Robust Surface Consolidation of Scanned Thick Point Clouds

Xiaochao Wang
School of Mathematical Sciences
Dalian University of Technology
Dalian, China
wangxiaochao18@gmail.com

Xiuping Liu
School of Mathematical Sciences
Dalian University of Technology
Dalian, China
xpliu@dlut.edu.cn

Hong Qin
Department of Computer Science
Stony Brook University (SUNY Stony Brook)
Stony Brook, NY 11794, USA
qin@cs.sunysb.edu

***Abstract*—This paper proposes a consolidation method for scanned point clouds that are usually corrupted by noises, outliers, and thickness. At the beginning, we construct neighborhood of a point based on shared nearest neighbor relationship. Then, the points with few number of neighbors are regarded as outliers and removed. After that, we propose a feature-aware projection operator to thin the thick point clouds by considering spatial distances, normal diversifications, and the squash directions of thick point clouds. Experiment results of scanned point clouds show that our method can consolidate the thick point clouds while preserving sharp features and geometry details.**

***Keywords*-Consolidation; Thick Point Clouds; Feature-preserving Reconstruction**

I. Introduction

Scanned point clouds can be easily obtained and widely used in the field of computer graphics and computer aided design. Due to the limitations of scanners or surrounding environments, scanned point clouds usually contain outliers and noises. In point clouds processing, an important challenge is the thick point clouds, which may occur in the misalignment of multiple scan fusion. When reconstructing surfaces from these raw point clouds, rough, fragmental and non-manifold parts may appear on constructed models. The thicker the point clouds are, the more challenges we will face. Therefore, it is important to thin the point clouds before downstream geometry processing.

In the literature, a large number of methods have been proposed to deal with raw point clouds [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11]. Alexa et al. [1] proposed a moving least squares (MLS) based method, in which the raw points are projected onto the fitting surfaces. This method assumes the smoothness of underlying surface, hence it could not deal with sharp features. In order to deal with sharp features, more improved MLS methods have been proposed, for example, Lipman et al. [3] proposed a data-dependent moving least squares for surface reconstruction and Fleishman et al. [7] developed a robust moving least squares method based on robust statistics.

In [8], Lipman et al. proposed a locally optimal projection operator (LOP) method, which is robust to noise and outliers. Huang et al. [9] further improved the LOP method by introducing local density weight. Recently, Huang et al. proposed another anisotropic LOP algorithm using the normal projection distance as the weight parameter in [12]. Liu et al. [10] proposed an iterative method to consolidate the scanned point clouds. For outdoor scenes, Wang et al. [13] proposed a consolidation method of point clouds by combining outliers filtering and noise smoothing. Most of them assume that the acquired point clouds are densely sampled and without thickness. If the scanned point clouds contain thickness, the previous algorithms may fail.

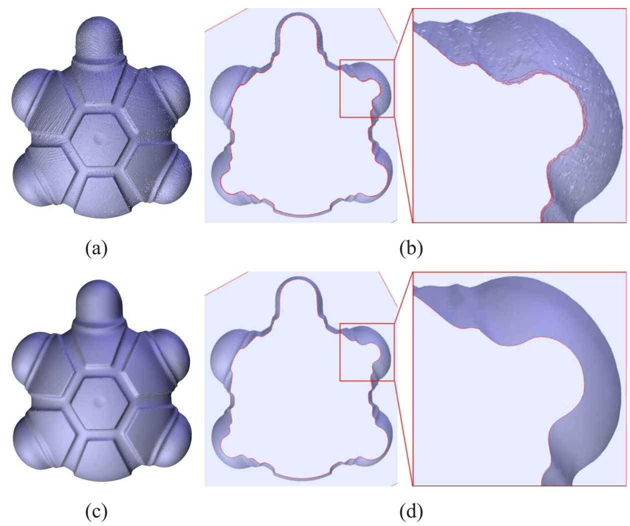

Figure 1. Thick point clouds consolidation. (a) and (c) Surfaces reconstructed directly from real scanned turtle toy model without and with our consolidation method, respectively. (b) and (d) Close-up cross-section views corresponding to (a) and (c). The side-by-side comparisons demonstrate the effectiveness of our consolidation algorithm, which not only removes noises while preserving features, but also successfully consolidates the thick point clouds into thin sheet.

