Crowdsourced Pedestrian Map Construction for Short-Term City-Scale Events

Ulf Blanke* , Robin Guldener† , Sebastian Feese‡, and Gerhard Tröster§

Wearable Computing Lab, ETH Zurich, Zurich, Switzerland

*blankeu@ethz.ch, †robing@student.ethz.ch, ‡sfeese@ethz.ch, §troester@ethz.ch

ABSTRACT

This paper targets the construction of pedestrian maps for city-scale events from GPS trajectories of visitors. Incomplete data with a short lifetime, varying localisation accuracy, and a high variation of walking behaviour render the extraction of a pedestrian map from crowd-sourced data a difficult task. Traditional network or map construction methods lean on accurate GPS trajectories typically obtained over longer time periods from vehicles at high speeds with less variation in locomotion. Not designed to operate under mobility conditions of pedestrians at large scale events they cannot be directly applied. We present an algorithm based on a crowd-sensing scheme to construct the pedestrian network during city scale events. In a thorough evaluation, we investigate the effect of trajectory quality and quantity on the map construction. To this end, we use a real world dataset with 25M GPS points obtained from 28.000 users during a three-day public festival event. Results indicate that with a short observation window of 30min the estimated pedestrian network can represent previously unseen trajectories with a median map-matching deviation in matching of only 5m and a map accuracy of more than 85%.

Keywords

mobility mining, crowd sourcing, pedestrian networks

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications – data mining, spatial databases and GIS

1. INTRODUCTION

Large scale events attract hundreds of thousands of visitors. Coming together in a public space visitors move in large numbers through a complex space defined by the event organiser. Temporary layouts are created to fulfil event requirements and navigate the masses to attractions, food locations or music stages. Within a few days a city infrastructure is heavily altered or created in open spaces such as for festivals. Roads are closed, pedestrian routes are carved, and new points of interests such as attractions or food booths are introduced. Effecting the behaviour of pedestrians, previously unseen and not anticipated mobility patterns in terms of paths taken or path occupancy can occur during the event.

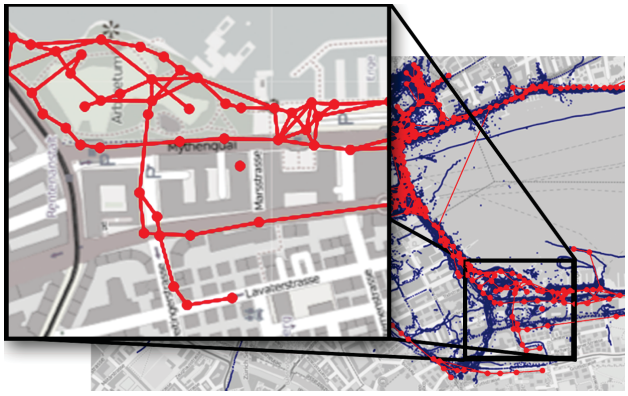


Figure 1: Algorithm output of a pedestrian map (red) from raw GPS trajectories (blue) of festival visitors.

Capturing and understanding pedestrian mobility can be a powerful ingredient to many applications for the operative level of event organization. Identifying the quickest route through the crowd for first emergency response, movement and density prediction for early warning, or planning for emergency booths with quickest reachability are just a few examples. Once a graphical representation [5] of potential pedestrian paths is available, crowd-mobility traces can be map-matched and attribute the graph, e.g., with speed or occupancy, to enable routing and other graph-based operations for pedestrian mobility. While commonly applied in automotive applications, it has not found application for pedestrian mobility yet, especially for the scenario of large-scale festivals with thousands of visitors.

A major inhibitor to allow routing or other graph-based operations is a missing routable map attached to the underlying mobility. Such maps can be obtained manually, but it becomes tedious with permanent changes in the infrastructure. Indeed automatic approaches have been proposed to crowd-source routable maps [8, 6] using in-car GPS devices. Increasing availability of GPS-enabled smartphones paired with phone customisation using apps and channel-

Urb-IoT 2014, October 27-28, Rome, Italy

DOI 10.4108/icst.urb-iot.2014.257190

ing through appstores, we also have a powerful instrument to collect required data for pedestrian mobility in a similar way.

