From (A) to (B): An Algorithmic Approach to Designing for Circulation of People in Buildings

Arash Naderpour¹, Brian R Johnson², Alex Anderson³
Design Machine Group, Department of Architecture, University of Washington, Seattle, USA

¹arash.naderpur@gmail.com, ²brj@uw.edu, ³ata@uw.edu
http://dmg.be.washington.edu

Abstract
One challenging consideration in architectural design is the movement, or circulation, of building occupants. Poor circulation design can lead to daily frustration—and hazardous conditions in emergencies, while good circulation design can promote productivity and occupant safety. Architects generally address the challenges with rules of thumb, personal experience, and simple calculations. Today, availability of computational power invites a more careful approach, but this depends on establishing: (a) effective representation of spaces, (b) an efficient means of determining routes, and (c) suitable design metrics. We present “A2B”, a toolkit that addresses the first two of these, and demonstrate it with a series of simple calculators. The toolkit consists of two components: a semi-automated workflow for converting regular architectural data (floor plans) from CAD or BIM applications into a format usable for analysis, and a modified version of a fast and accurate artificial intelligence pathfinding algorithm called Theta*. Prototype applications based on the toolkit and tested on an education building, illustrate how the A2B toolkit can be used in design evaluation.

Introduction
We refer to the act of moving purposefully from one place to another as navigation. The route followed during such movement is a path, and the process of selecting a path for future movement is known as pathfinding, a process which operates on a representation of physically possible movements known as a navigation model. Aggregate navigation, or circulation, has important implications in the design of buildings. Good circulation helps buildings operate efficiently by reducing congestion (Heerwagen, Kamp Schroer, Powell, & Loftness, 2004); (Waber, Magnolfi, & Lindsay, 2014). However, congestion can be either a positive or a negative quality in architectural space, as it can encourage occupant interaction in some environments (called productive congestion) or create frustration or danger in others, as during emergency egress.

As office work has become dominated by knowledge work, with its emphasis on teamwork, creativity and collaboration, the role of spatial layout and occupant circulation has been identified as an important component of productivity. Good spatial layout creates opportunities for conversations and social interaction as individuals come across one another throughout the normal working day (Brown & Duguid, 2000).

Good circulation is also essential for building safety. Emergency evacuation can be difficult in confusing settings, particularly because in emergency situations occupants must make quick decisions, and their problem-solving behaviour may be confounded by anxiety-induced stress (Passini, 1996). Poor circulation design can increase evacuation time for users of a building during emergency situations and make it hard for first responders and emergency teams to access critical areas.

Computing circulation metrics depends on establishing routes. Unlike route-finding in the automotive world, where road networks naturally form 2D planar graphs, architectural space is continuous, and routes are established dynamically. Further, traditional CAD/BIM representations do not include the necessary spatial connectivity data needed for finding routes through buildings. All of this makes automated analysis challenging.

Traditional tools for analysing circulation patterns in a building tend to rely on expert identification of critical paths, aggregate values based on area, and experience. Effective, high-end, agent-based crowd simulation models that predict the movement of people in a space have been developed (Kalay, 2004); however, they are computationally expensive and time consuming, often taking hours or even several days to return the results for a single design (Nagy, Lorenzo, James, & David, 2017). Our toolkit, A2B, addresses this problem with a data-conversion component that builds a navigation model and a route-finding component called Theta* that has a much lower computational cost (currently 1-2 hours on a laptop) than agent-based tools. Results, often in the form of 2D heat-maps, can offer insights to architects about the best locations for shared amenities in an office, about locations that make it convenient for people to collaborate with each other, about how long it will take to travel from one place to another inside a building, or about optimizing emergency egress routes. The toolkit could also be beneficial for other uses; for example, it could promote occupant health by providing a personalized calculation of how many calories would be burned during travel from one part of a building to another, or by demonstrating the aggregate benefits of routinely using stairs rather than elevators. These benefits would also contribute the efforts of designers and building owners seeking LEED or Living Building Certification.
Background

