Computational Visual Media
https://doi.org/10.1007/s41095-021-0265-1

Research Article

A Voronoi diagram approach for detecting defects in 3D printed fiber-reinforced polymers from microscope images

Xiang Li[1], **Sara McMains**[1] (✉)

Abstract Fiber-reinforced polymer (FRP) composites are increasingly popular due to their superior strength to weight ratio. In contrast to significant recent advances in automating the FRP manufacturing process via 3D printing, quality inspection and defect detection remain largely manual and inefficient. In this paper, we propose a new approach to automatically detect, from microscope images, one of the major defects in 3D printed FRP parts: fiber-deficient areas (or equivalently, resin-rich areas). From cross-sectional microscope images, we detect the locations and sizes of fibers, construct their Voronoi diagram, and employ α-shape theory to determine fiber-deficient areas. Our Voronoi diagram and α-shape construction algorithms are specialized to exploit typical characteristics of 3D printed FRP parts, giving significant efficiency gains. Our algorithms robustly handle real-world inputs containing hundreds of thousands of fiber cross-sections, whether in general or non-general position.

Keywords 3D printing (3DP), microscope image processing; fiber-reinforced polymer; Voronoi diagrams; α-shapes; resin-rich areas

1 Introduction

3D printing (3DP), or additive manufacturing, constructs 3D physical objects directly from digital models by successively depositing material layer by layer. The materials, methods, and applications of 3DP have been widely investigated due to advantages such as flexible customization, quick start to production, and the ability to fabricate complex geometries. Traditional 3DP systems use isotropic materials, such as metals or plastic filaments, whose mechanical properties are identical in all directions (though the layered prints are anisotropic). Recently, however, significant technical advances in 3DP processes have enabled printing with anisotropic materials to manufacture continuous fiber-reinforced polymer (FRP) composites [1, 2]. Compared to typical 3DP metal or plastic parts, FRP parts have superior strength-to-weight ratio in the fiber directions (e.g., lightweight 3D printed FRP parts have greater tensile strength than aluminum parts [2]).

Although the FRP 3D printing process is progressively improving, defect detection in printed parts remains largely manual and inefficient. Defect detection, failure analysis, and material and mechanical properties of a part are studied by examining cross-sectional microscope images [3]. In continuous FRP 3D printing, parts are typically printed with a unidirectional or cross-ply laminate structure [4, 5], in which all fibers in the part, or in the same layer respectively, have the same direction. Cross-sectional micrographs, taken transversely to the fiber direction, can effectively assist inspection and analysis of the material microstructure of each 3D printed layer (cross-ply structure) or the whole printed part (unidirectional structure). Figure 1 shows an example of a transverse microscope image of a cross-section of a unidirectional continuous FRP 3DP part, where the circular fiber cross-sections and the surrounding polymer (the resin, or matrix) can be seen. The strength and stiffness of the composite material is primarily provided by the fibers, with the polymer providing a cohesive matrix to protect and hold the fibers together, so a uniform fiber distribution and a high fiber volume fraction are desirable. Ideally, fibers are aligned and thus have

---

1 University of California Berkeley, Berkeley, California, 94720, USA. E-mail: X. Li, xli@berkeley.edu; S. McMains, mcmains@berkeley.edu (✉).

Manuscript received: 2021-08-09; accepted: 2021-11-17

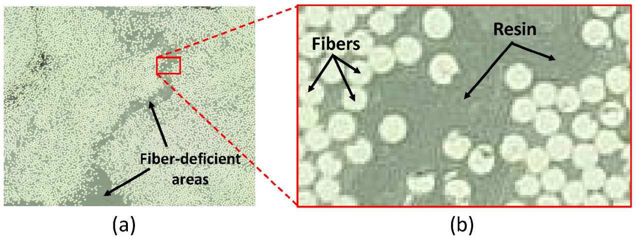

Fig. 1 (a) Transverse optical microscope image, and (b) a close-up, showing a cross-section of fibers and resin.

circular cross-sections in the transverse microscope images, but misaligned fibers may exist, indicating areas of reduced strength; these appear as ellipses (Fig. 2).

Because of their different characteristics, the geometric distribution of the fiber and resin is a key indicator when evaluating the quality of the part and printing process. As shown in Fig. 1(a), unevenly distributed fibers in the resin results in areas with insufficient reinforcement; such areas are commonly called *resin-rich areas* and lead to impaired mechanical properties and potential part failures [6, 7]. To emphasize the fact that such areas are actually defects, we will use the alternative term *fiber-deficient areas* in this paper. Currently, fiber-deficient area detection using microscope images is largely performed manually by experienced researchers [8].

In this paper, we present a novel algorithm to automatically detect fiber-deficient areas from microscope images of 3D printed FRP parts (see Fig. 3). This algorithm takes the result of circle and ellipse detection on the image as the locations of fibers, and determines the fiber-deficient areas from the input circles and ellipses using their Voronoi diagram and corresponding α-shape [9]. Our main contributions include:

- A novel approach to automatically find fiber-deficient areas in microscope images of 3D printed FRP parts using α-shapes that handles both aligned and misaligned fibers.
- Fast computation of the Voronoi diagram and α-shape of circles and ellipses using an approach specifically designed to exploit the characteristics of FRP composites. It is especially efficient when the majority of the fiber cross-sections are of similar size.
- Validation of our method on real-world microscope images. Our algorithm robustly detects fiber-deficient areas in real-world microscope images containing 100,000 fibers in 3.5 s, and 1,000,000 fibers in 70 s (excluding the time taken for circle and ellipse detection, which depends strongly on the resolution of the microscope image).

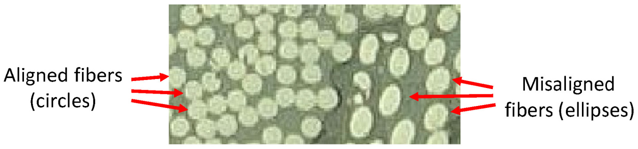

Fig. 2 Aligned fibers, with circular cross sections, and misaligned fibers, with elliptical cross sections.

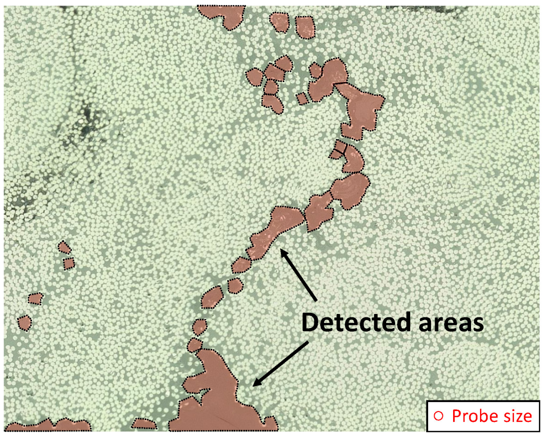

Fig. 3 Fiber-deficient areas detected by our algorithm, using as input the microscope image shown in Fig. 1(a). The probe radius used here is the mode of the detected fiber radii.

