DE GRUYTER
OPEN

BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **16**, No 6
Special issue with selection of extended papers from 6th International Conference on Logistic, Informatics and Service Science LISS'2016

Sofia • 2016 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.1515/cait-2016-0085

Generation Method and Application of Product-Oriented Medial Axis

Jianjun Peng[1], *Hui Wang*[1], *Jiping Li*[2], *Chuanming Song*[3]

[1]*School of Information Science and Engineering, Dalian Polytechnic University, Liaoning Dalian 116034, China*
[2]*College of Mathematics and Informatics, South China Agricultural University, Guangzhou Guangdong 510642, China*
[3]*School of Computer and Information Technology, Liaoning Normal University, Dalian 116029, China*
Emails: pengjj@dlpu.edu.cn wang_hui@dlpu.edu.cn li-jiping@163.com chmsong@lnnu.edu.cn

Abstract: *In this paper, the generation method of the medial axis in the arbitrary quadrilateral surface is proposed. It can provide a solution for the simplification of the complex fillet feature and the generation of the mesh in the model. By using the locus method associated with moving Frenet frame, we realize the simple and fast algorithm for generating the medial axis. As for the engineering problem, B-rep 3D solid models with clear boundary definition are mostly applied; the information of vertex, side and surface of the model, which is clearly stored in the model file, can be used to simplify the traditional locus method for generating the medial axis, in order to reduce the amount of data required by the generation. In this paper, we use the clear boundary information in the B-rep model as the condition for generating the medial axis and the characteristics of the bisector to eliminate the calculation of the branch points, reducing the factors affecting the accuracy of the medial axis. In order to ensure the accuracy of the medial axis, the density of the insertion points can be used for control.*

Keywords: *The medial axis, moving Frenet frame, mesh generation, the model feature simplification.*

1. Introduction

The medial axis is also referred to as the topological skeleton or center line, and is usually described as follows: In 2D closed region, there is a disc with variable radius. Keep that there are at least two tangent points between the disc and the boundary of the region, and then move the disc. After the path of the disc covers the whole region, the locus left by the centre of the disc forms the line said as the medial axis which is called the medial axis plane in the 3D model. The medial axis has a wide range of applications in each key link of the CAD/CAE system, such as model reconstruction [1], model analysis [2], computer vision [3], solid modeling and feature extraction in geometric design [4], etc. However, the medial axis is limited in some applications, because currently there is no fast and accurate algorithm for calculating it, especially in the curve boundary conditions.

In 1967, Blum [5] proposed the concept of the medial axis for the first time; it has the advantages of being unique, reversible, symmetrical, topological equivalence and so on. The medial axis is greatly affected by the continuity of the boundary, and it is difficult to extract the accurate curve boundary. 10 years ago, the bisector was introduced. It is a special case for the medial axis. However, this method needs pruning operation, increasing the complexity of the process. In recent years, Teixeira [6] and August, Tannenbaum and Zucker [7] have studied the relationship among the curvature, tangent and normal vector of the curved boundary and the medial axis, and the differential equation of the curve medial axis motion described by the mean curvature is given. Although there are a lot of algorithms about the generation of the medial axis, it is not an easy task to generate a stable and precise medial axis.

At present, the algorithms for the generation of the medial axis in the CAD/CAE system are generally divided into two categories: Locus method and Voronoi diagram [8]. Locus method is making a circle with variable radius tangent with at least two boundaries in the region according to the definition of medial axis, and the locus left by the circle center is the medial axis. However, the distance from each point to the boundary and whether or not this point is the branch one should be determined while applying that method; that is, the position with two or more tangent points, which increases the amount of data, inevitably affecting the efficiency of the calculation. Voronoi diagram is used to discretize the boundary of a given region by point set [9]. The vertexes of Voronoi diagram which originate from the discrete point set compose the medial axis. However, the algorithm is more suitable for the two-dimensional model, and it needs increasing number of additional selection conditions to expand it to three-dimensional model.

In this paper, we propose a simple and fast algorithm for generating the medial axis by using the locus method associated with moving the Frenet frame [10]. As for the engineering problem, B-rep 3D solid models with clear boundary definition are mostly applied [11], and the information of vertex, side and surface of the model which is clearly stored in the model file can be used to simplify the traditional locus method for generating the medial axis, in order to reduce the amount of data required by the generation. We use the clear boundary information in the B-rep model as the condition for generating the medial axis and the characteristics of the bisector to eliminate the calculation of the branch points in this paper, reducing the factors affecting the accuracy of the medial axis; that is, using boundary constraints to construct the medial axis. In order to ensure the accuracy of the medial axis, the density of the insertion points can be used for control.

2. Generation Algorithm of the medial axis

In this paper, we propose an algorithm for generating the medial axis of arbitrary quadrilateral by combining the locus method with moving Frenet frame. The flow chart for the Generation Algorithm of the medial axis is shown in Fig. 1.

