

# Visual Interactive Clustering and Querying of Spatio-Temporal Data

Olga Sourina and Dongquan Liu

School of Electrical & Electronic Engineering,  
Nanyang Technological University, Block S2,  
Nanyang Avenue, Singapore 639798  
eosourina@ntu.edu.sg  
<http://www.ntu.edu.sg/home/eosourina>

**Abstract.** Visualization techniques increase the user involvement in the interactive process of data mining and querying of spatio-temporal data. This paper describes a novel geometric approach to clustering and querying of spatio-temporal data. We propose the uniform geometric model based on function representation of solids to cluster and query time-dependent data. Clustering and querying are integrated with visualization techniques in one GUI. First, visual clustering with blobby model allows the user to see the result of clustering on the screen for different time points and/or time intervals and set the appropriate parameters interactively. After that, the user gets the data of clusters for the chosen time frames. Then, the user can visually query the cluster/clusters he/she is interested in with geometric primitive solids which currently are cubes, spheres/ellipsoids, cylinders, etc. Geometric operations of union, intersection and/or subtraction can be performed over the geometric primitive solids to get the final query shape. The user visually clusters spatio-temporal data and queries the clusters with geometric shapes through graphics interface accessing dynamically 3D projections of multidimensional points from database, warehouses or files. With the uniform geometric model of the clustering and querying of spatio-temporal data, 3D visualization tools can be naturally incorporated in one system to allow the user to visualize and query clusters changing over time.

## 1 Introduction

Clustering and querying of spatial data are classical problems in databases, and warehousing. Clustering algorithms can be applied for similarity search, customer segmentation, pattern recognition, trend analysis, etc. Numbers of algorithms for clustering multidimensional data have been proposed in the last few years, e.g. partitioning method such as k-means [1] and k-medoids [2], hierarchical method such as CURE [3], density-based method such as DBSCAN [4] and DENCLUE [5], etc. However, analysis of spatio-temporal data including clustering has received less attention. To be able to run on spatio-temporal data spatial clustering algorithms need temporal extensions [6, 7].

It becomes more and more important for the modern clustering systems to give the user an easy understanding of both the data set and the results [8]. Visualization offers

the user an intuitive way of analysis that can help to discover data patterns and structures. Data visualization techniques [9, 10] when incorporated with clustering algorithms could improve interpretability and usability of the data and clustering process. Visualization techniques could be used not only for the interpretation of the results but also for the interpretation of the whole process of clustering in order to help the user to come up with hypothesis and to set the values of parameters. For instance, the user could select the projection directions [11] for high dimension data set. To incorporate visualization techniques, the existing clustering algorithms use the result of clustering algorithm as the input for visualization system [10]. The drawback of such approach is that it can be costly and inefficient. The better solution is to combine two processes together, which means to use the same model in clustering and visualization.

On the other hand, development of query methods and graphical user interfaces is a new trend in data mining [12]. Querying of time-dependent data is a classical problem in temporal databases and warehousing. The goal of works in this area is to propose data representation model and query model able to handle time-dependent geometries including those changing continuously that describe moving objects [13, 14]. Spatio-temporal predicates are introduced to query time-dependent data [14].

In work [15], we proposed and fully described geometric query model with implicit functions. Then, in work [16], we proposed a solid-based clustering method. In this paper, we extend the uniform geometric model to handling time-dependent data. We proposed to apply solid-base clustering algorithm to data changing over time. The extended uniform geometric model allows integrate solid-based visual clustering and visual querying of time-dependent data in one GUI. Implicit function is used in spatio-temporal predicate implementation. This allows us to pose complex shape queries changing over the time. Our extended model allows us to integrate clustering, querying and visualization of spatio-temporal data. Spatio-temporal query languages are not discussed in this paper.

