The 19th Conference of Mechanical Engineering Network of Thailand
19-21 October 2005, Phuket, Thailand

Two-Level Adaptive Hierarchical Clustering Algorithm from Unorganized Points for Triangular Mesh

Viboon Sangveraphunsiri and Kiattisak Sritrakulchai
Robotics and Advanced Manufacturing Laboratory, Department of Mechanical Engineering
Chulalongkorn University
254 Phayathai Rd. Prathumwan, Bangkok 10330, Thailand
Tel. 66-2218-6448, 66-2218-6585 Fax 66-2218-6583
E-mail: Viboon.S@eng.chula.ac.th, reverserp@eng.chula.ac.th

Abstract

In this paper, we propose a new neural network based on our originated two-level adaptive hierarchical clustering algorithm. The algorithm is to manage and organize 3D point cloud for properly creating triangular mesh. Our algorithm is unsupervised, stable and flexible. Only one parameter, triangular edge length, is needed in the neural network. This helps to overcome the inconsistency and unstable of unorganized a cloud of points. The proposed two-level consists of the first level for clustering the cloud of points that has same order of limited length into a same cluster and the second level for clustering points available to generate triangular surface model or drape surfaces over the points of the same cluster. From the experimental result, it can be shown that the proposed method is very effective for clustering unorganized point clouds for generating triangular mesh.

Keywords: Unorganized Points / Two-Level / Hierarchical / Clustering Algorithm

1. Introduction

The hierarchical structured knowledge system using autonomous learning has received much attention, for examples [1-2]. By implementing an autonomous clustering process, we can rapidly organize the data space to overcoming the inconsistency and innate redundancy of human-generated abstractions. The hierarchical structure generated may then be used to discover trends and generalization information, to allow knowledge to be stored at the requisite level of granularity, and for further processing in the system. The clustering algorithms have been investigated previously and covered a vast range of methodologies. Jain and Dubes [3] discuss about the statistical approaches, or Song and Lee [1] and Hodge and Austin [2] discuss about the neural network techniques, used in a wide variety of applications, including pattern recognition, information retrieval, image processing, and natural language processing.

In this paper, we develop a new neural network based on two-level adaptive hierarchical clustering algorithm as compare to Hodge and Austin [2] that the multi-level is generated every time a change of network structure occurs and the GCS network (Growing Cell Structure) impose the strict network-topology preservation rule which can result in massive purges in the GCS network, causing much of the accumulated learning to be lost [4]. Our technique is a unique way for reverse engineering application when no CAD-data of the component available. Reverse engineering can be utilized as a design tool for developing creative product, designed by the stylist, for copying a model, and for quality control of product as well. Typical steps include scanning 3D-dimensional objects such as clay, wood or an existed physical object and generating CAD representation for CAD/CAM/CAE systems. Currently, a large cloud data points can be efficiently measured using the optical 3D measurement systems or automatic laser scanning system. The cloud of points is normally unorganized.

There are several works [5-13] proposed the technique to overcome the difficulty involved in the problem of surface reconstruction of unorganized points. They tried to correct the connectivity among the sampled points. The correct connectivity can give us a reconstructed surface mesh that faithfully represent the shape and topology of the original object from which the set of sample points were drawn. We will show that the algorithm can realize endless learning and automatic two-level clustering. Then, we call this algorithm the two-level adaptive hierarchical clustering algorithm.

2. Data Management System

The authors have designed a new algorithm, the two-level adaptive hierarchical clustering algorithm, which is the extended concept base on [2], [4], and [14], to store the representation of mapping point cloud to triangular mesh depending upon the distribution of the data in locality, to restrict data searching within the hierarchy tree for minimizing the search space and to retrieve data needed by using the word as indexing to access relevant data. As shown in Figure 1, the data recorded such as, point data prepared for creating triangular mesh, the triangular mesh structure data, and connectivity data among triangles, which are neighborhood of each triangle, are clustered, organized and recognized for effectively generating triangular mesh from unorganized cloud of points.

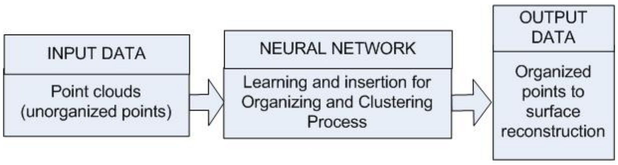

Figure 1: Overall data flow process of novel technique.

