

Visual Interactive 3-Dimensional Clustering with Implicit Functions

Olga Sourina, NTU

*School of Electrical & Electronic Engineering
Nanyang Technological University, Nanyang Avenue,
Singapore, 639798
eosourina@ntu.edu.sg*

Dongquan Liu, NTU

*School of Electrical & Electronic Engineering
Nanyang Technological University, Nanyang Ave
Singapore, 639798
pg03696626@ntu.edu.sg*

Abstract— Visualization techniques could enhance the current knowledge and data discovery methods by increasing the user involvement in the interactive process. In this paper, we propose a novel interactive clustering method based on geometric model with implicit functions and visualization techniques integrated in the GUI. First, visual clustering with blobby model allows the user to see the result of clustering on the screen and set the appropriate parameters interactively. After that, the user can get data of cluster in two ways. First method implies using solid-based subdivision algorithm. In the second method, the user needs to wrap the cluster he/she is interested in with geometric primitive solids that currently are cubes and/or spheres/ellipsoids. Geometric operations of union, intersection or subtraction can be performed over the geometric primitive solids to get the final wrapping shape. The user visually clusters the data and wraps the clusters with geometric shapes or even query clusters through graphics interface accessing dynamically 3-dimensional projections of multidimensional points from database or files.

Keywords—*data mining; visual clustering; implicit functions*

I. INTRODUCTION

Cluster analysis has been widely used in numerous applications. Numbers of clustering algorithms have been proposed in the last few years, e.g. partitioning method, hierarchical method, density-based method, etc. Some of them are well-established and proved to be successful in detecting certain cluster structures. Most of the existing methods are based on the following assumptions: the data sets should have the certain distribution or the number of clusters has to be known in advance. The traditional partitioning methods such as k-means [1] and k-medoids [2] face the problem to determine the number of clusters in advance. Moreover, the methods lack the ability to deal with the clusters of concave shape. On the other hand, hierarchical methods can detect clusters with irregular shapes, but the linkage methods in these algorithms predispose to find clusters with convex shapes. The improved hierarchical algorithms such as CURE [3] still cannot detect clusters with very complex shape because of the limitation of representation of clusters. The density-based algorithms such as DBSCAN [4] and DENCLUE [5] can find arbitrary shape clusters relatively efficient. But setting the parameters in DBSCAN could be a problem, and the efficiency of checking the connectivity between attractors in

DENCLUE could be improved. In the existed methods, cluster is usually defined as a collection of data objects that are similar to one another within the cluster and are dissimilar to the objects in other clusters [6]. This definition reflects the nature of cluster, but it does not describe characteristics of arbitrary shape cluster. The boundary of the cluster with the area inside the boundary could be used to describe cluster. In work [7], an extended definition of cluster was introduced. There, we proposed to define a cluster as a solid reconstructed on the points. 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. On the other hand, we intend to apply the clustering method on multi various data sets, where the shape of clusters can be arbitrary. Then, one of the clustering method requirements is setting parameters interactively to find clusters of arbitrary shapes. Moreover, in many application areas visualization of the data to be clustered is essential and helpful in further analysis. Therefore, an interactive clustering method with data visualization is desired.

The user of clustering algorithms is often involved in the clustering process. 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, 11] when incorporated with clustering algorithms could improve interpretability and usability of the data and clustering process. First, the user is interested in not only the results of clustering such as attributes of objects in the cluster, but also in the characteristics of cluster such as shape of cluster, and relationship between clusters. Second, since the data from different applications may be quite different in features, currently there is no clustering system that could deal with all possible types of data well. Therefore, the user should be involved into the clustering process to make it more efficient and accurate. Thus, 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 let the user help the system to make decisions or to set the values of parameters. For instance, the user could select the projection directions [10] 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 [11]. 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. As it was mentioned above, we extended the definition of cluster by adding the boundary of the cluster, and introduced a geometric model of the cluster as a solid. In this paper, we propose a new interactive 3D clustering method integrating visualization and clustering.

The aim of clustering is not only to find clusters and attributes of objects in the cluster, but finding the clusters of the certain shape could be even more valuable in some applications. If we find the formula of each cluster automatically or by wrapping the cluster, the search for similar shape clusters and comparison of clusters shapes could be possible [12].

Clustering is a classical problem in databases and warehousing. Development of query methods and graphical user interfaces is a new trend in data mining [6]. In this paper, we propose interactive visual 3-dimensional method based on the uniform geometric model of clustering and querying with implicit functions. With this model, we integrate visualization and clustering in one system. In work [13], the geometric query model with implicit functions was proposed and fully described.

