IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 1, 2018, accepted February 1, 2018, date of publication February 27, 2018, date of current version March 28, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2805781

Ridge-Valley-Guided Sketch-Drawing From Point Clouds

YINGHUI WANG[1,2], HUANHUAN ZHANG[1,2,3], XIAOJUAN NING[1,2], WEN HAO[1,2], ZHENGHAO SHI[1,2], MINGHUA ZHAO[1,2], HONGFANG ZHOU[1,2], LIANSHENG SUI[1,2], AND KE LV[4]

[1]Institute of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China
[2]Shaanxi Key Laboratory of Network Computing and Security Technology, Xi'an 710048, China
[3]College of Electronics and Information, Xi'an Polytechnic University, Xi'an 710048, China
[4]School of Engineering Science, University of Chinese Academy of Sciences, Beijing 100049, China

Corresponding author: Huanhuan Zhang (zhanghuanhuan0557@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61472319, in part by the Shaanxi Natural Science Foundation under Grant 2017JQ6023, and in part by the Shaanxi Science Research Plan under Grant 2015JZ015.

ABSTRACT Sketch-drawing is one of the simplest and most direct means to illustrate 3-D objects. It can not only present geometric features, but also greatly facilitate us to identify and understand the object. This paper presents a simple yet effective tool to generate view-dependent sketch-drawing from point clouds. First, we extract the ridge-valley lines in geometric regions by utilizing their curvature; meanwhile, we extract the contours of point clouds based on the object view-dependent; the obtained ridge-valley lines and contours are optimized and fused to describe the geometric features of the object. Second, we apply the change models of line thickness diffusion based on angle and depth to finish the line thickness variation. Finally, the shadow region of sketch-drawing is achieved by using the shadow generation model based on line density and gray value. Experimental results show that our proposed method can effectively complete the object sketch-drawing simulation from point clouds. Furthermore, our method is more flexible and robust than the existing algorithms; it does not require the preprocessing on the input point clouds.

INDEX TERMS Point clouds, ridge-valley lines, contours, sketch-drawing.

I. INTRODUCTION

Sketch-drawing is one of the simplest and most direct means to illustrate 3D objects, which has attracted a lot of attention in computer graphics and other related fields in the past decade [1]. This art form has a high value in theory and economy, which is widely used in real-life applications, such as medicine, industry, geography, artistic design and prototype description [2]. Figure 1 shows some sketch-drawing application examples. Note that sketch-drawing only carries a fraction of line information, while the surface information such as color and texture are lost. However, human beings can easily understand the sketch-drawing at a glance. It can be seen that sketch-drawing can be used to eliminate unnecessary visual clutter and depict essential information, which greatly facilitate us to identify and understand the object.

In recent years, a large amount of works on computer-generated sketch-drawing algorithms have been proposed, which can be roughly classified into two categories: image-space methods [3], and 3D-object methods [4]–[6]. For image-space methods, image processing techniques are applied to present the feature lines of an object, and these methods are usually time-saving since they avoid the expensive computations of different properties on surfaces. However, the drawing details are sometimes not obvious due to a lack of the full depth information of the object [7], and always suffer pixel-level artifacts and are difficult for shape stylization [8]. Different from image-based methods, 3D-object methods can also be used to draw various feature lines of an object, such as silhouettes, contours and ridge-valley lines from 3D model with regular topology. Unfortunately, most of the published works usually require that the input model is smooth and has good meshes quality [9]. On the other hand, the availability of 3D scanning devices nowadays leads to the proliferation of dense, massive and unstructured point clouds which no connectivity information is initially available. Obviously, it can first apply a careful preprocessing step (e.g. denoising, remeshing, etc.) before applying the sketch-drawing algorithms. But converting such data set to optimal polygonal meshes is non-trivial and generally time-consuming.

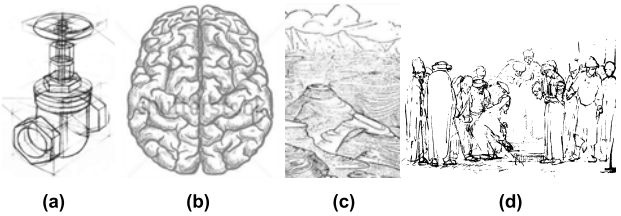


FIGURE 1. Application area. (a) Industry. (b) Medicine. (c) Geography. (d) Artistic design.

Recently point clouds have been touted by a growing number of researchers as the representation of choice for models of very high geometric complexity, typically obtained from laser range and optical scanners [10]. There are a few methods which achieve the silhouette extraction from point clouds [11], [12], but the line thickness changing and the shadow region generation under light conditions are not considered in those above methods. In our work, we explored a novel sketch-drawing method from the 3D raw point clouds and expanded the sketch-drawing application.

The rest parts of this paper are organized as follows: We briefly review the related work in Section 2. Section 3 presents the overview of our method. We present the algorithms to extract feature lines from point clouds in Section 4. Feature line diffusion is given in Section 5. Section 6 describes the shadow region generation. Experimental results are presented in Section 7. Finally, we conclude our work and highlight the future work in Section 8.

II. RELATED WORK

There exists extensive literature in computer-generated sketch-drawings and their applications. Rusinkiewicz *et al.* [1], Li *et al.* [7], and Zhang *et al.* [9], [13] provide good surveys. Here we only review those works related to 3D-object. We categorized them according to their methodologies.