We need to define some index symbols to help clustering, organizing, and recognizing data, so that we can implement more efficiently on reduced representations of the data and recorded database of data relation, the topology of the input point clouds, for creating triangular mesh.

M : Index set of all point data, $P_i \in M$; i =1,2, … , n, thought here as a probability distribution in $\mathbb{R}^3$ is used for generating mesh.

SM_m: The set of the partition of point data, subdivided from all point data. For the first level clustering, distance similarity will be used for generating partition of mesh at index m ; m =1,2, … , n .

$NC_{c,m}$: The sub-set of the partition of point data at index c of second level clustering and c =1,2, … , n. The $NC_{c,m}$ is the child at index c of the parent SM_m (partition of mesh at m).

$C_{v,c,m}$: The member of sub-set $NC_{c,m}$ at index v. $C_{v,c,m}$ will be called cell or member of $NC_{c,m}$ at v. v can be number 1, 2, …, n, where v indicated the point number in the point cloud. See example shown in the Figure 2.

$C_i = \{P_i\}$; $P_i \in M$, i =1,2, … , n. C_i is the coordinate of input point i .

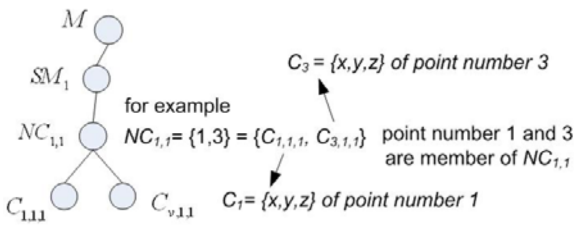

Figure 2: The symbols of index data

In addition to the symbol specified above, the parameter needed to start the searching which is specified by user is the limited length of similarity. The limited length of similarity, ρ, represents the limited distance among points used in the clustering process.

After the clustering process, the set of points which have similarity distribution will be clustered into the same partition which will be represented by n mutually disjoint subsets or clusters in the first level, that is $SM_1, SM_2,..., SM_m,...SM_n$. Each SM_m is divided into NC clusters or second level, $NC_1, NC_2,..., NC_c,...NC_n$. The triangular mesh will be generated from these NC clusters.

In conclusion, the organizing and clustering process, $\Phi(P_i)$, will perform a mapping sampled point (M) that associates a cluster of similarity of point data, P_i ; $i = 1, 2, \cdots, n$, into two-level hierarchical cluster in Figure 3, $\Phi(P_i): M \to SM \to NC$.

Where SM , level-one cluster, is a partition of M . Then, SM is a collection of nonempty sets, $\{SM_1, SM_2,..., SM_n\}$, such that, $M = SM_1 \cup SM_2 \cup \ldots \cup SM_n$, and sets, $SM_1, SM_2,..., SM_n$ are mutually disjoint, for all $m,\ l = 1, 2, \ldots, n,$ and $SM_m \cap SM_l = 0$, whenever $m \neq l$. The NC, level-two cluster, is the partition of SM_m. Then, NC is a collection of nonempty sets, $\{NC_1, NC_2,..., NC_n\}$, such that, $SM_m = NC_1 \cup NC_2 \cup \ldots \cup NC_n$, and sub-sets, $NC_1, NC_2,..., NC_n$ are mutually disjoint, for all $c,\ d = 1, 2, \ldots, n,$ and $NC_c \cap NC_d = 0$, whenever $c \neq d$.

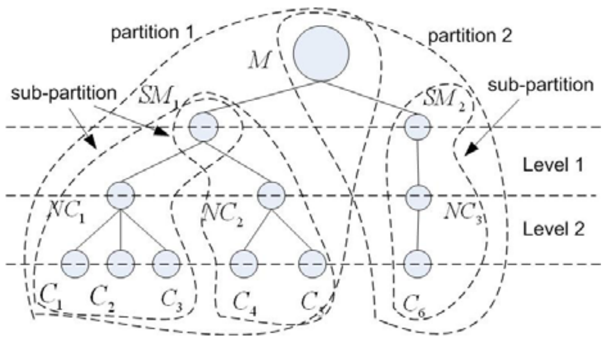

Figure 3: Subdivision into two levels

3. Organizing and Clustering Process $\Phi(P_i)$

Figure 4 illustrates the organizing and clustering process. The network starts from the typical or basic neural network, as shown in the Figure 4(a), which will consist of 4 nodes as point cloud (M) , a sub-partition (SM_1) of point cloud, a sub-partition $(NC_{1,1})$ of SM_1 and a cell which contain the point that passed through the process.

