CAPRIO: Graph-based Integration of Indoor and Outdoor Data for Path Discovery

Constantinos Costa
Dep. of Computer Science
University of Pittsburgh
Pennsylvania, USA
costa.c@cs.pitt.edu

Xiaoyu Ge
Dep. of Computer Science
University of Pittsburgh
Pennsylvania, USA
xiaoyu@cs.pitt.edu

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

ABSTRACT

Recently, navigation and localization systems have emerged to support queries like the shortest distance in either indoor or outdoor with additional constraints. These systems, however, neither combine the indoor and outdoor information nor consider the external natural conditions like the weather that one may face across an outdoor path.

In this demonstration paper we present *CAPRIO*, which proposes and implements a novel graph representation that integrates indoor and outdoor information to discover paths that personalize outdoor exposure while minimizes the overall path length. We also demonstrate how unifying the graph algorithms for indoor and outdoor navigation enables significant optimizations that would not be possible otherwise.

PVLDB Reference Format:
Constantinos Costa, Xiaoyu Ge, and Panos K. Chrysanthis. CAPRIO: Graph-based Integration of Indoor and Outdoor Data for Path Discovery. *PVLDB*, 12(12): 1878-1881, 2019.
DOI: https://doi.org/10.14778/3352063.3352089

1. INTRODUCTION

Recently, navigation and localization services have emerged to enhance the shortest path discovery in either indoor or outdoor. Particularly, with people spending the 90% of their time indoors [8], indoor navigation services optimized the shortest path search using magnetic fields for localization and a modified shortest path formulation [7]. An indoor environment has many elements with unique properties that are defining the indoor route [4]. 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. Many of these systems are incorporating machine learning techniques and augmented reality to provide new intuitive ways for path discovery [3].

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

Existing systems that consider indoor and outdoor information, use this information for localization and seamless transition from outdoors to indoors in order to improve navigation [5, 6]. These systems provide either an outdoor or an indoor path but not a combined one that considers the quality of the path. A unified model of indoor-outdoor space can be beneficial for the shortest path discovery [1]. For example, an outdoor path may not be an option due to extreme weather conditions (e.g., polar vortex [USA 2019 as seen in Figure 1 (right)], where low temperatures can cause frostbite very quickly to exposed skin) or natural disasters (e.g., a heatwave [Japan 2018 as seen in Figure 1 (left)], where high temperatures can cause hyperthermia). In these cases, the combination of indoor and outdoor information may produce a path that enables the usual activity of an individual.

In fact, we pose that the availability of services that combine indoor and outdoor information is *imperative* in situations where there are extreme weather conditions and natural disasters. *Minimizing the outdoor exposure* of a recommended path is vital to a pedestrian, since the aforementioned scenarios could be very dangerous for one's well being. In this demonstration 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 our proposed algorithm, coined *Graph Integrator and Path Discoverer* (GIPD), that integrates the external nodes (e.g., building) with the internal nodes (e.g., entrances/escalators/exits) to provide a path with minimum outdoor exposure and shortest distance. It achieves the integration in a manner that keeps the size of the graph to be searched for the shortest path to be no larger

Proceedings of the VLDB Endowment, Vol. 12, No. 12
ISSN 2150-8097.
DOI: https://doi.org/10.14778/3352063.3352089

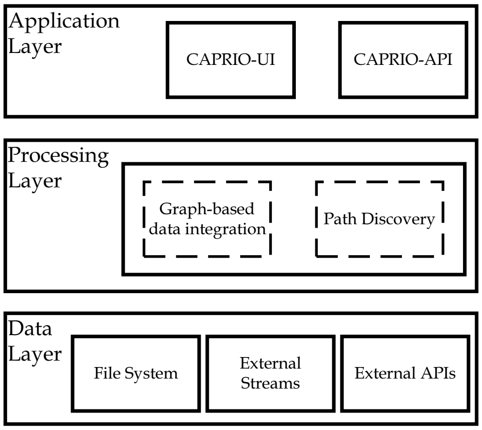


Figure 2: *CAPRIO* is an efficient graph-based data integration system that enables path discovery and targets to minimize the outdoor exposure o and the distance d.

than the external graph of the buildings. The *GIPD* algorithm allows CAPRIO system to provide a context-aware path in contrast with the traditional path from the existing well-known systems (e.g., Google Maps).

In summary, the contribution of this paper is the demonstration of the CAPRIO system and the effectiveness of its GIPD algorithm to combine navigation and localization services in order to personalize the outdoor exposure and minimize the distance of the path by using contextual information. In the next two sections, we will first overview CAPRIO and the *GIPD* algorithm (Section 2) and then discuss the demonstration specifics (Section 3).

2. CAPRIO SYSTEM

Our proposed CAPRIO system is structured in three layers (see Figure 2), namely *Data Layer*, *Processing Layer* and *Application Layer*.

