A METHODOLOGY FOR FINDING UNIFORM REGIONS IN SPATIAL DATA AND ITS APPLICATION TO ANALYZING THE COMPOSITION OF CITIES

A Thesis

Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

By

Zechun Cao

August 2013

A METHODOLOGY FOR FINDING UNIFORM REGIONS IN SPATIAL DATA AND ITS APPLICATION TO ANALYZING THE COMPOSITION OF CITIES

Zechun Cao

APPROVED:

Christoph F. Eick
Dept. of Computer Science

Ricardo Vilalta
Dept. of Computer Science

Germain Forestier
ENSISA - Universite de Haute Alsace, France

Dean, College of Natural Sciences and Mathematics

Acknowledgements

I owe my deepest gratitude to my advisor, Dr. Christoph Eick, for his excellent guidance, caring, patience, and providing me with an excellent atmosphere for doing research. I would like to thank Dr. Forestier, who let me experience the spatial data mining project in France and the practical issues beyond the textbook. I would also like to thank Dr. Vilalta for giving me great encouragement and helping me to correct the writing of this thesis. Last, but certainly not least, I would like to thank my parents and my wife. They were always there cheering me up and stood by me through the good times and bad, without which this thesis would never have reached fruition.

A METHODOLOGY FOR FINDING UNIFORM REGIONS IN SPATIAL DATA AND ITS APPLICATION TO ANALYZING THE COMPOSITION OF CITIES

An Abstract of a Thesis

Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

By

Zechun Cao

August 2013

Abstract

Cities all around the world are in constant evolution due to numerous factors, such as fast urbanization and new ways of communication and transportation. However, the evolution of the composition of a city is difficult to follow and analyze. Since understanding the evolution of cities is the key to intelligent urbanization, there is a growing need to develop urban planning and analysis tools to guide the orderly development of cities, as well as to enhance their smooth and beneficial evolution. Urban patches which represent uniform areas of a city play a key role in studying the composition of a city, as different types of urban patches typically are associated with different functions, such as recreational areas and commercial areas. In order to analyze the changes of the composition of cities, a polygon-based spatial clustering and analysis framework for studying urban evolution is proposed in this thesis. A spatial clustering algorithm named CLEVER is used to identify urban patches that are clusters of polygons representing different elements of the city based on a domain expert's notion of uniformity, which has to be captured in a plug-in interestingness function. The analysis methodology uses polygons as models for spatial clusters and histogram-type distribution signatures to describe their characteristics. Finally, popular signatures are introduced that describe distribution characteristics, which occur frequently in contiguous sub-regions of a spatial dataset, and an approach is presented that identifies and annotates urban patches with popular signatures. Experiments on datasets of the city of Strasbourg, France serve as an example to highlight the usefulness of the methodology.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"Urbanization is the physical growth of urban areas as a result of global change where increasing proportion of the total population becomes concentrated in towns. The United Nations reported that since 2008 more than half of the world's population is living in urban areas"[1]. Thus, mastering urban evolution became a major challenge for all major cities in the world. Consequently, there is a growing need to develop urban planning and analysis tools to guide the orderly development of cities, as well as enhance their smooth and beneficial evolution. The evolution of cities is a very dynamic activity; therefore, modeling the dynamics of urban evolution is a quite challenging task. Data describing city dynamics are widely available as they are collected on a regular basis, offering a great opportunity to develop urban computing techniques, which can be used to analyze and model urban evolution. Understanding and monitoring urban evolution allows urban planners to make smarter decisions because they can provide deep insights into a city with changing dynamics. Moreover, it offers an opportunity to improve people's knowledge about the impacts

from urbanization on the territory.

The step of urbanization leads to different functional regions in a city, called urban patches throughout the remainder of this thesis, such as residential areas, business districts, industrial and recreational areas. Different types of urban patches support different needs of people's lives and "serve as a valuable organization technique for framing detailed knowledge of a metropolitan area"[2]. Urban patches may be artificially created by urban planners, or may be the result of natural urban evolution; both could change functions and territories with the development of a city.

When studying urban evolution, the first challenge is to collect, extract and structure data so that they can be stored in a spatial-temporal database, storing very detailed information about a city's spatial composition at different times. As analyzing evolution on the raw data is not feasible, the second challenge is to summarize the composition of a city at a particular moment. In particular, in this step urban patches of a city are identified and annotated with signatures that contain summaries of their characteristics. The third challenge is to analyze and mine the obtained data to extract interesting knowledge on how a city changes over time. The last challenge is to develop simulation tools which aim at simulating a city's evolution based on rules which have been learnt from past experience. In this work, we are mainly focusing on the second challenge, also presenting some preliminary results on how to approach the third challenge.

In general, polygons play an important role in the analysis of spatial-temporal data as they provide a natural representation of geographical objects, such as buildings or countries. Furthermore, polygons can serve as models for spatial clusters and

(a) Year 1956 (b) Year 1976 (c) Year 1989 (d) Year 2002 (e) Year 2008

Figure 1.1: Building evolution of a neighborhood in Strasbourg, France between 1956 and 2008

can model nested and overlapping clusters. Polygons have been studied thoroughly in geometry, and powerful software libraries are available to manipulate, analyze, and quantify relationships among polygons. This work proposes novel polygon based data mining techniques to identify urban patches to study their evolution. We evaluate these techniques in case studies which center on the evolution of the building structure of the city Strasbourg in France. As depicted in Figure 1.1, polygons represent buildings in five different years of a neighborhood of Strasbourg, France.

The thesis's main technical contributions include:

1. Formal definition of the problem of finding uniform regions in spatial data as a maximization problem.

2. Developing a novel spatial clustering approach for identifying regions based on uniformity measures, which have to be expressed as reward-based fitness functions which are then maximized by the spatial clustering algorithm. The approach models the scope of spatial clusters as polygons and describes their characteristics using histogram-style distribution signature.

3. Introducing several interestingness measures to capture different notions of uniformity.

4. Introducing popular signatures which are frequently occurring distribution signatures in the subspaces of a spatial area of interest. A novel approach which summarizes the composition of a spatial dataset by annotating regions with popular signatures is presented.

5. Evaluating the proposed framework in a case study involving the building structure of the city of Strasbourg, France; in particular, the city is partitioned into uniform regions which are annotated with signatures and the benefit for domain experts of having such summaries is discussed.

The rest of the thesis is organized as follows: Chapter 2 gives the background information regarding urban planning with polygon definition and several polygonal distance functions. Chapter 3 explains the CLEVER clustering algorithm and its variant DCLEVER, which is more capable in handling spatial clustering problems. Chapter 4 introduces the spatial clustering approach to identify urban patches and different interestingness functions which capture different notions of uniformity. In Chapter 5, we present experimental results which identify urban patches based on three different notions of uniformity for neighborhoods of the city of Strasbourg in France. Chapter 6 concludes the thesis.

Chapter 2

Background and Related Work

2.1 Urban Evolution and Planning

"An urban area is characterized by higher population density and vast human features in comparison to areas surrounding it. Urban areas may be cities, towns or conurbations, but the term is not commonly extended to rural settlements such as villages and hamlets"[3]. More specifically, an urban area usually comprises several residential areas, industrial and business areas, and complex human settlement of variable size and industrial equipment with administrative functions.

Analyzing the past thirty years, urban evolution can be distinguished into two different categories. The first type of urban evolution usually occurs in the early period of city development, it is also called urban explosion in the territory. For instance, the urban evolution that happened at the end of 20th century only considers the efficiency rather than the comfort of urban living. The development of urban functions, such as habitations, commercial, services, and storage, usually needs the spatial support

outside the building realm[4]. Urban explosions sometimes incur great social expense without an explicit plan in terms of city organization. Therefore, urban explosions have a predominant feature, which can be depicted as spatial-territorial centrifugal expansion to different evolutionary poles with tendencies toward influencing and justifying future developments (Figure 2.1). This phenomenon is accompanied by a variety of other evolutionary trends related to movement and population dynamics, and dramatically changes the transportation module of the city.

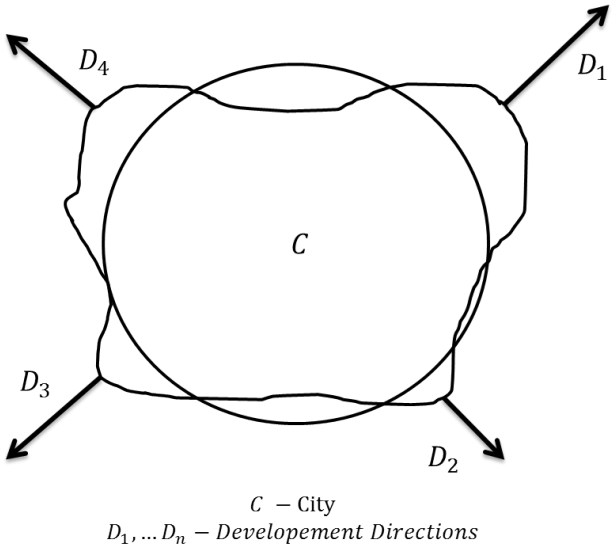


Figure 2.1: Urban Explosion Directions

Urban areas start experiencing another type of evolution even before being fully developed by spatial-territorial explosion, which is called *centripetal evolution*[4]. Centripetal evolution is characterized by city functional modifications, adaptations, restructuring and implementation. This phenomenon usually occurs in the center area of the city, as well as the areas with great development potential. In big cities, centripetal evolution is summarized by extending the commercial buildings,

transforming among different building types, positioning of important landmarks in specific locations in the city, and demolition of low-rise buildings and constructing of medium-and high-rise buildings, etc. Consequently, the second stage of urban evolution – centripetal evolution – focuses more on city structure change instead of spatial enlargement.

Rapid growth of urban areas makes it more and more attractive for people to live in the urban area. Along with the trend of people moving from villages and farms to cities, the rapid growing cities like Chicago (Figure 2.2) in the late 19th

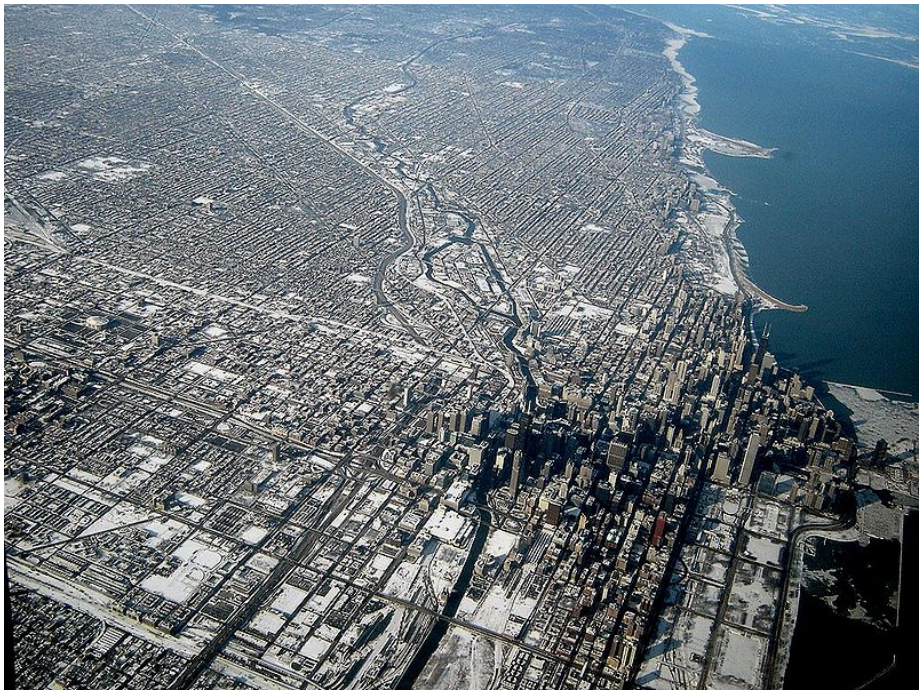


Figure 2.2: Chicago Downtown Aerial View (from Wikipedia)[1]

century and Mumbai a century later became the symbol of this rural-urban migration. "Urbanization, urbanisation or urban drift is the physical growth of urban areas as a result of global change"[1]. Urbanization can describe a specific condition at a set

time, i.e. the proportion of total population or area in cities or towns, or the term can describe the increase of this proportion over time. "So the term urbanization can represent the level of urban relative to overall population, or it can represent the rate at which the urban proportion is increasing"[1].

Today, this type of rapid urban growth usually occurs in developing countries, and may be attributed to the growth of new job opportunities. The rate of urbanization varies between countries, but Figure 2.3 depicts the rapid urbanization around the world. According to Figure 2.3, half of the world's population lived in urban areas at the end of 2008. There are many reasons for people to move out of rural area into big cities. Firstly, the desire of an individual or company to reduce the commuting time and cost associated with urban living. Secondly, people intend to take advantage of the increasing opportunities for jobs, housing, education and transportation in an urban area. Also, there are better basic services as well as other special services available in cities that are not found in rural areas. Lastly, urban areas usually have a great variety of entertainment activities and diverse social communities which makes it attractive to move to cities.

There are different types of urbanization that can be identified depending on the style of architecture, planning methods, as well as historic growth of areas. In the early stage of the developed area, urbanization traditionally showed the trend of concentration of human activities. This phenomenon is called *"in-migration"*[1], and usually refers to the migration from former colonies and similar areas to the center of the cities. Another interesting type of urbanization is called *"counter urbanization"*[1] or *"suburbaniztion"*[1], which is represented by the population flow

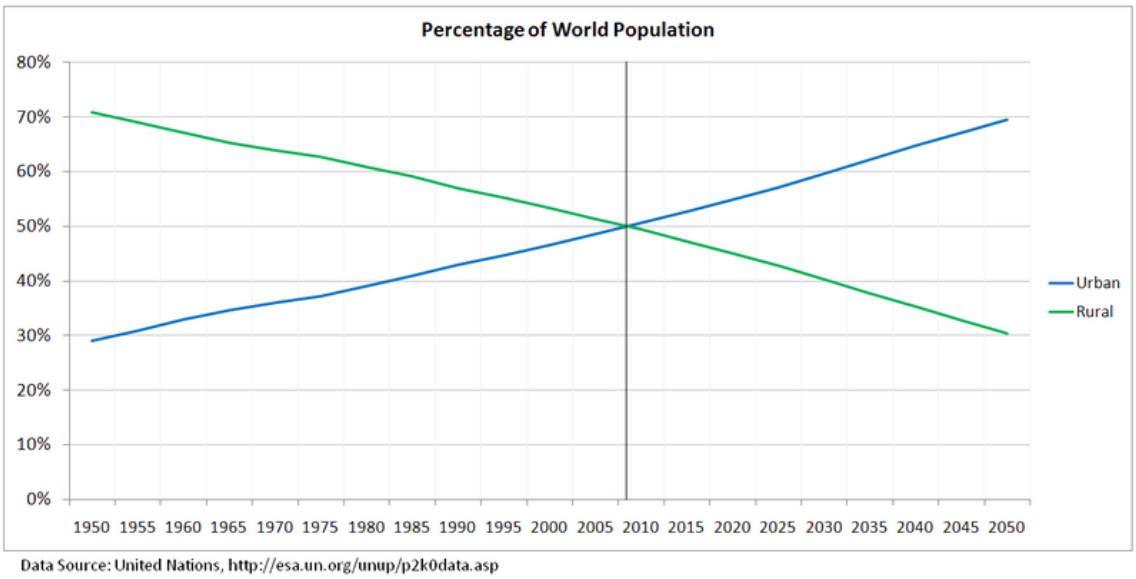


Figure 2.3: Percentage of World Population[1]

that shifts outward from an urban center. Counter urbanization usually is a recent development of modern cities, and is caused by improved public transportation, environmental pollution, and extremely high density populations in downtown areas.

