2012-06-20

A traffic accident risk mapping framework

WANG, JING

UNIVERSITY OF CALGARY

A TRAFFIC ACCIDENT RISK MAPPING FRAMEWORK

by

JING WANG

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF GEOMATICS ENGINEERING

CALGARY, ALBERTA

JUNE, 2012

Abstract

Identifying traffic accident concentration area is important for road safety improvements. Previous spatial concentration detection methods did not consider the severity levels of accidents, and the final traffic accident risk map for the whole study area ignores the different users’ requirements.

This thesis proposes an ontology-based traffic accident risk mapping framework. In the framework, the ontology represents the domain knowledge related to the traffic accidents and supports the data retrieval based on users' requirements. A new spatial clustering method, called DBCTAR (Density-based Clustering for Traffic Accident Risk), takes into account the numbers and severity levels of accidents is proposed for risk mapping. To demonstrate the framework and the new algorithm, the Ontology-based Traffic Accident Risk Mapping (ONTO_TARM) system and a web-based clustering service GeoClustering have been developed. Four case studies in the city of Calgary with final risk maps are presented and discussed.

Preface

The outcome of this research have been published and/or presented in a regular fashion, and listed as follows:

Book Chapter:

Wang, J., Wang, X., Liang, S.H.L.: GeoClustering: A Web Service for Geospatial Clustering. In: Li, S., Dragicevic, S., Veenendaal, B. (eds), Advances in Web-based GIS, Mapping Services and Applications. pp.37-54 (2011)

Journal Papers:

Wang, X., Wang, J.: Using Clustering Methods in Geospatial Information Systems. Geomatica. 64(3), pp.347-361 (2010)

Conference Proceedings:

Wang, J., Wang, X.: An Ontology-based Traffic Accident Risk Mapping Framework. 12th International Symposium on Spatial and Temporal Databases (SSTD 2011), Minneapolis, MN, USA, pp.21-38 (2011)

Map & Poster:

Wang, J., Wang, X.: A Traffic Accident Risk Mapping System, GeoAlberta Conference 2011 Map Gallery, Edmonton, Alberta, Canada (2011)

Acknowledgements

I would like to take this opportunity to thank all the people who made this work possible. My deepest gratitude and appreciation goes to my supervisor, Dr. Xin Wang, for her remarkable guidance. My sincere gratitude goes to Dr. Steve H.L. Liang, Dr. Danielle Marceau, Dr. Darren Bender and Dr. Caterina Valeo for teaching me and inspiring me over the past years.

I would also like to thank Dr. Quazi K. Hassan and Dr. Lina Kattan, for their time and efforts to read my thesis and provide valuable comments and suggestions.

I am most grateful to Mr. Gary Zhang, who offered me the internship at MRF Geosystems Corporation and contributed the NSERC/MITACS Industrial Postgraduate Scholarship. I would also like to acknowledge other financial support from Natural Sciences and Engineering Research Council of Canada, MITACS, Alberta Scholarship Programs, Faculty of Graduate Study, and the Department of Geomatics Engineering.

Particularly, I wish to express my deep appreciation to Dr. Richard Tay, for helping me get the real dataset.

Last but not least, I am indebted to my fellow students and friends Baijie Wang, Wei Gu, Lani Roux for their valuable comments and suggestions.

Dedication

I would like to dedicate this thesis to my parents Zigui Wei and Yunyi Wang, for their continuous love, trust, and support.

Table of Contents

List of Tables

List of Figures and Illustrations

List of Abbreviations

Symbol	**Definition**
API	Application Programming Interface
CSV	Comma-Separated Values
DBSCAN	Density-based Spatial Clustering of Applications with Noise
DBCTAR	Density-based Clustering for Traffic Accident Risk
GKD	Geographic Knowledge Discovery
GeoRSS	Geographically encoded objects for RSS
ISO	International Organization for Standardization
KDE	Kernel Density Estimation
KML	Keyhole Markup Language
OGC	Open Geospatial Consortium
ONTO_TARM	Ontology-based Traffic Accident Risk Mapping
OWL	Web Ontology Language
PIARC	Permanent International Association of Road Congresses
PDO	Property Damage Only
RDF	Resource Description Framework
RDFS	Resource Description Framework Schemas
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
XML	Extensible Markup Language

Chapter One: Introduction

1.1 Background Information

Road traffic accidents are a social and public health challenge, as they almost always result in injuries and/or fatalities (Anderson 2009). The World Health Organization estimates over 1 million people are killed each year in road collisions. This is equal to 2.1% of the annual global mortality and an estimated social cost of $518 billion (Peden et al. 2004). In Canada, about 3,000 people are killed every year on the roads (RememberRoadCrashVictims.ca 2009). In Alberta, the average time between collisions is 5 minutes (Tay 2006). Within only the City of Calgary, the total reportable collisions number was 39,542 in 2008 (Calgary Police Service 2009).

To significantly reduce traffic fatalities and serious injuries on public roads, we need to review the characteristics of traffic accidents and identify the hidden patterns behind the accidents' records, referring mainly to the actual knowledge contained in the collision data rather than the raw data records themselves. For example, road safety managers or residents may be interested in the accident patterns near their communities and not the data records.

Previous traffic safety studies show that, in most cases, the occurrences of traffic accidents are seldom random in space and time, but form clusters that indicate accident concentration areas in geographic space (Anderson 2009). A concentration area is defined as an area or location where there is a higher likelihood for an accident to occur based on historical data and spatial dependency. Thus, if we can identify the locations with the high risk on the roads, road safety managers can analyze the reasons behind the fact; and, the public can be aware of the danger, so that they can drive more carefully on the dangerous road or avoid it altogether.

The "risk" could be the result of number of accidents, severity level, and the measure of exposure. However, the measure of exposure (usually measured by exposure population or traffic volume) data is hard to get. So, in this thesis, the measure of exposure is considered as a constant in the same research area. The "risk area" in this thesis is more similar as "hotspot" or "black spot", which represents a road segment where road traffic accidents have historically been concentrated and accident severity level is also high.

This kind of strong need for actual knowledge - the risk area on the road, rather than for just the raw accident data from which the knowledge was derived, belongs to the emerging and developing research area called geographic knowledge discovery (GKD). GKD is the process of extracting information and knowledge from massive amounts of geo-referenced data. Among major GKD techniques, geospatial cluster analysis is an important and very useful method. Geospatial cluster analysis, also named geospatial clustering, is an approach to applying spatial clustering techniques on geographically based data. Spatial clustering is the process of grouping similar objects based on their distance, connectivity or relative density in space (Han et al. 2001). Clusters create an abstract representation of the original data, where similar points within the same groups are merged, based on location as well as on other non-spatial attributes. In addition, clustering methods can discern interesting spatial patterns and features, capture intrinsic relationships between spatial and non-spatial data, and present data regularity concisely and at higher conceptual levels (Ng & Han 2002).

Accident data can be represented as geospatial points. A cluster of accidents indicates patterns with high density accidents in that area. Therefore, geospatial clustering method may be able to determine the concentrated areas of accidents.

1.2 Problem Statement

Over the years, various spatial concentration detection methods and tools have been proposed and applied to discover traffic accident concentration patterns. However, previous traffic accidents concentration detection researches have several limitations. First, these methods do not consider the discrepancy of conditions and only generate one single map as a result. Traffic analysts and the general public are actually interested in accident concentration areas in terms of specific conditions, such as different time intervals, and weather or road surface conditions. These different conditions reflect different requirements from users. Therefore, risk maps that meet users' manifold requirements are necessary.

Nevertheless, integration of users' requirements into generating different concentration maps is not an easy task. The first subtask is the selection of the proper datasets based on different users' requirements. Selection at the data level necessitates good experience or understanding of the datasets in order to define the queries, which can sometimes be quite challenging for users who have no or only limited knowledge of the study area and dataset.

Second, existing concentration detection methods are limited by their lack of consideration of the attributes of the accidents, treating each accident as a point and applying traditional point pattern analysis methods to the extracted points. When defining the concentration area, they consider only the number of accidents, ignoring the severity levels associated with the accidents. However, accidents have different severity levels, including fatality, injury, and property damage only (PDO) (Doherty et al. 1998); and, each level should be treated differently.

Third, although various spatial clustering tools and applications have been proposed and developed over the years, to the best of my knowledge, most related clustering tools are not designed for geospatial clustering tasks. Most tools or applications involved in previous researches only utilize spatial analysis methods to identify traffic accident concentration areas. In addition, these tools are desktop-based sealed packages or closed systems; their openness and interoperability could be improved. Some companies do provide client libraries of APIs (application programming interfaces) for online geospatial clustering functions; however, most of these applications are used to simplify the map instead of for discovery of underlying cluster patterns. Moreover, users need to do extra programming to be able to use them.

1.3 Objective

The objective of this thesis is to design and implement a framework which can help users generate different concentration maps based on their requirements in terms of specific conditions, such as different time intervals, weather or road surface conditions, showing the “risk areas” on a road network based on the historical accident dataset.

The sub objectives can be summarized as follows:

- Proposes an Ontology-based Traffic Accident Risk Mapping (ONTO_TARM) Framework, so as to improve the limitations existing in previous research investigations. This framework generalizes the formalized methods and logical operations. It also supports the development work.
- Builds traffic accident domain ontology to organize data. Ontology is the explicit specification of a conceptualization (Gruber 1993). It provides domain knowledge relevant to the conceptualization and axioms for reasoning with it. For the accident domain ontology, it provides the knowledge pool for the reasoning, so that the framework can handle users' requirements at the knowledge level.
- Propose a density-based clustering algorithm for traffic accident risk (DBCTAR) to find accident concentrations with severity levels in geographical space. This method can separate objects into groups (called clusters) based on both spatial and non-spatial attributes. Clusters are regarded as network regions where the accidents are dense enough and above a certain severity level. These regions may have arbitrary shapes, and the accidents inside a region may be arbitrarily distributed.
- Builds a clustering service called GeoClustering to perform the geospatial clustering tasks. This web-based clustering service is aimed at the discovery of interesting patterns in geographical space. This easy-to-use service is designed around the concepts of "loose coupling" and "reuse" to improve openness and interoperability.

- Implements a system for the proposed ONTO_TARM framework. This system uses client and server architecture including a desktop interface and a web-based publishing platform.
- Conducts four case studies to demonstrate the system. Two cases generate maps in the same area under different temporal conditions and environmental conditions. One case demonstrates a risk map for a specific road only. Last case demonstrates the potential integration with other systems.

1.4 Organization of the Thesis

This thesis is organized as follows: Chapter 2 provides a literature review on the existing spatial clustering tools, the methods of identifying accident concentrations and the ontology in traffic accidents. Chapter 3 presents the ONTO_TARM framework with the proposed DBCTAR spatial clustering method. Chapter 4 describes the implementation of the risk mapping system and GeoClustering service. Chapter 5 gives four case studies and discusses the clustering results. Chapter 6 concludes the thesis and discusses future research directions.

Chapter Two: Literature Review

2.1 Introduction

This chapter presents an overview of related previous researches. Section 2.2 introduces traditional approaches to the road traffic accident concentration detection, mainly from the geographic perspective. Section 2.3 discusses the limitations of the current accident concentration detection approaches. Section 2.4 presents a comprehensive overview of spatial clustering methods. Section 2.5 presents an overview of the current geospatial clustering tools, mainly focusing on the online clustering tools. Section 2.6 briefly discusses the research works using ontology for traffic accidents.

2.2 Accident Concentration Detection

Identification of an accident concentration area in a road network is usually simplified into a task that detects concentrations of point events in a network. Various methods have been proposed and applied, mainly including spatial autocorrelation methods and kernel density methods.

2.2.1 Spatial autocorrelation methods

The autocorrelation methods detect whether a given point distribution differs from a random distribution throughout the study area (Boot & Getis 1988), such as Ripley's *K*-function(Ripley 1981), Getis's *G*-statistic(Getis & Ord 1992) and Moran's *I* (Moran 1950). These methods can be classified as global methods (Yamada et al. 2004) and local methods (Black & Thomas 1998), based on whether the methods apply the spatial autocorrelation significance test globally or locally within the study area.

The global methods examine whether a given point distribution differs from a random distribution. Positive spatial autocorrelation indicates that accident distribution is clustered, which means the concentration may happen in the study area. *K*-function is one of the evaluation methods. It is defined as the expected number of points within a distance *d* of an arbitrarily chosen point, divided by the density of points per unit area. The standard K-function method is based on the assumption of infinitely continuous planar space where distances are measured as a straight-line (Euclidean) distance. Yamada and Thill (2004) found a significant chance of over-detecting clustered patterns in planar K-function analysis of traffic accidents. They proposed a network K-function, which consider both the locational constraint by network and the distance measurement constraint, to resolve this problem. Shiode (2008) proposed a method for constructing a set of network-based quadrats as a basic statistic unit instead of conventional planar quadrats used for data aggregation. The limitation of these global methods is that they cannot reveal the location of clusters.

The Local Spatial-autocorrelation Methods are derived from Global Spatial-autocorrelation Methods and boomed since the 1990s (Getis 2008). Local methods need to aggregate point-based accidents into basic spatial units (BSUs). There is no unique solution for the division. For example, Flahaut et al. (2003) considered 100m long non-overlapping road section as basic spatial units and then apply local Moran's I method to determine significant clusters. However, if we conduct the research at a large scale using 200m long road section as BSUs on the same dataset, the results could be different. Thus different division methods of the BSUs may lead to different results at different scales. If the scale is too large or too small, it might mislead to false or inaccurate accident concentration result.

2.2.2 Kernel density methods

Kernel density methods aim at calculating and producing a density surface from point features. Here, the density is the total number of accidents per unit area. Usually, the methods divide the whole area into grid cells, and calculate the density of point features around each output raster cell. To do the calculation, a hump or kernel with a mathematical equation, called a kernel function, is applied to each accident point. A *kernel function* is a weighting function that is used to estimate variables' density ranging from 1 to 0 with a given radius, depending on its distance to the accident point. All the values from different points at a given cell are then totalled as the density estimation value.

The formula to calculate the kernel density values at location (*x*,*y*) is defined as:

$$D_{(x,y)} = \frac{1}{nr^2}\sum_{i=1}^{n} K(\frac{d_i}{r}),$$

where the *D*(*x*,*y*) is the density estimation value at location (*x*,*y*); *n* is the number of observations; *r* is the smoothing parameter called bandwidth (is the search radius, only points within r are used to estimate); *K* is the kernel function; d_i is the distance between the location (*x*,*y*) and location of the *i*th observation.

The traditional kernel method is a two-dimensional planar method, which generates a continuous raster surface with equal-sized cells covering the whole area in which the network located. The raster cells with high values indicate the accident concentration areas. The planar method has inherent limitations: First, all of the accidents are only located on streets. The cells that are located outside of the road have risk values that do not match the reality. Second, the density of the road network is ignored (Steenberghen 2010). Even if some grid cells have the same density values, they may include different lengths of road sections. The real density values of road network are, therefore, biased. Third, the choice of bandwidth affects the outcome surface.

To overcome these limitations, many studies have attempted to extend the conventional planar method to network spaces. Flahaut et al. (2003) developed a kernel density estimation method based on a simple network. Borruso (2005) considered the kernel as a density function based on network distances. Xie and Yan (2008) pointed out that point events in the network are better measured with density values per linear unit, but they did not consider the bias of their estimator explicitly. Okabe et al. (2009) discussed three types of the network kernel density estimation. The equal split kernel function and the equal split continuous kernel function have improved the kernel estimation methods. However, no kernel function exists that satisfies a combination of precisely estimating the density of events on a network without bias (Steenberghen 2010).

2.3 Limitations of Current Accident Concentration Detection Methods

Almost all the methods have their own weaknesses in addition to the limitations illustrated. First, all of the above methods handle accident analysis at the data level. They fail to take into consideration users' requirements. As discussed previously, accident distributions are totally different due to many factors, such as time, weather or road surface state. For example, Figure 2.1 provides a histogram showing the accident statistics on the 16th Avenue N, Calgary, Alberta, with the same time interval of the day for 4 years (1999-2002). From this chart, it can be seen that the accident numbers vary at different time intervals. This means in a specific time range of the day, the distribution of accidents should be different. Thus, two maps showing the concentration of accidents around 5:00AM and 8:00AM should be different. By a logical extension of this point, given other factors (such as the weather conditions), the risks of the road network should be different. However, current methods do not consider different factors. Although datasets can be generated from database query, the processing remains at the data level, not at the knowledge level. Therefore, current methods cannot satisfy users' needs.

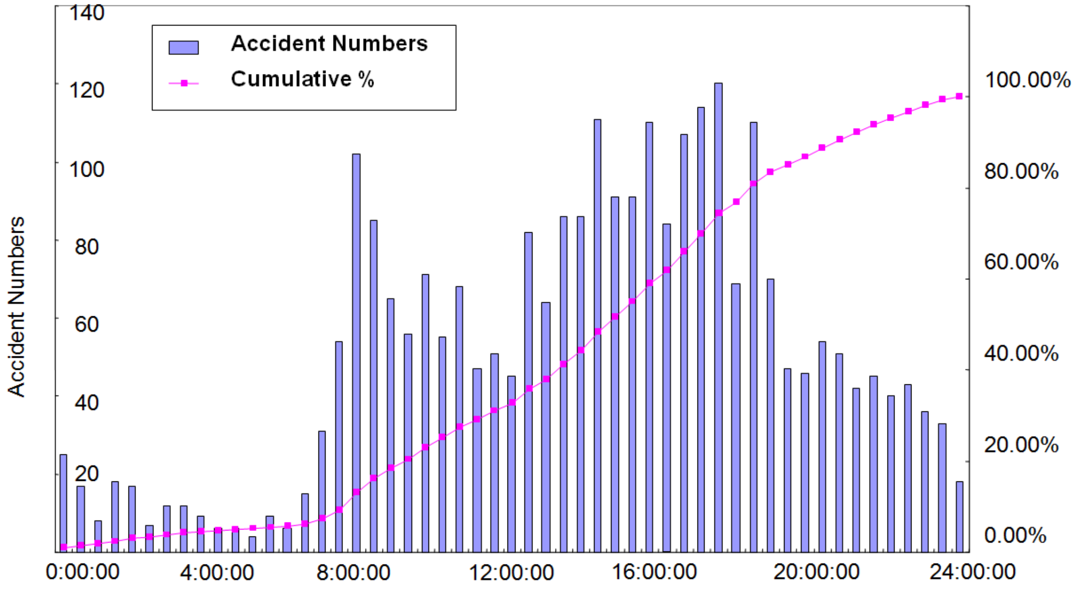


Figure 2.1 Accident statistics in the same time intervals on 16th Ave N in Calgary

Second, all of the above methods ignore the severity level of accidents. When users consider the accident risk of the road network, the assumption is that, if an area on the map is marked as high risk, that area should be more vulnerable to accidents. However, the nature of the accidents may be different from one another. One of the obvious distinctions is the severity level. For example, a rear-end accident should not be considered the same as an accident with a fatality. In the Figure 2.2 assume during the same time interval, both intersection A and B have 4 accidents. Around intersection A, there are two accidents with injury, and two accidents with property damage only. Around the intersection B, all the accidents are “accident with property damage only”. If we only count the total accident numbers at each intersection, A and B have the same risk. But if we consider the severity level, A should be considered as higher risk than B. Thus, the risk not only depends on the number of accidents, but also on the severity level of accidents. Unfortunately, most of the previous studies take the accident records as a point without considering the severity levels, and most of the statistical analyses are only based on the number of the accidents.

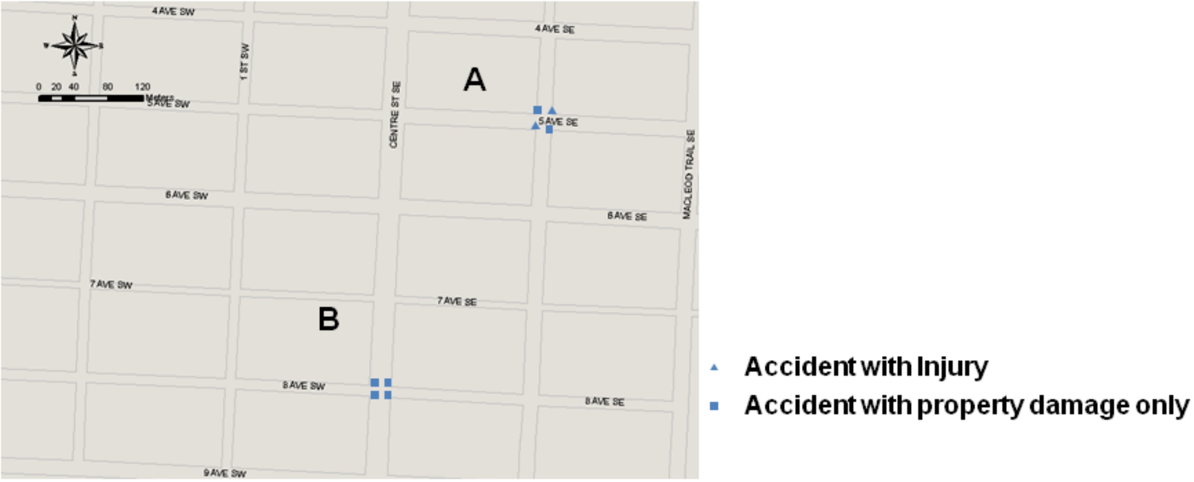


Figure 2.2 Accident numbers and severity levels

2.4 Geospatial Clustering

Geospatial clustering is the process of grouping a set of objects into groups (called clusters) based on their geographical locations and other attributes. Geospatial clustering can use general spatial clustering algorithms but emphasizes applying spatial clustering algorithms in geographical space. The rules which describe a step-by-step procedure for grouping the spatial objects based on pre-defined criteria among the spatial objects are called spatial clustering algorithms or methods. The “pre-defined criteria”, most time is based on the similarity. The function how to define the similarity among the spatial objects is called “distance function”.