The paper is organized as follows. Section 2 introduces the model defined with implicit functions that is used for interactive visual clustering of time-dependent data and describes similarity of the model with the model of density-based methods. Querying of time-dependent data based on the geometric query model is described in Section 3. Implementation of the system and examples of visual clustering and querying are discussed in Section 4. In Section 5, conclusion and future work are considered.

## 2 Solid-Based Clustering of Spatio-Temporal Data

The implicit modeling techniques are relatively new. This approach has become more sophisticated, generating new interest in computer graphics and related fields [17]. It uses implicit function instead of parametric function or explicit function as its mathematical foundation. In work [16], we proposed to define a cluster as a solid reconstructed on the points with the implicit functions. In this paper, we define cluster as a solid existing at time point. The solid not only describes the granular property of the cluster but also describes its boundary. With this definition, a new object could be easily identified to which cluster it belongs.

Let  $\mathbf{P}$  be a set of multidimensional points  $\mathbf{P}=\{[p_1, p_2, \dots, p_n, t]\} = \{[\mathbf{P}, t]\}$  in  $n$  dimensional Euclidean space  $E^n$ , and  $t$  is time in the  $n$ -dimensional Euclidean space  $E^n$ . Then, a solid reconstructed on the points can be described with function-based representation as follows:

$f(\mathbf{X}, \mathbf{P}) \geq 0$ , where  $\mathbf{X}$  belongs to  $E^n$ . The function can be defined by procedure. Such function defines closed  $n$ -dimensional geometric solid in  $E^n$  space under the following conditions:

- $f(\mathbf{X}, \mathbf{P}) > 0$  for the points inside the solid,
  - $f(\mathbf{X}, \mathbf{P}) = 0$  for the points on the solid boundary,
  - $f(\mathbf{X}, \mathbf{P}) < 0$  for the points outside the object,
- where  $\mathbf{X} = \{x_1, x_2, \dots, x_n\}$  is a position vector of the point in  $E^n$ .

The zero set of these functions provides surfaces, and the values that are greater or equal to zero define multidimensional geometric solid objects. We consider each cluster as a different solid per time point. For querying we use function-based representation of the query solid as well. For clustering, we implement solid-based subdivision algorithm for each time frame. In solid-based subdivision algorithm that was proposed and described for spatial data in work [18], we build the solid by computing the density function of points in the whole field, which means adding up all the influence functions of points inside the field. The sum is a field density function that consists of all influence functions of the points. The field density function can make a complete description of the whole data space. Parabolic function, square wave function and Gaussian function are some examples of basic influence function in density-based algorithms. The functions used in blobby, meta-balls or soft objects in Computer Graphics can also serve as the basic influence function for better efficiency of our method.

In this work, we use blobby model that is similar to the model used in density-based clustering methods. Blobby model was first accomplished by Blinn, and now the term blobby always includes other related models and is not limited only to the original model. The blobby primitive is described as follows:

$$f(r) = a * e^{-(r/b)}$$

where  $r$  is the distance from the center point of a primitive,  $a$  is the height of the function and  $b$  is related to the standard deviation which is Gaussian. At any point of the surface, the isosurface “potential” is equal to the sum of all the primitives’ contributions using the following function:

$$F(X, P) = \sum_{i=1}^N f(r) - T \geq 0$$

where  $N$  is the total number of blobby primitives and  $T$  is the threshold constant that determines the value of the isosurface.

In blobby model, the distance from the center point of the primitives describes the range that the primitive can influence on, the summary function adds up the influences of all primitives and the threshold determines the level of the isosurface

built. In our method, the same blobby model is used in the subdivision algorithm. By connecting all points with the same potential value (i.e. the threshold value), we can build a smooth implicit surface around the cluster. If there are many clusters in the data set, we will get many implicit surfaces with the same potential value. And each of the implicit surfaces will wrap the cluster. In our model, implicit surfaces serve as the boundaries of clusters. By substituting one point's coordinates into the blobby function and compare the result with  $T$ , we can easily know whether the point is inside or outside of the implicit surface. Thus, the solid-based subdivision algorithm is implemented as follows. We connect each point of each time point dataset with other points and sort the points into the clusters checking the values along the line connecting two points.