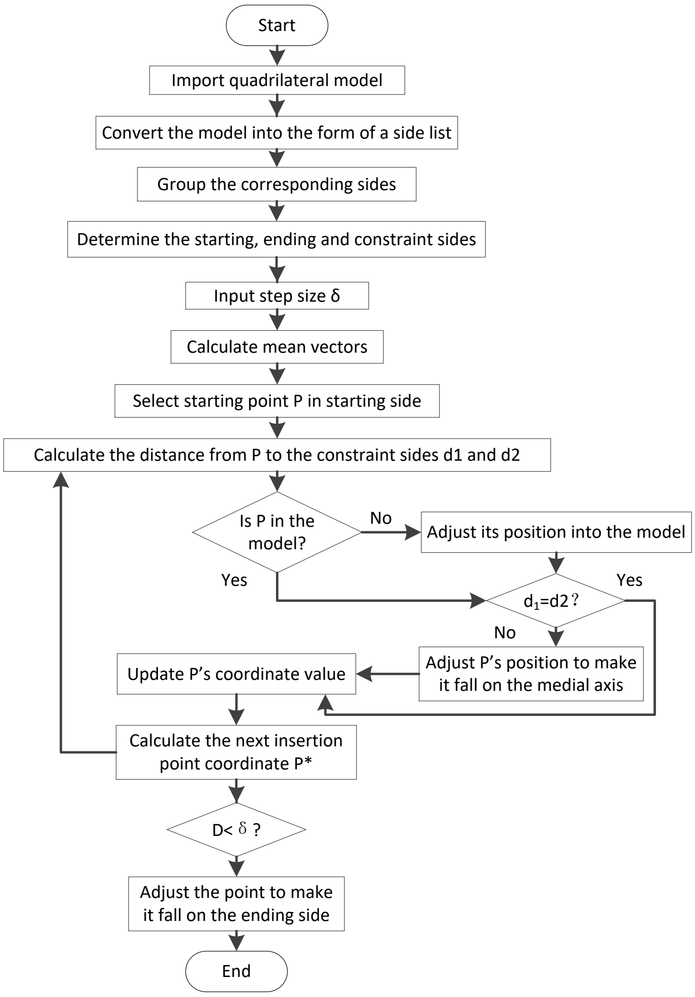

Start
Import quadrilateral model
Convert the model into the form of a side list
Group the corresponding sides
Determine the starting, ending and constraint sides
Input step size δ
Calculate mean vectors
Select starting point P in starting side
Calculate the distance from P to the constraint sides d1 and d2
Is P in the model?
No
Adjust its position into the model
Yes
d1=d2？
Yes
No
Adjust P's position to make it fall on the medial axis
Update P's coordinate value
Calculate the next insertion point coordinate P*
D< δ ?
Adjust the point to make it fall on the ending side
End

Fig. 1. The flow chart for the Generation Algorithm of the medial axis

Step 1. The quadrilateral model is transformed into a side-based element, and the sides are grouped for storage

For any quadrilateral model, the surface model is converted into the form of a side list, and the sides are divided into two groups according to the principle of "side and side are not adjacent". As shown in Fig. 2, A_1 and A_2 belong to one group, while B_1 and B_2 are in the other group.

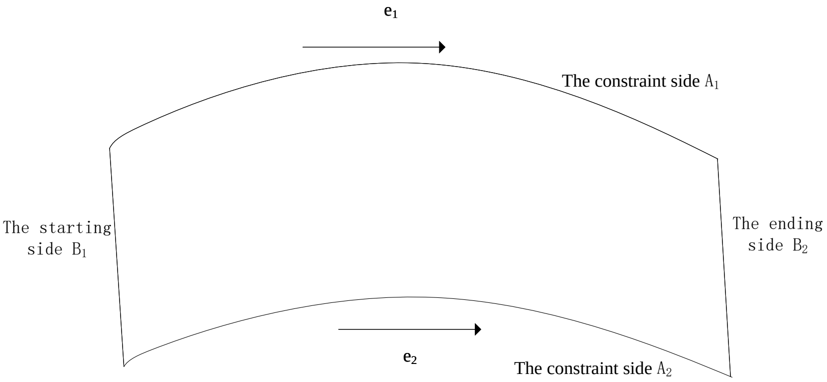


Fig. 2. A schematic diagram of grouping for each side of the quadrilateral

Step 2. Setting the constraint conditions and calculating the insertion point coordinates

We select the sides from one group as the starting side and the ending one, while those from the other group are the constraint sides. We insert the points according to the step size *δ* from the starting side to the ending one until the insertion point is close to the ending side.

The step size *δ* is the distance between the two insertion points, and it can control the number of insertion points. The smaller the step size is, the more the insertion points are, and the higher the medial axis precision is generated. We adopt one fifth of the shortest side in the model as *δ* here. As shown in Fig. 3, B_1 is set to start, and B_2 is the ending side. A_1 and A_2 are constraints. In the direction from the starting side to the ending one, we obtain the mean vectors and of the two constraints. Calculate by vector combination formula

(1) $$\vec{e} = \vec{e}_1 + \vec{e}_2,$$

that is, the interpolation direction of the insertion points.

In the starting side, the point P_1 is taken as the starting point. According to the interpolation direction and the step size *δ*, $P_2(x, y)$, the coordinate of the next insertion point, is calculated by

(2) $$\begin{cases} P_2.x = P_1.x + \vec{e}.x * \delta, \\ P_2.y = P_1.y + \vec{e}.y * \delta. \end{cases}$$

If the distance *d* from P_2 to the ending side is less than the step size *δ*, indicating that the new insertion point is close to the ending side, a new interpolation point is no longer required. In that case, P_2 is the ending point. Otherwise, continue to insert a new point *P**.

Step 3. Adjusting the position of each insertion point to make it fall within the model

We determine whether or not the insertion point is within the model region. If not, the insertion point is adjusted into the model region [12].