However, there are several important effects caused by urbanization that need to be discussed. Firstly, the environmental problem caused by urbanization has become a growing concern. Specifically, the heat generated by urban and industrial areas is highest in city centers. This phenomenon is known as the *"urban heat island"*[4], which is exacerbated by less vegetation and exposed soil in urban area. Hence, during warm daylight hours, less evaporative cooling in cities causes surface temperatures to rise higher than rural area[1]. Secondly, the economic changes in urban areas is also significant. In contrast to the fast growth of cities, a major phenomenon occurs in rural area which is known as *"rural flight"*[1]. Because of the rural flight, the size of labor markets in rural area is shrinking dramatically. As shown in Figure 2.4,

urban areas attract more younger people than rural areas. Thus, it is very difficult for rural families, especially small families, to improve their standard of living due to the outgoing flow of the younger population. Similar problems are more severe in the developing world, causing rising inequality between rural areas and cities.

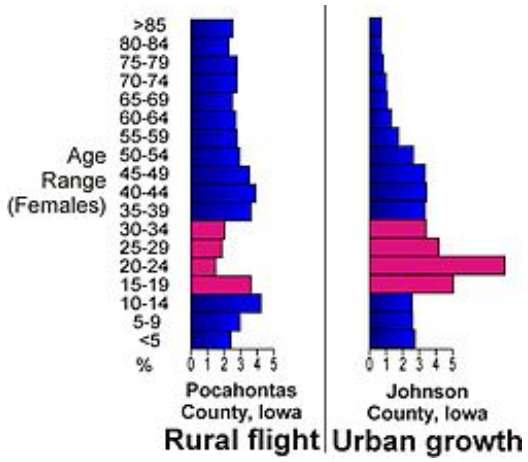


Figure 2.4: Population Comparison Between Rural and Urban Areas[1]

The evolution of any urban area or city reflects countless decisions and actions from the time of initial settlement to the present. Urban planning has always been a main concern for a long time: abundant evidence has been unearthed in the ruins of cities in China, India, Egypt, Asia Minor, the Mediterranean world, and South and Central America. Early examples of urban planning include orderly street systems with rectilinear and sometimes radial patterns; division of a city into specialized functional quarters; development of commanding central sites for palaces, temples and civic buildings; and advanced systems of fortification, water supply, and drainage[5]. Urban planning generally refers to the design and regulation of the use of space that focuses on housing, water system, transportation, etc. The various fields that play important roles in urban planning includes social and political concerns, engineering

and architecture etc.

In modern societies, more and more city planners have started to analyze the urban evolution issues, including land-use planing, densities, number and location of building permits, traffic flow, land price and rents, etc.[5]. Because public construction requires higher and higher expenses from the government, it becomes extremely important for a city to pursue a long term plan based on its own characteristics and needs. Moreover, the city blueprint should not only include the explicit plan on how to use the land, but also measure the benefits of social and physical infrastructure planning. The inability to create or implement a city blueprint is one of the most glaring failures in urban development.

2.2 Spatial Data Mining

Data mining is an important and an extremely active research field in the Computer Science area that focuses on finding valid, interesting, and useful patterns with large datasets. It has been playing an evolutionary and crucial role in laying the foundation for the next generation of major advances in many domains such as geography, biology, medicine, and social and political science.

With a variety of research domains in the data mining area, spatial and spatial-temporal data mining is becoming more and more interesting. Spatial data mining is one of the research areas in the data mining field that focuses on spatial datasets. However, extracting interesting patterns from spatial datasets is more complicated than from traditional datasets due to the complexity of the data types, spatial relationships, and spatial autocorrelation. As the rapid growth of the number of sensors

and their precision level, the explosive widespread use of large spatial datasets emphasizes the urgent need for automating the discovery of spatial knowledge. All of the difficulties stated above limit the usefulness of conventional data mining techniques for extracting interesting spatial patterns. The algorithms or tools for extracting spatial knowledge from large datasets are extremely important for many individuals or organizations making decisions; for example, the National Aeronautics and Space Administration (NASA), the National Imagery and Mapping Agency (NIMA), the National Cancer Institute (NCI), and the United States Department of Transportation (USDOT). Thus, spatial knowledge extraction applications are widely studied across public safety, transportation, earth science, epidemiology, and environmental management.

Moreover, spatial data mining usually is more challenging as it contains more complex data objects, such as trajectories and polygons. In conventional datasets, non-spatial attributes are used for characterizing non-spatial relations among data objects. However, as there is a need for defining spatial locations and extent of spatial objects, spatial datasets consist of spatial attributes in addition to non-spatial attributes. The spatial attributes exclusively depict the geographical information including longitude, latitude, convexity, elongation, and elevation. In contrast to explicit objects in non-spatial attributes, it is not straightforward to analyze the implicit relationships based on spatial attributes such as overlap, intersect, and orientation (Table 2.1). One possible way to deal with implicit spatial relationships is to materialize the relationships into traditional data input columns and then apply classical data mining techniques[6]. In order to reduce the loss of information during

Non-spatial Relationships (Explicit)	Spatial Relationships (Implicit)
Arithmetic	Set-oriented: union, intersection, membership
Ordering	Topological: meet, within, overlap
Is instance of	Directional: North, NE, left, above, behind
Subclass of	Metric: e.g., distance, area, perimeter
Part of	Dynamic: update, create, destroy
Membership of	Shape-based and visibility

Table 2.1: Non-spatial and Spatial Attributes Relationships[7]

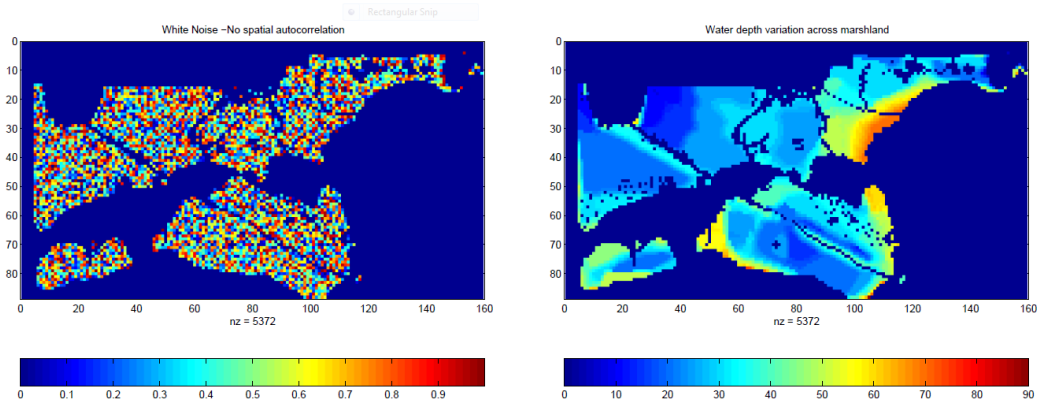
this materialization process, developing a model or technique to incorporate spatial attributes into the spatial data mining process is needed.

Traditional non-spatial data objects are considered as independent observation samples, whereas spatial data objects are usually autocorrelated. The property of similar things to cluster in space is so fundamental that geographers have elevated it to the status of the first law of geography: *"Everything is related to everything else but nearby things are more related than distant things"*[8]. Figure 2.5[7] demonstrates the difference between independent distribution dataset and spatial dataset with autocorrelated distribution.

Hence, the algorithms or techniques that ignore spatial autocorrelation usually perform poorly in the presence of spatial data. In reality, common sense or the domain experts' needs are converted into multiple computational constraints that are applied in the spatial data mining algorithms.

2.3 Polygon Model for Spatial Clustering

In geometry, a *polygon* is a flat shape consisting of straight lines that are joined to form a closed chain or circuit[9]. The segments that form the polygon figure are



(a) Attribute with an Independent Identical Distribution

(b) Attribute with Spatial Autocorrelation

Figure 2.5: Independent Identical Distribution and Spatial Autocorrelation

called *edges* or *sides*, and the points where two edges meet are the polygon's *vertices* or *corners*. Many types of polygons have been adopted in different domains to suit various needs. Instead of complex polygons that cross themselves, geographers and city planners are more interested in the polygons with closed polygonal chain and *simple polygons* which do not self-intersect. Simple polygons that are adopted in spatial analysis can be roughly classified into two types: *convex*, and *concave* polygons. In Figure 2.6, a convex polygon can be generalized as following properties of a simple polygon:

1. Every internal angle is less than or equal to 180 degrees.

2. Every line segment between two points inside the polygon remains inside or on the boundary of the polygon.

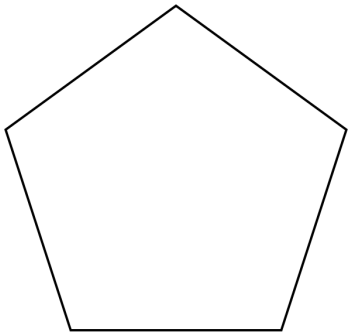


Figure 2.6: Convex Polygon

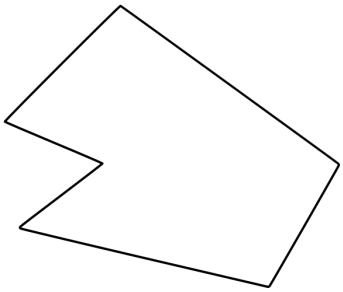


Figure 2.7: Concave Polygon

In contrast, concave polygon always has at least one interior angle with a measure that is greater than 180 degrees as in Figure 2.7.

Any polygon, simple or complex, convex or concave, has as many corners as it has sides. Each corner is characterized by an interior angle and an exterior angle. Basically, the interior angle is classified as the angle that faces the inner side of the polygon figure, and the exterior angle is the supplementary angle to the interior angle. In general, the sum of all the interior angles is related with the number of sides of the polygon figure. If a *n-gon* has n segments, the sum of degrees of all the interior angles can be computed using the following formula $\sum = 180 - \frac{360}{n}$. Although the interior angles can be determined using the formula, the sum of the

exterior angles of polygon will always be 360 degrees.

The area of the polygon is the size of the 2-dimensional region enclosed by the polygon[9]. The centroid, or geometric center, is the intersection of all the straight lines that start from vertices and divide the polygon into two equal regions. Suppose a simple polygon P has n segments where $n = 5$, then P can be represented by Figure 2.8:

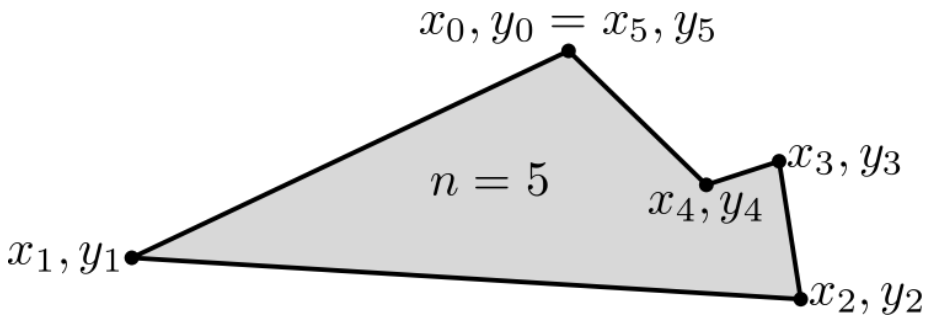


Figure 2.8: 5-segment Polygon

Generally, the area A of P with n segments can be determined by the following formula:

$$A = \frac{1}{2}\sum_{i=0}^{n-1}(x_i y_{i+1} - x_{i+1} yi) \tag{2.1}$$

Suppose point C is the centroid of P, the coordinates of C_x and C_y are given by Formula 2.2 and Formula 2.3:

$$C_x = \frac{1}{6A}\sum_{i=0}^{n-1}(x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \tag{2.2}$$

$$C_y = \frac{1}{6A}\sum_{i=0}^{n-1}(y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \tag{2.3}$$

In a broad sense, the definition of the area and centroid of the polygon might vary in different domains.

Among different spatial data types, the polygon is a very complicated and important model in spatial data mining applications that analyze the relationships and changes in spatial datasets. Moreover, the polygon model represents specific spatial objects more naturally and efficiently. Polygon analysis is particularly useful to mine relationships between multiple, related datasets, as it provides a useful tool to analyze discrepancies, progression, change, and emergent events[10]. Figure 2.9 shows an example of using polygons to represent the changes of earthquake areas around the world. The polygons in the figure provide a clear and recognizable result of the earthquake regions to domain experts by denoting the location and shape of the earthquake regions.