2 Related work

2.1 FRP 3D printing and defect detection

FRP composite materials are widely used in the aerospace, automotive, and many other industries. To reduce waste and handle intricate geometries, cutting edge methods use additive manufacturing techniques to fabricate FRP composite parts.

During development of FRP 3D printing processes, defect detection is an important tool to help researchers analyze the printed parts and improve their part quality. For FRP composites, common defects include voids, inclusions, and fiber-deficient areas [10]. Detecting voids and inclusions are both relatively well-studied in the material science community, and are now supported by image processing software such as ImageJ [11].

Compared to voids and inclusions, the detection of fiber-deficient areas is both more complex and less well-studied. A few researchers have attempted to automate estimating the location and distribution of fiber-deficient areas by utilizing Delaunay triangulations [7] or their dual, Voronoi diagrams [12, 13]. The main insight of these approaches is that spatial tessellations using the center points of the fibers can identify large tessellated regions as fiber-deficient areas. A limitation of such methods is that they treat fiber cross-sections as identical circles, ignoring variation in fiber diameters and ellipticity, which impairs their accuracy [12]. Tuning their parameters to choose appropriate thresholds for what constitutes a fiber-deficient area can be particularly problematic for real-world cases with these issues. Further discussion and comparison with our approach is provided in Section 7.

These methods, as well as ours, require fiber detection as a preprocess. We used a MATLAB implementation of the state-of-the art for detecting non-broken circular and elliptical fibers [14]. Broken fibers have little strength, so we should treat them as absent: areas where they are present should be flagged as fiber-deficient. The time taken for this watershed-segmentation-based fiber detection and classification method depends on the resolution of the input image and the number of fibers. For our $18,000 \times 10,000$ pixel test images, it takes just over 10 min to identify all 1.15 million fiber cross-sections present, on an Intel Core i7-9700K CPU with 16 GB RAM (without using MATLAB GPU calls). This method can achieve an accuracy of over 99.9%.

2.2 α-shapes

The α-shape is a computational geometry concept originally introduced for point set inputs [15] that captures the shape that could be accessed by a probe of radius α without intersecting the inputs. Generalizing α-shapes from point inputs to spheres, Kim et al. [16] elucidated the relationship between the α-shape of input consisting of spheres of different radii, their Voronoi diagram, and its dual triangulation, and proposed efficient algorithms for their construction. Inspired by this approach, we have developed a new method to construct the α-shape of input comprising circles and ellipses, and use its complementary regions to determine the fiber-deficient areas.

2.3 Voronoi diagram of circles and ellipses

The Voronoi diagram is a powerful computational geometry concept with numerous applications in science and engineering [17, 18]. Voronoi diagrams of circles and ellipses are a well-investigated topic because the cross-sections of many natural and industrial objects are circular or elliptical, including fibers.

Multiple algorithms have been proposed to calculate the Voronoi diagram of circles. Kim et al. [19, 20] proposed an algorithm that uses the ordinary Voronoi diagram of the circle centers as a seed, and then updates its topology by a series of edge-flipping operations. Jin et al. [21] reported a sweepline algorithm that handles circle inputs in arbitrary locations. The input circles are allowed to intersect or even fully contain each other. Lee et al. [22] designed an efficient topology-oriented incremental algorithm that robustly constructs the Voronoi diagram of 100,000 input circles in seconds, while handling degenerate cases. Some approaches for constructing the 3D Voronoi diagram of spheres, such as region expansion [23] or GPU ray-casting [24, 25], have 2D analogs for efficiently computing Voronoi diagrams of circles.

Beyond circles, the method of Emiris et al. [26, 27] can construct Voronoi diagrams of both circles and ellipses. However, it takes about 60 s to process 200 input ellipses [27], and is slow for applications with larger numbers of inputs.

Currently, no existing methods can efficiently construct the exact Voronoi diagram of inputs with large numbers of both circles and ellipses. Some approaches are efficient for circle inputs [19–22], but are difficult to extend to ellipse inputs; others handle both circle and ellipse inputs [26, 27], but are slow for large numbers of inputs.

Our inputs are predominantly composed of circles of essentially the same size with very few exceptional ellipses and large-size circles, and we have designed an efficient method to calculate exact Voronoi diagrams of inputs having these properties.

3 Algorithm overview

Our approach uses α-shapes to evaluate the proximity among the cross-section of fibers in the microscope image, and detect the areas where fibers are locally deficient.

Figure 4 illustrates the steps of our algorithm:

0. (Preprocessing). Detect the locations and sizes of the fiber cross sections in the input microscope

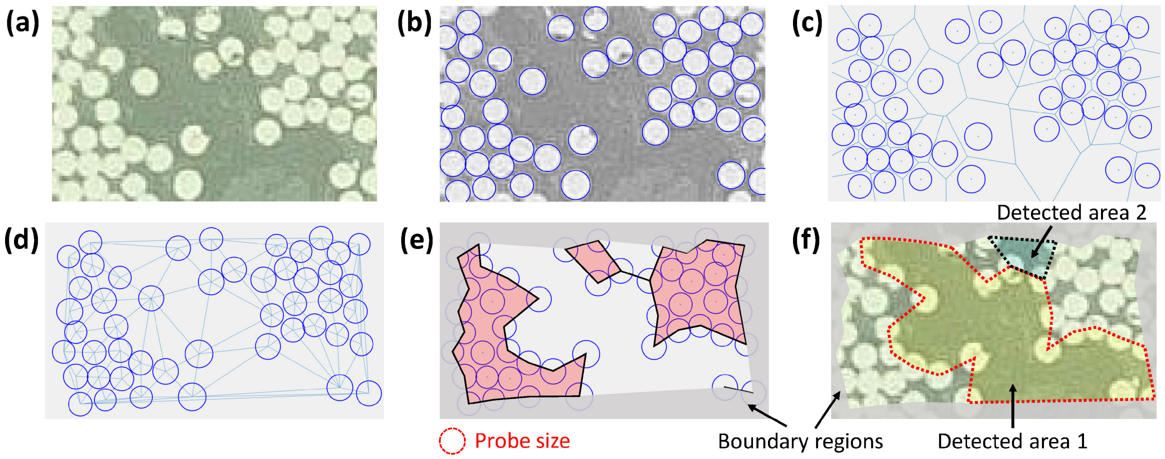

Fig. 4 Algorithm overview: (a) input microscope image, (b) circle detection, (c) Voronoi diagram, (d) dual triangulation, (e) α-shape in the refined region, (f) resulting fiber-deficient areas, the complement of the α-shape. The boundary region is the complement of the refined region, and is ignored.

image, using the watershed-segmentation-based algorithm in Ref. [14].