Simulating human movement inside buildings presents two general challenges:
Since CAD/BIM data sets are not generally developed for simulation purposes, they lack the spatial data needed for navigation (Johnson B. R., 2017).
The complexity of indoor space configuration and freedom of users’ movement inside buildings make automated route finding extremely challenging (Li, et al., 2012).
Nonetheless, recent studies have sought to overcome the challenges of simulating human movement and analysing interior building circulation without using agent-based crowd simulation methods. These methods derive a regular planar graph (a grid) from the floor plan, then build paths that follow edges or diagonals of the grid, often using the popular Dijkstra algorithm, which results in a characteristic zig-zag path that only approximates a natural walking route.
Nagy (Nagy, Lorenzo, James, & David, 2017) proposed one such methodology to evaluate and quantify productive congestion in their building designs, using an evenly sampled traversal graph and the Dijkstra algorithm. Although their method provides a measure of the productive congestion inside a building, it is not efficient in creating navigation models, and the Dijkstra paths leave it unclear whether their analysis results accurately predict what will actually happen.
Chen and Feng (Chen & Feng, 2009) developed two fast flow-control algorithms for simulating real-time emergency evacuation of populous places such as shopping malls, subway stations, and campus buildings. The first algorithm simulates situations in which there is no limit on the allowed number of evacuation paths, while the second limits the number of evacuation paths. They used the Dijkstra algorithm to search a graph generated on top of the floor plan to find the shortest paths between the starting location and each one of the exit doors. In addition to suffering from the Dijkstra zig-zag, it does not account for indoor objects and furniture that can impede movement. Finally, since their approach only generates a “first person” simulation, it provides limited feedback on other parts of the building.
Goetz and Zipf (Goetz & Zipf, 2011) developed a model that represents indoor environments with topologic, semantic and metric information that allows nearly length-optimal routing in the floor plans of complex buildings. Their navigation model included additional semantic information, allowing them to account for obstacles and special semantic areas inside rooms by adding additional nodes to the graph. However, the process of adding nodes around obstacles manually makes the production of the navigation model laborious and slow. In addition, the resulting paths do not cover all the indoor space, since nodes of the navigation graph are limited to doors and some predefined locations in each room.
Xu et al. (Xu, Wei, & Zlatanova, 2016) proposed an indoor pathfinding approach that takes obstacles of different shapes into account by describing them as points, lines, and polygons. They constructed the network of paths by subdividing the free space into triangles using the Delaunay triangulation algorithm and connecting the centre of gravity of each triangle, excluding edges that cross obstacles. Paths are found using the Visibility Graph (De Berg, Kreveld, Overmars, & Schwarzkopf, 2000) approach. Although routes generated in their approach are realistic looking and similar to human movement, their process of extracting the navigation model from the architectural documents and abstracting it is not completely automated, and their method of reducing abstraction in the navigation model to achieve higher accuracy significantly increases computational costs.
Although these studies addressed the challenges of occupant navigation, all of them suffer from several limitations:
Lack of automation in the process of extracting the navigation model.
High levels of abstraction in the navigation model, which fail to account properly for furniture and other obstacles.
Unrealistic zig-zag routes with many unnecessary heading changes generated by pathfinding algorithms.
Our toolkit addresses the general challenges of simulating circulation inside buildings while also overcoming the limitations of previous studies in this field.
The A2B Toolkit

The toolkit and applications described in this paper were implemented in Grasshopper, the visual programming environment included in Robert McNeel and Associates’ Rhinoceros modelling software. They consist of a mix of custom C# and existing Grasshopper components. Additional detail is available in (Naderpour, 2018).
Data Transfer
Although architecture firms follow different workflows in the design process and use a variety of software, most of them use a computer-aided drafting (CAD) or building information modelling (BIM) package for floor plan production. Accordingly, our toolkit starts with the transfer of a digital floor plan.
The most important requirement for developing the navigation model on a floor plan is identifying the humanly traversable, or walkable, areas in the building. This information is often implicit in BIM data (and it is easy to add to a CAD model). For Revit, this is found in the Room-Data. Figure 1 shows a Revit model with Room-Data for each walkable space highlighted with a distinct color. The rest of the transferred floor plan data helps with visualization, but also shows floor areas covered by furniture and other objects (cabinets, sinks, toilets, etc.) that would actually reduce the walkable floor area.
Navigation Model
Our conversion process begins by computing a square bounding box that contains the collection of room outlines (Figure 2). The navigation model is built by sampling this region using a rectangular grid. The grid spacing can be adjusted depending on the complexity of the floorplan. Complex floor plans with many pieces of furniture require a denser grid than open floor plans with
little furniture. The recommended range for grid size is from 20cm to 60cm, since humans cannot pass bottlenecks narrower than 20cm, and grid cells bigger than 60cm will make the navigation model excessively abstract (Figure 3).

