

# Incomplete Surface Reconstruction for Statue Point Cloud based on Symmetry

Jie SHAO

*School of Art and Design, Nanchang Hangkong University, Nanchang, china  
e-mail: 2262570580@qq.com*

**Abstract**—Advanced 3D scanning technologies enable us to obtain a great mound of point clouds for 3D objects. Among these point clouds, many of them are incomplete, so reconstruction from them is difficult. To address this problem, a novel method of reconstruction from incomplete statue point cloud is proposed. The proposed method complete the missing parts of statue models before reconstruction using the symmetry of statue model. Then the point cloud is reconstructed by Power Crust algorithm. Experiments show that our method can reconstruct the statue point cloud efficiently.

## 1. Introduction

With the development of 3D scanning technologies, we are now able to obtain a great mound of real objects' surface, and modeling complex objects from samples becomes a significant recent trend in geometric modeling [1][2][3]. But many of the samples acquired with 3D scanners is incomplete. Moreover, in the fields, such as archaeology, ancient architecture, almost all of the scanned models are incomplete. Therefore, it is crucial that we develop efficient algorithms for surface reconstruction from incomplete point cloud.

Symmetry is an essential and ubiquitous concept in nature, science, and art [4], and has been exploited in a variety of geometry processing applications, such as remshing [5], segmentation [6], shape matching [7] and shape retrieval [8]. Extracting symmetry from 3D model is essential for understanding and manipulating the geometric structure of 3D object. Lipman et al. [9] detect and quantify symmetry in point sets by analyzing symmetries in correspondence space with spectral analysis. The method is able to extract symmetries in point set containing noise and missing data. Ghosh et al. [10] solve for the optimal deformation that reconciles a set of local bilateral symmetries and apply it in the symmetrization of fossils. Jiang et al. [11] use symmetry to reconstruct 3D model of building from a single image. Ceylan et al. [12] propose a framework for image-based 3D reconstruction of urban

buildings based on symmetry priors. They detect symmetric line arrangements from image-level edges.

In this paper, we propose to exploit symmetry features to address the data missing issues, and apply it in building 3D model from incomplete statue point cloud. As man-made objects, statue often exhibit strict symmetry features and we can use the symmetry to estimate the geometrical structure and appearance of missing area from existing data. Based on this observation, we first detect symmetry at global scale and adopt it to repair for the missing data. We then reconstruct the point cloud using Power Crust[13] algorithm. Experiments show that the proposed algorithm gives a good visual effect of the resulting reconstruction.

## 2. Extract Symmetry

We first extract global symmetries from point set. We limit the symmetry transformation to reflection symmetry since it is the most dominant symmetry category in ancient architectures. We follow the basic idea in [4] and adapt it to point sets obtained from multi-view reconstruction method. We evenly sample 1000 to 2000 points in the original point set and each pair of sampling points is viewed as a pair of potential symmetric points. In 3D space, the plane of symmetry determined by this pair of points can be expressed as  $Ax+By+Cz+D=0$ . We represent the plane using a 4D vector  $(A, B, C, D)$  which defines the reflection transformation space, and each pair is associated with such a 4D vector. By looking at the original point set, we can

deduce that all of the 4D vectors should be clustered around the 4D vector corresponding to the reflection transformation actually present in the object, since there are more pairs agree on the true reflection plane. In practice, the sampling point sets exhibits approximate symmetries instead of precise symmetries. Thus, we utilize affinity propagation clustering algorithm to explore the distribution of the 4D vectors and find clusters in the transformation space.

### 3. Affinity Propagation Clustering Algorithm

AP has been proved to be useful for many domains such as in face images, gene expressions and text summarization [14-15]. The main idea of this algorithm is regarding each data point as a node in network and recursively transmits messages among nodes until a good subset appear. AP takes a collection of real-valued similarities among data points as input, denotes  $S(i, k)$  as similarity between point  $i$  and point  $k$ . Similarity  $S(i, k)$  indicates that how well the data point  $k$  is suited to be the exemplar for data point  $i$ . Generally speaking, each similarity is set to a negative squared error. In point cloud simplification, set

$$S(i, k) = -(\|x_i - x_k\|^2 + \|y_i - y_k\|^2 + \|z_i - z_k\|^2) \quad (1)$$

Where  $(x_i, y_i, z_i)$  is the coordinate of point  $i$ . Another parameter of input is  $S(k, k)$ , it locates the diagonal of  $S$ , and shows probability of point  $k$  to be chosen as exemplar. Larger values of  $S(k, k)$  are more likely to be chosen as exemplars. Therefore  $S(k, k)$  are called 'preferences'. In this sense the numbers of exemplars are influenced by the values of the  $S(k, k)$ .

