www.cadanda.com
Computer-Aided Design AND APPLICATIONS

Shape Aware Wireframe Generation for Meshes

Li Cao[1], Yike Xu[2], Yao Wu[3] and Siu-kong Koo[4]

[1]Hefei University of Technology, lcao@hfut.edu.cn
[2]Hefei University of Technology, yikexu@mail.hfut.edu.cn
[3]Hefei University of Technology, wuyao@mail.hfut.edu.cn
[4]Hong Kong Quantum AI Lab Limited, Koo@yangtze.hku.hk

Corresponding author: Li Cao, lcao@hfut.edu.cn

Abstract. Extracting wireframes from 3D meshes is a challenging problem. Existing methods are typically based on analyzing local shape properties, such as surface curvatures and angles between faces, which are generally sensitive to small features in the model and the complexity of the object. In this work, we propose a wireframe extraction method based on a geometric approximation of 3D meshes. Our idea is to utilize a well-established variational geometric partitioning method to derive a complete set of descriptive feature curves. The advantage of this strategy is that a reliable feature filtering mechanism can be inherently incorporated into the geometric approximation step, giving rise to descriptive wireframes that capture global structures of the 3D meshes. We experimented on various meshes. The results demonstrated that our method is superior to existing approaches in terms of the correctness and completeness of the extracted feature curves.

Keywords: mesh processing, variational shape approximation, wireframe, shape aware feature extraction
DOI: https://doi.org/10.14733/cadaps.2023.584-599

1 INTRODUCTION

Wireframes capture the essence of 3D meshes and are widely used in geometry processing (e.g., shape design [21] and sketch modeling [12]). The shapes of 3D meshes can be efficiently and intuitively represented by wireframes. It is unavoidable to have defects in 3D meshes being digitized in the real world, such as noise, missing surface details, over-smoothed in certain regions, etc. These defects hinder the applicability of the reconstructed models. An alternative solution for obtaining high-quality surface models is to extract the feature curves from the scanned data and use these feature curves to reconstruct better surfaces [6]. However, due to noise, missing data, outliers, and lack of topological information in point clouds, extracting reliable feature curves is tricky [11]. As a result, user input would become necessary for high-quality reconstruction.

Existing approaches for wireframe extraction from polygonal mesh rely on local shape properties, such as surface curvatures [14] and angles between faces/vertices [1]. These shape properties are too local to capture the global structure of 3D meshes, yielding unsatisfied feature extraction results. Methods based on length filtering [18] can generate sufficiently long feature curves. However, these methods neglect important surface information, such as swept surface. To achieve feature curves favoring both completeness and descriptiveness, more global information regarding the input models' structure must be considered.

In this work, we propose a method combining geometric approximation and feature line filtering to generate wireframes from 3D meshes. Our method is based on a well-established variational geometric partitioning strategy. Specifically, we exploit the Variational Shape Approximation (VSA) [4] algorithm to obtain the initial feature curves, followed by feature detection and filtering steps. Compared with existing approaches, our method can generate a complete set of feature lines which can be further refined and converted into a clean wireframe representation of the input model. Moreover, our filtering operation provides control over both sizes, i.e., surface areas, of the segmented surface patches and organization of individual feature lines, as shown in Fig. 1. The significant contributions of this work are as follows:

a) We propose a method incorporating geometric approximation into the feature curve extraction pipeline, which significantly improves the quality of the results.

b) We propose the idea of extending the 3D mesh feature curve to generate a complete wireframe structure.

c) We demonstrate a practical pipeline for evaluating the quality of a clean wireframe with different reconstruction methods.

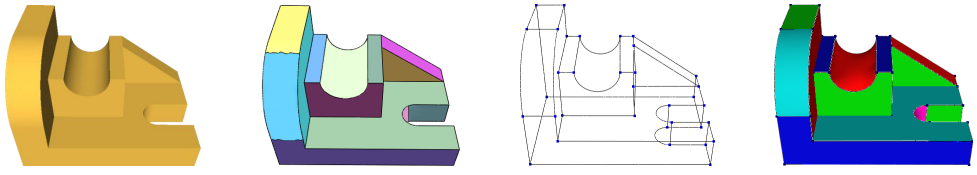


Figure 1: Results during operations. (a) Original mesh; (b) Patches from the VSA separation; (c) Wireframe from our method; (d) Reconstruction result

2 PRIOR WORKS

Wireframe is an important factor in lots of graphic applications, such as, remeshing [17], simplification [16], and reconstruction [26]. In most of this work, a well-extracted wireframe showed the high reliability of the original model. Due to the length limit of this paper, instead of showing a full scope of all previous work, a few representatives which are comparable our target will be described here.

2.1 Feature Detection for Meshes

Meshes have a wide range of applications and research value [23, 19]. The detection of feature curves from mesh has been widely studied in many works [24, 18]. In most of these studies, the geometry criteria used to extract feature lines are curvature and the derivatives of curvature. However, it is difficult and inefficient to calculate the correct result in many general situations, such as sharp edges. Torrente et al. [20] started the analysis of small features on meshes using an algebraic approach. Heat diffusion was introduced to deal with

this problem to provide better segmentation results [2]. Due to the limitation of local surface properties, these methods usually cannot detect features on smooth and transition regions.

2.2 Feature Curve Extraction

Many algorithms generate features by extracting patch boundaries from segmentation results. Based on the features detected, all feature curves can be generated for various applications, such as simplification [7]. Zhuang et al. [25] proposed a correlation clustering method to obtain feature lines. Besides curvature, dihedral angle [18], and discrete differential geometry [3] were also introduced to avoid costly computation. Most of these studies were based on the analysis of local surfaces, and a few of them introduced global surface analysis [10].

