Clustering spatio-temporal movement data

Amir Sarhangi

Nov, 2010

Clustering spatio-temporal movement data

by

Amir Sarhangi

Thesis submitted to the International Institute for Geo-information Science and Earth Observation in partial fulfilment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation, Specialisation: *Geoinformatics*.

Thesis Assessment Board

Thesis supervisors	Dr. O. Huisman, Dr. R. Zurita-Milla
Assessment Board Chair	Prof. Dr. M.J. Kraak (chair)
Thesis examiners	Dr. G. Andrienko

UNIVERSITY OF TWENTE.
FACULTY OF GEO-INFORMATION SCIENCE AND EARTH OBSERVATION

ENSCHEDE, THE NETHERLANDS

Disclaimer

This document describes work undertaken as part of a programme of study at the International Institute for Geo-information Science and Earth Observation (ITC). All views and opinions expressed therein remain the sole responsibility of the author, and do not necessarily represent those of the institute.

Abstract

Geographic Information (GI)community faced to the increase in availability of data during last decades due to advancement in the collection devices and also user requirements. Processing and extracting meaningful information from this large amount of data is the domain of KDD and currently Geovisual Analytics. Clustering in all these concepts plays an important role.

Spatio temporal movement data is one kind of data, in which interest is growing too fast. Clustering of spatio temporal movement data is quite a new area of research, which extends the existing clustering algorithms to appropriate use. OPTICS is a sophisticated algorithm which has been used mainly in trajectory clustering while the focuss of this research is mainly on point based clustering by the aid of 3D Euclidian distance function. In this work, original codes of OPTICS were extended to work in space-time, and implemented using the JAVA programming language. The algorithm tested over three human movement data sets: GPS, RFID and cell phone tracking data. The results are evaluated by visual interactive techniques and validity indices. In all data sets spatio-temporal clusters are achieved.

Keywords

Spatio-temporal data, clustering, Human movement, Geovisual analytics

Acknowledgements

Foremost, I would like to express my sincere gratitude to my supervisor Dr. Otto Huisman for the continuous support of my research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

I am heartily thankful to my second supervisor, Dr. Raul Zurita-Milla who guided and supported me in research skill class and during my thesis.

I owe my deepest gratitude to Ir. V. Bas Retsios who helped me too much. I really appreciate his knowledge and support during implementation of my thesis.

I would like to thank Dr. Ir. R.L.G. (Rob) Lemmens , Dr. C.P.J.M. Corne van Elzakker, Dr. M. Markus Gerke and Drs. M.T. Marga Koelen for helping me in the first days of my research.

It is a pleasure to thank JKIP coordinators Dr.B. Vosooghi and Mr. Huurneman because of their support during the course.

I would like to show my gratitude to Dr. Alimohamadi, Dr. Mesgari, Dr. Talei, Dr. Abkar and Ms. Silavi. I used their knowledge during my courses in KNT University.

I also wish to thank my family and relatives which supported me from first days of my studies here specially: Bashir, Azin, Nima, Arman, Sasan, Parya, Roza, Amin, Shirin and Navid...

Special thanks should be given to Narges. She showed me the prosperity. And my friends in Enschede, Amin, Nooshin, Majid, Nasrin, Babak, Aidin, Ario and Pooyan.

I would like to thank my JKIP classmates. We spent good days and difficult days with each other. I never forget them.

Last but not the least, I would like to thank my parents who thought me love and honesty. I dedicate this thesis to you.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this chapter the problem of huge amount of spatio-temporal data and role of clustering to deal with this problem will be discussed. Then aim of doing the research and issues relate to it will be identified. Related work of the research and methodology of doing it will be discussed too.

1.1 Motivation and problem statement

The availability of Geographic data has increased in many aspects over the last decades, in response to the needs of both public and private sectors. Various types of data are available. Perhaps the most common of these is GPS data, which is used in many applications such as tracking, positioning and navigation. More recently, new methods of data collection have allowed capture of cell phone data. These data types can be collected in large volumes, increasing numbers of location aware devices(e.g GPS, cell phones)and phones)and other tracking systems, opportunities to tracking people [39], animals or vehicles have grown rapidly and open new area to better understanding their behaviors [2]. These evolutions have enabled the study of more spatio temporal data.

Movement data as captured by GPS and other sensors is usually in the from x, y, t + attributes. This type of data is usually called 'trajectory' data, or geospatial lifelines. Geospatial lifeline is defined as a: "continuous set of positions occupied in space over some time period. Geospatial lifeline data consists of discrete space-time observations [...] describing an individual's location in geographic space at regular or irregular intervals" [35]. Another type of data which is available in large scale is cell phone tracking data. As Ahas, et al. mentioned, there is a great potential for using mobile positioning data in space-time behavior studies of tourism [1]. Here we could see the possibility of using large amount of passive mobile positioning in order to analysis of spatio-temporal events.

These large amounts of data need to be processed by new tools in order to extract useful information. Knowledge discovery in databases (KDD) has developed as a response to this problem [28].

KDD comprises five main steps: data selection and pre-processing, data en-

richment, data reduction and projection, data mining and finally pattern interpretation and recoding. One of the techniques which is used in the data mining level of KDD is clustering. Harvey et al. [28] define clustering as "groups of spatial object such that objects in the same group are similar and objects in different group are unlike each other". Clustering is used to extract meaningful information in the form of generic aggregate patterns(such as grouping and convergence) from large data set. This allows an analyst to work with a group of objects rather than individuals [45]. For example: in preceding cell phone tracking data set, clustering could be used to find popular visitor area like fishing areas at the weekend.

Due to the complexity of spatio temporal in comparison to spatial data, It is more complicated to do clustering in this field because conventional methods only focus on spatial part of data and don't consider time as an additional dimension. In addition adding one dimension with different nature to the location space makes clustering harder. Different methods addressed this issue in the literature. Relative Motion, Flocking, progressing clustering and interactive cluster analysis are some methods which try to solve this problem. In the following sections these methods and other methods will be discussed more.

The main problem identified here is which type of existing clustering methods are suitable to be extended for different types of human movement data, as different types of data have different resolution, scales, and uncertainty aspects. These issues are currently not well understood. By implementing the clustering algorithm and testing over different dataset, we could also evaluate the algorithm and suitability and possibly fitness for specific applications.

1.2 Research identification

The main aim of this research is to implement and test a space-time clustering algorithm in the area of spatio-temporal human movement data. This is to be achieved by reviewing existing clustering algorithms several reference datasets, and the results evaluated to help us understand the strength and weakness of the algorithm for different types of movement data. This could help one who wants to use this algorithm in an application involving KDD and data mining.

1.2.1 Research objectives

The goal this research will be dievided in four main objectives:

1. To review existing clustering algorithms and their application to human movement data
2. To find suitable clustering algorithm that could be extended to cope with spatio temporal data.
3. To find the suitable pseudo code for the algorithm(s) and implement them using different types of movement datasets for testing and evaluation

4. To critically evaluate the results by identifying issues and limitations in the implementation, and provide an agenda for future research

These objectives try to define the steps toward solving the identified problem.

1.2.2 Research questions

Toward reviewed research objectives some questions should be considered:

1. Which types of movement data exist ?
2. What are the main approaches used in relation to clustering of movement data and how can we categorize them?
3. What are the most widely used opensource clustering algorithms for human (point-based) movement data?
4. Which aspects of clustering algorithm should be considered to extend it to space-time clustering?
5. How can a space-time clustering algorithm deal with different characteristics of human movement data, such as scale, resolution, ?
6. How can we evaluate the results of clustering algorithm?

Answering these questions help us to reach the main goal.

1.2.3 Innovation aimed at

The novelty of this research is the implementation and evaluation of a spatio temporal clustering algorithm in uDig, based on an existing purely spatial algorithm. Open-source implementations of spatio-temporal clustering algorithms do not currently exist in the literature. By providing open source implementation this research provides a good framework to further expansion and development of spatio temporal clustering and an opportunity to better study and test of clustering algorithm over real data set(s).

1.2.4 Related work

There are quite a few works that relate to trajectory and its characteristics in the field of clustering [34]. In the case of generalizing movement, perhaps the most obvious work has been done by Laube et al. as relative motion(REMO). This work is based on analyzing groups of Moving Point Objects(MPO). It is based on two main properties: first defining REMO matrix which reflects the properties of geospatial life lines, for example azimuth, speed and acceleration. Second finding patterns in the matrix. This algorithm can find patterns like track, flock, leadership and also convergence and encounter. In this work, characteristics of movement objects like speed and azimuth are used to detect trajectories with the same characteristics as a*flock*. One of the main problems with this approach is that the algorithm just finds flocks in one time instance. [8]

Wachowicz et al. provide a new definition of the flock concept. According to this definition a set of parameters was defined as:*min-points, radios, min-time-slices* and *sampling rate*. The main assumption is that spatio temporal coherence exists among members of flock. Then by four steps flocks are detected by the algorithm. The first step is synchronization of samples in regular/irregular time steps. Because spatio-temporal data have different resolutions, they should be resampled to an equivalent reference. After this the location of sample points is calculated using nearest neighbor computation. In this step one trajectory considered as the base and by using predefined radius, the proximity to this base flock is calculated. By applying membership persistence analysis, flocks are detected. Each base and related trajectory which satisfies the proximity condition in two time instances consider as a base flock. The final step is membership-based pruning. In this step the extent of flock is computed and if It's less than user defined radius omit and do not consider as flock. This happens in the case of stationary flocks. In this work, the authors differentiate a stationary flock (e.g. people waiting in the bus station) from moving flock, disregarding the first and focusing only on moving flocks [8].

Other works have been done by Rinzivillo et al. and Andrienko et al. for clustering spatio-temporal data by using OPTICS [5], which belongs to the DBSCAN family [16]. Andrienko et al. [3] show that the algorithm could be applied to different types of data as: trajectories, events and spatio temporal distribution. The algorithm has been tested over two datasets . Similar approach by the same authors was proposed in progressive clustering. In this work, a simple distance function is applied to a big data set. Then another distance function is applied to remaining data set, and so on. Using this approach a quick division of large data set is possible and further work on interested data set is possible. But this approach requires the experience and knowledge of the operator or analyst because in each progressive step the a suitable distance function is needed. [45]

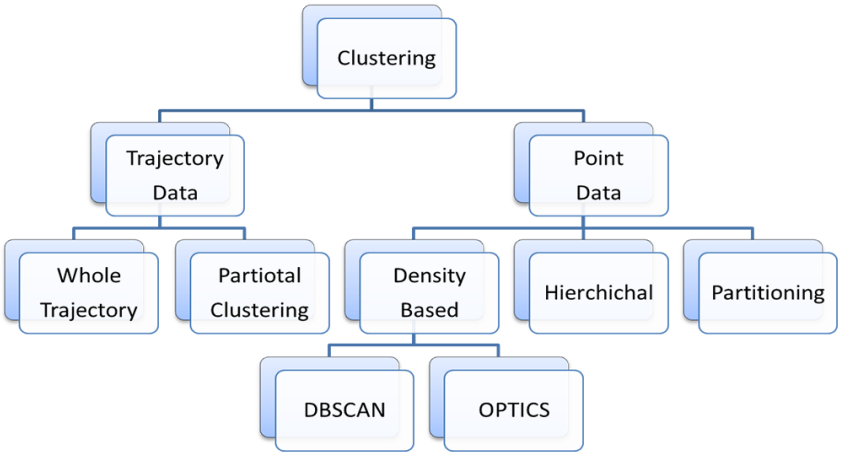

Figure 1.1: Categorization of different clustering method based on (28).

As shown in Figure 1.1, the strategies of finding cluster in point data can be categorized in three groups: *partition based, hierarchical*, and *density based* [31]. In the partition based approach, the data set is divided into the sub-data sets in a way that the objective function is optimized in each cluster . Then this procedure is iterated to reach the optimal solution for example in K-means method first random number of points are chosen as cluster centers which are equal to the number of clusters (defined by user). Then algorithm starts to assign points to cluster based on distance. For each cluster mean value are measured to find new center. The procedure iterates to find optimum clusters [28]. The CLARAN method [41] and SOM [32] follow this approach. The hierarchical approach uses a dendrogram to divide data set - like a tree in which clusters are shown as nodes and leaves are single objects. Finally, density-based methods relay on density notation [45]. Clustering in trajectory data we could consider two main methodologies, first whole trajectory which define clusters if the similarity is available in all length of trajectories, second partial clustering that divides cluster first in some parts, then try to find similarity between these parts. As shown in Figure 1.2, whole trajectory methods like OPTICS are failed to find cluster in four trajectories but partial clustering could find cluster in similar part of trajectories.

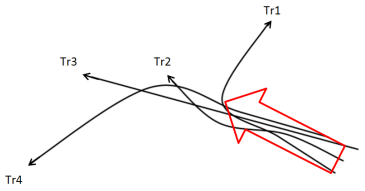

Figure 1.2: Partial clustering could find clusters in the parts of whole trajectories (28).

1.3 Methodology

The research will start with the literature review, to gather different algorithms and methodologies about clustering. Many algorithms exist(see Figure 1.1) . We should consider there are a lot of clustering methods which are used in other disciplines (like clustering based on spectral reflectance in remote sensing). On the other hand those methods may not be suitable for our propose which is dealing with time and location. Again there are the methods used in saptio temporal data but our concern here is human movement. These methods should be suitable for point data or trajectory data.

The objective here is to categorize them based on characteristics of data and possibly application. So we have to define the specific characteristic used as the basis for comparison. Of course among methodologies and algorithms which will be reviewed, some may be rejected in the first step. The main concern here is with human movement, therefore some of the algorithms that are not applicable to issue may be removed from the analysis. Also it is possible to use some idea in another scope (for example algorithms designed to work on raster or image data), to explore related methods.

Next step will be gathering information about different types of data sets used. It demands to have a short review on characteristics of these data. Different data has different characteristics for example GPS(Global Positioning System) data can be collected in different time frequency like 1sec, 20 sec, etc. sometimes the very good spatial accuracy is in demand then differential methods are employed. For cell phone tracking data, data resolution depends on call activity and this causes the variable temporal resolution.

In order to better test and evaluation the implementation of clustering algorithm, real data sets are needed. So at this stage, the algorithm and data sets will be available. The algorithm will be tested over different data sets. First, it will require implementation of the algorithm that will be tested. May be, It will be needed to access to pseudo codes, as these are not always published. In such cases, communication to authors will be essential to access the code. If I want to make my this part of methodology more clearer to see a good picture, I could list the steps as:

1. Identify algorithms suitable for human movement data and review them
2. Determine the most suitable ones(3min-5max) and select one of them
3. Find the relevant pseudo codes for the selected set
4. Implement them in Java/uDig
5. Experiment with clustering
6. Evaluation of clustering results

There are different algorithms available in the literature that are used for movement data. Some of them are used to analysis data, some of them aggregate and also there are algorithms to simplify and determine patterns in movement data . Therefore evaluation all of them are impossible because of time limitation. The selected set of algorithms will chosen based on some criteria like popularity , citation in the journals, novelty, access to pseudo codes and more importantly applicable for human movement data. Based on this It is expected to implement one algorithm and test it.

Having algorithm in the uDig it is possible to test the algorithm and see the results for different data characteristics. This includes:

- Spatial scale, e.g. what dimension of area means the extent of X and Y plane
- Temporal scale, e.g. time extent, one day, one week, etc.
- Resolution, e.g. the number of points acquired in time and location, granularity and
- Uncertainty , e.g. the tolerance X and Y in the plane

In an ideal case, data will input in the algorithm, algorithm run and results compared with respect to a 'reference model' (such as reality if available, or a

simplified reality). For example, in some data sets we know the relationship between members, the spatial resolution, temporal resolution and uncertainty. This will be the basis to compare, since it is unlikely that any reference datasets will be available for evaluating results. If there is no reliable data, a reference data set will be collected by using GPS. A group of between 10 and 30 people like students will be used to collect movement data. Predefined clusters will be produced in order to test the algorithm. These data can be used as control points. Also synthetic data could be used. Other datasets that may be used in the research include an RFID(Radio Frequency IDentification) evacuation trace dataset from the 2008 VAST challenge [22], a large wireless motion dataset from the InfoVis 2007 contest, and it is hoped a cellphone tracking dataset will be available to analyze. Different characteristics of these data sets are important for example GPS data are precise and cell phone tracking data are usually available in large scale while RFID data sets may use in indoor tracking individuals like in a mall.

Evaluation of results will be an important part of the research. It's could be in a comparative manner for example compare the parameters, number of clusters and compactness of each cluster. This evaluation must critically look at input, output and process for the evaluation of the results. The idea here is to try to relate results to the data characteristics and parameter setting. In this level visualization will be a helpful and valuable tool for interpretation. Visualization and interactive methods in clustering often is a part of algorithms [12] [19]. Table 1.1 shows possible algorithms and also different data sets. Another issue in evaluation process could be performance of algorithm for example computation time, and also size of datasets (for example what is processing time for 10 records or 1000 records?).

Table 1.1: The schematic form of representation of evaluation possible algorithms over data sets.

	Data sets		
Algorithms	GPS	RFID	CELL Phone Tracking
DBSCAN(Ester et al. [16])			
OPTICS(Ankerst et al. [5])			
REMO (Laube et al. [18])			
Progressive clustering (Rinzivillo et al. [45])			
Intractive clustering (Andrienko et al. [3])			
...			

First algorithm will be formalized in a logic language and then programmed into software. It is expected to use uDig open source GIS for implementation of the algorithm, and analysis, and visualization of results. The reasons for choosing this software are open source software, existing experiment in spatio temporal data and its ability to show clusters in Space-Time Cube [36]. Once implemented, we can test each of the parameters of the algorithms systematically To implement the selected algorithms in uDig, programming in Java will be required.

In order to visualize the results, the Space-Time Cube will be used. At the end the final result will show in a table which shows the properties and parameters of algorithm and also data. Evaluation of each parameter over algorithm, data set will show in the corresponded place. Each place may contain information about results of different parameters over different data sets, this could also contains size of clusters, compactness of cluster, specific characteristics of algorithms over specific dataset.

In the following chapters related subjects to the described problem, and objectives will be discussed as follows:

In Chapter 2 clustering in different disciplines including human movement data will be reviewed. Comparison, categorization and application of clustering algorithms will be considered there.

In Chapter 3 a density based clustering will be discussed more. OPTICS clustering will be discussed in details including pseudo codes, definitions and current applications.

In Chapter 4 the described OPTICS pseudo codes in the third chapter will be extended and implemented in uDig open source GIS.

In Chapter 5 the implanted algorithm will be tested over three different data sets including: GPS, RFID and cell phone tracking data set. The result also will be evaluated by visual interactive method and validity indices.

In Chapter 6 research will be summarized and dissuaded. Further work will be suggested also.

Chapter 2

Literature review

In this chapter after reviewing the clustering definition, application in different disciplines including data mining, KDD and Geovisual Analytics will be discussed. Then clustering algorithms will be categorized based on general view and human movement data. Issues relate to human movement data clustering include collecting method, available type of data, visualization, generalization of movement data and flocking pattern will be discussed in the final part of this chapter.

2.1 Clustering

2.1.1 Definition

The story of clusters started at beginning of the universe, when small particles formed clusters and eventually form different planets and stars also our world. This concept continued over time and forms part of every phenomenon. Particles try to connect to others and unify. Clustering is one of the basic and important matters in many disciplines. Including mathematics, philosophy, biology, astronomy, etc. Therefore, many definitions are available that try to explain clustering in their views, but they all identify that a fundamental part of clustering is the similarity. These are some available definitions exist in literature:"clustering as process of grouping set of physical or abstract object into class of similar object" [28]. The same definition is provided by Rinzillo et al., but instead of similarity talks about low variability that is same meaning of similarity and high variability that is dissimilarity. Based on this definition cluster is a group of object with low variability among members of cluster and high variability among different clusters [45]. As could be seen from these definitions and also other definitions, similarity is an important factor in the process of clustering. In fact clustering is nothing more than grouping similar things. The question which comes to mind is, why clustering? to answer this question some hints are considerable:

1. It's useful for many area to consider group of objects instead of individuals this could help to have an overview on all data and also easily working for analyst [3]

2. Finding opposite side of similarity means dissimilarity helps us to find peculiar and uncommon things

3. Gain knowledge about data, as an elementary process to more processes [45]

As could be seen from the definition there are broad range of application which could be involved by cluster analysis. In the next section some of them will be reviewed.

2.2 Application in different disciplines

Clustering is a "multidisciplinary activity". In a data perspective, it could be about graphs, data mining and data analysis, or when ones try to structure something by clustering, it refers to cognitive science. Clustering may be used to extract knowledge about something then machine learning comes to mind. Hence clustering covers quite broad activities and disciplines and also could be seen in different views like statistics' perspective, machine learning, classification and finally a data mining perspective in relation of large amounts of data to gain insight of structure of data or as a pre-processing step before further analysis [29].

To categorize these broad area of disciplines in application Jain.et al [29] provide four main categories:

1. Image segmentation

2. Object recognition

3. Information retrieval

4. Data mining

One of the areas which is used broadly in computer vision is image segmentation. This method has been used for many years and still are in use. The procedure of image segmentation is provided here [46]:

1. Measurements of image and project to feature space

2. Clustering those projected feature, the segments will formed

3. Image segments defined as a collection of "homogenous features" in image,respect to predefined properties like color gray level texture etc.

Application area again is quite broad include clustering used to segmentation of gray level [46]. For example, reading barcode in shops. This is possible by using thresholding clustering. Analysis of texture is another area of interest. Generally speaking by segmentation images, we gain less information in comparison the original one but motivated part here is finding relevant information [42].

Segmentation approach is used also in another area like finding pattern and model prediction. Analyst test hypothesis to find the pattern and predict the pattern. To do this first clustering could be used to group objects and then

find predict the pattern on groups. Visualization is important part of this work to see data in different level [29].

New challenges are multidimensional clustering and semantics in clustering. In object recognition, one of the applications of clustering is to view 3D objects and recognize them. A 3D object has many views, so to detection is possible by indexing in subset view of the object which means a subset of many different views available. Clustering is used to finding these subsets. For example, in hand writing detection. [29].

Information retrieval is used to automatically find information. This has a broad application in libraries. For example, bibliography assigns a combination of letter and number to categorize a book. The problem is to find the exactly relevant books. One of the solutions is classification of these assigned labels [29].

Data mining, increasing data recent years it is essential to find information among these large amounts of data again clustering gives the possibility to assist in this issue. The following part will discuss these issues more in detail.

2.3 Clustering and classification

Clustering is different from classification in terms of labeling clusters. Classification is a costly process, because providing training samples is costly [28]. Therefore, to have a suitable classification it's better to cluster data first and then label them [27]. Clustering compared to classification is an unsupervised process, meaning that the number of cluster is not determined in advance. The problem of an unsupervised process is in the evaluation of results. In contrast to supervised classification that uses independent training samples-sample not used in classification of data set- to control the results, in unsupervised clustering this possibility is not available. Therefore, another solution like validity indices is applicable. These indices use to evaluate the result of clustering in terms of quality and quantity of output clusters. [26] Quality of clustering means how good the cluster is produced. For measuring the cluster quality two main similarity conditions were introduced in the literacture [27]:

1. Intra-class: The relation(similarity) between the members of cluster.
2. Inter-class:The relation(similarity) between the clusters to each other.

To be a good cluster intra-class similarity should be high and inter-cluster similarity should be low. The quality measurement depends on method and also implementation which makes hard to measure the quality. Sometimes quality is measured by ability to detect patterns in data. Pattern is higher level of information and usually hidden in dataset. Data mining and its techniques(Clustering, classification, etc) try to find pattern in data. (More discussion available in section 3) Measuring the similarity is possible by distance function and separate function measures the quality and goodness of a cluster. This function is called quality function [27].

Classification is an effective process of dividing data set into classes and labeling them [27]. Another definition provided by Jain et al. has emphasized on *patterns* "clustering is an unsupervised classification of patterns (observations,

data items or feature space)into groups(clusters)" [29].In classification generally the problem is to assign a label to a new object, but in data clustering tries to group objects in the natural way.

Two main approaches in data clustering are *parametric* and *non-parametric*. Parametric approach uses predefined density functions to make the result optional. In non parametric there is no priori information [53] Spatio temporal movement data has random characteristics and usually incomplete observations(missing values) [45] therefore for clustering this kind of data (our case) we follow the second approach because there is not any about underlaying structure of data.

Groth et al. have emphasized classification as supervised *learning* and clustering as unsupervised *learning*. Clustering in this view is a method to finding similar trends. Classification is having a goal in mind, then profiling data and get more knowledge about data based on predefined profiles. In clustering there is no profiling data [23].

To better understand the differences between classification and clustering it's better to have a look at definitions of clustering in image processing to generalize them. Digital image classification assigns each pixel to class(s). This kind of taxonomy or classification has an important role in image analysis, since the results could be used directly in decision making.

In this perspective there are three main approaches in classification [1]:

- Supervised
- Unsupervised
- Hybrid

The differences is in using training samples. Supervised classification uses training samples, unsupervised doesn't use training samples and hybrid clustering is in between these two categories.

Unsupervised clustering means assign data(here pixel)to natural groups that exit in structure of data. Some advantages include:

- No need for prior knowledge about data, just apply algorithm to the data set and examine the result.
- Decrease human error. Generally a source of human error is in the procedure of data collection for data sampling is removed.
- Ability to find outliers. Because of to the natural structure of data

There are disadvantages for this approach-unsupervised clustering- of clustering like maybe found groups don't match to the desired groups of analysis. One solution to this limitation is using interactive methods and play with parameters until match the result.

Usually these steps are performed in most unsupervised classifications: first the analyst determines the number of groups based on knowledge (in some

[1] the rest of this section is summarized from [42]

Table 2.1: Clustering vs. Classification

Properties	Classification	Clustering
Process	Supervised	Unsupervised
Grouping	Force to go in classes	Natural structure
Output	Is known	Unknown
Quality	Known	No general method

method this step is not the case) then the center of clusters is found and the object assigned to the clusters. Then again, the centers of each cluster are formed and the process is repeated until the optimal clusters are formed. Within the performance, the analyst has a small effect on process and his/her job is to define constraints in beginning and also choosing appropriate algorithm.

Classification(supervised) approach has basic factors like effective approach to finding the distance measurement in space, finding content of classes and test division of classes. One of the advantages of this approach is giving the analyst a chance to determine output classes to match exactly his/her needs. At the same time it could be a disadvantage, because analyst forces data set to categorize in groups which maybe do not follow the natural structure of the data. Another disadvantage as mentioned before is providing training sample data that is a costly process [42]. Table 2.1 is shown the summarized comparison of clustering and classification.