AP has two kinds of messages to exchange between data points, and each takes a different competition into account, i.e., 'responsibility' and 'availability'. The 'responsibility'  $R(i, k)$  sends from data  $i$  to candidate exemplar point  $k$ , and reflects the accumulated evidence for point  $k$  as the exemplar for point  $i$ . The 'availability'  $A(i, k)$  reflects the accumulated evidence for appropriate of point  $i$  to choose point  $k$  as its exemplar.  $R$  and  $A$  update and exchange messages between pairs of points with known similarities. After several iterations, some points are assigned to other exemplars, and availabilities of them will drop below zero by updated rules.

TABLE I. ALGORITHM AP

| Algorithm AP: |  |
|---------------|--|
| Input         | $S, P$ (similarities and preferences of point sets).   |
| Output        | $D$ (exemplars for point sets)   |
| Step1         | initialize $A(i, k)=0$ .   |
| Step2         | update $R, R(i, k) = S(i, k) - \{A(i, j) + S(i, j)\}$ ,<br>$R(k, k) = S(i, k) - \{A(i, j) + S(i, j)\}$ |
| Step3         | update $A$ ,   |

|       |   |
|-------|---|
|       | $A(i, k) = \min\{0, R(k, k) + \sum_{j \in \{i, k\}} \max(0, R(j, k))\},$ $A(k, k) = \sum_{j \in \{i, k\}} \max(0, R(j, k))$ |
| Step4 | iteratively update terminated after a fixed number of iterations or messages stay constantly.                               |

From the setp2 of algorithm AP, equal mark both add  $A(i, k)$  to the left and right of

$$R(i, k) = S(i, k) - \{A(i, j) + S(i, j)\}, \text{ we get}$$

$$R(i, k) + A(i, k) = S(i, k) + A(i, k) - \{A(i, j) + S(i, j)\} \quad (2)$$

The value of  $R+A$  shows the effectiveness of affinity points and decides the probability of point  $i$  as exemplar of point  $k$ . In iterations responsibilities and availabilities are both related to the input similarities, and they are combined to confirm exemplars.  $R(k, k)$  and  $A(k, k)$  are increasing along with the accretion of  $S(k, k)$ , and then the probability of point  $k$  as exemplar is increasing. In order to void oscillations, the algorithm introduces damp factor, and generally set  $\lambda=0.5$ . In each iteratively update, the formulas are changed as follows.

$$R^{(t)}(i, k) = (1 - \lambda) * \{S(i, k) - \max\{A(i, j) + R(i, j)\}\} + \lambda * R^{(t-1)}(i, k) \quad (3)$$

$$A^{(t)}(i, k) = (1 - \lambda) * \{0, S(k, k) + \sum_{j \in \{i, k\}} \max\{0, R(j, k)\}\} + \lambda * A^{(t-1)}(i, k) \quad (4)$$

$$A^{(t)}(k, k) = (1 - \lambda) * \{\sum_{j \in k} \max\{0, R(j, k)\}\} + \lambda * A^{(t-1)}(k, k) \quad (5)$$

Points send messages like Fig. 1.

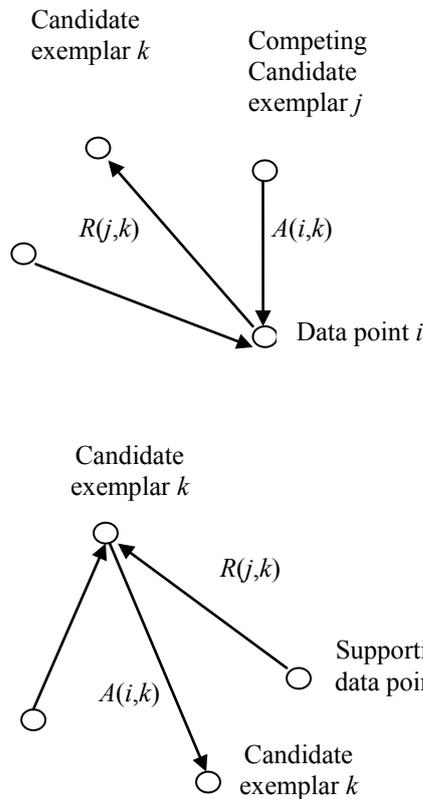


Figure 1. Sending Availabilities

#### 4. Reconstruction Algorithm

Our simplification algorithm is presented as follow:

TABLE II. Reconstruction Algorithm

| Reconstruction algorithm |   |
|--------------------------|---|
| Input                    | Original incomplete statue point cloud $D$                              |
| Output                   | Final statue point cloud $FD$   |
| Step1                    | Sampling the original point set, the percept is about 1/10              |
| Step2                    | Use AP clustering algorithm to cluster in symmetry transformation space |
| Step3                    | Use Power Crust algorithm to construct the point cloud                  |

#### 5. Experimental Results and Analysis

We experiment with the point-sampled model Twirl and Squirrel. The models both download from the web set of Stanford University computer graphics laboratory. Fig.3 illustrates the results of our proposed method on Twirl and Fig.4 illustrates the results of our proposed method on Squirrel.

From Fig.2, we can see, (a) is the original point clouds of Twirl, the number of the point is 7000. (b) is the original surface of Twirl model. In the original data, large part of data is missing in the right side, as shown in (c), the number of the remained points is 5235. (d) is the output of our algorithm running on the Twirl model. Our method still captures the symmetry accurately and completes the missing data successfully.

From Fig.3, we can see, (a) is the original point clouds of Squirrel, the number of the point is 9000. (b) is the original surface of Squirrel model. In the original data, large part of data is missing in the right side, as shown in (c), the number of the remained points is 6711. (d) is the output of our algorithm running on the Squirrel model. Our method still captures the symmetry accurately and completes the missing data successfully. Fig. 3 illustrates that the missing data in the ear, leg and feet of the Squirrel are completed.

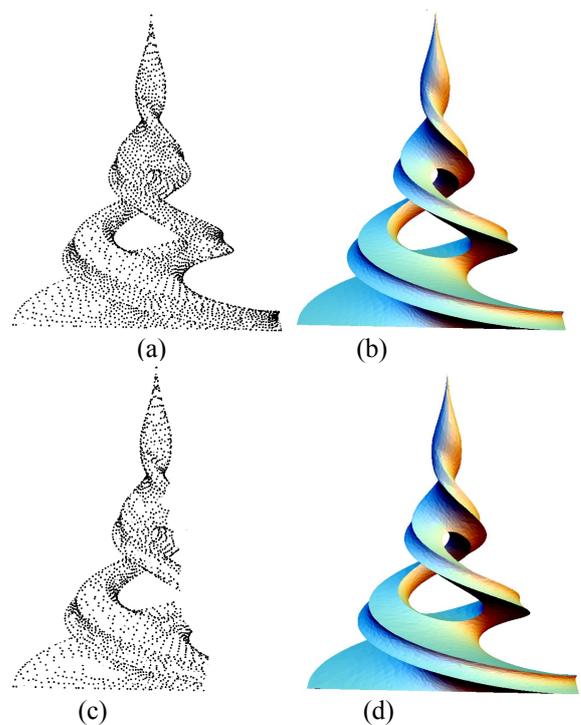
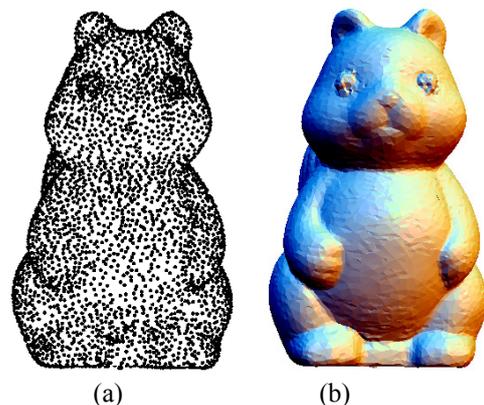


Figure 2. Simplification of Twirl and Squirrel model.



