



TECHNISCHE
UNIVERSITÄT
WIEN

Vienna University of Technology

MASTERARBEIT

EXTRACTION OF USER'S STAYS AND TRANSITIONS FROM GPS LOGS: A COMPARISON OF THREE SPATIO-TEMPORAL CLUSTERING APPROACHES

Ausgeführt am Institut für
Geoinformation und Kartographie
der Technischen Universität Wien

unter der Anleitung von
Univ.Prof. Mag.rer.nat. Dr.rer.nat. Georg Gartner, TU Wien
Univ.Lektor Dipl.-Ing. Dr.techn. Karl Rehl, TU Wien
Mag. DI(FH) Cornelia Schneider, Salzburg Research

durch
Francisco Daniel Porras Bernárdez
Austrasse 3b 209, 5020 Salzburg

Wien, 25 January 2016

Unterschrift (Student)

MASTER'S THESIS

EXTRACTION OF USER'S STAYS AND TRANSITIONS FROM GPS LOGS: A COMPARISON OF THREE SPATIO-TEMPORAL CLUSTERING APPROACHES

Conducted at the Institute for
Geoinformation und Kartographie
der Technischen Universität Wien

under the supervision of
Univ.Prof. Mag.rer.nat. Dr.rer.nat. Georg Gartner, TU Wien
Univ.Lektor Dipl.-Ing. Dr.techn. Karl Rehl, TU Wien
Mag. DI(FH) Cornelia Schneider, Salzburg Research

by
Francisco Daniel Porras Bernárdez
Austrasse 3b 209, 5020 Salzburg

Wien, 25 January 2016

Signature (Student)

ACKNOWLEDGEMENTS

First of all, I would like to express my deepest gratitude to **Dr. Georg Gartner** and **Dr. Karl Rehrl** for their supervision as well as **Mag. DI(FH) Cornelia Schneider** and all the time they have deserved to my person. I really appreciate their patience and continuous support.

I am also very grateful for the amazing opportunity that **Salzburg Research Forschungsgesellschaft m.b.H.** has given to me allowing me to develop my thesis during an internship at the institute. I will be always grateful for the confidence Dr. Rehrl and Mag. DI(FH) Schneider placed in me.

My special gratitude goes to **Dr. Richard Brunauer** for his guidance and enlightenment during the internship, **Simon Gröchenig** for his Java training, **Verena Venek** for her useful feedback and also the brilliant remaining team at SRFG because I have learnt from all of them.

My gratitude also goes to **Dr. Stefan Peters** and **Juliane Crone**, coordinators of the Cartography M.Sc. who have helped me always, the rest of the professors that have devoted their time to teach me and all my fellows and friends in the master. I thank every person and institution which has helped me during my studies.

A special mention goes to **Lisa, Gerhard, Lisbeth, Edi** and **Agi** for supporting me during the master with great kindness.

Furthermore, I would like also to thank TUM, TUW and TUD for granting me the opportunity of learning and growing within world-class universities. I will be always grateful to Germany and Austria for giving me the opportunity of receiving high quality education for free and, also Hope.

Finally, I need to thank my dear **Ángeles** because she is and always will be my angel. I thank my **brother** and my **parents**, who have devoted their lives to my human growth and specially my mother, **Martina** who is the most important person in the World for me and the greatest role model in which I will always inspire.

ABSTRACT

Analysis of movement behaviour of individuals has emerged as relevant research field and a wide range of potential applications have been proposed in previous literature. The advancement of positioning technologies and the development of hardware and software have contributed to the popularization of mobile devices and the expansion of Location Based Services. One of the consequences is the increase of mobility data available for developing new methods of analysis of movement behaviour.

Previous research on GPS data has mainly focused on trajectory analysis, although alternative approaches propose considering only the stationary parts. Some of these works aim to discover the places visited by the user and the stays performed on them as first step for a user's movement analysis. Clustering based approaches rely on different algorithms for clustering GPS logs collected by the user.

A general approach suitable for movement behaviour analysis is suggested. The aims of this general approach are **detecting the places visited** by a user as well as **characterising the stays** at these places **and the transitions** performed between them.

In order to detect the visited places, three spatio-temporal clustering approaches are proposed and evaluated under a common evaluation framework. This framework includes spatial and temporal measures to systematically assess three algorithms performing **incremental**, **density-based clustering** and a **combination** of both. Ground truth data collected by four users and tagged during collection process is used to test the validity of the approaches. The optimum parameter values for the algorithms are determined according to the results of the quality evaluation.

The characterisation of the user stays and transitions implies the extraction of them as well as the evaluation of this extraction comparing the three clustering algorithms. Two indices related with number and duration of stays and transitions are suggested for the assessment of the extraction accuracy. A movement behaviour profile of a user is developed and described.

Keywords: Movement behaviour, place discovering, clustering analysis, clustering comparison, GPS, incremental clustering, spatial-temporal data, data mining

ABBREVIATIONS

SRFG	–	Salzburg Research Forschungsgesellschaft m.b.H.
GPS	–	Global positioning system
GTD	–	Ground truth data
QE	–	Quality evaluation
SQL	–	Structured Query Language
OPTICS	–	Ordering Points to Identify the Clustering Structure
DBSCAN	–	Density-Based Spatial Clustering of Applications with Noise

LIST OF FIGURES

Figure 1. Thesis aim.....	13
Figure 2. Working principle of K-Means (Zhang Xiao 2015)	17
Figure 3. Core-distance (o) and reachability-distances for MinPts = 4 (Ankerst et al. 1999).....	20
Figure 4. Reachability plot with 3 clusters (Ankerst et al. 1999).....	20
Figure 5. Reachability plot showing a cluster (Ankerst et al. 1999).....	21
Figure 6. DJ-Cluster. (Changqing, Bhatnagar, et al. 2007).....	21
Figure 7. Relation between cluster radius and locations detected in (Ashbrook & Starner 2002).....	23
Figure 8. Illustration of the Time-Based Clustering algorithm (Kang et al. 2005).....	23
Figure 9. Pseudocode of the <i>i-cluster</i> algorithm (Hu & Wang 2007).....	25
Figure 10. Parsed points and detected stay points	25
Figure 11. OPTICS clustering of stay points	26
Figure 12. Circular buffer considered around a Tagged Place.....	27
Figure 13. Time tolerance representation	31
Figure 14. Method	36
Figure 15. Original track and smoothed version.	39
Figure 16. “Time-based” clustering algorithm (Kang et al. 2005).....	42
Figure 17. “Stay Point Detection” clustering algorithm (Ye et al. 2009).....	43
Figure 18. Incremental clusters from different days obtained at the same locations.....	44
Figure 19. Overlapping incremental clusters and a DBSCAN cluster of their centroids.....	44
Figure 20. Representation of places and transitions.....	46
Figure 21. Representation of <i>tagged</i> vs. <i>detected</i> stays and transitions.....	47
Figure 22. Parallel coordinates plot. Tested values for L.....	49
Figure 23. Parallel coordinates plot. L = 60.....	50
Figure 24. Parallel coordinates plot. L = 90.....	50
Figure 25. Parallel coordinates plot. L = 30.....	50
Figure 26. Parallel coordinates plot. L = 10.....	50
Figure 27. Parallel coordinates plot. L = 120.....	50
Figure 28. Process of incremental clusters grouping and convex hull creation.....	51
Figure 29. Comparison of clustering results: DBSCAN and own solution.....	52
Figure 30. DBSCAN clustering and stays extraction	53
Figure 31. Details of stay extraction	53
Figure 32. Simple QGIS process.....	54
Figure 33. OPTICS results from ELKI visualised in QGIS.....	54
Figure 34. Results of DBSCAN and OPTICS clustering from ELKI	55
Figure 35. Example of a quality evaluation log	56
Figure 36. Piece of code from the quality evaluation class	56
Figure 37. Representation of two incremental clusters and GPS points involved.....	57
Figure 38. Representation of visited places and stays at them.....	57
Figure 39. Representation of two visited places and a transition between them.....	59
Figure 40. Visualization of detected and tagged stays as stacks of cylinders.	61
Figure 41. Values of quality measures for first sub-approach.....	63
Figure 42. Detections and values from confusion matrix for first sub-approach.....	64
Figure 43. Relation runtime - number of points (Kang)	64

Figure 44. Example of Kang clustering results and relation with GTD	65
Figure 45. Values of quality measures for second sub-approach.	66
Figure 46. Detections and values from confusion matrix for second sub-approach.	67
Figure 47. Relation runtime - number of points (Ye).....	67
Figure 48. Values of quality measures for third sub-approach.	68
Figure 49. Detections and values from confusion matrix for third sub-approach.	69
Figure 50. ELKI DBCAN clustering runtimes.....	69
Figure 51. Relation runtime - number of points (DBSCAN)	69
Figure 52. Proportion of tagged stay time detected for User1	75
Figure 53. Values from confusion matrix for User1 Kang clustering.	76
Figure 54. Values for confusion matrix from stays extraction.....	77
Figure 55. Proportion of tagged transition time detected for User1.....	80
Figure 56. Values for confusion matrix from transitions extraction.....	81
Figure 57. Visualization of stays as stacks of cylinders. <i>Home1</i> and <i>Work</i> area.....	85
Figure 58. Visualization of stays as stacks of cylinders. <i>Home2</i> area.....	86
Figure 59. Visualization of detected and tagged stays as stacks of cylinders.	87
Figure 60. Combined visualization of transitions and stays with stacks of cylinders and pipes.	88

LIST OF TABLES

Table 1. Description of <i>Task 1</i>	14
Table 2. Description of <i>Task 2</i>	15
Table 3. Simulated values for q_t	30
Table 4: Simulated values for Q_{st} (Venek et al. 2015)	30
Table 5. Simulated values for Q_{st} (Venek et al. 2015).....	30
Table 6. Simulated values for Q_{it} (Venek et al. 2015).....	32
Table 7. Simulated values for Q_{it} (Venek et al. 2015).....	33
Table 8. Confusion matrix without True Negatives (TN).....	34
Table 9. Relation between time window width and mean velocity (Gröchenig & Hufnagl 2015)	39
Table 10. Parameters values for combinations	49
Table 11. Parameter values tested	51
Table 12. Parameter values tested	52
Table 13. Parameter values tested (DBSCAN)	55
Table 14. Example of stays table.....	58
Table 15. Example of transitions table	59
Table 16. Best values for each index and generating parameter settings.....	70
Table 17. Best F measures reached by the algorithms.....	71
Table 18. Best F measures clustering User1 GTD	71
Table 19. Quality of the extraction performed by the 3 algorithms using the best clustering parameters..	72
Table 20. Stays extraction performed by algorithms: Number of stays at 3 most visited places.....	73
Table 21. Stays extraction performed by algorithms: TOTAL of stays at visited places	74
Table 22. Stays extraction performed by algorithms: Stays duration at 3 most visited places.....	74
Table 23. Stays extraction performed by the algorithms: TOTAL stay durations at places.....	74
Table 24. Number of stays at each tagged place for each weekday.....	78
Table 25. Duration of stays at each tagged place for each weekday.....	79
Table 26. Number of transitions between tagged places for each weekday.	83
Table 27. Duration of transitions between tagged places for each weekday.	84
Table 28. Detected transitions of User1 during time intervals on Mondays	89
Table 29. Proportion between detected and real transitions of User1 on Mondays.....	91

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	3
ABSTRACT.....	4
ABBREVIATIONS	5
LIST OF FIGURES	6
LIST OF TABLES	8
TABLE OF CONTENTS	9
1. INTRODUCTION.....	11
1.1. Context and relevance of the topic.....	11
1.2. Scope of the work.....	13
1.2.1. Aim	13
1.2.2. Research questions.....	13
1.2.3. Tasks and objectives	14
1.3. Outline	16
2. THEORETICAL FOUNDATION	17
2.1. Clustering approaches	17
2.1.1. Partitioning-based clustering	17
2.1.2. Density-based clustering	18
2.1.3. Incremental clustering.....	22
2.2. Quality Evaluation framework	26
2.2.1. Quality measures.....	29
2.2.2. Confusion matrix	33
3. METHOD	36
3.1. Data pre-processing	36
3.2. Determination of visited places	40
3.2.1. Clustering	40
3.2.2. Quality evaluation	45
3.3. Characterisation of stays and transitions.....	46
3.3.1. Extraction of stays and transitions	46
3.3.2. Quality evaluation of the extraction of stays and transitions	46
3.3.3. Analysis of the stays and transitions extracted	48
4. IMPLEMENTATION	48
4.1. Determination of visited places	48
4.1.1. Incremental clustering (Kang)	48

4.1.2. Incremental + Density-based clustering (Ye + <i>ConvexHull</i>)	51
4.1.3. Density-based clustering	52
4.1.4. Quality Evaluation	56
4.2. Characterisation of stays and transitions.....	57
4.2.1. Extraction of stays	57
4.2.2. Extraction of transitions	59
4.2.3. QE of the stays and transitions extraction.....	60
4.2.4. Representation of stays and transitions.....	61
5. RESULTS AND DISCUSSION.....	62
5.1. Determination of visited places	62
5.1.1. Clustering results	62
5.1.1.1. Incremental clustering (<i>Kang</i>)	62
5.1.1.2. Incremental + density-based clustering (Ye + <i>ConvexHull</i>)	66
5.1.1.3. Density-based clustering (<i>DBSCAN</i>)	68
5.1.2. Algorithms assessment.....	70
5.2. Characterisation of stays and transitions.....	72
5.2.1. Algorithms assessment.....	72
5.2.2. Quality of the extraction with the <i>Incremental</i> approach	75
5.2.2.1. Extraction of stays	75
5.2.2.2. Extraction of transitions	80
5.2.3. Possible applications	85
5.2.3.1. Movement behaviour visualization.....	85
5.2.3.2. Future prediction	88
6. CONCLUSIONS AND OUTLOOK	92
6.1. Conclusions	92
6.2. Outlook	93
LITERATURE.....	95

1. INTRODUCTION

1.1. Context and relevance of the topic

Analysis of movement behaviour of individuals has emerged as relevant research field and a wide range of potential applications have been proposed in previous literature such as life patterns mining (Ye et al. 2009), prediction of user movements (Ashbrook & Starner 2003), frequent locations learning (Marmasse & Schmandt 2000) or supporting location-aware services (Bicocchi et al. 2008).

The first phase of movement behaviour analysis often requires the autonomous learning of the places visited by the subject. This implies the use of positioning techniques. Main positioning methods rely on satellites or mobile communication networks.

Positioning technologies have been evolving during the last decades and are used both in indoor and outdoor environments. For outdoor applications, the use of satellite systems for positioning offers higher accuracy in rural and urban environments in comparison to other methods. It also offers global availability, despite its multiple disadvantages generating systematic and random errors.

Meanwhile, mobile devices have evolved and diversified in terms of technology and design. Miniaturization and reduction of costs allow mobile platforms to include a growing number of sensors, especially smartphones or wearables which are becoming very popular. Standardization of hardware and software has helped trigger the popularization and development of mobile devices and new Location Based Services. As a side effect, an increasing stream of mobility data is available for developing new methods of analysis of movement behaviour if such data is properly collected.

Previous research on GPS data has mainly focused on movement pattern analysis based on the analysis of trajectories or part of them such as previous works on inference of user's significant places and current activity (Liao, Fox, et al. 2007), trip purpose (Wolf et al. 2001) or transportation mode (Zheng et al. 2008; Patterson et al. 2003; Liao, Patterson, et al. 2007; Reddy et al. 2008). This has revealed to be costly in terms of computational effort, in some cases (Buchin et al. 2011) with runtime complexities of $O(n^4)$.

Alternative approaches rely on considering only the stationary parts of trajectories instead of the mobile parts. In this direction, different groups have worked on identifying user's significant places (Ashbrook & Starner 2003; Cao et al. 2010; Changqing, Bhatnagar, et al. 2007; Hu & Wang 2007; Montoliu et al. 2013; Ye et al. 2009) and their automatic labelling with semantic meaning (Krumm & Rouhana 2013; Huang 2012; Montoliu & Martínez-sotoca 2012; Zhu et al. 2012; Zhu et al. 2013; Bicocchi et al. 2008; Castelli et al. 2007). Additionally, prediction of future movements has been a subject of research based on linear and probabilistic models (Etter et al. 2013; Gao et al. 2012; Hariharan & Toyama 2004; Krumm & Horvitz 2006; Liao, Patterson, et al. 2007; Wang & Prabhalla 2012) able to forecast the next location of the user and focused on transitions between locations.

Previous work based on stationary parts of trajectories could be classified into machine learning, fingerprinting and clustering based approaches. Analysis of user's movement behaviour typically starts with the discovery of the places visited by the user and the stays performed on them. Most of the clustering based approaches reviewed rely on partitioning, density-based or incremental clustering of the GPS logs collected by the user or mobile element.

The algorithms used for place detection are often evaluated individually. However, it is also possible to find performance evaluation of multiple algorithms with heterogeneous criteria (Changqing, Bhatnagar, et

al. 2007; Montoliu et al. 2013). The optimal values for the algorithm parameters depend on the resulting clusters and their relation with the real locations visited. Some authors base their parameter tuning on the number of places detected (Ashbrook & Starner 2003; Hu & Wang 2007) whereas others do not provide a thorough explanation of their criteria (Zheng et al. 2009).

Additionally, in some cases small data samples are used as ground truth data (Ashbrook & Starner 2002) in contrast with other works which cluster real-life long datasets without ground truth spatial data (Cao et al. 2010). Moreover, there is a disparity in the methodology used for generating ground truth data; some studies build a diary with times of visits to the places while collecting the data (Hightower et al. 2005), while others tag the visited places after the data collection (Krumm & Rouhana 2013).

To the best of our knowledge, there are no available systematic empirical evaluations in the literature which focus on a clustering approach of GPS logs under the following conditions:

- Comparing different classes of algorithms under a common assessment framework.
- Using a spatial and temporal accurate evaluation framework.
- Evaluating the optimal clustering parameters based on predefined spatio-temporal quality metrics.
- Using real-life ground truth data tagged during data collection and for different users.

The evaluation of different clustering algorithms is basic for the selection of an adequate approach to learn and represent the normal movement behaviour of a mobile element. The use of a systematic empirical evaluation framework enables the assessment of different approaches, not only clustering based. Additionally, given the spatio-temporal nature of the analysed phenomena, including the spatial and temporal dimensions in the evaluation might improve the assessment quality.

The quality of the evaluation would also benefit from an algorithm parameter selection based on the best clustering performance, instead of merely the number of clusters generated which often include irrelevant false positives.

Last but not least, the use of ground truth data collected by different users for the algorithms validation is important to take into account different movement behaviour patterns. Moreover, building a diary with spatial and temporal information of the stays at places during the data collection phase would avoid most of the errors caused by the limited human memory.

Mobile elements could be objects, animals or human beings: elders, children, etc. Among other applications, predicting irregular behaviour of mobile elements could allow the development of automated systems able to detect anomalous situations and start a human intervention to deal with potential problems and reduce the time needed to react to changes. This would be one of the fields this thesis aims to contribute aligned with SRFG¹ research objectives.

¹ SRFG: Salzburg Research Forschungsgesellschaft m.b.H.

1.2. Scope of the work

1.2.1. Aim

As a contribution for an integrated method to detect *irregular movement behaviour* of mobile elements, the present work aims to determine an **adequate general approach** to **detect** the **places** visited by a user and the **transitions** between such places. Moreover, such general approach must be able to **characterise** the **stays** performed in the places by the user and the **transitions** between them.

This work **implements** three existing clustering algorithms and **develops three different spatio-temporal sub-approaches** to detect the user's *visited places*. Then, an already existing theoretical **quality evaluation framework** is implemented to systematically **evaluate** the three sub-approaches and **determine the optimal** one to complete a **general approach** suitable for **user's movement behaviour** representation.

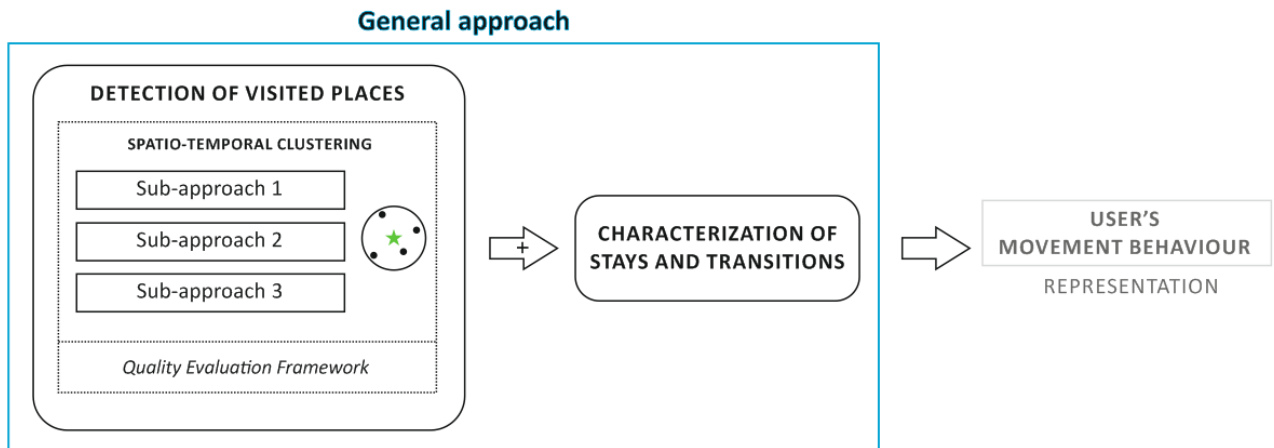


Figure 1. Thesis aim

1.2.2. Research questions

Different research questions and sub-questions have been identified so as to tackle the thesis aim.

- Which spatio-temporal clustering approach is the most adequate for the automatic detection of a user's visited places?
 - Which is the best algorithm to detect the places visited by the user?
 - What are the differences between the tested algorithms?
 - Which are the best values for the parameters of the clustering algorithms?
- Which approach is adequate to characterise the stays and transitions between the visited places?
 - Which algorithm performs the best stays and transitions extraction?
 - Which information can be extracted to represent the stays?
 - Which information can be extracted to represent the transitions?

1.2.3. Tasks and objectives

In order to initiate any kind of analysis of the mobility behaviour of users, we need to be able to determine places visited in their daily lives. This task is expected to generate a collection of clusters of GPS points which represent locations visited by the user during a period of time.

The spatial and temporal performance of the clustering algorithms is evaluated using a common quality evaluation framework. Clustering targets have been defined according to results presented in (Montoliu et al. 2013; Kang et al. 2005; Ye et al. 2009). Based on the quality evaluation, an assessment of the sub-approaches is generated and the best one is determined.

TASK 1	
Determination of visited places and evaluation of the detection quality	
GOAL	<ul style="list-style-type: none"> - Determining visited places in a user's daily life - Evaluation of clustering algorithms
RESULTS	<ul style="list-style-type: none"> - Clusters representing visited places - Comparison of clustering algorithms performance - Assessment of the algorithms and selection of the best
TARGET	Quality of the clustering: <ul style="list-style-type: none"> ➤ Spatial quality: <ul style="list-style-type: none"> - Precision of the clustering > 86 %. - Recall of the clustering > 76 %.
Sub-Task 1.1.	
Incremental clustering	
Algorithm	Incremental (Kang et al. 2005)
Sub-Task 1.2.	
Incremental + Density-based clustering	
Algorithm	Incremental (Ye et al. 2009) + DBSCAN (Ester et al. 1996)
Sub-Task 1.3.	
Density-based clustering	
Algorithm	DBSCAN (Ester et al. 1996)
Sub-Task 1.4.	
Quality evaluation	
	<ul style="list-style-type: none"> • Contribution to the development and implementation of a general quality evaluation framework common for the 3 spatio-temporal sub-approaches. • Comparison of quality evaluation results using a collection of different parameter settings for each algorithm. • Assessment of the 3 algorithms.

Table 1. Description of *Task 1*.

Once the places visited by the user are determined, the stays performed at them as well as the transitions executed between these locations are detected and characterised. The goals of the second task include the representation of stays and transitions with characteristic values and the evaluation of the stays and transitions extraction performed by the best sub-approach.

TASK 2	
Characterisation of stays at visited places and transitions between them	
GOAL	<ul style="list-style-type: none"> - Representing stays at visited places with characteristic values. - Representing transitions with characteristic values. - Evaluating the stays and transitions extraction performed by the best algorithm.
RESULTS	<ul style="list-style-type: none"> - Tables containing user's dwell time at visited locations (<i>stays</i>). - Tables containing transitions between visited locations and representative derived values (<i>transitions</i>). - Evaluation of the stays and transitions extraction.
TARGET	<p>Quality of the stays and transitions extraction:</p> <ul style="list-style-type: none"> ➤ Stays: <ul style="list-style-type: none"> - Precision of the stays detection > 50 %. - Recall of the stays detection > 50 %. ➤ Transitions: <ul style="list-style-type: none"> - Precision of the transitions detection > 50 %. - Recall of the transitions detection > 50 %.
Sub-Task 2.1.	
Extraction of stays at visited places	
	<ul style="list-style-type: none"> • Development of a Java process to extract dwell time in visited places.
Sub-Task 2.2.	
Extraction of transitions between visited places	
	<ul style="list-style-type: none"> • Development of a Java process to extract transitions between visited places.
Sub-Task 2.3.	
Quality evaluation of the stays and transitions extraction	
	<ul style="list-style-type: none"> • Implementation of a specific quality evaluation framework for the extraction of stays at visited places and transitions between them from a user dataset. • Comparison of the stays extraction performed by the 3 sub-approaches and determination of the best. • Comparison of quality evaluation results using a collection of different parameter settings for the spatially best algorithm.
Sub-Task 2.4.	
Analysis of the stays and transitions extracted	
	<ul style="list-style-type: none"> • Implementation of indicators for the assessment of the accuracy of the stays and transitions extracted. • Analysis of temporal patterns in a user's mobility behaviour. • Development of graphic representations of stays and transitions between visited places.

Table 2. Description of Task 2

Within multiple sub-tasks, a second quality evaluation framework is used to assess the extraction of stays and transitions. The *stays* detection performed by the 3 sub-approaches is compared and the best approach is selected. The best algorithm is tested with different parameter values and the extraction results are compared in the quality evaluation.

Finally, an analysis of the stays and transitions detected by the chosen spatio-temporal sub-approach is developed. Two indicators are used for a general assessment of the extraction accuracy and a simple approach for mobility behaviour analysis of a user is developed and presented.

1.3. Outline

The introduction presented in this **Chapter 1** has offered an overview of context and relevance of the topic of movement behaviour analysis based on GPS. The scope of this thesis has been defined with a double aim and two main research questions have been posed. Research tasks and their objectives have been described.

Chapter 2 Theoretical Foundation, provides the theoretical framework for this work. The most relevant clustering approaches for this thesis are described as well as the quality evaluation framework used to assess the spatio-temporal clustering sub-approaches presented.

Chapter 3 Method, offers a description of the method this thesis bases on. A (I) data pre-processing phase is required before the phase of (II) determination of visited places. The third phase for (III) characterisation of stays and transitions completes the method.

Chapter 4 Implementation, describes the implementation of the algorithms used to develop the three clustering sub-approaches as well as the quality evaluation. The general workflow is defined and the determination of the parameter values tested is explained. The characterisation of stays and transitions is divided to present each extraction individually as well as the quality evaluation of such extraction.

Chapter 5 Results and Discussion, presents the results of the clustering and the parallel quality evaluation with the corresponding interpretations. The performance of the algorithms for stays and transitions extraction is compared and the accuracy of such extraction is analysed. Possible applications of the general approach developed in this work are suggested.

Chapter 6 Conclusions and Outlook, develops a reflexion about the value of the general approach developed. Contributions and problems of the work are analysed and further research is proposed.

2. THEORETICAL FOUNDATION

The **Knowledge Discovery in Databases** (KDD) process include the Data Mining step which consist of applying data analysis and discovery algorithms that produce a particular enumeration of patterns over the data (Fayyad et al. 1996). Spatial Data Mining focuses on large spatial datasets what is more difficult that mining non-spatial datasets due to the complexity of the spatial data types, relationships and autocorrelations (Shekhar et al. 2003).

Clustering is one of the major data mining methods and one of the initial phases in supervised learning and prediction. It is one process for analysis of data at a higher level of abstraction, organising together individual elements into coherent clusters according to a similarity condition. A cluster is a collection of objects similar between them and different to objects included into other clusters.

Clustering algorithms has been widely used in literature to obtain spatio-temporal patterns from location data. Dealing with personal location, these patterns represent user's personal places which in some cases are considered significant in her daily life.

A quality evaluation framework developed at SRFG has been implemented as part of this thesis contribution. This framework has been designed to enable a systematic comparison of different approaches suggested for the detection of the places visited by a user. As previously mentioned, this thesis relies on such framework to compare 3 different spatio-temporal clustering approaches testing the performance of several clustering algorithms.

2.1. Clustering approaches

There are different approaches for spatial data clustering. The most relevant algorithms for our work that have been found in literature can be classified in **partitioning-based**, **density-based** and **incremental** clustering algorithms.

2.1.1. Partitioning-based clustering

These algorithms basically divide the objects of the dataset between different clusters such that each object is exclusively in one subset. The main drawback of this method is that the user has to indicate the number of clusters expected before starting the clustering process itself.

K-Means

It is an algorithm present in many of the reviewed works. K-Means (Macqueen 1967) assigns randomly all points to a predefined number of desired clusters K represented by their centroid. The Euclidean distance between points and the cluster centre is calculated and each point is assigned to its nearest centroid. Depending on the points included in the cluster, the centroids are recalculated. Such iterative process is repeated until centroids remain the same.

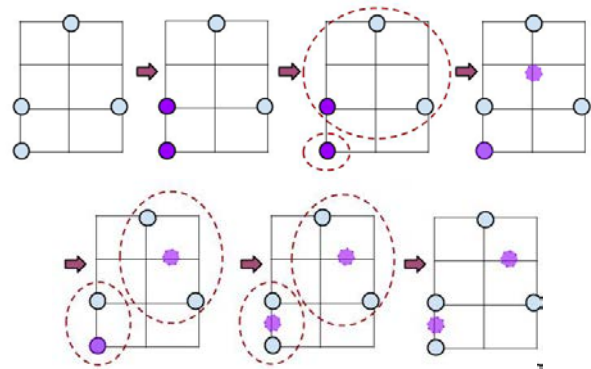


Figure 2. Working principle of K-Means (Zhang Xiao 2015)

However, different drawbacks have been reported in the literature (Changqing, Bhatnagar, et al. 2007), such as the necessity of specifying the number of clusters before the process starts or the high sensitivity to noise because of the inclusion of all the points in the clustering result. Furthermore, it only manages non-realistic spherical clusters and it is a non-deterministic algorithm given that the final result depends on the random assignment of points to the clusters at the beginning of the process. (Ashbrook & Starner 2003) used a version of the algorithm on a time-based adapted approach to determine significant places of users. (Kang et al. 2005) highlighted its computational costs and its necessity of including unimportant coordinates generating large and imprecise clusters. (Cao et al. 2010) compared it with *OPTICS* in their experiments, concluding that this density-based algorithm achieves better results.

2.1.2. Density-based clustering

This class of algorithms focuses on the number of points within a spatial region and the relation of neighbourhood between them. All the algorithms presented in this section build upon the widely used *DBSCAN*.

DBSCAN

The DBSCAN algorithm: *density-based spatial clustering of applications with noise* (Ester et al. 1996) has been widely used in research e.g. (Laasonen et al. 2004; Zhou et al. 2004). Further density-based algorithms have been developed on the basis of DBSCAN. Its most important characteristic is its ability for detecting clusters with different shapes within spatial databases of variable noise. Authors pointed out a good efficiency in large databases.

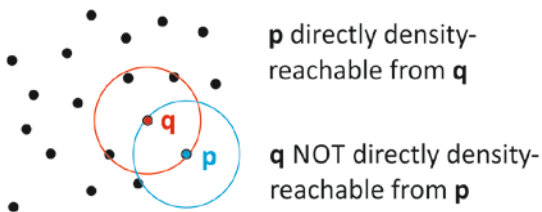
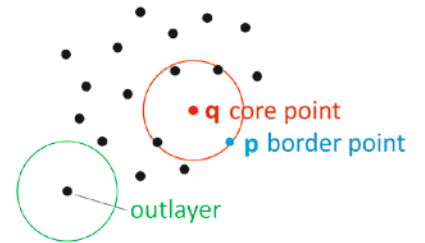
Two parameters are required as input: the radius of the neighbourhood **Eps** and the minimum number of points **MinPts** (density) that should contain. The density for each point depends on the number of points within the surrounding buffer of *Eps* radius. Parameters *MinPts* and *Eps* have to be set by the user and authors provide a method based in a k-dist graph so as to support the estimation of an optimal *Eps* value.

(Ester et al. 1996) define different concepts required for the adequate application of the algorithm:

- **Eps-neighbourhood of a point**

Defined by $N_{Eps}(p) = \{q \in D \mid dist(p, q) \leq Eps\}$

Two kinds of points are considered in a cluster: **core points** (inside the cluster) and **border points** (on the border). It is required that for every point *p* in a cluster *C* there is another point *q* in the cluster so that *p* is inside of the *Eps* neighbourhood of *q* and N_{Eps} contains at least *MinPts* points.



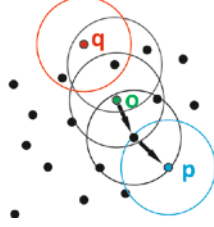
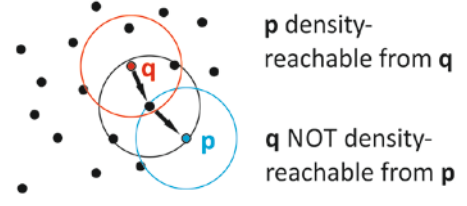
- **Directly density-reachable**

A point *p* is directly density-reachable from a point *q* (with respect to *Eps*, *MinPts*) if:

- 1) $p \in N_{Eps}(q)$
- 2) $|N_{Eps}(q)| \geq MinPts$ (core point condition)

- **Density-reachable**

A point p is density-reachable from a point q if there is a chain of points $P_1 \dots P_n, P_1 = q, P_n = p$ such that P_{i+1} is directly density-reachable from P_i . The notion of *density-connectivity* is introduced so as to cover the relation between border points.



p and q density-connected to each other by o

- **Density-connected**

A point p is density-connected to a point q if there is a point o such that both, p and q are density-reachable from o .

- **Cluster**

Let D be a database of points. A cluster C is a non-empty subset of D which satisfies the following conditions:

- 1) $\forall p, q$: if $p \in C$ and q is density-reachable from p . (Maximality)
- 2) $\forall p, q \in C$: p is density-connected to q . (Connectivity)

A cluster is defined to be a set of density-connected points which is maximal with respect to density-reachability.

- **Noise**

Let C_1, \dots, C_k be the clusters of a database D with respect to Eps_i and $MinPts_i$, $i = 1, \dots, k$. Then it is defined the noise as the set of points in the database D not belonging to any cluster C_i .

$$\text{Noise} = \{p \in D \mid \forall i: p \notin C_i\}$$

Algorithm description

1. Start with an arbitrary point p .
2. Retrieve all points density-reachable from p with respect to Eps and $MinPts$.
3. If p is a core point, a cluster is created.
4. If p is a border point, no points are density-reachable from p . Then,
5. Next point of the database is considered.

The main drawback of the algorithm is the difficulty to detect clusters of different densities. In (Changqing, Bhatnagar, et al. 2007; Chen et al. 2010) were reported several problems, such as not providing a strategy to efficiently handle large datasets and being very sensitive to the values of Eps (ϵ) and $MinPts$. Besides, (Montoliu et al. 2013) pointed out that DBSCAN tends to merge stay points with different semantic meaning in the same clusters.

OPTICS

Ordering Points to Identify the Clustering Structure (Ankerst et al. 1999) generalizes DBSCAN by creating a linear ordering of the points that allows the extraction of clusters with arbitrary values for ϵ . OPTICS does not produce a clustering of a data set explicitly; but instead creates an augmented ordering of the database

representing its density-based clustering structure. This cluster-ordering contains information which is equivalent to the density-based clustering corresponding to a wide range of parameter settings.