2.4 Clustering in data mining

Nowadays, we are facing an increasing amount of available data day by day or even second by second. This bulk of data motivates scientists to extract meaningful information to use in different disciplines. Knowledge Discovery Databases (KDD) were developed as a response to this issue. KDD attempts to find pattern among large data bases, which is different from simple query and traditional analysis, which don't penetrate in deep level of data. KDD, in fact, attempts to find whatever we cannot find by ordinary approaches, which are mainly costly in analyzing the large amount of data. KDD searches to find patterns in order to extract hidden information in the large amount of data. This is the similar definition of data mining: Data mining known as a solution to extract information from data in the form of patterns. In fact, data mining is one of the main levels of KDD towards finding interesting patterns. Extracting information in higher level is knowledge. Higher level means "idea and beliefs about the domain" [28]. The difference between KDD and data mining is in extraction level of information. KDD attempts to find knowledge from information extracted by data mining. KDD consists of five main steps [17]:

1. Data selection:subset of data in order to knowledge discovery
2. Pre-processing:find error, incomplete, noise and finding strategy to cleaning data

3. Transformation:finding effective representation of data.
4. Data mining:finding information from low level data
5. Interpretation and evaluation:of results of data mining and representation of them.

Figure 2.1 shows the KDD steps according to above definitions.

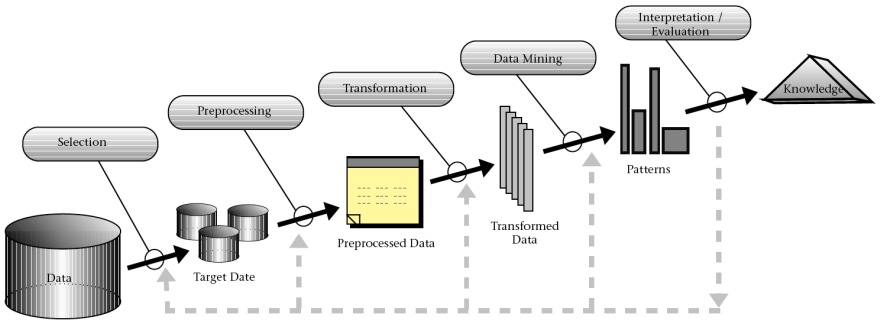

Figure 2.1: KDD process and data mining (17).

Data mining and clustering have a strong relationship to each other. Clustering is used in preprocessing step or even stand alone step towards finding pattern. It is used in segmentation, visualization and model prediction approaches in data mining. Mirkin et el. [38] provide a definition for clustering in data mining as"Clustering is a discipline devoted to reviling and describing cluster structure in data sets." The main concepts of this definition include: Data means "stored or recorded information" like data entries and attributes, Structure for clusters could be single, partition or hieratically and finally cluster description means"understanding and prediction clusters" [38]. In data mining prospective there is five steps to do clustering:

1. Defining problem and choosing data set. There is different data sets. First of all it should be decide to select which dataset is appropriate or even which subset of data set is important.
2. Preprocessing data: there is always typical problem in data sets like incomplete data records, or missing, values or unreliable data.
3. Finding clusters: using clustering algorithm to find clusters.
4. Interpretation of clusters to see how much tailor to application and relevant requirement and goals
5. Conclusion: sum up all together to have summary of pervious steps and come up to the solution relate to defined problem.

In data mining view, it doesn't matter what is the source of data and which method used to collect data. What is important here is data available with some properties. Data mining views data as a parcel: all have to be done is to extract information inside the parcel include patterns and regulations based on available properties. Clustering as a technique of data mining does an important job in this way [38].

Clustering in data mining covers quite broad area of application. Below some application area are reviewed.

1. A major area of data mining which uses clustering is world wide web. This type of data is named as unstructured data. Clustering use to facilitate search engines and grouping relevant words [38].

2. Another area of application is geological data mining. Clustering as technique of data mining uses to find oil well. The price of finding these oil are high an chance is low (1 from 10). There is large data of capable place available. By data mining and clustering it is possible to find oil well [29].

In pervious years, much work has been done for analyzing data. These attempts were whether completely visualization or completely automatic. Usually we are facing with the problem that is complicated and human intelligence need to be in the analysis process. Computers in another hand have an important role in analyzing data. As discussed before KDD attempts to analyze the large amount of data to *automatically* extract information(Knowledge). This causes many problems as KDD doesn't consider human expert in the process and analysis. For this reason KDD is called "black-box" because the user doesn't know the process of extracting knowledge [13].

On another hand visualization uses human "background knowledge, creativity and intention " to visualize data, results and process. The problem here is visualization methods are good to deal with small data set and fail in the large amount of data [13]. This condition gets worse in dealing to one kind of spatio-temporal movement data named trajectory. Trajectory data consists of location(x,y) and time(t). Visualizing this kind of data is mainly in Space Time Cube, which is a 3D space consists of 3 orthogonal axis [36]. Two horizontal axis are x and y or location and vertical axe is time. Visualizing large number of trajectories in the space time cube is hard to interpret and understand. Therefore, a solution for these two limitations will be Geovisual Analytics.

2.4.1 Geovisual Analytics

Geoisual Analytics approaches try to solve reviewed shortcomings in KDD and visualization and help decision making in many area of application like bioinformatic, climate change and pattern identification to name some [13]. As Bertini and Lalanne mentioned , approaches exist in this context in the literature could be projected on a line from pure data mining to pure visualization [10] at the other side For a method of data mining to be apply in Geovisual analytics there is some criterion [13]:

1. It should be fast, to answer in less than second

2. Understanding parameter by visualization
3. Interactive control of parameters: It gives the analyst facility to see the impact of changing parameter on results.

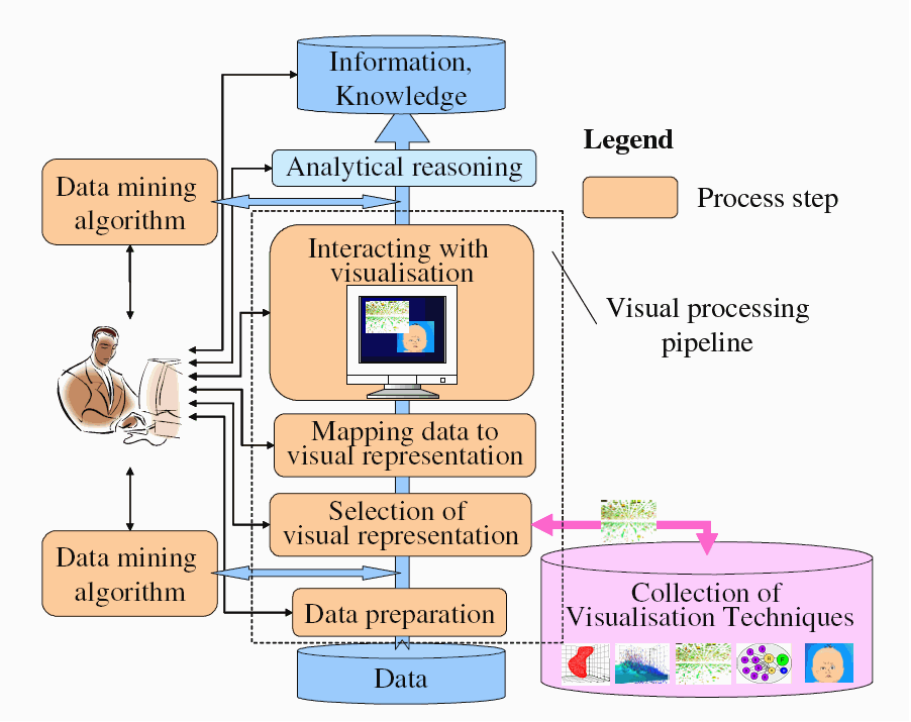

Figure 2.2: Geovisual Analytics (48).

In fact Geoisual Analytics collaborates strength of many disciplines like visualization , KDD, data mining and etc.(Figure 2.2) At this moment many scientists are working on this approach as a big project. [13]

2.4.2 Clustering in geographical data

Geographic data mining approach is used for finding interesting structures hidden in the large amounts of spatial data. Here the spatial dimension is major forms. Geographic analysis includes statistics, finding pattern, etc. In this situation clustering or cluster analysis would be a function that geographic community uses to analyze and mine interesting information and in higher level knowledge from data. For example, finding outliers to detect terrorist among a crowd, producing thematic maps by clustering feature space in remote sensing images [44], find a group of customers in marketing, cluster the same land use in the earth observation database, group of houses based on some criteria like location in city planning disciplines and finally earth quake study by clustering high risk centers [27].

In geographical data mining view data sets and acquisition of is important NOT because the method used to collect data or instrument or technique,

but because of characteristic that is inherent in the nature of data collection method. As discussed before data mining sees data what it is, as a parcel and doesn't care about where it comes from. To clarify this issue let's have a look at some kind of data collection methods and data sets available.

2.4.3 Collecting methods

One of the main methods of collection that has been used broadly is the global positioning system(GPS). This kind of data available in all over the world as long as the receiver could receive pseudo codes from at least three satellites simultaneously. GPS data generally are accurate. This intrinsic characteristic makes it possible to find tighter cluster. Cell phone tracking data are not as accurate as GPS data but available almost everywhere in large scale. This issue will be discussed more in following chapters. All in all, these methods of collection provide geographical data sets. We are trying to find behavior of these data. To finding micro-scale behaviors it is important to have higher resolution data and vice versa. Application plays a key role, means which area would be used to data mining. For example, consider a country wants to find the places that more attract tourists in order to provide facilities to better services. Here the problem is in macro-scale and low resolution data like cell phone tracking data is useful and sufficient. Consider in the mentioned situation a special company wants to show a park to the tourists. In other to avoid getting lost the tourist give them a PDA that uses global positioning system and a map to mark interesting places. Clearly, this is an example of micro-scale behaviors and of course high resolution data-GPS-is needed. Even in micro-scale behavior study, higher resolution data is needed when a group of tourists go to a museum and the accurate place of stuff is in demand. Perhaps RFID data here could be a solution to monitor the viewer.

Geographic data consists of data in different forms like raster data, satellite images, vector data like a map of a city, point data like position of nodes in a geodetic network and trajectories position of point data in time. One of the challenging areas of research in geographical data is dealing with movement data. This movement data could any kind of data that change the position in time like a pedestrian walks in pavement, a car in a highway, growing a city or moving an iceberg. Human movement data could be categorized in point data or trajectory data. This categorization will be discussed more in following sections.

"Spatial clustering is an important part spatial data mining" [27]. There is the difference between spatial data mining and data mining because of differences between data and spatial data. Spatial data has unique characteristics like effect each other and has relationship with each other. For example, attributes of near spatial locations most of the time show similarity, or maybe affect each other. A variety of methods for clustering data used in data mining have been developed. Methods have been used in different application and need different parameters such as the number of clusters. The differences between them may be so high that two methods for the same goal find different clusters, or even totally opposite cluster results.

2.5 Categorization of clustering algorithms

In this section and next section different categorization of clustering algorithms will be reviewed:gloabal categorization and geographical categorization.

2.5.1 Categorization of global clustering algorithms

According to taxonomy provided by Jain et al. clustering algorithms as shown in Figure2.3 divides to two major parts:partitional and hieratical.

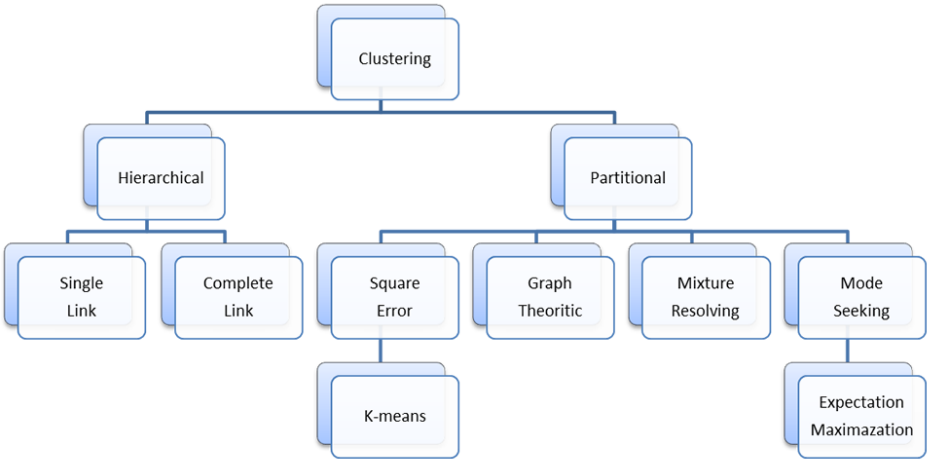

Figure 2.3: Categorization of clustering algorithms based on (29).

To describing this kind of taxonomy these comparisons should be done(summarized from Jain et al [29] :

- Agglomerative and divisive: agglomerative means first finding clusters base on pattern then merge clusters until satisfy the criterion to stop. Divisive approach consider whole as a cluster and then split to clusters.
- Monothetic and polythetic: Whether all features are considered simultaneously in clustering or in a defined order. Based on these more algorithms enter all features to computation distance simultaneously. This is called polythetic. In monothetic approach sequential clustering applies ordering to do clustering. Figure 2.4 shows the monothetic approach. First (x1) vertical line is drawn and then (v) data divided in two groups. The next level would be horizontal lines. H1 and H2 divide clusters made in previous step by vertical line. The major problem here is to operate on large data set and multi dimension data.

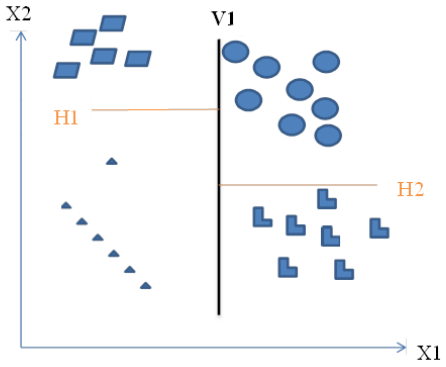

Figure 2.4: Monothetic clustering based on (29).

- Hard or fuzzy: Clustering, like classification, tries to assign each member of data set to divisions(or classes). In hard clustering, there is always a logic relationship-0 or 1- to each member. Therefore, the member of cluster(A)cannot a member of any other cluster(s). In the other hand fuzzy clustering assigns degrees of membership to a member of a class. Cluster will be formed by the higher amount of membership in each class, so a member of each cluster could be a member of another class based on the degree of membership. Advantages of this type of clustering are nearness to reality and also easily could convert to hard clustering by assigning maximum membership to each cluster.

- Incremental and non incremental: When deal on large data set, for example, cell phone data, constraints like memory and time play a main role. Therefore, clustering algorithms should be compatible to these issues. One way is by reducing the number of times that data scanned. This is called an incremental approach.

Hierarchical Clustering leads to structured form of data named dendrogram. Based on existing criteria hieratically clustering divides in tow main parts single link and complete link. Difference between these two is in the way they measure similarity and divide clusters. In single link minimum distance between a pair of pint in two clusters measures while in complete link the farthest distance is measured. These distances provide criteria to divide clusters from each other. Complete link produce more compress clusters rather than the single link.

Partitional clustering divides space to partitions and makes clusters in each partition. It doesn't lead to structured clusters like a dendrogram. These algorithms are suitable for clustering large amounts of data. The must famous

function in this category is K-means, which is simplest form of squared error algorithms. K- means is very popular because it's simple and also run in a short time. The major problem is choosing a start point because the algorithm is sensitive to this. For example the Figure 2.5 shows the impact of choosing the different start points on clustering the results. Ellipse clusters belong to choosing A,B and C as start points, and rectangle clusters belong to choosing A,D and F as start points.

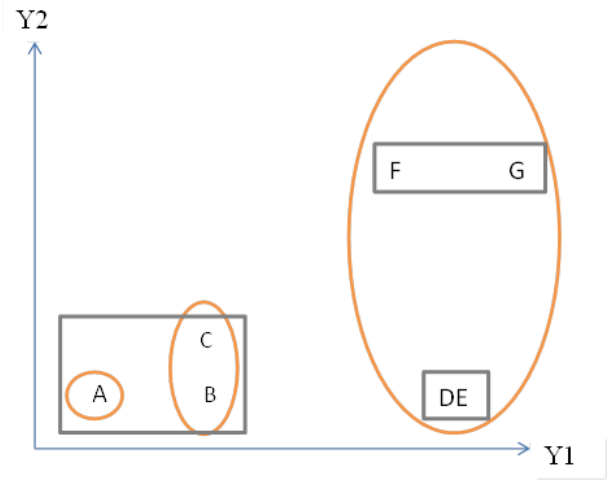

Figure 2.5: Choosing strat point in k-means based on (29).

Graph-theoretic clustering is one of partitional algorithms. In this algorithms "minimal spanning tree" [54] constructs edges by the longest length . Clusters are divided by this edge. This kind of clustering overlap hieratical clustering in terms of single link and complete link are forms of subgraphs minimal spanning tree of data. [21] . *Delauny graph* is one of the form that use in this category. Here all neighbor pairs connected to each other.

2.6 Comparison of clustering algorithms

As discussed before clustering algorithms cover a broad area of application and disciplines. Each area has its specification for algorithms and tries to tailor it to its disciplines. This issue causes different algorithms with different characteristics. Therefore, comparison of these algorithms is hard due to diversity of algorithms. Table 2.2 shows some of the common characteristics of clustering algorithms in three main categories discussed in pervious section and compare them in the framework [25].

Table 2.2: Comparison of Different clustering algorithms based on Halkdi et al (25)

Name	Category	Data type	Complexity	Clustering criterion
K-means	Patrtitional	Numerical	$O(\eta)$	Repeat until satisfy the threshold criterion
CURE	Hierarchical	Numerical	$O(\eta^2 \log \eta)$	Clusters with close representative member merge to each other
OPTICS	Density based	Numerical	$O(\eta \log \eta)$	Start in one point and find clusters which are density reachable and density connected

Name	Noise	Input parameters	Result	Shape
K-means	No	Number of clusters	Cluster centers	Linear shape
CURE	Yes	Number of cluster and representative	Assignment of data values to clusters	Arbitrary shape
OPTICS	Yes	Cluster radius and Minimum number of points	Reachability distance, ordering points	Arbitrary shape

2.7 Clustering and human movement data

When one talks about mathematics and statistics the first thing comes to mind is numbers. These two and other computation analysis tools are used increasingly in the area of geographic data. The strength is these methods are well developed and work in many areas. Everything is numbers, but spatial data is somehow different from non-spatial data. Geographical data-and also temporal data-has special properties that make it different from another type of data [6]. Therefore, to consider space and time simply as numerical values cause many problems and make them unapplicable data analysis. The question here what are these properties The? basic property of time and space is the dependency. In order order to work with geographical data we should consider "the first law of geography":"every thing is related to everything else but near things are more related to district things" [49]. In spatial term it is called spatial auto correlation. Usually in statistical view, data are independent and statical approaches operate under this assumption. While spatio temporal data violate this assumption(independency). In the first view, this property seems a problem to deal with spatio temporal data, but in fact this property help to understanding better pattern and global behavior of this kind of data in some ways like interpolation of spatio temporal data. For example some time there is a missing observation among data. Interpolation uses the concept of dependencies between pervious and next observations to estimate missing value. Another advantage of dependencies is spatial overlay of different kind of data, for

example raster data overly on vector data. There are some exceptions for first law of geography:when cross from mountain to land or water to field or maybe some area divided by mountains have different climate in sides. All of these issues show the complexity that analyst face to dealing with spatio temporal data. Therefore a global method to consider all factors of spatio temporal data in order to develop fully automated process is almost impossible. Geovisual analytics provides facility by developing computer, statical and human perception to answer question like how and when the condition changes. For example, whenever going from a jungle to a city the first low is violated. The rule of human is to understand and contribute this condition to analysis. [13]

Various types of geographical data exist. The most common type is point data because of our limitation to acquire continues data. Point data represent the position of an object in space, for example, example position of junction in a road network. As a matter of fact in clustering of geographical data (spatial or spatio-temporal)we are Captured point data because of nature of acquiring data which is discrete point data. By increasing number of location aware devices like GPS , PDA and many other devices possibility to tracking moving objects has increased. This leads to large amount of trajectory data as another type geographic data. Based on this two geographic types of data Miller & Han [28] provide two main categories of clustering algorithms that has been used in data mining. Figure 2.6 shows categorization of clustering algorithms in geographical data.

1. Point clustering
2. Trajectory clustering

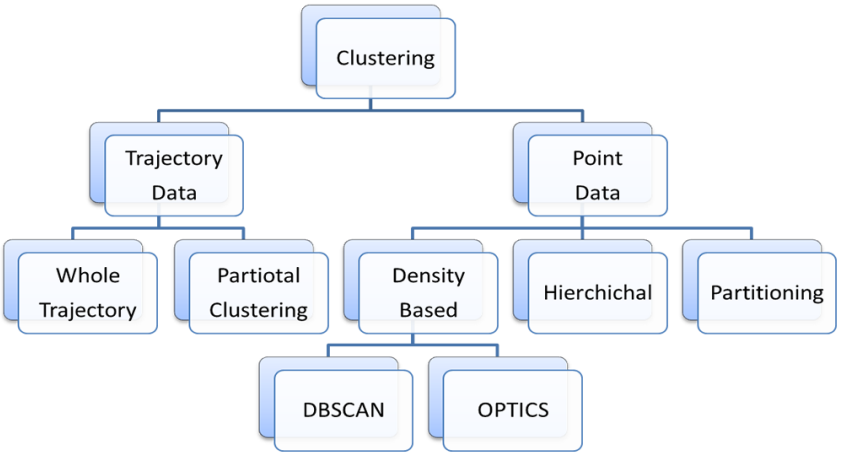

Figure 2.6: Categorization of different clustering method in geographical data based on (28).

Clustering in point data as discussed before divides in 3 main categorizes: *partition based, hierarchical*, and *density based* [31].

1. Partition based approach tries to partition data set and to make goal function optimum in the partitions [28]. The CLARAN method [41] and SOM [32] are examples of partitional approach.

2. The hierarchical approach divides data set in a tree like structure named dendrogram. This methods gives us the possibility of see structure of data in different granulates. [9] The example of this category is CURE [24]which is designed to large data bases.

3. Finally, density-based approach which tries to find more dense area in the data set [45].

Clustering in trajectory data are divided in to two main categories:

1. Whole trajectory: for clustering algorithm consider all length of trajectory

2. Partial trajectory: considering all length of trajectory maybe leads to loss some cluster structure. Partial clustering divides trajectories in parts based on some criteria and the cluster these parts instead of clustering whole trajectories. Figure 2.7 shows the partial clustering in four trajectories.

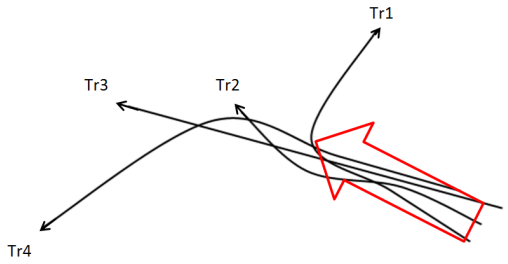

Figure 2.7: Partial clustering could find clusters in the parts of whole trajectories based on (28).

As discussed before, movement data is a complex type of data. A lot of research has been done on aggregation, generalization, clustering and visualization of these kinds of data. Here some of these researches will be reviewed.

Typical way to display trajectories is cartography Space-Time Cube [36]. As trajectories become more and more the display is affected and misunderstanding can happen. Work has been done to deal with this problem by generalize from 2D kernel density and 2D point data to 3D space -time density Figure 2.8. In this view data should be represented in 3D space. The issue of volume visualization is problematic here because human cannot see more than three dimensions. Three methods were used to volume visualization. These different visualizations could help to find spatio temporal patterns. For example, in all these visualization (visualization of space time density)hot spots spot shows the nearness of trajectories in space and time. That means many movements exist in same time and near places . One of the advantages of visualization like it is extraction spatio temporal pattern. Two main patterns are introduced here: temporal bridges and temporal towers. Temporal bridges

means several object exist at a certain time at a certain place, but in another time no object exists. Temporal towers means there are always many object exist near each other. Real dataset acquired from moving ship in gulf of Finland are used to show the results. Two types of ships were chosen: passenger and tanker. For each kind of ships daily density were calculated. These values were shown in voxels(3D pixel). As mentioned before three kinds of visualization were used.(Figure 2.8)The result visualized in the space time cube as expected before was cluttered. From moving ship in gulf of Finland are used to show the results. Two types of ships were chosen: passenger and tanker. For each kind of ships daily density were calculated. These values were shown in voxels(3D pixel). As mentioned before three kinds of visualization were used. In this situation analyze of results is a hard work. 3D space time density visualizations show the better result. This is help not only to better visualization view also to find patterns in these data. For example, temporal bridge was found in the Gulf of Finland. Patterns of passenger ships are very different from tanker ships. Direction and density are different also. For example, passenger ship are very dense between Helsinki and Tallin in the evening and morning.or some tankers stable for a long time [14].

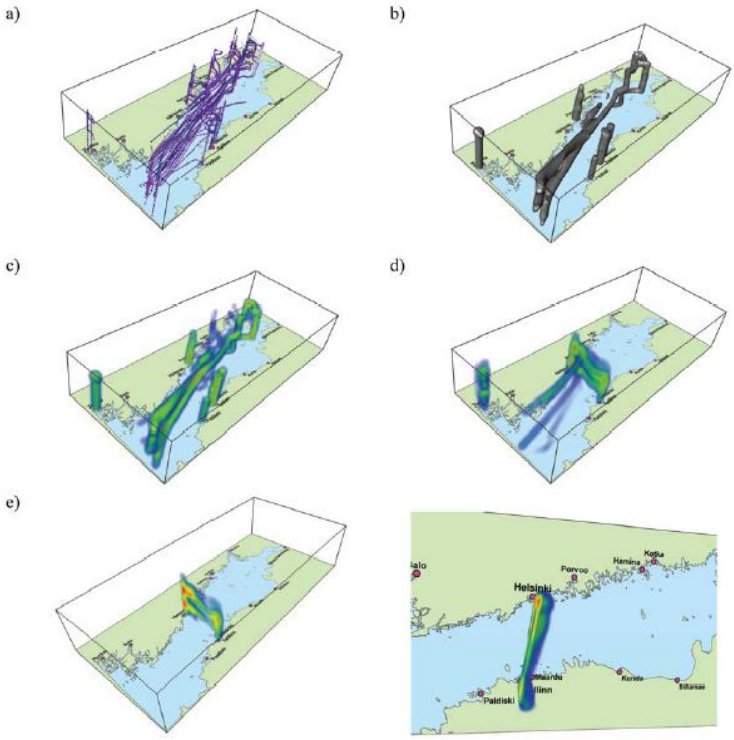

Figure 2.8: a) Space Time Cube.b&c&d) Using 3D space-time density in three differnet visualization form.e) Showing patterns in two views between Helsinki and Tallin (14)

In the case of generalizing movement, perhaps the most obvious work has been done by Laube et al. as relative motion(REMO) [18]. Study of moving point

objects covers a broad area of research like animal movement, traffic management and, etc. Also in location Based Services on of the crucial components is understanding of motion patterns. The challenge, here is to find appropriate tools for analyze and understanding this kind of data [43].

