Automated semantic trajectory annotation with indoor point-of-interest visits in urban areas

Victor de Graaff
Dept. of Computer Science
University of Twente
Enschede, The Netherlands
v.degraaff@utwente.nl

Rolf A. de By
Fac. of Geo-Information Science
& Earth Observation (ITC)
University of Twente
Enschede, The Netherlands
r.a.deby@utwente.nl

Maurice van Keulen
Dept. of Computer Science
University of Twente
Enschede, The Netherlands
m.vankeulen@utwente.nl

ABSTRACT

User trajectories contain a wealth of implicit information. The places that people visit, provide us with information about their preferences and needs. Furthermore, it provides us with information about the popularity of places, for example at which time of the year or day these places are frequently visited. The potential for behavioral analysis of trajectories is widely discussed in literature, but all of these methods need a pre-processing step: the geometric trajectory data needs to be transformed into a semantic collection or sequence of visited points-of-interest that is more suitable for data mining. Especially *indoor* activities in *urban areas* are challenging to detect from raw trajectory data. In this paper, we propose a new algorithm for the automated detection of visited points-of-interest. This algorithm extracts the actual visited points-of-interest well, both in terms of precision and recall, even for the challenging urban indoor activity detection. We demonstrate the strength of the algorithm by comparing it to three existing and widely used algorithms, using annotated trajectory data, collected through an experiment with students in the city of Hengelo, The Netherlands. Our algorithm, which combines multiple trajectory pre-processing techniques from existing work with several novel ones, shows significant improvements.

Keywords

Trajectory analysis, algorithm, smartphone, GPS, point-of-interest

Categories and Subject Descriptors

[**Information Systems Applications**]: Spatial-temporal systems—*Geographic information systems*

1. INTRODUCTION

The places a person visits regularly are a strong indication for personal preferences and needs. A person visiting a kindergarten each morning most likely has a young child, and a person visiting a concert hall on a regular basis probably has a more than average interest in music. This makes trajectory data a promising source to determine personal preferences and needs, as discussed in [1]. Furthermore, trajectory data can be used to assess the trustworthiness of UGC, as discussed in [2]: a negative review of a restaurant that the user visited in the past week can probably be trusted more than a negative review of a person who works at a nearby competitor. Before we can perform such an analysis, we need to know which places were visited. We define a relevant place as a *point-of-interest* (POI): *a location where goods and services are provided, geometrically described using a point, and semantically enriched with at least an interest category.* The type of trajectory analysis that is used to extract visited POIs from raw trajectory data has been discussed in the literature as *stop detection*, and we follow the definition of Yan et al. for a *stop*: a temporary suspension of the travel for some reason [3]. In this work, we are particularly interested in those stops that take place at a pre-defined POI.

Smartphones these days base the location on multiple information sources: WiFi positioning, (assisted) GPS positioning and cell phone tower locations. In the algorithm presented in this paper, we focus on the detection of POI visits based on the *raw* smartphone data, which does not reveal from which source the location was derived. This POI visit detection is a challenging task, especially for indoor activities in urban areas, since trajectory data is often incomplete, due to temporary inability to receive GPS signals from enough satellites, or imprecise, due to signal multipath. In the words of Alvares et al.: "to transform a sample trajectory into a semantic one (sequence of stops and moves) is not an easy task" [4], or as Yan et al. put it: "dense urban areas can have several different POIs. (..) Such large number makes it probabilistically intractable to infer the exact POI from imprecise location records" [5, 6]. Nevertheless, several attempts have been undertaken to extract POI visits from trajectory data. There are generally two ways to do this. The most common way is to detect POIs from slow movement over a longer period, and defining the locations at which this happens regularly as the POI

ACM 978-1-4503-3738-0.
http://dx.doi.org/10.1145/0000000.0000000

set, as for example by Ashbrook et al. [7] or Zheng et al. [8]. A less common way is to match trajectories with a given, or specifically collected, POI set, such as the one by Alvares et al. [9]. The advantage of the latter approach, is that more information on the matched POI may already be available, such as a name, address, website, and POI category. This makes the approach with a given POI set more suitable for a semantic analysis of the visited POIs, which is why we focus on this form of POI visit extraction. However, existing approaches have two important drawbacks: (1) they assume the availability of the GPS signal while residing at the POI, and (2) they do not take the accuracy of the GPS samples into account. The first drawback leads to the non-detection of indoor POI visits, while the latter leads to false positives for imprecise signals, as discussed in more detail in Section 2.

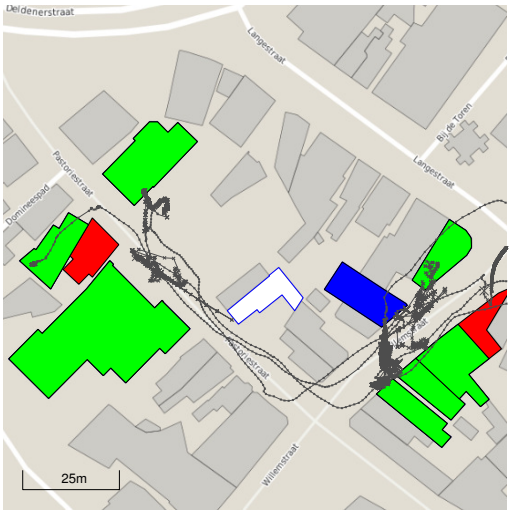


Figure 1: Trajectory annotation using Point-of-Interest Extraction (PIE). Green, blue and red polygons indicate the footprint of POIs that were extracted as visited. White and grey polygons were not extracted as visited. For green polygons proof was available of a visit, blue and white polygons were possibly visited, and red and grey polygons were definitely not visited in the real world.

