Engineering with Computers
DOI 10.1007/s00366-012-0256-z

ORIGINAL ARTICLE

A local adaptation-based generation method of medial axis for efficient engineering analysis

Yusheng Liu · Chuhua Xian · Ming Li · Haibin Xia · Shuming Gao

Received: 19 August 2011 / Accepted: 5 January 2012

Abstract Currently engineering analysis is regarded as an integrated part of design process and medial axis (MA) is often utilized. However, the generation of MA of complicated models is computation intensive since it is always generated from scratch even if a tiny modification is imposed. A novel local adaptation-based approach to generating the MA for efficient engineering analysis is proposed in this study. With this method, the MA of a resultant model constructed from two other models via a Boolean operation or parameter modification is generated by adapting the MAs of the operand models in a certain way, instead of regenerating the MA from scratch. First, several new properties of the MA which are the fundamental basis of the proposed method are investigated. Then, the boundaries that will vanish from or be added into the resultant model during the Boolean operation or parameter modification are found, and the region in which the MA segments (MASs) need to be regenerated is determined. Finally, the new MASs are generated for the region using an improved tracing method. The final MA of the resultant model is thus constructed by combining the newly generated MASs with the reserved MASs of the operated model(s). Some examples are given to illustrate the high computational efficiency of the proposed method for engineering analysis.

Keywords Medial axis · Medial axis transformation · Boolean operation · Local adaptation · Engineering analysis

Y. Liu (✉) · C. Xian · M. Li · H. Xia · S. Gao
State Key Laboratory of CAD & CG, Zhejiang University,
Hangzhou 310027, People's Republic of China
e-mail: ysliu@cad.zju.edu.cn

1 Introduction

With the rapid increase of computational ability, finite element analysis (FEA) that is one of the most important methods for engineering analysis is widely used in the design process of complex product to help find the design problems as early as possible. A lot of related commercial tools are developed to support FEA work. To speed up the engineering analysis using FEA, the medial axis (MA) [1], which can realize dimension reduction, is a primary approach for the simplification of the models. It is the locus of centers of maximal disks/balls contained within the boundary of a 2D/3D region. It possesses good properties, as Ramanathan and Gurumoorthy [2] have shown. For example, it simplifies the regional representation, while retaining the original information, as Patrikalakis and Maekawa [3] demonstrated.

There are a lot of research efforts for constructing the MA efficiently [4–9]. However, the computational efficiency of the MA is still not very satisfactory when it is used in engineering analysis. The reason is that thousands of iterations of MA generation process are needed when the engineering analysis is conducted to refine and optimize the design of complicated models.

One common characteristic of almost all the existing approaches is that the model is regarded as a whole part. In fact, complicated models can be regarded as an aggregation of simple modeling primitives combined by Boolean operations. Especially for the models before and after each iteration during FEA, there are only tiny modifications. Therefore, the MA of the model after FEA can be obtained efficiently and robustly by reusing the MA of the model before FEA.

Based on the above analysis, a novel local adaptation-based MA generation method is proposed for complicated

Published online: 20 January 2012

models when MA-based engineering analysis needs to be conducted for obtaining the optimal design. Here, 2D models whose MAs need to be generated are considered in this study. There are two distinct characteristics for the proposed approach. The first one is that its focus is on the modified part of the boundary when the modification operation is conducted on the models. Generally, the part of modified boundary is much smaller than that of the whole part when a minor modification is conducted on the complicated model. The second is that only the MA of modified boundary needs to be regenerated whereas the other part of MA of the unchanged boundary can be reused. In this way, when the model is modified by performing Boolean operations or parameter modifications during FEA, its MA is just adapted locally rather than regenerated from scratch. Therefore, the computational efficiency can be improved dramatically for complicated models and then speed up the engineering analysis process.

This paper is organized as follows. In Sect. 2, the related work is given. Some concepts, properties of the MA, and the method overview are described in Sect. 3. In Sects. 4 and 5, the method for finding the changed boundary and the region whose MA needs to be regenerated is discussed. The adapted tracing method for generating a new MA is described in Sect. 6. In Sect. 7, the degeneration problem of the MA is analyzed. Section 8 gives the implementation and some discusses. Finally, the conclusion and future work are given in Sect. 9.

2 Related work

Generally, the approaches to generating the MA can be classified into three categories: thinning, tracing, and Voronoi graph-based method.

The thinning method is used to generate the MA of a model by generating that of another, approximate model which can be calculated easily. The calculation precision is controlled by the approximation precision of the model. Nackman [10] put forward a method which substituted a polygon/polyhedron for the smooth boundaries and thus used the MA of the polygon as that of the input model. It was generalized from the method proposed by Bookstein [11], which could generate the linear skeleton. Based on the diffusing process of combined waves, Scott et al. [12] proposed a method to determine the symmetric axis of a model which was the superset of the MA. The method is very effective for 0–1 images, but the calculation error is considerably greater for color images. Siddiqi [13] put forward a vector field-based thinning method. In their method, each voxel was given a vector pointing to the nearest boundary point, and if the average flux for a voxel into its adjacent voxels was positive, the voxel must be on MA. Ragnemalm [14] determined the shortest Euclidean distance to the boundary for each voxel and obtained the MA by calculating the local gradient maxima. Hoff [15] implemented the algorithm with hardware, and the computational efficiency was improved dramatically. In Vleugels and Overmars's approach [16], voxelization was recursively conducted for a space that contained MA voxels, until the necessary resolution was reached. Foskey [17] combined the advantages of the above two methods: i.e. the implementation with both hardware and recursive division was used in their method.

Based on local continuity, the tracing method is widely used. It is a recursive method. Montanari [18] set forward a method to calculate the MA for the deformed multiply connected region. In their method, the key issue was to determine the crucial bifurcation points and offset the boundary inward. The bifurcation points are connected by appropriate linear or parabola segments to generate the whole MA. Lee [19] proposed a method whose computation complexity was $O(n\log n)$ for convex polygons and $O(n^2)$ for concave polygons. For the multiply connected region with h holes, Srinivasan and Nackman [20] put forward a method whose computational complexity was $O(nh + n\log n)$. Similarly, Gursoy and Patrikalakis [21–23] proposed a method for 2D models consisting of multiply connected regions with the boundary being line and arc. Culver [24] proposed a space decomposition-based reliable and efficient method for the MA of polyhedron. However, it is highly sensitive to the choice of polyhedral approximation. For different tessellations of the same shape, the obtained MAs are different. Similarly, Choi [25] proposed a domain decomposition-based method that could effectively reduce the computational efficiency for a complicated domain replaced by a set of simple domains. They used a G^1 cubic spline of Minkowski Pythagorean hodograph to represent the MA for desired degree of accuracy. Choset [26] put forward a progressive generation method of the MA by tracing, which was used for the behavioral layout of the robots with local distance information obtained from sensors. In [27], Chiang put forward a method for segmented C^2 curves in which the traced branches were joined through the polynomial functions. Sherbrooke [28] developed a different algorithm that built the linearly approximate segments of the curve in the direction of tracing, so that the computational complexity was reduced greatly. This method can be used for conveniently calculating the MA of polyhedron. Reddy and Turkiyyah [29] put forward a tracing-based method of MA generation with the help of a Voronoi diagram. Dutta and Hoffmann [30] proposed a CSG-based skeleton method. Instead of tracing the bisectors of the boundary entities, Ramanathan and

Gurumoorthy [2] proposed a method for generating the medial axis transformation (MAT) for planar domains with free-form boundaries by marching along the object boundary. Based on the moving Frenet frames and Cesaro's approach of differential geometry, Cao [31] proposed a tracing algorithm in which the mapping relationship between two boundaries and the MA had been established. The geometrical model of the MAT of the planar domains with curved boundaries has been generated. The advantage of the approach is that it is not sensitive to the topological singularity of the polygon approximation algorithm. Aichholzer [9] provided an efficient and stable MA algorithm for 2D free-form shapes using a piecewise circular boundary conversion. The advantage of their method is that it can guarantee convergence to the exact medial axis of the input shape. Recently, a GPU-based parallel generation of MA is proposed by Cao which is a novel technology for speeding up MA generation [32].

Based on the duality relation between Voronoi diagram and Delaunay triangulation, the MA can also be calculated approximately from the Voronoi diagram. If the sampling density approaches infinity, the Voronoi vertices in this case converge to MA. Lavender [33] generated the Voronoi diagram based on Octree (Quadtree on 2D), which divided the space into Voronoi zones according to a given resolution. For a 2D zone and 3D space, Brandt [34] proposed a continuous method to approximate the skeleton with boundary discretization. Dey [35] set forward a hierarchical and simplified method for generating MA which was scale and density independent. They developed two filters for selecting the correct MA points. Sheehy [36] tried to determine the topological structure of the MA based on the Delaunay triangulation of a region bounded by some points satisfying a certain distribution. Etzion and Rappoport [37] proposed a calculation method for Voronoi diagram based on space division, which allowed local and partial calculation of the Voronoi diagram. To make the algorithm more robust, they separated the symbolic representation from the geometric representation of the Voronoi diagram. Ramamurthy and Farouki [38] proposed an incremental method to compute the Voronoi diagram-based MA. In their method, a single boundary segment was added to an existing boundary-segment set at each step. After the Voronoi diagram of the 2D model was computed, the MA was then obtained by the refinement steps. Joachim et al. [39] proposed a highly efficient method for MA based on sampling the 2D model at first and then computing the MA of the union of inner Voronoi balls for the sampling set. In this algorithm, it was only necessary to compute the Voronoi diagram of the sampling set, and hence it was not only faster but also more robust.

3 Basic concepts and overview of the method

3.1 Basic concepts

To facilitate the description of the proposed local adaption-based MA generation method, the related concepts are given first.