2.4.1 Spatial clustering methods

Spatial clustering algorithms exploit spatial relationships among data objects to discern groupings inherent within the input data. The spatial clustering methods can be classified into five categories, based on the underlying clustering technique used (Han et al. 2001, Han & Kamber 2006).

1) Partitional methods: Partitional clustering methods partition points into clusters, such that the points in a cluster are more similar to each other than to points in different clusters. They start with some arbitrary initial clusters and iteratively reallocate points to clusters until a stopping criterion is met. These methods tend to find clusters with hyperspherical shapes. Examples of partitional clustering algorithms include k-means, PAM(Partitioning Around Medoids), etc (Kaufman & Rousseeuw 1990, Ng & Han 2002).

2) Hierarchical methods: Hierarchical clustering methods can be either agglomerative or divisive. An agglomerative method starts with each point as a separate

cluster, and successively performs merging until a stopping criterion is met. A divisive method begins with all points in a single cluster and performs splitting until a stopping criterion is met. Examples of hierarchical clustering methods are CURE(Clustering Using REpresentatives), BIRCH(Balanced Iterative Reducing and Clustering using Hierarchies), etc (Guha et al. 1998, Zhang et al. 1996, Karypis et al. 1999).

3) Density-based methods: Density-Based clustering methods try to find clusters based on the density of points in regions. Dense regions that are reachable from each other are merged to form clusters. Density-based clustering methods excel at finding clusters of arbitrary shapes. Examples of density-based clustering methods include DBSCAN(Density-based Spatial Clustering of Applications with Noise), OPTICS(Ordering Points To Identify the Clustering Structure), DBRS(Density-Based *clustering* with Random Sampling), etc (Ester et al. 1996, Ankerst et al. 1999, Wang & Hamilton 2003). 4) Grid-based methods: Grid-based clustering methods quantize the clustering space into a finite number of cells and then perform the required operations on the quantized space. Cells containing more than a certain number of points are considered to be dense. Contiguous dense cells are connected to form clusters. Examples of grid-based clustering methods include STING(STatistical INformation Grid), CLIQUE(CLustering In QUEst), etc (Wang et al. 1997, Agrawl et al. 1998, Sheikholeslami et al. 1998).

5) Model-based methods: Model-based methods hypothesize a model for each of the clusters and attempt to optimize the fit between the data and some mathematical models such as Autoclass and COBWEB (Cheeseman et al. 1993, Fisher 1987).

Among these different types of methods, a density-based clustering method is the most suitable method for traffic accident risk analysis, because it can discover arbitrarily shaped clusters based on density. The section 2.4.3 will discuss the classic density-based clustering method on which this research is based.

2.4.2 Distance functions

The “distance function” is the key component of any spatial clustering method in that it measures the similarity among spatial objects. Distance is a numerical description of how similar two objects are in space. According to Tobler’s First Law (Tobler 1970), geometric distance is usually used as the scale of similarity in the “ideal model”. However, sometimes the non-spatial attributes (alphanumeric attributes) of objects is also incorporated into the distance function.

The character of geometric distance is that it is defined by exact mathematical formulas that reflect the physical length between two objects in defined coordinate systems, such as Euclidean Distance, Manhattan Distance, Great Circle Distance, etc. Spatial clustering methods do not always use geometric distance; for example, if the distance can be defined as the shortest traveling time between two different addresses in a city. In this case, the distance function should take into account road networks, speed limitations, volume of traffic, number of traffic lights, and stop signs. In fact, the distance function is always tailored to different clustering purposes.

Spatial objects may have significantly different non-spatial attributes that distinguish them from each other and influence the clustering result. Consequently geometric distance will sometimes be extended to include not only coordinates but also non-spatial attributes. Non-spatial attributes can be classified into two categories: numerical and non-numerical. For numerical non-spatial attributes, the numerical values can usually be transformed into some standardized values, and calculated by using geometric distance functions as additional dimensions. For the non-numerical attributes, new functions are defined to transform non-numerical values to numerical, such as “weight” in GDBSCAN (Sander et al. 1998) or “purity” in DBRS (Wang & Hamilton 2003).

2.4.3 DBSCAN algorithm

DBSCAN (Density Based Spatial Clustering of Applications with Noise) (Ester et al. 1996) was the first density-based spatial clustering method proposed. The key concept is the definition of a new cluster or extension of an existing cluster based on a neighborhood. The neighborhood around a point of a given radius (*Eps*) must contain at least a minimum number of points (*MinPts*). Given a dataset *D*, a distance function *dist*, and parameters *Eps* and *MinPts*, the following definitions are used to define DBSCAN.

An arbitrary point $p(p \in D)$, the neighborhood of p is defined as $N_{Eps}(p) = \{q \in D \mid dist(p,q) \leq Eps\}$. If $| N_{Eps}(p) | \geq MinPts$, then p is a core point of a cluster. If p is a core point and q is p’s neighbor, q belongs to this cluster, and each of q’s neighbors is examined to see if it can be added to the cluster. Otherwise, point q is labelled as noise.

The expansion process is repeated for every point in the neighborhood. If a cluster cannot be expanded further, DBSCAN chooses another arbitrary unlabelled point and repeats the process. This procedure is iterated until all points in the dataset have been placed in clusters or labelled as noise. The pseudocode of DBSCAN is showing below.

```
DBSCAN (SetOfPoints, Eps, MinPts)//SetOfPoints is UNCLASSIFIED

  ClusterId := nextId(NOISE);
  FOR i FROM 1 TO SetOfPoints.size DO
    Point := SetOfPoints.get(i);
    IF Point.ClId = UNCLASSIFIED THEN
      IF ExpandCluster(SetOfPoints, Point, ClusterId, Eps, MinPts)
      THEN
        ClusterId := nextId(ClusterId)
      END IF
    END IF
    END FOR
END; // DBSCAN
```

```
ExpandCluster(SetOfPoints, Point, ClId, Eps, MinPts) : Boolean;

  seeds := SetOfPoints.regionQuery(Point,Eps);
  IF seeds.size < MinPts THEN // no core point
    SetOfPoint.changeClId(Point,NOISE);
    RETURN False;
  ELSE // all points in seeds are density reachable from Point
    SetOfPoints.changeClIds(seeds,ClId);
    seeds.delete(Point);
    WHILE seeds <> Empty DO
      currentP := seeds.first();
      result := SetOfPoints.regionQuery(currentP, Eps);
      IF result.size >= MinPts THEN
        FOR i FROM 1 TO result.size DO
          resultP := result.get(i);
          IF resultP.ClId IN {UNCLASSIFIED, NOISE} THEN
            IF resultP.ClId = UNCLASSIFIED THEN
              seeds.append(resultP);
            END IF;
            SetOfPoints.changeClId(resultP,ClId);
          END IF; //UNCLASSIFIED or NOISE
        END FOR;
      END IF; //result.size >= MinPts
      seeds.delete(currentP);
      END WHILE; //seeds <> Empty
    RETURN True;
  END IF
END; //ExpandCluster
```

Net-DBSCAN (Stefanakis 2007) is an example of a spatial clustering method used in GIS to extend DBSCAN to a network environment. However, this method can only deal with the nodes of a dynamic linear network. Yiu and Mamoulis (2004) proposed a modified DBSCAN method used for clustering objects on a spatial network. This method well defined the distance and initial clusters under the network environment.

Compared with traditional concentration detection methods, a density-based spatial clustering method can inherently discover the concentrations; dataset segmentation is not needed at the beginning of the process. It can also find arbitrary concentration shapes from the dataset. However, most density-based clustering methods inherited the limitation of original DBSCAN, which does not take into account any non-spatial attributes of the data point and cannot be directly applied to accident datasets on a spatial network.

2.5 Current spatial clustering applications

It is hard to find applications specifically designed for geospatial clustering tasks. However, general spatial clustering applications have been developed and used in both local and online environment for a long time. Most existing spatial clustering tools are included in desktop-based software or packages. Recently, some online geospatial clustering tools have been proposed and developed. This section will generally summarize the local general spatial clustering applications and discuss online geospatial clustering tools in details.

2.5.1 Local clustering applications

The related local desktop-based software or packages are widely used, such as SaTScan (Kulldorff 2009) or ClusterSeer (ClusterSeer 2009). These applications are effective for dealing with data available at local machines only, but cannot handle online data sources. For example, if users need to find spatial clusters of accidents records published by an RSS feed, the traditional software cannot handle the request directly. Rather, users need to download and transform the data before performing the clustering.

Besides, these applications are not designed with the interoperability. Interoperability is the ability of a system, or components of a system, to provide information portability and inter-application cooperative process control (Bishr 1998). Current clustering functions are usually packaged as part of a specific and proprietary system. It is difficult to be utilized by other applications without knowing the system API (Application Programming Interface) specifications. The input and output files are usually proprietary and defined by the system. Users cannot choose to use common spatial data standards or to use them as third-party clustering service components. Consequently, exchanging data between different systems or reusing the service is difficult, if not impossible.

In addition, these applications lack geo-visualization capability. The clustering results provided by these tools are mainly in text form or simple graphics. Users cannot visualize the clustering results on the map.

2.5.2 Online clustering applications

The online geospatial clustering tools are usually used for map feature simplification. Map scale reduction inevitably leads to conflict and congestion of map symbols. To make the maps legible, appropriate operations (e.g., selection, simplification, aggregation, etc.) must be employed to simplify map features (Yan & Weibel 2008). The most common example is cartographic generalization for online map symbols. For example, as shown in Figure 2.3, when a large number of photos are given for a geographic region such as the downtown San Francisco area, users may want to find a set of “representatives” to improve the display.

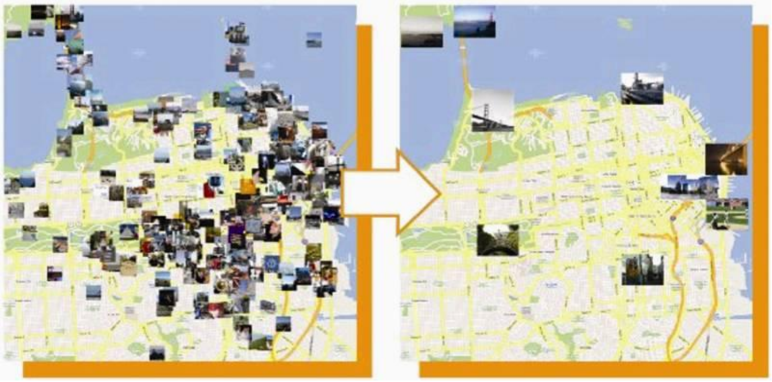


Figure 2.3 An online spatial clustering example - Tag Map (Jaffe et al., 2006)

Client-Server architecture is very popular among online clustering tools. It is a distributed application architecture that partitions tasks or workloads between service requesters (clients) and service providers (servers). The client is responsible for interacting with users. A server is usually a high-performance host that offers functions and/or resources. When users submit a request to the client, the client requests the server's content or service function. The server will respond to the user’s request. In terms of implementation, the underlying techniques can be divided into client-side and server-side methods.

2.5.2.1 Clustering on the client side

Client-side clustering methods are usually attached to specific online map services APIs. The following discussion will use Microsoft Bing Maps and Google Maps, the most popular mapping platforms, as examples to introduce client-side methods.

Microsoft Bing Maps AJAX Control API (former Virtual Earth Map Control API) offers a built-in method, `VEShapeLayer.SetClusteringConfiguration(type, options)`, which can set the method to determine which symbols are clustered, as well as how the clustering result is displayed (Microsoft 2009). The first parameter `type` has two values: `None` and `Grid`. In case of `None`, this method will return the original symbols. In case of `Grid`, a simple grid-based clustering algorithm will be used. In addition, users can override this method with the name of other clustering method functions in the form of `VEShapeLayer.SetClusteringConfiguration (algorithm, options)`, where `algorithm` is the name of the clustering method functions developed by users. `options` is specifying how the cluster result is displayed.

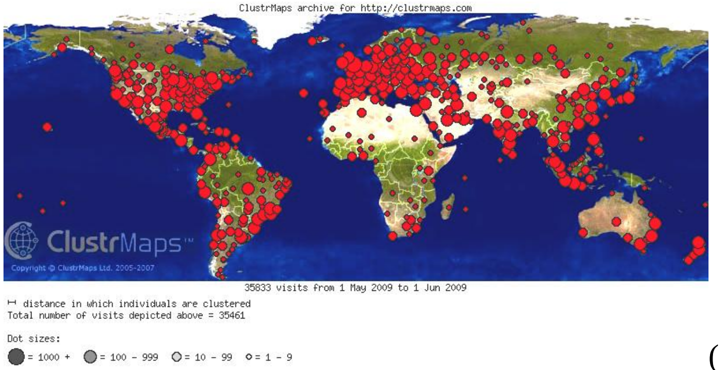
Google Maps API does not offer out-of-box functions. However, several Google Maps supporters lunched several projects to help users manage the symbols on the map, such as Google Maps Clustering API Project (Pearman 2009) and ACME cluster JavaScript library (ACME Labs 2009). Similar to the clustering function offered by the Bing Maps API, users can use these JavaScript APIs at the client side to group symbols on the map.

The clustering techniques at the client side are relatively simple. Grid-based clustering algorithms are the most common method. Most of these are implemented in JavaScript and the clustering work is usually performed by the browser. Subsequently the browser and online map services APIs display the clustering result as a layer superimposed on the map. This mechanism restricts the size of clustering data to be relatively small, usually under few thousand points.

2.5.2.2 Clustering on server side

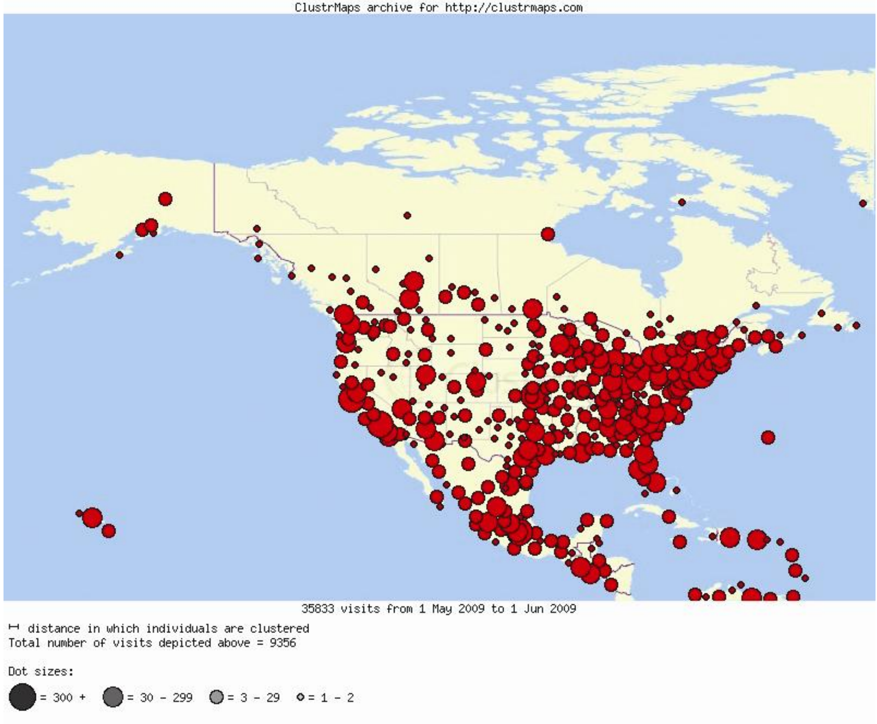
Server-side clustering methods are relatively independent from online map services. The implementation technique can be divided into either real-time methods or pre-processing methods. For real-time methods, the server responds to the user's query and performs clustering on the fly. This method cannot be used for applications with huge data since the response time is much too long. To improve the clustering response time for huge datasets, pre-processing methods are applied in which pre-processing methods pre-cluster the dataset at different zoom levels. Usually the server first converts each symbol's position to a pair of pixel coordinates on the screen at each zooming level. It then calculates the pixel distance between symbols and combines the symbols closest to one another into one cluster symbol. Finally, these cluster symbols are saved on the server and displayed at their pre-determined optimal zoom levels, depending on users' queries. Generally, there are two ways to save the pre-processing result: static raster images and vector point files.

ClustrMap (ClustrMap 2009), shown in Figure 2.4, illustrates the use of a static raster image. It is an archived clustering map based on the visitors to the website clustrmap.com from May 1st to Jun 1st 2009. This clustering map has two zooming levels: global and continental. Different images are returned according to the user's request. Figure 2.4(a) is the clustering map at the global scale while Figure 2.4(b) is the clustering map at the continental scale. The static images are usually updated after a fixed time period, for example every 24 hours.


ClustrMaps archive for http://clustrmaps.com
ClustrMaps™
Copyright © ClustrMaps Ltd, 2005-2007
35833 visits from 1 May 2009 to 1 Jun 2009
⊢ distance in which individuals are clustered
Total number of visits depicted above = 35461
Dot sizes:
● = 1000 +
● = 100 - 999
○ = 10 - 99
○ = 1 - 9

(a)

http://www2.clustrmaps.com/counter/maps.php?url=http://clustrmaps.com&clusters=yes&hist=2009-05-01_to_2009-06-01&type=small&category=plus&map=world


ClustrMaps archive for http://clustrmaps.com
35833 visits from 1 May 2009 to 1 Jun 2009
⊢ distance in which individuals are clustered
Total number of visits depicted above = 9356
Dot sizes:
● = 300 +
● = 30 - 299
● = 3 - 29
○ = 1 - 2

(b)

http://www2.clustrmaps.com/counter/maps.php?url=http://clustrmaps.com&clusters=yes&hist=2009-05-01_to_2009-06-01&type=small&category=plus&map=North %20America

Figure Copyright (C) ClustrMaps Ltd, 2012, www.clustrmaps.com, reproduced with permission.

Figure 2.4 Clustering results, in the form of static raster images, for two zooming levels: (a) Global, (b) Continental (ClustrMap 2009)

Figure 2.5 shows an example of using vector point files to display zoom-scale-dependent point-location information (Maiom 2009). This sequence of maps shows the location of real estate for sale or rent in Italy. Each real estate can be considered as a point. To avoid overlap and to make the map more legible, the system only shows the clustering result of points instead of the real locations at the small zoom level. During pre-processing, the system saves the clustering result at different zooming levels as “markers” into different xml files. Each “marker” has a new coordinate to represent the points in this cluster. When users view the map at different zooming levels, the server returns the corresponding xml files. Figure 2.5 shows “markers” for the city “Firenze” at different zoom levels.

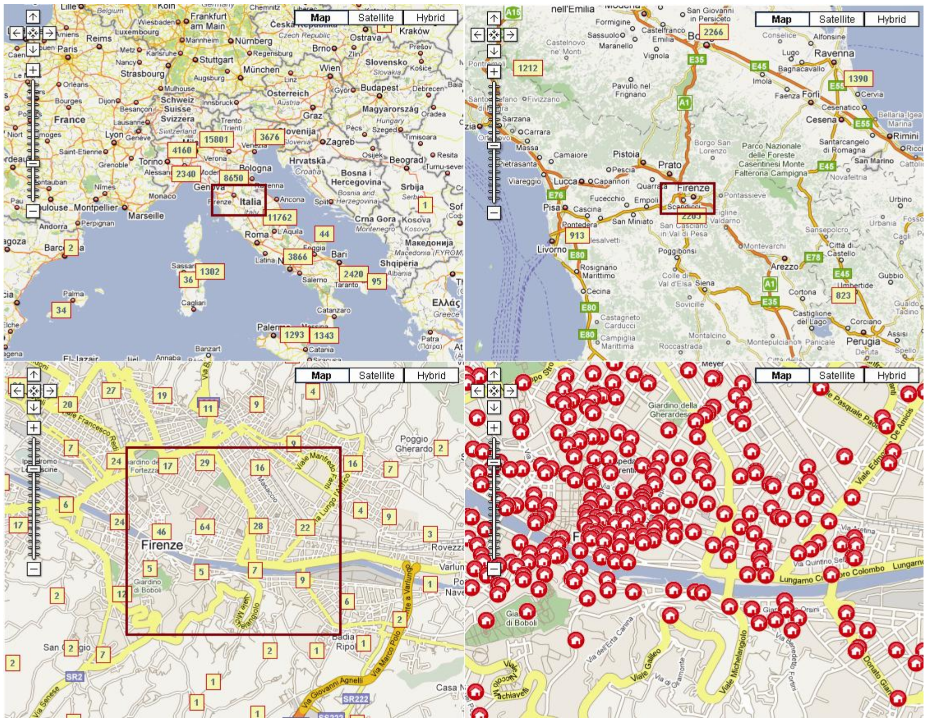


Figure 2.5 Clustering results in the form of vector “markers” for four zooming levels (Maiom 2009)

2.5.2.3 Limitations of current online clustering tools

Current online clustering tools utilize clustering technique to simplify map symbols for online map applications. These tools have some limitations:

First, existing online clustering APIs are not designed for general users, i.e., they are not easy to use. Some clustering function APIs are available on the Internet. For example, Microsoft Bing Maps Control AJAX API offers built-in clustering functions. Google Maps Clustering API Project and ACME cluster JavaScript library add clustering functions to Google Maps. However, these are designed for developers, not for general users. Thus, in order to implement their own clustering assignment, general users need to learn specific programming languages and API usages.

Second, in terms of clustering functions, most web-based map applications such as ClustrMap provide embedded clustering functions only for cartographic generalization and do not focus on clustering patterns. Grid-based clustering is widely used to combine neighbor points into one single cluster to reduce the total number of symbols being displayed in the current map view. Here clustering is used as a technique with which several points of interest can be represented by a single icon when they are close to one another. These tools are not concerned with patterns or with showing patterns on the map.

2.6 Ontology in Traffic Accidents Research

Ontology is an explicit representation of knowledge. It is a formal, explicit specification of shared conceptualizations, representing the concepts and their relations that are relevant for a given domain of discourse (Gruber 1993). Generally, ontology contains basic modeling primitives such as classes or concepts, relations, functions, axioms and instances (Gómez-Pérez et al. 2004).

