CAPRIO: Context-Aware Path Recommendation exploiting Indoor and Outdoor information

Constantinos Costa, Xiaoyu Ge, Panos K. Chrysanthis

Department of Computer Science, University of Pittsburgh, USA
{costa.c, xiaoyu, panos}@cs.pitt.edu

***Abstract*—During extreme weather conditions and natural disasters caused by meteorological phenomena, it is imperative to enable navigation that minimizes the outdoor section of recommended paths. Existing indoor-outdoor navigation and localization systems have evolved to support queries like the shortest distance, either outdoor or indoor, with additional constraints. However, most of them work in isolation and do not take into consideration the external natural conditions, like the weather, that an individual may experience walking outside during a polar vortex or heatwave.**

In this paper, we present *CAPRIO*, a context-aware path recommendation system whose objectives are two-fold: (i) minimizing outdoor exposure; and (ii) minimizing the distance of the recommended path. We propose a novel graph representation that integrates indoor and outdoor information to discover paths that satisfy outdoor exposure and distance constraints. We measure the efficiency of the proposed solution using two real datasets collected from the University of Pittsburgh and University of Cyprus campuses. We show that we can achieve comparable distance to the state-of-the-art in minimizing outdoor exposure.

***Index Terms*—indoor, outdoor, navigation, path recommendation, graph processing, context-aware.**

I. Introduction

Recently, new navigation and localization services have emerged to enhance shortest path discovery, particularly, because people are spending 90% of their time indoors [1]. New indoor navigation services optimize the shortest path search using magnetic fields for localization and a modified shortest path formulation [2]. An indoor environment has many elements with unique properties that define the indoor route [3]. On the other hand, outdoor systems are well established and enhanced through a variety of data collection and processing techniques, e.g., OpenStreetMap, Google Maps, Bing Maps, Here WeGo, TomTom, Waze. Several systems integrate social network data or crowdsourcing data to produce enriched path recommendations using machine learning to provide alternative navigation services through immersive technologies (e.g., augmented reality) [4], [5], [8].

Unfortunately, navigation solutions primarily focus on indoor or outdoor localization and navigation; Only a handful of the state-of-the-art solutions consider indoor-outdoor seamless transition techniques [6] rather than a unified model that can be beneficial for the shortest path discovery [7]. In many cases, the combination of indoor and outdoor information can produce new, context-aware paths that are essential for the everyday activities of an individual. The following real-world situations increase the possibility of predefined paths becoming non-viable options due to severe weather, or when emergency transportation services are needed the most:

Severe Weather Avoidance: Severe weather can lead to natural disasters like the tsunami in Indonesia 2004, or the heatwave in Japan 2018 as seen in Figure 1 (left), and unbearable outdoor conditions like the Polar vortex in the USA 2019 as seen in Figure 1 (right). *In this scenario, providing a context-aware path that minimizes outdoor exposure is vital for the people who must travel and simultaneously avoid the extreme weather conditions.*

Emergency Transportation: During severe weather conditions, people are more vulnerable to heat (e.g., hyperthermia, heat exhaustion, heat cramps, heat stroke) and cold-related illnesses (e.g., hypothermia, cold weather injuries, frostbite). *In this scenario, being able to transfer severely vulnerable people through a context-aware path with minimum outdoor exposure could play a crucial role during an emergency.*

Minimizing the outdoor exposure of a recommended path is of imperative importance to pedestrians, since the aforementioned scenarios could be very dangerous to one's well-being. The traditional navigation and localization services are not sufficient enough to recommend context-aware paths and improve the quality of the route. Thus, context-aware navigation and localization services should work together to minimize the outdoor exposure while considering the distance of the path.

Fig. 1. (left) The public are being advised to take every precaution to avoid the extreme heat in Japan (BBC 2018), (right) An elementary school closed due to cold weather in Des Moines, Iowa (CNN 2019).

