2021 22nd IEEE International Conference on Mobile Data Management (MDM) | 978-1-6654-2845-3/20/$31.00 ©2021 IEEE | DOI: 10.1109/MDM52706.2021.00050

HealthDist: A Context, Location and Preference-Aware System for Safe Navigation

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

Sayantani Bhattacharjee
Dept. of Computer Science
University of Pittsburgh
sab301@pitt.edu

Benjamin Graybill
Dept. of Computer Science
University of Pittsburgh
bcg36@pitt.edu

Constantinos Costa
Dept. of Computer Science
University of Pittsburgh
costa.c@cs.pitt.edu

Sudhir Pathak
Learning Research & Development Center
University of Pittsburgh
skpathak@pitt.edu

Walter Schneider
Learning Research & Development Center
University of Pittsburgh
wws@pitt.edu

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

***Abstract*—In this demo paper, we feature *HealthDist*, an innovative system that is an additional asset in the fight against the COVID-19 pandemic. *HealthDist* utilizes context (e.g., weather conditions), location (e.g., crowded areas), and user preferences to provide safe pedestrian paths which decrease the exposure to the virus causing COVID-19. Its modular design, consisting of four components, reduces the time and resources needed to provide accurate localization and indoor-outdoor path recommendations that satisfy the user's preferences. We demonstrate interactively using smartphones how *HealthDist* can provide real time navigation information within a university campus and illustrate the reduction of the COVID-19 exposure risk while satisfying the constraints defined by the user.**

Video: http://bit.ly/3bMicbs

***Index Terms*—indoor, outdoor, navigation, path recommendation, graph processing, disability, congestion forecasting.**

I. Introduction

COVID-19 is an airborne disease, which is highly contagious with the larger percentage of infected people not exhibiting symptoms. To reduce its spread, governments around the world have promoted the use of face masks, social distancing, contact tracing and isolation [1]. In this paper, we are demonstrating our innovative system, dubbed *HealthDist*, developed as part of the fight against the COVID-19 pandemic to promote these public health policies.

HealthDist is designed as part of the *CovidReduce* project (CovidReduce.org) to implement a holistic approach of *proactive* (contact avoidance) and *retroactive* (contact tracing) functionalities without violating privacy (shown in Figure 1). It utilizes the context (e.g., weather conditions), location (e.g., crowded areas) and user's preferences to recommend *safe* pedestrian paths that go through less congested hallways and corridors, hence decreasing the exposure to the virus causing COVID-19 and reducing the risk of severe illness. Furhermore, *HealthDist* is designed to serve as a highly accurate proximity detector, providing information about the distance of nearby individuals, work as a radar detector, providing information about potential encounters with moving, even out of view, individuals in their vicinity (path/trajectory prediction), and operate as an infection exposure meter, counting possible infection Quanta (i.e., infection dose). All these three services require accurate localization.

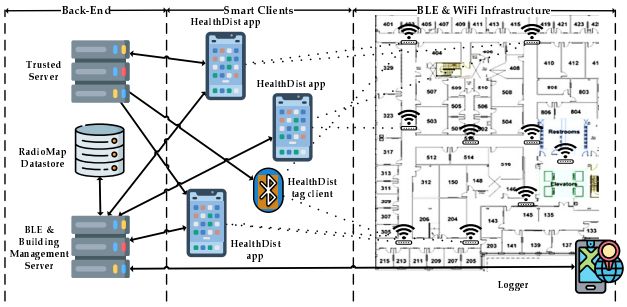


Fig. 1: The HealthDist system consists of three components: i) HealthDist Back-end, ii) HealthDist Smart Clients, and iii) HealthDist BLE Infrastructure

The current version of *HealthDist* fully implements the safe path recommendation and the Quanta meter, allowing users to see and reduce their daily risk status, when moving within a university campus. More details on HealthDist are described in the extended paper [2].

II. HealthDist Back-End

The *HealthDist* system builds upon our CAPRIO architecture [3] and consists of the *Data Layer*, *Processing Layer*, and *Application Layer*.

