

Extract Segmentation Lines of 3D Model based on Regional Discrete Curvature

Hui Jia¹ and Jiangang Zhang²

¹*School of Computer Science & Technology, Xi'an University of Posts and Telecommunications, Xi'an 710121, China;*

²*Information and Supervision Department, Xi'an Thermal Power Research Institute Co Ltd, Xi'an 710032, China.
jhnwu@126.com;marko_zhang@tpri.com*

Abstract

The discrete curvature is sensitive to the noise and heavy computation of feature value. In order to extract the segmentation lines of 3D meshes, this paper presents a novel and efficient algorithm which is based on regional discrete curvature. Extract prominent feature points, pre-segmentation and then obtain the constraint region. Calculate the curvature degree extremal points within the constraint region, finally utilize the curvature degree extremal points to fit plane which guided extracting the segmentation lines. This algorithm does not need to wipe off the noise at initial stage, the experiment results show that the proposed algorithm can accurately and frequently yield meaningful segmentation lines for models which has an evident branch structure.

Keywords: 3D model; mesh segmentation; segmentation lines; regional discrete curvature; watershed algorithm

1. Introduction

3D model segmentation is divided model according to the geometrical characteristics into a set of parts with limited number and each has a simple shape [1]. 3D model segmentation is a research hot spot at present[2,3]. The first step of many research methods of works such as model simplification, compression, fast modeling, model deformation, 3D texture mapping, 3D model retrieval is model segmentation. And most of this kind of segmentation is meaningful segmentation [1,5,6], namely according to the minima rule[4], the complex object is as a combination of simple basic elements. Human perception usually divides a surface into parts along the concave discontinuity of the tangent plane.

Several approaches have been discussed in the past for decomposing meshes. Aleksey Golovinskiy *et. al.*, [5] put forward method to mesh partitioning by random cut. The general strategy is to generate a random set of mesh segmentation and then to measure how often each edge of the mesh lies on a segmentation boundary in the randomized set. Kate et al [6] proposed hierarchical mesh decomposition algorithm using fuzzy clustering and cuts. This algorithm proposed a hierarchical mesh decomposition algorithm. First computed decomposition boundary between the components fuzzy. Then, focused on the small fuzzy areas and found the exact boundaries which go along the features of the object. The algorithm also avoided over-segmentation and jaggy boundaries between the components. Halim Benhabiles *et. al.*, [7] presented automatic 3D-mesh segmentation algorithm based on boundary edge learning. A large database of manually segmented 3D-meshes is used to learn a boundary edge function and a processing pipe line that produce smooth closed boundaries using this edge function was presented. Sun Xiaopeng *et. al.*, [8] presented a hierarchical mesh segmentation algorithm as semi-supervised k-means clustering and k-ring strip growing. Han Li *et. al.*, [9] proposed discrete curvature

constrained triangle mesh model segmenting technique which is an optimized algorithm. The algorithm classified the vertex attribute based on discrete curvature estimation, and it then combined the region growing method to adaptively determine the topological structure of the 3D models.

Most of the above methods need to convert 3D models into dual graph, facet segmentation turned to vertex segmentation [5-6]; most of them using discrete curvature as characteristics of segmentation. However, estimate the curvature in whole 3D mesh cause a lot of calculation, in addition the discrete curvature is sensitive to the noise, and therefore affect segmentation results of those algorithms. In this paper, the segmentation algorithm does not need to convert 3D model into a dual graph, but directly calculate discrete curvature at mesh vertexes, and the calculation is constrained by region, which greatly reduce the calculated quantity of the curvature.

The algorithm consists of three stages:

Choose prominent feature points. Prominent feature points can on behalf of the model's components, and generally reside on tips of components of a given model.

Pre-segmentation. Segment 3D model according to prominent feature points and geodesic distance, and get the constraint region.

Extract segmentation lines based on regional discrete curvature. Computing discrete gaussian curvature within constraint region and get extremal points, utilizing the curvature degree extremal points to fit plane which guaid extracting the segmentation lines.

2. Choose Prominent Feature Points

Prominent feature points are points which can represent the model's components and always reside on tips of components. Let v is the vertexes set of the model and $V = \{v_i \in R^3, 1 \leq i \leq n\}$, the algorithm to choose prominent feature points is as follows:

Step 1 Calculate geodesic distance $D(v_i, v_j)$ between vertexes, choose the vertex to all vertexes furthest distance as the first feature point, and add it to the feature set s .