In this paper, we present *CAPRIO*, which combines navigation and localization services to minimize the outdoor exposure and the distance of the path. The core of the system is a novel algorithm, coined *Graph Integrator and Path Discoverer* (*GIPD*), that integrates the external nodes (e.g.,

TABLE I
SUMMARY OF NOTATION

Notation	Description
v_{o_j}, V_o	outdoor/external vertex, set of all outdoor/external vertices $j = 1, \ldots, k$
v_{i_l}, V_i	indoor/internal vertex for an outdoor vertex v_{o_j}, set of all indoor/internal vertices $j = 1, \ldots, m$
e_{o_j}, E_o	outdoor/external edge, set of all outdoor/external edges $j = 1, \ldots, k$
e_{i_l}, E_i	indoor/internal edge, set of all indoor/internal vertices $j = 1, \ldots, m$
$G_o(V_o, E_o)$	outdoor/external graph
$G_i(V_i, E_i)$,	indoor/internal graph
o	outdoor exposure factor
d	distance
P	The recommended path

building) with the internal nodes (e.g., entrances, escalators, exits) to provide a path with minimum outdoor exposure and the shortest distance overall.

The *GIPD* algorithm achieves the integration in a manner that keeps the size of the graph used to search for the shortest path to be no larger than the external graph of the buildings. It allows the *CAPRIO* system to provide a context-aware path in contrast with the traditional path recommended by the existing, well-known systems mentioned above.

The **contributions** of this paper are summarized as follows:

- We propose a novel graph integration algorithm, coined *GIPD*, which solves the discovery of the path based on outdoor exposure and distance metrics (Section II).
- We propose, *CAPRIO*, a context-aware path recommendation prototype system that implements *GIPD* to support the exploration of indoor and outdoor environments (Section III & IV).
- We measure the efficiency of the proposed algorithm using two real datasets consisting of buildings in the University of Pittsburgh and the University of Cyprus campuses. Our evaluation shows that *CAPRIO* can reduce up to 60% of outdoor exposure, while in some cases at the cost of increased indoor distance (Section V).

II. GRAPH-BASED PATH DISCOVERY ALGORITHM

In this section, we introduce the details of our *Graph Integrator and Path Discoverer (GIPD)* algorithm (see Algorithm 1), which is where our main contribution lies. Additionally, this section formalizes our system model, assumptions, and problem. The main symbols and their respective definitions are summarized in Table I.

A. Problem Formulation

Given a specific set of indoor elements along with a source and a destination point, this work aims to minimize the outdoor exposure with the minimum distance producing a path from the source to the destination.

The efficiency of the proposed technique in achieving the above goal is measured by the following objectives:

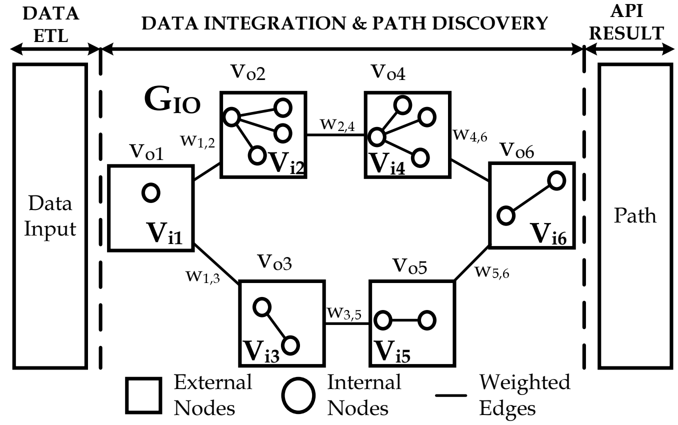

Fig. 2. *CAPRIO* is a system that can provide a recommended path with the minimum outdoor exposure and distance for each request along with the respect source and final destination.

Definition 2.1: Outdoor Exposure (O) is the percentage of the path that is outdoors.

Definition 2.2: Distance (D) is the distance of the path between the source and the destination point.

B. The GIPD Algorithm

In order to set the *GIPD* algorithm in to context it is useful to understand the operational aspects of the our proposed *CAPRIO* system. Consider the case of a request/query submitted to *CAPRIO* shown in Figure 2.

Firstly, *CAPRIO* is extracting, transforming, and loading the data into the system. Then, the *GIPD* algorithm integrates the internal V_i nodes representing entrances, escalators, or exits, and external V_o nodes representing buildings, using a unified graph $G_{IO}(V_i \cup V_o, E_i \cup E_o)$ as shown in Figure 2 (center).

