GIO.G: A Generator for Indoor-Outdoor Graphs to Simulate and Analyze Urban Environments

Vasilis E. Sarris
Dept. of Computer Science
University of Pittsburgh
Pittsburgh, USA
v.sarris@pitt.edu

Connor P. Sweeney
Dept. of Computer Science
University of Pittsburgh
Pittsburgh, USA
cps43@pitt.edu

Sean M. Linton
Dept. of Computer Science
University of Pittsburgh
Pittsburgh, USA
sean.linton@pitt.edu

Brian T. Nixon
Dept. of Computer Science
University of Pittsburgh
Pittsburgh, USA
nixon.b@cs.pitt.edu

Panos K. Chrysanthis
Dept. of Computer Science
University of Pittsburgh
Pittsburgh, USA
panos@cs.pitt.edu

Constantinos Costa
Rinnoco Ltd.
Limassol, Cyprus
costa.c@rinnoco.com

2024 25th IEEE International Conference on Mobile Data Management (MDM) | 979-8-3503-7455-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/MDM61037.2024.00050

***Abstract*—Pedestrian-focused modeling of urban environments is difficult due to a lack of publicly available realistic datasets, and the time and labor-intensive manual processes required to make one, creating barriers to effective evaluation and analysis. In this paper, we introduce GIO.G, a Generator for Indoor-Outdoor Graphs, designed to address these challenges and enhance pedestrian-focused simulation in urban environments. GIO.G offers configurable parameters such as building characteristics, urban density, and foot traffic congestion levels, enabling users to explore a wide range of scenarios with precision and scalability. Through a series of scenarios, we highlight GIO.G's unique features and showcase GIO.G's versatility and effectiveness in generating realistic Indoor-Outdoor Graphs.**

***Index Terms*—Data visualization, Data exploration, User interfaces, Graph drawing, Synthetic data, Indoor-Outdoor Graphs**

I. Introduction

Urban environments are dynamic hubs of activity, where the interplay between indoor spaces (e.g., campus buildings and convention centers) and outdoor areas (e.g., parks and plazas) profoundly influences pedestrians' daily experiences. Understanding and accurately modeling these interactions are essential for various pedestrian-centric tasks, including developing applications tailored to pedestrian needs, designing efficient and accessible transportation systems, and creating sustainable and livable city infrastructures.

Traditionally, representing urban environments from a pedestrian-centric perspective involves time-consuming manual modeling processes. These processes often require extensive data collection and specialized software tools, presenting a significant barrier to entry for those interested in studying pedestrian behaviors across diverse urban settings. Additionally, while crowdsourced platforms like OpenStreetMap (OSM) [1] provide valuable data, they often lack complete or consistent information about building topologies, posing challenges for accurate urban modeling efforts.

This work was partially funded by NIH award R01HL159805. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of any of the sponsors.

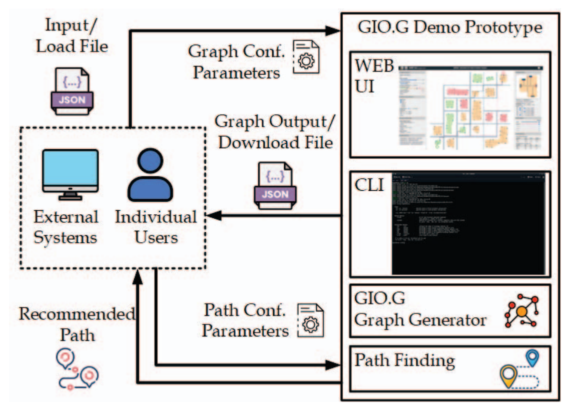

Fig. 1: Overview of the GIO.G Demo prototype system showcasing the data flow and its modules. It should be noted that GIO.G is also usable as a standalone command-line interface (CLI) application without the controller or web-interface.

To address the lack of publicly available pedestrian-focused urban modeling tools / datasets, we extended GIO.G (*Generator for Indoor-Outdoor Graphs*)[1] following the encouragement by the community after our Indoor-Outdoor Graph generator's introduction in [2]. Indoor-Outdoor Graphs are a hierarchical graph structure shown in Fig. 2 and detailed in Section II.