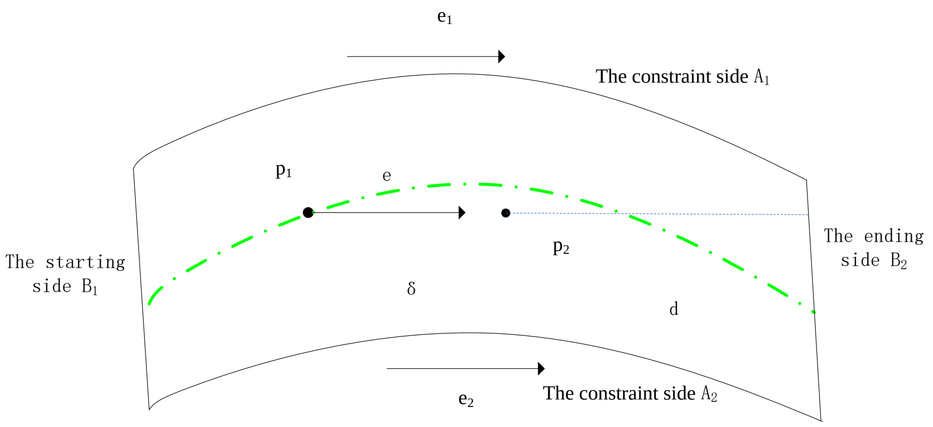


Fig. 3. Tracing algorithm of inserting points

As shown in Fig. 4, when the insertion point P is obtained, d_1 and d_2 are calculated, also the shortest distances from P to the two constraints, as well as q_1 and q_2, the points corresponding to the shortest distances. The direction vector, $\vec{e}_1 = q_1q_2/\|q_1q_2\|_2$ is obtained by Frenet frame formula. It is determined whether or not the position of the insertion point P is within the model region. If not, it will be adjusted into the model. Judgment and adjustment methods are as follows.

According to q_1 and q_2 corresponding to the shortest distances between the point P and the constraints, the distance $d = \|q_1q_2\|$ is obtained. The relationship among d_1, d_2 and d is compared to the shortest distances from P to the constraints.

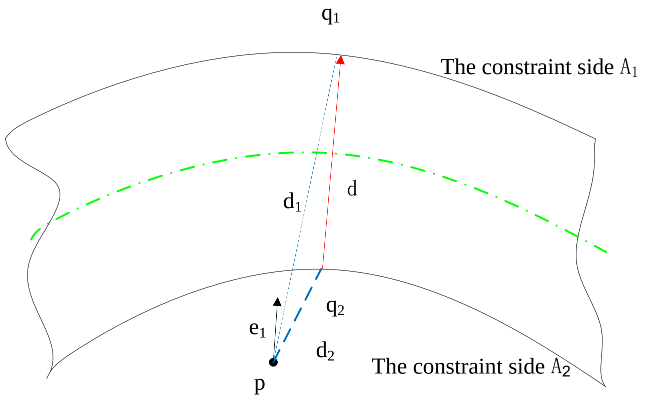


Fig. 4. Diagram of the algorithm for insertion points

(a) If $d_1>d$ or $d_2>d$, point P is outside the model. If $d_1>d_2$, point P moves in the direction of for a certain distance $\Delta = |(d_1 - d_2)|/2$； if $d_1<d_2$, point P moves in the opposite direction of for a certain distance $\Delta = |(d_1 - d_2)|/2$. After the insertion point P moves into the model, the point coordinates are updated.

(b) If $d_1<d$ and $d_2<d$, it shows that the point P is in the model. Then it is not necessary to adjust.

Step 4. Adjusting the position of each insertion point to make it fall on the medial axis

Adjusting the position of the insertion point in the model to make it fall on the medial axis.

As shown in Fig. 5, the insertion point is adjusted according to the following rules:

(a) If $d_1 > d_2$, P moves in the direction for a certain distance Δ;

(b) If $d_1 < d_2$, P moves in the opposite direction for a certain distance Δ;

(c) If $d_1 = d_2$ (approximately equal), update the coordinate value of P.

In the paper, the initial value of the moving distance Δ is set to be $|(d_1 - d_2)|/2$. When the two successive moving directions are inconsistent, the second moving distance Δ is reduced to be $|(d_1 - d_2)|/4$. By iterative algorithm, repeatedly adjust the position of the point until the distances that point P reaches the two constraints are equal. The value of moving distance is $\Delta = |(d_1 - d_2)|/2n$， including the value of n is the times of the inconsistency of the two successive moving directions. Adjust repeatedly the position of the point by iterative method. If it is difficult to make the distances between the point and the constrains exactly equal, being approximately equal, i.e., $d_1/\ d_2 = \varepsilon$ (ε is close to 1), is applicable. For high-precision medial axis, the closer is ε to 1, the better, but the calculation efficiency will be reduced. In the present invention, ε in each model is adopted as $0.99 < \varepsilon < 1.01$.

