International Journal of Grid Distribution Computing
Vol.7, No.4 (2014), pp. 105-112
http://dx.doi.org/10.14257/ijgdc.2014.7.4.10

Reduction Strategy of Point Clouds to Reconstruct Surface Based on Fuzzy Clustering

Liu Yan-ju[1], Jiang Jin-gang[2], Tao Bai-rui[1], Zhang Hong-lie[3] and Liu Yan-zhong[3*]

[1]Computer Center, Qiqihar University, Qiqihar, China
[2]College of Mechanical & Power Engineering, Harbin University of Science and Technology, Harbin, China
[3]College of Computer and Control Engineering, Qiqihar University, Qiqihar, China
**15146692464@163.com*

Abstract

In order to remove redundant data and resolve conflicts in point clouds, we proposed fuzzy clustering reduction strategy in this paper. Original point clouds are decreased before computing other pretreatments. The proposed method involves three processes: reduction of the original data using fuzzy clustering while the point clouds are divided into sub-domains using octree structure, generation of the sub-surface that is fitted the sub-surface by implicit function in each sub-domain, the normal alignment that are computed normal of sub-surface and inference the global normal of surface using iteratively propagate algorithm. The method is suitable to reduce mass point clouds to reconstruct surface that can keep the property of surface. The experimental results show that the model with less sharp feature is more effective than complex model to reduce point clouds by fuzzy clustering.

Keywords: *point clouds; surface reconstruction; reduction strategy; fuzzy clustering; implicit function*

1. Introduction

In general, modern 3D laser scanning are capable of producing point clouds that contain millions of sample points in order to achieve the desired acquisition surface. This will lead to more computation and storage of CPU. How to reduce the original point clouds efficiently and obtain the same result using the fewest points have been many researchers' issues at present. Hamann [1] proposes a data reduction scheme for triangulated surfaces according to curvature of triangulated mesh. But it could be used for STL files only. Proportional reduction [2] and average distance reduction [3] are proposed, respectively. It is not reasonable that they use the original point clouds to generate triangular meshes and then compress the mesh. If the point clouds could be reduced directly before generating meshes, this will reduce the amount calculation of generating triangular meshes and subsequent calculations. In addition, these methods lose many details after processing and make the sharp feature smoothly. In this paper, a reduction strategy for point clouds is proposed which point clouds are reduced by fuzzy clustering method. Typical data with sharp feature are kept and others around these data are reduced in proportion during the octree subdivision. Their curvature and normal vector are computed in each unit. At last, the point data retained are reconstructed by propagate algorithm.

This paper is organized as follows. We firstly review the related work of surface reconstruction, normal estimation, and reduction of point clouds in Sections 2. In Section 3, we give reduction strategy of point clouds using fuzzy c-mean clustering the algorithm of reduction data and parameters of the FCM Algorithm. In Section 4,

ISSN: 2005-4262 IJGDC

calculation of the normal vector, and the procedure of surface reconstruction from point clouds without normal in detail. The results are illustrated feasibility of the proposed approach in Section 5. Conclusions are given at last.

2. Related Work

Reducing the complexity of point clouds is one of the key preprocessing techniques for subsequent algorithms. Dey *et al.* [4] presented a point cloud simplification algorithm, which detect redundancy in the input point cloud with the help of the “Cocone” [5] and local feature size concepts. The method requires the computation and maintenance of 3D Voronoi diagrams and tends to be computationally and memory expensive. Sareen *et al.* [6] introduced each extracted contour was redefined as a cubic B-spline curve by reduced number of control points defined by a user defined reduction ratio. Their algorithm takes a random initial subset of the input point clouds and uses its 3D Delaunay triangulation to define a signed distance function over the set. Song and Feng [7] associated simplification algorithm to reduce the number of data points scanned from a mechanical part, in which the boundary surfaces [8] often contain sharp edges. Even highly dense input point clouds may be simplified to unevenly distributed output point sets. In either case, re-sampling of the input clouds may be necessary to support any effective further processing. Kim *et al.* [9] adapted both geometry and color attributes to reduce point clouds for surface splitting. Their given geometric and color error-based iterative simplification method produces point clouds with low approximation error. However it is very sensitive in execution time to the size of the input point clouds. Many existing data simplification techniques are complex and not directly applicable to spline-based surface models.