GIO.G is an open-source and configurable tool designed to streamline the modeling process and reduce manual effort, enabling researchers and practitioners to focus more on scenario exploration and analysis. GIO.G offers a systematic and automated approach to generating Indoor-Outdoor Graphs and synthetic foot traffic congestion data based on user-defined

[1]https://github.com/admtlab/GIO.G

2375-0324/24/$31.00 ©2024 IEEE
DOI 10.1109/MDM61037.2024.00050

TABLE I: Configurable GIO Parameters

Notation	Definition	Default
N	Number of Buildings	100
P	Total Area Covered by N Buildings	75%
BC	Buildings Clustered Together	15%
h	High Congestion Buildings	30%
m	Medium Congestion Buildings	40%
l	Low Congestion Buildings	30%
C	Buildings with Constant Congestion	False

environmental parameters. This scalability extends to large environments, making GIO.G suitable for a wide range of urban modeling applications. In this paper, we provide a detailed description of GIO.G and demonstrate its capabilities using pedestrian-path finding algorithms.

We begin by discussing Indoor-Outdoor Graphs and their significance in modeling pedestrian behaviors in urban environments (Section II). Next, we delve into GIO.G's configurable parameters, highlighting its flexibility in accommodating various environmental factors and user preferences (Section III-A). We present the underlying graph generation algorithm, outlining the technical aspects that enable GIO.G to efficiently simulate pedestrian dynamics (Section III-B).

The prototype of GIO.G consists of three modules: (i) a web interface; (ii) GIO.G (graph generator); and (iii) a path finding module (Figure 1). These are discussed in Section IV. The demonstration plan is presented in Section V.

II. Indoor-Outdoor Graphs

GIO.G's foundation is a bidirectional, weighted Indoor-Outdoor Graph $G(V_o, E_o, G_{indoor})$ consisting of *outdoor vertices* V_o, *outdoor edges* E_o, and *Indoor Graphs* $G_{indoor}(V_i, E_i)$. Indoor Graphs G_{indoor} are bidirectional weighted graphs that are comprised of indoor vertices V_i, representing only the doors allowing people to enter and exit buildings, and edges E_i representing the paths between any two indoor vertices. Buildings are represented by outdoor vertices, each of which has an Indoor Graph. The structure of an Indoor-Outdoor Graph enables seamless traversal through both indoor and outdoor spaces.

III. Graph Generation

In this section, we will describe the configurable parameters and the graph generation procedure.

A. Configurable Parameters

The GIO.G interface allows a user to specify several parameters to model an Indoor-Outdoor environment, summarized in Table I. These consist of the number of outdoor vertices in the graph N, the percentage of the grid that should be covered by outdoor vertices P, the percentage of outdoor vertices for which GIO.G will generate high congestion synthetic data h, the percentage of outdoor vertices for which GIO.G will generate medium congestion synthetic data m, and the percentage of outdoor vertices for which GIO.G will generate low congestion

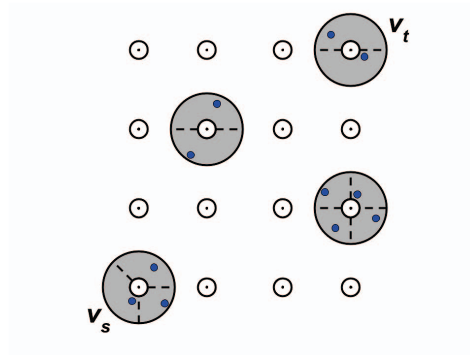

Fig. 2: Example of a fully generated Indoor-Outdoor Graph G using the graph generation parameters $N = 4, P = 0.5$.

synthetic data l. Alternatively, we also provide C, an option for GIO.G to generate the same synthetic congestion data for all outdoor vertices. In this context, "congestion" refers to pedestrian foot traffic. Finally, we have a parameter BC, the percentage of the buildings that should be merged into other buildings to create buildings that span multiple grid cells.