Different from DBSCAN, cluster memberships are not assigned. Instead, the object processing order and the information to assign cluster memberships is stored. According to (Ankerst et al. 1999) this information consists of two values for each object:

- **Core distance**

The core-distance of an object p is simply the smallest distance \mathcal{E}' between p and an object in its \mathcal{E} -neighbourhood such that p would be a core object with respect to \mathcal{E}' if this neighbour is contained in $N_{\mathcal{E}}(p)$. Otherwise, the core-distance is UNDEFINED.

- **Reachability distance**

The reachability-distance of an object p with respect to another object o is the smallest distance such that p is directly density-reachable from o if o is a core object. In this case, the reachability-distance cannot be smaller than the core-distance of o because for smaller distances no object is directly density-reachable from o . Otherwise, if o is not a core object, even at the generating distance \mathcal{E} , the reachability-distance of p with respect to o is UNDEFINED. The \mathcal{E} of an object p depends on the core object with respect to which it is calculated.

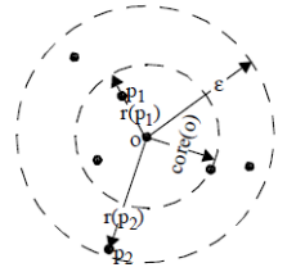


Figure 3. Core-distance (o) and reachability-distances for $\text{MinPts} = 4$ (Ankerst et al. 1999)

Each point is retrieved and the core condition is checked. When it is satisfied, the cluster grows including the neighbours of the point which are density-connected. In case the point is not a core object, the retrieval process proceeds on the next non-checked object of the database. The order of the points in the database does not influence the order of retrieval, which is determined by the distances between them.

OPTICS generates the augmented cluster-ordering consisting of the ordering of the points, the reachability-distance and the core-distance values. This information is sufficient to extract all density-based clusterings with respect to any distance \mathcal{E}' which is smaller than the generating distance \mathcal{E} from this order.

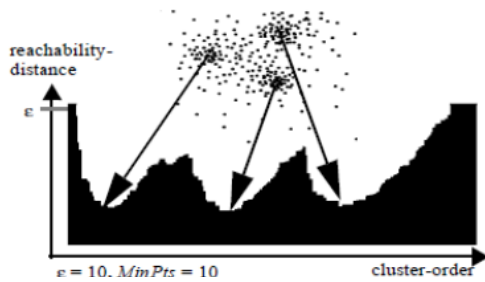


Figure 4. Reachability plot with 3 clusters (Ankerst et al. 1999)

An interactive analysis of the results is performed through reachability plots which are direct graphical representation of the cluster-ordering. The vertical axis represents the reachability distance and the horizontal reflects the order of clustering for each object.

The generating distance \mathcal{E} influences the number of clustering levels which can be seen in the reachability-plot. The smaller we choose the value of \mathcal{E} , the more objects may have an UNDEFINED reachability-distance. Therefore, we may not see clusters of lower density, i.e. clusters where the core objects are core objects only for distances larger than \mathcal{E} .

(Ankerst et al. 1999) presented also an algorithm for automatic analysis of the results of optics which idea is to identify potential start-of-cluster and end-of-cluster regions first, and then to combine matching regions into (nested) clusters. The reachability value of a point corresponds to the distance of this point to the set of its predecessors so clusters are dents in the reachability-plot. Basically, it is defined a reachability distance threshold and consecutive objects under it constitute a cluster.

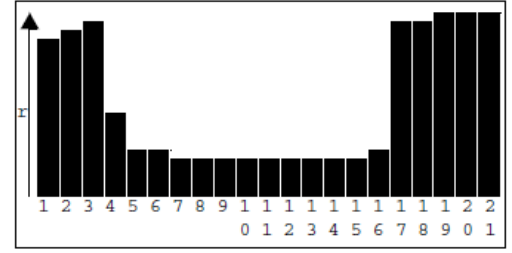


Figure 5. Reachability plot showing a cluster (Ankerst et al. 1999)

It provides better results on clustering points in data of varying density. (Zheng et al. 2008; Zheng et al. 2009) included OPTICS in their work to cluster user's transportation change points and (Ye et al. 2009) chose it to complement an incremental ("time-based") clustering approach.

DJ-Cluster

(*Density-and-Join-based*) is an algorithm presented in (Changqing, Frankowski, et al. 2007) which bases on DBSCAN, modified so as to deal with signal errors. Authors include a temporal pre-processing in order to guarantee that locations are really visited with enough frequency. Nevertheless, some useful information can be lost during this phase.

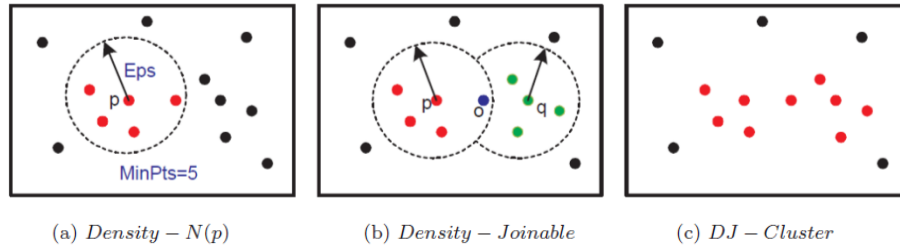


Figure 6. DJ-Cluster. (Changqing, Bhatnagar, et al. 2007)

DJ-Cluster requires at most a single scan of the data. For each point, it calculates its neighbourhood which consists of points within distance Eps , with the condition that there are at least $MinPts$ of such points. If no such neighbourhood is found, the point is labelled as noise; otherwise, the points are either created as a new cluster if no neighbour belongs to an existing cluster, or joined with an existing cluster if any neighbour belongs to the existing cluster.

Key properties:

- Every point is in exactly one cluster or is ignored as noise;
- There are always at least $MinPts$ points in each cluster;
- The algorithm partitions the input into non-hierarchical clusters;
- The clusters are mutually exclusive.

Authors reported great improvements over K-Means regarding recall and precision and a reduction in the time and memory requirements compared with DBSCAN. However, the algorithm tends to discover places with more GPS readings, or frequent places. Important and infrequent places may not be identified.

ST-DBSCAN

Proposal of (Birant & Kut 2007) is an improved version of DBSCAN. This algorithm is able to cluster points according to spatial, temporal and non-spatial features. It changes the epsilon parameter of DBSCAN by two parameters *Eps1* and *Eps2*. The similarity of points is defined by the combination of two density tests. The spatial dimension is considered with *Eps1* and *Eps2* serves for the non-spatial similarity measure.

Likewise DBSCAN, *MinPts* determines the minimum number of points that must constitute the neighbourhood around the considered point. The fourth parameter Δ_E is used to avoid the determination of clusters with small differences in the non-spatial values of the neighbouring points.

Authors point out the problems of the current density-based algorithms when dealing with clusters very close together. The values corresponding to border points in the cluster could be very different from those located in the opposite border whether the difference on values of neighbouring objects are small. Little value changes on neighbours may generate big value changes between starting and ending points of the cluster. However, the points should be within a certain distance from the mean value of the cluster.

So as to deal with the mentioned problem, the average value of a cluster is compared with the new (non-spatial) value on consideration. If the absolute difference between such values is above Δ_E , the new point is not included in the cluster. The average value of the objects of the cluster is referred as *Cluster_Avg* whilst the non-spatial value of an attribute is named *Object_Value*.

Another difference of this algorithm consists on the definition of a density-distance (*DensityFactor*). This distance is calculated as the division of the maximum density-distance by the minimum density-distance. These distances represent the largest and smallest distances between the point in consideration and its neighbours. It is defined as:

$$DensityFactor(C) = 1 / \left[\frac{\sum_{p \in C} density_distance(p)}{|C|} \right]$$

The *DensityFactor* of a cluster *C* represents the degree of density of the cluster. If *C* is a “loose” cluster the minimum density-distance will increase and the density distance will be very small, hence the *DensityFactor* of *C* will be close to 1. Otherwise, if *C* is a “tight” cluster the minimum density-distance will decrease and the density distance will be bigger, therefore the *DensityFactor* will result close to 0.

2.1.3. Incremental clustering

Several examples of this class of algorithm have been found in the literature, often referred as “*time-based clustering*”. Multiple approaches have been developed having all in common the computation of clusters incrementally as new location estimates are generated, therefore taking into consideration the time at which such coordinates have been obtained. Different coordinate-based systems have been used as source of location information.

The location-learning agent

(Marmasse & Schmandt 2000) presented the *location-learning agent* which observes user’s frequented locations over time and labels them. Their algorithm only recognizes locations where the GPS signal is lost. After the signal is lost within a given radius on 3 occasions, the agent infers that could be a building

and marks it as a relevant location. Nevertheless, some significant places (e.g. town square, parking lots) would not be discovered because GPS tracker is still able to obtain positioning information within these spaces. Also, not all buildings are opaque so data has to be analysed for stationary points.

Significant locations from GPS data

(Ashbrook & Starner 2002) worked on a two-step approach improving the previous work to determine significant locations. Also in this case, places are recognized where GPS signal is lost. Given that the signal loss still determines the detection of locations, mainly buildings are found whereas important outdoor places are ignored.

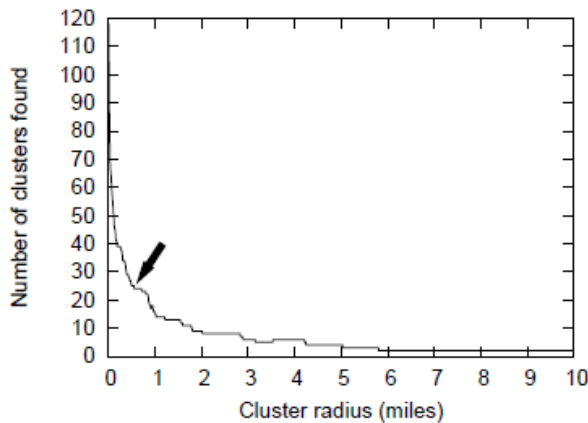


Figure 7. Relation between cluster radius and locations detected in (Ashbrook & Starner 2002).

generated cluster is prepared. Then, they look for a significant change in the slope of the curve, a “knee” which represents the radius just before the number of detections begins to converge with the number of points.

When the temporal difference between a track point and its previous one is greater than a threshold t , it is marked as a significant location. When analysing their data they observed a linear relationship between t and the number of locations detected. They arbitrarily determined 600 seconds as value for t .

Because of the fuzziness of locations, places are clustered with a variant of k-means. Authors aim to obtain locations with small radii but large enough to avoid representing the same significant location with different clusters. A plot with the number of locations detected in relation with the radius of the

Time-Based Clustering

(Kang et al. 2005) collected traces of location coordinates with a software client ([Place Lab](#)) which computes location coordinates by listening for RF emissions from known radio beacons in the environment (Wi-Fi fingerprinting).

Their approach clusters the parsed coordinates according to their associated timestamps while clusters where little time is spent are ignored. If the distance between incoming coordinates increases over a fixed threshold, a new cluster is formed.

Let us assume a user moves from place A to place B: while at place A, her coordinates are close (within some distance of each other) belonging to cluster A.

As user moves towards place B, her coordinates move away from cluster A and some small intermediate clusters are generated ($i1, \dots, i5$). A short time after arriving at place B, cluster B is formed.

If a cluster time duration is greater than a time threshold, it is considered to be a significant place. In Figure 8, clusters A and B are considered significant places while the others are ignored.

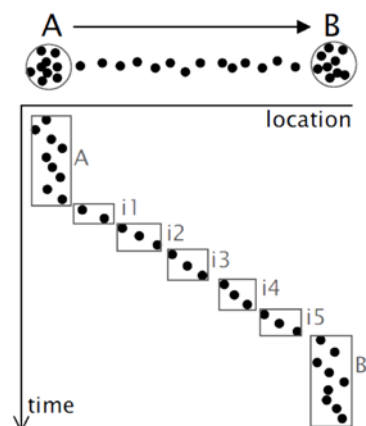


Figure 8. Illustration of the Time-Based Clustering algorithm (Kang et

The total number and size of extracted places depend on the distance and time parameters of the clustering algorithm. A greater distance threshold generates fewer, larger and less precise places. A smaller distance results in smaller and more precise places but may result in missed or fragmented places due to a possibly noise, scattered stream of coordinates.

Higher time thresholds results in places where the user has lasted larger timespans and may exclude places where less time was spent. Meanwhile, a smaller time limit increases the number of extracted locations where the user has stayed a short time. In order to detect frequent (and maybe shorter) visited places a second and smaller time threshold is used. Authors point out the need of adapting the parameters to the user's context, like the mean of transport.

As outlying coordinates are excluded clusters obtained are more likely to be fitted around significant places. Additionally, significant places can be extracted at run-time performing computations simple enough to run on an environment limited in resources such as mobile devices.

Nevertheless, (Changqing, Frankowski, et al. 2007) mentioned the lack of consideration of re-occurrence of readings at the same location making difficult discovering places visited with high frequency and short dwell time. Moreover, it was reported to require large storage capacity due to the continuous location data collection with very fine intervals.

i-cluster

(Hu & Wang 2007) presented an evolved version of the previous (Kang et al. 2005) which is referred as **TBC** in their work. They include a third time parameter t_{intv} and use an auxiliary data structure *Tempplaces*. *Tempplaces* stores those visited places with a stay duration smaller than t , that are temporally not considered as significant places by solution in (Kang et al. 2005). The additional threshold t_{intv} specifies the acceptable time for a revisit to the significant place. Two temporary clusters stored in *Tempplaces* would be merged if the user moves away from the current significant location and returns within t_{intv} .

Pseudocode of the algorithm is presented in Figure 9. **Pseudocode of the *i-cluster* algorithm (Hu & Wang 2007).** The spatial and temporal thresholds d and t are defined as in TBC while additional variables are used in *i-cluster*. The incoming GPS point is the input *loc*, whereas *cl* is the current cluster stored as its centroid. *Firsttimestamp*, *Lasttimestamp* and *Size* of the cluster are other self-descriptive variables and *Places* is a list to store the extracted significant places. The function *Distance()* calculates the distance from an input point to the cluster centroid and the function *Duration()* measures the time span of the user stay at the cluster. *Plocs* has the same use as in TBC (see previous algorithm).

i-Cluster

```

1:  if Distance(cl, loc) < d then
2:      add loc to cl // Add the new data to current cluster if it's within distance range
3:      clear plocs
4:  else
5:      if plocs.length > l then
6:          if Duration(cl) > t then
7:              add cl to Places // A significant place found
8:          else
9:              merged ← false // Add the temporary cluster to Tempplaces for potential merge
10:             add cl to the end of Tempplaces
11:             for j = Size(Tempplaces) - 2 to 0 do
12:                 tc ← jth cluster in Tempplaces
13:                 if (Firsttimestamp(cl) - Lasttimestamp(tc)) < tintv then
14:                     dist ← Distance(tc, clcentroid)
15:                     sum ← Duration(cl) + Duration(tc)
16:                     if dist ≤ d and sum ≥ t and merged = false then
17:                         merge cl, tc to a single cluster added to Places

```

```

18:         remove  $cl, tc$  from  $Tempplaces$ 
19:         merged  $\leftarrow$  true
20:     end if
21: else
22:     remove  $tc$  from  $Tempplaces$ 
23: end if
24: end for
25: end if
26: clear  $cl$ 
27: add  $plocs.end$  to  $cl$ 
28: clear  $plocs$ 
29: if  $Distance(cl, loc) < d$  then
30:     add  $loc$  to  $cl$ 
31: else
32:     add  $loc$  to  $plocs$ 
33: end if
34: else
35:     add  $loc$  to  $plocs$ 
36: end if
37: end if

```

Figure 9. Pseudocode of the *i-cluster* algorithm (Hu & Wang 2007).

Authors applied their algorithm to a very small sample of GPS points in their experiment. Optimal parameters values were determined as in (Kang et al. 2005) with a $d = 40$ meters, $t = 300$ seconds and $t_{intr} = 1200$ seconds.

This algorithm is reported to be space-efficient given that GPS data are not kept belonged to a cluster. Authors also pointed the limited space overhead induced and a tolerable time complexity of clusters merging in mobile devices. Their results showed a similar performance as the baseline *TBC* algorithm.

Stay Point Detection

More recently, (Ye et al. 2009) developed a similar algorithm as in (Kang et al. 2005). They introduce the notion of **stay points**. A stay point S represents a geographic region in which the user stays for a while. Therefore, each stay point carries its semantic meaning. Two types of stay points are considered: 1) user maintains stationary at a point for over a time threshold (enters a building); 2) user wanders around within a spatial region for over the time threshold (park, campus, etc.).

The mean longitude and latitude of the GPS points construct a stay point. In their experiments, a stay point is detected if individual spends more than 30 minutes within a range of 200 m. When stay points are detected, they use a stay point sequence $S = \{s_1, s_2, s_3, \dots, s_n\}$ to represent the individual's location history. The arrival time and leaving time respectively equals the timestamp of the first and last GPS point constructing this stay point.

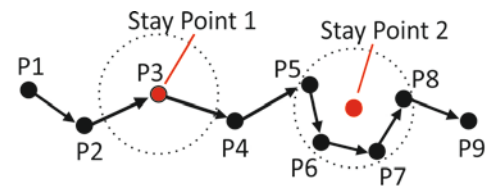


Figure 10. Parsed points and detected stay points

Because of the inaccuracy of positioning no two stay points have the same spatial coordinates despite, for instance, being the representation of the same significant place in different days. Hence, authors used a second modelling level to group stay points with the same semantic meaning. All individual's stay points are put into a dataset and clustered into several geographical regions. In comparison to k-means, density-based methods are capable of detecting clusters with irregular structure. They adopt the aforementioned clustering algorithm *OPTICS*; when there are at least a minimum number of points $MinPts$ within a search radius ϵ of an already clustered point, the new points are added to the cluster. In this way, a cluster is formed as a closure of points. Stay points of the same significant place are directly clustered into a density-based closure.

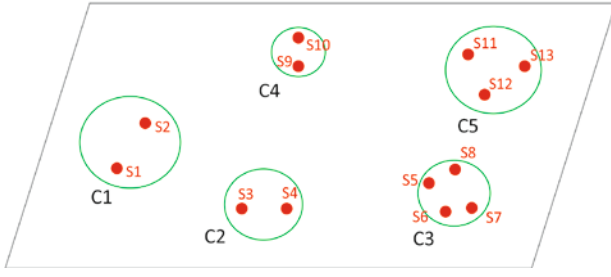


Figure 11. OPTICS clustering of stay points

After clustering the stay points, the individual stay point sequence is transformed into a location history sequence $C = \{c_1, c_2, c_3, \dots, c_n\}$. Each stay point S is substituted by the cluster C it belongs to. Meanwhile, the arriving time and leaving time of this stay point are retained and associated with the cluster. Therefore, there will be available records for visits to the same significant place on different days and/or moments of the day.

This algorithm performs offline whereas solution in (Kang et al. 2005) works online. Using both solutions as baseline in their experiments (Montoliu et al. 2013) obtained a better performance with this algorithm in comparison with (Kang et al. 2005) solution.

2.2. Quality Evaluation framework

This thesis has contributed with the implementation of a common evaluation framework, developed at SRFG (Venek et al. 2015), to test the effectiveness of different approaches and compare them equally. This work uses the evaluation framework for the quality assessment of the three **spatio-temporal clustering approaches** presented in this **thesis**; this means their performance quality so as to cope with the **task 1: detecting the user's visited places**.

Ground truth data

The dataset used to test the efficiency of the approaches presented in this thesis consists on **ground truth data** collected with GPS trackers by 4 people. Data collectors were researchers at SRFG that tracked their daily life and annotated the places visited. There are gaps in the data generated by the typical incidences a normal user experiments in the real world with this kind of system, i.e. battery constrains, not carrying or switching on the device, etc. Two different models of GPS trackers were used: *GPS Travel Recorder BT-Q1000XT* and *GPS Data Recorder CR-Q1100P* of *QSTARZ*. A sampling rate of 3 seconds was used for most of the collection so as to achieve an adequate data quality.

The 4 researchers built a trip protocol, registering the locations they visited during the 40 days data collection campaign. This route logs include the visited places and the intermediate stops (coordinates). Moreover, the starting and ending time (**tagged times**) of every trip carried out between these places were registered as well as the means of transport used: car driving, motorcycling, cycling, jogging and walking.

A post-processing of such data consisted on the extraction of the time invested in every stay at each location. Given that at the initial locations only was recorded the starting time of the trip, a time of 30 minutes was considered for the stay duration at such initial places.

The georeferencing of the locations was based on *OpenStreetMap* existing information. The 4 individuals manually annotated the OSM elements that represented the visited places. Then, the centroids of the stored elements were extracted and its coordinates were bound to the recorded places in the trip protocols. These visited locations will be referred as **tagged places** in the rest of this work.

Quality evaluation

The quality evaluation includes the spatial and temporal component on the task of detecting the **tagged places** provided as ground truth data. Consequently, measures to evaluate the spatial and temporal accuracy of the estimations have been designed (Venek et al. 2015). Moreover, the performances of the detection quality of the algorithms are compared in a **confusion matrix**.

Different parameter settings are tested for each algorithm to compare the results of the quality evaluation depending on the parameter values. Every algorithm generates a number of **detected places** or **detections** depending on its parameter settings.

Produced detected places are assigned to the tagged places for each test user. The way of relating both elements consisted on the consideration of a **circular buffer** of a determined **radius r** around the tagged places. If a detected place is located within such buffer, it is assigned to its corresponding tagged place. Hence, a tagged place is considered detected even if the coordinates are not exactly the same.

The buffer radius r is determined evaluating the diameter around tagged places (Venek et al. 2015). The square root of the area of each tagged place was calculated and the average of them was computed. Such average was approximately 35 meters. Then, the 95 % Horizontal Error of the Global Average Position Domain Accuracy of 18 meters was added twice to the average and a maximum diameter of 53 meters was obtained. (William J. Hughes Technical Center 2014). In (Venek et al. 2015) the minimum radius was chosen by the double of the GPS location error which is 18 meters. Within this range, the optimal radius was estimated by using one of the measures proposed (Q_{su}), trying to decrease its value. Finally, testing results with the 3 algorithms have suggested optimum results doubling the diameter so that the final considered radius is 53 meters.

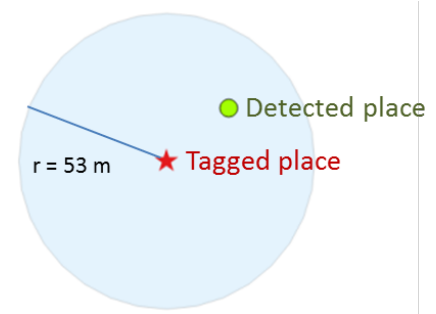
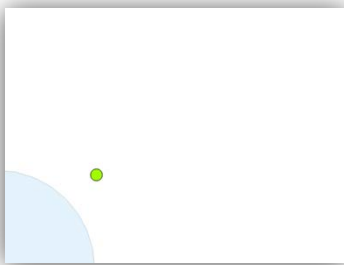


Figure 12. Circular buffer considered around a Tagged Place

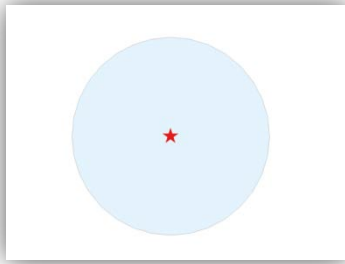
Now, spatial relations between tagged places and detected places are analysed so as to consider all the possible cases. In the graphics, tagged places are represented as *red stars* while detected places are depicted as *green points*. Circular buffers are displayed in light blue.

Five cases have been identified:



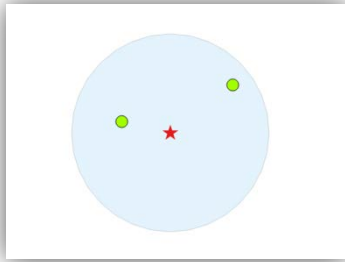
1) *Detection without tagged place*

Algorithm produces a detected place where no tagged place was reported.



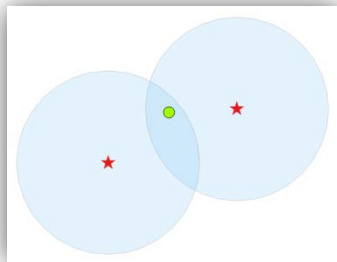
2) *Tagged place without detected place*

A reported tagged place cannot be assigned to any detected place.



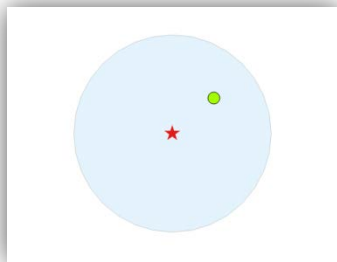
3) *Multiple detected places*

More than one detected place can be assigned to a tagged place.



4) *Multiple tagged places*

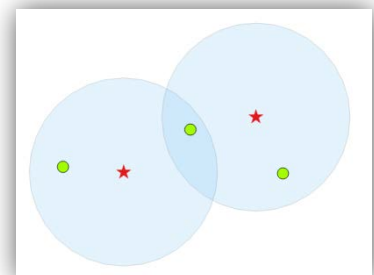
A detected place can be assigned to more than one tagged place.



5) *Detected place with tagged place*

One detection is assigned to one tagged place.

An additional case has been identified. It describes the combination of multiple detected places and multiple tagged places. In this case a detected place is related to more than one tagged place and at the same time it is one of multiple places detected within the circular buffer of another tagged place. To solve this situation, the multiple tagged places are considered as *detected places with tagged place*; the detected place is assigned to the spatially nearest tagged place. Nevertheless, one of the quality measures proposed evaluates the **uniqueness** of the detected places, dealing with the case *multiple tagged places*.



2.2.1. Quality measures

The described possible cases are considered and quantified under the defined quality measures. Four measures have been designed in order to evaluate the **spatial** and **temporal** accuracy of the detected places generated by each of the algorithms (Venek et al. 2015). The possible range of values for the measures vary from 0 to 1, corresponding 0 to the worst possible result on detecting a tagged place and 1 to the maximum quality of tagged places detection. The two first indices target the spatial domain whereas the third and fourth aim to evaluate the temporal domain.

1. *Spatial accuracy* (Q_{sa})

(Venek et al. 2015) This measure captures the degree of spatial accuracy of the clustering performed by the implemented algorithm. Accuracy is evaluated in relation to the distances between tagged places and detected places as well as the number of detected places.

Assuming a group of \mathbf{P} detected places and \mathbf{T} tagged places, a circular buffer of a determined radius \mathbf{r} is created around each of the tagged places \mathbf{T} . A detected place is spatially “assigned” to a tagged place if the Euclidean distance between the detected place \mathbf{P}_p and the tagged place \mathbf{P}_t is smaller than or equal to the radius \mathbf{r} (Venek et al. 2015). For each of the tagged places \mathbf{T} , the distances to its corresponding detected places \mathbf{P} are calculated. Hence, a distance matrix of elements $\mathbf{d}_{t,p}$ it is obtained for which $\mathbf{d}_{t,p} = 0$ if the Euclidean distance of tagged and detected place is larger than the radius \mathbf{r} .

$$\|P_p - P_t\|_2 = \sqrt{(x_p - x_t)^2 + (y_p - y_t)^2}$$

$$d_{t,p} := \begin{cases} \|P_p - P_t\|_2 & \text{for } \|P_p - P_t\|_2 \leq r \\ 0 & \text{otherwise} \end{cases}$$

Equation 1. Euclidean distance tagged-detected place

Then, the mean of the distances for each tagged place is computed. The values of the distance matrix are summarised and divided by the total number of assigned detected places received by the sign function of the distances:

$$\bar{d}_t = \frac{\sum_p d_{t,p}}{\sum_p \text{sgn}(d_{t,p})}$$

Equation 2. Mean distance

Now, the rightness of the detection of the tagged place is evaluated. For each tagged place, the *degree of correctness* \mathbf{q}_t of the detected places is computed. We can identify three cases; a fully correct detection if the mean of the distances is smaller or equal than half of the radius, a partially correct detection if the mean is between the radius and half of the radius and an incorrect detection otherwise.

$$q_t := \begin{cases} 1 & \text{if } \bar{d}_t \leq \frac{r}{2} \\ 1 - \frac{\bar{d}_t}{r} & \text{if } \frac{r}{2} < \bar{d}_t \leq r \\ 0 & \text{otherwise} \end{cases}$$

Equation 3. Degree of correctness

In Table 3 values simulated for the degree of correctness q_t of the detected places are presented, using the buffer radius of 53 meters. In this example, the measure is independent from the number of detected places considered for parameter derivation.

Finally, the spatial accuracy Q_{sa} can be computed as the division of the summarisation of q_t by the total number of tagged places T . Therefore Q_{sa} captures two cases: the multiple detected places and the detected place with tagged place. It reflects the accuracy of the correctly assigned detected places.

Table 3. Simulated values for q_t

\bar{d}	Radius r
	q_t
53.0	0.00
45.0	0.15
40.0	0.25
35.0	0.34
30.0	0.43
26.5	1.00

$$Q_{sa} := \frac{\sum_t^T q_t}{T}$$

Equation 4. Spatial accuracy

Table 4: Simulated values for Q_{sa} (Venek et al. 2015)

Q_{sa}	Total number of tagged places				
		10	20	30	40
	1	0.100	0.050	0.033	0.025
	2	0.200	0.100	0.067	0.050
	5	0.500	0.250	0.167	0.125
	10	1.000	0.500	0.333	0.250
	15		0.750	0.500	0.375
	20		1.000	0.667	0.500

Table 4 shows simulated values for the spatial accuracy. The sum of the degrees of correctness q_t of the detected places cannot be greater than the total number of tagged places.

2. Spatial uniqueness (Q_{su})

This measure (Venek et al. 2015) represents the uniqueness of recognizing tagged places. It is computed as one minus the division of the number of detected places assigned to multiple tagged places $N_{multiple}$ by the total number of detected places P .

$$Q_{su} := 1 - \frac{N_{multiple}}{P}$$

Equation 5. Spatial uniqueness

Table 5. Simulated values for Q_{su} (Venek et al. 2015)

Q_{su}	Total number of detected places				
		10	20	30	40
Num. of assigned detected places corresponding to multiple tagged places	1	0.900	0.950	0.967	0.975
	2	0.800	0.900	0.933	0.950
	5	0.500	0.750	0.833	0.875
	8	0.200	0.600	0.733	0.800
	10	0	0.500	0.667	0.750
	15		0.250	0.500	0.625
	20		0	0.333	0.500
	25			0.167	0.375
	30			0	0.250
	35				0.125
	40				0

The measure deals with the mentioned case 4: *multiple tagged places*, i.e. when a detected place can be spatially related to multiple tagged places. Therefore, a value of 1 for Q_{su} represents a situation in which no detected place can be related to more than one tagged place. In

Table 5 values simulated for Q_{su} are presented.

3. Temporal accuracy (Q_{ta})

This is the first of the two temporal measures presented in (Venek et al. 2015) and focus on the temporal accuracy of the clustering.

Likewise the spatial performance, the performance of the algorithms regarding the temporal dimension has been also investigated. The ground truth data collected includes the starting and ending times of the stays at the reported tagged places, i.e. the times at which the user arrives at the place and leaves it. The aim is to evaluate how well the stays are determined by the algorithms with respect to the real stays reported in ground truth data.

As explained before, the incremental algorithms cluster points which keep their timestamps so that it is possible obtaining clusters with their associated entry and exit times. Thus, every time a cluster is generated as a new detected place or is generated within an already known detected place, a visit or *stay* to such significant place is recorded with its corresponding starting and ending time. Meanwhile, the density-based algorithm requires the additional process to extract the stays at the significant places detected after clustering the whole dataset. In this case, the entry and exit time is obtained intersecting the GPS tracks with the clusters so as to extract the timestamps of the first and last point detected within a 53 m circular buffer around the detected place centroid.

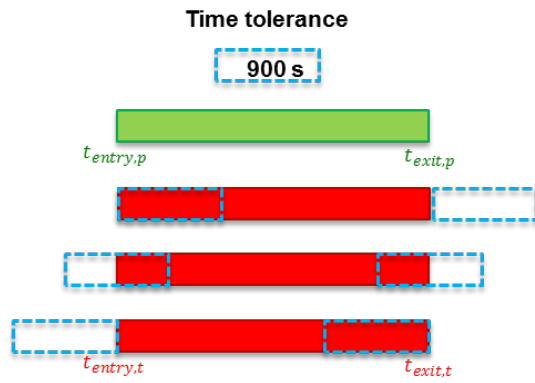


Figure 13. Time tolerance representation

A time tolerance has to be defined as done for the spatial accuracy assessment with the circular buffer radius. In (Venek et al. 2015) is specified a time interval as a tolerance deviation from tagged time to detected time. This means a tagged time is considered detected even if the mean detected time deviation is smaller or equal to the specified time interval in seconds. Two possible time intervals were tested: the first one relates to the defined minimum stay duration of 900 seconds, i.e. 450 seconds tolerance deviation at the entry and exit times. The other

interval corresponded to half of the minimum stay duration (450 sec), thus, 225 seconds as tolerance

deviation for entry and exit times. After tests, the first time interval of 900 sec was considered optimal and more realistic for the quality evaluation.

As the temporal accuracy is assessed with respect to the spatially assigned detected places (Venek et al. 2015), the number of detected places assigned to tagged places D is determined first. The objective then is to evaluate the correctness of the detected times related to the tagged places. The matter is determining if those spatially assigned detected places are also temporally correct.

The time differences between **detected** and **tagged entry point** as well as between **detected** and **tagged exit point** is determined for each tagged place. Then the mean of such difference values is computed. The unit of the mean is a time difference in seconds:

$$\Delta t_{t,p} = \frac{|(t_{entry,p} - t_{entry,t})| + |(t_{exit,p} - t_{exit,t})|}{2}$$

Equation 6. Mean time deviations from the tagged

The mean deviations from the tagged time $\Delta t_{t,p}$ which are larger than the specified time interval of 900 seconds are identified and to provide the deviation from the tolerance level. The result is a matrix with elements $t_{t,p}$:

$$t_{t,p} := \begin{cases} \Delta t_{t,p} & \text{for } \Delta t_{t,p} \leq 900 \\ 0 & \text{otherwise} \end{cases}$$

Equation 7. Value of t_p

The temporal accuracy Q_{ta} is equal to 0 in case the number of mean time deviations within the time tolerance of 900 sec is 0. Otherwise, its value is computed as one minus the sum of all time deviations divided by the product of the time interval (900 sec) multiplied by the number of identified time deviations.

$$Q_{ta} := \begin{cases} 0 & \text{for } \sum sgn(t_{t,p}) = 0 \\ 1 - \frac{\sum t_{t,p}}{900 * \sum sgn(t_{t,p})} & \text{otherwise} \end{cases}$$

Equation 8. Temporal accuracy

In Table 6. **Simulated values for Q_{ta}** values for Q_{ta} are simulated. The columns correspond to the number of correctly detected times and the rows indicate time deviations bellow 900 sec. In the first cell, one detected time's deviation from tagged place is smaller than 15 minutes which means Q_{ta} equals 0.933. The higher the mean deviation between tagged and detected times the smaller becomes Q_{ta} .

Table 6. Simulated values for Q_{ta} (Venek et al. 2015)

		Number of correctly detected times			
		1	5	10	15
Time deviations of < 900 sec	1	0.933	0.987	0.993	0.996
	2	0.867	0.973	0.987	0.991
	5	0.667	0.933	0.967	0.978
	10	0.333	0.867	0.933	0.956
	12	0.200	0.840	0.920	0.947
	15	0	0.800	0.900	0.933

4. Amount of temporal incorrectness (Q_{ti})

This measure reflects the *degree of incorrectness* or the amount of non-correctly detected temporal information (Venek et al. 2015). The number of spatially assigned detected places without any matching tagged time is computed as N_{incorr} . Then, it is divided by the number of **spatially assigned detected places D** . A **detected time** is considered as correctly assigned to a **tagged time** if it matches one of the times reported as ground truth.