In this paper, we present the trajectory annotation algorithm *Point-of-Interest Extraction* (PIE), illustrated in Figure 1, which is designed to overcome the aforementioned problems. In this figure, the circles indicate the trajectory samples. The green, blue and red areas indicate extracted POI visits. The PIE algorithm was designed especially for indoor activities, while using the GPS sensor of mobile devices. In this paper, we also present the results of a comparison with the existing algorithms of Alvares [9], Palma [10], and Rocha [11]. These three methods (especially the first two), are regularly used for POI extraction, due to their availability as an extension for the data mining tool Weka [4, 11]. For this comparison, we set up an experiment with students of the University of Twente, who used their own mobile devices as they were invited to visit pre-defined POIs as part of a treasure hunt game in the city of Hengelo, The Netherlands.

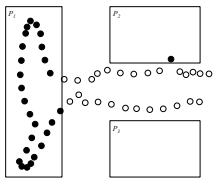
The remainder of this paper is structured as follows: Section 2 provides an overview of related work. The details of our approach are laid out in Section 3. Our validation method and results are presented in Section 4. Privacy considerations are discussed in Section 5. Section 6 finally, concludes this paper.

2. RELATED WORK

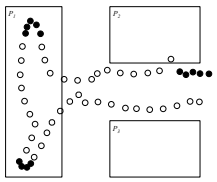
In this section, we focus on the explanation and illustration of the approaches with which we compare our work: the approaches of Alvares, Palma, and Rocha, who (in collaboration with each other) tried to solve the same problem as we are, using three different approaches. We conclude this section with a short discussion of other related work.

2.1 Alvares's IB-SMoT

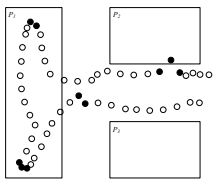
Alvares's *Intersection-Based Stops and Moves of Trajectories* (IB-SMoT) [4, 9] is the most straightforward of the three approaches: it intersects the trajectories with the polygons that represent the POIs, as illustrated in Figure 2a. The trajectory is split up into intervals that are either a *stop* at a POI, or a *move* between POIs. Those intervals during which the trajectory intersects with a known polygon are marked as a stop (illustrated by filled trajectory points in Figure 2a), and those intervals during which the GPS samples do not intersect with such a polygon are annotated as a move (illustrated by empty trajectory points). Besides a trajectory and a set of disjoint relevant polygons, this approach takes no input parameters. Drawbacks of Alvares's approach are: (1) distortion of the GPS signal, for example due to signal multipath, can easily lead to false positives, (2) absence of the GPS signal during indoor activities is not taken into account, and (3) the accuracy indicator of GPS signals is ignored, causing imprecise signals to be interpreted with equal importance as precise ones.



(a) Alvares's IB-SMoT



(b) Palma's CB-SMoT



(c) Rocha's DB-SMoT

Figure 2: Existing POI extraction approaches: circles represent trajectory points, filled circles represent trajectory points with significance in the illustrated approach

2.2 Palma's CB-SMoT

Palma's *Clustering-Based Stops and Moves of Trajectories* (CB-SMoT) [10], illustrated in Figure 2b, is based on a variation of the well-known clustering algorithm *DBSCAN* (Density-Based Spatial Clustering of Applications with Noise) [12]. Rather than setting the clustering parameter *Eps* (used to determine the maximum allowed distance between points before starting a new cluster) for *all* trajectories to the same value, they introduce a *quantile function* that takes an input parameter *area*, and uses this to automatically determine a proper setting for *Eps* per trajectory. In this way, trajectories with faster movement have distance threshold values that are higher than the trajectories of slower moving ones. Palma et al.'s approach also requires the input parameter *minTime*, a minimum time threshold that is used as an alternative for DBSCAN's minimum number of points to create a cluster. The clustered trajectory parts (illustrated by filled trajectory points) are then intersected with the known POI polygons. Because of the clustering approach, a reflecting signal while moving past a POI is generally discarded. However, the second and third drawback of Alvares's approach also apply here: due to the absence of GPS signals inside buildings, the clusters are often located outside the POIs parcel polygon, and cluster building is disturbed by imprecise signals.

2.3 Rocha's DB-SMoT

Rocha's *Direction-Based Stops and Moves of Trajectories* (DB-SMoT) [11], illustrated in Figure 2c, is based on the notion of heading change: if the direction of a trajectory frequently changes, this indicates a *stop* (illustrated by filled trajectory points). Besides a trajectory, their algorithm takes three parameters as input: the minimum direction change *minDC*, the minimum time interval to build a new cluster *minTime*, and the maximum tolerated consecutive number of trajectory samples that do not exceed *minDC* inside a cluster: *maxTol*. Rocha et al. validated their method using the fishing locations of Brazilian vessels, for which the POI areas are relatively large compared to the inaccuracy of the GPS signal. In the application for indoor urban activity detection, the location accuracy is much lower, as the receiver often cannot pick the optimal satellite constellation. Furthermore, just like in the other two approaches, knowledge of the signal's accuracy is not taken into account, causing inaccurate points to be regarded relevant for this geometric analysis.