To improve, in this paper, we develop a feature-preserving projection operator to consolidate the scanned point clouds. At first, some extreme outliers are removed by investigating the connection strength between the current vertex with their neighbors. Then, to reconstruct the unknown models and preserve the sharp features from the raw scattered point clouds, we propose an iterative scheme to thin the point clouds. The main contributions of our work are:

978-1-4799-2576-6/13 $31.00 © 2013 IEEE
DOI 10.1109/CADGraphics.2013.12

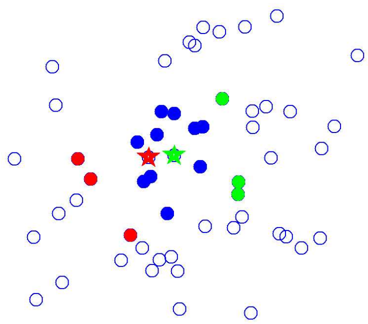

Figure 2. Illustration of shared nearest neighborhoods.

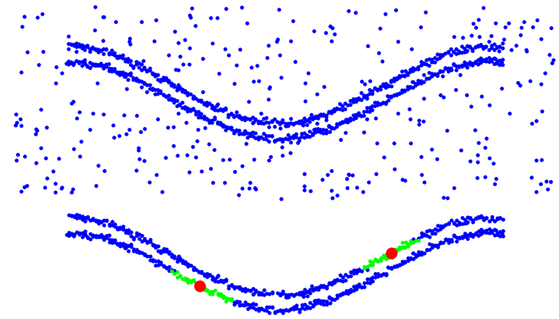

Figure 3. Neighborhood selection and outlier removal. Top: Close-by 2D point clouds with noises and outliers. Bottom: Point clouds with outliers removed and the neighborhoods of two red points are shown in green points.

- **Well-defined neighborhoods.** The neighborhoods of point clouds are well selected, which works well in noisy scanned point clouds.
- **Sharp and detail feature preserving.** Our algorithm can preserve the sharp features and geometry details during the process of consolidation.

The rest of the paper is organized as follows. Section II presents the scheme of neighborhood selection and proposes consolidation algorithm. Experimental results are illustrated in Section III. Finally, Section IV concludes the paper.

II. Algorithm

Given raw point clouds $P = \{p_i = (x_i, y_i, z_i)\} \subset R^3, i = 1, \cdots, n$, corrupted by noises, outliers, and thickness, we focus on obtaining clean and thin point clouds with the feature-preserving property. Our algorithm contains three parts: neighborhood selection, outlier removal, and thick point clouds thinning.

A. Neighborhood Selection and Outlier Removal

We first establish the neighborhood relationship for each point based on the shared nearest neighbor (SNN) clustering algorithm [14]. Then, the points with fewer neighbors are treated as outliers and removed.

To find a reasonable neighborhood for a point, we first measure the similarity of two points p and q by counting the number of shared nearest neighbors between them. The similarity of p and q is defined as:

$$S(p, q) = \#(NN(p) \cap NN(q)), \quad (1)$$

where $NN(p)$ and $NN(q)$ are the K-nearest neighbor sets of p and q. The more shared neighbors they have, the more confidence they locate in each other's neighborhood.

The illustration of shared nearest neighborhood is shown in Fig. 2, where the involved two points are marked in red and green stars. Their ten shared nearest neighbors are shown in blue with fourteen nearest neighbors used. Except the shared nearest neighbors, their individual remaining nearest neighbors are marked in the same color as the involved points.

After obtaining the similarities among points, the neighborhoods of the point clouds are established in a front propagation way. Specifically, for each point, a few number of nearest neighbor points are inserted into the front list, if they satisfy the following two conditions. First, the similarity between the added point and the current front point should be more than a given number, which depends on how many K-nearest neighborhoods are used (the default value is 40% of K). Second, the distance between them should be less than a specific threshold, which is usually set to several times of average distance within the entire data. The newly-added vertices become new front points by replacing the current front vertex. This propagation process is going on until there are no qualified candidates that can be searched or the maximum number of neighbors is reached.

Among the constructed neighborhoods, there are some points capturing few neighbor vertices, which may be far away from the main part of model or can not be confirmed from their surrounding neighbors. These points are naturally corresponding to outliers and should be removed together with their captured neighbors. The outlier removal result of the close-by 2D point clouds with noises and outliers is shown in Fig. 3, in which most outliers are effectively deleted. For remaining point clouds, repeat the above procedure, the neighborhood structures can be finally established. The pseudo-code of our neighborhood selection and outlier removal is documented in Algorithm 1.