The Data Layer is responsible for managing input data from various data sources such as local or distributed files, data streams, or other external APIs.

The Processing Layer is responsible for the core functionalities and services of the *HealthDist* system such as processing data for path recommendations, performing localization based on BLE devices, and providing congestion information. This layer comprises five modules: (i) the *Building Management*

978-1-6654-2845-3/21/$31.00 ©2021 IEEE
DOI 10.1109/MDM52706.2021.00050

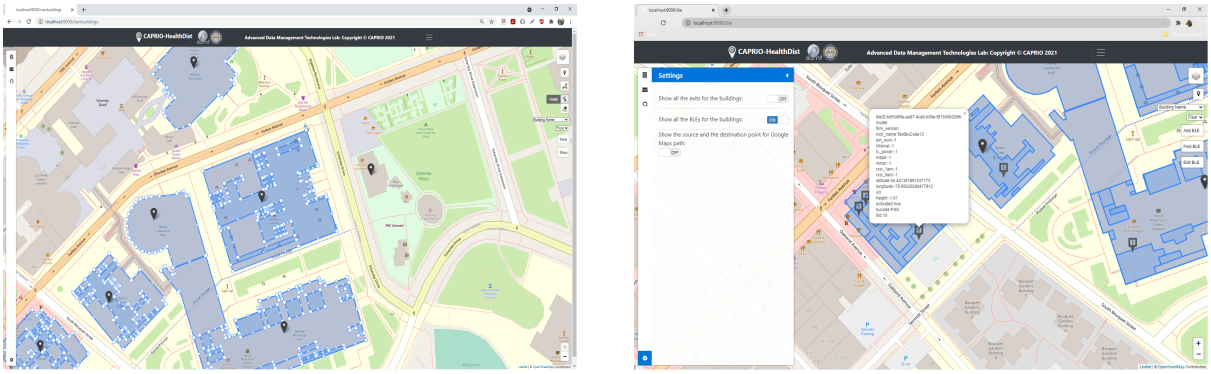


Fig. 2: (left) The *Building Management* (BM) allows the easy and simple management of buildings, and (right) the *BLE Management* (BLEM) supports efficient management of the BLE devices using an intuitive map-based interface.

(BM); (ii) the *BLE Management* (BLEM); (iii) *Radiomap Generation* (RG); (iv) *Localization* (RGL); and (v) the *Navigation Assistant* component. The *HealthDist* system preserves privacy by using unique, anonymized, and encrypted IDs (based on the Apple/Android SDK).

The Application Layer acts as the user interface for *HealthDist* as well as to the BML and BLEM Processing Layer modules, allowing for easy development with an open API. This layer utilizes a Leaflet JS map library to abstract the complexity of the system from the user.

III. Implementation

A. Building Management

The Building Management (BM) allows the user to create and use building geometries in order to improve localization. Its purpose is to avoid any discrepancies that exist in building outlines in map tiles as well as respective floor plans, and provide a higher level of accuracy during outdoor and indoor path recommendation. The BM helps create the outer and inner geometries of the buildings (e.g., corridors, exits, entrances) using interactive map-based interface and stores them in the form of GeoJSON objects in a NoSQL database (i.e., MongoDB) with the help of Scala and Play Framework.

The system allows the users to display, modify or delete the building geometries (i.e., floor plans) shown in Figure 2 (left). The geometric data is stored in a lightweight geographic JSON format, called GeoJSON, which supports all types of geometric shapes (e.g., Point, Multipoint, Line, Polyline, MultiPolyline, Polygon and MultiPolygon, efficiently). This format paired with MongoDB, facilitates 2dsphere indexing which is extremely helpful for localization. MongoDB enables efficient processing with semi-structured data and in turn increases the flexibility of the system while handling pre-existing geometries. Scala combines object-oriented and functional programming concepts and allows the execution of Java code. The Mongo module (play-mongo) in the Play framework, provides NoSQL support and permits the use of Scala and Java programming. As such, BM is an interactive, highly-scalable and flexible system that allows multi-language development.