2.4 Other related work

Besides POI visit extraction, other types of trajectory annotation have received their share of attention. Stefanakis showed for example, how trajectories can be annotated with geometric properties [13]. These properties can then be used for trajectory simplification, assisting us to select the relevant sampling points from the trajectories. An example of a useful trajectory simplification metric is the one from Chen et al. [14], which we discuss in the next section as a part of our approach. Guc et al. also annotate trajectories with POI visits [15], but this annotation is done manually. SeMiTri of Yan et al. annotates trajectories automatically, but since they find it probabilistically intractable to infer the exact POI, they annotate the trajectories with properties of the region where the trajectory was created [6]. Another algorithm that could be used for POI visit detection is *Continuous Nearest Neighbor* (CNN) Search by Tao et al. [16]. However, since by definition, their approach continuously picks the nearest POI, this approach leads to many false positives, albeit for very short time intervals. The CNN approach would therefore need to be extended with a filter of those POIs that should be marked as visited, and those that were simply nearby for a shorter amount of time. However, in this paper we prefer to focus on the comparison with existing algorithms, rather than adaptations of existing algorithms, and therefore we consider it outside the scope of this paper. Kang et al. [17], to conclude, came up with a time-based clustering approach based on WiFi positioning purely for POI visit detection. As stated in the introduction however, we focus on POI visit detection from smartphone trajectory data, which may come from multiple sources.

3. APPROACH

Our PIE algorithm is mainly based on four parameters from the smartphone location sensor readings and its spatio-temporal derivatives: the accuracy of the location samples as provided by the smartphone OS, reductions in speed, changes in direction and projection of signals onto parcel polygons. We begin this section by defining several geometric concepts that we use as selection filters, and then present the algorithm that combines these.

3.1 Definitions

Trajectory and trajectory sample point

The location signal in a mobile device is only accurate up to several meters (see for example [18]), depending on several factors, such as the vicinity of known WiFi networks, the quality of the GPS sensor, the number of detected satellites by the GPS sensor, their spatial arrangement in the sky, and the presence of reflecting surfaces. This is modeled in mobile devices using an accuracy parameter, measured in meters, which indicates the radius around the point in which the device may also be located. Even though the APIs of the large manufacturers have named this an *accuracy* parameter, we prefer to refer to this as an *inaccuracy* indicator, as its value increases with increasing inaccuracy.

DEFINITION 1. *A* trajectory sample point *is a spatio-temporal point, associated with an inaccuracy indication, represented by a tuple* $p = (x, y, t, i)$.

We express x, y, and i in meters, and t in seconds. Other units suitable for distance and time calculation can be used as well, as long as the thresholds discussed below are set accordingly.

DEFINITION 2. *A* trajectory *is a chronological sequence of sample points:* $T = \langle p_0, \ldots, p_n \rangle$.

Since trajectory samples with a high value for the inaccuracy indication are unsuitable for geometric calculations, we also introduce the *inaccuracy threshold* value i_{max}.

Staypoint

Li et al. introduced the concept of a *staypoint* [19]; it is illustrated in Figure 3. Zheng elaborated on this in [8], from

which we follow the definition: a geographical region where a user stayed over a time threshold T_r within a distance threshold of D_r. However, the definitions and algorithms of Li and Zheng do not take maximality into account. This causes multiple staypoints are created, based on sequences that contained each other, as for example is the case for p_4 and p_5 in Figure 3. To avoid this, and let a trajectory sample , we follow a slightly different definition.

DEFINITION 3. *A* candidate stop sequence *C is a non-empty subsequence of a trajectory T for which $p = \langle p_m, \ldots, p_n \rangle$, for which $\forall m < i \leqslant n$, $Dist(p_m, p_i) \leqslant D_r$, $Dist(p_m, p_{n+1}) > D_r$ or $n+1 \geq |T|$, and $Int(p_m, p_n) \geqslant T_r$, where Dist is the geospatial distance between two sample points and Int is the time difference between two sample points, $|p_i.t - p_j.t|$.*

DEFINITION 4. *A candidate stop sequence C of a trajectory T is a* stop sequence *S if and only if no other candidate stop sequence C' exists for T that contains all samples contained in C.*

DEFINITION 5. *A* stay point *s is defined as the centroid of a stop sequence S. s is a tuple (x, y, t_a, t_l, i) where x and y are the coordinates, t_a is the arrival time, t_d is the time of departure, and i is the maximum inaccuracy:*

(x, y) *is the centroid of* S,
$t_a = \min_{p \in S}(p_i.t)$,
$t_d = \max_{p \in S}(p_i.t)$,
$i = \max_{p \in S}(p_i.i)$.