In this work our goal is to automatically infer a pedestrian map[1] during large-scale festivals as shown in Figure 1. While a large corpus of work exists on estimating networks from crowd-sourced trajectory data, most of it focuses on vehicular data and cannot be applied directly. Mobility data obtained from pedestrians differs significantly from vehicles and imposes new challenges: First, pedestrian systems are highly dynamic. They have a large behavioural component, few physical constraints only, and opposed to vehicles they are not restricted to lanes. Pedestrians typically exhibit slow walking speeds, frequent direction changes, and can move freely across streets, pavements, and open spaces such as parks. Second, festival events last only up to days. Crucial for real-time applications is a quick construction within a few minutes or fraction of an hour. Third, operating in a crowd-sourced scheme obtained pedestrian GPS data can be highly noisy. Visitors are not instructed, carry the device freely in the pocket, and walk freely through the area. As a consequence GPS-reception can vary or fail completely leading to partially observed or imprecise trajectories.

We address these challenges and contribute an algorithm for pedestrian map construction specifically tailored to the use case of festivals. Using a large scale dataset [4] with 28.000 users contributing 24M GPS data points, we thoroughly analyse the quantity and quality required for creating the pedestrian network. We show that our crowd-sourced approach is able to construct a pedestrian network in less than $30min$ for a festival area of $1.5km^2$ with good generalization properties to different times of day of the event.

2. RELATED WORK

The ubiquity of GPS-enabled devices such as navigation systems in cars or smartphones gave opportunity to collect data from the masses and, for instance, to create automatically routable road maps [6], to detect changes in road networks [9] or to perform graph-based operations on road networks for further analysis [8]. Biagoni et al compile in a recent survey [3] a comprehensive list of map construction algorithms from vehicular GPS data and provide an overview of quantitative and qualitative assessment of the algorithm's performances. Unlike vehicles being constraint to roads with few direction changes, or indoor pedestrians being constraint by walls, festival visitors exhibit much fewer constraints — characteristics that have not been taken into consideration in these works. While the survey focuses on vehicular data, we will see in Section 3 that we follow common steps, but introduce modifications to address the challenges of pedestrian network construction.

In the domain of constructing pedestrian networks (or maps) indeed only few works exist. Mainly, automatic floor plan construction, e.g. for navigation, or location-based services in shopping malls have been targeted by research. Since GPS fails to obtain a position fix indoors, approaches vary in terms of modalities used. Deadreckoning using motion sensors to obtain pedestrian trajectories for map construction has been proposed in [13]. It does not require any infrastructure and has the ability to operate indoors. In [12] or [1] the user's trajectories are calibrated by WiFi-beacons with known position. Latter assume a given WiFi-infrastructure, which is annotated in a 2D map. In [10] the authors focus on pedestrian network construction using GPS. On average collected tracks contain 650 points, last for $10min$ and cover $670m$ of length. Note that these conditions do not apply for a realworld data collection as in festivals (see later in Section 5.1). In our case, lanes need to be constructed from multiple partially observed paths for a single lane, for which this algorithm has not been designed for.

[1]We use the terms *network*, *graph* or *map* synonymously.

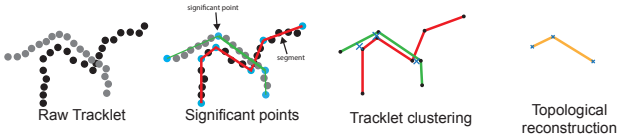


Figure 2: Overview of the pedestrian network construction algorithm.

Furthermore, investigated pedestrian datasets consist of few volunteers or co-working researchers instructed to walk continuously and creating clear trajectories [13, 10]. We investigate our algorithm on *crowd-sourced* data from thousands of non-instructed and partially observed visitors. With this large data corpus (see Section 4) we are able to investigate in depth the critical aspects of map creation such as data quantity and quality. In the next section we present our algorithm for crowd-sourced pedestrian map construction.

3. PEDESTRIAN MAP CONSTRUCTION

Our goal is to represent typical paths pedestrians take as a graph (or routable map). As input let $\{T_u\}$ be a set of trajectories for each user u. Each trajectory T_u consists of a sequence of $\mathbf{p} = (x, y, e, t)^T$, where x and y correspond to GPS-sourced longitude and latitude position obtained at time t with an estimated positioning error e. We follow typical steps outlined in [3]: (1) a filtering step to check for irregularities, (2) a data reduction step to significant points, and (3) a merging step to link significant points to form a pedestrian network.

Tracklet Segmentation.

