Demo: Towards Flexible and Scalable Indoor Navigation

Zhuqi Li
Peking University
lizhuqi@pku.edu.cn

Yuanchao Shu
Microsoft Research
yuanchao.shu@microsoft.com

Börje F. Karlsson
Microsoft Research
borje.karlsson@microsoft.com

Yiyong Lin
Microsoft Research
yiyolin@microsoft.com

Thomas Moscibroda
Microsoft Research
moscitho@microsoft.com

ABSTRACT

Bootstrapping efforts and scalability issues hinder large-scale deployment of indoor navigation systems. We present FOLLOWUS, an easily-deployable (bootstrap-free) and scalable indoor navigation system. In addition to robust navigation through real-time trace-following, FOLLOWUS integrates cloud services to process and combine traces at large scale. It can also leverage optional floor plans to further enhance navigation performance. We designed and implemented FOLLOWUS, including a mobile app and cloud services on Azure, and validate its real-world usability.

KEYWORDS

Navigation; indoor; cloud; app; trace; map; scalability; flexibility

1 INTRODUCTION

Despite years of research attention, localization-based indoor navigation has not found wide-spread practical use, largely due to the high burden on deployment and bootstrapping. They are usually built upon full-fledge indoor localization systems, which rely on specialized infrastructure (e.g. beacons) or incur high bootstrap costs. On the contrary, recently proposed lightweight peer-to-peer navigation systems promise easy and incremental deployment, without the need for extra infrastructure or maps. Despite these promising advantages, peer-to-peer navigation suffers from poor scalability and flexibility as navigation is only possible over pre-collected leader paths.

In this demonstration, we present FOLLOWUS, a new end-to-end indoor navigation system that combines the advantages of both localization-based and peer-to-peer navigation with increased flexibility and scalability. As in typical peer-to-peer navigation, users can walk along a path, share it with others and follow paths shared by others via a mobile client. During the navigation, FOLLOWUS can provide timely instructions to help user walk from starting point to destination by synchronizing magnetic sequence between leader trace and follower trace. FOLLOWUS also incorporates an intelligent cloud service to the lightweight navigation application to parse and combine ("stitch together") traces at scale. Based on the analysis of motion events and the matching results between sensor signals, FOLLOWUS is able to split, concatenate, and generate new traces according to users' searching input. In addition, FOLLOWUS can incrementally build the indoor trace graph and improve its quality with optional floor plan. In a nutshell, FOLLOWUS can scale both horizontally — reaching as many users as possible without requiring much effort of them — as well as vertically by efficiently computing new paths from traces shared by users.

MobiCom'17, October 16–20, 2017, Snowbird, UT, USA.

ISBN 978-1-4503-4916-1/17/10…$15.00
DOI: http://dx.doi.org/10.1145/3117811.3119854

2 OVERVIEW

It is important to highlight that FOLLOWUS is completely plug-and-play, meaning it imposes no infrastructure requirements on buildings, no bootstrapping effort such as creating Wi-Fi fingerprint databases, and it requires minimal effort from users. This allows the system to scale out to reach as many potential users and to work in as many locations as possible. FOLLOWUS traces *require* only readings from smartphones' magnetometer, accelerometer, gyroscope, and barometer sensors. Nevertheless, other signals (e.g. Wi-Fi RSSI, cellular signals) can easily be integrated into the system.

2.1 Usage scenarios

FOLLOWUS allows users to record and follow paths using its mobile client app or conduct *app-less navigation* with web-based offline instructions.

Recording (as "leader"). During recording, the FOLLOWUS client queries sensors and detects user motion events including steps, turns, and level changes (i.e., staircases, elevators, or escalators). When the leader arrives at the destination, sensor measurements and detection results are packed to build a reference trace and then sent to the cloud to be shared publicly or with specific users.