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

The *Processing layer* has a main module, which implements the *GIPD* algorithm with two components, namely *graph-based integration* and *path discovery*. The graph-based integration component converts the data from different sources into external and internal nodes and ingests them into a single, unified graph. The graph discovery component uses the unified graph to produce a path that meets the outdoor exposure requirements specified by a user and has minimal distance. Indoor spaces have many attributes and constraints such as the distance and the accessibility of space [2].

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 applications over CAPRIO architecture.

To understand the operational aspects of our proposed CAPRIO system, consider Figure 3, where we show an instance of the system for one request/query. Table 1 summarizes our notations.

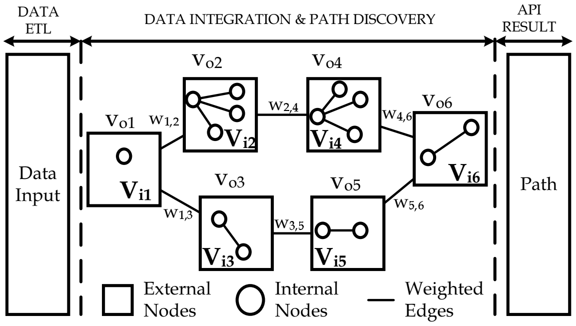


Figure 3: *CAPRIO* is a system that can provide a recommended path with the minimum outdoor exposure and distance for each request along with the respect source and final destination.

Table 1: 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 $j = 1, \ldots, m$
$G(V_o, E_o)$	outdoor/external graph
$G_i(V_i, E_i)$,	indoor/internal graph
o	outdoor exposure factor
d	distance
P	recommended path

Firstly, the CAPRIO is extracting, transforming and loading the data into the system. Next, the GIPD algorithm integrates the internal Vi and external Vo nodes using a unified graph $G_{IO}(V_i \cup V_o, E_i \cup E_o, W)$, enhanced with weights W, as shown in Figure 3. 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}. Finally, the algorithm produces a path between the source s and the final destination f using the well known Dijkstra algorithm on the projected graph $G_P(V_o, E_o, W)$, consisting of the external V_o nodes, edges E_o and W weights.

The crux of our GIPD 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.

2.1 Graph-Based Path Discovery Algorithm

In this section, we introduce the details of the *Graph Integrator and Path Discoverer* (GIPD) algorithm (Algorithm 1), which is where our main contributions lie.

