Int J Adv Manuf Technol
DOI 10.1007/s00170-006-0495-5

ORIGINAL ARTICLE

Viboon Sangveraphunsiri · Kiattisak Sritrakulchai

The development of a technique for 3D complex surface reconstruction from unorganized point cloud

Received: 14 July 2005 / Accepted: 10 October 2005

Abstract In this paper, we propose a new neural network based on our two-level adaptive hierarchical clustering algorithm. The algorithm is to manage unorganized points, so that the triangular mesh models can be correctly obtained by applying the triangular mesh creation algorithm. We also develop adaptive self-flipping triangle edges to improve triangular mesh structure. Only one parameter, the maximal edge length of triangle, is needed in the neural network. The proposed two-level consists of the first level for clustering the cloud of points that has same order of the maximal edge length into a same cluster and the second level for generating triangular surface model or drape surfaces over the points of the same cluster. The normal vector for the generated triangular 3D surface model can be obtained from the second level. This helps to generate the STL file or stereolithography format. From the experimental results it can be shown that the proposed method is very effective for clustering unorganized point clouds for generating a triangular mesh of complex surfaces.

Keywords Unorganized points · Surface reconstruction · Hierarchical · Clustering algorithm

1 Introduction

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. 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 clustering algorithms have been previously investigated and covered a vast range of methodologies. Jain and Dubes [1] discuss the statistical approaches, or Song and Lee [2] and Hodge and Austin [3] and Azuaje and Dubitzky [4] discuss 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 an adaptive hierarchical clustering algorithm as compared to Hodge and Austin [3] 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 [5]. Our technique is a unique reverse engineering application when no CAD-data of the component is 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 as well. Typical steps include scanning 3D objects such as clay, wood, or an existing 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.

Several works [6–14] have proposed techniques 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 represents the shape and topology of the original object from which the set of sample points were drawn. Our work is a unique technique where the algorithm used the series of slicing planes moving in the user-specified direction to sort and order the cloud points. Then the sorted data points are used in the organizing and clustering process of the algorithm. The slicing planes are used just to control to interval of the neighborhood points. We do not

V. Sangveraphunsiri (✉) · K. Sritrakulchai
Robotics and Advanced Manufacturing Lab,
Department of Mechanical Engineering,
Chulalongkorn University,
254 Phyathai Patumwan,
Bangkok 10330, Thailand
e-mail: viboon.s@eng.chula.ac.th
Tel.: +66-2-2186448
Fax: +66-2-2186583

do any point projection to the slicing planes, which is different from the process proposed by [15–17]. So, the triangular mesh and STL file were directly generated from the original point clouds. The algorithm can be used for complex surfaces reconstruction.

There is much research work on surface reconstruction using a neural network. There are two forms commonly used for surface reconstruction from unorganized points of a cloud. The first form is the Kohonen's self-organizing map. It needs to predetermine topology or connectivity of a triangular surface, which is the network structure, for reconstructing surface. Then a neural-network learning algorithm will be used for adjusting the 3D coordinates of each vertex of the triangular surfaces so that they are close enough to the input point with a specified constrain. Its disadvantage is the fact that the network structure had to be pre-specified in advance. The detail of the procedure can be consulted [6–8]. The choice of an unsuitable network structure, however, can badly degrade network performance.

The second form of neural network applied in order to overcome the drawback of Kohonen's model is the growing cell structures (GCS) [9–12]. This network is similar to the Kohonen's self-organizing maps. For the growing cell structure, the network structure is a triangle and the network nodes are vertices. The network structure starts with a very simple network, the triangle, and grows incrementally by adding one new node or deleting a node to the network, which depends on the distribution of input point set.

However, the results of the two methods of neural network mentioned above are unsuitable for high-accuracy applications such as rapid prototyping and manufacturing with CNC systems because the triangular 3D surface model created from the approximated points that are close to actual input points. They are not the original cloud points, and it is very difficult to assign the suitable parameters needed for training the learning network.

In this paper, we use only one point at each cycle in our algorithms. Every input point will be sorted and ordered by the sorting and ordering process. Our work is a unique technique; this process used the series of slicing planes moving in the user-specified direction, which is normally x-director of the object, to sort and order the point cloud. Then the sorted data points are used in the organizing and clustering process of the two-level adaptive hierarchical clustering algorithm. This algorithm will help to arrange the new input to a suitable group of points so that the more accurate triangular mesh can be created. After that, the polygonizing follows and the normal vectors for STL file will be created by the triangular mesh creation algorithm. So, our algorithm uses original points for generating the triangular mesh.