Study of moving point object could be reviewed in two prospectives:GISeinse and Database. In Data base view, here concentrate is on spatio-temporal databases. This kind of Data base is different from traditional DBMS(Data Base Management System)in the term of dynamic updating of data. In moving object databases because of changing the position of an object , data base should dynamically change the attributes of moving points [50]. In traditional database system query processing is just simple query without any analysis of data. For example group or generalize of stored data. The problem is to find a process to do analysis on data to come out with new information [20] [7].

Most of the works have been done in GISience were about the static aspects of phenomena like static position of houses in a city [36]. Handling temporal aspect of data is a hard work. Although some approaches try to address this issue like the space time prism but in compare to pervious pure spatial works, are still immature. KDD as defined before is dealing of large amount data, which is tracking data here. In data mining level of KDD by using clustering algorithms and analysis, we are trying to find patterns. In other hands GISience provides powerful tools to analyze geographical data. Special Kind of KDD named GKD stand for Geographic Knowledge Discovery tries to find pattern and extract information in the large amount of geographical data by help of geographic analysis tools. Therefore, by providing analysis tools in GI Science, It is possible to extract pattern and knowledge in huge spatial data. [34] As mentioned before KDD involve five main steps:"data selection, preprocessing ,enrichment, mining ,visualization and interpretation" [17]. REMO as an algorithm to find pattern in data involve in all five steps. For example, to create REMO matrix, it's essential to project data to the matrix(REMO has been discussed more in pervious chapter) [34].

Study of movement pedestrians is one of the broad and active research throw previous decades. These study must relate on the photo, video and other data type. Recently increasing facility to tracking individuals lead to provide good sources for data mining and extract the pattern. Although movement of the crowd is affected by constraints like traffic light, street, etc. But these constrain do not affect intrinsic behavior of the movement objects that is called pattern. Movement of pedestrians shows in lifeline or trajectories. Flocking approach as discussed before tries to find group among individuals. Individuals should be near in time and position to consider as a flock. Flocking is usually considered as a collective movement over time. In this view a flock is a group of objects that move together for a time period [8]. The important point that makes flock different from clustering is persistency among members of a flock over a period of time.

Definition of clustering of moving object is provided by Kalnis et al. as a group of objects close to each other for a period of time. Example of this kind of clusters is a convey of cars or migrant animals. In this work, DBSCAN [16] was used in each step of time. Cluster is found when a certain number of objects

remains in the cluster with respect to time $t-1$. It doesn't matter the members are the same or not for the whole process(all time steps). What is important density of the cluster should be big enough with respect to a threshold [30].

Although the definition of flock and moving cluster quite similar and confusing-for example migrant animal could be either a flock or moving cluster-but there are differences in member structure as discussed above. These two concepts are overlapped in each other. Figure 2.9 shows moving clusters and Figure 2.10 shows flock patterns [52]. In flock members should be the same from start to end. However in clustering members could leave or enter to cluster, as long as the density of cluster is persistent.

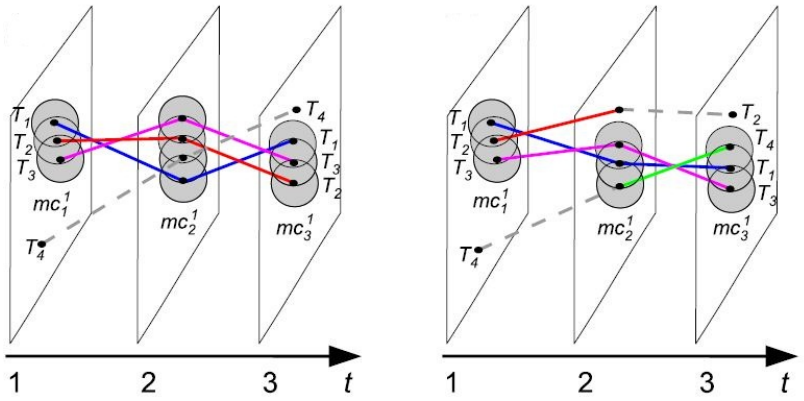

Figure 2.9: Moving clusters (52)

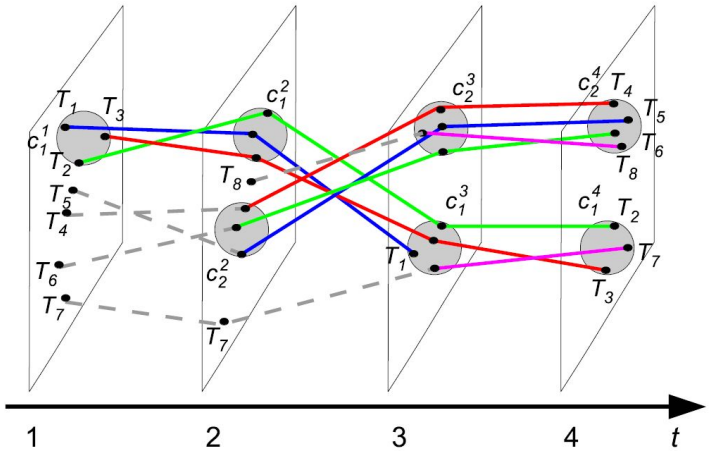

Figure 2.10: Flock patterns (52)

Chapter 3

The OPTICS Clustering

In this chapter OPTICS algorithm will be described as a sophisticated algorithm among density based algorithms. Definitions and pseudo codes relate to it will be defined. Current works in area of spatio temporal clustering will be reviewed at the end of this chapter.

3.1 Introduction

In pervious chapter, we saw definition and concept of clustering in different views. Of course important issue here is geographical data and specially those which relate to human movement data. As reviewed before strategy of clustering in geographical data is divided to two main categories [28]:

1. clustering of point data
2. clustering of trajectory data

Clustering of human movement data we are faced to the point data because of our limitation in acquiring completely continues data. Point data in this case relate to independent snapshot of position in the time. Our instruments and also our costs are limited in acquiring continues data here. Even if the possibility of continues positioning would be provided still the nature of data would be point data with high resolution. Therefore, we use different models like a continues function-trajectory- to model the reality. Again our entrance in this kind of modeling is point data.

Clustering of point data also is divided into three main categories: [28]

1. Partitional
2. Hierarchical
3. Density based

Partitional clustering as discussed before partitions space and evaluates the partitions based on some criterion. For example minimum square error. The most well-known methods in this category is:K-means which repeatedly search the data set for clusters and cluster centers. In each repetition the clusters

centers and therefore clusters will be enhanced and this process will continue until the predefined criterion for stop is satisfied. Hierarchical approach decompose the data and creates a tree like structure-in such a structure the leafs are members and the branches show different clusters and different branches are the member of another clusters and so on-, like CURE which define representative for clusters and assigns the members with near distance to this data to different clusters [24]. And finally density based clustering searches for high density structure based on predefined criterion. The well-known methods of this category are DBSCAN [16] and OPTICS(Ordering Points To Identify the Clustering Structure) [5].

In the pervious chapter and the beginning of this chapter the definition of point data and trajectory data were discussed. These data as a good representative of spatio-temporal movement data often have random component and low(or different) resolution component for example to build a trajectory of someone from home to work, certain amount of point is captured which maybe not showing completely behavior of him/her. In this case we don't know captured point are really coincide on the reality or even are desired(important point)or not . These characteristics motivate researchers to use density based clustering algorithm in this area because density base methods are noise tolerant and easily adapt to complex data [45].

In the following sections first Density based clustering methods will be discussed and the OPTICS algorithm will be discussed in terms of basic definitions, key parameters and pseudo codes. Finally applications of this method and distance functions will be reviewed. The remainder of this thesis will consider implementation of the OPTICS algorithm, and to extend it to work with spatiotemporal data.

3.2 Density based clustering [1]

As discussed in the pervious chapter cluster analysis could be used as an approach to get insight on data and also it could be consider as a step before more analysis for different purposes like pattern extraction, query optimization and etc. There are many methods and algorithms for clustering. Generally they suffer from main problem:determaining the initial values for parameters. Defining these parameters is very difficult due to variety of algorithms and data sets. Usually structure of data is so complicated which makes finding clusters by a general parameter impossible. For example in GPS data set of moving people in a city it's impossible to find a clustering parameter to extract all clusters (small, medium and big groups). Another problem is the sensitivity to these parameters which makes the result different.

One of the important properties of real data sets is that they are unpredictable. Therefore it is not possible to model them by a global density functions and variety of local density functions are used to model them. Density function is used to measure similarity and dissimilarity between objects. To make this clearer we will assess this issue as that is the key reason behind the develop-

[1] This section summarizes the seminal work of Ankerst et al. [5]

ment of OPTICS [5]. Consider Figure 3.1 . In this figure two groups of cluster are detectable: A, B, C and C1, C2, C3.

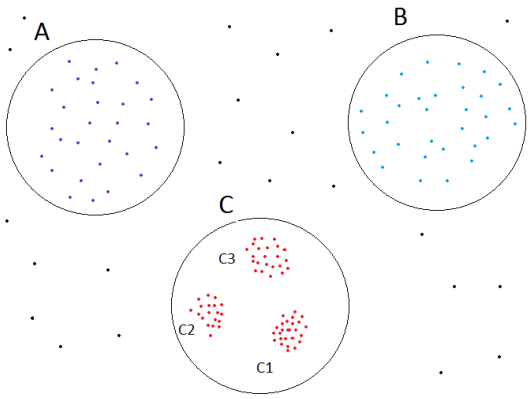

Figure 3.1: Two groups of cluster: first group A,B and C and second group C1, C2 and C3 based on (5)

This is clear if one density function find second group the first group would be noise. From this observation it is clear that by using a general density function we can not identity all existing structure in the data. Two solutions exist:

1. Dendrogram and hierarchical structure
2. Using concept of density based clustering for many parameters

The first solution fails because these kind of clustering techniques for large data sets are hard to interpret. As discussed before these techniques makes a tree like structure. In this structure each leaf(end member)is member of several clusters at the same time. Here the clusters are available in several levels e.g. leaves, different branches, and roots. The interpretation of clusters is done based on this levels of clustering. Therefore in large data set for example 1,000,000 records-it is hard to find the best level and good results. Another aspect of this kind of structure is visualization which will be very hard to construct in the large debases.

The second approach is density based clustering which as discussed before infinite parameter settings and infinite clusters are possible. The problem is clear because of too many possible answers. Therefore the solution which is offered here:finding the structure which includes all cluster structures. This is the basic concept of OPTICS. Lets have an overview on definition of density based clustering.

Density Based Clustering:" the basic concept in this clustering is that in a defined radius at least minimum number of points are available" [16]. Related definitions to support this basic definition are provided in the next chapter.

3.2.1 Definitions of density based clustering

1. Directly density reachable:object p is directly density reachable with respect to(w.r.t) object q if:
   (a) q is a core object
   (b) For p given ϵ and MinPts ,p in the radius

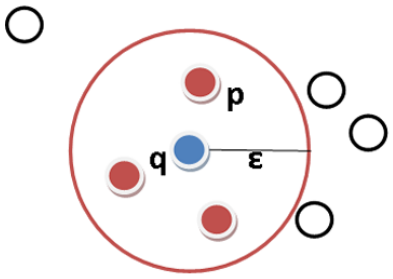

Figure 3.2: Directly Density Reachable w.r.t MinPts 3 and Epsilon neighborhood

2. Core object:objet q is a core object if the objets in the ϵ neighborhood of it is at least equal to MinPts

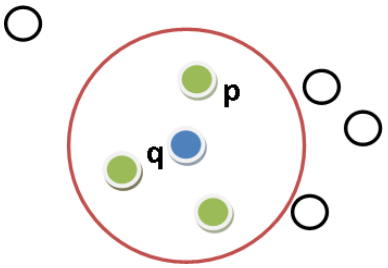

Figure 3.3: Core object

3. Density Reachable: object p is density reachable w.r.t object q if for a chain of objects like $p_1 \ldots p_i$ each object p_{i+1} is directly density reachable from p_i(p_i=p and p_1=q)

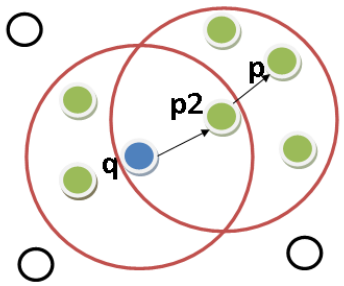

Figure 3.4: p is density reachable from q

4. Density connected:objects p and q w.r.t object o is density connected if both of them(p and q) are density reachable from object o.

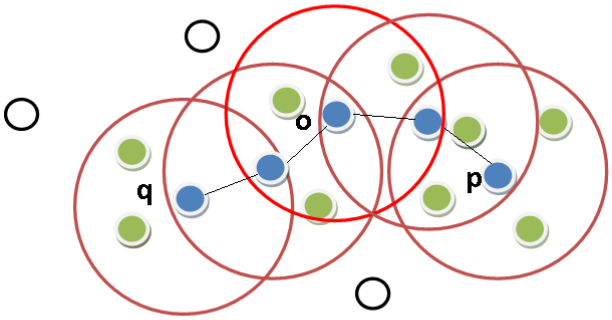

Figure 3.5: Objects p and q are density connected w.r.t object o

5. Cluster is detectable if has two main conditions:
   (a) Maximality:if object p is a member of cluster $\Re$, and object q is density reachable form p , q is a member of $\Re$ too.
   (b) Connectivity:if p and q is a member of cluster $\Re$, a and b which is density reachable from p and q should be density connected.
6. Noise:objects that don't belong to any clusters are noise.

DBSCAN is a clustering algorithm based on these definitions. First one arbitrary point in data base is chosen. If the chosen point was a core object, then cluster will be expand by the concept of density reachability(the points which are density reachable from the core object) and density connectivity(the points which are density connected by the core object). This means every point which is density reachable and density connected by core object point is assigned to

the cluster. The expansion will be done on the neighbors and continue until no core objects are remaining. Then again another object from the data set will be chosen and the procedure repeated for new clusters.

In spite to the benefits of this approach which makes it different from clustering methods, there are drawbacks like choosing ϵ which needs a very appropriate setting as maybe causes loosing or gaining clusters(could be considered as importance of this parameter in the final result indeed). By choosing small value(higher density)the structure of clusters will be different than for high values(lower density). Therefore the user should choose the parameter with respect to the structure of data(for example sparse or dense data set). But in the case of combination of different structure as discussed, it would be problematic.

The problem that we faced the previous case was that retrieved clusters from high ϵ values include low values. The purpose here is to find the lowest value of ϵ to ensure that all clusters(with different densities) are detectable. This is possible by the concept of density reachability, by considering the lowest density reachability. OPTICS doesn't actually make any clusters. In fact all it does is making an ordering database by the concept of *Reachability Distance* and core distance. Then by using DBSCAN it is possible to extract clusters from these concepts(reachability distance and core distance for all points in data set). It will be shown how these concepts are enough to extract any cluster. First it is important to provide an overview on two main definitions in OPTICS:

1. Core distance:for a given ϵ and MinPts, minimum distance $\acute{\epsilon}$ that makes the object core object(C-O).

2. Reachability distance:for a given object q from p:

$$ReachabilityDistance = \begin{cases} \text{U}NDEFINED & \text{if } p \text{ is not a C-O} \\ \text{m}ax(\text{Distance}(p,q),\text{Core-distance}(p)) & \text{if } p \text{ is C-O} \end{cases}$$

This is clear that distance of no object w.r.t core-object won't be lower than the core-distance, because in this situation the condition of the core-object violates. In this situation, that distance-the lower distance w.r.t core object- would define the neighborhood of the core-object, therefore the condition of MinPts in the neighborhood is not possible anymore and the object will not satisfy core object condition(defenition 2). OPTICS does ordering of data base and finds core distance and reachability distance for every point of data base(if available for all points, as mybe they-core distance and reachability distance- would not be available for some points).

3.2.2 Pseudo codes of OPTICS

After reviewing the concept and definitions of OPTICS It's worth having a brief review of the pseudo codes and see how it comes together. As reviewed before the aim of OPTICS is to build an ordered structure from data set with respect to object reachability distances. This ordered objects equal to many parameter settings of density based clustering.

Pseudo codes of OPTICS consists of three main loops:

1. Main loop (Figure 3.7)
2. Expand cluster (Figure 3.8)
3. Update order seeds (Figure 3.9)

Figure 3.6 shows all loops in a flowchart view.

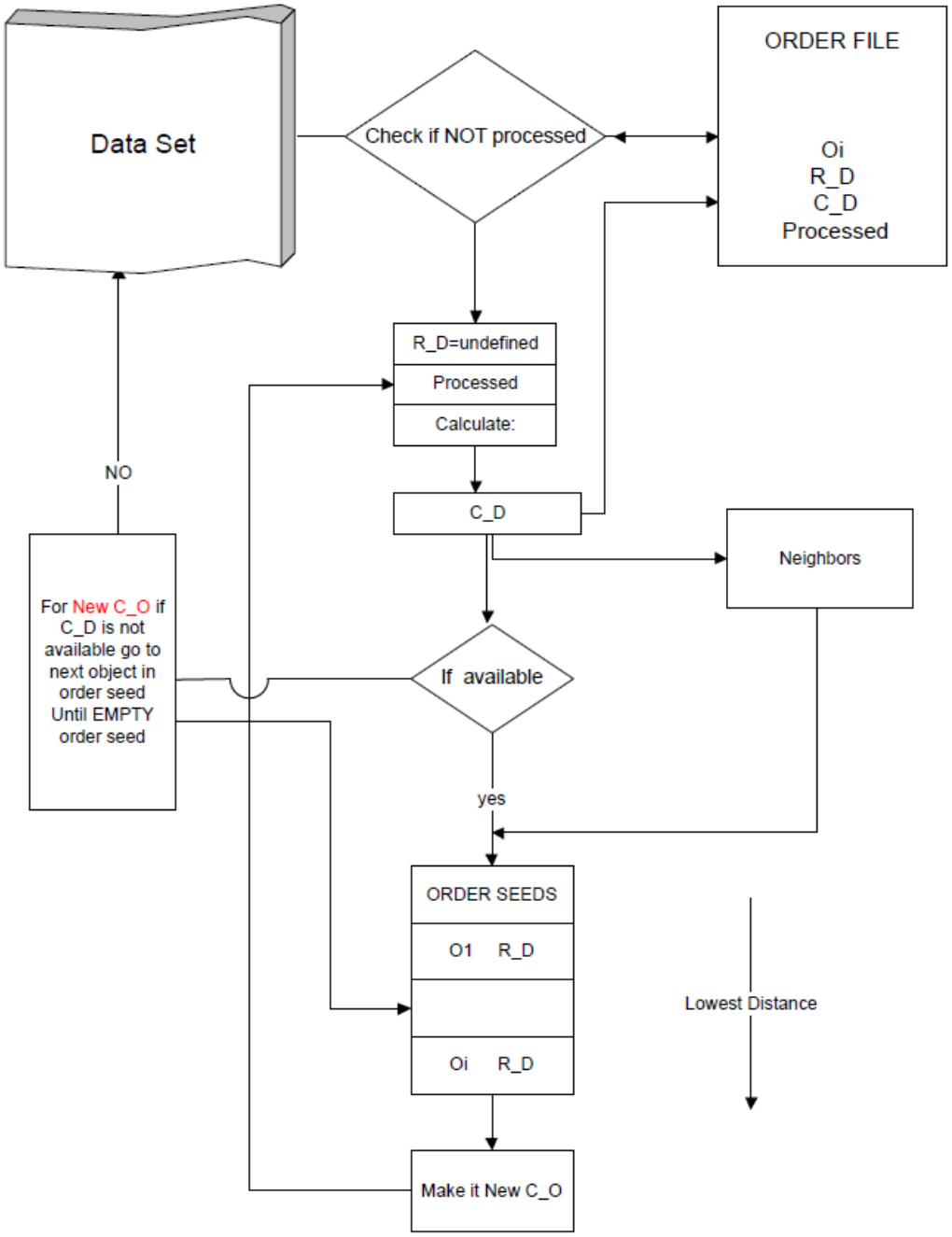

Figure 3.6: OPTICS flowchart based on Sarhangi(2010)

Algorithm starts by opening an empty order file in the main loop which is shown by Figure 3.7. It searches the domain(set of objects) for every single object. As long as object is not processed goes to the next loop(Expand cluster). This process repeats until all objects get to order file. An object goes to order file if three main measurements done.

- Reachability Distance (R-D)
- Core Distance(C-D)
- Mark as processed

Again the algorithm comes back to Expand cluster shown in Figure 3.8 and consider the object as new core object, finds core distance, neighbors, mark as processed and writes it in Order file. The difference between this new object and prior object is in previous case if C-D was not available, new object from the domain was chosen, but here (after Order seeds) if C-D is not available algorithms back to the que in Update Order Seeds and chooses the next object.
All this process repeats until all object writes in Order file.

```
OPTICS (SetOfObjects, ε, MinPts, OrderedFile)
  OrderedFile.open();
  FOR i FROM 1 TO SetOfObjects.size DO
    Object := SetOfObjects.get(i);
    IF NOT Object.Processed THEN
      ExpandClusterOrder(SetOfObjects, Object, ε,
                MinPts, OrderedFile)
  OrderedFile.close();
END; // OPTICS
```

Figure 3.7: OPTICS main loop (5)

The next loop after main loop is Expand cluster which assess objects passed the main loop with respect to ϵ and MinPts(minimum number of point). Following content will be assigned to object in the first step(Expand cluster loop):

- R-D=Undefined
- Processed✓
- Calculate C-D
- Find Neighbors

In this level object satisfies the order file condition, therefore, is written in order file. Then the algorithm checks the C-D for this object. If it isn't available-in other words object is not a core object- means there is no chance to expand cluster and goes back to(main loop) the next new object in data set. If it's a core object- C-D is available- check the neighbors for further expansion.(Update order seeds)

```
ExpandClusterOrder(SetOfObjects, Object, ε, MinPts,
OrderedFile);
  neighbors := SetOfObjects.neighbors(Object, ε);
  Object.Processed := TRUE;
  Object.reachability_distance := UNDEFINED;
  Object.setCoreDistance(neighbors, ε, MinPts);
  OrderedFile.write(Object);
  IF Object.core_distance <> UNDEFINED THEN
    OrderSeeds.update(neighbors, Object);
    WHILE NOT OrderSeeds.empty() DO
      currentObject := OrderSeeds.next();
      neighbors:=SetOfObjects.neighbors(currentObject, ε);
      currentObject.Processed := TRUE;
      currentObject.setCoreDistance(neighbors, ε, MinPts);
      OrderedFile.write(currentObject);
      IF currentObject.core_distance<>UNDEFINED THEN
        OrderSeeds.update(neighbors, currentObject);
END; // ExpandClusterOrder
```

Figure 3.8: OPTICS Expand cluster loop (5)

Update order seeds loop in Figure 3.9 is important loop because it finds a queue from neighbors. For a given object and its neighbors assigned reachability distance to all neighbors in a list. This list is built based on ascending R-D which Means the end member(s) of list is nearest object to core object. This neighbor will be chosen in order seed to next expansion in next level(Expand cluster).

```
OrderSeeds::update(neighbors, CenterObject);
  c_dist := CenterObject.core_distance;
  FORALL Object FROM neighbors DO
    IF NOT Object.Processed THEN
      new_r_dist:=max(c_dist,CenterObject.dist(Object));
      IF Object.reachability_distance=UNDEFINED THEN
        Object.reachability_distance := new_r_dist;
        insert(Object, new_r_dist);
      ELSE // Object already in OrderSeeds
        IF new_r_dist<Object.reachability_distance THEN
          Object.reachability_distance := new_r_dist;
          decrease(Object, new_r_dist);
END; // OrderSeeds::update
```

Figure 3.9: OPTICS Update seed loop (5)

According to the "*run-time*" calculations on different data set It's claimed that OPTICS run-time is 1.6 of DBSCAN run time because of similar structure in both algorithms.(Density based on predefined radius and MinPts).Cluster results extracted from OPTICS have different and some case undefinable by DBSCAN cluster results.

3.2.3 Visual technique in OPTICS

In previous section, it was discussed how Implication OPTICS on data set leads to order file which is an ordered structure of data set w.r.t reachability distances. Two ways available for using order file in visual techniques in OPTICS:

1. Have an overview on data and structure

2. Finding our desire details, after having general view, see trivial properties of data

Reachability plot gives powerful tools to Visualizing the OPTIC results. It has two axes the horizontal axis shows the ordered object, and vertical axis shows reachability distances for every single object. (See Figure 3.10)

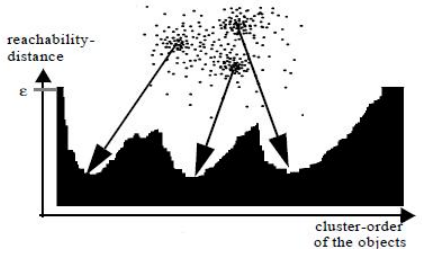

Figure 3.10: Reachability plot (5)

The advantage of this approach(visual techniques)is getting an overview in data set and this method is not sensitive(of course not completely)to value of ϵ and MinPts. In another these parameter should be big enough to find clusters for us. Again we emphasis that this property belongs to visual techniques, which try to have an overview on data. The impact of ϵ has been shown in Figure 3.11 below.

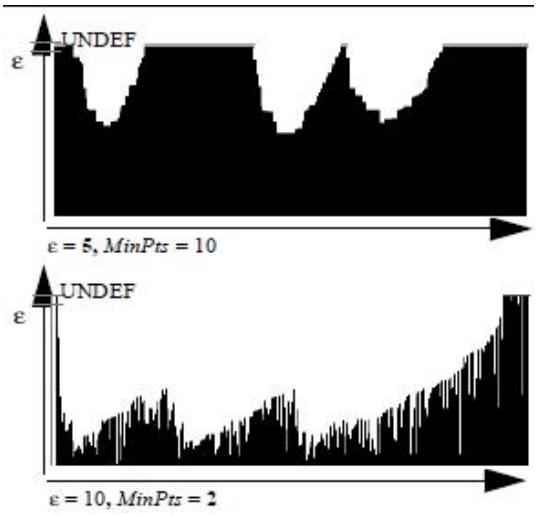

Figure 3.11: Impact of parameter settings in OPTICS (5)

As shown in the first part of Figure 3.11 lower value for ϵ and high value for MinPts leads to smooth diagram. In second part of this Figure 11 the high value of ϵ and low value of MinPts lead to rough structure. Although this variation has changed the shape of reachability plot somehow but the general structure remains unchanged. Theϵ parameter makes reachability distance undefined or defined. The less ϵ, the more undefined reachability distances. MinPts parameter impacts the shape of reachability plot. The more MinPts smoother shape

and less value leads to more tracks in reachability plot.