B. Generation Procedure Overview

1) Generating Outdoor Vertices: The generated Indoor-Outdoor graph is built upon a two-dimensional square grid. Each pair of (x, y) coordinates in the grid is a point where an outdoor vertex could be generated. The bounds of this grid B are determined by $\lceil \sqrt{N/P} \rceil$, the minimum bounds required to accommodate N buildings covering P percentage of the grid. Since outdoor vertices are generated over a square grid, the total number of locations where outdoor vertices can be placed T is B^2.

The first phase generates $\lfloor N * (1 - BC) \rfloor$ locations for the base outdoor vertices. These locations are picked from valid, unoccupied (x, y) coordinate pairs at random.

The second phase generates $\lceil N * BC \rceil$ extensions to buildings. These locations for building extensions are chosen by randomly selecting a base outdoor vertex and checking for an unoccupied, adjacent coordinate pair. If one is found, an extension is generated at that location, and "merged" with the original base outdoor vertex. Buildings are merged by combining their indoor vertices into one outdoor vertex, effectively making it a single that spans multiple grid points.

After the building merging process, at least one, if not both, provided generation parameters is guaranteed to fall short (Total number of buildings N or coverage P). The number of buildings will be $\lceil N * BC \rceil$ short of N and the total area of buildings covered will be $\lceil (T * (1 - BC)) - N \rceil$ short of covering a P proportion of T.

For this reason, a third and final phase generates additional, standalone outdoor vertices. By generating $min(\lceil N * BC \rceil, \lceil (T * (1 - BC)) - N \rceil)$ additional buildings, it guarantees at least one of the N or P parameters is completely fulfilled while the other is as close as possible.

Fig. 3: The front-end Web Interface prototype for GIO.G

2) Generating Indoor Graphs: Every outdoor vertex is randomly determined to have between 3-6 indoor vertices, the exact location of each is generated by adding a random polar coordinate offset (r, θ) to the (x, y) coordinates of the outdoor vertex. A polar coordinate offset allows us to easily set a bound on the maximum distance away from the outdoor vertex using r, and to equally partition the areas an indoor vertex can be placed using θ. This is important for three reasons:

1) Setting a lower bound for r guarantees there will always be at least some indoor time traversing a building;
2) Setting an upper bound for r guarantees the indoor vertices will never overlap between outdoor vertices;
3) Assigning ranges of θ to each indoor vertex allows us to distinctly partition the space around the outdoor vertex, guaranteeing no overlapping indoor vertices.

A visualization of the total space that indoor vertices can occupy, an example of a fully generated outdoor vertex, including lines to visualize the θ partitions, and a visualization of a fully generated random graph can be found in Figure 2.

Since Indoor-Outdoor Graphs are used to represent pedestrian environments, GIO.G also generates synthetic foot traffic data for each outdoor vertex as well. When each building is being generated it is labeled as either HIGH, MEDIUM, or LOW. Then for all 288 5-minute windows of a 24-hour day, a HIGH, MEDIUM, or LOW congestion Gaussian Distribution is sampled. Allowing not only variable congestion across the graph, but also throughout the day.

The building generation algorithm, without building clustering, can be found in [2].

IV. Demo Artifact

The demonstration GIO.G artifact consists of a front-end web interface module and two back-end modules, the GIO.G (graph generator) and path finding modules (Figure 1). All three modules communicate by exchanging graph JSONs. The back-end modules were developed using Scala and the Play Framework 2.7. The Path Finding module is an imported version of CAPRIO's core [3], [4].

The front-end web interface is built with standard JavaScript and HTML. The main graph component of the web interface is rendered using Konva.js, a lightweight 2D canvas drawing library, which uses coordinates provided by the Graph Generator and Path Finder to systematically create buildings, routes, and additional shapes.

A. Web Interface

The web interface consists of three panes (Figure 3). The center pane displays the graph generated by GIO.G, and supports controls for panning, zooming, path endpoint selection, and building selection.

The left pane of the web interface configures GIO.G generation options and displayed elements in the following sections:

- *Graph Generator:* Controls the configurable parameters sent to the Graph Generator module as described in Table I in Section III-A.
- *Path Recommendation:* Controls the configurable parameters sent to the Path Finder module, such as accessibility support, path endpoints, and path finding algorithms

Fig. 4: Example interaction (left) showing a randomly generated graph from GIO.G, (center) manually added and deleted buildings, and (right) newly merged buildings.