Trajectories can be gapped by missing or bad GPS-signal. We first filter all positions with an GPS-error $e > 20m$. To avoid erroneous trajectories by hallucinating connectivity of positions that are far apart, we first segment the trajectories into tracklets. To this end, we segment each trajectory T_u into a set tracklets $S = \{T_u^i\}$ for $i = 1..n_u$ for all users u. For instance, if two consecutive points $\mathbf{p}_1$ and $\mathbf{p}_2$ of T_u exceed a thresholding distance d the trajectory is segmented into $T_u^{(1)}$ and $T_u^{(2)}$ at these points and added to the set S. Considering physical proximity between consecutive positions ensures spatial-temporally continuous of resulting trajectories. We experimentally set d to $30m$.

Based on the input of the tracklets S we apply our approach of estimating the pedestrian network. Figure 2 depicts the three steps to obtain the topological reconstruction from raw tracklets S. First, we approximate each tracklet to extract *significant points*. We define significant points as turning points of the trajectory. Based on the approximated trajectories, we cluster trajectories based on the proximity of segments of pairwise significant points. Finally, we discover the topological reconstruction including intersections

by connecting significant points. Next, we describe these steps in more detail.

3.1 Approximation and Significant Points

We identify significant points $\mathbf{p}_s$ to capture relevant characteristics in a minimal data representation for further processing. A common method to reduce the set of raw GPS trajectories is to approximate by k-means clustering of GPS-location and heading θ [7]. However, this requires to set an initial k, thresholding manually a heading θ for a semi-random seed of the k centroids. A more intuitive and automatic approach we selected is based on a sliding-window and bottom up (SWAB) approximation [11]. It is typically used for approximating one-dimensional time series. Given GPS tracks—considered as 2-dimensional time series—we extended the cost function of SWAB to 2D as the mean error of a segment $\epsilon = \frac{1}{n}\sum_{i=0}^{n}|x'_i - x_i|$ where x'_i is the projection of x_i on the segment. Figure 3a shows SWAB-approximated tracklets for $\epsilon = 4m$. The red dots shows the set of significant points $\mathbf{p}_s$ per tracklet and essentially represent turns of the pedestrian. The red lines represent segments between pairwise significant points.

3.2 Clustering Significant Points

Given the approximated tracklet segments and significant points $\mathbf{p}_s$ from the step before, we now merge significant points from multiple users to form representative points $\mathbf{p}'_s$ of a shared path. To this end, we iterate through each segment consisting of pairwise significant points for each tracklet. Based on a distance d and bearing difference b we identify similar segments. For the set of similar segments, we re-adjust the significant points $\mathbf{p}_s$ by averaging the position with the closest significant point of the corresponding segment. Note that the approach of clustering (or merging) is similar to [7] except to our extending re-adjustment step. This step allows segments to represent multiple tracklets with minimal error. Figure 3b shows the original significant points in gray and their re-adjusted position in red. For example, the significant points from segment a and b are readjusted and merged to points 1 and 2. If not otherwise noticed we set $b = 10^\circ$ and $d = 10m$.

3.3 Topology inference

In a final step we connect clustered significant points by inferring paths and intersections. As we keep information about the originating tracklets of the significant points, we follow these assumptions: (1) we assume that two significant points are connected if a connection exists of the underlying segments. (2) Two points shall be connected if there is not a third point in-between the underlying segments. To this end, we first sort all points according to each of the underlying segments and then extract the immediate neighbour(s) for each point according to each of its segments. An observed point is only connected *directly* to its neighbour(s) (connection pair), i.e., if there is no point in between. Furthermore, a maximal neighbour perimeter p is considered for neighbours from the segment that the connection pair shares. For instance, in Figure 3(c) the significant points 4 and 6 are not connected even though they share a segment. Because point 5 is within a lanewidth l of a segment running from 4 to 6, and the points 5, 4 and 6 share a originating segment (a and c), the connecting segment e is ignored. Figure 3c depicts an example topology reconstructed by four tracklets. It can be seen that multiple partial observations uncover the topological form of a T-junction.

Figure 4: Area (green) of data collection of the festival Züri Fäscht.

4. DATASET

Every three years the city of Zürich is host to Switzerland's biggest event: The Züri Fäscht. During the event happening in the city center and around the lake of Zürich (see Figure 4), the urban infrastructure is redesigned by closing roads and deploying booths, music stages, attraction points. Over the course of three days one million daily visitors are entertained by large program of concerts and shows and move freely within the festival area.