2.3 Wireframe for Shape Approximation

Patch-based segmentation method can give us another solution to generate a wireframe of cloud points or meshes. Variational shape approximation (VSA) [4] is one of the well-known patch segmentation methods. Furthermore, quadric patches [22] was introduced to extend the method. Fernando et al. [5] proposed a new type of surface feature skeleton for surface abstraction. Lu et al. [15] proposed an automatic method based on the quadric surface fitting method to extract complete feature curve networks. Liu et al. [13] introduced a first end-to-end trainable deep network architecture to this field. Guo et al. [9] recovered the boundary representation by solving a global optimization maximizing the likelihood under structural validness constraints and applying geometric refinements.

2.4 Wireframe Quality Evaluation

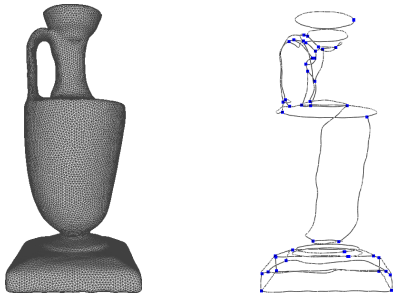
The evaluation of wireframe quality is an important research topic in computer-aided design. This task guarantees the wide application of wireframe. Lu et al. [15] used the hybrid distance between the fitting surfaces and the input mesh. Liu et al. [13] proposed to combine vertex position, average precision of wireframe structure and wireframe editing distance to evaluate the wireframes.These methods are all aimed at evaluating the accuracy of wireframe, without fully considering the performance of wireframe in practical applications[16, 26]. We apply wireframe to the field of model reconstruction to generate the reconstructed 3D model. The Hausdorff distance is introduced for quantitative analysis to compare the similarity between the reconstructed mesh and the original mesh, so as to evaluate the quality of the wireframe.

3 WIREFRAME EXTRACTION FOR MESHES

3.1 Motivation

Wireframes consist of streamlines for major curvature lines on a surface, and trim lines that define surface boundaries or sharp features [16]. Previous methods for generating wireframes focus on how to identify feature curves and connect them. However, the identification of feature curves may neglect some smoothly changed structures, such as sweep surfaces. As depicted in Fig. 2b, although the feature segments are well connected by the method in previous work [14], it is difficult to reconstruct the mesh with this kind of wireframe.

VSA can simplify complex geometry into a few basic shapes that retain essential input features. It is highly dependent on the number of initial seeds and their position at the beginning of the procedure. Therefore, this will lead to some boundary region processing errors. In this work, we would solve this kind of problem by introducing structure information when generating wireframes.



(a) (b)

Figure 2: Wireframe from previous work. (a) Mesh of Vase; (b) Wireframe from FCNE [14]

3.2 Method Description

This work proposes a new processing method to generate better wireframes. As shown in Fig. 3, the processing pipeline of shape aware wireframe generation (SWFG) for meshes is as follows:

a) **Separate patches.**Based on the VSA method, the original mesh can be divided into several patches. The number of patches varies with the complexity of the input mesh. We propose a search scheme to determine the number of patches. For each patch, its boundary will be recorded for further connection.

b) **Feature extraction.**After the above step, several separated patches are collected. The inner area will be processed to generate internal feature lines for one mesh patch. With the help of the filtering step, many segments of feature lines can be located. Small features in each patch will generate lots of short feature curves which should be kept for further analysis.

c) **Filter and extension.**Small feature lines are filtered for further processing. For all line segments generated in each patch, those useless ones should be filtered out to generate a clean wireframe. To achieve this goal, a threshold value is set to remove these short and separated line segments. The left feature curves need to be connected to represent the patch. The endpoints of line segments are extended to generate candidate points, which are vertices of the mesh. For these candidate points, a weight function is designed to find the suitable ones. The selected point will be added to the line segments gradually to form the resulting wireframe of each patch.

d) **Smooth and merge.**It is crucial connecting the wireframe of each patch to generate a complete one. For the saw-tooth style edge of the patches after VSA segmentation, a smoothing step is introduced to smooth the contour of these patches. The boundary curves of VSA patches and the feature curves from each patch are connected and placed to generate a complete wireframe.

After going through the above steps, a well-connected wireframe of the mesh model is generated, which can be used to reconstruct a new smooth mesh for further application.

3.3 Implementation

3.3.1 Generation of VSA patches

The VSA [4] method is introduced to separate the mesh into several patches without the calculation of the local differential coefficient. Based on the original VSA method, more additional steps are introduced to determine the proper number of patches of each mesh.

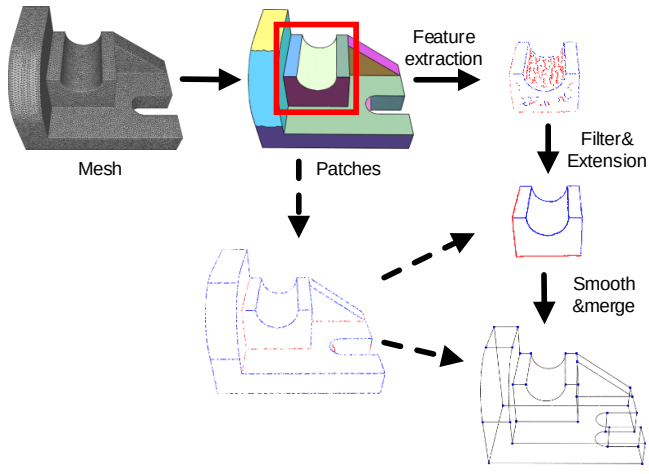


Figure 3: Processing pipeline of shape aware wireframe generation for meshes

a) **Merging patches.**After the VSA operation, patches can be collected. Sometimes, due to the random initial start positions of the seeds, some flat patches may divide into multiple patches, which need to be merged into one patch. The average normal vectors of neighbor patches will be calculated to decide if the merge is needed. This step can be viewed as an initial step of our extended operations.

b) **Additional separation.**For each resulting VSA patch, we calculate the square error of the normal vector q for all triangles in this patch. A threshold value α of the additional separation is needed for further operation. If α in the range of [0.01,0.05], this patch should be divided into more patches, until the threshold is reached. Otherwise, contour from a few VSA patches would not properly generate the reconstruction result. In the example of Fig. 4, the value of α is 0.03, and the resulting segmentation is more reasonable. In extreme cases, the reconstruction process would diverge and no meaningful results can be obtained. In Eq. 1, q_i is the normal vector of the i-th triangle.