A. SKETCH-DRAWINGS BASED ON SUGGESTIVE CONTOURS

3D-object sketch-drawing methods usually focus on how to extract the 3D model feature lines and how to describe the change of the feature line width to render the line art [4], [14]. DeCarlo *et al.* [15] described a new non-photorealistic rendering (NPR) technique based on suggestive contours which can naturally extend contours and convey the shape effectively. To address the problem that suggestive contours do not appear in the convex regions, DeCarlo and Rusinkiewicz [16] presented two new types of lines that convey highlights—suggestive highlights and principal highlights.

B. SKETCH-DRAWINGS BASED ON RIDGE-VALLEY LINES

Ridge-valley lines are powerful shape descriptors and widely used in sketch-drawing [5]. Judd and Adelson [8] introduced apparent ridges for non-photorealistic sketch-drawings, and this method capture important information about an object's shape. Aiming at sketch-drawing simplification, Ni *et al.* [17] proposed multi-scale sketch-drawings from 3D mesh models.

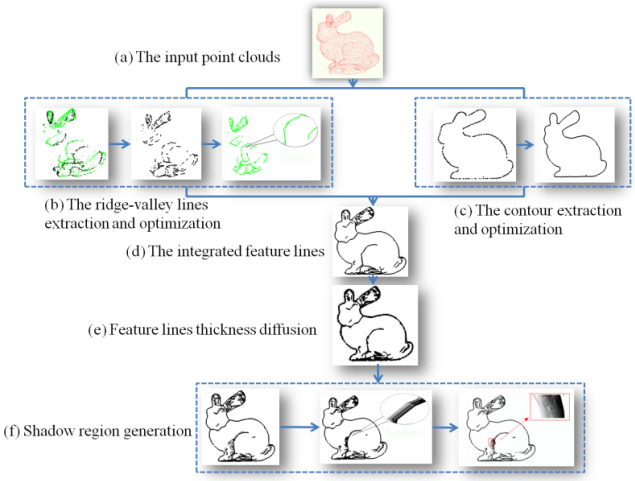


FIGURE 2. The framework of the proposed method.

C. SKETCH-DRAWINGS BASED ON EDGE DETECTOR

Inspired by the Canny edge detector, Xie *et al.* [18] derived photic extremum lines (PEL) on 3D surfaces. Zhang *et al.* [13] developed a new sketch-drawing technique based on the Laplacian-of-Gaussian (LoG) edge detector that can depict shapes with view-dependent feature lines in real time. This Laplacian line approach is numerically stable and did not suffer from precision issues. Inspired by the difference-of-Gaussian (DoG) edge detector, Zhang *et al.* [9] presented a simple approach to generate view-dependent sketch-drawings for 3D models, this method is robust and the lines are generated in a single pass.

Most of these study achievements show that the existing sketch-drawing methods can create similar results with hand-drawing from good meshes quality, and can express properties of the model very well. When these methods are implemented on the point clouds model, they need first preprocess the input point clouds model to obtain mesh with good quality. But, converting point clouds data set to optimal meshes is non-trivial and generally time-consuming.

III. OVERVIEW OF THE PROPOSED METHOD

Figure 2 illustrates the framework of the proposed method. It is composed of three main steps: 1) feature line extraction, 2) feature line thickness diffusion, and 3) shadow region generation.

A. FEATURE LINE EXTRACTION

First, we extract the ridge-valley lines (figure 2(b)) and contour (figure 2(c)) from point clouds, and connect and optimize these lines. Secondly, the optimized lines are fused to obtain the integrated feature lines (figure 2(d)).

B. FEATURE LINE THICKNESS DIFFUSION

Based on the extracted feature lines from the previous stage, we employ the proposed change models of line thickness

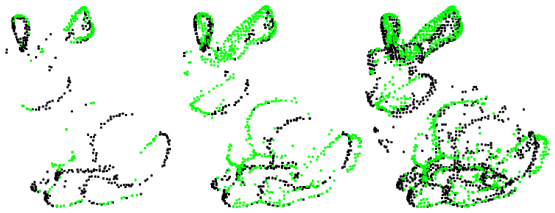


FIGURE 3. The parameter α controls the extracted number of ridge or valley points. The smaller the value of α, the more the number of ridge or valley points. (a) α=0.2. (b) α=0.1. (c) α=0.09.

FIGURE 4. The parameter r controls the connected ridge-valley lines length. The larger the value of r, the length the connected the ridge or valley segment lines. (a) r=0.1. (b) r=0.2. (c) r=0.3.

FIGURE 5. The optimized of ridge-valley lines.

diffusion based on angle and depth to finish the obtained feature line thickness diffusion (figure 2(e)).

C. SHADOW REGION GENERATION

At the final stage of our method, we introduce a novel shadow change model based on line density and gray value to generate the shadow region of the sketch-drawing (figure 2(f)).

IV. FEATURE LINES EXTRACTION

The critical questions for a computer-generated sketch-drawing method are: what kind of lines to extract, and where do we draw the lines [6]? Cole *et al.* [6] presented a comparative study in which artists made sketch-drawings to convey specific 3D shapes. They registered a large number of artistic drawings with rendered 3D models and then quantitatively compared the artistic drawings with computer-generated lines, such as contours, ridge-valley lines and Canny edges. In our work, we extract the ridge-valley lines (they are powerful shape descriptors [5]) and contours (they can express the model's outline and boundary shape information [7]) to represent the features of point clouds.

