The extraction of feature lines on 3D models: a survey

Weize Quan, Weiliang Meng, Xiaopeng Zhang
LIAMA-NLPR, Institute of Automation, CAS, China
weiliang.meng@ia.ac.cn, xiaopeng.zhang@ia.ac.cn

Abstract

Feature lines are perceptually important features on 3D models, and have extensive applications such as mesh simplification, re-gridding, shape analysis, non-photorealistic rendering and surface smoothing, etc.. This paper gives a comprehensive survey on this field, which has not been summarized before. According to whether feature lines are associated with the line of sight, feature lines can be classified into two categories: view-dependent curves and view-independent curves. Besides, view-independent curves can be further classified according to the type of 3D models, and view-dependent curves can be classified according to the property of feature lines. A complete classified method of feature lines on 3D models is proposed in this paper, and the challenges for feature extraction are also discussed briefly.

Keywords: feature lines, 3D models, view-dependent curves, view-independent curves

1. Introduction

Feature lines on 3D models are powerful shape descriptors, which can express the main geometry characteristics effectively. Usually an artist can capture essential features of objects and depict them into a stick figure using several strokes, while for most of other people this is a challenging task. Therefore, it is valuable to let the computer automatically extract these curves for us, and the curves also has other kinds of applications, such as surface segmentation[1], non-photorealistic rendering (NPR)[2;3;4], mesh remeshing[5], mesh simplification[6;7], surface reconstruction[8], and mesh denoising[9], etc..

In recent years, a series of viewpoints and methods for the extraction of feature lines are proposed by researchers. According to whether feature lines are associated with the line of sight, we can classify feature liens into two categories: view-independent curves and view-dependent curves.

View-independent curves are not related to the viewpoint, and can be calculated by analyzing geometric properties of surface itself. This category can be further divided into direct curvature estimation, tensor voting theory, Morse theory, surface approximation, graph theory, and other methods based on the type of 3D models and computing methods.

View-dependent curves are related to viewpoint, and the viewpoint must be located firstly before calculating them [10;11]. The subtypes of this category includes silhouettes [12], formulated silhouettes [13], suggestive contours[2], apparent ridges[3], Photic Extremum Lines (PELs) [14], principal and suggestive highlights[4], and Laplacian lines[15].

The final classified method of feature lines on 3D models proposed by us is shown in Figure 1. We will give more details on the following sections.

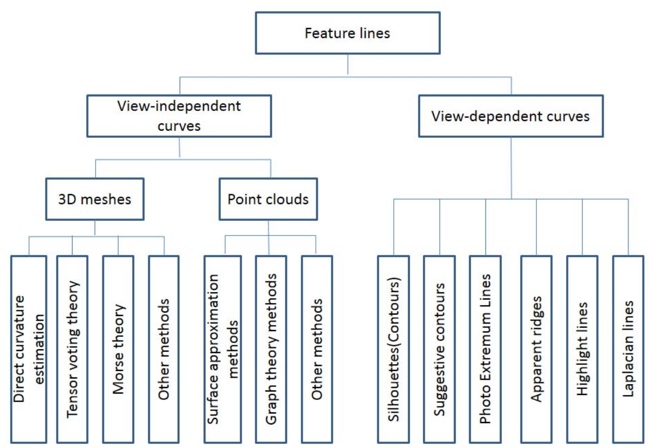


Figure 1. Our classification of feature lines on 3D models.

2. View-independent curves

Usually the computation of view-independent lines is relation to the topology of 3D models. 3D models can be continuous such as parametric surfaces, or discrete such as meshes and point clouds. For the continuous case, view-independent lines can be computed by simply calculating the derivation of the parametric equations. For the discrete case, it is more difficult and many researches are mainly focus on this case for the view-independent

978-1-4799-6854-1/14 $31.00 © 2014 IEEE
DOI 10.1109/ICVRV.2014.11

curves. Additionally, meshes and point clouds are the main 3D discrete models with great difference for the calculation as connection relationships can be used in mesh but not in point clouds, so we divide view-independent curves according to the 3D model types first.

2.1. Curves on 3D meshes

For the last decades, a wide variety of methods have been proposed in the field of the feature lines extraction on 3D meshes, especially on dense triangle meshes. At present, the common feature extraction techniques on 3D meshes can be classified into four categories: direct curvature estimation, tensor voting theory, Morse theory, and other methods.