1. Construct the Voronoi diagram based on the fiber geometries.
2. Construct the dual triangulation from the Voronoi diagram.
3. Calculate the refined region to focus on (as well as its complementary boundary region). Build the α-shape from the dual triangulation in the refined region, identifying areas where fibers are close together.
4. Calculate the complement of the α-shape, giving the fiber-deficient areas. Further check and divide them into sub-areas if necessary.

Transverse cross-sectional microscope images of 3D printed FRP materials have the following characteristics:

- A typical image contains hundreds of thousands of fiber cross-sections.
- Most fibers are aligned, appearing as circles of about the same size, with a few appearing larger. Only a few fibers are misaligned, appearing as ellipses.

Our algorithm is designed specifically for these input characteristics: construction of the Voronoi diagram and α-shape is most efficient for inputs that are primarily circles, with a relatively small number of ellipses and circles of large radii.

4 Preliminaries

Given a set of objects $O = \{O_1, \ldots, O_n\}$ in $\mathbb{R}^2$, the Voronoi diagram of O is defined as the partition of the plane into n Voronoi cells, where each Voronoi cell is the set of all points closer to a particular input object O_i than to O_j $(\forall j \neq i)$.

Within the Voronoi diagram, Voronoi edges are common boundaries between two adjacent Voronoi cells. Any point on a Voronoi edge is equidistant to its two corresponding input objects. The intersections of Voronoi edges are Voronoi vertices. Each Voronoi vertex is equidistant to all of its corresponding input objects.

Because of this equidistance property, a Voronoi vertex always corresponds to the center of an empty circle that is tangent to all of the vertex's corresponding objects. Following the nomenclature for Voronoi diagrams of points, we call this a *circumcircle*, centered at the Voronoi vertex, having as its radius the distance between the Voronoi vertex and the corresponding objects. It does not overlap with the interior of any input object (see Fig. 5(b)).

The calculation of Voronoi vertex geometry is identical to the calculation of the circumcircle of a set of three corresponding objects, circles and ellipses in our case. We use the circumcircle calculations for circles and ellipses detailed in Refs. [20, 26] respectively.

The dual triangulation (see Fig. 5(c)) is the dual structure of the corresponding Voronoi diagram. It satisfies the following duality properties, illustrated in Fig. 5(d):

- Each Voronoi vertex corresponds to a bounded face in the dual triangulation. The centers of the

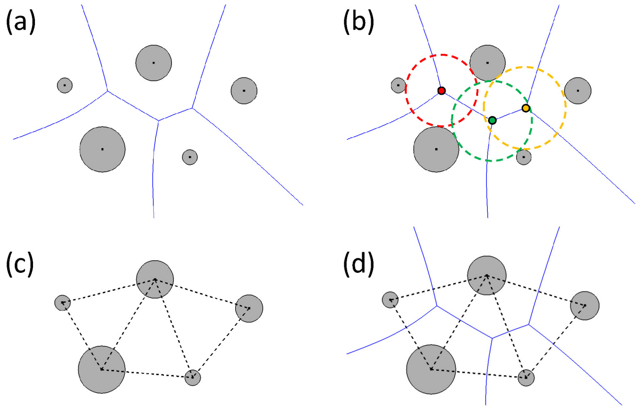

Fig. 5 Voronoi diagram and dual triangulation of a set of input circles with different radii: (a) Voronoi diagram, (b) Voronoi vertices and corresponding circumcircles, (c) dual triangulation of the same input, (d) superimposed Voronoi diagram and dual triangulation, illustrating their duality.

corresponding circles or ellipses of the Voronoi vertex are the dual triangulation face vertices.

- Each Voronoi edge corresponds to an edge (perpendicular to it for circle inputs) in the dual triangulation.
- Each Voronoi cell corresponds to a vertex in the dual triangulation. Vertices in the dual triangulation are centers of the input circles or ellipses.

We apply α-shapes to a set of two dimensional circle and ellipse inputs representing the fiber cross sections: imagine a probe with radius α moving around the whole 2D plane while not overlapping any of the inputs. The union of regions where the probe cannot reach is defined as the α-hull (see Fig. 6(a)). The curved edges on the boundary of the α hull are then replaced by straight edges to obtain the α-shape (see Fig. 6(b)). In our application, α-shape regions are places where fibers are close to each other, since the probe cannot be moved between them without colliding with the fibers, while complementary regions to the α-shape are fiber-deficient areas (see Fig. 6(b)).

The α-shape is constructed from the dual triangulation by first calculating the α-complex, the simplicial complex that is the subset (subcomplex) of the dual triangulation whose elements (triangular faces and edges) are inaccessible by the α-probe; the union of such elements is the α-shape (see Fig. 7). The details of this process are discussed in Section 6.

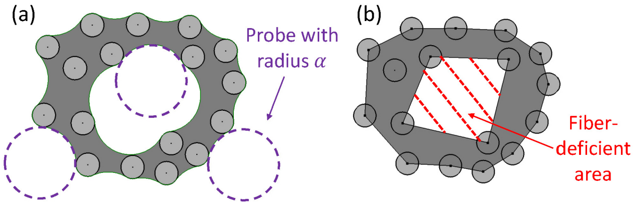

Fig. 6 (a) α-hull and (b) α-shape for circle inputs.

Note that the concepts of Voronoi diagrams and α-shapes were originally defined for point set inputs. Voronoi diagrams were then generalized to arbitrary objects, and these are sometimes referred to as *generalized Voronoi diagrams*, but often *generalized* is omitted. Similarly, although we generalize α-shapes to arbitrary objects, we just refer to them as α-shapes instead of generalized α-shapes.

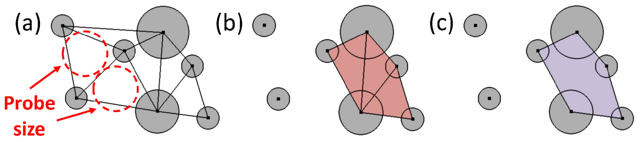

Fig. 7 Relationship between: (a) dual triangulation, (b) α-complex, (c) α-shape.