A. RIDGE-VALLEY LINES EXTRACTION

The ridge or valley detection method has been widely used by others before us. Pang [19] provides related methods. The lines we wish to capture correspond to ridges or valleys from point clouds. Firstly, we calculate each point curvatures via the least squares method [20]. The maximum curvature $k_{\max}$ and the minimum curvature $k_{\min}$ are found by traversing all the curvatures. Secondly we detect ridge or valley points. p_i is the ridge point when the curvature of p_i point meet the requirements of $k_i < 0, k_i < \alpha k_{\min}, 0 < \alpha < 1$($k_i$ is the curvature of p_i, α is threshold parameter), p_i is the valley point when the curvature of p_i point meet the requirements of $k_i > 0, k_i > \alpha k_{\max}, 0 < \alpha < 1$. The extracted ridge or valley points are shown in Figure 3, where the ridge or valley points are vary from sparse to dense with the parameter α decreases. The green color points are invisible ridge or valley points in Figure 3.

Then, we connect the obtained ridge points and valley points respectively. We first select a point as the starting point and choose the set of neighborhood points ($NBHD(p_k) = \{p_j, \|p_j - p_i\| < r, j = 0, 1, \cdots, k\}$) in the circle with r radius as the initial growth point. Secondly, the direction vectors are calculated using Principal Component Analysis, and the neighborhood points are projected onto these vectors. The distance of two projected points stops growing until we cannot find the next growth point. In this way, we can obtain the connected ridge-valley lines, as shown in Figure 4, where it can be seen that the larger the value of neighborhood radius r is, the longer the connected the ridge or valley segment lines are.

The connected ridge-valley lines are often fractured due to the points of the model with noise. To obtain the unbroken ridge-valley lines, the connected ridge-valley lines need to be optimized. In our work, we first get the tangential vector T through the vector at the vertex of the ridge-valley lines. Given the endpoint p in a ridge-valley line, and there existed another point q within the effective connection space meanwhile q is the endpoint of another ridge-valley line, furthermore, the tangential directions of the two endpoints are opponent, we connected the two endpoint of ridge-valley lines. Each endpoint of the ridge-valley line has an effective connection space, which is defined by a distance threshold αr, a tangential vector T and a predetermined angle β, where α is the distance threshold parameter, r is neighborhood radius with connecting the valley lines. The optimized of the ridge-valley lines are obtained according to the above methods, the results are shown in Figure 5.

B. CONTOUR EXTRACTION

Although the ridge-valley lines can describe the details of the objects, the ridge-valley lines themselves cannot generate the pleasing sketch-drawing. In order to better generate the perfect sketch-drawing from the point clouds, it is needed to extract the contour. Firstly, the model is projected on two-dimensional plane under a specific viewpoint to obtain a two-dimensional point set. Then, the left points of the projected model are selected as the starting point of the boundary point, and the local coordinate system is

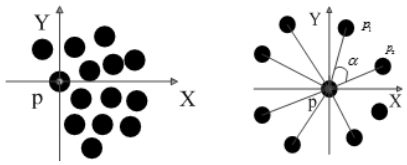


FIGURE 6. The local coordinate system.

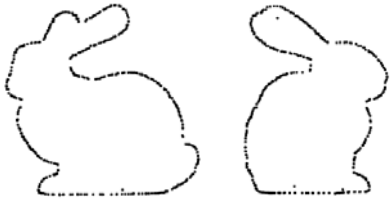


FIGURE 7. The obtained contour in different viewpoints.

FIGURE 8. Extraction of contour lines.

established. We used K-Nearest Neighbor (*KNN*) method to find the set $(NBHD\,(p_k) = \{p_j, \|p_j - p_i\| < r, j = 0, 1, \cdots, k\})$ of k neighbors and divided the region of p_k into four quadrants of the established local coordinate system as shown in Figure 6. The point p is the boundary point, if there are no neighbor points in one of the four quadrants. Otherwise we traversal the angle between neighbor points to the point p vector in $NBHD\,(p_k)$. When the angle is greater than a certain threshold, the point p is the boundary point. When the viewpoint changes, the point clouds model is rotated, and the rotation of the three-dimensional point cloud coordinate changes. Finally, the contour of point cloud model in a certain viewpoint is obtained as shown in Figure 7.

According to the obtained contour points, we can easily find that some interference points or noise points often appear around the normal boundary points. Figure 8 (a) shows the obtained contour points and the circles indicate interference points which often affect the correctness of the contour line connection. The Figure 8(b) shows a simplified outline of contour points, and c point is the interference point. According to the shortest distance connection method, the distance between the interference point c and the normal point p is less than the distance between the normal point d and the normal point p, then the point p and the point c are connected, this result in unpleasing counter.

To avoid the situation that the nearest point is the interference point, we select an edge point p as the initial growth point and select the point p_{front} near the point p in contour

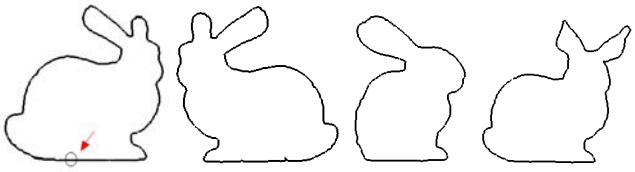


FIGURE 9. Contour lines of different view.

FIGURE 10. The integrated feature lines.

point set A. The direction vector is determined via the formula $n_p = p_{front} - p$. We apply *KNN* method to find the set $(NBHD\,(p_k) = \{p_j, \|p_j - p_i\| < r, j = 0, 1, \cdots, k\})$ of k neighbors, and sort the neighborhood points $NBHD\,(p_k)$ from small to large according to the distance between $NBHD\,(p_k)$ and p, that is $NBHD(p_k) = \{p_1 < p_2 < p_3 \cdots < p_k\}$. Finally, We select the point between which and the direction vector the distance is less than given the radius threshold $z_{\max}$ as the next initial growing point in $NBHD(p_k)$, and delete the point A in $NBHD(p_k)$, it stopping searching for the next growing point until the contour point set A is empty. The optimized of contour line is shown in Figure 9.