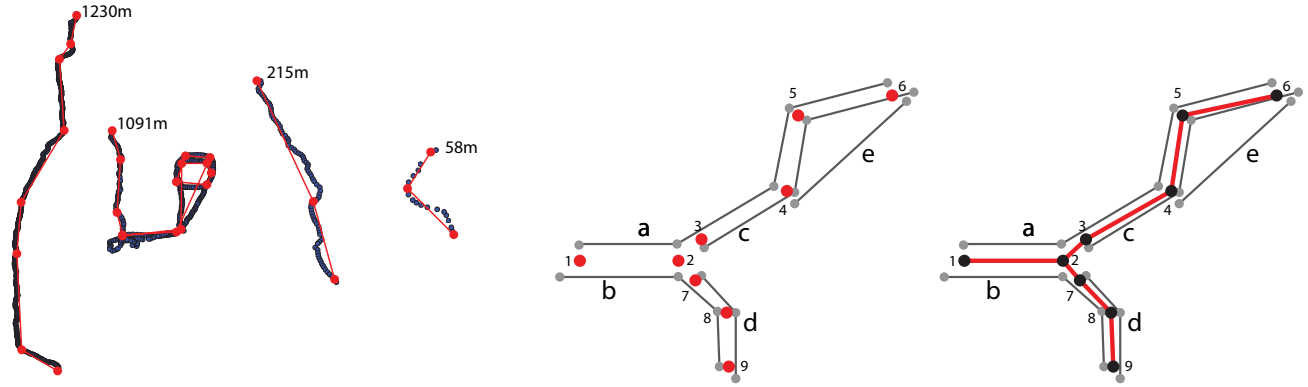
In [4] Blanke et al collect pedestrian trajectories based on GPS from the festival visitors. To this end, an official app has been deployed. It offers features for the user such as a program and a map and collects continuously the location of the user in the background. Each location is attached to an anonymous id that enables us to create location trajectories. Data has been transmitted to a server in $2min$-intervals. Thereby, the location data was available near to real-time for showing momentary crowd conditions. For the three-day period of the event, 24M individual locations from 29.000 visitors have been collected. With the size of the data corpus it offers an ideal testbed for the evaluation of our algorithm.

5. EVALUATION

Crowd-generated data can be highly noisy, erroneous and incomplete. Intuitively, the more data we have the more we can filter on quality aspects and potentially improve the algorithm's result. In this section, we investigate quantity and quality requirements of trajectory data for estimating a map in a crowdsourcing scheme. Before presenting the evaluation of the pedestrian map construction, we investigate the characteristics of the crowd-sourced localisation data.

5.1 Analysis of Localization characteristics

The data collection has been performed by visitors during their normal behavior. Variation in carrying the device, device type and model, crowd conditions, and nearby buildings or trees effect the reception and quality of the data. Furthermore device type and localisation management of the particular smartphone OS exhibit varying sampling rate and localisation methods (GPS, Wifi, Cell) that are not in control of the developer, but depend on the mobility behaviour of



(a) Result of SWAB (b) Clustered significant points (c) Inferred topology

Figure 3: (a) Result for 2D-SWAB in red for four tracklets with raw GPS-points in blue, (b) Example for merging significant points of similar trajectories. and (c) Example for a reconstructed topology for four tracklets creating T-junction.

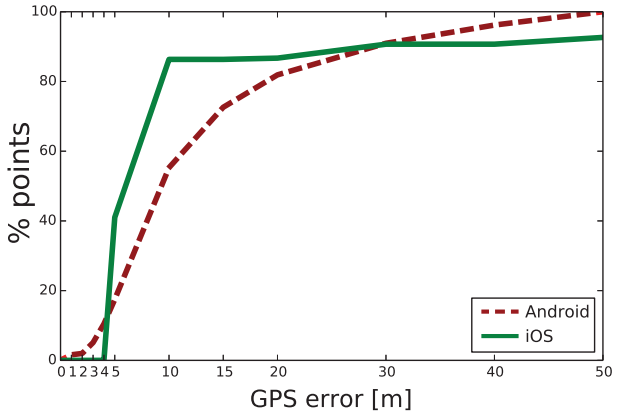


Figure 5: Maximum GPS error estimation for iOS and Android phones.

the user[2]. In the following, we show the distributions across users regarding *GPS accuracy* and *sampling frequency*, as well as the tracklet characteristics *length*, *duration*, and the *number* of originating GPS points.

