing space object is generated, and this is used for generating edges to the door center point. Similarly, an elevator also uses the center-point-based method for generating G_W on it, and the ramps are the same as stairs.

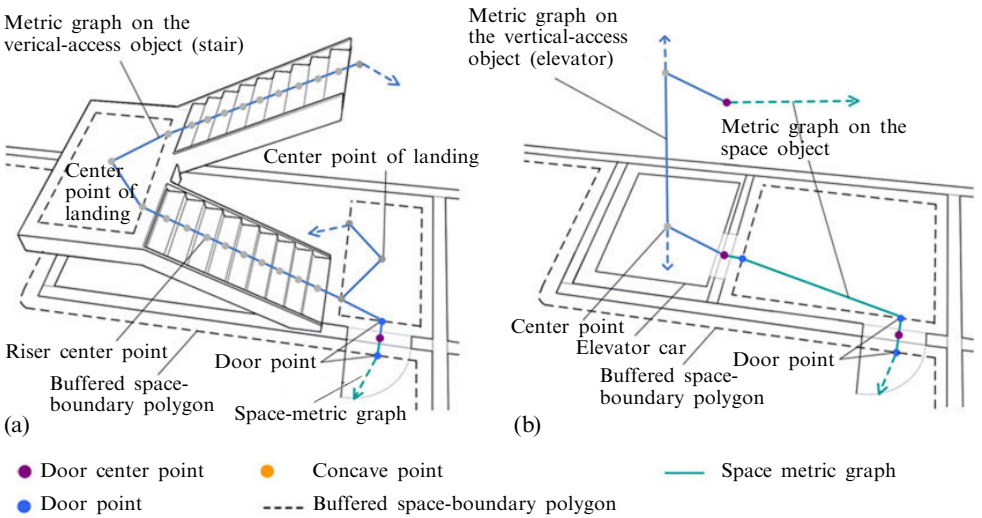


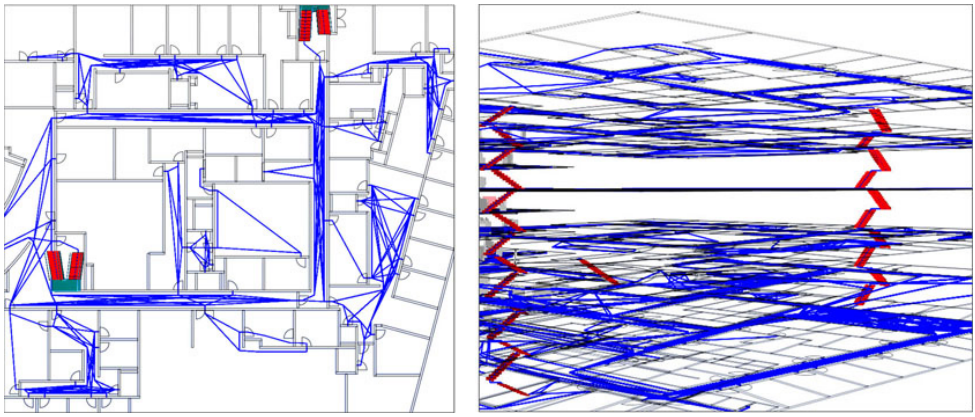
Figure 8. [In color online.] An example of the metric graph on vertical-access objects: (a) stair and (b) elevator.

4.6 Building metric graph and circulation graph

As described earlier, the building metric graph G_B is the union of gathered metric-graph sets: the space object metric graph G_S and vertical-access object metric graph G_W:

$$G_S \cup G_W \rightarrow G_B . \tag{27}$$

G_B covers the entire building-level circulation paths between any two spaces in the building. It is the main structure of the UCN that we have described in this paper. An example of the UCN is shown in figure 9. The test model shown in figure 9 has 6 levels, 1133 spaces, and 106 vertical-access objects including 65 elevator spaces and



(a) (b)

Figure 9. [In color online.] Building metric-graph examples representing (a) a selective floor and (b) interfloor circulation on the test building model.

41 stair spaces. In the graph structure, the building geometry generates 2206 door points, 4281 concave points, and a total of 16 067 edges.

As a subset of the G_B, a circulation-path graph could be retrieved from G_B. We now describe how it is generated by the operator CGraph( ) for retrieving a circulation graph between two different spaces s_j and $s_{j'}$ in B.

$$\mathrm{CGraph}(s_j, s_{j'}, G_B) \rightarrow g_{b,i}\,. \tag{28}$$

Input: two different space instances s_j and $s_{j'}$, and a building metric graph G_B.
Output: an instance of a circulation metric graph $g_{b,i}$ between spaces s_j and $s_{j'}$.