Ridge-valley lines can reflect the shape features under different viewpoints, while the contours are formed by the two-dimensional point sets under a specific perspective. Therefore, we should integrate the feature lines (ridge-valley lines and the contours) in the same coordinate system and the same model, and judge the visible of the ridge-valley lines by using ray tracing algorithm. The integrated feature lines are shown in Figure 10.

V. FEATURE LINES THICKNESS DIFFUSION

Based on the extracted feature lines from the previous stage, we employ the proposed change models of line thickness diffusion based on angle and depth to represent the line width.

A. DEPTH-BASED MODEL OF LINE THICKNESS DIFFUSION

According to the principle of painting art, a stroke close to the viewpoint is thicker than a stroke farther away from the viewpoint. In our work, we choose the minimum point as the initial viewpoint on Z axis. And the straight line along the positive direction of the z axis is the line of sight. We traverse the values of z coordinate to get the maximum value $z_{\max}$ and the minimum value $z_{\min}$. Then, the value of z is divided into different segments, each segment can be mapped to different thickness via (4), and the thickness t_p of the z point can be obtained. Finally, the width of the segments between the two

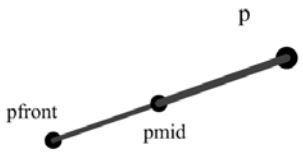


FIGURE 11. Segment thickness change model.

points is calculated according to the thickness value of two adjacent points. There is a segment $p_{front}p$ connected by p_{front} and p, p_{mid} is the middle point of segment $p_{front}p$, as shown in Figure 11. The width value of lines $p_{front}p_{mid}$ and $p_{mid}p$ can obtained according (4) and (5) respectively.

$$(z_{\max} - z_{\min})\,\eta + z_{\min} = z_p \tag{1}$$

$$(\rho_{\max} - \rho_{\min})\,\eta = t_p \tag{2}$$

$$t_p = \frac{z_p - z_{\min}}{z_{\max} - z_{\min}}\,(\rho_{\max} - \rho_{\min}) \tag{3}$$

$$t_{front,mid} = \left(t_{front} + \frac{t_p}{2}\right) cofi \tag{4}$$

$$t_{mid,p} = \left(\frac{t_{front}}{2} + t_p\right) cofi \tag{5}$$

Where $\rho_{\max}$ is the maximum thickness, $\rho_{\min}$ is the minimum thickness, $cofi$ is the control factor, η is parameter of controlling the lines thickness.

B. ANGLE-BASED MODEL OF LINE THICKNESS DIFFUSION

As shown in Figure 1, the line width edge of the sketch-drawing is becoming coarser when the degree of curvature changed greatly. The line width of the sketch-drawing is becoming narrower when the degree of curvature is trivial changed. In this paper, an angle-based line thickness algorithm is designed. Firstly, we calculate the angle between the point p and the two adjacent points, p_{front} and p_{next}, via (10), as shown in Figure 12. Then the thickness level value of certain point is found according to the defined angle (As shown in Table 1). Finally, the line between two adjacent points is divided into two parts according to the midpoint, and each segment width is calculated by (10) and (11).

$$n_{p_{front}p} = p_{front} - p \tag{6}$$

$$n_{p_{next}p} = p_{next} - p \tag{7}$$

$$\cos\alpha = \frac{n_{p_{front}p} \cdot n_{p_{next}p}}{\left|n_{p_{front}p}\right| \left|n_{p_{next}p}\right|} \tag{8}$$

$$\alpha = \cos^{-1} \frac{(p_{front} - p) \cdot (p_{next} - p)}{|p_{front} - p|\,|p_{next} - p|} \tag{9}$$

$$t_{front,mid} = \left(t_{front} + \frac{t_p}{2}\right) cofi \tag{10}$$

$$t_{mid,p} = \left(\frac{t_{front}}{2} + t_p\right) cofi \tag{11}$$

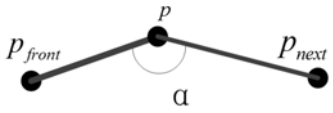


FIGURE 12. Angle thickness change model.

TABLE 1. Angle level.

α	$0<\alpha\le\frac{\pi}{9}$	$\frac{\pi}{9}<\alpha\le\frac{\pi}{6}$	$\frac{\pi}{6}<\alpha\le\frac{2\pi}{9}$	$\frac{2\pi}{9}<\alpha\le\frac{5\pi}{18}$	$\frac{5\pi}{18}<\alpha\le\frac{\pi}{3}$
t_α	10	9	8	7	6
α	$\frac{\pi}{3}<\alpha\le\frac{4\pi}{9}$	$\frac{4\pi}{9}<\alpha\le\frac{5\pi}{9}$	$\frac{5\pi}{9}<\alpha\le\frac{2\pi}{3}$	$\frac{2\pi}{3}<\alpha\le\frac{5\pi}{6}$	$\frac{5\pi}{6}<\alpha\le\pi$
t_α	5	4	3	2	1

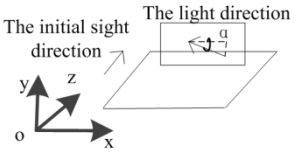


FIGURE 13. The light direction.