Figure 3.12 demonstrates the power of OPTICS in finding different clusters by using readability plot.

Figure 3.12: Clusters are valies in reachability plot (5)

3.3 Distance function

One of the important properties of OPTICS is using *distance function*. The idea of using distance function is to find similarity and dissimilarity of objects in the data set based on desired properties i.e. using different attribute to clustering like location in geographical view, the age of people, the sex, etc., some combination of these properties, or all of them. Rinzivillo et al define distance function as: [45]

"Clustering techniques use certain methods for estimating differences between objects in terms of relevant characteristics. These differences are also called *distances* and the methods for computing them are called *distance function*."

By using OPTICS as a fundamental framework for clustering and using appropriate distance functions it is possible to cluster various kinds of data-including point data, trajectory data- by different characteristics(spatial resolution, temporal resolution, sampling). This property of OPTICS-using distance function- allows it to be a powerful tool for grouping data. For example, by using appropriate distance functions it is possible to find flock patterns of the group of objects moving together for a given period of time.

Distance function could be considered in different views. Although there are not many distance functions implemented in the area of OPTICS clustering, but theoretically for clustering area it is possible to view distance functions as:

1. Exploring data
2. Mathematic function

3.3.1 Exploring data

The first view of distance functions is searching data, which tries to extract similar objects from data set by different characteristics. Finding clusters of people in a certain location or certain time or combination of time and location' For example, clustering in the entrance gate of a park in different times of day, week or month, is an example of such a distance function. Here distance function works like a complicated query to explore data. The brilliant work in this

area has been done by Andrienko et al. [3] They developed a library of different distance functions for different kinds of data. One of the simple distance function which introduced in the developed library is *similar destination*. Here the OPTICS only clusters the trajectories in a certain time and location(destination of trajectories). The result of such a function would be trajectories that go to the same, but not necessarily exact place in the same, but not necessarily exact time. Of course this is a quite simple function. This work will be discussed more in the section 4.2.

3.3.2 Mathematic

In the second view which is distance function in mathematical point of view, the problem of measuring similarity or distance between sets of points is considerable. For defining distance function in this area, the defenition of *metric* provided by Eiter and Mannila should be considered [15]. They defined function f:$\mathbb{D}\times\mathbb{D} \longrightarrow \Re^{+}$ as a metric on set $\mathbb{D}$, if for $x, y, z\in \mathbb{D}$:

1. $f(x, y) = 0$ if and only if $x = y$
2. $f(x, y)$=$f(y, x)$
3. $f(x, z)\leq f(x, y)$+$f(y, z)$

Every f function which satisfies the first and second situation is a distance function. Different distance functions were introduced in this category, like Handoff distance, Sum of minimum distance, surjection distance and Link distance [15] This definition could be extended to the clustering as object a and b are quite similar if the distance between them is zero(first situation), and object a and b are similar if distance between a w.r.t b is equal to distance of object b w.r.t a.

Han et.al[27] porvide a list of distance measure for objects. According to this list One of the popular distance measure is Minkowski distance:

$$d(i,j)=\sqrt[n]{|x_{i1}-x_{j1}|^n+|x_{i2}-x_{j2}|^n+\ldots+|x_{ip}-x_{jp}|^n}$$

i=$(x_{i1}, x_{i2}, ..., x_{ip})$ and j=$(x_{j1}, x_{j2}, ..., x_{jp})$ are objects in p dimension. When n=1 Manhatan distance will be driven:

$$d(i,j)=|x_{i1}-x_{j1}|+|x_{i2}-x_{j2}|+\ldots+|x_{ip}-x_{jp}|$$

And finally if n=2 Euclidian distance will be driven:

$$d(i,j)=\sqrt{|x_{i1}-x_{j1}|^2+|x_{i2}-x_{j2}|^2+\ldots+|x_{ip}-x_{jp}|^2}$$

Our distance function in Space-Time Cube clustering works based on the Euclidian distance in 3D space. We used X, Y and T in the Euclidian distance formula. The relation between time and location will be defined by speed parameter. This issue will be discussed in detail in Chapter 4.

3.4 Recent implementations of OPTICS

OPTICS gives powerful tools for clustering and has been used specially in movement data. The popularity of OPTICS in movement data back to the properties this kind of data which usually are known by trajectories. Different sampling rate both spatial and temporal, unpredictable nature of movement and other properties are motivated using density based clustering in this area.[45] Advantages of using OPTICS in this area are[27]:

1. Has information of broad range of clustering parameter setting
2. Powerful in interactive and automatic clustering
3. Could be used by visualization techniques

In the prior sections two methods which used OPTICS and their properties will be discussed.

3.4.1 Progressive clustering of movement data [2]

Trajectories are complex data type with different properties. Similarity between trajectories is found by distance function. Distance functions try to find similarity in terms of properties of trajectories. Therefore, using different distance functions tailored to different characteristics is essential to do clustering. The idea of general distance function (only one distance function for all properties)fails because:

1. Hard to construct
2. Difficult to interpretation of result
3. Not efficient for large data (increase run-time)

As discussed before the aim of clustering is finding similar object in data set in order to easy handling of data or as a primary step of other operations. Analyst cooperation is a main part of this process: i.e. the cooperation of the analyst through interactive display or the result of clustering. In the case of movement data, clustering problem back to trajectories. Trajectories have different characteristics based on geometry or dynamic like speed, azimuth, etc. Each distance function tries to measure similarity between these characteristics. Problem of general distance function is discussed before. The solution is to find distance function for each property. Progressive clustering works based on this issue, means different distance function in each step. This is the basic schema of "progressive clustering".

[2] This section summarizes the seminal work of [45]

How does it work?

Based on what has been discussed it is found that trajectories are complex data and general distance function failed in this data. This approach-progressive clustering- works based on step by step progress. In the first step one distance function exert on data set. Result of this step are processed by analyst, which knows the distance functions and properties. Based on goal of clustering another distance function implements on the result of previous step(s). This process continues until achieving the predefined goal. During one step, it is possible to use the combination of different distance functions. Visualization and interactive methods are important in this situation. One of the main advantages of this approach in addition to easy interpretation and easy analyzing, is handling large data sets, by using simple distance function to have an overview on whole data set and then using sophisticated distance function in the next level. That helps us to better handling large data sets.

The heart of this approach is OPTICS, which uses distance function(s). Distance function could be used for exploration of data or in mathematical perspective. From previous sections, it's known that OPTICS doesn't make clusters. Instead, it finds reachability distance for every point in data set and puts them in an order. OPTICS is sensitive to choosing MinPts parameter because of impact on picking up a point as core-object or not. For example, in a very dense data set low value of MinPts leads to increasing the number of core objets. In consequence reachability plot shape will be fluctuated(show many valleys). Or in contrast high value of MinPts leads to decreasing the number of core objects and smoother shape. However,in the case of sparse data small number or no core-object will be found. In fact "*Parameter of Minimum number of points should be adjusted to the data*".

There are two general ways to clustering trajectory data:

1. "feature-based model"
2. "direct geometric"

In *feature-based model* object directly projected to the model space, and similarity will be measured in this space. For example, each property projects to one axis. *Direct geometric* uses the properties of time and space to finding similarity. The properties include start time, end time, dynamic and destination, etc. Progressive clustering works based on direct geometric approach. Based on this, different distance functions are defined, include simple distance function like "*common destination*" which calculates the real distance on earth of end of trajectories or more complicated function like "*route similarity*" which is used for uncomplete trajectories(when trajectories doesn't match either in time or space). This distance function search on two trajectories for nearest points and compute the mean of measured distances.

Visualization of clusters

As discussed in Chapter 2 common way of visualizing clusters is "*Space-Time Cube*" [36]. But a problem arises when visualizing many trajectories. The so-

lution is clustering trajectories based on similarities in important places and visualizing them. Having this goal in mind two main issues arises:

1. Finding similar structure in movement data is very difficult
2. Choosing important point is a problem here

Rinzivillo et. al. [45] argue that “dynamic aggregators” are a solution.“A *dynamic aggravator* is a special kind of object that is linked to several other objects (members of an aggregate) and computes certain aggregate attributes such as the member count and the minimum, maximum, average,median, etc. from the attribute values of the members." Two main *dynamic aggeragator*s are defined:

1. *summation places* is a place that "*aware*" of all trajectories which pass from that place
2. *aggregate moves* finds trajectories which pass from two sequence points and compute for example mean for them.

Progressive clustering of cars in Milan

OPTICS has been used in this case study by different distance functions. GPS data set of Milan includes 6187 trajectories between 6 to 10 in the morning. OPTICS uses three parameters: distance function, MinNbs(MinPts) and ϵ. It’s clear at that moment people are going to work therefore "*common destination*" is suitable to detect clusters.

In order to find uneven data these steps have been done:

1. Set high values of MinNbs for whole data set
2. Manipulate the parameter by analyst based on the goal (finding important clusters not small clusters)
3. Finding clusters
4. Remove clusters from data
5. Using lower value of MinNbs on remaining data
6. Repeat the process until finding a reasonable clusters

In whole process it should be used one distance function by different parameter settings. Figure 3.13 shows the result of this process. Clusters are depicted by different colors. By this saucerful approach it is demonstrated how progressive clustering find the regions which people go to work.

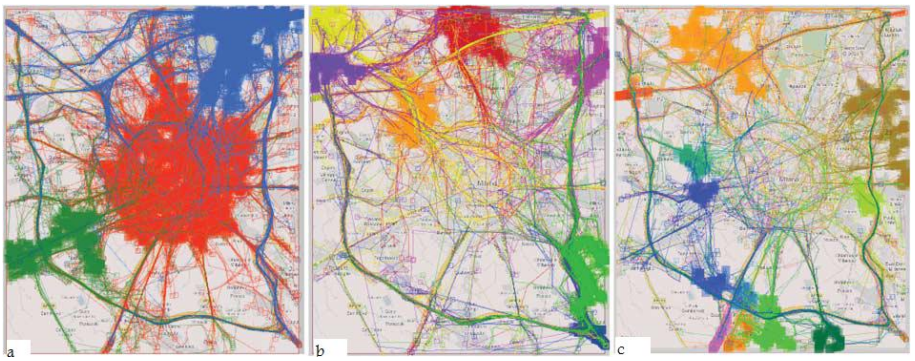

Figure 3.13: Using progressive clustering to find uneven data by common distention(a) parameter settings:e=500, MinNbs=10 (b) e=500, MinNbs=10 (c)e=500, MinNbs=5 (45)

Deeper understanding of data

In pervious section we saw how using "*common distention*" distance function clusters in the central part of city were detected. Consider one wants to have deeper view on data. This is possible by using "*route similarity*" distance function. For this task the biggest cluster(red) in last part is chosen. If the goal is finding the short cluster facing to the cit center, 63 percent of whole cluster will be noises. By using "blue " and "green " clusters the results were the same which comes to this hypothesis:Clusters with near start and destination is more similar to each other. Now If one wants to assess long trajectories, first short clusters produced in pervious step will be removed. Then by using "*route similarity*" on remaining data, clusters will be find. The results show interesting points. If the results divide to two categories,around the city and inside the city It is found that long clusters belong to both two categories. Also in the second category(inside the city) cluster facing toward city center are more than clusters that leave city center. The reason is quite clear and coincide to the working hour in morning.

Conclusion

Clustering will be successful if combine with the visualization tools which is using the result of clustering and combination of them with auxiliary data like maps. In this work application and performance were neglected. The emphasis here on using different distance functions in order to deal with different characteristics of data. These properties impacts the choosing of distance function and also the order of using distance functions.

3.4.2 Interactive Cluster Analysis [3]

Recent implementations of OPTICS are trying to do space-time clustering. It means clustering not only by near location but also by closeness in time. There-

[3] THIS SECTION IS SUMMARIZED FROM [3]

fore clustering of different types of data will be issue of recent works in this area.

In pervious section we discussed how trajectories as a complex type of data with different characteristics can not clustered by ordinary algorithms and we saw two general methods:

1. Choosing algorithm based on data characteristics
2. Using general algorithm like OPTICS and different distance functions

This article follows the second methods which is using OPTICS on three data types and different distance function. Three data sets include:

1. Objects have spatial and temporal characteristics like position of a point
2. Trajectories are moving objects like tracking a car
3. Spatial distribution of events "‘moving entities in different time interval" [3]likes combination of different objects

In the first case different distance functions are used, for example "*common distention*". Distance function in the second data type calculates distance in space and time and combines them as a final distance. In third data type first data divide to different parts and characteristics of points in each compartment are aggregated, which means on point as a representative of all points in compartment. Clustering will be done by comparing each part to corespondent part. For more effectiveness of clustering neighbors could enter to the similarity calculation. Even in the more advanced calculation it is possible to enter weighted neighbors to the calculation. General process could be summarized in three steps:

1. Divide data set in regular or irregular partitions
2. Count neighbor of data in each partition and aggregate them
3. Measuring distance between two spatial distributions in terms of aggregation distances

In following the examples of distance function in three types of data will be reviewed.

Case study:Evacuation traces

In this work RFID data set named "*evacuation traces*" has been used which contains 82 trajectories. This is small data set but author selected this data set to show the effectiveness of approach."*Common distention*" distance function was used to find people who want to quit the building and those staid in building. In this situation we are facing to the second type of data, which is trajectories and appropriate distance function need to be used for this data. This is quit useful to find outliers, for example, terrorists. Analyst manipulates distance threshold

until getting the coherence results. Figure 3.14 shows 4 detected clusters. Hollow squares show the start of clusters and filled squares shows the destination. Furthermore, 5 trajectories were not in clusters.

Figure 3.14: Common destination found 4 clusters (3)

Another experiment was done on this data set that consider third type of data: spatial distribution. People inside the building were divided to times with different values. These values demonstrate the importance. For example, in many moments there are no important phenomena but in a small portion of time many activities were done. By using clustering sequence clusters shows "relative stability"and fast movement consider as noise in data. In this experiment first data set -distributed people in the building-divides into rectangular parts. Then 837 time steps are converted to 160 time interval, which means each time interval includes five time steps except the last interval(836-837) Figure 3.15. Clustering was done on this time interval and at the end clusters with the same members in each interval show the final results. Four clusters were detected with size 3, 4, 24 and 76. Also it is possible to use Space-Time Cube to interpret the clusters. [36] For example in Figure 3.16 it is clear that violet clusters stops moving sooner than green clusters.

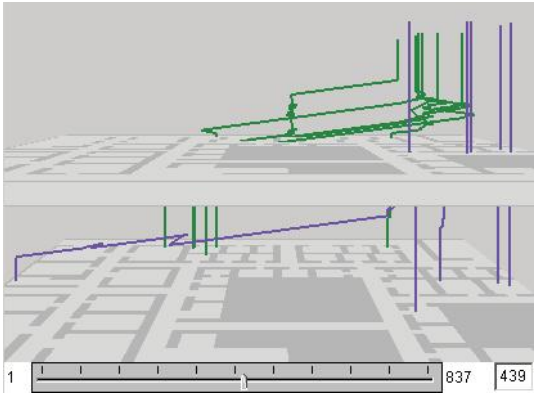

Figure 3.16: Violet clusters stops moving sooner than green clusters (3)

In pervious section we saw how by using "*Common distention*" distance func-

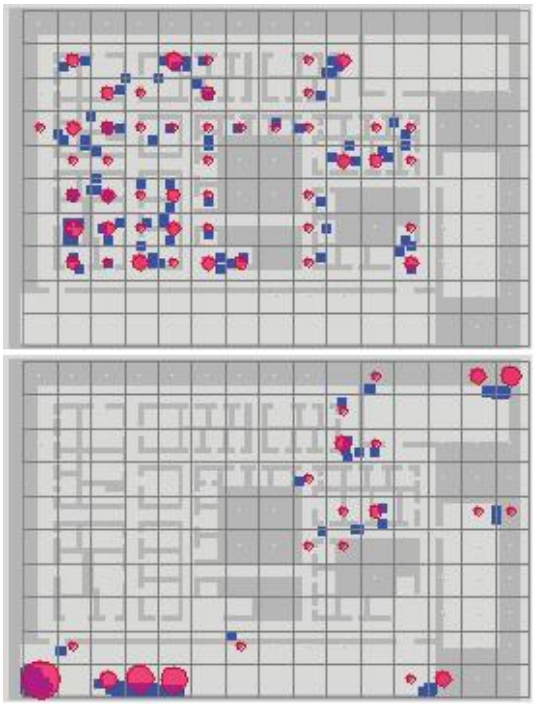

Figure 3.15: Top image show the presence count (red)and position of people(blue)in the first interval and bottom shows the last interval (836-837) (3)

tion special trajectories detect from special trajectories. Here we want to find trajectories based on movement route. The desire distance function is "'route similarity". Figure 3.17 shows the "route similarity" distance function finds nine clusters.

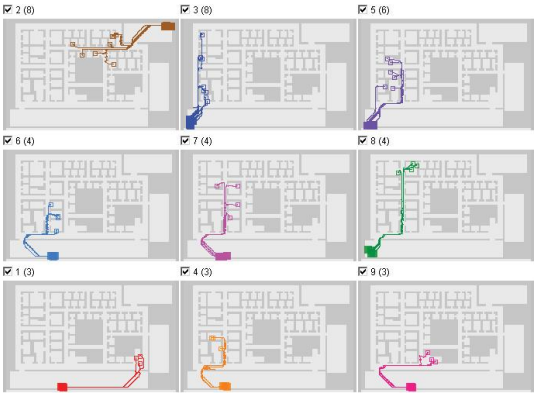

Figure 3.17: Route similarity finds 9 clusters (3)

As an another experiment point data set of migrant boats was experienced. This data set shows illegal migrant boats shows 917 records of landing and interdiction events over three years from 2005 until 2007. Two distance functions of spatial and spatiotemporal distances were used. First one returns simply the spatial cluster irrespective to time. And second one convert the time to spatial

distance and combine it with spatial distance in Euclidian distance formula. Figure 3.18 and 3.19 shows clusters by different color using spatial and spatiotemporal distance function respectively in this data set.

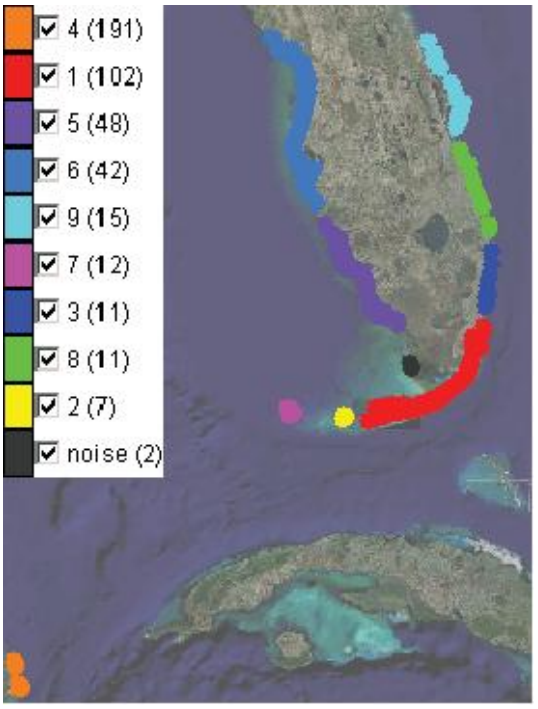

Figure 3.18: clustering by using spatial distance function (3)

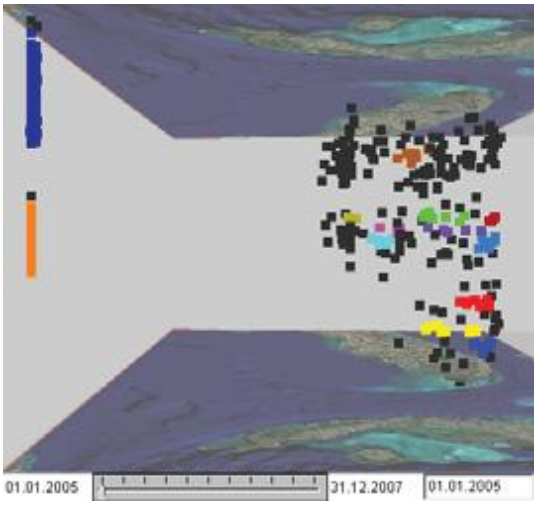

Figure 3.19: clustering by using spatio-temporal distance function (3)

Conclusion

As could be seen by different experiment "explatory analysis" and using visual exploration is an important part of cluster analysis. [12] [19] In fact, interactive

clustering helps us to generate and visualize space time clusters interactively. Another issue is clustering different types of data need to be handseled in *appropriate* way. Appropriate way means know the data and characteristics and devise suitable tools, which here are distance functions.

Chapter 4

Design and Implementation

This chapter describes the implementation of a space-time clustering algorithm based on OPTICS. In the framework of this chapter in the first part the preview is done on the overall approach on the moving to the implementation of OPTICS. Then the uDig software application and Space-Time Cube will be introduced. The OPTICS pseudo code programmed in java is reviewed in the section 4 and in the final section the problems and solutions, the parameters and overall procedure of developing space time clustering implementation will be reviewed.

4.1 Overall approach

The first idea of this work was finding open source implementation of OPTICS and distance functions. Unfortunately there are no open source implementations were found for spatio temporal clustering. We tested an implementation in the area of machine learning named Weka [37] which was not suitable for temporal clustering during to flat table data structure. Therefore, it was decided to implement the original OPTICS pseudo codes from Anksert et al. [5]. Also we have to make our distance function. Although implementation always was a challenging task but make us more accurate in the details of OPTICS algorithm and provide a good framework for further expansion and work in the area of space-time clustering.

Recent implementation of OPTICS-Space Time Clustering- tries to solve the problem of spatio-temporal clustering in human movement data. We used Java uDig to implement the algorithm.

This implementation of OPTICS performs 3D space time clustering. To make the 3rd dimension (Time)comparable to X and Y, user will be asked to enter the relationship between time and distance. This relationship will be defined by speed parameter first and later by $maxT$.

In this OPTICS implementation distance function is needed. Distance function as being discussed before, is used to measure similarity and dissimilarity. In our implementation distance function could be any desired function. As an example here we used euclidian distance in space as default distance function.

In the next section uDig and Space-Time Cube [36] will be introduced.

4.2 Java uDig

According to prior experiences e.g. flocking algorithm which had been implemented successfully before, it was decided to implement OPTICS in the uDig software application, using the JAVA programming language. UDig stands for user-friendly Desktop Internet GIS that is an open source desktop application which could be used either as a stand alone application,or extended by integrated plug-ins.

What has been done here is to add plug-in to the software by using Java code. uDig has advantages which motivated us to choose it:

- User friendly for user especially GIS users
- Desktop located, doesn't need to install
- Good in complicated analytical calculations

Space-Time Cube developed and implemented in the uDig as a plugin by Kraak et.al [36] help to better view and undrstand data and also clustering output.

General framework of uDig software according to Figure 4.1 contains main views like Layer view, Space-Time Cube view and table view. The main toolbar gives the user the possibility to zoom, pan, information query from layer, distance measurement and other ordinary tasks.

Data to be used in the space time clustering need to be in shapefile format by at least three attributes of X,Y(in meter, degree, etc.) and Time(in date, day, hour, etc.). According Figure 4.1 to The shapefile could be shown in uDig as a layer or as a table or in a Space-Time Cube.

In space-time cube window shown in Figure 4.2 an option will be available to perform OPTICS which by using this option the user interface will be available. After setting the parameters and performing OPTICS the progress indicator shows the running of programm.

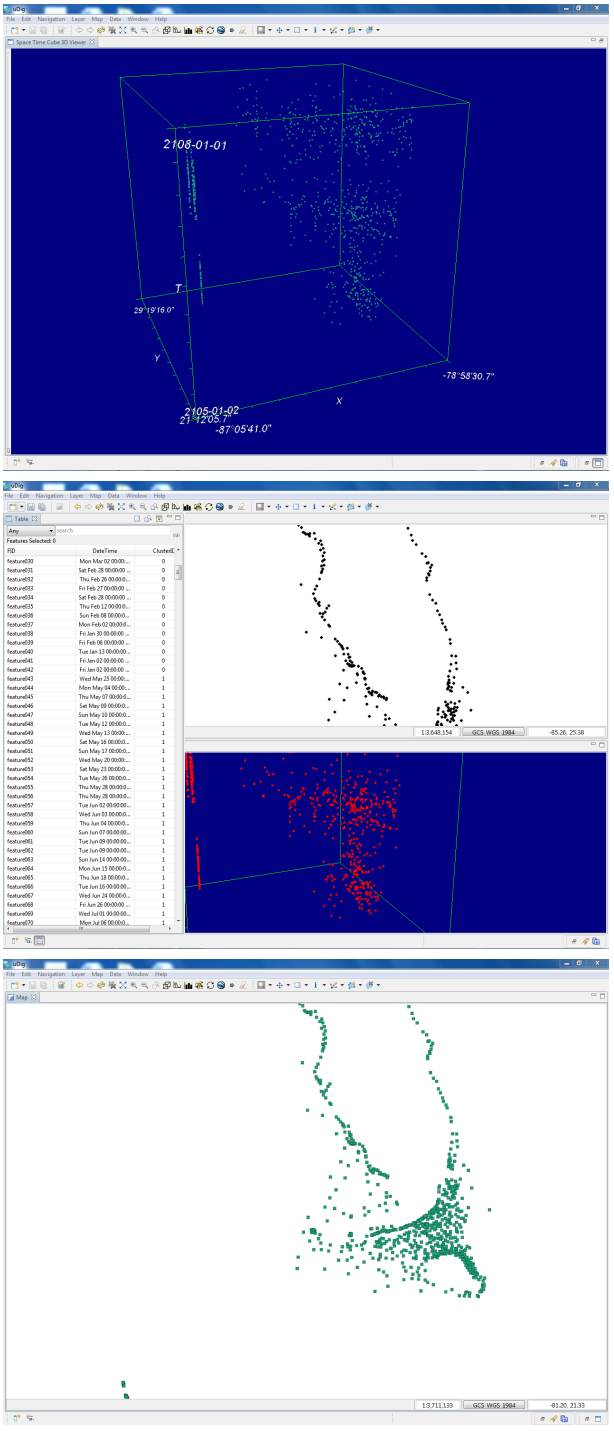

Figure 4.1: Top space-time cube, Middle table and layer and space-time cube view in main window, and bottom layer view

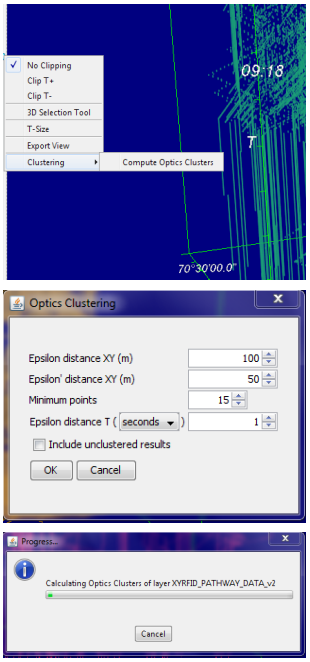

Figure 4.2: From top to bottom, Space Time Cube and OPTICS option, User interface and progress indicator

4.3 OPTICS.java Pseudo-code

After a review on uDig and new option in it which is OPTICS space-time clustering, It's worth to see what is happening behind the scene. We used a plug-in which is programmed in Java named Optics.java[Appendix B]and OpticsItem.java[Appendix A]. The code in Optics.java matches the pseudo-code introduced in chapter 3 section 2.2 as follows:

- **computeOpticsClustering()**:

  This is the main loop, that is first converting the uDig data in the format that Optics needs it, then executing getOpticsClustering(), then converting the data back to a format that is suitable for uDig.