Then, we apply the distance similarity equation for searching, $(D_v)_{c,m} = \left\| (C_v)_{c,m} - C_i \right\|$ Where $\| \,.\, \|$ denotes the Euclidean vector norm. So, $(D_v)_{c,m}$ is a set

$\{d_1, d_2, \cdots, d_v\}_{c,m}$, the distance between point data, $(C_v)_{c,m}$, (cell v of sub-set partition index c of set partition index m) and new sampled point data of cell C_i which is the 3D coordinate of input point data (P_i) as shown in Figure 4(b).

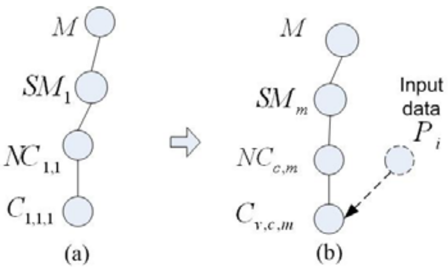

Figure 4: Basic neural network

The distance in the set $(D_v)_{c,m}$ is used for clustering by comparing to the limited length, ρ. In this clustering process, there are four possible cases or events of learning and insertion for every input point data as indicated in Figure 5. We will define the distance set L_g, $\{d_1, d_2, \cdots, d_g\}$; where the distance (d_g) is less than ρ. The number of members in the set $(L_g)_{c,m}$ will indicate the event that will occur. For example, if $g=1$ and the number of members in set $NC_{c,m}$ is equal to 1, it is indicated that case 1 will occur. If $g>1$, case 4 will occur. If $g=0$, case 2 will occur. If $g=1$ and the number of members in set $NC_{c,m}$ greater than 1, this indicate that case 3 will occur.

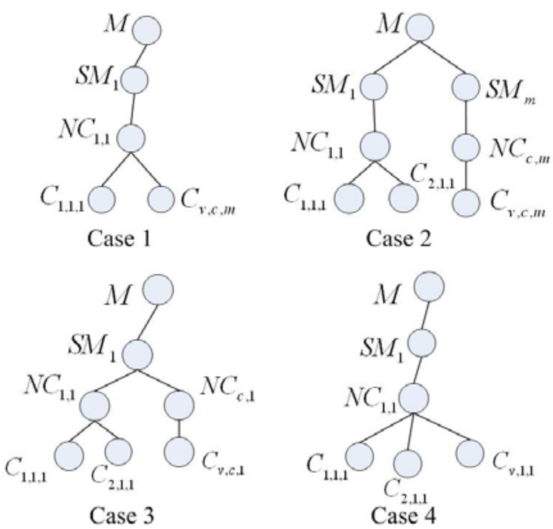

Figure 5: Learning and insertion of the four events for new fed input point data (P_i).

4. Results

To demonstrate our algorithm, we develop application modules using Python. Python is an object oriented script language. The main advantage of using Python is it provides us the ability to interface to other languages. Python can also implement in various operating system. Our tested platform is a Pentium III 700 MHz running Linux operating system. For post processor, we are using the Blender, a freeware, for graphic display of triangular mesh. We also developed our own post-process for graphic display [15]. The point clouds, using in the experimentation, were obtained by using both the commercial optical 3D-measurement systems and our developed automatic laser scanning machine [16].

The tested point cloud consists of 61 points as shown in Figure 6. We impose the limited length of similarity, ρ =5. The result of emulated triangular mesh is shown in Figure 7. The three partition meshes (SM) are created. Where SM_1 has 20 points and 17 triangular mesh. SM_2 and SM_3 have 3 points and 1 triangular mesh. Figure 7 can also be confirmed by visual inspection of nearly distribution of point data creating triangle.

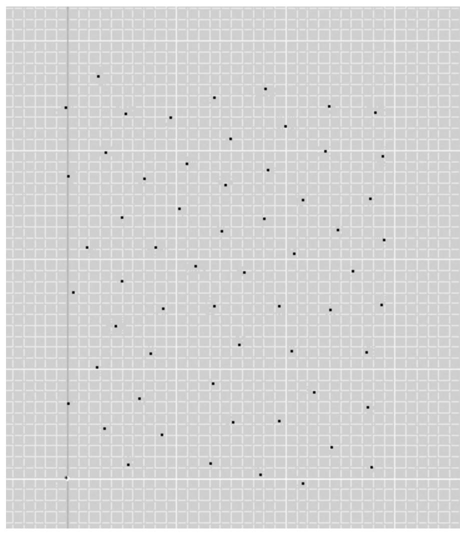

Figure 6: Point data.