Clustering techniques are mostly unsupervised methods that can be used to organize data into groups based on similarities among the individual data items. Most clustering algorithms do not rely on assumptions common to conventional statistical methods, such as the underlying statistical distribution of data, and therefore they are useful in situations where little prior knowledge exists. A fuzzy level set algorithm [10] is proposed to facilitate medical image segmentation. It is able to directly evolve from the initial segmentation by spatial fuzzy clustering. The controlling parameters of level set evolution are also estimated from the results of fuzzy clustering. Intuitionistic fuzzy C means clustering method [11] using intuitionistic fuzzy set theory. The intuitionistic fuzzy set theory considers another uncertainty parameter which is the hesitation degree that arises while defining the membership function and thus the cluster centers may converge to a desirable location than the cluster centers obtained using fuzzy C means algorithm. The potential of clustering algorithms to reveal the underlying structures in data can be exploited in a wide variety of applications, including classification, image processing, pattern recognition, modeling and identification.

Surface reconstruction continues to be an important application of computer vision. At present, MC, Delaunay triangulation, and implicit surface are three kinds of main algorithms. The interested readers are referred to [12] for a survey. The estimation of oriented normal field is fundamental to these works as oriented normal provide the first-order approximation and identify outside/inside of the underlying sharp. So it is very necessary and important to estimate reliable normal directly for point clouds. Hoppe *et al.* [13]developed MC algorithm, which estimate normal of each sampling point by calculating k nearest neighbors of sampling point [14] and then linear approximate for the section shape by these tangent planes. The normal of point is the eigenvector associate with the smallest eigenvalue. Gaussian weights are added to sample point’s neighbors [15] get more robust and accurate normal estimation using moving least squares (MLS) [16] in some conditions. Actually, these methods belong to low-pass filters and sharp features are smoothed out.

3. Reduction Strategy of Point Clouds using Fuzzy c-mean Clustering

Using FCM process, we can obtain *c*(<*n*) cluster centers. Then let *c* clusters be practical training data to reconstruct surface. In this sections, we assume that given a set of points *p={p₁, p₂, ..., pₙ}* ($p_i = [x_i, y_i, z_i]^T \in R^3$) that are scattered in 3D domain.

3.1. Reduce Point Clouds Framework during Partition of Unity

Consider the fuzzy C-Mean(FCM) clustering algorithm with following functions. First, it resolves conflicts in the original training data due to the reasonable distribution of the typical data. Second, it removes the redundant data from the original data, because there is not cluster centre around any redundant datum. Third, it improves the computation size by reducing the number of the training data. Now we summarized FCM as follows [17].

It minimizes the following objective function with respect to fuzzy memberships $\mu_{ki}(x_k)$ and cluster centers v_i:

$$J_s(U, v_1, v_2, ..., v_c) = \sum_{k=1}^{n}\sum_{i=1}^{c}(\mu_{ki})^s \|x_k - v_i\|^2 \tag{1}$$

where $\|x_k - v_i\|_a^2 = (x_k - v_i)A(x_k - v_i)^T$, A is a positive definite matrix, s($1 < s < \infty$) means a smoothing weight.

(a) Initialize parameters, c $(2 \le c < n)$ and s($1 < s < \infty$) and U(0) of fuzzy partitions randomly.

(b) Compute the cluster center v_i, for =1,…,c, by using U(t)

$$v_i = \sum_{k=1}^{n}(\mu_{ki})^s x_k \Big/ \sum_{k=1}^{n}(\mu_{ki})^s \tag{2}$$

(c) Update U(t) in the following: Calculate the sets I_k and I_k^* (k=1,2,…,n) as

$$I_k = \{i \mid 1 \le i \le c, d_{ki} = \|x_k - v_i\| = 0\} \tag{3}$$

$$I_k^* = \{1,2,...,c\} - I_k \tag{4}$$

if $I_k = \phi$ then $\mu_{ki} = 1 \Big/ \sum_{j=1}^{c}(d_{ki}/d_{kj})^{2/(s-1)}$ else

$$\begin{aligned} &\mu_{ki} = 0, \forall i \in I_k^*, \\ &\sum_{i \in I_k}\mu_{ki} = 1 \end{aligned} \tag{5}$$