Step 2 Iteratively add new feature points one by one. Calculate geodesic distance between all vertexes and each feature points in set s , choose the vertex has farthest distance to nearest feature point as new feature point. Assume s_1, s_2, \dots, s_n are n feature points, the next $n+1$ feature point s_{n+1} is defined as formula 1:

$$s_{n+1} = \max \{ \min_{i=1, \dots, n} D(v_i, s_i) \} \quad (1)$$

Step 3 The termination condition. When the k -th feature point has added in set s , we calculate the distance minimum value between the k -th feature point and other points in set s . This minimum value will gradually reduce as the point added, as formula 2:

$$G(k) = \min_{i < k} \{ D(s_k, s_i) \} \quad (2)$$

According to ref.[1], when the number of prominent feature points is enough to represent the component, $G(k)$ will decrease obviously. Consequently get first-order derivative, when $G(k)$ reach its maximum, k is the number of prominent feature points.

3. Pre-segmentation

The prominent feature points represent the component of 3D mesh, then we calculate the probability each vertex belong to the patch of each prominent feature point.

Let s_1, s_2, \dots, s_n are n feature points, $Patch(s_i)$ is the component represented by s_i , $d_{v_m}(s_i) = Dist(v_m, s_i)$ is the geodesic distance between vertex v_m and s_i , the probability v_m belong to $Patch(s_i)$ is:

$$p_{v_m}(s_i) = \frac{\sum_{k=1}^n d_{v_m}(s_k) - d_{v_m}(s_i)}{\sum_{k=1}^n d_{v_m}(s_k)} \quad (1)$$

Obviously the closer distance v_m and s_i , the higher probability v_m belong to $Patch(s_i)$. When v_m is exactly s_i , the probability is 1. Utilize formula 3 to calculate the probability every vertex belong to the patch each prominent feature point represented. Set a threshold value ε_1 :

$$v_m \in Patch(s_i) \text{ iff } \left(p_{v_m}(s_i) \geq 0.5 + \varepsilon_1 \right) \& \& \left(p_{v_m}(s_i) > p_{v_m}(s_j) \right) \quad (4)$$

If three vertexes of a face all satisfy formula 4, the face is belong to $Patch(s_i)$, else if one of them does not meet the formula 4, we regard the face as belonging to the patch to be segment $Patch(U)$. After pre-segmentation, 3D model has been segmented into $n+1$ $Patch$. $Patch(s_1)$, $Patch(s_2)$, ..., $Patch(s_n)$, $Patch(U)$. $Patch(U)$ is the constraint region during post-segmentation.

4. Post-Segmentation based on Regional Discrete Curvature

4.1. Calculate Discrete Curvature

Scholars have done plenty of studies of the discrete curvature estimation on the point cloud [10-11], among them Levin [12] put forward curvature estimation algorithm of point sets based on Moving-Least Square Surface (MLS). The basic idea is to define the MLS surface M as the surface projection of ψ_p , $M = \{x \in R^3 \mid \psi_p(x) = x\}$. Directly compute of surface curvatures for point-set surfaces based on a set of analytic equations derived from MLS. Amenta and Kil [13] gave a more precise definition for projection MLS surfaces as the local minima of an energy function $e(y, a)$ (y is a position vector and a is a direction vector) along the directions given by a vector field $n(x)$. As shown in Figure 1. Based on this definition, they derived a projection procedure for taking a point onto the MLS surface M implied by n and e , which can be illustrated in Figure 1. M is defined as zero solution set of implicit function $g(x)$.