3.1.1 Generative elements of MA

The boundary elements tangent to the maximum inscribed circle during the generation of MA. For a 2D model, each boundary edge and each concave vertex are generative elements of a certain MA. The generative elements of the same MA segment (MAS) are called as conjugates each other.

3.1.2 Local adaptation-based generation of MA

The construction of MA is based on locally adapting and reusing the MA(s) of the operated model(s) when the modification operation is conducted on the model(s). As shown in Fig. 1, given the MAs of models *A* and *B*, the MA of the resultant model, as shown in Fig. 1b, is constructed by combining the MAs of models *A* and *B* in a certain way rather than regenerating it from scratch.

3.1.3 Vanishing generative element (VGE)

The boundary edge segment of the operated model(s) that is to vanish in the resultant model after the Boolean operation or parameter modification. As shown in Fig. 1b, the dashed lines on the boundaries of models *A* and *B* do not appear in the resultant model. They are the VGEs after the Boolean union operation.

3.1.4 Added generative element (AGE)

It is the boundary edge that appears on the resultant model after the Boolean subtraction operation or parameter modification operation.

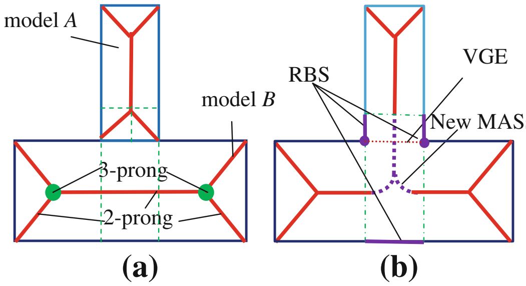


Fig. 1 Illustration of some concepts about local adaption-based MA generation method

3.1.5 Regeneration boundary set (RBS)

In order to correctly construct the MA of the resultant model by the local adaptation-based generation method after the Boolean operation or parameter modification is performed upon the model(s), the MASs of some boundary segments in the resultant model need to be regenerated. Such boundary segment set is called as regeneration boundary set (RBS). As shown in Fig. 1b, the RBS consists of purple thick solid boundary edge segments and concave vertices for the Boolean union operation.

3.1.6 Semi-continuity

Two models Ω_1 and Ω_2 are of semi-continuity if there is an affine transformation T satisfying:

For $\forall\delta > 0$, there must exist $\varepsilon > 0$ satisfying:

$$H(\Omega_1, \Omega_2) < \varepsilon \Rightarrow h(T(\Omega_1, \Omega_2)) < \delta \qquad (1)$$

Here, $H(\Omega_1, \Omega_2)$ is two-sided Hausdorff distance between models Ω_1 and Ω_2 and:

$$H(\Omega_1, \Omega_2) = \max(h(\Omega_1, \Omega_2), h(\Omega_2, \Omega_1)) \qquad (2)$$

Here, $h(\Omega_i, \Omega_j)$ is the one-sided Hausdorff distance from model i to model j.

3.1.7 Region separability

Given any two regions, if the modification that arises in any one region does not affect the other, the two regions are of region separability.

3.2 New properties of MA

Preparatory property 1 The number of N-prong ($N > 2$) MA points is finite [40]. According to the property, the emphasis of this study is focused on how to calculate the 2-prong MA points when regenerating the MA of the RBS. For the N-prong ($N > 2$) MA points, they can be obtained by finding the intersections of MA segments.

Preparatory property 2 Medial axis transformation is region separable [40]. According to the above preparatory property 1, if there are at least two 2-prong MA segments, there must exist at least one 3-prong MA point so that the circle centered at it is tangential with the boundary at more than two boundary points. The set of such circles divides the boundary into different overlapped sub-regions, and for each sub-region, the MAT is independent of the others. As shown in Fig. 2, if region 1, 2, or 3 in Fig. 2a is modified, the region whose MASs need to be modified are regions 1′, 2′ or 3′, respectively, as shown in Fig. 2b.

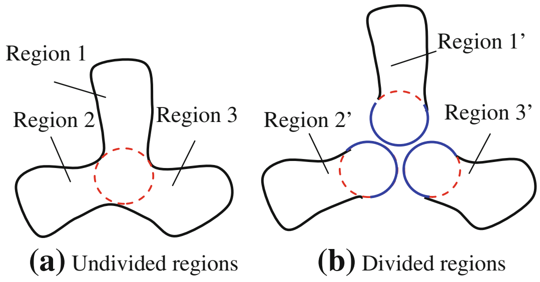


Fig. 2 Region separability of MA

Property 1 Medial axis transformation is semi-continuous.

It can be deduced from preparatory property 2 that the MAT of any given part of the boundary is limited in a finite region. That is to say, for $\forall\delta > 0$, we can always find $\varepsilon > 0$ that satisfies the requirement of the definition of *semi-continuity*. As shown in Fig. 1b, the RBS for Boolean union operation is well defined. The MA of the resultant model is obtained just by subtracting and adding some MASs of the operated model(s).

According to preparatory property 2, the modifications of MASs in one region will not affect those in other regions if these two regions are separable. According to property 1, the RBS of each Boolean operation is limited and well defined. Therefore, the MA of resultant model after a Boolean operation or parameter modification need not be regenerated from scratch. Only the region in which the MA needs to be added or deleted, i.e., RBS, must be identified and then its MA is regenerated.

Lemma 1 *Even if one generative element of a certain MAS is the VGE, its corresponding MAS must be deleted from the existing MA.*

Proof (Taking Boolean union as example) Suppose that models A and B are operated on by a Boolean union operation. Let e be a point on the MA, with p and q as its generative elements. Furthermore, suppose the generative element q is to vanish after the union operation and p will still exist. Obviously, circle $C(e, r) \subseteq A \cup B$ is still true for the unchanged radius r with the point e being the center. The following will prove that there is not any other tangent point except point p between the circle C and the boundary of $A \cup B$ using the method of contradiction.

First suppose that there is such a tangent point denoted as q'. Obviously, it is impossible that q' lies on the pre-operated boundary of model A, since the tangent point has been destroyed and vanished during the Boolean union operation. Similarly, it is impossible that q' lies on the boundary on (B–A), since it will not lie on the boundary of A if q' lies on the boundary of (B–A). Therefore point q'

should not lie on the boundary of model A. However, since $C(e,r) \subseteq A$ and $q' \in C$, thus q' must lie on the boundary of model A. There is a contradiction which is caused by the above assumption that there exists such a tangent point q'.

On the other hand, the radius r cannot be increased anymore, since the circle C is a maximally inscribed circle and tangent with a boundary at p. Therefore, the point e should not be the MA point of $A \cup B$ and must be deleted. The case for Boolean subtraction can be proved similarly.

Furthermore, the circle C will not be tangent with the boundary with the same radius if p and q are both deleted after Boolean operation. Surely a new circle C' can be obtained by increasing the radius r. However, it cannot be guaranteed that the circle C' is tangent with the boundary at two different points simultaneously and thus point e will not be a MA point and should be deleted.

Lemma 2 *If both generative elements of the MA point are not deleted after a Boolean operation, the MA point must still be a MA point of the resultant model after Boolean union operation. However, it may not be a MA point of the resultant model after Boolean subtraction operation.*

Proof The first half part of the lemma is to be proved at first. Suppose that a Boolean union operation is imposed on models A and B, point e is a MA point of pre-operated model A, and the maximum inscribed circle centered at point e with the radius r is tangent with the boundary of model A at points p and q. Since points p and q are still preserved in the resultant model after the Boolean union operation, the circle centered at point e with the radius r is still tangent with the boundary of model A at points p and q. Denote the circle with $C(e,\ r)$, $C(e,r) \subseteq A \subseteq A \cup B$ is still true after Boolean union operation. Therefore, point e is still the MA point of resultant model after Boolean union operation.

The second part of the lemma is explained with the example shown in Fig. 3. Here, the resultant model is obtained by subtracting model B from model A. Although the generative elements of MASs PQ and UV are still kept in the resultant model, they are not the MASs of the resultant model. The reason is that there are new boundary edges added into the resultant model which are nearer to them than their previous conjugates, and thus some new MASs, such as EF and MN, as shown in Fig. 3, have to be generated. The essential reason is that $C(e,r) \subseteq (A-B)$ is not guaranteed to be true for each center point e and radius r, though $C(e,r) \subseteq A$ is true.

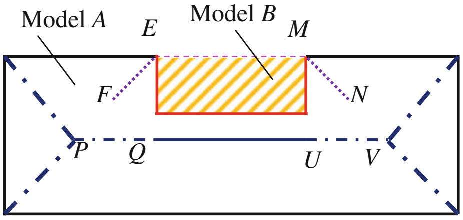


Fig. 3 Illustration of the effect of a Boolean subtraction on MA

Theorem 1 *The following four kinds of boundary elements should belong to the RBS: (1) the AGEs; (2) all the conjugates of AGEs; (3) all the new generated concave vertices; (4) all the conjugates of the VGEs. Furthermore, all the new MASs to be added into the final MA of the resultant model can only be generated based on the RBS.*

Proof It is obvious that the conclusion is correct for the first three parts. Thus, only the fourth part is proved here. Suppose that there are two boundary points $P, Q \in A \cup B(or A - B)$. According to preparatory property 2 and property 1, the following can be deduced:

- If P and Q do not belong to the RBS, there must be corresponding MA points to them, and thus it is impossible for them to be used to generate other MA points.
- If $P \in$ RBS and $Q \notin$ RBS (or vice versa), there must exist MA points corresponding to them, respectively, and thus it is impossible for point Q to be used to generate a MA point with point P.

Based on the above analysis, all the new MASs to be added into the MA of the resultant model can only be generated from generative elements of the RBS.