Particularly, the algorithm calculates the weight of each edge using the internal nodes. For example, the weight $w_{1,2}$ of the edge from v_{o_1} to v_{o_2} vertex is calculated using the V_{i_2}, which is a set of internal nodes for the vertex v_{o_2}. Then, the algorithm produces a path between the source s and the final destination f using the well known Dijkstra algorithm on the external graph. The crux of our algorithm is that it controls the scale of the graph to be traversed by the Dijkstra algorithm by fusing the internal graphs as weights in the external graph.

The weight of the external edge is computed through the following equation:

$$\begin{aligned} WEIGHT(v_{o_j}, v_{o_k}, o) = o * DT_i(v_{o_j}, v_{o_k}) \\ + DT_o(v_{o_j}, v_{o_k}) \end{aligned} \tag{1}$$

where o is a tunable outdoor exposure factor that ranges from $0 - 1$ that controls the trade-off between minimizing the outdoor exposure and the overall travel distance d with 0 being the minimum outdoor exposure; $DT_i(v_{o_j}, v_{o_k})$ is the internal travel distance within the node v_{o_j} that is required to go to the exit v_{jk}, which leads to the closest outdoor travel distance between v_{o_j} and v_{o_k}; and $DT_o(v_{o_j}, v_{o_k})$ is the external travel distance between v_{o_j} and v_{o_k}.

Algorithm 1 - *CAPRIO* Path Discovery Algorithm: *Graph Integrator and Path Discoverer (GIPD) algorithm*

Input: s: source;f: destination; V_o: outdoor vertices; V_i: indoor vertices; o: outdoor exposure factor; d: distance
Output: Recommended path P

▷ **Step 1: Graph creation and initialization**
1: $G \leftarrow V_o$ ▷ Initialize a graph using all vertices V_o
▷ **Step 2: Weights population**
2: **for all** $v_{o_j} \in V_o$ **do** ▷ For each edge calculate the weight
3: **for all** $v_{o_k} \in V_o$ **do**
4: **if** $v_{o_j} != v_{o_k}$ **then**
5: $w \leftarrow WEIGHT(v_{o_j}, v_{o_k}, o, d)$
6: **end if**
7: $G \leftarrow EDGE(v_{o_j}, v_{o_k}, w)$
▷ A new edge is added to graph G
8: **end for**
9: **end for**
▷ **Step 3: Shortest path discovery**
10: $P \leftarrow DIJKSTRA(G, s, f)$
▷ Execute Dijkstra algorithm over the newly created graph

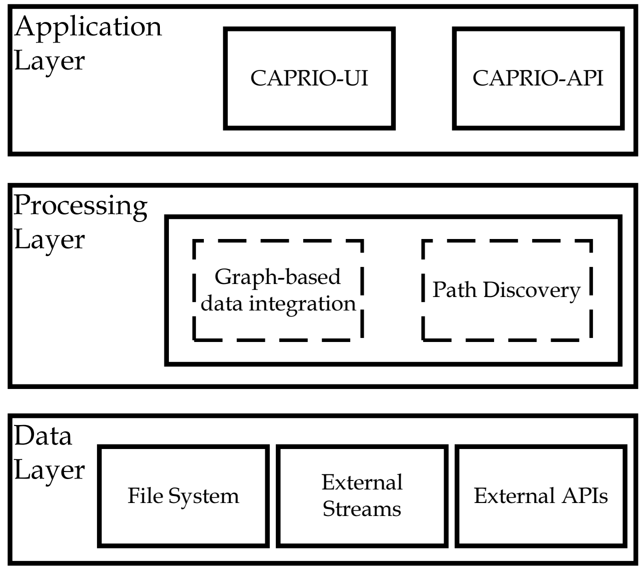

Fig. 3. *CAPRIO* is an efficient graph-based data integration system that enables path discovery and targets the minimization of the outdoor exposure o and the distance d.

Specifically, the *GIPD* algorithm works as follows: as illustrated in Algorithm 1, the graph G is initialized using all the vertices V_o. In the weights population step (Step 2 - lines 2-9), for each edge v_{o_j}, v_{o_k} of external nodes V_o, the weight is calculated using the $WEIGHT$ function (see Equation 1) based on the outdoor exposure o. Then, the $EDGE$ function creates a new edge with the weight w and adds the edge to the graph G. Finally, the shortest path discovery step (Step 3 - line 10) generates the path using the Dijkstra algorithm, which produces a path with minimum path weights.