$$\alpha=\frac{1}{n}\sum_{i=1}^{n}(q_i-\bar{q})^2 \tag{1}$$

$$\bar{q}=\frac{1}{n}\sum_{i=1}^{n}q_i \tag{2}$$

c) **Boundary area processing.**Check the triangles at boundary areas for each patch. Some triangles are wrongly classified due to the averaging of the normal vertices. These triangles will cause the expected smooth curves to become rough, especially for those sharp edges. The $\mathbf{L^{2,1}}$ distance between these triangles and nearby patches needs to be calculated to find the correct patch, as shown in Fig. 4.The calculation of $\mathbf{L^{2,1}}$ is based on the equation as follows,

$$\mathbf{L^{2,1}}(\mathbf{R_i},\mathbf{P_i})=\iint_{\mathbf{x}\in\mathbf{R_i}}||\mathbf{n}(\mathbf{x})-\mathbf{n_i}||^{\mathbf{2}}\mathbf{dx} \tag{3}$$

In Eq. 3, R_i is the area needs to be expanded, P_i is the i-th patch, x is the vertex in R_i, $\mathbf{n}(\mathbf{x})$ is the normal vector of the vertex x , $\mathbf{n_i}$ is the normal vector of the i-th patch.

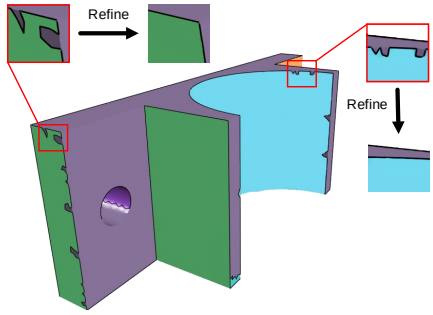


Figure 4: Processing wrongly classified triangles of the boundary area

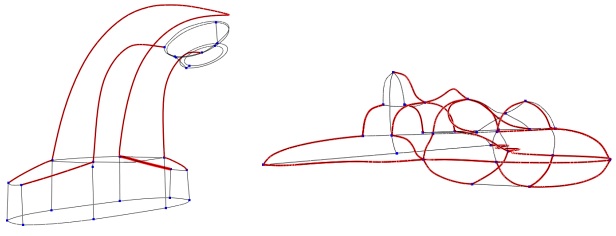
With the different number of seed points, some wireframe results can be collected for evaluating the performance. To better compare the similarity of the original mesh and the reconstructed one, Hausdorff distance is introduced for quantitative analysis.

$$\mathbf{H}(\mathbf{A},\mathbf{B}) = \mathbf{max}(\mathbf{h}(\mathbf{A},\mathbf{B}),\mathbf{h}(\mathbf{B},\mathbf{A})) \tag{4}$$

$$\mathbf{h}(\mathbf{A},\mathbf{B}) = \mathbf{max}(\mathbf{a}\in\mathbf{A})\mathbf{min}(\mathbf{b}\in\mathbf{B})||\mathbf{a}-\mathbf{b}|| \tag{5}$$

$$\mathbf{h}(\mathbf{B},\mathbf{A}) = \mathbf{max}(\mathbf{b}\in\mathbf{B})\mathbf{min}(\mathbf{a}\in\mathbf{A})||\mathbf{b}-\mathbf{a}|| \tag{6}$$

In Eq. 4, $A = \{a_1, a_2, ..., a_n\}$ and $B = \{b_1, b_2, ..., b_n\}$ are the vertices of the mesh to be compared. For reconstruction applications, a smaller Hausdorff distance is preferred [26]. Different methods are applied for the reconstruction based on the geometric features of meshes. The choice is based on a ratio between the number of sharp feature segments [16] and the number of all curve segments in the wireframe.



(a) (b)

Figure 5: Sharp feature segments (in red) of different wireframes. Ratio of sharp feature segments to all segments (a) Tap: 7:32, (b) Fighter: 26:55

If the ratio is less than 0.3 (Fig. 5a), the method from [16] will be used, otherwise (Fig. 5b), the method from [26] will be used. Take the Raised mesh as an example. The Hausdorff distance of the different number of patches is shown in Fig. 6. The x-axis indicates the number of different sites, while the y-axis shows the Hausdorff distance. The Raised model can be viewed as three major parts, so the number of patches is increased by 3 for each iteration. From Fig. 6, it can be concluded that n = 51 is optimal since the difference between n = 51 and n = 54 tends to be stable. The reconstruction meshes with different n are depicted in Fig. 7.

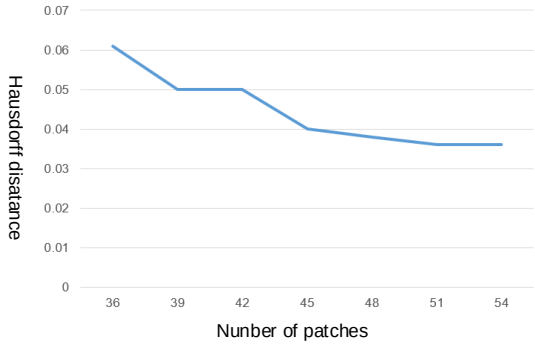


Figure 6: Hausdorff distance of reconstruction results by different wireframe

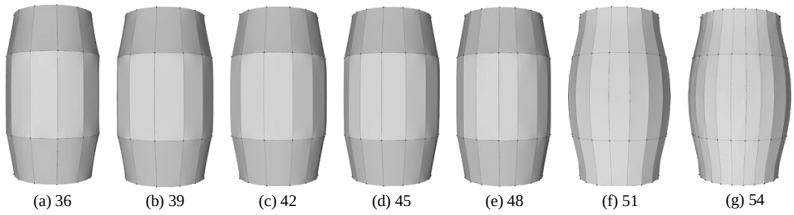


Figure 7: Reconstruction meshes with different n

3.3.2 Feature extraction

For general meshes, both boundary information and small features will lead to changes in the values of their differential coefficients. These are also the criteria to identify small features in existing solutions. Most feature-aware methods employ differential coefficients to filter the face normal, which causes the removal of small features. The key step of feature preserving methods is how to identify small features for different patches. The result from the above steps cannot generate feature lines for the inner area. Feature curves of inner area need to be extracted for each patch. In our method, a modified feature curve extraction method is employed for this purpose, as depicted in Fig. 8a. For each V of the patch M , a third-order polynomial equation is used to fit the small curved surface of V . V is employed as the original point of the local frame, the tangent plane of V is the XY plane and the normal direction of V is Z .