Function based shape modeling (or modeling with implicit functions) is becoming increasingly popular in computer graphics [14, 15]. The idea that complex geometric objects could be produced from a "small formula" is applied in this work. Using the function representation, geometric solids are defined with the inequality $f(x,y,z) \geq 0$, where the function f is positive for the points inside the solid, equal to zero on its border and negative outside the solid. The operations are defined as functions superposition, and therefore the result of any operation is a functionally defined solid as well, which can be used as an argument to other operations. We built the geometric model by using function representation of solids. It provides the possibility of further analysis such as the comparison of shapes of clusters and querying of clusters.

Thus, we can describe our novel interactive clustering method based on the geometric model with implicit functions and visualization techniques as follows. First, visual clustering allows the user to see visual result of clustering on the screen and set the appropriate parameters. After that, the user can get data of cluster automatically with the solid-based subdivision algorithm or can identify or query the cluster by wrapping it with the query solid. The subdivision algorithm will automatically use the parameters that were set interactively at the stage of visual clustering with implicit functions. Another opportunity for the user in our interactive method is to wrap or query the cluster he/she is interested in with geometric shapes that currently are cubes and/or spheres/ellipsoids. Geometric operations of union, intersection or subtraction can be performed over the geometric primitive solids to get the final wrapping shape or query shape. The user visually clusters the data and wraps or queries the clusters with geometric shapes through graphics interface accessing dynamically 3-dimensional projections of multidimensional points from

database or files. Our visual interactive clustering method is based on the uniform geometric model with implicit functions.

The paper is organized as follows. Section 2 introduces the model defined with implicit functions that is used for interactive visual clustering and describes similarity of the model with the model of density-based methods. In Section 3, solid-based subdivision algorithm is described. The wrapping/querying method based on the geometric query model is described in Section 4. Implementation of the system and examples of clustering are discussed in Section 5. In Section 6, conclusion and future work are considered.

II. VISUAL CLUSTERING WITH IMPLICIT FUNCTIONS

The implicit modeling techniques are relatively new. This approach has become more sophisticated, generating new interest in computer graphics and related fields. It uses implicit function instead of parametric function or explicit function as its mathematical foundation.

Let \mathbf{P} be a set of multidimensional points $\mathbf{P} = \{[p_1, p_2, \dots, p_n]\}$ 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. For wrapping and querying we use function-based representation of the query solid. In solid-based subdivision algorithm, 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, metaballs or soft objects in Computer Graphics [13] 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.

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 a which can make cluster shape “thinner”. We have to set interactively three parameters for our model a – scale factor, b – exponential factor, and T – threshold value.

III. SOLID-BASED SUBDIVISION ALGORITHM

Now we present the solid-based subdivision algorithm which is based on the blobby model. In Figure 1, we can easily identify that there are six clusters in the dataset. The reason why we can recognize the clusters is that we can clearly ‘see’ or visualize the gaps between different clusters. Let us reconstruct a solid on points to give a boundary to every cluster as shown in Figure 1. The boundaries represented by curves isolate the six clusters from each other.

If we draw a straight line between two objects that belong to different clusters, there must be some intersections of line with the boundaries of the two clusters as it is shown in Figure 2. For simplicity, only one object is shown in each cluster. Let us denote the interval between points of the line-boundary intersection as m – n . Obviously, all points belonging to interval m – n do not belong to the clusters. In other words, we can conclude that object p and q do not belong to the same cluster. From Section 2, we know that, given a set of objects, we can find the potential (influence) at any point of the space by summing up the potentials at that point due to every object. By connecting all points with the same potential value (i.e. the threshold value), we can build a smooth implicit surface around the cluster.

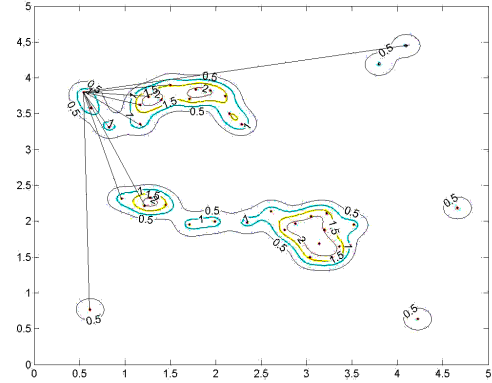


Figure 1. Sample Dataset 1 with cluster boundaries.

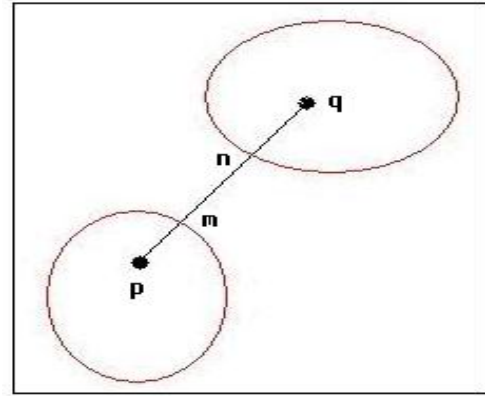


Figure 2. Boundaries and line intersection.