B. BLE Management

The BLE Management (BLEM) allows users to create, modify, and use Bluetooth Low Energy devices (BLEs) in order to improve localization and indoor path recommendation. Due to the inability to utilize GPS for indoor localization, BLEs are the primary means of localization for indoor path recommendation. When these devices are strategically placed inside a building, higher level of accuracy can be achieved during localization and indoor path recommendation.

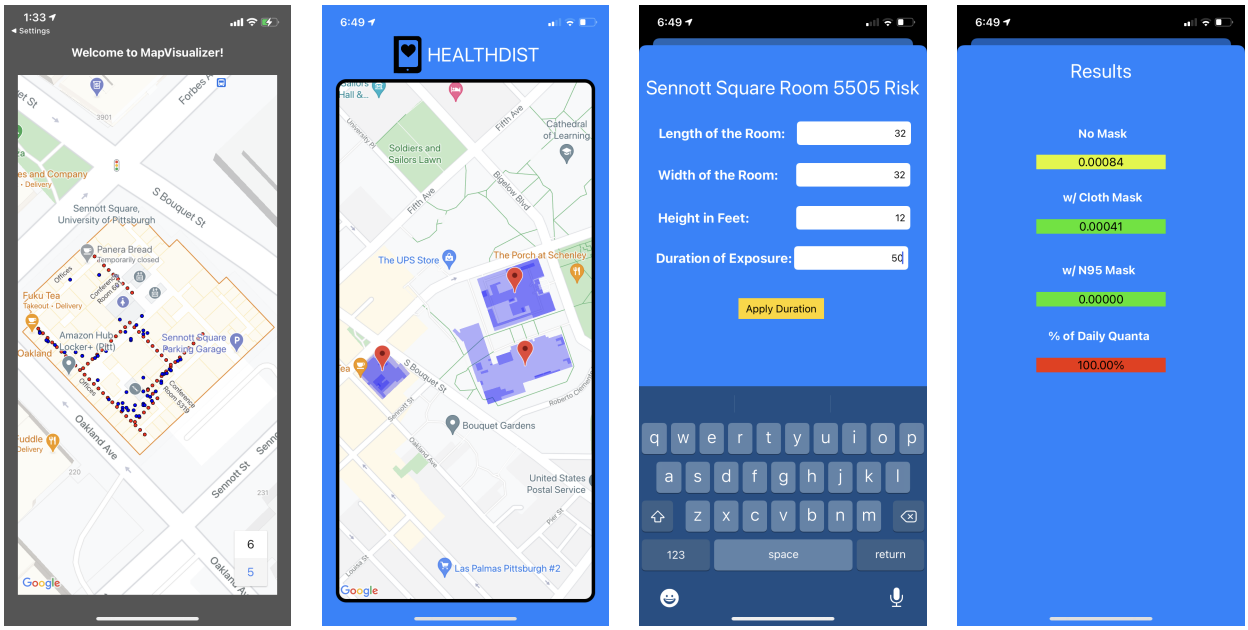
The BLEM allows the creation, modification, and displaying of BLEs using an interactive map-based interface shown in Figure 2 (right) and stores BLEs in the form of GeoJSON objects in MongoDB similarly to the BM, discussed above in Section III-A. The use of the GeoJSON allows all BLEs to follow a standard format as a Point for easy and efficient retrieval from MongoDB. Utilizing the 2dsphere indexing on MongoDB allows for efficient retrieval and modification of BLEs as each device has a unique latitude and longitude for each building and floor.

It is easy for a user to retrieve all information (e.g., ID, location) for a given BLE device by clicking the BLE's icon as shown in Figure 2 (right). In this way, a user can locate the BLE device they want to modify, retrieve the BLE's ID, and modify the contents of the BLE device by passing the UUID of the BLE device.