In the MLS surface, there are three main steps to estimate curvatures of points set. (1) Compute normal of the input points. (2) Project the sample points onto the MLS surface. (3) Compute curvatures of the MLS surface at the projected points. The formulas of computing Gaussian curvature and mean curvature for the implicit surface are shown as formula 5 and formula 6. $\nabla g(x)$ is the gradient of $g(x)$, $H(g(x))$ is the Hessian matrix of $g(x)$.

$$k_{Gaussian} = - \frac{\det \begin{bmatrix} H(g(x)) & \nabla g(x) \\ \nabla^T g(x) & 0 \end{bmatrix}}{\|\nabla g(x)\|^4} \quad (5)$$

$$k_{mean} = - \left(\frac{\nabla^T g(x) H(g(x)) \nabla g(x) - \|\nabla g(x)\|^2 \text{trace}(H(g(x)))}{\|\nabla g(x)\|^5} \right) \quad (6)$$

Step2.1 Calculate $\max(C_v(r))$

Step 2.2

for $v_i \in Patch(U)$

if $\max(C_v(r)) - C_{v_i}(r) < \varepsilon_2$

Insert v_i into CurExtrePoint

end

Output CurExtrePoint

The value of ε_2 is according to the experiment, as shown in figure 2, the curvature degree extremal value of m1611 in $Patch(U)$ is around 5, in experiment the value of ε_2 is 0.2, the vertexes curvature degree difference to extremum value less than 2 as the curvature extremal points.

4.3. Contour Completion

The CurExtrePoint is as the segmentation line of the 3D model. Most of the time, The CurExtrePoint is not a closed curve, and need to be completed and formed a closed loop around the 3D model. Let β is selected unclosed segmentation line in CurExtrePoint, the closing process can be defined as iteratively select optimal adjacent point of endpoint of β , until a closed loop around the 3D model is completed. The critical method to form a closed curve is how to find the adjacent vertex which is closest to the most salient feature area. The algorithm from ref.14[14] believe that the user often intends to cut a mesh along boundaries which is perpendicular to the medial axis, but this assumption is not suitable to all models. As shown in Figure 3. The segmentation lines of the eagle sometimes parallel to the medial axis, such as the line in the wing, sometimes have a certain eagle to the medial axis, such as the line in the legs.

In order to perform automatic contour closing method which completes β to a closed loop around the mesh, we use a method based on combination of multi-feature. The unclosed feature line is coplanar to the closed loop in three-dimensional space, then fitting planar with Constrained Least Squares, then scissor the 3D mesh with the fitting plane. The vertexes which adjacent the intersecting line of the fitting plane with the model can be defined as feasible solution region of the appending vertex. Determine the direction function which can be used to determine the direction of the appending vertex within feasible solution region. There is similarity of curvature characteristic between the appending vertex and the vertex of segmentation line, curve degree characteristic can be used to obtain higher accurate appending vertex.

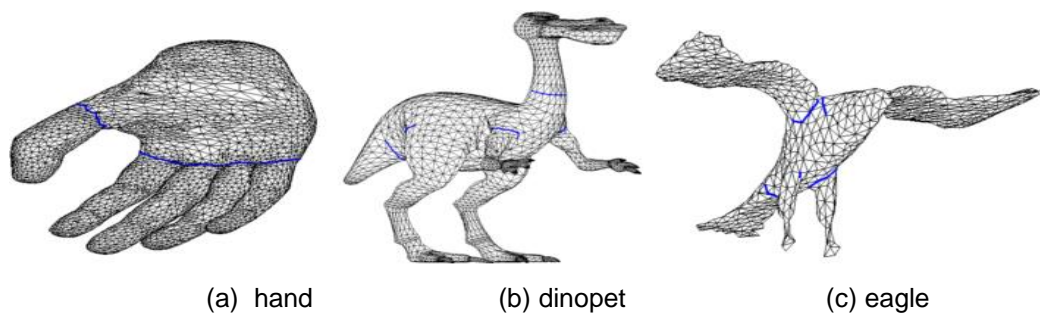


Figure 3. Segmentation Line

Feasible solution region Utilize the vertexes of the segmentation line to fit plane. The feasible solution region can be defined as vertexes set which each vertex has closest distance from vertex to the fitting plane. Let β is selected unclosed segmentation line, $n_1, n_2 \dots n_k$ are k vertexes of β . The plane fitting method utilized the coordinates of these k vertexes. The general expression for the plane equation is: $Ax + By + Cz + D = 0$ ($C \neq 0$), and $z = (-A/C)x + (-B/C)y + (-D/C)$. Note: $a_0 = -A/C, a_1 = -B/C, a_2 = -D/C$, and $z = a_0x + a_1y + a_2$. For all k vertexes ($k \geq 3$): $(x_i, y_i, z_i), i = 0, 1, \dots, k-1$, utilize points to fit plane, make the formula (8) minimum value.