5 Construction of the Voronoi diagram

5.1 Approach

Preprocessing the microscope image provides a set of circles and ellipses representing the locations and sizes of the detected fibers. In the next step, we build the Voronoi diagram of these circles and ellipses.

Recall that most fibers have circular cross-sections with identical radius, with a few exceptions having larger radius or that are elliptical. For convenience, we will call the majority, identical circles *regular* input sites, and the exceptional size circles or ellipses *special* input sites. The input sites may slightly overlap each other, but none are fully contained within others (i.e,. there are no hidden sites).

The construction of the Voronoi diagram follows a simple idea: we first build a Voronoi diagram assuming all inputs are regular input sites. We then expand the Voronoi cell of each special input site, one after another, gradually transforming the preliminary Voronoi diagram to the final Voronoi diagram. We call this process the cell-expansion method.

Figure 8 shows an example of the serial cell-expansion process. The input consists of four regular sites and two special sites (see Fig. 8(a)). We first calculate the Voronoi diagram assuming all input circles are regular sites (see Fig. 8(b)). Constructing the Voronoi diagram of a set of identical circles is the

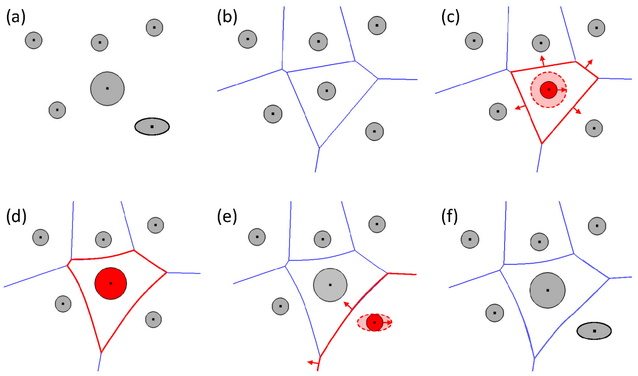

Fig. 8 Cell-expansion process: (a) input, (b) ordinary Voronoi diagram, (c, d) expanding the Voronoi cell of one special site, (e) expanding the Voronoi cell of the second special site, (f) result. The expanding input sites and Voronoi cells are in red.

same as constructing the ordinary Voronoi diagram of a set of points; in our implementation, we use the Triangle library by Shewchuk [28] to do so. This library uses a data structure that simultaneously represents the ordinary Voronoi diagram and its dual triangulation. Starting from the ordinary Voronoi diagram, we iteratively expand the Voronoi cells of each special site in random order, updating the cells' geometry and topology in the data structure (see Figs. 8(c)–8(e)), until all special sites have been processed (see Fig. 8(f)). Note that Voronoi edges that are straight line segments for equi-sized circle input generally become curved when their neighbors are special sites.

In Fig. 9, suppose we are extending the red circle and its Voronoi cell. Its *cell vertices* are v_1–v_4, and its *cell edges* are e_1–e_4. Voronoi edges connected to cell vertices, but not cell edges, are *radiating edges* of the expanding cell: e_5–e_8. Voronoi vertices connected to radiating edges but not cell vertices are *neighboring vertices* of the expanding cell: v_5 and v_6.

Some Voronoi cells are unbounded, with Voronoi edges having one or both or ends at infinity. We call such edges *infinite cell edges*; we treat them as connected to distinct cell vertices at infinity. An example of an unbounded Voronoi cell is shown in Fig. 10. Suppose we are expanding the red circle and its Voronoi cell. Then e_3 and e_4 are infinite cell edges connected to different cell vertices at infinity, v_4 and v_5 respectively.

As we increase the size of a particular input site, its corresponding Voronoi cell expands accordingly. Kim et al. [23] showed that for 3D Voronoi diagrams of spheres, during the Voronoi region expansion process, topology changes only occur at Voronoi vertices or edges. Thus, it is sufficient to only consider the status of the edges and vertices of the expanding Voronoi cell for topology changes. We state an analogous theorem for the 2D Voronoi diagram of circles and ellipses:

Theorem 1. *In a 2D Voronoi diagram of circles and ellipses, when a specific Voronoi cell is expanded, topology changes only occur at cell vertices, but not at cell edges.*

Proof Suppose the theorem is false, i.e., topology changes may also occur at cell edges. During the expansion process, such cell edges would intersect other Voronoi cells, generating new Voronoi vertices on the edge. Consider Fig. 11. As cell 1 expands, its cell edge (red) intersects another Voronoi cell (cell 2), locally dividing cell 3 into cells 3 and $3'$, and generating a new Voronoi vertex. As explained in Section 4, the new vertex has a corresponding

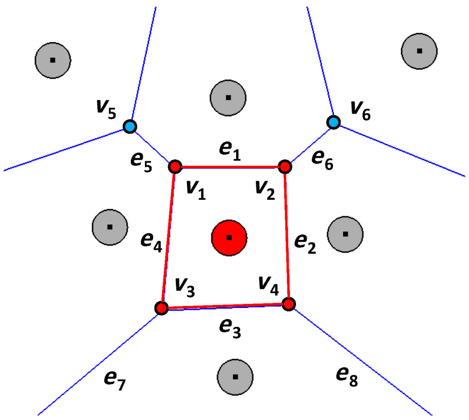

Fig. 9 An expanding Voronoi cell (red) with: cell vertices v_1–v_4, cell edges e_1–e_4, radiating edges e_5–e_8, and neighboring vertices v_5, v_6.

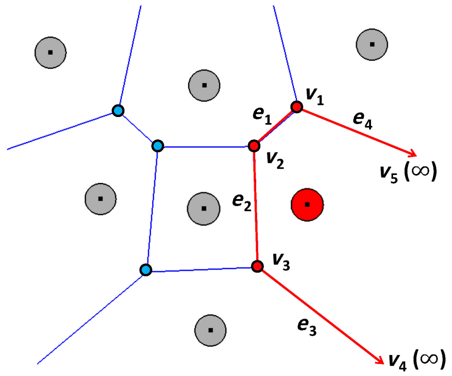

Fig. 10 An expanding unbounded Voronoi cell (red) with: regular cell vertices v_1–v_3, cell vertices at infinity v_4, v_5; regular cell edges e_1, e_2, and infinite cell edges e_3, e_4.

