J Ambient Intell Human Comput
DOI 10.1007/s12652-017-0550-0

ORIGINAL RESEARCH

Making tourist guidance systems more intelligent, adaptive and personalised using crowd sourced movement data

Anahid Basiri[1] · Pouria Amirian[2] · Adam Winstanley[3] · Terry Moore[4]

Received: 25 April 2016 / Accepted: 11 July 2017

Abstract Ambient intelligence (AmI) provides adaptive, personalized, intelligent, ubiquitous and interactive services to wide range of users. AmI can have a variety of applications, including smart shops, health care, smart home, assisted living, and location-based services. Tourist guidance is one of the applications where AmI can have a great contribution to the quality of the service, as the tourists, who may not be very familiar with the visiting site, need a location-aware, ubiquitous, personalised and informative service. Such services should be able to understand the preferences of the users without requiring the users to specify them, predict their interests, and provide relevant and tailored services in the most appropriate way, including audio, visual, and haptic. This paper shows the use of crowd sourced trajectory data in the detection of points of interests and providing ambient tourist guidance based on the patterns recognised over such data.

✉ Anahid Basiri
a.basiri@southampton.ac.uk

Pouria Amirian
pouria.amirian@os.uk

Adam Winstanley
adam.winstanley@mu.ie

Terry Moore
terry.moore@nottingham.ac.uk

[1] Department of Geography and Environment, The University of Southampton, Southampton, UK

[2] Ordnance Survey, Southampton, UK

[3] Department of Computer Science, Maynooth University, Maynooth, Ireland

[4] The Nottingham Geospatial Institute, The University of Nottingham, Nottingham, UK

Keywords Ambient services · Tourist guidance · Trajectory data mining · Touristic point of interest (PoI) · Spatio-temporal data

1 Introduction

According to the United Nations World Tourism Organization, over the last 3 years, the developed countries including Germany, the US, and China, exhibited the strongest growth rates and they had the highest international tourism expenditures. Despite this high growth rate, there are several challenges still tourism industry is dealing with. They include: offering information in several languages, providing more personalised suggestions and offers, offering more value to the user, and retaining your customers. In this regard, crowd sourced gather data, ambient intelligence (AmI) and mobile services can contribute significantly.

Most people enjoy the subject of travel, and it's easy to find active mobile users in the crowd who are happy to participate in projects on the topic. The participation of the crowd helps to give more realistic and up-to-date information about destinations already visited. This paper proposes a framework to use the crowd-sourced data for the purpose of tourist guidance and ambient services, in particular suggestion-making services. Crowd-sourced trajectory data is analysed to extract the sites/points of interest and attractive paths and places. The use of crowd trajectories can provide more up-to-date destination suggestions minimising the direct effect of commercialised advertisement, which might not reflect the public interest.

The contribution of the ambient intelligence (AmI) in travel and tourism related an application is also significant. This is mainly because the visitors, travellers, and the tourists are not usually very familiar with the visiting areas and

Published online: 01 September 2017

therefore they need an assistive, personalized, adaptive, ubiquitously aware of the location, and intelligent service (Manes 2002).

This paper shows the use of crowd-sourced trajectories of movements in an art gallery to analyse and recognise the patterns of movement, identify the point of interests (PoIs) using spatio-temporal data mining techniques and use the recognised patterns and PoI in more personalised, adaptive and intelligent recommender and suggestion making services. There are several research projects investigating the use of AmI in tourist assisted services (Huang et al. 2015; Umanets et al. 2014; Wang et al. 2016; Nakamura et al. 2015; Kabassi 2013; Sorrentino et al. 2015; Smirnov et al. 2013; Xu and Ke 2015; Tom Dieck and Jung 2015), there is still a need to use the patterns of volunteers' trajectories of movements as there are some information hidden within the data. However as reviewed by Yuan et al. (2016), majority of data mining techniques for the application of smart tourist services, are limited to online multimedia, such as travel (Vu et al. 2015), Check-in data (Lu et al. 2012), rather than movement behaviour and trajectories of travels. (Abramson et al. 2014) used the trajectories of Flickr geo-tagged photos to identify the points of interest. And Yuan et al. (2016) used the blog data mining to summarise and categorise tourist information. This paper uses the trajectories of movements, both indoors and outdoors, to extract knowledge about the interests, preferences, and demands of tourists. For example, some individuals skip particular paintings in a gallery and spend more time in some sections, as their interest could be different from the others. If such patterns are identified, the same suggestions can be made to a new individual who starts with the same interest area. In addition to this, the use of crowd-sourced data allows us to cluster tourists based on common interest without distracting them by keep asking preference-related questions. Their needs and demands can be automatically predicted if the shared patterns are available. Also, there is some additional information that can be provided to the visitor, with respect to the PoIs, if the interest and preferences are identified. In general tourist guidance, is one of the applications where AmI can have a great contribution to the quality of the service, as the tourists, who may not be very familiar with the visiting site, need a location-aware, ubiquitous, personalised and informative service. Such services should be able to understand the preferences of the users without requiring the users to specify them, predict their interests, and provide relevant and tailored services in the most appropriate way, including audio, visual, and haptic.

To identify such patterns and clusters, spatio-temporal data mining techniques (Cressie and Wikle 2015) can be very helpful tools. The spatio-temporal data mining techniques can identify the patterns hidden in data, detect the similarities and anomalies, and find classes and clusters. This paper focuses on using spatio-temporal data mining, and in particular trajectory mining techniques, to identify PoIs using anonymous trajectories of movements and use the PoIs in some applications including Tourist guidance and suggestion making services, and navigation services.

In this regard, trajectory data, without any reference to a moving user's identity, should be stored. It is possible to use many different anonymisers to make sure that anonymity of data is preserved; in this project, K-anonymity (Kalnis et al. 2007) has been applied. The contributors of the trajectories can simply upload their Global Positioning System (GPS) tracks or draw and store their movements' polylines on the maps. It is also possible to capture the trajectories indirectly from other resources, including CCTVs, Global Navigation Satellite System (GNSS) embedded in mobile phones or vehicles, accelerometers and so on. Having a huge amount of input data from a variety of resources makes the data management and storage challenging. Due to the volume of the input data, and also the variety of data resources and consequently the different schema for the datasets, the trajectory analysis should be done using big data analytics techniques. Therefore the input data is stored in a Graph database, Neo4j, to have more efficient trajectory data management and analysis trajectory elements (nodes and edges) in comparison with other solutions (Amirian et al. 2015). In order to extract patterns and identify rules, an inference engine is developed to apply big data mining techniques to detect anomalies, identify clusters and classify data (Baiget et al. 2008), and then extract patterns and rules (Kuntzsch and Bohn 2013; Zhang et al. 2013). Such information and knowledge can be used in pathfinding and routing decision-making and many other navigational suggestion-making applications.

This paper is structured as following; the next section discusses the nature of the trajectory data, i.e. trajectories, data preparation and management (in graph databases), then it discusses the trajectory data mining techniques. And Sect. 3 implements the data mining techniques and inference system with application to tourist guidance.

2 Trajectory mining

Data mining, as a field at the intersection of computer science and statistics, is a field, which attempts to discover patterns from large datasets. It utilizes methods at the intersection of artificial intelligence, machine learning, statistics, and database systems (Zheng 2015). The overall goal of the data mining process is to extract information from a dataset and transform it into an understandable structure for further use. Data mining involves six common classes of tasks:

1. Anomaly detection (outlier/change/deviation detection)—the identification of unusual data records, that might be potential errors or anomalies
2. Association rule learning (dependency modelling)—searches for relationships between variables. Using association rule learning, one can determine which variable are more related to another. This is sometimes referred to as market basket analysis.
3. Clustering—is the task of discovering groups and structures in the data that are in some way or another "similar", without using known structures in the data.
4. Classification—is the task of generalizing known structure to apply to new data.
5. Regression—attempts to find a function which models the data with the least error.
6. Summarization—providing a more compact representation of the dataset, including visualisation and report generation

In order to identify the patterns of movements among tourists' trajectories and recognise PoIs for recommender systems, several steps to be taken to accomplish the trajectory data mining process, see Fig. 1. The earliest steps in trajectory data mining process are data capture and preparation, which are explained in the next subsection.

