Generation of Hierarchical Multi-resolution Medial Axis for CAD Models

Housheng Zhu¹, Yusheng Liu^{1*}, Jianjun Zhao^{2*}

¹State Key Lab. of CAD&CG, Zhejiang University, Hangzhou, P.R. China, 310027

²School of Mechanical Science & Engineering, HUST, Wuhan, P.R. China, 430074

(ysliu@cad.zju.edu.cn)

Abstract

Medial axis (MA) is used as an effective description for objects in many engineering fields. A difficulty for the current methods for the generation of MA of CAD models is the balance between the efficiency and the quality. In this study, an approach to iteratively generating hierarchical multi-resolution MA is proposed. In each iteration, only a small part of MA that affects MA quality is refined, by which the time cost and the space cost are reduced greatly. First, the model is voxelized and its initial MA is generated by distance dilation method. Meanwhile, the MA quality is computed and evaluated. Second, if the MA quality doesn't satisfy the requirement, upgrade the MA level and re-compute the local MA in the affected region until the MA quality does. Finally, by combining the local MA in the affected region with the reused MA in other regions, hierarchical multi-resolution MA is obtained. Several examples are given to demonstrate the outperformance of the proposed method in terms of time and space.

Keywords: Hierarchical computation, Medial axis; Multi-resolution; CAD model; Distance dilation

1. Introduction

Medial axis (MA) is first proposed by Blum [1] as an effective description for objects. The MA of an object is the set of all points having more than one closest point on the object's boundary [2]. It has been widely used in many engineering fields, such as finite element analysis, shape analysis, robot route planning, solid modeling, and mesh generation [3-5]. For a complicated CAD model, as the

computational complexity is too high, it is difficult to compute its accurate MA. Therefore, researchers usually use approximation methods to obtain the MA of a CAD model. For example, the surface-based method [6] computes approximate MA represented by points. The MA usually consists of some MA segments, each of which is generated by one or two boundary faces. Here, an MA segment here may refer to a curve edge/face/point based on the context. Moreover, the connection between MA segments can be obtained directly during the generation process by checking common MA voxels of different MA segments. However, the resultant MA of the surface-based method is difficult to be transformed to MA segments.

Another work is thinning method based on distance transformation [7]. During the thinning process of a CAD model, the input model is usually first discretized into some voxels under a certain resolution, and then the MA represented by voxels is computed. However, it is not easy to satisfy both the requirement of the MA quality and the computational efficiency at the same time in the computation of the MA of a CAD model. If the model resolution is low, the MA quality will be low since the details of the MA cannot be properly represented by large voxels although the computational efficiency is high. On the contrary, the time cost and space cost will be intolerable although the MA quality is high if the model resolution is high. How to balance the quality and the efficiency is a difficult

problem during the MA generation. To control the resolution, Gao *et al.* [8] proposed a method of multi-resolution MA generation. In their method, the concept “dilation unit” is used to combine the small boundary faces with large boundary faces to ignore their impact on the MA. Their method can efficiently suppress noises and thus improve the MA robustness. However, since the voxel resolution is still constant in this method, the MA segments generated by non-noise small boundary faces that have to be concerned still cannot be well computed efficiently.

Based on above analysis, a method is proposed to compute the hierarchical MA of a CAD model with the help of multi-resolution voxels. The main idea is that based on the initial MA, local MA is refined iteratively to improve the MA quality until the requirement is satisfied. In the hierarchical MA, high-resolution voxels are used for the small boundary faces and MA segments generated by them to embody the details while normal-resolution voxels are for other boundary faces and MA segments generated by them. Therefore, high-quality MA can be generated at both low time cost and low space cost. The novelty of this study will be reflected in the following aspects. (1) A method to evaluate the MA quality is proposed based on boundary faces and MA segments. (2) In each model level, to refine the model based on the voxelized model efficiently, a method is proposed to determine the details of the model, called the re-voxelization region and conduct multi-resolution voxelization in the re-voxelization region by high-level voxels. (3) The affected region is proposed only in which the local MA needs to be refined to improve the MA quality. By combining the new local MA to the reused MA, the hierarchical MA is obtained. (4) The adaptive distance dilation is proposed to refine MA voxels by using boundary voxels in re-voxelization region.

Based on the adaptive distance dilation, an adaptive double queues distance dilation based algorithm is proposed, which can be used to compute voxel distances for generating MA voxels.

As the re-voxelization region and the affected region are both much smaller than the entire model, not only the high-quality MA can be achieved, but also low time complexity and space complexity is ensured.

This paper is organized as follows. Related work is introduced in Section 2 while the basic concept and method overview are introduced in Section 3. In Sections 4, 5 and 6, the multi-resolution voxelization, computation of multi-resolution voxel distances and multi-resolution MA generation method are detailed respectively. The complexity, robustness and accuracy are analyzed in Section 7. Some examples are illustrated in Section 8 while conclusions are given in Section 9.

2. Related work

The traditional methods of MA generation can be divided into three categories: thinning method, tracing method and Voronoi-graph based method. Here only the thinning method which is related to this paper is introduced. For more detail information about MA, readers can refer to Ref. [9].

In thinning method, a model is usually first voxelized to achieve its MA. The error of resultant MA is decided by the resolution of voxelization. Lam *et al.* [10] introduced thinning method systematically. Nackman [5] proposed a method which replaces the smooth boundary by a polygon/polyhedron, and thus MA of polygons can be used as MA of the input model. Based on the diffusion of the combined waves, Scott *et al.* [11] proposed a method to compute the symmetric axis of a method as a superset of the MA. This method is very effective for 0-1 images, but the computational error is considerably greater for

color images. Siddiqi *et al.* [12] proposed a vector field-based thinning method. In the approach by Vleugels *et al.* [13], voxelization was recursively performed on a space that contained MA voxels until the necessary resolution was reached. To preserve the topology of the model, Borgefors *et al.* [14] proposed a thinning method for 3D skeletonization that can be used to compute both the surface skeleton and the curve skeleton. In their method, the Euclidean distance can be represented by the local distances, and an optimal method is given to compute the minimum distance between two points [15].