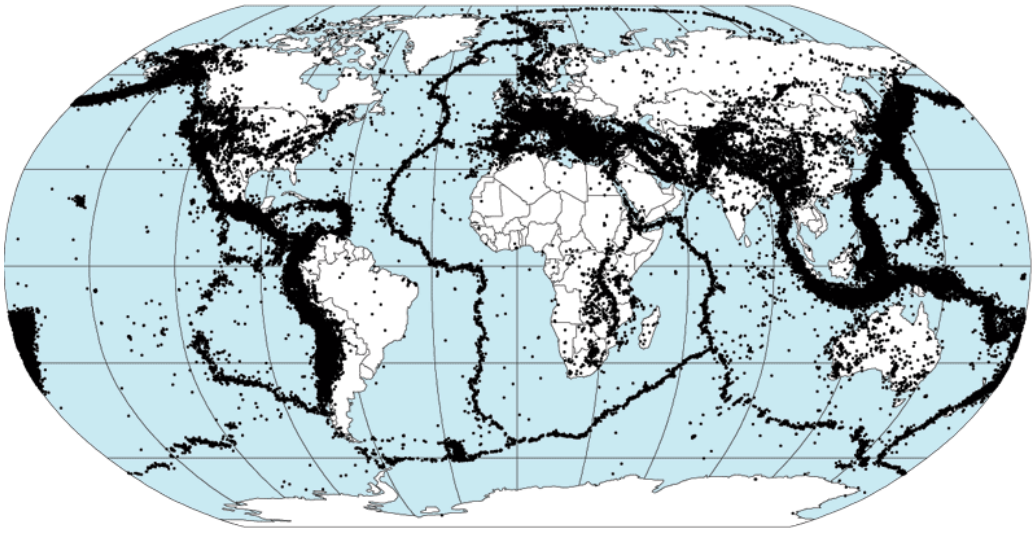


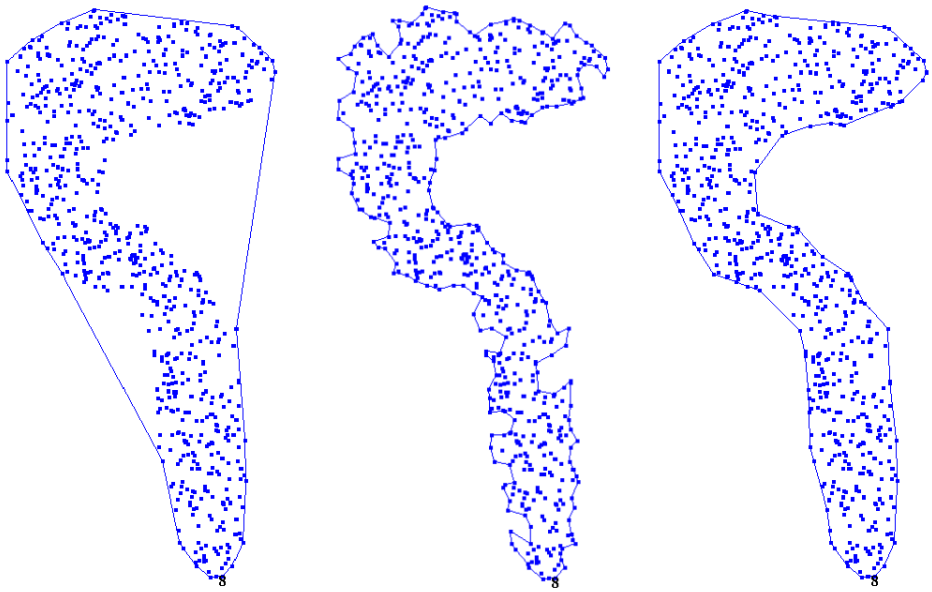
Figure 2.9: Representation of Spatial Changes by Using Polygons[11]

Convex Hull Method
Voronoi Diagram Method
Characteristic Shapes
Grid-based Method (S-shapes and R-shapes)
Gift-wrapping Method (Concave Hull Algorithm)
Alpha Shapes
A-shapes
Density Contouring Method (DContour)

Table 2.2: Polygon Model Generating Algorithms[12]

There are plenty of algorithms available regarding generating a polygon model based on a set of points, some of which are listed in Table 2.2. Since the computational complexity increases dramatically as the precision of polygon enhances, it is essential to generate efficient polygon model without a dramatic increase of the computational cost. Three typical polygon models generated by the same set of points are shown in Figure 2.10.

It is obvious that Figure 2.10(a) is not a good polygon model because it includes too much empty space, which is not relevant to the spatial object. The model with too much irrelevant information leads to inaccurate results by data mining algorithms, or wrong conclusions in statistical analysis. Therefore, it is necessary to generate a polygon model that excludes irrelevant information. On the other hand, Figure 2.10(b) is not an ideal model either due to the extremely high polygon complexity. Although it does not have large empty areas, it is more complex than it should be. This phenomenon is called overfitting, and it is computationally expensive to import such models into data mining algorithms. Figure 2.10(c) has the best overall quality among all three models. It represents the shape of the points set very well, and does not have too many edges and cavities. Therefore, it is more efficient



(a) A smooth polygon with large empty areas

(b) A tight polygon with too many edges and sharp angles

(c) A smooth polygon without large empty areas and with less cavities

Figure 2.10: Different Polygons Generated by Existing Algorithms[12]

in terms of storage space and processing cost compared to the other two models. In general, the desired polygon model should have the following properties for spatial data mining systems[12]:

1. The polygons should reflect the density of points in the dataset. Large empty areas inside the polygons are not desirable.

2. The polygons should be as smooth as possible: A smooth polygon is the one that does not have too many sharp angles (zigzags), and the number of edges is as small as possible. A very tight polygon that has too many cavities is not desirable.

Based on the constraints listed above, most of the polygon generation algorithms are not suitable for spatial data mining problems. However, the two-step fitness function method has proved to be effective in generating balanced polygon models.

According to the criteria of the effective model, it is desirable that the generated polygon has a small area as well as a short perimeter. Because area and perimeter are inversely proportional in most cases, a fitness function that is able to find the balance between smaller area and shorter perimeter is needed. Given the polygon p we can define the fitness function $f(p)$[12] as:

$$f(p) = area(p) \times perimeter(p) \tag{2.4}$$

where $area(p)$ is the area of polygon p and $perimeter(p)$ is the perimeter of polygon p. Fitness function Formula 2.4 is used to plug into multiple existing polygon model generation algorithms to obtain the optimized parameter setting. As stated above, polygons with smoother edges (lower number of edges and cavities) are preferred to polygons with a complicated shape. Hence a polygon smoothing step is included in the model generated in the first step, referred to pf, in order to reduce the number of edges under certain constraints. Generally, the second step is defined as a fitness function $f_2(p)$[12]:

$$f_2(p) = numEdges(p) \tag{2.5}$$

where $numEdges(p)$ represents the total number of edges in the polygon p. Moreover, Fitness function Formula 2.5 should be minimized with the restrictions defined as follows:

$$area(p) < area(pf) \times (1 + th\_area),\ (0 < th\_area < 1) \tag{2.6}$$

where th_area is the area increasing threshold,

$$numEdges(p) < numEdges(pf) \times (1 - th\_edges),\ (0 < th\_edges < 1) \qquad (2.7)$$

where th_edges stands for the number of edges decreasing threshold. The polygon model with a minimum number of edges whose area and number of edges meet both constraints is considered as the fittest model. By adopting the two-step fitness function method in Characteristic Shapes algorithm[13], a set of smooth polygons can be generated without cavity and overfitting problems based on Complex8 dataset as shown in Figure 2.11 and Figure 2.12.

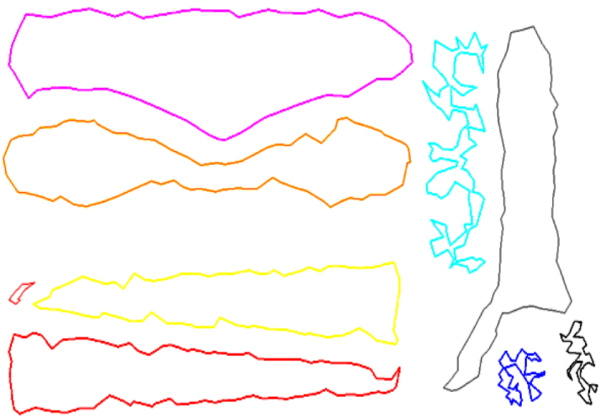


Figure 2.11: Polygons Generated for Complex8 Dataset at First Step

Figure 2.11 depicts the polygon group generated after the first fitness function, and Figure 2.12 illustrates the result after the smoothing process that occurs at the second step. It is obvious that the polygons smoothed by the second fitness function contain fewer edges while keeping the area increases less than the threshold. Thus, the existing polygon model generation algorithms work well with two fitness functions, which ensure quality as well as effectiveness.

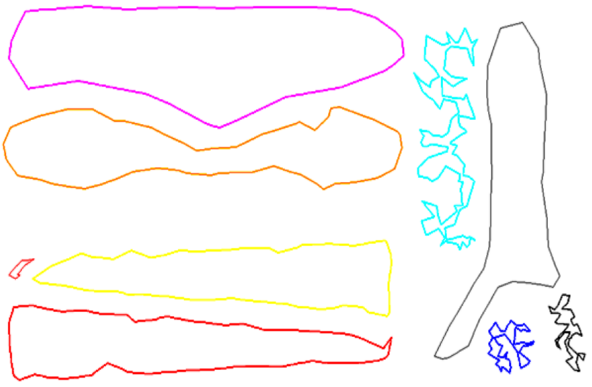


Figure 2.12: Polygons Generated for Complex8 Dataset at Second Step

2.4 Polygon Distance Functions

Spacial datasets can be divided into point spatial datasets, trajectory spatial datasets, and polygonal datasets. Points can be easily represented by using longitude and latitude, yet trajectory and polygonal datasets are more complicated in nature. Furthermore, it is very common that several spatial objects lie inside the same region that is shared by one or two polygons, e.g. rivers or bridges cross the lakes or small roads cut through highways. Therefore, it is essential to represent spatial objects accurately in order to do further analysis. Generally, spatial objects are represented by both spatial and non-spatial attributes that comprehensively describe spatial structure and organization information. The spatial structure usually includes the information related to geography knowledge, for example location, shape, height, etc. On the other hand, non-spatial attributes describe other information of the spatial object including age, texture, category, etc. Dissimilarity function, also called distance function, is used for measuring the difference or distance between two polygons along with the problems associated with their use. There are many existing dissimilarity functions

particularly designed for polygonal models that take both spatial and non-spatial attributes into account, and they will be introduced in the following paragraphs.

1. **Centroid Distance:** because a polygon has complex shape, the simplest way to approximate a polygon object is to represent each polygon as a single representative point. Hence, the distance between two polygons is the distance between two representative points. Figure 2.13 shows the distance between polygon $P1$ and $P2$ by using centroid distance function. The centroids are calculated based on Formula 2.2 and Formula 2.3. However, it is not very effective to use centroid distance function in this situation because the extent of the polygons is ignored by centroid approximation. Moreover, it makes this case even worse if the centroid is outside of the polygon.

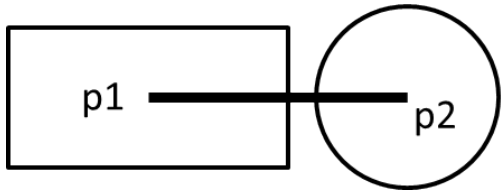


Figure 2.13: Centroid Distance Function

2. **Minimum Bounding Rectangle Distance:** the minimum bounding rectangle of a polygon can be used for approximating the shape of the polygon object. This distance function finds the distance between two polygons by using the distance between two minimum bounding rectangle. Specifically, the distance between two rectangles is the distance between the centers of two rectangles. Unfortunately, drawbacks exist with the centroid distance function that also limit the efficiency of minimum bounding rectangle distance functions due to

the ignorance of polygon extents.

3. **Separation Distance:** given two polygons $P1$ and $P2$, the distance d can be defined by the separation distance as defined by Formula 2.8

$$d = min\{d(Q1, Q2)\} \tag{2.8}$$

where $Q1$ and $Q2$ are the points set for polygon $P1$ and $P2$ respectively. From Formula 2.8, the distance between two polygons is determined by the minimum distance between any pair of points in $P1$ and $P2$. This distance function solves the bottlenecks of centroid distance function and minimum bounding rectangle distance by taking the extents of the polygons into account. However, it is not quite satisfactory for geospatial application because the distance will be zero even if two polygons share a point. As in Figure 2.14, the distances between $P1$

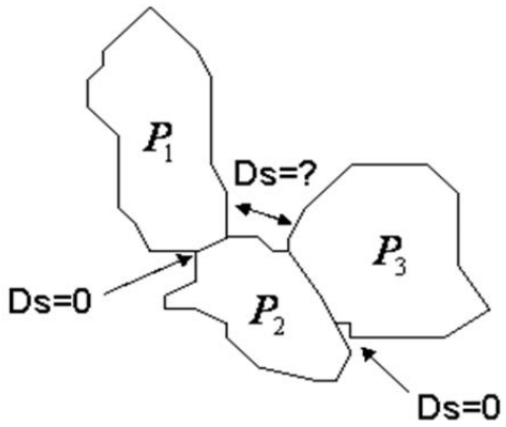


Figure 2.14: Separation Distance Function[14]

and $P2$, and $P2$ and $P3$ are zero, based on the definition of separation distance.

Thus, the distance between $P1$ and $P3$ should be zero based on the transitive relationship. However, it is obvious that $P1$ and $P3$ are separated and do not share a border in Figure 2.14. Hence, the separation distance function does not work well among adjacent polygon models.

4. **Min-Max Distance:** another way to measure the distance between polygon models is to find the maximum or minimum distance between each pair of vertices of polygons[14].

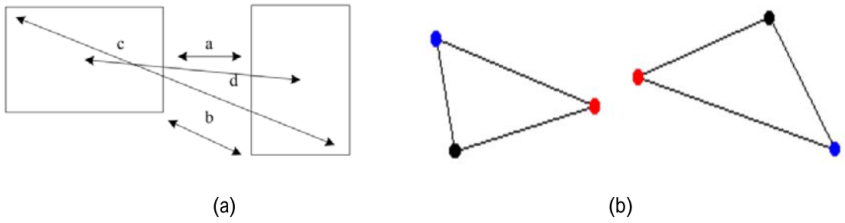


Figure 2.15: Separation Distance Function[14]

Several distance measurements are shown in Figure 2.15(a) including: separation distance (a), minimum distance between vertices (b), maximum distance between vertices (c), and centroid distance (d). The Min-Max distance b and c, violates the intuitive distance measurement in the geospatial application, causing confusion and inaccurate results. Moreover, the Min-Max distance function can not take the polygon shape into account as shown in Figure 2.15(b). In other words, the shortest and longest distance between two polygons (shown in red and blue respectively) remain the same as long as the specific vertices are fixed. Therefore, it is not feasible to use the Min-Max distance function in spatial datasets as the shape of the objects is very important.

5. **Hausdorff Distance:** Conventionally, a polygon object is represented by a set of points. In order to overcome the drawbacks of previously stated distance functions, the Hausdorff distance function has been proposed. It measures the distance between two sets of points A and B by using Formula 2.9.

$$h(A, B) = \max_{a \in A}(\min_{b \in B} d(a, b)) \tag{2.9}$$

where a and b are points sets of A and B, and $d(a, b)$ is the distance between two points a and b. Based on this defined formula, the distance between two polygons is determined by the maximum distance of a point in A to the nearest point in B. Generally, the centroid distance function usually performs well for convex polygon objects; however the Hausdorff distance function gives a more accurate distance between concave polygons, as can be seen in the second and third examples in Figure 2.16. The distance will often be underestimated or overestimated if the centroid of the polygon falls outside of the area of the polygon. In comparison, the Hausdorff distance D_h gives a more accurate solution than any of the discussed distance functions so far.