Navigation (as "follower"). Users can follow traces over any pre-generated path. Traces can either be pushed to the client app Inbox or pulled via Search functionality (e.g., from East Entrance to Room 1357). When following a trace, real-time navigation instructions including turns and floor/level changes are displayed in the app based on online synchronization between the reference trace and current ambient sensing information. Besides real-time navigation modes, FOLLOWUS also provides a web interface featuring static *app-less navigation* — users can view trace summaries online in a browser and perform self-navigation without app installation on the client.

2.2 Architecture

Figure 1 shows the system architecture and its main modules: Recording Manager and Navigation Manager (client-side), and Service API, Trace Processing Pipeline (TPP), and Path Builder (cloud-side).

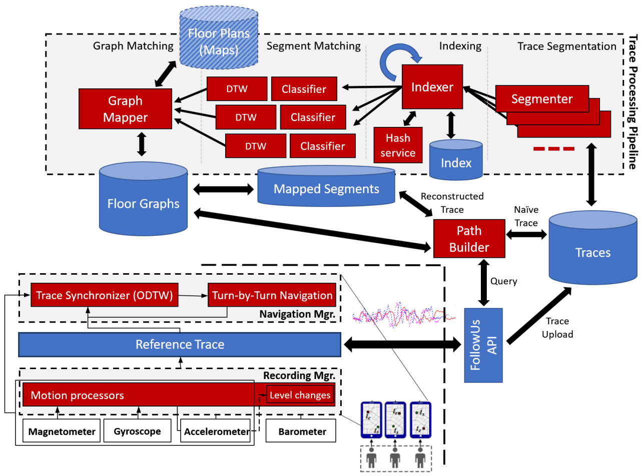


Figure 1: System diagram of FOLLOWUS.

Recording Manager fuels the whole system by providing trace data recorded from mobile devices (e.g., smartphones) during a users' walk. This module collects and processes IMU data and location-specific geomagnetic signals [1–3] to extract user motion patterns and construct a trace. The output of this module can be directly used by the Navigation Manager to provide real-time plug-and-play navigation service. The processing results yielded by this module are also pushed to the cloud through the FOLLOWUS Service API for storage and further processing by TPP, ensuring the scalability of the system.

Navigation Manager takes a reference trace as an input and navigates followers from the starting point to its selected destination from the database. A reference trace can be generated by either Recording Manager (local to the device) or Path Builder (queried from the backend service). During navigation, the module provide step-by-step navigation instructions, which is based on the synchronization result between geomagnetism readings from the reference trace and real-time measurements from the follower's Recording Manager[1]. The online dynamic time warping [2, 4] (ODTW) synchronization algorithm is designed to provide accurate navigation progress estimation with enhanced adaptability (capable to mitigate the effect of different walking speed) and efficiency (linear time complexity). To enhance navigation robustness, this module also contains a "deviation detection" component and the corresponding recovery mechanisms.

Service API handles trace and (media) annotation upload/download, data storage, user authentication, user notifications, and metadata search.

Trace Processing Pipeline ingests newly uploaded traces and is responsible for segmenting traces, identifying similarities between segments, and building logical graphs of building floor plans. Specifically, TPP will first split the incoming traces into trace segments by navigation events. Then trace segments will be organized by a high-efficient indexing structure to find the mutual matching candidates. Finally TPP utilizes the connectivity information and matching result between trace segments to build indoor trace graphs. Such graphs and mapped segments can later be used to produce new navigation paths. The pipeline design allows FOLLOWUS to leverage cloud capabilities to dynamically scale processing power as more data arrives. If indoor floor-plans are available, this module can automatically find best match paths and improve graph building efficiency.

Path Builder receives user queries for paths between given start and end points. If a path is readily available, it is immediately returned. Otherwise, PB will find an optimal path by searching the floor plan graphs from TPP and combines corresponding segments to create a new trace for download.