- **getOpticsClustering()**:

  This is pseudocode of Algorithm ExtractDBSCAN-Clustering. As discussed before OPTICS makes ordered file. Ordered file makes an order of each point w.r.t reachability distance of all points. This information alone does not make clusters. In order to make cluster from this information the algorithm ExtractDBSCAN-Clustering gets the Ordered file produced from OPTICS, $\acute{\epsilon}$ and MinPts as inputs. Then it assigns ordered points to the clusters if they were under the $\acute{\epsilon}$ threshold.

- **getOpticsOrdering()**:

This is pseudocode described in Figure 3.7 chapter 3: OPTICS main loop. In this level the algorithm starts by making an empty order file which eventually is filled by processed points. Each unprocessed point goes to the next level.

- **expandClusterOrder()**:

  This pseudocode is described in Figure 3.8 chapter 3: OPTICS expand cluster loop is the main loop of OPTICS(although the name main loop is assigned to the pervious loop). For each point from previous loop, if it was not processed, the algorithm starts to assign neighbors, Core-Distance with respect to MinPts and ϵ. This Object will marked as processed Object and if Core-Distance was available for it, goes to the next level(update()) for expansion of cluster from the neighbors of Object.

- **update()**: This pseudocode is described in Figure 3.9 chapter 3: OPTICS Update seed loop. The main task in this level is to make a queue from neighbors of previous Object which come from pervious level of expandClusterOrder() in order to further expand the cluster. This queue will be made based on Rechability Distance parameter introduced in chapter 3. The aim of this queue is to find the potential objects to be new center object, in order to expand the cluster. This queue will be made in a way that always selects nearer object in the neighborhood of previous center object.

- **getNeighbors()**:

  This is based on the definition of “Ne(p): The e-neighborhood of p", which are all objects within a distance-radius e, including p itself.

- **getCoreDistance()**:

  This is based on Definition 1 page 4 chapter 3: “core-distance of an object p”. According to this definition, for a given ϵ and MinPts in the neighborhood of object p, minimum distance $\acute{\epsilon}$ that makes the object p, core object.

The code in OpticsItem.java is responsible for storing each object and its data, and contains the "distance" function , getDistance() which is implemented as the Euclidean distance in 3 dimensions. This is what enables us to have space time clustering with respect to location(X,Y) and time. The discussion on this issue in section 5.2 of this chapter.

4.4 Developing the space time clustering

During development the problems in the first run of almost every program are inescapable. These problems have different origins like misundrstanding/misinterpreting the concepts, technical problems and limitation of software, etc. Solution of these problems needs an accurate view on the every detail/component of the programm. This section will review some problems in the implementation process.

4.4.1 Problems

Tracking data

The first run of space time clustering was on the GPS tracking data set which contains several paths(Figure3). The results showed that OPTICS basically re detect which points belongs to paths. By manipulating several parameters the results improved but not perfect result. A perfect result would show each path(trajectory) continually in one cluster or more clusters.

This problem shows that our distance function which treats each point independently is not completely suitable for finding or tracking data. The suitable distance function for tracking data is a function that completely follows the paths. Although this is predictable and also reasonable because our clustering tool is in the point level. It will be discussed in the next chapter that how by a simple query from the result of OPTICS clustering it is possible to extract the trajectories.

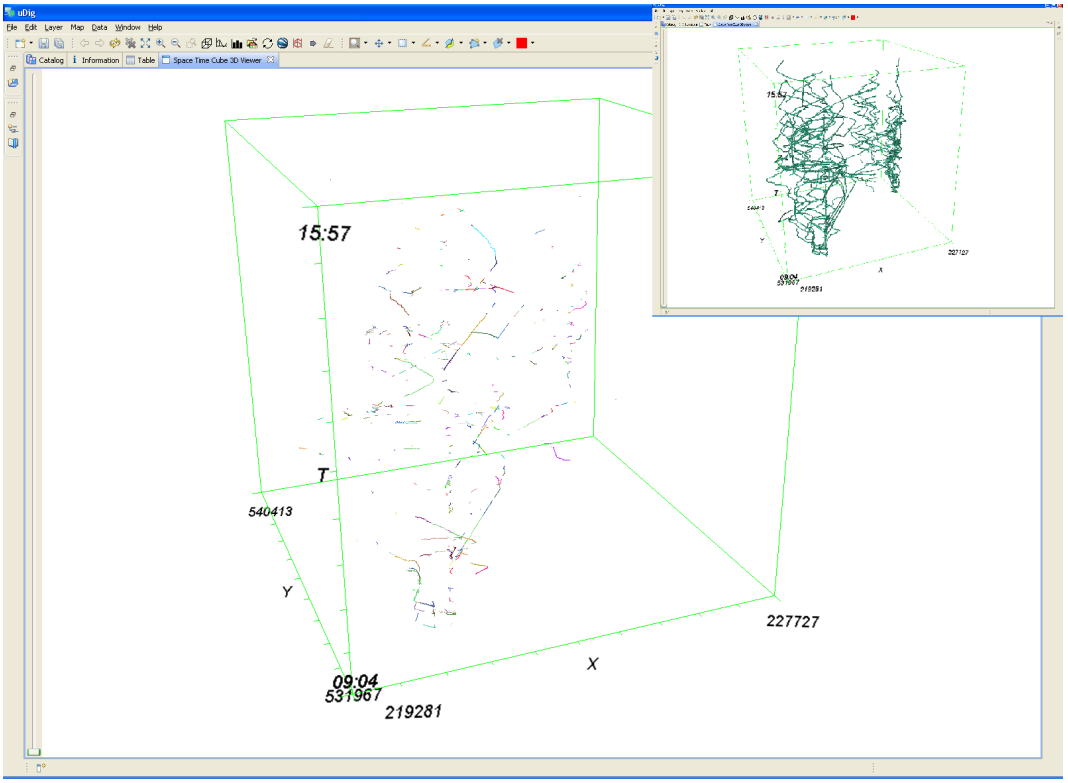

Figure 4.3: Top smaller window shows the real data set and bigger window shows the OPTICS results. Clusters represented by different colors

"Space-time clustering" and temporal clusters

Another data set used for testing was a GPS point data set named migrant boats from the VAST 2008 challenge [22]. This data set contains 917 records,

which have spatial positions of landing illegal migrant boats and time of each records between 2005 and 2007. In the first run, clustering was done successfully but the number of clusters was too high. After assessment, it was discovered that the clustering was done on single layers, layer by layer clustering. In another word algorithm had done clustering in each single time steps.(Figure 4.4) That was not what was intended. This required further work in order to make continuous clusters in location and time (in other words, other words temporal clusters). Theoretically, this could be done by the algorithm, because the distance function could be extended to make a sphere around the Core-object, which could be expanded in time and space dimension simultaneously.

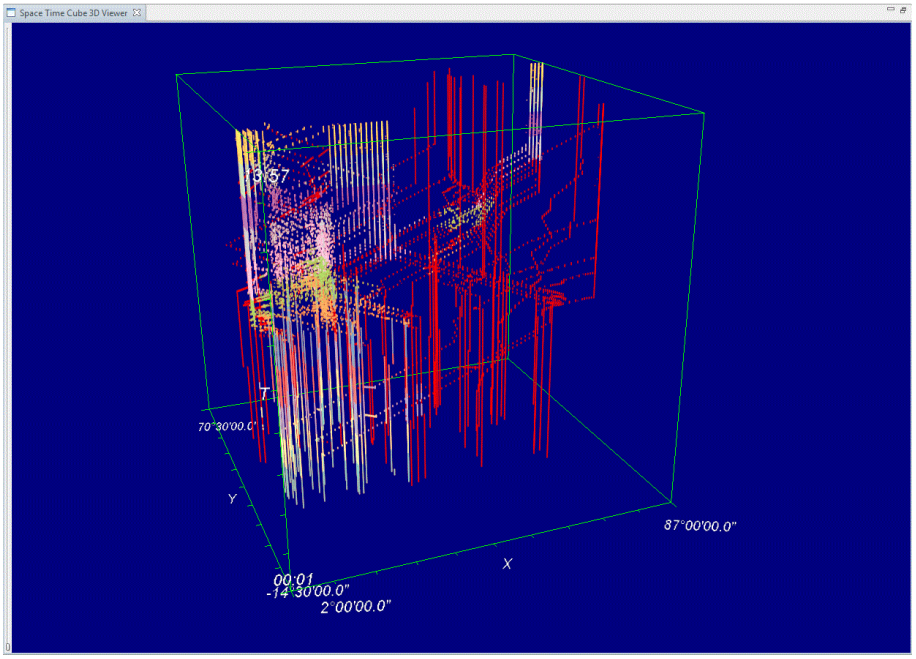

Figure 4.4: Layer by layer clustering result(here for better showing the layer by layer structure many layers merge in one color) the red color is un-clustered data and different colors shows different clusters

First effort was to change the parameters to get desirable results. By several parameter settings some times number of clusters decreased but the main problem still remaind: *There is no cluster expansion in the Z-dimension(time)*. After these experiments we found the solution to this problem was found to be the speed parameter setting. The points in the data set used were expressed in degrees(Latitude/Longitude). In this case speed=1 means 1 degree per second. The ϵ and $\epsilon\prime$ are in degrees too. Data used was in WGS84 coordinate system therefore 1 degree on a map is approximately 111 kilometers in reality. This means by using speed 1(the minimum speed setting) a point could move 111 kilometers in one second! to clarify this issue we picked up two first items of migrant data set (April 24 and April 26). Their spatial distance is less than 0.5 degree according to Euclidean distance. The temporal distance is

172800 seconds(April26 - April24 = 2 days = 2×24 hours = 2×24×60 minutes = 2×24×60×60 seconds = 172800 seconds). When speed was set 1 it means in 1 second, the traveled distance is equal to 1 degree. Therefore 172800 seconds ≡ 172800 degrees. Before that we knew that in this data set 1 degree≈111 kilometers, so speed=1 stands for 19million kilometers travel in space in 2 days which is quite unrealistic.

When ϵ=1, in fact OPTICS algorithm also decides that any items that differ more than 1 second in time, will not be in the same cluster and this caused problem in the first run. The solution is to recalibrate the speed for converting time to distance. The speed value should be smaller than 1 to find temporal clusters.

Let's consider this hypothetical situation:Two first items of migrant data set (April 24 and April 26) and ϵ=1 degree , which means items allow to be in the cluster when their distance is less than 1 degree. By setting speed correctly we want to answer the question of how much is time difference allowed to be? Here 2 days(temporal distance) should be comparable to 0.5 degree(spatial distance). Speed should be 0.5 degree per 2 days, or 0.5 ÷172800 seconds. Therefore speed should be 0.000002893518518 degree per second. This value was not accepted by the entrance in user interface, so by adjusting parts of the Java code, the temporal clusters were discovered.(Figure 4.5)

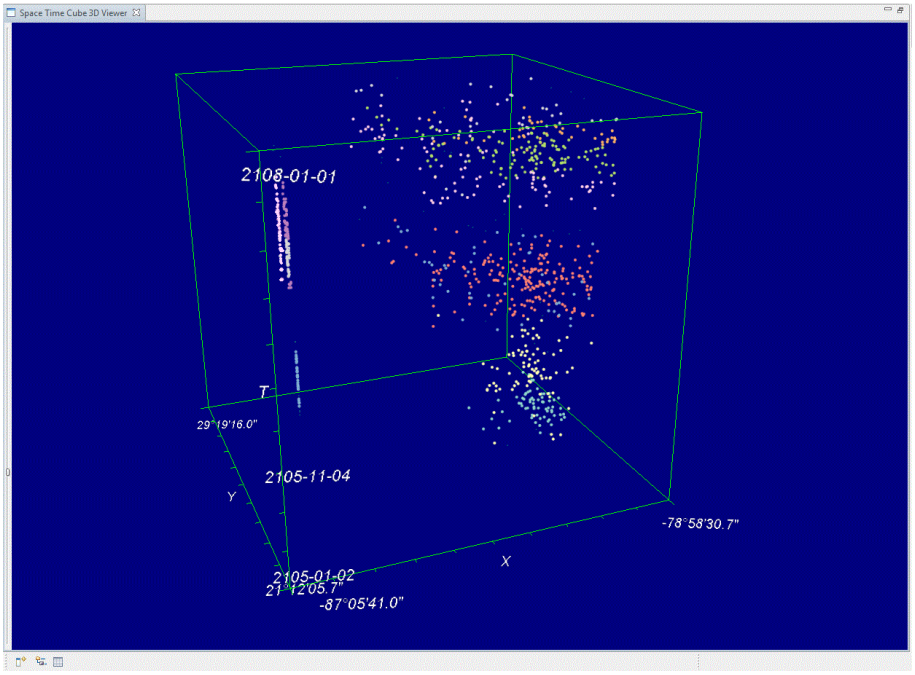

Figure 4.5: The spatio-temporal clusters with recalibrated speed parameter setting

4.4.2 "Space-time clustering" parameters

By clicking this option a user interface will appear to ask parameters which are:

1. ϵ
2. $\acute{\epsilon}$
3. MinPts
4. Speed, *MaxT*

The ϵ determines the area to be searched and $\acute{\epsilon}$ is determines the level of Reachability distance that clusters could be extracted. MinPts is the number of point in theϵ of center object. Speed parameter has an important rule in the relation between time and location in distance function. Figure 4.6 shows different parameter settings and results. Location in geographical point of view generally is defined in meter or degree(latitude and longitude). Time has different metric which mostly is date or hour minutes or seconds. Having time and location in the same formula requires that a relationship is defined between them. This relationship will be held by Speed parameter.

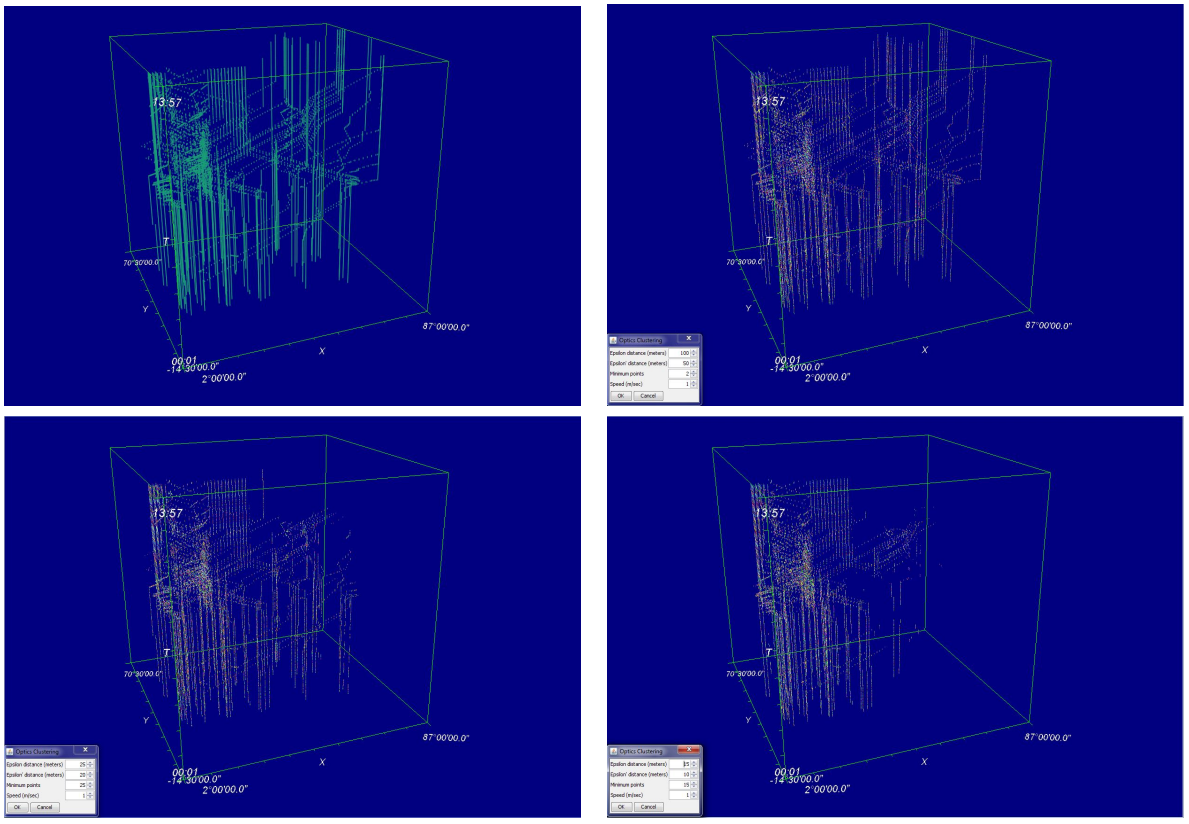

Figure 4.6: Top left data set, different parameter settings and clusters in diffrent colors

After this experiment we found out that using speed parameter is not handy and most of the time causes confusion. Because speed parameter first of all should be set in very low value(for this data set 0.000002893518518) for this

data set and then also desired temporal extension need very accurate and careful speed setting. Interpretation of result based on speed is another problem because before setting we should calculate the temporal extension. This issues motivated us to using time instead of speed in the algorithm. In new version time was used as maxT parameter. In the first effort it was in seconds, by for high extend temporal data set like this data set very big number was needed(2,000,000). In the refinement process of the versions the possibility to set the time as seconds, minute, hour day and week were added. In the next section the whole process of developing the space-time clustering is defined.

4.4.3 Enhancing "space-time clustering"

Besides reviewed problems, we enhanced the software step by step. Whenever the problem or limitation or suggestion we have been faced with, we use it to enhance the pervious version. The final version of Space-Time Clustering is the result of seven pervious versions, which has been edited gradually. Here is the review of them:

First version In this version we need to create a map, with at least one layer of points, with a date/time-attribute or equivalent. The layer should be loaded into the Space Time Cube, with the *stp* option (space-time-path) or the *regular+t* option (3D-points with time). In both two options OPTICS clustering are available by right-click on the vertical scrollbar of the Space Time Cube. If OPTICS takes more than a few seconds, a progress bar will appear, which can even be canceled.

Second version In the first version there are no layer as results. In fact the results are accessible just visually in the layer view and Space Time Cube view. Because the interpretation of results based on the layer and manipulate them in the table view the second version tries to solve this limitation. In the new version it is possible to have access to the layer of OPTICS result and also table associated with that layer. In this version the option *regular+t* should be used. If this layer loaded by<stp> option in uDig and by using different color the result apear in the form of vectors with different colors associated with clusters. However in the large data set it will be hard to interpret. The OPTICS layer could be export as shape file.

Third version Problem of getting smaller values remaind in the second version. The OPTICS form doesn't accept the small value which as we saw for speed parameter is needed. In this version this limitation is considered.

Fourth Version In pervious versions there are not effective connection between the original and OPTICS layer. In other word sometime finding associated attribute in the OPTICS cluster were problematic. This problem solved in the next version. In this version some attribute like number of cluster members are added to OPTICS. Using speed parameter is another problem. In this version "maxT" is replaced to speed. This parameter was in seconds. Very big parameter are accepted in this version (1,000,000). A technical problem happened in the closing OPTICS form in

this way that when closing the OPTICS form the programm runs. This problem is solved in this version.

Fifth version When running the OPTICS got too long(For example in RFID data set) the progress indicator was diapered. This causes doubt in working the OPTICS or not. Version five solve this problem.

Sixth version In the sixth some important enhancement were done. First OPTICS layer contains all the attribute of original layer plus ClsterID and ClusterSize. This layer is completely independent form original file therefore it could be save or edited. This layer gives the user the possibility to event includes uncluttered item from original layer to OPTICS layer to do this the check box which added to OPTICS form should be fill. The clustering form automatically shows the unit of input map layer("o" for degree and "m" for meter). Progress indicator is improved by the name of layer which OPTICS works on it.

Seventh version The final version calculates quality measurement parameter VNND [33]during clustering process.

Chapter 5

Test and Evaluation

In this chapter the Space-Time Clustering and its distance function will be tested over three different data sets. GPS, RFID and Cell phone tracking data are the data sets with different characteristics. In each section after reviewing the data set and its characteristics, parameter will be set for them and then summarized at the end of each sections. Also appropriate distance function will be suggested for data sets. At the end possibility of extracting trajectory data from existing point based implementation- Space-Time Clustering - will be discussed.

5.1 Introduction

Clustering algorithms are a solution to handle large amount of data that have been introduced in different areas and disciplines including human movement data. This type of data are in fact sampled points. Clustering human movement data it could be considered at least three X, Y and T(time)dimensions to create clusters. Density based clustering was considered as a solution to clustering in this area(spatio-tempral movement data), because of some properties of these kind of algorithm such as good efficiency, the ability to deal with noise and arbitrary shape clusters. [16]. As been discussed OPTICS is one of the density based algorithm of clustering which has been used successfully for clustering spatio-temporal data [3] [45]. In the pervious chapter it has been shown how this algorithm implemented and do the clustering successfully. Problems solvation and enhancement were done to the implementation. The final step will be test and evaluation algorithm by real data set in this chapter. Before that some issues should be considered.

In order to use OPTICS properly in clustering spatio-temporal movement data two main issues are considerable:

1. Distance function
2. parameter settings

Distance function as discussed in Chapter 3, could be seen in two main views:*spatio-temporal query* and *mathematical formula*.In mathematical view, the focus will be on distance function related to type of data, e.g. point data and trajectory

data. The OPTICS parameters were discussed in chapter 3 in detail. These parameters are the key variables in the result of clustering. On other hand, the input data should be considered in clustering. Different data sets have different characteristics like:

- Spatial and temporal extent: How long and how far data spread in space and time
- Resolution(granulity)of data
- Type of data: point data or trajectory data

In order to deal with different types of data, using different distance functions is possible. Andrienko et.al [3] used different distance functions in diverse types of spatio-temporal movement data. Different data sets with different characteristics need proper parameter setting to deal with these characteristics which means using parameters and distance function together it is possible to make the OPTICS a flexible tool for clustering different type of data by different characteristics.

In pervious works [4] [45] the focus was on trajectory data. Their distance function works based on trajectory data. We want to test the spatio-temporal distance function-3D Euclidian distance- in point data e.g. focus on the point data properties and behaviors on the term of point data. The details of space-time distance function available in chapiter 4.

In the framework of this chapter first an overview on distance functions in point data and trajectory data are done. Then space-time clustering in three different data sets with different characteristics will be tested. These data sets are: Cell phone tracking data, the GPS data set and RFID data set. First by using our distance function(3D spatio-temporal distance)parameter will be set to each data set then for each data set distance function will be suggested based on data type. At the end of this chapter it could be seen how trajectory clustering could be derived from our distance function.

5.2 Experiment

The whole procedure in this experience is loading data layers to uDig, running the OPTICS and interpretation of results based on structure of data sets and validity indices. Interpretation of result and parameters setting are done by visually interactive method and introduced validity indices. The combination of layer view of data, Space-Time Cube view, clustered result and table associated to the clustered layers allows to better interpret the results and therefore, adjust the parameters.

5.2.1 Evaluating quality of clustering results

The validity index to qualify the result of clustering was used. Variance of the nearest neighbor distance(VNND) [33] searches in the local environment of data and calculates the nearest distance of each point w.r.t other points in

the cluster. Then for each cluster the variance is calculated and summation of these values will be the VNND. This index compensates the weakness of other indices, which are unsuitable for arbitrary shape clusters. The lower amount of this index shows the more homogenous cluster. The mathematic formula according to [33] are:

The smallest distance between each point in the cluster:

$$dmin(x_i) = min_{x \in C_i}(d(x_i, x))$$

The average of minimum point distances in cluster:

$$\overline{dmin(x_i)} = \frac{\sum_{x_i \in C_i} dmin(x_i)}{sizeC_i}$$

And the variance will be defined as:

$$V(C_i) = \frac{1}{sizeC_i - 1} \sum_{x_i \in c_i} (dmin(x_i) - \overline{dmin(x_i)})^2$$

And finally VNND for all clusters is:

$$VNND = \sum_{i=1}^{n_c} V(C_i)$$

here x_i is the member of cluster C_i and $dmin$ is the shortest distance to the neighbors in the ith cluster and n_c is number of clusters.

One of the advantages of this index in addition to deal with arbitrary shape clusters is calculation the index during the clustering process which makes the calculation faster and therefore causes less running time. Implementation of this index was done in the final enhancement of the OPTICS implementation(See Chapter 4 section 4.3)

5.2.2 Points and trajectories

Practically in the spatio-temporal movement data, It is possible to consider two main types of data:

1. Point data
2. Trajectory data

Point data has at least three main attributes(Latitude, Longitude, Time)which define the position of the object in space and time. It means the position of an object in space and time is independently defined from other objects. One of the best models to show point data in this sense is Space-Time Cube [36] which uses the third axes as time. In the Space-Time Cube, the point data has a unique position in the 3D space.

Trajectory data is the second type. Here for a given ID there are three attributes(X,Y,Time). The relation between the ID and attributes could be considered as a function. In such a function having ID in given time the location will be derived. In other words, for a given object this kind of function assigns time

to the location. Such function is named as trajectory [45]. The visualization of the trajectory are shown best by vector in the Space Time Cube.

Having these types of data in mind, it could be categorized distance functions in the same two categories. Therefore, the distance function for point data and distance function for trajectory data are considrable. Distance functions for trajectory data were developed by Rinzivillo et al and Andrienko et al [45] [3] [4]. They developed the library of distance functions for trajectory data. In their work as discussed before spatial and temporal characteristics of objects were used independently or in combination to measure the distances.