$$h(x,y) = 1/2(b_0x^2 + 2b_1xy + b_2y^2) + 1/6(c_0x^3 + 3c_1x^2y + 3c_2xy^2 + c_3y^3) \tag{7}$$

With the computation result of the vertex V , e_{max} and e_{min} , which are the derivatives of principal direction corresponding to principal curvature k_{max} and k_{max} , feature points are collected.

$$e_{max} = \partial k_{max}/\partial t_{max} \tag{8}$$
$$e_{min} = \partial k_{min}/\partial t_{min} \tag{9}$$

The extracted feature curves are represented as a set of ordered vertices, $C = \{v_0, v_1, v_2, ..., v_n\}$.

3.3.3 Filtering and extension

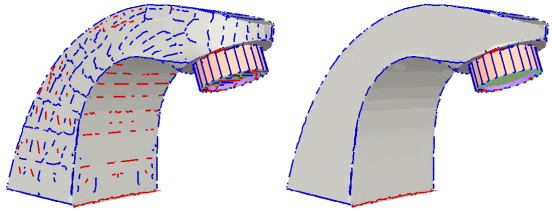
The distribution of small features for each patch is sparse and locally connected. Based on our observation, small features of triangle mesh cover a small area and represent the structure. In the presentation of line frames, they appear as some short connecting lines, which separate the patch into several components. As depicted in Fig. 8a, lots of short feature lines can be obtained from the above steps.

Inspired by the FCNE [14] method, we use the filtering algorithm proposed in [24]. A parameter is used to measure the intensity of feature lines. The threshold β is set according to the complexity of the model.

$$\beta = \frac{1}{n}\sum_{i=1}^{n} C_r(v_i) \tag{10}$$

$$C_r(v_i) = \sum_{i=1}^{n} \sqrt{{k_{\max}}^2 + {k_{\min}}^2} \tag{11}$$

Where $C_r(v_i)$ is the root mean square curvature of v_i. We use β to filter out small features. After filtering, only those feature curves longer than β are reserved for further analysis, as shown in Fig. 8b.



(a) (b)

Figure 8: Small feature lines are filtered by using $\beta = 0.004$. (a) Feature lines of Tap; (b) After filtering

The boundary and feature lines inside the patch will be used to continue the extension work. Neighbor patches sharing the same boundaries have different starting and endpoints. This will cause some line segments to be redundant. Since there are many redundant boundaries from other patches, removing these undesired boundaries and generating a concise wireframe is necessary.

The extension work starts from the endpoint of the feature curve. This step has two purposes. First, in the feature detection phase, some weak features are not detected, or are filtered out in the filtering phase. These features generally exist near the endpoint of the feature curve. The extension is to connect these features again. Secondly, wireframe is a set of closed curves, so only by extending the end point to close the curve can a complete wireframe be formed. When the extension passes through the weak feature area, it extends to the direction where the feature points are located, and the position of the feature points plays a dominant role in the extension process. When the extension passes through the none characteristic area, the original direction of the characteristic curve is the dominant factor guiding the extension direction. Extension is a search process. Heuristic search is adopted to avoid too much time cost caused by extension.

3.3.4 Smooth and merge

In the above steps, resulting patches from the VSA method have aliasing boundaries, which are caused by walking along the edge of triangles. This kind of situation would happen more frequently for meshes with small features. An edge smoothing operation is introduced to deal with the rough curves, as depicted in Fig. 9. The major steps of the smoothing operation are shown in Alg. 1.

Algorithm 1: Feature lines smoothing operation

```
Require: P: Patches;
  n = P.size(); Done = false; idx=0;
  while (not Done) do
    Sort(P,P.NumofFaces);
    Done=true;
    for (i=1 to n) do
      x = FindSharedTriangles(idx,i);
      if (x not less than 2) then
        Smooth(idx,i);
        Done=false;
      end if
    end for
  end while
  return
```

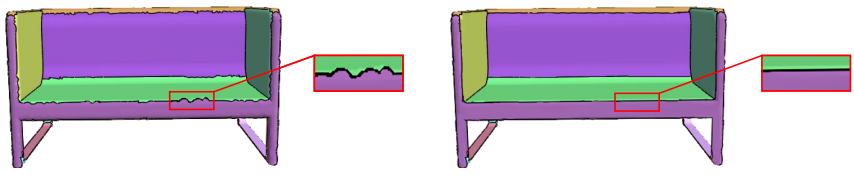


Figure 9: The smoothing operation for patches

Some feature lines with open-end vertices must be processed and attached to a nearby frame. A new solution is proposed to deal with this situation. The endpoints of line segments are extended to generate candidate points, which are vertices of the mesh. For these candidate points, a weight function is designed to find the suitable ones, as shown in Eq. 12. The selected points will be added to the line segments gradually to form the final wireframe.

$$F_n = \lambda L_{cur} + (1-\lambda) L_{ang} \tag{12}$$

$$L_{cur} = \left| \frac{k_{max} - |N_{V_i} \rightarrow k_{max}|}{k_{max} - k_{min}} \right| \tag{13}$$

$$L_{ang} = cos\angle t_{s,V_i} t^{+}_{avg} \tag{14}$$

In Eq. (12), L_{cur} is the curvature term of the candidate V_i, used to find the vertex of maximum curvature in the candidates. k_{max} and k_{min} is the one-round extension of the candidate vertex. $|N_{V_i} \rightarrow k_{max}|$ is the maximum curvature of the candidate vertex V_i, its range is [0,1]. L_{ang} is the term of angle, which is the angle between the line of candidate vertex and end points and the line of feature line, its range is [-1,1]. The λ is used to control the weight of the two terms for different meshes.