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 dataset with other points and sort the points into the clusters checking the values along the line connecting two points.

We have to note that the Sub-Divisional Algorithm can also act as a stand alone clustering algorithm even without being integrated into the system. It can discover arbitrary-shaped clusters in 2-dimensional and 3-dimensional spaces without visualization.

IV. VISUAL INTERACTIVE CLUSTERING USING WRAPPING

After we visualize clusters using interactively set parameters, we can get the objects belonging to one cluster using two different methods based on the uniform geometric

model. In this section, we describe our geometric query model consisting from geometric objects and operations. The model was introduced first in work [13]. 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.

Let P be a n -dimensional point, G_1 be a solid, bG_1 be a boundary of G_1 and iG_1 be an interior of G_1 . Then a point/solid classification is described by a 3-valued predicate:

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

We can consider our set of objects as multidimensional geometric points/objects, and generally multidimensional geometric primitive solids defined with functions $f(x_1, x_2, \dots, x_n) \geq 0$. The zero set of these functions provides surfaces, and the values that are greater or equal to zero define multidimensional geometric solid objects.

In our method, we wrap the clusters we are interested with cuboids that is an intersection of halfspaces and/or ellipsoids that are defined as follows:

Halfspace:

$$G_1: f_1(\mathbf{X}) = f_1(x_1, x_2, \dots, x_i, \dots, x_n) = (x_i - a) \geq 0$$

where a is some real number ($a \in R$).

Hyperellipsoid:

$$G_i: f_i(\mathbf{X}) = 1 - ((x_1 - x_{01})/a_1)^2 - ((x_2 - x_{02})/a_2)^2 - \dots - ((x_n - x_{0n})/a_n)^2 \geq 0$$

where $x_{0,1}, x_{0,2}, \dots, x_{0,n} \in R$ and $a_1, a_2, \dots, a_n \in R$.

We describe geometric operations as set-theoretic operations implemented in the form proposed by Ricci in work [16], where operations over implicit functions are considered. We apply them over primitive geometric solid objects to obtain complex geometric shapes.

Mathematically, **Union:** $G_3 = G_1 \cup G_2$ 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 $f_3 = f_1 \vee f_2 = \max(f_1, f_2) \geq 0$, where $G_3 \subset E_n$.

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 $f_2 = \neg f_1 \geq 0$. **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 $f_3 = f_1 \wedge (\neg f_2) = \min(f_1, \neg f_2) \geq 0$, where $G_3 \subset E_n$.

Thus, with this model we can wrap clusters with the final solid shape and/or query clusters.

V. IMPLEMENTATION AND RESULTS

Our system for visual interactive 3D clustering is based on the uniform geometric model with implicit functions. Visual clustering allows the user to set interactively the appropriate parameters for clustering. We developed the graphical user interface based on the geometric algebra. We use the geometric concepts to cluster and apply visualization techniques to interpret the clustering process. The points mapped from the database and the clustering process both is visualized. To get initial clusters on 3-dimensional points clouds default parameters are used. After initial visual clustering, appropriate parameters can be entered interactively using visual feedback. Cluster can be wrapped with a complex geometric query solid with union, intersection or other operations over primitive geometric solids. These primitive query solids currently are cuboids (box), and ellipsoid. We can fit any 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.

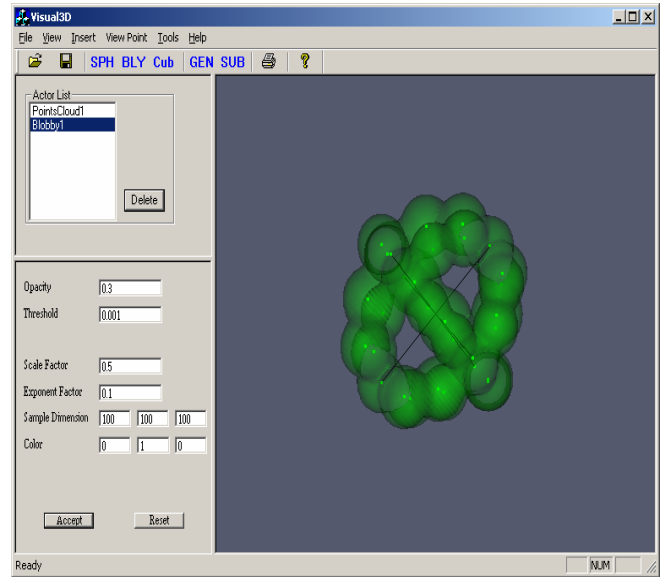


Figure 3. 3D visual clustering with uniform geometric model.