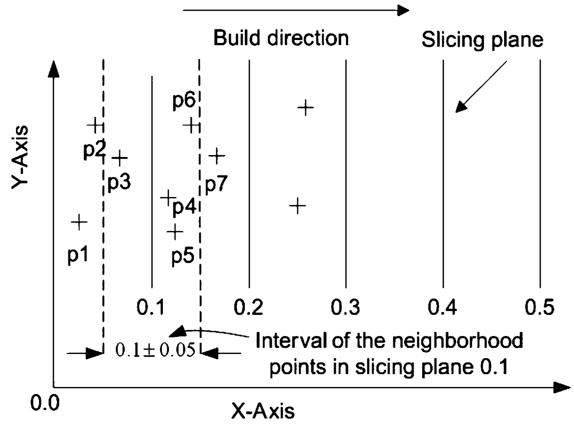


Fig. 2 Ordering of slicing planes and points within each plane

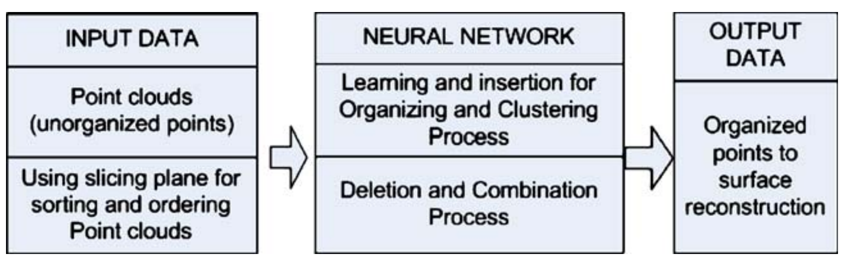
2 Data management system

Figure 1 illustrates the data flow process of our whole process. The point cloud will be sorted and ordered to minimize complexity or disordering. Then, in the neural network, the sorted data points are used in two processes as the organizing and clustering process and the deletion and combination process. Finally, the output data from the neural network are ready for generating triangular mesh with the incremental triangular mesh creation algorithm.

2.1 Sorting and ordering process

For example, in Fig. 2, the points are sorted by considering the distance within slicing planes along the X-axis. The slicing planes are 0.1 apart equally. We specify ±0.05 as the interval for grouping points to the nearest slice plane. Points p1 and p2 in Fig. 1 will be grouped to the slicing plane at 0.0 and ordered by the distance to the slicing plane. Similarly, points p3, p4, p5, p6, and p7 are ordered in the same way and grouped to the 0.1-slicing plane. In addition, we do not need to specify the whole cloud point for

Fig. 1 Overall data flow process of the novel technique



grouping. We can just specify only a portion of the cloud point in the grouping process if needed. This gives us more flexibility to manage the cloud point. After this arrangement, the organizing and clustering process will be preceded.

2.2 Organizing and clustering process

We need to define index symbols to help clustering, organizing, and recognizing point data used in the two-level adaptive hierarchical clustering algorithm. The point cloud will be grouped into a hierarchical structure as shown in Fig. 3 as follows.

M The set of all point data, where $P_i \in M$; i=1,2, … , n, is the point data in *M*. And n is the number of point in the set *M*.

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 these partitions of mesh at index m; m=1,2, … ,a, where a is the number of set partition of the point data.

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

$C_{v,c,m}$ The members of sub-set $NC_{c,m}$ at index v. So, $C_{v,c,m}$ will be called cell or members of $NC_{c,m}$ at v, where v = 1, 2, …, d, and d is the number of members of sub-set $NC_{c,m}$. So, v indicated the point number in the point cloud. See example shown in the Fig. 3.

In addition to the symbols specified above, the parameter for specifying the similarity of points will be specified by the user. This parameter is needed to start the searching algorithm. This parameter is called the length similarity or the maximal edge length (ρ), which is used for creating the triangular mesh. It represents the Euclidean distance measured among the points used in the clustering process.

After the clustering process, the set of points that have similarity distribution will be clustered into the same partition, which will be represented by a mutually disjoint subsets or clusters in the first level, that is $SM_1,SM_2,\ldots,SM_4,\ldots,SM_a$. Each SM_q is divided into *NC* clusters or second level, $NC_1,NC_2,\ldots,NC_h,\ldots,NC_b$. Then, the triangular mesh will be generated from these *NC* clusters.

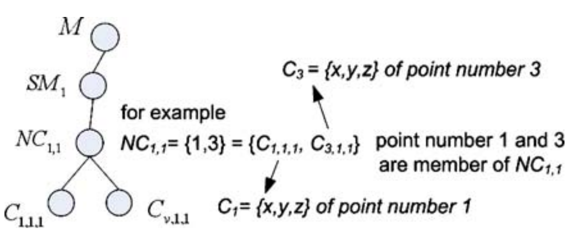


Fig. 3 The symbols of index data

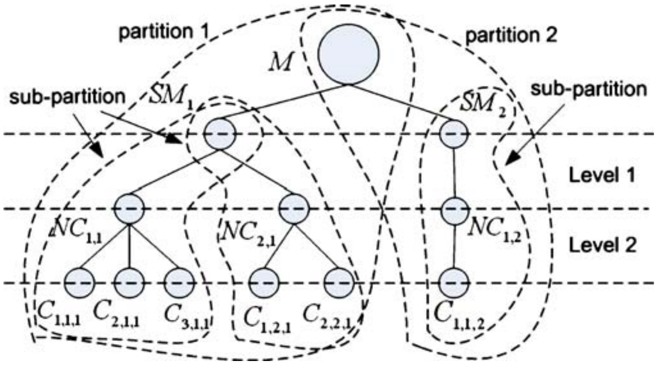


Fig. 4 Subdivision of partition in the two-level hierarchical clustering structure

In conclusion, the organizing and clustering process is to map sampled point (in *M*) to the associated two-level hierarchical cluster of similarity of point data as shown in Fig. 3.

Where *SM*, level-one cluster, is a set partition of *M*. Then, *SM* is a collection of nonempty sets, $\{SM_1,SM_2,\ldots,SM_a\}$, such that, $M=SM_1\cup SM_2\cup\ldots\cup SM_a$, and sets, $SM_1, SM_2,\ldots,SM_a$ are mutually disjoint, for all $SM_q \cap SM_l = 0$ where q, l=1,2,…,a, whenever $q \neq l$. The *NC*, level-two cluster, is the sub-set partition of SM_q. So, *NC* is a collection of nonempty sets, $\{NC_1,NC_2,\ldots,NC_b\}$, such that, $SM_m=NC_1\cup NC_2\cup\ldots\cup NC_b$, and sub-sets, $NC_1,NC_2,\ldots,NC_b$ are mutually disjoint, for all $NC_h \cap NC_e=0$ where $h,e=1,2,\ldots b$, whenever $h \neq e$.