Different distance functions with different complexity for trajectory data according to [45] are:

Common source distance function compares trajectories to see if they have the same origin. Therefore, clusters will be formed based on the start point clustering. The distance between the origins is spatial distance in the surface of the earth.

Common destination distance function is the same as common source distance function but the destination point of trajectories will be used for clustering.

Common source and destination combination of two described distance function leads to this distance function. In the simplest form, the measurements are done in both source and destination and then by averaging them final distance will be driven. In the more advance approaches not only the distances in the source and destination are computed, the distances in the mid points are calculated too. There are different strategy to finding midpoints, like k points by time which finds the desired number of midpoints by the user in the way that the time interval is equal and k point by distance does the same thing but the distances keep equal by choosing k midpoints by the user. Other strategies are time steps and distance steps which user defines desired time step and distance step for being midpoints respectively.

Route similarity The problem of reviewed distance function is that they work in the ideal trajectories e.g. the same durations, good sample intervals and etc. But in reality most of time these kind of data is not available. This distance function provides the possibilities to measure the similarity of incomplete trajectories and trajectories with different time duration. the distance function route similarity search for the nearest point in two under comparison trajectories and in the case of not matched positions the penalty distance will be calculated.

Route similarity and Dynamics distance function work like the rout similarity but compensate the weakens of it which is dealing with temporal aspect of trajectories. This distance function search first define the relative time then by considering this time in two trajectories search for nearest point in position.

The described distance functions are suitable for trajectory of movement data. The second category of distance functions is distance functions of point data. There are not many implemented distance functions in this area. The spatial and spatio-temporal distance function used by Andrienko et al. [3] are the example of such distance function swhich measure the distance of each point with respect to other points in 2D or 3D space respectively. Our distance function is also working in this area. As discussed before our distance function work based on the euclidian distance formula in 3D space. It computes the distance between points in time and space and combines them in the 3D Euclidian distance. In our distance function, each point separately measured against the other points.

5.3 GPS data

GPS(Global Positioning System)is one of the Global Navigation Satellite Systems, which provides the possibility positioning. This technology developed by United States for military purposes first and then became the worldwide system. Nowadays, GPS technology is used in different areas like navigation, orientation, tracking and even high accuracy positioning. [43] One of the interest of many scientists in different disciplines is tracking and analyzing moving objects. Tracking the moving entities means finding the location of moving object in different time steps. This function of GPS system provides useful tool to collect spatio-temporal data. Human movement is one of the main study in this field(collecting spatio-temporal data) like studies of individuals, crowds, pedestrians, location based services(LBS) and urban growth to name a few. [51] [47] By advancing the electronic technology different devices were developed in this area like data loggers, PDA, GPS navigation devices, Mobile phone with GPS etc. These devices provide a valuable source of spatio-temporal data.

Most research has focused on increasing accuracy and developing devices rather than application based study. These researches provide the possibility to positioning with high accuracy. For example, one of the methods of positioning named deferential GPS or DGPS makes the positional accuracy less than meter for different purposes, even navigation. In the DGPS navigation system, the moving object corrects its position by aid of bench mark stations, which accurately positioned before. In spite to the power of GPS in providing both spatial and temporal accuracy, the main disadvantage of this system is positioning in the city area specially in the area with tall buildings and also indoor positioning. Results depend on having suitable angle that receiver could see the satellite. Therefore, in urban area with tall building it's problematic to use GPS. In other hands GPS gives the possibility to have the point in the desired sample interval and distance interval. Searching for signal is one of the problems of GPS receiver, this process for high accuracy measurement is time consuming, and causes mis-positioning.

The first data set used for testing, and evaluation is GPS data from illegal migrant boat [22]. this data set contains 917 records between 2005 until 2007. Each record has some attributes. We selected three main attributes(X,Y,DATE)for

simplicity. The characteristics of this data set are summarized in the table 1.

5.3.1 Parameter setting

The first experiment on this data set back to use first run of the program when the layer by layer clusters were happened(Caphter 4). Different parameter settings lead to results with the different number of cluster. In some cases, the number of clusters got high but the temporal clusters didn't happend. Different parameter settings and combination were tested to get good result. First two ϵ and $\acute{\epsilon}$ parameters were changed until finding the good result. The good Results here means clusters, which tailored to the structure of data. After setting two parameters the other two parameters(Speed and Minpts) were changed. As expected from pseudo codes by changing the Minpts number of cluster varied. The higher value leads to fewer clusters and the smaller value leads to high number of clusters.

The problem of temporal clusters still had remained. After some experiments, the suitable values for ϵ and $\acute{\epsilon}$and Mipts parameters are found. By changing the speed parameter and keep other parameters constant the result doesn't change. As discussed in Chapter 4 that speed parameter should be set less than one in this data set. To review, Space time clustering does the clustering both in space and time. We follow euclidian distance function formula in 3D space. Therefore, the hight of the sphere made by the time should be comparable to the circle made by location.

The migrant data set bounded in the coordinates (-86.8, 21.2) (79.2, 29.3). From this coordinates it is found the highest distance(11.10) between points in the data set to set the ϵ parameter. The process started By setting the first parameter setting as ($\epsilon = 100$ and $\acute{\epsilon} = 50$, Minpts=15, ,speed=1) by changing the speed parameter in the best situation one big cluster with 890 members and other 14 members in the noise cluster(under threshold) were achieved. Thereupon by setting ϵ as the very high values one big cluster for all the data set was achieved. Therefore, the following hypothesis is driven from the experience:

> *if the ϵ and $\acute{\epsilon}$ were too big with respect to extent of data set almost all data points were arranged to one big cluster*

The experiment was continued to provide proof to the hypothesis, by setting the ϵ and $\acute{\epsilon}$ (60, 50, 40). The results proof the hypothesis as they didn't change considrably with respect to primary results. It should be noted that although all data assigned to one cluster but in fact, temporal cluster is driven with big size. Temporal cluster means member of cluster expands in time(/space) in different time steps. In another word algorithm is ok, but the parameter setting is not sufficient.

From the hypothesis, it turned out that smaller values for the ϵ and $\acute{\epsilon}$ are needed. Therefore, testing small values was started and after various testing and combination of parameters the parameters set as:$\epsilon = 15$ and $\acute{\epsilon} = 5$, Minpts=5 ,speed=0.00000289518518). By this parameter setting 32 clusters were achieved which some of them were noises-cluster under the threshold and with small members. It was a good result in compare to previous results.

One of the problems in clustering algorithms is parameter setting. For this reason, the structure of data should be considered. It would be a good evaluation to see how much clusters match to the structure of data. Generally, in layer view the structure of this data set contains two divided parts(Figure 5.1):

1. The big bulk of points in the east of map
2. The small number of points int the point the southern-west corner of map

The bulk of points in Space-Time Cube, contains too many points spread in all time duration. The small points contains three columns of points which expand in 2 different time periods. In order to have reasonable parameter setting three conditions were considered.

> **Criteria 1**:*the parameter setting should be in the way that put two bulk of data into two different clusters*

This is the first and simple criteria in order to start parameter setting. By different experiment and combinations of parameters the group of parameters which satisfied the first criteria were chosen. The second criteria is temporal criteria. As mentioned in the structure of data, there are three columns, which two of them are in the same time range- from 20/03/2107 until 29/12/2107 -and the third one is in different time range-From 27/04/2106 until 25/11/2106. [1]

> **Criteria 2**:*the parameter setting should be in the way that divide two columns with same time duration from the third columns with different time duration*

The filtered parameters from the first criteria go to the second criteria and if the criteria 2 was satisfied the final criteria will be considered. The final criteria is the spatial accuracy condition after two point arrows expanded in different times were divided from the third one, it should be seen the possibility to divide these two columns which have the same time range. This criteria are true for the big bulk of data in the east of the map.

> **Criteria 3**:*the parameter setting should be in the way that divide two columns with the same time expansion from each other*

By satisfying third criteria we could say parameter settings are tailored to the structure of data because both temporal and spatial clusters were divided from each other.

[1]The spreadsheet data was in the day. Therefore, when uDig convert it to years the data automatically shift 10 years forward.

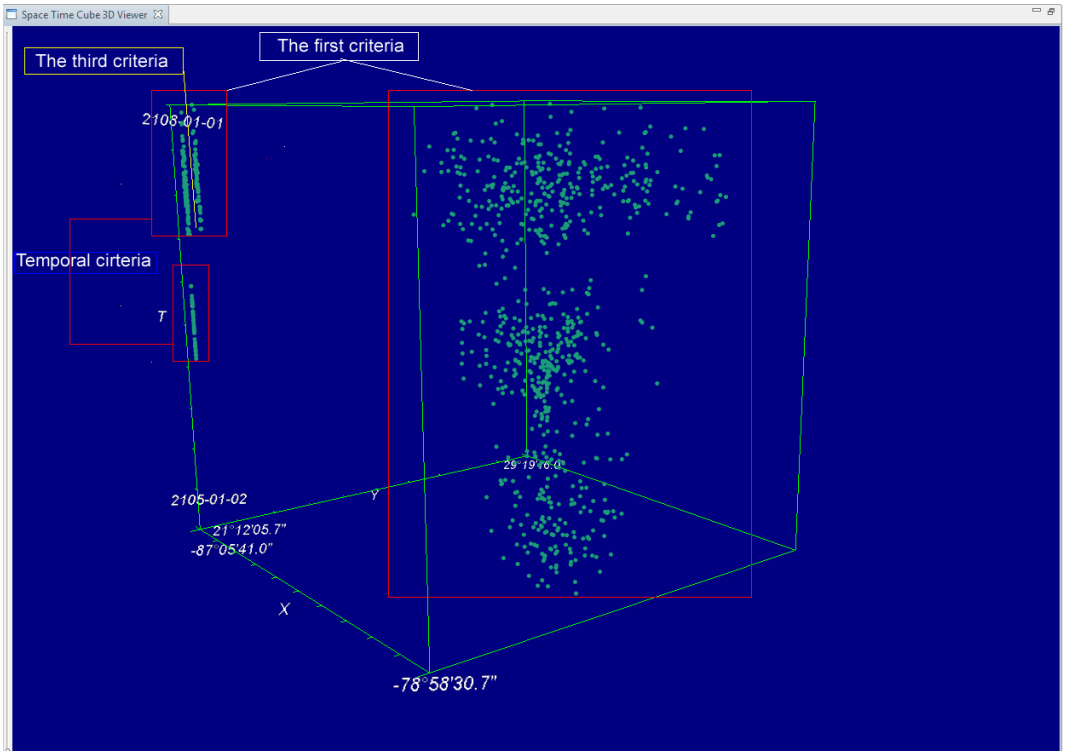

Figure 5.1: Three criterions and the cluster results

After finding the best parameters fit for the criteria some assessment like number of clusters, visually see the expansion of clusters were done to find the best parameter among acceptable parameters. In fact, some kind of interactive method and visually analyzing the results was done here. The parameter suggested for this data set was:$\epsilon = 15$ and $\acute{\epsilon} = 5$, Minpts=5 ,speed=-0.00000289518518). After removing the noises 20 clusters were found. There were two big clusters with 206 and 200 members and the third big cluster had 81 members, by satisfying the third criteria, we could see the different among the cluster of three columns. The running time of the algorithm was very fast(several seconds)due to the small number of records.

By setting the last version of our implementation by $maxT$ parameter, more accurate results were achieved rather than speed parameter setting before . The biggest cluster has 126 members and expands in the center big bulk of data. The second big cluster contains 108 member and expands more in the big bulk of data. The third big cluster belongs to the one of the columns in the corner of map by 106 members. Another clusters have less than 60 members. The VNND parameter for all individual cluster are less than 1 which shows homogenous clusters. Figure 5.2 shows the result of clustering with appropriate parameter setting. In Table 5.1 the parameter setting and some measurements like number of clusters and average of VNND for all clusters is shown.

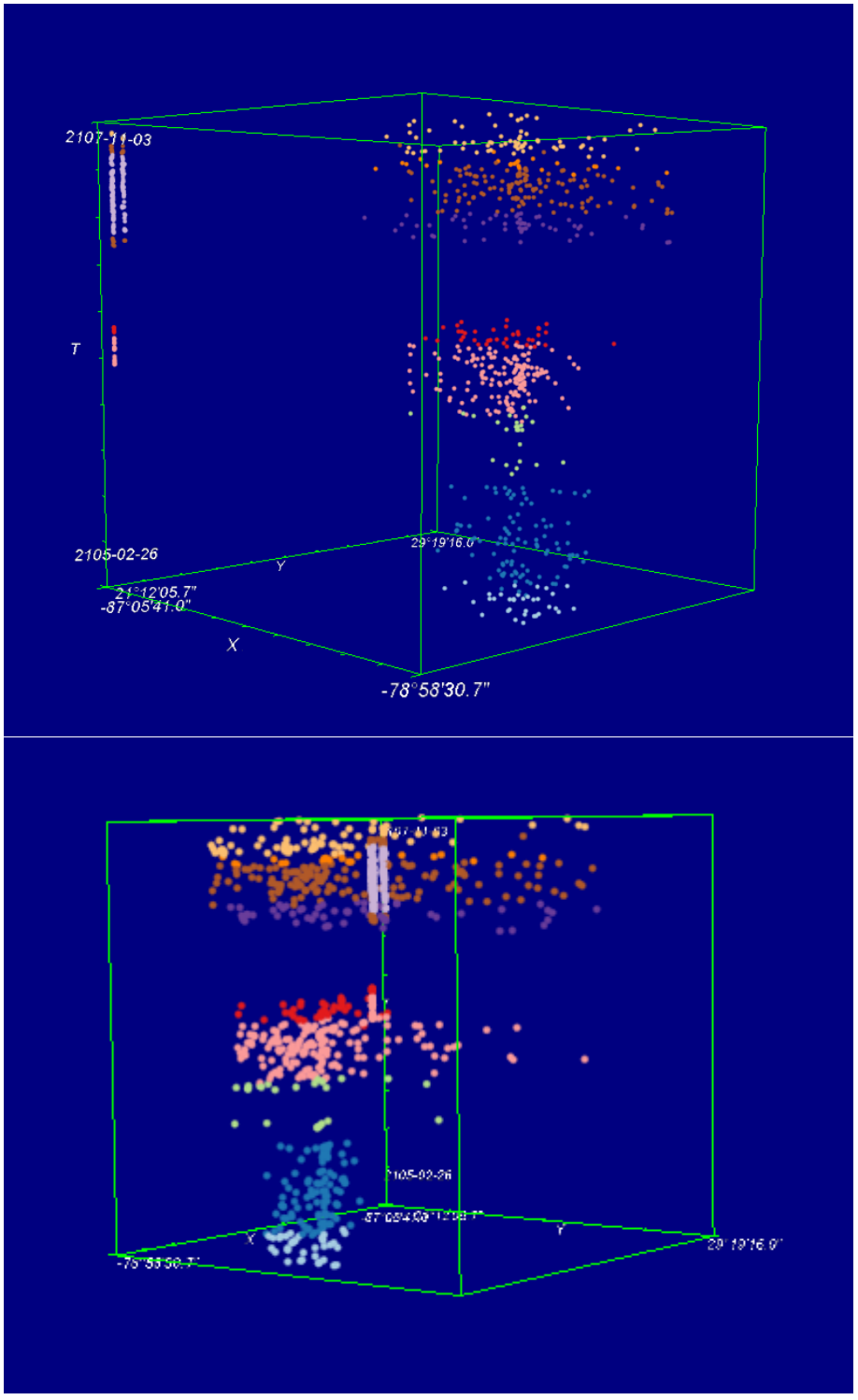

Figure 5.2: The spatio-temporal clusters with appropriate speed parameter setting

Table 5.1: Parameter setting for GPS data

	ϵ	$\acute{\epsilon}$	Minpts	MaxT	Number of Clusters	VNND	Run time
1	15	5	5	20 days	17	0.414288	00:00:01
2	100	50	15	15 days	3	0.64878	00:00:01
3	50	25	15	30 days	15	1.69	00:00:01

5.3.2 Distance function

The structure of this data set is the high resolution positional data and extend in the big time and big region. The structure of data is in the time of landing and interdiction of illegal migrants is unsuitable for making trajectory data. In other words, there is no reasonable continues route or trajectory could be extracted from the data set. Only snapshot of data are available. Therefore the distance function used in the second category (point data distance function)is more useful in this data set.

Table 5.2: Summarized table of migrant data set

GPS data		
Data properties	Spatial Resolution	High
	Temporal Resolution	Low
	Spatial extend	High
	Temporal extend	High
Data Type	Point Data	
Distance functions	Common source/destination	
	Common source and destination	
	Route similarity	
	Route similarity+ dynamics	
	Space time distance	✓

5.4 RFID data

RFID (Radio Frequency IDentification)is referred to a system which used to tracking/positioning object. The principal work of this system is like barcode scanning. Each RFID device should be scanned by the scanner to be positioned. RFID system consists of three main parts:scaning antenna, data interpreter and RFID device(tag). The scanning antenna sends the radio wave to the RFID device, the device responds to the antenna and the information coded on tags, which includes time and ID and other attributes processed in the data interpreter. This is the simple schema of RFID system. The accuracy of RFID devices are high generally and depend on the configuration of sensor(antenna) network and radio signals. This system in contrast to GPS system is suitable for indoor positioning, for example, inside a shopping mal, parking or even in the city. The advantages of this system in addition to indoor positioning, is using very light

tags, which are very cheap, don't need to be directly scanned(like barcodes) and capable of be tracked in everywhere as long as the network was available [40].

5.4.1 Parameter setting

The RFID data set used here is named evacuation traces from the VAST 2008 challenge [22] which contains the regular samples of people position, which wants to leave a building in an emergency situation. It contains 67797 positions from 81 persons in 837 time steps. The spatial and temporal resolution is high and the spatial extent is small(building level). The data set is surrounded by the rectangle from (2,1)to (87,55). The largest distance between points is 100.7m. Here the position of people because of the sensor positions has a regular structure.

First very big parameters were set as the initial experiment to cover whole data set. For example, by using ($\epsilon = 100$, $\acute{\epsilon} = 50$, Minpts=5 , maxT=300) almost entire data are assigned clusters. There are big clusters, which were not appropriate. This data set is an example of small scale data set, therefore, here the behavior of data in micro-level should be considered. This means by such a good resolution data it should be possible to find clusters, which are near in the distance-for example less than few meters- and time(in seconds). Otherwise by using large parameter, in fact, the micro-scale behavior will be lost. Another issue in choosing large values for ϵ and also maxT cause calculating too many distances, which means increasing the runtime. In this data set by $\epsilon = 100$ and maxT=500 in the CPU Intel(R) Core(TM)i5 2.67GHz and 3 GB ram memory takes one hour which is too long. Therefore, smaller values were tested. By several parameter settings, two parameter settings were chosen. The first one is ($\epsilon = 3$, $\acute{\epsilon} = 1.5$, Minpts=5 , maxT=300) which after removing the noises, 40 clusters were driven. The value of VNND for the biggest cluster was 0.0147716, which is quite suitable and show the homogeneity of the cluster. Another parameter setting is($\epsilon = 5$, $\acute{\epsilon} = 2.5$, Minpts=5 , maxT=300) which the result was almost the same but the clusters got smaller. By setting small values for parameters(item 4 in Table 5.3)although VNND value is low but many small clusters are resulted. Table 5.3 is summarized parameter settings for the RFID data set.

Table 5.3: Parameter setting for RFID data

	ϵ	$\acute{\epsilon}$	Minpts	MaxT	Number of Clusters	VNND	Run time
1	5	2.5	5	300 seconds	31	0.0776	00:03:51
2	3	1.5	5	300 seconds	40	0.0147716	00:03:28
3	50	25	15	300 seconds	192	3.429216	00:29:51
4	2	1	15	120 seconds	267	0.015	00:02:58

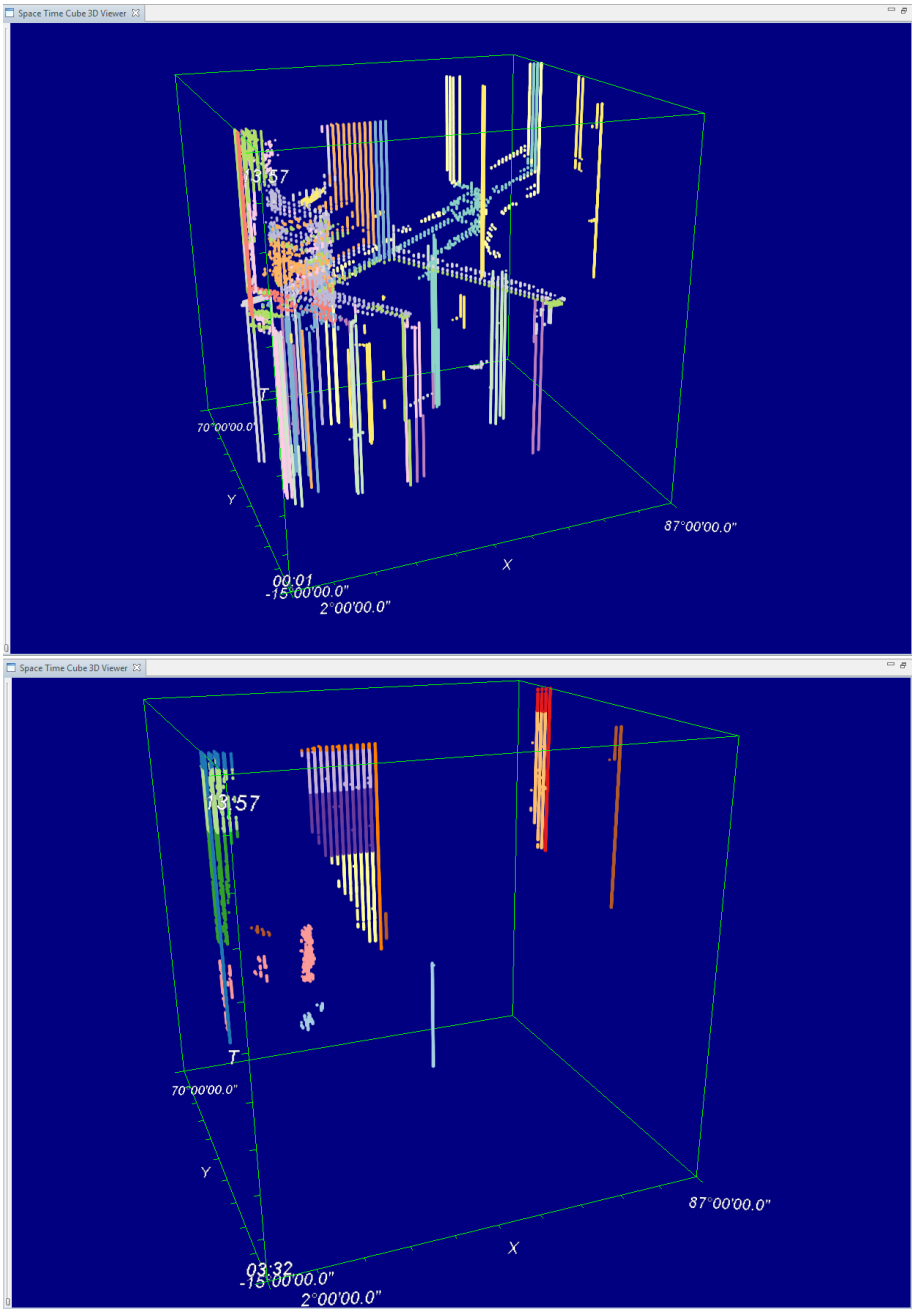

Figure 5.3: The spatio-temporal clusters with:top appropriate parameter setting, and bottom small parameter setting(item 4 in table 5.3)

5.4.2 Distance function

According to the structure of data, trajectory data will be easily driven. Therefore both point and trajectory distance functions are applicable here.

Table 5.4: Summarized table of evacuation traces data set

RFID data		
Data properties	Spatial Resolution	High
	Temporal Resolution	High
	Spatial extend	low
	Temporal extend	low
Data Type	Point Data	
Distance functions	Common source/destination	✓
	Common source and destination	✓
	Route similarity	✓
	Route similarity+ dynamics	✓
	Space time distance	✓

5.5 Cell phone tracking data

Mobile positioning is a valuable resource of geographical studies. By this technology large scale data will be acquired. Two methods which used in the cell phone tracking are passive positioning and active positioning. This kind of data is saved in the memory of operators. The procedure is provided here. For the active positioning service provider or operator asks the cell phone holder by Sms or call to record the positioning or not. Therefore, the user has freedom to be positioned or not. In other hands in passive positioning doesn't ask the user to positioning. The position of user automatically is saved in the memory of the operator. Different techniques are used in order to find position the user. Some cell phone devices have been equipped with GPS loggers. In this way positioning and tracking are available. The technique of positioning is depended on the system which has been used by the operator like Nokia or Simens systems. Using different techniques impact the accuracy of positioning. [1]

Another technique which has been used widely in passive mobile positioning(cell phone tracking)is cellular network positioning. Cellular network includes many antennas, which provide the signal to the cell phones. The distribution of these antennas is varied in different areas. For example, in a city the density of antennas is higher than the suburban area. Whenever a user making a *call activity* the position of the antenna which provide the signal, records in the memory of the operator. *Call activity* is different activities, which make the connection to the service provider like sending sms, making a call, turn off/on the phone or using GPRS and location based services. Therefore, positioning in this area depends on *when* and *where* making call activity. This is important for us in spatio-temporal clustering because *when* defines how often making a call activity or temporal resolution and *where* defines the spatial resolution-in region with high density of antenna better spatial resolution is available rather than the sparse antenna. In this way clustering algorithm should use appropriate distance function with good parameter settings. The positional accuracy is different from 100m to 1000m in cities and 450m to 2Km in suburban areas

but in the sparse network it decreases significantly to 1.5Km to 20Km. [1] To use this data it's important to know where are they coming from?

The cell phone tracking data set is used here is a subset of data that used in an Estonian study [1] contains 8719 records from 500 user in 20 hours(from 4 to 24). Each use has been assigned a random ID to identify. The data set is the example of large scale data, which expand in a very big area. The farthest distance between points is 352155/68 m (max distance of data set). The temporal extend is low in less than one day.

5.5.1 Parameter setting

The parameter settings for this data set are divided in two parts:parmaeter setting by *speed* parameter and parameter settings by *maxT* parameter. The process of finding appropriate parameter settings is empirical and is based on interactive testing of different parameters. In this procedure, the space-time clustering run in the data set in uDig. The result will be evaluated visually in the Space-Time Cube and further process like removing noise from results will be done in ArcGIS on the attribute table. The final filter would be VNND.