The cells of the grid are either defined as walkable or part of an obstacle. Most nodes are also walkable, except for the nodes at corners of obstacles, which are not walkable and cannot be parts of paths. To discriminate walkable nodes from obstacles, A2B applies a test at each of the grid cells (Figure 4). First, it creates a boundary polyline with 32 equally distributed sample points. Next, it tests each of these 32 points for containment within any of the room or furniture outlines that overlap or intersect the cell. If any sample point falls within an object outline, the cell is marked as an obstacle, as are its corner nodes, otherwise it is walkable, and its corners may be included in paths. This method distinguishes not only walkable cells and obstacle-cells, but also creates a buffer about the size of one cell around walls and obstacles, which makes the navigation model more realistic, since it is unlikely that people will walk right next to the walls or furniture inside a space. Figure 5 illustrates the final navigation model generated for a portion of the plan shown in Figure 1. Walkable cells are shown in green (0), while obstacle cells are shown in red (1). Orange cells are the buffer cells around walls and furniture and, like red cells, are not walkable. Furniture items present in the floorplan are treated as obstacles in the navigation model; this reduces the navigation model’s level of abstraction and increases realism of the paths generated by the toolkit.

**Theta**\(^*\) Pathfinding Algorithm

One of the main flaws of many pathfinding studies is the zig-zag routes generated by the pathfinding algorithms used. This is because routes produced by most of the commonly used pathfinding algorithms in indoor navigation research, like A\(^*\) and Dijkstra, are constrained to move parallel to the navigation grid’s edges, or along their diagonals, as illustrated in Figure 6 (Yap, 2002).

To overcome this drawback, we used the Angle-Propagation Theta* algorithm (Nash, Kenny, Sven, & Ariel, 2007), which is a variant of the A\(^*\) algorithm. It generates routes without limiting their headings. The main difference between Theta* and other pathfinding algorithms is that Theta* connects nodes of the navigation grid using a line-of-sight test that allows it to generate paths that deviate from gridlines. A simple analogy for a
line-of-sight test between nodes is putting a lamp on top of a node and shining it in the general direction of travel. Light cannot pass through blocked cells, so all the nodes that are not located in the shadow of blocked cells have line-of-sight to the node under the lamp (Figure 7).

Figure 5: Final navigation model. Zeros indicate walkable cells and ones indicate obstacle-cells.

Figure 6: Undesirable zig-zag path generated by A* algorithm.

Figure 7: Line-of-sight test of the node (S).

Figure 8: The route generated by Theta* algorithm is 5% shorter and has approximately 50% fewer heading changes.

Toolkit Applications

We built several analysis scripts to test the utility of the A2B toolkit, including a maximum exit route checker, a congestion prediction tool, and an occupant-density egress tool. All were tested on an educational building with a complex floor plan (Naderpour, 2018). Space constraints limit us to presenting details about just two of these here.

Congestion prediction prototype

In work environments, serendipitous congestion, which occurs when people come across each other while moving through the building, can have a positive effect on human experience. It can also help improve workplace performance by activating spaces and facilitating productive interaction between employees (Brown, et al., 2014). This is because casual daily conversations and interactions can help make raw ideas and concepts useful to an organization (Brown & Duguid, 2000). In healthcare settings, by contrast, congestion is usually considered a negative attribute of the space because it inhibits effective movement (Johnson & Happ, 1977).

Visualization of Results

In A2B, tight integration of the Grasshopper programming environment and the Rhino modelling and rendering environment makes the visualization of results straightforward. Our vector drawings, coloured “heat maps”, and “super cell” result aggregation strategies all flow from this work and are illustrated in the following examples.
Having information about the locations in a building with a high chance of congestion helps architects and interior designers decide where to position amenities and security features. For instance, architects can place information centres, or shared amenities adjacent to highly congested areas. The first prototype developed in this study identifies the locations with a high chance of congestion inside a given floor plan by measuring the density of possible routes in its spaces.