To reduce the computational complexity, many researchers have proposed various strategies. Sherbrooke *et al.* [16] built the linearly approximate segments of the curve in the direction of the tracing. Their method can be used for conveniently computing the MA of polyhedrons. Using a piecewise circular boundary conversion, Aichholzer *et al.* [17] handled models with free-form shapes to efficiently generate the MA. Moreover, with their method, convergence is guaranteed. To speed up the Voronoi graph-based MA generation method, Meijster *et al.* separated the grid points of the image into independent rows and columns; thus, it is well suited for parallelization on a shared-memory machine [18]. Hirata devised an efficient algorithm for each row to find the lower envelope of the minima of the set of distance functions [19].

Some researchers also considered the generation of multi-resolution MA. Miklos *et al.* [20] proposed a method which generated the MA in different abstract levels. Reniers *et al.* [21] robustly computed line skeleton and curve skeleton for 3D models under multiple levels. However, those methods [20, 21] cannot represent MA for details of the model efficiently. Gao *et al.* [8] also proposed a method to generate multi-resolution MA based on distance-dilation method in which the MA

voxel size is constant. When MA voxel size is small to represent details of the model, the computational cost of their method will be too high to be implemented.

Another typical work is the parallel computation of MA generation by multiple processors. A parallel banding algorithm based on the GPU was proposed in Ref. [22]. In this method, exact Euclidean distance transform (EDT) was computed efficiently for 2D images. However, the degree of parallelization is low when the texture sizes are high. Moreover, many researchers also proposed similar GPU based parallel methods [6, 23-26]. Recently, a parallel method based on multi-CPU for MA generation is proposed [27]. However, those methods cannot generate high-quality MA that maps to MA segments.

3. Basic Concept and Method Overview

3.1 Basic concept

To facilitate the reader's understanding, some related basic concepts are introduced here first here.

Definition 1: *The distance of voxel A* means the distance between voxel A and its nearest boundary voxel. The nearest boundary voxel is called touch voxel of voxel A [7]. Here the distance between two voxels refers to the Euclidean distance between their centers. Particularly, the distance of a boundary voxel is 0 and its touch voxel is itself.

Definition 2: *Hierarchical multi-resolution voxels* are hierarchical voxels which have different voxel sizes. Initially, a voxel of level 0 (called L_0 voxel for short) represent a voxel in

Fig. 1 The hierarchical multi-resolution voxels

basic resolution, which is also called a basic voxel with edge size of E_0 . Similarly, the edge size of an L_i voxel is called E_i . Here, $E_i = E_0/2^i$. A basic voxel can be virtualized and be divided into $8^i * L_i$ child voxels, which means that the *refining level* of the basic voxel is i (called R_i basic voxel for short). As shown in Fig.1a, an R_0 basic voxel A is shown. When the refining level of basic voxel A is upgraded to 1, it is divided into 8 L_1 child voxels as shown in Fig.1b. Then, when the refining level of basic voxel A is upgraded to 2, it is divided into 64 L_2 child voxels as shown in Fig.1c.

Definition 3: The MA level represents the hierarchy of MA of different qualities. The MA level of initial MA is 0, which means that the model only contains L_0 voxels. When the MA level is upgraded to i gradually, the model will contain L_i voxels. When the MA level is i , the voxelization is called L_i voxelization and the MA generation is called L_i MA generation.

Definition 4: The *re-voxelization region* is the region which contains the details of the model. As only those details affect the MA quality, they need to be re-voxelized.

Definition 5: The *affected region* is the region only in which the voxel distances and MA will change when the MA level is upgraded according to the change of *re-voxelization region*. The voxels in an affected region is called *affected voxels*.

Definition 6: The *basic resolution* is the resolution represented by voxel numbers on three coordinate axes or on longest coordinate axis only under MA level 0. When the MA level is upgraded to i , the resolution of L_i voxels becomes 2^i times of the basic resolution.

3.2 Method overview

With the consideration that B-Rep models are the dominant representations in the CAD system, the B-Rep model is used as the input model and its hierarchical MA is generated as

follows which is illustrated in Fig.2.

- (1) Multi-resolution voxelization: Based on the current MA level and current MA quality, determine the re-voxelization region and conduct local voxelization in this region to generate smaller voxels. Initially in L_0 voxelization, the entire model is voxelized.
- (2) Computation of the affected region: According to the change of boundary voxels in multi-resolution voxelization, compute the affected region under the current MA level. Initially in L_0 MA generation, the affected region is the entire model.

Fig. 2 The overview of the proposed method

- (3) Multi-resolution MA generation: Under the current MA level, compute the distances only for the voxels in the affected region. Then, compute the local MA in the affected region by using these voxel distances. Finally, combine the local MA in affected region with other MA to obtain the final MA.
- (4) Evaluation of the MA quality: Compute the MA quality of the current MA level. If the MA quality satisfies the requirement, the method ends and MA of all levels is saved. Otherwise, the MA level is upgraded by 1 and turn to step 1 to generate the MA of a higher level.

4. Multi-resolution Voxelization

In this section, multi-resolution voxelization is proposed. Initially, L_0 voxelization voxelizes the entire model and ray-tracing method [28] is used for voxelization. For voxelization of higher level, first, the re-voxelization region is determined. After that, voxelization is conducted in this region.

4.1 Determination of the Re-voxelization region

The upgrade of MA level is caused by the unsatisfactory MA quality. As a result, the re-voxelization region needs to cover the region that affects the MA quality. To measure the MA quality, quality metric of MA is proposed first.