3.3 Overview of the method

The tracing method is adapted to generate new MASs in this study. This approach differs from existing methods in that only the MA of the RBS is generated, instead of the whole region of the model. Moreover, the possible MAS types are also determined by considering the types of boundary elements. For planar shapes, these are point, straight line, arc and free-form curves. Here the free-form curve is approximated by the combination of straight lines and arcs. The types of combinations of generative elements are point–point, point–line, line–line, point–circle, line–circle and circle–circle. The types of MAS may be line, parabola, ellipse, or hyperbola. Some typical MA types as well as their generative elements are given in Fig. 4 [30].

As the first step, the key issue is to efficiently determine the RBS in order to shorten the computational time of generating MA by local adaptation. After that, the new MASs of the RBS are generated and combined with the reserved MASs of the model(s) to get the MA of the resultant model. The process can be divided into three steps, as shown in Fig. 5: (1) Determine the VGEs and AGEs of the operated model(s) according to the performed operation; (2). Generate the RBS; (3). Generate the final

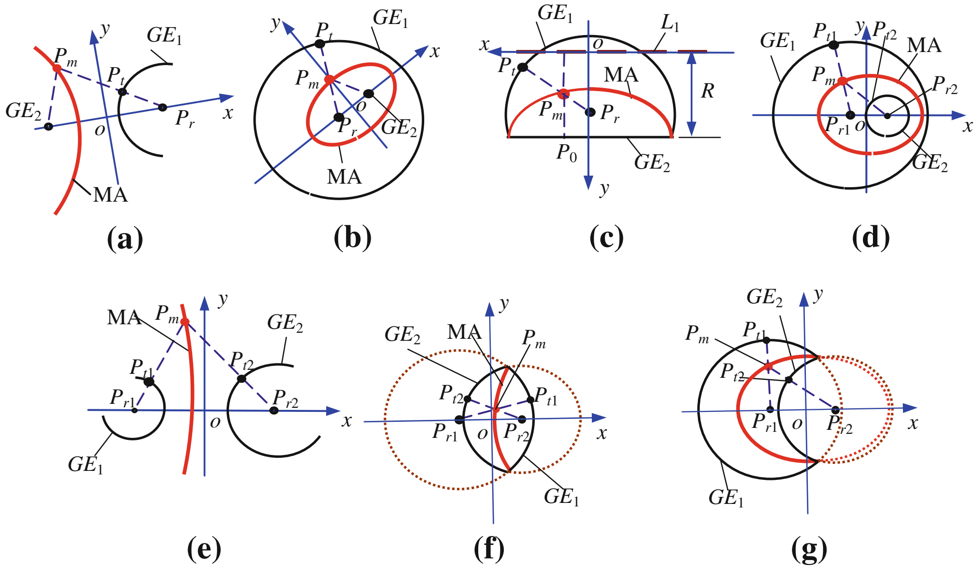


Fig. 4 MA types corresponding to the complicated type combinations. **a** The GEs are the arc and the point, and the point lies out of the circle of the arc; **b** The GEs are the circle and the point, and the point lies in the circle; **c** The GEs are the arc and the line; **d** The GEs are both circles, and one circle lies in another; **e** The GEs are both arcs, and they are separate; **f** The GEs are both arcs, and they intersect with both concave parts remaining; **g** The GEs are both arcs, and they intersect with one concave part and one convex part remaining. GEi($i = 1,2$) are the generative elements for the MA; Pm represents any point on MA; Pt, Pt$_i$($i = 1,2$) are the points on GEs for the MA point P$_m$; Pr$_i$($i = 1,2$) are the center points of the arc generative elements, respectively

MA of the resultant model. Each step is discussed in the following in detail. Without loss of generality, the whole process is illustrated with Boolean union operation whereas the special characteristics of Boolean subtraction and parameter modification are discussed separately.

4 Determination of the changed boundaries

Generally, after a Boolean operation or parameter modification is conducted, some boundary edge segments will vanish and some will appear in the resultant model. The changed boundaries are imperative for generating the MA of the resultant model. They imply the heuristics for determining the RBS of the resultant model. The following will discuss how to find the changed boundary edges for Boolean operation and parameter modification, respectively.

4.1 The changed boundaries of Boolean operations

It is observed from Fig. 6 that the green and yellow thick boundary edges are VGEs that will not appear in the resultant model if the Boolean union operation is imposed on models A and B. Similarly, the yellow thick boundary edges will be the AGEs whereas the green thick edges will be the VGEs if the Boolean subtraction operation $B - A$ is conducted. Noticeably, it is the VGEs and AGEs that form the exact boundary of the resultant model M_{I} of the intersection of model A and B. Based on the above analysis, the changed boundaries of the resultant model after the Boolean union and subtraction operation are formalized as follows:

$$\Delta\Omega(A \cup B) = \Omega(A \cap^{*} B) \tag{3}$$

$$\Delta\Omega^{-}(A - B) = \Omega(A \otimes A \cap^{*} B) \tag{4}$$

$$\Delta\Omega^{+}(A - B) = \Omega(A \cap^{*} B) - \Omega(A \otimes A \cap^{*} B) \tag{5}$$

Here, $\Omega(M)$ is the boundary of model M, $\Delta\Omega$ (Operation) is the changed boundary after the specific Operation. $\Delta\Omega^{-}(A - B)$ and $\Delta\Omega^{+}(A - B)$ stand for the vanishing and added boundaries, respectively, after the Boolean subtraction operation. Operation $\Omega(A \otimes A \cap^{*} B)$ stands for the boundary of model A contained in the resultant model of $A \cap^{*} B$. Operation $\cap^{*}$ stands for the irregular Boolean intersection.

Noticeably, the above Eqs. (3–5) are also effective for complicated cases in which there are inner loops contained in the models. As shown in Fig. 7a, there is a hole that is the inner loop for model A and B, respectively. The effective boundaries after various operations are shown in Fig. 7b–f. Here, the solid lines are valid for the corresponding resultant model whereas the dashed lines are just given to facilitate the understanding. In addition, the vertices of the resultant model M_{I} of the intersection of model A and B are worth being noted since they are used to break

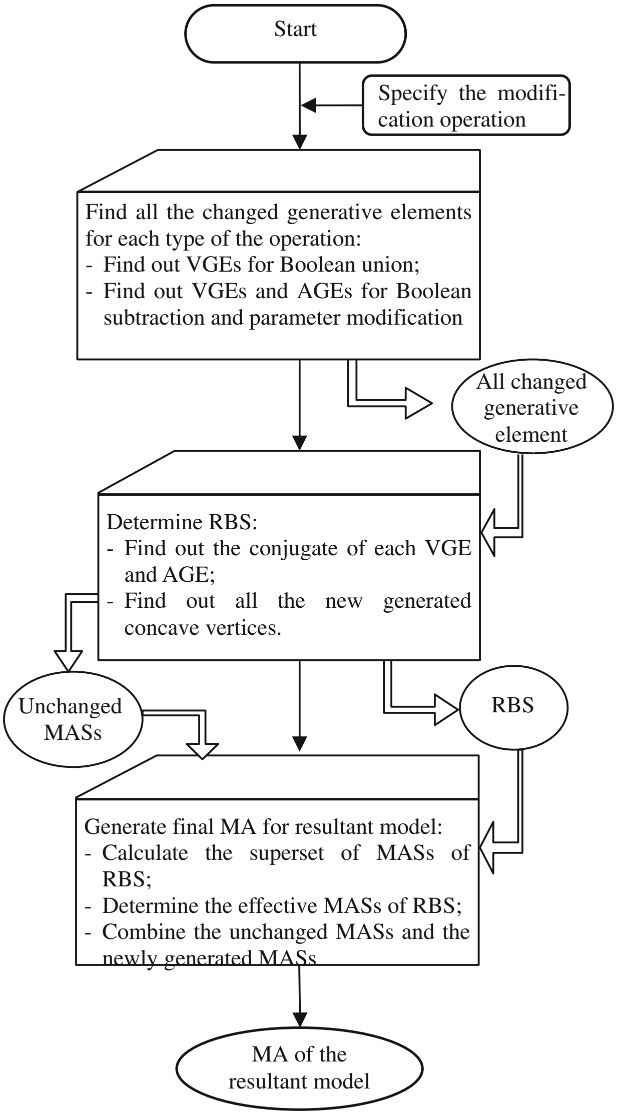

Start
Specify the modification operation
Find all the changed generative elements for each type of the operation:
- Find out VGEs for Boolean union;
- Find out VGEs and AGEs for Boolean subtraction and parameter modification
All changed generative element
Determine RBS:
- Find out the conjugate of each VGE and AGE;
- Find out all the new generated concave vertices.
Unchanged MASs
RBS
Generate final MA for resultant model:
- Calculate the superset of MASs of RBS;
- Determine the effective MASs of RBS;
- Combine the unchanged MASs and the newly generated MASs
MA of the resultant model

Fig. 5 Overview of the proposed method

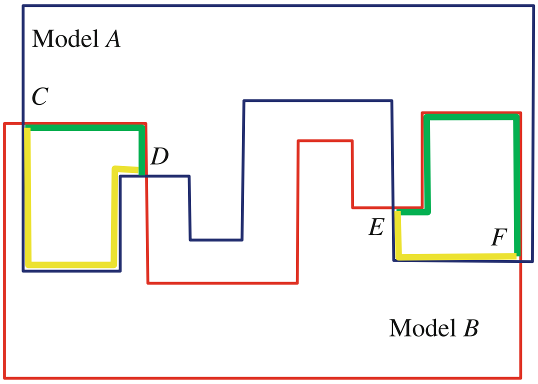


Fig. 6 Illustration of vanishing and added boundary edges

the pre-operated edges into two segments. Such broken points can be used as heuristic information for combining the newly generated MASs of the RBS with the reserved MASs of the operated model(s). We use a set named BPS to record them. Importantly, the concave vertices will be further used to generate the new MASs for the resultant model and need to be added into the RBS.