As we mentioned before, the formulae of blobby model is similar to the density-based clustering model of DBSCAN, OPTICS and DENCLUE. In density-based methods, Gaussian function is used as follows:

$$f^r(\vec{x}) = e^{-\frac{d(\vec{x}, \vec{r})^2}{2\sigma^2}}, \text{ where } d(\vec{x}, \vec{r}) \text{ is a distance between two points.}$$

We implemented visual clustering with blobby functions. The blobby formula has additional parameter  $\mathbf{a}$  which can make cluster shape "thinner". We have to set interactively three parameters for our model  $\mathbf{a}$  – scale factor,  $\mathbf{b}$  – exponential factor, and  $\mathbf{T}$  – threshold value.

We have to note that the solid-based subdivision algorithm can also act as a stand alone clustering algorithm even without being integrated with visualization techniques into the system.

### 3 Querying of Spatio-Temporal Data

After we visualize clusters per time point or/and time interval using interactively set parameters, we can query the clusters with geometric objects. In this section, we describe our geometric query model consisting from geometric objects generally changing over time and operations. The proposed model is an extension of the model that was introduced first in work [15]. As it was shown there, geometric interpretation of relational algebra selection operation can be phrased as follows: "find out the points that belong to the solid." In our model, the query solid can be a complex geometric solid. The complex query solid can be created with union, intersection, and other operations over primitive solids that are generally hyperhalfspaces, hypercuboids, hyperellipsoids, etc. Selection operation of relational algebra can be found in geometry as point/solid classification predicate. In this paper, we extend the model with time dimension. Then, the point/solid classification predicate can be described as follows. Let  $P$  be a point in Euclidean space  $E^n$  and  $t$  is time,  $G_1$  be a query solid described with implicit function  $f_1$  defined with time-dependent parameters and location changing over time,  $bG_1$  be a boundary of  $G_1$  and  $iG_1$  be an interior of  $G_1$ . Then a point/solid predicate is described with the implicit function representation of the geometric object  $G_1$  by a 3-valued predicate:

$$S_3(P, G_1) = \begin{cases} 0, & \text{if } f_1(x_1, x_2, \dots, x_n, t) < 0 \quad P \notin G_1 \\ 1, & \text{if } f_1(x_1, x_2, \dots, x_n, t) = 0 \quad P \in bG_1 \\ 2, & \text{if } f_1(x_1, x_2, \dots, x_n, t) > 0 \quad P \in iG_1 \end{cases}$$

In our model, query solid can have time-dependent parameters and/or coordinates that can be defined analytically or by procedure. Thus, the geometric query model consists of the following geometric objects:

- $n$ -dimensional points  $P = \{[x_1, x_2, \dots, x_n, t]\}$  where  $t$  is time;
- time-dependent 3-dimensional primitive geometric objects for the construction of a query solid using geometric operations.

The following is an implicit function representation of the primitive time-dependent 3-dimensional geometric solids that could be used for construction of geometric criteria:

**Halfspace:**

$$G_1: f_1(\mathbf{X}, t) = f_1(x_1, x_2, x_3, t) = (x_1 - a[t]) \geq 0$$

Where  $a$  is some real number ( $a \in \mathbb{R}$ )

**Sphere:**

$$G_1: f_1(\mathbf{X}, t) = r[t]^2 - (x_1 - x_{0,1}[t])^2 - (x_2 - x_{0,2}[t])^2 - (x_3 - x_{0,3}[t])^2 \geq 0$$

Where  $x_{0,1}, x_{0,2}, x_{0,3} \in \mathbb{R}$

**Ellipsoid:**

$$G_1: f_1(\mathbf{X}, t) = 1 - ((x_1 - x_{0,1}[t])/a_1[t])^2 - ((x_2 - x_{0,2}[t])/a_2[t])^2 - ((x_3 - x_{0,3}[t])/a_3[t])^2 \geq 0$$

where  $x_{0,1}, x_{0,2}, x_{0,3} \in \mathbb{R}$  and  $a_1, a_2, a_3 \in \mathbb{R}$ .

