# Indoor Trajectory Identification: Snapping with Uncertainty

**Ravi Shroff** and **Yilong Zha**
Center for Urban Science and Progress
New York University
1 Metrotech Center, 19th floor, Brooklyn, NY, 11201

**Richard Wang** and **Manuela Veloso** and **Srinivasan Seshan**
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA, 15213

## Abstract

We consider the problem of indoor human trajectory identification using odometry data from smartphone sensors. Given a segmented trajectory, a simplified map of the environment, and a set of error thresholds, we implement a map-matching algorithm in a urban setting and analyze the accuracy of the resulting path. We also discuss aggregation of user step data into a segmented trajectory. Besides providing an interesting application of learning human motion in a constrained environment, we examine how the uncertainty of the snapped trajectory varies with path length. We demonstrate that as new segments are added to a path, the number of possibilities for earlier segments decreases monotonically. Applications of this work in an urban setting are discussed, as well as future plans to develop a formal theory of odometry-based map-matching.

## Introduction

In this paper we examine the identification of walking trajectories of people equipped with mobile phone-based odometry sensors. Specifically, we build on prior work in (Wang, Veloso, and Seshan 2013; 2014) and implement a "snapping" algorithm to reconstruct human paths traversed in a real indoor environment, given an existing map of that environment. This algorithm searches for all plausible paths within specified error bounds using the map and a segmented trajectory derived from accelerometer and gyroscope measurements.

There are three major modalities for indoor human path identification; wifi, odometry, and vision-based. Vision-based systems rely on fixed infrastructure (cameras), odometry-based systems rely on mobile sensors, and wifi-based systems require both fixed wifi access points in conjunction with a wifi-enabled mobile device. Odometry offers several advantages over vision-based systems. First, it is easy to anonymize since images of people are not stored. Second, there is no confusion between individuals since data is tied to a person's phone. Third, there is no requirement for the installation of fixed hardware (this is also an advantage over wifi based approaches), making scaling more cost-efficient. We emphasize that the odometry-based approach does not use GPS, and in fact uses no data apart from smartphone sensor measurements and an underlying "topological map" of the space.

Our technique of map matching is borrowed from the navigation algorithms for GPS(Quddus, Ochieng, and Noland 2007). It was first used to handle indoor path identification tasks with a wheeled robot (Wang, Veloso, and Seshan 2013; 2014), and proved robust in several real settings. For the path identification task for humans, existing works are mainly based on probability distributions (Rai et al. 2012; Ferris, Fox, and Lawrence 2007). We also mention the related work in (Woodman and Harle 2008) which uses a foot-mounted inertial measurement unit, polygonal building map, and particle filter, to accurately detect a human trajectory in a 3-dimensional indoor environment.

In this paper, the "snapping" technique of (Wang, Veloso, and Seshan 2013) is applied to humans for the first time, and the only sensors used are the accelerometer and gyroscope in a commonly available smart phone, the Samsung Galaxy S4. The process of deriving a walker's trajectory from raw data is as follows:

1. Collect raw data and identify steps (i.e. step length and heading).
2. Combine steps into a segmented trajectory.
3. "Snap" the segmented trajectory to a given map using specified error thresholds.

In a companion paper (Zha et al. 2015) we focus on the details of part 1. In this work however, we assume that step lengths and headings (with error) are given and discuss parts 2 and 3 above.

Given the complexity of real situations, we focus on one indoor setting, the highly structured environment of New York University's Center for Urban Science and Progress (CUSP). "Highly structured" here refers to a regime of many narrow, straight corridors and restricted spaces, as opposed to wide open, possibly curved spaces. The snapping algo-

Copyright © 2015, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

rithm is implemented for a complicated trajectory and fairly restrictive error thresholds, and demonstrates accurate performance. We introduce metrics to compare how uncertainty in a trajectory changes with number of path segments, compute these metrics in our example, and verify that they conform to intuition.

The results of this project and further work has a multitude of potential applications in an urban setting. City agencies can understand how indoor public spaces are used and learn aggregate patterns of pedestrian movement (for example, to link walking patterns to health outcomes). Retailers can use trajectory knowledge to optimize store layout and cultural institutions can learn which exhibits are most viewed. We note that currently a user's path is determined by data collected from his or her personal smart phone. A thorough discussion of effective large-scale data collection strategies and solutions to anonymity and privacy concerns is outside the scope of this paper, although certainly a prerequisite to implementation. We conclude with a discussion of further research directions regarding human trajectory identification in indoor spaces.