2.1.1. Direct curvature estimation. Direct curvature estimation methods are used to calculate the feature lines that are defined relative to the curvature, such as crest lines, which are defined via extrema of the principal curvatures along their corresponding curvature lines. However, practical detection of curvature extrema is widely considered as a difficult computational task because this procedure involves estimating of third-order and fourth-order surface derivatives [16;17], and there are some researchers specially study related works [18;19;20;21;22].

In order to conquer the problem, polynomial fitting strategies are developed to achieve an accurate estimation of the principal curvatures and their derivatives on each mesh vertex. Cazals et al.[19] propose polynomial fitting to estimate differential properties without the assumption that the mesh normal is already given. Under the premise that the vertex normal is preliminary estimated, Goldfeather et al.[20] propose a third-order method for estimating the principal directions, which makes full use of the normal vectors, and obtains a better result than the method proposed in [19]. Yoshizawa et al.[7] improve the adjacent-normal cubic-order approximation algorithm in [20], and their method leads to a faster crest lines detection.

Ohtake et al.[16] propose an implicit surface fitting algorithm for detecting ridges and valleys on triangle meshes. In their work, the multi-level implicit surface fitting is combined with finite difference approximation for high-quality estimation of curvature tensor and their derivatives. Implicit surface fitting is a promising method for ridge-valley lines detection, but the computation time cost is high. Kim et al.[23] employ enhanced moving-least-squares approximation to estimate curvatures and corresponding derivatives on each vertex. They reduce the time-complexity compared to Ohtake et al.[16] in approximation, which can be seen from their table of curvatures and their derivatives estimation statistics (Table 1).

Table 1. The comparison of running time of Kim's method and Ohtake's method. (From [23]).

Model	Num. of triganles	Estimating modified MLS approximation	Estimating curvatures and their derivatives	Estimating RBF approximation [16]	Estimating curvatures and their derivatives [16]
Feline	99,732	1.2	9.7	22	101
Dinosaur	112,623	1.5	10.9	23	129
Teeth	233,204	3.1	21.9	52	274
Dragon	15,000	1.8	14.5	38	202
Saddle	8192	0.1	0.4	1	2
Fandisk	51,874	.06	7.3	11	34
Mechanical part	340,480	15.4	85.1	27	306

However, polynomial fitting strategies or implicit surface schemes are all still time-consuming. Hildebrandt et al.[24] use discrete differential operators on 3D meshes, which can avoid those preprocessing steps and is more efficient. The crucial step is the discretization of differential geometric equations.

2.1.2. Tensor voting theory. A symmetric semi-definite matrix at each vertex can be obtained via normal tensor voting method, and then the vertices can be classified via eigen-analysis of this matrix to detect the creases. Methods based on tensor voting theory can detect sharp features with robustness to noisy meshes, which are usually adopted by researchers to analyze a surface shape [25].

Page et al.[26] use normal-vector voting to estimate curvature and extract creases on a triangle mesh. Kim et al.[27] use tensor voting theory to detect features of triangular meshes and handle multiple attributes. Their method utilizes a tensor voting technique [25] for classifying features and mainly includes three steps: eigen-analysis of a normal voting tensor, clustering of vertices, along with region growing and cleaning.

2.1.3. Morse theory. Morse theory is a very powerful tool for studying the geometric and topological proprieties of differential manifolds, which mainly studies the critical points of smooth functions on manifolds. Therefore, the Morse theory can be used to calculate the critical points of the curvature function to detect feature lines.

Várady et al.[28] present segmentation techniques deriving from the Morse theory to construct feature skeletons. Sahner et al.[29] propose a method based on discrete Morse Theory needing no derivative estimation, and curvedness measure [30] is used as the scalar indicators on the vertices of the mesh instead of the principal curvatures. Weinkauf et al.[31] propose a novel method based on a topological analysis of the principal curvature functions without curvature derivatives, and the topology of a discretized function is calculated based on discrete Morse theory [32].