(d) Repeat (b) and (c) until ||*U(t)-U(t-1)*||< ε. ε is a positive and small enough real number, which is a terminal criterion. The matrix norm is taken as [17]

$$\|U(t) - U(t+1)\| = \max_{k,i}\{|\mu_{ki}(t) - \mu_{ki}(t+1)|\} \tag{6}$$

3.2. Parameters of the FCM Algorithm

Before using the FCM algorithm, the following parameters must be specified: the number of clusters (*c*), the ‘fuzziness’ exponent (*m*), the termination tolerance (ε), and the norm inducing matrix (*A*). Moreover, the fuzzy partition matrix (*U*) must be initialized. The choices for these parameters [18] are now described one by one.

The number of clusters *c* is the most important parameter, in the sense that the remaining parameters have less influence on the resulting partition. When clustering real data without any a priori information about the structures in the data, one usually has to make assumptions about the number of underlying clusters. The chosen clustering algorithm then searches for *c* clusters, regardless of whether they are really present in the data or not.

Validity measures are scalar indices that assess the goodness of the obtained partition. Clustering algorithms generally aim at locating well separated and compact clusters. When the number of clusters is chosen equal to the number of groups that actually exist in the data, it can be expected that the clustering algorithm will identify them correctly. When this is not the case, misclassifications appear, and the clusters are not likely to be well separated and compact. Hence, most cluster validity measures are designed to quantify the separation and the compactness of the clusters.

$$\chi(Z;U,V)=\sum_{i=1}^{c}\sum_{k=1}^{N}\mu_{ik}^{m}\|z_k-v_i\|^2 \Big/ c\cdot\min_{i\neq j}(\|v_i-v_j\|^2) \tag{7}$$

It has been found to perform well in practice. This index can be interpreted as the ratio of the total within-group variance and the separation of the cluster centers.

The weighting exponent *m* is a rather important parameter as well, because it significantly influences the fuzziness of the resulting partition. As *m* approaches one from above, the partition becomes hard ($\mu_{ik}\in\{0,1\}$) and v_i are ordinary means of the clusters. As $m\to\infty$, the partition becomes completely fuzzy $\mu_{ik}=1/c$ and the cluster means are all equal to the mean of Z.

The FCM algorithm stops iterating when the norm of the difference between *U* in two successive iterations is smaller than the termination parameter ε. For the maximum norm $\max_{ik}(|\mu_{ik}^{(l)}-|\mu_{ik}^{(l-1)})$, the usual choice is $\varepsilon=0.001$, even though $\varepsilon=0.001$ works well in most cases, while drastically reducing the computing times.

The shape of the clusters is determined by the choice of the matrix *A*. *A* common choice is *A=I*, which gives the standard Euclidean norm:

$$D_{ik}^{2}=(z_k-v_i)^T(z_k-v_i) \tag{8}$$

4. Implicit Function Locally Fitting Framework

In 3D space, the polynomial term for thin-plate is $P(x)=m_0+m_1x+m_2y+m_3z$. The unique implicit function is found by solving for the weights w_i of the radial basis functions and for the coefficients *m*$_0$, *m*$_1$, *m*$_2$ and *m*$_3$ of *P(x)*. The unknowns are solved by constructing the following linear system formed by applying (9) to each constraint c_i. We use LU decomposition to solve this system in this paper.

4.1. Implicit Model for Point Clouds

For any implicit model, implicit function can be written as following:

$$f(X_i)=\sum_{i=1}^{n}w_i\phi(\|X_i-c_i\|)+p(X_i) \tag{9}$$

where X_i is point on the surface. *n* is the numbers of point clouds. c_i is added offset point in equation which is not on the surface. w_i is weight value of RBF at c_i. $\phi(\|\cdot\|)$ is basis function. $P(X_i)$ is a first-order linear polynomial.

$$p(X_i)=\lambda_0+\lambda_1x_i+\lambda_2y_i+\lambda_3z_i \tag{10}$$

where $P(X_i)$ is satisfied with orthogonal to weight coefficients', i. e. when $f(c_i)=h_i$ is known, there is the equation.