GPS accuracy.

We used the GPS accuracy estimation from the phones as defined by Android and iPhone[3]. Figure 5 shows the accuracy in meters as a cumulative distribution for both smartphone brands iOS and Android. It can be seen that we can reach for >80% of iOS devices an accuracy of 10m, while Android reaches 20m accuracy for the same percentage. We see that the quality of GPS traces varies across users. Therefore, we investigate the effect of the GPS accuracy on the map construction in Section 5.3.

Sampling frequency.

A critical aspect to obtain user trajectories is the continuity of the location sampling. Differences in the location management of the respective OS or the GPS reception can influence the sampling rate. Figure 6 shows the time interval between sampling two consecutive points, accumulated over the entire dataset. We see that consecutive points with less than 10s delay dominate the dataset with over 90%. That means that in 90% of the data approximately a GPS point is available every 10-15m assuming normal pedestrian speeds.

Figure 7 shows the cumulative distribution of multiple tracklet characteristics relevant for the map construction:

Tracklet length and duration.

The dashed blue line shows the cumulative distribution for distance walked by a pedestrian for each tracklet. For example 50% of the tracklets are shorter than 100m lasting about 2-3min (green line), and only 20% are longer than 400m. Note that this differs to previous work which assume long paths taken by single users that cover a significant part of the map [13, 10]. Festival paths span over kilometres of which we capture only sporadic and partial observations. Therefore, we investigate potential effects of the track length on the quality of the map construction.

Tracklet points.

The number of points (red line) per tracklet reflects the sampling frequency and is assumed to be critical to obtain significant points. 50% of the tracklets contain 21 points—a fraction compared to previous work [10]. We assume that few tracklets with a high number of points capture most of the maps information and are better suited for map construction for a large festival area.

As mentioned before, we obtained data from regular users that did not take special care of an accurate GPS-localization. Consequently, the characteristics of the obtained data varies greatly as can be seen by the slowly saturating distribution gradient (instead of a steep gradient) in Figure 7. To understand the impact of these characteristics on the construction of the pedestrian map, we investigate these by filtering each individually in the next section.

[2]See Apple iOS Location and Maps Programming Guide.
[3]See the *android.location.Location.getAccuracy()* reference and for Apple iPhones iOS 6: Understanding Location Services knowledge database.

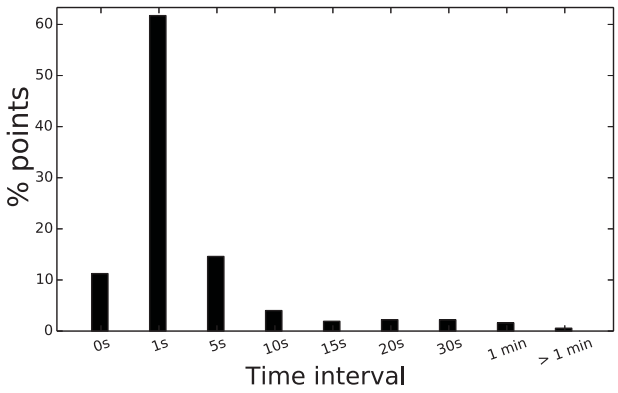


Figure 6: GPS accuracy estimation for iOS and Android phones.

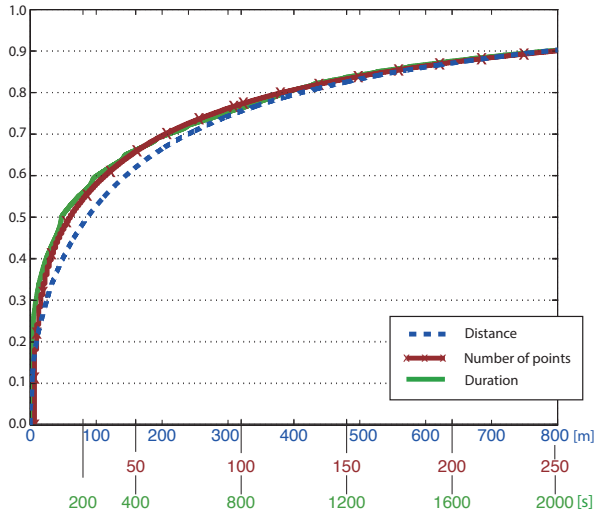


Figure 7: Cumulative distribution over the tracklet characteristics: length in meters, number of samples, and duration in seconds.