**Cone:**

$$G_1: f_1(\mathbf{X}, t) = ((x_1 - x_{0,1}[t])/a_1[t])^2 - ((x_2 - x_{0,2}[t])/a_2[t])^2 - ((x_3 - x_{0,3}[t])/a_3[t])^2 \geq 0$$

where  $x_{0,1}, x_{0,2}, x_{0,3} \in \mathbb{R}$  and  $a_1, a_2, a_3 \in \mathbb{R}$ .

**Cylinder:**

$$G_1: f_1(\mathbf{X}, t) = ((x_1 - x_{0,1}[t])/a_1[t])^2 - ((x_2 - x_{0,2}[t])/a_2[t])^2 \geq 0$$

Where  $x_{0,1}, x_{0,2} \in \mathbb{R}$  and  $a_1, a_2 \in \mathbb{R}$ .

By further declaring that our model is open to any type of objects that can be defined implicitly with some functions  $f(x_1, x_2, x_3, t) \geq 0$ , we could avoid the problem of a minimum set of primitives and to change this set depending on the application problem to be solved.

Geometric operations are applied to primitive geometric objects to obtain complex geometric shapes at each time point. The analytical definition of set-theoretic operations is realized in the form proposed by Ricci [19], where operations over implicit functions are considered. Affine transformations (translation, rotation and scaling) are also used to increase an expressive power of the proposed geometric model. Geometric operations include set-theoretic union, intersection, difference and orthographic projection.

Mathematically,

**Union:**  $G_3 = G_1 \cup G_2$  of two objects  $G_1 \subseteq E^n$  and  $G_2 \subseteq E^n$  with the descriptive functions  $f_1$  and  $f_2$  will be defined as  $f_3 = f_1 \vee f_2 = \max(f_1, f_2) \geq 0$ , where  $G_3 \subseteq E^n$ . **Intersection:**