4.1.1 MA quality metric

An MA is a set of some connected MA segments which are generated by interactions between boundary faces. Normally, an MA segment is generated by one or two boundary faces. Therefore, a boundary face maps to some MA segments. As shown in Fig.3a, six boundary faces of a cuboid generate 13 MA

segments, as shown in Fig.3b. These six boundary faces are called *MA-sensitive boundary faces* as they are involved in MA generation. On the contrary, some boundary faces are too small to be involved in MA generation and they are called *MA-insensitive boundary faces*. For example, the cuboid in Fig.3a has a small concave part marked by a red circle. However in Fig.3b, L_0 MA doesn't contain any MA segment generated by the boundary faces of the concave part, which means that those boundary faces are MA-insensitive boundary faces. Specifically, an MA-sensitive boundary face needs to map to a normal MA segment. Here a normal MA segment refers to an MA segment which is generated by one or two boundary faces and contains at least one MA voxel which is not shared by other MA segments. The restriction of a normal MA segment avoids the overlap of different normal MA segments which will affect the quality of MA segments. Based on the analysis above, the MA quality metric is defined as follows.

$$MA\ Quality\ metric = SBF_No / BF_No \quad (1)$$

Fig.3 The single resolution method and multi-resolution method

Here, SBF_No refers to the number of MA-sensitive boundary faces while BF_No refers to all boundary faces. The value of MA quality metric, which is between 0 and 1, indicates the MA quality under current MA level. To satisfy the requirement, the quality metric of the generated MA has to be larger than the

required value. In this study, the required metric value is set to 0.95. Increasing the required metric value will obtain MA of higher levels but the computation cost will also increase.

4.1.2 Determination of re-voxelization region

The need of MA level upgrade is only caused by MA-insensitive boundary faces which affect the MA quality. Therefore, only basic voxels that are related to MA-insensitive boundary faces need to be refined in voxelization. Based on the analysis above, the re-voxelization region is defined as follows.

$$Re\text{-voxelization region} = \bigcup_{i=1}^{ISBF_No} voxel_set_i \quad (2)$$

In equation (2), *ISBF_No* refers to the number of MA-insensitive boundary faces while *voxel_set_i* refers to the basic voxels which intersect with the bounding box of the *i*-th MA-insensitive boundary face. As the level of smallest voxels is equal to the current MA level, the refining level of basic voxels in re-voxelization region needs to be upgraded to the current MA level.

As shown in Fig.4a, a small boundary face *F* is marked in red. *L₀* voxelization is shown in Fig.4b. Here, purple voxels are boundary voxels while pink voxels are body voxels. This color convention is used all through this paper. As shown in Fig.4c, according to equation (2), the basic voxel *A* that intersects with the bounding box of face *F* is the re-voxelization region during *L₁* voxelization since boundary face *F* is the only MA-insensitive boundary face. Therefore, the refining level of basic voxel *A* is upgraded to 1. Similarly, in *L₂* voxelization, basic voxel *A* is in re-voxelization region and its refining level is upgraded to 2.

4.1.3 Smoothness of basic voxels in re-voxelization region

During the upgrade of MA level, sudden change may appear between basic voxels. After the refining levels of basic voxels are determined in the re-voxelization region, two basic voxels whose differences of refining levels are 2 may connect. The connected voxels whose differences of levels are 2 will

induce unsmooth connection between voxels. To avoid this problem, the refining levels of basic voxels in the re-voxelization region need to be smoothed.

Specifically, for any basic voxel whose refining level is *R* in the re-voxelization region, if the refining levels of its face-adjacent basic voxels are smaller than *R*-1, upgrade them to *R*-1. As shown in Fig.4d, in *L₂* voxelization, the levels of basic voxel *B*, *C* and *D* are upgraded from 0 to 1 since the refining level of basic voxel *A* in re-voxelization region is 2.

Fig. 4 *L₀/L₁/L₂* re-voxelization region of a 2D model *M*

4.2 Multi-resolution voxelization in the re-voxelization region

Based on the refining levels of voxels in the re-voxelization region, multi-resolution voxelization will be conducted. For a *R_i* basic voxel *A* in re-voxelization region, 8^i child *L_i* voxels will be generated. These 8^i *L_i* voxels may be outside voxels, boundary voxels or body voxels. For those boundary voxels, their boundary faces also need to be determined.

For a model with a large number of boundary faces, using ray-tracing method [28] directly in basic voxel *A* to find boundary

Fig. 5 Illustration of MA generation

voxels by all boundary faces is too expensive. In fact, only the boundary faces whose bounding boxes are intersected with basic voxel A may be possible to be the boundary faces of these $8^i L_i$ voxels. Therefore, only these boundary faces need to be involved in the ray-tracing method for basic voxel A with edge size of E_i .

After the re-voxelization in re-voxelization region is finished, the voxels outside re-voxelization region can be reused. As shown in Fig.4d, other voxels are reused except basic voxel A , B , C and D in the re-voxelization region. Compared with the entire model, the re-voxelization region is usually small and thus the computational complexity of multi-resolution voxelization is usually quite low.

5. Computation of Multi-resolution Voxel Distances

After the voxelization process is completed, the affected region needs to be computed first. Initially during the process of L_0 MA generation, the affected region is the entire model. The process of MA generation in which the affected region is computed incre-

mentally is illustrated in Fig.5. As shown in Fig.5a, all boundary voxels in purple and body voxels in red are in the affected region during the generation of L_0 MA.