N_{incorr} describes the number of detected places which are spatially assigned to tagged places and of which any **detected time** cannot be matched to a **tagged time**. On the other hand, N_{corr} indicates the number of detected places spatially assigned to tagged places and of which at least one **detected time** can be matched to a **tagged time**. D is obtained by counting the detected places which have been spatially assigned to one of the tagged places with time information (time data was not provided for all of the tagged places in ground

truth data). Hence, the amount of incorrectly matched detected times within the spatially assigned detected places is determined as:

$$Q_{ti} := 1 - \frac{N_{incorr}}{D} = \frac{N_{corr}}{D}$$

Equation 9. Amount of temporal incorrectness

Again, possible values have been simulated for Q_{ti} in Table 7. The closer to 1, the better the assignment of detected times to tagged times performs.

Table 7. Simulated values for Q_{ti} (Venek et al. 2015)

		D			
		10	20	30	40
N_{incorr}	1	0.900	0.950	0.967	0.975
	5	0.500	0.750	0.833	0.875
	10	0	0.500	0.667	0.750
	15		0.250	0.500	0.625
	20		0	0.333	0.500
	25			0.167	0.375
	30			0	0.250
	35				0.125
	40				0

2.2.2. Confusion matrix

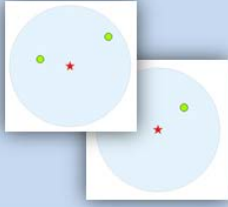
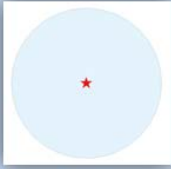


As presented in (Venek et al. 2015) a confusion matrix contains information about actual and predicted classifications performed by a classification system. This information generally includes four measures (Leroy 2011): **true positive**, **true negative**, **false positive** and **false negative**. True negatives (TN) would represent all possible GPS positions which are not considered as detected or tagged place, i.e. an infinite number. Therefore, our case is not a typical two class problem so **three** categories are considered:

- 1) True positive (TP). A tagged place is detected.
- 2) False negative (FN). A tagged place is not detected.
- 3) False positive (FP). A tagged place is detected where there is none.

Regarding the five possible cases previously identified, the confusion matrix excludes (3) *Multiple detected places* and (4) *Multiple tagged places* because there is not any differentiation between those two cases; a detected place is always assigned to the spatially closest tagged place. The category true positives (TP) counts unique tagged places with at least one detected place within its surrounding buffer. This means it does not distinguish if more than one detected place lies inside the buffer as cases (3) and (4) do.

The false negatives account for tagged places which have not be related to any detected place within that circular buffer. The detected places not related to any tagged place buffer are categorised as false positives (FP).

Table 8. Confusion matrix without True Negatives (TN)

Actual \ Predicted	Predicted	
	Positive	Negative
Positive	 TP	 FN
Negative	 FP	 TN

Precision and recall

In (Venek et al. 2015) precision, or positive predictive value, is computed by dividing the true positives by the total number of predicted positives (Leroy 2011). This value represents the proportion of correct detected places in relation with the total produced. It is a measure of the *exactness* or *quality* of the algorithm.

$$Precision = \frac{TP}{TP + FP}$$

Recall, or sensitivity, is the ratio of true positives and actual positives (Leroy 2011). This value represents the proportion of tagged places detected. Recall is a measure of the *completeness* or *quantity* of the algorithm.

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

A high precision indicates that the detected places produced by the algorithm are relevant as targeting true tagged places. On the other hand, high recall indicates most of the existing tagged places are detected by the algorithm.

The F measure describes the ratio of precision and recall. It is a weighted mean or rather harmonic mean of the two statistical values (Leroy 2011). This indicator determines the effectiveness of retrieval. The higher the F measure value, the better is the algorithm.

$$F \text{ measure} = \frac{2 * (Precision * Recall)}{Precision + Recall}$$

False Negative rate and False Discovery rate

In (Venek et al. 2015) the False Negative Rate (FNR) describes the relationship between false negatives and actual positives (Leroy 2011). This ratio indicates the amount of tagged places which are not detected.

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

The False Discovery Rate (FDR) computes the ratio of false positives and predictive positives (Leroy 2011). This measure shows the proportion of incorrect detected places, i.e. detected places not assigned to a tagged place.

$$FDR = \frac{FP}{TP + FP}$$

Both of the statistical measures can be derived by either precision or recall. The FDR can be related to the precision and the FNR can be related to the recall (Leroy 2011).

$$FDR = 1 - Precision \qquad FNR = 1 - Recall$$

3. METHOD

Detecting the places visited by the user in her daily life is a clustering task within this work. This thesis performs *incremental clustering*, *density-based clustering* and a *combination* of both as part of three sub-approaches suggested to solve this task. Two **incremental algorithms** are implemented within an already existing *Java* environment (at SRFG) and a well-known **density-based algorithm** is tested within the *ELKI*² *Java* framework. **Ground truth data** (GTD) consisting of GPS tracks from 4 volunteers will be processed so as to cluster the track points according to spatial and/or temporal conditions.

Then, a spatio-temporal quality evaluation framework is used to assess the clustering performance of the 3 sub-approaches in order to select the best algorithm or combination of algorithms for determining the places visited by the user.

Nevertheless, an initial phase is needed before the places detection. A pre-processing of the GTD is required in order to deal with the GPS errors affecting the data collection. This procedure was already designed at SRFG.

Finally, a novel approach is developed for **characterising** the different **stays at the places visited** by the user as well as the **transitions** between them.

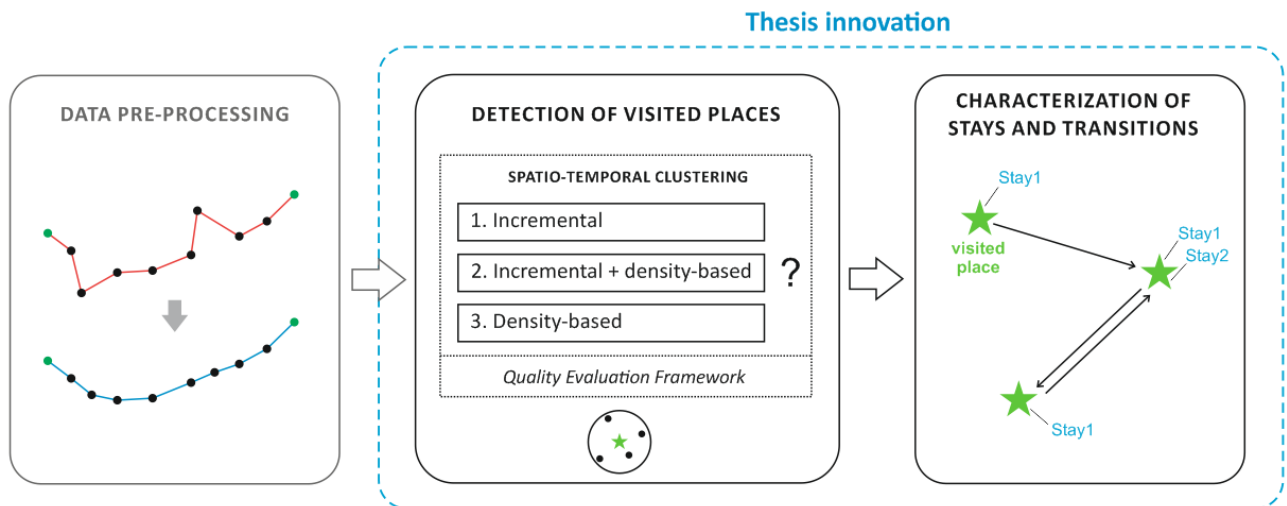


Figure 14. Method

3.1. Data pre-processing

Real life operation of GPS tracking devices implies several shortcomings regarding correctness. A pre-processing procedure for GPS trajectories **already developed** at SRFG (Gröchenig & Hufnagl 2015) will be applied on raw data with the **goal** of dealing with positioning errors. As a **result**, GPS tracks are obtained filtered and smoothed as the input for the subsequent clustering task.

POSITIONING ERRORS

² ELKI: Environment for Developing KDD-Applications Supported by Index-Structures (LMU Munich).

Errors can be classified in systematic and random errors depending on the cause (Griffin 2012):

- Systematic errors can depend on the geometry of the satellites: lack of visibility (less than 4) or a high Positional Dilution of Precision (PDOP) due to a short distance between them. On the user side, warm or cold start problems can occur.
- Random errors can be generated by issues related to the satellite orbit, the satellite clock or the receiver as well as ionospheric and tropospheric disturbances. The local geometry of the surroundings could determine a loss of signal as well as multipath effects.

So as to deal with random errors different **smoothing** techniques can be applied, whereas systematics errors are handled with **filtering** techniques. Nevertheless, filtering also produces the smoothing of a track.

(Gröchenig & Hufnagl 2015) have built their pre-processing scheme upon specific literature focused on diverse filtering and smoothing techniques. On this context, filtering consists on applying a process that removes from a signal some feature not desired, being a signal any time or spatial-varying quantity. Smoothing consists in creating an approximating function intended to capture relevant patterns in the data while excluding noise.

METHOD

➤ Filtering

Researchers at SRFG (Gröchenig & Hufnagl 2015) implemented a filtering based on velocity and acceleration. Unlike common filters based on same principle, their system not always removes the current point if a value exceeds the range. Instead, sometimes it is removed the previous and this is done iteratively.

Steps:

1. Removal of track points with equal timestamp

Obviously, a GPS device cannot be located at more than one location at a given time. Given that sometimes the GPS device reports consecutive locations with the same timestamp, the first track point is assumed to be the correct one and the following ones are removed.

2. Removal of track points with equal geometry

After losing the signal, some GPS devices send again the last known location with different timestamp and several times. In theory, while the device is static the identical geometry can be sent multiple times. However, in reality the location varies slightly even when the GPS is not moved at all and the coordinates estimated are spotted around the current location. Therefore, if consecutive track points have an equal geometry (not slightly variation), the first one is stored while the following ones are removed.

3. Correction of tunnels

When the GPS device enters a tunnel, some low-quality locations are sent before the signal is lost. After exiting such structures, around three seconds are required before receiving again a correct signal.

Hence, if the median of the sample rate of a track is lower than 1.1 seconds and two consecutive track points have a temporal separation greater than 5 seconds, this section is considered to be a tunnel and the following track points are removed:

- Current track point and successors within a time range of 4 seconds
- Previous track point
- All the predecessors to the previous within a time range of 3 seconds if have a time gap to their previous point lower than 5 seconds.

4. Track points with excessive acceleration (positive or negative)

If the acceleration between track points is greater than 20 km/h per second and the temporal difference to the pre-predecessor is smaller than 1.1 s, the previous point is removed. If a track point was removed the values of the current point are recalculated and the process is repeated. An acceleration of 20 km/h has been considered because it represents the maximum acceleration for cars.

5. Removal of spikes

If the speed difference between current and previous track point is above 40 km/h and between current and next track point is smaller than -20 km/h, the previous point is removed. If the speed difference between current and previous track point is below -40 km/h and between the current and the next one is greater than 20 km/h, the current track point is removed.

This process is done up to 3 times and the timestamps are not taken into consideration. Its application is only on low sampled tracks because high accelerations in higher sampled tracks are removed in the previous step.

➤ Smoothing

In this phase, researchers have used a kernel based approach with a triangular function as kernel function. Positions are selected for the average calculation and the time window is dynamic, decreasing with increasing velocity.

Initially, a fix time window of 5 seconds before and after the current position is used to calculate a mean velocity for such current position. This is a weighted arithmetic average with triangular function.

The obtained mean velocity allows for the calculation of the width of the time window, according to the following formula:

$$t_{wk} = \begin{cases} \min(20; 1 + \frac{30}{v_{mean_k}}), & v_{mean_k} > 0 \\ 20, & v_{mean_k} = 0 \end{cases}$$

Equation 10. Width of the time window (Gröchenig & Hufnagl 2015)

t_{wk} Width of the time window at position k in seconds
 v_{mean_k} Mean velocity at position k in m/s²

The time window is at least 1 s and at most 20 s wide.

Velocity [m/s]	Velocity [km/h]	Window width [s]	Num. of Positions in Window at Sample rate 1s
0	0	20	19
1	3.6	20	19
2	7.2	16	15
3	10.8	11	11
4	14.4	8.5	9
5	18	7	7
10	36	4	3
20	72	2.5	3
30	108	2	1

Table 9. Relation between time window width and mean velocity (Gröchenig & Hufnagl 2015)

The time window serves for the selection of the positions used for calculation of the average latitude, longitude and altitude. For each position in the window a weighting factor is calculated. Such factor is dependent on the time difference to the position currently calculated.

$$w_i = \max(0; 1 - \frac{abs(\Delta t_i)}{\frac{t_{wk}}{2}})$$

Equation 11. Weighting factor (Gröchenig & Hufnagl 2015)

Δt_i Time difference between current position k and position i (seconds)
 t_{wk} Window width at position k (seconds)

Then, the calculation of the average is performed with the following formula:

$$\bar{x}_k = \frac{(\sum_{i=k-N_{prev}}^{k-1} w_i \cdot x_i) \cdot N_{foll} + x_k + (\sum_{i=k+1}^{k+N_{foll}} w_i \cdot x_i) \cdot N_{prev}}{(\sum_{i=k-N_{prev}}^{k-1} w_i) \cdot N_{foll} + 1 + (\sum_{i=k+1}^{k+N_{foll}} w_i) \cdot N_{prev}}$$

Equation 12. Average of quantity at index k (Gröchenig & Hufnagl 2015)

x_k Quantity which is calculated (Lat, Lon, Alt) at index k
 \bar{x}_k Average of quantity at index k
 N_{prev} Number of positions in time window before position k
 N_{foll} Number of positions in time window after position k
 w_i Weighting factor for index i

In this formula, the sums of the positions before the current one are weighted with the number of the positions in the window after the current and reverse. This avoids a systematic shift of the position if positions in time window are not symmetrically distributed.

Figure 15. Original track and smoothed version.



3.2. Determination of visited places

3.2.1. Clustering

Clustering requirements

The mobility behaviour of the GTD³ collectors has specific characteristics such as different daily routines, short and long travelling distances or use of different transportation means. We are not interested in the semantic meaning of the user's visited places, the significance of the locations or the transportation mode of the individual. We are also not interested on the user's activity at her visited places.

In order to obtain compact and precise clusters, the system should ignore isolated points acquired during the transitions between the visited locations. This will facilitate the detection of visited places. Ideally, the algorithm should be capable of working autonomously identifying the changing collection of detected places and informing whether the user is at one of them.

We are interested in the dwell time at the detected places as well as the time spans invested during the transitions between them. The quality evaluation framework designed at SRFG and implemented during the development of this thesis calculates different spatial and temporal quality measures so as to evaluate

³ Ground Truth Data.

the performance of the different approaches. Hence, the implementation of the incremental algorithms should keep track of the timestamp of the points clustered so as to calculate the time invested within every detected place (stay duration). For the density-based algorithm, an auxiliary process will be implemented so as to extract such time from the whole dataset after the clustering process ends.

Selected clustering approaches

Taking into account our clustering requirements, two incremental clustering algorithms from (Kang et al. 2005) and (Ye et al. 2009) would be tested as previously stated. The work presented at (Ye et al. 2009) indeed requires a second clustering of the initial clusters. As mentioned before, their chosen complementary algorithm is the density-based OPTICS (Ankerst et al. 1999). Hence, for the accomplishment of this thesis objectives it was considered interesting emulating such perspectives with the adoption of **two** spatio-temporal clustering sub-approaches; an “*incremental*” and a combined “*incremental + density-based*”.

In order to conduct the comparison under a wider perspective a complementary **third** sub-approach has been adopted. The density-based clustering algorithm DBSCAN (Ester et al. 1996) will be tested as an independent solution, thus the final combination of clustering sub-approaches remains as:

- *Incremental* (Kang et al. 2005)
- *Incremental* (Ye et al. 2009) + *density-based*
- *Density-based* (Ester et al. 1996)

OPTICS (Ankerst et al. 1999) is a further development of the well-known DBSCAN algorithm. OPTICS generalizes DBSCAN and does not produce a clustering of a data set explicitly i.e. it basically allows for the determination of the optimum parameters for DBSCAN. Therefore, OPTICS will be used only initially so as to determine the best parameters for the application of DBSCAN within the second and third sub-approaches adopted in this thesis.

The incremental algorithms presented in (Kang et al. 2005) and (Ye et al. 2009) (from now on also referred to as “**Kang**” and “**Ye**”) will be implemented as independent Java classes within a previously existing Java framework. The third algorithm (DBSCAN) will be applied with the [ELKI](#) Java software from the [LMU University](#).

Nevertheless, as DBSCAN is based in density it requires using the whole collection of data under analysis so as to determine the user’s visited places according to the specific density of points for such dataset. This means the timestamps of the GPS logs are not relevant and points are clustered only according to their spatial relationship with their neighbouring points, i.e. points collected at different dates can be clustered together so that **final detected places do not carry any temporal information** apart from their spatial validity for the temporal **period** covered by the whole dataset.

Therefore, so as to accomplish our objectives an additional Java class has to be implemented in order to determine the **starting** and **ending time** of the stays at these detected places. It will imply a piecewise comparison of the original GPS tracks with the previously detected places (from the whole dataset) in order to collect the timestamps of the first and last GPS points detected within each visited place.

3.2.1.1. Incremental clustering

The time-based clustering presented in (Kang et al. 2005) has been chosen as first option because it fits our clustering requirements meanwhile authors reported a good precision (near 79 %) and recall (near 94 %) in their experiments on detecting places from 19-days long traces.

Incremental algorithm 1

```

cluster(loc)
input: measured location loc
state: current cluster cl,
       pending locations plocs,
       significant places Places

1:  if distance(cl, loc) < d then
2:    add loc to cl
3:    clear plocs
4:  else
5:    if plocs.length > L then
6:      if duration(cl) > t then
7:        add cl to Places
8:        clear cl
9:        add plocs.end to cl
10:       clear plocs
11:      if distance(cl, loc) < d then
12:        add loc to cl
13:        clear plocs
14:      else
15:        add loc to plocs
16:    else
17:      add loc to plocs

```

Figure 16. “Time-based” clustering algorithm (Kang et al. 2005).

The main parameters ***d*** and ***t*** are **distance** and **time** thresholds. ***cl*** (current cluster) is a temporal cluster, ***plocs*** is a list of pending coordinates used to filter outliers and ***Places*** is the list of detected places (“*significant places*”). When a new point (coordinates) is parsed to the algorithm, if its distance to the centre of the current cluster (***cl***) is < ***d***, the point is included in ***cl*** (lines 1-2); otherwise it is added to the list of pending coordinates ***plocs*** (17). If the temporal length of ***plocs*** grows > ***L***, they consider the user is really moving away from the cluster ***cl*** and a new cluster ***cl*** is started (5-13). ***Plocs*** is cleared any time a new point is within ***d*** meters from the current cluster ***cl*** centre (3, 10, 13). On leaving ***cl***, if more than ***t*** seconds were spend inside, then ***cl*** is added to the list of detected places ***Places*** (7).

After checking if the time length of the ***cl*** in consideration is greater than ***t*** and before adding it to the list of detected places ***Places***, a *merging condition* is tested: if the cluster’s centroid is at a distance < ***d***/3 of an already existing detected place in the list, then the cluster is merged with that place, otherwise it is added as a new detected place.

Despite the authors initially mention only two parameters distance ***d*** and time ***t***, it is easy realizing that a third parameter ***L*** has to be set. This time parameter is used to determine if the user is really moving from the current clustered position (candidate to be a detected location): “*If plocs grows beyond L seconds worth of coordinates, we decide the user is really moving away from cl and start a new cluster*”. There is no mention to any reference value in their work and obviously it will have an influence on the clustering results.

Moreover, the *merging condition* works itself as a density-based post-clustering of the initially detected places. Authors set the distance to check to ***d***/3, and a change in this value will also affect the number of detected places as well as our quality evaluation results afterwards.

3.2.1.2. Incremental clustering and density-based clustering

The work presented in (Ye et al. 2009) has been chosen as the second option, firstly because of the interest of this thesis on time-based clustering solutions and secondly because of its interesting combined approach incremental/density-based. Additionally, (Montoliu et al. 2013) reported a recall of 76 % and a precision of 81 % using this solution in one of their experiments on stay point learning.

As previously exposed, this approach combines an incremental clustering algorithm so as to obtain *stay points* for the user and a second clustering level in order to deal with the issue of the fuzziness of locations. The visited places often are detected multiple times during the period of GPS data analysed. Every time a place is detected, the resulting cluster of GPS points is different even for the same place. Therefore, also the centroid of such cluster varies.

Incremental algorithm 2

Input: A GPS log P , a distance threshold $distThreh$
and time span threshold $timeThreh$
Output: A set of stay points $SP = \{S\}$

```

1:   $i=0, pointNum = |P|$ ; // the number of GPS points in a GPS logs
2:  while  $i < pointNum$  do,
3:     $j:=i+1$ ;
4:    while  $j < pointNum$  do,
5:       $dist = Distance(p_i, p_j)$ ; // calculate the distance between two points
6:      if  $dist > distThreh$  then
7:         $\Delta T = p_j.T - p_i.T$ ; // calculate the time span between two points
8:        if  $\Delta T > timeThreh$  then
9:           $S.coord = \text{ComputMeanCoord}(\{p_k \mid i \leq k \leq j\})$ 
10:          $S.arrvT = p_i.T$ ;  $S.levT = p_j.T$ ;
11:          $SP.insert(S)$ ;
12:          $i:=j$ ; break;
13:        $j:=j+1$ ;
14:  return  $SP$ .
```

Figure 17. “Stay Point Detection” clustering algorithm (Ye et al. 2009).

Figure 17 presents the pseudo-code of the incremental algorithm applied to extract stay points from GPS data. The GPS tracks of the user are parsed so as to detect areas within a distance threshold in which the user stays for a period over a time threshold. In their experiments a stay point is detected if the individual dwells more than 30 minutes within a range of 200 meters. Each detected stay point stores temporal information as arrival time $S.arrvT$ and leaving time $S.levT$ respectively extracted from the timestamp of the first and last GPS point included in this cluster.

Figure 18 shows two incremental clusters obtained for two different days. Despite they represent the same place (building, park, square...) their centroids are different. Each centroid has an associated user’s **visit** or **stay** with a time start ts and time end te .

Authors then used OPTICS to perform a density-based clustering of the initial stay points (clusters) obtained by the incremental algorithm. Such initial clusters are represented by their centroids which now will be the elements clustered by OPTICS. The new clusters obtained with OPTICS constitute the final representation of detected places. Given that the arriving time and leaving time of each stay point was retained, a list of arriving and leaving times (stays) is stored for each detected place.

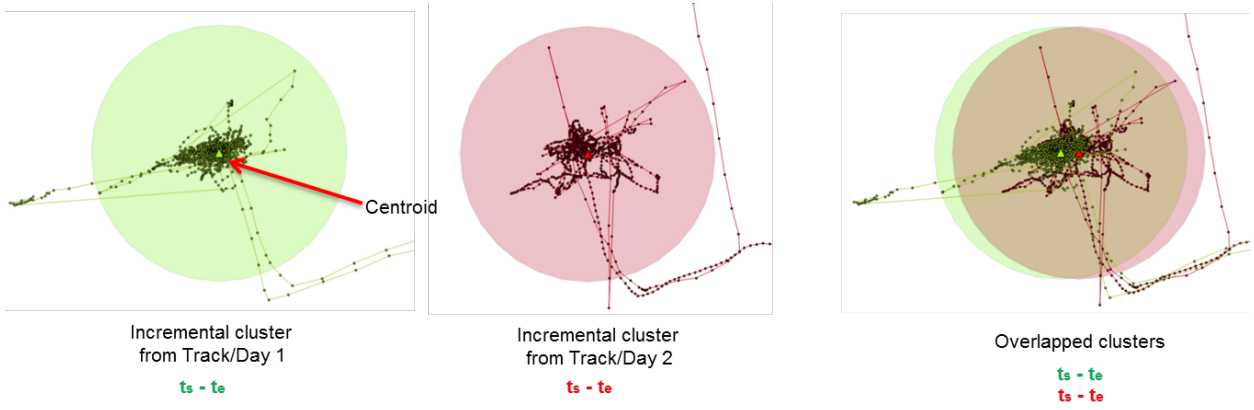


Figure 18. Incremental clusters from different days obtained at the same locations

As pointed out in the literature review (19), OPTICS is a generalization of DBSCAN which basically allows for the determination of an optimum value for DBSCAN's Epsilon parameter (Eps). Such value depends on the specific dataset and expected clustering results. Thus, DBSCAN is the algorithm that is finally used for the tests.

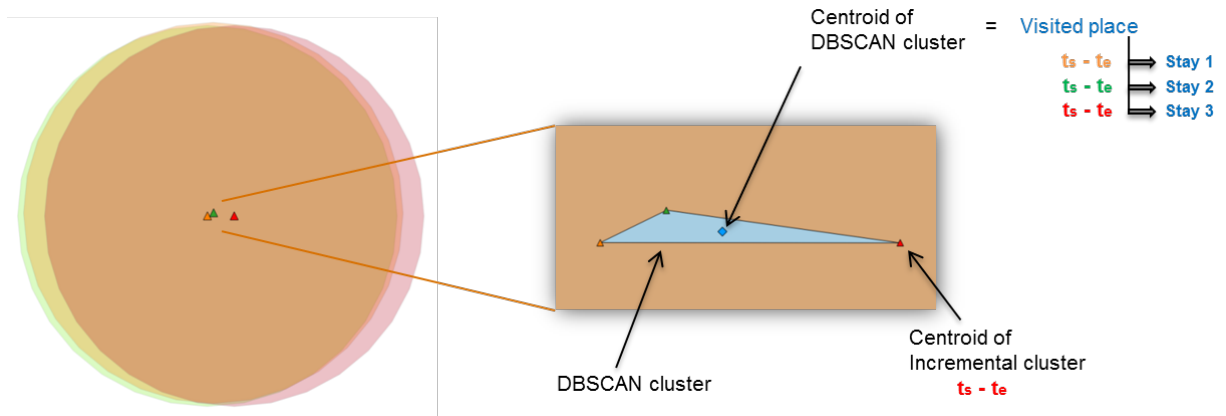


Figure 19. Overlapping incremental clusters and a DBSCAN cluster of their centroids

Figure 19 represents 3 overlapping incremental clusters obtained at different moments. DBSCAN has been applied to cluster the incremental centroids. The centroid of the DBSCAN cluster represents the new coordinates of the visited place whereas the stays performed in each incremental cluster are stored and assigned to the final place (stays 1, 2 and 3).

Initially, OPTICS was applied with ELKI in order to determine the optimal epsilon for DBSCAN. The minimum number of points $MinPts$ to include in the neighbourhood of a considered cluster was set to one because of the nature of the dataset. Many of the theoretical places detected by the incremental algorithm during the time period of the dataset are represented by only one stay point, i.e. a unique visit to a user's potential detected place during the period evaluated.

The optimum epsilon is chosen depending on the number of final clusters obtained and their relation with the tagged places reported as ground truth data. Then, such Eps and a $MinPts$ of one is used to apply DBSCAN to the initial clusters resulting from the incremental algorithm.

There are available Java implementations for DBSCAN although an alternative **own solution** was developed and tested for this thesis. This solution consists on the grouping of the clusters generated by the incremental algorithm (points) according to the distance between them. Then, a convex hull is

generated around the grouped points so as to calculate the centroid of every group. Further explanation is provided in section 4.1.2.

3.2.1.3. Density-based clustering

The well-known DBSCAN completes the comparison with a specific density-based approach. This option was considered interesting in order to compare the performance of the other alternatives which take advantage of the temporal dimension of the user's mobility data and do not focus only on the spatial dimension as DBSCAN does.

This algorithm has been chosen because of its ability for detecting clusters with different shapes within spatial databases of variable noise and a relatively good efficiency in large databases.

OPTICS is used as in the previous sub-approach. In this case, on the ground truth data of one of the users, with more than 400.000 points, so as to determine an adequate initial value for *Eps* and *MinPts*. Then, a group of parameter settings is chosen for testing the results of the clustering through the designed quality evaluation framework.

ELKI is also used to perform the clustering, given the huge improvement in performance this software provides. Unlike other Java implementations of DBSCAN, ELKI can use different index structures for sub-quadratic runtime and supports arbitrary data types and distance functions.

3.2.2. Quality evaluation

An integrated Java process allows for the automatic quality evaluation of each of the clustering performed by the presented sub-approaches. As described before, 4 spatio-temporal quality measures are calculated and a confusion matrix is built in order to obtain typical metrics relevant to assess the clustering performance.

- **Quality measures**

- *SPATIAL*

- 1. Spatial accuracy
 - 2. Spatial uniqueness

- *TEMPORAL*

- 3. Temporal accuracy
 - 4. Amount of temporal incorrectness

- **Confusion matrix**

- Precision
 - Recall
 - F measure

3.3. Characterisation of stays and transitions

3.3.1. Extraction of stays and transitions

Every place detected by the algorithms is given an identifier and is represented by the cluster centroid; this is a pair of coordinates. The user spends a variable amount of time at each visited place. Each one of the visits performed by the user is considered a **stay** at such location. All the stays at visited places will be *stored* associated to their corresponding visited place.

A **transition** refers to the change of location and time invested in the movement from a first detected place (**origin**) to a second detected place (**destination**). In both places, it has been previously identified a *stay* with its corresponding dwell time and therefore starting and ending time.

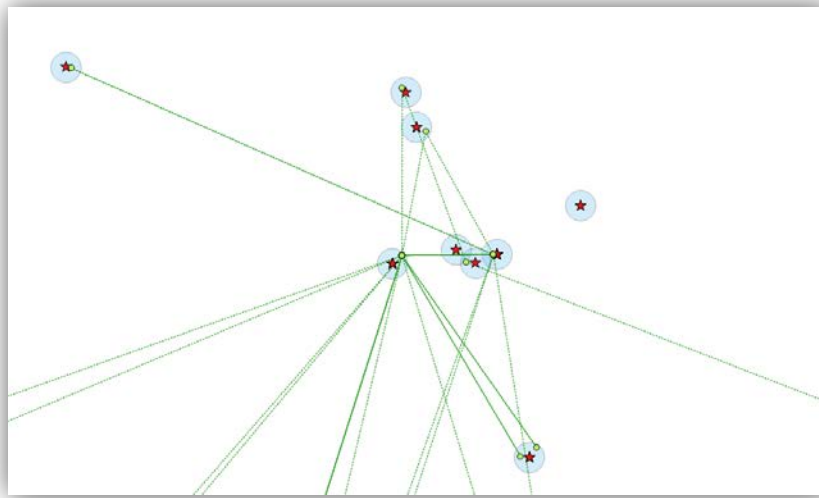


Figure 20. Representation of places and transitions.

3.3.2. Quality evaluation of the extraction of stays and transitions

The quality evaluation framework for the general performance of the different approaches has been presented in section 2.2. Now, it will be partially adapted to specifically evaluate the extraction of **stays** at visited places as well as **transitions** between them.

Once the 3 spatio-temporal sub-approaches are evaluated under the common framework, the best algorithm in terms of spatial precision and accuracy is chosen in order to assess the stays/transitions extraction it is able to perform. This means, once the clusters are generated (detected places) and spatially assigned to the tagged places, we test how well the algorithm performs on detecting the stays or visits to these places as well as the movements between them.

Hence, this secondary quality evaluation also includes a spatial and temporal component. In this case, the evaluation focuses on the detection of the reported **stays** at the **tagged places** and the **transitions** between them. As we are dealing with two tasks, they are considered separately but with the same evaluation structure: first, it is considered the relation between tagged time and detected time and then, the performance of the algorithm is compared in a **confusion matrix**.

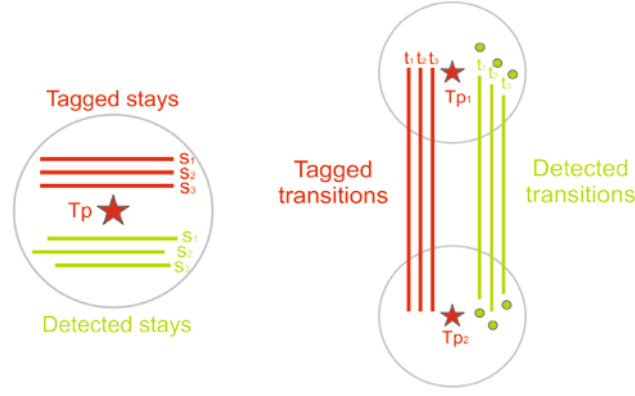


Figure 21. Representation of *tagged* vs. *detected* stays and transitions

Detected stays are related to the **tagged stays** and **detected transitions** are related to **tagged transitions**. A tagged stay is considered as found by the same approach as for calculating the general quality measures. Once a detected place has been spatially assigned to a tagged place through the *circular buffer* of 53 meters, the corresponding detected stay is matched to any of the tagged stays as when calculating the **Temporal accuracy (Qta)**. The same interval of 900 seconds will be considered as tolerance deviation from tagged time to detected time. We compare the detected stay with all the stored stays at the tagged place. Then, it is determined the time differences between **detected** and **tagged entry** point and between **detected** and **tagged exit** point for each tagged stay:

$$\overline{t_{t,d}} = \frac{|(t_{entry,d} - t_{entry,t})| + |(t_{exit,d} - t_{exit,t})|}{2}$$

We compute the mean deviations and if one of them is smaller or equal to the time interval of 900 seconds, the tagged stay is considered as detected. Thus, the **ending time** of the **transition** previous to such visit as well as the **starting time** of the subsequent **transition** has been also matched, unless we are considering the first detected stay. Then, obviously, it will exist only one related transition and its starting time (departure) will be equal to the ending time of that first stay.

As previously mentioned, the two extraction results are evaluated separately: first we consider the *proportion of time extracted* and then we create a *confusion matrix* for both the stays and transitions extracted.

➤ PROPORTION OF TIME EXTRACTED

It consists on the simple computation of the total time detected by the algorithm divided by the total time tagged in the ground truth data. It is calculated for both the stays and the transitions:

$$St_{ext} := \frac{St_d}{St_t} \quad Tt_{ext} := \frac{Tt_d}{Tt_t}$$

➤ CONFUSION MATRIX

Two confusion matrices will be generated also for this evaluation: the first one to analyse the real and predicted stays extracted by the algorithm, and the second one does the same for transitions.

The three classes considered are the following:

- 1) True positive (TP). A tagged stay/transition is detected.
- 2) False negative (FN). A tagged stay/transition is not detected.
- 3) False positive (FP). A stay/transition is obtained when there is no tagged stay/transition.

3.3.3. Analysis of the stays and transitions extracted

Two indicators are used to assess the accuracy of the stays and transitions detected by the best clustering sub-approach. Then, a simple analysis of the mobility behaviour of a user is developed as a sample of the potential applications of the general approach presented in this thesis.

4. IMPLEMENTATION

4.1. Determination of visited places

In the following sections, a description of the implementation of the algorithms and the auxiliary Java classes developed for this thesis is presented. Additionally, the developed workflow that included external software is described.

The two incremental algorithms have been implemented within a pre-existing Java framework at SRFG. For the application of the DBSCAN algorithm the already mentioned *ELKI* software has been used. The free and open source GIS software *QGIS*⁴ has served for visualization purposes as well as some processing tasks.

The implementation of the quality evaluation framework works as an integrated single process from the start of any of the clustering algorithms up to the generation of the output values for the quality measures and the confusion matrix. Furthermore, additional Java classes have been implemented for each algorithm so as to allow for the batch processing of the datasets with different parameter settings.

4.1.1. Incremental clustering (Kang)

As a first step, the pseudo-code (page 42) provided in (Kang et al. 2005) has been implemented and initial clustering tests were performed over the users' datasets. Then, the algorithm was improved in order to allow for the extraction of dwell times at the detected clusters (visited places). Further improvements included the extraction of transition times between the visited places and the development of common structures to provide input to the quality evaluation class.