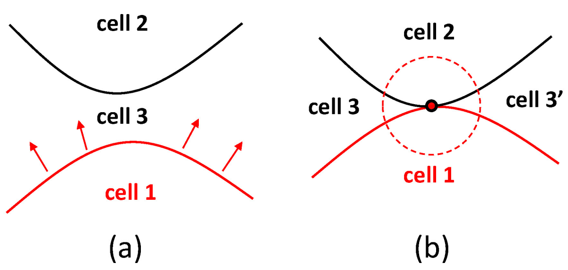

Fig. 11 Topology changes at cell edges during expansion. This situation only exists for non-convex inputs, and not for circle and ellipse inputs.

circumcircle that is tangent to the objects in each of its generating cells. So in this case, the input object of cell 3 must have tangent points to the circumcircle for both cell 3 and cell 3′. However, this contradicts our assumption of convex inputs (circles and ellipses), since a convex shape only has at most one tangent point to a circle. Hence, the theorem is valid. ■

During the serial cell-expansion process for all special sites, during the expansion of each special site, our algorithm follows a two-step process to detect potential vertex status and topology changes:

1. Check if the expanding site has infinite cell edges, which may generate new vertices.
2. Next, for each cell vertex, check its interaction with neighbors by tracing along radiating edges. Each vertex might collide with a neighbor vertex, disappear, or remain unchanged in its topology.

We now explain each step in detail.

5.2 Checking infinite cell edges

During the cell-expansion process, if the current expanding site has an unbounded Voronoi cell, we check all of its infinite cell edges to detect their potential intersections with infinite edges not belonging to the expanding cell.

In Fig. 12(a), consider infinite cell edge e_1. As the input site C_1 expands, e_1 moves towards its neighboring Voronoi cell corresponding to C_2 and potentially intersects with another infinite edge e_3. Edges e_1 and e_3 are shared by sites C_1–C_2, and C_2–C_3, respectively. If they intersect, there must be a new circumcircle generated by C_1, C_2, and C_3. Applying the robust methods described in Ref. [20] (or [26] for ellipses), we find circumcircles from sites C_2, C_3, and the expanded C_1. As illustrated in Fig. 12(b), in this case, the newly generated circumcircle is detected from the calculation. From the existence of the new circumcircle, we confirm the intersection of e_1 and e_2, and then update the geometry and topology of the Voronoi diagram (Fig. 12(c)).

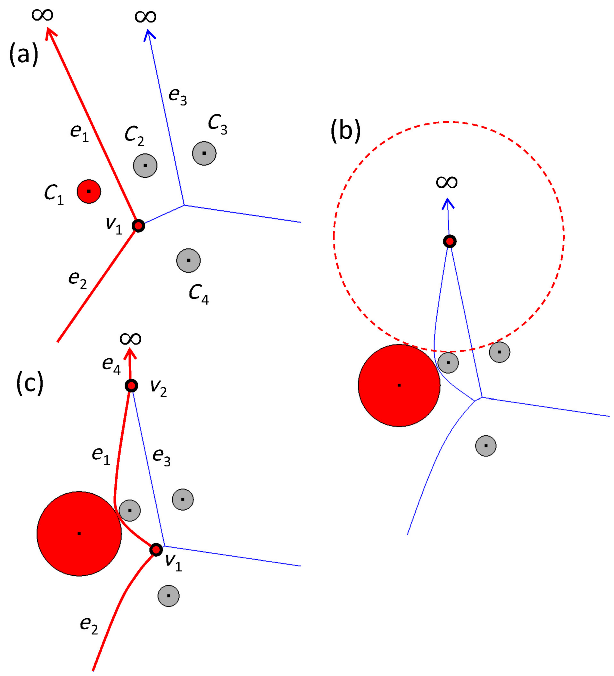

Fig. 12 An expanding unbounded Voronoi cell with infinite cell edges. (a) Before expansion, (b) a newly generated circumcircle exists, (c) after expansion, with a new generated vertex v_2 and edge e_4.

A disconnected edge is a special infinite edge both of whose ends extend to infinity. Following the process described above, we check for newly generated vertices at each of its ends. As shown in Fig. 13, new vertices may be generated at both of a disconnected edge's ends.

5.3 Tracing cell vertices along radiating edges

5.3.1 *Approach*

For each special input, after checking its infinite cell edges and updating the topology and vertex geometry accordingly, we obtain an updated list

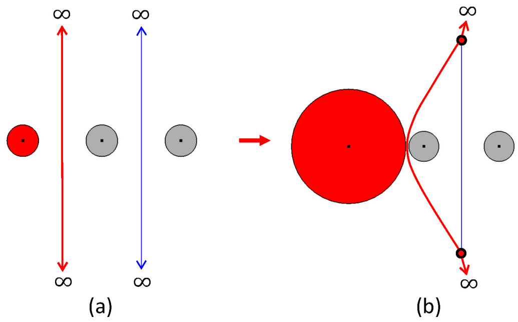

Fig. 13 Disconnected edges in the cell-expansion process

of its cell vertices. When a Voronoi cell expands, each of its cell vertices moves outward along its corresponding radiating edge because the vertex must remain equidistant from the two sites generating the radiating edge. In this second step of the cell expansion process, we trace each of the non-infinite cell vertices along its radiating edge, detecting potential topology changes. We consider two cases.

5.3.2 *Radiating edge extends to a neighbor vertex*

For each cell vertex we are tracing, if its radiating edge extends to a neighbor vertex, the cell vertex might collide with this neighbor vertex.

See Fig. 14(a). Assume we are analyzing cell vertex v_1 of the expanding site C_1. As site C_1 expands, v_1 moves along its radiating edge e_1 towards neighbor vertex v_2 (see Fig. 14(b)). No topology changes occur until v_1 collides with v_2. At the collision point, vertices v_1 and v_2 merge into a single cell vertex and radiating edge e_1 disappears (see Fig. 14(c)). After the collision, the new cell vertex splits into two cell vertices, which move along each of the newly adjacent radiating edges, generating a new cell edge e_2 (see Fig. 14(d)). This process is called *edge flipping* because it locally flips an old Voronoi edge (e_1) to a new one (e_2), with no other topology changes elsewhere.

We can perform a simple check for the collision between cell vertices and corresponding neighboring vertices by checking if the expanding site collides with the corresponding circumcircle of the neighbor vertex. In Fig. 14, neighbor vertex v_2 is the center of the circumcircle of its three corresponding sites C_2, C_3, and C_4. There is no topology change when the expanding site C_1 does not intersect the circumcircle (see Fig. 14(b)); two vertices degenerate to one when C_1 is tangent to the circumcircle (see Fig. 14(c), and a edge flip operation is needed when C_1 overlaps the circumcircle (see Fig. 14(d)).