5.1 Computation of the initial affected region

After the multi-resolution voxelization process, some old boundary voxels vanished and some new boundary voxels are generated. As shown in Fig.5b, some L_1 red boundary voxels are newly generated on the top during L_1 voxelization. In the authors' previous work [7], it was proved that the boundary change by Boolean union and Boolean subtraction, only affects the voxel distances and MA in the affected region as follows:

- (1) After a Boolean union operation, the affected region contains the voxels whose touch voxels are vanished boundary voxels or the dual boundary voxels of vanished boundary voxels.
- (2) After a Boolean subtraction operation, the affected region is incrementally decided by the distance dilation of new boundary voxels.

The boundary change during the multi-resolution voxelization process consists of Boolean union operations and Boolean subtraction operations. Therefore, the final affected region consists of the two parts above. Here, the first part is called *initial affected region* and can be computed easily. Then the second part is incrementally decided to obtain the final affected region. As shown in Fig.5c, the touch voxels of the red voxels are new boundary voxels or the dual boundary voxels of vanished boundary voxels. Therefore, they are affected voxels. The incremental computation of the final affected region will be described in the next section.

5.2 Computation of the final affected region based on distance dilation

Fig. 6 Neighbor voxels in different cases

To incrementally compute the affected region, the distance dilation [7] is adapted here. Since the old distance dilation can only be used for single-resolution voxels, it needs to be improved for multi-resolution voxels.

5.2.1 Traditional distance dilation for multi-resolution voxels

Suppose the touch voxel and distance of voxel A is known. The process using the distance information of voxel A to compute that of its neighbor voxel B is called the distance dilation from voxel A to voxel B [7]. Here, the two voxels that are face-adjacent, edge-adjacent or point-adjacent are called neighbor voxels of each other. The process of distance dilation is described as follows

- (1) If the touch voxel C of voxel A can provide the smallest distance for voxel B , voxel C is the new touch voxel of voxel B by replacing the old one.
- (2) If the touch voxel C of voxel A can provide the same smallest distance for voxel B , voxel C is another touch voxel of voxel B .

Generally, a voxel has 26 neighbor voxels. However, this number may change slightly for the connected voxels of different levels. As shown in Fig.6a, if the level of its

face-/edge-/point-adjacent neighbor voxel is equal or smaller than itself for the red voxel, there is only one face-/edge-/point-adjacent neighbor voxel on each face/edge/point of the red voxel, which is marked in yellow/purple/brown respectively. As shown in Fig.6b, if the level of its face-/edge-/point-adjacent neighbor voxel is l times larger than itself, there may be 4^l face-/2 l edge-/1 point-adjacent neighbor voxels on each face/edge/point of the red voxel, which are marked in yellow/purple/brown respectively.

5.2.2 Adaptive distance dilation for refining MA voxels

During the traditional distance dilation from voxel A of high level to voxel B of low level, there may be a problem as follows: The levels of generated MA voxels will be lower than that of the boundary voxels. To solve this problem, an *adaptive distance dilation method* is proposed as follows (Suppose the dilation is from voxel A to its neighbor voxel B):

(1) If the level of voxel A is higher than voxel B , the refining level of the basic voxel which contains voxel B need to be upgraded to that of the basic voxel which contains voxel A . After that, traditional distance dilation is conducted from voxel A to its connected child voxels of voxel B .

(2) If the level of voxel A is equal or smaller than voxel B , traditional distance dilation is conducted from voxel A to voxel B .

As shown in Fig.6c, the adaptive distance dilation will be conducted from the L_1 red voxel to its L_0 yellow face-adjacent voxel and its L_0 purple edge-adjacent voxel respectively. However, their levels are lower than that of the red voxel. After the refining levels of the basic voxels that contain the yellow voxel and the purple voxel are upgraded to 1, the red voxel, purple voxel and yellow voxel are all of the same level and the distance dilations from the red voxel to the yellow voxel and purple voxel are conducted respectively as shown in

Fig.6d. In the same time, the new cyan voxels which don't contain distance information are also generated.

5.2.3 Adaptive double-queue distance dilation based algorithm for the computation of multi-resolution voxel distances

The double queues distance dilation (DQDD) based algorithm is used to compute voxel distances by distance dilation and was introduced in the authors' previous work [7].

The key data structures of this algorithm are two queues, which are named *current queue* and *next queue*. However, this algorithm is only used for computing single-resolution voxel distances and adaptive distance dilation is not involved. Therefore, the MA level cannot be upgraded to increase the MA quality. To solve this problem, by using adaptive distance dilation, the adaptive DQDD based algorithm is proposed for computing the distances of multi-resolution voxels as follows.

Fig. 7 The steps of DQDD-based algorithm

- (1) The initialization of double queues. The iteration round is set to 0. The affected boundary voxels are put into the current queue while the next queue is set to NULL.
- (2) Take a voxel named voxel A from the current queue. If its distance is in the current range, conduct the adaptive distance dilation process from it to each of its neighbor voxels. After the distances of all its neighbor voxels are updated, put it into the next queue. Otherwise, if its distance is not in the current range, put it into the next queue.

- (3) If the next queue is not empty, it becomes the new current queue and the new next queue is set to NULL. And then update the iteration round and turn to Step 2. Otherwise, stop the algorithm.

To illustrate the proposed adaptive DQDD-based algorithm, a model is used as shown in Fig.7. Here, the current queue is marked in blue while next queue is marked in green. And E_0 is set to 2. L_0 and L_1 voxelization is shown in Fig.7a and Fig.7b respectively. Therefore, the affected boundary voxels are put into the current queue (they are set to blue) as shown in Fig.7b in the 0th iteration. As

shown in Fig.7c, the next queue is generated by the current queue. After the first iteration is conducted, the next queue becomes the new current queue as shown in Fig.7d. Here, the adaptive distance dilations are conducted since the distance dilations from voxels of high level to those of low level exist. The refining levels of the corresponding basic voxels are upgraded as shown in Fig.7e and the new next queue is generated as shown in Fig.7f. The algorithm continues until the next queue is empty.