In the pervious chapter the speed parameter discussed. First experiment on this data set was on the version with speed parameter. The process started by this parameter settings:$\epsilon = 100$ and $\acute{\epsilon} = 50$, Minpts=15 ,speed=1 which finds two clusters. The temporal clusters were found by this parameters(despite the GPS data set which the speed parameter should be very low)but the temporal differences between the members of clusters were under the minutes. It turns out later that in fact by this parameters,it is still in doubt whether temporal clusters is obtained or not because the algorithm with time version could not find temporal clusters by setting time less than 30 seconds. Another issue was the same position of people in the data set. In the clusters result the position of several people were the same and therefore go to the same clusters. This is unrealistic because several people couldn't occupy the exact same position. But by considering how cellular network position the cell phones the answer will be defined. In fact these people in the cluster in reality don't have near position and just because the have been gained signal by the same tower, they have same positions. This aspect of cell phone tracking data set effect the result of clustering considerably. By decreasing the MinPts parameter the number of clusters increases and the clusters expand in Minutes. Using Minpts=5 and same parameter settings gave 35 reasonable clusters. We start another exploration by setting ϵ big values(1000,500and 200)the number of clusters were too high(5983,1123).

Another experiment was changing speed parameter. By increasing speed the results don't change considerably. By decreasing the speed the number of clusters decreased but the members of clusters increased. It means more members each cluster has. This is the good sign toward avoiding layer by layer cluster and therefore, having temporal clusters. By gradually refitment of results it is found these parameters good for this data set:$\epsilon = 50$ and $\acute{\epsilon} = 10$, Minpts=5,speed=00002. By these parameters 8361 features clustered in 322 clusters Which 216 of them were reasonable clusters and rest of them were

noises.

As discussed before the difficulty in speed parameter setting and interpretation of it, motivated us to use the time parameter instead of speed parameter setting. In the new version(see chapter4 section 4.3) the *maxT*replaced to the speed parameter. Using this parameter allows to define temporal clusters in desired time(seconds, minutes, hour day and week). This parameter setting is handy and easy to interpret. First experiment on this data set by new maxT parameter was done as follows. Firstly, the dense area visually was found and the approximate bounding box of this area-dense area- was drowned. This bounding box was used to find the farthest distance in the dense area which is 23870m. The ϵ parameter used to define searching area for finding core objects. It should be big enough to cover all points. However, having very big ϵ although insures that no point is left but increases the costs(the number of calculated distance increases, in consequence, the running time increases too) . By using this distance as ϵ parameter, It is insured that no points will be left, and also the running time is optimum. The second epsilon($\acute{\epsilon}$) are set 10000(near the half of ϵ). The number of members in clusters set to 30(Minpts)and the maxT parameter set to one hour. Setting time less than minutes as discussed before doesn't lead to temporal clusters in this version. It should be noticed that by setting hour as maxT the temporal clusters with difference in seconds will be driven, but in fact, the seconds don't have an impact on the calculation process of clustering in this data set. The number of clusters were 174. There were three clusters with more than 1000 members. The biggest cluster contains 2443 members. This clustered distributed in the dense area(bounding box)and the time expansion is from 7:53 to 13:54. The second big cluster has 1694 members in the approximately in the same area but with different time duration from 12 to 19. The interesting pattern here is the location of members in described two clusters are near to each other but the time expansions are different. The situation was the same for the third big cluster with 1122 members expanded from 11:55 to 15:24. This issue brings us to this hypothesis:

> *By increasing the maxT parameter the cluster with the near location should merge to each other and the number of members increases.*

By the same parameter setting and increasing maxT to five hours the three clusters in pervious experiment mainly merge to the one big cluster with 6330 members. This cluster expanded whole daytime. Four main clusters were found in other areas with relatively small members(100 500 members). The run time for this parameter setting was about 30 minutes by using CPU Intel(R) Core(TM)i5 2.67GHz and 3 GB ram memory which relatively was a big run time(normally clustering in this data set takes under minute time). By setting time parameter to 10 one big cluster was found, which prove our hypothesis.

By setting parameter in low values $\epsilon = 100$ and $\acute{\epsilon} = 50$, Minpts=15, one hour the number of clusters increases and local clusters were found. The hypothesis is still true here. By increasing time to two hours the cluster members increases (from 1833 to 3204 total members).

As discussed before low resolution data are suitable for study macro-scale problems. The parameters should be set by the consideration of the macro-

scale problem. Therefore, it's not reasonable to use local parameter settings e.g. parameters with low values. For example, using $\epsilon = 5m$ or MinPts=5 in this data set are not an intelligent choice. Because low resolution cell phone data set is not accurate enough. These parameter maybe give the results in specific situation but there is no grantee that they are match to the reality.

Table 5.5: Parameter setting for Cell phone tracking data

	ϵ	$\acute{\epsilon}$	Minpts	MaxT	Number of Clusters	Mean of VNND	Run time
1	23870	10000	30	1 hour	174	696735.68	00:01:03
2	23870	10000	30	5 hours	5	18393241.37	00:14:23
3	100	50	15	1 hour	103	3.971343	00:00:04

At the end of exploring parameters in this data set it should be noted that our distance function successfully finds the cluster in large scale data. Here the application should be considered and relate to parameter settings. For example, clustering of people who go to work inside the city need different parameters rather than clustering of people want to go outside of town in holidays. The first cluster needs small epsilon and specific time duration rather than second one. As discussed before low spatial resolution data in this data set limit our expectation of the clustered results as It's not reasonable to study micro scale behavior of data.

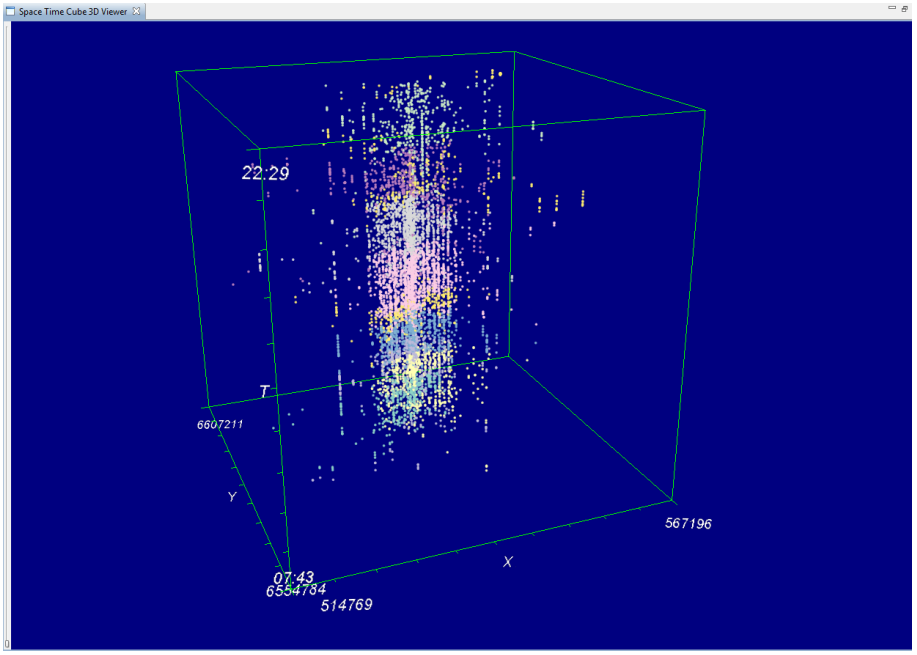

Figure 5.4: The spatio-temporal clusters parameter setting top item 1 in the Table5.5

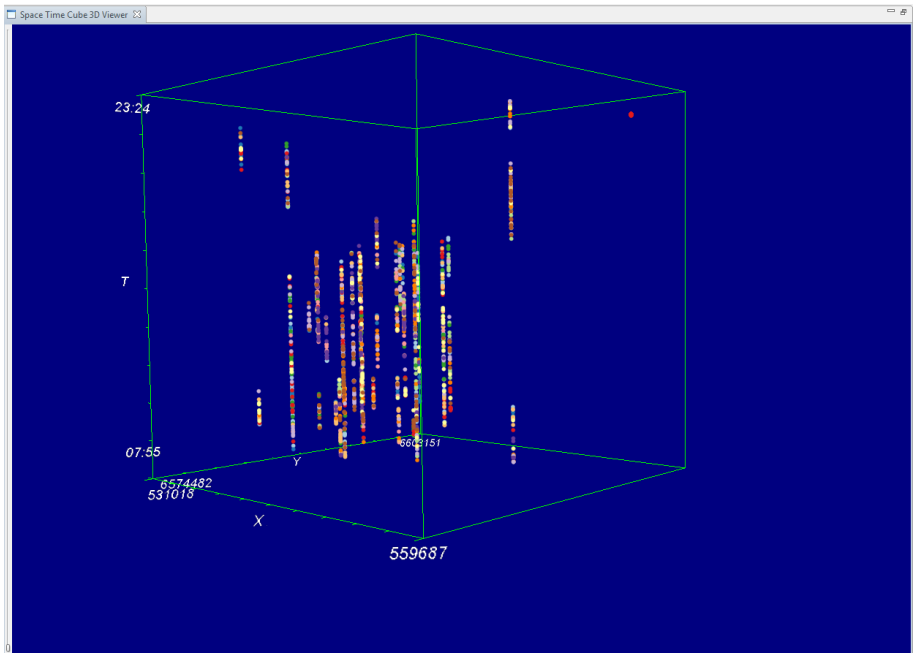

Figure 5.5: The spatio-temporal clusters parameter setting item 3 in the Table5.5

5.5.2 Distance function

The distance function route similarity is useful in certain places for this data set. For example, along a highway which the antennas are placed. Inside the city this data set is unsuitable. Another issue is irregular sampling of this data set because the positions are available when the user makes a call activity but what will happen if the user doesn't make call activity for a long time? Considering these issues make using trajectory based distance function harder. Point based clustering are more reasonable because treats the points independently from other points. In our view spatio-temporal distance function is a suitable choice for this data set. In other had technically trajectory data is available therefore trajectory based distance function could be used as well.

Table 5.6: Summarized table of cell phone tracking data set

Cell phone data		
Data properties	Spatial Resolution	low
	Temporal Resolution	High
	Spatial extend	High
	Temporal extend	low
Data Type	Point Data	
Distance functions	Common source/destination	✓
	Common source and destination	✓
	Route similarity	✓
	Route similarity+ dynamics	✓
	Space time distance	✓

5.6 Toward trajectory clustering in space-time clustering

Trajectory data as been discussed perviously in detail as a model of reality which represents spatio-temporal movement data. Clustering of trajectories because of its complexity,is some times difficult due to incomplete trajectories, low resolution data etc. Therefore, complex distance functions like *route similarity* or *route similarity and dynamics* are used to cluster them [45]. On other hand, our clustering tool does the clustering at point level, which means each point is treated individually. For example, for a given person in RFID data set, there could be many clusters, or maybe just few points belong one cluster at certain moment. Here the trajectories could be drawn easily from clusters by a simple *query* by ID. Such a query could be done in database or even in spread sheet of data. As an example in the clustered result from RFID data set we picked up one cluster with 12 persons. If the trajectory of these 12 persons in the original data are extracted and compare to the position of the portion of point in the clustered result, trajectory clusters can be easily derived. For better understanding of the issue, see Figure 2 below. This is the layer of a clustered result which different colors show different persons. It could be seen that when the persons are station due to good resolution of data the trajectories for each person(here constant trajectory) are easily detectable. But when people move there are missing values for trajectories which should be extracted by query in data base.

This theory is very simple but practically in the complicated trajectory clustering some issues arise. It should be considered which cluster should be considered to extract the trajectories, what is the best cluster that shows the cluster of trajectories, how many points of a trajectory should be in point clusters to consider the trajectory in the cluster and, etc. By this method-extracting trajectory from point clusters- in fact, the main base as clusters are available, all need to be done is *mining* meaningful trajectory clusters.

At the end of this section it is concluded that Space-Time Clustering is

achieved in all data sets. For each data set possible distance functions are suggested. In the next chapter all the procedure of research and the results of this Chapter will be summarized and discussed.

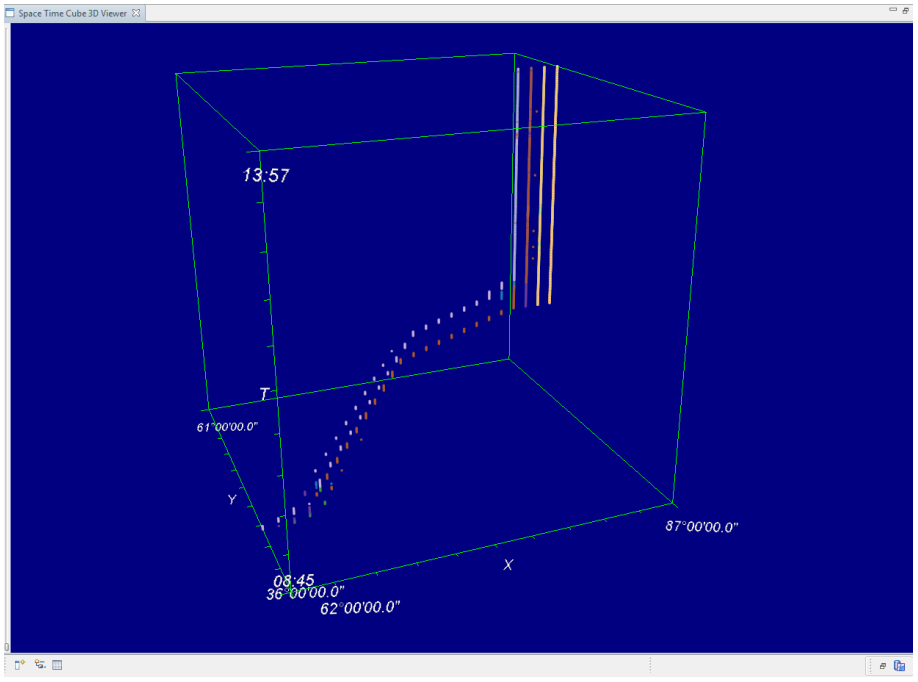

Figure 5.6: A cluster with different members by different color

Chapter 6

Discussion Conclusion and Recommendation

In this chapter first the summary of research with related chapters, will be reviewed. Then different issues relate to the research will be discussed. The conclusion of research will be provided next and at the agenda for further work for this research will be suggested at the end.

6.1 Summary of research

In this research clustering algorithms in the area of human movement data has been implemented and evaluated for the first time in the uDig by java programming language.

According to the first Chapter the main aim of this research is defined as above sentence. Toward this goal four main steps were defined as follows. Reviewing clustering algorithms and application in human movement data have been done in Chapter 2. In this chapter clustering definition, applications and its rule in data mining, KDD and Geovisual Analytics were studied. Also human movement data divided in two main types: point data and trajectory. According to these two types Clustering algorithms categorized in partitional, hierarchical and density based algorithms for point data, whole trajectory and partial trajectory for trajectory data. In this research the focus was on point data because data are usually points. Among point data-clustering algorithms-category density based clustering is considered as a good solution for clustering spatio temporal data because the properties like good efficiency, noise tolerant and finding arbitrary shaped clusters.

In Chapter 3 density based clustering and specifically a well known algorithm in this category named OPTICS was descirbed in details. This algorithm has been used before for clustering spatio temporal data, but there were not open source implementation available for it. Therefore the original pseudo codes described in chapter 3 was implemented.

In Chapter 4 implementation and expansion of OPTICS in uDig open source GIS was done. In this chapter implemented pseudo codes and gradually refinement of implementation is described. OPTICS uses distance function to find

similarity between objects. In order to expand this concept to space time clustering appropriate distance function which finds similarity in space and time, was introduced and implemented. The problem of finding temporal clusters was solved first by speed parameter and later by maxT parameter. Another parameters needed are ϵ and $\acute{\epsilon}$ which define searching area and clustering threshold respectively. Minpts defines the number of neighbors in the ϵ neighborhood. By the end of this chapter clustering algorithm was ready to test.

In chapter 5 Space time clustering algorithm was tested on three data sets(GPS, RFID and Cell phone tracking). These data sets have different characteristics. In order to deal with these characteristics appropriate parameter setting and distance function are needed. Using developed distance function and different parameter settings clusters in space and time achieved in all data sets. To evaluate the result visual interactive techniques in combination with validity indices named VNND were used. By visualization the cluster shape, number of clusters, members and by VNND the homogeneity of clusters were evaluated. The results shows the implementation successfully finds clusters in different data sets.

6.2 Discussion

6.2.1 Distance function

"Distance function" can be interpreted as a 'query' function, or as a separate mathematical function that determines similarity of points for clustering. Here we take this second view. Another measurement of distances -or differences- which could be placed in this view is *quality measurement*. Further discussion will provided in the subsequent subsection.

The developed and implemented distance function in this research is a purely mathematical function for determining which points should belong to a cluster using Euclidean distance measure. In this distance function a sphere is defined around the core- object as a searching bound(threshold).(see the definition one chapter 3 section 1). In this view, the Time and Location have equivalent values in the geometrical point of view because the Location is the horizontal circle and the Time is the vertical circle with same radios. If it is desired that the search area should be larger spatially than temporally, the horizontal search circle should be bigger(or vice versa). For example in finding the clusters of people before earthquake and after earthquake in a city which searching area is big and the time short. Therefore, an ellipsoid with semi major axes along the X, Y plane should be replaced to the sphere(the current distance function). In this case new distance function with additional parameters like ellipticity is needed. This parameter according to ellipsoid formula ($\frac{x^2}{a^2}$+$\frac{y^2}{b^2}$+$\frac{z^2}{c^2}$=1) will be defined as a relation between a, b and c. For example if a, b, c were equal the shape will be an sphere or if $a = b$ and $c > a$ the ellipsoid expands temporally rather than spatially.

Distance function and clustering quality measurement

As been discussed before in chapter 2, clustering is an unsupervised process which means it is not known how good clustered results are. Therefore, some measurements are provided to measure the quality of clustering. These measurements are named validity techniques of which there are divide in two main categories: external and internal validation techniques. External validation techniques tries to measure the quality w.r.t known labeled objects or ground truths, while internal validation techniques try to find how well clustering algorithm tailors to the internal structure of data and does not use pre-knowledge labeled data. [25] The simple quality measurement in this area is *symmetric set difference*, which determine the quality based on the number of identical point the clusters as follows. Let C1 and C2 be clusters, the *symmetric set difference* will be defined as [11]:

$$d_{\Delta}(C1, C2) = |C1 \bigcup C2| - |C1 \bigcap C2|$$

Another Quality measurement of two clusters have been done by distance function *minimal matching distance*. According to this distance function dissimilarity of points in two clusters is measured. In the case of unmatching number of members in clusters the penalties will be applied. [11]

Another issue in distance function and similarity measurement would be In this view distance functions are defined for the different kind of variables like Binary variables(0,1), Nominal variables, Ordinal variables, Ratio scaled variables, and finally Vector variables. [27]

6.2.2 Development of software

One of the difficulties to expansion of OPTICS to space-time according to 3D Euclidian distance formula, is two metrics in the same formula. Space and Time have different metrics. Speed parameter was considered as a solution to convert time to the space metrics. However different issues and difficulty were occurred. For example, when the coordinate system of the input shapefile layer, is in decimal degrees first the degree should converted to meter and then appropriate speed should be set. Even different positions in the same coordinate system like WGS84- from north to west- should have different speed parameters. Therefore *maxT* parameter was replaced to the speed parameter. This parameter as shown in table 6.1 is easy to set and interpret.

Another issue of current implementation is the problem of clusters under the threshold. Minpts parameter defines the neighborhood threshold for the clusters. Therefore, the number of members in a cluster shouldn't be under Minpts. In some parameter setting this problem occurred. The solution of this problem is removing these clusters from the result and reconsider the implementation and definition to see the source of this problem.

During the research, parallel to the technical enhancing of space time clustering some enhancing were done to enhance the software. New options and possibility could be added to the work that so far done.

As discussed in the Geovisual Analytics section in Chapter 2, contribution of humans by visualizing the result of algorithm and interpretation based on it

plays a key role in better understanding the result and finding patterns. Clustering tools provided in this research in the Geovisual Analytics point of view should be able to support the preceding idea. One of the of the weaknesses of uDig is in visualizing clusters by different colors, which assign maximum twelve colors to the clusters' results. However, it has been seen that number of clusters are more. Therefore, different clusters are assigned in the same color. Another weakness is the size of the clustered point results in the Space-Time Cube which should be changed manually by the XML file. One of the possible ways of interpret the results of clustering is getting query from the result which shows every information relating to point by clicking on each point. This possibility is available in the 'layer view' result of clustering but not in the result demonstrated by Space-Time Cube. The other possibility is extending the simple available in the Space-Time Cube to include attribute queries. This gives the user functions like selection by desired criteria, remove clusters, statistics, etc. In this research selection by attribute was done in ArcGIS.

Run time of the algorithm is another issue that should be addressed. In some experience in RFID data set(70,000 records) by high value parameters(ϵ=100, MaxT=500 seconds) it has been seen that run time took one hour.

6.2.3 OPTICS and trajectories

The possibility of showing the result in the trajectory form should be considered. OPTICS is more about clustering points (cases like the migrant boats data set) and not really designed for trajectories. It finds the point clusters. As discussed in Chapter 5 trajectories could be extracted from the result of space time clustering. However there are some issues here, like in which level by how many members and in which clusters, trajectories should be extracted. Visualization in this situation could help to better understanding of the result.

6.2.4 Parameter setting and distance function

Clusters extended in space and time were achieved by appropriate parameter setting. One of difficulty of working with clustering algorithm is setting appropriate parameters. In this work several experiments were done to find appropriate parameters. Such a parameters should find clusters according to the structure of data and with good homogeneity which were achieved by visual interactive analysis and VNND indices respectively.

Table 6.1 summarized parameter settings and measurement relate to each parameters in three data sets. From this table some general role could be extracted. For example in all data set, by increasing the parameters the number of clusters decreases. This acceptable according the algorithm and it means by big parameter clusters unify in bigger clusters.

Run time of the algorithm is measured for different parameter settings. By increasing the time threshold and searching area the run time will be increased. The biggest run time according to the table 6.1 belongs to RFID data set by very big parameter setting. However, from the evaluation of VNND parameter it is obvious that this parameter setting is not suitable for the data set. The lowest

run time belongs to GPS data set.

Distance function which has been used here, has found spatio-temporal clusters. Another possible distance functions which could be used for data sets are suggested in table 6.2. According to this table data sets in special format(RFID, Cell phone tracking) like ID, X, Y, Time could be clustered both by point based distance function and trajectory based distance functions. Data set without ID like GPS data just could be clustered by point based distance function which has been implemented here.

Table 6.1: Summarized table of parameter setting and result

	ϵ	$\acute{\epsilon}$	**Minpts**	**MaxT**	**Number of Clusters**	**VNND**	**Run time**
GPS data							
1	15	5	5	20 days	17	0.414288	00:01
2	100	50	15	15 days	3	0.64878	00:01
3	50	25	15	30 days	15	1.69	00:01
RFID data							
4	5	2.5	5	300 seconds	31	0.0776	03:51
5	3	1.5	5	300 seconds	40	0.0147716	03:28
6	50	25	15	300 seconds	192	3.429216	28:51
7	2	1	15	120 seconds	267	0.015	02:58
Cell phone tracking data							
8	23870	10000	30	1 hour	174	696735.68	01:03
9	23870	10000	30	5 hours	5	18393241.37	14:23
10	100	50	15	1 hour	103	3.971343	00:04

Table 6.2: Summarized table of distance functions and data characteristics

Data set	GPS	RFID	Cell phone tracking
Data Type	Point Data		
Common source/destination		✓	✓
Common source and destination		✓	✓
Route similarity		✓	✓
Route similarity+ dynamics		✓	✓
Space time distance	✓	✓	✓
Spatial Resolution	High	High	Low
Temporal Resolution	Low	High	High
Spatial Extent	High	Low	High
Temporal Extent	High	Low	Low

Application should be considered in parameter setting. It should be known for which propose the results want to be used. To micro-scale behavior, it is important to have high resolution data. The example of high resolution data here

is RFID data set. In this data set cluster in micro-scale is easily detectable by small parameter settings. Item 7 in Table 6.1 is example of such a parameter setting. However low resolution data like cell phone tracking data is not suitable for study micro-scale behavior for example clustering people in a building. In this case all people maximum assign to one cluster because the position of them are the same in data set. In other hand this data set in quite suitable for study macro-scale behavior due to high extended data set in space. For example clustering of tourist in the country or county.

6.3 Conclusion

Geographic Information (GI)community faced to the increase in availability of data during last decades due to advancement in the collection devices and also user requirements. Processing and extracting meaningful information from this large amount of data is the domain of data mining, KDD and currently Geovisual Analytics. Clustering in all these concepts plays an important role.

Spatio temporal movement data is one kind of data,in which interest is growing too fast. Clustering of spatio temporal movement data is quite a new area of research, which extends the existing clustering algorithms to appropriate use. In the categorization of existing clustering algorithms density based algorithms are appropriate to clustering spatio temporal movement data because of good efficiency, their ability to deal with noise, their ability to extract arbitrary shaped clusters, which covers the properties of spatio temporal data like random component, low or different resolution both in time and location, etc.

OPTICS is a sophisticated algorithm in this category. This algorithm has been used mainly in trajectory clustering while the focuss of this research is mainly on point based clustering. This clustering algorithm uses distance function in order to find similarity between different objects. These properties of OPTICS- using distance function- in combination with appropriate parameter setting provides a flexible clustering tool for diverse types of spatio temporal movement data with different characteristics.

The original aim was to find an existing open source implementation of OPTICS and make it available in uDig as a plugin to the existing Space-Time Cube environment. There were not appropriate open source codes. Therefore, the original codes of OPTICS were extended to work in space-time, and implemented using the JAVA programming language.

In this research OPTICS and specific distance function based on 3D Euclidian distance is implemented and tested over three different data sets. This distance function successfully finds the spatio temporal clusters in the point data sets. In the case of trajectory data set, using query from results is suggested. In fact, the implemented distance function extracts directly point clusters, and indirectly- by help of simple query- trajectory clusters. The advantage of using this distance function for trajectory data are, data doesn't need to be in special order to enter to clustering process, and simple computation. The parameter setting is a drawback of this algorithm-like many clustering algorithms-, as dif-

ferent parameters lead to different results. The parameter settings in order to deal with different characteristics like resolution, spatial extends and temporal extends were done for all three data sets. The result showed that the clusters expanded in time and location, which means spatio temporal clustering is achieved.

There is no general definition for quality of clusters. Therefore, evaluation of clusters is done based on two levels: visual interactive method and validity indices. By visual exploration of data set and clusters in space time cube and layer view, it is possible to see how much result fits to the structure of data. By this approach, a set of good clusters is defined. Further evaluation is done on the good clusters by VNND validity indices. VNND measures the compactness of clusters. The lower value of VNND means more homogenous clusters. This evaluation is done for all three data sets and final results were achieved(table 6.1).