We designed our proposed algorithm to support any indoor path recommendation system as a plugable internal travel distance calculation engine.

III. The *CAPRIO* Architecture

We express our proposed architecture in three layers (see Figure 3), namely *Data Layer*, *Processing Layer*, and *Application Layer*.

The *Data layer* transforms the data from various sources into a predefined format to ship them over to the Processing Layer. The input data can be regular files on a local or distributed file system, data streams, or external APIs.

The *Processing layer* has a main module with two components, namely *graph-based integration* and *path discovery*. The core module that runs the *GIPD* algorithm converts the data from different sources into external and internal nodes. Then, the core module will integrate the nodes into a graph to produce a path that minimizes outdoor exposure and distance. The *GIPD* algorithm is currently using Anyplace to calculate the internal travel distance. Indoor spaces have many attributes and constraints such as the distance and the accessibility of space [9]. Door to Door distance impacts the calculation and the processing of the graph-based data integration module.

The *GIPD* algorithm is being triggered through a web request using the *CAPRIO* API in order to discover the context-aware path using the above-integrated graph. Initially, the algorithm calculates the weights of each edge of the external graph by examining the travel distance within each internal graph (i.e., nodes). Once the weight for each edge has been assigned, traditional graph techniques could be applied on the external graph that resulted from the previous step in order to obtain the path.

The *Application layer* is equipped with an easy-to-use map-based web interface layer that hides the complexity of the system through a simple and elegant web interface. Additionally, it provides an open API to enable the development of smart application over *CAPRIO* architecture.

CAPRIO has a modular design and an exposed API to allow the scalability and extensibility of the system. This allows the core of the system to be updated using different graph-based algorithms without affecting the user interface.

One of the main advantages of *CAPRIO* is the ability to produce a unified, context-aware path that considers both the external graph that consists of buildings and streets and the internal graph that consists of the entries and exits inside each building. To do so, *CAPRIO* has employed state-of-the-art techniques for construction and integration of both the external and internal graphs.

In particular, for the construction of external graph and routing, *CAPRIO* relies on the Google Maps API, and for the construction of internal graph and routing *CAPRIO* relies on the state-of-the-art indoor navigation system Anyplace [1].

To combine the external graph with the internal graph of each building, *CAPRIO* uses both the street distance reported