$$\begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1k} & 1 & c_1^x & c_1^y & c_1^z \\ \phi_{21} & \phi_{22} & \cdots & \phi_{2k} & 1 & c_2^x & c_2^y & c_2^z \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi_{k1} & \phi_{k2} & \cdots & \phi_{kk} & 1 & c_k^x & c_k^y & c_k^z \\ 1 & 1 & \cdots & 1 & 0 & 0 & 0 & 0 \\ c_1^x & c_2^x & \cdots & c_k^x & 0 & 0 & 0 & 0 \\ c_1^y & c_2^y & \cdots & c_k^y & 0 & 0 & 0 & 0 \\ c_1^z & c_2^z & \cdots & c_k^z & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_k \\ \lambda_0 \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_k \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{11}$$

From equation (3), only find offset points set and $\phi(\|\cdot\|)$ can realize the implicit surface reconstruction. The following is to determine the basis function $\phi(\|\cdot\|)$ and offset point c_i.

4.2. Stitch the Local Surface Framework by Advancing Front Algorithm

After the data reduction and partition are completed and fitted local surface, we obtain several sub-domain data sets Ω_i and sub-surfaces f_i (x, y, z). The normal of implicit sub-surface at the node p_i, ($p_i \in \Omega_i$) can be computed, but these normal vectors are unaligned. In the section, we inference orientation of normal and stitch local surface by advancing front algorithm at the same time.

We choose to infer the normal locally and iteratively propagate the partially determined results over the entire model to obtain the global solution, which its procedure for repeated alignment is similar to [19]. The difference is the selection of the initial point that is checked by an energy function to match between neighboring local surfaces. The maximum coordinate component of the point is defined the initial point in this paper. If the *z* coordinate component of p_i is the maximum, its normal vector matches the *z* axis after comparing. In this way, the local surface including p_i is defined the initial surface with the normal vector and its orientation, and then propagate its neighbors surface. The alignment procedure in detail can be summarized as algorithm 1.

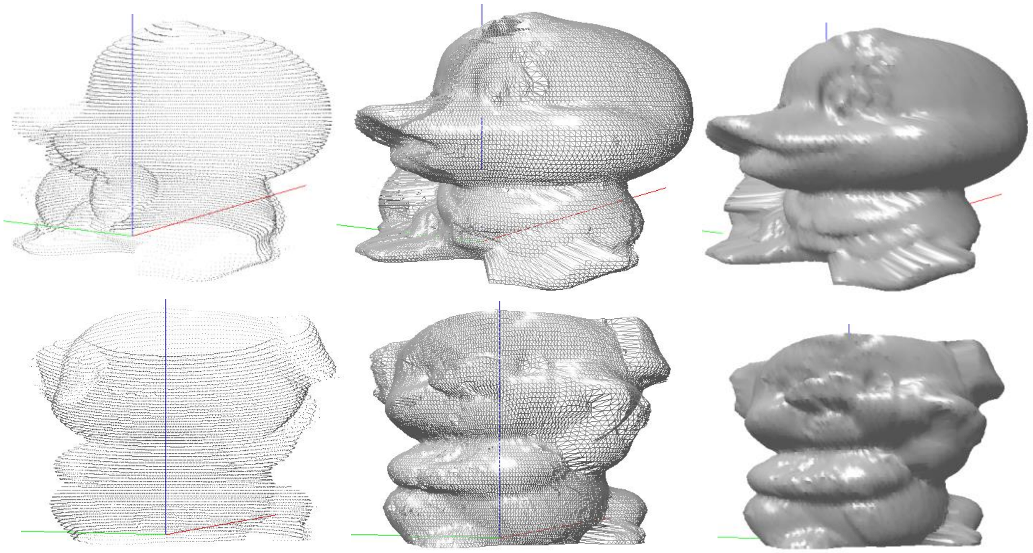
```
Algorithm 1: Inference orientation of normal and stitch local surfaces algorithm
Step 1. Compare all coordinate components of points and find the maximum;
Step 2. Put the local surface $\Omega_i$ including maximum point into active list;
Step 3. Mark other local surfaces as 0 and put into wait list F;
Step 4. while  F is not empty
            $\Omega_{active}$=F.front
            Construct graph G with $\Omega_{active}$
            Execute BP inference on G
            If $n_{active} \cdot n_i \neq +1$ Update normal of the nodes G
            Find and insert adjacent grids of G into F
Step 5. End
```

Initially, find the maximum coordinate component in the set of reduced points by comparison. The active list contains the maximum point unit only. All the local surfaces are marked as 0 means unaligned except the initial surface. During each processing, the first unit of the active list is selected to build a local graph G. We search the unit labeled 0 for the four neighbors of Ω_{active} and link Ω_{active} with them and marked 1 means aligned. The algorithm marches iteratively by an active octree unit Ω_{active} to traverse the unaligned regions of the model until all the sub-surfaces have been properly normal. The unit adjacent to a graph node are activated and inserted into the active list. To complete this inference procedure, the consistently sub implicit patches are then smoothly blended to form the overall approximate surface G at the same time.