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: Graph-Based Integration
 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: Path Discovery
10: $P \leftarrow DIJKSTRA(G, s, f)$
    ▷ Execute Dijkstra algorithm over the newly created graph
```

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 the state-of-the-art techniques for construction and integration of both the external and internal graphs.

In particular, CAPRIO relies on the Google Maps API to construct the external graph and provide the routing between buildings, while CAPRIO relies on the state-of-the-art indoor navigation system, Anyplace [8], to construct the internal graph and provide the routing the internal paths. To combine the external graph with the internal graph of each building, CAPRIO is using both the street distance reported by the Google Maps API and the internal travel distance of each building computed by the Anyplace in order to produce the integrated weights of each external edge.

The weight of this 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 $DT_i(v_{o_j}, v_{o_k})$ 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}; $DT_o(v_{o_j}, v_{o_k})$ is the external travel distance between v_{o_j} and v_{o_k}; and 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. By means of the outdoor exposure factor, users can customize their outdoor exposure based on their given circumstances.

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 considering the travel distance within each internal graph (i.e., nodes). Once the weight for each edge has been assigned, traditional graph techniques (e.g., Dijkstra) will be applied on the external graph that was resulted from the previous step to obtain the path.

Particularly, 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 being calculated using the $WEIGHT$ (Eq. 1) based on the outdoor exposure o. Then the $EDGE$ function creates a new edge with the weight w and adds it to the graph G. Finally, the shortest path discovery step (Step 3 - line 10) generates the path using Dijkstra algorithm that produces a path with minimum path weights.

3. DEMONSTRATION SCENARIO

During the demonstration, the attendees will be able to comprehend the key concepts of CAPRIO, the visualization abstraction, as well as the performance of our propositions by interacting with a user-friendly interface. Below we will present more details on the implementation of CAPRIO, and then discuss our demo plan.

3.1 Demo Artifact

We have implemented a prototype of CAPRIO using an interactive map, 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. Particularly, 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; (ii) the graph tab that animates the path using a graph visualization to provide visually the algorithms and the techniques behind the paths illustrated in Figure 4; and (iii) the settings tab that activates/deactivates elements on the main interface.

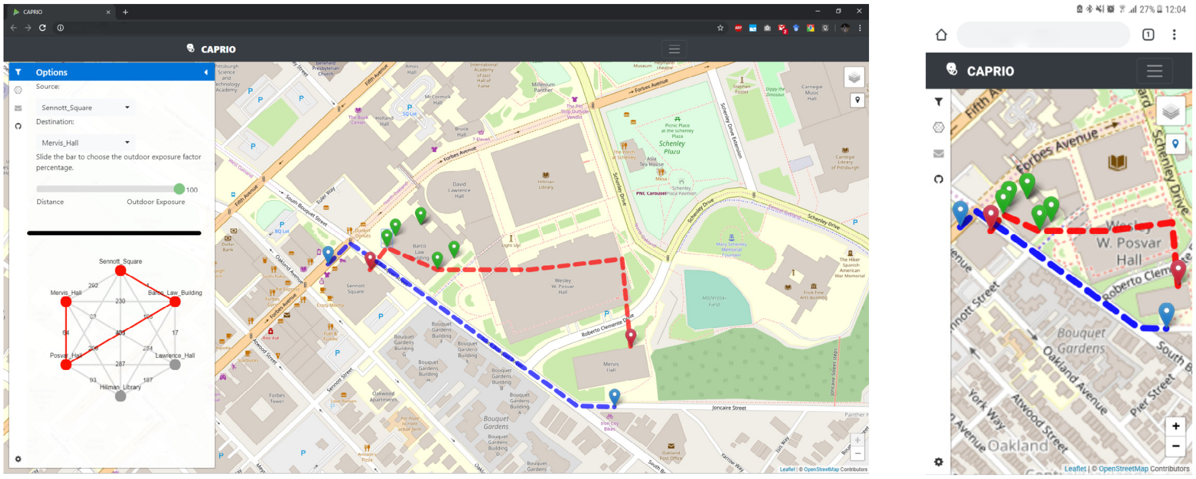
The hardware stack of our CAPRIO installation resides on a dedicated server. The server is featuring 12GB of RAM with 4 Cores (@ 2.90GHz). During the demonstration, we will connect over cable or Wi-Fi to the CAPRIO web service and enable the users to interact with our intuitive web interface, as described next. We shall also have video recordings at hand, in case the network is unstable at the conference.

3.2 Demo Plan

Equipment: The conference attendees will have the opportunity to interactively engage with the CAPRIO GUI using a standard laptop, a tablet and smartphones we will bring along at the conference.

---

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



Source:

Sennott_Square

Destination:

Mervis_Hall

Slide the bar to choose the outdoor exposure factor percentage.

100

Figure 4: (left) 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 (blue line). *GIPD* algorithm can be visualized as an animated graph that shows the resulted graph along with the path comparison on top of Google Maps. (right) The user interface follows the responsive design and it can be used as a mobile application for path navigation

Datasets: We will pre-load a variety of real datasets to the CAPRIO back-end. The loaded data exposes the graph-based data integration and will be very useful to visually show how the CAPRIO path recommendation engine works in real time over real data.

Scenarios: Particularly, CAPRIO server will be publicly available to allow attendees to change the parameters of the system and to see the result in real time on the interface along with the animated graph that we discussed in Section 2.

In order to present the benefits of our propositions to the attendees, we will provide visual cues that will enable the audience to understand the performance benefits (i.e., outdoor exposure and distance).

4. CONCLUSIONS

In this demo 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 the outdoor and indoor information. The goal of the system is to extract a path that satisfies the distance and the outdoor exposure requirements according to the user's preference.

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

5. ACKNOWLEDGMENTS

We thank Jonathan Michael Albert and Chihao Sun, members of our group at the University of Pittsburgh for the assistance with the gathering of the real-world data that is used as part of this demo.

6. REFERENCES

[1] S. K. Jensen, J. T. V. Nielsen, H. Lu, and M. A. Cheema. Outdoor-indoor space: Unified modeling and shortest path search. In *Proc. of the Eighth ACM Intl Workshop on Indoor Spatial Awareness*, pages 35–42, 2016.

[2] H. Lu, X. Cao, and C. S. Jensen. A foundation for efficient indoor distance-aware query processing. In *IEEE 28th Intl Conference on Data Engineering*, pages 438–449, 2012.

[3] F. Mata and C. Claramunt. A social navigation guide using augmented reality. In *Proc. of the 22Nd ACM Intl Conference on Advances in Geographic Information Systems*, pages 541–544, 2014.

[4] C. Salgado, M. A. Cheema, and D. Taniar. An efficient approximation algorithm for multi-criteria indoor route planning queries. In *Proc. of the 26th ACM Intl Conference on Advances in Geographic Information Systems*, pages 448–451, 2018.

[5] 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.*, 13(2):12:1–12:28, 2017.

[6] A. Vanclooster and P. De Maeyer. *Combining Indoor and Outdoor Navigation: The Current Approach of Route Planners*, pages 283–303. Springer Berlin Heidelberg, 2012.

[7] H. Wu, S. He, and S.-H. G. Chan. Efficient sequence matching and path construction for geomagnetic indoor localization. In *Proc. of the 2017 Intl Conference on Embedded Wireless Systems and Networks*, pages 156–167, 2017.

[8] D. Zeinalipour-Yazti, C. Laoudias, K. Georgiou, and G. Chatzimilioudis. Internet-based indoor navigation services. *IEEE Internet Computing*, 21(4):54–63, 2017.