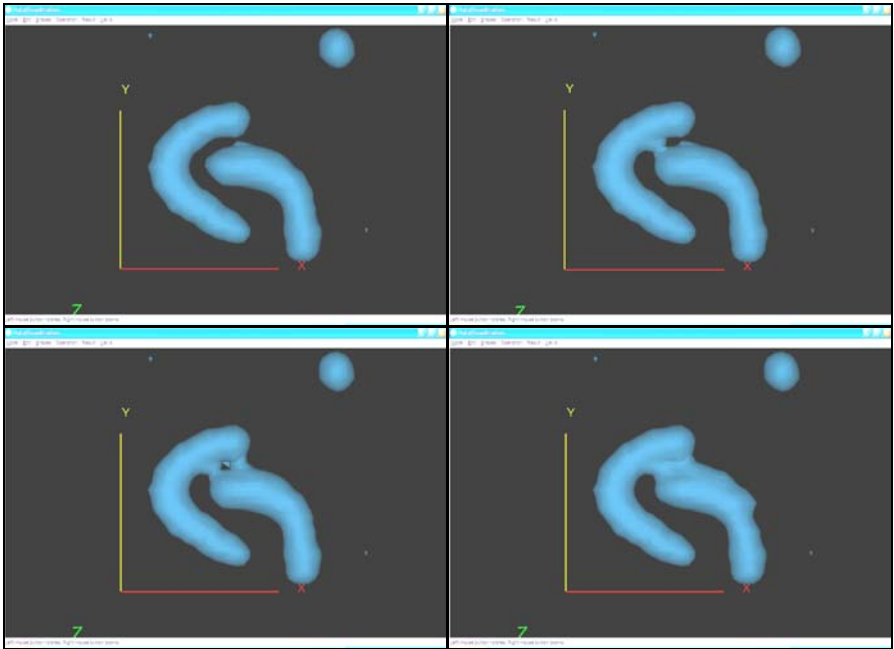
$G_3=G_1 \cap G_2$  of two objects  $G_1 \subset E^n$  and  $G_2 \subset E^n$  with the descriptive functions  $f_1$  and  $f_2$  will be defined as  $f_3=f_1 \wedge f_2 = \min(f_1, f_2) \geq 0$ , where  $G_3 \subset E^n$ . **Complement:**  $G_2 = \neg G_1$  of object  $G_1 \subset E^n$  with the descriptive functions  $f_1$  will be defined as  $f_2 = -f_1 \geq 0$ . **Difference:**  $G_3 = G_1 \setminus G_2$  between objects  $G_1 \subset E^n$  and  $G_2 \subset E^n$  with descriptive functions  $f_1$  and  $f_2$  will be defined as  $f_3 = f_1 \wedge (\neg f_2) = \min(f_1, -f_2) \geq 0$ , where  $G_3 \subset E^n$ . **Translation:**  $G_2 = T(G_1)$  of object  $G_1 \subset E^k$  with descriptive functions  $f_1$  by  $a_1, a_2, \dots, a_n$  will be defined as  $f_1(x_1 - a_1, x_2 - a_2, \dots, x_n - a_n) \geq 0$ . **Rotation:**  $G_2 = R(G_1)$  of object  $G_1 \subset E^k$  with descriptive functions  $f_1$  of angle  $\alpha$  about some axis will be defined as  $f_1(x'_1, x'_2, \dots, x'_n) \geq 0$  where  $[x'_1 \ x'_2 \dots \ x'_n \ 1] = R^{-1} [x_1 \ x_2 \dots \ x_n \ 1]$  and  $R^{-1}$  is an inverse matrix of rotation. **Scaling:**  $G_2 = S(G_1)$  of object  $G_1 \subset E^k$  with descriptive functions  $f_1$  in  $s_1, s_2, \dots, s_n$  times will be defined as  $f_1(x_1/s_1, x_2/s_2, \dots, x_n/s_n) \geq 0$ .

Thus, with this model we can query clusters changing over time with the final query shape.

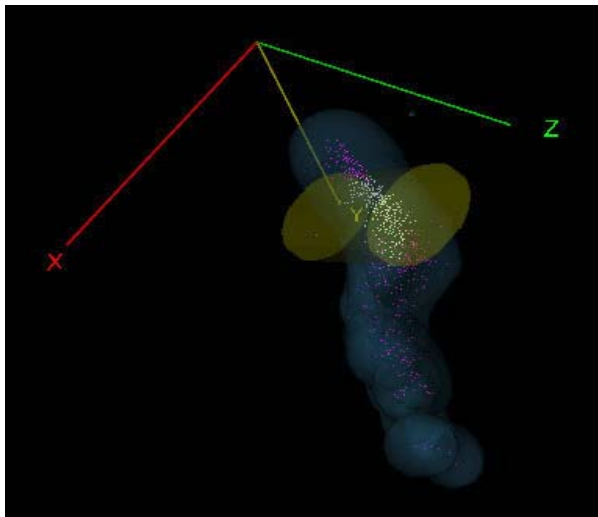
## 4 Visual Clustering and Querying

Our system for visual interactive 3D clustering and querying of spatio-temporal data is based on the uniform geometric model with implicit functions. Visual clustering allows the user to set interactively the appropriate parameters for clustering data changing over time. We developed the graphical user interface based on the geometric algebra. We use the geometric concepts to cluster and query spatio-temporal data and apply visualization techniques to interpret the clustering process and querying. The points mapped from the database and the clustering process both is visualized. To get initial clusters on 3-dimensional points clouds changing over time default parameters are used. After initial visual clustering, parameters can be changed at any time point. Cluster can be queried with a complex geometric query solid with union, intersection or other operations over primitive geometric solids. These primitive query solids currently are cuboids (box), cylinders, cones, and ellipsoid. We also can fit interactively any chosen cluster with the wrapping solid and keep the cluster implicit formulae in the database. In addition, any point belonging to the visualized solid can be located and identified in the database, data warehouse or file.