Algorithm parameters

As described in 3.2.1.1 the algorithm has three parameters:

- Parameter **d** (distance)
- Parameter **t** (time)
- Parameter **L** (time, secondary. *Page 42*)

Naturally, different input values for these parameters will produce diverse clustering results. The development of a quality evaluation framework for all the algorithms provides a fast and homogeneous system to compare the performance of the algorithm depending on the parameter settings.

⁴ QGIS Desktop 2.8.2. [OSGeo](http://qgis.org/) 2015

(Kang et al. 2005) report from their experiments an optimum configuration for d and t so that the number of detected places keeps stable depending on the values of parameters. Initially, it is suggested a d between 30 and 50 meters and a t of 300 seconds. When validating with longer GPS traces, they reach the maximum recall and precision with a distance threshold of 30 meters and a time threshold of 1800 seconds. These have been the values used for the initial tests of our algorithm and afterwards for choosing a group of parameter settings so as to evaluate the performance once the quality evaluation framework is established. Then, it has been determined the best setting to specifically obtain maximum precision, recall and/or f measure.

Parameter L

It has been pointed out the absence of a reference value for parameter L in Kang's work. Thus a subjective value of 100 seconds was used for initial tests. Thereafter, a group of values was selected for each parameter so that 80 different combinations of parameters were tested in the quality evaluation.

Table 10. Parameters values for combinations

d (m)	t (s)	L (s)
20	600	10
30	900	30
40	1200	60
50	1500	90
		120

The analysis of the influence of L values on the clustering results poses a problem of multidimensional multivariate data analysis. Hence, a parallel coordinates plot was considered a useful visualization for the purpose of determining an optimum value for L. The free software [XDAT](#) is used to generate such graphic and multiple

versions of it after data filtering.

Figure 22 presents a plot of different values. The first column includes the tested values for L, whereas the third one displays the number of detections (clusters generated). The remaining columns correspond to the values obtained for the 3 most relevant measures of the quality evaluation: precision, recall, F measure. Given that the **F measure** is a weighted mean of precision and recall (Leroy 2011), it determines the effectiveness of the clustering. A graduated colour scale has been applied to represent the values of this measure so that it is easier tracing the relation between the considered variables.

Further graphics have been produced so as to facilitate the analysis of these relations. The relations for each tested value of L have been filtered in order to allow for a clearer interpretation.

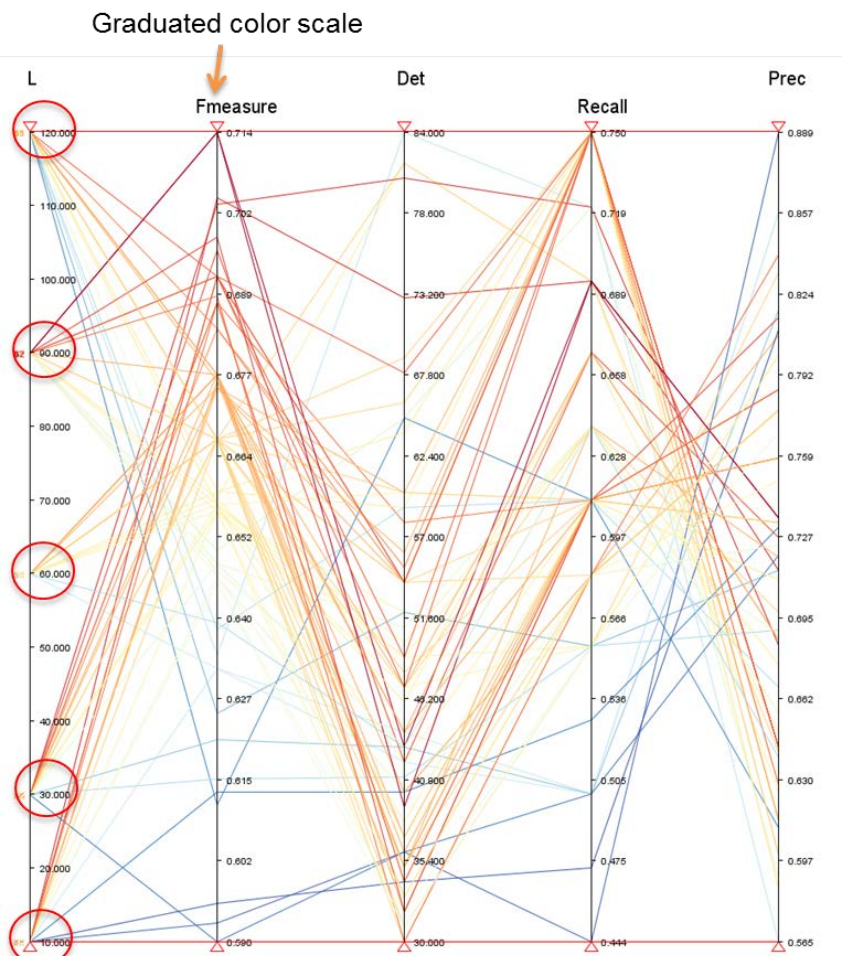


Figure 22. Parallel coordinates plot. Tested values for L

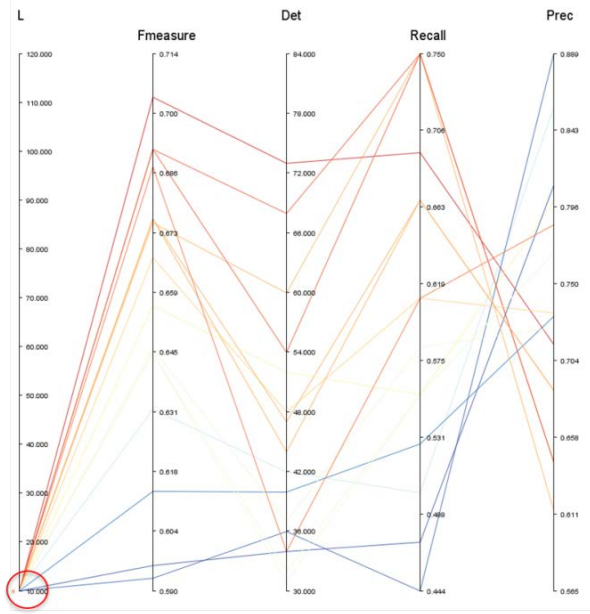


Figure 26. Parallel coordinates plot. L = 10

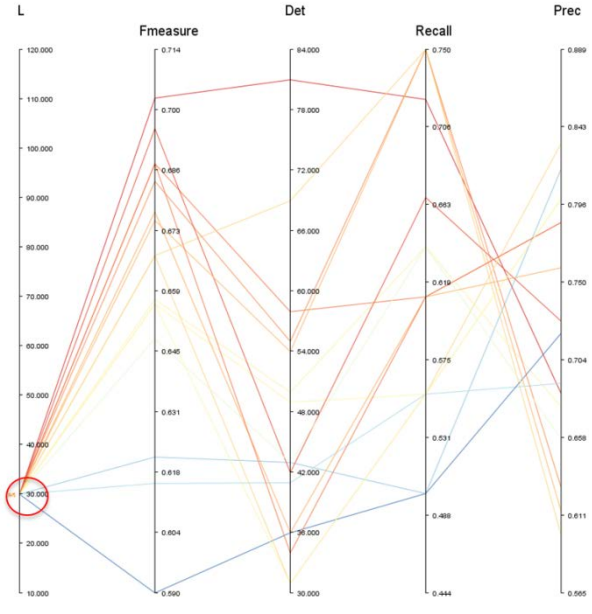


Figure 25. Parallel coordinates plot. L = 30

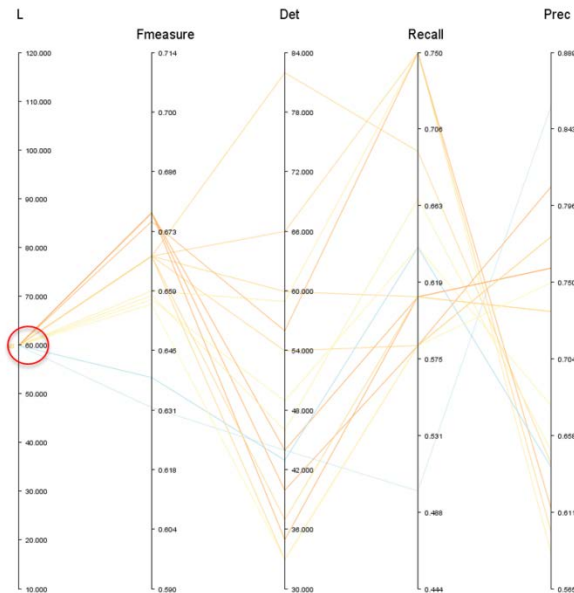


Figure 23. Parallel coordinates plot. L = 60

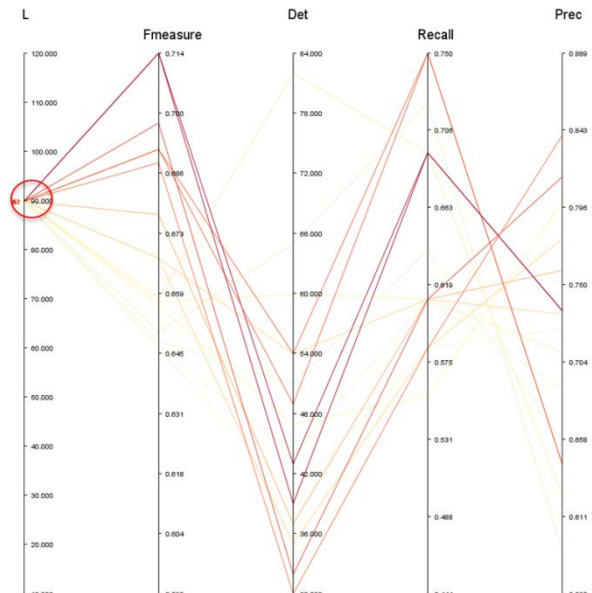


Figure 24. Parallel coordinates plot. L = 90

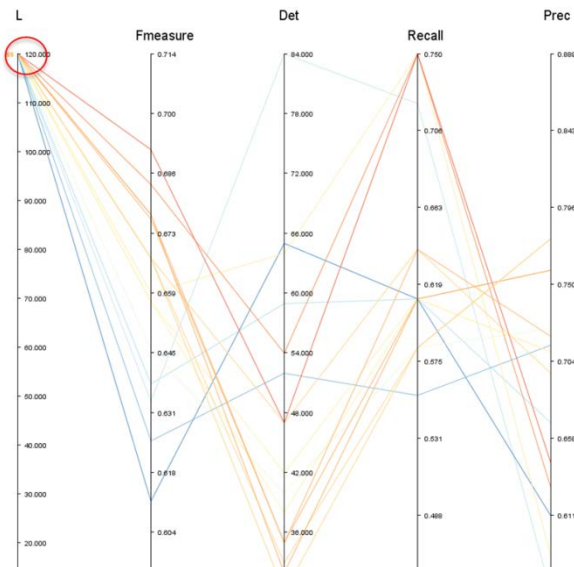


Figure 27. Parallel coordinates plot. L = 120

The value of $L = 90$ seconds (Figure 24) is identified as the optimum, given that the range of **F measure** values obtained is relatively narrow and located in the upper part of the F measure column. Moreover, the number of detected places generated with the algorithm is generally low whereas the range of recall values is higher than with other L (e.g. 10 or 30).

As described in page 42, *Kang* created this time parameter to allow the algorithm determining if the user is really moving from the current clustered position.

Parameter settings

Having determined an optimum value for L , different combinations of values⁵ for the other 2 parameters were chosen in order to test the clustering performance of this algorithm.

Table 11. Parameter values tested

d (m)	t (s)
20	300
30	600
40	900
53	1200
60	1500
70	1800
80	2100
90	
100	

4.1.2. Incremental + Density-based clustering (Ye + *ConvexHull*)

Likewise Kang's algorithm, the implementation from the original pseudo-code (page 43) is an improved version to allow for the extraction of stays and transition times. Moreover, an alternative to DBSCAN is developed in this thesis for the second level clustering performed in (Ye et al. 2009). Developing our own solution was considered interesting for the density-based clustering of the initial clusters generated by the incremental algorithm.

Grouping and convex hull

Our solution consists on the grouping of the initial clusters. This means considering each cluster centroid and evaluating the Euclidean distance which separates it from the neighbouring points. If such distance is below a **grouping radius** value, a new grouping cluster is generated with both points. Otherwise, a new cluster is created.

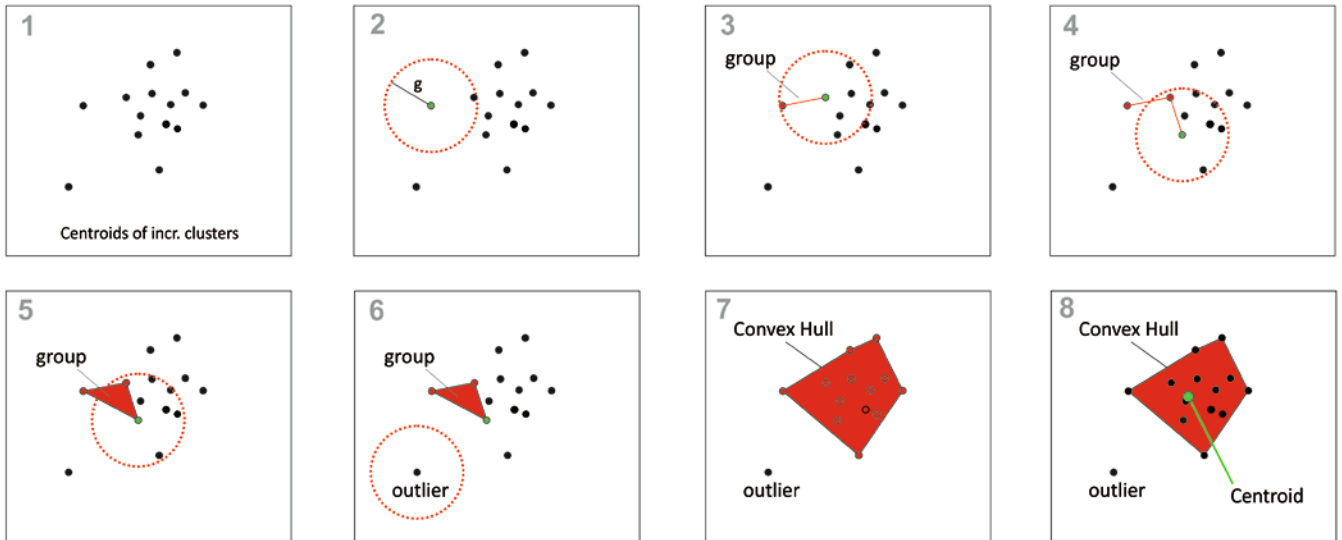


Figure 28. Process of incremental clusters grouping and convex hull creation

Once the groups are formed, a convex hull is created with the points belonging to each group. The centroid of each convex hull is the final location representing the visited place.

⁵ The series of values for parameter d is broken with $d = 53$ in order to make it coincident with the buffer radius used in the quality evaluation.

In order to choose an adequate *grouping radius*, OPTICS was applied with ELKI over the resulting clusters from Ye's algorithm and an optimum *Eps* of 40 meters was determined. This *Eps* or search radius was used for clustering with DBSCAN as well as for our solution.

In our case, we are not considering a minimum neighbourhood of points so as to create a new cluster as DBSCAN does (MinPts). Nevertheless, it is unnecessary due to the low number and tight distribution of detected places or clusters generated by the incremental algorithm. A comparison between the results of our solution and those obtained with DBSCAN showed a very similar performance in terms of recall, precision and F measure values.

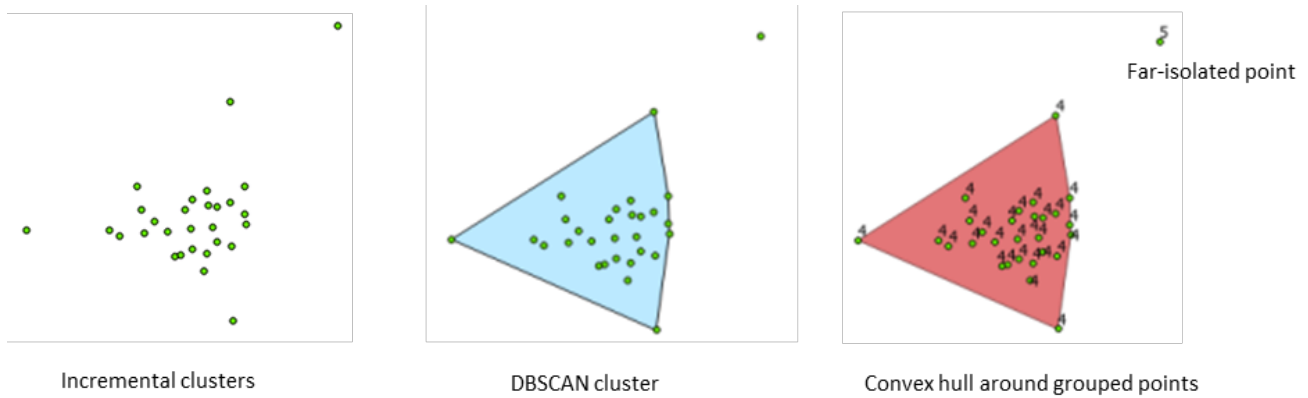


Figure 29. Comparison of clustering results: DBSCAN and own solution

Our implemented solution is more convenient for our objectives because it is **fast** and **integrated** in an already existing framework. Moreover, no **additional** software **tool** is required.

Algorithm parameters

In 3.2.1.2 the two main parameters of the algorithm are described:

- Parameter **d** (distance)
- Parameter **t** (time)

Experiments in (Ye et al. 2009) led them to determine an optimal distance threshold of 200 meters and a time of 1800 seconds. These values were used as reference for the initial tests of our implementation.

Parameter settings

Subsequently, as done with the first incremental algorithm, different values were chosen to build combinations of input values for the parameters. Results of the clustering will be also compared in our experiments.

4.1.3. Density-based clustering

The DBSCAN algorithm also requires the determination of the optimal parameter values. As pointed out, OPTICS enables the estimation of adequate values for DBSCAN parameters and ELKI is an interesting option for this purpose. ELKI has the advantage of testing additional algorithms within the same platform. This allowed us for the use of OPTICS with a high runtime performance over our ground truth

Table 12. Parameter values tested

d (m)	t (s)
25	300
50	600
100	900
200	1200
300	1800
400	2400
500	3000
600	
700	

datasets. These datasets have an average of approximately 440.000 track points collected during the 40 days campaign.

As previously stated, the extraction of temporal information from this density-based clustering required an additional processing of the datasets. Firstly, the whole dataset of each user is clustered to obtain the visited places corresponding to the complete analysed period. Secondly, the dataset is parsed again for a piecewise comparison of all the GPS tracks with the previously identified clusters (detected places). Both, the minimum stay duration and the buffer radius defined for the quality evaluation are used for the extraction of the stays.

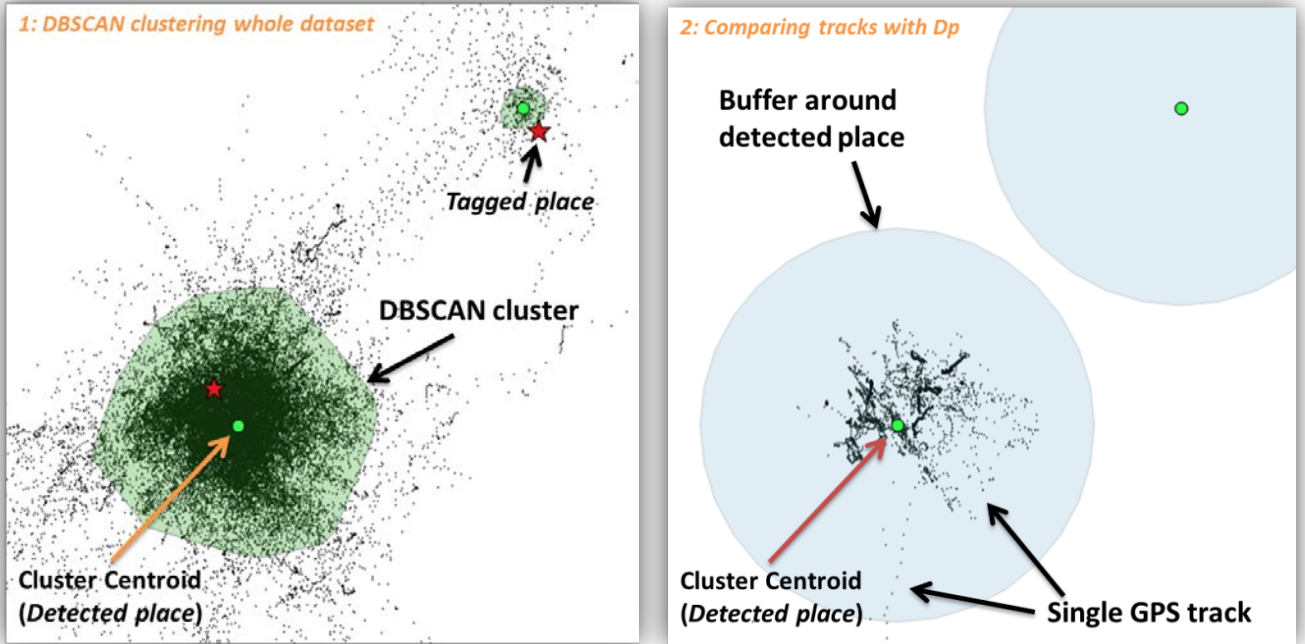


Figure 30. DBSCAN clustering and stays extraction

If a track point is identified within a buffer around a detected place, a new “visit” is created and the point timestamp is stored as starting time of such visit or stay. Following track points are parsed and the first one detected out of the buffer is used to extract the ending time of the stay. If the temporal length of the stay is smaller than the minimum stay duration it is discarded, unless another visit has already been detected within the same buffer so that both are merged. Then, if the total length of the merged stay is greater than 900 seconds, it is stored within the detected place. Additionally, if the time span between two consecutive stays at the same visited place is below 900 seconds, both stays are merged as one longer stay.

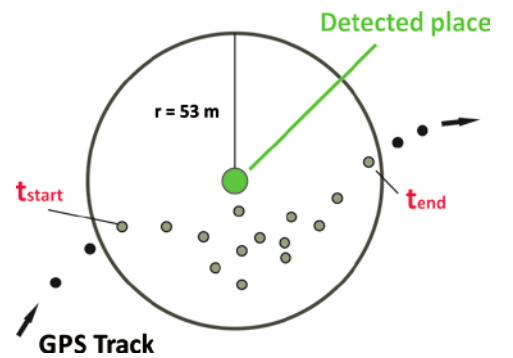


Figure 31. Details of stay extraction

Algorithm parameters

The two parameters of DBSCAN are independent from time (see 3.2.1.3):

- **Eps** (*search radius*)
- **MinPts** (*minimum points within the neighbourhood*)

A subjective search radius of 20 meters and minimum number of points of 50 was used to perform a clustering with OPTICS over the whole dataset of one of the users. Obtaining a reachability plot (page 19) for 0.4 million points was not feasible, thus an alternative visual analytical approach was adopted in order to estimate an appropriate *Eps* for initial tests with DBSCAN.

ELKI produces a text file with all the clustered points and their corresponding reachability distance. Such resulting file from OPTICS was displayed in QGIS, grouping the points according to their reachability in intervals of 2 meters. In Figure 33 points with a maximum reachability of 20 meters are represented whilst those not displayed have an infinite reachability, this basically means they would never be included in a cluster if used a search radius of 20 m.

As pointed out, OPTICS only requires a maximum epsilon and offers a cluster-ordering which contains information equivalent to the density-based clustering corresponding to a wide range of Eps values. In other words, the reachability distance value informs about the necessary epsilon in order to include such point in a cluster if DBSCAN would be applied over the dataset. Hence, the real locations visited (tagged places) where represented so as to enable us to have a sense about the needed Eps to generate clusters suited - in space and number - to the real locations we are supposed to detect.

Figure 32. Simple QGIS process

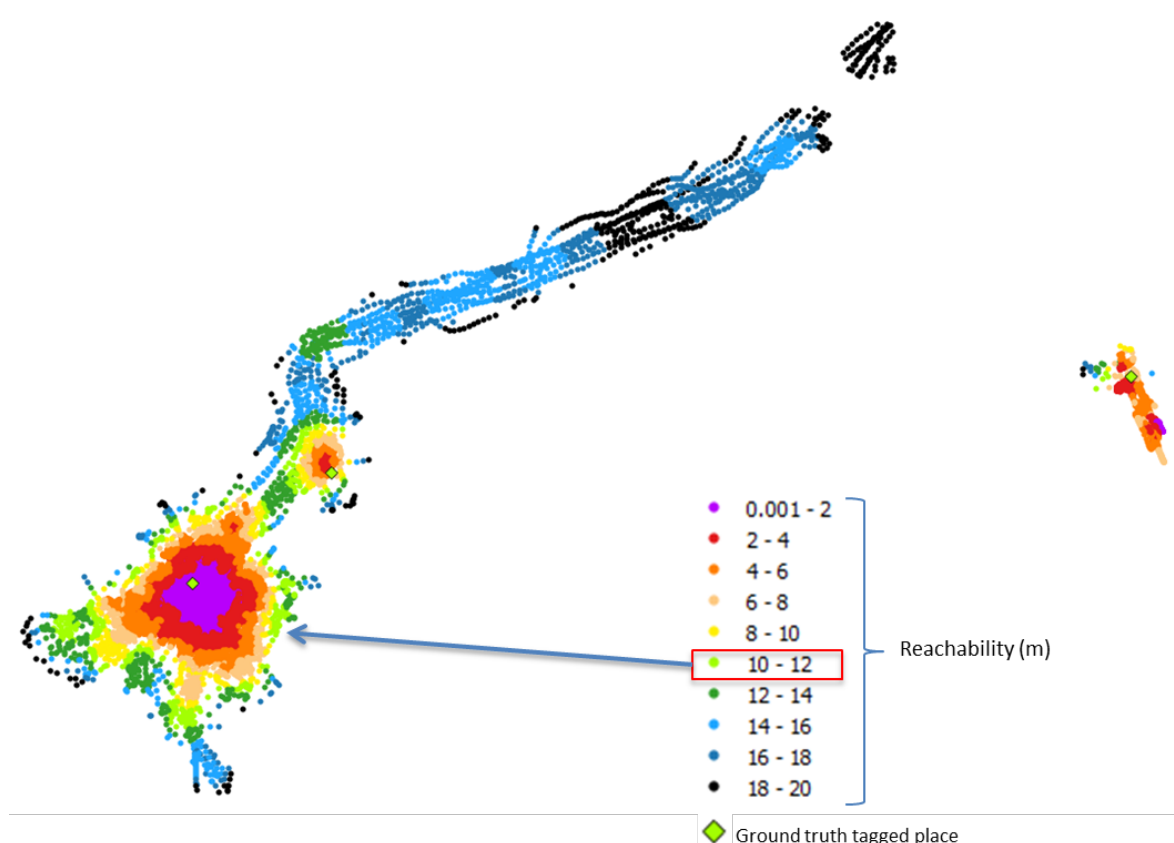
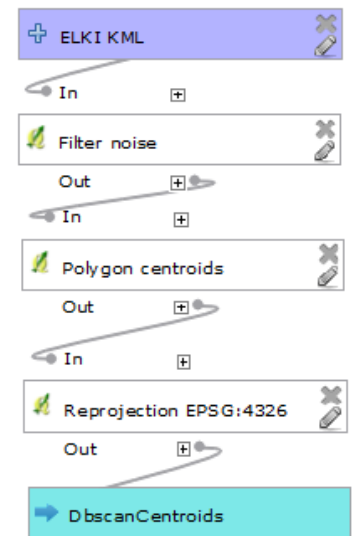
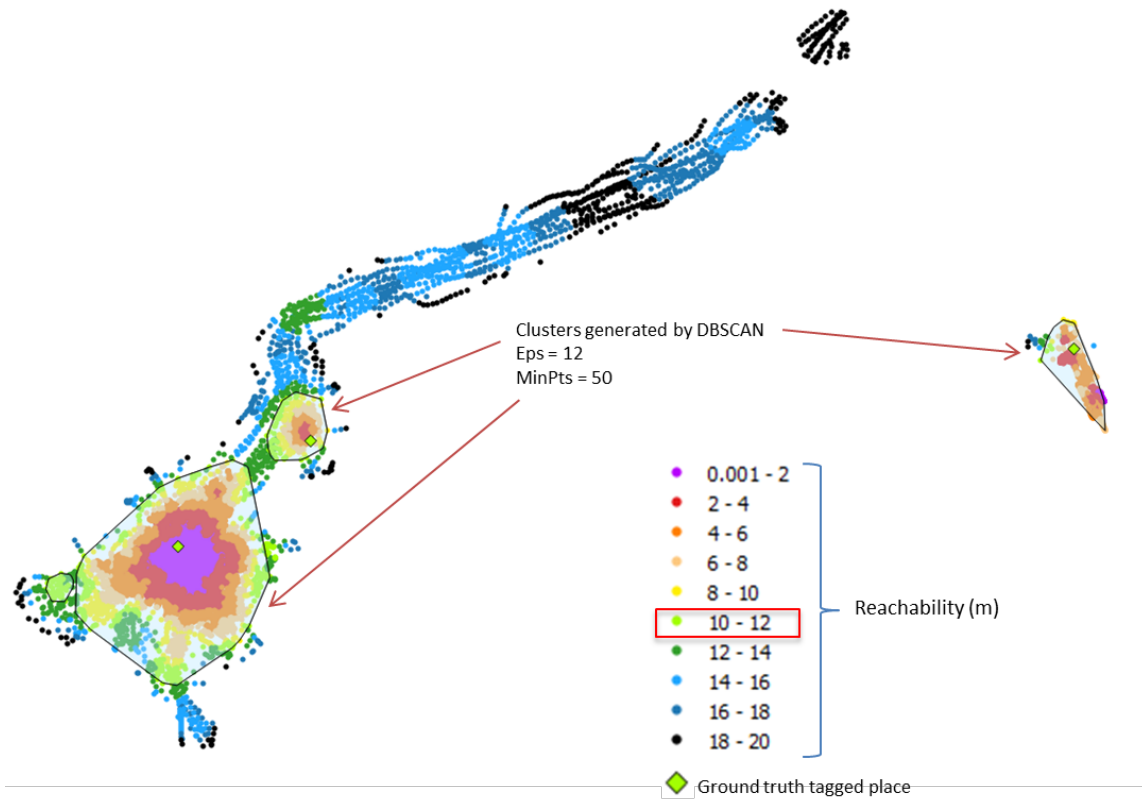


Figure 33. OPTICS results from ELKI visualised in QGIS

A visual exploration of the OPTICS results on display led us to choose an *Eps* of 12 meters because of the number of clusters expected and their position in relation with the ground truth tagged places. Therefore, a clustering of the whole dataset was carried out with DBSCAN and the determined *Eps*. In

Figure 34 resulting clusters are represented overlaid to the previous information. It is easy checking the reach of the clusters that include all the points with a reachability value from 0.001 (purple) to 12 meters (light green).



In case an *Eps* of 14 meters would be used, then the three clusters at the left would result in a single cluster including all the points up to the dark green coloured ones.

Figure 34. Results of DBSCAN and OPTICS clustering from ELKI

Parameter settings

For this algorithm different parameter values were also selected for testing in the experiments. In this case, the values are input for manual DBSCAN clustering in ELKI. Then, the files generated by ELKI are semi-automatically processed in QGIS so as to filter noise, calculate centroids of the clusters and change projection of their coordinates (Figure 32). Finally, such coordinates are fed into the Java process designed to extract the temporal information and trigger the quality evaluation.

Table 13. Parameter values tested (DBSCAN)

MinPts	Eps (m)
20	2
30	3
40	6
50	9
60	12
70	15
80	18
90	
100	
110	
120	

4.1.4. Quality Evaluation

Within this thesis the general quality evaluation framework (developed at SRFG) has been implemented based on the pre-existing Java environment and receiving further improvements from colleagues. The input for this process comes from any of the tested algorithms and the output consist on a quality evaluation log containing the parameter values used for the clustering as well as the quality measures and confusion matrix values corresponding to such parameter settings.

Parameters	Q1	Q2	Q4	Recall	Precision	Fmeasure	DetPlaces	GTDStayPr	GTDTraistRecall	stPrecision	stFmea	trRecall	trPrec	trFmea		
(20.0_300_90_20.0_53.0_900)	0.531	0.966	0.535	0.509	0.750	0.466	0.574	89.000	0.288	0.431	0.406	0.259	0.316	0.496	0.259	0.341
(20.0_600_90_20.0_53.0_900)	0.490	0.948	0.503	0.533	0.694	0.694	0.694	58.000	0.257	0.420	0.344	0.316	0.330	0.460	0.356	0.401
(20.0_900_90_20.0_53.0_900)	0.475	1.000	0.500	0.611	0.611	0.759	0.677	43.000	0.284	0.349	0.356	0.376	0.366	0.403	0.364	0.382
(20.0_1200_90_20.0_53.0_900)	0.489	1.000	0.460	0.667	0.611	0.759	0.677	40.000	0.262	0.331	0.333	0.370	0.351	0.360	0.342	0.351
(20.0_1500_90_20.0_53.0_900)	0.450	1.000	0.502	0.690	0.556	0.800	0.656	34.000	0.232	0.251	0.289	0.356	0.319	0.288	0.308	0.297
(20.0_1800_90_20.0_53.0_900)	0.393	1.000	0.518	0.731	0.500	0.818	0.621	30.000	0.220	0.222	0.267	0.348	0.302	0.230	0.262	0.245
(20.0_2100_90_20.0_53.0_900)	0.366	1.000	0.546	0.750	0.472	0.810	0.596	28.000	0.211	0.214	0.250	0.341	0.288	0.223	0.267	0.243
(26.5_300_90_26.5_53.0_900)	0.597	0.986	0.285	0.561	0.778	0.538	0.636	70.000	0.441	0.465	0.533	0.405	0.460	0.532	0.339	0.415
(26.5_600_90_26.5_53.0_900)	0.532	0.981	0.333	0.611	0.694	0.625	0.658	53.000	0.454	0.491	0.494	0.471	0.482	0.532	0.435	0.479
(26.5_900_90_26.5_53.0_900)	0.486	1.000	0.403	0.710	0.611	0.688	0.647	41.000	0.414	0.416	0.450	0.494	0.471	0.460	0.441	0.451
(26.5_1200_90_26.5_53.0_900)	0.464	1.000	0.424	0.759	0.611	0.733	0.667	37.000	0.392	0.400	0.433	0.513	0.470	0.417	0.436	0.426
(26.5_1500_90_26.5_53.0_900)	0.424	1.000	0.436	0.769	0.583	0.750	0.656	33.000	0.390	0.334	0.406	0.518	0.455	0.367	0.418	0.391
(26.5_1800_90_26.5_53.0_900)	0.442	1.000	0.553	0.760	0.583	0.750	0.656	32.000	0.349	0.323	0.367	0.489	0.419	0.345	0.414	0.376
(26.5_2100_90_26.5_53.0_900)	0.390	1.000	0.635	0.739	0.528	0.731	0.613	30.000	0.288	0.282	0.322	0.446	0.374	0.309	0.387	0.344
(30.0_300_90_30.0_53.0_900)	0.613	0.985	0.328	0.622	0.806	0.558	0.659	66.000	0.483	0.415	0.556	0.444	0.494	0.504	0.340	0.406
(30.0_600_90_30.0_53.0_900)	0.531	0.978	0.351	0.667	0.722	0.703	0.712	46.000	0.488	0.493	0.522	0.525	0.524	0.540	0.469	0.502
(30.0_900_90_30.0_53.0_900)	0.495	0.974	0.410	0.700	0.667	0.750	0.706	38.000	0.462	0.420	0.478	0.541	0.507	0.468	0.464	0.466
(30.0_1200_90_30.0_53.0_900)	0.464	1.000	0.363	0.750	0.639	0.767	0.697	35.000	0.447	0.404	0.472	0.578	0.520	0.432	0.469	0.449
(30.0_1500_90_30.0_53.0_900)	0.441	1.000	0.341	0.769	0.611	0.815	0.698	31.000	0.449	0.363	0.456	0.594	0.516	0.396	0.462	0.426
(30.0_1800_90_30.0_53.0_900)	0.457	1.000	0.395	0.769	0.611	0.846	0.710	30.000	0.426	0.341	0.428	0.579	0.492	0.381	0.465	0.419

Figure 35. Example of a quality evaluation log

Quality evaluation parameters

As described in 2.2 the quality evaluation requires two parameters:

- **BufferRadius** (m)
- **TimeInterval** (s)

The **BufferRadius** stands for the radius used for the buffer created around tagged places so as to spatially assign the detected places obtained with each algorithm processing. The **TimeInterval** is required for the calculation of the temporal quality measures. As described before, the values of these parameters were set to 53 meters for the buffer radius and 900 seconds for the time interval.