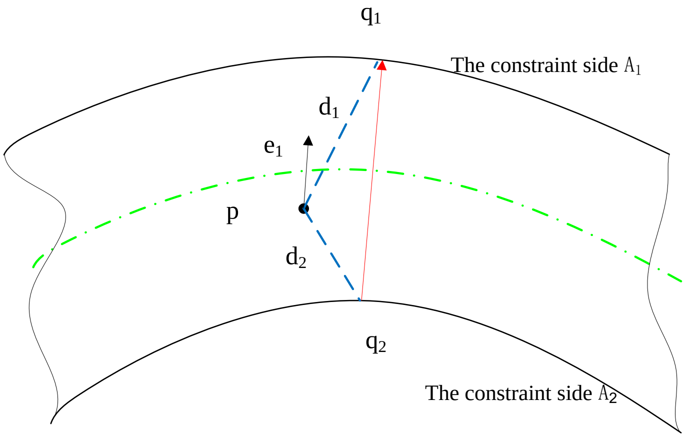
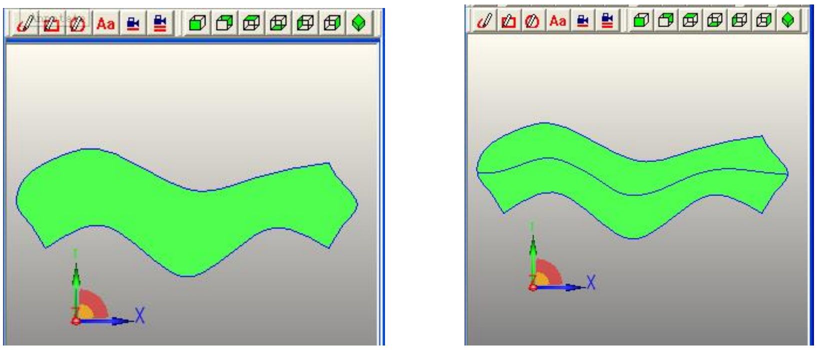


Fig. 5. Diagram of adjusting the position of insertion point on medial axis

Step 5. B-spline curve is obtained by using the points on the medial axis to form the smooth curve which is approximately the medial axis of the surface model [13]

Fig. 6 shows the effect diagram of the medial axis of the quadrilateral with four irregular bending sides. In the CAD models of industrial production, many models are regular. But there are some irregular models. The algorithm of generating the medial axis is tested by using the irregular boundary model with high complexity. As shown in Fig. 6a, the quadrilateral with irregular bending sides is adopted. After introducing that model, apply the medial axis generation algorithm, and the effect is shown in Fig. 6b. Although the bilateral bending causes the direction of the insertion point to be not exactly equal to the direction vector of the constraint, the insertion point can fall on the medial axis by the adjustment. And then use the points on the medial axis to construct B-spline curve which can equally divide the medial axis of the bending quadrilateral, to fulfil the symmetry of the medial axis.



(a) (b)

Fig. 6. The effect diagram of the medial axis of the quadrilateral with four irregular bending sides

3. The fillet feature simplification method based on Medial Axis Algorithm

According to the method of generating the fillet feature surface, the fillet surface can be divided into the point-transition fillet surface and the side-transition one [14]. We refer to the simplified method for the point-transition fillet feature [15, 16]. The side-transition fillet feature is made up of four sides. It can be divided into two groups according to the characteristics of the sides. Constraints can belong to one group, while the common ones- to the other, in order to meet the above generation algorithm of the medial axis. Therefore, the medial axis which can be generated on the fillet feature surface provides the basis for realizing the simplification of the fillet feature.

The fillet feature simplification method based on Medial Axis Algorithm is shown in Fig. 7.

Step 1. The model is kept by the surfaces as element, and each surface is distinguishing

The three-dimensional solid model is transformed into the form of the surface list. The surfaces are divided into three categories: Smooth surface, supporting surface and protection surface, stored as three lists, see [14].

Step 2. The supporting surface is kept by the sides as element, and the sides are classified and stored

A supporting surface is removed from the list of supporting surfaces and stored in the form of sides. Then, the sides are divided into three categories: Smooth, semi smooth and non smooth sides, stored in three lists [17].