Areas of a plan that many occupants of the building pass through to access other areas have a high chance of becoming congested. In other words, congestion is more likely to occur in areas of the floor plan that contain many of the routes that occupants take when they move around. Congestion happens when people must walk close to each other. For instance, if two or more individuals approach each other in a narrow space, at some point congestion will occur between them. If the same number of individuals pass through a wide space, congestion will probably not occur. Consequently, congestion is a function of two parameters:

- Number of routes in a sector of the building.
- Area of the sector.

After generating the navigation model (Figure 9, in white), a grid of user-defined “super cells” was created on top of the navigation model. This grid divides the floor plan into areas for measuring congestion (Figure 9, in red). Since the authors could find no research concerning the ideal distance between people for congestion to occur, the size of the super cells was made a user-generated input parameter for this analysis.

$$D = \frac{R}{N \times S^2}$$

Figure 9: Supper cells outlined in red on top of the navigation model, in white.

The toolkit requires the user to provide sets of start points and destination points, from which the toolkit generates paths. In testing, we selected initial locations for both students and faculty. This accounts for students coming and going to class or going to visit their professors in their offices, for example, or for professors going to their colleagues’ offices for meetings. We further assumed that exterior doors, interior stairwells, and elevators are also feasible origin and destination points.

Figure 10 illustrates the 23,350 possible routes generated in the floor plan, for 150 building occupants, between their start points and possible destination points.

In this case we defined the density of routes passing through each supercell as the metric for congestion. Route density was calculated using Equation (1), where $D$ represents density of routes, $R$ is number of routes crossing the super cell, $N$ is the number of walkable navigation cells within the super cell, and $S$ is the cell width of the navigation model.

$$D = \frac{R}{N \times S^2}$$

Figure 10: All the possible routes (white polylines), generated between start and destination points (green circles).

Comparing the results of the analysis with what students and faculty experience during a regular day in the first floor of the building indicates that the results are close to reality, especially during class changes. For instance, the areas around the core of the building are extremely congested locations for brief periods between classes (Figure 12). In addition, the super cells that contain the stairs are highly congested, since individuals need to use them to access the building’s rest rooms. Although these results model the operation of an existing building, the toolkit clearly could provide architects with insights about the quality of the circulation in a proposed building during the design process.
Emergency egress prototype

Emergency evacuation is another situation which is extremely sensitive to the circulation design of the building. While complicated and inappropriate circulation design might increase casualties during a disaster, proper circulation design might save lives and reduce confusion. Therefore, many regulations and rules related to emergency evacuation are provided in building codes. However, rules related to congestion and traffic during egress, which can be more dangerous than the disaster itself in some scenarios, are missing in building codes (Canter, 1980).

Literature related to emergency egress indicates that the most crucial parameter during evacuation of a building is time, which is affected by occupants’ uncertainty when deciding which exit path to take, length of the route to exits, and congestion during an evacuation (Canter, 1980). The circulation design of the building has a direct impact on all these parameters. The second prototype was thus intended to provide data that might be used during the design process, though we used the same floor plan as the first prototype.

The first step in developing an emergency evacuation analysis tool was to develop a new metric for identifying locations in the floorplan with a high chance of occupant collisions during emergency exiting. This provides different information than simple route density, since few collisions occur during ordinary egress because in day-to-day use people pass through the areas of the building at different times. Conversely, locations with a lower density of routes might be highly congested locations during egress if many individuals pass through them at the same time. We needed a time-dependent value.

We chose the highest instantaneous occupant density in each super cell as a metric. To calculate this, individual timelines were computed, assuming all occupants began evacuating at the same time using paths leading to their closest exits. At each time step, Equation (2) was used to calculate occupant density in each super cell, and the maximum values were retained. $D$ represents occupant density in a super cell during the time interval, $H$ is the occupant load in the super cell at that time, $N$ is the number of walkable navigation cells in the super cell, and $S$ is the cell-size of the navigation model.

$$D = \frac{H}{N \times S^2}$$

The super cells in this analysis were reduced to 2m x 2m since congestion during emergency egress occurs in smaller areas than at other times.

In contrast to the first prototype in which the start and destination points were interchangeable and routes...
between all of them were required for the analysis, start points and destination points in the emergency evacuation scenario have to be distinct sets of points. All the possible locations of individuals, including lecture-room seats, as well as staff and faculty offices, are start points, and all the exit doors are destination points during emergency egress. Figure 13 indicates the selected start and destination points in the sample building.