5.2.4 Incremental computation of the final affected region

The final affected region is computed incrementally by adaptive DQDD-based algorithm from affected boundary voxels in the entire model. As shown in Fig.5d, after the first round of adaptive DQDD-based algorithm is conducted, the voxels in final affected region are marked in red and are of the current MA level. Therefore, the MA voxels in the affected region will be of the current MA level.

However, there may be still a problem after the affected region is computed. Some voxels will lose their distance information during the adaptive distance dilation process. As shown in Fig.5d, the newly generated L_1 cyan voxel doesn't contain any distance information after adaptive distance dilation. Some of these new voxels may belong to the affected region while the others may not. To compute distances for those unaffected voxels, using the affected voxels only is not enough and their neighbor unaffected voxels are needed in the second round of adaptive DQDD-based algorithm. These neighbor unaffected voxels are called seed voxels and marked in yellow as shown in Fig.5e.

5.3 Distance computation of multi-resolution voxels

After the adaptive DQDD-based algorithm is conducted twice, the distances of

multi-resolution voxels are computed as follows.

- (1) Adaptive DQDD-based algorithm from the affected boundary voxels in the entire model. After the initial affected region is computed, this algorithm is used to compute the incremental affected region and the final affected region are obtained.
- (2) Adaptive DQDD-based algorithm from the affected boundary voxels and seed voxels in the affected region. After the affected region is computed, the seed voxels are needed in another adaptive DQDD-based algorithm. They are put into the initial current queue. This algorithm is used to compute voxel distances in the affected region.

6. Generation of Multi-resolution MA Voxels

After the voxel distances in affected region is computed, the MA in the affected region can be computed based on them. An MA voxel needs two touch points since an MA point has at least two nearest points on the boundary according to its definition. However, this condition may not be enough since the distance errors exist in a discretized model.

Fig. 8. General process of MA voxel determination

The *dilation region* of a boundary face is the region that contains only those voxels whose touch voxels are on the face [7]. The boundary of two dilation regions has same distance to two boundary faces. Therefore, the voxels lying on it can be considered as MA voxels. As shown in Fig.8a, the red boundary of the dilation regions of boundary faces 1-4 is

considered as the MA and the voxel on the MA are considered as MA voxels as shown in Fig.8b. For example, MA₁₂ is generated by the boundary of the dilation regions of boundary faces 1 and 2 as shown in Fig.8b. A special condition is that the boundary of different dilation regions contains two layers of voxels. As shown in Fig.8a, the boundary of dilation regions of boundary faces 2 and 4 is two layers of voxels which are marked in yellow. This problem is called double-layer problem. To solve it, one layer of voxels is selected as MA voxels from the two layers which is marked in red as shown in Fig.8b.

```
// Algorithm input: model M
// Algorithm output: Hierarchical MA by multi-resolution voxels
Set MA level to 0
Voxelization
Adaptive DQDD-based algorithm from all boundary voxels in the entire model // Distance computation
MA computation
Compute the MA quality metric
while (MA quality metric < required metric )
    Upgrade the MA level by 1
    Determination of re-voxelization region
    Multi-resolution voxelization in re-voxelization region
    Computation of the initial affected region
    Adaptive DQDD-based algorithm from affected boundary voxels in the entire model // incremental computation of the final affected region
    Adaptive DQDD-based algorithm from affected boundary voxels and seed voxels in the affected region // Distance computation in the affected region
    MA computation in the affected region and combine it with the reused MA
    Compute the MA quality metric
end while
Store hierarchical multi-resolution MA
```

Fig. 9 Pseudo code of the proposed hierarchical multi-resolution method

After the MA in the affected region and the reused MA are combined, the final MA under the current MA level is obtained. As shown in Fig.5f, the resultant MA voxels in brown are the combination of MA voxels in the red affected region and the reused MA voxels outside the affected region. Then, the MA quality is evaluated as introduced in subsection 4.1.1. If the MA quality doesn't satisfy

the requirement, the MA of higher level is computed to improve its quality. The pseudo-code of the proposed hierarchical multi-resolution method is given in Fig.9.

7. Analysis of Accuracy, Complexity and Robustness

To prove the efficiency and effectiveness of the proposed method, the accuracy is analyzed first. After that, the time complexity and cost complexity are discussed. Finally, the effectiveness of the proposed method is proved in further by analyzing robustness.

7.1 Accuracy analysis

For single-resolution MA generation method with a resolution of $No_x * No_y * No_z$, the MA accuracy is $\sqrt{3}/2 * E_0$ [7]. Similarly, it can be concluded that for hierarchical multi-resolution MA generation method with a basic resolution of $No_x * No_y * No_z$, the accuracy of L_i MA is $\sqrt{3}/2 * E_i$. In addition, for the same model, only when the resolution of single-resolution method is $2^i No_x * 2^i No_y * 2^i No_z$, its MA accuracy can be equal to that of L_i MA of hierarchical multi-resolution method whose resolution is $No_x * No_y * No_z$. On another hand, for these two conditions, the resolutions of MA voxels of details are both $2^i No_x * 2^i No_y * 2^i No_z$, so their MA qualities are also the same.

Based on the analysis above, it can be seen that for the same requirement of MA quality and MA accuracy, the basic resolution of hierarchical multi-resolution method can be one 2^i -th that of the single-resolution method. The next subsection will compare their differences of computational complexities.

7.2 Complexity analysis

In this subsection, the space complexities and time complexities of single-resolution method and hierarchical multi-resolution method are analyzed and compared.