Quality of the 3 implemented solutions proposed in this thesis is evaluated. The algorithms are tested on the same *Intel Xeon W3565* CPU 3.20GHz machine with 10GB of memory running *Microsoft Windows 7*.

```

761 CSVFileWriter csvWriterStayTimesGTD = new CSVFileWriter(homeDirectory + "\\" + analysisMethod.toString().toLowerCase() + "\\" + analysisSubject + "\\GTDStayTime
762 csvWriterStayTimesGTD.writeLine("TpGeometry;TpID;VisitID;TimeStart;TimeEnd;Duration(s);NumMonth;NumDayStarted;DayOfWeek");
763
764 for (TaggedPlace tPlace : taggedPlaces) {
765     for (int tw = 0; tw < tPlace.getTimeElements().size(); tw++) {
766         // Writing each of the timeElements as an independent line so as to be able to represent both the dP and the coordinates of each visit to such dP
767         TimeElement teW = tPlace.getTimeElements().get(tw);
768         if (teW.getTimeEntry() == null) {
769             break;
770         }
771         String dayOfWeek = formatDay.format(teW.getTimeEntry());
772         // Date of the stay starting is extracted by parts
773         Calendar calEntry = Calendar.getInstance();
774         calEntry.setTime(teW.getTimeEntry());
775         int cYear = calEntry.get(1);
776         int cMonth = calEntry.get(2);
777         int cDay = calEntry.get(5);
778         int dayEnd = cDay + 1;
779
780         // For extraction of statistical data. If the stay only covers one day, it is written
781         if (isSomeDay(teW.getTimeEntry(), teW.getTimeExit()) == true) {
782             csvWriterStayTimesGTD.writeLine(tPlace.getLocation() + ";" + tPlace.getNumberId() + ";" + tw + ";" +
783                 + dateTimeFormat.format(teW.getTimeEntry()) + ";" + dateTimeFormat.format(teW.getTimeExit()) + ";" + teW.getDuration() + ";" + (cMonth + 1)
784                 + totalGTDStays = totalGTDStays + 1;
785         } else {
786             // If the stay affects more than one day, time of the stay during first day is written and rest of the time is written in the following day */
787             Calendar calEnd = calEntry;
788             calEnd.set(cYear, cMonth, dayEnd, 0, 0, 0);
789
790             Date endDate = calEnd.getTime();
791             Date endDate1 = new Date(endDate.getTime() - 1); // First part of the stay ends just one millisecond before the following day: 23:59:59
792             String dayOfWeekCont = formatDay.format(endDate);
793
794             csvWriterStayTimesGTD.writeLine(tPlace.getLocation() + ";" + tPlace.getNumberId() + ";" + tw + ";" +
795                 + dateTimeFormat.format(endDate1) + ";" + dateTimeFormat.format(endDate) + ";" + teW.getDuration() + ";" + (cMonth + 1)
796                 + totalGTDStays = totalGTDStays + 1;
797         }
798     }
799 }

```

Figure 36. Piece of code from the quality evaluation class

4.2. Characterisation of stays and transitions

4.2.1. Extraction of stays

The **duration** of a user's **stay** within a detected place will depend on the input values for the time parameter of every algorithm, in the case of the incremental approaches. Nevertheless, it has been mentioned the lack of consideration of the time by DBSCAN. Hence, an auxiliary Java class is implemented so as to extract these stays from the user's dataset.

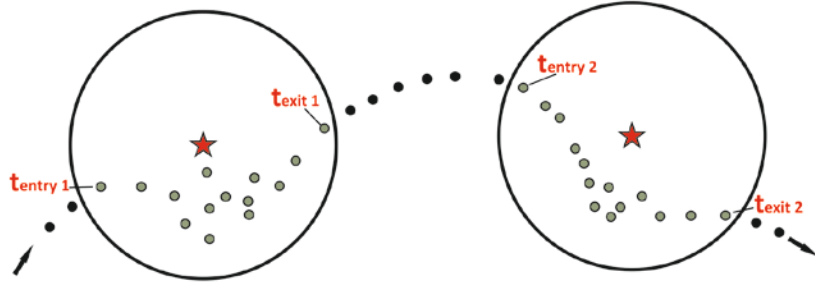


Figure 37. Representation of two incremental clusters and GPS points involved

The figure above illustrates two incremental clusters of track points. The red stars represent the centroid of visited places. Basically, the duration of such stays is calculated subtracting the timestamp of the last GPS point included in the cluster from the timestamp corresponding to the first one; both times are respectively stored as **starting (tentry)** and **ending (texit)** times of the stay. Moreover, the stay or visit is uniquely identified within its corresponding detected place.

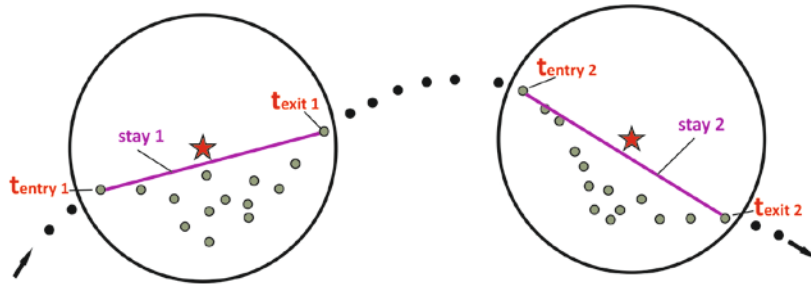


Figure 38. Representation of visited places and stays at them

CSV files which tabulate the stay times extracted by the algorithm or from the GTD are generated as output of the quality evaluation. These tables contain extracted and derived information from the user dataset. A section of one of the *detected stays* table is shown in Table 14.

First column contains the different detected place IDs whereas the second column display the ID of every of the stays or “visits” done to such detected place. Table is ordered by *DpID* and *VisitID* so times (*Start* and *End*) follow a chronological order with a quite continuous series because the *Detected Place 1* is one of the most visited places by this user (Home place). Some VisitIDs are repeated because our *quality evaluation* process splits the visits to “sleeping places” one millisecond before midnight and distributes the duration of the stay between the two affected days. Hence, the subsequent data mining can consider the correct amount of time stayed at each place on the corresponding weekday. The fifth column stores the calculated duration of every stay in seconds. The rest of the columns contain month, day of month and weekday of the stay.

Table 14. Example of stays table

DpID	VisitID	TimeStart	TimeEnd	Duration(s)	Month	Day	WD
1	0	12/08/2014 16:27	12/08/2014 17:48	4904	8	12	Tue
1	1	13/08/2014 16:28	13/08/2014 17:10	2488	8	13	Wed
1	2	13/08/2014 17:45	13/08/2014 23:59	22450	8	13	Wed
1	2	14/08/2014 00:00	14/08/2014 08:16	29814	8	14	Thu
1	3	14/08/2014 16:14	14/08/2014 16:32	1061	8	14	Thu
1	4	17/08/2014 18:32	17/08/2014 23:59	19659	8	17	Sun
1	4	18/08/2014 00:00	18/08/2014 08:04	29042	8	18	Mon
1	5	18/08/2014 16:18	18/08/2014 16:53	2104	8	18	Mon
1	6	18/08/2014 17:41	18/08/2014 18:28	2819	8	18	Mon
1	7	19/08/2014 16:01	19/08/2014 16:37	2165	8	19	Tue
1	8	19/08/2014 18:09	19/08/2014 18:46	2231	8	19	Tue
1	9	20/08/2014 16:34	20/08/2014 23:59	26728	8	20	Wed
1	9	21/08/2014 00:00	21/08/2014 08:11	29498	8	21	Thu
1	10	21/08/2014 19:21	21/08/2014 23:59	16725	8	21	Thu
1	10	22/08/2014 00:00	22/08/2014 08:21	30063	8	22	Fri
1	11	22/08/2014 13:32	22/08/2014 14:29	3402	8	22	Fri
1	12	25/08/2014 16:39	25/08/2014 17:27	2859	8	25	Mon
1	13	26/08/2014 17:31	26/08/2014 18:41	4190	8	26	Tue
1	14	28/08/2014 17:11	28/08/2014 17:35	1428	8	28	Thu
1	15	28/08/2014 17:52	28/08/2014 23:59	22058	8	28	Thu
1	15	29/08/2014 00:00	29/08/2014 06:33	23606	8	29	Fri
1	16	29/08/2014 11:14	29/08/2014 13:57	9797	8	29	Fri
1	17	02/09/2014 15:30	02/09/2014 23:59	30545	9	2	Tue

This base information is then processed in *SQLite* so as to derive new information relevant for a characterisation of the user's movement behaviour. *SQLite* allows for the automation of the analysis which is very convenient for further processing of results from all the algorithms as well as different combinations of parameter settings for each algorithm.

So as to analyse **the stays extracted** two indicators have been designed to enable a rough evaluation of the **extraction accuracy**. The **detected** stays are compared with the real **tagged** stays reported in GTD⁶:

- **1. Number of stays at each visited place per weekday**
 - 1.1. Number of detected occurrences
 - 1.2. Number of tagged occurrences
 - 1.3. Proportion of tagged stays detected
- **2. Duration of the stays at each visited place per weekday**
 - 2.1. Duration of detected stays
 - 2.2. Duration of tagged stays
 - 2.3. Proportion of tagged stays duration detected

⁶ Detected stays produced by the algorithm which did not match with tagged stays are ignored.

4.2.2. Extraction of transitions

Likewise the dwell time extraction, these times are obtained by difference; in this case the starting time of a stay is subtracted from the ending time of the previous one so as to get the time invested in the lapse between such stays. Additionally, the information regarding the origin and destination of each transition is stored using the coordinates of the corresponding detected places.

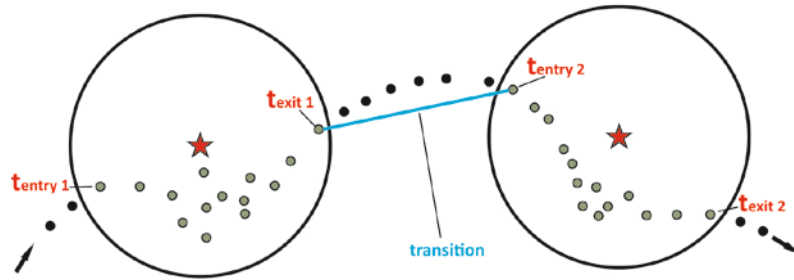


Figure 39. Representation of two visited places and a transition between them

An estimation of the average speed at which the transition was performed is computed and stored as well as the Euclidean distance between origin and destiny. This information does not take into consideration the real trajectories followed by the user and thus the real length of the movements. It is an indicative and further research would be needed, e.g. for determining the means of transport.

A time of transition between two stays does not imply a change of location between two different user's visited places. This could mean the user has not moved to another place considered relevant in his daily life (> 900 seconds visit duration) or the algorithm has not performed well enough on detecting intermediate visits to another place. These transitions will present the same origin and destination.

Table 15. Example of transitions table

TrID	Orig.	Dest.	TimeDeparture	TimeArrival	Distance(m)	Duration(s)	Sp(km/h)	Mon	Day	WD	SameDay
1	1	2	12/08/2014 17:48	12/08/2014 17:55	441.0	405	3.920	8	12	Tue	TRUE
2	2	3	12/08/2014 20:12	13/08/2014 08:17	490.8	43477	0.041	8	12	Tue	FALSE
3	3	1	13/08/2014 16:01	13/08/2014 16:28	320.4	1623	0.711	8	13	Wed	TRUE
4	1	1	13/08/2014 17:10	13/08/2014 17:45	0.0	2131	0.000	8	13	Wed	TRUE
5	1	3	14/08/2014 08:16	14/08/2014 08:23	320.4	423	2.726	8	14	Thu	TRUE
6	3	1	14/08/2014 16:10	14/08/2014 16:14	320.4	264	4.368	8	14	Thu	TRUE
7	1	4	14/08/2014 16:32	14/08/2014 18:47	45377.3	8107	20.150	8	14	Thu	TRUE
8	4	5	14/08/2014 19:28	14/08/2014 19:29	113.6	96	4.260	8	14	Thu	TRUE
9	5	6	15/08/2014 09:59	15/08/2014 10:03	860.0	192	16.125	8	15	Fri	TRUE
10	6	5	15/08/2014 11:59	15/08/2014 12:02	860.0	195	15.877	8	15	Fri	TRUE
11	5	5	15/08/2014 13:28	15/08/2014 14:42	0.0	4429	0.000	8	15	Fri	TRUE
12	5	6	15/08/2014 17:55	15/08/2014 17:59	860.0	198	15.636	8	15	Fri	TRUE
13	6	5	15/08/2014 21:12	15/08/2014 21:15	860.0	195	15.877	8	15	Fri	TRUE
14	5	7	16/08/2014 10:24	16/08/2014 10:44	1905.3	1210	5.669	8	16	Sat	TRUE
15	7	5	16/08/2014 11:02	16/08/2014 14:18	1905.3	11756	0.583	8	16	Sat	TRUE
16	5	5	16/08/2014 18:15	16/08/2014 18:40	0.0	1533	0.000	8	16	Sat	TRUE
17	5	8	17/08/2014 09:51	17/08/2014 10:00	605.3	491	4.438	8	17	Sun	TRUE
18	8	5	17/08/2014 10:48	17/08/2014 12:54	605.3	7614	0.286	8	17	Sun	TRUE
19	5	9	17/08/2014 13:54	17/08/2014 14:42	8827.8	2899	10.962	8	17	Sun	TRUE
20	9	5	17/08/2014 16:18	17/08/2014 16:53	8827.8	2135	14.885	8	17	Sun	TRUE
21	5	1	17/08/2014 17:15	17/08/2014 18:32	45479.6	4640	35.286	8	17	Sun	TRUE
22	1	3	18/08/2014 08:04	18/08/2014 08:13	320.4	555	2.078	8	18	Mon	TRUE
23	3	1	18/08/2014 16:10	18/08/2014 16:18	320.4	447	2.580	8	18	Mon	TRUE
24			8/08/20	6 3	8/08/20	0 0	29 6	0 000	8	8	

In this case a CSV file also tabulates the **transitions** extracted from the GTD. The Table 15 shows a section of this CSV. An ID is assigned to each detected transition and also the IDs of origin and destination are stored. The **departure time** corresponds to the ending time of the previous stay whereas the **arrival time** is equivalent to the starting time of the following stay, at the arrived visited place. The Euclidean distance between departing and arriving place is calculated as well as the duration of the movement in seconds. Then, an estimate of the speed is calculated but only as an indication given that we are not working with real trajectories, thus we ignore the real distance travelled. Number of month, day and weekday is also stored.

The column *SameDay* is useful for storage of Boolean information about the number of days affected in such transition. In some cases, the hypothetical transition starts at the evening of one day and finishes at the morning of the following day, lasting e.g. 12 hours as in $TrID = 2$, in the table. **These cases represent a lack of detection of an intermediate stay. Almost all of these long stays are periods at “sleeping places” and most of them are not detected because the ground truth data itself.** Battery shortage during night and GPS cold start errors at the beginning of the new day mainly could explain multiple gaps in GTD. In these cases, the distance between the last GPS point of one day and the first point of the following day is considerably greater than the value for the distance parameter (d) of the algorithm; hence, preventing the detection of a cluster or visit (cluster re-detection) at such *sleeping places*. Moreover, this explains also part of the time not detected at stays by the algorithm. This time corresponds to *sleeping periods*.

This transition information is also mined in SQLite so as to derive new information relevant for the characterisation of the user’s movement behaviour.

Another two indicators, equivalent to those presented for stays, have been designed for assessment of the **accuracy of the extracted transitions**. The detected transitions are now compared with the tagged transitions between the tagged places:

- **1. Number of transitions between tagged places per weekday**
 - o 1.1. Number of detected occurrences
 - o 1.2. Number of tagged occurrences
 - o 1.3. Proportion of tagged transitions detected
- **2. Duration of the transitions between tagged places per weekday**
 - o 2.1. Duration of detected transitions
 - o 2.2. Duration of tagged transitions
 - o 2.3. Proportion of tagged transitions duration detected

4.2.3. QE of the stays and transitions extraction

The specific quality evaluation of this extraction is fully integrated within the general Java process. The calculated values for the metrics described in 3.3.2 are included in the output CSV log.

4.2.4. Representation of stays and transitions

The free and open source software QGIS is used to generate multiple visualizations with the *Qgis2threejs*⁷ plugin. Different types of visualizations were prepared for comparison between GTD and detected places, stays and transitions.

This contributed generating insight into the GTD allowing us to detect movement behaviour patterns. In this regard, a possible application for the general approach developed in this thesis is presented at the end of this document.

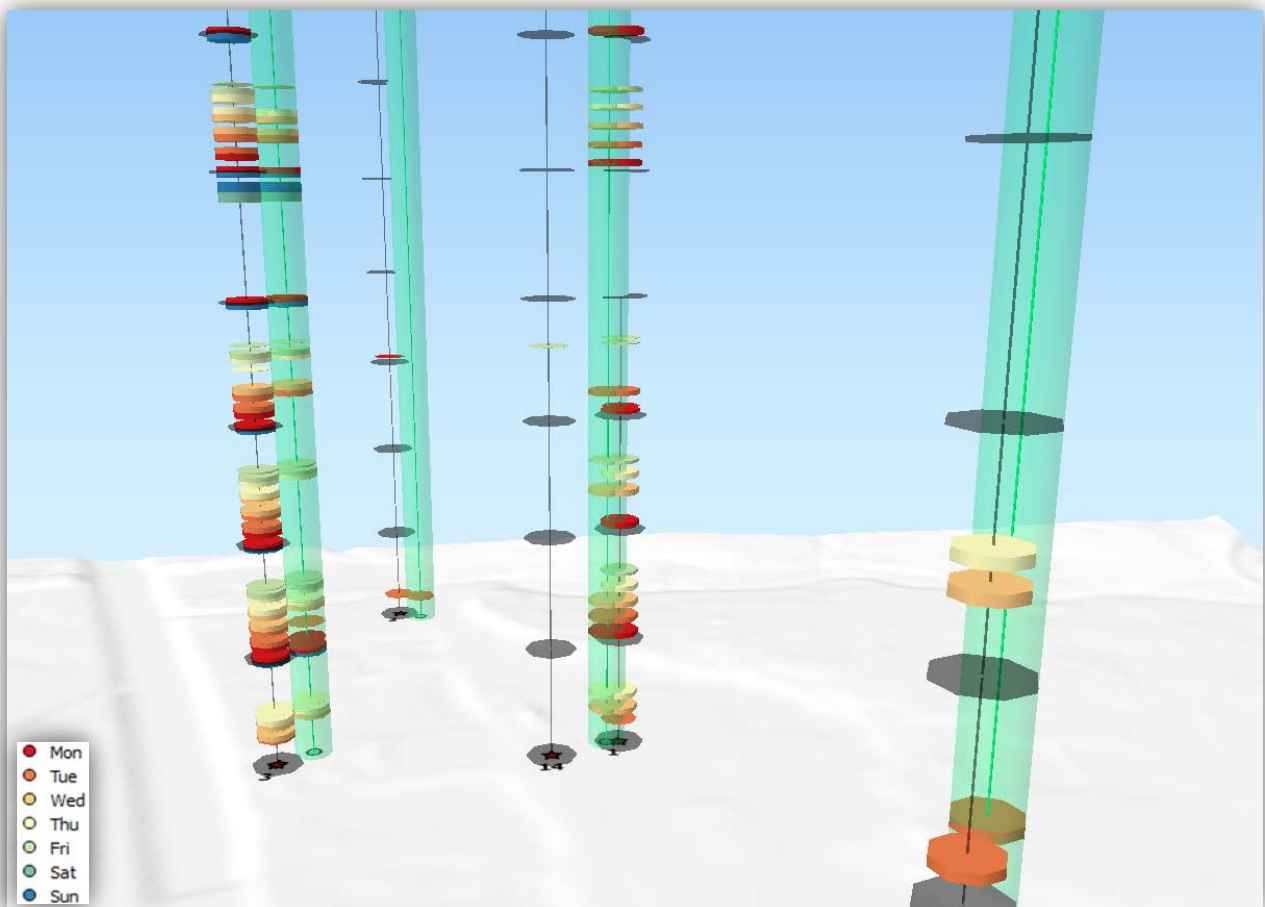


Figure 40. Visualization of detected and tagged stays as stacks of cylinders.

⁷ *Qgis2threejs* has been developed by [Minoru Akagi](#). It allows for 3D visualization powered by *WebGL* and the *three.js* JavaScript library.

5. RESULTS AND DISCUSSION

Different experiments have been conducted so as to evaluate the quality performance of every sub-approach on **determining the user's visited places**, as well as the performance of the algorithms on dealing with four datasets.

Outputs of the algorithms have been mined in order to **characterise the stays and the transitions** of the user between her identified visited places. A quality evaluation of the time extraction performed by the best clustering sub-approach is presented and several indicators have been calculated to assess the stays and transitions extraction accuracy.

5.1. Determination of visited places

Every sub-approach was applied over the whole dataset of each of the four individuals. For testing different parameters combinations batch processing was used. The output was collected in a database. Then, Quality Evaluation (QE) results for the **four persons** were **averaged** and are presented in the following sections.

The values obtained for the four quality measures and the three main measures derived from the confusion matrix are presented in a table. Additionally, the number of detected places produced by the clustering is also presented. A conditional formatting of the data with a colour scale has been applied to each measure matrix in order to facilitate the visualization of the information.

5.1.1. Clustering results

5.1.1.1. Incremental clustering (*Kang*)

Quality results

A graduated colour scale is used in Figure 41 so as to represent the data. Better values are coloured in green whereas worst values in red. Percentile 50 of the values is represented in yellow. Nevertheless, the number of detected places is an exception because a low number of detections have been considered better (green) than a high number of them (red).

First, the four quality measures are considered. The maximum ***spatial accuracy (Qsa)*** is achieved with 300 seconds and 53 meters as input parameters for our implementation of Kang's algorithm. However, the best ***spatial uniqueness (Qsu)*** can be obtained with a wide range of combinations, having generated very high values for all the parameter settings. This means that detected places have been related to more than one tagged place in only very few cases. In other words, only a few clusters have been created in overlapping areas of more than one circular buffer around a tagged place.

Qsa		Distance d (m)								
Time t (s)		20	30	40	53	60	70	80	90	100
300		0.471	0.493	0.506	0.507	0.501	0.490	0.486	0.473	0.461
600		0.411	0.432	0.469	0.475	0.470	0.455	0.450	0.445	0.421
900		0.380	0.396	0.439	0.443	0.444	0.437	0.429	0.432	0.407
1200		0.355	0.360	0.386	0.395	0.399	0.398	0.394	0.397	0.376
1500		0.335	0.338	0.357	0.366	0.372	0.384	0.378	0.379	0.367
1800		0.306	0.336	0.351	0.353	0.362	0.381	0.373	0.369	0.353
2100		0.295	0.317	0.344	0.339	0.358	0.368	0.361	0.367	0.352

Qsu		Distance d (m)								
Time t (s)		20	30	40	53	60	70	80	90	100
300		0.992	0.996	1.000	0.996	1.000	0.996	0.996	0.996	0.998
600		0.987	0.995	1.000	1.000	1.000	0.994	0.994	0.994	0.991
900		1.000	0.993	1.000	1.000	1.000	0.993	0.993	0.993	0.995
1200		1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.994
1500		1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.994
1800		1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.993
2100		1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.993

Qta		Distance d (m)								
Time t (s)		20	30	40	53	60	70	80	90	100
300		0.556	0.522	0.542	0.574	0.590	0.611	0.606	0.577	0.572
600		0.518	0.506	0.579	0.624	0.648	0.652	0.643	0.614	0.615
900		0.531	0.503	0.543	0.570	0.612	0.624	0.620	0.589	0.580
1200		0.529	0.493	0.566	0.593	0.619	0.629	0.621	0.593	0.586
1500		0.521	0.523	0.599	0.583	0.622	0.641	0.616	0.594	0.598
1800		0.525	0.556	0.615	0.570	0.621	0.656	0.633	0.600	0.597
2100		0.526	0.558	0.616	0.582	0.633	0.659	0.640	0.602	0.603

Qti		Distance d (m)								
Time t (s)		20	30	40	53	60	70	80	90	100
300		0.452	0.612	0.664	0.685	0.719	0.788	0.785	0.792	0.817
600		0.492	0.669	0.667	0.728	0.748	0.784	0.789	0.792	0.799
900		0.549	0.649	0.705	0.713	0.751	0.777	0.782	0.775	0.808
1200		0.581	0.651	0.739	0.739	0.770	0.778	0.791	0.791	0.813
1500		0.570	0.645	0.737	0.745	0.763	0.774	0.774	0.784	0.798
1800		0.602	0.651	0.729	0.745	0.759	0.770	0.774	0.803	0.809
2100		0.602	0.671	0.743	0.776	0.772	0.775	0.773	0.783	0.792

Figure 41. Values of quality measures for first sub-approach.

The maximum *amount of temporal incorrectness* (Qti) is reached with the combination of 100 meters for d and 300 seconds for t . Maximum Qti indicates that 81.7 % of the times recorded by the test users in their trip protocols have been identified under the quality conditions defined (page 29). Nevertheless, the worst values for the *temporal accuracy* (Qta) are generated when using distance thresholds of 20 and 30 m. Qta represents how accurate has been the identification of the times detected by the corresponding parameter settings, thus a high Qti does not mean the time detection has been accurate.

The higher *temporal accuracy* (Qta) is produced with threshold values of 70 m and 2100 s. Such common high values for the two temporal measures indicate a relatively high performance of the algorithm on detecting the real time of the visits to the spatially detected tagged places.

Regarding the most relevant measures derived from the **confusion matrix**, the maximum **recall** of 74.5 % is obtained with the combination 300 seconds / 30 meters, whereas the best **precision** of 84.6 % is reached with the parameter values 2100 seconds / 20 meters. Nevertheless, the higher **F measure** (62.9 %) is produced by the combination 900 s / 80 m. The lowest number of **detected places** is produced by the higher tested value for t , 35 minutes.

Taking the F measure as the most adequate indicator of the clustering performance, the best combination (900 s / 80 m) can reach a 59.7 % and 67.1 % for recall and precision respectively.

However, the algorithm reached a maximum F measure of 74.3 % for **User1**, with a recall of 72.2 % and a precision of 76.5 % using a different combination of parameters.

Recall	Distance d (m)									
Time t (s)	20	30	40	53	60	70	80	90	100	
300	0.697	0.745	0.731	0.728	0.728	0.719	0.720	0.698	0.658	
600	0.586	0.629	0.648	0.649	0.648	0.644	0.654	0.645	0.603	
900	0.514	0.548	0.577	0.581	0.581	0.576	0.597	0.589	0.561	
1200	0.471	0.499	0.513	0.516	0.518	0.527	0.541	0.531	0.505	
1500	0.445	0.475	0.480	0.484	0.489	0.498	0.507	0.504	0.500	
1800	0.414	0.461	0.462	0.464	0.485	0.489	0.497	0.484	0.473	
2100	0.399	0.435	0.449	0.437	0.467	0.472	0.484	0.463	0.460	

Precision	Distance d (m)									
Time t (s)	20	30	40	53	60	70	80	90	100	
300	0.441	0.470	0.428	0.416	0.413	0.407	0.392	0.372	0.351	
600	0.624	0.610	0.594	0.579	0.585	0.552	0.559	0.556	0.517	
900	0.728	0.722	0.695	0.673	0.661	0.641	0.671	0.664	0.636	
1200	0.750	0.731	0.736	0.724	0.725	0.698	0.706	0.695	0.673	
1500	0.803	0.783	0.776	0.745	0.762	0.754	0.749	0.736	0.704	
1800	0.828	0.810	0.790	0.775	0.786	0.757	0.777	0.760	0.744	
2100	0.846	0.824	0.802	0.787	0.795	0.764	0.786	0.778	0.757	

Fmeasure	Distance d (m)									
Time t (s)	20	30	40	53	60	70	80	90	100	
300	0.539	0.574	0.538	0.527	0.525	0.518	0.506	0.484	0.457	
600	0.600	0.618	0.619	0.611	0.614	0.594	0.602	0.596	0.554	
900	0.600	0.620	0.628	0.621	0.617	0.604	0.629	0.621	0.593	
1200	0.574	0.588	0.602	0.600	0.601	0.598	0.609	0.598	0.574	
1500	0.568	0.587	0.591	0.583	0.592	0.596	0.601	0.593	0.580	
1800	0.548	0.583	0.581	0.578	0.596	0.590	0.602	0.588	0.574	
2100	0.538	0.564	0.572	0.557	0.584	0.578	0.593	0.574	0.567	

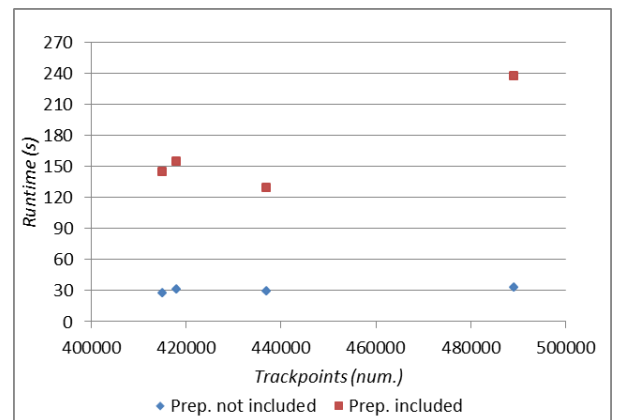
Detections	Distance d (m)									
Time t (s)	20	30	40	53	60	70	80	90	100	
300	115	98	101	101	100	99	99	101	102	
600	67	62	63	62	61	63	63	62	63	
900	47	45	46	47	48	48	47	47	48	
1200	40	40	39	39	39	40	40	41	40	
1500	35	36	34	35	35	35	36	36	38	
1800	31	33	33	32	33	34	34	34	34	
2100	29	31	31	30	32	33	33	32	33	

Figure 42. Detections and values from confusion matrix for first sub-approach.

Runtime

The runtimes of the Java implementation of this sub-approach have been considered. The whole process includes GPS tracks pre-processing, clustering, stays and transitions extraction, and quality evaluation. This time has been related with the total number of GPS points parsed, different for each of the four users who collected our ground truth data. The graphic at the right presents the runtimes both including and excluding the pre-processing of the GPS trackpoints.

Figure 43. Relation runtime - number of points



As observed in the graphic, there is not a clear relation between runtime and number of points parsed. The algorithm's average runtime for the four persons, excluding pre-processing, was **30 seconds**.

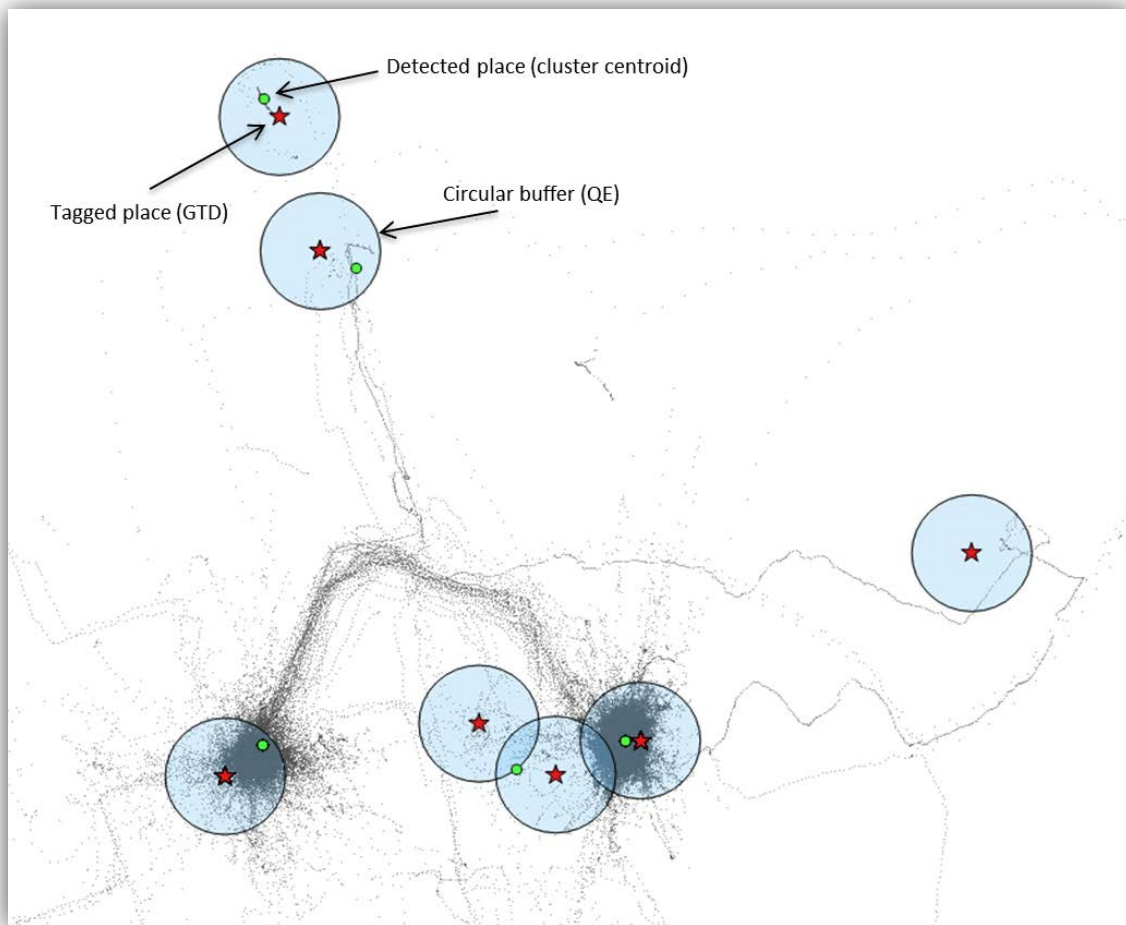


Figure 44. Example of Kang clustering results and relation with GTD

5.1.1.2. Incremental + density-based clustering (*Ye + ConvexHull*)

Quality results

Again, the results obtained from the quality evaluation are presented in the same way as in the previous sub-approach. Same colour coding has been applied; however no direct comparison is possible because parameters values tested are different due to the specificities of each approach which determine a logic range of values to test. The temporal measures reveal a relatively good performance, also on this approach, on extracting the times of the spatially assigned tagged places.