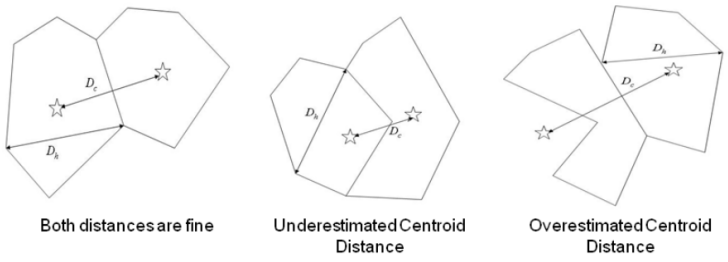


Figure 2.16: Hausdorff and Centroid Distance Comparison[14]

6. **Fréchet Distance:** The Féchet distance function is intuitively defined by

imagining that a dog and its handler are walking on their respective polygon boundaries. Although the speed of the walking is adjustable, both can not go backward. Therefore, the Fréchet distance between these two polygon boundaries is defined as minimal length of any leash necessary for the dog and the handler to move from the starting points of the two curves to their respective endpoints. Let S be a metric space. A curve A in S is a continuous map from the unit interval into S. Let A and B be two given curves in S. Then, the Fréchet distance between A and B is defined as the infimum over all reparameterizations α and β of $[0,1]$ of maximum over all $t \in [0,1]$ of the distance in S between $A(\alpha(t))$ and $B(\beta(t))$[15]. In mathematical notion, the Fréchet distance $F(A,B)$ is:

$$F(A,B) = \inf_{\alpha,\beta} \max_{t\in[0,1]} \{d(A(\alpha(t)), B(\beta(t)))\} \tag{2.10}$$

where d is the distance function of S. Because the Fréchet distance function takes the flow of two curves into account, it provides a better measurement of similarity for curves over the Hausdorff distance function. However, it is important to mention that the Fréchet distance function is used for shape matching instead of measuring the geographic distance between two polygons in spatial data mining applications. The Fréchet distance function has been shown to have a better performance than Hausdorff distance function in shape measurement.

2.5 Related Work

Work in[16, 17] proposed a region discovery framework based on a fitness function to maximize. The framework adapts four representative clustering algorithms, exemplifying prototype-based, grid-based, density-based, and agglomerative clustering algorithms to optimize the fitness function. The fitness function is defined according to the application, and the goal is to model the interestingness of a region. Other works find uniform regions for spatial regression[18, 19] by using quite different methods. Both approaches partition the space into regions related with different regression functions. Uniformity in this work is associated with a set of points sharing the same or a similar relationship between a dependent variable and a set of independent variables.

Joshi et al.[20] proposes a dissimilarity function for clustering geo-spatial polygons. The proposed dissimilarity function takes into account different characteristics of the polygon separated in different groups: non-spatial attributes, intrinsic spatial attributes, and extrinsic spatial attributes. The dissimilarity function computes the dissimilarity between polygons as a weighted function that computes the distance between two polygons in the different attribute spaces. This approach is different from our approach, which supports plug-in interestingness functions that allow assessing cluster quality using non-distance-based interestingness measures. Moreover, our approach generates clusters which are contiguous in the subspace of the spatial attributes.

The topic discovery approach[2, 21] to identify urban regions has gained some popularity recently. The idea of this approach is to "learn" popular signatures,

called topics, from small sub-regions of a city. Then the next regions that have a strong association with a single or a mixture of topics are identified. There are two major differences between our approach and the topic discovery approach. Firstly, our approach is supervised based on a domain expert's notion of uniformity, which has to be expressed by a plug-in interestingness function. But popular signatures are identified by an unsupervised topic discovery approach in the other approach. Secondly, the topic discovery approach requires an apriori given partitioning of the city as an input. However our approach uses spatial clustering algorithms to determine such a partitioning which is optimal with respect to a given notion of uniformity. For example, the popular signature clustering approach that was proposed in this paper directly identifies the scope of a particular popular signature, which is the union of contiguous spatial clusters matching this signature.

Chapter 3

CLEVER

3.1 CLEVER

Before the discussion of discovering uniform regions in the next chapter, we need a spatial clustering algorithm that is capable of finding contiguous spatial clusters maximizing a plug-in reward function, which captures a particular notion of uniformity. A spatial clustering algorithm named CLEVER[22, 23] will be adapted for the task to find uniform clusters in spatial datasets. As the algorithm relies on computing distances between polygons when used to cluster buildings, we first introduce the distance function we use for this purpose, before discussing CLEVER in more detail.

CLEVER is a prototype-based, k-medoid-style[24] spatial clustering algorithm which employs randomized hill climbing to maximize a plug-in reward function. Reward functions are assumed to have the following form when assessing the quality of

a clustering $X = \{c_1, \ldots, c_k\}$:

$$q(X) = \sum_{c \in X} reward(c)^{\beta} = \sum_{c \in X} i(c) \times |c| \tag{3.1}$$

where $|c|$ denotes the number of objects in a cluster c, and $i(c)$ is an interestingness function which assesses how interesting the cluster c is. Three different such interestingness functions will be introduced in Chapter 4. Moreover $\beta \geq 1$ is a parameter which determines how much reward is put on cluster size; β indirectly controls the numbers of clusters in X, as cluster size is rewarded using a non-linear function. Usually fewer clusters are obtained when larger values for β are used. The reward function assesses the quality of a clustering as the sum of the rewards of the individual clusters; The pseudo-code of CLEVER is given in Algorithm 1.

Input: *Dataset O, k', neighborhood-size, p, q, β, object-distance-function d,* imax
Output: *Clustering X, fitness $q(X)$, rewards for cluster in X*
Algorithm:

1. *Create a current solution by randomly selecting k' representatives from O.*
2. *If* imax *iterations have been done terminate with the current solution.*
3. *Create p neighbors of the current solution randomly using the given neighborhood definition.*
4. *If the best neighbor improves the fitness q, it becomes the current solution. Go back to step* 2.
5. *If the fitness does not improve, the neighborhood of the current solution is re-sampled by generating $p \times q$ more neighbors. If re-sampling does not lead to a better solution, terminate; otherwise, go back to step* 2 *replacing the current solution by the best solution found by re-sampling.*

Algorithm 1: CLEVER algorithm pseudo-code

CLEVER maintains a current set of representatives which are objects in the dataset and forms clusters by assigning the remaining objects in the dataset to the

closest polygon in the representative set. It samples p solutions in the neighborhood of the current representative set by adding, deleting, and replacing representatives. This process continues as long as a better clustering with respect to $q(X)$ is found. The algorithm begins its search from a randomly created set of k' representatives, where k' is an input parameter of the algorithm. CLEVER has recently been generalized to cluster complex spatial objects, such as lines and polygons.

To give an example let us assume we cluster a dataset $O = \{o_1, \ldots, o_{200}\}$ with k' set to 3. In this case, the algorithm starts with a random representative set, let us say $\{o_3, o_9, o_{88}\}$, and forms clusters by assigning the remaining 197 objects to the closest representative which takes $O(k \cdot (n - k))$, where n is the number of objects in the dataset and k is current number of representatives. Next, the algorithm samples p new clusterings in the neighborhood of the current solution by inserting, deleting, or replacing representatives. For example, assuming p is 3, the algorithm might create clusterings for the representative sets $\{o_3, o_9, o_{88}, o_{92}\}$,$\{o_3, o_{88}\}$, and $\{o_3, o_{17}, o_{88}\}$, all of which have been obtained by a single insertion/deletion/replacement applied to the current representative set $\{o_3, o_9, o_{88}\}$. Next, the algorithm computes $q(X)$ for these three clusterings, and if the best of the three clusterings improves the clustering quality, its representative set becomes the new current solution; otherwise, the algorithm terminates. In general, assuming that CLEVER runs for t iterations its complexity[1] is of the order of $O(t \cdot p \cdot k \cdot n)$ with t and k usually being much smaller than n.

[1]The analysis of the complexity of CLEVER is further complicated by the fact that the number of representative/clusters change between iterations; that is, the algorithm might start with $k' = 100$ clusters but the final clustering might contain 83 or 113 clusters. That is, CLEVER seeks for "optimal" number of clusters with respect to the dataset O and the fitness function q.

3.2 DCLEVER

As CLEVER needs to search the neighborhood intensively, distance computing becomes very expensive if CLEVER is applied on complex spatial objects. In order to address this problem, a variant of CLEVER — called DCLEVER — is created to avoid redundant distance computation. Hence, DCLEVER works like CLEVER except that CLEVER computes distance between objects on the fly, whereas DCLEVER first reads the entries in the distance matrix.

DCLEVER requires a distance matrix D as an additional input parameter. The distance matrix has to be created in a pre-processing stage. The pre-processing is a stand-alone computing procedure that takes a dataset O as input, and produces distance matrix D as output. The pseudo-code of distance computing procedure is shown in Algorithm 2.

Input: *Dataset O, object-distance-function d*
Output: *Distance matrix D*
Algorithm:

1. *Compute the distance between objects based on input distance function d.*
2. *Store the distance value in an upper triangular matrix D.*

Algorithm 2: Distance computing procedure pseudo-code

In step 2, the distance matrix is generated as an upper triangular matrix. As in the case study that will be introduced in Chapter 4, the number of objects in spatial data mining problem is huge. By storing the distance matrix in upper triangular matrix, DCLEVER has lower memory consumption and shorter executing time compared to full distance matrix loading strategy.

3.3 Experiment

In this section, original CLEVER and DCLEVER are compared based on the earthquake dataset. The dataset contains 2000 objects with three attributes: longitude, latitude, and earthquake depth. The interestingness function used in the experiment rewards high variance with respect to earthquake depth, which means CLEVER and DCLEVER will maximize the cluster's depth variance. Hausdorff distance function is used in this experiment to compute the distance between different objects, other parameters are set as in Table 3.1.

Table 3.1: Experiment Parameter Setting.

p	q	β	k'	η	Threshold
20	20	2.8	8	2.0	1.2

CLEVER and DCLEVER are run separately under the same initialization and sampling rate. In order to load the matrix into the memory, DCLEVER takes $48.1s$ to load the distance matrix that is $16MB$ from the hard drive. Table 3.2 lists the computation time comparison between CLEVER and DCLEVER for 5 different runs.

Table 3.2: CLEVER and DCLEVER Comparison.

Run	No. of Iterations	No. of Clusterings	No. of Regions	CLEVER	DCLEVER
1	26	1660	15	$6.1s$	$48.1s + 3.9s$
2	30	2480	10	$8.3s$	$48.1s + 5.7s$
3	29	3500	7	$8.8s$	$48.1s + 6.2s$
4	40	2220	12	$9.1s$	$48.1s + 6.5s$
5	23	1770	14	$5.7s$	$48.1s + 4.1s$

Because DCLEVER eliminates the demands of distance computation that occurs in CLEVER by pre-processing, it is easier to apply expensive distance functions in DCLEVER such as Hausdorff distance function, Féchet distance function, etc.,

because DCLEVER computes the distance matrix in a pre-processing procedure. Moreover, DCLEVER increases the computation performance by 20% compared to the original CLEVER algorithm as in Table 3.2 although this does not compensate for $48.1s$, which is the distance matrix loading time of DCLEVER. As the computation time of CLEVER is positively correlated with the number of iterations and clusterings, a large number of iterations makes DCLEVER more attractive than CLEVER, e.g. when the number of iterations exceeds 700. However, the loading time of distance matrix for DCLEVER increases following $O(n^2)$ as the number of objects n increases, which kills DCLEVER with big datasets.

Chapter 4

Identifying Uniform Regions in a City

4.1 Problem Definition

Cities all around the world are in constant evolution due to numerous factors, such as fast urbanization and new ways of communication and transportation. However, the evolution of the composition of a city is difficult to follow and analyze. Since understanding the evolution of cities is the key to intelligent urbanization, there is a growing need to develop urban planning and analysis tools to guide the orderly development of cities, as well as to enhance their smooth and beneficiary evolution.

When studying urban evolution, the first challenge is to collect, extract, and structure data so that they can be stored in a spatial-temporal database, storing very detailed information about a city's spatial composition at different times. As analyzing city evolution using the raw data is not feasible, the second challenge is

to summarize the composition of a city at a particular point of time. In particular, in this step urban patches of a city are identified and annotated with signatures which are with summaries of their characteristics. The third challenge is to analyze and mine the obtained data to extract interesting knowledge on how a city changes with respect to time. The last challenge is to develop simulation tools which aim at simulating a city's evolution based on rules which have been learnt from past experience.

Spatial clustering groups the objects in a spatial dataset and identifies contiguous regions in the space of the spatial attributes. Spatial clustering algorithms can be used for hotspot discovery, change analysis, and data summarization. One important spatial clustering task is to create a partitioning of a given space into uniform regions based on a domain experts notion of uniformity, such as partitioning the space of a city into different urban patches. However, traditional clustering algorithms are not suitable for this task as they minimize distance-based objective functions, whereas assessing uniformity relies on non-distance based uniformity measures, such as purity, entropy or variance with respect to a continuous non-spatial attribute. In this work, we develop novel spatial clustering algorithms which identify uniform regions in a spatial dataset by maximizing a plug-in measure of uniformity.

So far we did not clearly discuss what distinguishes a uniform region of a city from one that is not uniform. In general, we assume that distribution signatures are used to characterize the objects that belong to an urban patch. Examples of such signatures include histogram-style building type signatures which give the proportions

of different building types that occur in an urban patch, such as 15% are commercial buildings and 85% are residential buildings. Moreover, the similarity between different building type signatures can be easily assessed: for example, we could take the Euclidian distance between the vectors associated with different building type signatures. More formally, we are interested in obtaining spatial clusters using the following maximization procedure:

Input: *a dataset O containing spatial objects belonging to p classes*
Task: *Find a spatial clustering $X = \{C_1, \ldots, C_k\}$ of O such that*

(1) $C_i \subseteq O$ for $i = 1, \ldots k$

(2) $C_p \bigcap C_q = \emptyset$

which maximizes the following objective function $\varphi(X)$:

$$\varphi(X) = \sum_{C \in X \text{ and } C' \in X \text{ and neighboring } (C,C')} d(s(C), s(C'))/b \qquad (4.1)$$

where b is the number of pairs of neighboring clusters in X, $s(C)$ denotes the signature of cluster C and d is a distance function which assesses the similarity of two signatures.