Figure 4 shows that the two-level hierarchical clustering structure is more than one partition, SM_1 and SM_2, and/or the number of sub-partitions is greater than 1, $NC_{1,1}$ and $NC_{2,1}$.

2.3 Learning and inserting process

Figure 5 illustrates the organizing and clustering process. The network starts from the simple neural network structure, as shown in the Fig. 5a, which will consist of four nodes as point cloud (*M*), a partition (SM_1), a sub-partition ($NC_{1,1}$) of SM_1 and a cell ($C_{1,1,1}$) which contains a point.

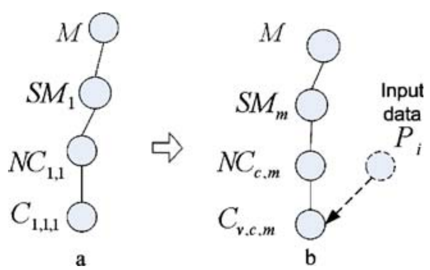


Fig. 5 Starting neural network structure

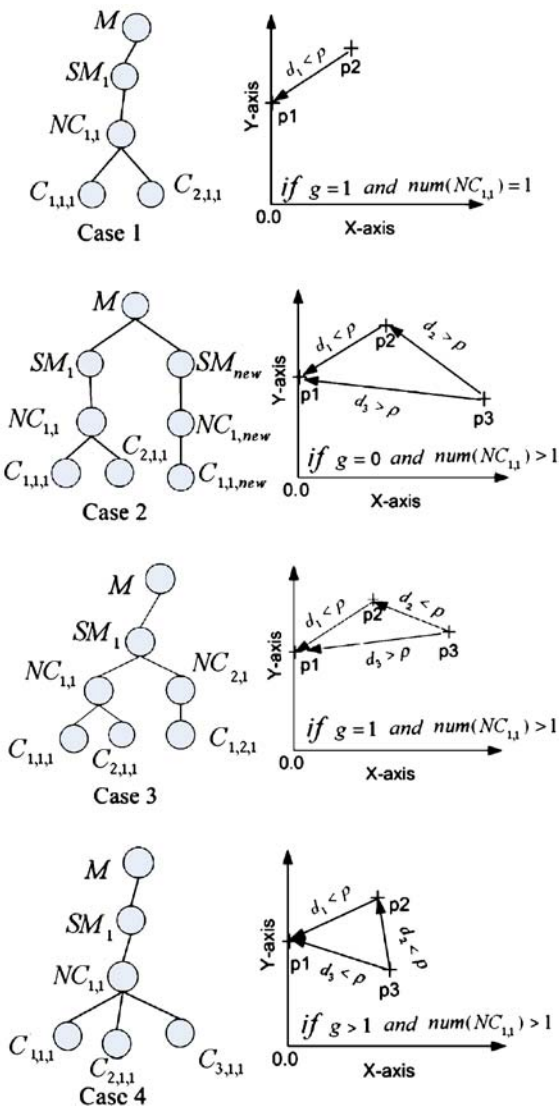


Fig. 6 Learning and insertion of the four events for new fed input point data (P_i)

Then, we apply the distance similarity equation for searching as $(D_v)_{c,m} = \left\|(P_v)_{c,m} - P_i\right\|$, where $\|.\|$ denotes the Euclidean vector norm. So, $(D_v)_{c,m}$ is a set $\{d_1,d_2,\ldots, d_v\}_{c,m}$, the distance between point data, $(P_v)_{c,m}$, (point v of

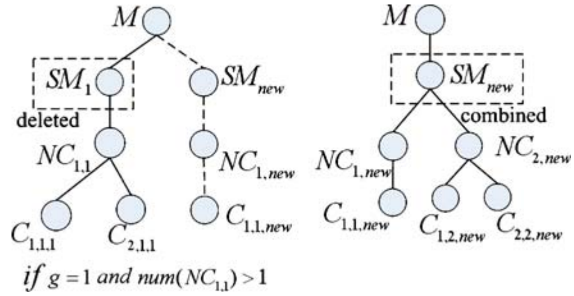


Fig. 8 Deletion and combination for case 3

sub-set partition index c of set partition index m) and new sampled point data P_i which is the 3D coordinate of the input point data as shown in Fig. 5b. The distance in the set $(D_v)_{c,m}$ is used for clustering by comparing to the length similarity, ρ. In this clustering process, there are four possible cases or events of learning and insertion for every input point data as indicated in Fig. 6. We will define the distance set L_g, $\{d_1,d_2,\ldots,d_g\}$; 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.

Figure 6 shows the four cases of the learning and insertion of the new fed input point data (P_i) (in this example case is point P_2). Case 1: When d_1<ρ, which means that the input point P_2 can be inserted into the sub-partition $NC_{1,1}$ and the total number of points in this set is two, which is not enough to create the triangular mesh. In other words, the condition of case 1 occurs when the number of members of the set Lg =1 or g=1 and the number of members in set $NC_{1,1}$ is equal to one. Case 2: The new input point is fed (in this example is P_3). If d_2>ρ and d_3>ρ, this means that the new point P_3 is not close to any point P_1 and P_2 in sub-partition $NC_{1,1}$ or g=0. So, the new partition needed to be created is shown in Fig. 6. Case 3: The new input point P_3 is fed. If d_2<ρ (g=1) and $d3$>ρ, which mean that point P_3 cannot be inserted into the sub-partition $NC_{1,1}$. So, we need to create a new sub-partition $NC_{2,1}$ and point P_3 will be inserted into the $NC_{2,1}$. Case 4: If d_3<ρ (or g>1), then point P_3 will be inserted into sub-partition $NC_{1,1}$. So, the number of points in the sub-partition is enough to create a triangular mesh.