2.1 Input data: trajectories

A trajectory is the trace of movement of a moving object. The moving object can be a pedestrian, driver, a group of animals, a natural phenomenon like a tornado, that moves in space and time (Mazimpaka and Timpf 2016). Trajectories can have several components, however location and time are the main components of trajectory data.

There are some preconditions that should be satisfied to make sure that the results are valid and are not based on incomplete input datasets. If the input dataset is small (for example in terms of the number of samples and the extent and density/frequency of trajectories with respect to space and time), then outputs (the rules and patterns) may be valid only for a specific situation, area, and time interval. The level of robustness of output knowledge is strongly correlated with input sample size. In this regard, for successful pattern recognition and rule extraction there are some rules-of-thumb (Basiri et al. 2016b).

1. The sample size of trajectory data has to be large, otherwise the training and control/test process may not find all patterns contained in the data.
2. The sample data should be dense enough to give complete spatial coverage.

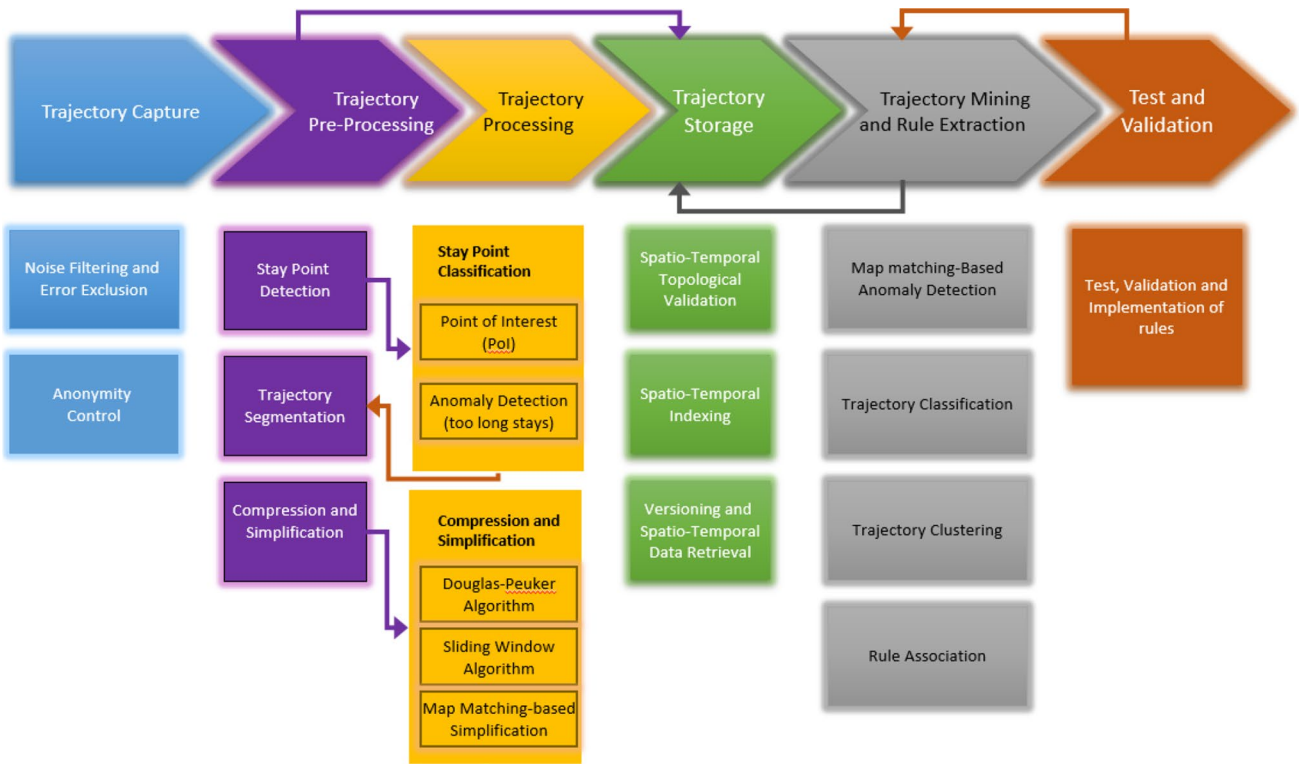

Fig. 1 Trajectory mining process

3. The sample data should be frequent enough to give complete temporal coverage (different days, times, weekends, seasons, and so on)
4. The sample data should cover all (at least most) travel behaviours and modes. This helps spatio-temporal data mining to exclude anomalies and exceptions, and also it helps to find clusters and classes, which share common patterns easier and with a greater level of certainty.

The larger input datasets may lead to more realistic results since anomalies will have less weight in pattern recognition process. However capturing input data with an acceptable level of quality is a big challenge as there is not a globally available, accurate, cheap or ideally free-to-use positioning technique, which can localise the users seamlessly indoors and outdoors. Due to this issue, tracking data can be captured from several different sources such as Global navigation Satellite Systems (GNSS) receivers, e.g. GPS receivers embedded in mobile phones and In-Vehicle Navigation Devices, RFID tags and readers, video cameras and CCTVs, Bluetooth networks (Basiri et al. 2015b, 2017). Also large input data storage, retrieval and analysis could only be efficiently handled if the trajectories are stored in a graph database. This paper uses a graph database, Neo4j, to store and retrieve trajectories. As the name implies graph databases are based on graph theory and employ nodes, properties and edges as their building blocks. The nodes and edges can have properties. In the graph databases, various nodes might have different properties. The graph databases are well suited for data, which can be modelled as networks such as road networks, social networks, biological networks and semantic webs. Their main feature is the fact that each node contains a direct pointer to its adjacent node, so no index lookups are necessary for traversing connected data. As a result, they can manage a huge amount of highly connected data since there is no need for expensive join operations. Some of the graph databases support transactions in the way that relational databases support them. In other words, the graph database allows the update of a section of the graph in an isolated environment, hiding changes from other processes until the transaction is committed. Trajectory data and in general geospatial data can be modelled as graphs, since graph databases support topology natively, topological relationship (especially connectivity) between geospatial data can be easily managed by this type of NoSQL databases (Amirian et al. 2015). In most GIS workflows, topological relationships play a major role. In addition since each edge in a graph database can have a different set of properties, they provide flexibility in the traversal of the network, based on various properties. For example, it is possible to combine time, distance, a number of points of interest, and the user preferences in finding best path and the mentioned path would be unique for each user. In summary, the storage model of graph databases is a graph and there is a need for mapping layer whenever another data structure is needed in the application layer.

In addition, the protection of the contributors/users' privacy is one of the key challenges (Basiri et al. 2015b) as there are many pieces of private and confidential information hidden in the data and any links to the identity of the users/contributors could be damaging to their safety, security and privacy. It is very important to make the trajectories anonymised and this can be done using different anonymisers such as K-anonymity (Gruteser and Grunwald 2003). Most existing anonymisers on tracking data adopt a K-anonymity. In order to do so, the location of the user got by a query and K to the anonymiser, which is a trusted third party (Kalnis et al. 2007; Mokbel et al. 2006) in centralized systems or a peer in decentralized systems (Ghinita et al. 2007a, b). The anonymiser removes the ID of the user and cloaks the exact user location. Then anonymiser sends the location and query to the spatial database or location-based services server.

2.2 The proposed methodology

Having the data anonymised, then trajectories are ready to go through the next stage and be pre-processed. The pre-processing stage is to make them easier to store (using segmentation, compression and simplification techniques) and also semantically more understandable (by identifying the stay points and using the map matching techniques). The pre-processing step is actually based on the fact that all the points in a trajectory are not equally important and meaningful (Zheng 2015). The pre-process step, including stay point detection, trajectory segmentation, trajectory simplification and compression make the trajectories ready to be stored and retrieved in a more efficient way. In addition to the efficiency, trajectory segmentation and stay point detection make the trajectories and some of the points semantically meaningful and easier to interpret. Such stay points can be used in recommended places to visit (Basiri et al. 2014, 2015a), estimate the actual travel time and petrol consumption (Shang et al. 2014).