5. Results

The purpose of the experiments is to evaluate the performance of our proposed framework by reconstructing three models. Experimental results present in this section are generated on a PC equipped with an Intel Core 2 processor at 2.93GHz and 2GB main memory. The input data set is from 3D laser scanner, PICZA LPX-250. In the experiment, the models of rabbit, spider, and head are selected. The scanned numbers of the models are showed in Table 1 and their stereo pictures of point clouds can be seen in Figure 1(a). However, the greater the amount of data is, the more computation is. It is necessary to reduce the data before computation, especially for the simple model.

According to the reduction strategy in this paper, the amount of data for two models is reduced. After comparing, $\lambda = 0.0003$ could obtain the best result. Half of data is cut in one half on the whole. Due to the model of duck is not sharp, data have been decreased by almost two-thirds. On the contrary, the models of pig are more complex and its data are decreased one-second. In particular, the face of pig model is more sharply, surface reconstruction can not fit the desired accuracy if the data are decreased in this area. We only reduced the data of face range according to the value of λ. In Figure 1 (b), the reduced model of point clouds can be seen. The amount of data and the time of data reducing are shown in Table 1 too.



a. Original point b. Reduction point c. Surface reconstruction

Figure 1. Duck and Toy Pig Model of Point Clouds

Table 1. Comparison between Original Data and Reduced Data

Model	Original Data Size	Reduced Data Size	Time to Reduce Data
Duck	35647	11216	2.32
Toy pig	171875	85674	4.84

Local variational implicit surfaces are computed and refined until desired accuracy 6×10^{-4} is achieved. From Table 2, it takes longer time to compute normal and reconstruct surface using original data than to do that using reduced data. For the simple model such as duck, its time of reconstruction using reduced data is reduced two-thirds due to simple models contains less data. While the model of toy pig is more complex than duck, they contain more data. Although the surface reconstruction using reduced data costs one-second time than that using original data, we spend some time to reduce data. In order to keep the model accuracy, less data is decreased. Figure 1(c) illustrates the capability of the proposed approach and reconstruct surfaces from point clouds without normal efficiently. Table 2 summarizes the computation time of three models evaluated.

Table 2. Computational Time (Seconds)

Model	Original data		Reduced data	
	Normal computing	Global Surface Adjusting	Normal computing	Global Surface Adjusting
Duck	6.45	5.29	2.26	1.17
Toy pig	13.76	10.14	6.31	5.73

6. Conclusions

A novel reduction strategy from point clouds is proposed. Two key advantages of the scheme are the reduction algorithm for original data and the normal of surface obtained by iteratively propagate algorithm. While the global point clouds are divided into several smaller sub-domains, the original data are decreased by fuzzy c-mean clustering at the same time. So we can simplify the problem of reconstruction and save much computation time and storage of CPU. We fit the sub-surface by implicit function and compute the normal of sub-surface in each sub-domain. Finally, the overall surface is formed after repeatedly inferring normal among the sub-surfaces. The approach is more obvious to the relatively simple model than the complex model. The future work should be improved the reduction algorithm for the complex model and the inference algorithm of normal orientation.

Acknowledgment

This work was supported by National Natural Science Foundation of China No. 51205093 and Natural Science Foundation of Heilongjiang Province of China No. F201336 and the Education Department of Heilongjiang Province of China, No. 12531765.

References

[1] B. Hamann, “A Data Reducation Scheme for Triangulated Surfaces [J]”, Computer Aided Geometric Design, vol. 11, no. 2, (**2004**), pp. 197-214.

[2] M. Pauly, R. Keiser, L. P. Kobbelt and M. Gross, “Shape Modeling with Point-sampled Geometry [J]”, ACM Trans. Graph., vol. 22, no. 3, (**2003**), pp. 641-650.