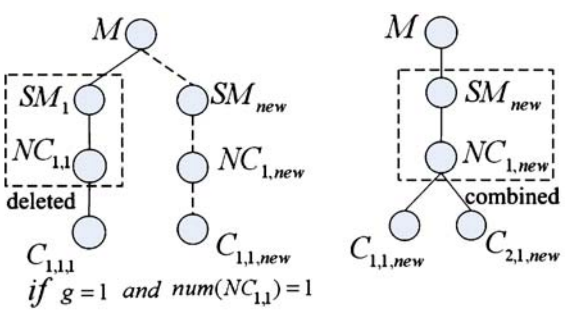


Fig. 7 Deletion and combination for case 1

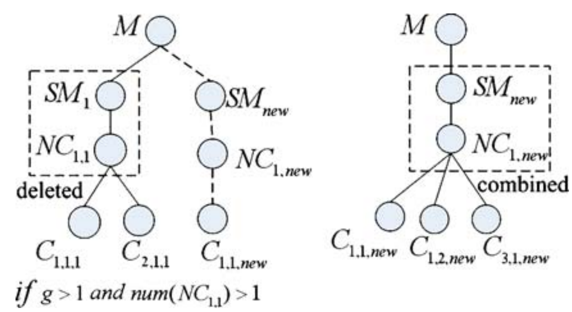


Fig. 9 Deletion and combination for case 4

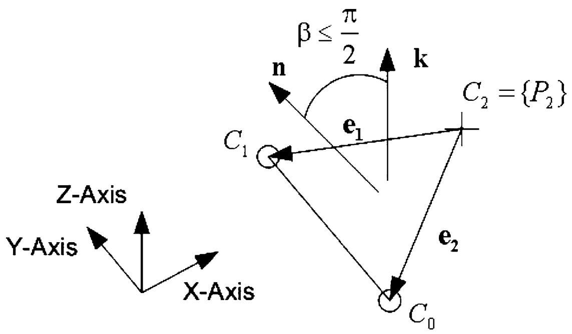


Fig. 10 The normal vector direction of the first triangle

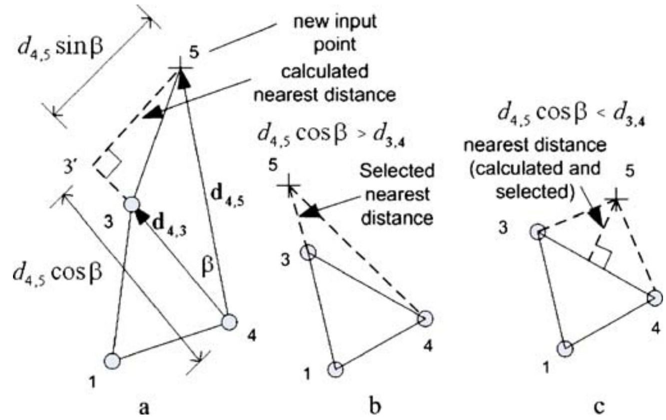


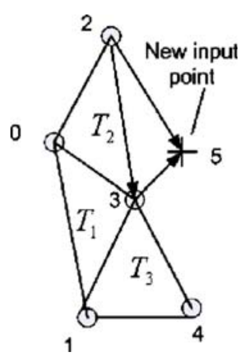
Fig. 12 Condition for searching the nearest distance

2.4 Deletion and combination process

The deletion and combination process will be used when the number of members of set $Lg \geq 1$, that is case 1, case 3 and case 4 occur, in the organizing and clustering process. In Fig. 7, If the new input point P_2 of two-level hierarchical clustering structure, SM_{new}, $NC_{\text{new},1}$, $C_{1,1,\text{new}}$, is closed to P_1 of SM_1, $NC_{1,1}$, $C_{1,1,1}$ and $d_1<\rho$, this means that SM_1, $NC_{1,1}$ and SM_{new}, $NC_{\text{new},1}$ can be combined as shown in Fig. 7 for case 1. After the combination, SM_1 and $NC_{1,1}$ set will be deleted. If the new input point P_3 is closed to P_2 but further to P_1 of SM_1, $NC_{1,1}$, $C_{1,1,1}$,$C_{2,1,1}$ with $d_2<\rho$ and $d_3>\rho$, this is means that SM_1 and SM_{new} can be combined as shown in Fig. 8 for case 3 and SM_1 should be deleted. If the new input point P_3 is closed to P_1 and P_2 with $d_2<\rho$ and $d_3<\rho$, this means that SM_1, $NC_{1,1}$ and SM_{new}, $NC_{\text{new},1}$ can be combined as shown in Fig. 9 for case 4 and SM_1 and $NC_{1,1}$ set should be deleted.

After the deletion and combination process, we eliminate the duplicate set of points. Each NC will contain the points and neighborhood points that have the same similarity or distance between these points less than the specified length similarity. This will provide the set of points for creating triangular mesh more efficiently.

Fig. 11 The computation for the other triangles



3 The triangular mesh creation

The triangles are preceded whenever the number of cells within the sub-set of the partition of point data, $NC_{c,m}$, is greater than or equal to three, and the number of members in the set $(L_g)_{c,m}$ is greater than 1 or g>1. This is the case 4 as described in the organizing and clustering process.