Let us consider examples of the visual 3-dimensional clustering. First, 3D projections of multidimensional points from database or file are visualized as clouds of points. Then, the points are clustered visually with blobby functions and subdivision algorithm. In Fig. 3, the result of visual clustering with parameters $T=0.001$, $a=0.5$, and $b=0.1$ is pictured. The user can also wrap the cluster with union of ellipsoids or just query clusters. In Fig. 4 and 5 wrapping with ellipsoids and union of ellipsoids is shown. Method of choosing projections for 3-dimensional clustering is out of the scope of the paper.

The system is implemented with the software Visualization Toolkit (VTK) where visualization is implemented with marching cube algorithm [17].

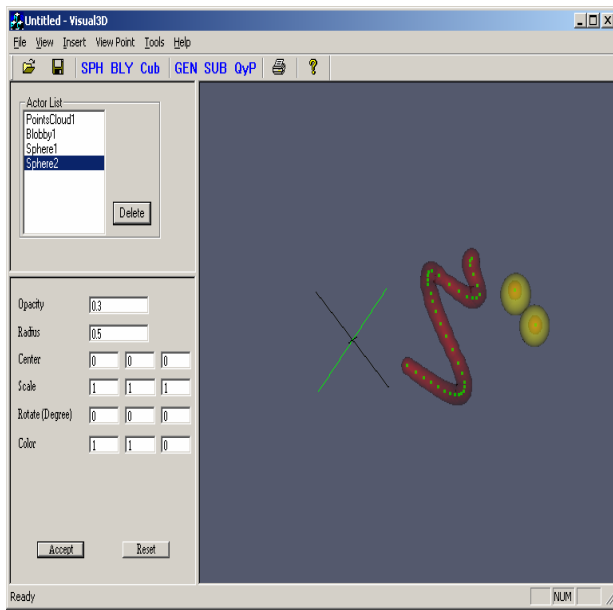


Figure 4. Wrapping the clusters with ellipsoids.

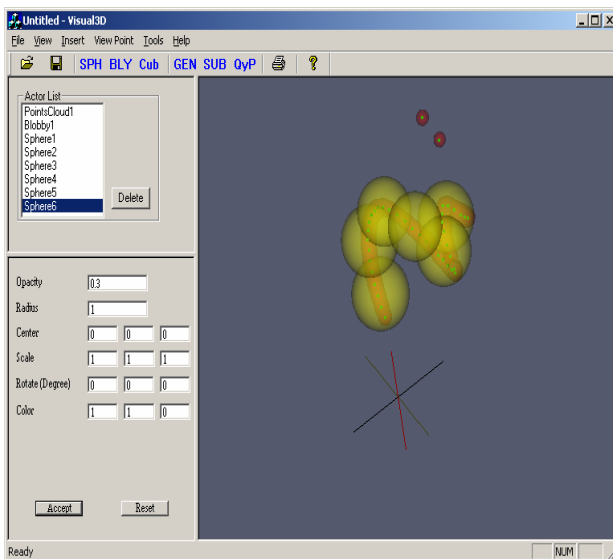


Figure 5. Wrapping the cluster with union of ellipsoids.

VI. CONCLUSION AND FUTURE WORK

We have presented visual interactive 3-dimensional clustering method based on the uniform geometric model with implicit functions. Visualization, clustering and querying are integrated in the system prototype. We conclude that our interactive method has a great potential for interactive data clustering. 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. Moreover, due to the rapid development of visualization and virtual reality techniques, the proposed

method can be implemented with VR interface. 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 datasets and to design the system for large databases. 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] J. Han, M. Kamber, *Data Mining Concepts and Techniques*. San Francisco, CA: Morgan Kaufmann Publishers, 2000.
- [7] 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.
- [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] 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.
- [11] 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
- [12] F.C. Langbein, B. Mills, A.D. Marshall, R.R. Martin, "Recognizing Geometric Patterns for Beautification of Reconstructed Solid Models", in *Proc. of Int. Conf. Shape Modeling and its Applications*, Italy, 2001.
- [13] Sourina O., Boey S.H., "Geometric Query Types for Data Retrieval in Relational Databases", *Data & Knowledge Engineering*, Elsevier Science B.V., Vol. 27, 2, 1998, pp. 207 - 229
- [14] Bloomenthal J., *An Introduction to Implicit Surfaces*, Morgan-Kaufmann, 1997.
- [15] G. Turk, Huong Quynh Dinh, J.F. O'Brien, "Implicit Surfaces that Interpolate", in *IEEE Int. Conf. Shape Modeling and its Applications*, Italy, 2001.
- [16] Ricci A., "A constructive geometry for computer graphics", *The Computer Journal*, Vol. 16, 2, 1973, pp. 157-160.
- [17] Schroeder W., Martin K., Loresen B., *The Visualization Toolkit*, Prentice Hall, 1998.