Figure 15 shows a special case where the neighbor vertex is in non-general position (i.e., the vertex is shared by more than three sites). In this case, after the cell vertex collides with this neighbor vertex, new cell vertices are generated on each of the radiating edges from the original neighbor vertex, and new cell edges are generated between these vertices, proceeding in clockwise or counter-clockwise order.

After each of the topology change events described above, new vertices are generated. We iteratively check each newly generated vertex along its radiating edge, until no further topology changes occur.

5.3.3 *Radiating edge extends to infinity*

See Fig. 16. When a cell vertex v_1 has a radiating edge e_1 that extends to infinity, it moves along this radiating edge. During the expansion of the corresponding cell, the vertex v_1 may disappear at a critical point. At the critical point, the original radiating edge e_1 disappears along with the cell vertex v_1, and instead the two cell edges connected to this cell vertex (e_2 and e_3) become infinite at this end (see Fig. 16).

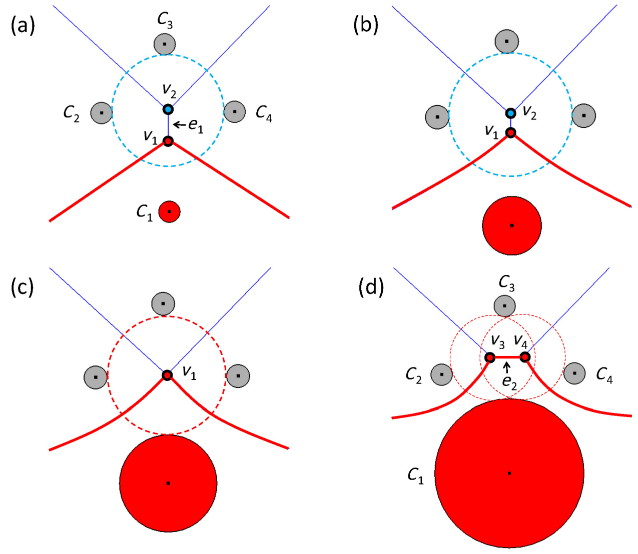

Fig. 14 During expansion of a special site (red), the cell vertex v_1 collides with neighbor vertex v_2, causing a topology change.

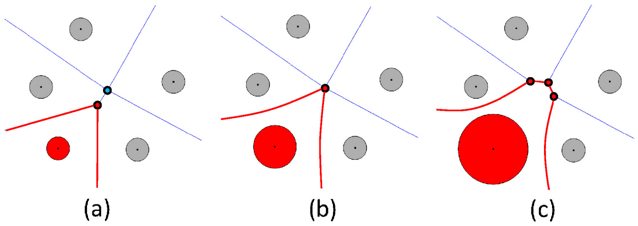

Fig. 15 A special case: a neighbor vertex in non-general position.

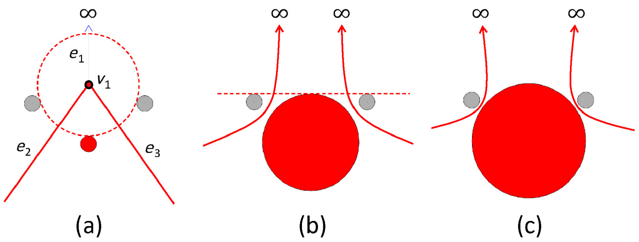

Fig. 16 During the cell-expansion process, a cell vertex vanishes when its radiating edge extends to infinity

We detect this topology change by calculating the circumcircle of the expanding cell vertex's corresponding input sites. If the circumcircle exists, the vertex is still moving along the radiating edge, and no topology change occurs (see Fig. 16(a)); the critical point occurs when all three corresponding inputs are tangent to a line (see Fig. 16(b)). Starting from this critical point, no circumcircles exist for the corresponding inputs, and the topology of the Voronoi diagram must be updated (see Figs. 16(b) and 16(c)).

5.4 Overall cell-expansion process

Figure 17 summarise our cell-expansion process. The input is the ordinary Voronoi diagram, which is then processed in two levels of iteration in our method.