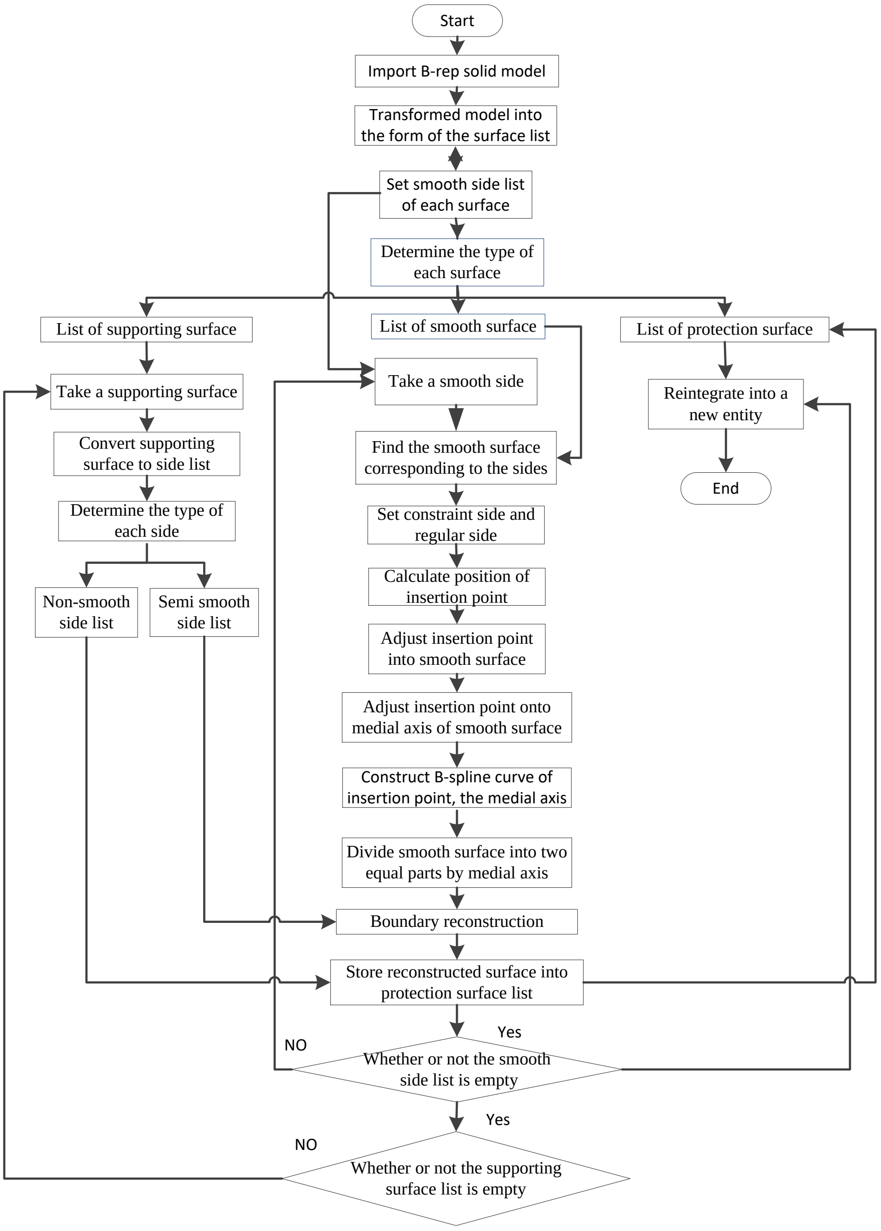

Start
Import B-rep solid model
Transformed model into the form of the surface list
Set smooth side list of each surface
Determine the type of each surface
List of supporting surface
List of smooth surface
List of protection surface
Take a supporting surface
Take a smooth side
Reintegrate into a new entity
End
Convert supporting surface to side list
Find the smooth surface corresponding to the sides
Determine the type of each side
Set constraint side and regular side
Non-smooth side list
Semi smooth side list
Calculate position of insertion point
Adjust insertion point into smooth surface
Adjust insertion point onto medial axis of smooth surface
Construct B-spline curve of insertion point, the medial axis
Divide smooth surface into two equal parts by medial axis
Boundary reconstruction
Store reconstructed surface into protection surface list
NO
Yes
Whether or not the smooth side list is empty
Yes
NO
Whether or not the supporting surface list is empty

Fig. 7. Diagram of the fillet feature simplification method based on Medial Axis Algorithm

As shown in Fig. 8, the supporting surface F_1 is removed from the list, and its sides (e_1, e_2, e_3, e_4) are deposited respectively in the lists; e_4 is the common side of the supporting surface and smooth surface, i.e., smooth side, deposited in the smooth side list; e_2 and e_3 have the smooth points, namely semi smooth sides, deposited in semi smooth side list; e_1 is non smooth side, deposited directly into the non smooth side list.

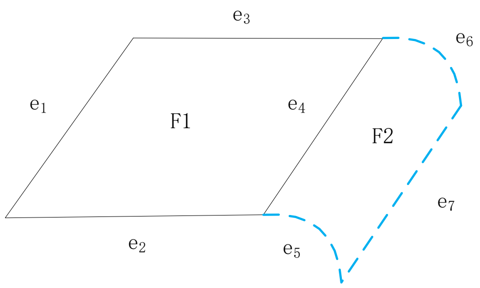


Fig. 8. Diagram of classification for the sides of supporting surface

Step 3. Generating the medial axis and decomposing the smooth surface

A side is removed from the smooth side list. According to that smooth side, we find the corresponding smooth surface from the list of smooth surfaces. The smooth surface is stored as a list of sides. The smooth surface comprises two smooth sides and two regular sides. With a smooth side as a constraint condition, regular side as a starting side and the ending one, we use the medial axis algorithm to generate the medial axis of the smooth surface. The medial axis divides the smooth surface into two equal surfaces. At the same time, the two ends of the medial axis divide the two regular sides into four sections which are stored in the list of the segmentation sides.

As shown in Fig. 9, the smooth surface F_2 can be found via the smooth side e_4 of the supporting surface F_1. The medial axis e_8 is generated with the smooth sides e_4 and e_7 as the constraints, as well as non smooth sides e_5 and e_6 as the starting side and the ending one. Divide the smooth surface F_2 into two surfaces F_3 and F_4; at the same time, the two ends of the medial axis divide the side e_5 into two sections a_1 and a_2， and e_6 is divided into b_1 and b_2. The four sides are deposited in the segmentation side list.

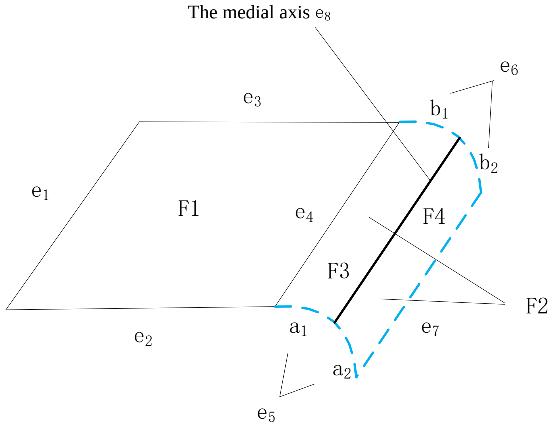


Fig. 9. Diagram of smooth surface decomposition

Step 4. Reconstructing the external contour of the supporting surface

The decomposed smooth surface and the corresponding supporting surface are reintegrated into a surface. The original supporting surface and the independent smooth surface no longer exist. The reconstructed supporting surface is stored in the protection surface list.