## Data

The snapping algorithm takes three inputs: a traversed path in the form of a collection of segments, a topological map of the ambient space, and a set of error thresholds. The algorithm compares the traversed path to the topological map and determines which paths are within the error thresholds. We discuss the creation of traversed path segments from user step data in the next section.

We denote a traversed path by $\Gamma_n = \sum_{k=1}^{n}(S_k, \theta_k)$, a formal sum of segment-angle pairs. Here $S_k$ is the $k^{th}$ segment and $\theta_k$ is the angle between $S_k$ and $S_{k-1}$, with $\theta_1 = 0$. Given $\Gamma_n$, let $\Gamma_j$, with $j \leq n$ denote the $j^{th}$ *partial path*. We assume that each segment and angle measurement contains some unknown error from the true values, and for convenience we frequently identify a segment $S$ with its length.

The next input to the snapping algorithm is a topological map, i.e. a simplified representation of an environment such as a floor of an office building, or platform of a subway station. A topological map is a collection of segments specified by their endpoints, representing hallways and their intersections, such that segments only intersect at endpoints. This also encodes lengths of hallways and angles between hallways. Note that a long hallway with several intersections will comprise multiple segments of varying lengths in the topological map that correspond to all possible sections of the hallway. Given a text file of the $(x, y)$ coordinates of all walls in an environment, we use a simple point-and-click program to select intersections and construct all possible edges between intersections that do not pass through a wall.

Finally, we set two allowable error thresholds, a length threshold $(dm_1, dm_2)$ and angular threshold $da$. The length threshold consists of both an allowable percentage error $dm_1$ and absolute error $dm_2$. For example, a trajectory segment $S$ is within the length threshold of a topological map segment $T$, if either $\frac{|S-T|}{T} < dm_1$ or $|S - T| < dm_2$. We specify both $dm_1$ and $dm_2$ to account for large percentage errors in short segments, and large absolute errors in long segments. The angular threshold is a constant measured in radians and comparison between angles is performed similarly. Note that a complete trajectory identification implementation that converts raw data into segments and then snaps segments to a topological map may have additional parameters that affect the accuracy of the input trajectory. These parameters are discussed in the next section and in (Zha et al. 2015).

## Methods

### Segment Identification

As introduced above, the traversed path $\Gamma_n$ describes the raw human trajectory in the form of segment-angle pairs. To obtain $\Gamma_n$, we first extract the steps of the walker from the accelerometer and gyroscope signals of a mobile phone, in the form of length-heading pairs. The steps where a turn occurs are identified and steps between turns are integrated into longer segments, representing straight walks in each hallway.

The task of step identification based on sensor signals from a mobile phone is challenging. Unlike wearable sensors, the user activity, device type and position the phone is carried on the body may lead to complex variations in input data. In our other work, we compared peak counting and template matching methods for step identification(Zha et al. 2015), but in this paper we assume that we have step information for the whole walk with some error. Although simply connecting all steps yields a trajectory, in order to use the snapping algorithm efficiently we aggregate steps into segments.

We regard a series of consecutive steps with small total change in heading as a straight walk and a series with larger change of heading as a turn. Explicitly, consider a sequence of steps $\{x_i\}_{i=1}^{m}$ with respective headings $\{\psi_i\}_{i=1}^{m}$, let $\tau > 0$ denote the turning threshold, and $w$, an odd integer greater than 3, the window length parameter. For each $i$, consider the window $W_i$ of $w$ consecutive steps centered at step $i$ (restricting the window size for $i$ near 1 and $m$ as necessary), and let $d\psi_i$ be the largest difference between all headings in $W_i$. Given a consecutive sequence $d\psi_i, \ldots, d\psi_{i+k}$, $k \geq 0$, each of which is greater than $\tau$, we say a turn occurred at index $median(\{i, \ldots, i + k\})$. After the turns are identified, the segments are calculated as the summed length of the steps between two turns and the angles of the segments are calculated as difference in headings between consecutive turns.