Stay points refer to the locations where users/contributors have stayed for a while. The stay points can be simply identified if the location of the user is not changing over a period of time. However due to positioning services' inaccuracy and errors (Pang et al. 2013), it commonly happens that the user stays stationary for a while but the positioning technology generates different readings (Zheng 2015). In order to detect such stay points/area, there are several algorithms and methods.

Basiri et al. (2016a) proposed an approach that checks the travel speed for each segment and if it is smaller than a threshold then the average of these two points are replaced/

stored as the "stay point". It is possible to put distance and temporal interval threshold separately, instead of the speed of movement at each segment (i.e. if the distance between a point and its successor is larger than a threshold and also the time span is larger than a given value) as Li et al. (2008) proposed. Yuan et al. (2016) proposed using the density-clustering algorithm to identify the stay points. This paper uses this approach in order to identify the speed clustering algorithms to identify the stay points. This can be viewed as the combination of segments' speed threshold and the density-clustering algorithms.

Data mining techniques, which have been implemented in some pattern recognition projects, are based on static approaches (conventional data mining techniques), which are not fully compatible with the spatial and temporal aspect of trajectory data. Also in some research projects (Yavas et al. 2005; Monreale et al. 2009) where dynamic data mining has been applied, spatial and temporal aspects of input data were considered separately. This approach does not include spatio-temporal relationships which can help to identify some other rules.

The problems with most spatial and temporal (not spatio-temporal) data mining techniques, which have been used for pattern recognition, are:

- Spatio-temporal topological relationships are ignored. The spatial relations, both metric (such as distance) and non-metric (such as topology, direction, shape, etc.) and the temporal relations (such as before and after) are information bearing and therefore need to be considered in the data mining methods. Also, some spatial and temporal relations are implicitly defined and they are not explicitly encoded in a database. These relations should be extracted from the data. However, there is always a trade-off between pre-computing them before the actual mining process starts (eager approach) and computing them on the fly when they are actually needed (lazy approach). Moreover, despite much formalization of space and time relations available in spatio-temporal reasoning, the extraction of spatial/temporal relations implicitly defined in the data introduces some degree of certainty that may have a large impact on the results of the data mining process.
- Working at the level of stored data, that is, geometric representations (points, lines and regions) for spatial data or timestamps for temporal data, is often undesirable. Therefore, complex transformations are required to describe the units of analysis at higher conceptual levels, where human-interpretable properties and relations are expressed.
- Spatial resolution or temporal granularity can have a direct impact on the strength of patterns that can be discovered in the datasets. General patterns are more likely to be discovered at the lowest resolution/granularity level. On the other hand, large support is more likely to exist at the higher levels of resolution. To have a better support and also having more clusters, a higher level of resolution and granularity are needed. In this stage it worth to mention that, lack of spatial and temporal accuracy and precision of input data can be compensated, up to some extent, by number of input trajectories. So for large enough datasets, less accurate trajectories can let us have appropriate results as if accurate data was available.

In order to consider spatio-temporal relationships in the data mining process, a spatio-temporal database, which can store all required aspect of spatio-temporal objects, should be firstly generated. This helps to use, modify and analyse different characteristics and relationships between/of trajectory data. Then using such a database and having large enough input datasets, it would be possible to find similarities, clusters, classes, anomalies, etc. and finally find rules and patterns contained within the movement trajectory clusters.

The final step of knowledge discovery from data is to verify that the patterns produced by the data mining algorithms occur in the wider dataset. Not all patterns found by the data mining algorithms are necessarily valid. It is common for the data mining algorithms to find patterns in the training set which are not present in the general dataset. This is called overfitting. To overcome this, the evaluation uses a test set of data on which the data mining algorithm was not trained. The learned patterns are applied to this test set and the resulting output is compared to the desired output. The accuracy of the patterns can then be measured from how many are correctly classified. A number of statistical methods may be used to evaluate the algorithm, such as ROC curves.

If the learned patterns do not meet the desired standards, then it is necessary to re-evaluate and change the pre-processing and data mining steps. If the learned patterns do meet the desired standards, then the final step is to interpret the learned patterns and turn them into knowledge.

However data mining techniques are very well-developed, in order to have a better pattern recognition process and consider all aspects of trajectory data, it is highly recommended to apply spatio-temporal data mining techniques not to have the problem described in the last subsection in the last subsection. In this case, spatio-temporal relationships are also considered in this process.

As discussed by Abraham and Roddick (1998), the forms that spatio-temporal rules may take are extensions of their static counterparts and at the same time are uniquely different from them. Five main types can be identified:

- *Spatio-temporal associations* These are similar in concept to their static counterparts as described by Agrawal et al. (1993). Association rules are of the form X → Y (c%, s%) where the occurrence of X is accompanied by the occurrence of Y in c% of cases (while X and Y occur together in a transaction in s% of cases).
- *Spatio-temporal generalisation* This is a process whereby concept hierarchies are used to aggregate data, thus allowing stronger rules to be located at the expense of specificity. Two types are discussed in the literature; spatial-data-dominant generalisation proceeds by first ascending spatial hierarchies and then generalising attributes data by region, while nonspatial-data-dominant generalisation proceeds by first ascending the spatial attribute hierarchies. For each of these different rules may result.
- *Spatio-temporal clustering* While the complexity is far higher than its static, non-spatial counterpart the ideas behind spatio-temporal clustering are similar—that is, either characteristic features of objects in a spatio-temporal region or the spatio-temporal characteristics of a set of objects are sought (Ng and Han 1994; Ng 1996).
- *Evolution rules* This form of rule has an explicit temporal and spatial context and describes the manner in which spatial entities change over time. Due to the exponential number of rules that can be generated, it requires the explicit adoption of sets of predicates that are usable and understandable. Example predicates include Follows, coincides, parallels and mutates (Allen 1983; Freksa 1992; Hornsby and Egenhofer 1998).
- *Meta-rules* These are created when rule sets rather than datasets are inspected for trends and coincidental behaviour. They describe observations discovered amongst sets of rules. For example, the support for suggestion X is increasing. This form of rule is particularly useful for temporal and spatio-temporal knowledge discovery.

In order to extract patterns and rules of movements, all input trajectory data are randomly divided into two feature classes. The first one, which is called, training data, is used for pattern recognition and rule learning purposes. Another set of input data, which is called control data, is used to control how the learnt rules and recognised patterns fit into this set of data. After analysing and finding patterns in the training data, the inference system will apply the extracted patterns on the control data to see how similar input control data and estimated results are. If very similar, it is possible to infer that a pattern was discovered and any new data can be analysed using that pattern.

The clusters and classes which can be generated using above-mentioned techniques are used to generate rule sets. The final step of knowledge discovery from data is to verify that the patterns produced by the data mining algorithms occur in the wider dataset. Not all patterns found by the data mining algorithms are necessarily valid. It is common for the data mining algorithms to find patterns in the training set which are not present in the general dataset.

If the learned patterns do not meet the desired standards, then it is necessary to re-evaluate and change the pre-processing and data mining steps. If the learned patterns do meet the desired standards, then the final step is to interpret the learned patterns and turn them into knowledge. In the next section, implementation and the interpretation of such patterns are explained.