As shown in Fig. 10, the supporting surface F_1 comprises four sides: e_1, e_2, e_3 and e_4. The contour of the reconstructed surface comprises e_1, e_2, e_3, e_8, a_1 and b_1. The new surface boundary is the extension of the original boundary, with the generated medial axis replacing the smooth side e_4; the semi smooth sides e_2 and a_1 are integrated into a new side, and the semi smooth sides e_3 and b_1 into a new one. Therefore, the new contour is made up, stored in a new side list. The surface constructed according to the new side list is an integration of the original fillet surface and the supporting surface, stored in the protection surface list (Fig. 11).

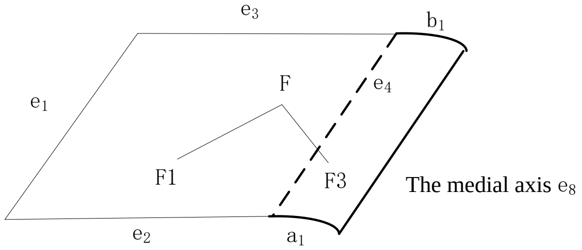


Fig. 10. Diagram of the construction for the "side" of the fillet surface and the supporting surface

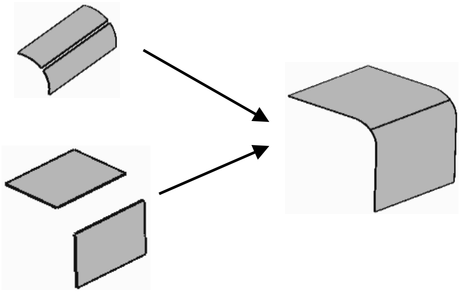


Fig. 11. Diagram of the construction for the "surface" of the fillet surface and the supporting surface

Step 5. Reconstructing the supporting surface to simplify the model feature

When the smooth side list and the supporting surface list are empty, the decomposition of all the smooth surfaces and the reconstruction of all the supporting surface contour structure are completed. All the surfaces in the protection surface list are reconstructed into an entity, to simplify the fillet feature of the target object. That is, deleting the fillet feature and the supporting surface of the original model, replacing them with a new generated surface, and reconstructing them, to form a complete model.

As shown in Fig. 12, the original protection surface (Fig. 12a), and the additional the reconstructed supporting surface (Fig. 12b), are stored in the protection surface list. Reconstructing those surfaces into an entity, we can obtain a simplified model of Fig. 12c.

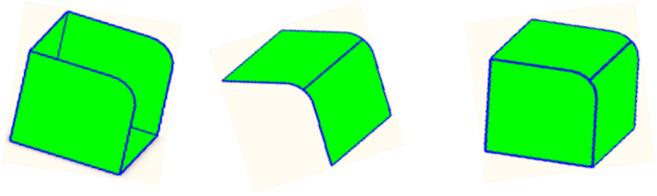


Fig. 12. Diagram of 3D entity reconstruction:
Protection surface (a); reconstructed surface (b); simplified model (c)

If there are concave and convex features on the supporting surface or the fillet surface of the original model, the inner and outer rings which represent the features should be fused with the outer boundary of the new bending surface, as shown in Fig. 13.

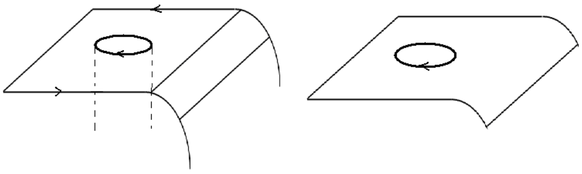


Fig. 13. The integration of the surfaces with concave and convex features

4. Mesh generation method based on the Medial Axis Algorithm

The mesh generation is to disperse a continuous region into finite elements, and set up finite nodes in each element [18]. The continuous region is seen as the collection of the elements with connection at nodes [19]. That is to say, the mesh generation is to divide a given region into several sub regions, decomposing the given region into several sub regions.

In the algorithm of the medial axis, the boundary is used as the constraint condition for generating the medial axis. We use the medial axis to segment the model, and the boundary of each sub region is also restricted by the boundary, making the boundaries of each sub region similar. The medial axis generated from the given region divides the region into two equal parts. If the two regions continue to be divided, a number of approximately equal sub regions are obtained. We swap the relationship between the constraint sides and the starting, the ending sides, and again implement the same division. The obtained series of the continuous elements are the generated mesh.

As shown in Fig. 14a, the closed region is divided into the quadrilateral comprising *ab*, *bc*, *cd*, and *da*; *ab* and *cd* are used as the constraint condition to generate the medial axis *ef*, which decomposes the region into two regions, as shown in Fig. 14b. And then determine it with condition: The shortest side in each region is less than or equal to the initial mesh boundary setting value. If the determinant condition is not satisfactory, re-divide it. As shown on Fig. 14c, *ae*, *fd* and *eb*, *cf* are used as constraints respectively to regenerate two medial axes, and the original region is divided into four sub regions, which meets the determinant condition for the termination. Swap the relationship between the constraint sides and the starting, ending sides of the original model, and implement the same division with *bc* and *da* as the constraint condition, to generate three medial axes on Fig. 14d. Six medial axis generated from the division in two different directions intersect, and the intersection nodes are the mesh nodes connected into a series of continuous quadrilateral, that is the mesh element, as shown on Fig. 14e.