4.2 The changed boundaries after parameter modification

Modifying parameters is a usual means of refining models. In this study, the result of parameter modification on the model is replaced by that of the Boolean operation. As shown in Fig. 8, the resultant model is the same as unifying model B with model A when the size of dimension d_1 reduced to $d_1 - \delta_1$. Similarly, the resultant model is the same as subtracting model C from model A when the size of dimension d_1 is increased to $d_1 + \delta_2$. Therefore, to obtain the new boundaries after parameter modification, the models before and after the operation are compared firstly. Then, whether the Boolean union or subtraction operation is used is determined based on the comparison result. After that, they can be obtained by the same method as described in "Subsect. 4.1".

5 Generation of the RBS

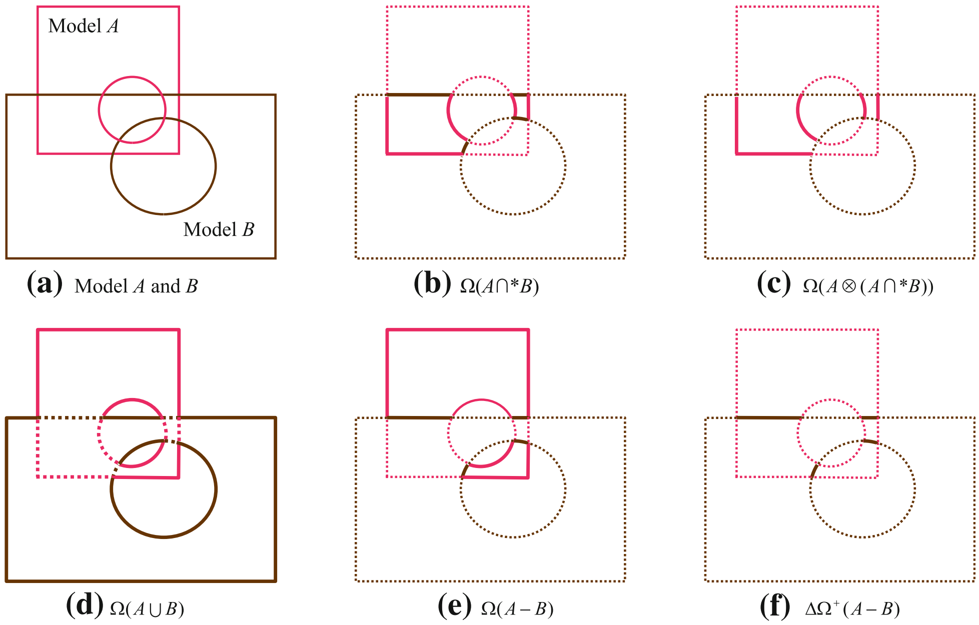
After all the VGEs and AGEs are identified, the next step is to evaluate their effects on the MA, and then generate the RBS.

5.1 The RBS of VGEs

According to theorem 1, the RBS of VGEs consists of their conjugates. The main task for generation RBS of each VGE is to identify its conjugate which is accomplished with the following steps:

1. Find out its corresponding MASs in the operated models. Specifically, for Boolean union operation, the MASs of both operated models should be identified. However, for the Boolean subtraction operation, only those of the subtracting model need to be identified. As shown in Fig. 9, the MASs $\{V_1C, CV_2\}$ and $\{AB\}$ are found for model P and model Q, respectively, when Boolean union operation is operated on them.
2. Determine whether the number of MASs in a specific operated model is greater than one. If so, the VGE must be divided into several segments for the model by

Fig. 7 The relationship of the boundary for different operations



(a) Model A and B **(b)** $\Omega(A\cap^{*}B)$ **(c)** $\Omega(A\otimes(A\cap^{*}B))$

(d) $\Omega(A\cup B)$ **(e)** $\Omega(A-B)$ **(f)** $\Delta\Omega^{+}(A-B)$

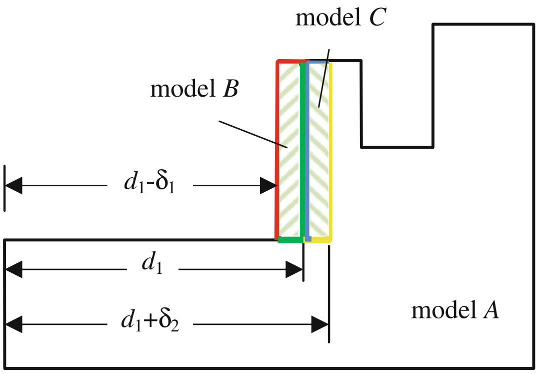


Fig. 8 Illustration of the subtracted and added boundary edges of the model when using parameter variations

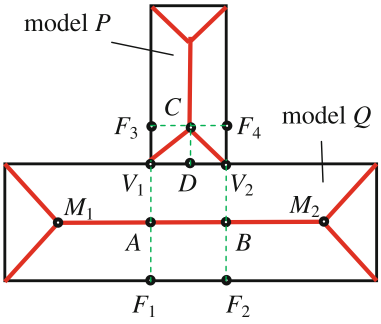


Fig. 9 Illustration of generation of the RBS for vanishing edges

making auxiliary straight lines from the endpoints of the MASs that are perpendicular to the VGE (or its tangent line for the arc type of generative element), respectively. The intersection points of the auxiliary straight lines and the VGE (or its tangent line) are also recorded, and they are regarded equivalently with the other endpoints of the VGE. As shown in Fig. 9, there are two MASs in model P for VGE V_1V_2, thus it should be divided into two parts: V_1D and DV_2 for model P.

3. For each model, from the endpoints of the VGE, make auxiliary straight lines that are perpendicular to the VGE itself and record the intersection points of the straight lines with the MAS of the VGE. As shown in Fig. 9, the intersection point sets are $\{V_1, C, V_2\}$ and $\{A, B\}$ for models P and Q, respectively.
4. For each model, from the intersection points on the MASs, make auxiliary straight lines that are perpendicular to the corresponding conjugates (or to their tangent lines for the arc type of generative element) of the VGE segment and record the intersection points of the straight lines with the conjugates. The edge segments between every two intersection points belong to the RBS. As shown in Fig. 9, the edge segments V_1F_3, V_2F_4, and F_1F_2 are the edge segments belonging to the RBS.
5. Add the newly generated concave vertices into the RBS and Refine it.

5.2 The superset of the RBS of AGEs

According to theorem 1, AGEs of the resultant model from the subtracted model must belong to the RBS. However, determining the conjugates of each AGE which also belong to the RBS is not a trivial problem. The reason is that the distance relationship is changed and thus the partner relationships are changed accordingly because of the adding of the AGEs from the subtracted model. In this study, a heuristic-based method is proposed to identify a superset of the RBS which contains not only all the conjugates of AGEs but also some redundant boundary edge segments. Although some redundant boundary edge segments are also added in the superset, the efficiency is still much higher than that of finding the exact RBS of AGEs. The proposed heuristic-based method is described as follows after the AGE set is given:

Step 1 Find the following three vertex set: (i) all the intersections between the input AGEs and the existing boundary edges, $\{V_{ab,\ i}\}$ ($i = 0, \ldots, m$); (ii) all the intersections between the AGEs and the existing MASs, $\{V_{am,\ i}\}$ ($i = 0, \ldots, n$); (iii) all the concave vertices on the AGEs, $\{V_{cc,\ i}\}$ ($i = 0, \ldots, k$).

Step 2 If the set $\{V_{ab,\ i}\}$ is empty, go to step 3. Otherwise, Put the boundary edges that intersect with the AGEs into the superset of RBS since there must be new MASs between the AGEs and the boundary edges that intersects with them. As shown in Fig. 10a, *DH* and *EP* must generate new MASs with a part or the whole of *DG* and *EF*, respectively, and thus they belong to superset of RBS when model *B* is subtracted from model *A*.

Step 3 If the set $\{V_{am,\ i}\}$ is empty, go to step 4. Otherwise, there must be at least one MAS that is destroyed by the AGEs. The destroyed MASs will not exist in the resultant model and thus the MASs of their generative elements must be regenerated. Therefore, the generative elements of the destroyed MASs belong to the superset of RBS. As shown in Fig. 10b, the MAS *KX* is destroyed when model *C* is subtracted from model *A*. Its generative elements, i.e., part of *DK* and part of *KJ*, have to be put into the superset of RBS.

Step 4 If the set $\{V_{cc,\ i}\}$ is empty, go to step 5. Otherwise, sort the concave vertices in a sequence. For each concave vertex, find two nearest boundary edges or concave vertices which satisfy the following conditions:

(i) It does not intersect with the AGE one of whose vertices is the concave vertex itself.

(ii) The perpendicular line from the concave vertex to the nearest boundary edges cannot intersect with other AGEs or other boundary edges.

As shown in Fig. 10a, the nearest boundary edges are *HI* and *LM* for concave vertex *G* when model *B* is subtracted from model *A* and the above conditions are both satisfied. For concave vertex *W* when model *C* is subtracted from model *A*, the initial nearest boundary edges are *OP* and *EX*. However, the perpendicular line from vertex *W* to *OP* will intersect with the AGE *ST* and thus *OP* does not satisfy the needed condition.