We can obtain better triangular mesh by increase the limited length up to a certain number. Figure 8, 9 and 10 are the results of triangular mesh created by specifying $\rho=6$, $\rho=10$ and $\rho=40$, respectively.

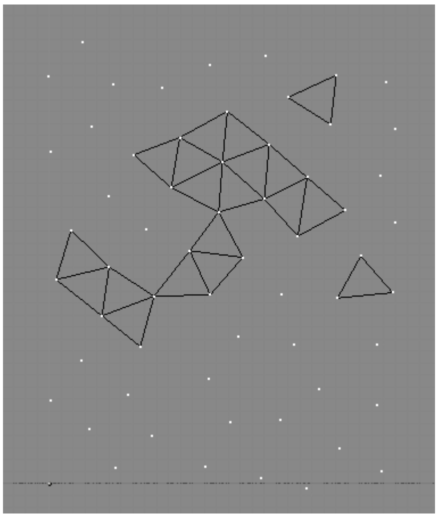

Figure 7: Point data for creating triangular mesh, $\rho = 5$.

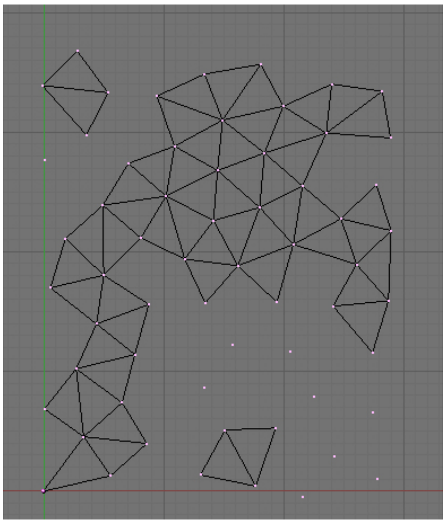

Figure 8: Point data for creating triangular mesh, $\rho = 6$.

As shown in Figure 8, they still have three partitions in M. The biggest partition has 44 points and 48 triangles. The other two partitions have 4 points and 2 triangles. As $\rho = 10$ and $\rho = 40$, shown in Figure 9 and 10, both cases have only one partition but differently the number of triangles. Obviously, the quality of the triangular mesh in both cases is different. Figure 9 shows the better mesh quality compared to mesh quality of Figure 10. Figure 10, the limited length is specified larger than requirement. Consequently, the several undesirable triangles are created near the boundary of surface.

Obviously, this algorithm can realize endless learning and automatic two-level hierarchical clustering. Then, we call this algorithm the two-level adaptive hierarchical clustering algorithm.

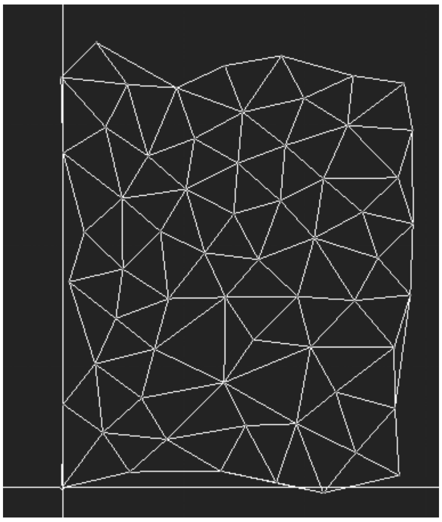

Figure 9: Point data for creating triangular mesh, $\rho = 10$.

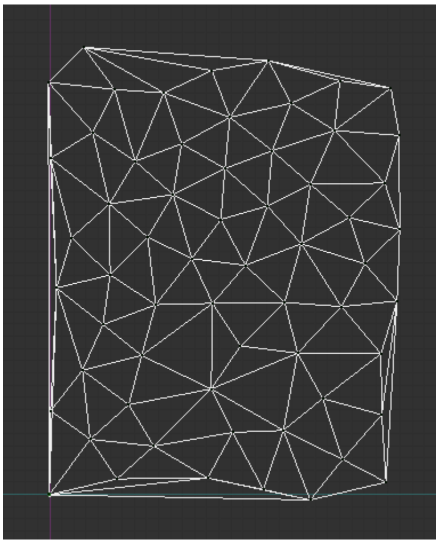

Figure 10: Point data for creating triangular mesh, $\rho = 40$.