During neighborhood selection, potential neighbor vertices are added in a probing way, only the vertex with the strong connection to the current vertex is selected. This scheme makes our method be able to obtain reasonable neighborhoods in close-by regions, which reflects the intrinsic structures of underlying surfaces and provides faithfully information for the subsequent processes.

B. Thick Point Clouds Consolidation

Till now, we have constructed the neighborhood structures of the input point clouds, which will be consolidated by the newly-devised method. The goal of our work is to

Algorithm 1 Neighborhood Selection and Outlier Removal

Input: data set $P \in R^3$

Initialization:

NN:K-nearest neighborhoods of P
h: average distance among P
for each $i \in P$ **do**
  $N\{i\} = []$: the selected neighborhoods
  $TFT = []$: temporary front set
  $FT \leftarrow NN\{i\}$: the propagation front set
  repeat
    for each $j \in FT$ **do**
      for each $k \in NN\{j\}$ **do**
        Compute distance $D(j,k)$
        Compute similarity $S(j,k)$
        if $D(j,k) \leq \lambda * h$ & $S(j,k) \geq \omega * K$ **then**
          $TFT \leftarrow k$
        end if
      end for
    end for
    $N\{i\} \leftarrow TFT - FT$
    update $FT \leftarrow TFT - FT$
  until Three times are reached
end for
Delete the points with the number of neighborhoods smaller than K

Output: The neighborhood selection of each point $N\{i\}$

consolidate the raw thick point clouds, which means squash the thick point clouds into thin sheets. To this end, with the estimated normals by traditional principal component analysis, feature-preserving projection operator for point $p \in P$ with neighborhoods $NB_p = \{q_i\}, i = 1, \cdots, m$ is defined as:

$$\begin{aligned} \hat{p} &= p + r \cdot n_p, \\ r &= \frac{1}{\sum_{q \in NB_p} w_q} \sum_{q \in NB_p} w_q \cdot (q - p) \cdot n_p, \\ w_q &= w_d \cdot w_{nn} \cdot w_{nd}, \end{aligned} \tag{2}$$

where n_p is the normal of point p and r is the step-size of displacement in normal direction. The weight w_q involves three geometric aspects, each of them is defined as:

$$\begin{aligned} w_d &= exp(-\frac{\|q-p\|^2}{2\alpha^2}), \\ w_{nn} &= exp(-\frac{(n_q \cdot n_p)^2}{2\beta^2}), \\ w_{nd} &= exp(-\frac{\|q-p\|^2 - ((q-p)\cdot n_p)^2}{2\gamma^2}), \end{aligned} \tag{3}$$

where w_d is a spatial weight in Gaussian filter with standard deviation parameter α, which determines the influence region inversely proportional to Euclidean distance. The second weight w_{nn} is a feature-preserving weight, with parameter β that penalizes the weight of a point with significantly different normal with n_p. The third term w_{nd} considers the special distribution of thick point clouds, which indicates that, the smaller vertical distance a point has towards the normal direction, the larger influence it will have, even though they may have the same Euclidean distance to the current point. This weight is controlled by the parameter γ.

Different from traditional projection operators, our new operator not only considers the spatial distance information and the normal diversification, but also emphasizes the importance of the squash directions of thick point clouds. These properties enable us to consolidate the thick point clouds into thin and clean data in a feature-preserving manner, which can be seen from the comparisons of our projection operator with bilateral denoising method [15] in 2D thick and noisy point clouds in Fig. 4. At the top of Fig. 4, the different weight distributions of bilateral method and our method are shown, in which our method assigns more weight to the points closing to normal direction.