Let us consider examples of the visual 3-dimensional clustering of data changing over time. First, 3D projections of multidimensional points from database or file are visualized as clouds of points and can be viewed at any time point and/or interval. Then, the points are clustered visually with blobby functions and subdivision algorithm and can be viewed at any time point and/or interval as well. In Fig. 1, the visual clustering of spatio-temporal data is shown. In Fig. 2, an example of querying of time-dependent data is shown. First, 3-dimensional projection of multidimensional points is mapped from the database and visualized as point clouds. A blobby solid can be reconstructed at each time point to show shape of point clouds changing over time. Then, a solid query is posed and time interval is chosen. Here, a query solid is a cylinder that does not change its parameters and location over time interval. The result of the query is time-dependent data and is visualized as set of snapshots or as file with animation.



**Fig. 1.** 3D visual clustering of spatio-temporal data



**Fig. 2.** Querying of the cluster with cylinder shape

With the proposed query model, the user could specify a query solid for each time point defining time-dependent primitive solids parameters and location analytically or

by procedure. Currently, with the implemented GUI, the user constructs the query shape that does not change over time interval.

Geometric objects can be drawn opaque or transparent. We employ visualization techniques and advanced computer graphics algorithms for the implementation of the user interface.

In Fig. 3 wrapping with ellipsoids and union of ellipsoids is shown. The system is implemented with the software Visualization Toolkit (VTK) where visualization is implemented with marching cube algorithm [20].

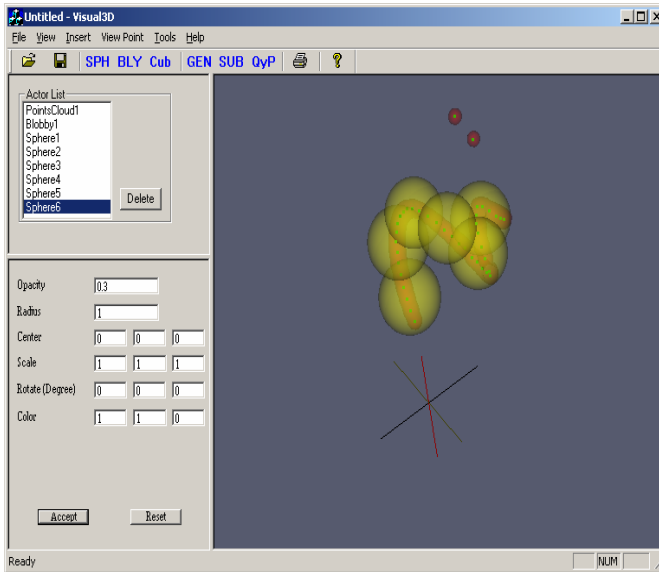


Fig. 3. Wrapping the cluster with union of ellipsoids

## 5 Conclusion and Future Work

In this paper, we introduced geometric approach to clustering and querying of time-dependent data in spatio-temporal databases, and data warehouses. We have presented visual interactive 3-dimensional method of clustering and querying spatio-temporal data. Visualization, clustering and querying are integrated in the system prototype. We conclude that our interactive method has a great potential for interactive data clustering and querying of data changing over time. The nature of our geometric model has the advantage of easy integration with visualization techniques. Thus, the user can be involved into the clustering and querying process in order to make more efficient and intuitive decisions on the data changing over time. The work completed by now mainly has focused on the testing of the method on small datasets. We are planning to improve and test our algorithms on large data changing over time and to design the system for large spatio-temporal databases and warehouses. In future, we also are planning to continue our research in the interactive geometric clustering

looking for the optimal parameters. We are going to make further improvements on the algorithms and visualization techniques to make the clustering and querying process more efficient and intuitive to the user. The human vision is the most experienced in the interpretation of realistic representations. The application of advanced computer graphics algorithms and visualization techniques for graphical data mining languages and representation of the data mining results could help to explore the data through more intuitive interface employing even modern VR tools.