C. Localization

The Localization module is implemented as the *LocationAccuracy* library to support navigation, proximity and radar detection on smartphones. It is also used for improving congestion forecasting. The localization library incorporates modified versions of Anyplace algorithms that rely on the Radio Maps, which are generated based on fingerprinting [4].

The four Anyplace algorithms are described in the extended *HealthDist* paper [2] and consist of: *Unweighted K-NN, Weighted K-NN, Probabilistic Minimum Mean Square Error*, and *Weighted Probabilistic Minimum Mean Square Error*.



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

Fig. 3: (a) Visualizing the localization using the Unweighted Probabilistic Minimum Mean Square Error; (b) HealthDist iOS application for several University of Pittsburgh campus buildings; (c) User interface for entering information for Sennott Square room 5505; (d) Calculated risk of exposure using the parameters provided in the previous image (c).

These algorithms were translated and adapted to be used in an iOS application and are written in a Swift static library that can be imported into any iOS application bundle.

For testing and demonstration, we created an app that allows us to use the radiomap files to show correlation as to where the logged latitude and longitude points are in red and the predicted locations are in blue as seen in Figure 3 (a). The app also logs to the console the average accuracy for a given set of points based on the difference from the actual/logged position and the predicted value from the algorithm.

D. Navigation Assistant

The Navigation Assistant module is responsible for providing safe indoor and outdoor path recommendations in order to better support *HealthDist*'s contact avoidance objective. This module is able to provide safe path recommendations through the use of deep-learning based congestion forecasting and an A* based algorithm, recently added to our CAPRIO system. The deep-learning based congestion forecasting uses a spatial index coined CM-Structure which allows for the efficient retrieval of trained long short-term memory (LSTM) models for each indoor corridor. The CM-Structure maps each corridor inside a building to a set of LSTM forecasting models. These models are trained on generated indoor traffic patterns and provide predictions of a corridor's congestion at a specific time with high accuracy. The A* based algorithm, called *ASTRO*, uses the congestion predictions as one of the weights while searching for the best path to recommend. That is, a path that contains higher congestion will have a higher cost than a path that goes through less congested routes. By factoring predicted congestion in as a cost to ASTRO's path finding algorithm, paths with lower indoor congestion are recommended supporting *HealthDist*'s contact avoidance objective.

IV. HealthDist Smart Client

The HealthDist iOS app is a multiview application of the *HeathDist*'s path recommendation and Quanta counting services. It is programmed using Apple's developer tools and XCode swift coding environment. The app uses that data from the BM in the GeoJSON format to display the buildings on a map generated by Google Maps for the iOS API.

The user is able to select a building by clicking on a marker shown in Figure 3 (b) and open a new prompt for information about the selected building and the user. This information includes basic characteristics of the space where the user is located (e.g., size and ventilation), as shown in Figure 3 (c), the congestion and the duration of exposure for the selected location, and whether the user wears no mask, a cloth mask, and a N95 mask. Once the necessary information is provided, the application calculates and displays the risks of COVID-19 exposure in Quanta, as shown in Figure 3 (d).

The user can change their given parameters or select a new location from the original map view. Details about these parameters can be found in the web-based Quanta calculator available at CovidReduced.org.

V. HealthDist BLE Infrastructure

The BLE infrastructure of HealthDist consists of 500 meters long range BLE devices placed 20 feet apart inside of buildings. By placing a sufficient number of BLE devices in building corridors HealthDist is able to obtain a more accurate Radio Map which are used in HealthDist's localization as discussed in Section III-C.

A. RadioMap Generation

1) Collecting the Data: Radio maps are generated by aggregating data collected from samples taken along a path of latitude and longitude coordinates. To streamline the process of collecting data for the generation of these radio maps a logging app was built for both Android and iOS platforms which allows users to select paths and incrementally take signal samples at points along the path chosen.