In summary, we are interested in obtaining a spatial clustering in which the average Euclidian distance between the signatures of neighboring clusters is as large as possible. It should be emphasized that only distances between neighboring clusters are considered in the definition of φ. In order to find uniform partitions, we can devise a search procedure which maximizes the disagreement of neighboring clusters with respect to their signatures.

However, developing a spatial clustering algorithm which directly maximizes $\varphi(X)$ is quite challenging, as this would require to identify and to keep track of

which spatial clusters are neighboring in order to compute $\varphi(X)$, which leads to quite significant clustering overhead, and to theoretical problems[1]. Consequently, we are using different heuristics to find uniform spatial clusters without having to deal with the question which clusters are neighboring, and rely on approaches which use simplified versions of $\varphi(X)$ instead; in particular:

1. We use prototype-based spatial clustering algorithms that are guaranteed to obtain contiguous spatial clusters without the necessity of knowing which clusters are neighboring. These algorithms maximize reward functions which encourage the merging of similar neighboring clusters and the splitting of non-homogeneous clusters if it leads to a significant increase in the total reward.

2. We reformulate the above optimization task in two ways:

    i. We make the problem supervised, by using interestingness functions which assess the quality of spatial clusters based on uniformity measures which capture a domain expert's notion of uniformity. Moreover, as we will see later, those uniformity measures assume that certain signatures are more desirable than other signatures. Two such interestingness functions will be introduced in Sections 4.2.1 and 4.2.2.

    ii. Instead of comparing the signatures of all neighboring clusters — as φ does —

[1]If prototype-based clustering algorithms, such as K-medoids or K-means are used, a Voronoi tessellation can be used to derive cluster models from the set of cluster prototype which are convex polygons; unfortunately, it is not computationally feasible to compute Voronoi cells in higher dimensional spaces, as the complexity of the algorithm is exponential with respect to the dimensionality of the dataset. Consequently, it is only feasible to compute the Voronoi tessellation in $1D$, $2D$, and for small datasets in $3D$. For density-based clustering algorithm the situation is even worse; for example, we are not aware of any methods which are capable of producing cluster models from a DBSCAN clustering.

we employ an approach which identifies a set of popular[2] signatures and then uses those signatures to annotate clusters. In particular, this approach seeks for a spatial clustering which maximizes the match of a cluster's signature with the closest signature in the popular signature set, as will be explained in Section 5.2.3.

4.2 Interestingness Functions for Uniform Regions

In this section, two interesting functions are described, which will be used later to identify urban patches based on two notions of uniformity: uniformity with respect to building sizes and uniformity with respect to proportions of building types.

4.2.1 Purity

The purity interestingness function is used for analyzing interestingness with respect to a categorical non-spatial attribute. Purity interestingness $i(r)$ of a cluster r is computed using the following formula:

$$i(r) = \begin{cases} 0 & max_c p_c(r) < th \\ & c \in cl(O) \\ (max_c p_c(r) - th)^\eta & \text{otherwise} \end{cases} \tag{4.2}$$

where $cl(O)$ is the set of classes in the dataset O, p_c is a function that computes the proportion of a class c in cluster r, $\eta > 0$ is the scaling factor, and $th > 0$ is the threshold. For example, assuming that $th = 0.4$, $\eta = 1$, and r contains examples of

[2]Popular signatures are distribution characteristics which occur frequently in contiguous subspaces of a spatial dataset.

3 classes distributing $(0.6, 0.3, 0.1)$: $i(r) = 0.6 - 0.4 = 0.2$ for cluster r. In general when using the purity interestingness function, we are interested to obtain clusters which are dominated by instances of a single category.

4.2.2 Low Variance Interestingness Function

The low variance interestingness function assesses the variance of a continuous attribute in a cluster; the interestingness of the cluster c is inversely proportional to the variance of the continuous attribute in the cluster:

$$i(r) = \begin{cases} 0 & var(c) > var(D) \\ 1 & var(c) = 0 \\ min(1, (log_{100} var(O))/var(c))^{\eta} & \text{otherwise} \end{cases} \tag{4.3}$$

where $var(c)$ is the variance for a continuous attribute in cluster c, $var(O)$ is variance of the continuous attribute in the dataset O, and $\eta > 0$ is the scaling factor. For example, assuming $\eta = 2$ and the variance of the continuous attribute in cluster c is 4% of its variance of the dataset, we obtain:

$$i(c) = (log_{100} 25)^2 = 0.0906$$

In general, if this interestingness function is used, low variance clusters are obtained, somewhat similar to regression trees.

4.2.3 Discovering Uniform Regions Using Popular Signatures

Analyzing change in urban environments allows identifying new trends in a city's composition, helps assessing if certain policies have been successful, and produces valuable knowledge to predict how a city will evolve in the future. However, change analysis for urban data is challenging, as it requires aggregation and summarization of the data to be useful for city planners and scientists. There are many ways to aggregate and summarize urban data. Past work[2, 25, 21] in this area relies on identifying urban patches which represent homogeneous areas in a city and then analyze how the scope and signature of urban patches change.

4.2.3.1 Features of the Proposed Urban Patch Analysis Framework

The urban patch analysis framework proposed in this thesis relies on:

1. Polygonal spatial clustering algorithms, which support plug-in interestingness functions to capture different notions of uniformity, are used to identify urban patches.

2. Histogram-style signatures are used to characterize the distribution of spatial clusters; in general, signatures are normalized vectors. Different types of signatures are used for different measures of interestingness.

3. Polygons are used as models of urban patches and concave hull algorithms are used to approximate the scope of an urban patch.

4. Evolution is analyzed by comparing how the scope and signatures of spatial clusters change.

5. Three change analysis approaches are proposed:

    - The first approach applies a spatial clustering algorithm with a plug-in uniformity measure and then directly analyzes how the scope and signatures of spatial clusters change.
    - The second approach, which is advocated in[2], clusters the signatures and replaces individual region signatures by popular signatures; popular signatures are signatures which occur frequently in contiguous subspaces of a city. Finally, change is analyzed by analyzing how the scope of each popular signature changed.
    - The third approach is an improvement of approach b. Here a spatial clustering algorithm is used to identify regions which exhibit particular popular signatures.

Determining the scope of a spatial cluster is a challenging task. The easiest approach is to compute the convex hull of the spatial objects in the cluster, but the obtained convex hull polygon is usually not very tight, frequently enclosing empty spaces, as can be seen in the scope visualizations in Figure 4.1, which uses the convex hull as a cluster model. Alpha shapes[26] and the concave hull algorithm[27] generalize the convex hull algorithm, allowing for the generation of much tighter polygons which might contain holes; both algorithms recently gained popularity and became part of spatial extensions of popular database systems, such as *Postgres*

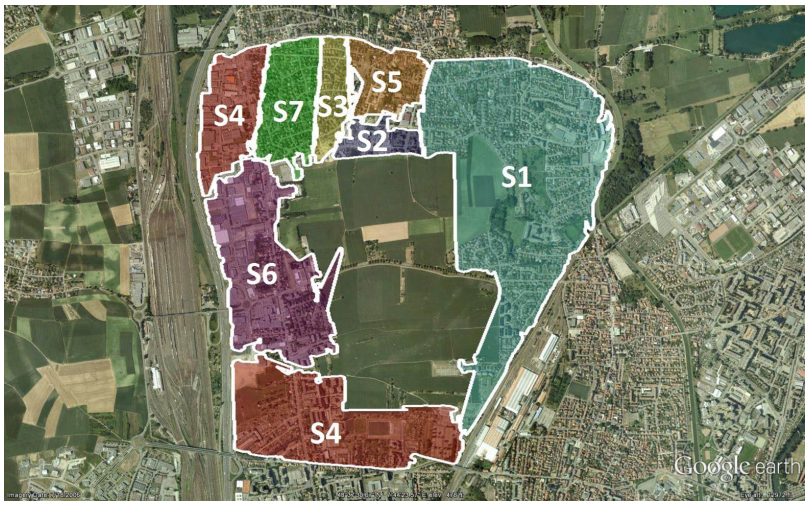


Figure 4.1: Example of a Spatial Clustering of Buildings Annotated by Popular Signatures

and *Microsoft SQL Server*. However, in cases that the density in spatial clusters is not uniform, these methods are generating less than perfect polygons; recent research[12, 28] tries to enhance these methods to deal with varying densities. In our proposed methodology we use the concave hull algorithm for computing the scope of spatial clusters; we believe this approach is more effective than using the convex hull algorithm, as it wraps a much tighter line around a set of polygons, resulting in less overlap with respect to the scope for neighboring clusters.

4.2.3.2 Distribution Signatures Used in the Thesis

Basically, we use two kinds of signatures in this thesis which correspond to two interestingness functions introduced earlier. The first type of signature captures the distribution of a continuous attribute in a cluster; in particular the following 7-value signature is used in the experiments:

Let μ be the mean value for building size in the datasets and σ the standard deviation for the observed building sizes.

1. Mean value for building size in the cluster, reported as a z-score based on (μ, σ)

2. Standard deviation s_C for building size in the cluster, reported as:

$$min(1, max(-1, log_{10}(s_C/\sigma))) * 0.5 + 0.5$$

3. Proportion of buildings whose size is below $\mu - \sigma$

4. Proportion of buildings whose size is above $\mu - \sigma$ and below $\mu - 0.25 * \sigma$

5. Proportion of buildings whose size is above $\mu - 0.25 * \sigma$ and below $\mu + 0.25 * \sigma$

6. Proportion of buildings whose size is above $\mu + 0.25 * \sigma$ and below $\mu + \sigma$

7. Proportion of buildings whose size is above $\mu + \sigma$

For example, the signature $(1, 0.5, 0, 0, 0, 0.6, 0.4)$ indicates that the mean value of the continuous attribute in cluster c is exactly one standard deviation larger than the mean value of the continuous variable in the entire dataset; its variance is identical with the variance in the dataset; 40% of the values are higher than the mean value of the dataset plus one standard deviation and the remaining 60% of the values of the continuous attribute are between 0.25 and 1 times of standard deviations plus the mean value of the continuous variable in the dataset.

A second type of signature is used to analyze the decomposition of a city with respect objects belonging to $p(p > 1)$ categories. In this case, signatures $S(V) = (s_l, \ldots, s_p)$ with $s_l + \cdots + s_p = l$ give the proportions of the examples in V belonging to

the p categories in a cluster. For example, the signature $(0.6, 0, 0, 0, 0.4, 0)$ indicates that 60% of the objects belong to the first category, and 40% of the objects belong to the fifth category.

Urban patch discovery and change analysis using these two signatures will be further discussed in the experimental evaluation section of this thesis.

4.2.3.3 Characterizing Spatial Clusters Using Popular Signatures

If urban patches are identified using spatial clustering in conjunction with the purity interestingness function, we could identify regions that are dominated by a single category, such as industrial areas with a large percentage of industrial buildings. However, many urban patches are characterized by particular proportions of class densities without a dominating class. For example, collective houses usually have a lot of garages next to them and these kinds of regions cannot be discovered by using a spatial clustering algorithm in conjunction with the purity interestingness function. This is the motivation for the following alternative approach which seeks to find regions which exhibit popular building type signatures which occur frequently in contiguous subspaces of a city. In general, we assume that urban patches serving different functions to the citizen of a city are characterized by significantly different building-type signatures, whereas different regions serving the same or a similar function, such as two industrial areas, exhibit similar building signatures. In general, when using popular signature clustering, the quality of a clustering X with respect to a signature set P is measured using the following quality measure:

$$Quality(X, P) = \sum_{c \in X} (\frac{|c|}{|X|}) * d(sig(c), closest(sig(c), P)) \tag{4.4}$$

where $X = c_1, \ldots, c_k$ denotes a spatial clustering and $sig(c)$ denotes the signature of a clustering c; P is a set of popular signatures; $closest(s, P)$ is a function which computes the closest signature in P to s; d is the Euclidian distance function; and '||' computes set cardinality, e.g. $\{a, b, c\} = 3$.

In summary, the quality of X with respect to P is computed as the weighted sum of the Euclidian distance of each cluster signature to its nearest signature in P. Each distance is weighted by the number of objects in the cluster. The Euclidian distance of a cluster signature to its closest signature in P receives a higher weight if this cluster contains more objects. As the closest signature in P will be used as a summary of each cluster, it is desirable that this signature in P is close to the cluster signatures; therefore, the Euclidian distance between those two signatures assesses the error in this approach.

Next, we introduce an algorithm to identify regions which exhibit popular signatures based on the quality measure introduced in the previous paragraph. It first collects signatures using a sampling approach; second, from the collected signatures it identifies a set of popular signatures using a clustering approach; third, it uses a spatial clustering algorithm to identify regions which exhibit one of those popular signatures; finally, a color display is created from the spatial clustering result in which the closest popular signature for each region is represented in a different color. The four procedures to identify and visualize regions which exhibit popular signatures are described in more detail:

Procedure 1: Extract a large number of signatures from randomly generated spatial clusters.

Step 1: randomly picking k representative objects in the dataset.

Step 2: Obtain k spatial clusters by assigning the other objects in the dataset to the closest of the k representatives.

Step 3: Compute the signatures for each cluster.

Step 4: Repeat step 1 to 3, also varying k, until a large set of signatures S is obtained.

Step 5: Remove outliers from S.

Procedure 2: Generate set of popular signatures P using a clustering algorithm to cluster the signatures in S.

Step 1: Cluster the signatures into a small number of (e.g. 5-15) clusters.

Step 2: Return the centroids of each cluster as the set of popular signatures.

Step 3: Return the best popular signature set P.

Procedure 3: Generate spatial clustering for each dataset associating popular signatures with obtained clusters. To obtain spatial clusters which are annotated by popular signatures, apply CLEVER to each dataset using the following interestingness function $i(c)$:

$$i(c) = \begin{cases} 0 & d(sig(c), closest(sig(c), P)) > D \\ D - d(sig(c), closest(sig(c), P))^{\theta} & \text{otherwise} \end{cases} \tag{4.5}$$

Procedure 4: Generate the color map of each clustering; the concave hull of each cluster is colored based on its closest signature in $\{s1, \ldots, s7\}$; e.g. if $s1$ is the closest signature in $\{s1, \ldots, s7\}$ to cluster 1, cluster 1 is colored red, if $s2$ is the closest signature in $\{s1, \ldots, s7\}$ to cluster 2, cluster 2 is colored in orange.