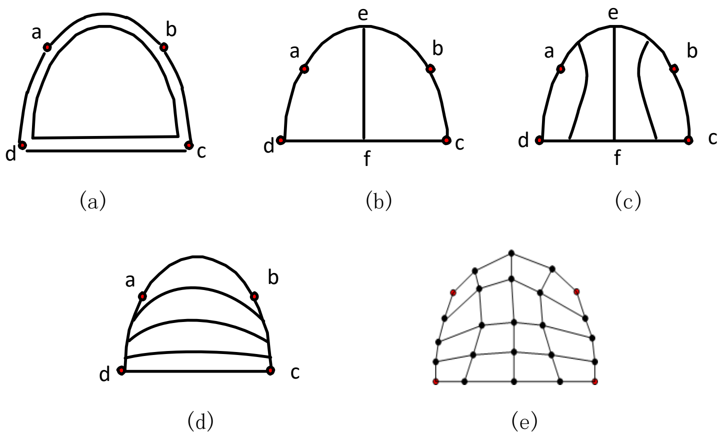
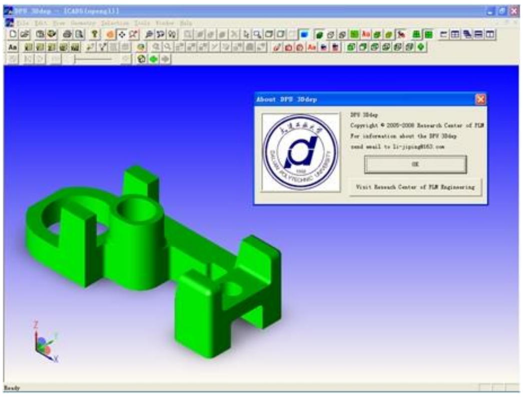


Fig. 14. Generation of the mesh based on the Medial Axis Algorithm

5. Method verification

The above simplification method of the model fillet feature based on the Medial Axis Algorithm and the generation method of the mesh have been integrated in the self-research three-dimensional CAD data processing software system DPU 3Ddep, see Fig. 15 [17].

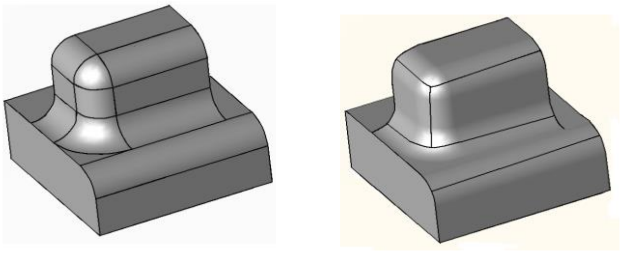

DPU 3Ddep - [CAD5(openg1)]
File Edit View Geometry Selection Tools Window Help
About DPU 3Ddep
OK
Visit Research Center of PLM Engineering
Ready

DPU 3Ddep
Copyright © 2005-2008 Research Center of PLM
For information about the DPU 3Ddep
send email to li-jiping@163.com

Fig. 15. CAD data processing software system DPU 3Ddep

5.1. Verification for the simplification method of fillet feature

Fig. 16a shows a mechanical part composed of 31 surfaces which contain a plurality of fillet features. The medial axis is calculated from the smooth surfaces and the outer contour is reconstructed from the supporting surfaces. Using the method in the paper, the simplified model is suggested as shown on Fig. 16b with the number of surfaces reduced to 11. The test result shows that the external shape of the models which fillet features are simplified do not change, indicating that the method is feasible and effective.



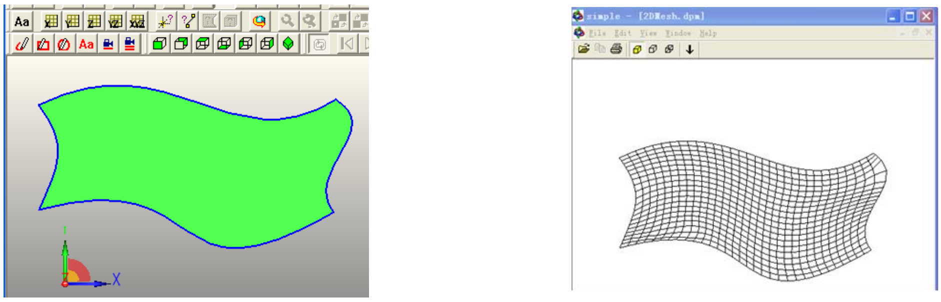
(a) (b)

Fig. 16. Effect diagram of fillet simplification of mechanical part model:
Mechanical part model (a); simplification effect (b)

5.2. Verification for the mesh generation method

The algorithm is verified by using the model according to the mesh generation method based on the medial axis algorithm, and the result is visualized by Hoops.

As shown in Fig. 17a, the quadrilateral with high complexity, irregular shape and complete curved sides is used to verify the algorithm of mesh generation. On Fig. 17b we can see the number of quadrilateral surfaces of the model decomposition; 31 medial axes are generated on the surfaces of the model. When each side is discretized, and 33 boundaries are added, the total number of generated points is 693. Even if the inner quadrilateral element is affected by the boundary bending deformation, each quadrilateral element seems structured, and the mesh quantity can be changed to control the mesh quality.