[3] J. H. Park, Y. Park and Y. Han, *et al.* “Effect of Outer Surface Sealing Treatment on the Reduction of Surface Check Occurrence During the Drying of Center-Bored Round Timber [J]”, vol. 32, no. 2, (**2014**), pp. 236-243.

[4] T. K. Dey, J. Giesen and J. Hudso, “Decimating Samples for Mesh Simplification [C]”, In Proc. 13th Canadian Conf. on Computational Geometry, Waterloo, Canada, (**2011**), pp. 85–88.

[5] T. K. Dey, R. Dyer and L. Wang, “Localized Cocone Surface Reconstruction [J]”, Computer and Graphics, vol. 35, (**2011**), pp. 483-491.

[6] K. K. Sareen, G. K. Knopf and R. Canas, “Contour-based 3D Point Cloud Simplification for Modeling Freeform Surfaces [C]”, Science and Technology for Humanity (TIC-STH), IEEE Toronto International Conference, (**2009**), pp. 381 – 386.

[7] H. Song and H. Feng, “A Progressive Point Cloud Simplification Algorithm with Preserved Sharp Edge Data [J]”, The International Journal of Advanced Manufacturing Technology, vol. 45, no. 5-6, (**2009**), pp. 583-592.

[8] C. Chen and K. Cheng, “A Sharpness-dependent Filter for Recovering Sharp Features in Repaired 3D Mesh Models [J]”, IEEE Transactions on Visualization and Computer Graphics, vol. 14, no. 1, (**2008)**, pp. 200-212.

[9] D. B. Kim, R. Pajarola and K. H. Lee, “Efficient Reduction of Point Data Sets for Surface Splatting Using Geometry and Color Attributes [C]”, The International Journal of Advanced Manufacturing Technology, Online First™, vol. 12, (**2011**), November.

[10] B. N. Li, C. K. Chui, S. Chang and S. H. Ong, “Integrating spatial fuzzy clustering with level set methods for automated medical image segmentation [J]”, Computers in Biology and Medicine, vol. 41, no. 1, (**2011**), pp. 1-10.

[11] T. Chaira, “A novel intuitionistic fuzzy C means clustering algorithm and its application to medical images [J]”, Applied Soft Computing, vol. 11, no. 2, (**2011**), pp. 1711-1717.

[12] Y. Zhang, Y. Liu and X. Dai, “Categories and New Developments of Three-dimensional Reconstruction Using Triangulation Mesh [J]”, ICIC Express Letters, vol. 5, no. 1, (**2011**), pp. 95-100.

[13] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald and W. Stuetzle, “Surface Reconstruction from Unorganized Points [C]”, In Proceedings of ACM SIGGRAPH, (**1992)**, pp. 71-78.

[14] G. M. Nielson, L. Zhang, K. Lee and A. Huang, “Spherical Parameterization of Marching Cubes IsoSurfaces based upon Nearest Neighbor Coordinates [J]”, Journal of Computer Science and Technology, vol. 24, no. 1, (**2009**), pp. 30-38.

[15] P. Stelldinger, L. J. Latecki and M. Siqueira, “Topological Equivalence between A 3D Object and the Reconstruction of Its Digital Image [J]”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, (**2007**), pp. 126-140.

[16] G. Xue and Y. Zhang, “A Parallel Backprojection Algorithm for Spiral Cone-Beam CT Exact Reconstruction [J]”, JDCTA: International Journal of Digital Content Technology and its Applications, vol. 4, no. 6, (**2010**), pp. 134-143.

[17] L. Bai, J. Liang, C. Dang and F. Cao, “A novel fuzzy clustering algorithm with between-cluster information for categorical data [J]”, Fuzzy Sets and Systems, vol. 215, no. 16, (**2013**), pp. 55-73.

[18] H. Izakian and W. Pedrycz, “Agreement-based fuzzy C-means for clustering data with blocks of features [J]”, Neurocomputing, vol. 127, no. 15, (**2014**), pp. 266-280.

[19] G. Dziuk and C. M. Elliott, “An Eulerian Approach to Transport and Diffusion on Evolving Implicit Surfaces [J]”, Computer Visual Science, vol. 13, (**2010**), pp. 17–28.