2.3 Data flow

As shown in Figure 1, data flows from client (as a leader) to cloud, and then from cloud back to client (as a follower) in FOLLOWUS. The cloud component not only provides storage, communication interfaces, telemetry, and account management, but also features an intelligent trace processing pipeline that makes FOLLOWUS flexible and scalable. In "naïve" navigation, where one can only follow the exact same paths from others, each path recording produces and uploads a trace. To enable "cross-trace" navigation, FOLLOWUS employs a novel *trace segmentation and indexing* technique. Segments are then efficiently matched, indexed, and persisted to allow the construction of *segment graphs*. A segment graph is incrementally built from trace segments and can eventually provide users with an exponentially increasing number of paths. Additionally, floor plans can naturally be leveraged to improve both performance and efficiency of the graph generation process.

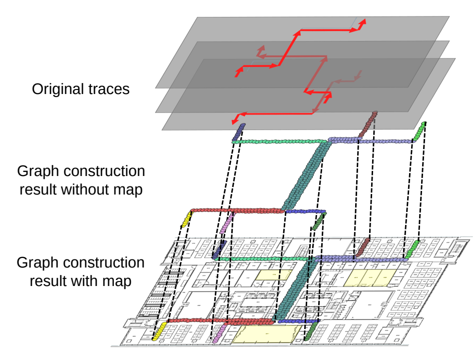


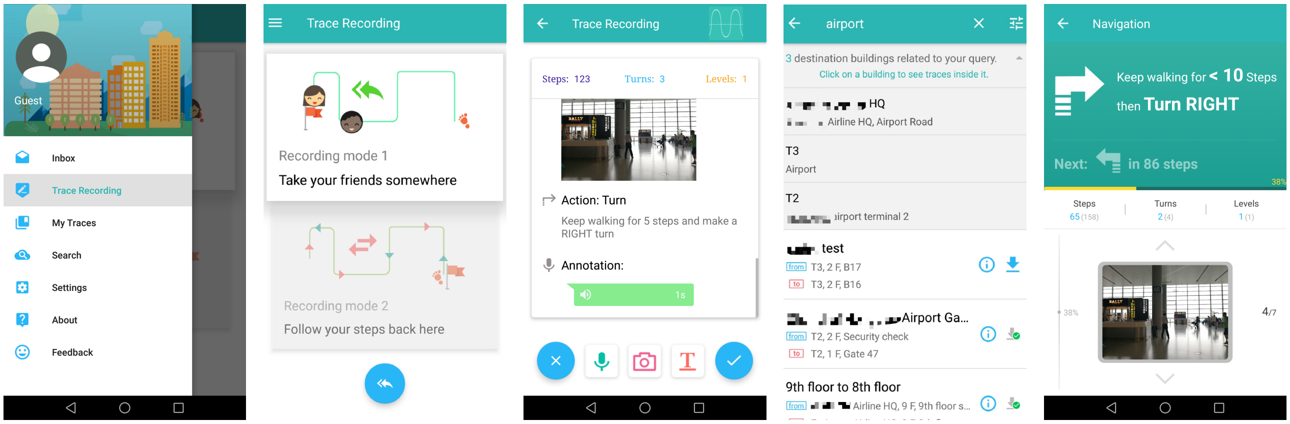
Figure 2: Graph construction visualization.

3 IMPLEMENTATION

We implemented FOLLOWUS as an Android app and a set of backend cloud services on Azure[2]. The entire system consists of $\approx$ 56k lines of code (loc), broken into: Mobile Core 8.9kloc; FOLLOWUS App 29kloc; Trace Processing Pipeline 13.6kloc; Path Builder 1.7kloc, and FOLLOWUS Service API 2.5kloc. Mobile Core encompasses platform-independent algorithms and functions to be re-used by other applications and services. The FOLLOWUS app makes use of the core modules to accomplish trace collection and real-time navigation tasks. Basic implementation of data collection, motion

[1]Note that other location-specific signals like Wi-Fi RSSI can also be naturally leveraged.

[2]A public version is available to download at https://aka.ms/mspathguide [5].