The last few years have seen a growing interest (Peuquet 2002) in approaches that have domain ontology add a conceptual level over the data, which is used as a middle layer between the user and the dataset, especially with spatial data. Several ontological approaches are proposed for road accidents. Hwang (2003, 2004) built a high-level conceptual framework that includes traffic accident domain ontology. However, this research focused on the task ontology and did not consider the disparity of accidents. Yue et al. (2009) presented an ontology-based prototype framework for traffic accident management from a hierarchical structured point of view. However, the ontology was designed only for the traffic management system. It rarely considered the spatial knowledge associated with the accidents.

The Web Ontology Language (OWL), as one of the most widely used ontology languages, is designed for use by applications that need to process the content of information instead of just presenting information to humans (Smith 2004). It is endorsed by the World Wide Web Consortium (W3C). OWL provides additional vocabulary along with a formal semantics and facilitates greater machine interpretability on top of RDF/XML-based serializations, such as XML, RDF, and RDF Schema (RDFS).

2.7 Traffic Accidents Road Safety Research

The traffic accident is usually affected by three factors: the road environment, the condition of vehicles and the skill, concentration and physical state of road users (Geurts

2003). As the absolute number of accidents cannot well reflect the road safety, the concept of risk indicator is proposed.

Currently the main form of numerically risk assessment in road transport area is usually defined by recorded numbers of fatalities (or casualties) and some measures of exposure (Shen et al. 2012). European Road Safety Observatory defined the risk as a ratio of road safety outcomes and some measures of exposure (ERSO 2007). Different researchers may use different indicator (e.g. size of population, time in traffic, traffic density) of exposure to describe risk from different points of view. International Traffic Safety Data and Analysis Group (IRTAD, 2012) pointed out that the three most frequently used measures of exposure to risk are: population size, the number of registered vehicles, and the distance travelled.

Chapter Three: An Ontology-Based Traffic Accident Risk Mapping Framework

3.1 Overview

This chapter presents the ontology-based traffic accident risk mapping (ONTO_TARM) framework. As stated in Chapter 1, accidents are not random in geographic space, but form clusters on road networks. If we can generate maps showing the high-risk road segments, it will help people reduce the road accidents. Previous studies have proposed various spatial concentration detection methods; however, they only generate one single map without consideration of the discrepancy of conditions.

Users with individual objectives may have different requirements. For example, a traffic analyst may be interested in an accident concentration area for the downtown area during workdays at rush hours, so that the most vulnerable locations with accidents can be located and the reasons behind these accidents can be analyze to improve road safety. A new driver may be interested in a map of the northwest part of the city during winter weekends, in order to avoid dangerous areas when practising driving in this area during the winter time.

The integration of users' requirements into map generation process is necessary. The first subtask is the selection of the proper datasets based on different users' requirements. One naive option is the translation of users' requirements into traditional database queries. For example, in the former example of downtown workday rush hours, "downtown area" and/or "rush hours" must be defined, so users can handle the traffic accident data at the data level, which can sometimes be quite challenging as the user may have no or only limited knowledge of the study area and dataset. However, if knowledge of the study area and datasets are well defined and represented, users' requirements can be handled at the knowledge level.

After selecting the proper datasets, the second task in the generation of a concentration map is the application of proper traffic accident concentration detection methods. Existing traffic accident concentration detection methods do not consider the distinctions of each accident, only the number of accidents. Accidents with fatalities and injuries put more strain on the network than property damage only (PDO) accidents. An intersection with frequent fatal accidents may be more dangerous than an intersection with PDO accidents, in cases where both intersections have the same number of accidents.

To address users' requirements at the knowledge level, the traffic accident domain ontology (TADO) is constructed, and an ontology-based reasoning process is involved. TADO is built at a high generic level with a conceptual and taxonomical representation of accident data. It provides domain knowledge, including non-spatial and spatial concepts, as well as definitions relating to the traffic accidents. This enables users to pose semantic queries with a semantic representation of traffic accident concepts. Therefore, TADO can provide a knowledge source that supplements domain experts and integrates users' goals into the selection procedure.

To consider the severity level of each accident, the density-based clustering algorithm for traffic accident risk (DBCTAR) with a risk model is proposed. The DBCTAR, which is extended from DBSCAN, inherits the advantages of discovering arbitrary shapes and, in particular, identifying the regions in the data space where the objects are dense. In addition, the DBCTAR is designed under a network environment with the ability to consider the severity level through a newly added parameter, “Risk Index”.

3.2 ONTO_TARM Framework

This thesis structures and organizes the core components of domain ontology and the newly proposed clustering algorithm to establish the ontology-based traffic accident risk mapping (ONTO_TARM) framework. This framework illustrates the proposed work, broadens the concept of generic risk mapping solution in such a way that readers can have a better understanding of the whole mapping procedure, and better supports future implementation and development work. Figure 3.1 shows the proposed ONTO_TARM framework.

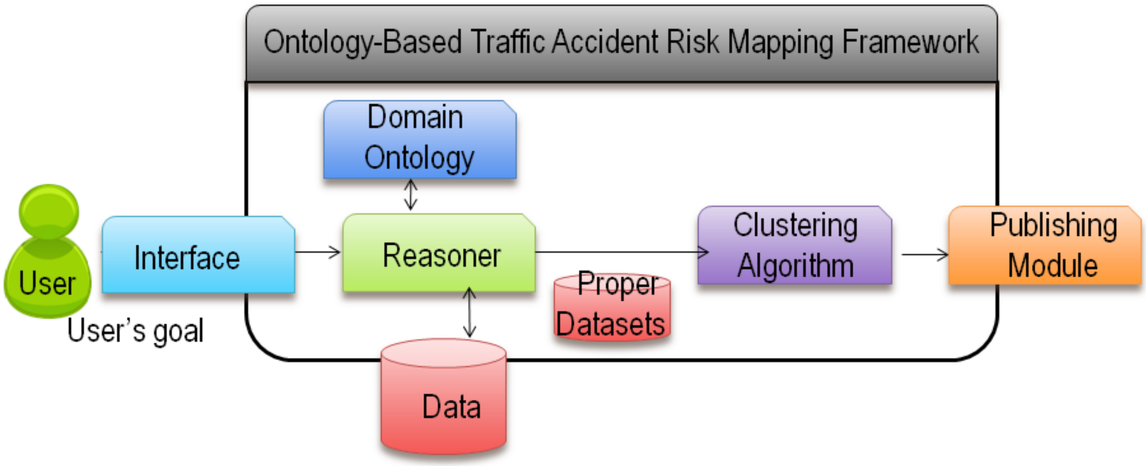


Figure 3.1 Entire Ontology-based traffic accident risk mapping (ONTO_TARM) framework

The ONTO_TARM framework includes an interactive input module, an accident domain ontology, an ontology reasoner, datasets, a clustering algorithm with a risk model and a map publishing module.

With this framework, the users' goals provide the input, which can be represented with some predefined words. For example, users can input the time range or select predefined time sections, such as morning or rush hours. The traffic accident domain ontology represents the knowledge of accidents. Spatial concepts are well defined; and, each accident record is described by several characteristics, such as crash time, location and environmental factors. The ontology reasoner is used to reason the knowledge represented in the ontology. It contains the classification and decomposition rules. The users' requirements are translated into a set of subtasks by performing reasoning on the ontology. The traffic accident datasets contain all the accident records. In this research, it is assumed the datasets include all the traffic accident data that meet different users' requirements. Appropriate datasets can be chosen by a selection procedure guided by the ontology with respect to the user's goals. The clustering algorithm is used to find the traffic concentration areas based on users' goals. After identifying the proper datasets, the proposed density-based spatial clustering method with a user selected traffic accident risk model, offered by the web-based geospatial clustering service, are applied to the dataset to identify the traffic accident risk area. The publishing module is the output part of this framework for the final map generation and publication. The different traffic accident risk maps that meet users' requirements can be finally generated and published.

The whole framework works as follows: The interface handles users' goals as inputs, sending them to the reasoner. The reasoner parses the users' goals into tasks based on the domain ontology, conducts queries for each task and returns with the proper dataset. Finally, the clustering algorithm is run on the proper dataset, and a risk map is generated and published. In the following subsections, each component is introduced and discussed. This chapter is presented from a design point of view. Detailed implementations of the framework are discussed in Chapter 4.

3.3 Traffic Accident Domain Ontology

Traffic accident domain ontology (TADO) provides formal descriptions of the classes of concepts and the relationships among those concepts that describe road traffic accidents. With TADO, users can retrieve the proper datasets without knowing the details of the area. The structure of TADO is based on Wang et al.(2010).

Definition 1 (**Domain ontology structure**) An ontology structure of a domain is a 7-tuple $O := \{D, C, R, A, H^C, prop, att\}$, where D is the domain context identifier, C is a set called concept, R is the relation identifiers (C and R are disjoint and provide necessary conditions for membership), and A is a set of attributes to describe C and R. H^C, which is a concept hierarchy classification, is a set of hierarchical trees that define the concept taxonomy in the domain. The *prop* function relates concepts non-taxonomically: $R \rightarrow C \times C$. Each attribute in A can be treated as a specific kind of relation, where the function *att* relates literal values to concepts: $A \rightarrow C$. Elements C and R can be regarded as the high-level encapsulation of the analysis and design model for the ontology.

Definition 2 (**Classification**) H^C is a set directed, transitive relations: $H^C = \{ h^C \subseteq C \times C \}$, where $h^C(C_1, C_2)$ means that C_1 is a sub-concept of C_2 in the relation h^C. Usually, H^C includes a set of classification instances. Depending on the application, the classification constraints may be different. Even the same concept can be categorized into several categories.

Each component of the top-level ontology is discussed in detail.

Domain Context Identifier

In TADO, the domain context identifier *D* is *TrafficAccidentRecordDomain.*

Concepts

Accident records include spatial and non-spatial information. For example, each accident record has the attributes of location and accident time. Therefore, the concept set *C* of TADO includes three main classes: *GeospatialThing*, *AccidentRecord* and *AccidentCondition* to represent this information.

For the spatial information, the ontology conceptual tree is extended from the Cyc knowledge base (Cyc 2009) by altering the *GeographicalThing* to *GeospatialThing* with customized spatial classes. The Cyc knowledge base was selected because it is the most commonly used ontology and it contains a great quantity of common sense knowledge encoded in formal logic. *GeospatialThing* is defined as an abstract class to provide the basic classes of geospatially related concepts or entities that can be used to describe the locations of accidents. It includes subclasses *GeometricThing, FixedStructure* and *GeographicalRegion*. *FixedStructure* presents the facilities related to the accidents, such as the road. *GeographicalRegion* describes the geographical area with a specific boundary. Any geographical region used in TADO is an instance of *GeographicalRegion.* Various geographical regions, such as *Province*, *City*, *County*, *Community* and *CitySection,* are defined. *Province*, *City* and *County* are defined as regions with political boundaries. *Community* is derived from the census subdivisions and can be classified into city sections. *CitySection* is the region in the city that has formed over a long historical period. For example, Calgary is an instance of the class *City*. Within the boundaries of Calgary, there are around 100 communities, with each community belonging to at least one of the five Calgary *CitySections*, which are the NW, SW, SE, NE and downtown areas.

The non-spatial information describes the non-spatial properties of the accidents. It includes two main subclasses, *AccidentRecord* and *AccidentCondition*. *AccidentRecord* represents the class of the available accident record data. Any record used in TADO is an instance of *AccidentRecord*. Non-spatial properties of this class are defined in *AccidentCondition*, such as *TemporalConditions* and *EnvironmentalConditions*. The *TemporalConditions* class includes different abstract classes based on different time scales, from hourly to yearly. The temporal concepts, such as rush hours and slow hours, are also defined. The *EnvironmentalConditions* define various accident related environmental factors. Examples of these classes are *WeatherConditions* and *RoadConditions*.

Relations

Relations consist of the relationships among *GeospatialThing*, *AccidentRecord* and *AccidentCondition*. *Geospatial Relation* and *AccidentCondition Relation* are the two major types of relations. *Geospatial Relation* includes the spatial relationships among *GeospatialThing*. There are three kinds of geospatial relations: direction, distance and topological relations. A direction relation describes the orientation in space of some objects, such as *north*, *south*, *up*, *down*, *behind* and *front*. A distance relation specifies the distance from an object to a reference object. Some examples of distance relations are *far* and *close-to* (near). A topological relation describes the location of an object relative to a reference object (Egenhofer 1991). Topological relations include *disjoint*, *contains/insideof*, *overlap*, *cover/covered* and *meet*. *AccidentCondition Relation* defines relations between *AccidentRecord* and *AccidentCondition*. The relationships also include temporal and non-temporal relationships. Examples of temporal relationships are *at time point of, around time, in the range of, early than, later than*. An example of a non-temporal relationship is *with the condition of*.

Attributes

Attributes define the attributes and properties of the above classes and their subclasses. One example of spatial attributes is *location*. Some examples of non-spatial attribute include *hasName*, *hasValue*, *hasTime*, *hasDate*.

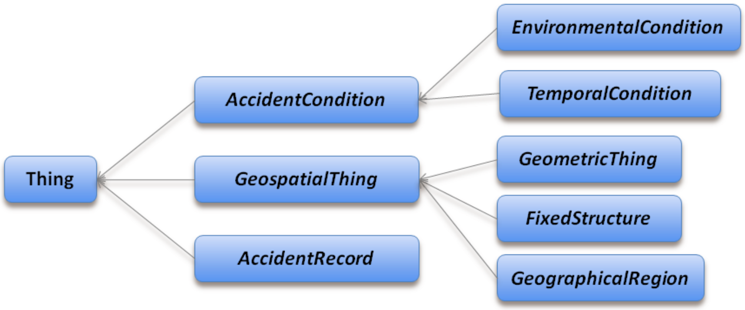


Figure 3.2 Top-level conceptual three in TADO

Classification

Classification includes the hierarchical classification used for TADO. Figure 3.2 shows the top-level ontology defined in TADO and the hierarchical classification of *GeospatialThing* and *AccidentCondition*. As shown in Figure 3.3, *GeospatialThing* is the top class of all spatial things in TADO. In the three subclasses, the *GeometricThing* class includes abstract geometric shapes. *FixedStructure* presents the facilities related to the accidents and includes classes such as *Building*, *Station*, *Roadway*. The *Roadway* class includes subclasses *Expressway*, *Highways*, *Majorroad*, and *Localroad*. Under the class *GeographicalRegion*, we have *EcologicalRegion*, *GeocuturealRegion*, *GeopoliticalRegion*. Subclasses *Country*, *Province*, *City*, *County* and *Community* belong to the *GeopoliticalRegion*.

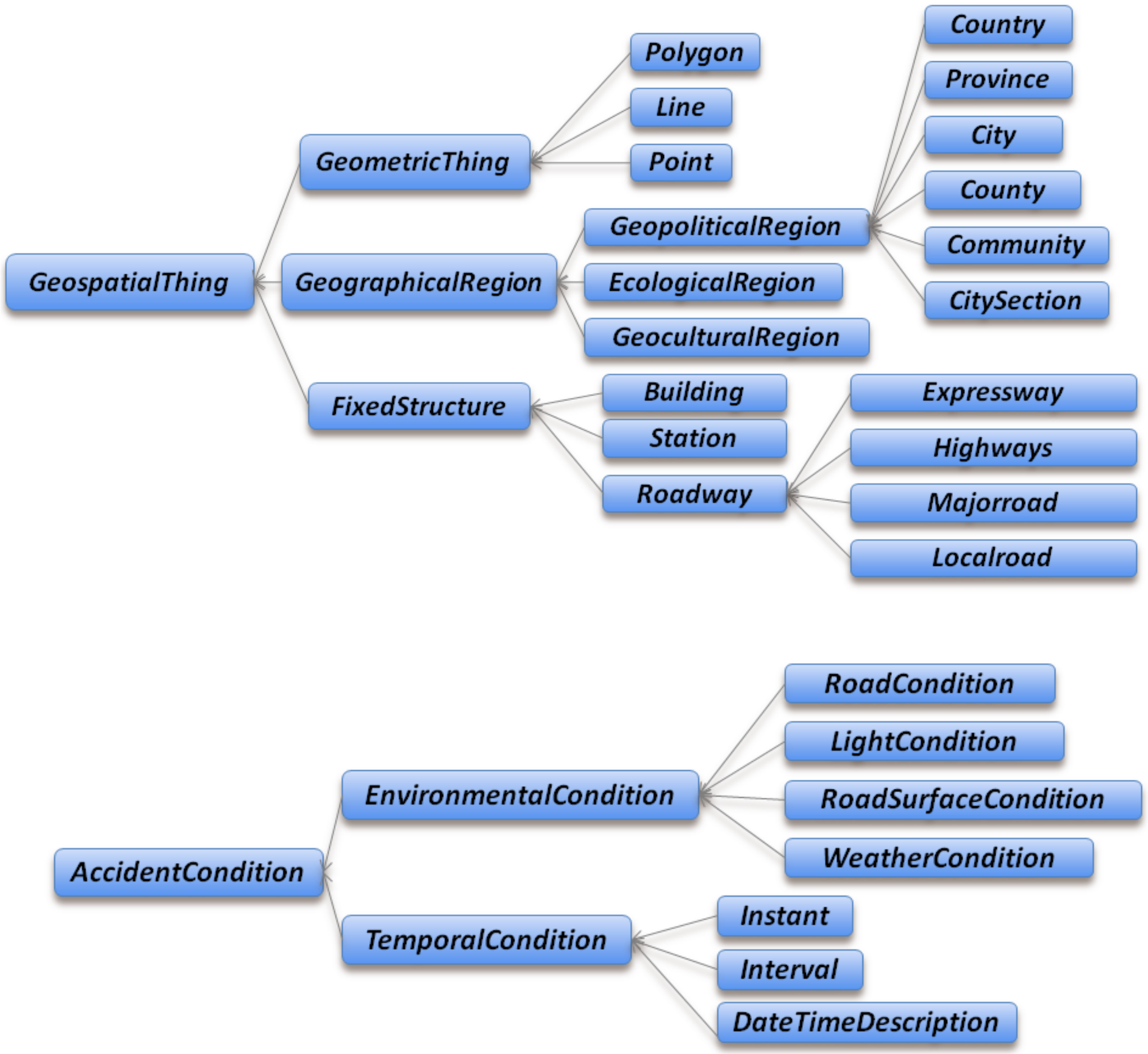


Figure 3.3 Classification of TADO

AccidentCondition can be classified into *TemporalConditions* and *EnvironmentalConditions*. The *TemporalConditions* include *Instant*, *Interval* and *DateTimeDescription* class. The *EnvironmentalConditions* include the *WeatherCondition*, the road *RoadSurfaceCondition*, the *RoadCondition*, *LightCondition*. Each has detailed subclasses. For example, the *WeatherCondition* includes *SevereWeather* and *FairWeather*. The *high_wind*, *fog_smog_smoke_dust*, *hail_sleet*, *raining* and *snow* are all in the severe weather condition class.

Function *prop*

The function *prop* relates concepts non-taxonomically among the concepts. It can be an instance of geospatial relation or non-geospatial relation, such as *underconditionof*() and *insideof*(). Here, using *insideof* as an example. In Figure 3.3, *City* and *Community* are two classes (concepts) in the ontology, and class *City* is not a super-class of the class *Community*; therefore, *insideof*(*City*, *Community*) represents whether a community is inside a city. Thus, *insideof* defines one type of relationship between instances of the two classes.

Function *att*

The function *att* is used to describe the properties or attributes of a class. For example, *hasName* is used to define the names of instances of each class.

Reasoner

In the framework, the ontology reasoner is used to reason the knowledge represented in the ontology. The input of the reasoner is the user's goals, and the output is a set of proper accident records selected from the raw dataset. After generating a general task from a user's goals, the spatial task identifies the target geographical area. The non-spatial task identifies the proper temporal and environmental factors. For example, if a user's goal is the generation of a risk map for accidents that happened in rush hours on workdays in downtown Calgary, the reasoner first finds "downtown Calgary". A spatial query task is generated as shown in Figure 3.4.

The non-spatial task is materialized by the task "accidents that happened in rush hours on workdays with severe weather". This task includes two main components: a temporal condition task and an environmental condition task, as shown in Figure 3.5. The non-spatial attributes are both complex tasks that need further decomposition. The temporal condition task is composed of two subtasks: finding "rush hours" and finding "working days". The weather condition task is finding "severe weather" and will return conditions including high_wind, fog_smog_smoke_dust, hail_sleet, raining, and snow.

```
sub-task: findDowntownAreaTask
defgoal find Calgary Downtown Area
Input:
(object (is-a City) (object?ci) (hasName "Calgary"))
(object (is-a CityfSection) (object?cs)
  (hasName "Downtown Area") (insideOf?ci))
(object (is-a community) (object?co) (insideOf?ci)
   (belong-section?cs))
Output:
(object (is-a $?community) (object? co))
```

Figure 3.4 Pseudocode of spatial query task findDowntownAreaTask

```
sub-task: findAccidentConditionTask
defgoal find Accident Conditions
Input:
(object EnvironmentalCondition?ec
     (RoadSurface-condition "dry"),
(RoadCondition"straight" || "curve"),   (WeatherCondition
findSevereWeatherTask()) (LightCondition
"artificial"||"nature"))
(object TemporalCondition?tc
     (DateTimeDescription? findWorkingdaysTask())
     (Interval? findRushHoursTask()))
(object (is-a AccidentCondition) (object?ac) (include?ec &
tc))
Output:
(object (is-a $?AccidentCondition) (object?ac))

sub-task: findSevereWeatherTask
defgoal find severe weather conditions
Input:
(object (is-a WeatherCondition) (object?we) include?
"high_wind"||"fog_smog_smoke_dust"||
"hail_sleet"||"raining"||"snow")
Output: (object (is-a WeatherCondition) (object?we))

sub-task: findWorkingdaysTask
defgoal find working days
Input: (object (is-a calendarDay) (object?cd) is-a?weekday
is-not-a?holiday )
Output: (object (is-a calendarDay) (object?cd))

sub-task: findRushHoursTask
defgoal find rush hours
Input: (object (is-a timerange) (object?tr)
equal?TimeofRushHour )
Output: (object (is-a timerange) (object?tr))
```

Figure 3.5 Pseudocode of non-spatial query task findAccidentConditionTask

3.4 Density-based Clustering Algorithm for Traffic Accident Risk (DBCTAR)

The traffic accident risk in this thesis is derived from the accident concentration area. The accumulated accident number is the most common way to reflect the risk level. However, as fatality and injury accidents put more strain on the road network and increase the economic burden on society, these accidents need to be considered differently from PDO accidents, in order to account for their larger effects (Rifaat et al. 2010). Therefore, the risk area should be defined by both the frequency and degree of the severity.