Regarding the choice of parameters, $w$ should be larger when the space is more open. For example, in a shopping mall with wide intersections, people may take more steps to finish a turn than in office environments. However, with a large $w$ the window sometimes can contain more than one turn, resulting in mis-identification of turns. On the other hand, the turning threshold $\tau$ should be small enough to recognize turns, but much larger than the error in individual step headings, so avoid over-counting turns.

Note that the task of separating turns and straight walks can be ambiguous due to the difficulty in defining a turn.

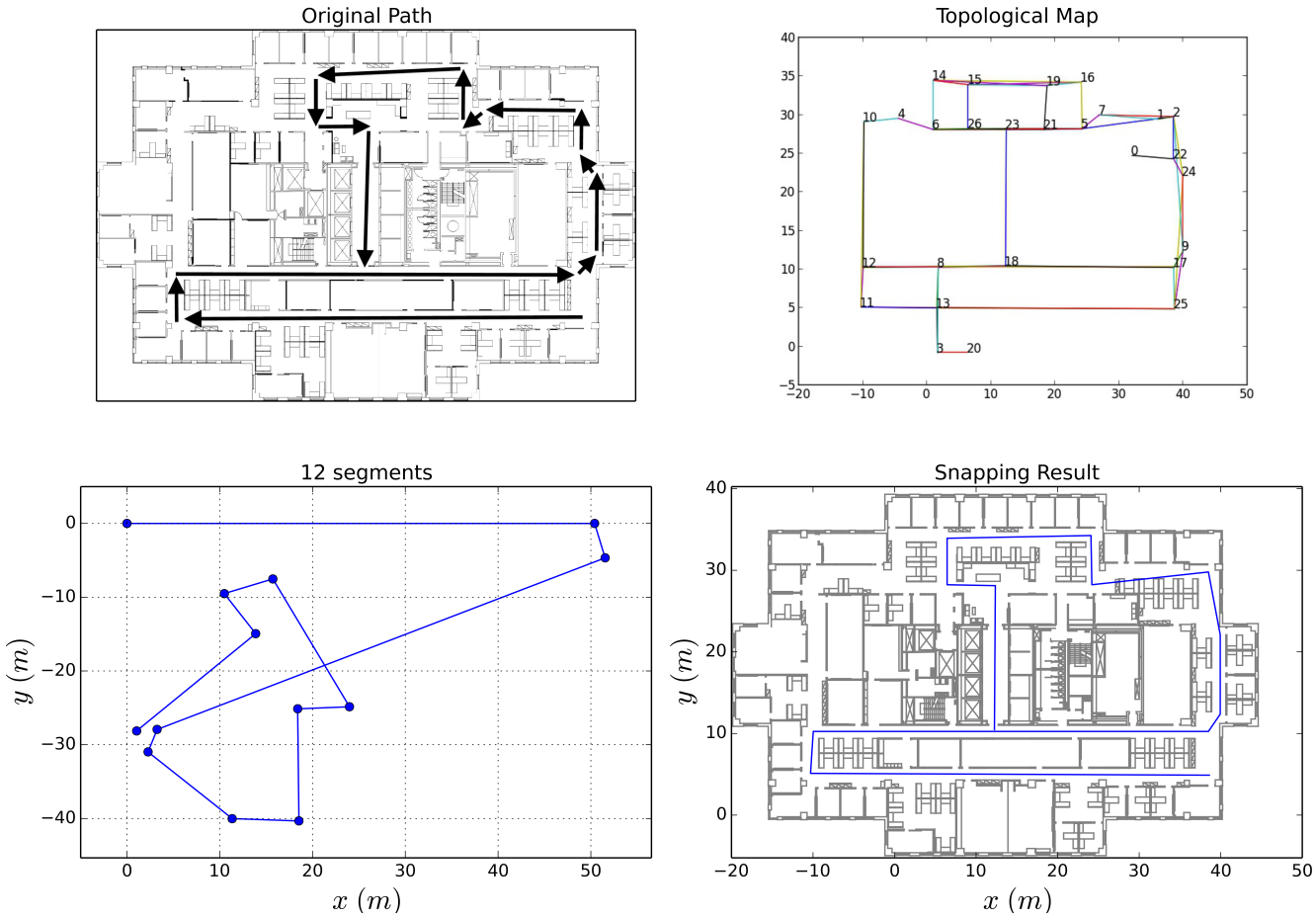


Figure 1: The original path displays the actual path traversed at CUSP, whereas the topological map represents the intersections and hallways used by the snapping algorithm. The segments plot shows the segmented input trajectory to the snapping algorithm (note that error in both length and angle is apparent). The result of the snapping algorithm is displayed in the lower right, and conforms closely to the original path, although direction of motion is not indicated.