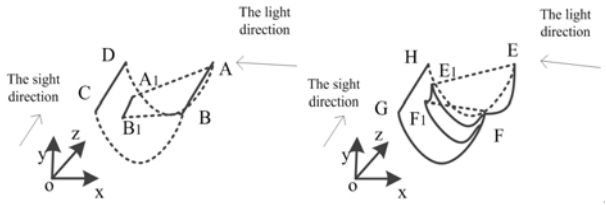


FIGURE 14. Shaded area.

VI. SHADOW REGION GENERATION

The shadow region is an indispensable part of the sketch-drawing. In our work, it consists of three steps to generate the shadow region. Firstly, the shadow region model is defined, and then the line density of the shadow region is calculated. Finally, the gray value of lines is expressed.

A. SHADOW REGION DEFINITION

We should set the initial sight direction along the positive of the z axis, and choose the light direction which is generated the angle of 45 degrees with the xoy plane and parallel to the xoz plane, as shown Figure 13. The light passes through a certain ridge point to generate a projection point in the concave-convex area on the model. The projection lines generated by the ridge lines are connected for forming a projection area. We should judge whether the projection area is obscured part and define the visible projection area.

A simplified diagram of a concave curve is shown in Figure 14. The ridge line consists of a plurality of connected broken line segments, which can approximate a curve. Therefore, the shadow region of one of the broken line

FIGURE 15. Shaded region definition.

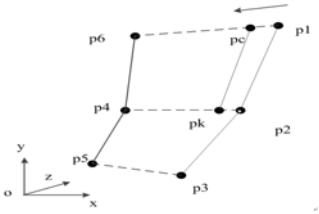


FIGURE 16. Shadow lines description.

segments can be extended to the entire ridge line. The curved *ABCD* surface is a concave curve, *AB* is the ridge lines on the curved surface. The projection point of A is A1, the projection point of point B is B1, and the projection point segment is A1B1. The projection line of the ridge line EF is E1F1, the curve EE1 is the line of intersection of the plane and the curve surface where the line segment EE1 is parallel to the light, and the curve FF1 is the line of intersection of the plane and the curve surface where the line segment FF1 is parallel to the light. The curve surface EFF1E1 is the projection area. Then we use the ray trace algorithm to determine the unobstructed area of curve surface EFF1E1 along the line of sight, this area is the shadow region which we can see in line of sight area. The shadow region is projected on the two-dimensional plane and the contour is integrated. The shadow region is defined in Figure15.

As shown in Figure 16, the projection point of p_1, p_2, p_3 are p_6, p_4, p_5 respectively and we calculate the direction vector n of the shadow lines p_1p_6 by Eq (13). Then the ridge-valley point p_1 moves a certain distance along the direction vector t to get the p_c point by Eq (13). Similarly, we calculate the direction vector n of the shadow lines p_2p_4 bby Eq (12) and obtain the point p_k. After that, we connect the two points p_c and p_k to form a segment p_cp_k which is a shadow line of a ridge-valley line. In this way, we can obtain the multi shadow line in one ridge-valley line. Due to the shadow lines are become denser when distance from the ridge valley line becomes closer. As shown in Eq (14), the value of given t is non-linear growth.

$$n = p_6 - p_1 \tag{12}$$

$$p_c = p_1 + t.n \tag{13}$$

$$t = k \times d, \quad k \in \left\{\frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}\right\} \tag{14}$$

Where n is the direction vector, t is the distance value, d is the parameter of controlling the line gray value, k is the parameter of controlling the line density.

FIGURE 17. Graduated color description.

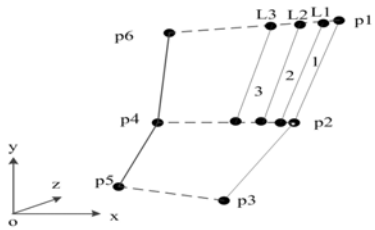


FIGURE 18. Brightness region division.

Finally, the region close to feature lines is dark and the region farther away feature lines is bright. In our work, we construct a graded grayscale model (as shown in Figure 18) where the pixel values include white and gray with different degrees of grayscale.

B. THE GRAY VALUE OF LINES EXPRESSION

The shadow region of a ridge-valley line is divided into n blocks and the gray value of each block is defined according the distance of the feature lines to generate the change of the grayscale value in shadow region. As shown in figure 19, the ridge-valley L line is composed of segments p_1p_2 and p_2p_3.For the ridge-valley p_1p_2, a plurality of hatched lines l_1, l_2 and l_3 are calculated under a specific light. The line l_2 and p_1, p_2 constitutes a closed quadrilateral, meanwhile the quadrilateral is marked as area1. We do it the same as to area1 the area 2 and area 3 are marked. Area 1 is darker than area 2, area 2 is darker than area 3.

VII. EXPERIMENTAL RESULTS AND DISCUSSION

The proposed method is implemented on a standard workstation equipped with Intel(R) Core (TM) 2Duo CPU with 4 GB of main memory, NVIDIA graphics card, and Windows 7.1 OS. We use OpenGL and Visual C++ to realize our method. The most of datasets adopted in our implementation is freely available and the detailed information can be found in Aim@shape [21].