Qsa		Distance (m)								
Time (s)		25	50	100	200	300	400	500	600	700
300		0.482	0.489	0.485	0.430	0.403	0.391	0.346	0.288	0.251
600		0.394	0.453	0.461	0.400	0.357	0.353	0.311	0.276	0.234
900		0.331	0.389	0.420	0.384	0.351	0.343	0.302	0.267	0.228
1200		0.300	0.353	0.391	0.363	0.328	0.306	0.279	0.254	0.220
1800		0.234	0.305	0.343	0.333	0.313	0.301	0.261	0.238	0.222
2400		0.212	0.289	0.309	0.301	0.277	0.267	0.230	0.194	0.203
3000		0.178	0.237	0.247	0.280	0.259	0.247	0.212	0.178	0.184

Qsu		Distance (m)								
Time (s)		25	50	100	200	300	400	500	600	700
300		0.998	0.996	1.000	0.996	0.997	0.998	0.997	0.996	0.998
600		0.989	0.994	1.000	0.992	0.993	0.997	0.997	0.997	0.997
900		0.986	0.992	1.000	1.000	0.989	0.996	0.995	0.996	0.996
1200		0.993	1.000	1.000	1.000	0.986	0.995	0.994	0.996	0.994
1800		0.991	1.000	1.000	1.000	0.984	0.994	0.992	0.995	0.992
2400		1.000	1.000	1.000	1.000	0.989	1.000	0.992	0.992	0.992
3000		1.000	1.000	1.000	1.000	0.988	1.000	0.991	0.990	0.990

Qta		Distance (m)								
Time (s)		25	50	100	200	300	400	500	600	700
300		0.421	0.555	0.538	0.497	0.448	0.445	0.398	0.333	0.398
600		0.566	0.627	0.643	0.556	0.532	0.463	0.414	0.340	0.382
900		0.517	0.551	0.581	0.526	0.516	0.507	0.427	0.352	0.356
1200		0.533	0.552	0.541	0.515	0.502	0.477	0.397	0.334	0.323
1800		0.559	0.567	0.626	0.571	0.511	0.513	0.389	0.364	0.275
2400		0.610	0.586	0.665	0.593	0.513	0.510	0.386	0.358	0.305
3000		0.625	0.562	0.680	0.610	0.531	0.510	0.442	0.379	0.289

Qti		Distance (m)								
Time (s)		25	50	100	200	300	400	500	600	700
300		0.664	0.713	0.681	0.776	0.707	0.708	0.743	0.682	0.743
600		0.582	0.645	0.695	0.790	0.730	0.745	0.758	0.690	0.755
900		0.539	0.649	0.676	0.780	0.744	0.750	0.774	0.706	0.758
1200		0.531	0.670	0.666	0.783	0.744	0.747	0.774	0.703	0.755
1800		0.539	0.669	0.704	0.774	0.751	0.736	0.747	0.698	0.739
2400		0.601	0.713	0.707	0.783	0.781	0.768	0.761	0.777	0.745
3000		0.619	0.731	0.746	0.796	0.762	0.768	0.759	0.794	0.746

Figure 45. Values of quality measures for second sub-approach.

The best **Qsa** is produced with the combination 300 s / 50 m for clustering. The values for **Qsu** are close to 1 for the majority of the tested settings also in this case. On the other hand, $t = 50$ minutes and $d = 200$ meters generate the best **Qti**, but the maximum **Qta** is obtained with 50 minutes / 100 m.

The confusion matrix measures reveal a maximum recall of 72.3 % for the setting 300 s/ 100 m whereas the best precision (79.8 %) is achieved with the pair 3000 s / 50 m. Nevertheless, the higher F measure (57.5 %) required 20 minutes for Time and 100 meters for Distance. In this case, the precision and

recall obtained were respectively a 64.2 % and a 52.5 %. The lower number of *detected places* was produced by the higher *time* tested of 50 minutes.

Recall	Distance (m)									
Time (s)	25	50	100	200	300	400	500	600	700	
300	0.661	0.711	0.723	0.658	0.634	0.619	0.543	0.480	0.414	
600	0.546	0.620	0.645	0.600	0.574	0.550	0.490	0.446	0.394	
900	0.453	0.523	0.563	0.545	0.543	0.499	0.454	0.418	0.370	
1200	0.401	0.462	0.525	0.500	0.492	0.454	0.435	0.388	0.353	
1800	0.330	0.405	0.449	0.450	0.440	0.421	0.404	0.337	0.335	
2400	0.295	0.359	0.403	0.415	0.384	0.374	0.354	0.319	0.306	
3000	0.247	0.307	0.335	0.384	0.365	0.352	0.336	0.414	0.283	

Precision	Distance (m)									
Time (s)	25	50	100	200	300	400	500	600	700	
300	0.376	0.353	0.312	0.249	0.217	0.210	0.199	0.193	0.169	
600	0.504	0.486	0.475	0.387	0.345	0.319	0.279	0.259	0.231	
900	0.575	0.591	0.578	0.509	0.469	0.419	0.361	0.331	0.284	
1200	0.622	0.622	0.642	0.587	0.542	0.481	0.423	0.387	0.335	
1800	0.665	0.691	0.679	0.638	0.600	0.549	0.502	0.482	0.410	
2400	0.694	0.718	0.721	0.690	0.639	0.580	0.541	0.514	0.454	
3000	0.791	0.798	0.767	0.739	0.694	0.631	0.564	0.555	0.484	

Fmeasure	Distance (m)									
Time (s)	25	50	100	200	300	400	500	600	700	
300	0.474	0.467	0.432	0.360	0.319	0.310	0.288	0.277	0.239	
600	0.515	0.540	0.544	0.468	0.428	0.401	0.353	0.333	0.288	
900	0.501	0.548	0.565	0.521	0.499	0.453	0.400	0.378	0.318	
1200	0.483	0.526	0.575	0.538	0.512	0.465	0.427	0.400	0.341	
1800	0.438	0.507	0.537	0.524	0.505	0.475	0.446	0.428	0.368	
2400	0.410	0.474	0.513	0.515	0.478	0.453	0.423	0.404	0.364	
3000	0.375	0.441	0.465	0.504	0.477	0.451	0.418	0.403	0.356	

Detections	Distance (m)									
Time (s)	25	50	100	200	300	400	500	600	700	
300	99	115	132	145	160	165	150	143	132	
600	61	70	77	86	92	95	96	100	93	
900	44	48	54	59	66	67	69	73	70	
1200	36	40	45	47	52	54	57	58	57	
1800	27	31	36	39	41	43	44	44	44	
2400	23	26	30	33	33	35	35	36	36	
3000	17	20	24	28	28	31	31	31	31	

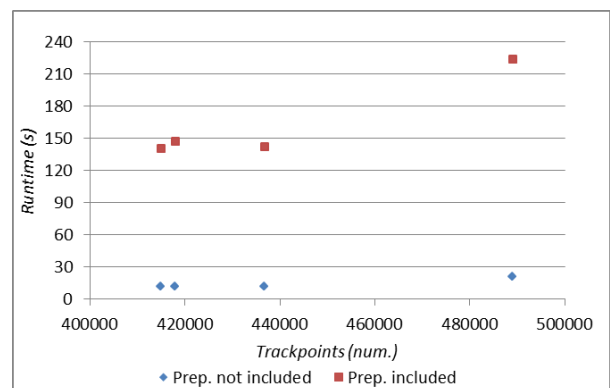
Figure 46. Detections and values from confusion matrix for second sub-approach.

Runtime

Likewise the previous approach, the runtimes of this solution have been considered.

The average runtime for processing the datasets of the four users, excluding pre-processing and including our *ConvexHull* solution, was **14 seconds**.

Figure 47. Relation runtime - number of points (Ye)



5.1.1.3. Density-based clustering (DBSCAN)

Quality results

The last quality evaluation in this case shows worst results for the temporal quality measures in comparison with other two algorithms. The setting with 9 meters as search radius (Eps) and 20 as number of points within neighbourhood ($MinPts$) achieves the best Q_{sa} . Meanwhile, Q_{su} reach high values as with the other approaches. On the side of the temporal measures, the best Q_{ta} of 0.852 is produced with a combination 2 Eps / 80 $MinPts$ whereas the maximum Q_{ti} is generated by the a search radius Eps of 15 meters and a neighbourhood of 120 points for the clustering, then a 58.9 % of the times provided in the identified tagged places were correctly detected by the sub-approach.

Qsa	Eps (m)						
MinPts	2	3	6	9	12	15	18
20	0.434	0.435	0.486	0.493	0.479	0.431	0.408
30	0.421	0.429	0.481	0.486	0.492	0.469	0.438
40	0.389	0.419	0.457	0.485	0.492	0.468	0.446
50	0.349	0.383	0.437	0.468	0.487	0.467	0.452
60	0.313	0.368	0.407	0.448	0.478	0.480	0.469
70	0.310	0.365	0.404	0.436	0.459	0.475	0.467
80	0.298	0.352	0.394	0.434	0.456	0.461	0.461
90	0.295	0.329	0.401	0.439	0.447	0.467	0.475
100	0.273	0.308	0.376	0.420	0.443	0.443	0.463
110	0.255	0.283	0.368	0.395	0.418	0.436	0.459
120	0.257	0.274	0.367	0.389	0.415	0.431	0.439

Qsu	Eps (m)						
MinPts	2	3	6	9	12	15	18
20	0.993	0.996	0.997	0.999	0.998	0.999	0.999
30	0.990	0.995	0.994	0.998	0.998	0.998	0.999
40	0.994	0.990	0.997	0.999	0.997	0.997	0.999
50	0.983	0.987	0.998	0.998	0.998	0.997	0.998
60	0.984	0.993	0.998	0.996	0.998	0.992	0.998
70	0.991	0.991	0.997	0.998	0.995	0.993	0.998
80	0.983	0.996	0.995	1.000	1.000	1.000	0.997
90	0.990	0.991	0.995	1.000	1.000	1.000	0.997
100	0.996	0.984	0.994	1.000	0.996	1.000	0.992
110	0.992	0.986	0.994	1.000	1.000	1.000	0.994
120	0.991	0.985	0.993	1.000	1.000	1.000	0.994

Qta	Eps (m)						
MinPts	2	3	6	9	12	15	18
20	0.810	0.831	0.796	0.722	0.750	0.769	0.754
30	0.717	0.799	0.763	0.777	0.661	0.734	0.780
40	0.753	0.609	0.717	0.782	0.698	0.654	0.753
50	0.780	0.611	0.725	0.672	0.711	0.644	0.714
60	0.791	0.703	0.628	0.666	0.636	0.620	0.583
70	0.825	0.678	0.604	0.616	0.677	0.614	0.612
80	0.852	0.695	0.595	0.632	0.602	0.619	0.630
90	0.709	0.724	0.676	0.676	0.580	0.620	0.635
100	0.648	0.574	0.632	0.648	0.609	0.657	0.628
110	0.719	0.579	0.566	0.582	0.513	0.638	0.621
120	0.587	0.632	0.528	0.601	0.543	0.598	0.617

Qti	Eps (m)						
MinPts	2	3	6	9	12	15	18
20	0.060	0.082	0.152	0.225	0.237	0.319	0.369
30	0.091	0.132	0.191	0.291	0.351	0.352	0.389
40	0.123	0.172	0.264	0.331	0.377	0.362	0.405
50	0.155	0.230	0.321	0.365	0.398	0.417	0.415
60	0.182	0.295	0.311	0.430	0.459	0.458	0.451
70	0.220	0.304	0.385	0.408	0.466	0.486	0.515
80	0.177	0.342	0.415	0.412	0.470	0.502	0.514
90	0.221	0.332	0.464	0.458	0.498	0.488	0.539
100	0.207	0.327	0.504	0.497	0.541	0.536	0.513
110	0.250	0.391	0.472	0.534	0.535	0.552	0.515
120	0.294	0.396	0.521	0.569	0.547	0.589	0.558

Figure 48. Values of quality measures for third sub-approach.

Analysing the results for the confusion matrix, a good maximum recall of 77 % is obtained with the Eps 6 and $MinPts$ 20. Nevertheless, the best precision (66.3 %) required 120 points within the neighbourhood and 2 meters of Epsilon. Then, **the best combination of precision and recall determined by the higher F measure of 56.1 % was achieved with 18 m Eps and 110 points of neighbourhood.** The lowest amounts of detected places are produced by the greatest values of $MinPts$.

Recall		Eps (m)						
MinPts		2	3	6	9	12	15	18
20		0.651	0.670	0.770	0.753	0.727	0.667	0.615
30		0.605	0.628	0.698	0.727	0.706	0.698	0.649
40		0.553	0.581	0.654	0.706	0.697	0.679	0.664
50		0.508	0.557	0.622	0.660	0.682	0.669	0.648
60		0.458	0.530	0.591	0.624	0.656	0.671	0.662
70		0.449	0.505	0.567	0.613	0.644	0.663	0.652
80		0.428	0.481	0.548	0.612	0.616	0.646	0.649
90		0.415	0.469	0.543	0.594	0.608	0.649	0.657
100		0.382	0.429	0.508	0.568	0.597	0.629	0.639
110		0.360	0.391	0.491	0.536	0.575	0.599	0.630
120		0.360	0.379	0.486	0.520	0.550	0.579	0.602

Precision		Eps (m)						
MinPts		2	3	6	9	12	15	18
20		0.200	0.179	0.160	0.151	0.144	0.130	0.125
30		0.299	0.270	0.240	0.225	0.204	0.200	0.180
40		0.343	0.357	0.303	0.294	0.270	0.248	0.236
50		0.408	0.448	0.368	0.350	0.331	0.298	0.277
60		0.430	0.489	0.436	0.393	0.388	0.363	0.338
70		0.475	0.521	0.465	0.420	0.427	0.400	0.382
80		0.521	0.533	0.500	0.480	0.458	0.447	0.418
90		0.571	0.556	0.515	0.501	0.477	0.477	0.456
100		0.587	0.544	0.543	0.516	0.500	0.497	0.479
110		0.621	0.577	0.588	0.548	0.515	0.520	0.508
120		0.663	0.618	0.621	0.562	0.521	0.523	0.515

Fmeasure		Eps (m)						
MinPts		2	3	6	9	12	15	18
20		0.306	0.281	0.265	0.252	0.240	0.217	0.208
30		0.400	0.376	0.355	0.342	0.315	0.310	0.281
40		0.423	0.441	0.412	0.411	0.387	0.361	0.348
50		0.452	0.494	0.460	0.456	0.442	0.411	0.387
60		0.442	0.507	0.500	0.481	0.484	0.468	0.446
70		0.461	0.512	0.509	0.496	0.510	0.497	0.479
80		0.469	0.504	0.521	0.536	0.523	0.526	0.505
90		0.480	0.508	0.527	0.541	0.533	0.547	0.536
100		0.462	0.478	0.524	0.538	0.542	0.551	0.546
110		0.454	0.461	0.535	0.540	0.540	0.554	0.561
120		0.464	0.463	0.545	0.537	0.533	0.548	0.553

Detections		Eps (m)						
MinPts		2	3	6	9	12	15	18
20		299	303	316	294	286	287	272
30		188	185	193	202	201	196	200
40		146	129	145	153	153	158	157
50		117	100	113	118	125	130	133
60		96	83	93	99	103	108	111
70		84	73	81	91	93	97	98
80		76	67	69	80	81	84	90
90		70	64	64	72	74	80	83
100		62	59	57	64	69	75	76
110		56	49	53	57	64	66	72
120		52	44	49	53	60	62	67

Figure 49. Detections and values from confusion matrix for third sub-

Runtime

For this density-based sub-approach, it has been considered the performance of ELKI when clustering with DBSCAN and making use of a kd-tree as indexing structure. The Figure 50 shows a plot of the runtimes required for the processing of *User1* dataset (418.017 trackpoints) depending on the search radius selected *Eps*. Only the times for two different neighbourhoods (*MinPts* 30 and 60) have been represented because all the tested *MinPts* present a very similar behaviour.

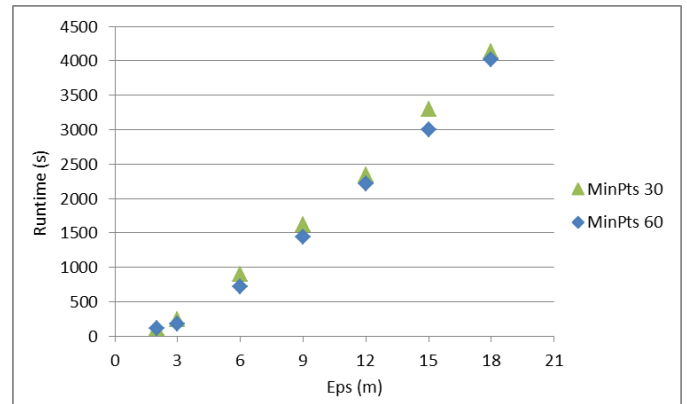


Figure 50. ELKI DBSCAN clustering runtimes

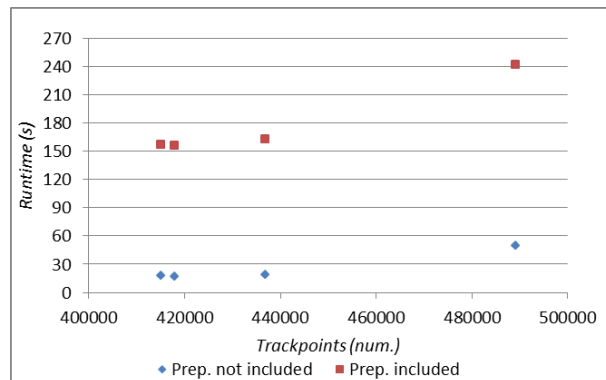


Figure 51. Relation runtime - number of points (DBSCAN)

Then, the runtime of our Java processing has been considered. In this case, the process starts with the feed of the ELKI clusters and includes GPS tracks pre-processing, dwell times extraction, transitions extraction and quality evaluation. The plot again compares the whole runtime needed with the time required when pre-processing is excluded. The average runtime for our Java class, excluding pre-processing, was **26 seconds**.

5.1.2. Algorithms assessment

The quality evaluation measures capture the temporal and spatial accuracy of the detection of tagged places. Nevertheless, the general performance of the algorithms has been compared with a confusion matrix.

As we have presented and analysed in the previous sections, different parameters settings for the algorithms generate variable results for each of the quality measures and confusion matrices. Maximum values for each one of the indices would indicate the best performance in a specific aspect of evaluation. However, the assessment of the overall performance of each spatio-temporal approach depends on all the aspects.

Table 16 presents the higher values reached for each of the indices on consideration and the parameters values which generated such results. This is a summarisation of the results already commented.

Table 16. Best values for each index and generating parameter settings

Best values for each measure and generating parameters						
Algorithm	KANG		YE		DBSCAN	
	Parameters	Value	Parameters	Value	Parameters	Value
Recall	30 / 300	0.745	100 / 300	0.718	6 / 20	0.770
Precision	20 / 2100	0.846	26.5 / 3000	0.781	2 / 120	0.663
Fmeasure	80 / 900	0.629	100 / 1200	0.575	18 / 110	0.561
Qsa	53 / 300	0.507	50 / 300	0.489	9 / 20	0.493
Qsu	Multi	1.000	Multi	1.000	Multi	1.000
Qta	70 / 2100	0.659	100 / 3000	0.680	2 / 80	0.852
Qti	200 / 900	0.838	200 / 3000	0.796	15 / 120	0.589
Avg. Runtime	80 / 900	30	100 / 1200	14	18 / 110	26

➤ Temporal performance

The maximum *temporal accuracy* Q_{ta} can be reached by our implementation of **DBSCAN** (0.852). However, the most correct time detection is carried out by our implementation of **Kang** as we can conclude from the *amount of temporal incorrectness* Q_{ti} . It indicates that an 83.8 % of the times detected by Kang are correct, while **DBSCAN** detects a greater proportion of incorrect times.

➤ Spatial performance

The higher *spatial accuracy* Q_{sa} can be reached by **Kang** (0.507) whereas all the algorithms are able to generate a 100 % of clusters uniquely related to one possible tagged place, instead of located between two or more of them. This is represented by the value (1.000) of the *spatial uniqueness* Q_{su} .

The confusion matrix completes the evaluation of the spatial performance and at the end it is the key element to select the best clustering approach. The temporal performance fully depends on the spatial performance because, within our framework, there is no possible time detection for clusters not spatially related to any tagged place. An algorithm or parameters configuration can detect a high proportion of tagged times (high Q_{ti}) but it considers only detected places previously spatially assigned to tagged places, so if another algorithm or parameters configuration is able to detect more tagged places, it could extract more correct times despite reaching a lower Q_{ti} than the first algorithm.

➤ Overall performance

The confusion matrix values offer the best estimation on the performance of an algorithm. In (Page 33) the relevance of the F measure is explained; it describes *the ratio of precision and recall and determines the effectiveness of retrieval*. Hence, it will be used to select the best approach for detecting visited places in a user's daily life.

Table 17 shows the best **F measure** achieved by each algorithm, the corresponding quality results and the parameters settings generating such optimal values.

Table 17. Best F measures reached by the algorithms

Best F measure and associated measures			
Algorithm	KANG	YE	DBSCAN
Parameters (d/t)	80 / 900	100 / 1200	18 / 110
	Value	Value	Value
Recall	0.597	0.525	0.630
Precision	0.671	0.642	0.508
Fmeasure	0.629	0.575	0.561
Detections	47	45	72
Qsa	0.432	0.391	0.459
Qsu	0.993	1.000	0.994
Qta	0.589	0.537	0.621
Qti	0.775	0.671	0.515

Kang reached the higher average F measure for the **four users (62.9 %)**. **Ye** would be the second option and **DBSCAN** the last one. Nevertheless, **DBSCAN** reached the best recall (63 %) but the lowest precision (50.8 %). This algorithm generates the highest number of detected places and its precision also indicates the large proportion of false positives it produces. Nevertheless it offers the best *spatial* and *temporal accuracies* (Q_{sa} and Q_{ta}), although we have already explained how it is possible.

Therefore, the **Incremental** approach based on **Kang's** algorithm constitutes the best solution to achieve the first aim of this thesis. Moreover, it is the best spatio-temporal clustering sub-approach this thesis can contribute for the analysis of a user's movement behaviour.

The ground truth data collected by each user involves different sampling periods, spatial extensions, means of transport, movement behaviours and even fieldwork incidences. Hence, there is a variation on the performance results of the algorithm for every user dataset. When testing our implementation of Kang the QE results for **User1** were the best. This occurs also when clustering with the other algorithms.

Table 18. Best F measures clustering User1 GTD

Best F measure and associated measures			
Algorithm	KANG	YE	DBSCAN
Parameters	53 / 900	100 / 1200	6 / 120
	Value	Value	Value
Recall	0.722	0.639	0.667
Precision	0.765	0.719	0.800
Fmeasure	0.743	0.676	0.727
Detections	36	32	38
Qsa	0.554	0.505	0.530
Qsu	1.000	1.000	0.974
Qta	0.483	0.393	0.486
Qti	0.815	0.913	0.567

The optimum parameters for Kang when considering **User1** are a **d** of **53 meters** and a **t** of **900 seconds** which generates the best clustering in terms of **F measure (74.3 %)** corresponding to a recall of 72.2 % and a precision of 76.5 %. User1 dataset also improves the results for **Ye** and **DBSCAN** clustering.

The SQL language has been used for mining the output generated by the clustering QE and the subsequent extraction of stays and transitions which allows us to obtain a movement behaviour profile of the user.

5.2. Characterisation of stays and transitions

As pointed out, according to the general QE results the best clustering sub-approach is the **Incremental** which is built upon Kang's algorithm. When clustering the individual GTD datasets the best marks are achieved with *User1* data, probably due to incidences on data collection non solvable by the work presented here. Hence, the rest of the thesis focuses on **User1 ground truth data**.

Firstly, we perform an assessment of the **three algorithms** comparing the quality of the extractions performed and analysing the accuracy of the stays detection carried out. Secondly, we present the quality evaluation of the extraction performed by the **best algorithm** as well as an analysis of the accuracy of the stays and transitions detected by this sub-approach.

5.2.1. Algorithms assessment

QUALITY OF THE EXTRACTION

The quality evaluation presented in 3.3.2 is applied to the three algorithms.

➤ Proportion of time extracted

Total time detected by the algorithm divided by the total time tagged in the ground truth data, for both the stays and the transitions:

$$St_{ext} := \frac{St_d}{St_t} \qquad Tt_{ext} := \frac{Tt_d}{Tt_t}$$

➤ Confusion matrix

Two confusion matrices are generated: the first one to analyse the tagged and detected stays extracted by the algorithm, and the second one does the same for transitions. We obtain recall, precision and **F measure** of both tasks.

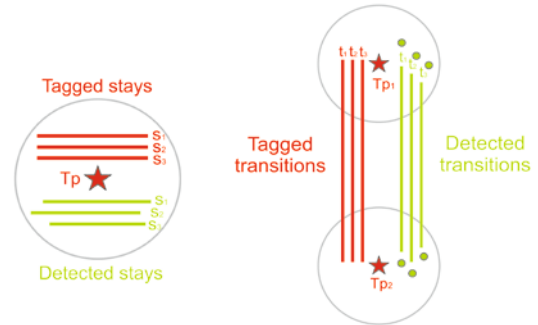


Table 19. Quality of the extraction performed by the 3 algorithms using the best clustering parameters

Quality of the extraction with best clust. parameters			
Algorithm	KANG	YE	DBSCAN
Parameters	53 / 900	100 / 1200	6 / 120
	Value	Value	Value
StayT_ext	0.619	0.175	0.468
TranT_ext	0.493	0.295	0.373
StRecall	0.600	0.317	0.450
StPrecision	0.684	0.467	0.600
StFmeas	0.639	0.377	0.514
TrRecall	0.568	0.331	0.374
TrPrecision	0.594	0.383	0.452
TrFmeas	0.581	0.355	0.409

As presented in the last section, QE of the clustering determined the best parameter values for each algorithm. Using this settings for clustering, the quality evaluation of the stays and transitions extraction generated the values presented in Table 19.

Also for the extraction, the *Incremental* approach is the best option. **Kang** is able to extract a 61.9 % of the real time spent in stays and 49.3 % of time spent in transitions. Despite **DBSCAN** reached good values for clustering, the F measure of the stays and transition extraction only achieves 51.4 % and 40.9 % respectively.

Now, we analyse the extraction performed by the three approaches in order to assess the accuracy of the stays detection. We **only** compare the **stays** extraction because it is more relevant than the **transitions** extraction due to the influence of the mentioned pitfall of the transitions between “sleeping places” (see page 60). Moreover, Kang was only able to detect less than a 50 % of the time invested in transitions, whereas the other algorithms performed much worst.

ANALYSIS OF THE EXTRACTION

For a simplified comparison, only the values for the 3 most visited places of *User1* will be considered. After interviewing *User1* and asking him about the semantic meaning of the places, we know these are the **main home** (*Home1*), the **secondary home** (*Home2*) and the **work place** (*Work*) of *User1*. They represent around 83 % of the total time reported on GTD, therefore these could be considered the most representative locations for the user and an adequate election for comparing the algorithms.

➤ Number of stays at tagged places

Table 20. Stays extraction performed by algorithms: Number of stays at 3 most visited places

3 most stayed places								
Number of stays at each tagged place for each weekday								
Detected Occurrences								
Place	Mon	Tue	Wed	Thu	Fri	Sat	Sun	SUM
KANG								
Home1	4	3	6	8	7	1	4	33
Home2	0	0	0	1	10	14	12	37
Work	3	4	4	2	2	0	0	15
YE								
Home1	1	1	1	0	4	0	0	7
Home2	0	0	0	1	3	6	7	17
Work	2	2	3	3	2	0	0	12
DBSCAN								
Home1	3	3	4	3	3	1	3	20
Home2	0	0	0	1	10	13	10	34
Work	2	0	3	1	4	0	0	10
GTD								
Home1	11	10	11	12	8	2	7	61
Home2	0	0	0	3	11	16	13	43
Work	8	5	4	6	6	0	0	29

Proportion of tagged occurrences detected								
Place	Mon	Tue	Wed	Thu	Fri	Sat	Sun	PROP
KANG								
Home1	0.36	0.3	0.55	0.67	0.88	0.5	0.57	0.54
Home2				0.33	0.91	0.88	0.92	0.86
Work	0.38	0.8	1	0.33	0.33			0.52
YE								
Home1	0.09	0.1	0.09	0	0.5	0	0	0.11
Home2				0.33	0.27	0.38	0.54	0.4
Work	0.25	0.4	0.75	0.5	0.33			0.41
DBSCAN								
Home1	0.27	0.3	0.36	0.25	0.38	0.5	0.43	0.33
Home2				0.33	0.91	0.81	0.77	0.79
Work	0.25	0	0.75	0.17	0.67			0.34
GTD								
Home1	1	1	1	1	1	1	1	1
Home2				1	1	1	1	1
Work	1	1	1	1	1			1

Considering the real time tagged as **GTD**, the detected place semantically tagged as *Home1* presents a higher number of visits during working days whereas the location tagged as *Home2* concentrates its visits during weekends. Obviously, there are no visits to *Work* during weekends. *User1* explained that most of the weekends he is moving to his second residence (*Home2*) in a smaller settlement and during this period, he enjoys plenty of different outdoor activities.

Analysing the distribution of stays at the 3 most visited places during the week; we can observe that Kang outperforms the other algorithms. *Home2* is the best identified place with a similar performance of both Kang and DBSCAN. Nevertheless, Kang extracted an 18 % more stays at *Work* and a 21 % more stays at *Home1* than our implementation of DBSCAN.

Table 21. Stays extraction performed by algorithms: TOTAL of stays at visited places

TOTALS									
Number of stays at each tagged place for each weekday									
Detected stays									
Alg.	Mon	Tue	Wed	Thu	Fri	Sat	Sun	SUM	
KANG	8	9	11	13	23	20	24	108	
YE	4	4	5	6	12	11	15	57	
DBSCAN	6	4	8	6	20	19	18	81	
GTD	24	22	21	28	30	26	29	180	

Proportion of tagged stays detected									
Alg.	Mon	Tue	Wed	Thu	Fri	Sat	Sun	PROP	
KANG	0.33	0.41	0.52	0.46	0.77	0.77	0.83	0.6	
YE	0.17	0.18	0.24	0.21	0.4	0.42	0.52	0.32	
DBSCAN	0.25	0.18	0.38	0.21	0.67	0.73	0.62	0.45	
GTD	1	1	1	1	1	1	1	1	

Now, if we consider the *TOTALS* we can compare the total number of stays detected. 180 stays were reported in GTD and Kang has been able to detect 108 (60 %) of them, whereas DBSCAN has only reached a 45 %. Regarding the distribution of such visits during the week, Kang shows also the highest detection rates. GTD shows a higher number of visits after Thursdays explained by the variety of outdoor activities around the second home reported by the researcher. These imply multiple stays long enough to be extracted and Kang performed quite well on detecting them.

➤ Duration of stays at tagged places

Table 22. Stays extraction performed by algorithms: Stays duration at 3 most visited places

3 most stayed places									
Stays duration at each tagged place for each weekday									
Detected stays duration (h)									
Place	Mon	Tue	Wed	Thu	Fri	Sat	Sun	SUM	
KANG									
Home1	23.48	17.97	40.55	42.20	28.68	11.78	30.35	195	
Home2	0.00	0.00	0.00	4.52	37.43	86.80	61.32	190	
Work	23.48	22.02	31.85	13.48	9.10	0.00	0.00	99.93	
YE									
Home1	0.60	0.62	1.05	0.00	5.95	0.00	0.00	8.22	
Home2	0.00	0.00	0.00	0.65	5.05	16.82	7.37	29.9	
Work	15.63	6.92	24.10	16.33	9.10	0.00	0.00	72.08	
DBSCAN									
Home1	16.65	17.97	31.00	22.53	10.55	11.78	23.52	134	
Home2	0.00	0.00	0.00	4.52	37.43	81.47	50.42	174	
Work	10.68	0.00	25.33	7.78	11.07	0.00	0.00	54.87	
GTD									
Home1	76.23	65.90	67.03	60.38	36.58	12.28	47.70	366	
Home2	0.00	0.00	0.00	5.22	42.98	96.15	61.45	206	
Work	36.73	22.52	31.85	29.77	15.47	0.00	0.00	136	

Proportion of tagged stays duration detected									
Place	Mon	Tue	Wed	Thu	Fri	Sat	Sun	PROP	
KANG									
Home1	0.31	0.27	0.6	0.7	0.78	0.96	0.64	0.53	
Home2				0.87	0.87	0.9	1	0.92	
Work	0.64	0.98	1	0.45	0.59			0.73	
YE									
Home1	0.01	0.01	0.02	0	0.16	0	0	0.02	
Home2				0.12	0.12	0.17	0.12	0.15	
Work	0.43	0.31	0.76	0.55	0.59			0.53	
DBSCAN									
Home1	0.22	0.27	0.46	0.37	0.29	0.96	0.49	0.37	
Home2				0.87	0.87	0.85	0.82	0.84	
Work	0.29	0	0.8	0.26	0.72			0.4	
GTD									
Home1	1	1	1	1	1	1	1	1	
Home2				1	1	1	1	1	
Work	1	1	1	1	1	1		1	

When considering the extraction of the time invested at the 3 most stayed places *Kang* also offered the best performance. *Home1* is the worst detected in terms of total time at stays (53 %). *Home2* is also very well detected with a 92 % of the real stayed time extracted.

In this case, *Ye* is better than *DBSCAN* only on detecting the total time invested at *Work* (13 % more time extracted). This algorithm had a really poor performance for *Home1* (2 %).

Table 23. Stays extraction performed by the algorithms: TOTAL stay durations at places

TOTALS									
Stays duration at each tagged place for each weekday									
Detected stays duration (h)									
Alg.	Mon	Tue	Wed	Thu	Fri	Sat	Sun	SUM	
KANG	48.2	50.9	75.8	62.8	82.6	107	103	530	
YE	17.5	16.2	28.6	19.6	24.3	24.9	19.1	150	
DBSCAN	28.6	26.7	59.7	36.8	63.2	101	84.3	401	
GTD	126	123	134	126	103	123	122	856	

Proportion of tagged stays duration detected									
Alg.	Mon	Tue	Wed	Thu	Fri	Sat	Sun	PROP	
KANG	0.38	0.41	0.57	0.5	0.8	0.87	0.84	0.62	
YE	0.14	0.13	0.21	0.16	0.24	0.2	0.16	0.18	
DBSCAN	0.23	0.22	0.45	0.29	0.61	0.82	0.69	0.47	
GTD	1	1	1	1	1	1	1	1	

Regarding the total time in stays extracted, **Kang** is again the best option despite a lower performance on Mondays and Tuesdays. **Ye** is by far the worst option for stays extraction, with only 150 hours correctly detected (18 %).

Other parameter settings produce slightly higher time extraction for the other algorithms, although with the handicap of a lower F measure from their global quality evaluation.

In order to characterise the stays and transitions between visited places specific data extraction procedures have been implemented within each sub-approach Java process. According to the results of the **global** quality evaluation performed on our improved implementation of the algorithms, the best one (Kang et al. 2005) has been chosen for the **clustering** of the dataset and the subsequent **extraction** of the stays and the transitions between the detected places.

5.2.2. Quality of the extraction with the *Incremental* approach

The structure of the previous section is emulated for both stays and transitions: first, it is evaluated the **quality of the extraction** with the proportion of time extracted and the **confusion matrix** and then, the accuracy of the extracted stays or transitions is **analysed**.

5.2.2.1. Extraction of stays

QUALITY OF THE EXTRACTION

➤ Proportion of stay time extracted

Different parameter settings are compared for the extraction of time in stays but the result corresponding to the best clustering parameters is highlighted in red.

StayT_ext	Distance d (m)								
Time t (s)	20	30	40	53	60	70	80	90	100
300	0.288	0.483	0.560	0.627	0.615	0.600	0.616	0.655	0.625
600	0.257	0.488	0.545	0.612	0.614	0.599	0.614	0.654	0.530
900	0.284	0.462	0.551	0.619	0.613	0.598	0.613	0.653	0.638
1200	0.262	0.447	0.541	0.599	0.613	0.598	0.613	0.653	0.623
1500	0.232	0.449	0.530	0.598	0.611	0.611	0.612	0.652	0.622
1800	0.220	0.426	0.534	0.601	0.611	0.611	0.611	0.651	0.622
2100	0.211	0.403	0.533	0.601	0.611	0.611	0.610	0.650	0.619

Figure 52. Proportion of tagged stay time detected for User1

This table for *User1* shows the proportion between total time detected at stays by the algorithm and the total time tagged in tagged places from GTD. As we can observe, the maximum value of a 65.5 % of total tagged time detected is obtained with a parameter setting of 300sec/90m. Nevertheless, according to the clustering QE, the **F measure** corresponding to this setting is as low as a 48.4 % because a very low **precision** of the clustering (35.1 %) as we can see in Figure 53.