7.2.1 Space Complexity

The major space cost for MA generation comes from the voxels. For the single-resolution method with a resolution of $No_x*No_y*No_z$, the space cost is as follows:

$$M_{\text{single}} = O(No_x*No_y*No_z) \quad (3)$$

Similarly, for the hierarchical multi-resolution method with a resolution of $No_x*No_y*No_z$ for the MA level of i , the memory cost is as follows:

$$M_{\text{multi}} = O(\alpha * 8^i * No_x*No_y*No_z) \quad (4)$$

Here, $\alpha(0<\alpha<1)$ refers to the ratios of basic voxels in the affected region to all basic voxels. Because the refining level of basic voxels in the affected region is i , 8^i new child voxels are needed for each of them.

The single-resolution method with a resolution of $2^i*No_x*2^i No_y*2^i No_z$ and the hierarchical multi-resolution method for the MA level of i with a basic resolution of $No_x*No_y*No_z$ have the same MA accuracy and MA quality. However, it can be seen from equations (3) and (4), the ratio of their memory cost is α which is far smaller than 1. Therefore, the space complexity of the proposed hierarchical multi-resolution method is far smaller than that of the single-resolution method.

7.2.2 Time complexity

As mentioned above, the main steps of MA generation are voxelization, computation of voxel distances and the final MA generation. Suppose the time cost of these three steps are T_v , T_d and T_g respectively for each voxel. For the single-resolution method with a resolution of $No_x*No_y*No_z$, the time cost is

$$T_s = No_x*No_y*No_z * (T_v + T_d + T_g) \quad (5)$$

Similarly, for hierarchical multi-resolution method with a resolution of $No_x*No_y*No_z$ for the MA level of i , the memory cost is

$$T_m = 8^i * No_x*No_y*No_z * (\beta * T_v + \alpha * T_d + \alpha * T_g) \quad (6)$$

Here, $\beta(0<\beta<1)$ refers to the basic voxels in re-voxelization region.

The single-resolution method with a resolution of $2^i*No_x*2^i No_y*2^i No_z$ and the hierarchical multi-resolution method for the MA level of i with a basic resolution of $No_x*No_y*No_z$ have the same MA accuracy and MA quality. It can be seen from equation (5) and (6) that the ratio of their time cost is between α and β which is far smaller than 1, which proves that the time complexity of the proposed hierarchical multi-resolution method is far smaller than that of the single-resolution method.

7.3 Robustness and sensitivity analysis

Generally, the robustness of MA and the sensitivity of MA are mutually exclusive. When the MA robustness is strong, the MA sensitivity will be poor, and vice versa. If small faces are regarded as noise faces for the consideration of MA robustness, their impact on the MA needs to be eliminated. Otherwise, if they are regarded as normal faces for the consideration of MA sensitivity, their impacts on the MA need to be computed. In the proposed hierarchical multi-resolution MA, its robustness is guaranteed since small faces cannot be recognized by the large voxels. Meanwhile, the MA sensitivity can also be obtained since the small faces can be recognized by the small voxels for high MA level. By computing the MA of different levels, both MA robustness and MA sensitivity are achieved.

8. Implementation and Experiments

The proposed method is implemented with visual studio 2008 and the geometric modeling kernel ACIS 17.0. To demonstrate the efficacy and efficiency of the proposed method, three groups of experiments are conducted as follows. The detailed steps of the proposed method are illustrated in the first group in which the MA level is upgraded from 0 to 2 gradually. In the second group, the proposed hierarchical multi-resolution method and the single-resolution method are conducted for two models respectively. For each model, the computation time and the number of involved voxels are listed to compare the

differences of time and space cost between the two methods. The single-resolution method is used for comparison because it is a method which is commonly used. In the third group, the comparison between the proposed method and the previous method is given by a model.

8.1 Illustration of the whole method

To illustrate the detailed steps of the proposed method, a typical model as shown in Fig.10 is used. The color convention for this section is also given in Fig.10.

Fig. 10 The detail steps of the proposed method

The original model is shown in Fig.10a. It contains a small concave part which is marked by a red circle. The results of L_0 voxelization and L_0 MA generation are shown in Fig.10b and Fig.10c respectively. Since the MA level is 0, the affected region is the entire

model. After the MA quality is computed, L_1 re-voxelization region is determined. New boundary voxels in re-voxelization region are L_1 voxels in dark green while voxels outside re-voxelization region can be reused. Then, the computed affected region is marked in

blue as shown in Fig.10e. In Fig.10f, the seed voxels around affected region are marked in yellow. Finally, the generated MA in the affected region and the reused MA in other region are combined into L_1 MA, as shown in Fig.10g. The process of computing L_2 MA is similar with that of computing L_1 MA as shown in Fig.10h-k. Since the quality metric of L_2 MA is 1, L_3 MA needn't be computed and the method ends.

Note that L_0 MA doesn't contain MA generated by boundary faces of small concave part since the boundary voxels of MA level 0 are large. On the contrary, L_1 MA and L_2 MA contain the MA generated by boundary faces of small concave part since the boundary voxels of L_1 MA and L_2 MA are small. It can be seen clearly that the MA of low level can suppress noises while the MA of high level can contain details.

Fig. 11 The voxelization and MA generation results in level 0-2 for Model 1

8.2 Performance demonstration of the proposed method

To show the efficiency and effectiveness

of the proposed hierarchical multi-resolution method, some models with details are used here to compare the differences of the hierarchical multi-resolution method and the sin-

gle-resolution method on efficiency and effectiveness. Two test models (Model 1, 2) are shown in Fig.11a and Fig.12a respectively, and their details are marked by red circles. The details of Model 1, 2 are two small concave parts on each model respectively. For Model 1, L_1 MA and L_2 MA generated by the hierarchical multi-resolution method with a resolution of 40 is compared with the MA generated by the single-resolution method with a resolution of 80 and 160 respectively since they have the same MA quality and MA accuracy. The $L_0/L_1/L_2$ voxelization and MA for Model 1 by using the hierarchical mul-

ti-resolution method are shown in Fig.11a/11c/11e and Fig.11b/11d/11f respectively. The voxelization and MA of Model 1 by using the single-resolution method are shown in 11g/11i and 11h/11j respectively.