Different 3D point clouds models are tested to illustrate the effectiveness of our method. Figure 19 shows different rendered models using our method, including the horse, bird, maxplanck, camel, rocker and lucy models. Figure 19(b) is the contour extraction results. Figure 19 (c) is the ridge-valley lines extraction results. Figure 19(d) is our result by combining contour lines and ridge-valley lines, and figure 19(e) is our result of the ridge-valley lines and contour line diffusion. Figure 19 (f) is the final sketch-drawing results. We find that our sketch-drawing method can adequately depict each object's local information and global shape. From Figure 19 (b) (c) (d), it can be found that our method can effectively extract the ridge-valley lines and the contour line from point

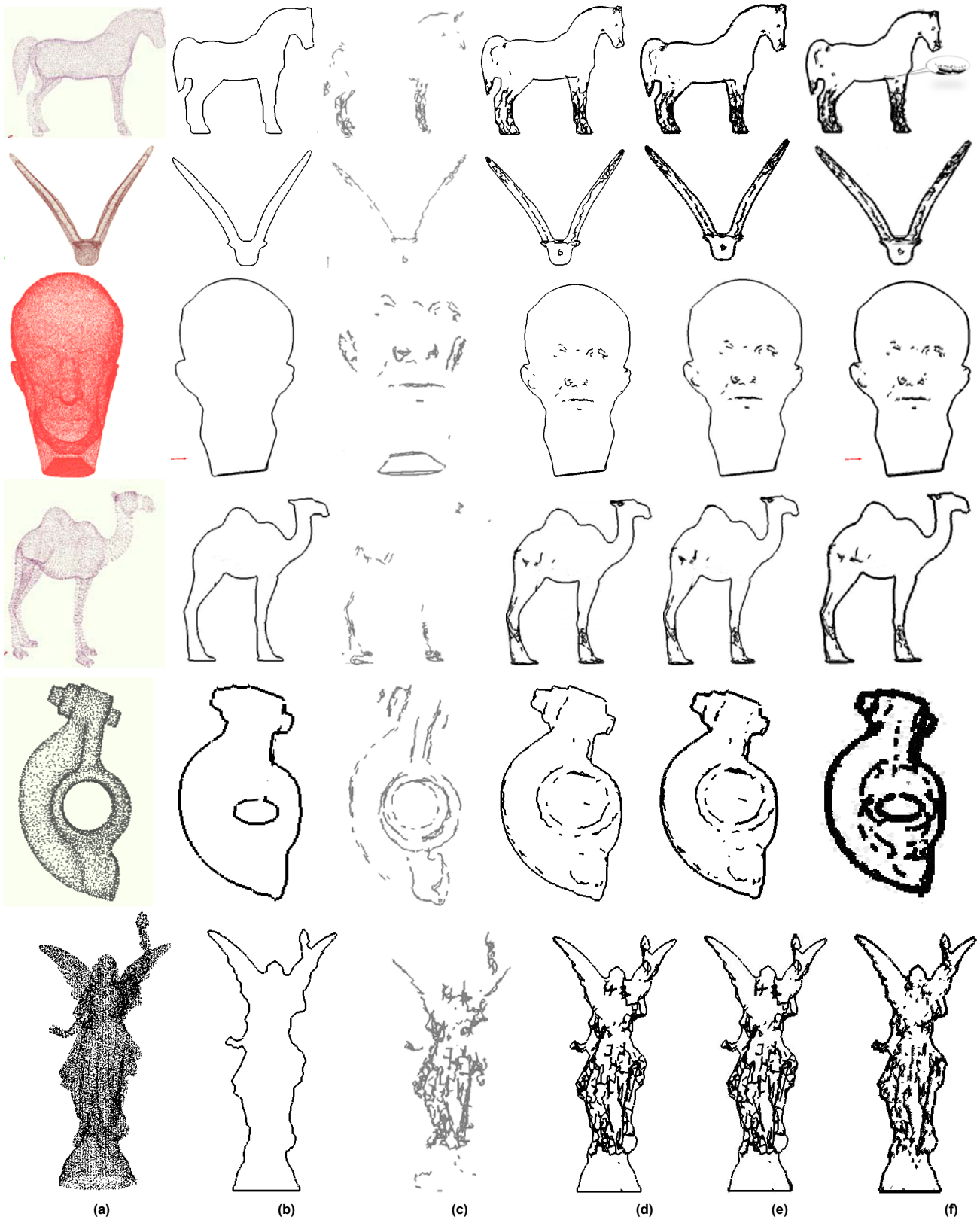


FIGURE 19. The sketch-drawing result from point clouds (a)the raw point clouds (b)The countor line extraction (c)The ridge-valley lines extraction (d)The ridge-valley lines and countor line intergation (e) The ridge-valley lines and countor line diffusion (f) The shadow region generation.

TABLE 2. Model – time contrast.

model / time(s)	Bunny	Bird	maxplanck	Horse	Camel	Rocker	Buffalo	Lucy
The number points	5326	11790	49132	7268	6603	5658	10850	17915
ridge-valley points extraction	2.715	5.375	25.715	3.682	3.247	2.945	5.375	9.674
ridge-valley points connection	0.338	0.738	4.338	0.591	0.586	0.432	0.738	2.976
contour points extraction	1.810	3.818	20.810	2.321	1.925	1.754	3.818	6.880
contour line connection	0.367	1.129	10.367	0.927	0.689	0.596	1.129	6.579
Drawing	1.521	2.373	12.521	1.818	1.875	1.547	2.373	5.847
Total (s)	7.151	13.782	73.751	9.864	8.322	7.274	13.433	31.956

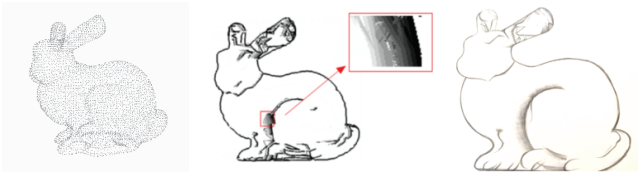


FIGURE 20. Comparison with traditional drawings. (a) the raw point clouds (b) our method (c) sketch-drawing by an artist.