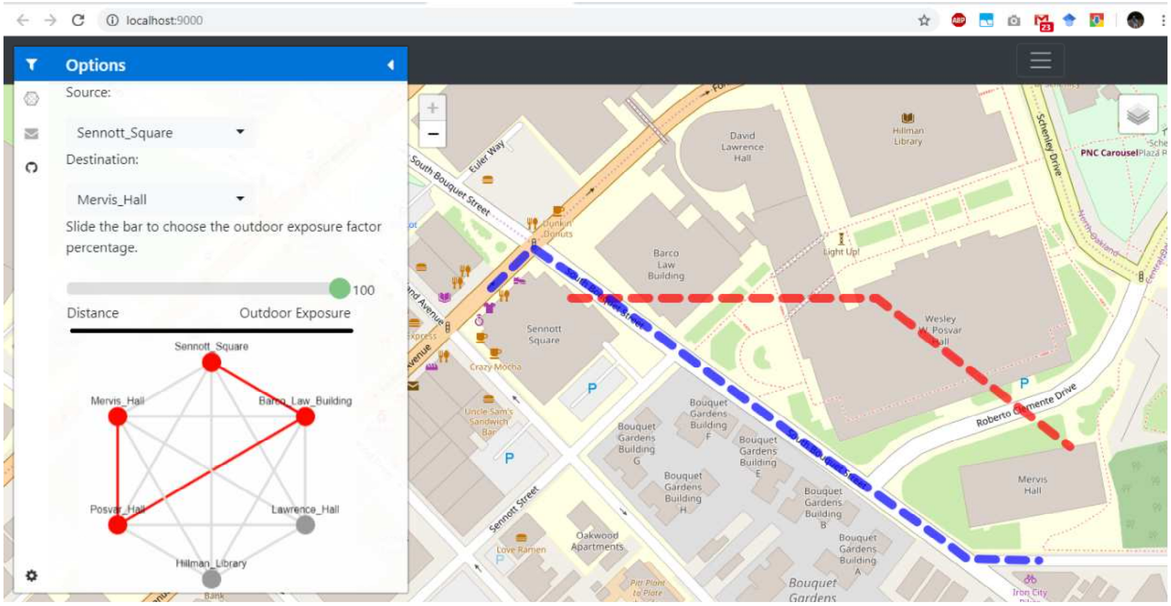

Fig. 4. (top) The *CAPRIO* data exploration user interface was developed on top of Google Maps, which enables the direct comparison between our recommended path (red line) and paths from traditional navigation systems, like Google Maps (blue line). (bottom) The *CAPRIO*'s *GIPD* algorithm can be visualized as an animated graph that shows the resulted graph along with the path comparison on top of Google Maps.

by the Google Maps API and the internal travel distance of each building produced by Anyplace in order to calculate the integrated weights of each external edge.

IV. *CAPRIO* Prototype Description

We have implemented a prototype of *CAPRIO* incorporating an interactive map and integrating several graph techniques in the back-end, which was developed using Play Framework 2.7[1]. The *CAPRIO* web interface is implemented in HTML5/CSS3 along with extensive usage of Leaflet[2] and Cytoscape.js[3].

An illustrative path exploration interface is shown in Figure 4. We have implemented a query sidebar that allows the user to execute a variety of template queries. The query sidebar has three main tabs: (i) the options tab that enables the user to choose the source and the destination for the recommended path along with its outdoor exposure/distance preference, shown in Figure 4 (top); (ii) the graph tab that animates the path using a graph visualization to provide visually the algorithms and techniques behind the paths, illustrated in Figure 4 (bottom); and (iii) the settings tab that activates/deactivates elements on the main user interface.

V. Experimental Methodology and Evaluation

This section presents an experimental evaluation of our proposed *CAPRIO* system and its core GIPD algorithm. We start out with the experimental methodology and setup, followed by two experiments. In the first experiment, the performance of *CAPRIO* is compared against two baseline approaches with respect to various metrics over two real datasets. The second experiment examines the influence of the shortest path control parameters on the performance of *CAPRIO*.

[1]Play Framework: https://www.playframework.com/
[2]Leaflet: https://leafletjs.com/
[3]Cytoscape.js: http://js.cytoscape.org/

A. Methodology

This section provides details regarding the algorithms, metrics, and datasets used for evaluating the performance of the proposed approach.

Testbed: Our evaluation is carried out on a dedicated Windows 10 server. The server is featuring 12GB of RAM with 4 Cores (@ 2.90GHz), a 500 GB SSD and a 750 GB HHD.

Algorithms: The proposed *CAPRIO* is compared with the following approaches:

- **GMaps:** This is the industrial solution by Google, called Google Maps.
- **USP:** We have implemented the unified graph modeling for shortest path proposed in [7].

Note that GIPD is the core algorithm of the proposed solution in this paper, Algorithm 1.

Datasets:

- **PITT:** This is a real dataset that was collected inside the University of Pittsburgh campus and consists of 6 buildings with exits ranging between 2 to 6 per building.
- **UCY:** This is a real dataset that was collected inside the University of Cyprus campus and consists of 11 buildings with exits ranging between 1 to 7 per building.

Metrics: We evaluate the performance of *CAPRIO* using the metrics defined in Section II-A in all experiments:

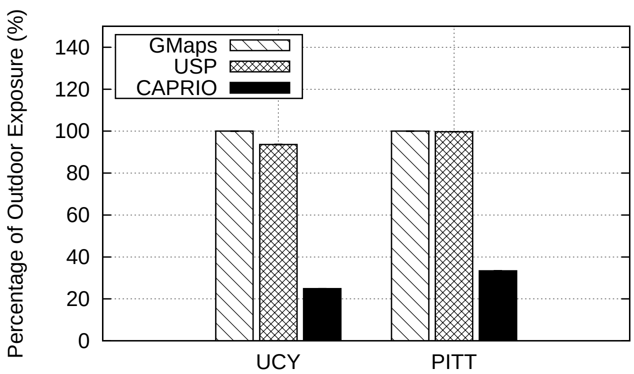

Fig. 5. The outdoor exposure percentage comparing *CAPRIO*, USP, and Google Maps (GMaps) for UCY and PITT dataset using $O = 0\%$, $SPA = Dijkstra$.