Since the risk areas are arbitrary shapes on the road network, the proposed clustering method is a density-based clustering method for traffic accident risk (DBCTAR). This clustering method is extended from DBSCAN, which is described in Section 2.4 of Chapter 2.

To consider the severity level of each accident, it is proposed to assign different weights to accidents with different severity levels. Within a given accident dataset *D*, a variable *RiskIndex* is defined as follows:

$$RiskIndex = \sum_{i=1}^{n} W_i * Count(S_i) / E \qquad \textbf{1})$$

where S_i is the *i*th severity level, *Count*() is a function to get the total number of accidents at that level, and W_i is the weight assigned to the *i*th severity level. *E* is the exposure coefficient, which should depend on the measure of traffic density. The *Riskindex* not only considers the number of accidents, but also takes into account the severity level. A new parameter *MinRisk*, which is the threshold of *RiskIndex*, is also defined.

Gien a dataset *D*, a symmetric distance function *network_dist*(), parameters *Eps* and *Minpts*, and a threshold *MinRisk*, the following definitions are used to define DBCTAR.

Definition 1 For $p \in D$, The neighbourhood of a point *p*, denoted by $N_{Eps}(p)$, is defined by $N_{Eps}(p) = \{q \in D \mid network\_dist(p,q) \leq Eps\}$.

Definition 2 A point *p* is densidty-reachable from a point *q* with respect to *Eps*, *MinPts* and *MinRisk* if $| N_{Eps}(p) | \geq MinPts$ AND $RiskIndex(p) > MinRisk$.

Definition 3 A point *p* is density-reachable from a point *q* with respect to *Eps* and MinPts if there is a chain of points $p_1,\ldots,p_n$, $p_1=q$, $p_n=p$ such that p_{i+1} is directly density-reachable from p_i.

Definition 4 Let D be a set of accident points. A concentration-based cluster C is a non-empty subset of D satisfying the following condition:

1) $\forall p, q$: if $p \in C$ and *q* is density-reachable from *p* with respect to *Eps*, *MinPts* and *MinRisk*, then $q \in C$;

2) $\forall p, q \in C$: *p* is density-connected to *q* with respect to *Eps*, *MinPts* and *MinRisk*.

With DBCTAR, when the cluster extends an existing cluster from a neighborhood, the neighborhood around a point of a given radius (*Eps*) must contain at least a minimum number of points (*MinPts*) and has a *RiskIndex* larger than *MinRisk.* This algorithm is used in a network environment; therefore, it uses road network distance *network_dist* rather than the Euclidian distance. The core point is an accident that has at least *MinPts* accidents within the search distance *Eps*; and, the *RiskIndex* of the accidents within the search distance is larger than *MinRisk.* This core point criteria can be stated as follows: For $p \in D$, the neighborhood of p is defined as $N_{Eps}(p) = \{q \in D \mid network\_dist(p,q) \leq Eps\}$. If $| N_{Eps}(p) | \geq MinPts$ AND $RiskIndex(p) > MinRisk$, then p is a core point of a cluster.

If *p* is a core point and *q* is *p*'s neighbor, then *q* belongs to this cluster; and, each of *q*'s neighbors is examined. Otherwise, if *p* is not a core point, point *q* is labeled as noise. The algorithm ends when every point is classified as in a cluster or labeled as noise.

Ideally, no two accidents have the same severity level. However, for the practical cases, assigning unique weights to each accident is not feasible. In road safety research, accident records are usually classified into 3 classes: fatality, injury and PDO. Accident with fatalities and/or injuries can be converted into equivalent property damage only (EPDO) accidents (Rifaat et al. 2010).

$$\text{EPDO} = W_1 * Count(\text{Fatal}) + W_2 * Count(\text{Injury}) + W_3 * Count(\text{PDO}) \qquad \textbf{2})$$

EPDO is calculated by assigning different weighting schemes, as shown in Table 3.1. One of the most commonly used conversion weight settings is recommended by PIARC (Permanent International Association of Road Congresses) with W_1=9.5; W_2=3.5; and W_3=1 (Rifaat et al. 2010).

In the previous definition of *Riskindex*, the exposure coefficient *E* is not easy to get. In this research, we assume that all the accidents in the given dataset have the same traffic density. Thus, for the practical cases, *E* is considered as a constant. So, EPDO can be considered as a simplified format of *Riskindex*. When implement the DBCTAR algorithm in the prototype, the *Riskindex* is simplified as follows:

$$RiskIndex = W_1*Count(\text{Fatal}) + W_2*Count(\text{Injury}) + W_3*\ Count(\text{PDO})$$

3)

where W_1, W_2 and W_3 depend on the risk index model that the user selects from Table 3.1. These models use different weighting schemes that reflect different perspectives of the significance of each kind of accident. For example, Transport Canada uses the weight of 13.88 for accidents with injury, which suggests the injury accident is more important than the weight of 3.5 recommended by the PIARC.

Table 3.1 Different weight models for accident severity level

Model	Ratio	Source
1	1:1:1	Simple Total Crash Count
2	9.5:3.5:1	PIARC
3	76.8:8.4:1	North Carolina DOT
4	136.13:4.94:1	Ohio DOT
5	779.9:13.88:1	Transport Canada
6	1300:90:1	Federal Highway Administration

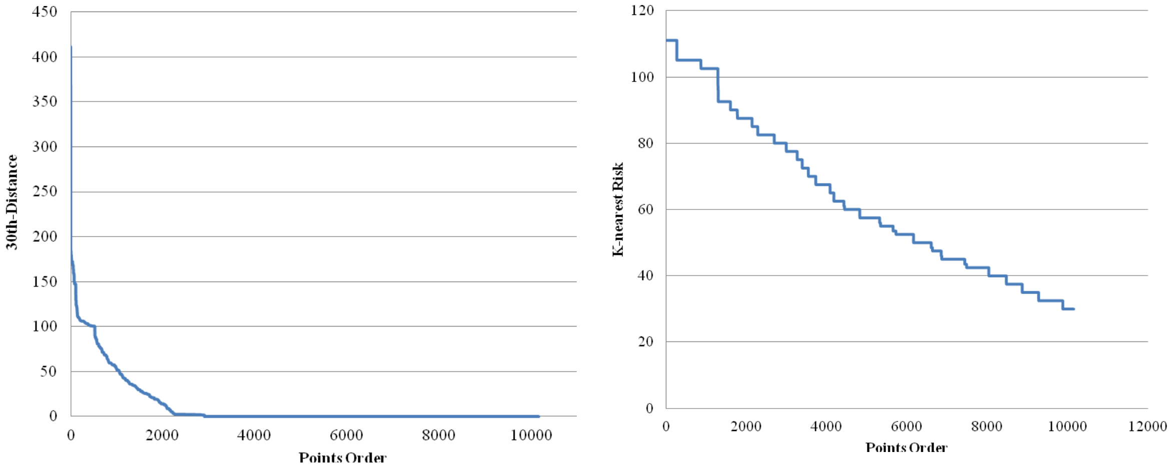
(DOT: Department of Transportation) (Rifaat et al. 2010)

Following is the pseudo code of DBCTAR algorithm:

```
DBCTAR_V1 (SetOfPoints, Eps, MinPts, MinRi)//SetOfPoints is
UNCLASSIFIED
 ClusterId := nextId(NOISE);
 FOR i FROM 1 TO SetOfPoints.size DO
  Point := SetOfPoints.get(i);
  IF Point.ClId = UNCLASSIFIED THEN
   IF ExpandCluster(SetOfPoints, Point, ClusterId, Eps, MinPts,
MinRi) THEN
      ClusterId := nextId(ClusterId)
   END IF
  END IF
 END FOR
END; // DBCTAR
ExpandCluster(SetOfPoints, Point, ClId, Eps, MinPts, MinRi) :
Boolean;
 seeds := SetOfPoints.regionQuery(Point,Eps);
//returns the Eps-neighborhood of Point in SetOfPoints
 IF seeds.size < MinPts OR getRiskIndex(seeds) < MinRi THEN // no
core point
  SetOfPoint.changeClId(Point,NOISE);
  RETURN False;
 ELSE // all points in seeds are density reachable from Point
  SetOfPoints.changeClIds(seeds,ClId);
  seeds.delete(Point);
  WHILE seeds <> Empty DO
   currentP := seeds.first();
   result := SetOfPoints.regionQuery(currentP, Eps);
   IF result.size >= MinPts AND getRiskIndex(result) >= MinRi THEN
      FOR i FROM 1 TO result.size DO
      resultP := result.get(i);
     //SetOfPoints.get(i) returns the i-th element of  SetOfPoints.
      IF resultP.ClId IN {UNCLASSIFIED, NOISE} THEN
            IF resultP.ClId = UNCLASSIFIED THEN
                  seeds.append(resultP);
            END IF;
      SetOfPoints.changeClId(resultP,ClId);
      END IF; //UNCLASSIFIED or NOISE
      END FOR;
   END IF; //result.size >= MinPts
   seeds.delete(currentP);
  END WHILE; //seeds <> Empty
  RETURN True;
 END IF;
END; //ExpandCluster

getRiskIndex (SetofPoints) : Double
 VAR
  c1,c2,c3: INTEGER;
  ri: DOUBLE;
  c1=0;c2=0;c3=0;
  FOR i FROM 1 TO SetofPoints.size DO
      CASE SetofPoints.SeverityLeveal of
            "fatal" : c1++
            "injury" : c2++;
            "property only" : c3++;
      END;
  END;
  ri := getRiskIndexModel(c1,c2,c3);
 //getRiskIndexModel depends on the risk model chosing.
 RETURN ri;
END;// getRiskIndex
```

To determine the parameter *MinRisk*, a method similar to the *k-dist* function (Ester 1996) is adopted. First, the most significant *k-dist* graph is built to identify the most suitable *k* value. Based on our experiment, with the accident dataset, this value could be larger than the normal value. Then calculate the *k*-nearest neighbor's *Riskindex* and sort these values by distance. The threshold *MinRisk* point is located near the first "valley" of the sorted *k*-nearest *Riskindex* graph. The total risk of *eps* distance graph and the real results are also used as the reference.



(a) *k-dist* (*k*=30) graph (b) *k*-nearest *Riskindex* graph (*k*=30)

Figure 3.6 Parameter settings graph for DBCTAR

Figure 3.6 shows the parameter setting graph based on a 10154 accidents dataset. Figure (a) is the *k-dist* graph when *k*=30. From this figure, when the *MinPts* set to 30, the *Eps* is around 100. Figure (b) is the *k*-nearest *Riskindex* graph when *k*=30. The *MinRisk* value is near the first "sharp drop" of the curve, around 90.

Chapter Four: Implementation

4.1 Overview

To demonstrate the ONTO_TARM framework, a system has been developed to generate traffic risk maps based on users' requirements and publish maps on the web. Figure 4.1 shows the structure of the system, which consists of three components, pre-processing, server and client.

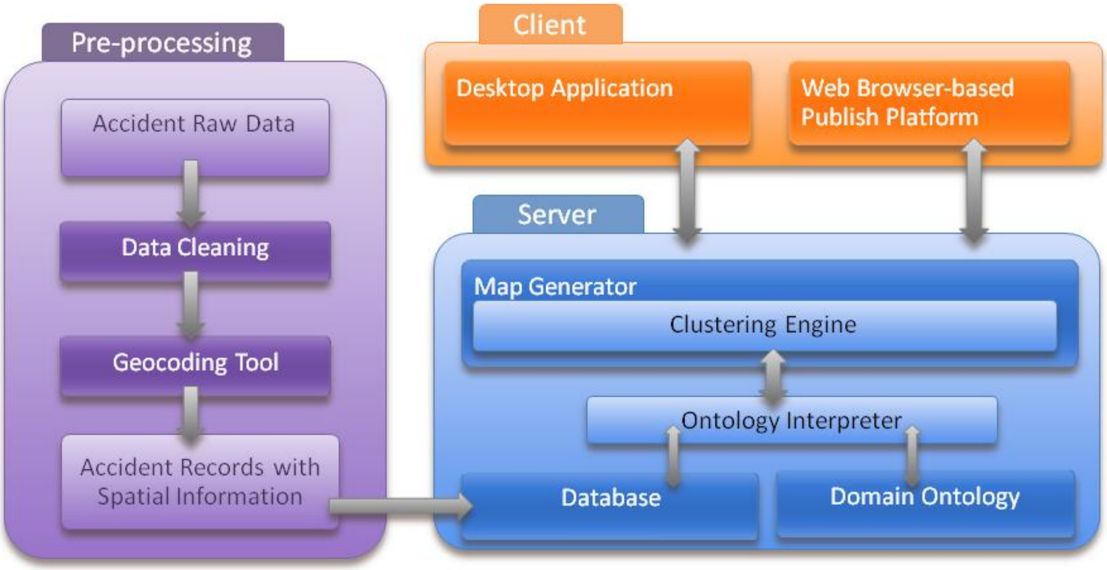


Figure 4.1 Structure of the ONTO_TARM system

The pre-processing component helps the user clean the raw data. It also includes a geocoding tool, which is used to add spatial coordinates for the original traffic accident text records.

The system uses client and server architecture. The client side offers a graphic user interface to help users input different requirements and demands and view traffic accident risk maps in two- or three-dimensional views. Two versions of the interface have been developed: a desktop version and a web-based publishing platform.

The server side of the system handles the dataset selection and generates maps based on users' requirements. It includes five main parts: the database, domain ontology, ontology interpreter, clustering engine and map generator. The database stores and manages the processed traffic accident data with coordinate information. The accident domain ontology is represented by Protégé-OWL (an ontology editor and knowledge-base framework with OWL). To enable ontology reasoning for the dataset, the ontology interpreter utilizes the Protégé-OWL reasoner to communicate with database.

The clustering engine has been designed as a new web-based clustering platform to improve the limitations of current online clustering tools, as discussed in Chapter 2. In the current version, it implements a simplified version of the density-based clustering algorithm for traffic accident risk (DBCTAR) and is used to find the traffic concentration areas based on users' goals with consideration of both the density and severity levels of accidents. The map generator transfers the clustering result from point sets into risk maps.

Details of each system component and implementations are discussed in the following sections.

4.2 Geocoding Tool

Geocoding is a process that finds associated geographic coordinates for the accident locations from the accident addresses. In general, the locations of the traffic accident records are described by the intersection of the streets or a street address, instead of by coordinates. For example, "1st St. & 5th Ave SE" represents the location of an accident in the intersection of 1st Street and 5th Avenue in the southeast of the city; and, “333 2nd Ave SW” describes an accident location as a street address. Since the clustering engine cannot directly handle street addresses in the algorithm, geocoding is needed to determine the latitude and longitude coordinates of the accident records.

In the ONTO_TARM system, the geocoding tool, called geocoding service, is offered by Google Maps JavaScript API V3 Services. A comparison of the geocoding services among Bing Maps Geocode Service, Bing Spatial Data Services - Geocode Dataflow API and ESRI Geocoding Engine led to the conclusion that Google Maps V3 Service has better accuracy than other services for our sample dataset. This tool uses JavaScript to run the batch process for the comma-separated values (CSV) text files. Each accident record has a unique identification and formatted text location description, which are stored in a line of the CSV file. The tool adds the latitude and longitude coordinates for each accident record of CSV files.

In this thesis, the maximum accuracy of the geocoding result is at the street level as the location description is based on the street level address. This research assumes the geocoding result at the street level represents the real accident distribution. Sometimes the locations of the accidents may not reflect the true accident location due to inaccuracy or input errors of the street address records. In these cases, the accuracy of geocoding is at the city level, and the locations of accidents may be geocoded as the centre of the city. These records are excluded from the research dataset.

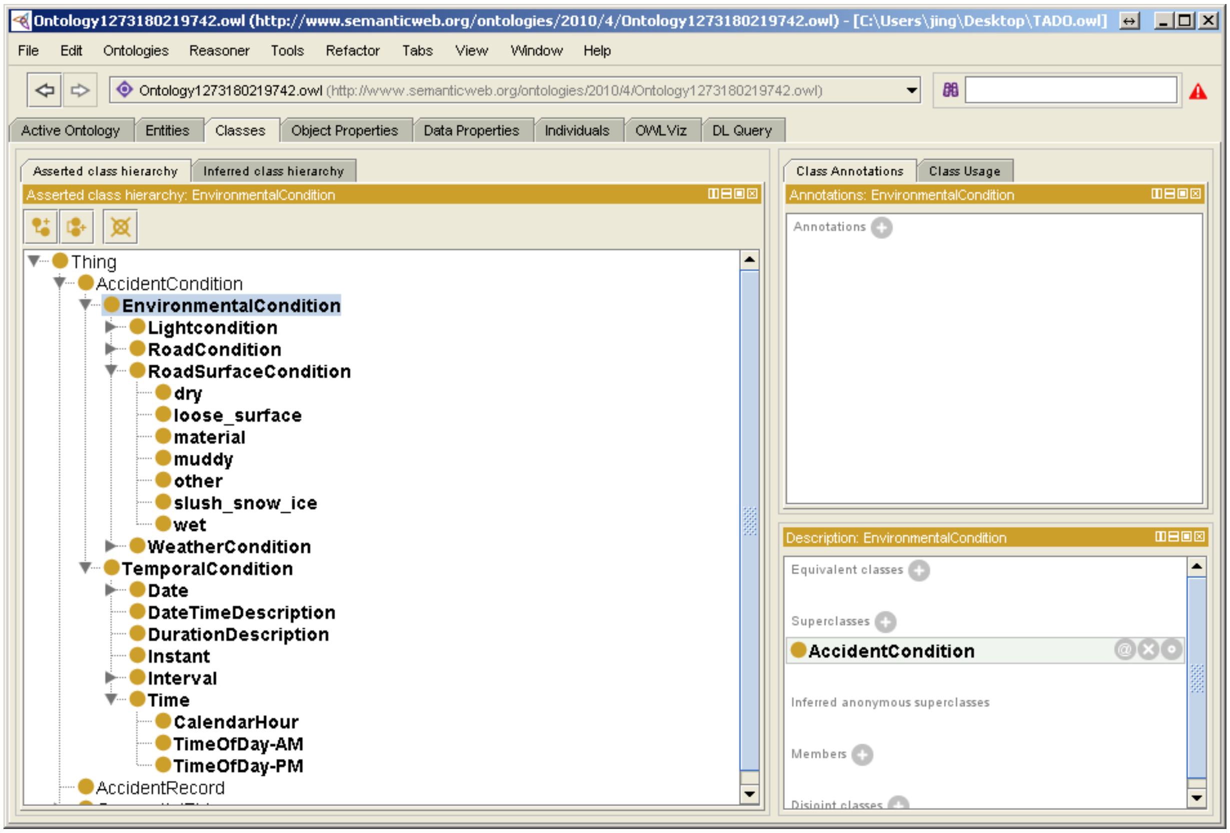
4.3 Server-side Components and Implementation

4.3.1 Database, Ontology and Ontology Interpreter

The accident database, domain ontology and ontology interpreter work jointly to host the geocoded accident dataset and enable the ontology reasoning for the dataset. The database stores and manages the accident data and is the data source for the ontology interpreter, which works as middleware between the clustering engine and the database to perform the ontology query. The main component of the ontology interpreter is derived from the ontology reasoner, which uses the accident domain ontology as the knowledge source for the reasoning. .

For the database, the ONTO_TARM system uses ESRI's File Geodatabase (ESRI 2009) to host the accident dataset. For the reasoning part, the accident domain ontology is represented with Protégé-OWL 4.0 software (Protégé 2009) and saved in one OWL file. OWL is one of the most widely used ontology languages. It is characterized by formal semantics and RDF (Resource Description Framework) / XML (Extendible Markup Language) based serializations for the semantic web. The traffic accident domain ontology (TADO) with Protégé-OWL is shown in Figure 4.2.

The Ontology Interpreter is implemented by a customized Jena reasoner with the Jena API (Jena 2010). Jena API is a Java API for RDF. In the current version, Jena works on the level of RDFS (Resource Description Framework Schemas). The ontology query works on top of the abstract class to returns an XML file, which contains the query parameters to perform a manually selection on the file goedatabase via ArcCatalog (ESRI 2009) and can be saved into simple FeatureClasses. The clustering engine actually works on the saved simple FeatureClasses via a client application.