3 Implementation

The first step is capturing the trajectory. This can use many different positioning and tracking technologies and methods including Global Navigation Satellite Systems (GNSS), Wireless Local Area Network (WLAN), Radio Frequency Identification (RFID), cameras, mobile networks, Inertial

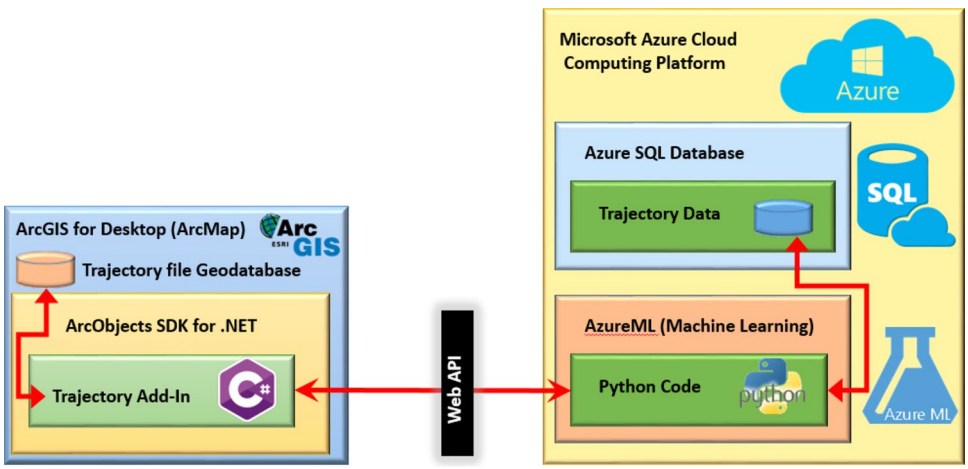

Fig. 2 The high-level architecture of the system (Basiri et al. 2016c)

Navigation Systems (INS), Bluetooth networks, tactile floors, and UltraWide Band (UWB) (Basiri et al. 2015b).

We implement this process using an inference engine, developed as an ArcGIS add-in. Tracking data, which has been captured over a period of two months using a mobile app installed on mobile devices, was analysed based on the algorithm. Figure 2 shows the high-level architecture of the system.

As it illustrated in the figure, data mining and machine learning functionalities are hosted in Microsoft Azure cloud computing platform. At the other hand, geospatial functionalities (such as visualization, generalization) are implemented as an ArcGIS Add-in. Communication between the cloud-hosted machine learning functionality and the ArcGIS is performed over the Web.

Fig. 3 One of the mounted cameras' coverage areas

In order to show that there is a strong correlation between the reliability of the output from data mining and the bounding box (both spatially and temporally) in which the trajectory data are located, two sets of trajectory data for different cities (Hanover in Germany and Maynooth in Ireland) were captured. Trajectories were captured over 2 months (July 2013–August 2013) using a mobile app which can be downloaded from servers at the Institute of Cartography and Geoinformatics (IKG) at the Leibniz University of Hanover and the National University of Ireland, Maynooth (NUIM). The app, which has been used for data capture, can be downloaded from https://www.ikg.uni-hannover.de/index.php?id=635&L=1. Positional data was captured using GPS and mobile networks (usually when the user is moving outdoors) or it can be captured from QR codes affixed to most of the major turning points and important features (especially for indoor localisation) using a QR code reader app (Basiri et al. 2014; 2016a). It is also possible to capture the trajectory of movement from a network of ceiling mounted cameras (such as CCTVs), as it is shown in Figs. 3 and 4.

In camera installation phase, the final goal was maximizing the room coverage and also having a more overlapping area, which is covered by more than one camera. Having the overlapping areas is very important since all analysis is doing over anonymous data and users are identified using a random number (Object ID) assigned. So in order to follow the user roaming from one camera's coverage area to another's, it is very important to have an area, which is covered by two cameras to find the corresponding absolute position of a user in overlapping area, as it shown in Fig. 4.

There are two pre-processing stages on the data: anonymity control and noise filtering/error exclusion.

Due to privacy and data protection issues, it is highly important to anonymise the data especially when the data

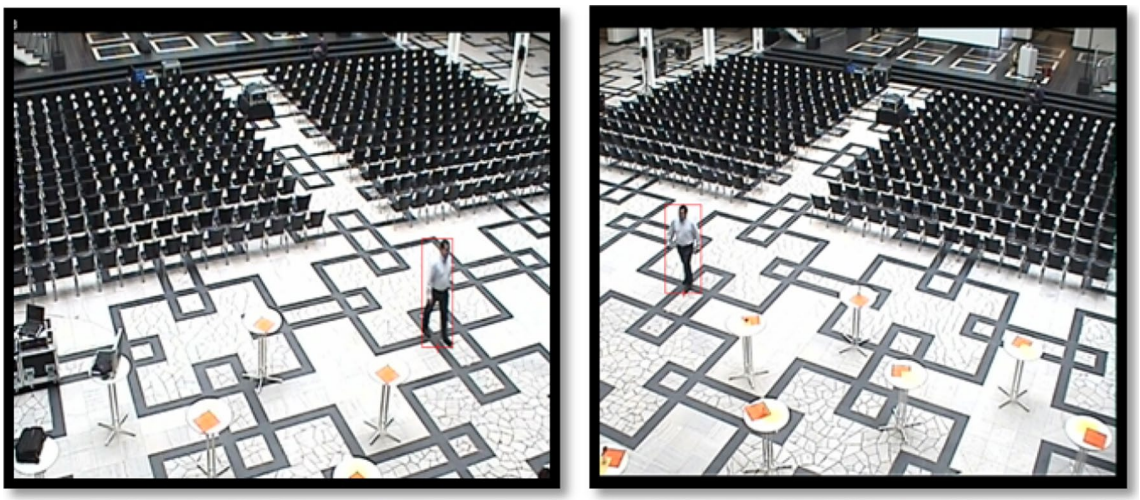

Fig. 4 Minimum bounding rectangle of a user located in the overlapping area of two cameras

is from CCTV cameras (Gidofalvi et al. 2007). Therefore tracking data is stored without any reference to a user's identification. There are various anonymisers (Chow and Mokbel 2011). This paper uses the K-anonymity program (Kalnis et al. 2007), which is trusted third party software often used on tracking data (Chow and Mokbel 2011). The data are stored in centralized systems or on decentralized peer devices (Ghinita et al. 2007b). The anonymiser removes the ID of the user and cloaks the exact user location in the spatial database.

Anonymised trajectories can have some points that are not perfectly accurate or in some cases even valid. Such errors and noises should be filtered in advance to minimise the invalid results at the end of the data mining process. It is very important to remember this phase is being carried out to filter the errors and noises and it is not to exclude the abnormalities and anomalies, as they might be helpful for some applications and scenarios. The noises and errors are in the data due some reasons including poor and multipath positioning signals or tracking the reflection (for example on windows) rather than the actual location of the users by CCTV camera. In order to detect and exclude noises and errors there are some methods available, such as Kalman and Particle filtering and mean (or median) filters, described and reviewed by Lee and Krumm (2011).

This application, however, used a heuristic-based method, as the other filters replace the noise/error in the trajectory with an estimated value and this may have a significant impact on the output of trajectory mining (i.e. the recognised patterns and rules). It calculates the distance and the travel time between each consecutive couple of points in the trajectory and then the travel speed for each segment can be easily calculated. Then it is possible to find out the segments whose travel speeds are larger than a threshold (for example 360 km/h). If the travel mode for each trajectory is also identifiable (identified by the contributors or based on statistical methods which can find of the consecutive segments with the almost the same average speed classifiable into three classes of pedestrian, car/bus/train, and bicycle), then it is possible to have different thresholds depend on the travel mode (a pedestrian cannot walk faster than 20 km/h.).

This paper only uses the travel modes specified by the contributors and simply ignore the possibility of error/noise detection using the statistical methods. This is due to some transitional segments (e.g. from pedestrian to car and then again to pedestrian mode) or some anomalies (which are not due to errors or noises) that might be removed if their spatial characteristics and relationships with surrounding spatial features (i.e. map matching) are not considered. Such segments need to be carefully kept for the next steps of trajectory data mining process as they potentially can have valuable information.