5.2 Evaluation scheme and criteria

We perform a quantitative evaluation of filtering the characteristics of GPS accuracy, track length, and number of points for the map construction. Since filtering reduces the quantity of data, we also investigate the observation period to counter-steer the data reduction. A quick map construction is crucial for the festival use case. Therefore, we vary the observation period for short times from 10min to 120min. We set the observation start time to the start of the event at 5pm, Friday 06/05/2013.

We defined two evaluation criteria *accuracy* and *trajectory representation* to evaluate the quality of the resulting graph:

Accuracy.

We calculate the accuracy by counting the hits of graph components (edges, nodes) into lanes and open spaces on a map (see Figure 8a). For open spaces, we annotate the entire walkable space. If a graph component (edge and node) covers a lane or zone, the component is marked as correct, otherwise as a miss. By dividing the correct components by the total number of graph components, we obtain the graph accuracy.

(a) Accuracy (b) Representativeness

Figure 8: Metrics used for quantitative evaluation. (a) Accuracy of constructed pedestrian map compared to annotated streetmap. The red color shows an inaccurate segment. (b) Error of representing previously unseen trajectories (blue).

Trajectory Representativeness.

Previously unseen trajectories should be well represented by the constructed map. To this end, we construct a map for a specific time period and select n random trajectories from other times. The random n trajectories are map-matched using point matching as in [2]. Each raw GPS point is matched by orthogonal projection to the nearest map edge. We report on the median distance of original GPS points and their projection on the edge. Note that we favour the median over average, to avoid the effect of outliers. For example, some visitors used the boat across the lake, which has not been considered in the map. This leads to a high distances to the graph that is not necessarily relevant and thereby artificially increases the average distance.

5.3 Results

We show the impact of GPS accuracy (Figure 9), track length (Figure 10) and the number of available points per tracklet (Figure 11) by sweeping a filter-threshold over these parameters. The x-axis represents the observation period as measure for data quantity. The y-axis shows the *accuracy*, respectively median distance per trajectory as the *representativeness* measure.

GPS accuracy.

In Figure 9 we see that filtering GPS-accuracy with more than $9m$ significantly improves map accuracy, but comes at the cost of representativeness by increased distance to unseen trajectories. This is the result of lower data quantities for constructing paths to represent new trajectories. Increasing the time period for construction, increases the required quantity, and the median distance drops below $2m$. However, this comes at the cost of accuracy, dropping from a maximum accuracy of 90% at $30min$ and 6m GPS accuracy (see annotation) to 78%. Interestingly, one can observe a large jump in GPS accuracy of larger values than $10m$. As can be seen in Figure 5 the distribution exhibits a steep gradient for iOS devices accounting for 90% of the data having an accuracy of $10m$ and greater. An optimal tradeoff can be probably found at $30min$, filtering GPS at $8m$ accuracy.

This leads to a median distance greater than $5m$ and an accuracy of almost 90%.

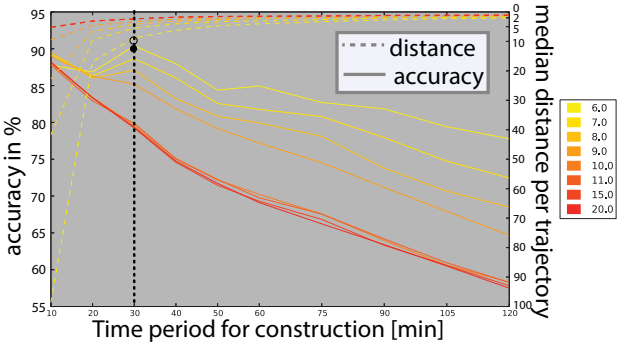


Figure 9: Results for filtering tracklets by GPS-accuracy

Tracklet length.

For results of varying the minimum track length we see a similar result in Figure 10. By filtering the tracks by their length the accuracy can improve up to 20% when observing $120min$ of data maintaining a median distance of less than $2m$. As can be seen with the dashed annotations, a similar accuracy can be obtained with a stronger filter on the minimum track length and the observation time. This means that short observations—but from many—can benefit the map construction in short time. This is favourable over waiting for longer tracks and applying a minimum length filter.

Tracklet points.