The user starts by clicking the "Start route selection" button and then chooses a starting and ending point. The app then creates a path of red location dots every two meters in a straight path between the chosen starting and ending points. This becomes the position of where the data collector will be sampling for local signals. These signals come from BLE devices placed every twenty feet around the building. The purpose of this process is to eliminate GPS from trying to detect where the collector is during sampling. By having the user select the points and creating intermediaries we are relying on the human selection of the coordinates and there is no concern of GPS inaccuracy.

When the collector (i.e., the user using the logger) goes to the starting point, and clicks the "Start/Restart Sampling" button, the traversal of the chosen path begins. At each point on the path, the collector sets how many samples to collect and hits "Take next samples" for the app to begin monitoring for what is called a beacon region which is defined by a UUID. Each second the logger gathers all the beacons that are broadcasting and logs the UUID, major, minor, bearing, and RSSI values to a file in our app data (separate files for each latitude and longitude coordinate along the path). Once the specified number of samples is taken the collector will repeat the process for each subsequent point on the path.

2) Aggregrating the Data: The data for each path is collected and then is placed into directories that correlate to what location they were taken from. This often is data collected over multiple passes of the same path. In order to get an accurate value for each point the data is looked over and cleaned for any readings that are blatantly incorrect and then the RSSI values for each BLE present in the samples are averaged to produce a final RSSI values for each present BLE at that point. Once this is done we have a coordinate that has a single value for each BLE that it could connect to at that location. We do this for each point along the scanned path and that gives us one data point every two meters. These singular paths are then put together to form a Radio Map for the entire floor of a building which can be fed to our localization algorithms for location prediction.

VI. Demonstration Scenario

During the demonstration, the attendees will be able to comprehend the key concepts of *HealthDist*, the visualization abstraction, as well as the performance of our propositions by interacting with a user-friendly interface.

A. Demo Artifact

We have extended our prototype of *CAPRIO*, which incorporates an interactive map and integrating several graph techniques in the back-end, which was developed using Play Framework 2.7 and MongoDB 4.4. The *HealthDist* web interface is implemented in HTML5/CSS3 along with extensive usage of Leaflet and jQuery.

We have implemented a query sidebar that allows the user to switch between three main views: (i) The path recommendation interface, which enables the user to choose the source, the destination and the accessibility of the recommended path along with its outdoor exposure/distance and the congestion preference; (ii) The BM interface, which allows the user to edit, delete and add the geometry of any building using multiple layer interactively, shown in Figure 2 (left); and (iii) The BLEM interface, which supports the management of the BLE devices and uses an intuitive map-based interface that allows the user to place BLE devices inside a building, shown in Figure 2 (right). The HealthDist open API provides the user with necessary information to calculate a COVID-19 risk exposure score, shown in Figure 3 (d).

B. Demo Plan

Equipment: The conference attendees will have the opportunity to interactively engage with the *HealthDist* GUI using a standard laptop, a tablet and smartphones.

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

Scenarios: *HealthDist* server will be available to allow attendees to change the parameters of the system and to see the result in real time on the interface. 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, distance, congestion and COVID-19 risk exposure score).

References

[1] N. Ahmed, R. A. Michelin, W. Xue, S. Ruj, R. Malaney, S. S. Kanhere, A. Seneviratne, W. Hu, H. Janicke, and S. K. Jha, "A survey of covid-19 contact tracing apps," *IEEE Access*, vol. 8, pp. 134 577–134 601, 2020.

[2] C. Costa, B. T. Nixon, S. Bhattacharjee, B. Graybill, D. Zeinalipour-Yazti, W. Schneider, and P. K. Chrysanthis, "A context, location and preference-aware system for safe navigation," in *IEEE MDM*, pp. 1–8, 2021.

[3] 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 MDM*, pp. 230–231, 2020.

[4] D. Zeinalipour-Yazti, C. Laoudias, K. Georgiou, and G. Chatzimilioudis, "Internet-based indoor navigation services," *IEEE Internet Comput.*, vol. 21, no. 4, pp. 54–63, 2017.