Recall	Distance d (m)									
Time t (s)	20	30	40	53	60	70	80	90	100	
300	0.750	0.806	0.806	0.806	0.778	0.778	0.778	0.722	0.667	
600	0.694	0.722	0.750	0.750	0.750	0.750	0.750	0.694	0.611	
900	0.611	0.667	0.694	0.722	0.722	0.722	0.722	0.667	0.583	
1200	0.611	0.639	0.611	0.639	0.639	0.667	0.667	0.611	0.556	
1500	0.556	0.611	0.583	0.611	0.611	0.611	0.611	0.583	0.556	
1800	0.500	0.611	0.583	0.583	0.611	0.611	0.611	0.556	0.528	
2100	0.472	0.556	0.583	0.583	0.611	0.611	0.611	0.556	0.528	

Precision	Distance d (m)									
Time t (s)	20	30	40	53	60	70	80	90	100	
300	0.466	0.558	0.547	0.518	0.483	0.491	0.418	0.366	0.333	
600	0.694	0.703	0.692	0.675	0.692	0.628	0.628	0.595	0.524	
900	0.759	0.750	0.781	0.765	0.743	0.743	0.743	0.706	0.677	
1200	0.759	0.767	0.815	0.793	0.793	0.774	0.774	0.733	0.714	
1500	0.800	0.815	0.840	0.815	0.846	0.880	0.846	0.778	0.714	
1800	0.818	0.846	0.875	0.840	0.880	0.880	0.846	0.800	0.760	
2100	0.810	0.833	0.875	0.840	0.880	0.880	0.846	0.800	0.760	

Fmeasure	Distance d (m)									
Time t (s)	20	30	40	53	60	70	80	90	100	
300	0.574	0.659	0.652	0.630	0.596	0.602	0.544	0.486	0.444	
600	0.694	0.712	0.720	0.711	0.720	0.684	0.684	0.641	0.564	
900	0.677	0.706	0.735	0.743	0.732	0.732	0.732	0.686	0.627	
1200	0.677	0.697	0.698	0.708	0.708	0.716	0.716	0.667	0.625	
1500	0.656	0.698	0.689	0.698	0.710	0.721	0.710	0.667	0.625	
1800	0.621	0.710	0.700	0.689	0.721	0.721	0.710	0.656	0.623	
2100	0.596	0.667	0.700	0.689	0.721	0.721	0.710	0.656	0.623	

Figure 53. Values from confusion matrix for User1 Kang clustering.

This means this parameter settings for Kang generates plenty of false detected places (false positives) whereas a high number (68.9 %) of the tagged places are also detected (true positives). The total amount of tagged time at these 68.9 % of tagged places detected is maximal for the algorithm.

Nevertheless, we have chosen the combination of parameters that performs the best possible clustering in terms of spatial recall (72.2 %) and precision (76.5 %), through the **higher F measure value** (74.3 %). This guarantees that despite the proportion of total tagged time extracted in this case (61.9 %) is slightly lower than the possible maximum, the clusters generated are more significant as being closer to the real visited places of *User1*. Hence, the characterisation of the stays and transitions in between will reflect more closely the real movement behaviour of the user, as detecting less false stays and transitions or ignoring less of the real ones.

➤ Confusion matrix

Now, the values obtained for the confusion matrix generated from the stays extraction will be analysed. Precision, recall and F measure of the *stays extraction* performed by Kang for *User1* show a different distribution compared to the confusion matrix values generated from the clustering QE of the algorithm.

The maximum stays extraction F measure (65.7 %) is reached with a parameter setting 1200s/90m, but again it corresponds to a lower clustering performance compared to the selected parameters for the analysis.

TrRecall	Distance d (m)									
Time t (s)	20	30	40	53	60	70	80	90	100	
300	0.496	0.504	0.511	0.525	0.525	0.518	0.518	0.561	0.511	
600	0.460	0.540	0.540	0.561	0.590	0.554	0.568	0.597	0.576	
900	0.403	0.468	0.532	0.568	0.568	0.576	0.568	0.604	0.590	
1200	0.360	0.432	0.489	0.547	0.554	0.554	0.554	0.583	0.547	
1500	0.288	0.396	0.439	0.496	0.504	0.511	0.525	0.561	0.511	
1800	0.230	0.381	0.432	0.489	0.504	0.504	0.504	0.540	0.504	
2100	0.223	0.345	0.396	0.475	0.504	0.504	0.482	0.504	0.475	

TrPrecision	Distance d (m)									
Time t (s)	20	30	40	53	60	70	80	90	100	
300	0.259	0.340	0.392	0.399	0.408	0.407	0.393	0.411	0.376	
600	0.356	0.469	0.503	0.542	0.569	0.527	0.552	0.589	0.571	
900	0.364	0.464	0.552	0.594	0.594	0.606	0.617	0.656	0.661	
1200	0.342	0.469	0.548	0.613	0.636	0.616	0.636	0.669	0.650	
1500	0.308	0.462	0.517	0.595	0.614	0.617	0.640	0.672	0.623	
1800	0.262	0.465	0.536	0.613	0.636	0.631	0.625	0.670	0.636	
2100	0.267	0.444	0.509	0.600	0.636	0.631	0.615	0.642	0.611	

TrFmeas	Distance d (m)									
Time t (s)	20	30	40	53	60	70	80	90	100	
300	0.341	0.406	0.444	0.453	0.459	0.456	0.447	0.474	0.433	
600	0.401	0.502	0.521	0.551	0.580	0.540	0.560	0.593	0.573	
900	0.382	0.466	0.542	0.581	0.581	0.590	0.592	0.629	0.624	
1200	0.351	0.449	0.517	0.578	0.592	0.583	0.592	0.623	0.594	
1500	0.297	0.426	0.475	0.541	0.553	0.559	0.577	0.612	0.561	
1800	0.245	0.419	0.478	0.544	0.562	0.560	0.558	0.598	0.562	
2100	0.243	0.389	0.445	0.530	0.562	0.560	0.540	0.565	0.534	

Figure 54. Values for confusion matrix from stays extraction

Our parameter combination reaches a 63.9 % for F measure, corresponding to a detection of a 60 % of the real stays reported in GTD (recall); meanwhile, a 68.4 % of the stays detected by the algorithm match with real tagged stays (precision).

ANALYSIS OF THE EXTRACTION

The indicators proposed for the analysis of the accuracy of the stays extracted are calculated for *User1*. As we have explained, the parameter settings used for this extraction are the best clustering parameters of $d = 53$ meters and $t = 900$ sec.

- 1. **Number** of stays at each tagged place per weekday
 - o 1.1. Number of detected occurrences
 - o 1.2. Number of tagged occurrences
 - o 1.3. Proportion of tagged stays detected
- 2. **Duration** of the stays at tagged visited place per weekday
 - o 2.1. Duration of detected stays
 - o 2.2. Duration of tagged stays
 - o 2.3. Proportion of tagged stays duration detected

Now, the resulting tables are presented and the indicators are explained further for a better understanding of the reader.

➤ 1. Number of stays at each tagged place per weekday.

Table 24. Number of stays at each tagged place for each weekday.

Number of stays at each tagged place for each weekday									
Detected stays									
TpID	Mon	Tue	Wed	Thu	Fri	Sat	Sun	SUM	
1	3	4	4	2	2	0	0	15	
2	0	1	0	0	0	0	0	1	
3	4	3	6	8	7	1	4	33	
4	0	0	0	1	10	14	12	37	
5	0	0	0	0	1	0	0	1	
6	0	0	0	1	0	0	0	1	
7	0	0	0	0	2	1	2	5	
8	0	0	0	0	0	0	1	1	
9	0	0	0	0	0	0	1	1	
10	0	0	0	1	0	0	0	1	
11	0	1	0	0	0	0	0	1	
12	0	0	1	0	0	0	0	1	
13	0	0	0	0	1	0	0	1	
14	0	0	0	0	0	0	0	0	
15	0	0	0	0	0	1	0	1	
16	0	0	0	0	0	1	0	1	
17	0	0	0	0	0	1	0	1	
18	0	0	0	0	0	0	1	1	
19	0	0	0	0	0	0	1	1	
20	1	0	0	0	0	0	0	1	
21	0	0	0	0	0	0	0	0	
22	0	0	0	0	0	0	0	0	
23	0	0	0	0	0	0	0	0	
24	0	0	0	0	0	0	0	0	
25	0	0	0	0	0	0	0	0	
26	0	0	0	0	0	0	0	0	
27	0	0	0	0	0	0	1	1	
28	0	0	0	0	0	0	1	1	
29	0	0	0	0	0	1	0	1	
30	0	0	0	0	0	0	0	0	
SUM	8	9	11	13	23	20	24	108	

Tagged stays									
TpID	Mon	Tue	Wed	Thu	Fri	Sat	Sun	SUM	
1	8	5	4	6	6	0	0	29	
2	1	1	0	0	0	0	0	2	
3	11	10	11	12	8	2	7	61	
4	0	0	0	3	11	16	13	43	
5	0	0	0	1	1	1	1	4	
6	0	0	0	1	0	0	0	1	
7	0	0	0	0	2	2	2	6	
8	0	0	0	0	0	0	1	1	
9	0	0	0	0	0	0	1	1	
10	0	0	0	1	0	0	0	1	
11	0	1	1	1	0	0	0	3	
12	0	0	1	0	0	0	0	1	
13	0	0	0	0	1	0	0	1	
14	0	0	0	0	1	0	0	1	
15	0	0	0	0	0	1	0	1	
16	0	0	0	0	0	1	0	1	
17	0	0	0	0	0	1	0	1	
18	0	0	0	0	0	0	1	1	
19	0	0	0	0	0	0	1	1	
20	1	0	0	0	0	0	0	1	
21	1	0	0	0	0	0	0	1	
22	1	0	0	0	0	0	0	1	
23	1	3	2	2	0	0	0	8	
24	0	1	0	0	0	0	0	1	
25	0	1	1	1	0	0	0	3	
26	0	0	1	0	0	0	0	1	
27	0	0	0	0	0	0	1	1	
28	0	0	0	0	0	0	1	1	
29	0	0	0	0	0	1	0	1	
30	0	0	0	0	0	1	0	1	
SUM	24	22	21	28	30	26	29	180	

Proportion of tagged stays detected									
TpID	Mon	Tue	Wed	Thu	Fri	Sat	Sun	PROP	
1	0.38	0.80	1.00	0.33	0.33			0.52	
2	0.00	1.00						0.50	
3	0.36	0.30	0.55	0.67	0.88	0.50	0.57	0.54	
4				0.33	0.91	0.88	0.92	0.86	
5				0.00	1.00	0.00	0.00	0.25	
6				1.00				1.00	
7					1.00	0.50	1.00	0.83	
8							1.00	1.00	
9							1.00	1.00	
10				1.00				1.00	
11		1.00	0.00	0.00				0.33	
12			1.00					1.00	
13					1.00			1.00	
14					0.00			0.00	
15						1.00		1.00	
16						1.00		1.00	
17						1.00		1.00	
18							1.00	1.00	
19							1.00	1.00	
20	1.00							1.00	
21	0.00							0.00	
22	0.00							0.00	
23	0.00	0.00	0.00	0.00				0.00	
24		0.00						0.00	
25		0.00	0.00	0.00				0.00	
26			0.00					0.00	
27							1.00	1.00	
28							1.00	1.00	
29						1.00		1.00	
30						0.00		0.00	
PROP	0.33	0.41	0.52	0.46	0.77	0.77	0.83	0.60	

The values have been displayed with a graduated colour scale as done with clustering QE results. It represents data from higher values, in green, to lower values, in red. The “SUM” column and row are explained separately.

First of all, it has to be pointed out that 73.83 % of the stays have been done at the three most visited places in which 83 % of the tagged time has been spent. These places are the **main home** or *Home1* (ID 3), a **secondary home** or *Home2* (ID 4) and the **work place** or *Work* (ID 1) of *User1*.

In the central section, the number of visits to each place has been counted by weekday. For instance, the **tagged place 3** (*Home1*) has been the most visited. 11 visits have been reported on Mondays during the whole GTD period, whereas only two have been tagged on Saturdays. In the section at the right, for instance, we can observe that only a 36 % of the visits reported to tagged place 3 on Mondays have been detected by the algorithm.

The **column** SUM summarises the number of visits to each place while the **row** SUM summarises the number of visits done in each weekday. The **column** PROP at the right border represents the proportion between the total detected stays at each tagged place and the total of stays reported in such tagged place as GTD. For example, 33 visits have been detected to tagged place 3, which are the 54 % of the 61 visits

tagged for such place. The **row** PROP reflects the proportion between the total visits detected in one weekday and the total visits tagged in such day.

We can observe that the algorithm detects more than 52 % of the visits done to the most visited tagged places during the GTD collection (1, 3 and 4). Some stays at tagged places which have received only one visit during the data collection have been detected (6, 8 to 10, 12, 13, 15 to 20, 27 to 29). For these places the stays detection rate is perfect (100 %).

On the other hand, stays at tagged places 21, 22, 24, 25 and 30 have not been detected. In these cases, the initial reason of the lack of temporal detection is explained because such tagged places had not been spatially detected in the clustering. Thus, there is no possible time extraction because any detected place was spatially assigned to these tagged places. In some occasions, a manual inspection of the detected places spatially not assigned to tagged places showed a time correspondence of stays detected with stays tagged. These tagged stays are therefore correctly detected but not reported due to their belonging to detected places (clusters) out of the 53 meters circular buffer around any tagged place.

Finally, regarding the proportion of GTD stays detected for every weekday, the best detection rates correspond to weekends (> 77 %) whereas the worst performance corresponds to Mondays (33 %).

➤ 2. Duration of the stays at each place per weekday

Table 25. Duration of stays at each tagged place for each weekday.

Stays duration at each tagged place for each weekday																										
Detected stays duration (h)									Tagged stays duration (h)									Proportion tagged stays duration detected								
TpID	Mon	Tue	Wed	Thu	Fri	Sat	Sun	SUM	TpID	Mon	Tue	Wed	Thu	Fri	Sat	Sun	SUM	TpID	Mon	Tue	Wed	Thu	Fri	Sat	Sun	PROP
1	23.48	22.02	31.85	13.48	9.10	0.00	0.00	99.93	1	36.73	22.52	31.85	29.77	15.47	0.00	0.00	136.3	1	0.64	0.98	1.00	0.45	0.59			0.73
2	0.00	2.27	0.00	0.00	0.00	0.00	0.00	2.27	2	0.50	2.27	0.00	0.00	0.00	0.00	0.00	2.77	2	0.00	1.00						0.82
3	23.48	17.97	40.55	42.20	28.68	11.78	30.35	195.0	3	76.23	65.90	67.03	60.38	36.58	12.28	47.70	366.1	3	0.31	0.27	0.60	0.70	0.78	0.96	0.64	0.53
4	0.00	0.00	0.00	4.52	37.43	86.80	61.32	190.1	4	0.00	0.00	0.00	5.22	42.98	96.15	61.45	205.8	4				0.87	0.87	0.90	1.00	0.92
5	0.00	0.00	0.00	0.00	0.93	0.00	0.00	0.93	5	0.00	0.00	0.00	0.10	0.93	3.72	1.60	6.35	5				0.00	1.00	0.00	0.00	0.15
6	0.00	0.00	0.00	0.65	0.00	0.00	0.00	0.65	6	0.00	0.00	0.00	0.65	0.00	0.00	0.00	0.65	6				1.00				1.00
7	0.00	0.00	0.00	0.00	5.07	3.13	5.80	14.00	7	0.00	0.00	0.00	0.00	5.07	5.85	5.80	16.72	7					1.00	0.54	1.00	0.84
8	0.00	0.00	0.00	0.00	0.00	0.00	0.83	0.83	8	0.00	0.00	0.00	0.00	0.00	0.00	0.83	0.83	8							1.00	1.00
9	0.00	0.00	0.00	0.00	0.00	0.00	1.52	1.52	9	0.00	0.00	0.00	0.00	0.00	0.00	1.52	1.52	9							1.00	1.00
10	0.00	0.00	0.00	1.93	0.00	0.00	0.00	1.93	10	0.00	0.00	0.00	1.93	0.00	0.00	0.00	1.93	10				1.00				1.00
11	0.00	8.70	0.00	0.00	0.00	0.00	0.00	8.70	11	0.00	8.70	9.20	9.38	0.00	0.00	0.00	27.28	11		1.00	0.00	0.00				0.32
12	0.00	0.00	3.42	0.00	0.00	0.00	0.00	3.42	12	0.00	0.00	3.42	0.00	0.00	0.00	0.00	3.42	12			1.00					1.00
13	0.00	0.00	0.00	0.00	1.40	0.00	0.00	1.40	13	0.00	0.00	0.00	0.00	1.40	0.00	0.00	1.40	13					1.00			1.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	14	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.50	14					0.00			0.00
15	0.00	0.00	0.00	0.00	0.00	1.00	0.00	1.00	15	0.00	0.00	0.00	0.00	0.00	1.00	0.00	1.00	15						1.00		1.00
16	0.00	0.00	0.00	0.00	0.00	0.70	0.00	0.70	16	0.00	0.00	0.00	0.00	0.00	0.70	0.00	0.70	16						1.00		1.00
17	0.00	0.00	0.00	0.00	0.00	2.48	0.00	2.48	17	0.00	0.00	0.00	0.00	0.00	2.48	0.00	2.48	17						1.00		1.00
18	0.00	0.00	0.00	0.00	0.00	0.00	0.17	0.17	18	0.00	0.00	0.00	0.00	0.00	0.00	0.17	0.17	18							1.00	1.00
19	0.00	0.00	0.00	0.00	0.00	0.00	0.32	0.32	19	0.00	0.00	0.00	0.00	0.00	0.00	0.32	0.32	19							1.00	1.00
20	1.27	0.00	0.00	0.00	0.00	0.00	0.00	1.27	20	1.27	0.00	0.00	0.00	0.00	0.00	0.00	1.27	20	1.00							1.00
21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	21	0.80	0.00	0.00	0.00	0.00	0.00	0.00	0.80	21	0.00							0.00
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	22	0.22	0.00	0.00	0.00	0.00	0.00	0.00	0.22	22	0.00							0.00
23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	23	9.87	14.37	10.88	10.58	0.00	0.00	0.00	45.70	23	0.00	0.00	0.00	0.00				0.00
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	24	0.00	4.78	0.00	0.00	0.00	0.00	0.00	4.78	24		0.00						0.00
25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	25	0.00	4.28	8.32	7.55	0.00	0.00	0.00	20.15	25		0.00	0.00	0.00				0.00
26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	26	0.00	0.00	3.13	0.00	0.00	0.00	0.00	3.13	26			0.00					0.00
27	0.00	0.00	0.00	0.00	0.00	0.00	1.83	1.83	27	0.00	0.00	0.00	0.00	0.00	0.00	1.83	1.83	27						1.00		1.00
28	0.00	0.00	0.00	0.00	0.00	0.00	1.22	1.22	28	0.00	0.00	0.00	0.00	0.00	0.00	1.22	1.22	28							1.00	1.00
29	0.00	0.00	0.00	0.00	0.00	0.73	0.00	0.73	29	0.00	0.00	0.00	0.00	0.00	0.73	0.00	0.73	29						1.00		1.00
30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	30	0.00	0.00	0.00	0.00	0.00	0.35	0.00	0.35	30						0.00		0.00
SUM	48.2	50.9	75.8	62.8	82.6	106.6	103.3	530.4	SUM	125.6	122.8	133.8	125.6	102.9	123.3	122.4	856.5	PROF	0.38	0.41	0.57	0.50	0.80	0.87	0.84	0.62

In the central part of Table 25, the duration of the visits to each tagged place has been summarised by weekday. For example, as seen before, the **tagged place 3 (Home1)** is the most visited in terms of

number of visits and also in terms of total duration of such stays. On Mondays, 76.23 hours have been spent at this place, whereas the algorithm has detected only 23.48 (31 %).

The **column** SUM summarises the total hours stayed at each place while the **row** SUM summarises the total time stayed at places in each weekday. The **column** PROP at the right border represents the proportion between the total detected stay time at each tagged place and the total stay time reported at such tagged place as GTD. For example, 195 hours of dwell time have been detected at tagged place 3, which are the 53 % of the 366.1 stayed hours tagged for such place. The **row** PROP reflects the proportion between the total stay time detected in one weekday and the total stay time tagged in such day.

We can observe that the algorithm detects more than 53 % of the time spent at the most visited tagged places during the GTD collection: 1, 3 and 4 i.e. *Work*, *Home1* and *Home2*. The places at which the detection of stay occurrences was perfect also show a 100 % of stay time detected. Similarly, the places not spatially detected in the clustering which did not present any stay detected, obviously, does not present any stay duration.

Finally, regarding the proportion of GTD stayed time detected for every weekday, the best detection rates correspond also to weekends (> 80 %) whereas the worst performance corresponds again to Mondays (38 %). Proportions of total time detected are greater than the proportions of stays occurrences detected analysed in the previous section. This can be explained because the time spent in detected tagged stays is greater than the time spent in non-detected tagged stays.

5.2.2.2. Extraction of transitions

QUALITY OF THE EXTRACTION

➤ Proportion of transition time extracted

TranT_ext	Distance d (m)								
Time t (s)	20	30	40	53	60	70	80	90	100
300	0.431	0.415	0.370	0.361	0.364	0.373	0.365	0.404	0.293
600	0.420	0.493	0.471	0.483	0.519	0.466	0.482	0.503	0.468
900	0.349	0.420	0.493	0.493	0.493	0.520	0.506	0.545	0.530
1200	0.331	0.404	0.480	0.499	0.500	0.500	0.503	0.524	0.502
1500	0.251	0.363	0.420	0.445	0.446	0.450	0.457	0.480	0.468
1800	0.222	0.341	0.415	0.443	0.446	0.449	0.449	0.471	0.450
2100	0.214	0.296	0.363	0.418	0.446	0.449	0.425	0.423	0.421

Figure 55. Proportion of tagged transition time detected for User1.

This table for *User1* shows the proportion between total time during transitions detected by the algorithm and the total transition time tagged from GTD. As we observed in Figure 55, the maximum value of a 54.5 % of total tagged transition time detected is obtained with a parameter setting of 900s/90m. However, according to the global QE, the **F measure** corresponding to this setting reaches only a 68.6 % because of a lower **recall** of the clustering (66.7 %). We have already explained within the user stays section the reasons for using the parameter combination 900 sec /53 m for the extraction analysis.

It must be pointed out that in the GTD revision two anomalous artefacts have been found. These are two transitions of more than 10 hours (49.860 and 145.800 sec.). As these have been interpreted as GTD collection errors, they have been filtered out for the quality evaluation and characterisation of the extraction.

➤ Confusion matrix

As done before for the stays, the values from a confusion matrix have been generated from the transitions extraction so as to evaluate the performance of the process.

The maximum F measure (62.9 %) is reached with a parameter setting 900s/90m, but again it corresponds to a lower clustering performance compared to the selected parameters for the characterisation.

TrRecall		Distance d (m)								
Time t (s)		20	30	40	53	60	70	80	90	100
300		0.496	0.504	0.511	0.525	0.525	0.518	0.518	0.561	0.511
600		0.460	0.540	0.540	0.561	0.590	0.554	0.568	0.597	0.576
900		0.403	0.468	0.532	0.568	0.576	0.568	0.604	0.590	
1200		0.360	0.432	0.489	0.547	0.554	0.554	0.583	0.547	
1500		0.288	0.396	0.439	0.496	0.504	0.511	0.525	0.561	0.511
1800		0.230	0.381	0.432	0.489	0.504	0.504	0.504	0.540	0.504
2100		0.223	0.345	0.396	0.475	0.504	0.504	0.482	0.504	0.475

TrPrecision		Distance d (m)								
Time t (s)		20	30	40	53	60	70	80	90	100
300		0.259	0.340	0.392	0.399	0.408	0.407	0.393	0.411	0.376
600		0.356	0.469	0.503	0.542	0.569	0.527	0.552	0.589	0.571
900		0.364	0.464	0.552	0.594	0.594	0.606	0.617	0.656	0.661
1200		0.342	0.469	0.548	0.613	0.636	0.616	0.636	0.669	0.650
1500		0.308	0.462	0.517	0.595	0.614	0.617	0.640	0.672	0.623
1800		0.262	0.465	0.536	0.613	0.636	0.631	0.625	0.670	0.636
2100		0.267	0.444	0.509	0.600	0.636	0.631	0.615	0.642	0.611

TrFmeas		Distance d (m)								
Time t (s)		20	30	40	53	60	70	80	90	100
300		0.341	0.406	0.444	0.453	0.459	0.456	0.447	0.474	0.433
600		0.401	0.502	0.521	0.551	0.580	0.540	0.560	0.593	0.573
900		0.382	0.466	0.542	0.581	0.581	0.590	0.592	0.629	0.624
1200		0.351	0.449	0.517	0.578	0.592	0.583	0.592	0.623	0.594
1500		0.297	0.426	0.475	0.541	0.553	0.559	0.577	0.612	0.561
1800		0.245	0.419	0.478	0.544	0.562	0.560	0.558	0.598	0.562
2100		0.243	0.389	0.445	0.530	0.562	0.560	0.540	0.565	0.534

Figure 56. Values for confusion matrix from transitions extraction

Our parameter combination reaches a 58.1 % of F measure, corresponding to a detection of a 56.8 % of the real transitions reported in GTD (recall); meanwhile, a 59.4 % of the transitions detected by the algorithm match with real tagged transitions (precision).

ANALYSIS OF THE EXTRACTION

- 1. Number of transitions between tagged places per weekday
 - o 1.1. Number of detected occurrences
 - o 1.2. Number of tagged occurrences
 - o 1.3. Proportion of tagged transitions detected

- 2. Duration of the transitions between tagged places per weekday
 - o 2.1. Duration of detected transitions
 - o 2.2. Duration of tagged transitions
 - o 2.3. Proportion of tagged transitions duration detected

➤ 1. Number of transitions between tagged places per weekday

Table 26 shows the possible transitions reported on GTD as well as those detected by the algorithm. In the central part of the table, we can find the number of times a transition has been counted for a specific weekday. For example, the transition between tagged places 3 and 1 (*Home1* -> *Work*) has been the most common during the 40 days GTD collection period. Up to 3 times this transition has been done by *User1* on Mondays from a total of 20. The left part of the table shows the algorithm has extracted 12 (60 %) of these transitions between tagged places 3 and 1.

The column SUM summarises the number of times a transition has been counted while the row SUM summarises the number of transitions counted in each weekday. The column PROP reflects the proportion between the total detected occurrences of one transition and the total tagged occurrences of such transition. Meanwhile, the row PROP represents the proportion between the total transitions detected in one weekday and the total transitions tagged for such day.

The algorithm has been able to extract most of the transitions (60 %) between *Home1* and *Work* (03-01) and (79 %) the opposite trip (01-03). Moreover 80 % of the transitions starting and ending at *Home1* (03-03) have been also detected; these are short trips around *Home1* that if include stops, these are too short to be significant e.g. traffic light. Nevertheless, the algorithm only extracted 17 % of the trips around *Work* (01-01). On the other hand, most of the transitions affecting tagged place 4 (*Home2*) have been detected, including 90 % of the trips around *Home2* (04-04).

The algorithm only extracts 30 % of the transitions between *Home1* and *Home2* (03-04) whereas it reaches a 40 % for the opposite direction (04-03). Obviously, any of the transitions affecting the tagged places 21, 22, 24 and 25 have been detected; those are some of the mentioned spatially non-detected tagged places. As reported for stays, some of these transitions can be identified when manually comparing timestamps of detected places close to tagged places but far than the 53 meters circular buffer.

Finally, if we consider the proportion of GTD transitions detected for every weekday, transitions done in weekends present the best detection rates as when analysing stays. In this case, the worst performance corresponds to Wednesday with a 31 % of transitions detected.

Table 26. Number of transitions between tagged places for each weekday.

Transitions between tagged places for each weekday									
Detected transitions									
Tran	Mon	Tue	Wed	Thu	Fri	Sat	Sun	SUM	
01-01	0	1	0	0	0	0	0	1	
01-02	0	0	0	0	0	0	0	0	
01-03	3	3	3	2	4	0	0	15	
01-31	0	0	0	0	0	0	0	0	
02-01	0	0	0	0	0	0	0	0	
02-03	0	0	0	0	0	0	0	0	
03-01	3	1	1	3	4	0	0	12	
03-03	1	1	1	1	0	0	0	4	
03-04	0	0	0	0	2	0	0	2	
03-10	0	0	0	0	0	0	0	0	
03-11	0	0	0	0	0	0	0	0	
03-12	0	0	0	0	0	0	0	0	
03-13	0	0	0	0	1	0	0	1	
03-19	1	0	0	0	0	0	0	1	
03-26	0	0	0	0	0	0	1	1	
04-03	0	0	0	0	0	0	2	2	
04-04	0	0	0	0	3	4	2	9	
04-05	0	0	0	0	1	1	1	3	
04-06	0	0	0	0	0	0	0	0	
04-07	0	0	0	0	2	2	2	6	
04-08	0	0	0	0	0	0	1	1	
04-09	0	0	0	0	0	0	1	1	
04-14	0	0	0	0	0	1	0	1	
04-16	0	0	0	0	0	1	0	1	
04-17	0	0	0	0	0	0	0	0	
04-28	0	0	0	0	0	1	0	1	
05-04	0	0	0	0	1	0	0	1	
06-04	0	0	0	1	0	0	0	1	
07-04	0	0	0	0	2	1	2	5	
08-04	0	0	0	0	0	0	1	1	
09-04	0	0	0	0	0	0	1	1	
10-03	0	0	0	1	0	0	0	1	
11-03	0	1	0	0	0	0	0	1	
12-03	0	0	0	0	0	0	0	0	
13-04	0	0	0	0	1	0	0	1	
14-15	0	0	0	0	0	0	0	0	
15-04	0	0	0	0	0	1	0	1	
16-04	0	0	0	0	0	1	0	1	
17-18	0	0	0	0	0	0	0	0	
18-04	0	0	0	0	0	0	1	1	
19-20	0	0	0	0	0	0	0	0	
20-21	0	0	0	0	0	0	0	0	
21-22	0	0	0	0	0	0	0	0	
22-23	0	0	0	0	0	0	0	0	
22-24	0	0	0	0	0	0	0	0	
23-22	0	0	0	0	0	0	0	0	
24-22	0	0	0	0	0	0	0	0	
24-25	0	0	0	0	0	0	0	0	
25-22	0	0	0	0	0	0	0	0	
26-27	0	0	0	0	0	0	1	1	
27-03	0	0	0	0	0	0	1	1	
28-29	0	0	0	0	0	0	0	0	
29-04	0	0	0	0	0	0	0	0	
31-01	0	0	0	0	0	0	0	0	
SUM	8	7	5	8	21	13	17	79	
Tagged transitions									
Tran	Mon	Tue	Wed	Thu	Fri	Sat	Sun	SUM	
01-01	2	1	0	2	1	0	0	6	
01-02	1	1	0	0	0	0	0	2	
01-03	4	3	4	4	4	0	0	19	
01-31	0	0	0	0	1	0	0	1	
02-01	1	0	0	0	0	0	0	1	
02-03	0	1	0	0	0	0	0	1	
03-01	5	3	4	4	4	0	0	20	
03-03	1	1	1	1	0	1	0	5	
03-04	0	0	0	1	3	0	0	4	
03-10	0	0	0	1	0	0	0	1	
03-11	0	1	1	1	0	0	0	3	
03-12	0	0	1	0	0	0	0	1	
03-13	0	0	0	0	1	0	0	1	
03-19	1	0	0	0	0	0	0	1	
03-26	0	0	0	0	0	0	1	1	
04-03	0	0	0	0	0	0	5	5	
04-04	0	0	0	0	3	5	2	10	
04-05	0	0	0	1	1	1	1	4	
04-06	0	0	0	1	0	0	0	1	
04-07	0	0	0	0	2	2	2	6	
04-08	0	0	0	0	0	0	1	1	
04-09	0	0	0	0	0	0	1	1	
04-14	0	0	0	0	0	1	0	1	
04-16	0	0	0	0	0	1	0	1	
04-17	0	0	0	0	0	0	1	1	
04-28	0	0	0	0	0	1	0	1	
05-04	0	0	0	1	1	1	1	4	
06-04	0	0	0	1	0	0	0	1	
07-04	0	0	0	0	2	2	2	6	
08-04	0	0	0	0	0	0	1	1	
09-04	0	0	0	0	0	0	1	1	
10-03	0	0	0	1	0	0	0	1	
11-03	0	1	0	1	0	0	0	2	
12-03	0	0	1	0	0	0	0	1	
13-04	0	0	0	0	1	0	0	1	
14-15	0	0	0	0	0	1	0	1	
15-04	0	0	0	0	0	1	0	1	
16-04	0	0	0	0	0	1	0	1	
17-18	0	0	0	0	0	0	1	1	
18-04	0	0	0	0	0	0	1	1	
19-20	1	0	0	0	0	0	0	1	
20-21	1	0	0	0	0	0	0	1	
21-22	1	0	0	0	0	0	0	1	
22-23	0	1	0	0	0	0	0	1	
22-24	0	1	1	1	0	0	0	3	
23-22	0	1	0	0	0	0	0	1	
24-22	0	1	0	1	0	0	0	2	
24-25	0	0	1	0	0	0	0	1	
25-22	0	0	1	0	0	0	0	1	
26-27	0	0	0	0	0	0	1	1	
27-03	0	0	0	0	0	0	1	1	
28-29	0	0	0	0	0	1	0	1	
29-04	0	0	0	0	0	1	0	1	
31-01	0	0	0	0	1	0	0	1	
SUM	18	16	15	22	25	20	23	139	
Proportion of tagged transitions detected									
Tran	Mon	Tue	Wed	Thu	Fri	Sat	Sun	PROP	
01-01	0.00	1.00			0.00	0.00		0.17	
01-02	0.00	0.00						0.00	
01-03	0.75	1.00	0.75	0.50	1.00			0.79	
01-31					0.00			0.00	
02-01	0.00							0.00	
02-03		0.00						0.00	
03-01	0.60	0.33	0.25	0.75	1.00			0.60	
03-03	1.00	1.00	1.00	1.00		0.00		0.80	
03-04				0.00	0.67			0.50	
03-10				0.00				0.00	
03-11		0.00	0.00	0.00				0.00	
03-12			0.00					0.00	
03-13					1.00			1.00	
03-19	1.00							1.00	
03-26							1.00	1.00	
04-03							0.40	0.40	
04-04				1.00	0.80	1.00		0.90	
04-05			0.00	1.00	1.00	1.00		0.75	
04-06			0.00					0.00	
04-07				1.00	1.00	1.00		1.00	
04-08							1.00	1.00	
04-09							1.00	1.00	
04-14						1.00		1.00	
04-16						1.00		1.00	
04-17							0.00	0.00	
04-28						1.00		1.00	
05-04			0.00	1.00	0.00	0.00		0.25	
06-04			1.00					1.00	
07-04				1.00	0.50	1.00		0.83	
08-04							1.00	1.00	
09-04							1.00	1.00	
10-03				1.00				1.00	
11-03		1.00		0.00				0.50	
12-03			0.00					0.00	
13-04					1.00			1.00	
14-15						0.00		0.00	
15-04						1.00		1.00	
16-04						1.00		1.00	
17-18							0.00	0.00	
18-04							1.00	1.00	
19-20	0.00							0.00	
20-21	0.00							0.00	
21-22	0.00							0.00	
22-23		0.00						0.00	
22-24		0.00	0.00	0.00				0.00	
23-22		0.00						0.00	
24-22		0.00		0.00				0.00	
24-25			0.00					0.00	
25-22			0.00					0.00	
26-27							1.00	1.00	
27-03							1.00	1.00	
28-29						0.00		0.00	
29-04						0.00		0.00	
31-01					0.00			0.00	
PROP	0.44	0.44	0.33	0.36	0.84	0.65	0.74	0.57	

➤ 2. Duration of the transitions between tagged places per weekday.