After both nearest boundary edges or concave vertices which satisfy the above conditions are found, the following different processes are conducted according to the types of the found elements:

- Put both the found concave vertices or one concave vertex and one boundary edge into the superset of RBS directly if there is at least concave boundary vertex that is found.
- If both the found elements are edges, determine whether the two found edges are connected. If so, put both of them into superset of RBS directly. Otherwise, determine whether the connected edges between them also need to be put into the super set of RBS. There is at least a boundary concave vertex lying between the found boundary edges since they are the nearest edges of the new concave vertex and not connected directly. According to the pre-property 2, the AGEs do not affect the MASs of other boundary edges and thus the connected boundary edge between the found boundary edges need not be put into the superset of RBS. As shown in Fig. 10a, *HI* and *LM* need to be put into the superset of RBS since they are the nearest boundary edges of concave vertex *G*. However, the connection edges such *IJ* and *JK* need not be put into the superset of RBS.

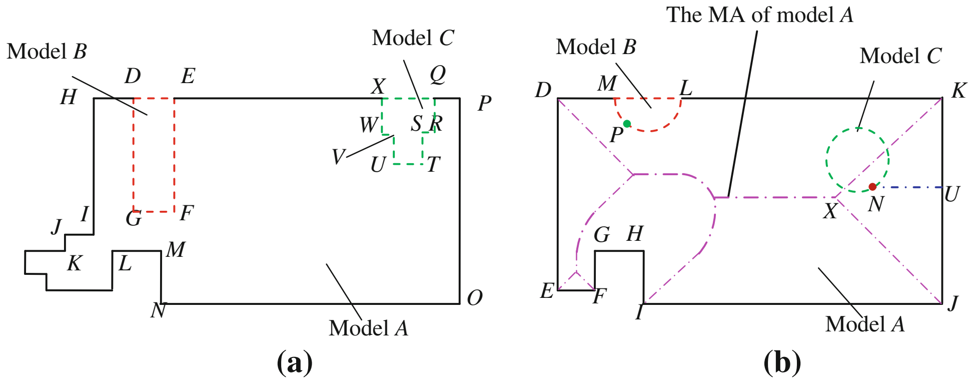


Fig. 10 Illustration of the superset of RBS when a model is subtracted from another model. **a** There are concave vertices when new boundary edges are added. **b** There is no concave vertex when new boundary edge is added

- *Step 5* If there is no concave vertex on the AGEs, an arbitrary seed point on the AGE is selected at first and a nearest boundary edge is found which satisfies the above two conditions. Here, only one nearest boundary edge needs to be found since there is no concave vertex on the AGEs which means that the AGE is a smooth curve. As shown in Fig. 10b, there is no new concave vertex to be generated when model *B* or *C* is subtracted from model *A*. The random point, *P* or *N* on the AGE, is selected as the seed point and then the nearest boundary edge satisfying the above conditions, *DK* and *KJ*, respectively, is found.
- *Step 6* Refine the superset of RBS by eliminating the same boundary edges of being put in repeatedly and combining some segments that belong to the same boundary edge. As shown in Fig. 10a, the nearest boundary edges *EX* and *ON* can be found for concave vertex *W* when model *C* is subtracted from model *A*. Similarly, the boundary edge *ON* can also be found for concave vertex *U*. Therefore, it should be eliminated for refining the super set of RBS.

6 Generation of the MASs of the RBS

The generation of MASs of the RBS is divided into two steps. Firstly, the superset of MASs of the RBS, i.e., all the possible MASs are created. Then, the effective MASs are selected, and the incorrect and redundant ones are eliminated. After that, the newly generated MASs of the RBS are combined with the unchanged MASs of the operated model(s) to obtain the complete MA for the resultant model.

The superset of new MASs is the set consisting of all the MASs which are generated based on the combination of every two elements in the RBS, and thus it is composed of all the effective new MASs, as well as the unnecessary new MASs. Obviously, it is easy to get the superset of new MASs based on the basic MA types discussed in "Subsect. 3.3". To obtain the correct MASs from the superset of candidate MASs for the RBS, it is necessary to select suitable MASs from the superset at first, and then determine the endpoints of the selected MASs. In this study, the tracing method is used to effectively select suitable MASs. First, the start tracing point is determined with the help of the broken points. It is an efficient and effective way to take a broken point as the start point to trace the new MASs, since the new MAS must be C^0 continuous with the reserved MASs of the operated model(s).

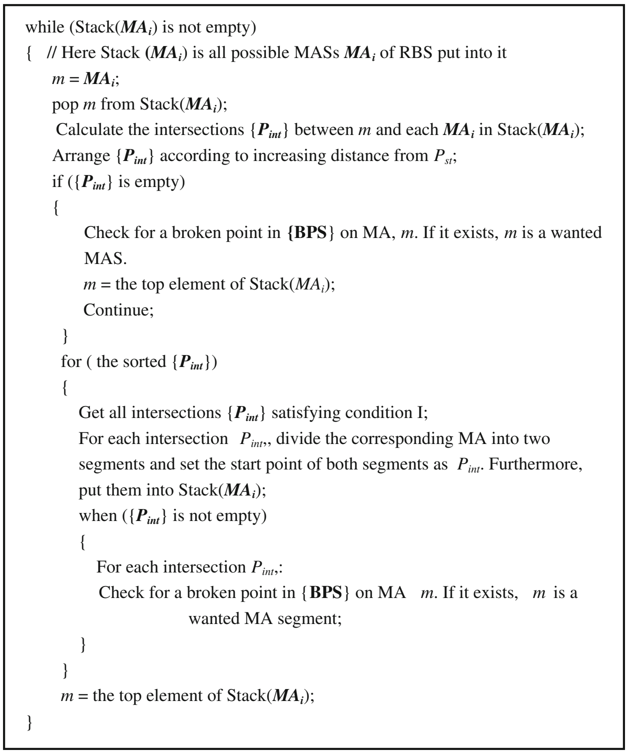
According to the tracing method, after an effective MAS and its endpoints are determined, the next effective MASs that are adjacent to it can be determined similarly. According to the property of the MA, the number of MASs connecting one MAS at one endpoint must be less than 2, regardless of the degeneration. Each generative element of the 2-prong MAS or the adjacent boundary of each generative element must be that of the next MAS, according to property 1. Based on the above analysis, a modified tracing method is described for determining the endpoint of one MAS and the next effective MAS (denoted as MA_i hereafter):

1. Find all MASs that intersect with MA_i and arrange them and the intersection points according to the distances between $P_{st,i}$ and the intersection points in increasing order. Here, $P_{st,i}$ is the start point of MA_i.
2. Make a circle disc with the given intersection point as its center and the distance between the given intersection point and each generative element of MA_i as its radius. Then check whether the following condition is satisfied:

Condition I The circle disc lies inside the model, and besides the tangent points with the generative element of MA_i, there is only one other tangent point with the boundary edges of the model.

If so, the intersection point is the endpoint of MA_i.

3. Divide each of the next segment(s) $MA_{i+1,j}$ $(j = 1, 2)$ into two segments with $P_{st,i+1}$ being their start point.



```
while (Stack($\boldsymbol{MA}_i$) is not empty)
{   // Here Stack ($\boldsymbol{MA}_i$) is all possible MASs $\boldsymbol{MA}_i$ of RBS put into it
    $m = \boldsymbol{MA}_i$;
    pop $m$ from Stack($\boldsymbol{MA}_i$);
    Calculate the intersections {$\boldsymbol{P}_{int}$} between $m$ and each $\boldsymbol{MA}_i$ in Stack($\boldsymbol{MA}_i$);
    Arrange {$\boldsymbol{P}_{int}$} according to increasing distance from $P_{st}$;
    if ({$\boldsymbol{P}_{int}$} is empty)
    {
        Check for a broken point in {BPS} on MA, $m$. If it exists, $m$ is a wanted
        MAS.
        $m$ = the top element of Stack($MA_i$);
        Continue;
     }
     for ( the sorted {$\boldsymbol{P}_{int}$})
     {
        Get all intersections {$\boldsymbol{P}_{int}$} satisfying condition I;
        For each intersection  $P_{int}$,, divide the corresponding MA into two
        segments and set the start point of both segments as  $P_{int}$. Furthermore,
        put them into Stack($\boldsymbol{MA}_i$);
        when ({$\boldsymbol{P}_{int}$} is not empty)
        {
           For each intersection $P_{int}$,:
           Check for a broken point in {BPS} on MA   $m$. If it exists,   $m$ is a
                          wanted MA segment;
        }
     }
     $m$ = the top element of Stack($\boldsymbol{MA}_i$);
}
```

Fig. 11 Pseudo code of the algorithm for generating the effective MASs for the RBS

Furthermore, decide which segment is the effective MAS. Generally, a MAS will not continue to be effective when passing through the intersection point with consideration of the property of local continuity.

4. Repeat the above steps until there is no intersection point between the current MAS MA_i and any of the remaining MASs or until none of the intersection points satisfy condition I.

The pseudo-code of the algorithm is given in Fig. 11.

After all the new effective MASs are determined for the RBS, it is easy to obtain the final MA of the resultant model by combining new MASs with the reserved MASs of the operated model(s).

7 Degeneration analysis

As mentioned above, the number of n-prong ($n > 2$) MA points is finite, i.e., most of MA points are 2-prong, and they are always formed as curve segments such as parabola, hyperbola, and ellipse. A 3-prong MA point indicates that a bifurcation point exists which is connected to three 2-prong MASs. Usually, the 3-prong MA points are separated from each other, as shown in Fig. 12a (M_1 and M_2 are both 3-prong MA points). However, several 3-prong MA points may be coincident in some cases as shown in Fig. 12b. In this study, it is defined as n-dimensional degeneration (denoted as n-degeneration) when there are $n + 1$ 3-prong MA points in coincidence.