After defining start and destination points, the shortest paths to the exit door closest to each individual were generated. In order to do that, the toolkit generated shortest paths between each start point and all the available exit doors and selected the shortest path for each start point. The closest exit door for each occupant was found based on the shortest realistic walking distance, rather than the Euclidean distance. Figure 13 also shows the paths selected between each occupant and their closest exit. These paths were generated using the AP-Theta* solver, which is part of the toolkit.

The simulation also requires information about travel speeds during evacuation. Much research has been done measuring travel speed on different types of surfaces (horizontal, inclined, etc.). Several factors influence a person’s speed, including age, gender, grouping (family, friends, etc.), physical ability, and environmental conditions. Using data provided in (Fahy & Proulx, 2001) human movement speeds during emergency egress inside public buildings (theatres and educational buildings) are between 0.33 m/s and 2.33 m/s. We chose 1.33 m/s as the average speed of occupants for the test example. When the simulation starts, all occupants begin to move toward the closest exit at the indicated speed. At each time step their positions are updated, a census is taken, and an occupant density computed for each super cell. When the last occupant reaches an exit, time stops. For visualization purposes, we defined an upper limit of 1.25 people per square meter (five occupants in a 2m x 2m super cell) for the occupant density of each cell. In Figure 14 cells with density higher than the upper limit are coloured red, while the rest of the super cells are coloured a shade of blue, with dark blue representing the least congested cells.

Figure 15 is an enlarged view showing the number of walkable navigation model cells in each super cell (designated as A), as well as the maximum occupancy of each super cell during the evacuation period (designated as B). For example, one congested super cell in the upper left corner of the image has 30 walkable cells and had a maximum occupancy of 11 people during the evacuation. The resulting heat map identifies the same choke-points as seen in the actual building during class changes. Tools like the emergency evacuation analysis prototype in this project may provide useful information about emergency evacuation of a building and identify dangerous locations during emergency egress.

Conclusion

In this paper we discussed algorithmic approaches to analysing circulation in buildings. We presented a computational toolkit, A2B, that extracts a navigation model from BIM or CAD data and generates circulation routes between any two locations inside the building. These paths lack the zig-zag character of many earlier projects. We illustrated use of the toolkit by describing two applications: one for detecting areas of a floor plan with a high chance of (positive or negative) congestion, and one for identifying egress choke-points, in both cases with visual feedback for designers to consult during the design process.

Figure 14: Egress peak occupant density heat map.

Figure 15: Enlarged peak occupant density heat map with area and maximum occupancy data superimposed.

The main advantages of our circulation toolkit in comparison to similar computational tools are:

Our circulation toolkit is relatively quick and computationally cheap in comparison to agent-based crowd simulation tools, though it identifies paths inside a building with a level of accuracy comparable to agent-based crowd simulation tools.
Since our approach in generating the navigation model from a regular architectural floor plan is semi-automated, it would be easy to incorporate into routine practice, and it treats furniture as obstacles during simulation, which is a rare feature in similar tools.

The pathfinding algorithm used in developing the circulation toolkit, a modified version of Theta*, is an appropriate pathfinding algorithm for architectural analysis.

**Directions for Future Work**

While the circulation toolkit produces routes that more accurately predict human behaviour than its rigid predecessors, there are many parameters in real-world events that influence individuals, but which computational simulation tools discussed here do not include. Evaluating the outputs of the prototype application in comparison to actual human experience requires additional research.

The recent advent of extremely fast and inexpensive CPUs and GPUs means computational power and speed will no longer be a significant problem for agent-based crowd simulation tools. However, integrating our approach with agent-based simulation approaches to develop additional methods for simulating human movement inside buildings is a promising area of future research.

Finally, while this work seeks to aid designers, occupant navigation in complex and unfamiliar buildings could also benefit from navigation tools comparable to GPS systems used in cars. We implemented Theta* in a smartphone app to demonstrate the feasibility, but such systems require data (maps) and position information for building interiors which is rarely available. Using inertial or camera data, possibly in a shared app, it may be possible to dynamically map existing buildings and track progress along a route, but further development is required.

**References**