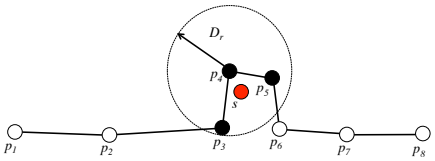


Figure 3: A trajectory, where three trajectory points $\langle p_3, p_4, p_5 \rangle$ form a staypoint s (illustration after [8])

Heading change

Chen et al. introduced several forms of *heading change* for their trajectory simplification method proposed in [14]. We adopt their concept of *neighbor heading change*, which corresponds with the definition of *direction change* used by Rocha in [11]. The *heading change* is the change in heading that takes place at a point p_i with respect to its direct neighbor points in the trajectory:

DEFINITION 6. *The* heading change *θ for a point n in the trajectory T is given by:*

$$\theta_i = \frac{180}{\pi} cos^{-1}\left(\frac{a \cdot b}{||a|| ||b||}\right)^{\circ},$$

where $a = \begin{bmatrix} p_i.x - p_{i-1}.x \\ p_i.y - p_{i-1}.y \end{bmatrix}$, $b = \begin{bmatrix} p_{i+1}.x - p_i.x \\ p_{i+1}.y - p_i.y \end{bmatrix}$.
For the first or last point in a trajectory, the heading change is defined as $\theta = 0^{\circ}$.

Where the algorithm of Rocha et al. looks for repeatedly changing directions, we are only interested in the direction change at the selected point. Small direction changes in a trajectory normally indicate a person travelling, while arriving at their destination, the movement deviates from this to enter their destination. Of course, people may have many other reasons to change direction, such as following a road network. Therefore, filtering based on direction changes alone does not suffice, but it is merely one of the features in our approach to filter out points. To extract those points where people change direction, we use a *direction change threshold* θ_{min}, where $0 \leqslant \theta_{min} \leqslant 180^{\circ}$. Note that a simplified trajectory is still a trajectory, and this definition therefore allows us to filter out sample points prior to applying this threshold to filter out points.

Parcel polygons and polygons-of-interest

We define a *polygon-of-interest* (POLOI) as *a location where goods and services are provided, geometrically described using a polygon, and semantically enriched with at least an interest category.* A *parcel polygon* is similar to a POLOI, but does not necessarily describe a location where goods and services are provided, but may also be another type of parcel, such as a residence. In [20], we discussed how a POLOI can be estimated given a POI set using open data. In the algorithm presented in this paper, we assume the availability of polygons for *all* nearby parcels as the polygon set P. Those elements in P that also represent a POLOI are contained in a subset of P, referenced as P_{POLOI}. The result of the PIE algorithm, which is the *set of visited POLOIs*, is denoted as P_{VP}.

Polygon projection and maximum projection distance

GPS signals are typically unavailable inside buildings. Therefore, we use selected points, derived from the GPS signal right before or right after a stop, to determine which place has been visited. These points are projected onto the nearest parcel polygon. We call this *polygon projection.* To limit the projection distance, we introduce the absolute *maximum projection distance* π_{max} within which the point is still considered indicative of a parcel visit.

3.2 Algorithm

In our PIE algorithm, we combine the concepts discussed above. First, we filter out those points for which the inaccuracy value exceeds the accuracy threshold i_{max}. These points are too imprecise to do further calculation with. Secondly, we extract the staypoints from a trajectory. Then, we determine the direction change between the staypoints to determine whether any staypoint is an indication for a visited location or should be attributed to natural behavior in traffic. Thereafter, we project the selected points onto the nearby polygons, taking the maximum projection distance π_{max} into account. Those polygons that are POLOIs, are added to the result set of visited polygons that are not POLOIs. Note that, unlike in the work of Palma and Chen, the heading change is taken into account *after* staypoint detection: the advantage is that this filters out those staypoints that were on a relatively straight path with respect to the previous and next staypoints. This leads to the formalization of our algorithm shown in Algorithm 1.

4. VALIDATION

We begin this section with a description of the validation approach, followed by the way we collected and cleaned the ground truth data. Then, we discuss the metrics used to evaluate how our algorithm performs in comparison to ex-

Algorithm 1 Point-of-Interest Extraction (PIE)

input: T // trajectory as sequence of sample points
i_{max} // maximum inaccuracy in meters
D_r // staypoint distance threshold in meters
T_r // staypoint time threshold in seconds
θ_{min} // minimum direction change in degrees
π_{max} // maximum projection distance in meters
P // set of all existing polygons
P_{POLOI} // subset of P, containing POLOIs
output: P_{VP} // subset of P_{POLOI}, containing vst. POLOIs

procedure PIE
  // Select points based on inaccuracy
  $accuratePoints \leftarrow \{\, p \in T \mid p.i \leq i_{max} \}$;
  // Select points based on staypoints
  $stayPts = stayPointDetection(accuratePoints, T_r, D_r)$;
  // Select points based on heading change
  $selectedPoints \leftarrow \{\, p \in stayPts \mid$
    $headingChange(stayPts, p) \geq \theta_{min} \}$;
  // Select nearest polygon, apply π_{max}
  // and check if it is a POLOI
  $P_{VP} \leftarrow \{\, p \in selectedPoints \mid$
    $polygon = getNearestPolygon(P, p)$
    $\wedge\, dist(polygon, p) \leq \pi_{max}$
    $\wedge\, polygon \in P_{POLOI} \}$;
  return P_{VP}
end procedure

isting algorithms. Finally, we present the results of the validation with a short discussion.