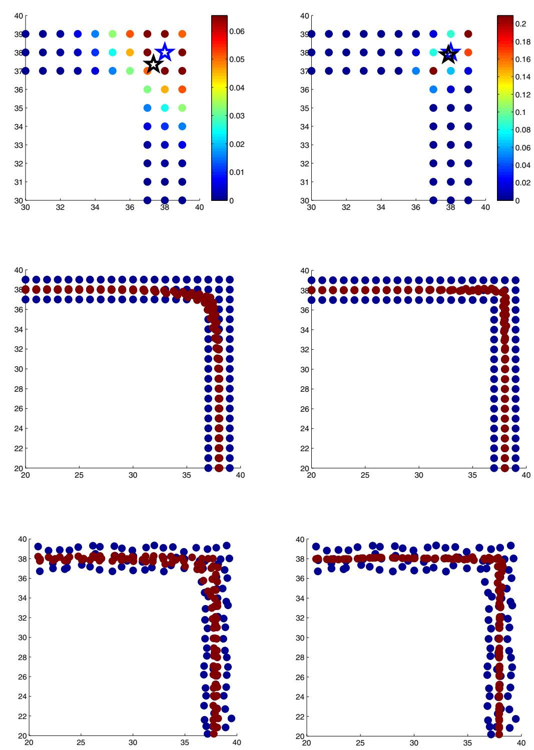

Figure 4. Illustration of projection weights and thinning results. Top left and right show the projection weight illustrations of bilateral method [15] and our method for blue star points, respectively. The projected points are shown in black stars. The middle and bottom rows show the thinning results of bilateral method (left) and our method (right) for noise-free and noisy thick point clouds, respectively.

III. Experimental Results

In this section, we apply our new consolidation method to raw scanned point clouds to show the efficiency and robustness of the proposed algorithm. To show the thinning results, the consolidated point clouds are reconstructed by the Ball Pivoting algorithm [16] and are converted into mesh models.

Thick point clouds consolidation. We first show a reconstructed turtle toy model from noisy, raw thick point clouds in Fig. 1. From the zoomed region and cross section of original reconstruction model in Fig. 1(b), we can see that the thick point clouds lead to multi-layer surfaces with noise. After consolidation using our method, the thin point clouds are obtained and the reconstructed model is presented in single sheet with noise removed (Fig. 1(d)).

In Fig. 5, we compare our method with Wang et al.'s method [13] on a raw scanned door knob model in terms of noise removal and thick point clouds thinning. Our proposed method can successfully handle the thick point clouds and heavy noises. Additionally, our method achieves better result than Wang's method, which can be observed from Fig. 5 together with the corresponding cross-sectional views.

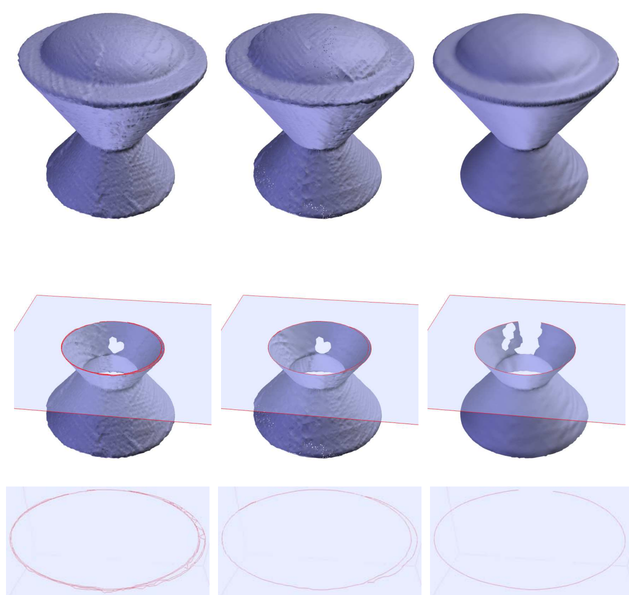

Figure 5. Consolidation results of scanned door knob model. Top left: Reconstructed surface of raw point clouds. Top middle and right: Consolidation results of Wang's method [13] and our method, respectively. The middle and bottom figures are corresponding cross-sectional views of top row.

In Fig. 6, the result of a scanned CAD model is used to further illustrate the effectiveness of our consolidation method in items of the property of sharp feature-preserving. The proposed operator considers the spatial distance information and the normal diversification, and in particular, it emphasizes the importance of the squash directions of thick