In the central section of the Table 27 a summarisation of the duration of the tagged transitions between tagged places and for each weekday is presented. For instance, the greater time invested in movement has been in the transition 04-04 and the algorithm has correctly extracted a 98 % of this time. This is interesting because it represents movements around place *Home2* and *User1* explained he enjoys plenty of outdoor activities when he visits this location during weekends. Further analysis about User1 mobility will be presented in the following section.

Regarding *Home2* (ID 04), most of the time invested in transitions to and from this place was extracted, reaching a 98 % of the time in trips around *Home2* (04-04). Kang has been able to detect only a 45 % of the tagged time in transitions *Home1-Home2* (03-04) and a 40 % for the opposite direction (04-03).

Finally, the daily transition time detection performed better for weekends, with more than 54 % of the time extracted. In comparison, only a 21 % of the time invested in transitions on Mondays has been detected.

5.2.3. Possible applications

The general approach presented in this thesis allows the detection of the places visited by a mobile element or human user. It also provides an approach to extract the stays performed at these places as well as the transitions between the locations. Moreover, two indicators have been suggested to determine the accuracy of the extraction whilst facilitating the detection of user's movement behaviour patterns.

Two possible applications of our approach are suggested in this section: the visualisation of the movement behaviour and the prediction of the future movements of the user.

5.2.3.1. Movement behaviour visualization

Different visualizations were prepared so as to facilitate the analysis of the GTD as well as the detected visited places, stays and transitions. This gave us a better insight into the data and good sense of the accuracy of the extraction performed by the sub-approaches.

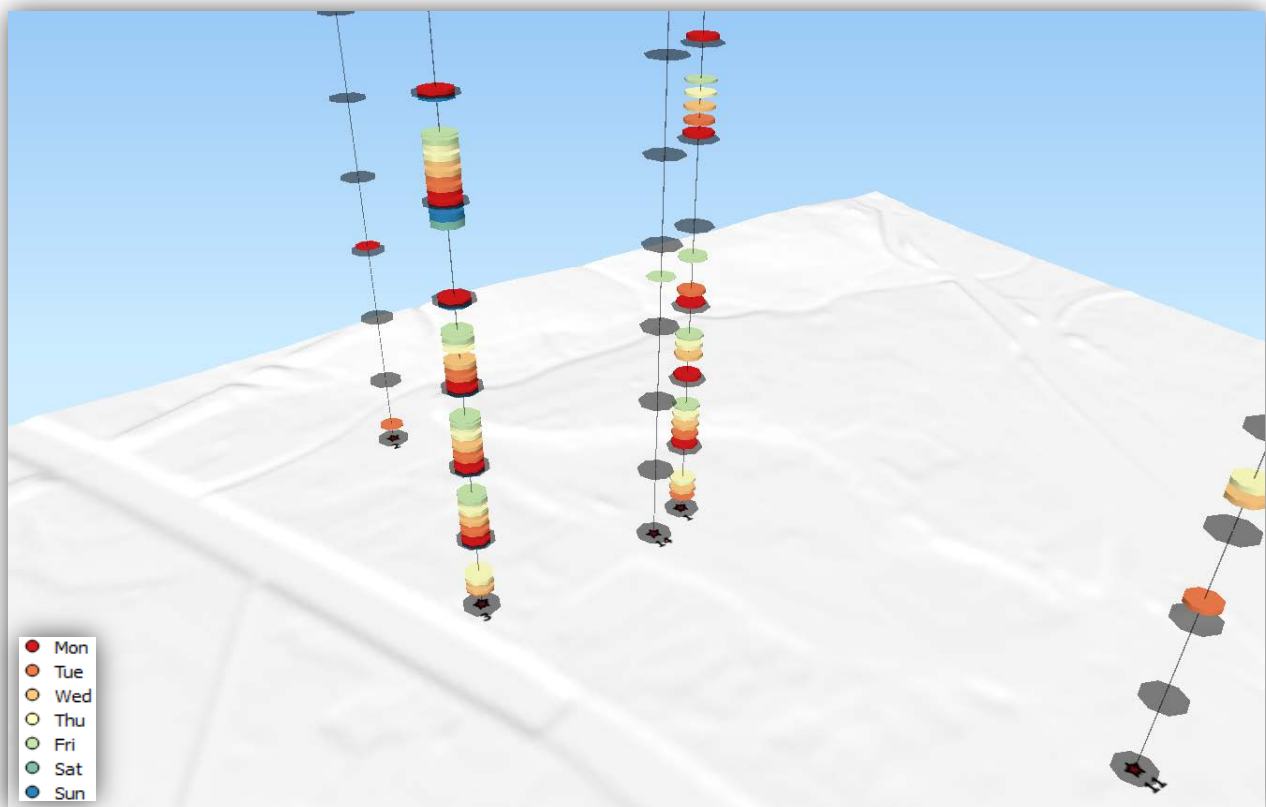


Figure 57. Visualization of stays as stacks of cylinders. *Home1* and *Work* area.

A first option consists of the representation of the stays at tagged places as cylinders. The diameter of such cylinder is fixed at an adequate value for a correct display. The base of each cylinder is set at a height which represents the time start of the stay. The height of each cylinder is obtained extruding the base proportionally to the duration of the stay it represents. Each stay has been represented with a colour depending on the weekday of the stay.

The first stay starts on 2014-08-12 (Tuesday) and as time 0, it has been chosen the date 2014-08-11 because it is a Monday. This has helped for the representation of the natural weeks affected by the GTD collection (black transparent circles). Hence, we can observe 7 weeks of data piled up but the total period of GTD covers only 41 days. First day is a Tuesday and we can observe the first stay as an orange form at the base of tagged place 1 (*Work*).

This representation allows for a quick overview of the stays distribution of the user. It is possible observing the sequence of stays during the week and the regularity of the of the user's behaviour between different weeks. For instance, we can realise that most of the weekends user is not at places 1 or 3 (*Work* and *Home1*), he usually is at *Home2* as we mentioned in the previous sections. Similarly, most of working days User1 spends time at *Home1*. User also stays some time on Friday (before moving to *Home2*).

An additional aspect of the information on display is the regularity and the height of the gaps between stays. Gaps at *Work* stack are very regular during the week because stays usually have a similar length of 8 hours (8 a.m. to 16 p.m.), whereas gaps at *Home1* separate at least two main blocks which mainly represent mornings (0 a.m. – 8 a.m.) and evenings (16 p.m. – 0 a.m.) before and after working.

We can detect a gap in the fifth week at *Home1* and *Work* which is explained by the assistance of the user to an international conference. In the figure it is also possible identifying three long stays at place 11 during working time (gaps in the *Work* stack): these were full day training sessions at an academic institution of the city.



Figure 58. Visualization of stays as stacks of cylinders. *Home2* area.

Figure 58 shows the area around tagged place 4 (*Home2*). We can observe a number of short stays in the surroundings which represent some of the outdoor activities of User1 during weekends. Moreover, the small height of the gaps between stays reflects the mentioned relatively short stays out of *Home2*. This could explain the higher rates of detection of total stay duration during weekend, because user spends most of the time at *Home2* which is a tagged place well detected by the algorithm. User can better avoid GPS problems related to low battery and cold start. Here also week five presents an absence of stays due to the mentioned international conference.

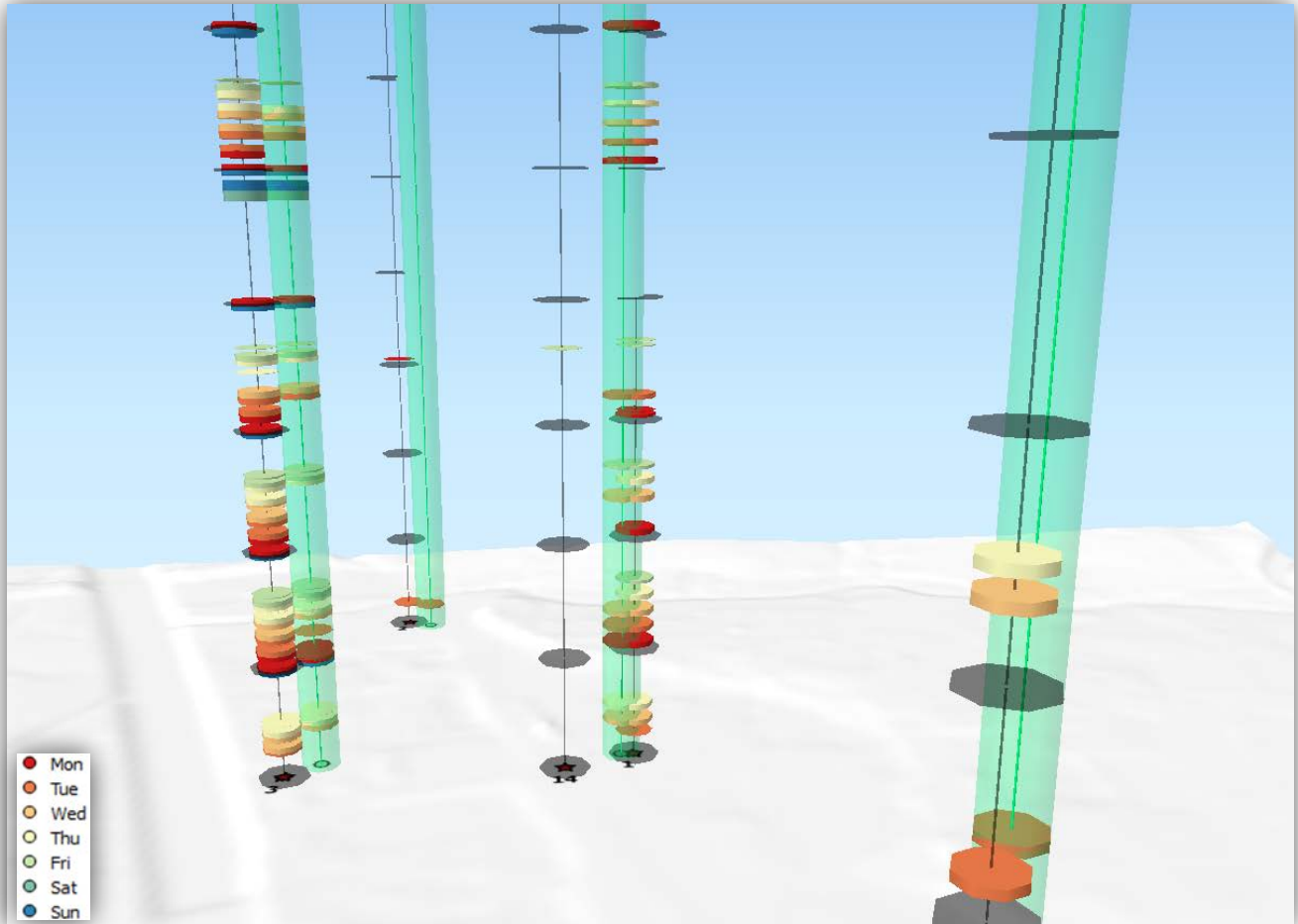


Figure 59. Visualization of detected and tagged stays as stacks of cylinders.

In Figure 59 an additional set of stacks has been added representing the detected stays within a transparent green cylinder. The green tubes are displaced because they are located exactly in the locations detected by the algorithm. These are the clusters or detected places which have been spatially related to tagged places closer than 53 meters; the rest of detected places are not displayed, despite being relatively close in some cases. Thus we can observe one-to-one the correspondences between tagged and detected stays.

Visualization of transitions

The transitions are represented as pipes piled up and coloured according to the weekday they were done. The tagged places have been represented as thin cylinders for reference while the natural weeks have been again identified with black dishes. The pipes are located at a height proportional to the starting time i.e. arriving time is not represented due to the lack of inclination in the lines.

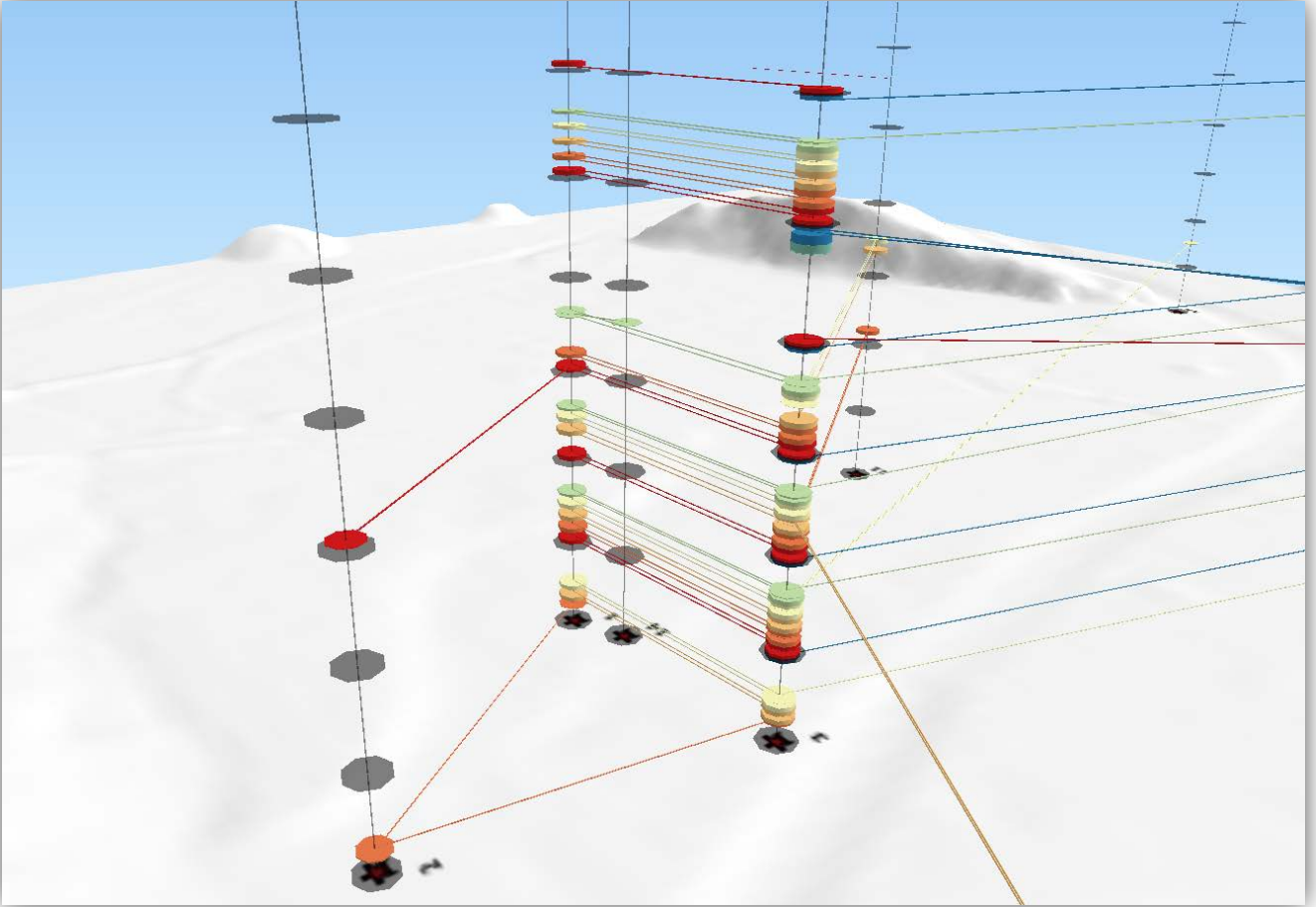


Figure 60. Combined visualization of transitions and stays with stacks of cylinders and pipes.

We can easily observe in Figure 60 the general patterns on the movement behaviour of the user. It is also possible differentiating at least two transitions every working day between *Home1* and *Work* for most of the weeks. Meanwhile, most of Fridays and Sundays there is a transition between *Home1* and *Home2* (green and blue pipes point to such place despite not visible in the figure). Presumably, the transition on Friday is towards *Home2*, whereas the return trip is represented by the Sunday pipes (blue). Another conclusion we can draw is that the user usually pass through *Home1* (tag 3) when coming from *Work* (tag 1) and before going to *Home2*.

Furthermore, we can appreciate again the already mention assistance to a full day training on weeks 3 and 4, out of the work place (tag 11 at the background of the figure). There is no transition from *Home1* to *Work* within such 3 days.

5.2.3.2. Future prediction

Prediction of the future stays and transitions could be the next step in further research to establish a presence probability model. The aims of this thesis are out of the objectives of such phase; however, an example of transitions prediction is presented in this section.

Transitions have been mined to extract the transitions between detected places for each weekday during different time intervals. ***The IDs of the transitions are different from those assigned in GTD.***

Table 28. Detected transitions of User1 during time intervals on Mondays

Transitions during time intervals each week day [MONDAY]																						
Detected Occurrences																						
Transit:	0-6	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	22	SUM				
01-01	0	0	1	0	0	0	0	0	0	0	0	2	1	0	0	0	0	4				
01-02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
01-03	0	0	0	4	0	0	0	0	0	0	0	0	0	1	0	0	0	5				
01-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
01-05	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
01-11	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	2				
01-13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
01-14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
01-23	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2				
01-31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
01-33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
02-03	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
03-01	0	0	0	0	0	0	0	0	0	0	1	4	0	0	0	0	0	5				
03-03	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	3				
03-10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
03-12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
04-05	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
05-01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
05-03	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
05-05	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
05-06	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
05-07	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
05-08	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
05-09	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
05-15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
05-18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
05-19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
05-34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
06-05	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
07-03	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
07-05	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
07-07	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
08-05	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
09-05	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
10-01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
11-01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
12-03	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
13-05	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
14-01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
15-16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
16-17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
17-05	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
18-05	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
19-05	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
19-20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
20-21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
21-19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
21-22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
22-21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
23-24	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	2				
24-24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
24-25	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	2				
24-26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
24-29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
25-24	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1				
25-25	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	2				
26-24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
26-27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
27-28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
28-24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
29-30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
30-01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
31-32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
32-01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
33-05	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
34-35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
35-36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
36-05	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
SUM	0	1	2	7	0	0	1	1	2	1	3	6	2	1	0	1	0	28				

The Table 28 shows all the transitions detected by the algorithm for User1 on Mondays. Most of the intervals represent 1 hour except the intervals 00:00-6:00, 20:00-21:59 and 22:00-23:59. Intervals cover from the time starting the interval until 1 millisecond before the time starting the following period.

In Table 29 are presented the relations between the detected occurrences of a transition during a time interval and the total transitions detected during such interval. For instance, the transition 03-01 represents *Work* to *Home1* in this case (just the opposite IDs to GTD). This transition has been detected 4 times between 16:00 and 16:59; this represents 65 % of the transitions performed by the user every Monday between 16:00 and 16:59.

This establishes a probability for the next transition according to the previously learned occurrences. For example, there is a 65 % probability that the next Monday between 16:00 and 16:59 *User1* will move from *Work* to *Home1*. Meanwhile, there is a probability of 57 % that in such weekday between 08:00 and 08:59 *User1* moves from *Home1* to *Work* (01-03).

The row PROP stores the proportion between all the transitions detected within a time interval on Mondays and all the transitions detected on such day. The column PROP show the proportion between the total detected occurrences of a transition on Monday and all the transitions detected on such day. For instance, the probability that the user moves between 08:00 and 08:59 next Monday is of 25 % whereas from 0:00 to 5:59 is 0 %.

Obviously the quality of such simple prediction would depend on the quality of the transitions extraction that the algorithm is able to perform.

Table 29. Proportion between detected and real transitions of User1 on Mondays

[illegible]

6. CONCLUSIONS AND OUTLOOK

6.1. Conclusions

The aim of this thesis was developing and evaluating a general approach suitable for movement behaviour analysis of a mobile element or user. Such approach had to determine the places visited by the user as well as characterise the stays performed at these places and the transitions between them. The detection of visited places was based on the clustering of GPS logs and three spatio-temporal sub-approaches had to be developed and evaluated to select the most adequate. The characterisation of stays and transitions had to include the extraction of them, the quality evaluation of such extraction and the analysis of the stays and transitions detected.

Three spatio-temporal sub-approaches have been proposed for the **detection of visited places**. These approaches have been evaluated under an existing common evaluation framework implemented with this thesis within a unified Java process. For **characterisation of stays and transitions**, the three algorithms have been evaluated and compared in terms of quality of the extraction. The stays and transitions extracted have been analysed regarding accuracy of the detection of real stays and transitions (tagged GTD). The research questions posed at the beginning can be now answered according to the implementation of the method carried out, the results obtained and the level of achievement of the targets established.

Regarding the first question, the spatio-temporal clustering approach most adequate for the automatic detection of a user's visited places is the “**Incremental approach**” (3.2.1.1), based on *Kang's* algorithm. Kang is able to achieve a maximum recall and precision of 80.6 % and 88 % respectively outperforming the results presented in (Montoliu et al. 2013). Nevertheless, **the most equilibrated clustering** offered an **F measure** of **74.3 %**, representing a **72.2 %** of **recall** and a **76.5 %** of **precision**. The corresponding parameter values for clustering are a **distance** of **53** meters and a **time** of **900** seconds. **DBSCAN** performs the second best clustering and *Ye* the worst in comparison with *Kang*. *Kang* and *Ye* are suitable for implementation within a mobile environment working continuously as new GPS data is collected. On the other hand, **DBSCAN** requires the whole dataset to produce clusters and the algorithm parameters are much more dependent on the amount of data and the user's mobility patterns. It also demands much higher computing resources.

Within the combined approach “**Incremental + density-based**” (3.2.1.2), the *Convex Hull* solution developed to complement the *Ye* algorithm worked well in comparison with the original use of *OPTICS* presented in (Ye et al. 2009). Nevertheless, the inclusion of additional parameters required by both options increased the difficulties on determining the optimum values for all the parameters and the general complexity of such approach. The “**Density-based approach**” (3.2.1.3), based on **DBSCAN** required the implementation of an additional process to extract temporal information related with the visits at places. This process added an extra time parameter which determines the minimum stay duration and one of the consequences is the dismissing of clusters representing places visited for periods shorter than this duration. Hence, there is a negative impact in the QE results of the clustering performance.

Regarding the second research question, an approach for characterising user's stays and transitions has been presented. Our **Incremental** sub-approach based on *Kang* was able to produce the best extraction of stays and transitions according to the quality evaluation performed. The best possible **stay** extraction reaches a detection of 65.5 % of the real time spent at places. Concerning the **stay extraction task**, the optimal clustering parameters values produce an F measure of 63.9 % with a recall of 60 % and a precision of 68.4 %. On the other hand, the best possible **transition** extraction reaches a detection of 54.5

% of the real time spent during transitions. Regarding the **transition extraction task**, is reached an F measure of 58.1 % which represents a recall of 56.8 % and a precision of 59.4 %.

Using different clustering parameter values, it is possible achieving slightly higher rates for the extraction of time in tagged stays (up to 65.5%). Nevertheless the clustering performed in these cases offers very low precisions as generating a high number of false positives. Hence, the clusters created are less realistic and will produce false stays and transitions and worst behaviour profiling. *Duration* and *weekday* of the stay is information basic for the characterization, whereas *duration*, *origin*, *destination* and *weekday* are fundamental to characterise the transitions. The *Euclidean* distance and the *speed* are useful to detect anomalous situations and errors in transitions. *Start* and *ending day* are relevant to detect anomalous stays and transitions.

The extraction of transitions performed by the algorithms has been relatively poor. The main reason identified is the inability to detect part of the stays at *sleeping places*. These are the places the user sleeps in and therefore, most of the GPS tracks should finish and start at them. In some cases the user forgets switching on the receiver or the device runs out of battery and then, some visits at places are not properly registered. Hence, some tracks finish at locations far from the starting point of the following track. If this happens at a sleeping place, such place is not detected by an incremental algorithm. *Kang* and *Ye* require a flow of parsed points close to each other by less than a distance threshold (*d* parameter). Our implementation of *DBSCAN* is also affected by this problem, because the complementary process for stays extraction (4.1.3) also requires a continuous flow of points. Thus, when a stay is not detected the previous and following transitions are merged as one and the origin, destination, distance, duration and speed are erroneous. Even short stays not detected create this conflict and the transitions related are not able to be matched with the real transitions.

Another pitfall related with GTD is the lack of detection of some tagged places. In a few cases, places visited by the users and tagged with ID and time information of the visits are impossible to detect due to GPS errors. In these cases the GPS points collected by the devices represent erroneous locations such that the clusters created by the algorithms do not represent the real locations visited by the user. Then, the quality evaluation process is not able to spatially assign these detected places to the real tagged places as not being located inside the 53 m circular buffers (2.2). Therefore, the performance of the clustering as well as the stays and transitions extraction is reduced.

A detailed analysis of the movement patterns of one user has been developed comparing the real behaviour with the detected by the best algorithm. Finally, two possible applications of the general approach presented in this thesis have been suggested; first, the user's movement behaviour visualisation and second, the future prediction of user movements.

The general approach presented in this thesis is suitable for movement behaviour analysis of a mobile element using GPS logs as input.

6.2. Outlook

Different changes could be implemented in the clustering approaches so as to improve the clustering performance and allow a new assessment of the different options.

The Incremental approach should consider the means of transport of the user so that the parameters are adjusted dynamically depending on the detected speed. As pointed out, the algorithm parameters have an important influence in the number and representativeness of the clusters generated. Moreover, QE results of clustering tests have shown that the performance of the detection depends on the length of the typical stay of each user which seems to vary according her specific movement behaviour. Hence, the time

parameter could be configurable by the user in case this approach would be implemented within a mobile application. Then, the algorithm could detect the stays at visited places better.

In order to improve the Incremental + Density-based approach, some modifications could be done in the incremental algorithm so as to enable a better detection of the sleeping places. The *Convex Hull* solution would require further testing in order to determine optimum values for the grouping radius and an improvement on the visited places detection. Both modifications could increase the stays and transitions extraction performance and accuracy.

The Density-based approach presented some drawbacks such as a minimum stay duration parameter, required to extract visits at the places. Further testing could determine a better value to increase the performance of the extraction. Additional checks by the algorithm considering consecutive revisiting could improve the accuracy of the stays extracted. Despite this approach requires the whole dataset for clustering and demands higher computing power, complementary solutions could be implemented for a mobile environment. For instance, client-server architectures could be used to reduce the computing requirements inside the mobile device. This obviously would require a good broadband speed for communication of data.

The general approach presented in this work should be tested with additional ground truth data. The data collection campaign could be specially designed to avoid the identified problems regarding GPS errors. Alternative and more realistic options could include further data pre-processing to avoid spatial discontinuity between consecutive tracks. Different radius for the circular buffer used in the QE influence the number of places detected as well as stays and transitions extracted. Hence, further testing is required to determine an optimum radius.

LITERATURE

- Ankerst, M. et al., 1999. Optics: Ordering points to identify the clustering structure. In *ACM Sigmod Record*, pp. 49–60. Available at: <http://dl.acm.org/citation.cfm?id=304187>.
- Ashbrook, D. & Starner, T., 2002. Learning significant locations and predicting user movement with GPS. *Proceedings. Sixth International Symposium on Wearable Computers*, pp.101–108. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1167224> [Accessed July 14, 2014].
- Ashbrook, D. & Starner, T., 2003. Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, 7(5), pp.275–286. Available at: <http://www.springerlink.com/content/t41dukuu8p2ulek9/> [Accessed July 22, 2014].
- Bicocchi, N. et al., 2008. Supporting location-aware services for mobile users with the whereabouts diary. In *Proceedings of the 1st international conference on MOBILE Wireless MiddleWARE, Operating Systems, and Applications*. Innsbruck, pp. 1–6. Available at: <http://portal.acm.org/citation.cfm?id=1361500>.
- Birant, D. & Kut, A., 2007. ST-DBSCAN: An algorithm for clustering spatial-temporal data. *Data and Knowledge Engineering*, 60(1), pp.208–221.
- Buchin, M. et al., 2011. Segmenting trajectories: A framework and algorithms using spatiotemporal criteria. *Journal of Spatial Information Science*, (3).
- Cao, X., Cong, G. & Jensen, C.S., 2010. Mining significant semantic locations from GPS data. *Proceedings of the VLDB Endowment*, 3(1-2), pp.1009–1020. Available at: <http://portal.acm.org/citation.cfm?id=1920841.1920968> [Accessed August 13, 2014].
- Castelli, G., Mamei, M. & Rosi, A., 2007. The Whereabouts Diary. In J. Hightower, B. Schiele, & T. Strang, eds. *Location- and Context-Awareness SE - 11*. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 175–192. Available at: http://dx.doi.org/10.1007/978-3-540-75160-1_11.
- Changqing, Z., Frankowski, D., et al., 2007. Discovering personally meaningful places. *ACM Transactions on Information Systems*, 25(3), p.12–es.
- Changqing, Z., Bhatnagar, N., et al., 2007. Mining personally important places from GPS tracks. In *Proceedings - International Conference on Data Engineering*. IEEE, pp. 517–526. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4401037> [Accessed August 13, 2014].
- Chen, L., Lv, M. & Chen, G., 2010. A system for destination and future route prediction based on trajectory mining. *Pervasive and Mobile Computing*, 6(6), pp.657–676. Available at: <http://dx.doi.org/10.1016/j.pmcj.2010.08.004>.
- Ester, M. et al., 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Second International Conference on Knowledge Discovery and Data Mining*, pp. 226–231. Available at: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.20.2930>.
- Fayyad, U., Piatetsky-Shapiro, G. & Smyth, P., 1996. Knowledge Discovery and Data Mining: Towards a Unifying Framework. In *Proc 2nd Int Conf on Knowledge Discovery and Data Mining Portland OR*, pp. 82–88.
- Griffin, T.W., 2012. GPS CaPPture: A System for GPS Trajectory Collection, Processing, and Destination Prediction.
- Gröchenig, S. & Hufnagl, M., 2015. *Pre-Processing of GPS Data. Internal report. SRFG unpublished research*,.
- Hightower, J., Consolvo, S. & LaMarca, A., 2005. Learning and recognizing the places we go. In M. Beigl et al., eds. *UbiComp 2005: Ubiquitous Computing*. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 159–176. Available at: http://dx.doi.org/10.1007/11551201_10.
- Hu, D.H. & Wang, C., 2007. GPS-based Location Extraction and Presence Management for Mobile Instant Messenger. *Embedded and ubiquitous computing*, (60533040), pp.309–320.
- Huang, C., 2012. Mining users behavior and environment for semantic place prediction. *Nokia Mobile Data Challenge*, 2012. Available at: [http://idb.csie.ncku.edu.tw/paper/conference/Mining Users's](http://idb.csie.ncku.edu.tw/paper/conference/Mining%20Users's)

- Kang, J.H. et al., 2005. Extracting places from traces of locations. *ACM SIGMOBILE Mobile Computing and Communications Review*, 9(3), p.58. Available at:
<http://portal.acm.org/citation.cfm?doid=1024733.1024748> [Accessed August 13, 2014].
- Krumm, J. & Rouhana, D., 2013. Placer: semantic place labels from diary data. *UbiComp*, (UbiComp), p.9. Available at: <http://dl.acm.org/citation.cfm?doid=2493432.2493504> [Accessed August 13, 2014].
- Laasonen, K., Raento, M. & Toivonen, H., 2004. Adaptive On-Device Location Recognition. In A. Ferscha & F. Mattern, eds. *2nd International Conference LNCS 3001, Springer Verlag*. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 287–304. Available at:
http://dx.doi.org/10.1007/978-3-540-24646-6_21.
- Leroy, G., 2011. *Designing User Studies in Informatics*, London: Springer London. Available at:
<http://link.springer.com/10.1007/978-0-85729-622-1>.
- Liao, L., Patterson, D.J., et al., 2007. Learning and inferring transportation routines. *Artificial Intelligence*, 171, pp.311–331.
- Liao, L., Fox, D. & Kautz, H., 2007. Extracting Places and Activities from GPS Traces Using Hierarchical Conditional Random Fields. *The International Journal of Robotics Research*, 26(1), pp.119–134. Available at: <http://ijr.sagepub.com/cgi/doi/10.1177/0278364907073775> [Accessed August 13, 2014].
- Macqueen, J., 1967. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1, pp.281–297. Available at:
http://books.google.com/books?hl=en&lr=&id=IC4Ku_7dBFUC&oi=fnd&pg=PA281&dq=SO+ME+METHODS+FOR+CLASSIFICATION+AND+ANALYSIS+OF+MULTIVARIATE+OBSERVATIONS&ots=nMYdB0OhuL&sig=N46jBcReO1y74cX1CtHG6qgiB8Q.
- Marmasse, N. & Schmandt, C., 2000. Location-aware information delivery with comMotion P. Thomas & H.-W. Gellersen, eds. *Handheld and Ubiquitous Computing*, 1927, pp.157–171. Available at:
http://dx.doi.org/10.1007/3-540-39959-3_12.
- Montoliu, R., Blom, J. & Gatica-Perez, D., 2013. Discovering places of interest in everyday life from smartphone data. *Multimedia Tools and Applications*, 62, pp.179–207.
- Montoliu, R. & Martínez-sotoca, J., 2012. Semantic place prediction by combining smart binary. *Mobile Data Challenge 2012 Workshop*, (McC), pp.1–6.
- Patterson, D.J. et al., 2003. Inferring High-Level Behavior from Low-Level Sensors A. Dey, A. Schmidt, & J. McCarthy, eds. *UbiComp 2003 Ubiquitous Computing*, 2864, pp.73–89. Available at:
http://dx.doi.org/10.1007/978-3-540-39653-6_6.
- Reddy, S. et al., 2008. Determining transportation mode on mobile phones. *Proceedings - International Symposium on Wearable Computers, ISWC*, pp.25–28.
- Shekhar, S., Zhang, P. & Huang, Y., 2003. Trends in Spatial Data Mining. *Science*, 7, pp.357–379. Available at:
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.77.1454&rep=rep1&type=pdf>.
- Venek, V. et al., 2015. *Presence Probability Model - Quality Evaluation. Internal report. SRFG unpublished research.*,
- Wolf, J., Guensler, R. & Bachman, W., 2001. Elimination of the Travel Diary: Experiment to Derive Trip Purpose from Global Positioning System Travel Data. *Transportation Research Record*, 1768(1), pp.125–134.
- Ye, Y. et al., 2009. Mining Individual Life Pattern Based on Location History. In *2009 Tenth International Conference on Mobile Data Management: Systems, Services and Middleware*. pp. 1–10. Available at:
<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5088915>.
- Zhang Xiao, Y., 2015. Data Mining Self Notes.
- Zheng, Y. et al., 2009. Mining interesting locations and travel sequences from GPS trajectories. *Proceedings of the 18th international conference on World wide web - WWW '09*, p.791. Available at:
<http://portal.acm.org/citation.cfm?doid=1526709.1526816>.

- Zheng, Y. et al., 2008. Understanding mobility based on GPS data. In *Proceedings of the 10th international conference on Ubiquitous computing - UbiComp '08*. UbiComp '08. New York, New York, USA, NY, USA: ACM Press, p. 312. Available at: <http://doi.acm.org/10.1145/1409635.1409677> [Accessed August 13, 2014].
- Zhou, C. et al., 2004. Discovering personal gazetteers. In *Proceedings of the 12th annual ACM international workshop on Geographic information systems - GIS '04*. p. 266. Available at: <http://dl.acm.org/citation.cfm?id=1032222.1032261>.
- Zhu, Y. et al., 2013. Feature engineering for semantic place prediction. *Pervasive and Mobile Computing*, 9(6), pp.772–783. Available at: <http://dx.doi.org/10.1016/j.pmcj.2013.07.004>.
- Zhu, Y., Sun, Y. & Wang, Y., 2012. Nokia Mobile Data Challenge : Predicting Semantic Place and Next Place via Mobile Data. In *Mobile Data Challenge by Nokia*. Newcastle, pp. 1–6.