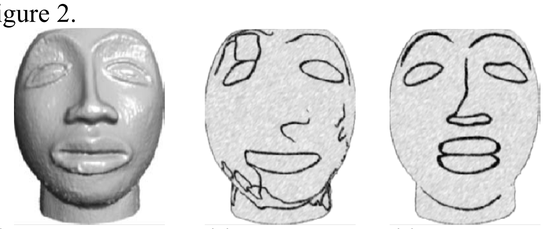
2.1.4. Other methods. Integral invariants are first come up and used for planar curve matching in [33]. Clarenz et al.[34] propose a classification tool based on moment analysis in the local neighborhood, and detect features by integral invariants. Lai et al.[35] accomplish the multi-scale recognition of features through integral invariants of local geodesic neighborhoods and can be used to classify and edit features.

In [36], snakes, i.e., active contour models, are proposed first and used to edge or line detection. Inspired by this, Lee et al.[37] give a geometric snake as an extension of snakes for detecting the salient features on 3D meshes. A geometric snake can automatically move via minimizing an energy function, which is defined by normal variations of the neighbor faces at a mesh vertex. Particularly, a user can interactively edit the snake to obtain the better result.

2.2. Curves on Point clouds

For many feature extraction work on point clouds, the strategies can be divided into three kinds: graph theory methods, surface approximation methods, and other methods. Methods based on graph theory can construct local topological connection, while surface approximation methods such as Robust Moving Least Square (RMLS)[38], local MLS polynomial [39] are mainly used to estimate curvature information. Other methods include multi-scale operator [40], statistical method [41], and so on.

2.2.1. Graph theory methods. Gumhold et al.[42] use a neighbor graph to connect nearby points, and extract features from point clouds on the local neighbor graph only without surface reconstruction. Pauly et al.[43] extends the method in [42] by utilizing a multi-scale classification that enhance the robustness of extraction algorithm. In contrast to [42], there is an apparent different point in [43]: a multi-scale classification operator is used to analyze feature to improve the reliability and robustness in the noisy meshes as shown in Figure 2.



(a) original point clouds (b) single-scale (c) multi-scale

Figure 2. Multi-scale feature detection (c) has the higher robustness than single-scale feature detection on a noisy point cloud data. (From [43]).

Unlike Gumhold et al.[42] and Pauly et al.[43], Demarsin et al.[44] combine normal estimation with graph theory to detect closed sharp lines in point clouds. In this method, a minimum spanning tree is constructed at the level of clusters, which is different from [42; 43] at the level of individual points. Due to clustering reduces the size of point clouds, the algorithm is practical for large point clouds.

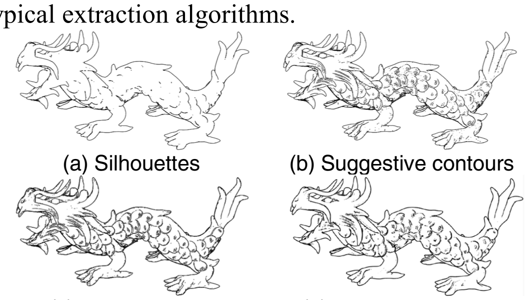
2.2.2. Surface approximation methods. Daniels et al.[45] use RMLS [38] to locally fit multiple surfaces to the neighbors of potential feature points. However, this algorithm is limited since the RMLS is time-consuming. Therefore, Pang et al.[46] use a local MLS polynomial [39] to approximate the neighbors of each point, and the principal curvatures and their directions are computed based on the local surface approximation method.

2.2.3. Other methods. Liu et al.[47] use a multi-scale operator, namely, the Difference of Normal(DoN)[40], to detect feature lines from unorganized noisy point clouds, and the points whose DoN values are larger than a specific threshold are regarded as candidate feature points. After removing the outliers far away from the potential feature lines, the feature lines are obtained by directly connecting the vertices in order.

Weber et al.[41] introduce a statistical method to extract sharp feature lines based on Gauss map clustering. In this method, the points which are impossible to belong to sharp features are abandoned, and then the remaining potential feature points go through Gauss map clustering which is an iterative selection process with an adaptive local sensitivity parameter.