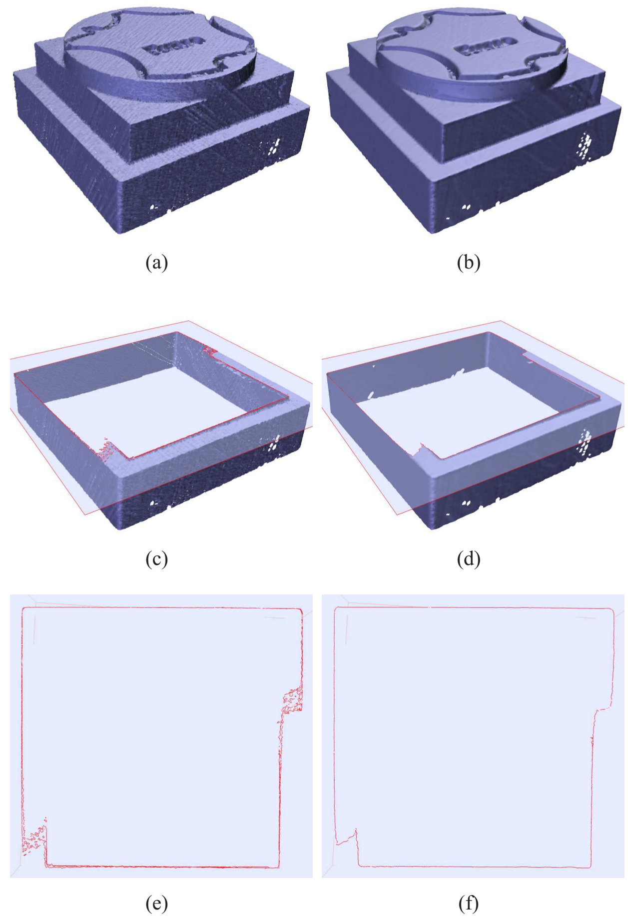

Figure 6. Results of our method being applied to scanned CAD model Nasa. (a) The reconstructed result of raw point clouds. (b) Our consolidation result. The cross sections corresponding to (a) and (b) are shown in (c-d) and (e-f), respectively.

point clouds, which makes our algorithm successfully deal with thick point clouds and preserve sharp features. The properties of the algorithm become more evident when we render cross-sections of two reconstruction surfaces (see Fig. 6(c-f)).

Our method can also preserve geometry details. In the same scheme, as shown in Fig. 7, the regular and explicit texture details are successfully recovered from noisy scanned golf ball model. Focusing on cross sections, we can observe that the raw thick point clouds are correctly thinned.

More results of our consolidation method are shown in Fig. 8 and Fig. 9, and they all demonstrate the attractive behavior of our method in thick point clouds thinning, while preserving sharp features and geometry details.

Parameters and Timing. In neighborhood construction, K-nearest neighbors of the front vertices are added at each time when they fall into specific distance threshold. The process of propagation will be repeated several times. We set a default value of $K = 25$ and repeat three times in our

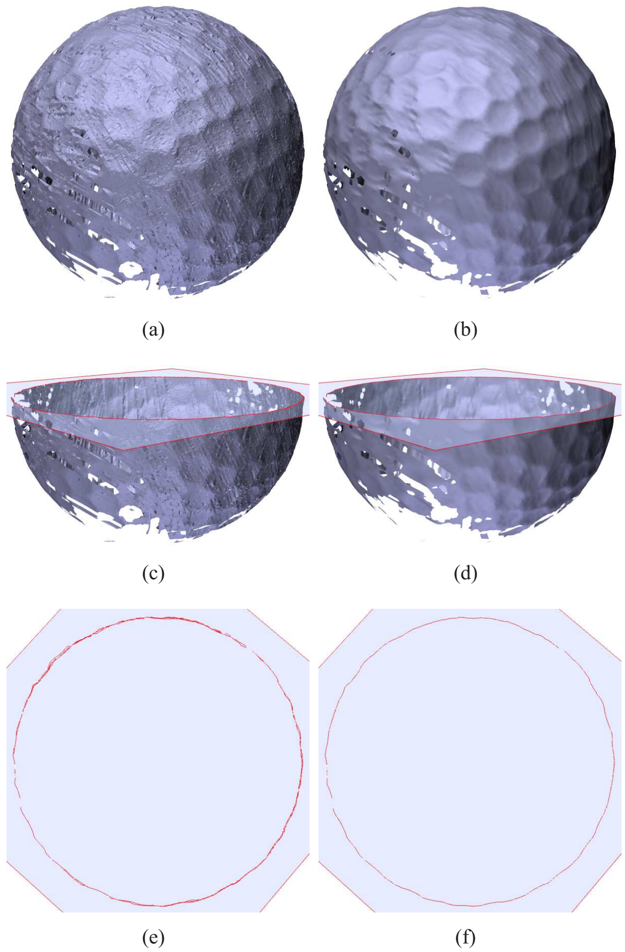

Figure 7. Consolidation results of scanned golf ball with regular texture details. (a) is the reconstructed surface from raw point clouds. (b) is the reconstruction surface after applying our method. The cross sections of (a-b) are placed in the middle and the bottom of the figures.