Again, we can observe a similar trend for filtering the minimum number of points per tracklet (see Figure 11); the stronger we filter, the higher the accuracy. But it also lowers the representativeness. This is to be expected. Less data remains which can be remedied by increasing the observation period. The effect of filtering the number of tracklet-points leads to improvement of similar order of magnitude. The highest improvement of accuracy is again at $120min$ from 60% to 75% with a median distance of less than $2m$. We took a closer look at the absolute number of used points with similar quality (see annotation). Interestingly, we saw for the shorter observation time that only 33% less points

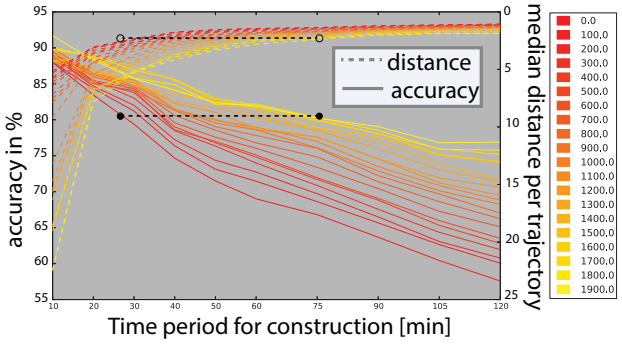


Figure 10: Results for filtering tracklets by observed distance.

have been used. This suggests that a shorter window with tracklets of fewer points (i.e. partial observations) can indeed help to generate a map within a short cycle.

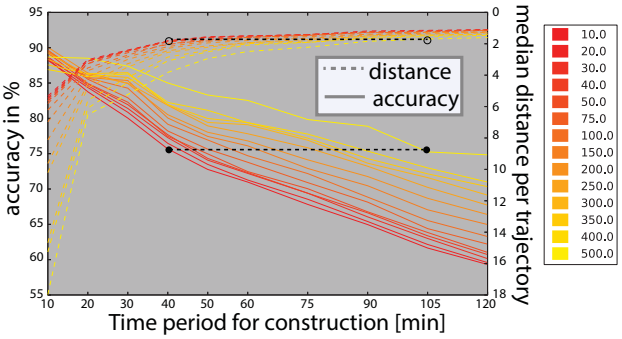


Figure 11: Results for filtering by number of points per tracklet

Discussion.

Overall we can see that by increasing the observation period the representativeness increases. This is as expected as more data reveals more potential paths and leads to a pedestrian network with higher density. At the same time the accuracy decreases. With a higher network density it is likely that more edges lie outside of the annotations. For an observation period longer than $30min$, we see that the representativeness begins to saturate. We can also see that the accuracy has an overall slight "dent" of improvement at $30min$. Therefore, we see this observation period as ideal which also fits the requirements of a quick network construction.

For visual inspection we show in Figure 12 the output for a pedestrian network constructed for $30min$ filtering GPS accuracy greater than $7m$. It can be seen that the graph captures the walked paths well (A). However, it can be also seen in (B) that a mesh is created that interconnects neighbouring lanes. While pedestrians indeed may switch lanes at any place, a more simplified representation is often more desirable. This can be remedied with graph-based smoothing, which we keep as a future improvement.

6. CONCLUSION

We presented an algorithm for pedestrian network construction for large scale events with thousands of visitors. Imposing challenges by noisy and partial trajectory observations, we investigated data quality and quantity required for map construction. We evaluated our algorithm quantitatively in terms of representativeness and accuracy. In a crowd-sourced scheme, we are able to create an accurate network within less than $30min$. Our algorithm is able to connect partial observations and to reveal underlying paths of visitors. Thereby, we are able to address the specific requirement of quick map generation for a short life cycle usage during festivals. This enables graph-based operations for application such as navigation, mobility prediction or planning for festivals without requiring a previously established pedestrian infrastructure.