According to the definition of MA, it is easy to determine whether there is a degeneration instance and what dimension each degeneration instance is. However, it is not a trivial task to obtain the MA by local adaptation when degeneration instances exist. The key problem here is that the partner relationship of the generative elements must be recombined for the coincident 3-prong MA points with different generative elements. That is, the generative elements which come from different coincident 3-prong MA points may be recombined to act as the generative elements of each 3-prong MA point when the degenerated, coincident 3-prong MA points are separated from each other because of a Boolean operation or parameter modification. As shown in Fig. 12c, the generative elements for 3-prong MA points C are recombined with those of 3-prong MA points M_1 and M_2 in Fig. 12a.

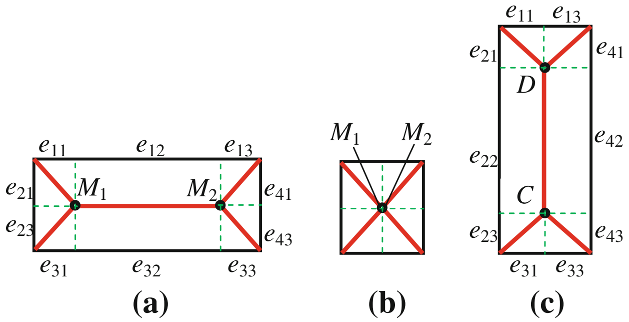


Fig. 12 Recombination of the GEs of the coincident 3-prong MA points

To solve the problem, a special structure is devised to record the degenerated MA point in which, for each coincident 3-prong MA point, the generative elements are in turn saved individually for each of its 2-prong MA points. Therefore, the intersection of 2-prong MA segments of a certain coincident MA point can be recombined freely. Here, the reason for 2-prong MASs used as the elementary unit of degenerated 3-prong MA points is that the generative elements are fixed for the 2-prong MA point.

8 Implementation and discussion

The proposed method was implemented with visual C++.net 2005 based on the geometric modeling kernel ACIS 13.0. The experiments were conducted on a personal computer with an Intel® Core i5 2.8G CPU and 4G RAM. To test the computer efficiency of the proposed method, three groups of experiments were carried out. The first is about the comparison of the computational time of the proposed method with that of state-of-art methods after the local modification was carried out on the complicated model. The second is about the statistic analysis of the relationship between the calculation time and the number of edges. The last is about trying to utilize the proposed method to generate the MA incrementally during the modeling process. Before discussing the examples, the algorithm complexity is analyzed approximately.

8.1 Algorithm complexity analysis

Suppose that the number of boundary elements (including the boundary edges and the concave vertices) is n for model A. For each boundary element, the maximum number of MA one of whose generative elements is the boundary element is n-1. Therefore, the number of the affected MA is $k(n\text{-}1)$ if the number of the deleted and vanishing boundary elements is k, i.e., O(kn). The computation complexity of the proposed method is O(k^2n^2), which is quite unsatisfactory. However, it is noticeable that the number of the affected boundary elements is finite for each modification. Especially when Boolean union operation is conducted to add a tiny modification for a complicated model during engineering analysis. That is, the algorithm complexity becomes O(n^2).

Furthermore, it is clear that it is impossible that each boundary element can generate MA with all other boundary elements. Exactly, the number of conjugates for each boundary element is no more than two according to preparatory property 1. Therefore, the algorithm complexity is approximately O(1). That is, the algorithm complexity is almost independent on that of the model for most cases. It is a pretty important property for the proposed method. The following examples will demonstrate the property for generating MA during engineering analysis.

8.2 Comparison of calculation time

In the first example, our proposed method is compared with those of Hesselink [4] and Chin [5]. Their computation efficiencies are both linear time. Here, the four used example models with local modification are shown in Figs. 13, 14, 15 and 16a, b. The new generated MASs of the RBS are marked in green. Table 1 shows the calculation time of the three methods. It can be seen from Table 1 that, compared to Hesselink's and Chin's method, the computational time of the proposed method is dramatically reduced.

8.3 Statistic analysis of the relationship between the calculation time and the edge number

To further explore the relationship between the model complexity and the computational time, a more elaborate test was conducted. Here, 10 groups of models, each of which had 10 different models with the same number of edges were utilized. These models were collected and modified from Princeton Shape Benchmark (PSB) (http://www.shape.cs.princeton.edu/benchmark/). Some typical CAD models are selected and their 2D contours are utilized here. Some random modifications were given for each model. The computation time was summed and averaged

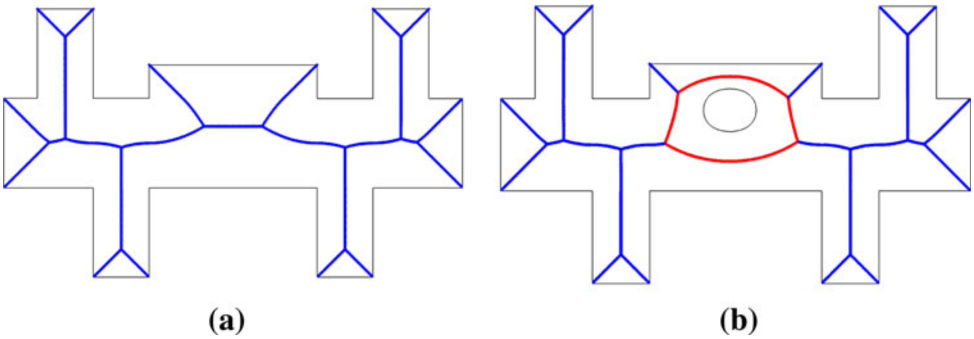


Fig. 13 The model modified by adding a hole

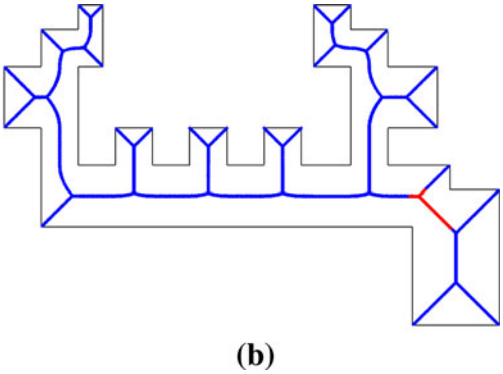


Fig. 14 The model modified from 38 edges to 42 edges

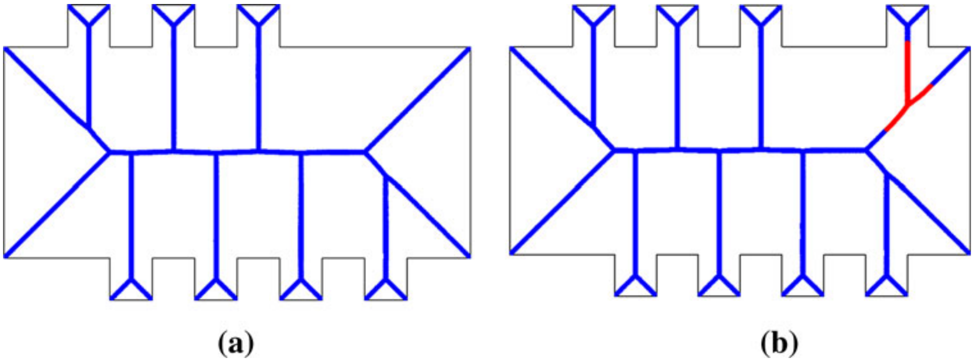


Fig. 15 The model modified from 60 edges to 64 edges

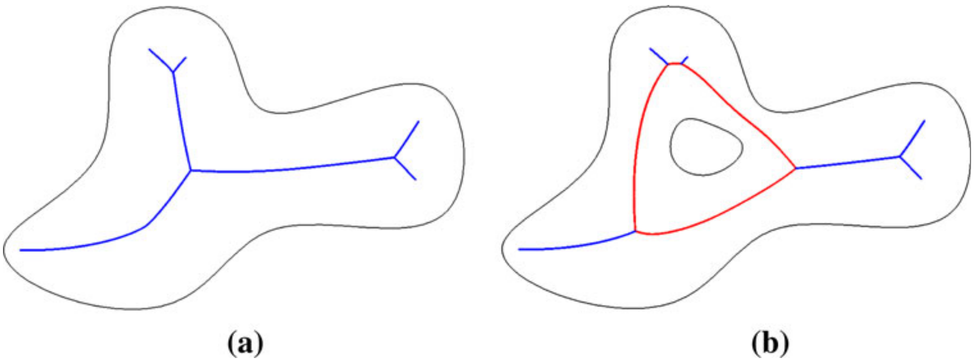


Fig. 16 The model modified with free-form edges

Table 1 Comparison of calculation time of different methods

Figure no.	Calculation time of different methods (s)		
	Hesselink's method	Chin's method	Our method
13	0.22	0.238	0.042
14	0.327	0.342	0.043
15	0.449	0.515	0.043
16	0.212	0.226	0.040

for each group. The result of the relationship between the computational time and the number of edges is given in Fig. 17. It can be seen from Fig. 17b that when the model is complicated and just locally modified by the Boolean operations during engineering analysis, the computational time almost stays within a fixed range for the proposed method. The calculation time is not absolutely the same and there are also some small differences since the number of Boolean union operations and that of subtraction operations are not the same. Our experience has shown that the Boolean subtraction operations involve a larger RBS than Boolean union operations do and thus more time is needed.

On the other hand, the calculation time increases approximately in a linear proportion for the methods of Hesselink [4] and Chin [5] in which the MA is regenerated from scratch for the whole model. Based on the analysis, it is obvious that the computational efficiency of the proposed method is a dramatic improvement over the state-of-art methods, and its computational time is almost independent of the number of edges but dependent on the RBS. Noticeably, the proposed method is not as effective for simple models as it is for complicated models. For example, the MA of a circle is one point, i.e., the center of the circle. If a smaller, concentric circle is subtracted from it at the center position, the MA of the resultant model is also a circle which is totally different from those of the individual primitives. The reason is that the RBS in this case is the total boundary, i.e., the circle. In this case, the MA should be regenerated for the whole model even though the proposed method is applied. Fortunately, the models to be analyzed in engineering are always complicated. And the computation time will be very small even if the simple model appears in engineering design and analysis since there are a finite number of boundary edges for the simple models.