**Algorithm 1** `Snap(currPoint,Segs,path)`

**if** $Segs$ **is** $empty$ **then**
  $outputpath()$;
  $exit()$;
**end if**
$AdjSegs = getAdjSegments(currPoint)$;
$MatchedSegs = underThresh(Segs[0], AdjSegs)$;
**for** $\forall$ $segment \in MatchedSegs$ **do**
  $newpath = [path, segment]$;
  $newPoint = otherEnd(segment, currPoint)$;
  $Snap(newPoint, Segs[1:], path)$;
**end for**

In a straight hallway people may not walk straight, and in a shallow-angled intersection , people may walk as if in one straight hallway. This kind of ambiguity can lead to error in segment identification, but a flexible topological map can account for this. In general, we would rather consider two real segments as one rather than break one actual segment into two. For example, in Figure 1, the segment identification algorithm ignored two small turns. The modification of the topological map is to actually allow some intersections that pass through walls to be connected by artificial "hallways". To build this topological map, one can first connect all adjacent intersections without passing through walls, and then artificially connect non-adjacent (but close) intersections which don't require a big turn to reach one from the other.

## Trajectory Snapping

The snapping algorithm finds all sequences of segments in the topological map that match with trajectory segments within the given error thresholds. In order to search all possibilities, the algorithm used a depth first graph search algorithm. The algorithm can be described as the recursive function **Snap**.

In function $getAdjSegments()$ the input is an intersection in the topological map and the returned value is the set of segments in the topological map with the specified intersection as an endpoint. In the function $underThresh()$ the first input is a segment, the second input is a set of segments, and the returned value is the subset of the second input consisting of those segments within the error thresholds of the first input. Finally, the function $otherEnd()$ takes two inputs, a segment and an endpoint of that segment. The returned value is the other endpoint of the segment.

When the algorithm terminates there may be several potential trajectories produced as output. We pick the one with smallest summed error as our final result. If there is no output, one can lower the thresholds $(dm_1, dm_2), da$ and rerun until a trajectory is successfully snapped. Note that there is an inverse relationship between the size of the error thresholds and the likelihood of successful snapping, and a direct relationship between the size of the error thresholds and the size of the algorithm's search space.

# Results

We implement the snapping algorithm in NYU's Center for Urban Science and Progress, which occupies the 19th floor of 1 MetroTech Center in downtown Brooklyn, NY. This setting is the floor of an indoor office building, with narrow corridors, offices, and cubicles. Most hallways are between 5 and 50 meters long, and most angles between corridors are right angles. Figure 1 shows both the underlying map of CUSP, along with the actual 14-segment long path traversed, as well as the topological map used for snapping. Note that in (Wang, Veloso, and Seshan 2013; 2014), the implementation was in the Gates Building at Carnegie Mellon University, which has a significantly less symmetric floor plan than CUSP.

Data was collected from an android application installed on the second author's handheld Samsung Galaxy S4 smartphone and converted into a segmented motion trajectory that also appears in Figure 1. Note that the segmented motion trajectory is comprised of 12 segments rather than the 14 actually traversed, and there is visible error in both segment angle and length. We used $dm_1 = 0.25$, $dm_2 = 5$, and $da = 0.8$, roughly $\frac{\pi}{4}$ radians, as error thresholds. We took the window length parameter $w = 5$ and the turning threshold $\tau = 0.4$ radians. We also assumed a length of $0.8$ meters for every two user steps when converting the raw input data into a segmented trajectory.