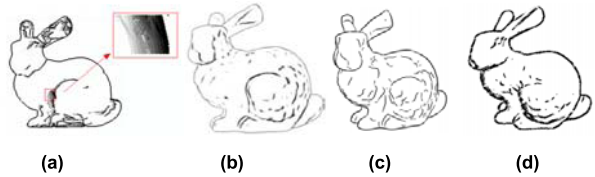


FIGURE 21. Comparison with other sketch-drawing methods (a) Our method (b) Difference-of-Gaussian (c) Suggestive Contours (d) Lines from diffuse shading.

cloud model. The ridge-valley lines and contour are fused after excluding the invisible ridge-valley lines to obtain the better integration results. We employ the proposed change models of line thickness diffusion based on angle and depth and add a brush model to finish the lines thickness diffusion. From Figure 19 (e), we can clearly find that a stroke close to the viewpoint is thicker than a stroke farther away the viewpoint. The line width of the sketch-drawing is becoming wider with the degree of curvature increasing. Otherwise, the line width of the sketch-drawing is becoming narrower.

In our work, we analyze the algorithms that affect the time complexity. In the process of sketch-drawing, the ridge-valley points and extracting contour points extraction are directly affects the time efficiency of the whole sketch-drawing as shown in Table 2.

In order to test the capability of our method, we compare our method with the technique used by artists, Figure 20 shows the result. From Figure 20, it can be seen that our sketch-drawing method can effectively depict the geometric feature lines of the model and the shaded areas, and has better expression for shading line processing, line brightness change and drawing cable.

We further compare our method with widely-used sketch-drawing methods which include Difference-of-Gaussian [13], Suggestive Contours [15], and lines from diffuse shading [3].These results are presented from Figure 21 (b)-(d). Although these methods achieved pleasing result, most of these approaches are very sensitive to the mesh quality, and usually require appropriate preprocessing operations (e.g. smoothing, remeshing, etc.) on the input meshes. Furthermore, suggestive contours must be used together with silhouettes, since they cannot illustrate salient features in convex regions. However, our method does not require the preprocessing on the input 3D model which is raw point clouds.

VIII. CONCLUSIONS AND FUTURE WORK

We have described a novel sketch-drawing method from the raw point clouds, which combines ridge-valley lines and the contour for better expressing the shape information. We conduct a wide range of experiments on real-world data sets and demonstrate the capability of our method to illustrate point clouds models. Our approach not only generates visually pleasing results based on point clouds, but also enriches the artistic sketch-drawing effect.

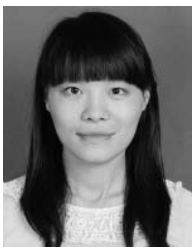
Our current implementation can be easily improved by employing a GPU-based acceleration scheme due to the parallel nature of ridge-valley lines extraction. Considering about how to render sketch-drawing for various kinds of 3D point clouds models and improve the sketch-drawing performance will be our future work.

REFERENCES

[1] S. Rusinkiewicz, F. Cole, D. DeCarlo, and A. Finkelstein, "Line drawings from 3D models," in *Proc. SIGGRAPH*, Los Angeles, CA, USA, Aug. 2008, Art. no. 39.
[2] T. Strothotte and S. Schlechtweg, *Non-Photorealistic Computer Graphics: Modeling, Rendering, and Animation*. San Francisco, CA, USA: Morgan Kaufmann, Jan. 2002.
[3] Y. Lee, L. Markosian, S. Lee, and J. Hughes, "Line drawings via abstracted shading," *ACM Trans. Graph.*, vol. 26, no. 3, Jul. 2007, Art. no. 18.
[4] E. Kalogerakis, D. Nowrouzezahrai, P. Simari, J. McCrae, A. Hertzmann, and K. Singh, "Data-driven curvature for real-time line drawing of dynamic scenes," *ACM Trans. Graph.*, vol. 28, no. 1, Jan. 2009, Art. no. 11.
[5] F. Cole, K. Sanik, D. DeCarlo, A. Finkelstein, S. Rusinkiewicz, and M. Singh, "How well do line drawings depict shape?" *ACM Trans. Graph.*, vol. 28, no. 3, Jul. 2009, Art. no. 28.
[6] F. Cole *et al.*, "Where do people draw lines," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 1–11, Jan. 2008.
[7] Z. Li, S. Qin, X. Jin, Z. Yu, and J. Lin, "Skeleton-enhanced line drawings for 3D models," *Graph. Models*, vol. 76, no. 6, pp. 620–632, Nov. 2014.
[8] T. Judd, F. Durand, and E. Adelson, "Apparent ridges for line drawing," *ACM Trans. Graph.*, vol. 26, no. 3, Jul. 2007, Art. no. 19.
[9] L. Zhang *et al.*, "Efficient and robust 3D line drawings using difference-of-Gaussian," *Graph. Models*, vol. 74, no. 4, pp. 87–98, Jul. 2012.
[10] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "BundleFusion: Real-time globally consistent 3D reconstruction using on-the-fly surface reintegration," *ACM Trans. Graph.*, vol. 36, no. 3, Jun. 2017, Art. no. 24.
[11] H. Xu, M. X. Nguyen, X. Yuan, and B. Chen, "Interactive silhouette rendering for point-based models," in *Proc. 1st Eurograph. Conf. Point-Based Graph. (SPBG)*, Zurich, Switzerland, Jun 2004, pp. 13–18.
[12] M. Olson, R. Dyer, H. Zhang, and A. Sheffer, "Point set silhouettes via local reconstruction," *Comput. Graph.*, vol. 35, no. 3, pp. 500–509, Jun. 2011.