3.4 Discussion

There are some interesting details in the pipeline. Based on SWFG, two major factors are helping the generation of a promising wireframe for reconstruction.

3.4.1 VSA provides useful structure information

With the help of the VSA method, global information on models can be obtained. Global information is not preserved in the methods that rely only on feature lines. Take the Telescope as an example, as depicted in Fig. 10b. Few feature lines can be found for the sweep surfaces. SWFG method can give a better result with the global structure information collected from patches, as depicted in Fig. 10c.

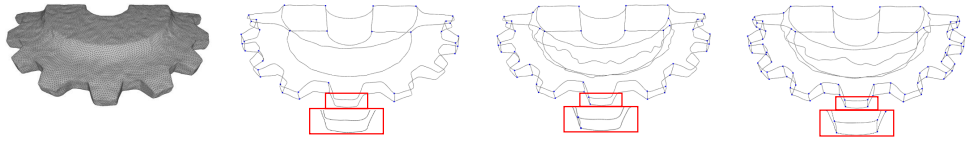
(a) (b) (c)

Figure 10: The proposed SWFG can generate meaningful wireframe for models with sweep surface

For the mesh of Vase from FCNE [14], all separated feature lines can be connected to produce a better result than DFN [3]. However, merging all separated parts may not generate the desired result, as depicted in Fig. 2b. With the global structure information collected from patches, SWFG method can give a better result, as depicted in Fig. 13c.

3.4.2 Preserving small feature curves

With the segmentation results from the VSA method, more feature lines can be collected for the extension. The boundary of small features removed with a large is shown in Fig. 11b and Fig. 11c. These short feature lines between the top and bottom surfaces are discarded due to their short lengths and unobvious curvature. With the help of VSA patches, SWFG method can preserve more short feature lines, as shown in Fig. 11d.



(a) (b) (c) (d)

Figure 11: Comparison with previous methods on model of Gear. (a) Mesh of Gear; Result of (b) FASCC [25], (c) FCNE [14] and (d) SWFG

4 PERFORMANCE

This section provides experimental results to validate the proposed method, which is run on a desktop computer with a 2.50 GHz CPU, 16 GB of memory and RTX 3060 GPU. We tested SWFG method on various 3D meshes from the dataset [8]. We compared our algorithm with previous methods, including DFN [3], FCNE [14], FASCC [25] and PC2WF [13]. The results from SWFG method are evaluated by reconstructing the mesh from the feature curves. The Hausdorff distance and the root mean square (RMS) distance between the reconstructed surfaces and the original mesh were computed for comparison. We also give a comparison of time efficiency.

4.1 Meshes

For the 3D meshes tested, the number of patches is given in Table 1. For these models in Table 1, the wireframe of Gear, Holder, Raised, Vase, and Telescope have been given before. The results for other meshes are given in Fig. 14. In each sub-figure of Fig. 14, the original mesh, VSA patches, wireframe generated and the reconstructed mesh are given, respectively. From the results, it can be concluded that the wireframe generated by SWFG method is a good representation of the original mesh and work well for surface reconstruction.

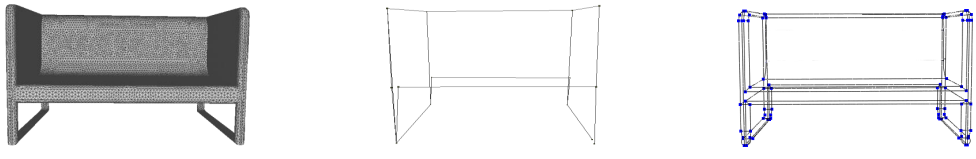
Table 1: The number of patches for meshes

No.	Mesh	Patches Num	Figure	No.	Mesh	Patches num	Figure
1	Buckle	20	Fig. 14f	7	Raised	51	Fig. 7
2	Fighter	29	Fig. 14a	8	Roller	74	Fig. 14c
3	Gear	43	Fig. 11	9	Sofa	54	Fig. 14e
4	Holder	20	Fig. 1	10	Tap	22	Fig. 14d
5	Latch	86	Fig. 14g	11	Telescope	90	Fig. 10
6	Mixer	34	Fig. 14b	12	Vase	35	Fig. 2

4.2 Comparison

To analyze the performance of SWFG method, we compared SWFG method with four methods: detection feature networks (DFN) [3], feature-aligned segmentation using correlation clustering (FASCC) [25], feature curve network extraction (FCNE) [14], and point cloud to wireframe (PC2WF) [13]. We perform experimental evaluations on 3D meshes of varying complexity.

Liu et al. [13] proposed n end-to-end trainable deep network architecture, to extract the wireframe model of the 3D point cloud. Since PC2WF [13] outputs straight lines only, we did not run it on the whole mesh shapes; instead, we provide qualitative results for the PC2WF [13] method only on the small subset of shapes presented in Fig. 12.