4.1 Approach

To validate our approach, we rebuilt the approaches of Alvares, Palma and Rocha, based on their papers. Just like our approach, the approaches by Palma and Rocha require several parameters to be set. For these three approaches, we used a ten-fold cross validation to detect the proper parameter settings using a brute force approach. To limit the number of combinations, we set the parameters $i_{max} = 10$m, $\pi_{max} = 10$m, and $\theta_{min} = 30^\circ$ for the PIE algorithm, while detecting T_r and D_r automatically.

4.2 Data collection

To collect validation data, we set up an experiment similar to the one described in [2]. During the welcome week of the University of Twente, called the *Kick-In*, the new students were invited to participate in a treasure hunt-like game that lasted four hours. The students were supposed to carry out exercises, and upload a picture of this activity with the specifically designed mobile application *Kick-In Discover Hengelo*, illustrated by the screenshot in Figure 4. The exercises could be carried out in any given order, and the best picture was rewarded with a prize.

The application, which was built on the PhoneGap platform [21], was available on both Android phones and iPhones. Ten of the 24 exercises had to be carried out at a specific POI. The employees of these POIs prepared a set-up for the respective exercise, such as a poker table, or a karaoke set. The organization of the event, which we had collaborated with while designing the app, had the possibility to send messages to all users at once. They used this to motivate the students to move on to the next exercise roughly every half hour. In the background of the application, the trajectory was sampled with explicit prior user consent. The pictures taken by the students, were linked to their trajectory and formed a proof of their visit to that specific POI. We used this proof as a manual ground truth POI visit annotation of the trajectories.

An unforeseen, yet important side-effect of this data collection, is that the resulting annotation is not complete: students did not upload a picture from their own device at every location they went to, but at several occasions, one picture was uploaded with several people on it. We address this when discussing the validation metrics in Section 4.4.

(a) For each exercise the students were invited to upload a picture

(b) A map helped the students to move between participating POIs

Figure 4: Screenshots from the *Kick-In Discover Hengelo* app

For the polygon sets P (and thus P_{POLOI}), we used authoritative data from a public web feature service, offered by the Dutch government through the open data initiative *Nationaal Georegister*[1]. To thoroughly test the algorithms for false positives, we assumed *all* parcel polygons in the vicinity of the event to be POLOIs, and thus in this case: $P = P_{POLOI}$.

4.3 Data cleaning

Several pictures were uploaded outside the time window of the event, due to people playing around with the app before and after the event. Furthermore, several pictures did not meet the exercise requirements. For example, exercises were carried out while not being at the correct location, or the picture was entirely black. These pictures were discarded as annotations after manual inspection. Trajectory samples that were not within the time window of 2 minutes before the user's first annotation and 2 minutes after his last annotation were discarded as well. Trajectories that did not contain at least 50 samples with an inaccuracy value below

[1] http://www.nationaalgeoregister.nl

30 were also excluded from the validation data, as these students most likely did not have GPS positioning turned on. This resulted in 23 valid trajectories from a wide range of device types, containing a total of 30,500 trajectory samples and 128 annotations.

4.4 Validation metrics

Our validation metrics are based on those that are commonly accepted in *information retrieval*: *precision*, *recall*, and their harmonic mean *F measure*. These metrics are indications for the relationship between the numbers of *true positives* (TP), *false positives* (FP), *true negatives* (TN) and *false negatives* (FN). Since the annotations were not entirely complete (as discussed above), we introduce a new category: *probable positives* (PP). These are the locations where students may have been, since these POIs participated in the event, but for which we do not have proof in the form of a (valid) photo. We extend the notion of *precision*, the fraction of retrieved POIs that are relevant, accordingly by interpreting PPs as TPs:

$$OptimisticPrecision = \frac{TP + PP}{TP + PP + FP}$$

We also use the more conservative *pessimistic precision*, in which we consider PPs as FPs:

$$PessimisticPrecision = \frac{TP}{TP + PP + FP}$$

For the fraction of relevant POIs that are retrieved, or *recall*, we use the regular formula:

$$Recall = \frac{TP}{TP + FN}$$

For our validation, we consider precision and recall equally important, in which case the optimistic and pessimistic F measures are formally given as:

$$OF = \frac{2 * OptimisticPrecision * Recall}{OptimisticPrecision + Recall}$$

and

$$PF = \frac{2 * PessimisticPrecision * Recall}{PessimisticPrecision + Recall}$$

4.5 Results

In Figure 5, we illustrate POI visits as extracted by the different approaches for the same trajectory. A green polygon indicates a true positive for POI visit extraction, a blue polygon a probable positive, a red polygon a false positive, a yellow polygon a false negative, and grey polygon a true negative. White polygons indicate locations that participated in the event, but were not visited by this student.