3.1 The triangle creation

For the first triangle, when the number of cells is three or the number of members in the set $NC_{c,m}$ is equal to three, a triangle can be created by setting $\beta \leq \pi/2$ as shown in Fig. 10. The β is angle between vector k, a vector along Z-axis specified by user for slicing plane, and n, the normal vector of the triangle T_1, calculated as shown in Fig. 10.

$$\mathbf{n} = \frac{\mathbf{e_1} \times \mathbf{e_2}}{|\mathbf{e_1} \times \mathbf{e_2}|}, \tag{1}$$

$$\beta = \cos^{-1}\left[\frac{\mathbf{n}\bullet\mathbf{k}}{nk}\right], \tag{2}$$

For example, $e_1 = \overrightarrow{C_2C_1}$ and $e_2 = \overrightarrow{C_2C_0}$, if $\beta \leq \pi/2$, then the triangle can be created along the ordered cells $\{C_2,$

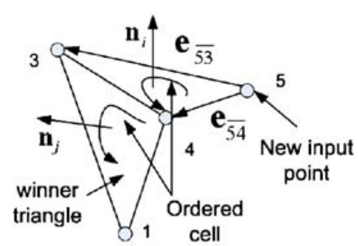


Fig. 13 The newly created triangle with respect to the winner triangle

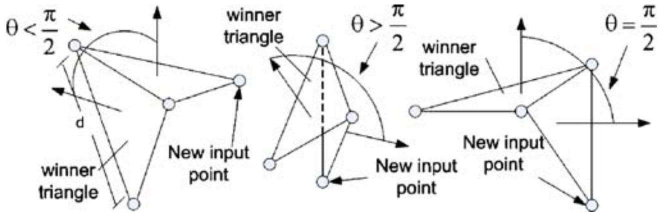


Fig. 14 The angle between the existing triangle and the newly created triangle

$C_1,C_0\}$; otherwise, the order of the cells is $\{C_2,C_0,C_1\}$. In other words, the triangle composed of three edges is $\{\overline{C_2C_1}, \overline{C_1C_0}, \overline{C_0C_2}\}$ or $\{\overline{C_2C_0}, \overline{C_0C_1}, \overline{C_1C_2}\}$ according to the direction of the normal vector.

The other triangles are preceded after the first triangle was created as shown in Fig. 11.

We need to search the best edge, called the winner edge, of the existing triangular mesh for sharing with the new created triangle of the new input point, for example point 5 shown in Fig. 11. For efficiency and effective searching, our algorithm searches only the best edge within the boundary triangular mesh (BTM). The BTM is composed of the edges of the triangles that do not share any other edge of triangles or the edges at outside of the triangular mesh in 3D space. Using BTM will reduce computation and searching time, especially for a large data. For example in Fig. 11, the BTM of the new input point 5 composed of edges $\overline{C_0C_1}$, $\overline{C_3C_2}$, $\overline{C_2C_0}$, $\overline{C_4C_3}$, and $\overline{C_1C_4}$.

Figure 12 shows two possibilities of the nearest distance calculated from the new input point. The first possibility is $d_{4,5}\cos\beta > d_{3,4}$, shown in Fig. 12a,b. So, the nearest distance is $d_{3,5}$. The second possibility is $d_{4,5}\cos\beta < d_{3,4}$, then the nearest distance is $d_{3',5} = d_{4,5}\sin\beta$ as shown in Fig. 12c. The angle β is calculated from Eq. 2.

The ordering of cells of this new triangle is based on the normal vector of the winner triangle as n_j and n_i in Fig. 13.

There are some special cases, such as surface discontinuity or a surface with a very high steep, point error, or noise due to the combination of more than one point cloud. We need to check whether these conditions occur by inspecting the angle between the normal vector of the winner triangle and the new triangle including the neighborhood triangle of the winner triangle.

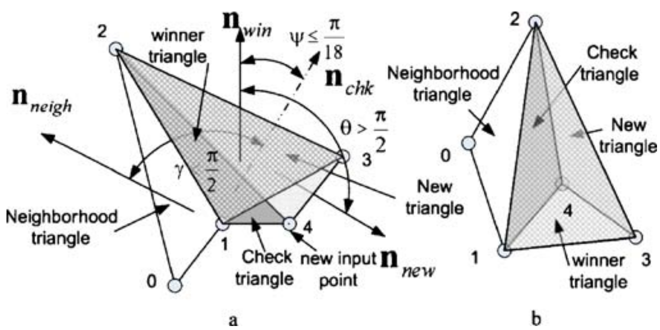


Fig. 15 Comparison of overlapped triangle for subdivision to three triangles

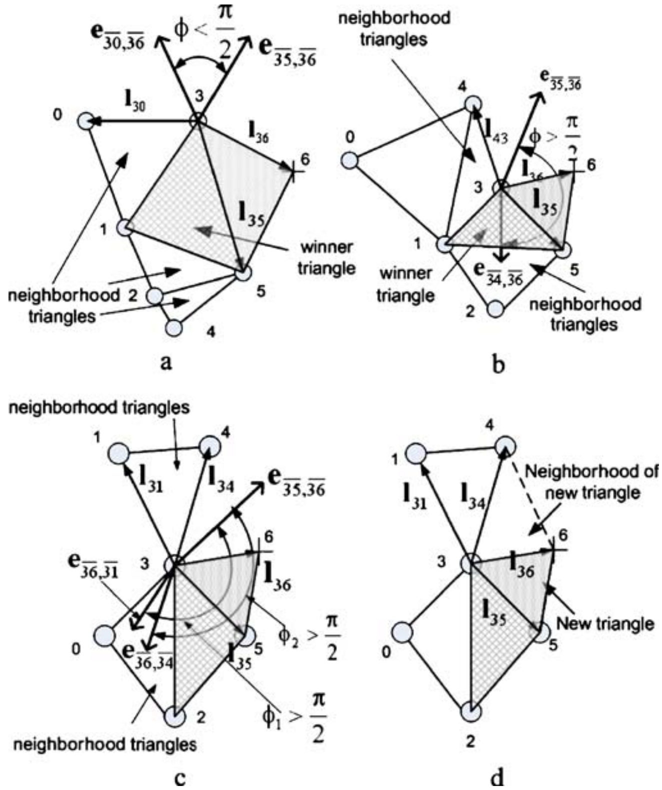


Fig. 16 Condition for creating neighborhoods of a new triangle

For example, in Fig. 14, it shows the direction change of angle θ, which is the angle between the newly created triangle for a new input point and the winner triangle. If θ<π/2, the new triangle is created by sharing its edge with an edge of the winner triangle. But if θ>π/2, this means that a high steep surface or surface discontinuity occurs. To