Here Model 1 is only a CAD model bounded by parametric surfaces. To further demonstrate the robustness of the proposed method, Model 2 bounded by freeform surfaces is conducted by the proposed method with a resolution of 50. Fig.12 shows the similar results for Model 2.

Fig. 12 The voxelization and MA generation results in level 0-2 for Model 2

The time cost and involved voxel numbers of the two models in each step are shown in Table 1. It can be seen from Table 1 that, for Model 1, L_1/L_2 MA by using the hierarchical multi-resolution method with a basic resolution of 40 and the MA by using the sin-

gle-resolution method with a resolution of 80/160 have the same MA quality. However, the time and space cost of the former one is much lower. The detail ratios of time cost and involved voxel numbers between the two methods are shown in Table 2, and similar

conclusions can be deduced. As shown in Table 2, Model 2 has the similar results with Model 1.

It can be concluded from the above experiments that the proposed hierarchical multi-resolution method can generate hierarchical

multi-resolution MA with much lower time and space complexity than that of the single-resolution method without reducing the MA quality, which proves the efficiency and effectiveness of the proposed method.

Table 1 The execution time and involved voxels of hierarchical multi-resolution method and single resolution method for Model 1-2

Model ID	Mode	Basic Res.	Level	Voxelized Voxel No	Dilated voxels No.	V- time	D-time	Total time	BF No.	SBF No.	MA quality
1	Single	40	0	7332	7332	981	86	1067	43	34	0.791
	Multi	40	1	120	823	266	90	356	43	36	0.837
	Single	80	0	50875	50875	3476	598	4074	43	36	0.837
	Multi	40	2	8	1584	478	1497	1975	43	40	0.93
	Single	160	0	382784	382784	13589	4783	18372	43	40	0.93
2	Single	50	0	8487	8487	6284	93	6377	56	46	0.821
	Multi	50	1	261	274	1699	9	1708	56	53	0.946
	Single	100	0	58864	58864	24936	752	25688	56	53	0.946
	Multi	50	2	10	73	6142	69	6211	56	56	1.000
	Single	200	0	436580	436580	102195	5467	107662	56	56	1.000

Note: V-time—Voxelization time; D-time—Dilation time; BF No.—Boundary Face number; SBF No.—MA Sensitive Boundary Face number

Table 2 The comparison between hierarchical multi-resolution method and single resolution method for Model 1-2

Model ID	MA Level	Voxelized Voxel No	Dilated voxels No.	Voxelization time	Dilation time	Total time	MA quality
1	1	0.2%	1.6%	7.6%	15.1%	8.7%	100%
	2	0.1%	0.4%	3.5%	31.3%	10.8%	100%
2	1	0.4%	0.5%	6.8%	1.2%	6.6%	100%
	2	0.1%	0.1%	6.0%	1.3%	5.8%	100%

Table 3 The comparison between distance-dilation method and the proposed method in different levels

MA Level	Dilation unit threshold of the distance-dilation method	Dilation unit threshold of the proposed method	Voxel edge size of the proposed method	Voxel edge size of the proposed method
0	4% the total area	User defined	Normal voxel edge size	Normal voxel edge size
1	2% the total area	User defined	Normal voxel edge size	1/2 normal size in detail part automatically
2	1% the total area	User defined	Normal voxel edge size	1/4 normal size in detail part automatically
3	--	User defined	--	1/8 normal size in detail part automatically

8.3 Comparison with the previous method

The proposed method is inspired by the distance-dilation method which was proposed by Gao *et al* [6]. In their method, the resultant MA of a CAD model can be generated under three MA levels. In each level, small boundary surfaces under a certain threshold (4%/2%/1%) are merged into a virtual surface called dilation unit. However, the voxel edge size of their method is fixed. To generate MA of multi-size voxels, the distance dilation method was improved as the double queues distance dilation (DQDD) based algorithm [7], and is further improved as the adaptive DQDD based algorithm of the proposed method in this paper. Comparing with the distance dilation method [6], the proposed method can generate MAs of all levels. For each MA level, the voxel edge size is half the voxel edge size of its previous MA level in detail part rather than the fixed size. What's more, the dilation unit threshold

of each level is user defined rather than a fixed value. The detail comparison between these two methods is shown in Table 3.

To further compare these two methods by examples, Model 3 is illustrated in Fig. 13. For the distance dilation method, since the detail part is too small to be a dilation unit (far smaller than the threshold 4%/2%/1%) and the voxel edge size is fixed, the MAs of three levels are similar. For the proposed method, the detail part can be automatically searched and its MA can also be represented by smaller voxels automatically.

The comparison on the execution time and the MA quality for the two methods are given in Fig.14 and Fig.15 respectively. Obviously, under higher levels, the proposed method can produce the same MA quality by a much smaller cost than the distance dilation method.

Fig. 13 The MA in level 0-2 for Model 3 under the distance dilation method and the proposed method

Fig. 14 The execution time of the two methods for Model 3 under different MA levels and resolutions (unit:ms, R refers to the basic resolution)

Fig. 15 The MA quality of the two methods for Model 3 under MA different levels and resolutions (R refers to the basic resolution)

9. Conclusions

In this study, a method is proposed to generate the hierarchical multi-resolution MA for CAD models. The contributions of this paper are

- (1) The hierarchical MA and multi-resolution voxels are proposed to compute MA with different qualities. In hierarchical MA, the MA robustness and MA sensitivity are both

considered. For each MA level, voxels of different resolutions are used.

- (2) A reasonable method is proposed to evaluate the MA quality. By evaluating the impacts of all boundary faces on the normal MA segments, MA quality is computed. If the MA quality doesn't satisfy the requirement, the MA level can be automatically upgraded to obtain the MA with bet-

ter quality iteratively.