The straightforward intersection-based approach by Alvares in Figure 5b leads to many false positives, due to the inclusion of location samples where signal multipath was in play. Similarly, Palma's clustering-based approach in Figure 5c suffers from this: detected clusters are often located in front of the POI, rather than intersecting with the polygon. Without a projection on the nearest polygon, this leads to many false negatives. The direction-based approach of Rocha in Figure 5d suffers from the constant reflection of signals in urban areas: many clusters are detected due to inaccuracies of the signal, rather than actual back-and-forth movement. As a result, clusters are created that were caused by signal multipath.

In Figure 6, we show the results of our validation aggregated over all valid trajectories. PIE outperforms the three existing approaches in pessimistic precision, optimistic precision, and recall, and as a result in F measure as well. We are able to extract visited POIs with a precision of 57.9%, classical precision of 44.7%, and recall 68.0% of the visited POIs. This corresponds with an optimistic F measure of 0.625, substantially higher than the optimistic F measures for Alvares (0.441), Palma (0.371), and Rocha (0.443). The pessimistic F measure for PIE equals 0.540, also substantially higher than those for Alvares (0.335), Palma (0.331), and Rocha (0.344).

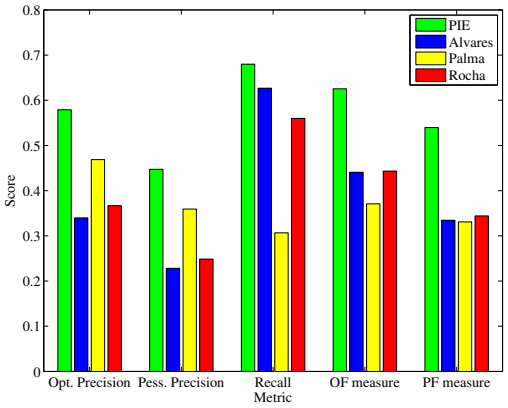


Figure 6: Validation results using described metrics

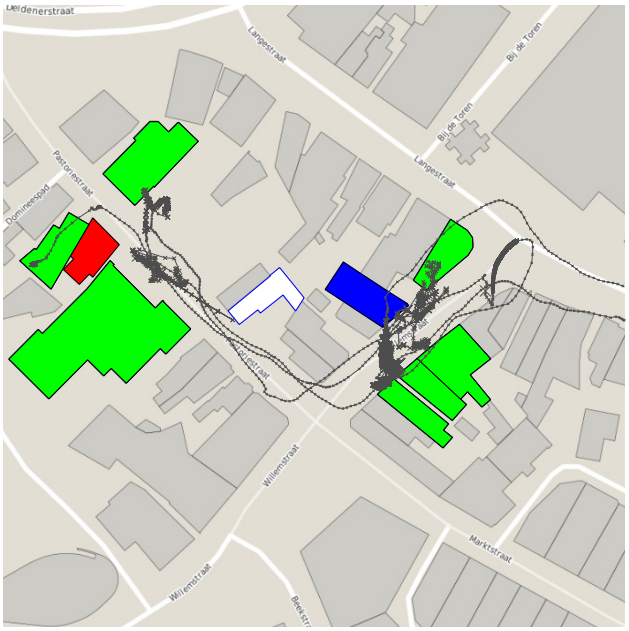
To give an indication of appropriate parameter settings for the PIE approach: nine validation batches from the 10-fold cross validation used a distance threshold $3m \leqslant D_r \leqslant 5m$ and a time threshold $11s \leqslant T_r \leqslant 14s$. We also experimented with a relative value for the π_{max} parameter, that relates the projection distance to the distance to the second nearest parcel, but this led to significantly inferior results. Another idea we experimented with was to take signal loss into account, but since this can be caused by several factors, this turned out to be a rather weak indicator for parcel or POI visits.

5. PRIVACY CONSIDERATIONS

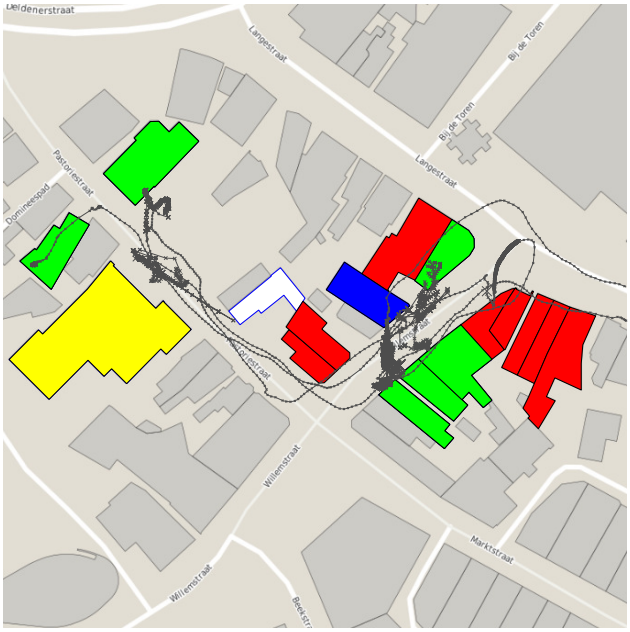
With great power, comes great responsibility. Algorithms like ours can be used to the benefit of a user, just as well as to his discomfort. Users are often careless about the permissions that a mobile application requires. We consider it the application developer's moral obligation to reduce the collection of private data to a minimum. Therefore, we share this algorithm with the following three privacy considerations:

Information ownership

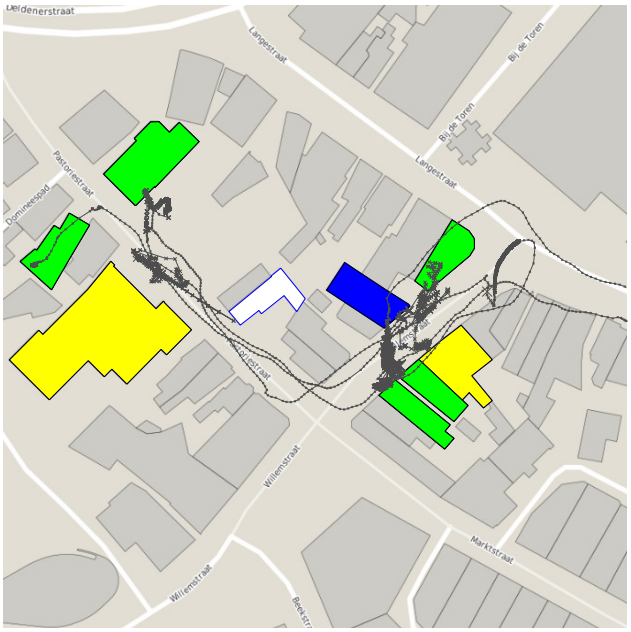
Trajectory data is personal data that belongs to a user. Therefore, which POI a person visited is information that shall be available to only that person, unless he agrees to share this information. Extracted POI visits shall remain on the device of the user. Only an aggregated profile, based on numerous POI visits, shall be sent to a service that uses



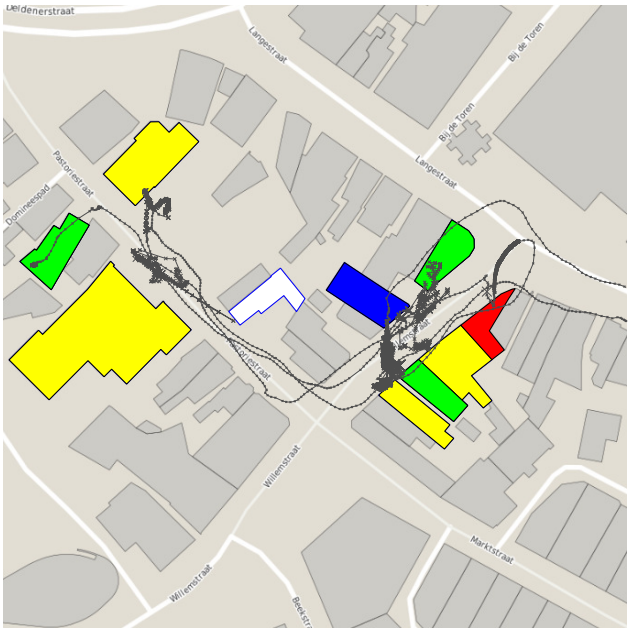
(a) PIE: clusters outside POLOIs still lead to POI/POLOI visit detection



(b) Alvares: many false positives due to simple intersection



(c) Palma: missed polygons due to location of clusters outside POLOIs



(d) Rocha: missed polygons due to location of clusters outside POLOIs; going around a corner also causes cluster creation

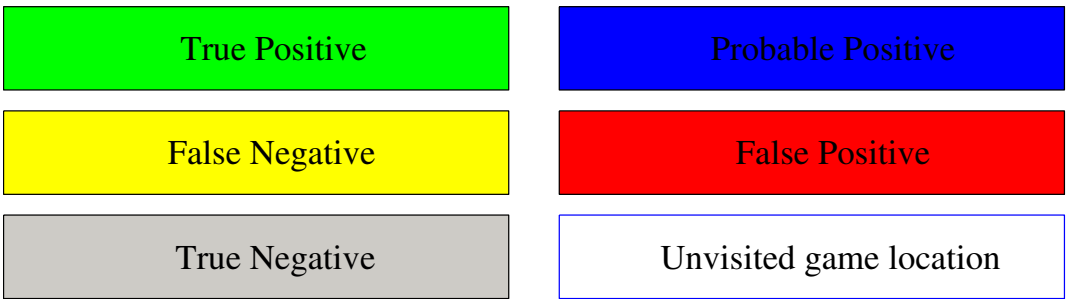


Figure 5: POI extraction from trajectory #110 using different algorithms.

these. For example: the fact that a user has children, can be beneficial to offer the right services at the right time. However, where his children go to school, is too personal to share.

Informed user consent
For certain services, it is required to know the *exact* POI where a person is currently located. One can think of a tag for social media, or, more importantly, an emergency call application for an elderly person. In this case, we consider it reasonable to share the exact POI, but only with the *consent of the informed user*.

POI set scope
POI sets shall contain only those POIs that increase the quality of the service. The scope of object types contained in the POI set shall be carefully selected for each application, and shall certainly not contain location types that a person may not be willing to share in public.