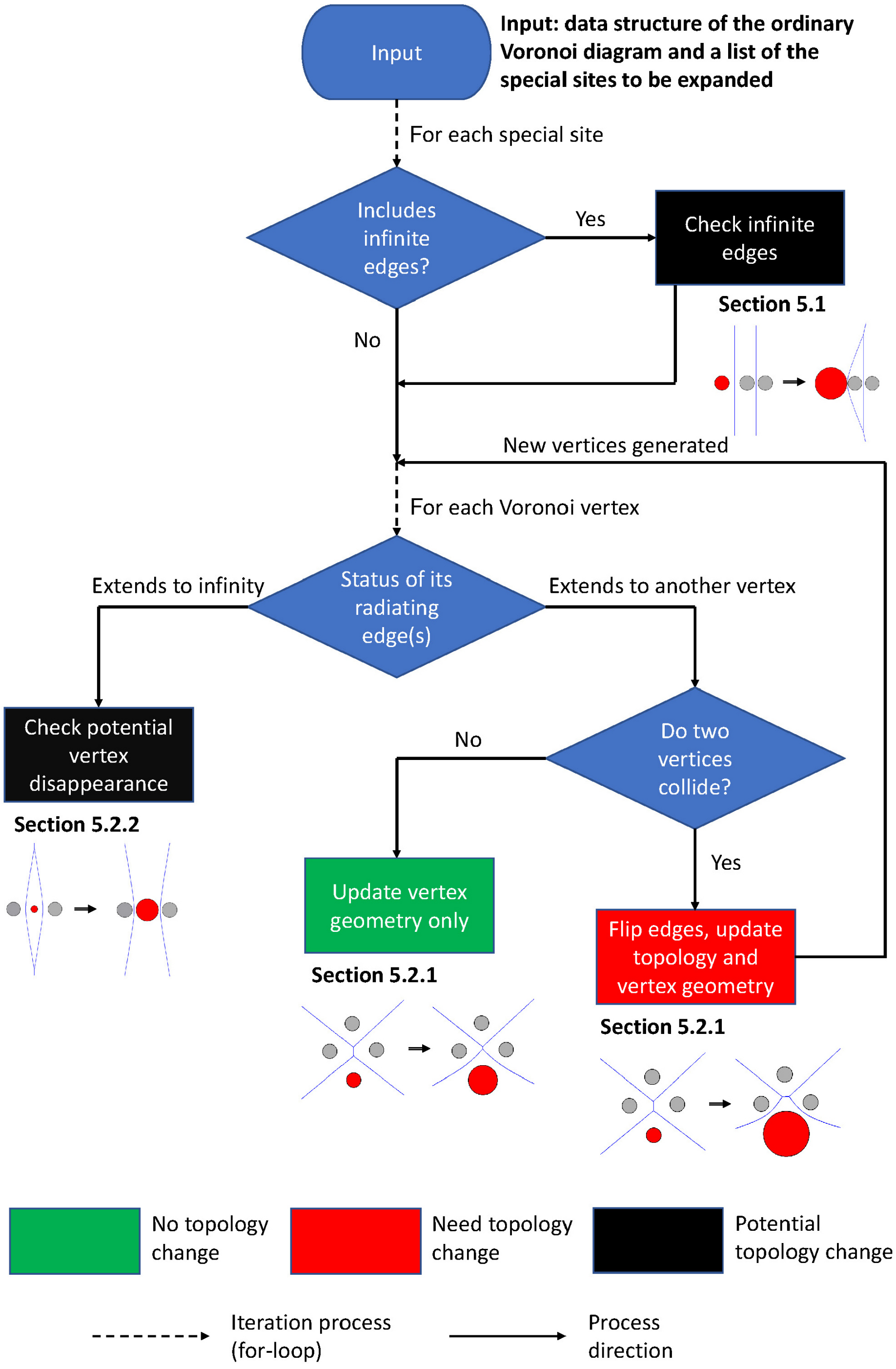

Fig. 17 Summary of the cell-expansion process

We iterate over the special sites to expand their corresponding Voronoi cells. For each expanding cell, we first check for topology changes in its infinite edges and then perform another level of iteration to detect other topology changes by tracing each cell vertex along its corresponding radiating edge(s).

During the process, we gradually update the topology and vertex geometry of the input Voronoi diagram until all special sites have been expanded. We do not calculate the curved edge geometry because it is not used in computing the α-shape in subsequent steps.

The cell expansion method works for both circle and ellipse inputs. Detecting topology changes is based on detecting the intersections between the expanding object and each of the corresponding circumcircles formed by its neighboring vertices.

6 Construction of the α-shape

6.1 Determining the refined region

After calculating the Voronoi diagram, we directly obtain the dual triangulation, as detailed previously in Section 4. As Fig. 20(a) shows, at the boundary of the dual triangulation, skinny triangles can occur. In a later step of constructing the α-shape (Section 6.3), we will determine if the α probe can be placed in a dual triangulation cell by comparing the size of its corresponding circumcircle with the α value. In that step, such a boundary skinny cell will be falsely detected as a fiber-deficient area because the size of its circumcircle does not indicate the actual size of the empty space between the three corresponding input objects. To avoid this miscalculation, we must cull these *boundary-artifact* skinny cells from the dual triangulation. After this culling process, we determine fiber-deficient areas on the remaining dual triangulation area (see Section 6.3).

In the dual triangulation, for any triangle having at least one edge at the boundary, we determine if it is a boundary-artifact triangle by checking if its corresponding circumcircle entirely stays outside this triangle. If so, the size of the circumcircle depends more on the boundary geometry than the actual empty space of the boundary-artifact triangle (see Fig. 18). We then cull all such boundary-artifact cells from the dual triangulation. We call the remaining part of the dual triangulation the *refined dual triangulation* (see Fig. 20(b)), and its corresponding region the *refined region*. The region outside the refined region is the boundary region (Fig. 20(c)), which is not considered in the following process.

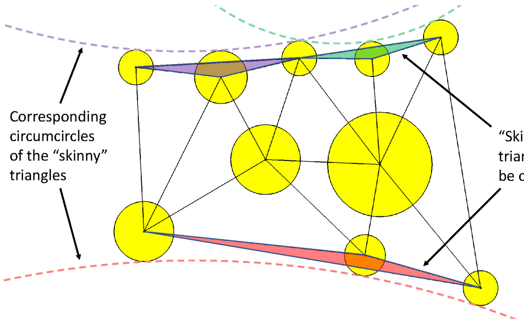

Fig. 18 Boundary-artifact triangles and their corresponding circumcircles.

6.2 Determining the threshold for α values

In the microscope image, if all fibers are evenly distributed, there can be no fiber-deficient areas. See Fig. 19(a). In this optimal situation, the fibers are evenly distributed in a hexagonal lattice, with the plane tiled by a triangular pattern with vertices located at the centers of the three neighboring fiber cross-sections. The largest possible probe (with radius α_{thresh}) that can be placed into this pattern is the circle centered at the center of the triangle, tangent to all of the three fiber cross-sections (see Fig. 19(b)). Any geometry change to the fibers will cause what could be considered fiber-deficient areas, where probes with radii larger than α_{thresh} can be inserted. Thus only probes with radii larger than the threshold probe size (α_{thresh}) could indicate fiber-deficient areas.

This (α_{thresh}) is thus a function of the fiber volume fraction v_{f} (the ratio of the volume of fibers to the total volume) as well as the fiber radius R. Analyzing the tiled triangular geometry (see Fig. 19(b)) to find the ratio of the fiber cross-section areas to the whole cross-section area in this ideal case, we set it equal

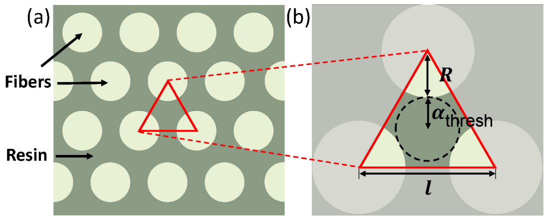

Fig. 19 Geometric relation between the threshold α value α_{thresh} and the radius of the fiber cross-sections R.