For future work, we plan to add attributes of underlying mobility and investigate graph-based prediction algorithms

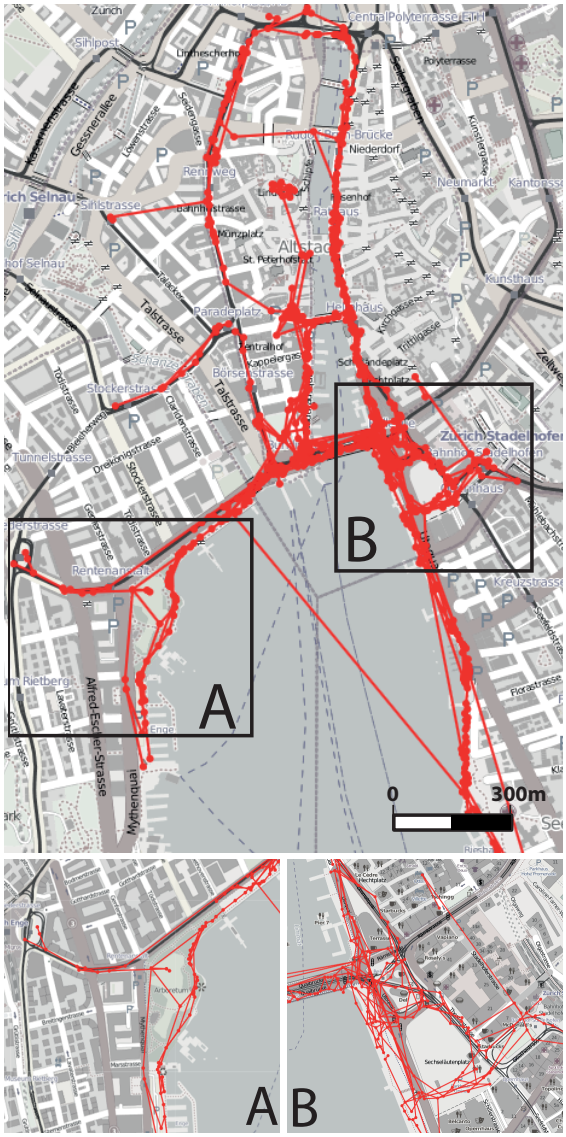


Figure 12: Example output for a 30min window.

for identifying trending densities, as well as routing applications such as estimated arrival times for emergency units.

7. ACKNOWLEDGEMENTS

We thank the organizers of the Züri Fäscht, especially Thomas Irninger and Roland Stahel for supporting marketing the app for data collection as well as the City Police Zürich, in particular Adrian Zemp and Andreas Moschin. This work has been supported by AWS in Education Research Grant award.

8. REFERENCES

[1] M. Alzantot and M. Youssef. Crowdinside: automatic construction of indoor floorplans. In *SIGSPATIAL*, 2012.

[2] K. Asakura, M. Takeuchi, and T. Watanabe. A pedestrian–oriented map matching algorithm for map information sharing systems in disaster areas. *IJKWI*, 2012.

[3] J. Biagioni and J. Eriksson. Inferring road maps from global positioning system traces. *Journal of the Transportation Research Board*, 2012.

[4] U. Blanke, G. Tröster, T. Franke, and P. Lukowicz. Capturing crowd dynamics at large scale events using participatory gps-localization. *ISSNIP*, 2014.

[5] J. A. Bondy and U. S. R. Murty. *Graph theory with applications*. 1976.

[6] L. Cao and J. Krumm. From gps traces to a routable road map. In *SIGSPATIAL*, 2009.

[7] S. Edelkamp and S. Schrödl. Route planning and map inference with global positioning traces. In *Computer Science in Perspective*. Springer, 2003.

[8] D. Guo, S. Liu, and H. Jin. A graph-based approach to vehicle trajectory analysis. *Journal of Location Based Services*, 2010.

[9] T. Guo, K. Iwamura, and M. Koga. Towards high accuracy road maps generation from massive gps traces data. In *IGARSS*, 2007.

[10] P. Kasemsuppakorn and H. A. Karimi. A pedestrian network construction algorithm based on multiple gps traces. *Transportation research part C: emerging technologies*, 2013.

[11] E. Keogh, S. Chu, D. Hart, and M. Pazzani. Segmenting time series: A survey and novel approach. In M. Last, A. Kandel, and H. Bunke, editors, *Data Mining in Time Series Databases*, number 57 in Machine Perception and Artificial Intelligence, pages 1 – 22. World Scientific Publishing, Singapore, 2004.

[12] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang. Walkie-markie: indoor pathway mapping made easy. In *USENIX conference on Networked Systems Design and Implementation*, 2013.

[13] H.-X. T. Xiuming Zhang, Yunye Jin and W.-S. Soh. Cimloc: A crowdsourcing indoor digital map construction system for localization. In *ISSNIP*, 2014.