Ontology1273180219742.owl (http://www.semanticweb.org/ontologies/2010/4/Ontology1273180219742.owl) - [C:\Users\jing\Desktop\TADO.owl]
File
Edit
Ontologies
Reasoner
Tools
Refactor
Tabs
View
Window
Help
Ontology1273180219742.owl (http://www.semanticweb.org/ontologies/2010/4/Ontology1273180219742.owl)
Active Ontology
Entities
Classes
Object Properties
Data Properties
Individuals
OWLViz
DL Query
Asserted class hierarchy
Inferred class hierarchy
Asserted class hierarchy: EnvironmentalCondition
Class Annotations
Class Usage
Annotations: EnvironmentalCondition
Annotations
Description: EnvironmentalCondition
Equivalent classes
Superclasses
Inferred anonymous superclasses
Members
Disjoint classes

```
Thing
  AccidentCondition
    EnvironmentalCondition
      Lightcondition
      RoadCondition
      RoadSurfaceCondition
        dry
        loose_surface
        material
        muddy
        other
        slush_snow_ice
        wet
      WeatherCondition
    TemporalCondition
      Date
      DateTimeDescription
      DurationDescription
      Instant
      Interval
      Time
        CalendarHour
        TimeOfDay-AM
        TimeOfDay-PM
  AccidentRecord
```

Superclasses: AccidentCondition

Figure 4.2 TADO in Prot ég éOWL

4.3.2 Clustering Engine

The clustering engine is the core component of the ONTO_TARM system. It is designed as an independent and generic Web-based clustering service called GeoClustering following the concept of Web Service. Web Service represents a convergence of the service oriented architecture (SOA) and the Web (W3C 2004). The web server is responsible for communications with client applications and response for the clustering tasks from the client. The goal of GeoClustering is not only to build a component of the ONTO_TARM system, but also a generic clustering tool to improve the limitations of current geospatial clustering tools.

The current beta version is available at http://www.geoclustering.net. A user guide is provided at http://www.geoclustering.net/help.php. Current version only considers the clustering objects as points. It can perform density-based clustering and visualize the results. The proposed spatial clustering algorithm DBCTAR and classical spatial clustering algorithm DBSCAN have been implemented. The distance function between two objects is defined as the shortest geometric distance on the spherical Earth surface in current version. The architecture of GeoClustering is shown in Figure 4.3. It adopts the client and server architecture. Details of the data format, server side and client side are illustrated as below.

To achieve better interoperability, GeoClustering uses XML as the data transfer format. The server can fetch the XML files directly from other websites and return the clustering results in the same format. Further discussion concerning the data format of user input and system output is discussed in 4.3.2.1. The clustering engine offers a friendly web page interface for general users. Users could submit clustering requests via browser or other applications. The APIs are in the form of an endpoint URL address. It will be the lowest level of the interface for using the clustering service and it is developed around the ‘reuse’ design paradigm. With this API, other web applications or services can call the geospatial clustering service directly. The client component will be discussed in detail in Section 4.3.2.2. The server will perform the clustering procedure after receiving the GeoClustering API request and parameters and then give a returned result. This will be discussed in 4.3.2.3.

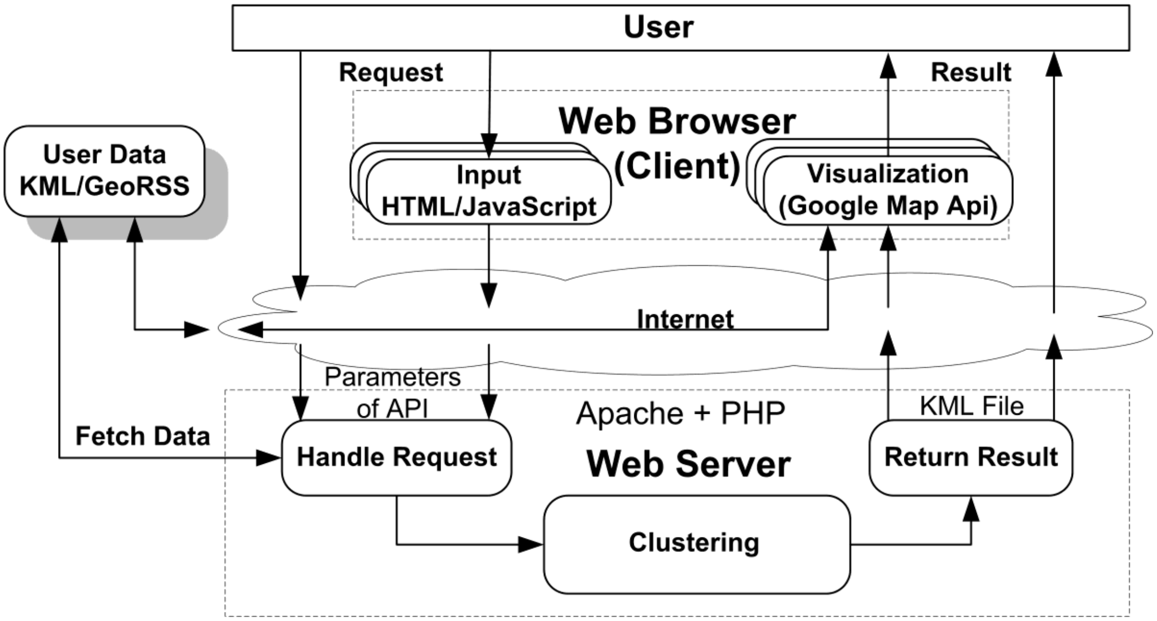


Figure 4.3 GeoClustering architecture

4.3.2.1 Data formats

In order to enable interoperability between heterogeneous data sources on the Internet, international organizations such as the World Wide Web Consortium (W3C), the International Organization for Standardization (ISO) and Open Geospatial Consortium (OGC), have been making significant efforts to define data exchange standards and protocols. With the proliferation and implementation of various ISO and OGC standards, spatial data in the form of standard XML (extensible markup language) format has become more and more popular.

To meet the requirement for interoperability, the standard XML file format is used in the data transaction in the GeoClustering platform. The system accepts two kinds of standard XML files from users: KML (Keyhole Markup Language) and GeoRSS. The clustering results are returned in the form of KML.

KML is a tag-based structure with nested elements and attributes and is based on the XML standard. It is used for expressing geographic annotation and visualization on existing or future Web-based, two-dimensional maps and three-dimensional Earth browsers (OGC 2008). KML v2.2 was adopted as an official OGC implementation standard on April 14, 2008.

Below is part of an example of KML input accident data file for GeoClustering. Each accident is represented as a point which under the `<Placemark>` node. This sample file contains an accident record with the name of 'Accident0001'. The time property of the accident is under the `<TimeStamp>` node. The geometry property of the accident is described in the `<coordinates>` element under the `<Point>` node. The geographic location is defined by longitude, latitude, and latitude. Usually each point must contain longitude and latitude value, while latitude is optional.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.2">
  <Document>
    <name>Accident record clustering input example</name>
    <Placemark>
      <name> Accident0001</name>
      <description>Attribute1:3</description>
      <TimeStamp>
      <when>2008-09-22T09:00:01-07:00</when>
      </TimeStamp>
      <Point>
        <coordinates>-114.132446,51.079529,0</coordinates>
      </Point>
     </Placemark>
      …
</Document>
```

```
</kml>
```

GeoRSS means geographically encoded objects for RSS (really simple syndication) feeds. It is based on RSS 2.0 and adds location information to data items (OGC 2006). An example of a GeoRSS input file, which includes the same point in the example KML file, is shown below. The name of that point is contained by `<title>`. Time property is described in `<description>`. The geographic location is defined by `<geo:lat>` and `<geo:long>` element.

```
<?xml version="1.0"?>
  <rss version="2.0"
  xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
  xmlns:dc="http://purl.org/dc/elements/1.1">
<channel>
  <item>
        <title>Accident0001</title>
        <description>2008-09-22T09:00:01-07:00,3</description>
        <geo:lat>51.079529</geo:lat>
        <geo:long>-114.132446</geo:long>
  </item>
</channel>
</rss>
```

4.3.2.2 GeoClustering client

In the GeoClustering, GeoClustering API is offered in the form of an endpoint URL address. Users can access the clustering service with the assigned clustering parameter values through the URL. HTTP GET or POST actions are the only supported request formats. The API endpoint URL is as follows:

```
http://localhost/cluster.php?<Data URL>&[File Type]&<Algorithm
Name>&<Clustering Parameters>
```

To request the clustering service, invoke as follows:

```
http://localhost/cluster.php?url=http://earthquake.usgs.gov/eqcenter
/catalogs/shakerss.xml&filetype=georss&algorithm=DBCTAR&param1=9&par
am2=0.3&param3=12
```

```
Data URL :: = "url =" <the url of remote point data file>

File Type :: = "filetype =" <kml | georss>

Algorithm Name :: = "algorithm =" <algorithm name: DBCTAR | DBSCAN |
| …>
Clustering Parameters :: = {"param"<number>"=" <the value of

parameter for the chosen algorithm>}
```

The algorithm name could be any implemented algorithms in GeoClustering. The following "`param`" is the corresponding parameters for the algorithm. In the example above, when `algorithm=DBCTAR`, then `param1` is the minimum number of points (*MinPts*), `param2` is the radius (*Eps*) in km, and `param3` is the minimum risk index (*MinRisk*).

To respond to the API requests, an XML response is returned. When unusual situations occur during the clustering procedure, two types of error message are sent back.

`<b>Parameters missing</b>` or

`<b>Data cannot be found at URL or data syntax error</b>`

When the correct clustering result is achieved through the clustering procedure, the response is presented in KML format.

```
  <?xml version="1.0" encoding="UTF-8"?>
  <kml xmlns="http://earth.google.com/kml/2.2">
    <Document>
      <name>GeoClustering Result</name>
      <Snippet> </Snippet>
      <description>
            Total Clusters:4 <br />   <!-- Cluster numbers -->
      Time:<b>0.191354036331</b> Seconds<br /> <!-- Run time -->
      </description>
<!-- Style begin -->
  …
    <StyleMap id="clustericon1">
      …
    </StyleMap>
    …
<!-- Style end -->
<!-- Accidents info begin -->
    <Folder>
            <Placemark>
        <name>Cluster 1</name>
        <TimeStamp><when></when></TimeStamp>
        <styleUrl>#clustericon1</styleUrl>
        <description>Accident0001 2008-09-22T09:00:01-07:00
        </description>
        <point>
          <coordinates>-114.132446,51.079529,0</coordinates>
        </Point>
      </Placemark>
      <Placemark>
        <name>Cluster 1</name>
        …
      </Placemark>
      <Placemark>
        <name>Cluster 2</name>
        …
    </Folder>
 <!-- Accidents info end -->
  </Document>
</kml>
```

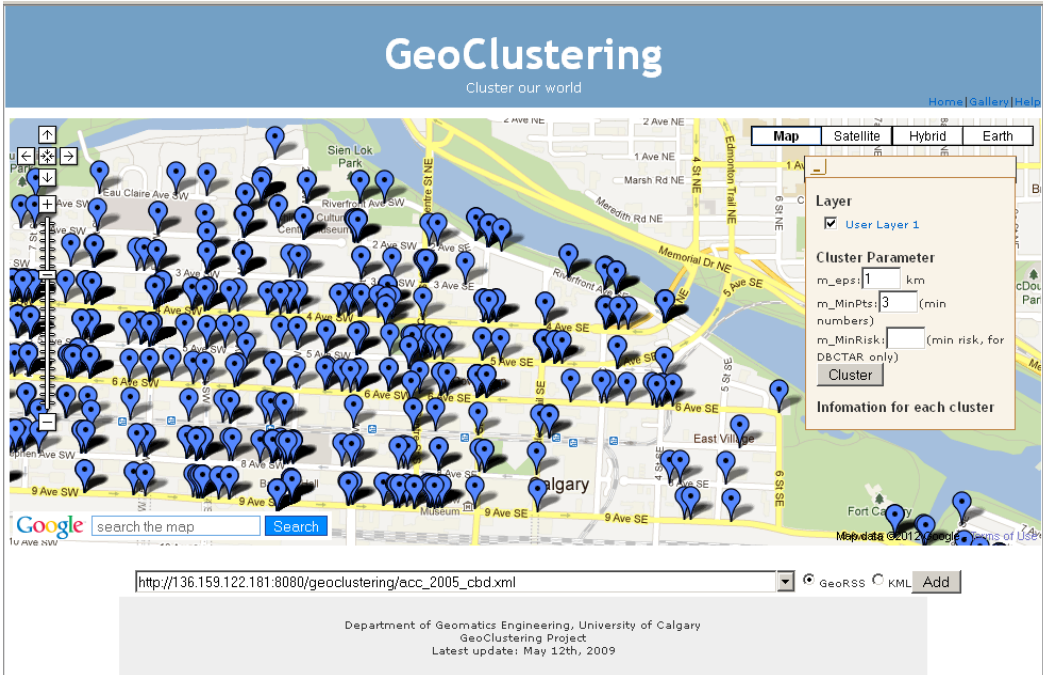
In the returned KML file, general clustering result information is included in element `<description> under <Document>`. Number of clusters established and running time are recorded here. The following part is the style information, which defines the icon style for the point in each cluster. When one point needs to use this style, it cites the id of `<StyleMap>`. All the points in the same cluster have the same cluster id and the same value under `<name>` and `<styleUrl>` tags. The original value under `<name>` and `<TimeStamp>` of each accident will be added into that point's `<description>` element. In the returned example file, the original point with name "`Accident0001`" becomes the point with the name "`Cluster 1`" and in the `<description>` part of this point the original information "`Accident0001 2008-09-22T09:00:01-07:00`" is added.

When there is no cluster found in the dataset following the clustering procedure, an alert response message is returned.