6. CONCLUSION & FUTURE WORK

In this paper, we introduced and validated our POI visit extraction algorithm PIE. Using a combination of several geometric and spatiotemporal processing steps, we are able to infer the visited POIs with significantly better results than those of existing approaches, as this algorithm is specifically designed for urban indoor trajectory analysis. Even for those trajectories for which typical challenges for this type of trajectory analysis, such as signal loss and inaccuracy, play a role, PIE manages to retain a combination of high precision and high recall. In the near future, it is our aim to combine this algorithm with POI collection techniques and POI-to-POLOI transformation techniques to create a holistic framework to transform trajectories into user profiles.

7. ACKNOWLEDGEMENTS

This publication was supported by the Dutch national program COMMIT/.

8. REFERENCES

[1] V. de Graaff, M. van Keulen, and R. A. de By, "Towards geosocial recommender systems," in *4th Intern. Workshop on Web Intelligence & Communities (WI&C 2012), Lyon, France*, ACM, 2012.

[2] V. de Graaff, D. Pfoser, M. van Keulen, and R. A. de By, "Spatiotemporal behavior profiling: A treasure hunt case study," in *Proc. of Web & Wireless GIS*, Springer Verlag, 2015.

[3] Z. Yan, J. Macedo, C. Parent, and S. Spaccapietra, "Trajectory ontologies and queries," *Transactions in GIS*, vol. 12, no. s1, pp. 75–91, 2008.

[4] L. O. Alvares, A. Palma, G. Oliveira, and V. Bogorny, "Weka-STPM: from trajectory samples to semantic trajectories," in *Proceedings of the XI Workshop de Software Livre, WSL*, vol. 10, pp. 164–169, 2010.

[5] Z. Yan, D. Chakraborty, C. Parent, S. Spaccapietra, and K. Aberer, "SeMiTri: a framework for semantic annotation of heterogeneous trajectories," in *Proc. of the 14th Int. Conf. on EDBT*, pp. 259–270, ACM, 2011.

[6] Z. Yan, D. Chakraborty, C. Parent, S. Spaccapietra, and K. Aberer, "Semantic trajectories: Mobility data computation and annotation," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 4, no. 3, p. 49, 2013.

[7] D. Ashbrook and T. Starner, "Using GPS to learn significant locations and predict movement across multiple users," *Personal and Ubiquitous Computing*, vol. 7, no. 5, pp. 275–286, 2003.

[8] Y. Zheng and X. Xie, "Learning location correlation from GPS trajectories," in *11th Intern. Conf. on Mobile Data Management*, pp. 27–32, IEEE, 2010.

[9] L. O. Alvares, V. Bogorny, B. Kuijpers, J. A. F. de Macedo, B. Moelans, and A. Vaisman, "A model for enriching trajectories with semantic geographical information," in *Proceedings of the 15th Int. Conf. on Adv. in GIS*, p. 22, ACM, 2007.

[10] A. T. Palma, V. Bogorny, B. Kuijpers, and L. O. Alvares, "A clustering-based approach for discovering interesting places in trajectories," in *Proceedings of the 2008 ACM symposium on Applied computing*, pp. 863–868, ACM, 2008.

[11] J. A. M. Rocha, G. Oliveira, L. O. Alvares, V. Bogorny, and V. C. Times, "DB-SMoT: A direction-based spatio-temporal clustering method," in *Intelligent systems (IS), 2010 5th IEEE international conference*, pp. 114–119, IEEE, 2010.

[12] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise.," in *KDD*, vol. 96, pp. 226–231, 1996.

[13] E. Stefanakis, "SELF: Semantically enriched line simplification," *Int. Journal of GIS*, pp. 1–19, 2015.

[14] Y. Chen, K. Jiang, Y. Zheng, C. Li, and N. Yu, "Trajectory simplification method for location-based social networking services," in *Proc. of the 2009 Int. WS on LBSN*, pp. 33–40, Citeseer, 2009.

[15] B. Guc, M. May, Y. Saygin, and C. Körner, "Semantic annotation of GPS trajectories," in *11th AGILE international conference on GIS*, 2008.

[16] Y. Tao, D. Papadias, and Q. Shen, "Continuous nearest neighbor search," in *Proceedings of the 28th Int. Conf. on VLDB*, pp. 287–298, 2002.

[17] J. H. Kang, W. Welbourne, B. Stewart, and G. Borriello, "Extracting places from traces of locations," in *Proc. of the 2nd ACM Int. WS on Wireless mobile appl. and services on WLAN hotspots*, pp. 110–118, ACM, 2004.

[18] P. A. Zandbergen, "Accuracy of iPhone locations: A comparison of assisted GPS, WiFi and cellular positioning," *Trans. in GIS*, vol. 13, pp. 5–25, 2009.

[19] Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W.-Y. Ma, "Mining user similarity based on location history," in *Proc. of the 16th Int. Conf. on Advances in GIS*, SIGSPATIAL '08, pp. 34:1–34:10, ACM, 2008.

[20] V. de Graaff, R. A. de By, M. van Keulen, and J. Flokstra, "Point of interest to region of interest conversion," in *Proc. of the 21st Int. Conf. on Advances in GIS*, SIGSPATIAL'13, pp. 388–391, ACM, 2013.

[21] S. Allen, V. Graupera, and L. Lundrigan, "PhoneGap," in *Pro Smartphone Cross-Platform Development*, pp. 131–152, Springer, 2010.