8.4 Incremental generation of MA by the proposed method

In the third experiment, the proposed method is used to try to generate the MA incrementally during the modeling

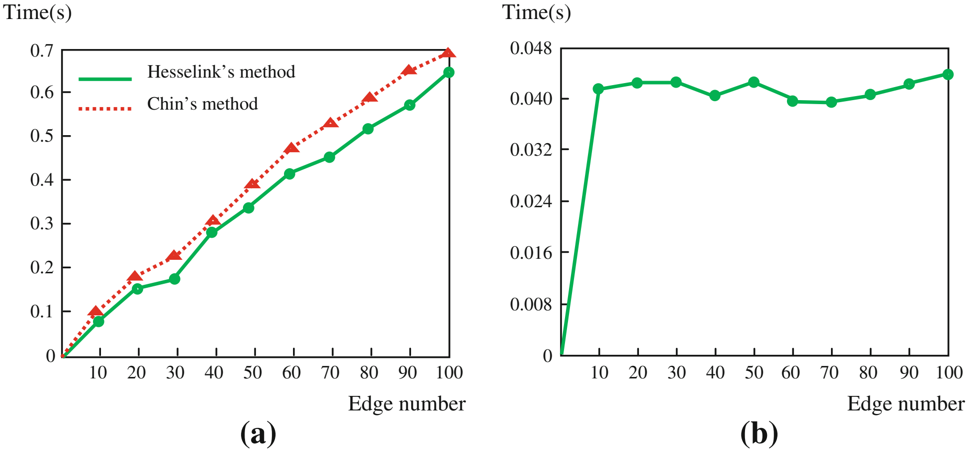


Fig. 17 The relationship between the calculation time and the number of edges. **a** For the Hesselink's and Chin's method. **b** For the proposed method

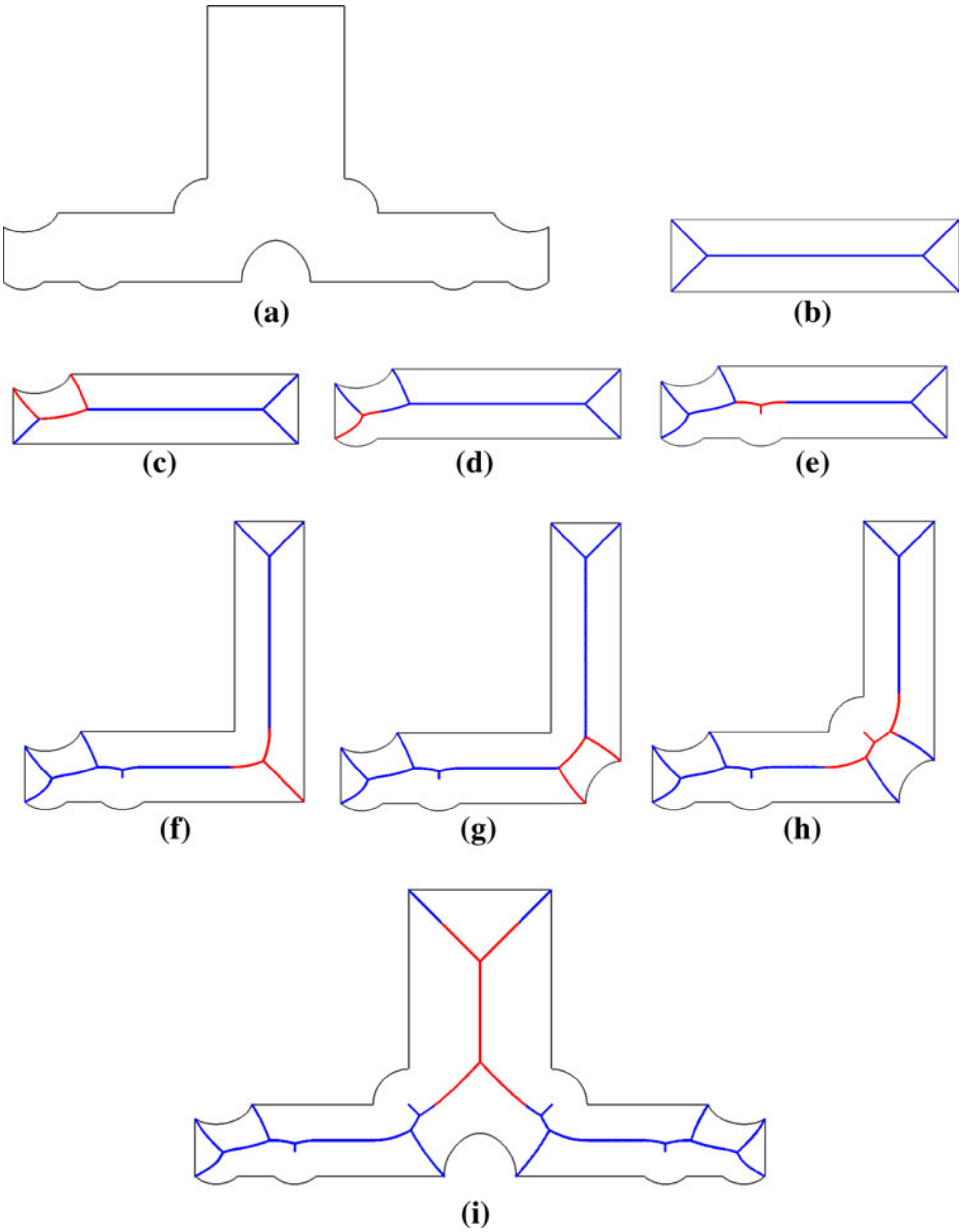


Fig. 18 An artifact and the incremental generation process of its MA

process. Figure 18b–i illustrates the modeling process and the corresponding MA of the given model shown in Fig. 18a. In each step of the example, the MASs marked with blue solid lines are the reused ones. The new ones generated for the RBS are marked with red color. The computational time for each step which includes that for conducting the Boolean operation and that for calculating the new MASs is given in Table 2. It can be seen from Table 2 that the total computation time for each step is almost the same.

Moreover, to test the influence of different orders of Boolean operations, several groups of Boolean operations with different orders are tested to construct the model of Fig. 18a. One typical modeling process is given in Fig. 19. The calculation time is given in Table 3. It can be seen from Tables 2 and 3 that their computational time is similar and thus the influence of the modeling orders can be neglected.

9 Conclusions

A novel, local adaptation-based approach to generating the MA of a 2D model is proposed for efficient engineering analysis in this study. The major contributions of the work include:

Table 2 The MA calculation time of each step for the model shown in Fig. 17a

Step	*c*	*d*	*e*	*f*	*g*	*h*	*i*
T1 (s)	0.029	0.028	0.028	0.031	0.028	0.027	0.030
T2 (s)	0.011	0.013	0.013	0.012	0.010	0.010	0.014
T3 (s)	0.040	0.041	0.041	0.043	0.038	0.037	0.044

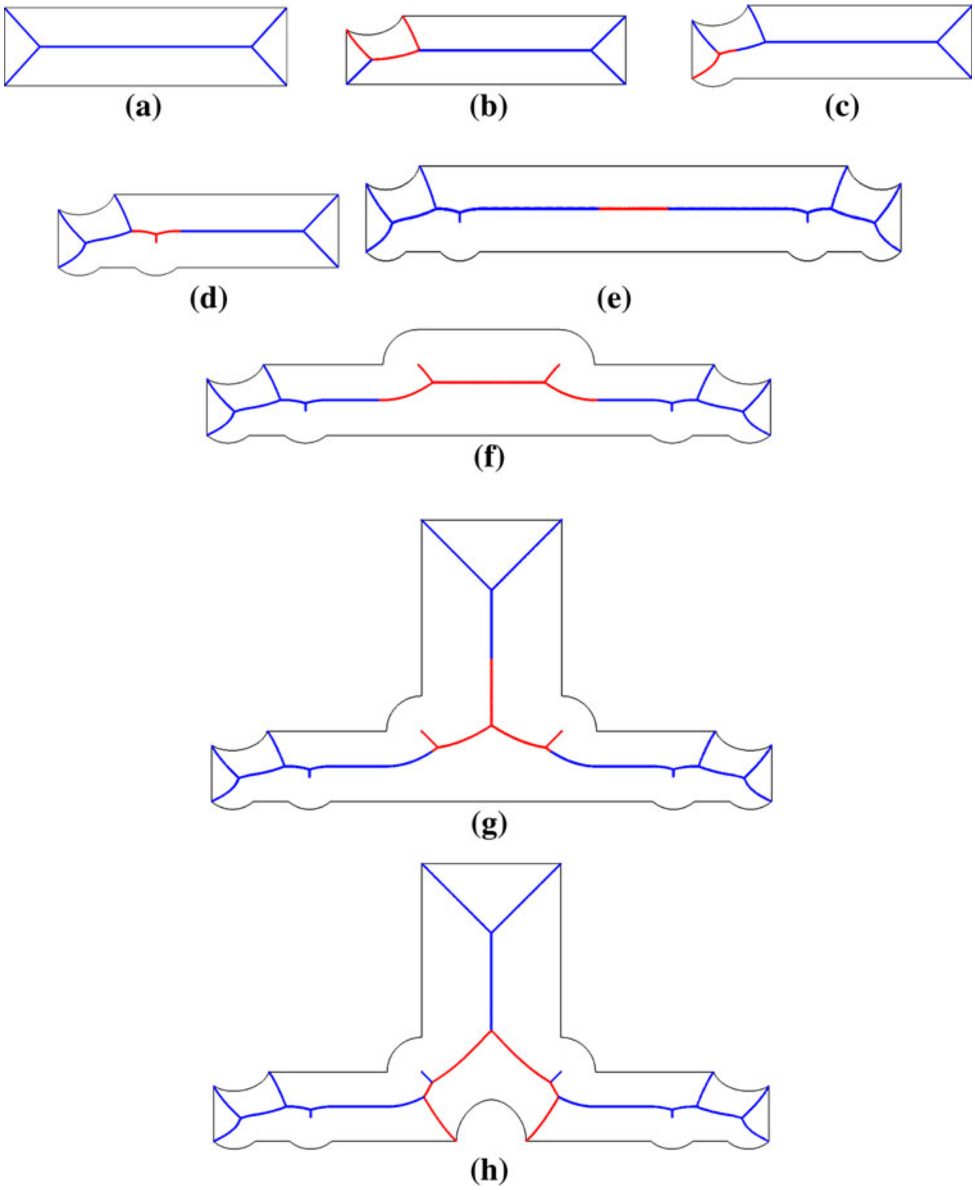


Fig. 19 The incremental MA of another modeling process different from that of Fig. 18