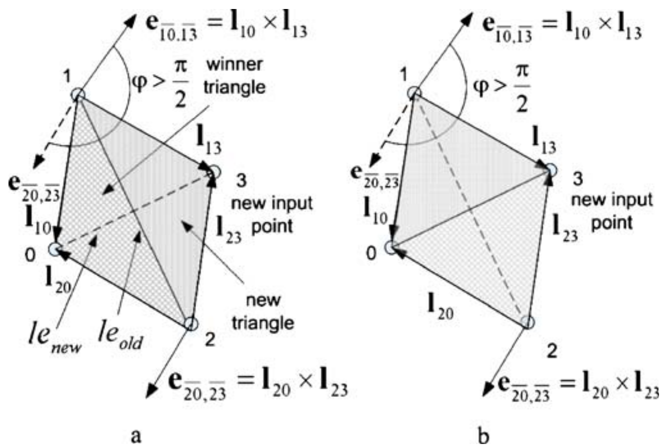


Fig. 17 The two triangles for self-flipping triangle edges of the triangular mesh structure

Table 1 The specifications of test models

Model	Max edge length (ρ)s	Characteristic surface	Number of points	Number of triangles
Human face	3.6	Surface with holes, open surface	5,716	11,379
Pan handle	2.8	Discontinuous surface (grooves), close surface	5,271	10,538
Teeth	1.6	Very complexity	7,886	15,487
Car seat	3.2	Complexity, sharp edge	9,680	19,037

confirm the case, we need to examine the neighborhood of the winner triangle to see whether at least one $\gamma<\pi/2$ met. So, the new triangle can be created. Otherwise, the new input point will be moved to the new extra case, as compared to the cases explained in Section 3, where the distance of the new input point is within the length similarity (g>0) but the angle $\theta>\pi/2$ as well as $\gamma>\pi/2$. The γ is the angle between the neighborhood of the winner and the check triangle; the check triangle is created for inspection to tell us about the closeness of the new triangle and the winner triangle. This case means that the new input point (point 4 in Fig. 15) is very close to the winner triangle. The closeness is defined by the angle ψ. The angle ψ is the angle between the normal vector of the winner triangle and the normal vector of the check triangle as illustrated in Fig. 15. The check triangle is formed by an edge of the neighborhood triangle and the edge of the new triangle. An example of a check triangle is shown in Fig. 15.

In addition to condition 1, if $\psi\leq\pi/18$, as shown in Fig. 15, this means that the new input point is very close to the winner triangle. The two triangles, the new triangle, and the check triangle, will almost overlap the winner triangle. This will affect the quality of the triangular mesh. So, to overcome the problem, we need to divide the winner triangle into three triangles so that the normal vectors of the additional three triangles will point in the same direction.

The value $\pi/18$ for the angle ψ is heuristic value. Alternatively, we can specify this value manually within the range (0, $\pi/2$), which depends on the complexity of surface. This range can be re-specified by the user if needed. For a smooth surface, the range of angle θ can be made smaller to reduce the effect of noise.

Details of the algorithm are as follows:
IF the number of members in set NC_c= = 3 and g > 1
COMPUTEn and β
IF $\beta\leq\pi/2$
COMPUTE order of cell in counter-clockwise direction
ELSE
COMPUTE order of cell in clockwise direction
COMPUTE create normal vector
ELSE IF the number of members in NC_c> 3 and g > 1
SET Near_EachEdge=[ ]
FOR every vertices that satisfy the BTM condition
COMPUTE the distance between the new input point and vertices
IF $d_{4,5}\cos\beta > d_{3,4}$
Add.Near_EachEdge = $d_{3,5}$
ELSE
Add.Near_EachEdge = $d_{4,5}$ sinβ
COMPUTE min(Near_EachEdge)
COMPUTE create normal vector of the new triangle
COMPUTE angle θ, γ, ψ
IF $\theta>\pi/2$
IF $\gamma<\pi/2$
IF $\psi\leq\pi/18$
COMPUTE split triangle to three triangles
ELSE
COMPUTE create new trianglecreate normal vector
ELSE
INCREASE create new partition SM_{new}, $NC_{1,new}$, $C_{1,1,new}$
ELSE
COMPUTE create new trianglecreate normal vector

3.2 Creation of the neighborhood triangles of a new triangle

After creating a new triangle from the shared edge, $\overline{C_3C_5}$, of the winner triangle, we also need to create neighborhood triangles of the new triangle. The neighborhood triangles and the new triangle have to share the same common point, C_3 and C_5. At each common point, the angle ϕ will be calculated and used for deciding whether the neighborhood

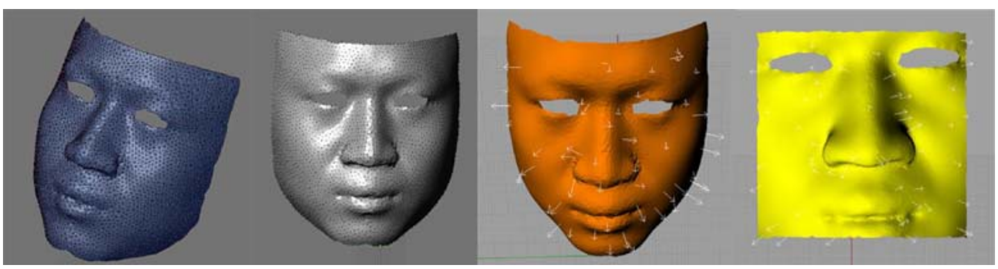


Fig. 18 The human face model for open discontinuous surface (two holes represent eye sockets)