## References

1. J. Hartigan, and M. Wong, "A K-means Clustering Algorithm", *Applied Statistics*, 28, 1979, pp. 100-108.
2. L. Kaufman, and P. Rousseeuw, *Finding Groups in Data: A Introduction to Cluster Analysis*. New York, John Wiley and Sons, 1990.
3. Sudipto Guha, R. Rastogi, K. Shim, *CURE: A Clustering Algorithm for large databases*. Technical report, Bell Laboratories, Murray Hill, 1997.
4. M. Ester, Hans-Peter Kriegel, Jorg Sander, Xiaowei. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", *Proc of KDD-1996*, 1996.
5. A. Hinneburg, D.A. Keim, An Efficient Approach to Clustering in Large Multimedia Databases with Noise, *American Association for Artificial Intelligence*, 1998.
6. U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyyh, From data mining to knowledge discovery: An Overview, In *Advances in Knowledge Discovery and Data Mining*, Cambridge, MA: MIT Press, 1996, pp.1-34.
7. N.J. Miller, and J. Han, Geographic data mining and knowledge discovery: An Overview, In *Geographic Data Mining and Knowledge Discovery*, London, New York: Taylor & Fransis, 2001, pp. 3-32.
8. D.A. Keim, "Information Visualization and Visual Data Mining", *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, 1, 2002, pp. 1-8.
9. T.C. Sprenger, M.H. Gross, A. Eggenberger, M. Kaufmann, A Framework for Physically-Based Information Visualization, in *Proceedings of Eurographics Workshop on Visualization '97*, Boulogne sur Mer, France, April 28-30, 1997, pp. 77-86.
10. T. C. Sprenger, R. Brunella, M. H. Gross. "H-BLOB: A Hierarchical Visual Clustering Method Using Implicit Surfaces," Department of Computer Science, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland
11. A. Hinneburg, D.A. Keim, and M. Wawryniuk, "HD-Eye: Visual Mining of High-Dimensional Data", *IEEE Computer Graphics and Applications*, September/October 1999, pp. 22-31.
12. J. Han, M. Kamber, *Data Mining Concepts and Techniques*. San Francisco, CA: Morgan Kaufmann Publishers, 2000.
13. R. H. Güting et al, A Foundation for Representing and Querying Moving Objects. *ACM Transaction on Database Systems*, Vol. 25, No. 1, March 2000, pp. 1-42.
14. M. Erwig, M. Schneider, Developments in Spatio-Temporal Query Languages, *IEEE Int. Workshop on Spatio-Temporal Data Models and Languages (STDML)*, 1999, pp. 441-449.
15. O. Sourina, S.H. Boey, Geometric Query Types for Data Retrieval in Relational Databases, *Data & Knowledge Engineering*, Elsevier Science B.V., Vol. 27, 2, 1998, pp. 207 – 229

16. O. Sourina, and L. Dongquan, "Geometric approach to clustering and querying in databases and warehouses", in *Proc. of Cyberworlds 2003*, Singapore, Dec. 2003, pp. 326-333.
17. Bloomenthal J., *An Introduction to Implicit Surfaces*, Morgan-Kaufmann, 1997.
18. O. Sourina, and D. Liu, Visual interactive 3-dimensional clustering with implicit functions, In *Proc. of IEEE CIS 2004*, Dec. 2004.
19. Ricci A., A constructive geometry for computer graphics, *The Computer Journal*, Vol. 16, 2, 1973, pp. 157-160.
20. Schroeder W., Martin K., Loresen B., *The Visualization Toolkit*, Prentice Hall, 1998.