$$S = \sum_{i=0}^{n-1} (a_0x_i + a_1y_i + a_2 - z_i)^2 \quad (8)$$

In order to make S the minimum value, $\frac{\partial S}{\partial a_n} = 0, n = 0, 1, 2$. That is to say:

$$\begin{vmatrix} \sum x_i^2 & \sum x_i y_i & \sum x_i \\ \sum x_i y_i & \sum y_i^2 & \sum y_i \\ \sum x_i & \sum y_i & n \end{vmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} \sum x_i z_i \\ \sum y_i z_i \\ \sum z_i \end{pmatrix}. \text{ Resolute this system of linear equations, we}$$

can get a_0, a_1, a_2 . The plane equation is $z = a_0x + a_1y + a_2$. Let $d_{v_i} = \text{dis}(v_i, z)$, if $d_{v_i} < \varepsilon$, append v_i to the feasible solution region E . As shown in Figure 4(a), β is the red line, Figure 4(b) is the feasible solution region of the contour.

Determine the direction The second feature is to determine the direction of the appending vertex. Let v^p and v^q are two endpoint of β . Select the one adjacent vertexes v_{1-ring}^p of v^p within E , if $v^q \in v_{1-ring}^p$, then we have found the loop include β ; otherwise choose the vertex from v_{1-ring}^p which has closest distance to the fitting plane as the appending vertex. But this method sometimes can not get the ideal vertex, as shown in Figure 4(c), the red points are along the correct direction, but the blue points is wrong although those points have more close distance, thus correct direction must be defined. Let $\eta_{v_i} = \cos(\overrightarrow{v_k v_p}, \overrightarrow{v_p v_i})$, $v^k \in \beta$, v^k is adjacent to v^p . v^i is adjacent to v^p too, but $v^i \in v_{1-ring}^p$. If $\eta_{v_i} > 0$, the angle between $\overrightarrow{v_k v_p}$ and $\overrightarrow{v_p v_i}$ is less than 90 degrees, the direction of v^i was basically consistent with β , otherwise v^i is opposite direction.

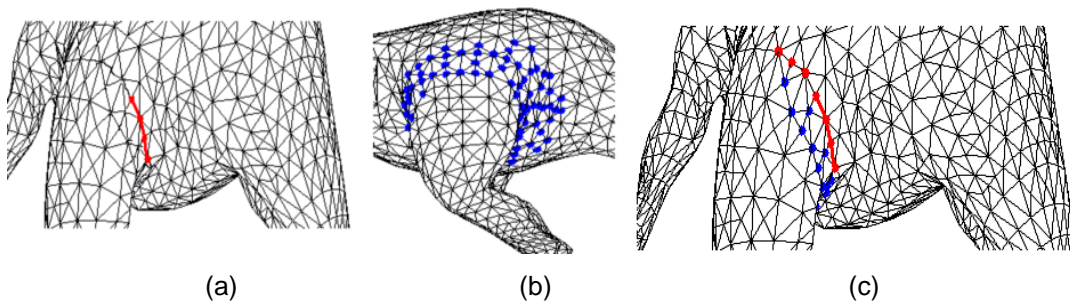


Figure 4. Contour Completion Algorithm

Curvature characteristic The third feature is the curvature of the appending vertex. Although the appending vertex is not necessarily belong to the concave area, it always has a large bending degree. The curvature degree value can better reflect the point bending

degree of the model, so according to formula 3 calculate the curvature degree values $c_v(r)$ of v_{1-ring}^p , and get curvature characteristic.

We select among candidate vertexes using the following vertex cost function: $cost(v_i) = \omega_1 \eta_{v_i} \square \omega_2 C_v(r)$, $v_i \in E$. ω_1 and ω_2 is the weights of directional characteristic η_{v_i} and curvature characteristic $C_v(r)$, values are normalized in the range [0:1].