3. View-dependent curves

While view-independent curves just reflect geometric properties of 3D objects, the view-dependent curves describe properties of the surface geometry, lighting, material, and viewing locations [48]. There are many view-dependent curves such as Silhouettes, Suggestive contours, PELs, Apparent ridges, Highlight lines, Laplacian lines as partly shown in Figure 3. Because these lines are often used in form of combination for describing shape [49;50;51], they are difficult to be classified from the perspective of extraction techniques. In this paper, they are classified in the view of definition with some typical extraction algorithms.



(a) Silhouettes (b) Suggestive contours

(c) Apparent ridges (d) Laplacian lines

Figure 3. Some view-dependent curves. (From [15]).

3.1. Silhouettes (Contours)

For a smooth surface, the silhouette (contour) can be defined as the loci of points where the normal vector are orthogonal to the view vector. Saito et al.[52] use image processing techniques for drawing contours (silhouettes). Gooch et al.[53] use a Gauss map for locating all silhouettes under orthographic projection. Hertzmann et al.[54] detect silhouettes based on geometric duality, and can be applicable to both perspective and orthographic projection. Buchanan et al.[55] introduce a data structure, namely, the edge buffer, to render silhouette easily. Lee et al.[50] give a GPU-based technique to produce silhouettes, ridges, and a generalization of suggestive contours.

3.2. Suggestive contours

In many cases, however, contours alone cannot convey salient and important aspects of a 3D object. DeCarlo et al.[2] introduce suggestive contour that exceeds contours and creases. Suggestive contours are curves along with the zero-crossings of the radial curvature and on where the surface bends away from the viewer.

DeCarlo et al.[2] propose two methods to compute suggestive contours on 3D triangle meshes, including object-space algorithm and image-space algorithm. In [56], DeCarlo et al. further analyze the movement of suggestive contours according to viewpoint changes, and extend pervious work[2] on static suggestive contours to dynamic environments.

3.3. Photic Extremum Lines (PELs)

The feature lines such as ridge-valley lines and suggestive contours are only determined by local geometry attributes and the view location. Xie et al.[14] propose Photic Extremum Lines (PELs) that reflects significant variations of luminance. Compared with other existing lines, PELs are more flexible and provide users more freedom to obtain ideal visualization.

In [14], the PEL is defined as a collection of points where the variation of luminance has a local maximum in the direction of its gradient. The extraction procedure consists of smoothing normal, calculating the gradient of illumination, computing the directional derivatives in the direction of gradient, detecting and tracing the zero-crossing to obtain the PEL.

3.4. Apparent ridges

Due to none of the formerly feature lines alone can capture all visually-relevant shape information on 3D objects, Judd et al.[3] present apparent ridges that are defined as the loci of points that maximize a view-dependent curvature. Standard techniques [21] are utilized to estimate view-dependent curvature, the view-dependent curvature derivatives based on the finite differences, and an approach similar to the method in [16] is used to find the zero-crossings of the view-dependent curvature derivatives. After trimming based on the setting threshold, apparent ridges are obtained finally.

3.5. Highlight lines

Existing lines are classically drawn in black on a white backgrounds, DeCarlo et al.[4] define two new types of highlight lines: suggestive highlights and principal highlights, which are drawn in white on dark background. The definitions of highlight lines are based on definitions of suggestive contours and ridge-valley lines, so the detection of suggestive highlights is similar to suggestive contours [2], and the detection of principal highlights is analogous to the method in [16].

3.6. Laplacian lines

Traditional methods generating feature lines on 3D objects need computing high-order derivatives, which are time-consuming and sensitive to noise. Zhang et al.[15] define Laplacian Lines as the zero-crossings points of the Laplacian of the surface illumination. Because the Laplacian of surface normal is view-independent, it can be pre-computed and thus reduce the running time.

4. Conclusion

In this paper, we give an extensive review of the extraction of feature lines on 3D models. The feature extraction of 3D models is one of the most interesting issues in the field of computer graphics. We classify the feature lines into two categories: view-dependent curves and view-independent curves, according to whether they are associated with the light of sight.

The extraction of view-independent curves usually involves estimating of third-order and fourth-order surface derivatives that are sensitive to noise. Despite various schemes such as smoothing or interactive schemes are used to improve the robustness of extraction techniques, this still is a challenge task that needs to be further studied. Furthermore, the normal tensor voting method that needs no higher-order derivatives but only first-order derivatives can extract sharp feature lines and show robustness to noise models, and thus have potential research value.