We have plotted the result of the snapping algorithm in the lower right panel of Figure 1, which conforms quite closely to the actual traversed path, although direction of motion is not indicated. Although the segmented input trajectory combined two pairs of short segments, yielding 12 segments rather than the actual 14, the snapping algorithm correctly snapped all twelve segments to their best match. In this implementation there were two output paths that were admissible within the given error thresholds, but given multiple admissible paths, we rank them by total percentage error (the sum of percent angular and percent length errors over all trajectory segments) and choose the lowest-error path to plot.

Next, we examine how uncertainty in the snapped path varies with the addition of new segments. In particular, we look at both how the total number of possible snapped paths varies, as well as how the number of possible matches for a given segment varies.

First, we introduce some notation to facilitate a discussion of uncertainty in the snapped trajectory as the number of segments in a path increases. Fix a topological map $M$, a set of error thresholds $E$, and an input trajectory of $n$ segments, $\Gamma_n = \sum_{k=1}^{n}(S_k, \theta_k)$ as above. We define

- $\rho_j(S_k) =$ the number of matches for segment $S_k$, considered as a segment in $\Gamma_j$, where $j \geq k$, when $\Gamma_j$ is snapped to $M$ with thresholds $E$.
- $\rho(\Gamma_j) =$ the number of trajectories for $\Gamma_j$ that snap to $M$ with thresholds $E$.

We expect $\rho_j(S_k)$ to decrease monotonically as $j$ increases, for a fixed $k$. That is, adding additional constraints to $\Gamma_j$ can only decrease the number of possible matches for $S_k$. Intuitively, segments early in a trajectory become

| | $\Gamma_2$ | $\Gamma_3$ | $\Gamma_4$ | $\Gamma_5$ | $\Gamma_6$ | $\Gamma_7$ | $\Gamma_8$ | $\Gamma_9$ | $\Gamma_{10}$ | $\Gamma_{11}$ | $\Gamma_{12}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Segment 1 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Segment 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Segment 3 | - | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Segment 4 | - | - | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Segment 5 | - | - | - | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 |
| Segment 6 | - | - | - | - | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| Segment 7 | - | - | - | - | - | 2 | 1 | 1 | 1 | 1 | 1 |
| Segment 8 | - | - | - | - | - | - | 1 | 1 | 1 | 1 | 1 |
| Segment 9 | - | - | - | - | - | - | - | 2 | 2 | 2 | 2 |
| Segment 10 | - | - | - | - | - | - | - | - | 2 | 2 | 2 |
| Segment 11 | - | - | - | - | - | - | - | - | - | 3 | 2 |
| Segment 12 | - | - | - | - | - | - | - | - | - | - | 1 |
| Full path | 2 | 2 | 1 | 2 | 3 | 2 | 1 | 2 | 2 | 3 | 2 |