4.3 Concave Hull Algorithm

In general, determining the scope of a spatial cluster is a challenging task. The goal is to create a spatial representation of a set of spatial objects in order to easily visualize it on the plane. One of easiest approaches is to compute the convex hull of the spatial objects in the cluster. However, the obtained convex hull polygon is usually not very tight and frequently enclosing empty spaces. In our proposed methodology, we used the concave hull algorithm for computing the scope of spatial clusters; we believe this approach is more effective than the convex hull algorithm, as it wraps a much tighter line around a set of spatial objects, resulting in less overlap with respect to the scope of neighboring clusters.

As we will introduced in the next chapter, the case study we conducted in this thesis adopts the concave hull functions in open-source Geographic Information Systems (GIS) platform - *OpenGIS*. The function provides the creation of concave polygon by the control of two main parameters — the compression parameter $0 \leq c \leq 1$ and boolean hole control value h. The higher value of c gives faster processing speed but looser defined concave hull shape as output. The hole control value determines if the output hull shape can have hole inside if possible. For the experiments in Chapter 5, we set parameters $c = 0.5$ and $h = ture$ for better performance and visualization.

Chapter 5

Case Study: Clustering Building Dataset

5.1 Identifying Uniform Regions in City

Figure 5.1 gives an example spatial clustering result in which buildings of different types (e.g., schools and industrial buildings) of a city are clustered.

The proposed methodology characterizes spatial clusters using their *scope* and *signature*. The scope of a spatial cluster captures the model of a cluster. In our approach, we use polygons as models for spatial clusters as depicted in Figure 5.1; that is, if a spatial object is inside the polygon which describes the scope of a spatial cluster, it belongs to that spatial cluster. Secondly, the proposed methodology uses signatures to annotate spatial clusters. Signatures summarize the distribution of the objects that belong to a cluster. As the clusters in the example contain buildings

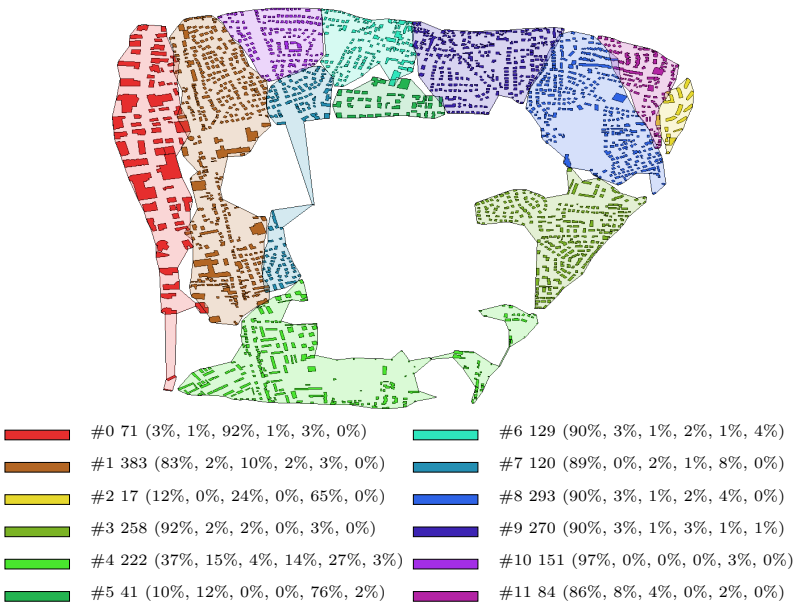


Figure 5.1: Example of a Spatial Clustering of Buildings Belonging to Different Building Types

belonging to different types, building type histograms are used as signatures to annotate spatial clusters. In our case study, there are six building types: single house, garages, industrial buildings, light buildings, collective buildings, and schools. For example, the leftmost cluster is identified as cluster 0 containing 71 buildings, and its building type signature is $(3\%, 1\%, 92\%, 1\%, 3\%, 0\%)$, indicating that 3% of the buildings in cluster 0 are single houses, 1% are garages, 92% are industrial buildings, 1% are light buildings, 3% are collective houses, and there are no schools in this spatial cluster. Moreover, the similarity between different building type signatures can be easily assessed: for example, we could use the Euclidian distance between the vectors associated with different building type signatures to assess the similarity of the contents of two spatial clusters.

Since understanding the evolution of cities is the key to intelligent urbanization, there is a growing need to develop urban planning and analysis tools to guide the orderly development of cities, as well as to enhance their smooth and beneficiary evolution. However, it is a big challenge for urban planners to come up with methodologies to analyze how cities are changing. Partitioning a city into uniform regions facilitates this task, as change can be analyzed based on higher level of granularity instead of on the raw data. A uniform region of a city will be called urban patch from now on. In this section, we present a set of experiments which use the methodology, which was introduced in Chapter 2, to extract urban patches from a building dataset. As buildings are represented as polygons, we use Hausdorff distance[29] to compute the distance between buildings in the experiments.

Table 5.1: Building Size Statistics

	Min	Max	Mean	Standard Deviation
2008	9.89	10384.34	256.56	511.18
5% Outliers Removal	49.77	1148.75	201.01	167.19

Table 5.2: Building Size Signature with Cluster Data for the First 6 Clusters from Clustering Result in Year 2008

Cluster ID	Mean	Standard Deviation	$\mu-1*\sigma$	$\mu - 0.25 * \sigma$	$\mu + 0.25 * \sigma$	$\mu+1*\sigma$	$\mu+1*\sigma$	No. of Buildings
0	-0.17	0	0%	11%	88%	1%	0%	122
1	0.15	0.35	0%	14%	48%	31%	8%	65
2	2.70	0.73	0%	0%	0%	19%	81%	16
3	0.44	0.46	0%	8%	37%	47%	7%	59
4	-0.21	0.11	0%	54%	42%	4%	0%	128
5	-0.21	0.02	0%	34%	65%	1%	0%	74

In this context, metrics for evaluation the homogeneity of a group buildings are

very important as they impact how a city is partitioned into urban patches characterized by signatures. In this section, we report the results of a series of experiments in which the CLEVER spatial clustering algorithm is used in conjunction with the three uniformity metrics, introduced in Chapter 2, to obtain interesting, uniform regions for the city of Strasbourg, France. In particular, we use a spatial building dataset of the city of Strasbourg, France, describing the buildings in a neighborhood of the city in 2008. In the frame of the GeOpenSim project, a temporal topographic database of the city of Strasbourg has been created[30].

The goals of this experimental evaluation are as follows:

1. Demonstrate how the proposed methodology works for a set examples.
2. Shed light on the computational challenges associated with finding uniform regions in spatial datasets.
3. Assess the benefits of the methodology for urban planners.
4. Assess what the spatial clustering algorithm CLEVER is capable to accomplish and what its limitations are-such an analysis is novel as the past two paper on CLEVER just introduced its pseudo code of CLEVER[22], and introduced parallel versions of CLEVER using OpenMP and Cuda[23].

5.2 Experiments Results

5.2.1 Building Size Distribution Experiment

Table 5.1 gives a statistical summary with respect to building sizes for the 2008 building dataset, as well as the statistics after removing 2.5% of the largest buildings

and 2.5% of the smallest buildings.

Table 5.2 summarizes a single building size distribution clustering result for the 2008 building dataset based on the signature definitions in Section 4.2. We only list the signatures of the first six clusters instead of 34 in total due to space limitation. These results were generated using CLEVER and low variance interestingness function for building size with $\theta = 2$ and $\beta = 1.001$. The first column reports the average building size using z-scores which were computed using $\mu = 256$ and $\sigma = 511$, based on the 2008 building size statistics. Columns 3-7 give the percentages of buildings in different bins of a 5-bin building size z-score histogram, based on μ and σ. The second column reports the cluster standard deviation in relationship to the dataset standard deviation. Entries above 0.5 indicate that the cluster's standard deviation is the same or larger than the dataset's standard deviation, whereas an entry of 0 indicates that the cluster's standard deviation is 1/10 or less of the dataset standard deviation. Each cluster represents an urban patch which ideally contains buildings with similar building sizes, as the employed interestingness function rewards low building size variance in a cluster.

In general, the data set is skewed with respect to building sizes as the minimum z-score of the average building sizes in Table 5.2 column 2 is just -0.21, whereas cluster 2 has a z-score of 2.7. Although cluster 2 is uniform in that it contains 16 very large buildings and no small buildings, its standard deviation is significantly above the average standard deviation in the dataset. Another complication is that in some parts of the city small buildings, particularly "small" garages, are collocated with large commercial and apartment buildings; clusters 1 and 3 in Table 5.2 represent

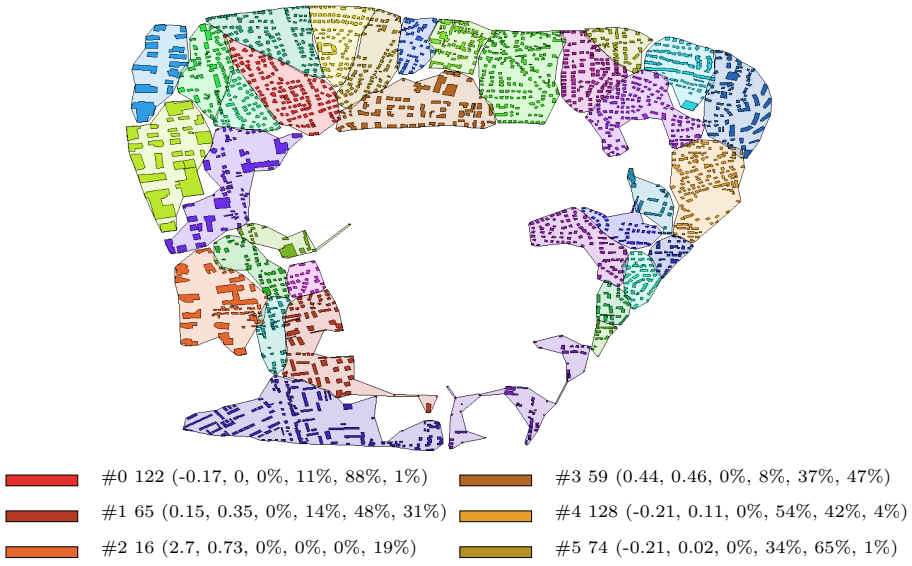


Figure 5.2: Visualization of Building Size Clusters of Year 2008 in Table 5.2.

such mixed building size clusters as their histogram contains building sizes belonging to four different bins. Finally, clusters 0, 4, and 5 in Table 5.2 are quite homogeneous with respect to their building size, exhibiting a very low standard deviation. In general, 30 of the obtained 34 clusters exhibit a building size standard deviation that is lower than the dataset standard deviation. Figure 5.2 visualizes the scope of the 34 building clusters of year 2008 with first 6 cluster signatures reported; it can be seen that our clustering algorithm successfully identifies the urban patches with similar building sizes. In the display, the first entry represents the cluster number, followed by the number of objects in the urban patch, followed by its building size distribution signature.

5.2.2 Building Type Purity Experiment

There are six different building types in the dataset: single house, garage, commercial building, light building, collective house, and school. In year 2008, 78% of the buildings were single houses; commercial buildings were 7%; collective houses were 8%; 4% of the buildings were garages, and 3% of the buildings were light building; finally, 1% of the buildings were schools. Building type signatures describe the characteristics of each urban patch which can help domain experts to better understand the composition of a city.

Figure 5.1 visualizes and lists the building type signatures of 12 clusters for the year 2008; they were generated by CLEVER using the purity interestingness function with $th = 0.5$, $\eta = 2$ and $\beta = 1.2$, as discussed in Chapter 4.2. Cluster 0 contains 92% commercial buildings; therefore, cluster 0 is labeled as a business urban patch. Cluster 10 is a residential area because 97% of the buildings in cluster 10 are single houses. There are 76% of collective houses in cluster 5, which indicates a living area with a lot of apartment complexes. Both garages and schools constitute very small percentages in the whole dataset, but garages and schools are more frequent in the collective housing areas in clusters 4 and 5. Surprisingly they are not present in cluster 2. Figure 5.1 verifies that our approach is able to identify contiguous urban patches dominated by buildings of a single type.

5.2.3 Popular Building-type Signature Experiment

Popular building-type signatures describe compositions of urban patches which frequently occur in different parts of a city. To obtain a set of popular signatures, we

first randomly created 1000 small spatial clusters and extracted their building-type signature. Next, we apply a distance-based outlier detection technique to remove 10% of the building type signatures as outliers — signatures were sorted by their 3-nearest neighbor distance to the other signatures in the set. Signatures with the largest 3-nearest neighbor distance were removed from the signature set. Next, we clustered the remaining signature set using K-means with different k values ranging between 6 and 10 several times, and identified the clustering with the lowest squared average distance of the objects in the dataset to the cluster centroid they belong to. Finally, we extracted the centroids from the best clustering as popular signatures. Table 5 lists nine popular building-type signatures that were obtained as the result of this process. The building type distributions of the entire dataset are also given for comparison. For example, signature $S4$ could be used to label a residence patch since 99% of the buildings are single houses, whereas signature $S7$ is a mixed housing signature, describing an area where buildings of many building-types are mixed together with a much higher density of garages and schools.

Table 5.3: Popular Building Type Signatures in 2008

Signature ID	Single House	Garage	Commercial Building	Light Building	Collective House	School
$S1$	77%	3%	2%	2%	17%	0%
$S2$	87%	4%	1%	3%	4%	1%
$S3$	2%	6%	0%	0%	92%	0%
$S4$	99%	0%	0%	0%	0%	0%
$S5$	48%	1%	46%	3%	2%	0%
$S6$	4%	0%	96%	0%	0%	0%
$S7$	37%	22%	4%	1%	32%	4%
$S8$	62%	6%	13%	12%	4%	1%
$S9$	85%	1%	14%	0%	0%	0%
Dataset	78%	4%	7%	3%	8%	1%

Table 5.3 summarizes a popular signature clustering result which was created using CLEVER and the popular signature interestingness function with parameters $D = 0.13$, $\theta = 2$, and $\beta = 1.005$. We use 0.1 as the threshold for the Euclidian distance of the cluster signature to its closest popular signature to indicate a good match. 14 out of the 16 urban patches shown in Table 5.4 have good matches with their popular signatures. Cluster 3 is quite unusual as it is dominated by light buildings and is not close to any popular signature in Table 5.3 at all, which is indicated by its very high Euclidian distance of 0.49 to its closest popular signature. As it turns out there is a single, very small region in the city with a high density of light buildings. As this signature occurs only in a single small area, it does not belong to the popular signature set. This observation will be confirmed later in the last experiment, discussed in Section 5.2.4.