5. The Experiment Results Analysis

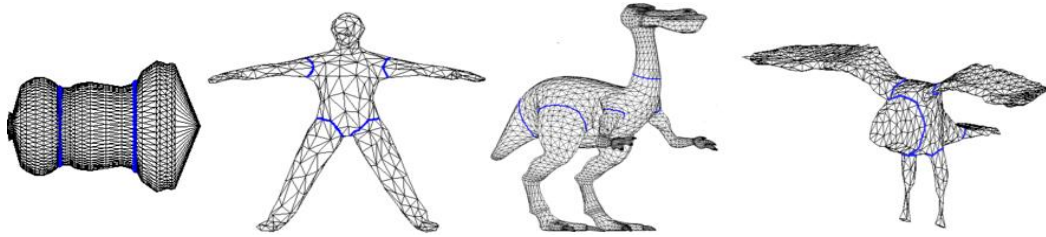


Figure 5. Closed Segmentation Line

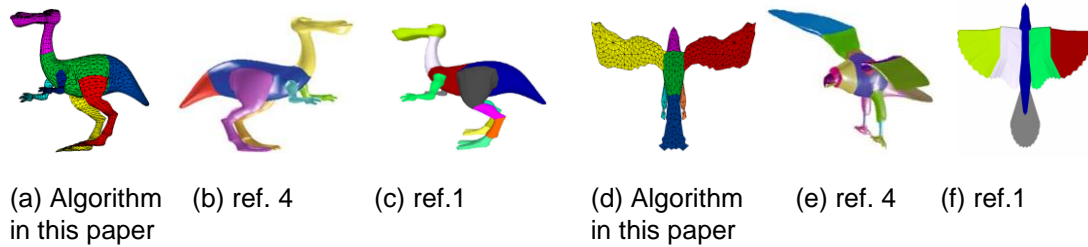


Figure 6. Segmentation Result Comparison

Our segmentation algorithm was evaluated in Windows Operation System, the testing environment is Intel Core i3 3.3GHz CPU, 4G internal memory. In order to compare quantitatively to the most recent algorithms, the Princeton benchmark[15] was used to our 3D-mesh segmentation evaluation. Take m1611 for example, use MDS(multi dimensional scaling)[11] which turn the model into pose-invariant, identify three feature points as prominent feature points, and carry out pre-segmentation. In Figure 2(a), the blue part is the patch to be segment $Patch(U)$, calculate the discrete curvature degree values of $Patch(U)$, as Figure 2(b), and get the curvature extremal points as segmentation line, as figure 2(c). According to segmentation line, we complete and form a closed loop around the 3D model, and get the final segmentation result. Figure 5 shows the closed segmentation lines of 3D models, Figure 6 shows the segmentation result comparison between this paper and reference. As shown in Figure 6 (a), the algorithm of this paper can segment dinopet into head and neck, limbs, torso, and tail. The experimental results of limbs, torso and tail are similar to ref.4, but the experimental results of ref.4 are more accurate in the head and neck. Because curvature degree values are all positive, ignore concave and convex feature of the surface. Thus the segmentation line is located on the extremal area of convex region. Ref.1 can segment 3D model hierarchically, in this paper, the number of partition is depend on the number of prominent feature point. Table 1 is the relevant experimental data about this paper. The pretreatment stage include the time of choosing the prominent feature points automatically and pre-segmentation of the 3D model. The post-segmentation stage include the calculating the curvature of $Patch(U)$, getting the segmentation lines and contour completion. Algorithm in this paper the geodesic distance calculation is the most time-consuming stage, the time complexity is $O(n^2)$, n is the vertex number of a model. The geodesic distance calculated only once,

does not need to compute them repeatedly during post-segmentation stage. The time complexity of post-segmentation is $O(C)$, C is the vertex number of $Patch(U)$, but $C \ll n$, thus the time complexity of this paper is $O(n^2)$. Because the algorithm of ref.1 can get hierarchical segmentation, the time complexity is $O(n^2 \log(n) + I n^2)$, I for k-means clustering iteration time. Obviously the speed of this paper is better than that of ref.1. The algorithm of ref. 5 had to calculate the curvature value of all the 3D model, larger amount of calculation, and easy cause over-segmentation because of the curvature sensitivity to the noise.