(a) (b) (c)

Figure 12: (a) Mesh of Sofa; (b) Result of PC2WF [13];(c) Result of SWFG

All generated wireframes are employed to do the reconstruction for comparison. Reconstruction experiments can demonstrate the stability and utility of wireframes. Hausdorff distance is calculated to evaluate their performance, as shown in Table 2. As mentioned before, a wireframe with a smaller reconstruction value is

Table 2: Comparison of the Hausdorff and RMS distance for different methods

No.	Mesh	DFN		FASCC		FCNE		SWFG	
		Hausdorff	RMS	Hausdorff	RMS	Hausdorff	RMS	Hausdorff	RMS
1	Buckle	N.A.	N.A.	0.0171	0.0289	0.0168	0.0230	**0.0160**	**0.0193**
2	Fighter	0.0659	0.0947	0.0530	0.0882	**0.0418**	**0.0568**	0.0436	0.0635
3	Gear	0.0260	0.0240	**0.0147**	**0.0138**	0.0254	0.0218	0.0161	0.0191
4	Holder	N.A.	N.A.	0.0475	0.0584	**0.0432**	**0.0510**	0.0454	0.0536
5	Latch	N.A.	N.A.	0.0267	0.0538	0.0232	0.0425	**0.0197**	**0.0209**
6	Mixer	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	**0.0772**	**0.1572**
7	Raised	0.0235	0.0289	0.0194	0.0215	0.0218	0.0234	**0.0154**	**0.0165**
8	Roller	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	**0.0601**	**0.0732**
9	Sofa	N.A.	N.A.	0.0788	0.1031	N.A.	N.A.	**0.0485**	**0.0253**
10	Tap	0.1221	0.1894	0.0918	0.1062	0.0769	0.0971	**0.0553**	**0.0885**
11	Telescope	N.A.	N.A.	0.0238	0.0439	N.A.	N.A.	**0.0207**	**0.0392**
12	Vase	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	**0.0311**	**0.0883**

better. From the data given in Table 2, it is clear that SWFG method can generate better wireframes than existing methods.

For the meshes in Table 2, compared with the FCNE [14] method, the SWFG method can reduce by more than 15% of the reconstruction error in general. For the Latch and Tap meshes, it can even reduce more than 20%. This method can also process some meshes which cannot be processed by DFN [3], FASCC [25], and FCNE [14] methods. In Table 2, the tag N.A. means that the wireframe from the corresponding method cannot satisfy the requirements for the reconstruction, as depicted in Fig. 13.

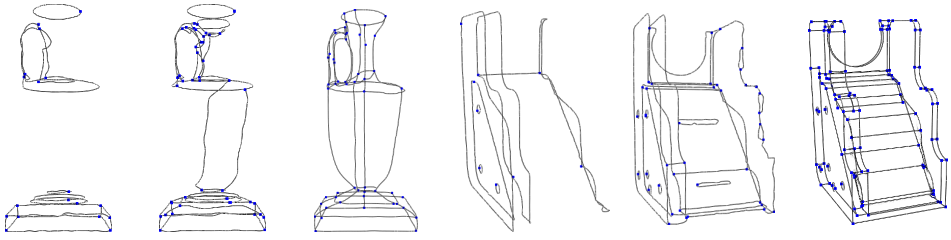


Figure 13: Wireframes result from different methods on models of Vase and Roller; (a), (d) Result of DFN [3]; (b), (e) Result of FCNE [14]; (c), (f) Result of SWFG.

RMS value is also given in Table 2. The reconstruction results from the wireframe generated by SWFG method have smaller RMS. Such a result shows that the overall quality of the wireframe is also better than existing methods.

Running time is shown in Table 3. VSA calculation is a little time-consuming, so our algorithm does not

perform particularly well in running time. But our running time still has better performance. DFN [3] consumes little time typically only uses curve length and curvature to remove more curves without the need for circle information in the network. FCNE [14], compared to DFN [3], combines curvature and surface fitting error to produce a satisfactory curve network. FASCC [25] spends more time highlighting the feature lines while avoiding weak or non-feature lines. In Table 3, we also show the number of triangular meshes of each model and the CPU and memory occupancy of SWFG. Because the number of triangular mesh patches will directly affect the CPU and memory usage of SWFG in the calculation process, thus affecting the running time.

Table 3: Comparison of running time (unit: s)

No.	Mesh	DFN	FASCC	FCNE	SWFG	Triangles	CPU	Memory
1	Buckle	N.A.	7.2591	**2.4821**	3.8341	48476	13.7%	179.8MB
2	Fighter	**0.1753**	3.4228	2.5325	2.2359	29770	11.3%	130.7MB
3	Gear	**0.3384**	4.1502	0.9739	1.3547	29596	11.2%	129.3MB
4	Holder	N.A.	6.9731	1.8738	**1.4945**	31910	11.6%	134.6MB
5	Latch	N.A.	8.4378	**2.9202**	4.4384	59466	14.9%	257.9MB
6	Mixer	N.A.	N.A.	N.A.	**2.5811**	34080	12.4%	152.1MB
7	Raised	**0.0944**	1.3586	0.3251	0.5023	35454	12.6%	159.3MB
8	Roller	N.A.	N.A.	N.A.	**5.6732**	60772	15.2%	265.7MB
9	Sofa	N.A.	9.3459	N.A.	**3.2394**	47480	13.4%	173.7MB
10	Tap	**0.3630**	6.3921	4.7220	4.2529	50240	13.9%	205.3MB
11	Telescope	N.A.	1.3529	N.A.	**0.5435**	25414	10.5%	120.4MB
12	Vase	N.A.	N.A.	N.A.	**0.6523**	26160	10.8%	122.4MB

4.3 Limitation

We have successfully extracted complete wireframes from various 3D meshes. Certain types of meshes, such as those with many asymmetric features and hole information, may not be processed well with SWFG method. Multi-scale problems for large and complex geometric models, SWFG method may lead to some loss of details. It would be difficult to choose the number of patches in those cases, as it would be hard to balance the computation cost and result accuracy. The adaptability of SWFG method to meshes from low sampling resolution is uncertain. Because the mesh with low resolution cannot produce good valley and ridge lines. SWFG needs more optimization in the running time. Further studies will be carried out to improve SWFG method so that fewer patches are needed while maintaining a good result.

5 CONCLUSION

In this work, we propose a wireframe extraction method based on a geometric approximation of 3D meshes. We utilized a well-established variational geometric partitioning step to derive a set of patches for surrounding structure information and analyze each patch to collect relatively small feature curves.

The advantage of this strategy is that a reliable feature filtering mechanism can be inherently incorporated into the geometric approximation step, giving rise to descriptive wireframes that capture the global structures

of the models. Experiments on various kinds of meshes have been carried out and the results demonstrated that SWFG method is superior to existing approaches in terms of correctness and completeness of the extracted feature curves. We can generate better wireframes with minor reconstruction errors for meshes with sweep surfaces or complex structures.