```
  <b>Internal error or no Cluster found in the dataset</b>
```

Besides the API, GeoClustering also offers a web page as GUI. The whole web page is implemented by HTML, CSS and JavaScript. To have a better user experience, AJAX (Asynchronous JavaScript and XML) have been used. The third party Web Map APIs are included in the web page. Microsoft Bing Maps and Google Maps are the two major players in the area of Web Maps. Comparing Google Maps API with Bing Maps

API, we choose Google Maps as it has a better 3D extension with the Google Earth Plug-in. Figure 4.4 shows the web page interface of GeoClustering.



Layer

☑ User Layer 1

Cluster Parameter

m_eps: 1 km

m_MinPts: 3 (min numbers)

m_MinRisk: (min risk, for DBCTAR only)

Cluster

Infomation for each cluster

http://136.159.122.181:8080/geoclustering/acc_2005_cbd.xml ● GeoRSS ○ KML Add

Department of Geomatics Engineering, University of Calgary
GeoClustering Project
Latest update: May 12th, 2009

Figure 4.4 GeoClustering web interface

4.3.2.3 GeoClustering server

The web server takes the role of communicating with the client and responding to the clustering task requested from the client. Among various web servers and corresponding server script languages available, Apache and PHP is selected as the implementation option because they are open source software, easy to deploy, platform independent, reliable and secure.

Figure 4.5 shows the work flow at the server side. When the server receives a request, it first verifies whether the request is complete or not. A complete request contains at least three essential parts: `Data URL, Algorithm Name and Clustering`

`Parameters. File Type` is optional. Failure to include any of the three essential parts leads to failure of the verification.

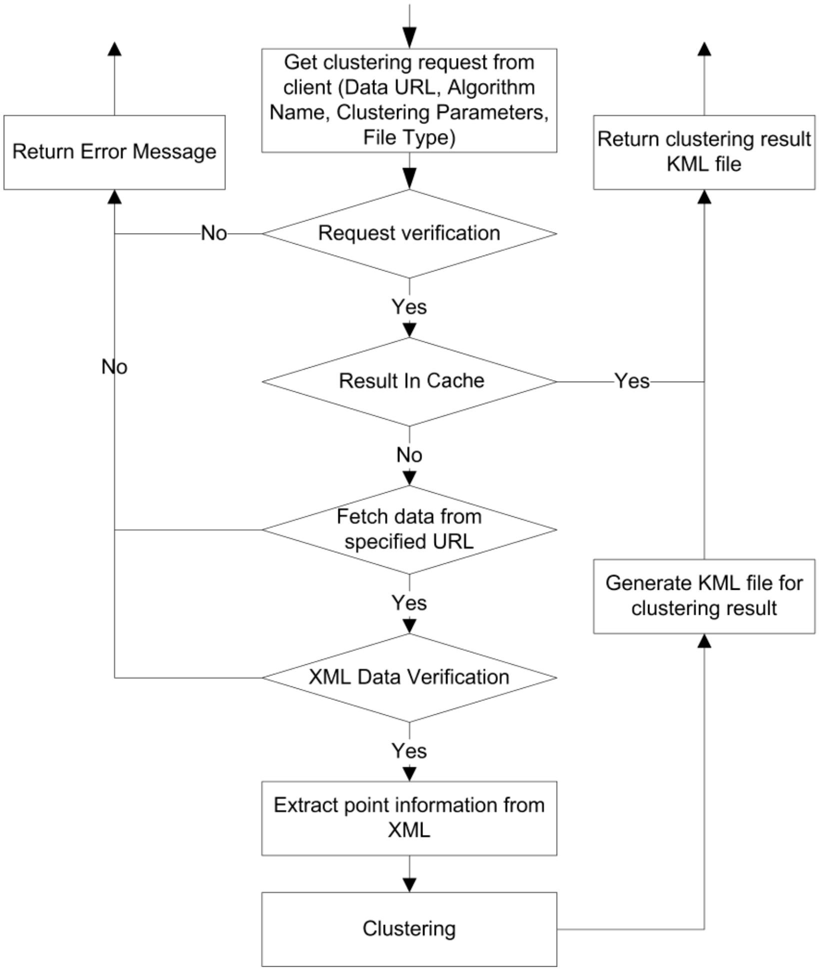

Get clustering request from client (Data URL, Algorithm Name, Clustering Parameters, File Type)
Return Error Message
Return clustering result KML file
Request verification
No
Yes
Result In Cache
Yes
No
No
Fetch data from specified URL
Yes
Generate KML file for clustering result
XML Data Verification
Yes
Extract point information from XML
Clustering

Figure 4.5 Server-side work flow

After the verification, the server calls for the function that detects whether there is a target xml results file in the “Cache”. Since clustering sometimes is time-consuming, a buffer system is implemented to save the clustering result for a short period of time. If the same clustering requests are sent the second time, the result can be quickly returned to client. The buffer can speed up the GeoClustering performance, especially when users want to compare the result with previous requests.

If there is no clustering result in the "Cache", the server will fetch the remote XML file (the user data) through the `Data URL` submitted. If the remote file is unreachable, an error message will be returned. If the server receives the remote data successfully, then the server will verify the completion of the XML files. Any incomplete or incorrect syntax will cause abnormal termination and an error message will be returned. Then, the server extracts all of the points' information from the XML files fetched in the last step and pass the points to the clustering algorithm module in the form of an array. Next, the algorithm module uses the algorithm specified by the request, together with clustering parameters, to conduct the clustering. A new array with the cluster information is then returned to the next module. Finally, in the next module, the server generates KML files from the returned array.

4.3.3 Map Generator

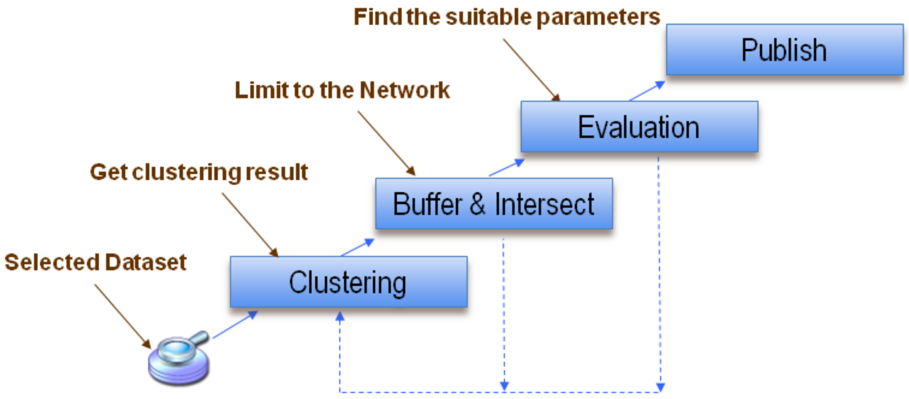


Figure 4.6 Workflow of the map generator

The map generator transforms the clusters into traffic accident concentration areas with different colors for better visualization. Figure 4.6 shows the process of changing the clusters into maps.

The clustering result consists of a few sets of points. Each set represents one cluster. The next step is the transfer of these point sets into an area. One of the traditional methods is the generation of the convex hull of the cluster. However, the spatial information of most accidents is based on geocoding results. Therefore, multiple points may overlap. In this case, the convex hull becomes a point. In this system, the map generator uses buffering and intersection operations to solve the problem. A dissolved buffer is generated for all the points in one cluster. The buffer result is then intersected and clipped by the road network polygons. The newly generated polygons have the same risk index value forms as the original clustering result. All the polygons are represented by gradient colours based on their values.

4.4 Client-Side Components and Implementation

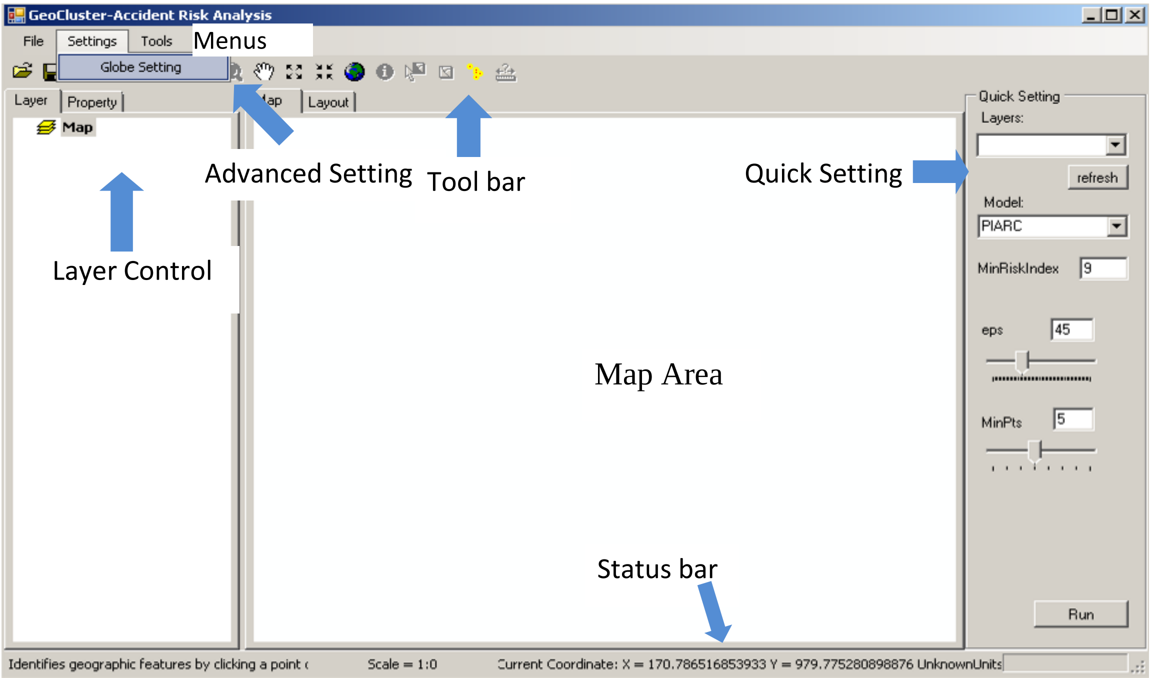
The desktop version of the client-side of the system also has a graphical user interface implemented using C# with ArcGIS Engine 9.3 (ESRI 2009), as shown in Figure 4.7. The menus and toolbar are located at the top of the interface. Each toolbar button corresponds to a function, such as open project document, save, add layer, pan, zoom, etc. The menus offer more options, such as the global setting, which leads the user to an advanced settings form, as shown in Figure 4.8.

Under the toolbar, there are three components, which from the left side to the right side are layer control, map area, quick setting panel. Layer control allows the user to control the visibility of various layers on the map. Map area enables the user to interact with the map and reach the final result. The quick setting panel lets the user set the clustering parameters. The status bar, which shows the coordinates of the cursor and the current system message, is located at the bottom of the interface.

In the “Advanced Setting Window” shown in Figure 4.8, all the environmental settings such as the location of the configuration file, default folder to save the temp file, etc., can be changed by the user.

The traffic concentration map can be exported as a KML file. An online platform based on Google Maps with a three-dimensional (3D) viewer is also implemented with Apache 4 to publish the KML file. The screenshot of the publishing platform is shown in Figure 4.9(a). Users can also manually export the KML file and load it to the 3D map viewer, as shown in Figure 4.9(b).

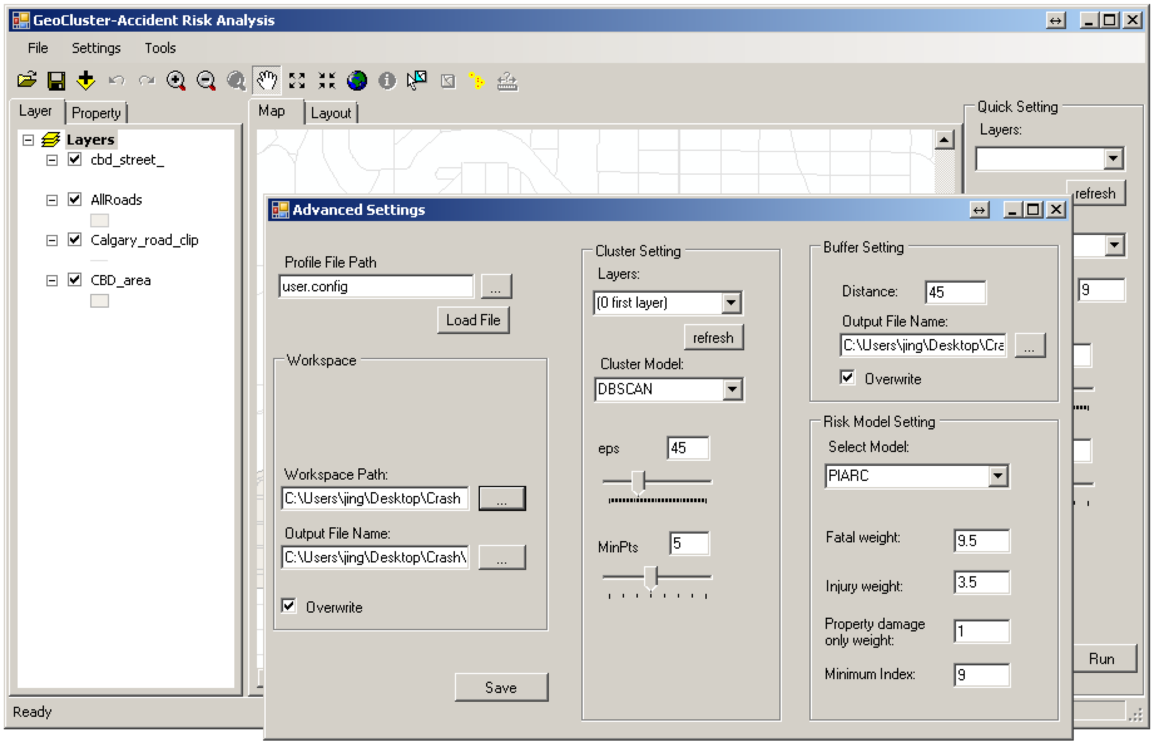
Case studies in the City of Calgary are conducted to test this ONTO_TARM system with real datasets. Details of the case studies are discussed in Chapter Five.



Quick Setting
Layers:
Model: PIARC
MinRiskIndex 9
eps 45
MinPts 5

Identifies geographic features by clicking a point (Scale = 1:0 Current Coordinate: X = 170.786516853933 Y = 979.775280898876 UnknownUnits

Figure 4.7 Main interface of system



Layers
☑ cbd_street_
☑ AllRoads
☑ Calgary_road_clip
☑ CBD_area

Quick Setting
Layers:
9

Profile File Path
user.config

Workspace
Workspace Path: C:\Users\jing\Desktop\Crash
Output File Name: C:\Users\jing\Desktop\Crash\
☑ Overwrite

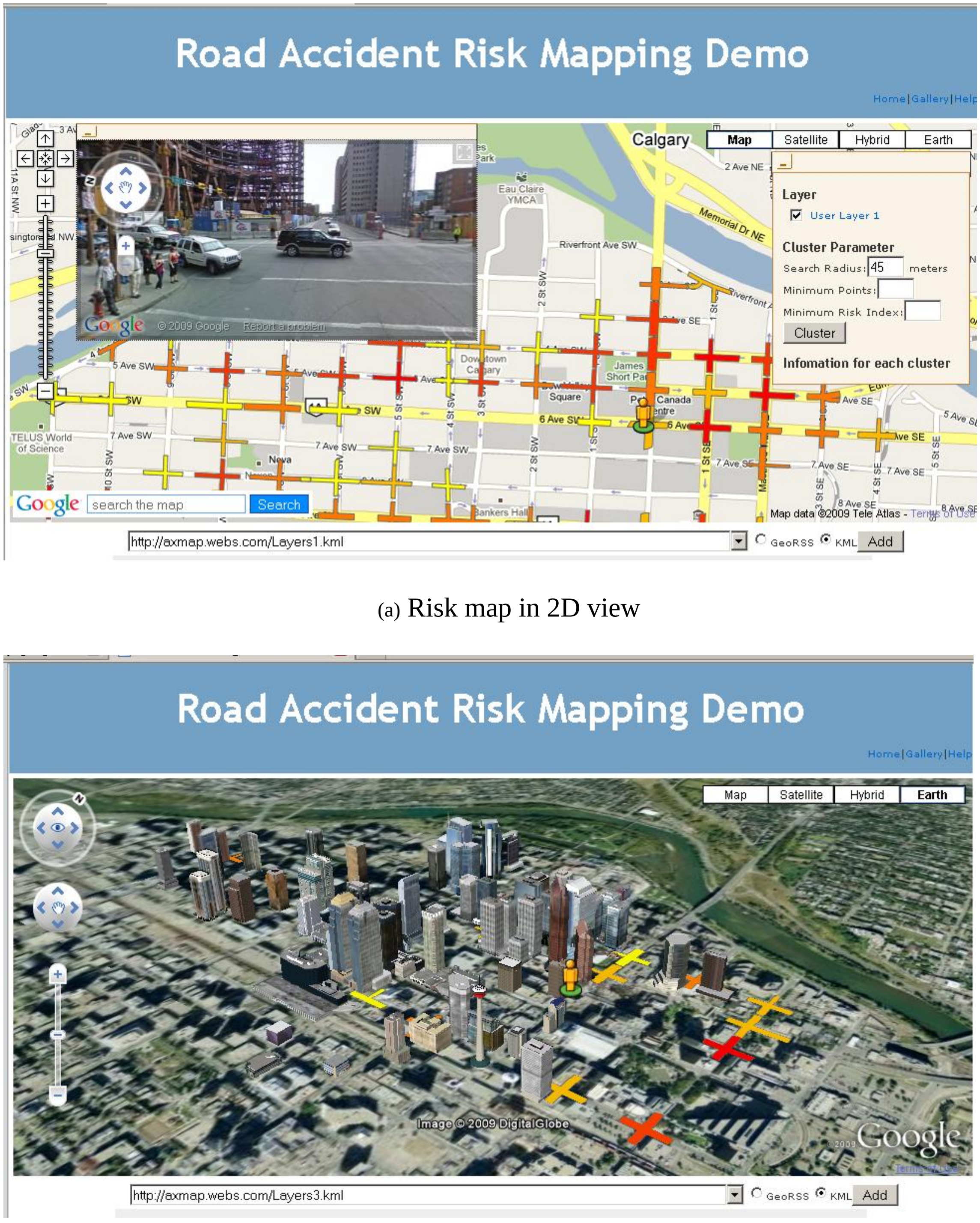
Cluster Setting
Layers: (0 first layer)
Cluster Model: DBSCAN
eps 45
MinPts 5

Buffer Setting
Distance: 45
Output File Name: C:\Users\jing\Desktop\Cra
☑ Overwrite

Risk Model Setting
Select Model: PIARC
Fatal weight: 9.5
Injury weight: 3.5
Property damage only weight: 1
Minimum Index: 9

Ready

Figure 4.8 Advanced settings of the system



Road Accident Risk Mapping Demo

Home|Gallery|Help

Layer

☑ User Layer 1

Cluster Parameter

Search Radius: 45 meters

Minimum Points:

Minimum Risk Index:

Cluster

Infomation for each cluster

http://axmap.webs.com/Layers1.kml ○ GeoRSS ● KML Add

(a) Risk map in 2D view

Road Accident Risk Mapping Demo

Home|Gallery|Help

http://axmap.webs.com/Layers3.kml ○ GeoRSS ● KML Add

(b) Risk map in 3D view

Figure 4.9 Road accident risk mapping web publishing platform

Chapter Five: Case Studies

In this chapter, four case studies with risk maps generated by the density-based clustering algorithm for traffic accident risk (DBCTAR) method are presented to illustrate the application of the proposed ONTO_TARM framework and the function of the developed ONTO_TARM system.

The first case study demonstrates the ontology reasoning and a comparison between the risk map result and one generated by using kernel density method, one of the widely used geography spatial analysis methods. In addition, this case also demonstrates how to use the clustering engine, GeoClustering Web service, as an independent platform. The second case study compares risk maps in the same area under different conditions, thereby proving that, even within the same area, different environmental factors may lead to diversified results. The third case study demonstrates a risk map for a specific road only, instead of a whole area, with a comparison between the result from the DBCTAR and that from a simple count number method. The fourth case study illustrates potential integration with other systems.

5.1 Study Area and Data Description

The main study area is in the city of Calgary, located in the southern part of Alberta, Canada. As of 2011, the City of Calgary is the third-largest municipality area in Canada. Based on the tested dataset, around 35,000 valid traffic accidents are recorded within the city boundary each year.

The data used for the case studies includes traffic accident data, road network data and basic geographic data. The traffic accident testing data was extracted from the

Alberta Collision Database, which is owned by the Government of Alberta’s Ministry of Transportation. It includes all the reported collisions on the roads within the province of Alberta from 1999 to 2005. The total number of records is more than 770,000. Each record has 56 properties in the case table and 33 properties in the object table. The case table has the general description of the collision, including case number, date, time, location, severity level, road class, road alignment, weather conditions, and total vehicles. The object table has all the vehicle related information, such as driver’s age, sex, vehicle condition. We assume that this dataset may not include some non-reported accidents but can represent the real accident situation and the injuries or fatalities are not caused by their own health issue.