Table 1. The Experimental Model and Relevant Data

model name	vertex number	face number	prominent feature points number	pretreatment/s	post-segmentation/s
m1611	1474	2944	3	1.245	0.171
m235	408	812	5	0.332	0.043
Denopet	4500	8996	7	5.212	0.680
eagle	1000	1996	7	0.987	0.194
hand	4696	9388	6	5.208	0.703

6. Conclusion

The algorithm of this paper firstly obtaining the prominent feature points automatically and pre-segmentation the 3D model, and then using the patch to be segment $Patch(U)$ as the the constraint region, calculating discrete curvature within $Patch(U)$ and getting the curvature extremum points as the segmentation lines and then finally using contour completion algorithm achieve the end segmentation lines. The characteristic of geodesic distance is calculated only once and discrete curvature is calculated within a limited range, this improve the efficiency of the characteristics calculation and significantly reduce the effect of noise on the calculation of curvature. The algorithm can better achieve meaningful segmentation for 3D models with obvious branching structure.

In addition, the following works still to be considered: firstly, how to enhance the separability of 3D model without obvious branching structure according to the curvature change, and obtain the precise segmentation lines. Secondly, hierarchical segmentation algorithm should be considered. Thirdly how to construct a learning system for a group of relevant models and achieve consistent segmentation is the key issue to be focused.

Acknowledgements

We gratefully acknowledge the supported by Xian university of posts and telecommunications youth fund (ZL2013-28).

References

- [1] S. Shlafman, A. Tal and S. Katz, "Metamorphosis of polyhedral surfaces using decomposition", Computer Graphics Forum, vol. 3, no. 21, (2002).
- [2] V. G. Kim, W. Li, N. J. Mitra, S. Chaudhuri, S. DiVerdi and T. Funkhouser, "Learning part-based templates from large collections of 3D shapes", ACM Transactions on Graphics (TOG), vol. 4, no. 32, (2013).
- [3] Y. Wang, S. Asafi, O. van Kaick, H. Zhang, D. Cohen-Or and B. Chen, "Active co-analysis of a set of shapes", ACM Transactions on Graphics(TOG), vol. 6, no. 31, (2012).
- [4] D. Hoffman and W. Richards, "Parts of recognition", Cognition, vol. 1, no. 18, (1984).

- [5] G. Aleksey and F. Thomas, "Randomized Cuts for 3D Mesh Analysis", ACM Transactions on Graphics, vol. 5, no. 27, **(2008)**.
- [6] S. Katz and A. Tal, "Hierarchical mesh decomposition using fuzzy clustering and cuts", ACM, vol. 3, no. 22, **(2003)**.
- [7] H. Benhabiles, G. Lavoué, J. P. Vandeborre and M. Daoudi, "Learning Boundary Edges for 3D Mesh Segmentation", Computer Graphics Forum, Blackwell Publishing Ltd, vol. 8, no. 30, **(2011)**.
- [8] S. Xiaopeng, Z. Qi and W. Xiaopeng, "Semi-supervised 3D Mesh Hierarchical Segmentation", Journal of Computer-Aided Design&Computer Graphics, vol. 4, no. 22, **(2010)**.
- [9] Li. Han, G. Xiaoshan and C. Bingzhi, "Discrete Curvature Constrained Triangle Mesh Model Segmenting Technique", Journal of Computer-Aided Design&Computer Graphics, vol. 6, no. 21, **(2009)**.
- [10] P. Yang and X. Qian, "Direct computing of surface curvatures for point-set surfaces", Eurographics symposium on point-based graphics, **(2007)**.
- [11] M. Meyer, M. Desbrun, P. Schröder and A. H. Barr, "Discrete differential geometry operators for triangulated 2-manifolds", Visualization and mathematics, vol. 2, no. 3, **(2002)**.
- [12] D. Levin, "Mesh-independent surface interpolation", Geometric Modeling for Scientific Visualization, **(2003)**.
- [13] N. Amenta, Y. J. Kil, "Defining point set surfaces", ACM Transactions on Graphics (TOG), vol. 3, no. 23, **(2004)**.
- [14] Y. Lee, S. Lee, A. Shamir, "Mesh scissoring with minima rule and part salience", Computer Aided Geometric Design, vol. 5, no. 22, **(2005)**.
- [15] X. Chen, A. Golovinskiy, T. Funkhouser, "A benchmark for 3d mesh segmentation", ACM Transactions on Graphics (SIGGRAPH), vol. 3, no. 28, **(2009)**.
- [16] S. Katz, G. Leifman and A. Tal, "Mesh segmentation using feature point and core extraction", The Visual Computer, vol. 8, no. 21, **(2005)**.

Authors



Hui Jia

Author's profile.



Jiangang Zhang

Author's profile.