(ASTRO [5], ASTRO-C [2], ASTRO-Greedy (ASTRO-G) [5], Dijkstra's, Greedy).

- *Manage Graph Data:* Provides the ability to select a preset graph, to upload a custom graph file, and to export the current graph.
- *Display Options:* Contains buttons to toggle visibility of different elements on the main display, such as buildings, roads, paths, congestion colors, and selection highlights.

The right pane of the web interface shows building and path information in the following sections:

- *Building Editor:* Provides interactive building modifications through adjusting the selected building's open/closed status, congestion level, and managing its list of doors. Doors can be created/deleted, update status for open/closed and accessibility, and re-positioned by dragging. There are additional buttons for creating or deleting buildings, and merging the selected building with an adjacent one. Additionally, the building editor supports panning and zooming.
- *Path Legend:* Identifies the line style for each path recommendation algorithm selected in the Path Finding section. Individual paths can also be displayed or hidden.
- *Path Statistics:* Displays statistics comparing the travel time (total time, indoor time, and outdoor time in seconds) and congestion level (average, minimum, and maximum number of $people/m^2$) of the recommended paths.

V. Demonstration

During the demonstration, attendees will have the opportunity to experience the streamlined process of generating and customizing an indoor-outdoor graph via a web-based interface (Figure 3) using a standard laptop.

A. Demo Plan

Scenario 1: The first scenario asks the user to define a set of parameters. GIO.G will use these parameters to generate a potential realization of the environment and display them using the web-based user interface.

Scenario 2: The second scenario asks the user to modify the graph topology by using the web-based user interface. The user will have the ability to create / merge / delete buildings, modify the location of building entrances, and change a building's congestion level.

Scenario 3: The third scenario provides the users with the ability to see how these indoor-outdoor graphs can be used in the context of a real-world application. The users will be able to compare various path finding algorithms performed on GIO.G-generated / user-modified graphs. They will be able to compare them visually as well as quantitatively via statistics.

B. Example Interaction

As an illustration of the three demo scenarios above, assume a user from the University of Pittsburgh wanted to model part of their campus. There are about 25 buildings in a roughly suburban area, additionally, there are 4 large buildings that cover multiple plots. Since it is a college campus, most buildings have a decent amount of foot traffic throughout the day (Figure 4-left). To generate a similar graph, the user would specify $N = 25$, $P = 0.75$, $BC = 0.15$, $h = 0.40$, $m = 0.40$, $l = 0.20$, and $C = False$. To make the generated graph more resemble their campus, they may add some new buildings not already included or remove some existing buildings that occupy spaces where there are public green spaces (Figure 4-center). Finally, they may merge some adjacent buildings and adjust their entrances and exit doors to represent larger lecture halls and libraries (Figure 4-right).

Additionally, they would be able to see an example application built upon the graph via the path finding options available in the web interface.

References

[1] O. contributors, "Planet dump retrieved from https://planet.osm.org," https://www.openstreetmap.org, 2017.

[2] V. E. Sarris, P. K. Chrysanthis, and C. Costa, "Recommending the Least Congested Indoor-Outdoor Paths without Ignoring Time," in *Symposium on Spatial and Temporal Data*, 2023, p. 121–130.

[3] C. Costa, X. Ge, and P. K. Chrysanthis, "CAPRIO: Graph-Based Integration of Indoor and Outdoor Data for Path Discovery," *Proc. VLDB Endow.*, vol. 12, no. 12, p. 1878–1881, Aug. 2019.

[4] C. Costa, X. Ge, E. McEllhenney, E. Kebler, P. K. Chrysanthis, and D. Zeinalipour-Yazti, "CAPRIOv2.0: A Context-Aware Unified Indoor-Outdoor Path Recommendation System," in *IEEE Mobile Data Management*, 2020, pp. 230–231.

[5] C. Anastasiou, C. Costa, P. K. Chrysanthis, C. Shahabi, and D. Zeinalipour-Yazti, "ASTRO: Reducing COVID-19 Exposure through Contact Prediction and Avoidance," *ACM Trans. Spatial Algorithms Syst.*, vol. 8, no. 2, Dec. 2022.