A concise wireframe can be used to finish various applications, such as remeshing, simplification, rendering, etc. SWFG method can provide a high-quality wireframe for those applications to better reconstruct the original meshes.

ACKNOWLEDGEMENTS

This work is supported by the grant of the National Science Foundation of China, No.61602146, and the grant of the Natural Science Foundation of Anhui Province, No.202104e11020006.

ORCID

Li Cao http://orcid.org/0000-0003-4025-8690
Yike Xu http://orcid.org/0000-0003-4583-4589
Yao Wu http://orcid.org/0000-0003-3464-6291
Siu-kong Koo http://orcid.org/0000-0001-8162-5181

REFERENCES

[1] Arhid, K.; Zakani, F.; Bouksim, M.; Aboulfatah, M.; Gadi, T.: An efficient hierarchical 3d mesh segmentation using negative curvature and dihedral angle. International Journal of Intelligent Engineering and Systems, 10(5), 143–152, 2017. http://doi.org/10.22266/ijies2017.1031.16.

[2] Benjamin, W.; Polk, A.W.; Vishwanathan, S.; Ramani, K.: Heat walk: Robust salient segmentation of non-rigid shapes. In Computer Graphics Forum, vol. 30, 2097–2106. Wiley Online Library, 2011. http://doi.org/10.1111/j.1467-8659.2011.02060.x.

[3] Cao, Y.; Yan, D.M.; Wonka, P.: Patch layout generation by detecting feature networks. Computers and Graphics, 46, 275–282, 2015. ISSN 0097-8493. http://doi.org/10.1016/j.cag.2014.09.022.

[4] Cohen-Steiner, D.; Alliez, P.; Desbrun, M.: Variational shape approximation. In ACM SIGGRAPH 2004 Papers, SIGGRAPH 2004, 905–914. Association for Computing Machinery, New York, 2004. http://doi.org/10.1145/1186562.1015817.

[5] de Goes, F.; Goldenstein, S.; Desbrun, M.; Velho, L.: Exoskeleton: Curve network abstraction for 3d shapes. Computers and Graphics, 35(1), 112–121, 2011. http://doi.org/10.1016/j.cag.2010.11.012.

[6] Dey, T.K.; Wang, L.: Voronoi-based feature curves extraction for sampled singular surfaces. Computers and Graphics, 37(6), 659–668, 2013. http://doi.org/10.1016/j.cag.2013.05.014.

[7] Digne, J.; Cohen-Steiner, D.; Alliez, P.; De Goes, F.; Desbrun, M.: Feature-preserving surface reconstruction and simplification from defect-laden point sets. Journal of mathematical imaging and vision, 48(2), 369–382, 2014. http://doi.org/10.1007/s10851-013-0414-y.

[8] Gao, X.; Shen, H.; Panozzo, D.: Feature preserving octree-based hexahedral meshing. In Computer graphics forum, vol. 38, 135–149. Wiley Online Library, 2019. http://doi.org/10.1111/cgf.13795.

[9] Guo, H.; Liu, S.; Pan, H.; Liu, Y.; Tong, X.; Guo, B.: Complexgen: Cad reconstruction by b-rep chain complex generation. ACM Transactions on Graphics, 41(4), 2022. http://doi.org/10.1145/3528223.3530078.

[10] Lai, Y.k.; Zhou, Q.y.; Hu, S.m.; Wallner, J.; Pottmann, H.: Robust feature classification and editing. IEEE Transactions on Visualization and Computer Graphics, 13(1), 34–45, 2007. http://doi.org/10.1109/TVCG.2007.19.

[11] Le, T.; Bui, G.; Duan, Y.: A multi-view recurrent neural network for 3d mesh segmentation. Computers and Graphics, 66, 103–112, 2017. http://doi.org/10.1016/j.cag.2017.05.011.

[12] Li, C.; Pan, H.; Bousseau, A.; Mitra, N.J.: Free2cad: Parsing freehand drawings into cad commands. ACM Trans. Graph. (Proceedings of SIGGRAPH 2022), 41(4), 93:1–93:16, 2022. http://doi.org/10.1145/3528223.3530133.

[13] Liu, Y.; D'Aronco, S.; Schindler, K.; Wegner, J.D.: Pc2wf: 3d wireframe reconstruction from raw point clouds. In International Conference on Learning Representations, 2021.

[14] Lu, Z.; Guo, J.; Xiao, J.; Wang, Y.; Zhang, X.; Yan, D.M.: Feature curve network extraction via quadric surface fitting. The Eurographics Association, 2019. http://doi.org/10.2312/pg.20191338.

[15] Lu, Z.; Guo, J.; Xiao, J.; Wang, Y.; Zhang, X.; Yan, D.M.: Extracting cycle-aware feature curve networks from 3d models. Computer-Aided Design, 131, 102949, 2021. http://doi.org/10.1016/j.cad.2020.102949.

[16] Pan, H.; Liu, Y.; Sheffer, A.; Vining, N.; Li, C.J.; Wang, W.: Flow aligned surfacing of curve networks. ACM Transactions on Graphics, 34(4), 2015. http://doi.org/10.1145/2766990.

[17] Pan, Q.; Xu, G.; Zhang, Y.: A unified method for hybrid subdivision surface design using geometric partial differential equations. Computer-Aided Design, 46, 110–119, 2014. http://doi.org/10.1016/j.cad.2013.08.023.

[18] Stylianou, G.; Farin, G.: Crest lines for surface segmentation and flattening. IEEE Transactions on Visualization and Computer Graphics, 10(5), 536–544, 2004. http://doi.org/10.1109/TVCG.2004.24.

[19] Takashima, H.; Kanai, S.: Shape descriptor-based similar feature extraction for finite element meshing. Computer-Aided Design and Applications, 18(5), 1080–1095, 2021. http://doi.org/10.14733/cadaps.2021.1080-1095.