[13] L. Zhang, Y. He, J. Xia, X. Xie, and W. Chen, "Real-time shape illustration using Laplacian lines," *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 7, pp. 993–1006, Jul. 2010.
[14] S. Grabli, E. Turquin, F. Durand, and F. X. Sillion, "Programmable rendering of line drawing from 3D scenes," *ACM Trans. Graph.*, vol. 29, no. 2, Mar. 2010, Art. no. 18.
[15] D. DeCarlo, A. Finkelstein, S. Rusinkiewicz, and A. Santella, "Suggestive contours for conveying shape," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 848–855, Jul. 2003.
[16] D. DeCarlo and S. Rusinkiewicz, "Highlight lines for conveying shape," in *Proc. 5th Int. Symp. Non-Photorealistic Animation Rendering*, San Diego, CA, USA, Aug. 2007, pp. 63–70.
[17] A. Ni, K. Jeong, S. Lee, and L. Markosian, "Multi-scale line drawings from 3D meshes," in *Proc. Symp. Interact. 3D Graph. (SI3D)*, Redwood City, CA, USA, Mar. 2006, pp. 133–137.
[18] X. Xie, Y. He, F. Tian, H. S. Seah, X. Gu, and H. Qin, "An effective illustrative visualization framework based on photic extremum lines (PELs)," *IEEE Trans. Vis. Comput. Graphics*, vol. 13, no. 6, pp. 1328–1335, Nov. 2007.
[19] X.-F. Pang, M.-Y. Pang, and C.-X. Xiao, "An algorithm for extracting and enhancing valley-ridge features from point sets," *Acta Autom. Sinica*, vol. 36, no. 8, pp. 1073–1083, Aug. 2010.
[20] G. G. Maisuradze and D. L. Thompson, "Interpolating moving least-squares methods for fitting potential energy surfaces: Illustrative approaches and applications," *J. Phys. Chem. A*, vol. 107, no. 37, pp. 7118–7124, Jun. 2003.
[21] *Stanford 3D Scanning Repository, AIMSHAPE Shape Repository*. Accessed: Jan. 2011. [Online]. Available: http://www.aimatshape.net

YINGHUI WANG received the Ph.D. degree from Northwest University, Xi'an, China, in 2002. From 2003 to 2005, he was a Post-Doctoral Fellow at Peking University, Beijing, China. He is currently a Professor with the Institute of Computer Science and Engineering, Xi'an University of Technology, China. His research interests include image analysis and pattern recognition.



HUANHUAN ZHANG received the B.S. and M.S. degrees in 2010 and 2013, respectively. She is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Xi'an University of Technology, China. She is currently a Lecturer with Xi'an Polytechnic University. Her current research interests include 3-D image analysis, visualization, and point cloud processing.

XIAOJUAN NING received the Ph.D. degree from the Xi'an University of Technology, Xi'an, China, in 2011. She is currently an Associate Professor with the Xi'an University of Technology. Meanwhile, she has cooperated with the Laboratory in Computer Science, Automation and Applied Mathematics, and the National Laboratory of Pattern Recognition, Institute of Automation, University of Chinese Academy of Sciences. Her current research interests include scene modeling and shape analysis.

WEN HAO received the B.S. and M.S. degrees from the Department of Computer Science, Shannxi Normal University, Xi'an, China, in 2008 and 2011, respectively, the Ph.D. degree from the Xi'an University of Technology, Xi'an, in 2015. She is currently with the Institute of Computer Science and Engineering, Xi'an University of Technology. Her research interests include pattern recognition and point cloud processing.

ZHENGHAO SHI received the M.S. degree in computer application technology from the Xi'an University of Technology, Xi'an, China, in 2000, and the Ph.D. degree in architecture of computer system from the Xi'an Institute of Microelectronics Technology, Xi'an, in 2005. He was a Post-Doctoral Researcher with the Department of Computer Science and Engineering, Nagoya Institute of Technology, Nagoya, Japan, from 2008 to 2009. He is currently a Professor with the Institute of Computer Science and Engineering, Xi'an University of Technology. His research interests include image information processing.

MINGHUA ZHAO received the Ph.D. degree from Sichuan University, Chengdu, China, in 2006. She is currently with the Institute of Computer Science and Engineering, Xi'an University of Technology, Xi'an, China. Her research interests include digital image processing, pattern recognition, computer graphics, and computer vision.

HONGFANG ZHOU received the Ph.D. degree from Xi'an Jiaotong University, Xi'an, China, in 2006. She is currently with the Institute of Computer Science and Engineering, Xi'an University of Technology, Xi'an. Her research interests include data mining, artificial intelligence, and the Internet of Things.

LIANSHENG SUI received the M.S. degree from the Institute of Computer Science and Engineering, Xi'an University of Technology, Xi'an, China, in 2000, and the Ph.D. degree from Xi'an Jiaotong University, Xi'an, in 2003. He is currently an Associate Professor with the Institute of Computer Science and Engineering, Xi'an University of Technology. His research interests include computer graphics and digital image processing.

KE LV received the M.S. and Ph.D. degrees from Northwest University, Xi'an, China, in 1998 and 2003, respectively. He was a Post-Doctoral Researcher with the Key Laboratory of Complex Systems and Intelligence Science, Institute of automation, University of Chinese Academy of Sciences, Beijing, China, in 2003. He is currently with the Chinese Academy of Sciences. His research interests include digital image processing, computer graphics, and intelligent information processing technology.