An ArcGIS add-in has been developed, as shown in Fig. 5, to visualise, process and analyse the input trajectory data. As illustrated in Fig. 5, a trajectory analyser in a dockable window is available to ArcMap. The first tab creates a feature class by reading the input XML file of the recorded points (i.e. GNSS logs, scanned QR-Codes, mobile cell-ID

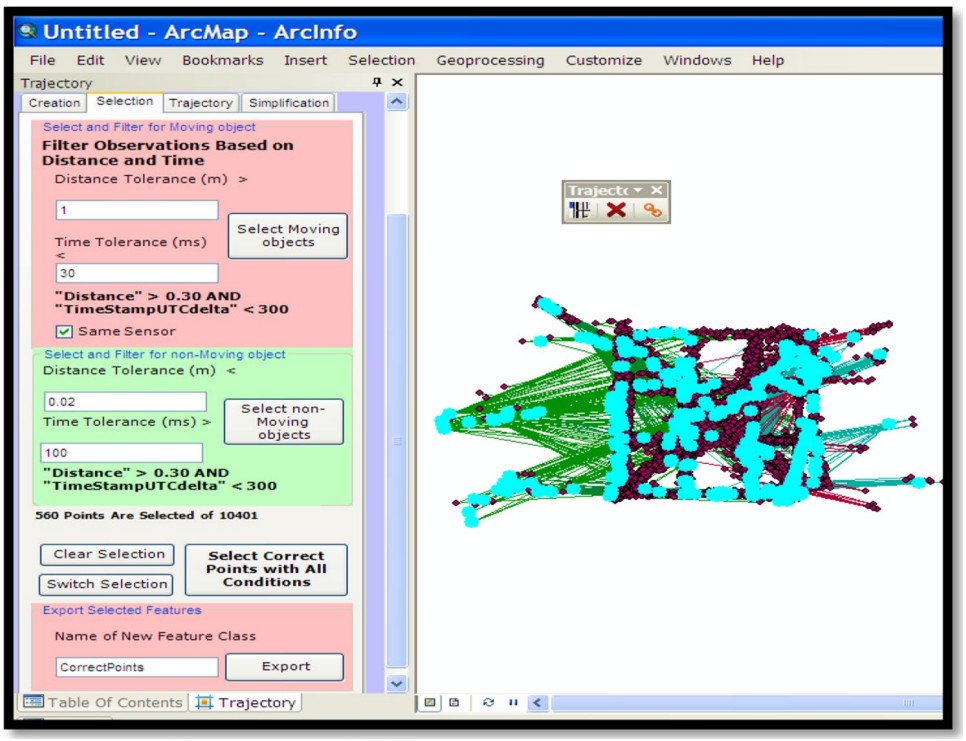

Fig. 5 Trajectory analyzer dockable window (developed ArcGIS add-in)

locations). It can add two columns to the created feature class which calculates the distance between each adjacent point pair (the length of each segment) and the speed of movement of the user passing that segment. The travel speed is compared to the threshold (depending on the travel mode and if the travel mode is not specified by the contributors it is being set to 360 km/h, however this value is a user-defined value and can be easily changed). The travel speed is being stored as an attribute data to each segment as later on it is being used for rule association and also pattern recognition steps. This paper uses the ensemble methods, in addition to some rules of thumb, to identify anomalies and also predictive rules.

A large number of points had labelled to produce a large training set. If a large number of classifiers with accuracy slightly better than a random guess combined together, then the accuracy of the ensemble is superior to most of the known classification algorithms. The idea of ensemble learning is to build a prediction model by combining the strengths of a collection of simpler base models (Hastie et al. 2013; Davidson-Pilon 2015).

Most of the ensemble models in off-the-shelf packages use the binary decision trees as the weak learners and produce an ensemble of hundreds or thousands of binary decision trees. However, this research uses a novel approach for combining other types of classification methods as well as binary decision trees. In MajorityVoteEnsembleClassifier it is possible to use the powerful classification algorithms like logistic regression and support vector machines and improve the accuracy of them by combining these models. Following piece of code, illustrate the code of the mentioned classifier.

The python code (developed by authors) implements some machine learning algorithms (such as MajorityVoteEnsembleClassifier, is explained later) by extending the scikit-learn package. The scikit-learn package is the most widely used machine learning package in python programming language. As illustrated by code snippet by inheriting from BaseEstimator class in scikit-learn package, the implemented algorithms inherit all methods similar to the other models in scikit-learn.

The python code then hosted in AzureML service of Microsoft Azure cloud computing platform. Microsoft's Azure Machine Learning (AzureML) (Tejada 2016; Barnes 2015) dramatically simplifies machine learning model deployment by enabling data scientists to deploy their final models as web services that can be invoked from any application on any platform, including desktop, smartphone, mobile and wearable devices. Hosting the python code of this research in AzureML allows using the same model from various platforms without worrying about scalability issues. The data for training the models are stored in SQL Azure database as trajectory database.

C# programming language was used for implementing the ArcGIS Add-in. ArcGIS Desktop Add-Ins is the preferred way to customize and extend ArcGIS for Desktop applications. With Add-Ins all the functionalities of ArcGIS for Desktop applications are available through ArcObjects, which is a set of components that constitute the ArcGIS platform (Amirian 2013). The Trajectory Add-in implemented as a Dockable window in ArcMap. The first tab creates a feature class in a local file geodatabase by reading the input XML file of the recorded points.

```
from sklearn.base import BaseEstimator
from sklearn.base import ClassifierMixin
from sklearn.preprocessing import LabelEncoder
from sklearn.externals import six
from sklearn.base import clone
from sklearn.pipeline import _name_estimators
import numpy as np
import operator

class MajorityVoteEnsembleClassifier(BaseEstimator, ClassifierMixin):
def __init__(self, classifiers, decisionVote='classlabel', weights=None):
```

In summary, this algorithm can be used as anomaly detection algorithm when the training data are composed of points with two labels (valid and invalid-anomaly). Also, it is possible to provide weights for input training dataset in order to learn from the mistakes of previous classifiers in different iterations.

A majority vote ensemble classifier based on the training data (points with x, y) this model can find the points with valid or invalid positions. So, it can be used as anomaly detection algorithm (binary) as well as classification algorithm (Multi-class) for detecting the type of point. The classifier uses set of weak and strong learners to find an answer with lower variance than the single classifiers. If

boosting is used, the bias is reduced as well. Parameters include:

- classifiers: array-like, different instances of classifier objects, shape = [n_classifiers]
- Different classifiers for the ensemble like SVM, Penalized Linear Models (Lasso, Ridge), Logistic Regression and so on.
- decisionVote: str, {'classlabel', 'probability'} (default = 'classlabel')

If 'classlabel' the prediction is based on the argmax of class labels. Else if 'probability', the argmax of the sum of probabilities is used to predict the class label (recommended for calibrated classifiers). If decisionVote is 'classlabel' and y contains just two values, this ensemble method detects the anomalies.

- weights: array-like, shape = [n_classifiers], optional (default = None)

```
        self.classifiers = classifiers
        self.named_classifiers = {key: value for key, value
                                  in _name_estimators(classifiers)}
        self.decisionVote = decisionVote
        self.weights = weights
    def fit(self, X, y):
```

The parameters of the Fit classifier are

- X : {array-like coordinates of training data},
- shape = [n_samples,2]
- Matrix of training samples [x, y].
- y : array-like, shape = [n_samples]
- Vector of target class labels.

for anomaly detection y has two unique values. The anomaly label is −1 and for ordinary points the label can be one positive value. For classification, y can be any positive value (unique value of y is equal to number of target classes.)

```
            if self.decisionVote not in ('probability', 'classlabel'):
                raise ValueError("vote must be 'probability' or 'classlabel; got
(vote={})", decisionVote)
            if self.weights and len(self.weights) != len(self.classifiers):
                raise ValueError('Number of classifiers and weights must be equal; got
{} weights, {} classifiers'
                                 ,(len(self.weights), len(self.classifiers)))
            self.lablenc_ = LabelEncoder()
            self.lablenc_.fit(y)
            self.classes_ = self.lablenc_.classes_
            self.classifiers_ = []
            for clf in self.classifiers:
                fitted_clf = clone(clf).fit(X, self.lablenc_.transform(y))
                self.classifiers_.append(fitted_clf)
            return self
        def predict(self, X):
```

If a list of 'int' or 'float' values are provided, the classifiers are weighted by importance; Uses uniform weights if 'weights = None'. weights can be used when for learning from mistakes (to set more weights to data points which are classified incorrectly to make the classifier focus on correction the classifier in less iterations.)