For point clouds, although there are some methods such as graph theory methods, surface approximation methods, and statistical methods can extract some feature lines, how to effectively obtain more detailed feature lines on unorganized noisy point clouds still needs more attention.

View-dependent feature lines take different properties of mesh into account, and are often used in form of combination. In practice, the approximate selection of type of curves and corresponding algorithms remain a challenge. Future promising methods should introduce interactive tools that can select type of curves and support artificial modification.

Above all, we describe the classify methods of feature lines first, and then the challenges for feature extraction are discussed briefly. We believe that this survey can give valuable insights into this important research topic and may be useful for researchers to resolve relevant research problems.

Acknowledgment

This work is supported by National High-tech R&D Program (2013AA013701).

References

[1] Georgios, S., and Gerald, F., “Crest lines for surface segmentation and flattening”, IEEE Trans. Vis. Comput. Graphics, 10(5), 2004, pp. 536-544.

[2] Decarlo, D., Finkelstein, A., Rusinkiewicz, S., and Santella, A., “Suggestive contours for conveying shape”, ACM Trans. Graph., 22(3), 2003, pp. 848-855.

[3] Tilke, J., Fredo, D., and Edward, H.A., “Apparent ridges for line drawing”, ACM Trans. Graph., 26(3), 2007, pp. 19-26.

[4] Decarlo, D., and Rusinkiewicz, S., “Highlight lines for conveying shape”, In Proceeding of NPAR, 2007. pp. 63-70.

[5] Attene, M., Falcidieno, B., Rossignac, J., and Spagnuolo, M., “Sharpen & Bend: Recovering curved sharp edges in triangle meshes produced by feature insensitive sampling”, IEEE Trans. Vis. Comput. Graphics, 11(2), 2005, pp. 181–192.

[6] Ohtake, Y., and Belyaev, A.G., “Automatic Detection of geodesic ridges and ravines on polygonal surfaces”, The Journal of Three Dimensional Image, 15(1), 2001, pp. 127–132.

[7] Yoshizawa, S., Belyaev, A., and Seidel, H.-P., “Fast and Robust Detection of Crest Lines on Meshes”, In Proc. of ACM Symposium on Solid and Physical Modeling, 2005, pp. 227-232.

[8] Hsu, S.-H., and Lai, J.-Y., “Extraction of geodesic and feature lines on triangular meshes”, International journal of advanced manufacturing technology, 42(9), 2009, pp. 940-954.

[9] Fan, H.Q., Yu, Y.Z., and Peng, Q.S., “Robust Feature-Preserving Mesh Denoising Based on Consistent Subneighborhoods”, IEEE Trans. Vis. Comput. Graphics, 16(2), 2010, pp. 312-324.

[10] Kolomenkin, M., Shimshoni, I., and Tal, A., “Demarcating Curves for Shape Illustration”, ACM Trans. Graph., 27(5), 2008, pp. 157:1-157:9.

[11] Kim, Y.J., Lee, Y.J., Kang, H., and Lee, S., “Stereoscopic 3D Line Drawing”, ACM Trans. Graph., 32(4), 2013.

[12] Hertzmann, A., “Introduction to 3D Non-Photorealistic Rendering: Silhouettes and Outlines”, Non-Photorealistic Rendering, SIGGRAPH Course Notes, 1999.

[13] Whelan, J. C. and Visvalingam, M., “Formulated Silhouettes for Sketching Terrain”, In Proc. TPCG '03, 2003, pp. 90-97.

[14] Xie, X.X., He, Y., Tian, F., Seah, H.-S., Gu, X.F., and Qin, H., “An Effective Illustrative Visualization Framework Based on Photic Extremum Lines (PELs)”, IEEE Trans. Vis. Comput. Graphics, 13(6), 2007, pp. 1328-1335.

[15] Zhang, L., He, Y., Xie, X.X., and Chen, W., “Laplacian Lines for Real-Time Shape Illustration”, In Proc. I3D '09, 2009, pp. 129-136.

[16] Ohtake, Y., Belyaev, A., and Seidel, H.P., “Ridge-valley lines on meshes via implicit surface fitting”, ACM Trans. Graph., 23(3), 2004, pp. 609-612.