A circulation graph instance $g_{b,i}$ is determined by its end vertices: two different door center points as a door-to-door circulation representation. On the basis of the two end vertices which denote two different spaces, the edges are retrieved by the shortest-path-finding algorithms, similar to the method used in SGraph( ). CGraph( ) first retrieves the shortest spatial topology between s_j and $s_{j'}$, such as the topological graph example in figure 1, and then each associated circulation space returns the shortest space metric graph using two end-door vertices for connecting them based on the method described earlier in figures 5 and 6. In other words, a series of SGraph( ) of the circulation spaces and their doors provides the result set of CGraph( ), while two different doors of circulation spaces are continuously connected to each other, from s_j through $s_{j'}$. Some examples are shown later in figure 11. In a reverse way, a set that contains all circulation graphs between any two spaces in the building is equivalent to the building metric graph G_B: $\forall g_{b,i} | g_{b,i} \in G_B \rightarrow G_B$.

Synthetically, we illustrate an overview process of the metric-graph generation as shown in figure 10. It depicts how a given IFC building model derives the metric graphs facilitated by the operators we declared. Those high-level operators can be implemented on the basis of several well-known geometric algorithms and libraries; we implemented it using SMC.

4.7 Computing walking distances

On the basis of the circulation graph representing a route between two spaces, the metric distance is determined by the sum of the gross length of the circulation metric-graph instance $g_{b,i}$. It has a series of vertices and edges, and a length of edge is gathered by standard coordinate-geometry-based calculation. In other words, all edges

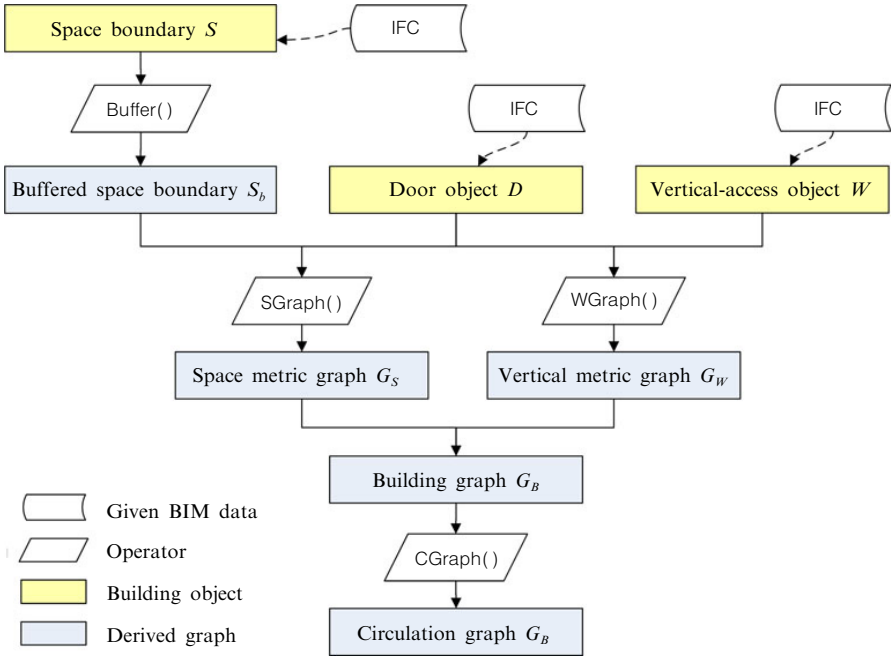


Figure 10. [In color online.] Overview of the universal circulation network generation process. Operators are Buffer( ), SGraph( ), WGraph( ), and CGraph( ): the collection of several low-level geometric operations. IFC = industry foundation classes; BIM = building information modeling.

(a) (b) (c)

Figure 11. [In color online.] Circulation metric-graph examples and their walking distance measurement on a building, captured from the Solibri model checker. The measured distances are; (a) 116′ 2″; (b) 163′ 5″; and (c) 201′ 3″.

are length-weighted; thus we call the metric graph a length-weighted graph structure for computing distances. These have already been obtained and used in previous sections in the shortest-path-finding method for generating appropriate metric-graph edges. A metric distance of the route can be computed by the metric-graph structure using the following equation:

$$d(g_{b,i}) = \sum_{i=1}^{m} l_i \,, \tag{29}$$

where $d(g_{b,i})$ is the metric distance between two spaces represented by $g_{b,i}$, m is the number of edges of $g_{b,i}$ between two spaces, and $l_i = \{l_1, l_2, l_3 \ldots l_m\}$ is the calculated-length set of edge instances through $g_{b,i}$.

Figures 11(a)–(c) represents some visualized examples of circulation graphs retrieved from the UCN. They return the distances 116′ 2″, 163′ 5″, and 201′ 3″, respectively. All colored spaces indicate the spaces involved in the circulation path, and figure 11(c) includes a vertical-access object—a stair within the path. Metric distances are directly and instantly calculated from the circulation graph generated by the model. Metric distances should be differentiated as either horizontal or vertical, because equal metric distance instances but in different conditions should be given different weights. Our focus is on defining the algorithms for the UCN; thus we compute them using the same distance weights.