6.4 Agenda for further research

The previous section has discussed the issues and limitations of the implementation of a space-time clustering algorithm. There are a number of things that need further research:

1. Automatic Parameter setting: the possibility to set parameters automatically according to the structure of data set
2. Large data bases: enhance issue relate to efficiency of algorithm to deal with large data sets
3. Context in clustering: how to contribute environment of clustering data to the clustering process and evaluation results
4. Filtration of result: remove bad members and bad clusters from the result automatically
5. Other disciplines: adapt distance function or using different distance function to other disciplines

In the following the more discussion will be provided.

6.4.1 Automatic Parameter setting

One of the difficulties in clustering algorithms is parameter setting as different parameters lead to different results. In the current implementation four parameters are needed, which are $\epsilon, \acute{\epsilon}$, Minpts, maxT. Possibility of automatic or semi automatic setting of this parameter is provided as follows.

The ϵ defines the searching area. According to the [5] this parameter should be big enough to ensure no point is left unprocessed in the data set. To set searching area efficient(by minimum distance measurement and cover all local densities), according to the algorithm definitions ϵ at least should be set as *max core-distance+1*. However, the difficulty here is that the core-distance value is

a function of ϵ variation. In other words by defining ϵ the points will be core-objects with specific core-distances. Therefore, the ϵ parameter should be set independent of OPTICS definitions. One possible way is using the minimum bounding box in the biggest dense area. By defining the dense areas, it would be possible to extract appropriate ϵ value to ensure covers all points. In this view dense area could be extracted visually from space time cube or even layer view.

The $\acute{\epsilon}$ determine in which level of rechability distance the clusters could be extracted. In other words it extracts clusters from the data set by desired threshold in rechability plot.(chapter 3 setion 2). Ankerst et al. has defined that by using the steep of rechability plot it is possible to automatically find the clusters. [5]

The Minpts parameter directly affect the core-object condition as it defines which object could be core object. This parameter should be set according to the density of data set, for example in a compact data set small Minpts leads to many core object and therefore number of clusters will be increased. [45] Therefore it should be set according points spread in the space.

The final parameter is maxT which has a same situation as Minpts as it should be set according to the structure of data in the time dimension. It defines the desired time interval of clusters.

6.4.2 Large data bases

One of the important aspects of clustering algorithm in the data mining and KDD is dealing with large amounts of data. [28] In such a data set huge number of distances should be calculated. This process is very time consuming and causes the big run time. As previewed before in the area of Geovisual Analytics the quickness of the algorithm is important too. [13] Therefore, making the algorithm efficient is very important. Possible way to deal with this issue is making index of data and using aggradation methods.One of the works which could be categorized in this area tries to increase the efficiency of density based clustering algorithm by avoiding unnecessary distance measurement. Based on OPTICS and DBSCAN definitions, these algorithms work based on ϵ range queries, and therefore, many distance calculations. In the case of large data set and "complex object"-like spatio-temporal moving object- ϵ needs to be introduced efficiently in order to avoid unnecessary distance measurement. "multi step query processing paradigm" was used directly into clustering algorithm in order to accelerate the algorithm by reduce the complex distance measure. In this approach complex distance measurements are neglected until the final result is no longer affected. In other words the priority is with the simple distance measures, as long as we don't loose of important results. [11]

In the current implementation when data set gets big, making order file will be hard. This difficulty is until the order file has been made. As long as the order file is made it will be easier to change the second e because it works like a threshold and does not measure any distances.

6.4.3 Context in clustering

Context is assigned to any information from the environment. This information could be used in the process of computing the algorithm or to interpret the result. Environment is not just physical environment that data are in it. In spatio temporal movement data context could be either the time as *when* the data is happening or the surroundings as *where* data is happening. Examples of simple contexts are working hour for *when* and the commercial area in a city as *where*. The challenge, here is how to contribute the context to the spatio-temporal clustering. For example, in clustering students of ITC it is important to know time of activity and the kind of activity they have(practical, field work, lecture, etc.). One simple way to better interpretation of clustering result is using a map in the background of the location and use a calender-that shows days, special occasions, etc. in the time dimension. By this information, it's possible to better interpret the clusters as the *where* and *when* question is known.

6.4.4 Filtration of result

The clustering tool introduced in this research is supposed to create the clusters which pass some criterions like suitable reachability distance and suitable number of neighbors. Among the accepted clusters sometimes the degree of belonging to the cluster is needed in order to filter out bad members.(the members with the low degree of belonging to the cluster). This filtering could be done by comparing members of the cluster to one representative point. The best representative point is the center of the cluster. In the spatio temporal clustering the clusters are expanded in the three dimensional space. Therefore, the center is either centroid of the 3D shape or if the cluster is considered as mass is the center of mass. By measure the distance to center the degree of belonging will be defined.

One of the filters has been used to remove bad clusters is removing by query in the table associated to the clustering results. Bad clusters here mean clusters with members under threshold or high VNND quality measurement. The process of removing is manually and therefore, time consuming and includes human errors. The automatic remove bad clusters is a good solution. One of the possibilities in order to automatic remove bad clusters is using variance of individual clusters. The procedure is provided as follows. First variance of individual clusters is derived form VNND. Then by averaging all variances an expected value is calculated. The standard deviation of cluster's variance to the expected value should be under a certain threshold- for example, 95 percent in normal distribution- to accept the cluster. Using such a method in spite to being automatic, has its drawback when the variance of clusters has significant differences. In this case, the important clusters may be excluded. For example, in clustering of cars in a city to show the traffic area may be a big cluster which has a big standard deviation will be eliminated by this method and just dense clusters in other areas will be accepted.

6.4.5 Other disciplines

It is believed that current OPTICS implementation which works based on 3D Euclidian distance function could be used for clustering other three dimensional data. For example, air pollution sensors in a city record the degree of pollution in the different moment. The position of sensors is known. Therefore, by using clustering tool and degree of pollution it is possible to define dangerous areas. If the forth dimension or time added to the clustering according to the Minkowski distance formula introduced in Chapter 3 it would be possible to use distance function in four dimensions. However, it is not simple to measure distance of three different kind of data(space, time and interval scale) in one distance function.

Another issue introduced in chapter 5 section 6 is adding query to current distance function in order to deal with multi dimensional data. It has been discussed how by using simple query trajectory data extracted from clusters of point data. If another attributes like age, sex and job are desirable to add to the clustering result, it should be discussed in which level, by which weight they should contribute in the clustering process.

Appendix A

[OpticsItem.java]

```
public class OpticsItem {
        private boolean processed;
        private double reachabilityDistance;
        private double coreDistance;
        private double x;
        private double y;
        private double z;
        private String featureID;
        private int order; // temporary
        private int nrNeighbors; // temporary
        public static double UNDEFINED = Double.MAX_VALUE;

        public OpticsItem(double x, double y, long time, double speed, String
    featureID) {
                this.x = x;
                this.y = y;
                this.featureID = featureID;
                this.z = (time / 1000.0) * speed;
                this.processed = false;
                this.reachabilityDistance = UNDEFINED;
                this.coreDistance = UNDEFINED;
                this.order = -1;
        }

        public boolean isProcessed() {
                return processed;
        }
        public void setProcessed(boolean processed) {
                this.processed = processed;
        }

public void setReachabilityDistance(double reachabilityDistance) {
```

```
            this.reachabilityDistance = reachabilityDistance;
        }

        public double getReachabilityDistance() {
            return reachabilityDistance;
        }

        public void setCoreDistance(double coreDistance) {
            this.coreDistance = coreDistance;
        }

        public double getCoreDistance() {
            return coreDistance;
        }

        public double getDistance(OpticsItem opticsItem) {
            return Math.sqrt((x - opticsItem.x) *
    (x - opticsItem.x) + (y - opticsItem.y) *
    (y - opticsItem.y) + (z - opticsItem.z) *
    (z - opticsItem.z));
        }

        public double getX() {
            return x;
        }

        public double getY() {
            return y;
        }

        public String getFeatureID() {
            return featureID;
        }

        public void setOrder(int order) {
            this.order = order;
        }

        public int getOrder() {
            return order;
        }

        public void setNrNeighbors(int nrNeighbors) {
            this.nrNeighbors = nrNeighbors;
        }

        public int getNrNeighbors() {
```

```
                return nrNeighbors;
        }
}
```

Appendix B

[Optics.java]

```
package org.n52.udig.stc.optics;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import org.geotools.feature.Feature;
import org.n52.udig.stc.Shape3DLayerFactory;
import org.n52.udig.stc.TimeMapping;
import org.n52.udig.stc.uielements.StcProgressMonitor;

import com.vividsolutions.jts.geom.Coordinate;

public class Optics {

        private final double epsilonXY;
        private final double epsilon1;
        private final int minPts;
        private final double epsilonT;
        private Map<String, Object[]> attributesMap;

        public Optics(double epsilonXY, double epsilon1, int minPts, double...
                                                                epsilonT) {
                this.epsilonXY = epsilonXY;
                this.epsilon1 = epsilon1;
                this.minPts = minPts;
                this.epsilonT = epsilonT;
        }

        public void computeOpticsClustering(Shape3DLayerFactory layerItem,...
                                        StcProgressMonitor monitor) {
                System.out.println("Started␣OPTICS!");
```

```
                // STEP 1: Read-in user-data
                List<OpticsItem> objSet = new ArrayList<OpticsItem>();
                try {
                Coordinate[] coordinates = layerItem.getCoordinates();
                        TimeMapping[] times = layerItem.getStartTimes();
                        Feature[] features = layerItem.getFeatures();
                        for (int i = 0; i < coordinates.length; ++i)
                objSet.add(new OpticsItem(coordinates[i].x, coordinates...
  [i].y, times[i].toLong(), epsilonXY / epsilonT, features[i].getID()));
                } catch (Exception e) {
                        e.printStackTrace();
                }

                monitor.setMaximum(objSet.size());

                // STEP 2: Perform clustering calculation
List<List<OpticsItem>> clusters = getOpticsClustering(objSet,...
                                  epsilonXY, epsilon1, minPts, monitor);

                // STEP XX: Get results

                if (clusters != null && !monitor.isCanceled()) {
                        attributesMap = new HashMap<String, Object[]>();
                        Integer clusterID = 0;
                        for (List<OpticsItem> cluster : clusters) {
                                Integer clusterSize = cluster.size();
Double varianceNearestNeighborDistance = null;
                                if (clusterSize > 1) {
                                        double dminC = 0;
                                        for (int i = 0; i < .....
clusterSize; ++i)
                                                dminC += cluster.get(i).
.....
getCoreDistance();
                                        dminC /= clusterSize;
                                        varianceNearestNeighborDistance =
 0.0;
                                        for (int i = 0; i....
 < clusterSize; ++i) {
                        double dminX = cluster.get(i).getCoreDistance();
                        varianceNearestNeighborDistance += .....
          (dminX-dminC)*(dminX-dminC);
                                        }
varianceNearestNeighborDistance /= (clusterSize - 1);
                                }
                                for (OpticsItem item : cluster) {
                        Double coreDistance = item.getCoreDistance();
```

```
                                        Double reachabilityDistance =
                                         item.getReachabilityDistance();
                                        Integer opticsOrder = item.getOrder();
Integer nrNeighbors = item.getNrNeighbors();
                                        attributesMap.put(item.getFeatureID(),
{clusterID, clusterSize, varianceNearestNeighborDistance, coreDistance,...
 reachabilityDistance, opticsOrder, nrNeighbors});
                                }
                                ++clusterID;
                        }
                } else
                        attributesMap = null;

                System.out.println("Done!");
        }

        // ExtractDBSCAN-Clustering
        private List<List<OpticsItem>> getOpticsClustering(List<OpticsItem>...
                   objectSet, double epsilon, double epsilon1, int minPts,
                   StcProgressMonitor monitor) {
                if (epsilon1 <= epsilon) {
        List<OpticsItem> orderedItems = getOpticsOrdering(objectSet,
epsilon, minPts, monitor);
                        if (orderedItems != null) {
                                List<List<OpticsItem>>....
                           resultList = new ArrayList<List<OpticsItem>>();
                                List<OpticsItem> currentCluster = null;
                                for (OpticsItem object : orderedItems) {
                                        if (object.getReachabilityDistance() >
epsilon1) {
                        // assuming epsilon1 <= epsilon < UNDEFINED
                                                if (object.getCoreDistance()..
<= epsilon1) {
                                        // ClusterId := nextId(ClusterId)
                                                        if (currentCluster !=.
null && currentCluster.size() > 0)
resultList.add(currentCluster);
currentCluster = new ArrayList<OpticsItem>();
                        // Object.clusterId = ClusterId
                        currentCluster.add(object);
                                                }
                        // ELSE Object.clusterId := NOISE
                                                } else if (currentCluster != null)....
                       // Object.reachability_distance <= epsilon1
                                                currentCluster.add(object);
                                        }
        if (currentCluster != null && currentCluster.size() > 0)
```

```
                                        resultList.add(currentCluster);
                                    return resultList;
                                }
                            }
                            return null;
                    }

                    private List<OpticsItem> getOpticsOrdering(List<OpticsItem>....
objectSet, double epsilon, int minPts, StcProgressMonitor monitor) {
                            List<OpticsItem> resultList = new ArrayList<OpticsItem>();
                            List<OpticsItem> orderSeeds = new ArrayList<OpticsItem>();
                            for (OpticsItem item : objectSet) {
                                    if (monitor.isCanceled())
                                            return null;
                            if (!item.isProcessed())
                                    expandClusterOrder(objectSet, item, epsilon,....
minPts, resultList, orderSeeds, monitor);
                                }
                            return resultList;
                    }

                /**
                 * Expands the ClusterOrder for this dataObject
                 * @param object Start-OpticsItem
                 * @param seeds SeedList that stores dataObjects with....
                  reachability-distances
                 */
                private void expandClusterOrder(List<OpticsItem> objectSet,....
                 OpticsItem object, double epsilon, int minPts,.....
                 List<OpticsItem> resultList, List<OpticsItem> orderSeeds,....
                  StcProgressMonitor monitor) {
                    List<OpticsItem> neighbors = getNeighbors(objectSet,....
                    object, epsilon);
                    monitor.worked(1);
                    object.setProcessed(true);
                    object.setReachabilityDistance(OpticsItem.UNDEFINED);
                    object.setCoreDistance(getCoreDistance(object, neighbors,....
                     minPts));

                    object.setOrder(resultList.size());
                    object.setNrNeighbors(neighbors.size());
                    resultList.add(object);

                    if (object.getCoreDistance() != OpticsItem.UNDEFINED) {
                        update(orderSeeds, neighbors, object);
                        while (orderSeeds.size() > 0 && !monitor.isCanceled()) {
                            OpticsItem currentObject = orderSeeds.remove(0);
```

```
                // next()
                neighbors = getNeighbors(objectSet, currentObject,...
                 epsilon);
                if (!currentObject.isProcessed())
                        monitor.worked(1);
                currentObject.setProcessed(true);
                currentObject.setCoreDistance(getCoreDistance(....
                currentObject, neighbors, minPts));

                currentObject.setOrder(resultList.size());
                currentObject.setNrNeighbors(neighbors.size());
                resultList.add(currentObject);

                if (currentObject.getCoreDistance() !=....
                 OpticsItem.UNDEFINED) {
                    update(orderSeeds, neighbors, currentObject);
                }
            }
        }
    }

    /**
     * Updates reachability-distances in the Seeds-List
     * @param seeds UpdateQueue that holds DataObjects with their....
      corresponding reachability-distances
     * @param neighbours List of DataObjects that were found in....
      epsilon-range of centralObject
     * @param centerObject
     */
    private void update(List<OpticsItem> orderSeeds, List....
     <OpticsItem> neighbors, OpticsItem centerObject) {
        double c_dist = centerObject.getCoreDistance();

        for (OpticsItem object : neighbors) {
            if (!object.isProcessed()) {
                double new_r_dist = Math.max(c_dist,....
                 centerObject.getDistance(object));
                object.setReachabilityDistance(new_r_dist);
                // insert object in orderSeeds, so that....
             orderSeeds remains sorted by reachabilityDistance
                orderSeeds.remove(object);
                int i = 0;
                while ((i < orderSeeds.size()) && ...
            (new_r_dist > orderSeeds.get(i).getReachabilityDistance()))
                    ++i;
                orderSeeds.add(i, object);
            }
```

```
        }
    }

        private List<OpticsItem> getNeighbors(List<OpticsItem>...
 objectSet, OpticsItem opticsItem, double epsilon) {
                List<OpticsItem> neighbors = new ArrayList<OpticsItem>();
                for (OpticsItem item : objectSet) {
                        if (!item.equals(opticsItem) &&....
 item.getDistance(opticsItem) <= epsilon)
                                neighbors.add(item);
                }
                return neighbors;
        }

        private double getCoreDistance(OpticsItem ....
opticsItem, List<OpticsItem> neighbors, int minPts) {
                if (neighbors.size() < minPts - 1)
                        return OpticsItem.UNDEFINED;
                else {
                        double coreDistance = OpticsItem.UNDEFINED;
                        for (OpticsItem item : neighbors)
                                coreDistance = Math.min(coreDistance,...
 item.getDistance(opticsItem)); // assumes UNDEFINED > any valid distance
                        return coreDistance;
                }
        }

        public Map<String, Object[]> getAttributesMap() {
                return attributesMap;
        }
}
```

Bibliography

[1] Rein Ahas, Anto Aasa, Antti Roose, lar Mark, and Siiri Silm. Evaluating passive mobile positioning data for tourism surveys: An estonian case study. *Tourism Management*, 29(3):469 – 486, 2008.

[2] Mattias Andersson, Joachim Gudmundsson, Patrick Laube, and Thomas Wolle. Reporting leaders and followers among trajectories of moving point objects. *GeoInformatica*, 12:497–528, 2008.

[3] Gennady Andrienko and Natalia Andrienko. Interactive cluster analysis of diverse types of spatiotemporal data. *SIGKDD Explor. Newsl.*, 11(2):19–28, 2009.

[4] Gennady Andrienko, Natalia Andrienko, and Stefan Wrobel. Visual analytics tools for analysis of movement data. *SIGKDD Explor. Newsl.*, 9(2):38–46, 2007.

[5] Mihael Ankerst, Markus M. Breunig, Hans peter Kriegel, and Jorg S. Optics: Ordering points to identify the clustering structure. pages 49–60. ACM Press, 1999.

[6] Luc Anselin. What is special about spatial data? alternative perspectives on spatial data analysis. Technical Report 89-4, National Center for Geographic Information and Analysis, 1989.

[7] Stan Aronoff. *Geographic Information Systems: A Management Perspective*. WDL Publications, Ottawa, Canada, 1989.

[8] Miriam Baglioni, José Antônio Fernandes de Macêdo, Chiara Renso, Roberto Trasarti, and Monica Wachowicz. Towards semantic interpretation of movement behavior. In *AGILE Conf.*, pages 271–288, 2009.

[9] Pavel Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002.

[10] Enrico Bertini and Denis Lalanne. Investigating and reflecting on the integration of automatic data analysis and visualization in knowledge discovery. *SIGKDD Explorations*, 11(2):9–18, December 2009.

[11] Stefan Brecheisen, Hans-Peter Kriegel, and Martin Pfeifle. Multi-step density-based clustering. *Knowledge and Information Systems*, 9(3):284–308, 2006.

[12] P. Compieta, S. Di Martino, M. Bertolotto, F. Ferrucci, and T. Kechadi. Exploratory spatio-temporal data mining and visualization. *Journal of Visual Languages and Computing*, 18(3):255 – 279, 2007. Visual Languages and Techniques for Human-GIS Interaction.

[13] G. Ellis D. Keim, J. Kohlhammer and F. Mansmann. (forthcoming) Mastering the information age: Solving problems with visual analytics. 2010.

[14] Urska Demsar and Kirsi Virrantaus. Space-time density of trajectories: exploring spatiotemporal patterns in movement data. *International Journal of Geographic Information Science*, 24(10):1527–1542, 2010.

[15] Thomas Eiter and Heikki Mannila. Distance measures for point sets and their computation. *Acta Informatica*, 34:109–133, 1997.

[16] Martin Ester, Hans peter Kriegel, Jorg S, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press, 1996.

[17] Usama Fayyad, Gregory Piatetsky-shapiro, Padhraic Smyth, and Terry Widener. The kdd process for extracting useful knowledge from volumes of data. *Communications of the ACM*, 39:27–34, 1996.

[18] Peter Fisher, Patrick Laube, Marc Kreveld, and Stephan Imfeld. Finding remo detecting relative motion patterns in geospatial lifelines. In *Developments in Spatial Data Handling*, pages 201–215. Springer Berlin Heidelberg, 2005.

[19] Peter Gatalsky, Natalia Andrienko, and Gennady Andrienko. Interactive analysis of event data using space-time cube. In *Proceedings of the Information Visualisation, Eighth International Conference*, pages 145–152, Washington, DC, USA, 2004. IEEE Computer Society.

[20] Reginald G. Golledge. The nature of geographic knowledge. *Annals of the Association of American Geographers*, 92:1–14(14), 2002.

[21] J. C. Gower and G. J. S. Ross. Minimum spanning trees and single linkage cluster analysis. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 18(1):54–64, 1969.

[22] G. Grinstein, C. Plaisant, S. Laskowski, T. O'Connell, J. Scholtz, and M. Whiting. Vast 2008 challenge: Introducing mini-challenges. *Visual Analytics Science and Technology, 2008. VAST '08. IEEE Symposium on*, pages 195 –196, 2008.

[23] Robert Groth. *Data mining: a hands-on approach for business professionals*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1998.

[24] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Cure: an efficient clustering algorithm for large databases. *Information Systems*, 26(1):35 – 58, 2001.

[25] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. On clustering validation techniques. *J. Intell. Inf. Syst.*, 17(2-3):107–145, 2001.

[26] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. Cluster validity methods: Part i. *ACM SIGMOD Record*, 31(3):19–27, 2002.

[27] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques (2 nd edition)*. Morgan Kaufmann, 2006.

[28] Jiawei Han Harvey J. Miller. *Geographic Data Mining and Knowledge Discovery, Second Edition*. CRC Press, 2009.

[29] A K Jain, M N Murty, and P. J. Flynn. Data clustering: A review. *ACM computing surveys*, 31(3), 1999.

[30] Panos Kalnis, Nikos Mamoulis, and Spiridon Bakiras. On discovering moving clusters in spatio-temporal data. In *In SSTD*, pages 364–381. Springer, 2005.

[31] Leonard Kaufman and Peter J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis (Wiley Series in Probability and Statistics)*. John Wiley & Sons, 1990.

[32] T. Kohonen, M. R. Schroeder, and T. S. Huang, editors. *Self-Organizing Maps*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.

[33] Ferenc Kovács and Renáta Iváncsy. Cluster validity measurement for arbitrary shaped clusters. In *AIKED'06: Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases*, pages 372–377, Stevens Point, Wisconsin, USA, 2006. World Scientific and Engineering Academy and Society (WSEAS).

[34] Patrick Laube, Stephan Imfeld, and Robert Weibel. Discovering relative motion patterns in groups of moving point objects. *International Journal of Geographical Information Science*, 19:639–668(30), 2005.

[35] Mark D. M. Geospatial lifelines. In *In integrating spataial and temporal databases*, volume 98471. Dagstuhl Seminars, 1998.

[36] Kraak M-J. The space-time cube revisited from a geovisualization perspective. 21 International cartographic conference, 2003.

[37] Geoffrey Holmes Bernhard Pfahringer Peter Reutemann Ian H. Witten Mark Hall, Eibe Frank. The weka data mining software: An update. *SIGKDD Explorations*, 11(1), 2009.

[38] Boris Mirkin. *Clustering for data mining: a data recovery approach*. Computer science and data analysis series:3. Chapman & Hall/CRC, Boca Raton etc, 2005.

[39] David Mountain and Jonathan Raper. Modelling human spatio-temporal behaviour: a challenge for location based services. In *In 6th International conference of geocomputation: geocomputation*, 2001.

[40] The RFID Network. What is rfid? Website, Oct, 16 , 2010. `http://rfid.net/basics/190-what-is-rfid`.

[41] Raymond T. Ng and Jiawei Han. Efficient and effective clustering methods for spatial data mining. In *VLDB '94: Proceedings of the 20th International Conference on Very Large Data Bases*, pages 144–155. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1994.

[42] Maria Petrou and Panagiota Bosdogianni. *Image Processing: The Fundamentals*. John Wiley & Sons, Inc., New York, NY, USA, 1999.

[43] Jonathan Raper, Georg Gartner, Hassan Karimi, and Chris Rizos. A critical evaluation of location based services and their potential. *J. Locat. Based Serv.*, 1(1):5–45, 2007.

[44] Elaine Rich. *Artificial intelligence*. McGraw-Hill, Inc., New York, NY, USA, 1983.

[45] Mirco Nanni Fosca Giannotti Natalia Andrienko Salvatore Rinzivillo, Dino Pedreschi and Gennady Andrienko. Visually driven analysis of movement data by progressive clustering. *Information Visualization*, (7):225–239, 2008.

[46] Bruce J. Schachter, Larry S. Davis, and Azriel Rosenfeld. Some experiments in image segmentation by clustering of local feature values. *Pattern Recognition*, 11(1):19 – 28, 1979.

[47] Noam Shoval. Tracking technologies and urban analysis. *Cities*, 25(1):21 – 28, 2008.

[48] Simeon J. Simoff, Michael H. Böhlen, and Arturas Mazeika. Visual data mining: An introduction and overview. In *Visual Data Mining: Theory, Techniques and Tools for Visual Analytics*, volume 4404 of *Lecture Notes in Computer Science*, pages 1–12. Springer-Verlag, Berlin, Heidelberg, 2008.

[49] W. R. Tobler. A computer movie simulating urban growth in the detroit region. *Economic Geography*, 46:234–240, 1970.

[50] Goce Trajcevski, Ouri Wolfson, Klaus Hinrichs, and Sam Chamberlain. Managing uncertainty in moving objects databases. *ACM Trans. Database Syst.*, 29(3):463–507, 2004.

[51] Stefan van der Spek, Jeroen van Schaick, Peter de Bois, and Remco de Haan. Sensing human activity: Gps tracking. *Sensors*, 9(4):3033–3055, 2009.

[52] Marcos R. Vieira, Petko Bakalov, and Vassilis J. Tsotras. On-line discovery of flock patterns in spatio-temporal data. In *GIS '09: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 286–295, New York, NY, USA, 2009. ACM.

[53] Xu-Lei Yang, Qing Song, and Yi-Lei Wu. A robust deterministic annealing algorithm for data clustering. *Data Knowl. Eng.*, 62(1):84–100, 2007.

[54] C. T. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Trans. Comput.*, 20(1):68–86, 1971.