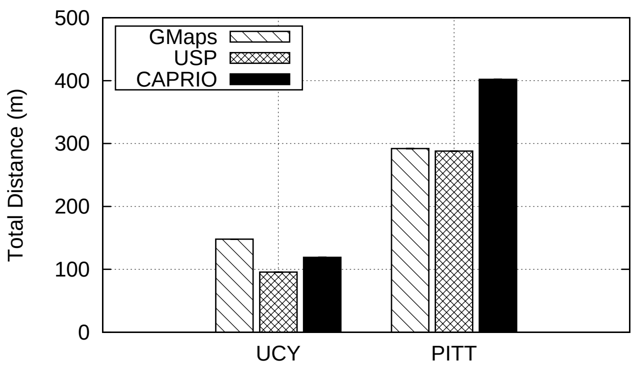

Fig. 6. The distance of the recommended path comparing *CAPRIO*, USP, and Google Maps (GMaps) for UCY and PITT dataset using $O = 0\%$, $SPA = Dijkstra$.

- **Outdoor exposure (*O*):** measures the total outdoor exposure of the recommended path, as a percentage of the whole path.
- **Distance (*D*):** measures the distance of the generated paths.

B. CAPRIO Performance

In the first experiment, we evaluate the performance of the proposed *CAPRIO* algorithm against the two state-of-the-art solutions over the datasets introduced in Section V-A.

In Figure 5, we can easily observe that *CAPRIO* outperforms *GMaps* and *USP* having the lowest outdoor exposure. Particularly, the proposed *CAPRIO* algorithm provides around 35% less outdoor exposure O compared to *GMaps* and *USP* approaches. This is due to the fact that *CAPRIO* prioritizes indoor paths through the integration process for the produced graph, described in Section III.

In terms of distance *D*, *CAPRIO* has slightly longer distance around $100m$ additional distance, in comparison with *GMaps* and *USP* for the PITT dataset, as shown in Figure 6. For the UCY dataset, *CAPRIO* has the slightly shorter distance, around $30m$, in comparison with the *GMaps* and around $20m$ longer than *USP*. This happens due to the fact that the distances

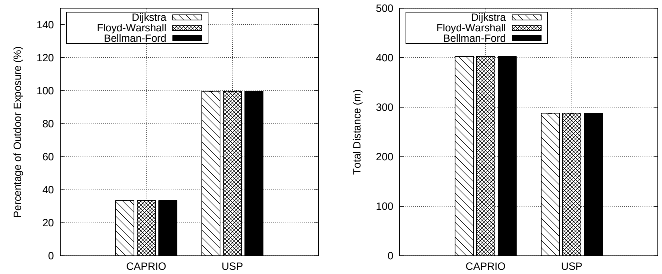

Fig. 7. *CAPRIO* provides the flexibility to run any shortest path algorithm over the result integrated graph. (Left) Outdoor exposure for all three well known SPAs, (Right) Distance for the three well known SPAs

inside the University of Cyprus campus are relatively small and walking through buildings can result in shorter overall distances than an outdoor-only path. The path can be provided by a shortest path algorithm parameter that will be investigated in the next experiment.

Figure 5 and Figure 6 clearly illustrate the trade-off between the outdoor exposure O and the distance D objectives on the results of the compared approaches, since $GMaps$ (only outdoor) approach obtained the worst possible outdoor exposure $O = 100\%$ with a relatively short distance D.

C. Shortest Path Algorithms (SPA)

In this experiment, we examine the influence of the shortest path algorithm parameter on the performance of the proposed *CAPRIO* in terms of O and D.

Figure 7 shows that any shortest path algorithm can be chosen as the SPA parameter without affecting the overall performance of the *GIPD* algorithm in terms of outdoor exposure and distance. This allows *CAPRIO* to be: (i) extremely versatile by replacing the last step of *GIPD* algorithm with any shortest path solution; and (ii) very flexible considering the integration of the weights for the final graph based on different approaches.