5 Summary

We have defined a new method for computing walking distances within buildings using a metric-graph structure based on the UCN that can be applied to any given building information model. It provides an exact method for distance measurement based on a pedestrian's visibility in traversing a space. Moreover, it reflects some theoretical rationale on architectural-research fields, such as the concept of an efficient and visible path. It will be applicable for several practical purposes such as distance measurement, checking egress-path distances regulated by fire codes, and gross walking-distance checking based on the number of traverses. Simultaneously it provides another metric method of producing a well-represented circulation graph for various circulation-analysis purposes. For a visibility-based approach to circulation modeling, we propose concave points and convex points which are populated by buffer polygons derived from the space polygons. This generates the graph structure in computational coherence and accuracy, facilitated by BIM. Algorithms for implementing the UCN structure were realized in SMC, and they have been tested using various example models. Test results show very precisely measured distances with visually good paths. The UCN is embedded in a wider set of tools that are being developed for use in automated design review projects. In several actual-design review projects, it has proved that it is practical and beneficial not only for distance measurement but also for the visualization of pedestrian circulation especially in the building-circulation review as one of the design processes. We expect these to be a practical use of our approach to generating the metric graph, and some more connotative architectural and theoretical issues could have a place in further research based on our proposed method. As a BIM-enabled graph application, our research was motivated and implemented by the power of BIM. We hope it has a positive and active influence upon future BIM-enabled applications in various projects.

Acknowledgements. This paper is based on a research project supported by a contract from the BIM Initiative Program, Office of the Chief Architect, United States General Services Administration (GSA). Developing the UCN is part of the research and development of the GSA Courts Design Guide Automation Project. The project has been funded since 2006, and is still ongoing under the supervision of Charles Eastman at Georgia Institute of Technology. We wish to express our gratitude to the GSA, Pasi Paasiala at Solibri, and Paola Sanguinetti, Hugo Sheward, and Sherif Abdelmohsen at Georgia Institute of Technology for their helpful cooperation and suggestions for the research and implementation. For more information, visit http://www.gsa.gov/bim, http://www.solibri.com, and http://dcom.arch.gatech.edu/gsa.

References

Benedikt M L, 1979, "To take hold of space: isovists and isovist fields" *Environment and Planning B* **6** 47 – 65

Brown B B, Werner C M, Amburgey J W, Szalay C, 2007, "Walkable route perceptions and physical features" *Environment and Behavior* **39** 34 – 61

CGAL, 2006 *Computational Geometry Algorithm Library User and Reference Manual: All Parts* Release 3.2.1, http://www.cgal.org/Manual/3.2/doc_html/cgal_manual/title.html

Duckham M, Kulik L, 2003, "Simplest paths: automated route selection for navigation" *Lecture Notes in Computer Science* Spatial Information Theory **2825** 169 – 185

Dym C L, Henchey R P, Delis E A, Gonick S, 1988, "A knowledge-based system for automated architectural code checking" *Computer-Aided Design* **20** 137 – 145

Eastman C M, 1999 *Building Product Models: Computer Environments Supporting Design and Construction* (CRC Press, Boca Raton, FL), http://www.crcpress.com

Eastman C M, 2008 *AEC Integration Lab.,* http://bim.arch.gatech.edu

Eastman C M, Teicholz P, Sacks R, Liston K, 2008 *BIM Handbook—A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors* (John Wiley, Haboken, NJ)

Gross J, Yellen J, 1998 *Graph Theory and its Applications, Discrete Mathematics Series* (CRC Press, Boca Raton, FL), http://www.crcpress.com

GSA, 2008 *BIM Guide Series 2: Spatial Program Validation* US General Services Administration, http://www.gsa.gov/bim

IAI, 2008 *IFC Specifications* International Alliance for Interoperability, http://iai-international.org

Kannala M, 2005, "Escape route analysis based on building information models: design and implementation", MSc thesis, Department of Computer Science and Engineering, Helsinki University of Technology, Helsinki

Kuipers B, Tecuci D, Stankiewicz B, 2003, "The skeleton in the cognitive map: a computational and empirical exploration" *Environment and Behavior* **35** 80 – 106

Lozano-Perez T, Wesley M A, 1979, "An algorithm for planning collision-free paths among polyhedral obstacles" *Communications of the ACM* **22** 560 – 570

NFPA 101, 2006 *Life Safety Code: Measurement of Travel Distance to Exits* NFPA101.7.6 and chapters 12 – 42, National Fire Protection Asociation, http://www.nfpa.org/catalog/

NY Building Code, 2004 *Building Code of the City of New York* Department of Citywide Administrative Services New York, http://www.nyc.gov/dcas/

Peponis J, Wineman J, Bafna S, Rashid M, Kim S H, 1998, "On the generation of linear representations of spatial configuration" *Environment and Planning B: Planning and Design* **25** 559 – 576

Turner A, Doxa M, O'Sullivan D, Penn A, 2001, "From isovists to visibility graphs: a methodology for the analysis of architectural space" *Environment and Planning B: Planning and Design* **28** 103 – 121

Turner A, Penn A, Hillier B, 2005, "An algorithmic definition of the axial map" *Environment and Planning B: Planning and Design* **32** 425 – 444

Werner S, Krieg-Brückner B, Herrmann T, 2000, "Modelling navigational knowledge by route graphs" *Spatial Cognition II* LNAI 1849 295 – 316