If decisionVote equals to 'classlabel' and there are just 2 values in y, then the ensemble works like anomaly detector. The parameters include

- X : array-like coordinates of training data}
- shape = [n_samples,2]
- Matrix of training samples [x, y].

This returns maj_vote : array-like, shape = [n_samples], Predicted class labels. And "−1" means an anomaly. All the other values can be interpreted as class labels.

close to zero) and replace them with a single point with a description of the time interval during which the user's speed was zero.

```
if self.decisionVotevote == 'probability':
            maj_vote = np.argmax(self.predict_proba(X), axis=1)
        else:
            predictions = np.asarray([clf.predict(X)
                                      for clf in self.classifiers_]).T
            maj_vote = np.apply_along_axis(
                                      lambda x:
                                      np.argmax(np.bincount(x,
                                                weights=self.weights)),
                                      axis=1,
                                      arr=predictions)
        maj_vote = self.lablenc_.inverse_transform(maj_vote)
        return maj_vote
    def predict_proba(self, X):
probas = np.asarray([clf.predict_proba(X)
                             for clf in self.classifiers_])
        avg_proba = np.average(probas, axis=0, weights=self.weights)
        return avg_proba
    def get_params(self, deep=True):
        """ Get classifier parameter names for GridSearch"""
        if not deep:
            return super(MajorityVoteEnsembleClassifier,
self).get_params(deep=False)
        else:
            out = self.named_classifiers.copy()
            for name, step in six.iteritems(self.named_classifiers):
                for key, value in six.iteritems(step.get_params(deep=True)):
                    out['%s__%s' % (name, key)] = value

            return
```

This part, Predict class probabilities for X (test data and unseen data), can have following parameters:

- X : array-like coordinates of training data}
- shape = [n_samples,2]
- Matrix of training samples [x, y].
- Training data, where n_samples is the number of data points and the number of features is 2.

This part returns:

- avg_proba : array-like,
- shape = [n_samples, n_classes]
- Weighted average probability for each class per sample.
- Average probability of a point being classified as anomaly.

In addition to the classification of segments, this function also helps to exclude redundant data. For example, it is possible to find points where the user has been stationary (speed

By calculating the correlation between trajectory data, it is possible to discover other modes of classification. Spatio-temporal clustering helps to identify such classes and to discover underlying rules and patterns through identifying parameters which are highly correlated and extracting rules. Evolution rules are applied at this stage through functions on the selection tab to discover spatial and temporal rules. Figure 5 shows clusters and classes of vertices of trajectories depending on the area of search, buffer threshold, which limits numbers of trajectories with the same types (modes) within an area. There is a buffering since there are always inaccuracies due to positioning technologies. In addition to the buffer area, there is buffering for temporal aspect of input data, which limit time interval when trajectories can be matched. Because of the large number of input features, spatial and temporal thresholds are used to cluster and identify associations. Such thresholds depend on the density, frequency and the nature of the input data. They can be changed using this tab according to experts' comments and recommendations.

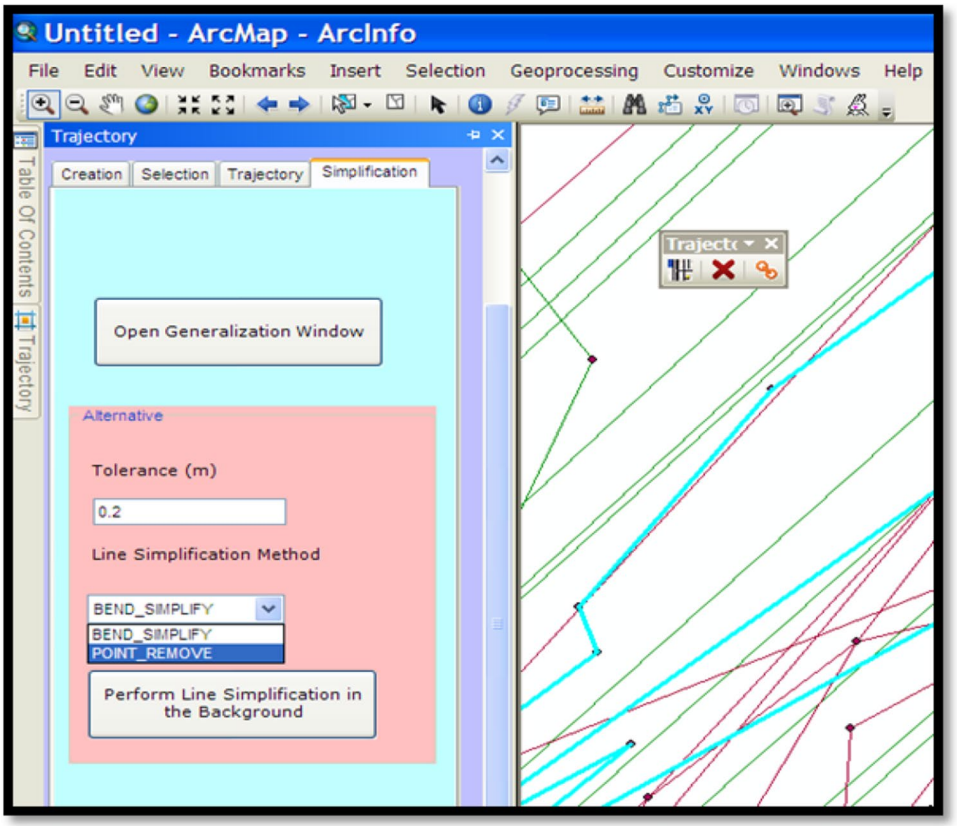

Fig. 6 Finding matched trajectories tab, simplification tab

Having the data anonymised and error/noise excluded, then trajectories are ready to go through the next stage and be pre-processed.

This paper uses this approach in order to identify the speed clustering algorithms to identify the stay points. This can be viewed as the combination of segments' speed threshold and the density-clustering algorithms.

Abnormalities and anomalies are detected (using statistical analysis), then some clusters are identified (using association rules between average speed between two points, number of stops, duration of stops, spatio-temporal topological relationships between trajectories and available features on the maps). Then rules and patterns can be recognised using relevant parameters and criteria, including speed, spatial and temporal correlations between segments and trajectories and also some trajectory matching algorithms. Some of the examples of such rules and patterns are listed below:

If the average speed of movement is more than 50 km/h and the trajectory is matched by the street network, then the travel mode is car; if instead the trajectory is matched with bus lines and there are stops (the speed for a short period of time, becomes zero) at the bus stops then the travel mode may be bus. Such rules can be used in the phase of "recognising user's current situation".

The inference can be more complicated than simple if-then rules. It is possible to find the pattern over input dataset using data mining techniques. By analysing input trajectories captured from different types of users, it is possible to automatically find some similarities and patterns.

In order to extract patterns and rules of movements, all input trajectory data are randomly divided into two feature classes. The first one, which is called, training data, is used for pattern recognition and rule learning purposes. Another set of input data, which is called control data, is used to control how the learnt rules and recognised patterns fit into this set of data. After analysing and finding patterns in the training data, the inference system will apply the extracted patterns on the control data to see how similar input control data and estimated results are. If very similar, it is possible to infer that a pattern was discovered and any new data can be analysed using that pattern.

In this paper since prior knowledge about the input data was included (such as a reference map or additional spatial data), it was decided to evaluate the correctness and logic of output patterns and rules using standards, "common sense" rules-of-thumb and expert comments as well as control data test. However there is a need to compare the results of this approach with other approaches results to evaluate them. One of the early inferred rules and patterns is about identifying the travel mode using speed and behaviour of movement. Based on the speed of movement it is possible to classify data into the four categories of pedestrian, bicycle,

wheelchair and vehicle. Using patterns of movement, it is possible to find some rules which distinguish between public transportation and cars—public transportation stops regularly at very specific points with very low correlation to time, that is whenever a vehicle arrives at the station, it usually stops. Such rules and patterns should be confirmed by control data. However, if no reference data is available, expert comments, logical rules and standard specifications are also part of this process.