(a) Side menu (b) Two recording modes (c) Trace recording (d) Trace search (e) Real-time navigation

Figure 3: App screenshots.

detection, and trace synchronization in FOLLOWUS is generally similar to [2, 6], whereas details and improvements are omitted due to space limitations. Trace Processing Pipeline and Path Builder are implemented to split, match, concatenate, and generate traces at scale. Beyond the core modules, the app UI and user experience has been optimized throughout the development cycle and feedback from testing. Figure 3 shows app snapshots, and Figure 4 shows a web-based offline instructions summary that can be used if a user does not have the app on her phone.

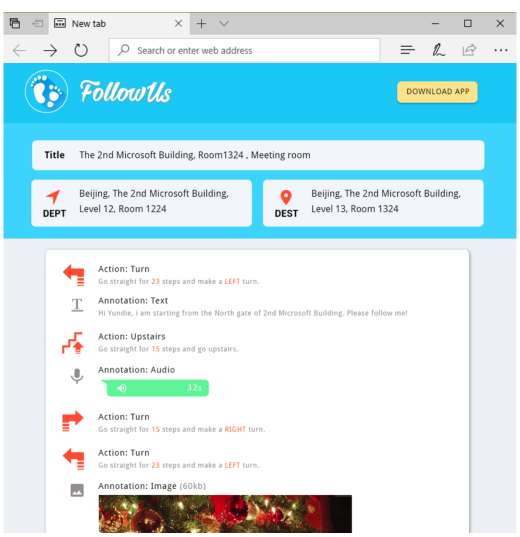


Figure 4: Offline (app-less) navigation.

4 DEMONSTRATION

In this demonstration, we will show the app and its capability of navigation. The demonstration will cover the three different modes under which FOLLOWUS can be used.

Ad-hoc navigation. In ad-hoc navigation demonstration, we will record an indoor trace with one smartphone and share it to another smartphone. Then we will use the second smartphone to follow the shared trace from the starting point to the destination. This mode can help subsequent visitors find a meeting room regardless of whether infrastructure and floor plans are available.

One-to-many navigation. In one-to-many navigation demonstration, we will record one indoor trace and share it to the public. Users can use the search function to find the trace and perform real-time navigation. This mode can help customers find desired stores in a shopping mall in an efficient way.

Reverse navigation. In reverse navigation demonstration, we will use a smartphone to record a trace from the starting point to the destination, and then navigate user back to the starting point. This mode can be used to record a trace from a parking spot to the elevator, and later follow it in reverse to find the car.

The demo only requires three Android smartphones (higher than 5.0). In ad-hoc navigation and one-to-many navigation mode, smartphones should be connected to Internet to allow trace uploading and downloading. In reverse navigation mode, no Internet access is required. The demo only needs few minutes to setup, including app installation and sensor calibration. This demo sets minimal requirements on the test sites. It can be demonstrated in any indoor environment.

REFERENCES

[1] Yuanchao Shu, Cheng Bo, Guobin Shen, Chunshui Zhao, Liqun Li, and Feng Zhao. Magicol: Indoor Localization Using Pervasive Magnetic Field and Opportunistic WiFi Sensing. *IEEE Journal on Selected Areas in Communications*, 33(7):1443–1457, 2015.
[2] Yuanchao Shu, Kang G Shin, Tian He, and Jiming Chen. Last-Mile Navigation Using Smartphones. In *ACM MobiCom*, 2015.
[3] Jaewoo Chung, Matt Donahoe, Chris Schmandt, Ig-Jae Kim, Pedram Razavai, and Micaela Wiseman. Indoor location sensing using geo-magnetism. In *ACM MobiSys*, 2011.
[4] Meinard Müller. Dynamic time warping. *Information retrieval for music and motion*, 2007.
[5] Microsoft. Microsoft Path Guide. https://aka.ms/mspathguide.
[6] Yuanqing Zheng, Guobin Shen, Liqun Li, Chunshui Zhao, Mo Li, and Feng Zhao. Travi-navi: Self-deployable Indoor Navigation System. In *ACM MobiCom*, 2014.