[17] Mao, Z.h., Lee, K., Cao, G., and Ma, Y.z., “Interactive Feature Extraction on 3D Meshes”, Computer-Aided Design and Applications, 8(5), 2011, pp.785-793.

[18] Page, D. L., Koschan, A. F., Sun, Y., Paik, J., and Abidi, M. A., “Robust Crease Detection and Curvature Estimation of Piecewise Smooth Surfaces from Triangle Mesh Approximations Using Normal Voting”, CVPR, 1, 2001, pp. 162-167.

[19] Cazals, F., and Pouget, M., “Estimating Differential Quantities using Polynomial fitting of Osculating Jets”, In Proc. Symposium on Geometry Processing, 2003, pp. 177-187.

[20] Goldfeather, J., and Interrante, V., “A Novel Cubic-Order Algorithm for Approximating Principal Direction Vectors”, ACM Trans. Graph., 23(1), 2004, pp. 45-63.

[21] Rusinkiewicz, S., “Estimating Curvatures and Their Derivatives on Triangle Meshes”, In Proc. 3DPVT '04, 2004, pp. 486-493.

[22] Kalogerakis, E., Simari, P., Nowrouzezahrai, D., and Singh, K., “Robust statistical estimation of curvature on discretized surfaces”, In Proc. of the Fifth Eurographics Symposium on Geometry Processing, 2007, pp. 13-22.

[23] Kim, S.-K., and Kim, C.-H., “Finding ridges and valleys in a discrete surface using a modified MLS approximation”, Comput. Aided Des., 37, 2005, pp. 1533-1542.

[24] Hildebrandt, K., Polthier, K., and Wardetzky, M., “Smooth Feature Lines on Surface Meshes”, In Proc. SGP’05, 2005, pp. 85:1-85:6.

[25] Medioni, G., Tang, C.-K., and Lee, M.-S., “Tensor Voting: Theory and Applications”, In Proc. RFIA, 2000.

[26] Page, D. L., Koschan, A. F., Sun, Y., Paik, J., and Abidi, M. A., “Normal Vector Voting: Crease Detection