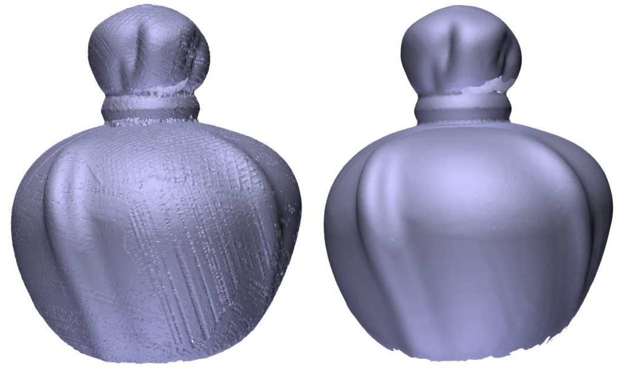

Figure 8. The consolidation results of perfume bottle model. The left one is the reconstruction result of original point clouds and the right one is the reconstruction result after consolidation.

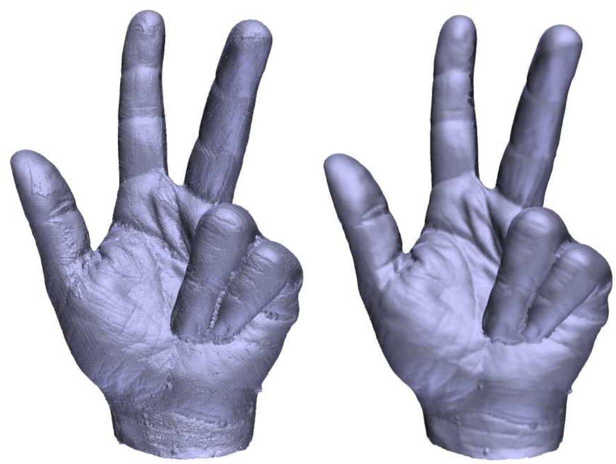

Figure 9. The consolidation results of hand model. The left picture is the reconstruction result of original point clouds and the right one is the reconstruction result after consolidation.

experiments, which works well for most cases. The choice of distance parameter depends on the sampling strategy of point clouds. If the sampling is very sparse, the distance threshold should be set larger, usually $[4-7]$ times of average distance among the whole data set will be utilized.

In the definition of projection operator, three parameters α, β and γ are adopted to control the influence range of each weight. Specifically, α controls the weight in terms of the Euclidean distances between neighbors and the current vertex. The smaller the distance is, the higher the value will be assigned. β is used to control the weight in terms of normal diversification, which relies on the sharpness of features. It will be set to a smaller value, such as $60°$, if a sharper feature is what we prefer. The third parameter γ emphasizes the importance of points with smaller distances orthogonal to the current normal, which makes our operator easier to “shrink” the thick point clouds into thin sheets. The parameters α and γ are related to the distance and are set to be the multiples of average distance of points in our work.

All results shown in our experiments are obtained after three iterations. The reconstruction parameters of Ball Pivoting algorithm are set by default values, i.e., clustering radius is 20% of ball radius and angle threshold is $90°$, while leaving the parameter of pivoting ball radius to be adaptive in different cases.

The proposed algorithm is implemented in MATLAB without any code optimization on a PC with 2.50Hz Intel Core CPU and 4.0GB RAM. All the timing statistics of tested models with used parameters are documented in Table I.

IV. Conclusion

In this paper, we have proposed a consolidation method for raw scanned point clouds. Neighborhood construction scheme is first developed in a front propagation way, then the points with few neighbors are removed as outliers. Con-

Table I
PARAMETERS AND RUNNING TIMES OF OUR METHOD IN MINUTES. NCT: NEIGHBORHOOD CONSTRUCTION TIME IN ONE ITERATION. NET: NORMAL ESTIMATION TIME IN ONE ITERATION. PT: PROJECTION TIME IN ONE ITERATION. TOTAL: TOTAL TIME AFTER THREE ITERATIONS.