Table 3 The MA calculation time of MA for the modeling process shown in Fig. 18

Step	*b*	*c*	*d*	*e*	*f*	*g*	*h*
T1 (s)	0.029	0.028	0.028	0.027	0.030	0.030	0.030
T2 (s)	0.011	0.013	0.013	0.011	0.011	0.010	0.015
T3 (s)	0.040	0.041	0.041	0.038	0.041	0.040	0.045

T1, T2 and T3 stand for the calculation time for each Boolean operation, the time for calculating the new MASs and the total calculation time of each modeling step

1. The concept of RBS is set forward, and the method of automatically determining the RBS of Boolean union and subtraction operations as well as parameter modifications is given, which guarantees that the original MASs can be reused as much as possible to generate the MA of the resultant model.
2. Several new properties of the MA which are related to local adaptation are put forward as a fundamental basis for the proposed method. With the help of these properties, it is proved that the modified region of MA is often finite for complicated models when the changed boundary is local.
3. An approach to generating the MA efficiently is proposed when local modifications happen for the model refinement during engineering analysis. With this method, the MA of the model after engineering analysis can be generated based on local adaptation and reusing those of the operated model(s), rather than regenerating it from scratch. Therefore, the computational efficiency can be improved dramatically for complicated models with minor modifications. Similarly, the proposed method can also be used in the modeling process so that the MA can be generated incrementally.

The future work will proceed in two directions. The first is to add more modeling primitives, and the second is to extend the method to 3D models.

Acknowledgments The authors are appreciated for the support from the 863 High-Technology Project of China (No. 2011AA100804), NSF of China (61173126, 91024007, 70901052) and Zhejiang Provincial Natural Science Foundation of China(R1110377).

References

1. Blum H (1967) A transformation for extracting new descriptors of shape. Models for the Perception of Speech and Visual Form. MIT Press, Weinant Wathen-Dunn, pp 362–381
2. Ramanathan M, Gurumoorthy B (2002) Constructing medial axis transform of planar domains with curved boundaries. Comput Aided Des 35:619–632
3. Patrikalakis NM, Maekawa T (2002) Shape interrogation for computer aided design and manufacturing. Springer, Berlin
4. Hesselink WH, Roerdink JBTM (2008) Euclidean skeletons of digital image and volume data in linear time by the integer medial axis transform. IEEE Trans PAMI 30(12):2204–2217
5. Chin F, Snoeyink J, Wang CA (1999) Finding the medial axis of a simple polygon in linear time. Discret Comput Geom 21:405–420
6. Held M (2001) VRONI. An engineering approach to the reliable and efficient computation of Voronoi diagrams of points and line segments. Comput Geom 18:95–123
7. Lee DT (1982) Medial axis transformation of a planar shape. IEEE Pattern Anal Mach Intell 4:363–369
8. Yap CK (1987) An O(nlogn) algorithm for the Voronoi diagram of a set of simple curve segments. Discret Comput Geom 2:365–393
9. Aichholzer O, Aigner W, Aurenhammer F et al (2009) Medial axis computation for planar free_form shapes. Comput Aided Des 41:339–349
10. Nackman LR (1982) Curvature relations in three-dimensional symmetric axes. Comput Graph Image Process 20:43–57
11. Bookstein FL (1979) The line skeleton. Comput Graph Image Process 11:123–137
12. Scott GL, Turner SC, Zisserman A (1989) Using a mixed wave diffusion process to elicit the symmetry set. Image Vis Comput 7(1):63–70
13. Siddiqi K, Bouix S, Tannenbaum A, et al. (1999) The Hamilton-Jacobi skeleton. In: International Conference on Computer Vision (ICCV), pp 828–834
14. Ragnemalm I (1993) Pattern recognition letters. Pattern Recogn Lett 14(11):883–888
15. Hoff KE, Keyser J, Lin M, et al (1999) Fast computation of generalized Voronoi diagrams using graphics hardware. Comput Graph 33(Annual Conference Series):277–286
16. Vleugels J, Overmars M (1995) Approximating generalized Voronoi diagrams in any dimension. Technical Report UU-CS-95-14, Department of computer science, Utrecht University
17. Foskey M, Lin M, Manocha D (2003) Efficient computation of a simplified MA. In: CD proceedings of the ACM symposium on solid and physical modeling
18. Montanari U (1969) Continuous skeletons from digitized images. J Assoc Comput Mach 16(4):534–549
19. Lee DT (1982) MA transformation of a planar shape. IEEE Trans Pattern Anal Mach Intell 4(4):363–369
20. Srinivasan V, Nackman LR (1987) Voronoi diagram for multiply connect polygonal domains, I: algorithm. IBM J Res Dev 31(3):361–372
21. Gursoy HN, Patrikalakis NM (1991) Automated interrogation and adaptive subdivision of shape using MA transform. Adv Eng Softw Workstn 13(5/6):287–302
22. Gursoy HN, Patrikalakis NM (1992) An automated coarse and fine surface mesh generation scheme based on MA transform, part I: algorithms. Eng Comput 8(3):121–137
23. Gursoy HN, Patrikalakis NM (1992) An automated coarse and fine surface mesh generation scheme based on MA transform, part II: implementation. Eng Comput 8(4):179–196
24. Culver T, Keyser J, Manocha D (1998) Accurate computation of MA of a polyhedron. Technical Report TR98-034, UNC-Chapel Hill
25. Choi HI, Choi SW, Han CY et al (2008) Two-dimensional offsets and medial axis transform. Adv Comput Math 28:171–199
26. Choset H (1997) Incremental construction of the generalized Voronoi diagram, the generalized Voronoi graph, and the hierarchical generalized Voronoi graph. In: 1st CGC workshop on computation geometry
27. Chiang C-S (1992) The Euclidean distance transform. Ph.D. thesis, Department of Computer Science, Purdue University, West Lafayette, Report CSD-TR 92-050
28. Sherbrooke EC, Patrikalakis NM, Brisson E (1995) Computation of MA transform of 3-D polyhedral. In: ACM solid modeling, pp 187–199
29. Reddy JM, Turkiyyah GM (1995) Computation of 3D skeletons using a generalized Delaunay triangulations technique. Comput Aided Des 27(9):677–694
30. Dutta D, Hoffmann CM (1990) A geometric investigation of the skeleton of CSG objects. UM-MEAM-90-02
31. Cao L, Jia Z, Liu J (2009) Computation of medial axis and offset curves of curved boundaries in planar domains based on the Cesaro's approach. Comput Aided Geom Des 26:444–454
32. Cao T, Tang K, Mohamed A, Tan T (2010) Parallel banding algorithm to compute exact distance transform with the GPU. In:

Proceedings of the ACM SIGGRAPH symposium on interactive 3D graphics and games (I3D). New York, 19–21 Feb 2010

33. Lavender D, Bowyer A, Davenport J et al (1992) Voronoi diagrams of set-theoretic solid models. IEEE Comp Graph Appl 12(5):69–77
34. Brandt JW (1994) Convergence and continuity criteria for discrete approximations of the continuous planar skeleton. CVGIP: Image Underst 59(1):116–124
35. Dey TK, Woo H, Zhao W (2003) Approximate MA for CAD models. In: Proceedings of the solid and physical modeling 2003, Seattle, Washington, 16–20 June, 2003
36. Sheehy DJ, Armstrong CG, Robinson DJ (1996) Numerical computation of medial surface vertices. In: Mullineux G (ed) The mathematics of Surfaces VI. IMA, Oxford University Press, Oxford
37. Etzion M, Rappoport A (1999) Computing the Voronoi diagram of a 3D polyhedron by separate computation of its symbolic and geometric parts. in W. F. Bronsvoort and D. C. Anderson, editors. In: Proceedings of fifth symposium on solid molid modeling and applications, Ann Arbor, ACM, pp 167–168
38. Ramamurthy R, Farouki T (1999) Voronoi diagram and medial axis algorithm for planar domains with curved boundaries I: Theoretical foundations. J Comput Appl Math 102:119–141
39. Joachim G, Balint M, Mark P (2007) Medial axis approximation of planar shapes from union of balls: a simpler and more robust algorithm. In: Canadian Conf. on Computational Geometry, Ottawa, Canada, 20–22 Aug, 2007
40. Choi HI, Choi SW, Moon HP (1997) Mathematical theory of medial axis transform. Pac J Math 181(1):1997