One of the biggest challenges in this project is redundant data. It is possible to have more than one trajectory for each user at the same time interval, with different shapes and sizes, since users can be viewed by different cameras synchronously. If two or more cameras detect a person at the same time in their overlapping area, then we will have two or more trajectories stored for that person, as it is shown in Fig. 6. The simplest policy is to consider all trajectories ignoring this fact that some of them may not show different users' movements since they are redundant data for the same user moving in a same period of time. This policy may lead to having a higher weight for trajectories located in overlapping areas. In order to handle this, we need to find only one trajectory, which shows the user's movement as detail as possible. This becomes more complex where there is no link between user and trajectory. Because of privacy issues, there is no link between users and their trajectory. That means, it is not single user ID assigned to user and based on which the corresponding trajectories (got from different cameras) can be identified. In this regard trajectories belong to a single movement made by single user should be identified, and then they should be transformed into one polyline to show user's movement over a period of time.

In order to find the trajectories to be matched and aggregated into one trajectory, we need to find correlated vertices, which represent one object captured by different cameras, and replace them with one vertex. In order to do this, the simplest policy can be finding vertices, which are spatially and temporally near to each other. Because the vertices, which are recorder spatially and temporally near to each other, are more likely to represent one user and the small differences between them can be because of camera synchronisation drift, calibration and instrumental errors. However this approach may be the simplest approach to find trajectories representing a single user's movement, it has got some issues. First of all, and the most important one, is mismatching issue. It is possible to find many trajectories that satisfy the conditions, i.e. captured within quite the same temporal interval and also spatially near, while some of these trajectories belong to another user and they represent another user's movements. In addition, finding the best time interval and also spatial buffering radios for matching process in which trajectories are analysed is quite tricky. Figure 4 shows selection tab in the developed ArcGIS-Add-in which allows selecting vertices captured spatially and temporally close to each other. Then a column will be added to the attribute table to show which segments should be matched. Spatial and temporal thresholds should be given or previously set, for example, the vertices whose distance from the surrounding vertices are less than 2 cm and also have been captured with a 100 ms time interval are selected. Finding the best temporal and spatial threshold depends on the applications, movements' characteristics, experts' comments, equipments' configurations and settings, etc.

This might be the simplest approach, however because of mismatching and also being so experience and application dependent, makes it unreliable. In this regard, this paper proposes using data mining techniques to find matching segment. This means, introducing fixed grid windows in which segments whose patterns are quite same considered to be matched. In contrast with the previous approach which only considers spatial and temporal proximity to find matching segments, this approach considers the patterns of movements. Figure 6 shows how two trajectories are found as matching trajectories using this approach. The green trajectory is captured by camera1 and the red trajectory is captured by camera2. As cameras locations, confutations and settings are different they may capture different trajectories with different numbers of vertices, as it shown in Fig. 6. However the proposed approach considers pattern of movement within a temporal and spatial window. For example the green and the red trajectories in Fig. 6 can be found as matching trajectories since the general trends and the patterns of movement in the predefined window, including four time intervals and a spatial bounding box, is quite same. Those very small noises (which are differences of the trajectories from the general trend) are ignorable according to the predefined settings.

Both approaches, finding near vertices based on a spatial and temporal threshold and also the pattern-based approach, have been implemented in the developed ArcGIS add-in to make choices and also make easier to understand what would be results of both approaches on the same dataset.

Now the data has been pre-analysed and it is possible to define some rules based on which Points of Interests (PoIs) can be extracted. This may be very helpful in tourist guidance or any indoor navigation services. Since cameras (CCTVs) are usually available indoors, especially in galleries and museums. In order to find PoIs, it is possible to select vertices where many users stay for a period of time (probably to see an interesting feature, such as sculpture, painting, etc.). In order to find such points, easily one can use selection tab and select non-moving users. If the number of users who stays in this point (or very close to this point) was more than a threshold, then we can export that point (or area) as a new feature class, called point of interests. Again

spatial and temporal threshold and also the number of users are selected based on the application and experts comments. These PoIs can be used to make navigational suggestions for newcomers.

4 Conclusion

Future mobile services will be more anticipatory of users' needs. Users look for a mobile service/application which can monitor or understand their current situations, predict their needs and demands, and then provide the most appropriate set of services in the most comfortable way. This paper describes a method based on data-mining to detect and extract existing patterns over pedestrian trajectories. The trajectories of anonymised users tracked in surveillance cameras are considered as sequences of time-discrete observations of those users' positions. These trajectories are processed to find the patterns within pedestrians' movements. For example, it is possible to find attractive/important points or features in a gallery, based on the most common users' trajectories; i.e. where there is no movement over a period of time for many tourists. These points can be considered as the point of interests to be applied in giving navigational instructions or suggestions to other users; i.e. it is possible to recommend a new user to go to that point, since many people have visited that location. In order to extract such patterns efficiently from spatio-temporal data of pedestrians, including time and position of users, an ArcGIS add-in has been developed to implement the method. This represents, stores, analyses and extracts patterns of pedestrians' trajectories based on the spatio-temporal data mining methods described in the paper.

Acknowledgements This work was financially supported by EU FP7 Marie Curie Initial Training Network MULTI-POS (Multi-technology Positioning Professionals) under Grant No. 316528.

References

Abraham T, Roddick JF (1998) Opportunities for knowledge discovery in spatio-temporal information systems. Australas J Inf Syst 5(2):3–12

Abramson D, Lees M, Krzhizhanovskaya V, Dongarra J, Sloot P, Bermingham L, Lee I (2014) International conference on computational sciencespatio-temporal sequential pattern mining for tourism sciences, procedia computer science, Vol 29, 2014, pp 379–389 **(ISSN 1877–0509)**

Agrawal R, Imielinski T, Swami AN (1993) A mining association rules between sets of items in large databases. In: Proceedings ACM SIGMOD conference on management of data. ACM, New York, pp 207–216

Allen JF (1983) Maintaining knowledge about temporal intervals. CACM 26(11):832–843

Amirian P (2013) Beginning ArcGIS for desktop development using .NET. Wiley, New Jersey

Amirian A, Basiri, Gales G, Winstanley A, McDonald J (2015) The next generation of navigational services using OpenStreetMap data: the integration of augmented reality and graph databases, OpenStreetMap in GIScience, Springer, Berlin, pp 211–228

Asahara KA, Maruyama KS (2011) Pedestrian-movement prediction based on mixed Markov-chain model. In: Proceedings of the 19th ACM SIGSPATIAL international conference on advances in geographic information systems (GIS '11), pp 25–33

Ashbrook D, Starner T (2003) Using GPS to learn significant locations and predict movement across multiple users. Personal Ubiquitous Comput 7:275–286

Baiget P, Sommerlade E, Reid I, Gonzàlez J (2008) Finding prototypes to estimate trajectory development in outdoor scenarios. In: Proceedings of the first international workshop on tracking humans for the evaluation of their motion in image sequences (THEMIS2008)

Barnes J (2015) Azure machine learning microsoft azure essentials. Microsoft Press, Redmond, Washington, USA

Basiri A, Amirian P (2014) Automatic point of interests detection using spatio-temporal data mining techniques over anonymous trajectories. In: Proceedings of the international conceference on computational science and its applications—ICCSA 2014, pp 185–198

Basiri A, Amirian P, Winstanley A (2014) The use of quick response (QR) codes in landmark-based pedestrian navigation. Int J Navig Obs 2014:1–7

Basiri A, Marsh S, Moore T, Amiran P (2015a) Automatic detection of points of interest using spatio-temporal data mining. J Mob Multimedia 11(3&4):193–204

Basiri A, Peltola P, Figueiredo e Silva P, Lohan ES, Moore T, Hill C (2015b) Indoor positioning technology assessment using analytic hierarchy process for pedestrian navigation services. In Localization and GNSS (ICL-GNSS). IEEE, pp 1–6

Basiri A, Amirian P, Winstanley A, Marsh S, Moore T, Gales G (2016a) Seamless pedestrian positioning and navigation using landmarks. J Navig 69:24–40

Basiri A, Mike J, Pouria A, Amir P, Monika S, Adam W, Terry M, Lijuan Z (2016b) Quality assessment of OpenStreetMap data using trajectory mining. Geo-spat Inform Sci 19(1):56-68