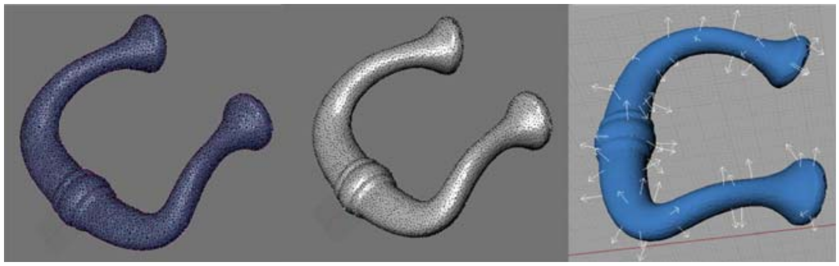


Fig. 19 The panhandle model for close surface

triangle will be created. For example, in Fig. 16a, at the common point C_3, the angle ϕ is calculated from the two vectors, the vector edge of the new triangle $e_{\overline{35},\overline{36}}$ and the vector edge $e_{\overline{30},\overline{36}}$ formed by the BTM. The angle $\phi<\pi/2$, which means that none of them is not qualified for creating the required neighborhood triangle. Otherwise, the newly created neighborhood triangles will overlap with the winner triangle.

Details of the algorithm are shown as follows:

COMPUTE angle ϕ of the neighborhood triangle of the new triangle, which depends on BTM for each common point
SET NumNeighCreated=0
COMPUTE NumNeighCreated equals the number of ϕ, for ($\phi>\pi/2$)
IF NumNeighCreated > 1
IF vertices of the new neighborhood = = vertices of edge winner
COMPUTE minimum (angle ϕ) and create neighborhood triangle
ELSE
COMPUTE create every neighborhood triangle
ELSE IF NumNeighCreated = =1
COMPUTE create neighborhood triangle

Figure 16 illustrates that there are three possibilities for creating new neighborhood triangles. In the first possibility, as shown in Fig. 16a, there is no new neighborhood created as $\phi<\pi/2$. The second possibility, only triangle $\{C_4, C_3, C_6\}$ was created as shown in Fig. 16b, because the triangle does not overlap with the winner triangle. The third possibility is shown in Fig. 16c. Because $\phi_1>\pi/2$ and $\phi_2>\pi/2$, we need to select only one of them by selecting the smaller angle ϕ for creating the new neighborhood triangle. $\{C_4, C_3, C_6\}$ as shown in Fig.16(d).

4 Self-flipping triangle edges

To improve the triangular mesh structure, we present self-flipping triangle edges. For example, in Fig. 17, the two triangles, shared with the edge l_{12}, are triangle $\{C_2, C_1, C_0\}$ and $\{C_3, C_1, C_2\}$. The length of the edge between C_1 and C_1 before adaptation is le_{old}.. After adaptation, the edge l_{03} will be used to separate the triangular mesh instead of l_{12}. Because the length le_{new} is less than the length le_{old}., the adaptation of connectivity of the triangular mesh will modify existing triangular mesh as shown in Fig. 17a to the new triangular mesh as shown in Fig. 17b. This will improve the quality of the mesh.

More systematic checking whether the self-flipping triangle edges should be applied can be considered from the angle φ as shown in Fig. 17. The angle φ is the angle measured between the normal vector of the two triangles formed by the non-share edges of the existing two triangles as illustrated in Fig. 17. If $\varphi>\pi/2$, then the adaptation of the two triangles occurs. The adaptation is continued for every sharing edge of two triangles. This will improve the quality of mesh.

5 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 with the ability to interface to other languages. Python can also be implemented in various operating systems. Our tested platform is a Pentium III 700 MHz running the Linux operating system. For post processor, we are using the Blender (freeware) for graphic display of the triangular mesh. We also developed our own post-processor for graphic display [18]. The point clouds, using in the experimentation, were obtained by using the