and Curvature Estimation on Large, Noisy Meshes", Graph. Models, 64(3-4), 2002, pp. 199-229.
[27] Kim, H. S., Choi, H. K., and Lee, K. H., "Feature detection of triangular meshes based on tensor voting theory", Comput. Aided Des., 41(1), 2009, pp. 47-58.
[28] Várady , T., Facello, M. A., and Ter'ek, Z., "Automatic extraction of surface structures in digital shape reconstruction", Comput. Aided Des., 39(5), 2007, pp. 379-388.
[29] Sahner, J., Weber, B., Prohaska, S., and Lamecker, H., "Extraction Of Feature Lines On Surface Meshes Based On Discrete Morse Theory", IEEE-VGTC Symposium on Visualization, 27(3), 2008, pp. 735-742.
[30] Koenderink, J. J., and van Doorn, A. J., "Surface shape and curvature scales", Image Vision Comput., 10(8), 1992, pp. 557-565.
[31] Weinkauf, T., and Günther, D., "Separatrix Persistence: Extraction of Salient Edges on Surfaces Using Topological Methods", Computer Graphics Forum, 28(5), 2009, pp. 1519-1528.
[32] Forman R., "A user's guide to discrete morse theory", In Proc. of the 2001 Internat. Conf. on Formal Power Series and Algebraic Combinatorics, 2001.
[33] Manay, S., Hong, B.-W., Yezzi, A. J., and Soatto, S., "Integral invariant signatures", In Proc. ECCV'04, 2004, pp. 87–99.
[34] Clarenz, U., Rumpf, M., and Telea, A., "Robust feature detection and local classification for surfaces based on moment analysis", Trans. Vis. Comput. Graphics, 10(5), 2004, pp. 516-524.
[35] Lai, Y.-K., Zhou, Q.-Y., Hu, S.-M., Wallner, J., and Pottmann, H., "Robust Feature Classification and Editing", Trans. Vis. Comput. Graphics, 13(1), 2007, pp. 34-45.
[36] Kass, M., Witkin, A., and Terzopoulos, D., "Snakes: Active contour models", International Journal of Computer Vision, 1998, 1(4), pp. 321–331.
[37] Lee, Y.j., and Lee, S., "Geometric Snakes for Triangular Meshes", Computer Graphics Forum, 21(3), 2002, pp. 229-238.
[38] Fleishman, S., Cohen-Or, D., and Silva, C. T., "Robust Moving Least-squares Fitting with Sharp Features", In Proc. ACM SIGGRAPH, 2005, pp. 544-552.
[39] Guennebaud, G., and Gross, M., "Algebraic point set surfaces", ACM Trans. Graph.,26(3), 2007.
[40] Ioannou, Y., Taati, B., Harrap, R., and Greenspan, M., "Difference of Normals as a Multi-Scale Operator in Unorganized Point Clouds", In Proc. 3DIMPVT '12, 2012.
[41] Weber, C., Hahmann, S., and Hagen, H., "Sharp Feature Detection in Point Clouds", Shape Modeling International Conference, 2010, pp. 175-186.
[42] Gumhold, S., Wang, X.L., and Macleod, R., "Feature extraction from point clouds", In Proc. 10th International Meshing Roundtable, 2001, pp. 293-305.
[43] Pauly, M., Keiser, R., and Gross, M., "Multi-scale Feature Extraction on Point-Sampled Surfaces", Computer Graphics Forum, 22(3), 2003, pp. 281-289.
[44] Demarsin, K., Vanderstraeten, D., Volodine, T., and Roose, D., "Detection of closed sharp edges in point clouds using normal estimation and graph theory", Comput. Aided Des., 39(4), 2007, pp. 276-283.
[45] Daniels, J. II, Ha, L. K., Ochotta, T., and Silva, C. T., "Robust Smooth Feature Extraction from Point Clouds", In Proceedings of the IEEE International Conference on Shape Modeling and Applications 2007, pp. 123-136.
[46] Pang, X. F., Pang, M. Y., and Song, Z., "Extracting Feature Curves on Point Sets", I.J. Information Engineering and Electronic Business, 3, 2011, pp. 1-7.
[47] Liu, X. S., and Jin, C., "Feature Line Extraction from Unorganized Noisy Point Clouds", Journal of Computational Information Systems, 10(8), 2014, pp. 3503-3510.
[48] Cole, F., Golovinskiy, A., Limpaecher, A., Barros, H. S., Finkelstein, A., Funkhouser, T., and Rusinkiewicz, S., "Where Do People Draw Lines?", ACM Trans. Graph., 2008, 27(3), pp. 88:1-88:11.
[49] Burns, M., Klawe, J., Rusinkiewicz, S., Finkelstein, A., and DeCarlo, D., "Line Drawings from Volume Data", ACM Trans. Graph., 24(3), 2005, pp. 512-518.
[50] Lee, Y.J., Markosian, L., Lee, S., and Hughes, J. F., "Line drawings via abstracted shading", ACM Trans. Graph., 26(3), 2007, pp. 18:1-18:5.
[51] Kalogerakis, E., Nowrouzezahrai, D., Simari, P., Mccrae, J., Hertzmann, A., and Singh, K., "Data-Driven Curvature for Real-Time Line Drawing of Dynamic Scenes", ACM Trans. Graph., 28(1), 2009,pp. 11:1-11:13.
[52] Saito, T., and Takahashi, T., "Comprehensible Rendering of 3-D Shapes", In Proc. SIGGRAPH'90, 1990.
[53] Gooch, B., Sloan, P.-P, J., Gooch, A., Shirley, P., and Riesenfeld, R., "Interactive Technical Illustration", In Proc. I3D '99, 1999, pp. 31-38.
[54] Hertzmann, A., and Zorin, D., "Illustrating smooth surfaces", In Proc. SIGGRAPH, 2000, pp. 517-526.
[55] Buchanan, J. W., and Sousa, M. C., "The edge buffer: A data structure for easy silhouette rendering", In Proc. NPAR'00, 2000, pp. 39-42.
[56] DeCarlo, D., Finkelstein, A., and Rusinkiewicz, S., "Interactive Rendering of Suggestive Contours with Temporal Coherence", In Proc. NPAR'04, 2004, pp.15-145.