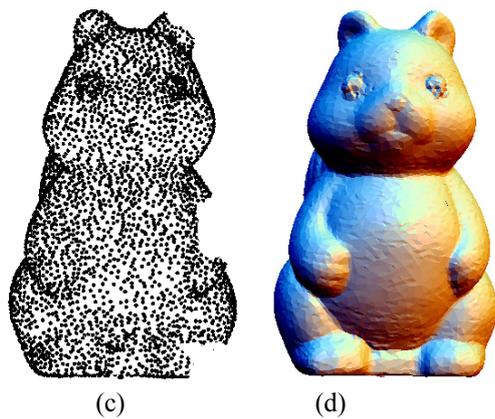


Figure 3. Simplification of Squirrel model.

Just as our expectation, almost all the missing parts of point-sampled models are reconstructed and the density of final point cloud is varied according to the surface's detail. For example, the samples are still very dense in the region like the top of Twirl and the eyes of Squirrel.

## 6. Conclusion

We have presented a novel reconstruction method for statue point-sampled models using symmetry. Our method first automatically extracts symmetries and completes the missing data using the extracted symmetry transformations. Next we use Power Crust algorithm to construct the statue point cloud. Experiments demonstrate the effectiveness of our method. However, for a surface without obvious symmetry plenty of textures, our method might incur visual artifacts. We plan to explore texture synthesizing techniques to achieve more realistic results in the future.

## References

1. Hyojoo Son, Changwan Kim, Kwangnam Ch. Rapid 3D object detection and modeling using range data from 3D range imaging camera for heavy equipment operation, *Automation in Construction*, 2010:19(7), 898-906.
2. Shengjun Liu, Charlie C.L. Wang. Orienting unorganized points for surface reconstruction. *Computers & Graphics*, 2010:34(3),209-218.
3. Wei-Cheng Xie, Xiu-Fen Zou, Jian-Dong Yang, Jie-Bin Yang. Iteration and optimization scheme for the reconstruction of 3D surfaces based on non-uniform rational B-splines. *Computer-Aided Design*, 2012: 44(1),1127-1140.
4. N. J. Mitra, L. J. Guibas, and M. Pauly, "Partial and approximate symmetry detection for 3d geometry," *ACM Trans. Graph.*, vol. 25,no. 3, pp. 560-568, 2006.
5. J. Podolak, A. Golovinskiy, and S. Rusinkiewicz, "Symmetry-enhanced remeshing of surfaces," in *Proceedings of the fifth Eurographics symposium on Geometry processing*, 2007, pp. 235-242.
6. T. Riklin-Raviv, N. Kiryati, and N. Sochen, "Segmentation by level sets and symmetry," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1,ser. CVPR '06*, 2006, pp. 1015-1022.
7. R. Gal and D. Cohen-Or, "Salient geometric features for partial shapematching and similarity," *ACM Trans. Graph.*, vol. 25, no. 1, pp. 130-150, 2006.
8. M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Symmetry descriptors and 3d shape matching," in *Proceedings of the 2004 Eurograph-ics/ACM SIGGRAPH symposium on Geometry processing*, 2004, pp.115-123
9. Y. Lipman, X. Chen, I. Daubechies, and T. Funkhouser, "Symmetry factored embedding and distance," *ACM Trans. Graph.*, vol. 29, no. 4,pp. 103:1-103:12, 2010.
10. D. Ghosh, N. Amenta, and M. Kazhdan, "Closed-form blending of local symmetries," *Computer Graphics Forum*, vol. 29, no. 5, pp. 1681-1688, 2010.
11. N. Jiang, P. Tan, and L.-F. Cheong, "Symmetric architecture modeling with a single image," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 113:1-113:8, 2009.
12. D. Ceylan, N. J. Mitra, H. Li, T. Weise, and M. Pauly, "Factored facade acquisition using symmetric line arrangements," *Computer Graphics Forum (EUROGRAPHICS)*, 2012.
13. N.Amenta, S.Choi, T.K.Dey and N.Leekha. A simple algorithm for homeomorphic surface reconstruction. *Internat. J. Comput. Applications*, 12, 2002, pp.125-141.
14. B.J.Frey and Delbert Dueck, "Clustering by Passing Messages Between Data Points", *SCIENCE*,vol.315, Feb 2007.
15. Ding-yin Xia, Fei WU, "Local and global approaches of affinity propagation clustering for large scale data", *Journal of Zhejiang University SCIENCE.*: PP 1373-1381, Sep 2008.