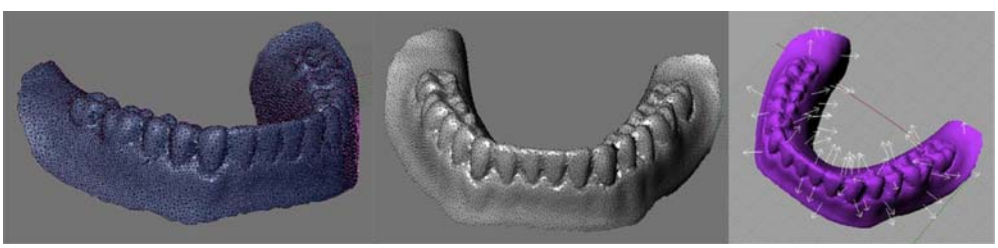


Fig. 20 The teeth model for an open very complex surface

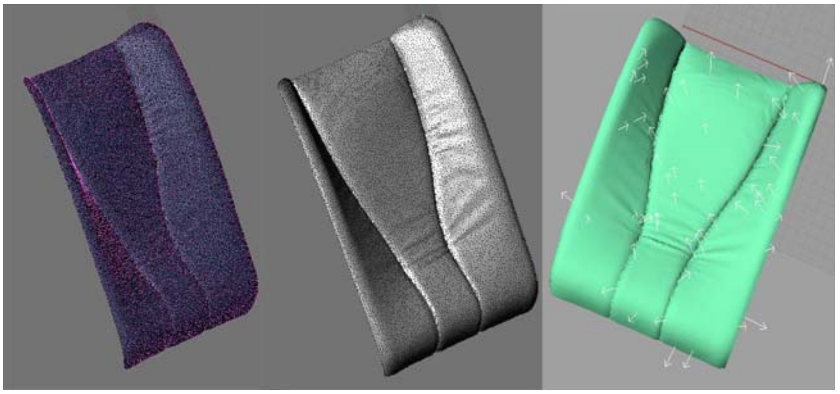


Fig. 21 The car seat model for open very complex surface

optical 3D measurement systems. Both commercial optical 3D measurement systems as well as our developed automatic laser scanning machine [19] were used for the measurement. The outputs of the testing are the reconstruction surface from unorganized point clouds. In the experiment, we specify 0.1 mm for the offset interval of the slicing planes. The offset interval is set equally. The other parameter is the angle $\psi \leq \pi/18$ for checking overlap of the triangles.

Four examples are presented here to illustrate the efficacy of the algorithm. Table 1 shows the characteristic of the test models, the specified maximal edge length, the number of points, as well as the number of triangle created.

The results for creating STL models are shown in Figs. 18, 19, 20 and 21. In Fig. 18, it is also shown that this algorithm will provide us the flexibility to manage the point cloud. Only a portion of the point cloud can be specified for regenerating the surface. In all of the examples, we also show the normal vector of the generated surface models, which is very useful information for informing the quality of the regenerated surface.

6 Conclusion

In this paper, we presented an algorithm for generating a triangular mesh from unorganized points. The unorganized points are sorted and ordered by the sorting and ordering process. The sorted data points are used in the organizing and clustering process by using the two-level adaptive hierarchical clustering algorithm. Then, the organized data points are used for creating a triangular mesh with the triangular mesh creation algorithm. We also develop an adaptive self-flipping triangle edge to improve the triangular mesh structure. These algorithms are used for surface reconstruction and creating an STL model. We demonstrated our algorithm with several experiments. Only one parameter, ρ, needed the organizing and clustering process. An STL file can be generated by our algorithm. The STL format is more versatile in many applications. The STL model can be transferred to general CAD/CAM software. From the experiments, we found that our algorithms work satisfactorily for complex surfaces, such as open-surface models, discontinuous surface models (for example, a surface with holes and without holes, and close surface models).

References

1. Jain A, Dubes R (1998) Algorithms for clustering data. Prentice Hall, Englewood Cliffs, NJ
2. Song H, Lee S (1998) A self-organizing neural tree for large set pattern classification. IEEE Trans Neural Networks 9(3)
3. Hodge V, Austin J (2001) Hierarchical growing cell structures: TreeGCS. IEEE Trans Knowl Data Eng 13(2):207–218
4. Azuaje F, Dubitzky W, Black N, Adamson K (2000) Discovering relevance knowledge in data: a growing cell structures approach. Syst Man Cyb, Part B. IEEE Trans 30 (3):160–448
5. Burzevski V, Mohan CK (1996) Hierarchical growing cell structures. IEEE International Conference Neural Network 3:1658–1663
6. Yu Y (1999) Surface reconstruction from unorganized points using self-organizing neural networks. IEEE Visualization 99 Proceedings Conference, pp 61–64
7. Mangiameli P, Chen S, West D (1996) A comparison of SOM neural network and hierarchical clustering methods. Eur J Oper Res 93:402–417
8. Fritzke B (1993) Kohonen feature maps and growing cell structures - a performance comparison. In: Giles CL, Hanson SJ, Cowan JD (eds) Advances in neural information processing systems-5-(NIPS-92)
9. Ivrissimtzis I, Jeong W, Seidel H (2003) Using growing cell structures for surface reconstruction. Proceedings of the Shape Modeling International, pp 78–86
10. Fritzke B (1996) Growing self-organizing networks - why?. ESANN'96: European Symposium on Artificial Neural Networks, pp 61–72
11. Fritzke B (1993) Growing self-organizing network for unsupervised and supervised learning. Technical Report ICSTR-93-026. International Computer Science Institute, Berkeley
12. Kawahara S, Saito T (1996) On a novel adaptive self-organizing network. CNNA'96, Fourth IEEE International Workshop on Cellular Neural Networks and Their Applications, pp 41–46
13. Hoppe H, Derose T, Duchamp T, McDonald J, Stuetzle W (1992) Surface reconstruction from unorganized points. SIGGRAPH'92 Proceedings Conference, pp 71–78
14. Amenta N, Bern M, Kamvysselis M (1998) A new Voronoi-based surface reconstruction algorithm. SIGGRAPH'98 Proceedings Conference, pp 415–421

15. Liu GH, Wong YS, Zhang YF, Loh HT (2003) Modelling cloud data for prototype manufacturing. J Mater Process Technol 138:53–57
16. Zhang YF, Wong YS, Loh HT, Wu YF (2003) An adaptive slicing approach to modeling cloud data for rapid prototyping. J Mater Process Technol 140:105–109
17. Wu YF, Wong YS, Loh HT, Zhang YF (2004) Modelling cloud data using an adaptive slicing approach. Compute Aided Design 140:213–240
18. Sangveraphunsiri V, Uttamang K (2003) Development of a 3-D solid modeling system based on the parasolid kernel. 12th International Pacific Conference on Automotive Engineering, IPC-12, Bangkok, Thailand
19. Sangveraphunsiri V (2000) Information technology for Innovation of manufacturing. Proceedings of Asian Academy Seminar on Advanced Manufacturing system, Hyderabad, India