[20] Torrente, M.L.; Biasotti, S.; Falcidieno, B.: Recognition of feature curves on 3d shapes using an algebraic approach to hough transforms. Pattern Recognition, 73, 111–130, 2018. http://doi.org/10.1016/j.patcog.2017.08.008.

[21] Xu, B.; Chang, W.; Sheffer, A.; Bousseau, A.; McCrae, J.; Singh, K.: True2form: 3d curve networks from 2d sketches via selective regularization. Transactions on Graphics (Proc. SIGGRAPH 2014), 33(4), 2014. http://doi.org/10.1145/2601097.2601128.

[22] Yan, D.M.; Wang, W.; Liu, Y.; Yang, Z.: Variational mesh segmentation via quadric surface fitting. Computer-Aided Design, 44(11), 1072–1082, 2012. http://doi.org/10.1016/j.cad.2012.04.005.

[23] Yingzhong Zhang, X.L.; Jia, J.: A compact face-based topological data structure for triangle mesh representation. Computer-Aided Design and Applications, 16(3), 539–557, 2019. http://doi.org/10.14733/cadaps.2019.539-557.

[24] Yoshizawa, S.; Belyaev, A.; Seidel, H.P.: Fast and robust detection of crest lines on meshes. In Proceedings of the 2005 ACM Symposium on Solid and Physical Modeling, SPM '05, 227–232. Association for Computing Machinery, New York, NY, USA, 2005. http://doi.org/10.1145/1060244.1060270.

[25] Zhuang, Y.; Dou, H.; Carr, N.; Ju, T.: Feature-aligned segmentation using correlation clustering. Computational Visual Media, 3(2), 147–160, 2017. http://doi.org/10.1007/s41095-016-0071-38.

[26] Zhuang, Y.; Zou, M.; Carr, N.; Ju, T.: A general and efficient method for finding cycles in 3d curve networks. ACM Transactions on Graphics, 32(6), 2013. http://doi.org/10.1145/2508363.2508423.

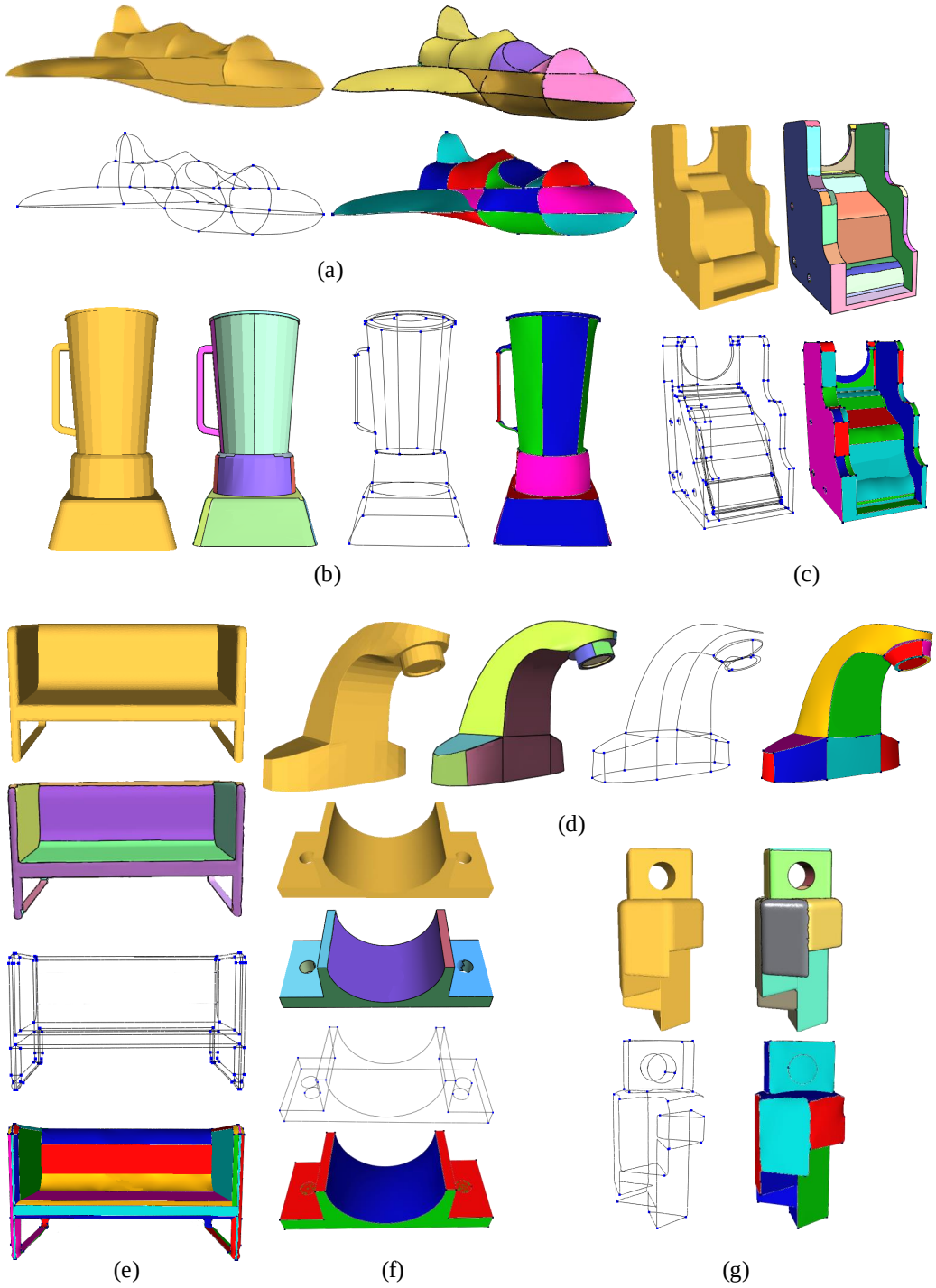


Figure 14: The original mesh, VSA patches, wireframes and the reconstructed mesh of different models.(a) Fighter (b) Mixer (c) Roller (d) Tap (e) Sofa (f) Buckle (g) Latch