VI. Related Work And Background Work

In this section, we present the background and related work in systems that can provide outdoor, indoor, or combined navigation and path recommendation.

A. Outdoor Path Recommendation

Recently, outdoor route recommendation systems have been enriched with external information that may affect the duration of the route. For example, *Dejavu* is a path navigation system that utilizes cell-phone sensors to provide accurate and energy-efficient outdoor localization [10]. Gervey et al. demonstrate how an alternative outdoor path can be generated based on the safety of a route [11]. On the other hand, there are systems that are trying to provide the fastest and simplest route for a destination [12]. Mata et al. show how social network data from a user profile may affect the outdoor path recommendation [13] and enrich it using augmented reality navigation [4]. Other systems focus on the temporal or personal preferences of the user to discover outdoor activities [14].

B. Indoor Path Recommendation

Indoor localization and navigation services have emerged due to the rapid growth of new large buildings (e.g., shopping centers, campuses, building complexes). Anyplace is an infrastructure-free indoor navigation system that uses sensing data from smartphones to determine the user's location [1]. Delail et al. proposed a context-aware system that enriches the indoor information by using augmented reality to provide indoor navigation [15]. Indoor environment and context are very important to determining the best indoor navigation route, especially to impaired people [16], [17]. Park et al. propose an indoor pedestrian network data model for emergency transportation services [18]. Additionally, Afyouni et al. illustrate how to process indoor continuous path queries over traditional database management systems using a hierarchical, context-aware data model [19].

C. Indoor-Outdoor Path Recommendation

The majority of the indoor-outdoor systems focus on the seamless transition between indoor and outdoor navigation [20]. Additionally, *IODetector* detects the indoor-outdoor environment changes accurately and efficiently, allowing the development of context-aware mobile applications [21]. *IONavi* is a joint indoor-outdoor navigation solution that uses mobile crowdsensing to create a collection of indoor-outdoor paths. Then produces an indoor-outdoor path based on the generated collection [8]. Similarly, *CrowdNavi* solves the last-mile navigation problem using crowdsourcing and the guider-follower model [22]. Jensen et al. presented a unified model of indoor and outdoor spaces that can provide the shortest path by exploiting the nature of buildings and roads [7].

VII. Conclusions

In this paper, we present a novel graph-based data integration and routing system, coined *CAPRIO*, that leverages existing graph exploration algorithms and systems to unify both indoor and outdoor information. The goal of the system is to extract a path that satisfies the distance and the outdoor exposure requirements according to a user's preference. Our experimental results also confirm our initial hypothesis that *GIPD* can provide paths using the indoor and outdoor information to minimize the outdoor exposure and the distance.

In the future, we aim to integrate richer contextual information (e.g., traffic, building accessibility, weather conditions) into both the internal and external nodes to produce a more robust and context-aware system that can better assist the user in finding and determining the most appropriate path based on all available information.

VIII. Acknowledgments

We would like to thank Thalia Chrysanthis for her help during the reviewing process, as well as the anonymous reviewers for their helpful comments. This work was supported, in part, by NIH under award U01HL137159. This paper does not represent the views of NIH.

References

[1] D. Zeinalipour-Yazti, C. Laoudias, K. Georgiou, and G. Chatzimilioudis, "Internet-based indoor navigation services," *IEEE Internet Computing*, vol. 21, no. 04, pp. 54–63, jul 2017.

[2] H. Wu, S. He, and S.-H. G. Chan, "Efficient sequence matching and path construction for geomagnetic indoor localization," ser. EWSN '17. Junction Publishing, 2017, pp. 156–167.

[3] C. Salgado, M. A. Cheema, and D. Taniar, "An efficient approximation algorithm for multi-criteria indoor route planning queries," ser. SIGSPATIAL '18. ACM, 2018, pp. 448–451.

[4] F. Mata and C. Claramunt, "A social navigation guide using augmented reality," ser. SIGSPATIAL '14. ACM, 2014, pp. 541–544.

[5] G. Chatzimilioudis, A. Konstantinidis, C. Laoudias, and D. Zeinalipour-Yazti, "Crowdsourcing with smartphones," *IEEE Internet Computing*, vol. 16, no. 5, pp. 36–44, Sep. 2012.