The traffic accident testing data are cleaned and geocoded before being used in the case studies. We assume that the geocoding result can represent the location of real accidents; all the accidents are happened on the road - not include the accidents happened in the parking lot. The data from 1999 to 2003 is used to generate traffic accident risk maps, and the data from 2004 and 2005 is used for validation. We assume that during this time period the traffic facilities in the research area have not been improved.

The road network data are extracted from the data media offered by ESRI – North American Street Map. The basic geographic data, including administrative boundaries, land cover and hydro network, are downloaded from GeoBase (GeoBase 2009). The community boundary dataset is derived from the Canada Census Dataset of 2006.

5.2 Case One

In the first case study, the user’s goal is to generate a risk map of the downtown area of Calgary during morning rush hours. This task refers to the downtown area of Calgary. Without geographic knowledge of Calgary or a definition of rush hours, the traditional method cannot proceed, due to the lack of domain ontology. However, ONTO_TARM can generate the task based on the user’s goal and perform the spatial reasoning. The spatial query task is shown in Figure 3.4. In the ontology, Calgary is an instance of the *City* class; and, all census units – communities in Calgary – are represented as instances of the *Community* class. The downtown area is an instance of *CitySection.* This task finds the communities inside Calgary that belong to the downtown area, returning with five communities: Eau Claire, Chinatown, Downtown west end, Downtown east village and Downtown commercial core.

As the second step, non-spatial reasoning is generated to filter the dataset. The non-spatial task is similar to the task shown in Figure 3.5 and is represented in Figure 5.1.

```
sub-task: findAccidentConditionTask
defgoal find Accident Conditions
Input:
(object EnvironmentalCondition?ec
  (RoadSurface-condition "dry"), (RoadCondition "straight"
|| "curve"),
  (WeatherCondition
"clear"||"high_wind"||"fog_smog_smoke_dust"||"hail_sleet"||
"raining"||"snow"||"other_weather_con")
  (LightCondition "artificial"||"nature"))
(object TemporalCondition?tc(Interval?
findRushHoursTask()), Interval? findMorningTask())
(object (is-a AccidentCondition) (object?ac) (include?ec &
tc))
Output:
(object (is-a $?AccidentCondition) (object?ac))

sub-task: findRushHoursTask
defgoal find rush hours
```

```
Input: (object (is-a timerange) (object?tr)
equal?TimeofRushHour)
Output: (object (is-a timerange) (object?tr))

sub-task: findMorningTask
defgoal find rush hours
Input: (object (is-a timerange) (object?tr)
equal?TimeofMorning)
Output: (object (is-a timerange) (object?tr))
```

Figure 5.1 Pseudocode of non-spatial query task findAccidentConditionTask

The temporal condition tasks include two subtasks: finding the rush hours and the morning duration. The final dataset based on the ontology-based query includes 869 records. The DBCTAR then identifies clusters of this dataset. Maps derived from the clustering results are generated by the map generator.

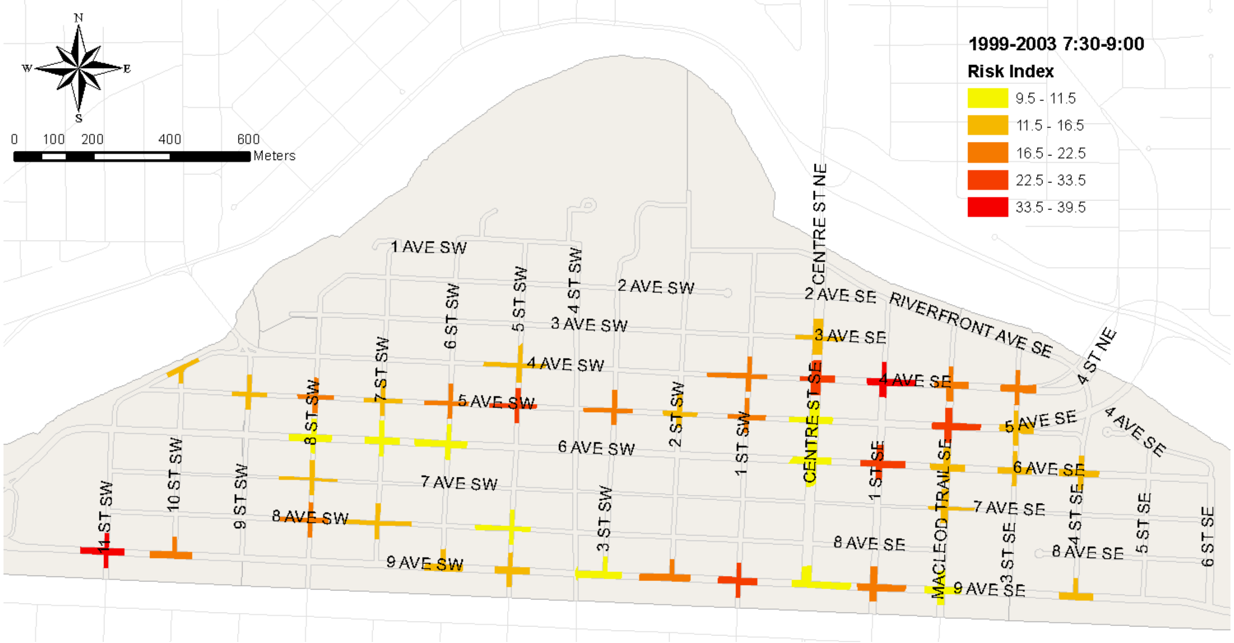


Figure 5.2 Result from the map generator – risk map of morning rush hour (7:30-9:00AM) of Calgary downtown area

Figure 5.2 shows one of the traffic accident risk maps. This map was generated with the risk model parameters recommended by PIARC. The parameter *MinRisk* was set to 10, *Eps* was set to 45 metres, and *MinPts* was set to 5. There are 43 intersections marked as risk areas on the map. The risk indexes for each area range from 10 to 43. The area with highest risk index is located at 4 Ave SE crossing ^{1St} St. SE. The validation with the 2004 dataset shows that 54.8% of the accidents were located in the risk area; 2005 dataset shows that 56.0% of accidents were located in the risk area.

A comparison was conducted between the risk maps of the same downtown area generated by the DBCTAR method and the traditional kernel density method. Both methods use the same dataset satisfying the user specified spatial and temporal conditions. shows the density estimation result when the radius was set to 45 metres and the cell size was 4.08 square metres. The kernel function was based on the quadratic kernel function described in Silverman (1986). The final map was derived from the normalized kernel density result ranging from 0 to 1. Since the risk index value in the DBCTAR map ranges from 10 to 43, it can be projected to the range from 0.232 to 1. Therefore, for the kernel density map, only the raster cells with a value larger than the corresponding minimum risk value of 0.232 have been extracted.

On the kernel density result map, 46 road intersections or segments are identified that had enough density to be marked as risk areas. The highest density area was also located at Macleod Trail SE crossing 5th Ave SE. When compared with the risk areas identified by the DBCTAR, the two maps are consistent; and, most of the risk areas are the same. The two maps have 41 intersections in common. The number of intersections or segments that were only identified by the kernel method was 5, which are marked by the blue circles in . There are 2 risk areas that only appear on the map generated by the DBCTAR, marked by the red squares.

Both maps have some imperfections since we only have two year data for the evaluation. At certain intersections, marked by the triangle symbols, where both methods indicated intersections with high density or risk, there were no accidents in these areas in 2004 and 2005.

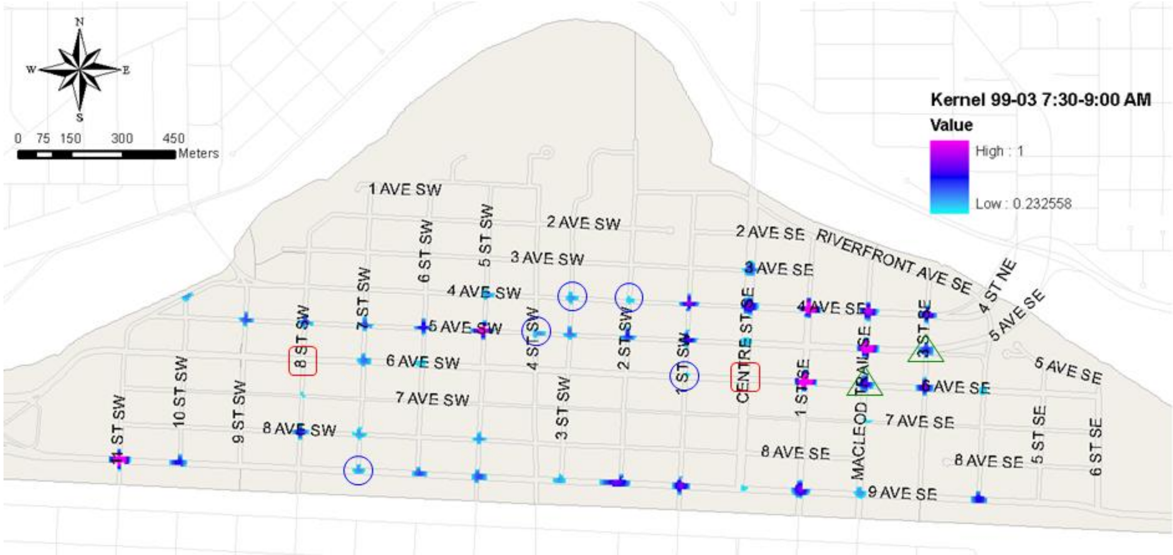
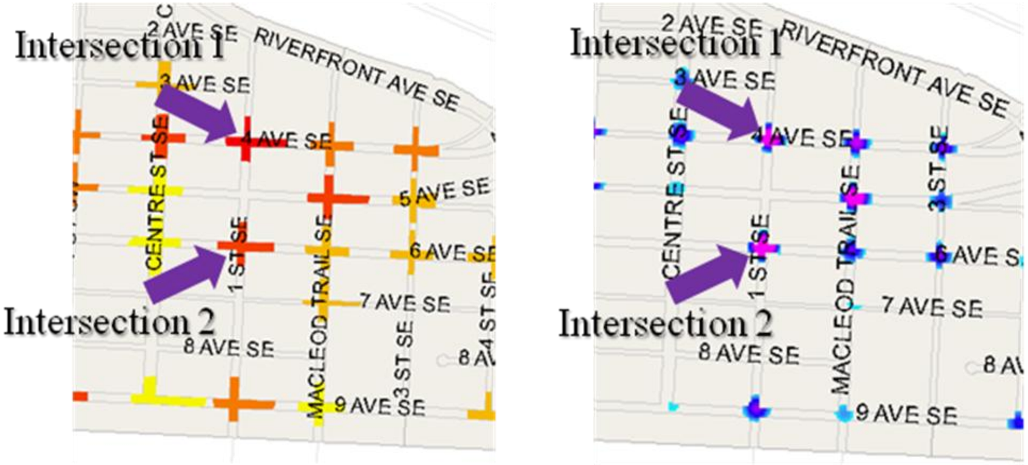


Figure 5.3 Comparison between the kernel density method and DBCTAR methods



(a)Kernel density result (b) DBCTAR result

Figure 5.4 Comparison of two intersections with the kernel density method (a) and the DBCTAR method (b) in the downtown area, as indicated by the arrows

If we take a closer look at some common intersections identified by both maps, the DBCTAR method is more suitable for determining the accident risk at certain intersections. Figure 5.4 (a) and (b) are zoomed-in maps of the same downtown area produced by the kernel density and DBCTAR methods, respectively. One intersection (shown by the up arrow) is the intersection of 4th Ave SE and 1th St SE; and, the second intersection (shown by the down arrow) is the intersection of 6th Ave SE and 1th St SE. According to the 1999-2003 dataset, there were 23 accidents (16 accidents were PDO, 7 were injury) that happened around the first intersection; and, there were 26 accidents (23 accidents were PDO, 3 were injury) happened around the second intersection.

Table 5.1 Comparison at intersections

	Intersection 1	Intersection 2
Location	4th Ave SE and 1th St SE	6th Ave SE and 1th St SE
Total number of accidents in 1999-2003	23	26
Number of PDO accidents	16	23
Number of accidents with injury	7	3
Risk index with DBCTAR (PIARC model)	43.0	33.5
Average density estimation (per 100m^2)	65.6	73.8
Accidents located in the risk area in 2004	6	0
Accidents located in the risk area in 2005	2	1

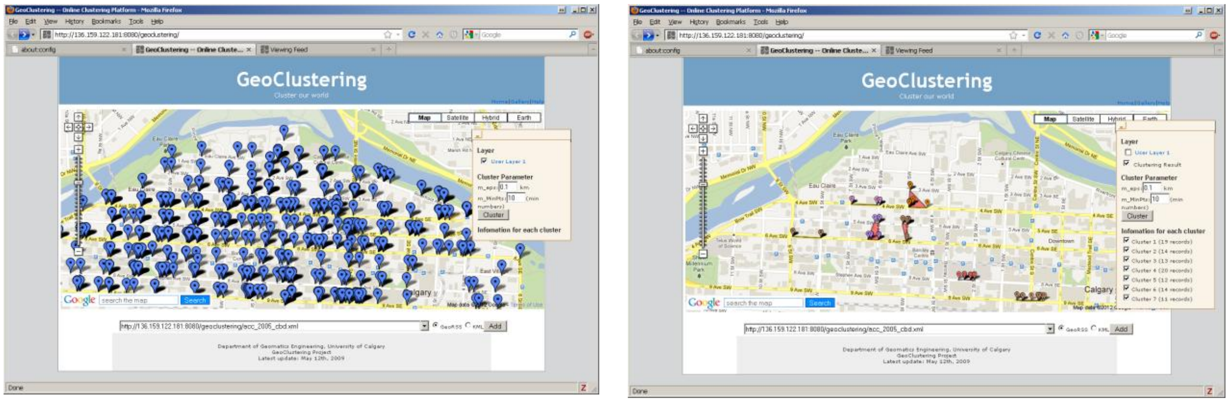
The kernel density method only considers the total number of accidents; therefore, Intersection 1 in Table 5.1 has an average density potential of 65.6/100m^2, which is less than Intersection 2, with an average density value of 73.8/100m^2. However, according to the DBCTAR method with the PIARC model, Intersection 1 has a higher index (43.0) than Intersection 2 (33.5), as this method also takes into account the severity level of the accidents. During 2004 and 2005, there were 6 and 2 accidents near the first intersection, and only 1 accident in 2005 was located in the second intersection.

In general, with the kernel density method, high density means high risk; therefore, the first intersection should be less dangerous than the second one. The opposite result occurs for 2004 and 2005. The DBCTAR method can identify similar patterns as the kernel density method. However, since the DBCTAR method also considers the severity level of each accident, we have a better matching result than that of the kernel density method. Due to the limited data source (only 7 years of data), no further comparisons are conducted to verify that the DBCTAR can surpass the kernel density method at all times. However, this case is good enough to demonstrate that the DBCTAR method is more suitable for determining the accident risk under particular situations.

In order to demonstrate the generic web-based clustering service, GeoClustering, platform, a practical application simulation was conducted. We assumed the previous dataset for the “downtown area at rush hours in the morning” was saved in GeoRSS format, which contains 869 points with geographical coordinate information and short descriptions. This GeoRSS file is saved on a web server, which can be accessed from the Internet.

If a user wants to use a traditional density method, DBSCAN algorithm, to identify those areas in downtown Calgary with high density using this dataset, the user can go to the GeoClustering web page interface at http://www.geoclustering.com. To load the dataset, simply input the URL of the saved GeoRSS file. The result is shown in

Figure 5.5(a). When the radius (*Eps*) is set to 0.1 km, and the minimum number of points is set to 10, seven clusters are generated for the area as shown in Figure 5.5(b).



http://136.159.122.181:8080/geoclustering/acc_2005_cbd.xml ● GeoRSS ○ KML

Layer
☐ User Layer 1
☑ Clustering Result

Cluster Parameter
m_eps: 0.1 km
m_MinPts: 10 (min numbers)

Infomation for each cluster
☑ Cluster 1 (15 records)
☑ Cluster 2 (14 records)
☑ Cluster 3 (13 records)
☑ Cluster 4 (20 records)
☑ Cluster 5 (12 records)
☑ Cluster 6 (14 records)
☑ Cluster 7 (11 records)

Department of Geomatics Engineering, University of Calgary
GeoClustering Project
Latest update: May 12th, 2009

(a)Dataset loading result

(b)Dataset clustering result

Figure 5.5 GeoClustering with the Calgary downtown area at rush hours in the morning dataset (1999-2003)

5.3 Case Two

In the second case study, the user's goal is to generate a risk map "between 8:00-10:00pm" and "under severe weather conditions" in the downtown area of Calgary. The spatial query task is the same as Case One. However, this time, for the first map, the non-spatial task only includes one temporal condition task that finds the 8:00-10:00pm interval shown in Figure 5.6. For the second map, the find severe weather task is shown in Figure 3.5.