Figures	Points	Parameters			Timing (m)			
		α	β	γ	NCT	NET	PT	Total
Turtle toy (Fig.1)	735914	5	60	4	5.87	1.37	4.63	39.97
Door knob (Fig.5)	189153	6	60	4	1.53	0.34	1.18	10.41
Nasa (Fig.6)	444312	4	60	2	3.81	0.82	2.84	24.94
Golf ball (Fig.7)	526921	4	60	2	4.39	0.94	3.62	30.41
Perfume bottle (Fig.8)	415587	5	60	3	3.59	0.80	2.81	24.05
Hand (Fig.9)	654658	3	60	2	5.19	1.22	4.39	36.58

sidering spatial distance information, normal diversification, and the squash directions of thick point clouds, we have designed a new projection operator to consolidate the raw scanned point clouds. We have tested our algorithm on real scanned point clouds to demonstrate the effectiveness of our method in thinning thick point clouds with the properties of sharp feature and geometry detail preserving. In the future, we plan to explore and add the hole-filling functionality into our consolidation framework.

ACKNOWLEDGEMENTS

All scanned point clouds used in this paper are courtesy of AIM@SHAPE Repository. The authors would like to thank Jun Wang for providing comparing results. This work is supported by the National Natural Science Foundation of China (No. u0935004, 61173102, 61190120, 61190121, 61190125, and 61370143), the scholarship from China Scholarship Council, and National Science Foundation of USA (IIS-0949467, IIS-1047715, and IIS-1049448).

REFERENCES

[1] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva, "Point set surfaces," in *Proceedings of the conference on Visualization '01*. Washington, DC, USA: IEEE Computer Society, 2001, pp. 21–28.

[2] B. Mederos, L. Velho, and L. de Figueiredo, "Moving least squares multiresolution surface approximation," in *Computer Graphics and Image Processing, 2003. SIBGRAPI 2003. XVI Brazilian Symposium on*, 2003, pp. 19–26.

[3] Y. Lipman, D. Cohen-Or, and D. Levin, "Data-dependent mls for faithful surface approximation," in *Proceedings of the fifth Eurographics symposium on Geometry processing*. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2007, pp. 59–67.

[4] G. Guennebaud and M. Gross, "Algebraic point set surfaces," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 23 – 31, Jul. 2007.

[5] H. Xie, J. Wang, J. Hua, H. Qin, and A. Kaufman, "Piecewise c1 continuous surface reconstruction of noisy point clouds via local implicit quadric regression," in *Visualization, 2003. VIS 2003. IEEE*. IEEE Computer Society Washington, DC, USA, 2003, pp. 91–98.

[6] H. Xie, K. McDonnell, and H. Qin, "Surface reconstruction of noisy and defective data sets," in *Visualization, 2004. VIS 2004. IEEE*. IEEE Computer Society Washington, DC, USA, 2004, pp. 259–266.

[7] S. Fleishman, D. Cohen-Or, and C. T. Silva, "Robust moving least-squares fitting with sharp features," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 544–552, Jul. 2005.

[8] Y. Lipman, D. Cohen-Or, D. Levin, and H. Tal-Ezer, "Parameterization-free projection for geometry reconstruction," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 22 – 27, Jul. 2007.

[9] H. Huang, D. Li, H. Zhang, U. Ascher, and D. Cohen-Or, "Consolidation of unorganized point clouds for surface reconstruction," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 176–182, Dec. 2009.

[10] S. Liu, K.-C. Chan, and C. C. L. Wang, "Iterative consolidation of unorganized point clouds," *IEEE Comput. Graph. Appl.*, vol. 32, no. 3, pp. 70–83, May 2012.

[11] H. Avron, A. Sharf, C. Greif, and D. Cohen-Or, "L1-sparse reconstruction of sharp point set surfaces," *ACM Trans. Graph.*, vol. 29, no. 5, pp. 135–146, Nov. 2010.

[12] H. Huang, S. Wu, M. Gong, D. Cohen-Or, U. Ascher, and H. R. Zhang, "Edge-aware point set resampling," *ACM Trans. Graph.*, vol. 32, no. 1, pp. 9–20, Feb. 2013.

[13] J. Wang, K. Xu, L. Liu, J. Cao, S. Liu, Z. Yu, and X. D. Gu, "Consolidation of low-quality point clouds from outdoor scenes," *Computer Graphics Forum*, vol. 32, no. 5, pp. 207–216, Aug. 2013.

[14] L. Ertöz, M. Steinbach, and V. Kumar, "Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data," in *Proceedings of the Third SIAM International Conference on Data Mining (SDM 2003)*, 2003, pp. 47 – 58.

[15] S. Fleishman, I. Drori, and D. Cohen-Or, "Bilateral mesh denoising," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 950–953, Jul. 2003.

[16] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 4, pp. 349–359, Oct. 1999.