Table 5.4: Popular Building Type Clustering Results for 2008

Cluster ID	Single House	Garage	Commercial Building	Light Building	Collective House	School	No. of Buildings	Closest Signature	Distance
0	89%	4%	2%	0%	5%	0%	56	$S2$	0.04
1	75%	7%	4%	0%	13%	0%	69	$S1$	0.07
2	73%	8%	6%	2%	12%	0%	52	$S1$	0.09
3	29%	2%	9%	45%	15%	0%	55	$S8$	0.49
4	72%	6%	11%	1%	10%	0%	157	$S1$	0.13
5	88%	4%	2%	3%	5%	0%	199	$S2$	0.02
6	100%	0%	0%	0%	0%	0%	112	$S4$	0.01
7	44%	1%	46%	5%	4%	0%	100	$S5$	0.05
8	87%	4%	1%	3%	3%	1%	335	$S2$	0.01
9	85%	1%	13%	1%	1%	0%	320	$S9$	0.01
10	77%	5%	8%	0%	10%	0%	39	$S1$	0.09
11	77%	3%	1%	1%	17%	2%	198	$S1$	0.03
12	36%	20%	3%	4%	34%	4%	142	$S7$	0.05
13	99%	1%	0%	0%	0%	0%	121	$S4$	0.01
14	98%	2%	0%	0%	0%	0%	57	$S4$	0.02
15	89%	0%	0%	0%	11%	0%	27	$S2$	0.09

Figure 5.3 visualizes the spatial clusters summarized in Table 6 with their associated popular signatures added. Different colors in the display are used to indicate

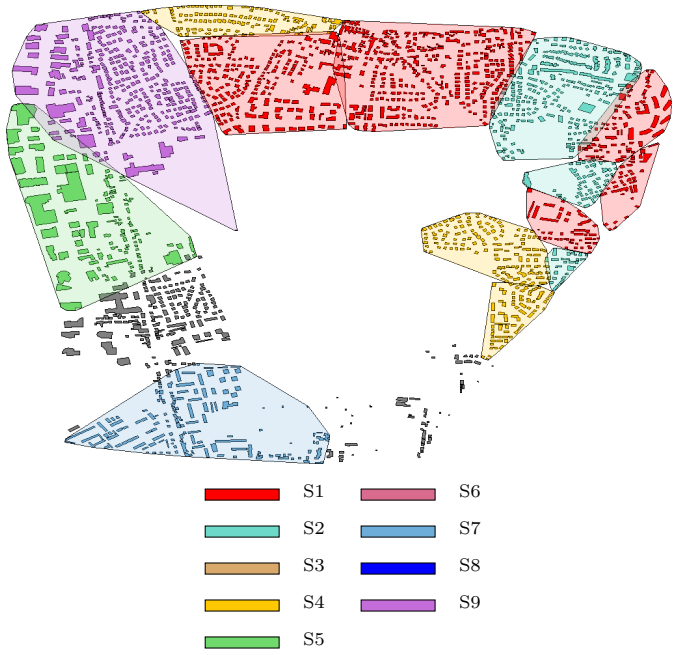


Figure 5.3: Visualization of 14 Clusters Summarized in Table 5.4 Annotated with Their Popular Signature.

different signatures. Moreover, clusters whose signatures are not close to any popular signature, namely clusters 3 and 4 in Table 6, are not labeled with popular signatures. Therefore, the display contains buildings shown in grey color which do not belong to any cluster. We claim that such displays are very helpful for domain experts as they facilitate identifying urban patches in different parts of city which exhibit the same popular signature. Our approach uses a spatial clustering algorithm — and not predetermined regions — to identify the scope of a popular signature and annotating only those regions which match a popular signature well. We claim that the urban patches identified by our approach, exhibit a much better match with the popular signature set.

Table 5.5: Query Signatures Used in the Experiment

Signature ID	Color	Single House	Garage	Commercial Building	Light Building	Collective House	School
$Q1$	Red	29%	2%	9%	45%	15%	0%
$Q2$	Green	70%	0%	0%	0%	0%	30%
$Q3$	Blue	2%	6%	0%	0%	92%	0%

5.2.4 Querying Spatial Dataset with Signatures

Although the presented popular signature mining algorithm was originally developed to determine the scope of a set of popular signatures, it can be used with any signature set P. This enables us to use the same algorithm for querying spatial datasets for the presence of particular "query signatures". For example, in the experiment summarized in Table 6 we came across cluster 3, which was dominated by light buildings, and we might be interesting to see if its signature (named $Q1$ in Table 5.5) occurs in other areas of the city. Along the same line we might want to see, if there are regions with a high density of schools in a residential area (e.g. we could us $Q2$ in Table 5.5). Finally, if the popular signature $S3$ occurs anywhere in the dataset ($Q3 = (2\%, 6\%, 0\%, 0\%, 92\%, 0\%)$), as it did not match any cluster signature, as reported in Table 5.3.

Table 5.6: Clusters Matching Query Signatures

Cluster ID	Matched Signature	Single House	Garage	Commercial Building	Light Building	Collective House	School	Distance
5	$Q1$	29.63%	1.85%	9.26%	44.44%	14.81%	0%	0.009
11	$Q3$	2.78%	5.56%	0%	0%	91.67%	0%	0.010
13	$Q2$	66.67%	0%	0%	0%	0%	33.33%	0.047

Figure 5.4 and Table 5.6 gives the result running CLEVER with the popular signature interestingness function for signature set $P = \{Q1, Q2, Q3\}$ with parameters $D = 0.2$, $\eta = 3$, and $\beta = 1.2$. The spatial clusters in Figure 5.4 are annotated with

corresponding signature colors if the distance of the cluster signature to its closest query signature in P is 0.2 or less; otherwise they are marked as white. Table 5.6 lists the signatures for three clusters that are close to query signatures as well as the number of objects, closest query signature and the distance to its closest query signature. The experiment took 28 seconds wall clock time and the algorithm needed 44 iterations evaluating 2670 clusterings. As can be seen, the algorithm rediscovered the same region with a majority of light buildings identified by the popular signature clustering algorithm but no other regions which express this signature. Moreover, a single region which almost perfectly matches the popular signature $S3$ was found. Finally, we were able to find a single region with a mixture of schools and single houses, but the match of the regions' signature with $Q2$ is of medium quality, as the Euclidian distance between the two signatures is about 0.04.

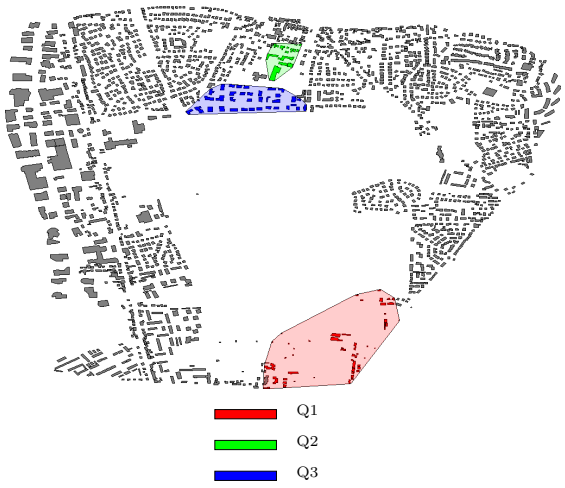


Figure 5.4: Visualization of Clusters Matching Query Signatures

5.2.5 Validating and Sensitivity Analysis Based on Popular Signature Clustering Experiment

CLEVER has been designed to find a "good" solution for what is, in general, an NP-hard problem relying on randomized hill climbing. As all optimization procedure which start with randomly created initial solutions, CLEVER — as K-means — is sensitive to initialization, as different initializations may lead to different, alternative solutions. In this section, we discuss the results of two experiments which analyze CLEVER's sensitivity to initialization, and how close CLEVER gets to the "optimal" solution.

To analyze CLEVER's sensitivity to initialization, we ran the building type purity clustering procedure 20 times with parameters $k' = 20, \beta = 1.05, \eta = 3$, and $th = 0.5$, and collected the following run characteristics: $q(X)$, number of the clusters in the final clustering, number of iterations, and the number of clusterings generated during the run. The sampling procedure used in this (and the next) experiment is as follows: first sample 15 clusterings in the neighborhood of the current clustering, then — if there is no improvement — 30 solutions, and finally 180 solutions; if none of the 225 sampled clusterings improves the current clustering, the search ends. According to the results reported in Table 5.7, CLEVER terminated after at an average 32 iterations and searched at an average 1400 clusterings. Although CLEVER starts from different initial clusterings, the quality of the clustering results are relatively stable around 729 with a standard deviation of 24. However, the number of final clusters obtained differs quite significantly between the twenty runs, ranging between 3 and 23. This fact indicates that the obtained 20 final clusterings — although having

Table 5.7: Building Type Purity Sensitivity Results

Run ID	$q(X)$	**No. of Clusters**	**No. of Iterations**	**Generated Clusterings**
1	776.81	7	38	1635
2	764.68	8	43	1920
3	756.20	10	25	645
4	747.56	11	39	1830
5	746.39	12	29	1245
6	744.51	9	30	1470
7	741.23	11	24	1170
8	738.21	3	31	1470
9	737.03	13	29	1245
10	736.27	16	45	1950
11	727.90	11	39	2010
12	726.31	8	48	2175
13	719.12	10	23	960
14	716.62	23	36	1395
15	715.18	14	20	525
16	710.86	16	26	1380
17	707.44	9	31	1140
18	693.47	18	37	1605
19	688.78	16	31	1665
20	685.85	16	24	1005
Mean	729.02	12.05	32.40	1422
SD	24.63	4.55	7.88	444.40
Max	776.81	23.00	48.00	2175.00
Min	685.85	3.00	20.00	525.00

a similar quality with respect to $q(X)$ — differ from each other.

Table 5.8: Building Type Signature Mining Sensitivity Results.

Run ID	$q(X)$	**No. of Clusters**	**No. of Iterations**	**Generated Clusterings**
1	1.45	14	24	1365
2	1.44	19	20	1110
3	1.43	25	35	2115
4	1.41	20	27	1575
5	1.39	14	25	1185
6	1.39	25	19	915
7	1.37	19	13	420
8	1.35	25	31	1680
9	1.35	19	13	405
10	1.34	25	19	885
11	1.33	25	31	1665
12	1.32	19	25	825
13	1.31	26	26	1050
14	1.31	17	19	915
15	1.31	19	21	1290
16	1.30	24	30	1515
17	1.30	17	16	840
18	1.24	23	25	855
19	1.23	22	11	390
20	1.21	21	19	720
Mean	1.34	20.90	22.45	1086.00
SD	0.07	3.77	6.58	464.97
Max	1.45	26.00	35.00	2115.00
Min	1.21	14.00	11.00	390.00

For the second experiment, we first partitioned the spatial dataset into 20 contiguous regions. Next, we computed the building type distribution for each of those regions, and used the obtained 20 signatures as "popular" signatures. Finally, we used the popular signature clustering procedure in conjunction with this signature set to see how close our approach can get to the "optimal" solution which consists of the 20 regions whose popular signature perfectly matches the region's signature.

Table 5.9: Ground Truth ($q(X) = 2.58$).

Cluster ID	No. of Objects	Single House	Garage	Commercial Building	Light Building	Collective House	School
$S0$	48	95.83%	0%	0%	0%	4.17%	0%
$S1$	100	88%	2.67%	4%	0%	5.33%	0%
$S2$	75	88%	2.67%	4%	0%	5.33%	0%
$S3$	56	62.50%	1.79%	21.43%	7.14%	7.14%	0%
$S4$	58	67.24%	10.34%	17.24%	1.72%	3.45%	0%
$S0$	48	95.83%	0%	0%	0%	4.17%	0%
$S5$	126	92.86%	5.56%	0%	0.79%	0.79%	0%
$S6$	146	91.78%	0%	0%	0%	8.22%	0%
$S7$	101	71.29%	3.96%	10.89%	1.98%	11.88%	0%
$S8$	147	36.73%	18.37%	2.72%	4.08%	34.01%	4.08%
$S9$	171	78.95%	4.68%	4.68%	1.75%	9.94%	0%
$S10$	114	63.16%	6.14%	0.88%	1.75%	22.81%	5.26%
$S11$	50	92%	0%	4%	2%	2%	0%
$S12$	130	97.69%	0%	0.77%	0%	1.54%	0%
$S13$	91	58.24%	2.20%	5.49%	26.37%	7.69%	0%
$S14$	77	40.26%	1.30%	57.14%	1.30%	0%	0%
$S15$	56	100%	0%	0%	0%	0%	0%
$S16$	139	93.52%	2.16%	0%	0%	4.32%	0%
$S17$	75	85.33%	6.67%	2.67%	1.33%	4%	0%
$S18$	137	78.10%	0.73%	20.44%	0.73%	0%	0%
$S19$	142	88.73%	2.11%	1.41%	5.63%	0.70%	1.41%

In this experiment, we ran CLEVER 20 times with parameter settings $k' = 20, \beta = 1.05, D = 0.1, \eta = 3$. The results of this experiment are reported in Table 5.8 and the ground truth Table 5.9. The best two clustering results are visualized in Figure 5.6 and Figure 5.7, using a different color for each popular signature; clusters which do not match any popular signature are visualized in white.

In general, we draw the following conclusions from the second experiment:

1. The optimal solution which perfectly approximates the 20 signatures has a $q(X)$ value of 2.58 — and based of the popular signature clustering interestingness function introduced earlier — each of the 20 clusters has the maximum interestingness of $0.1^3 = 0.001$. For example, a cluster whose signature has a "small" distance of 0.03 to the closest popular signature, has a significantly lower interestingness of

Table 5.10: Best Clustering Result ($q(X) = 1.45$).

Cluster ID	**No. of Objects**	**Closest Signature**	**Distance**
0	68	$S1$	0.013
1	129	$S15$	0.011
2	229	$S0$	0.003
3	206	$S3$	0.06
4	302	$S18$	0.012
5	176	$S8$	0.021
6	176	$S10$	0.074
7	207	$S9$	0.006
8	55	$S5$	0.020
9	187	$S5$	0.006
10	61	$S9$	0.034
11	162	$S19$	0.018
12	56	$S13$	0.254
13	25	$S5$	0.037

Table 5.11: Second Best Clustering Result ($q(X) = 1.44$).