```
sub-task: findTimeIntervalTask
defgoal find rush hours
Input: (object (is-a timerange) (object?tr) startat?(is-a
TimeOfDay-PM 8) endat?(is-a TimeOfDay-PM 10))
Output: (object (is-a timerange) (object?tr))
```

Figure 5.6 Pseudocode of non-spatial query task findTimeIntervalTask

After reasoning is performed to filter the dataset, 848 records were returned as the final dataset for the first map; and, 497 records were returned as the final dataset for the second risk map. Figure 5.7 shows one of the final traffic accident risk maps between 8:00-10:00pm in the downtown area. The map has been generated with the risk model parameters recommended by PIARC. The parameter *MinRisk* was set to 9, *Eps* was set to 45 metres, and *MinPts* was set to 5.

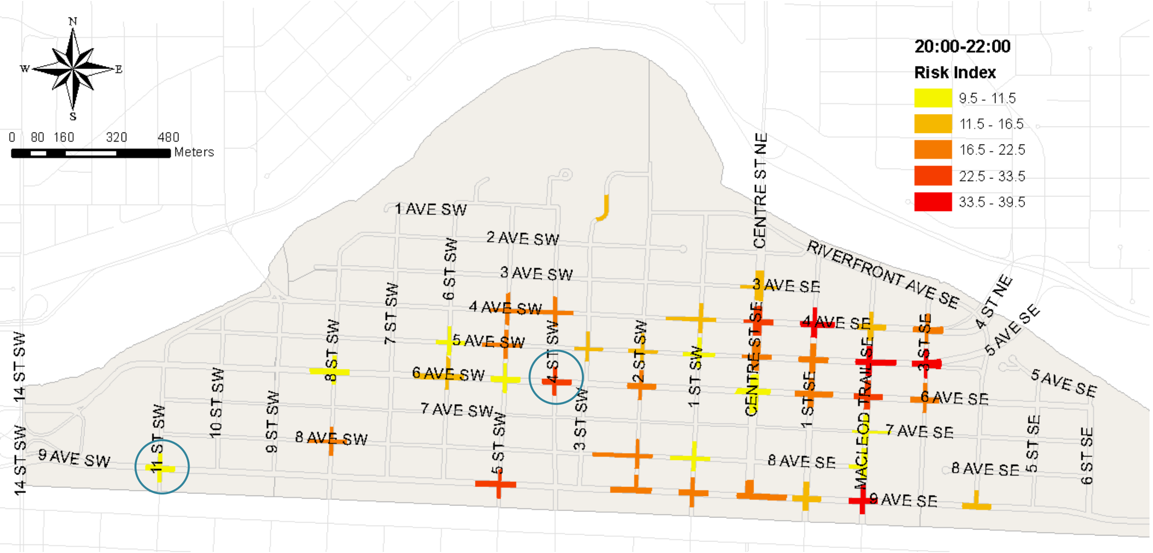


Figure 5.7 Risk map of 8:00-10:00 pm of the Calgary downtown area

When compared the result of Case One, we noticed the risk values of several intersections were quite different. For example, the intersection at 9th Ave SW crossing 11th St SW, had a high risk of 35.5 in the risk map of Case One. However, in the risk map of this case, it has a relatively lower risk index value of 9.5. Contrary to the previous example, the intersection of 6th Ave SW and 4th St SW was even not marked as a risk area in the Case One map; however, it has a high risk index value of 27.5 in Case Two.

Another comparison was conducted between two different weather conditions in the downtown area. Figure 5.8 shows a risk map in the downtown area under the snowing condition, and Figure 5.9 presents a risk map in the same area under raining condition. For the snowing condition, the highest risk intersection is at Macleod Trail SE crossing 6th Ave SE. The highest risk intersection under the raining condition is at the 5th Ave SE crossing 3rd St SE.

This case study demonstrates that, even in the same area, the risk map can be different with different factors (such as temporal and weather conditions).

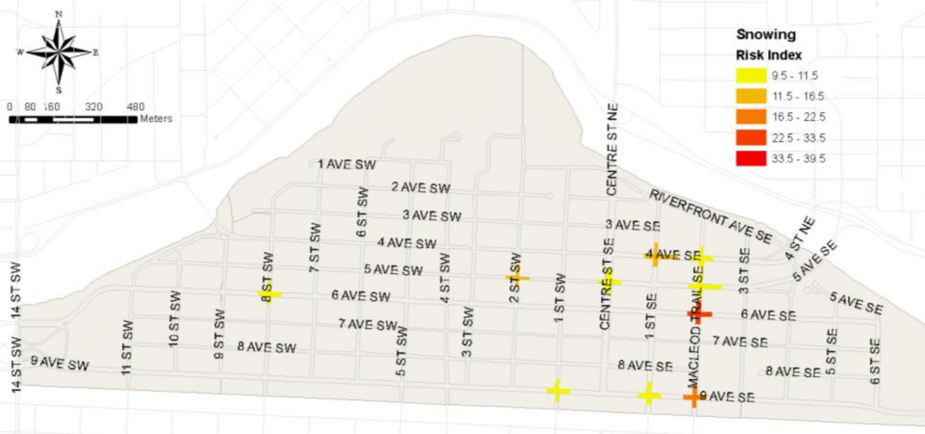


Figure 5.8 Risk map under snowing condition of the Calgary downtown area
(*MinRisk* =8, *Eps*=45, and *MinPts*=5, 372 records)

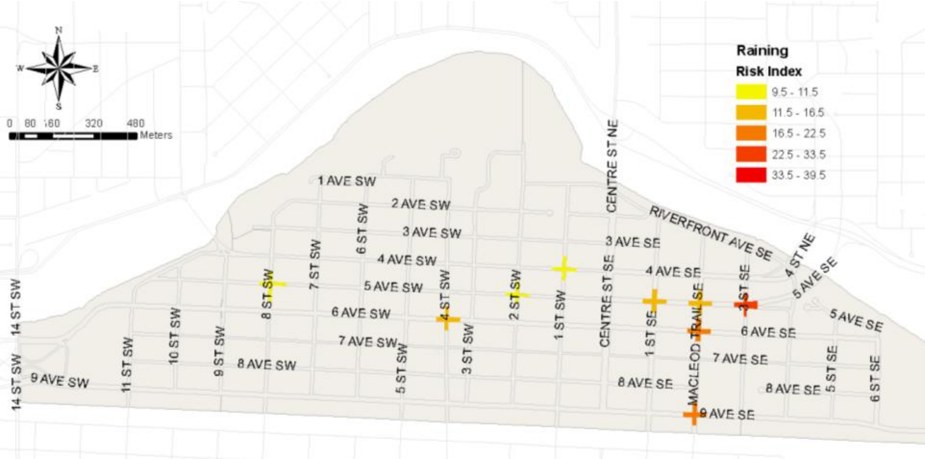


Figure 5.9 Risk map under raining condition of Calgary downtown area
(*MinRisk* =8, *Eps*=45, and *MinPts*=5, 279 records)

5.4 Case Three

In the third case study, the user's goal is to generate a risk map “on the Deerfoot Trail in Calgary”. In this case, the Google Geocoding service cannot return a valid result for most records. Lots of accident geocoding results are manually adjusted based on a road network with linear address information from the City of Calgary. In this case, there is no non-spatial reasoning required, and the spatial query task filtered the 1999-2003 dataset to limit accidents on the “Deerfoot Trail” only. The 2004 and 2005 dataset in the same area was used as the validation dataset.

Figure 5.10 shows one of the final risk maps. This map was generated with the risk model parameters recommended by Transport Canada. The parameter *MinRisk* was set to 80, *Eps* was set to 65 metres, and *MinPts* was set to 60.

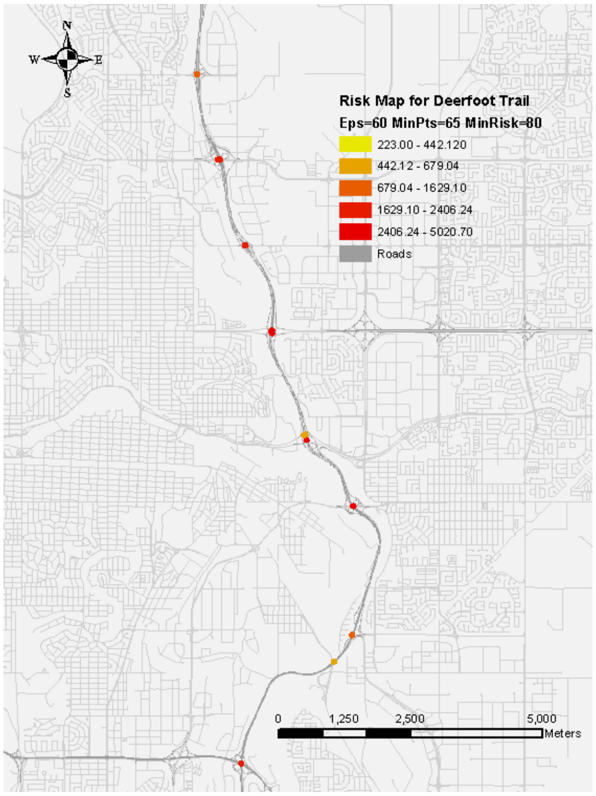


Figure 5.10 Risk map for the Deerfoot Trail in Calgary (extract)

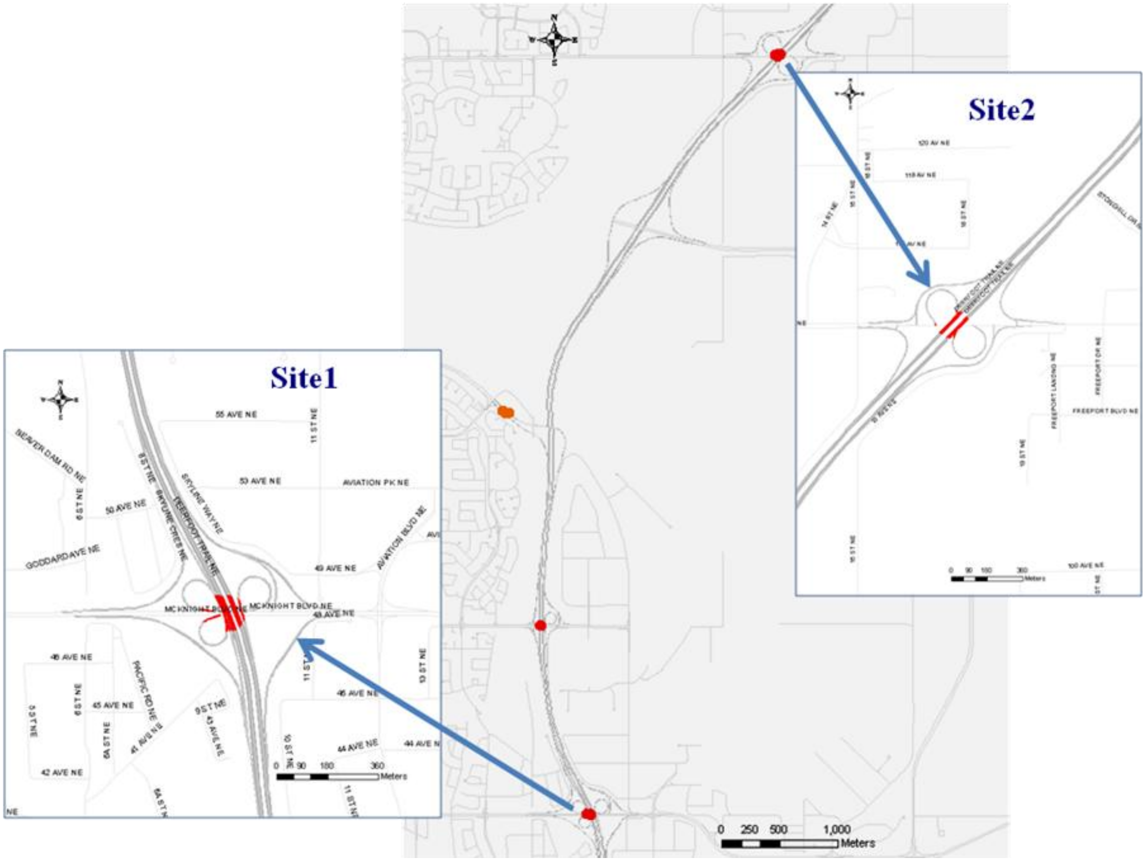


Figure 5.11 Comparison of two road segments on Deerfoot Trail

Table 5.2 Comparison of road segments on Deerfoot Trail

		Site 1	Site 2
Location		Mcknight Blvd NE	Country Hills Blvd NE
1999-2003	Fatality	1	1
	Injury	58	52
	PDO	271	284
	Total Count	330	337
	Risk Index	1855.94	1785.66
2004	Fatality	0	0
	Injury	18	9
	PDO	77	27
2005	Fatality	0	0
	Injury	17	8
	PDO	71	36

This risk map generated by the DBCTAR was also compared with the simple count method that is currently in use by the Calgary Police (Calgary Police Service 2009). Figure 5.11 shows locations of two selected road segments and Table 5.2 shows the detailed result at each location.

According to the 1999-2004 dataset, 330 accidents (271 accidents with PDO, 58 with injuries, 1 with fatalities) occurred around the Mcknight Blvd NE intersection; and, 337 accidents (284 accidents with PDO, 52 with injuries, 1 with fatalities) happened around the Country Hills Blvd NE intersection, respectively. Site 1 had a smaller number of accidents than did site 2; however, according to our DBCTAR method with the Transport Canada model, site 1 had a higher index (1855.94) than site 2 (1785.66), as this method also takes into account the severity level of the accidents. During 2004, there were 18 accidents with injuries and 77 accidents with PDO in site 1, and only 9 accidents with injury and 27 accidents with PDO were located in site 2. During 2005, there were 17 accidents with injuries and 71 accidents with PDO in site 1, and only 8 accidents with injuries and 36 accidents with PDO were located in site 2. This is another example that shows the DBCTAR method has a better performance in assessing accident risk.

5.5 Case Four

The fourth case study presents ideas about potential usage of the risk map for providing personal navigation assistance to the user. If a user wants to go to Bowness Park from the University of Calgary, the user has two route options, as shown in Figure 5.12. These two routes have almost the same distance and travelling time. In this case, user may refer to the “risk” to determine which route should be taken.

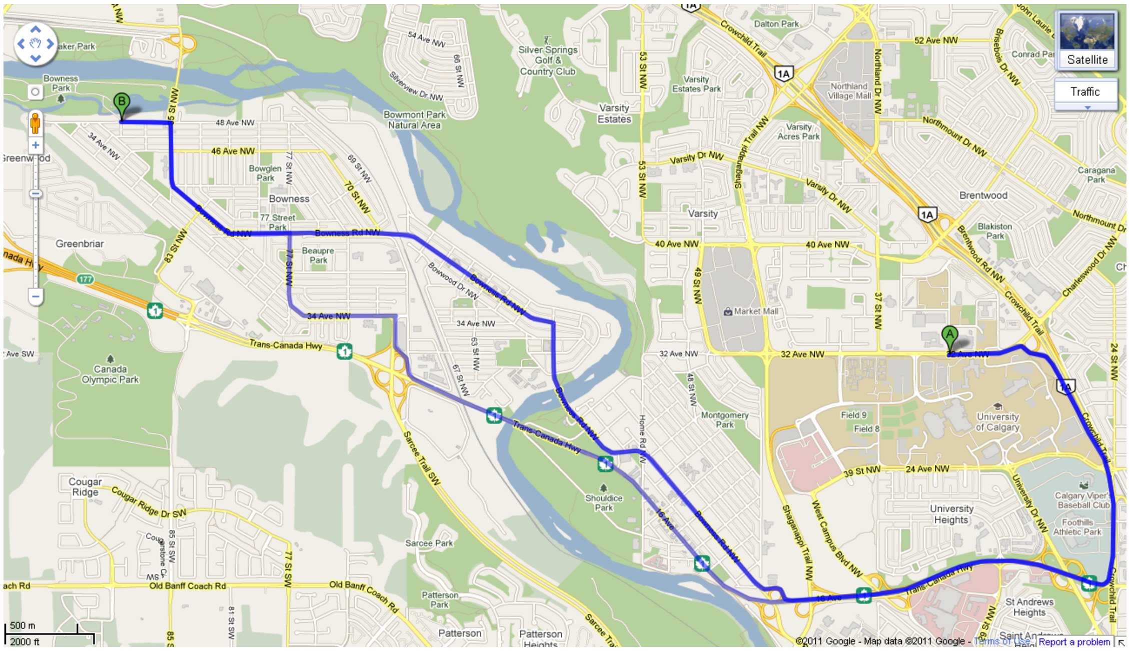


Figure 5.12 Two routes from the University of Calgary to Bowness Park

The ONTO_TARM system can generate a risk map around these communities to help the user to determine which route is better. As mentioned before, all census units, i.e. communities in Calgary, are represented as instances of the *Community* class, which are also geometric polygons. The route from the University of Calgary to Bowness Park can be extracted as instances of geometric line. With the extracted communities' geometric polygons, the reasoner uses the geometric lines as the input. The spatial relationship "meet" between the route and each community is then checked. The spatial query result returns 12 communities on the given routes.

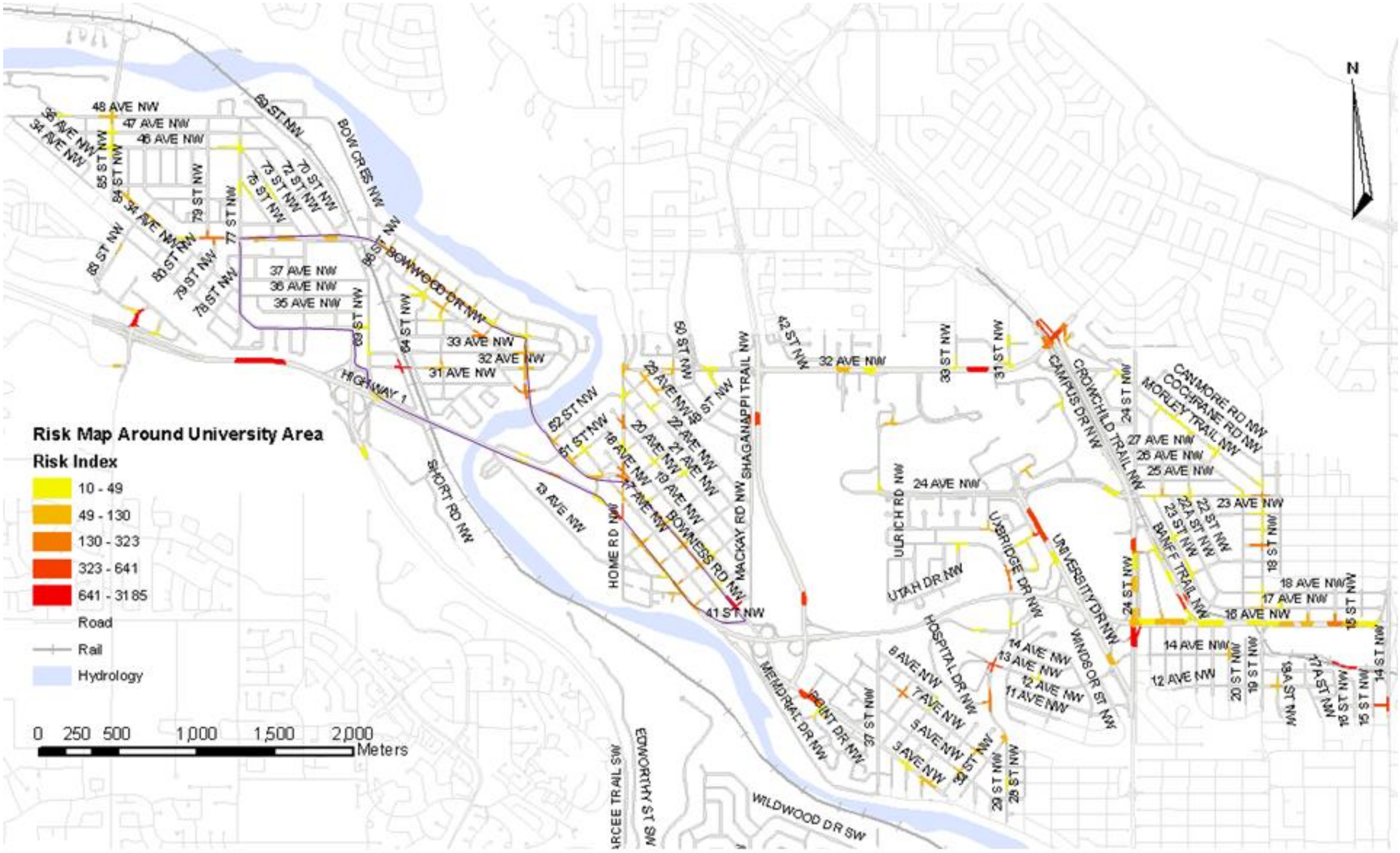


Figure 5.13 Risk map of the community around University of Calgary

In this case, the users do not give detailed requirements on the non-spatial reasoning; therefore, the full dataset is returned. The final dataset based on the ontology-based query includes 11,198 records. The risk map is generated and is shown in Figure 5.13. If we overlap the two different routes on the risk map and compare the accumulated total risk value, the user can have an intuitive view for determination. As the risk value for northern route is 3241.3, and the southern route is 1880.56, the better choice is to take the southern route.

Chapter Six: Conclusions and Future Works

6.1 Conclusions

This thesis proposes an ontology-based traffic accident risk-mapping (ONTO_TARM) framework. In ONTO_TARM, the ontology represents the domain knowledge, including the non-spatial and spatial concepts and definitions related to the traffic accidents, and helps to retrieve the most suitable dataset from the raw historical datasets based on users' goals for the generation of their own risk maps.

In addition, a geospatial clustering method – the density-based clustering algorithm for traffic accident risk (DBCTAR) – is proposed. This new clustering method has been extended from DBSCAN with consideration of both the total accident numbers and the severity levels of the accidents. In a simplified version, the value of equivalent property damage only is calculated for each cluster and used as the risk index value. The proposed method is adapted for a road network environment. The clustering result shows the boundary of each cluster subject to the boundary of the network.

A new web-based geospatial clustering service for discovering hidden patterns – GeoClustering – is also proposed and implemented. It is used as the clustering engine of the ONTO_TARM framework. It is an open, easy-to-use generic geospatial clustering web service, which can be used independently. This web service can be used to identify clusters from distributed data sources, with free access at anytime from anywhere. By using open and interoperable service interfaces, users are able to cluster their data online and visualize the clustering patterns on the map easily and conveniently.

Finally, case studies based on real traffic accident data have been implemented within a prototype of the proposed ONTO_TARM framework. The preliminary results achieved from the case studies are promising.

6.2 Future Works

This research can be extended in the future in the following ways: First, a system prototype would benefit from an improved ontology reasoner and a more powerful map generator. The current prototype is still a proof-of-concept type, using the Jena reasoner actually only works on the level of RDFS (Resource Description Framework Schemas) and, it could be upgraded to handle more complex ontology queries. The efficiency of the map generator needs to be improved. In our experiments, the execution time for more than 10,000 selected records with a complicated road network could take over 10 minutes. The most time-consuming work (>80% of the total running time) is generating maps from the clustering result.

Second, the current system cannot provide automatic recommendations for the weight model selection and clustering parameter settings. For the given case studies, the *k-dist* graph (Ester et al. 1996) and *k*-nearest *riskindex* graph are used to help users set parameters. How to set parameters automatically is remaining a question.

Third, the risk index can be better defined. For example, the affect of traffic volume or other exposure measurement should be used. In addition to the severity level, other properties of the accidents can be adopted in the risk index model. Also, the DBCTAR algorithm may need new rules to merge clusters at intersections.

Fourth, the GeoClustering could be improved in the following aspects:

1) More algorithms and advanced techniques could be developed and integrated into the platform. To reduce the execution time, clustering, spatial indexing, and data compression techniques may be used to improve network data transmission.

2) More data types need to be supported by the platform. The current version of GeoClustering only supports static datasets. Dynamic and continuous datasets, such as real time sensor data, have also been collected and need to be analyzed as well. Therefore, some new clustering methods and mechanisms dealing with real-time datasets can be added.

3) It is better to harness collective intelligence from users. In the web 2.0 era, web users should be treated as participants and contributors rather than simple data consumers. With implementation of online communication functions in the near future, users are encouraged to share and discuss the patterns of clustering results interactively. In return, such collective intelligence will provide users with a better understanding of spatial data and more ways to help them select suitable clustering parameters.

References

ACME Labs: JavaScript Utilities, http://acme.com/javascript/#Clusterer. (Last accessed on June 22, 2009).

Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic subspace clustering of high dimensional data for data mining applications. SIGMOD '98 Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data. 94–105. ACM, New York, NY, USA (1998).

Anderson, T.K.: Kernel density estimation and K-means clustering to profile road accident hotspots. Accident Analysis & Prevention. 41, 359–364 (2009).

Ankerst, M., Breunig, M.M., Kriegel, H.-P., Sander, J.: OPTICS: ordering points to identify the clustering structure. SIGMOD '99 Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data. 49–60. ACM, New York, NY, USA (1999).

Apache Jena project team: Apache Jena, http://incubator.apache.org/jena/ (Last accessed on Dec 20, 2010).

Bishr, Y.: Overcoming the semantic and other barriers to GIS - interoperability. International Journal of Geographical Information Science. 12, 299–314 (1998).

Black, W.R., Thomas, I.: Accidents on belgium’s motorways: a network autocorrelation analysis. Journal of Transport Geography. 6, 23–31 (1998).

Boots, B.N., Getis, A.: Point pattern analysis. Sage Publications, Newbury Park, CA (1988).

Borruso, G.: Network Density Estimation: Analysis of Point Patterns over a Network. International Conference on Computational Science and Its Applications – ICCSA 2005. 126–132 (2005).

Calgary Police Service: 2008 Report to the Community. http://www.calgarypolice.ca/pdf/AR08.pdf (Last accessed on June 22, 2009).

Cheeseman, P., Kelly, J., Self, M., Stutz, J., Taylor, W., Freeman, D.: AutoClass: a Bayesian classification system. Readings in knowledge acquisition and learning: automating the construction and improvement of expert systems. 431–441 (1993).

ClusterSeer: http://www.terraseer.com/products_clusterseer.php (Last accessed on June 22, 2009).

ClustrMaps: http://clustrmaps.com/ (Last accessed on June 22, 2009).

Cycorp, Inc.: http://www.cyc.com/2003/04/01/cyc (Last accessed on June 22, 2009).

Doherty, S.T., Andrey, J.C., MacGregor, C.: The situational risks of young drivers: The influence of passengers, time of day and day of week on accident rates. Accident Analysis & Prevention. 30, 45–52 (1998).

Egenhofer, M.J., Franzosa, R.D.: Point-set topological spatial relations. International Journal of Geographical Information Science. 5, 161–174 (1991).

ERSO(European Road Safety Observatory), Definition road safety risk indicator, http://ec.europa.eu/transport/wcm/road_safety/erso/data/Content/definition_road_safety_risk_indicator.htm (2007) (Last accessed on June 22, 2009)

ESRI: Esri Products, http://www.esri.com/products/index.html (Last accessed on June 22, 2009).

Ester, M., Kriegel, H., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. KDD '96 Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (1996).

Fisher, D.H.: Knowledge acquisition via incremental conceptual clustering. Machine Learning. 2, 139–172 (1987).

Flahaut, B., Mouchart, M., Martin, E.S., Thomas, I.: The local spatial autocorrelation and the kernel method for identifying black zones: A comparative approach. Accident Analysis & Prevention. 35, 991–1004 (2003).

GeoBase: http://www.geobase.ca/geobase/en/index.html (Last accessed on June 22, 2009).

Getis, A.: A History of the Concept of Spatial Autocorrelation: A Geographer's Perspective. Geographical Analysis. 40, 297–309 (2008).

Getis, A., Ord, J.K.: The Analysis of Spatial Association by Use of Distance Statistics. Geographical Analysis. 24, 189–206 (1992).

Geurts, K., Wets, G.: Black Spot Analysis Methods: Literature Review, Flemish Research Center for Traffic Safety, Diepenbeek, Belgium. (2003).

Gómez-Pérez, A., Fernández-López, M., Corcho, O.: Ontological Engineering: With Examples from the Areas of Knowledge Management, E-Commerce and the Semantic Web. Springer (2004).

Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquisition. 5, 199–220 (1993).

Guha, S., Rastogi, R., Shim, K.: CURE: an efficient clustering algorithm for large databases. SIGMOD '98 Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data. 73–84. ACM, New York, NY, USA (1998).

Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann, San Francisco (2006).

Han, J., Kamber, M., Tung, A.K.H.: Spatial Clustering Methods in Data Mining: A Survey. In: Miller, H. J. and Han, J. (eds.) Geographic Data Mining and Knowledge Discovery, Taylor and Francis, London (2001).

Hwang, J.S.: Ontology-based spatial clustering method: case study of traffic accidents. Presented at the Student Paper Sessions, UCGIS Summer Assembly (2003).

Hwang, S.: Using Formal Ontology for Integrated Spatial Data Mining. Computational Science and Its Applications – ICCSA 2004. 1026–1035 (2004).

IRTAD(International Traffic Safety Data and Analysis Group): IRTAD Annual Report 2011. Organization for Economic Co-operation and Development/International Transport Forum (OECD/ITF), Paris. (2012).

Jacquez, G.M.: Spatial Cluster Analysis. In: Wilson, J.P. and Fotheringham, A.S. (eds.) The Handbook of Geographic Information Science. 395–416. Blackwell Publishing Ltd (2008).

Jaffe, A., Naaman, M., Tassa, T., Davis, M.: Tag Maps, http://www.slideshare.net/mor/tag-maps (Last accessed on June 22, 2009).

Karypis, G., Han, E.-H., Kumar, V.: Chameleon: hierarchical clustering using dynamic modeling. Computer. 32, 68–75 (1999).

Kaufman, L., Rousseeuw, P.J.: Finding groups in data: an introduction to cluster analysis. John Wiley & Sons, New York (1990).

Kulldorff, M.: SaTScan User Guide, http://www.satscan.org/techdoc.html. (Last accessed on June 22, 2009).

Lee, J.-G., Han, J., Kamber, M.: An Overview of Clustering Methods in Geographic Data Analysis. In: Miller, H. and Han, J. (eds.) Geographic Data Mining and Knowledge Discovery, Second Edition. 149–187. CRC Press (2009).

Maedche, A., Zacharias, V.: Clustering Ontology-Based Metadata in the Semantic Web. Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery. 348–360. Springer-Verlag, London, UK (2002).

Maiom: maiom, http://www.maiom.com/mappa/ (Last accessed on June 22, 2009).

Microsoft Corp.: Virtual Earth Developer Center, MSDN, VEShapeLayer.SetClusteringConfiguration Method, http://msdn.microsoft.com/en-us/library/cc966930.aspx (Last accessed on June 22, 2009).

Miller, H.J.: Geographic Data Mining and Knowledge Discovery. In: Wilson, J.P. and Fotheringham, A.S. (eds.) The Handbook of Geographic Information Science. 352–366 (2008).

Moran, P.A.P.: Notes on Continuous Stochastic Phenomena. Biometrika. 37, 17–23 (1950).

Ng, R.T., Han, J.: CLARANS: A Method for Clustering Objects for Spatial Data Mining. IEEE Transaction on Knowledge and Data Engineering. 14, 1003–1016 (2002).

OGC: GeoRSS White Paper, http://www.opengeospatial.org/pt/06-050r3 (2006) (Last accessed on June 22, 2009).

OGC: KML, http://www.opengeospatial.org/standards/kml (2008) (Last accessed on June 22, 2009).

Okabe, A., Satoh, T., Sugihara, K.: A kernel density estimation method for networks, its computational method and a GIS-based tool. International Journal of Geographical Information Science. 23, 7–32 (2009).

Okabe, A., Yamada, I.: The K-Function Method on a Network and Its Computational Implementation. Geographical Analysis. 33, 271–290 (2010).

Pearman, M.: Google Maps API Projects, http://googlemapsapi.martinpearman.co.uk/home.php (Last accessed on June 22, 2009).

Peden, M., Scurfield, R., Sleet, D., Mohan, D., Hyder, A.A., Jarawan, E., Mathers, C. eds: World report on road traffic injury prevention. World Health Organization, Geneva (2004).

Peuquet, D.J.: Representations of Space and Time. Guilford Press (2002).

Prot ég éTeam: The Prot ég éhomepage, http://protege.stanford.edu/ (Last accessed on June 22, 2009).

RememberRoadCrashVictims.ca: http://www.RememberRoadCrashVictims.ca (Last accessed on June 22, 2009).

Rifaat, S.M., Tay, R., De Barros, A.: Effect of Street Pattern on Road Safety: Are Policy Recommendations Sensitive to Aggregations of Crashes by Severity? Transportation Research Record: Journal of the Transportation Research Board. 2147, 58–65 (2010).

Ripley, B.D.: Spatial Statistics. John Wiley & Sons, New York (1981).

Sander, J., Ester, M., Kriegel, H.P., Xu, X.: Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications. Data Mining and Knowledge Discovery. 2, 169–194 (1998).

Sheikholeslami, G., Chatterjee, S., Zhang, A.: WaveCluster: a wavelet-based clustering approach for spatial data in very large databases. The VLDB Journal. 8, 289–304 (2000).

Shen Y. Ruan D., Hermans E., Brijs T., Wets G., and Vanhoof K.: Road safety risk evaluation and target setting using data envelopment analysis and its extensions. Accident Analysis & Prevention. In Press. (2012).

Shiode, S.: Analysis of a Distribution of Point Events Using the Network-Based Quadrat Method. Geographical Analysis. 40, 380–400 (2008).

Silverman, B.W.: Density Estimation for Statistics and Data Analysis. Chapman and Hall (1986).

Smith, M.K., Welty, C., McGuinness, D.L.: OWL Web Ontology Language Guide, http://www.w3.org/TR/owl-guide/ (2004) (Last accessed on June 22, 2009).

Steenberghen, T., Dufays, T., Thomas, I., Flahaut, B.: Intra-urban location and clustering of road accidents using GIS: a Belgian example. International Journal of Geographical Information Science. 18, 169–181 (2004).

Steenberghen, T., Aerts, K., Thomas, I.: Spatial clustering of events on a network. Journal of Transport Geography. 18, 411–418 (2010).

Stefanakis, E.: NET-DBSCAN: clustering the nodes of a dynamic linear network. International Journal of Geographical Information Science. 21, 427–442 (2007).

Tay, R.: Saving Lives on Canadian Roads. http://www.publichealthworks.ca/archive/2006/2006_05_Tay_SavingLivesonCanadianRoads.pdf (2006) (Last accessed on June 22, 2009).

Tobler, W.: A Computer Movie Simulating Urban Growth in the Detroit Region. Economic Geography. 46, 234–240 (1970).

W3C: Web Services Architecture, http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/ (2004) (Last accessed on June 22, 2009).

W3C: OWL 2 Web Ontology Language Document Overview, http://www.w3.org/TR/owl2-overview/ (Last accessed on June 22, 2009).

Wang, W., Yang, J., Muntz, R.R.: STING: A Statistical Information Grid Approach to Spatial Data Mining. Proceedings of the 23rd International Conference on Very Large Data Bases. 186–195 (1997).

Wang, X., Gu, W., Ziebelin, D., Hamilton, H.: An ontology-based framework for geospatial clustering. International Journal of Geographical Information Science. 24, 1601–1630 (2010).

Wang, X., Hamilton, H.J.: DBRS: a density-based spatial clustering method with random sampling. Proceedings of the 7th Pacific-Asia conference on Advances in knowledge discovery and data mining. 563–575 (2003).

Wang, X., Hamilton, H.J.: Towards an ontology-based spatial clustering framework. Proceedings of the 18th Canadian Society conference on Advances in Artificial Intelligence. 205–216 (2005).

Xie, Z., Yan, J.: Kernel Density Estimation of traffic accidents in a network space. Computers, Environment and Urban Systems. 32, 396–406 (2008).

Yamada, I., Thill, J.-C.: Comparison of planar and network K-functions in traffic accident analysis. Journal of Transport Geography. 12, 149–158 (2004).

Yan, H., Weibel, R.: An algorithm for point cluster generalization based on the Voronoi diagram. Computers & Geosciences. 34, 939–954 (2008).

Yiu, M.L., Mamoulis, N.: Clustering objects on a spatial network. SIGMOD '04 Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data. 443–454. ACM, New York, NY, USA (2004).

Yue, D., Wang, S., Zhao, A.: Traffic Accidents Knowledge Management Based on Ontology. FSKD '09 Proceedings of the 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery. 447–449 (2009).

Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: an efficient data clustering method for very large databases. SIGMOD '96 Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data. 103–114. ACM, New York, NY, USA (1996).