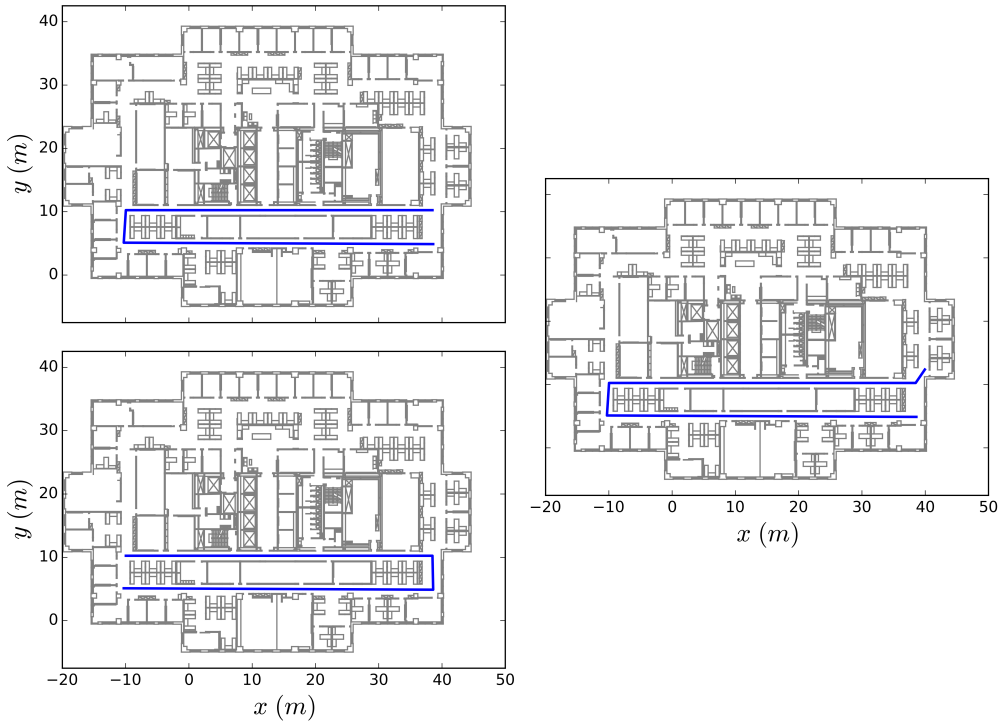


Figure 2: The $(i, j)$ entry in the table above shows the number of possibilities for Segment $i$ in the partial path $\Gamma_j$, where $1 \leq i \leq 12$ and $2 \leq j \leq 12$. The $j^{th}$ entry in the last row in the table gives the total number of possible matches for $\Gamma_j$, i.e. the maximum of all entries in the $j^{th}$ column of the table. Note that each of the first twelve rows decreases monotonically as $j$ increases, illustrating that earlier segments are "locked in" as new segments are added to the path. The three plots give an illustration of this for $\Gamma_3$ and $\Gamma_4$. Specifically, the two left plots demonstrate the two possible matches for $\Gamma_3$, and the two possibilities for Segments 1, 2, and 3 in $\Gamma_3$. The right plot demonstrates that the addition of Segment 4 gives a unique result for $\Gamma_4$, and hence there is now only one possible match for Segments 1, 2, and 3 in $\Gamma_4$.

"locked in" as newer segments are added, although the overall path may retain ambiguity. We observe this behavior also by looking at the number of matches for segment $S_k$, $k = 1, \ldots, 12$ in the context of the path traversed at CUSP. This is displayed in the table in Figure 2, where the number of matches for any given segment $S_k$ only stays the same or decreases as the number of segments in the path it belongs to increases. Explicitly, $S_3$, for example, has two possible matches in $\Gamma_2$ and $\Gamma_3$, then has only one possible match in $\Gamma_4, \ldots, \Gamma_{12}$, as illustrated in the plots in Figure 2.

On the other hand, $\rho(\Gamma_j)$ may fluctuate non-monotonically as $j$ increases from 1 to $n$. For example, $\rho(\Gamma_j) < \rho(\Gamma_{j+1})$ may occur if $S_{j+1}$ matches several segments in the topological map within given error thresholds. On the other hand, $\rho(\Gamma_j) > \rho(\Gamma_{j+1})$ if the addition of $S_{j+1}$ eliminates possible matching trajectories for $\Gamma_j$. We

observe this behavior in the 12-segment path traversed at CUSP, in the last row of the table in Figure 2. The total number of matches varies between 1 and 3, and increases and decreases as more segments are added.

## Discussion and Conclusion

The above results provide a first application of smartphone-based human trajectory identification using the o-snap algorithm of (Wang, Veloso, and Seshan 2014) in an indoor setting. We emphasize that this method does not use any fixed infrastructure or GPS, but rather just data obtained from the sensors inside a commonly available phone. However, there are numerous further engineering challenges to be overcome, and this work may provide the foundation for myriad applications of trajectory identification in an urban setting.

A first challenge is the extension of the snapping approach to open spaces, rather than the narrow corridors found in office buildings. It is not clear how to best construct a topological map for an open space, such as an indoor lobby or mall plaza. If one attempts to create artificial hallways in a topological map of an open space, i.e. to represent the possible routes people take as a sequence of short straight lines, the algorithm search space may grow tremendously. Furthermore, the shorter such artificial hallways are, the more prone the output trajectory may be to accumulated drift error. In addition, it is not clear that humans in a wide open space walk in straight lines; perhaps they walk along curved paths. More generally, it is not clear how to construct a topological map that takes into account curved walls. Open spaces – particularly in crowded urban settings – may also be full of people, forcing a trajectory to change direction or pause frequently. Adapting the snapping technique to take pausing and mid-segment changes in direction into account is another direction of research.