5. Conclusion

we presented the two-level adaptive hierarchical clustering algorithm. The algorithm is to manage and organize 3D point cloud for properly creating triangular mesh. Our algorithm is unsupervised, stable and flexible. Only one parameter, the length of edge of triangle or limited length, is needed in the neural network.

The experimental results show that, our algorithm works satisfactorily for unorganized point clouds in many different situations by specifying limited length parameter. As shown in Figure 12, it reveals that our algorithm has the flexibility and stability. We could use the limited length, ρ, more than suitable value , no effect happened in organization of all data points.

References

[1] Song, H., and Lee, S., “A Self-Organizing Neural Tree for Large Set Pattern Classification,” IEEE Trans. Neural Networks, Vol. 9, no.3, 1998.

[2] Hodge, V., and Austin, J., “Hierarchical Growing Cell Structures: TreeGCS,” IEEE Transactions on Knowledge and Data Engineering, Vol. 13, No. 2, 2001, pp. 207-218.

[3] Jain, A., and Dubes, R., “Algorithms for Clustering Data,” Englewood Cliffs, N.J., Prentice Hall, 1998.

[4] Burzevski, V., Mohan, C. K., “Hierarchical Growing Cell Structures,” IEEE International Conference, Neural Network, Vol 3, 1996, pp 1658-1663.

[5] Ivrissimtzis, I., Jeong, W., and Seidel, H., “Using Growing Cell Structures for Surface Reconstruction,” Proceedings of the Shape Modeling International, 2003, pp. 78-86.

[6] Yu, Y., “Surface Reconstruction from Unorganized Points Using Self-Organizing Neural Networks,” IEEE Visualization 99 Proceedings Conference, 1999, pp. 61-64.

[7] Hoppe, H., Derose, T., Duchamp, T., McDonald, J., and Stuetzle, W., “Mesh Optimization,” SIGGRAPH’93 Proceedings Conference, 1993, pp.19-26.

[8] Hoppe, H., Derose, T., Duchamp, T., McDonald, J., and Stuetzle, W., “Surface Reconstruction from Unorganized Points,” SIGGRAPH’92 Proceedings Conference, 1992, pp. 71-78.

[9] Amenta, N., Bern, M., and Kamvysselis, M., “A New Voronoi-Based Surface Reconstruction Algorithm,” SIGGRAPH’98 Proceedings Conference, 1998, pp. 415-421.

[10] Fritzke, B., “Growing Self-Organizing Networks – why?,” ESANN’96: European Symposium on Artificial Neural Networks, 1996, pp. 61-72.

[11] Fritzke, B., “Growing Self-Organizing Network for Unsupervised and Supervised Learning,” Technical Report ICSTR-93-026, International Computer Science Institute, Berkeley, 1993.

[12] Kawahara, S., and Saito, T., “On A Novel Adaptive Self-Organizing Network,” CNNA’96, Fourth IEEE International Workshop on Cellular Neural Networks and Their Applications, 1996, pp. 41-46.

[13] Fritzke, B., “Kohonen Feature Maps and Growing Cell Structures – A Performance Comparison,” Advances in Neural Information Processing Systems-5-(NIPS-92), C.L. Giles, S.J. Hanson, and J.D. Cowan, eds, 1993.

[14] Azuaje, F., Dubitzky, W., Black, N., and Adamson, K., “Discovering Relevance Knowledge in Data: A Growing Cell Structures Approach,” Systems, Man and Cybernetics, Part B, IEEE Transactions, Vol. 30, Issue 3, 2000, pp. 448-160.

[15] Sangveraphunsiri, V. and Uttamang, K., “Development of A 3-D Solid Modeling System Based on The Parasolid Kernel,” The 12th International Pacific Conference on Automotive Engineering, IPC-12, Bangkok, Thailand, 2003.

[16] Sangveraphunsiri, V., “Information technology for Innovation of Manufacturing,” Proceeding of Asian Academy Seminar on Advanced Manufacturing system, Hyderabad, India, 2000.