- (3) The re-voxelization region is proposed to refine the model locally when MA level upgrades. The affected region is proposed to simplify the computation of its voxel distances and MA generation. Therefore, the information outside the affected region can be reused to improve the efficiency greatly. Meanwhile, the re-voxelization region is proposed to simply the voxelization by reusing the region outside the affected region.
- (4) The adaptive distance dilation is proposed which can refine the basic voxels in distance dilation automatically. By using adaptive distance dilation, the adaptive DQDD-based algorithm is proposed to compute the affected region and multi-resolution voxel distances. The refined MA by multi-resolution MA voxels with higher MA quality is generated.

However, this method has some shortcomings. First, the efficiency should be improved when there are many detailed parts in a model. Second, this method should be extended to mesh models.

Acknowledgements

The authors appreciate the financial supports from the NSF of China (61572427 and 61370182), National Science and technology support Program (2014BAD04B02) and Key Project of Science and Technology of Zhejiang Province (2014C01052).

Reference

1. Blum H. A transformation for extracting new descriptors of shape. *Models for the Perception of Speech and Visual Form*. Cambridge: MIT Press, 1967: 362–380.
2. http://en.wikipedia.org/wiki/Medial_axis
3. Dorado R. Medial axis of a planar region by offset self-intersections. *Computer-Aided Design*, 2009, 41(12): 1050-1059.
4. Chin F, Snoeyink J, Wang CA. Finding the medial axis of a simple polygon in linear time. *Discrete and Computational Geometry*, 1999, 21(3): 405-420.
5. Nackman LR. Curvature relations in three-dimensional symmetric axes. *Computer Graphics and Image Processing*, 1982, 20(1): 43-57.
6. Ma J, Bae SW, Choi S. 3D medial axis point approximation using nearest neighbors and the normal field. *Visual Computer*, 2012, 28(1):7–19.
7. Zhu HS, Liu YS, *et al.* Constructive generation of the medial axis for solid models. *Computer-Aided Design*, 2015, 62:98-111.
8. Gao W, Gao SM, Liu YS, *et al.* Multiresolutional similarity assessment and retrieval of solid models based on DBMS. *Computer-Aided Design*, 2006,38(9): 985-1001.
9. Lakshmi J K, Punithavalli M. A survey on skeletons in digital image processing. // *Proceedings of International Conference on Digital Image Processing*. Los Alamitos: IEEE Computer Society Press, 2009:260-269.
10. Lam L, Lee SW, Suen CY. Thinning methodologies – a comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, September 1992, 14(9):869-885.
11. Scott GL, Turner SC, Zisserman A. Using a mixed wave diffusion process to elicit the symmetry set. *Image and Vision Computing*, 1989, 7(1): 63-70.
12. Siddiqi K, Bouix S, Tannenbaum A, *et al.* The Hamilton-Jacobi skeleton. // *Proceedings of International Conference on Computer Vision*. Los Alamitos: IEEE Computer Society Press, 1999: 828-834.
13. Vleugels JM, Overmars MH. Approximating generalized Voronoi diagrams in any dimension. Technical Report UU-CS-95-14, Department of computer science, Utrecht University, 1995.
14. Borgefors G, Nyström I, *et al.* Computing skeletons in three dimensions. *Pattern Recognition*, 1999, 32(7): 1225-1236.
15. Borgefors G. On digital distance transforms in three dimensions. *Computer Vision and Image Understanding*, 1996, 64(3):368-376.
16. Sherbrooke EC, Patrikalakis NM, Brisson E. Computation of MA transform of 3-D polyhedral. // *Proceedings of ACM Solid Modeling*. New York: ACM Press, 1995: 187-199.
17. Aichholzer O, Aigner W, Aurenhammer F, *et al.* Medial axis computation for planar free-form shapes. *Computer-Aided Design*, 2009, 41(5): 339-349.
18. Meijster A, Roerdin JBTM, *et al.* A general algorithm for computing distance transforms in linear time. *Mathematical Morphology and its Applications to Image and Signal Processing*. Springer US Press, 2000: 331-340.
19. Hirata T. A unified linear-time algorithm for computing distance map. *Information Processing Letter*, 1996, 58(3):129-133.
20. Miklos B, Giesen J, Pauly M. Discrete scale axis representations for 3D geometry. // *Proceedings of ACM SIGGRAPH*. New York: ACM Press, 2010: 394–493.
21. Reniers D, Wijk JJ, Telea A. Computing multiscale curve and surface skeletons of genus 0 shapes using a global importance measure. *IEEE Transaction Visual Computer Graph*, 2008,

- 14(2): 355-368.
22. Cao TT, Tang K, Mohamed A, *et al.* Parallel Banding Algorithm to Compute Exact Distance Transform with the GPU. Proc. of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games I3D, 2010, 1(2): 83-89.
23. Rong GD, Tan TS. Jump flooding in GPU with applications to Voronoi diagram and distance transform. Proc. ACM Symposium on Interactive 3D Graphics and Games (I3D), 2006, 109-116.
24. Sud A, Govindaraju N, Gayle R, Manocha D. Interactive 3D distance field computation using linear factorization. Proc. ACM Symposium on Interactive 3D Graphics and Games (I3D), 2006, 117-124.
25. Takashi M, Suzuki H. Sparse grid distance transforms. Graphical models, 2010, 72(4): 35-45.
26. David C, Montanvert A. Optimal separable algorithms to compute the reverse Euclidean distance transformation and discrete medial axis in arbitrary dimension. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29(3): 437-48.
27. Zhu HS, Liu YS, *et al.* Calculating the medial axis of a CAD model by multi-CPU based parallel computation. Advances in Engineering Software, 2015, 85: 96-107.
28. Glassner AS. An introduction to ray tracing. Academic press Ltd., London, UK, 1989.