Cluster ID	**No. of Objects**	**Closest Signature**	**Distance**
0	300	$S18$	0.006
1	70	$S15$	0
2	45	$S5$	0.026
3	175	$S9$	0.005
4	84	$S0$	0.008
5	128	$S6$	0.005
6	59	$S11$	0.028
7	82	$S13$	0.050
8	30	$S10$	0.197
9	172	$S8$	0.017
10	78	$S5$	0.027
11	114	$S16$	0.028
12	92	$S10$	0.020
13	140	$S1$	0.026
14	10	$S11$	0.069
15	61	$S5$	0.024
16	212	$S3$	0.049
17	106	$S19$	0.034
18	81	$S0$	0.007

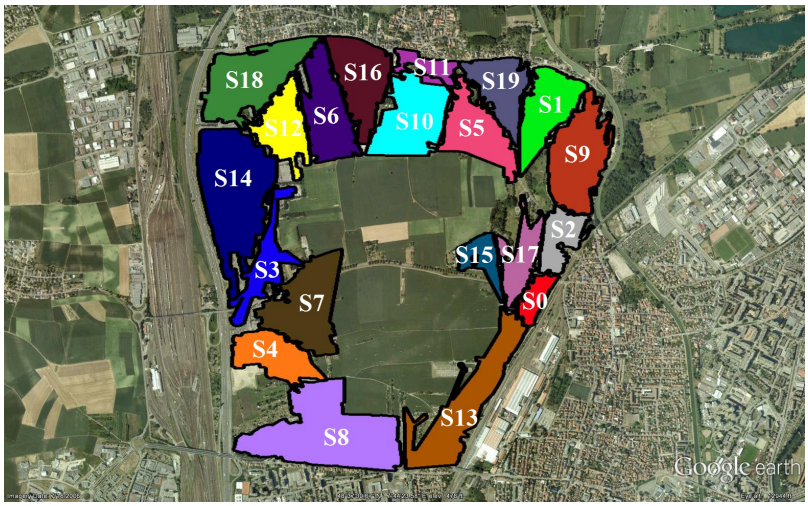


Figure 5.5: Ground Truth ($q(X) = 2.58$).

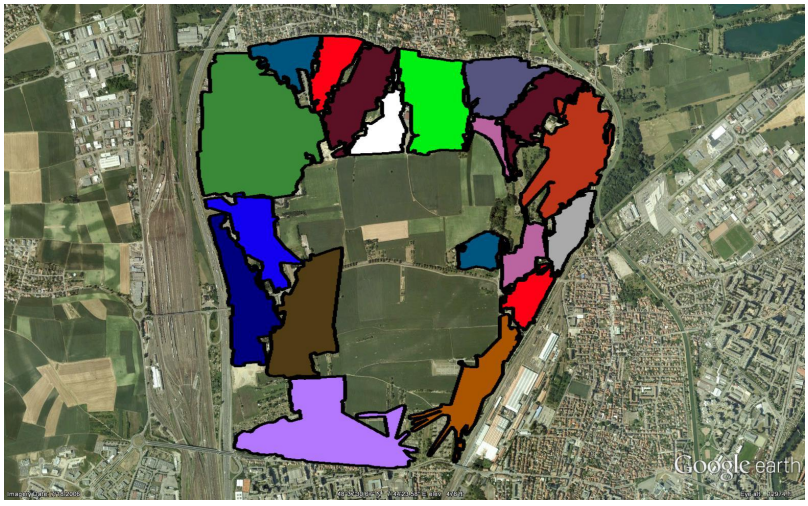


Figure 5.6: Best Result ($q(X) = 1.45$).

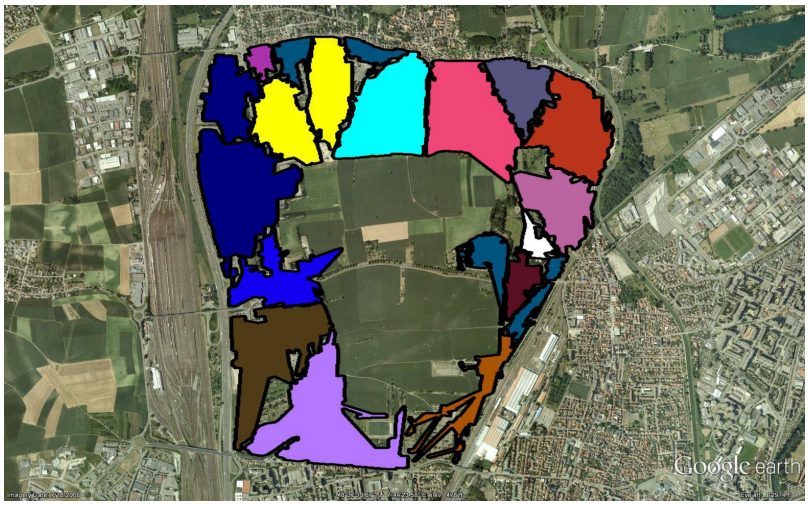


Figure 5.7: Second Best Result ($q(X) = 1.44$).

$0.07^3 = 0.000343$, due to the fact that we use a cubic interestingness function.

2. The clustering quality for the 20 clusterings is quite stable, ranging from 1.21 to 1.45, but they are not very close to 2.58, the quality of the optimal solution.

3. There is not a lot of agreement between the clustering results, neither between the optimal result and the other results nor between the two clustering results.

4. There seem to be many alternative ways to partition the datasets based on popular signatures; that is, many alternative partitioning exist which receive the same or similar rewards. For example, often one popular signature can be expressed as a linear combination of other popular signatures; consequently, we could have a cluster which perfectly matches a popular signature and its partitioning into two sub-clusters perfectly matches two other popular signatures.

5. We observed that there is much more agreement between final clusters in the first experiment that used the purity interestingness function than in the second experiment.

6. The result suggest that it might be beneficiary to extract "good" clusters from multiple runs and combine them into a final clustering, instead of trying to find all clusters in a single run.

7. In the experiment we used a low maximum sampling rate of 225; it might be interesting to see if parallel versions of CLEVER which allow for the use of much higher sampling rates get closer to the optimal solution.

Table 5.12: Performance Characteristics of the Reported Clustering Results

	No. of Iterations	No. of Clusterings Generated	Time Elapsed
Section 5.2.1	25	1215	32.36s
Section 5.2.2	30	1485	32.92s
Section 5.2.3	35	1590	33.65s
Section 5.2.4	44	2670	28.26s
Section 5.2.5	34	1950	36.15s

5.2.6 Performance Analysis for CLEVER

Table 5.12 gives some performance characteristics for the clustering results that were reported in Sections 5.2.1 to 5.2.5 in terms of iterations needed, number of clusterings generated, and wall clock time. CLEVER was run on a dataset containing 2039 objects on a computer with the processor running at 3 GHz and 8 GB main memory.

Chapter 6

Conclusion

This thesis introduces a spatial clustering methodology which identifies contiguous regions in the space of the spatial attributes which are uniform with respect to their signatures, and which represent statistical summaries for the objects belonging to a particular cluster. The second idea advocated in the thesis is to mine spatial data for the presence of particular signatures. These two types of signature-based spatial clustering have broad applications in urban computing, environmental sciences, ecology, and geo-targeting.

The proposed methodology defines the task of finding uniform regions formally as a maximization problem. Various objective functions and corresponding algorithms are introduced. In particular, we introduce a prototype-based clustering algorithm named CLEVER, which identifies uniform regions in a spatial dataset by maximizing a plug-in measure of uniformity, relying on a randomized hill climbing approach. A variant of CLEVER – DCLEVER – is also proposed to avoid computing distance on the fly, and outperforms CLEVER in small datasets with large amount of iterations.

Moreover, polygon models which capture the scope of a spatial cluster and histogram-style distribution signatures are used to annotate the content of a spatial cluster; both play a key role in summarizing the composition of a spatial dataset. We claim that the presented approach is novel and unique as existing clustering algorithms are not suitable for this task, because they minimize distance-based objective functions, whereas assessing uniformity relies on non-distance-based uniformity measures.

The proposed methodology is evaluated by a challenging real-world case study centering on analyzing the composition of the city of Strasbourg in France based on building characteristics. First, we identify uniform regions using two different interestingness functions based on the variance of building sizes and based on the purity of building types in a cluster. Second, an approach is presented which determines popular distribution signatures, and then uses a spatial clustering approach to identify urban patches with a good match with particular popular signatures. The efficacy of the two approaches is demonstrated through the experimental evaluation.

Applying the methodology presented in this thesis faces several challenges, such as sensitivity to initialization, finding more suitable algorithms to compute scope of a set of spatial clusters, providing a better theoretical foundation for signature mining, the capability to identify spatial clusters of arbitrary shape, and the need to run spatial clustering algorithms multiple times. Finally, as the computational complexity of signature mining is usually very high, there is a need for parallel signature mining algorithms. Our current and future work centers on dealing with these challenges.

In addition to the methodology proposed by this thesis and its main focus, it's not

trivial to mention some of other works that have been implemented to support this thesis. A KML file parsing program with database connection has been developed to import the original data into a geo-supported database. Moreover, the clustering results generated by DCLEVER are visualized by a self-developed software that produces the KML file.

Bibliography

[1] Wikipedia, "Urbanization — wikipedia, the free encyclopedia," 2012, [Online; accessed 4-April-2012]. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Urbanization&oldid=484629309

[2] J. Yuan, Y. Zheng, and X. Xie, "Discovering regions of different functions in a city using human mobility and pois," in *Proceedings of KDD Conference*, Beijing, China, 2012.

[3] Wikipedia, "Urban area – wikipedia, the free encyclopedia," 2012, [Online; accessed 18-March-2012]. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Urban_area&oldid=477723785

[4] M. Grigorovschi, "City planning evolution – urban development directions in the transition period," *Institutul Politehnic din Iasi. Buletinul. Sectia Constructii. Arhitectura*, vol. 56, no. 1, p. 63, 2010. [Online]. Available: http://www.ce.tuiasi.ro/~bipcons/Archive/174.pdf

[5] B. Das, "Urban planning in india," *Social Scientist*, vol. 9, no. 12, pp. 53–67, Dec. 1981. [Online]. Available: http://www.jstor.org/stable/10.2307/3517133

[6] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[7] S. Shekhar, P. Zhang, and Y. Huang, "Trends in spatial data mining," *Science*, vol. 7, no. 20, pp. 357–379, 2003.

[8] W. Tobler, *Cellular Geography*. Dordrecht: Reidel, 1979.

[9] Wikipedia, "Polygon — wikipedia, the free encyclopedia," 2012, [Online; accessed 7-April-2012]. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Polygon&oldid=484521966

[10] S. Wang, C.-S. Chen, V. Rinsurongkawong, F. Akdag, and C. F. Eick, "A polygon-based methodology for mining related spatial datasets," in *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Data Mining for Geoinformatics*, ser. DMG '10. New York, NY, USA: ACM, 2010, pp. 1–8. [Online]. Available: http://doi.acm.org/10.1145/1869890.1869891

[11] Wikipedia, "Earthquake — wikipedia, the free encyclopedia," 2013, [Online; accessed 14-July-2013]. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Earthquake&oldid=560813801

[12] F. Akdag, "Algorithms for creating polygon models for spatial clusters," Master's thesis, University of Houston, 2010.

[13] M. Duckham, L. Kulik, M. Worboys, and A. Galton, "Efficient generation of simple polygons for characterizing the shape of a set of points in the plane," *Pattern Recognition*, vol. 41, no. 10, pp. 3224–3236, 2008.

[14] D. Joshi, "Polygonal spatial clustering," Ph.D. dissertation, University of Nebraska, 2011.

[15] Wikipedia, "Frchet distance — wikipedia, the free encyclopedia," 2012, [Online; accessed 13-May-2012]. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Fr%C3%A9chet_distance&oldid=482704260

[16] W. Ding, R. Jiamthapthaksin, R. Parmar, D. Jiang, T. Stepinski, and C. F. Eick, "Towards region discovery in spatial datasets," in *Proc. Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD)*, September 2008, pp. 88–99.

[17] C. F. Eick, B. Vaezian, D. Jiang, and J. Wang, "Discovery of interesting regions in spatial datasets using supervised clustering," in *Proc. 10th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, September 2006.

[18] O. U. Celepcikay and C. F. Eick, "A regional regression framework for georeferenced datasets," in *Proc. 17th ACM SIGSPATIAL International Conference on Advances in GIS (GIS)*, November 2009, pp. 326–335.

[19] S. Vucetic and Z. Obradovic, "Discovering homogeneous regions in spatial data through competition," in *Proc. ICML Conference*, 2000, pp. 1095–1102.

[20] D. Joshi, A. Samal, and L. Soh, "A dissimilarity function for clustering geospatial polygons," in *Proceedings of the 17th ACM SIGSPATIAL International*

Conference on Advances in Geographic Information Systems (GIS), 2009, pp. 384–387.

[21] Z. Yin, L. Cao, J. Han, C. Zhai, and T. Huang, "Geographical topic discovery and comparison," in *Proceedings of WWW Conference*, Hyderabad, India, 2011.

[22] C. F. Eick, R. Parmar, W. Ding, T. Stepinki, and J.-P. Nicot, "Finding regional co-location patterns for sets of continuous variables in spatial datasets," in *Proc. 16th ACM SIGSPATIAL International Conference on Advances in GIS (GIS)*, 2008.

[23] C.-S. Chen, N. Shaikh, P. Charoenrattanaruk, C. F. Eick, N. Rizk, and E. Gabriel, "Design and evaluation of a parallel execution framework for the clever clustering algorithm," in *Proc. ParCo Conference*, 2011.

[24] L. Kaufman and P. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, 1990.

[25] J. Lesbegueries, N. Lachiche, A. Braud, G. Skupinski, A. Puissant, and J. Perret, "A platform for spatial data labeling in an urban context," in *International Opensource Geospatial Research Symposium (OGRS)*, Nantes, France, 2009.

[26] H. Edelsbrunner, D. Kirkpartick, and R. Seidel, "On the shape of points in the plan," in *IEEE Transactions on Information Theory*, 1983, pp. 551–559.

[27] A. Moreira and M. Santos, "Concave hull: a k-nearest neighbors approach for the computation of the region occupied by a set of points," in *Proc. International Conference on Computer Graphics Theory and Applications (GRAPP)*, 2007, pp. 61–68.

[28] B. Presles, J. Debayle, Y. Mailot, and P. J.-C., "Automatic recognition of 2d shapes from set of points," in *Proc. ICAR Conference*, 2011, pp. 183–192.

[29] N. Cressie, *Statistics for Spatial Data*. Wiley, 1993.

[30] A. Ruas, J. Perret, F. Curie, A. Mas, A. Puissant, G. Skupinski, D. Badariotti, C. Weber, P. Gancarski, N. Lachiche, A. Braud, and J. Lesbegueries, "Conception of a gis platform to study and simulate urban densification based on the analysis of topographic data," *Cartography and GIScience*, vol. 1, pp. 413–430, 2011.