(a) (b)

Fig. 17. Mesh generation of the quadrilateral with four curved sides:
The quadrilateral with four curved sides (a); the generation of the mesh parameter (b); the generation result of the mesh (c)

6. Conclusion

The medial axis plays a key role in the model simplification, model reconstruction, mesh generation and solid modeling which are the indispensable parts of the CAD/CAE system. In this paper, we propose a simple and fast algorithm for generating the medial axis by locus method associated with moving Frenet frame, based on which we realize the simplification algorithm of the complex fillet features and achieve the simplification effect without changing the model structure. Besides, we apply the medial axis to the mesh generation, to realize the mesh generation of the quadrilateral region; the visual function of the data file in hoops makes users understand the quality of the mesh. In the future, the medial axis will be applied to the hexahedral finite element mesh generation [20]. Since the medial axis has been verified for plane, it can also generate the high-quality hexahedral mesh by becoming the medial surface in 3D space to equally divide the space [21].

Acknowledgments: The authors wish to express their gratitude towards the financial supports from Science and Technology Department of Liaoning province (No 2014026001, No 20141112) and Education Department of Liaoning province (No L2014219).

References

1. He, S., Y. Choi et al. A 3D Shape Descriptor Based on Spectral Analysis of Medial Axis. – Computer Aided Geometric Design, Vol. **39**, 2015, pp. 50-66.
2. Nishino, K., G. Omori et al. Three-Dimensional Dynamic Analysis of Knee Joint During Gait in Medial Knee Osteoarthritis Using Loading Axis of Knee. – Gait & Posture, Vol. **42**, 2015, No 2, pp. 127-132.
3. Saúde, A. V., M. Couprie, R. A. Lotufo. Discrete 2D and 3D Euclidean Medial Axis in Higher Resolution. – Image and Vision Computing, Vol. **27**, 2009, No 4, pp. 354-363.
4. Quadros, W. R. LayTracks3D: A New Approach to Meshing General Solids Using Medial Axis Transform. – Procedia Engineering, Vol. **82**, 2014, pp. 72-87.
5. Blum, H. A Transformation for Extracting New Descriptors of Shape. – Models for the Perception of Speech & Visual Form, Vol. **19**, 1967, pp. 362-380.
6. Teixeira, R. C. Medial Axes and Mean Curvature Motion I: Regular Points. – Journal of Visual Communication & Image Representation, Vol. **13**, 2002, No 1, pp. 135-155.
7. August, J., A. Tannenbaum, S. W. Zucker. On the Evolution of the Skeleton. – In: ICCV, 1999.
8. Held, M. VRONI: An Engineering Approach to the Reliable and Efficient Computation of Voronoi Diagrams of Points and Line Segments. – Computational Geometry, Vol. **18**, 2001, No 2, pp. 95-123.
9. Deza, M., M. D. Sikirić. Voronoi Polytopes for Polyhedral Norms on Lattices. – Discrete Applied Mathematics, Vol. **197**, 2015, pp. 42-52.
10. Cao, L., J. Liu. Computation of Medial Axis and Offset Curves of Curved Boundaries in Planar Domain. – Computer-Aided Design, Vol. **40**, 2008, No 4, pp. 465-475.
11. Boussuge, F., et al. Extraction of Generative Processes from B-Rep Shapes and Application to Idealization Transformations. – Computer-Aided Design, Vol. **46**, 2014, pp. 79-89.
12. Zhu, Y., et al., Computing a Compact Spline Representation of the Medial Axis Transform of a 2D Shape. – Graphical Models, Vol. **76**, 2014, No 5, pp. 252-262.
13. Ahn, S., et al. Differentiation of Human Pluripotent Stem Cells into Medial Ganglionic Eminence vs Caudal Ganglionic Eminence Cells. – Methods, Vol. **101**, 2016, pp. 103-112.
14. Peng, J., R. Guo et al., Research and Application of NC Machining Simulation System. – Journal of Chinese Computer Systems, Vol. **31**, 2010, No 6, pp. 1240-1244.

15. Kim, B. C., D. Mun. Feature-Based Simplification of Boundary Representation Models Using Sequential Iterative Volume Decomposition. – Computers & Graphics, 2014, No 38, pp. 97-107.
16. Kwon, S., B. Kim et al. Simplification of Feature-Based 3D CAD Assembly Data of Ship and Offshore Equipment Using Quantitative Evaluation Metrics. – Computer-Aided Design, 2015, No 59, pp. 140-154.
17. Li, J., G. Tong et al., Automatic Small Blend Recognition from B-Rep Models for Analysis. – Engineering with Computers, Vol. **25**, 2009, No 3, pp. 279-285.
18. Yang, S., E. Dilay et al. Volume Element Model Mesh Generation Strategy and Its Application in Ship Thermal Analysis. – Advances in Engineering Software, 2015, No 90, pp. 107-118.
19. Fogg, H. J., C. G. Armstrong, T. T. Robinson. Enhanced Medial-Axis-Based Block-Structured Meshing in 2D. Computer-Aided Design, 2015.
20. Wei, Q., Y. Hatori, K. Sakai. Neural Construction of 3D Medial Axis from the Binocular Fusion of 2D MAs. – Neurocomputing, 2015, No 149, pp. 546-558.
21. Zhang, X., Y. Xia et al. Medial Axis Tree – An Internal Supporting Structure for 3D Printing. – Computer Aided Geometric Design, 2015, No 35-36, pp. 149-162.