Some other ways to extend our trajectory identification method include identification of starting location by the user and allowing the user to travel between building floors (and hence incorporating several topological maps into the same snapping algorithm). We anticipate that large-scale experiments incorporating many different users and devices would be necessary before any real-world implementation of the snapping algorithm is feasible. As noted in the introduction, there are important privacy issues to be considered, for even if a user's identifying information is completely erased from his or her trajectory, it may still be possible to de-anonymize users by correlating trajectories with external data sets. Additionally, processes involving automatic data collection from a user's smartphone will have to be developed to encourage large-scale adoption of our technique. Derived statistics from a large corpus of trajectory data will also be a useful output for both researchers and city agencies, to understand where city residents walk and how movement patterns change over time or in response to particular events.

Some proposed urban applications of this work and its extensions include:

- Event detection in subway stations, malls, or arenas. Given an anonymized, aggregated output of real-time pedestrian trajectories, city or security agencies may determine anomalous walking patterns (e.g. pedestrians avoiding an unsafe obstacle).
- Optimal resource allocation. Understanding highly trafficked pedestrian areas could inform the number and location of garbage cans, water fountains, and provide a more efficient alternative than assuming uniform spatial resource consumption.
- Learning exhibit preferences of museum patrons. Installing a trajectory data collection application on guided-tour iPads would allow curators at cultural institutions to understand which exhibits are preferred by which visitors, and could therefore help improve the user's experience.

In addition to the aforementioned engineering-focused extensions to this project, we also plan to construct a general theoretical framework to analyze snapping algorithm performance for a given topological map $M$. To motivate this idea, suppose $M$ is highly symmetric (for instance, the boundary of a square), then the snapping algorithm will be unable to distinguish between many different paths in the absence of a fixed starting location. Even if a starting location is given, one may have an $M$ that is still unable to distinguish between different paths if the allowed error thresholds for displacement and heading are sufficiently, but not unreasonably, large. However, intuition suggests that for a sufficiently irregular map $M$, the longer a path is, the fewer potential snapped paths will be produced by the algorithm. We anticipate that such a theory may begin by creating for each $M$, an associated function

$$f_M : \mathbb{R}^k \times P_M \longrightarrow \mathbb{Z}_{\geq 0}, \qquad (\theta, \Gamma) \longmapsto l$$

where $\theta$ is a vector of error thresholds, $P_M$ is the set of all possible paths on $M$, and $l$ is the number of paths within the error thresholds that snap to the given path $\Gamma$ on $M$. For $\theta$ in some range (depending on $M$), $f_M$ may be a decreasing function of the length of $\Gamma$. A potential application of this theory would be to give a quantitative measure of how accurately we may expect the snapping algorithm to perform in a real-world situation.

## References


Ferris, B.; Fox, D.; and Lawrence, N. 2007. Wifi-slam using gaussian process latent variable models. In *Proceedings of the international joint conference on artificial intelligence (IJCAI)*.

Quddus, M. A.; Ochieng, W. Y.; and Noland, R. B. 2007. Current map-matching algorithms for transport applications: State-of-the-art and future research directions. *Transportation Research Part C: Emerging Technologies* 15(5):312–328.

Rai, A.; Chintalapudi, K. K.; Padmanabhan, V. N.; and Sen, R. 2012. Zee: Zero-effort crowdsourcing for indoor localization. In *Proceedings of the 18th annual international conference on mobile computing and networking*, 293–304. ACM.

Wang, R.; Veloso, M.; and Seshan, S. 2013. Iterative snapping of odometry trajectories for path identification. In *2013 RoboCup International Symposium*.

Wang, R.; Veloso, M.; and Seshan, S. 2014. Optimal snapping of odometry trajectories for route identification. In *Proceedings of ICRA '14, the IEEE international conference on robotics and automation*.

Woodman, O., and Harle, R. 2008. Pedestrian localisation for indoor environments. In *Proceedings of the 10th international conference on Ubiquitous computing*, 114–123. ACM.

Zha, Y.; Shroff, R.; Wang, R.; and Veloso, M. 2015. Indoor trajectory identification: accurate step detection. Submitted to AAAI 2015 Workshop on AI for Cities.