Basiri A, Pouria A, Peter M (2016c) Using crowdsourced trajectories for automated OSM data entry approach. Sensors 16(9). https://doi.org/10.3390/s16091510

Basiri A, Lohan ES, Moore T, Winstanley A, Peltola P, Hill C, Amirian P, Figueiredo e Silva P (2017) Indoor location based services challenges, requirements and usability of current solutions. Comp Sci Rev 24:1–12

Browarek S (2010) High resolution, low cost, privacy preserving human motion tracking system via passive thermal sensing, Master Thesis, Department Electrical Engineering and Computer Science, MIT

Chen Z, Xia J, Caulfield C (2014) A survey of a personalised location-based service architecture for property hunting. J Spat Sci 59(1):63–78

Chen TCT, Honda K, Wang YC (2015) Int J Internet Manuf Serv 4(1):54–61

Chow CY, Mokbel MF (2011) Privacy of spatial trajectories. In: Zheng Y, Zhou X (eds) Computing with spatial trajectories. Springer, Berlin

Cressie N, Wikle CK (2015) Statistics for spatio-temporal data. Wiley, New Jersey

Davidson-Pilon C (2015) Bayesian methods for hackers: probabilistic programming and Bayesian inference. Addison-Wesley, Boston

Dodge S, Weibel R, Lautenschütz A-K (2008) Towards a taxonomy of movement patterns. Inf Vis 7:240–252

Freksa C (1992) Using orientation information for qualitative spatial reasoning. Theor Methods Spatio Temporal Reason Geogr Space LNCS 639:162–178

Ghinita G, Kalnis P, Skiadopoulos S (2007a) MobiHide: a mobilea peer-to-peer system for anonymous location-based queries. Adv Spat Temporal Databases Lect Notes Comput Sci 4605:221–238

Ghinita G, Kalnis P, Skiadopoulos S (2007b) PRIVE: anonymous location-based queries in distributed mobile systems. In: Proceedings of the 16th international conference on World Wide Web. ACM, Banff, Alberta, Canada, pp 371–380

Gidofalvi G, Huang X, Pedersen TB (2007) Privacy preserving data mining on moving object trajectories. In: Proceedings of the 8th IEEE international conference on mobile data management, Mannheim, Germany, May 7–11, pp 60–68

Gruteser M, Grunwald D (2003) Anonymous usage of location-based services through spatial and temporal cloaking. In: Proceedings of the 1st international conference on mobile systems applications and services. ACM, San Francisco, California, pp 31–42

Hastie T, Tibshirani R, Friedman J (2013) The elements of statistical learning: data mining, inference, and prediction, 2nd edn. Springer, Berlin

Hornsby K, Egenhofer MJ (1998) Identity-based change operations for composite objects. In: Poiker T, Chrisman N (eds) Proceedings of 8th international symposium on spatial data handling. International Geographical Union, Vancouver, Canada, pp 202–213

Huang K, Zhu J (2015) Research design of intelligent tourist guide system and development of APP.

Kabassi K (2013) Personalisation systems for cultural tourism. In: Tsihrintzis GA, Virvou M, Jain LC (eds) Multimedia services in intelligent environments. Springer, Heidelberg, pp 101–111

Kalnis P, Ghinita G, Mouratidis K, Papadias D (2007) Preventing location-based identity inference in anonymous spatial queries. IEEE Trans Knowl Data Eng 19(12):1719–1733

Kuntzsch C, Bohn A (2013) A framework for on-line detection of custom group movement patterns. In: Krisp JM (ed) Progress in location-based services, lecture notes in geoinformation and cartography. Springer, Heidelberg, pp 91–107

Lee WC, Krumm J (2011) Trajectory processing. In: Zheng Y, Zhou X (eds) Computing with spatial trajectories. Springer, Berlin, pp 1–31

Li Q, Zheng Y, Xie X, Chen Y, Liu W, Ma M (2008) Mining user similarity based on location history. In: Proceedings of the 16th annual ACM international conference on advances in geographic information systems. ACM, New York

Lu EHC, Chen CY, Tseng VS (2012) Personalized trip recommendation with multiple constraints by mining user check-in behaviors. In: Proceedings of the 20th international conference on advances in geographic information systems, ACM, pp 209–218

Manes G (2002) The tetherless tourist: ambient intelligence in travel and tourism. Inf Technol Tour 5(4):211–220

Mazimpaka JD, Timpf S (2016) Trajectory data mining: a review of methods and applications. J Spat Inf Sci 13:61–99

Mokbel MF, Chow C-Y, Aref WG (2006) The new casper: query processing for location services without compromising privacy. In: VLDB 2006

Monreale A, Pinelli F, Trasarti R, Giannotti F (2009) WhereNext: a location predictor on trajectory pattern mining. In: Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining, pp 637–646

Nakamura H, Gao Y, Gao H, Zhang H, Kiyohiro A, Mine T (2015) Adaptive user interface for personalized transportation guidance system. In: Tourism informatics. Springer, Berlin, pp 119–134

Ng RT (1996) Spatial data mining: discovering knowledge of clusters from maps. In: Proceedings of 1996 ACM-SIGMOD workshop on research issues on data mining and knowledge discovery. ACM, New York

Ng RT, Han J (1994) Efficient and effective clustering methods for spatial data mining. In: Proceedings of VLDB 1994, September 12–15, Santiago, Chile

Pang LX, Chawla S, Liu W, Zheng Y (2013) On detection of emerging anomalous traffic patterns using GPS data. Data Knowl Eng 87:357–373

Shang J, Zheng Y, Tong W, Chang E, Yuan NJ, Zheng Y, Xie X, Wang Y, Zheng K, Xiong H (2014) Discovering urban functional zones using latent activity trajectories. IEEE Trans Knowl Data Eng 27(3):1041–1347

Smirnov A, Kashevnik A, Balandin SI, Laizane S (2013) Intelligent mobile tourist guide. In: Internet of things, smart spaces, and next generation networking. Springer, Berlin, pp 94–106

Sorrentino F, Spano LD, Scateni R (2015) SuperAvatar children and mobile tourist guides become friends using superpowered avatars. In: 2015 International conference on interactive mobile communication technologies and learning (IMCL). IEEE

Tejada Z (2016) Mastering azure analytics: architecting in the cloud with azure data lake, HDInsight, and Spark. O'Reilly

Tom Dieck MC, Jung T (2015) A theoretical model of mobile augmented reality acceptance in urban heritage tourism. Curr Issues Tour 1–21. https://doi.org/10.1080/13683500.2015.1070801

Umanets A, Ferreira A, Leite N (2014) GuideMe—a tourist guide with a recommender system and social interaction. Proc Technol 17:407–414

Vu HQ, Li G, Law R, Ye BH (2015) Exploring the travel behaviors of inbound tourists to Hong Kong using geotagged photos. Tour Manag 46:222–232

Wang X, Li XR, Zhen F, Zhang J (2016) How smart is your tourist attraction?: measuring tourist preferences of smart tourism attractions via a FCEM-AHP and IPA approach. Tour Manag 54:309–320

Xu Q, Ke W (2015) The construction of wisdom scenic area research based on tourist experience: a case of summer palace. In: LISS 2014. Springer, Berlin, pp 675–680

Yavas G, Katsaros D, Ulusoy O, Manolopoulos Y (2005) A data mining approach for location prediction in mobile environments. Data Knowl Eng 54(2):121–146

Yuan H, Xu H, Qian Y, Li Y (2016) Make your travel smarter: summarizing urban tourism information from massive blog data. Int J Inf Manag 36(6):1306–1319

Zhang L, Dalyot S, Sester M (2013) Travel-mode classification for optimizing vehicular travel route planning, progress in location-based services, Springer, Berlin, pp 277–295

Zheng Y (2015) Trajectory data mining: an overview. ACM Trans Intell Syst Technol 6(3):29. http://www.crowdsourcing.org/editorial/crowdsourcing-applications-for-online-tourism-portals/31290. Accessed 2017