[6] H. S. Maghdid, I. A. Lami, K. Z. Ghafoor, and J. Lloret, "Seamless outdoors-indoors localization solutions on smartphones: Implementation and challenges," *ACM Comput. Surv.*, vol. 48, no. 4, pp. 53:1–53:34, Feb. 2016.

[7] S. K. Jensen, J. T. V. Nielsen, H. Lu, and M. A. Cheema, "Outdoor-indoor space: Unified modeling and shortest path search," ser. ISA '16. ACM, 2016, pp. 35–42.

[8] X. Teng, D. Guo, Y. Guo, X. Zhou, Z. Ding, and Z. Liu, "Ionavi: An indoor-outdoor navigation service via mobile crowdsensing," *ACM Trans. Sen. Netw.*, vol. 13, no. 2, pp. 12:1–12:28, Apr. 2017.

[9] H. Lu, X. Cao, and C. S. Jensen, "A foundation for efficient indoor distance-aware query processing," in *2012 IEEE 28th International Conference on Data Engineering*, April 2012, pp. 438–449.

[10] H. Aly, A. Basalamah, and M. Youssef, "Accurate and energy-efficient gps-less outdoor localization," *ACM Trans. Spatial Algorithms Syst.*, vol. 3, no. 2, pp. 4:1–4:31, Jul. 2017.

[11] M. Garvey, N. Das, J. Su, M. Natraj, and B. Verma, "Passage: A travel safety assistant with safe path recommendations for pedestrians," ser. IUI '16 Companion. ACM, 2016, pp. 84–87.

[12] D. Sacharidis and P. Bouros, "Routing directions: Keeping it fast and simple," ser. SIGSPATIAL'13. ACM, 2013, pp. 164–173.

[13] F. Mata and C. Claramunt, "A mobile trusted path system based on social network data," ser. SIGSPATIAL '15. ACM, 2015, pp. 101:1–101:4.

[14] S. Rahimiaghdam, P. Karagoz, and A. Mutlu, "Personalized time-aware outdoor activity recommendation system," ser. SAC '16. ACM, 2016, pp. 1121–1126.

[15] B. A. Delail, L. Weruaga, and M. J. Zemerly, "Caviar: Context aware visual indoor augmented reality for a university campus," ser. WI-IAT '12. IEEE Computer Society, 2012, pp. 286–290.

[16] N. Fernando, D. McMeekin, and I. Murray, "Poster: Context aware route determination model for mobile indoor navigation systems for vision impaired people," ser. MobiSys '16 Companion. ACM, 2016, pp. 24–24.

[17] B. Li, J. P. Muoz, X. Rong, Q. Chen, J. Xiao, Y. Tian, A. Arditi, and M. Yousuf, "Vision-based mobile indoor assistive navigation aid for blind people," *IEEE Transactions on Mobile Computing*, vol. 18, no. 3, pp. 702–714, March 2019.

[18] S. Park, S. Kim, and K. Yu, "Designing of indoor linkable pedestrian network data model for the transportation vulnerable," ser. ICDSP 2018. ACM, 2018, pp. 57–60.

[19] I. Afyouni, C. Ray, S. Ilarri, and C. Claramunt, "A postgresql extension for continuous path and range queries in indoor mobile environments," *Pervasive Mob. Comput.*, vol. 15, no. C, pp. 128–150, Dec. 2014.

[20] T. Springer, "Mapbiquitous – an approach for integrated indoor/outdoor location-based services," in *Mobile Computing, Applications, and Services*, J. Y. Zhang, J. Wilkiewicz, and A. Nahapetian, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 80–99.

[21] M. Li, P. Zhou, Y. Zheng, Z. Li, and G. Shen, "Iodetector: A generic service for indoor/outdoor detection," *ACM Trans. Sen. Netw.*, vol. 11, no. 2, pp. 28:1–28:29, Dec. 2014.

[22] Q. Wang, B. Guo, Y. Liu, Q. Han, T. Xin, and Z. Yu, "Crowdnavi: Last-mile outdoor navigation for pedestrians using mobile crowdsensing," *Proc. ACM Hum.-Comput. Interact.*, vol. 2, no. CSCW, pp. 179:1–179:23, Nov. 2018.