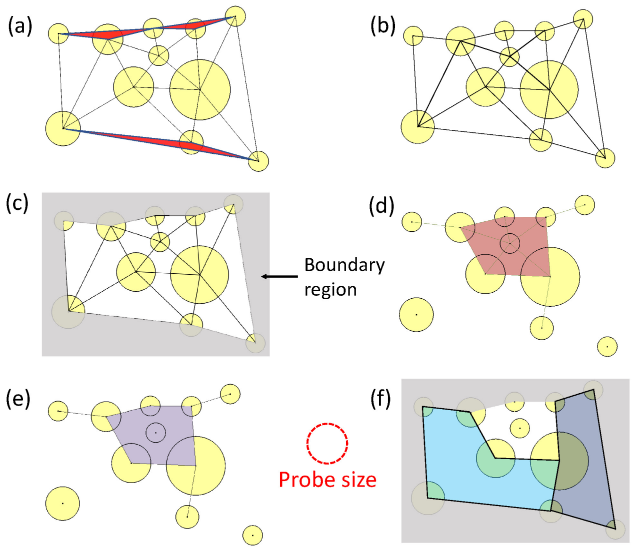

Fig. 20 Determining fiber-deficient areas from the dual triangulation: (a) dual triangulation, with boundary-artifact cells rendered in red, (b) refined dual triangulation, (c) boundary region, the complement of the refined dual triangulation, (d) α-complex, (e) α-shape, (f) detected fiber-deficient areas.

to the actual v_f for the manufacturing process and solve for α_thresh. Letting ℓ denote the side length of the equilateral triangle, we have

$$v_\mathrm{f} = \frac{\pi R^2}{\sin(60°)\ell^2} \tag{1}$$

For any equilateral triangle, the distance of its center from its vertices satisfies

$$R + \alpha_\mathrm{thresh} = \frac{\sqrt{3}}{3}\ell \tag{2}$$

Substituting for ℓ in Eq. (2) by an expression in terms of R and v_f from Eq. (1), we can derive the following relation:

$$\alpha_\mathrm{thresh} = \left(\frac{\sqrt{3}}{3}\sqrt{\frac{\pi}{\sin(60°)v_\mathrm{f}}} - 1\right)R \tag{3}$$

From the above equation, if the printed composite has, for example, nominal fiber volume fractions of 50%, 60%, or 70%, the threshold α value α_thresh is $0.555R$, $0.420R$, or $0.314R$ respectively. The choice of α_thresh in different applications is further discussed in Section 7.

6.3 Constructing the α-shape in the refined region

In Ref. [29], Kim et al. described a process for constructing the α-complex from the dual triangulation of circle inputs. We extend this idea to construct the α-complex of circles and ellipses from their refined dual triangulation. The process is has two parts.

First, for each edge in the refined dual triangulation, we check if the probe can traverse it by comparing the probe's diameter (2α) with the shortest length between the edge's corresponding objects (see Fig. 21). For a dual triangulation edge between circles, the shortest length equals the edge length (distance between circle centers) minus the radii of the two corresponding circles; for a dual triangulation edge between an ellipse and another circle or ellipse, the shortest length no longer has a simple relation to its edge length. We apply the method proposed by Zheng et al. [30] to calculate the shortest distance between two arbitrary ellipses, or a circle and an arbitrary ellipse. If the edge's corresponding shortest length is greater than 2α, the probe is able to freely traverse such an edge without colliding with either of the objects, so we keep this edge in the α-complex; if not, the probe is unable to traverse the edge, so we remove it from the α-complex.

Next, for each cell in the refined dual triangulation (connecting a triplet of circle centers), we check if the probe can be placed into it by comparing the diameter of the cell's corresponding circumcircle with 2α. This is the circumcircle we calculated when constructing the Voronoi diagram, the largest circle that can possibly be placed between the three corresponding inputs (see Figs. 5 and 7). If the circumcircle's diameter is smaller than 2α, the probe can not be placed in this cell, and we keep this dual triangulation cell in the α-complex; otherwise, the probe can be placed in the cell, and we remove the cell from the α-complex.

The α-complex (Fig. 20(d)) is a subset of the dual triangulation; it may contain dangling edges and interior voids. The above processes for checking edges and cells are separate: removing a cell does not necessarily mean its three corresponding edges will also be removed from the α-complex.

From the α-complex, we merge all neighboring cells,

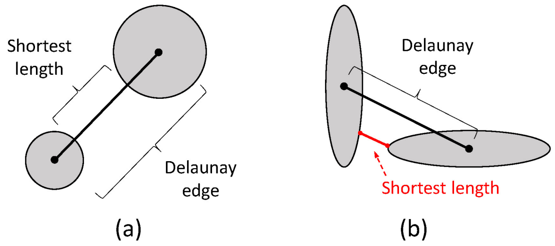

Fig. 21 Shortest length between: (a) two circles, (b) two ellipses.

and remove the edges between merged cells. The new structure is the α-shape (see Fig. 20(e)), indicating areas within which fibers are sufficiently dense that no probes could be placed there.

To find the fiber-deficient areas, in the refined region, we take the complement of the α-shape. If a complementary area is fully divided by the dangling edges in the α-shape, we treat its sub-areas as separate fiber-deficient areas (see Fig. 20(f)). It is necessary to distinguish such sub-areas because they do not form a fully connected region in which fibers are deficient. For example, see Fig. 22: although fibers are deficient inside each of the two detected sub-areas, there are enough closely spaced fibers at the boundary where they meet to increase its local strength. Since the fiber-deficient areas are not continuous across their locally reinforced contacting boundary, considering them as separate sub-areas helps in subsequent quality analysis and failure prediction processes.

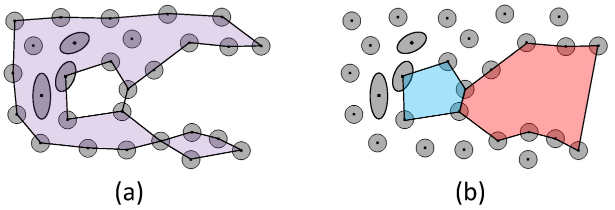

Fig. 22 A fiber-deficient area divided by a dangling edge: (a) α-shape, (b) two fiber-deficient areas.

7 Results

We have tested our algorithm on 30 high-resolution real-world microscope images from 3D printed FRP parts. Our images are all approximately 18,000 pixels $\times 10,000$ pixels, and contain about 1.15 million fiber cross-sections each, with the mode of the radii of fiber cross-sections around 7 pixels. Because of the inherent variability in microscopy and circle/ellipse detection processes, it is common to have a ± 1 pixel deviation in the detected circle radii or ellipse minor axes (for a misaligned fiber, the minor axis equals the original fiber radius). Therefore we use $\pm 15\%$ ($\pm\lceil 1/7 \rceil$) as our tolerance: if a fiber cross-section does not deviate by more than 15% from the mode of the detected radii R_{mode}, we consider it to have radius R_{mode} just like most other fiber cross-sections.

The proportion of expanded and misaligned fibers is highly dependent on the quality of the 3D printing process: poorer manufacturing leads to more expanded and misaligned fibers. In our test images, we found that no more than 1% of the fiber cross sections were larger size (expanded) circles ($radii > 1.15R_{\text{mode}}$) and no more than 0.008% were ellipses. No shrunken fibers ($radii < 0.85R_{\text{mode}}$) were found.

Our algorithm robustly handled all 30 real-world examples we tested, whether the fibers are circles or ellipses, in general position or not (see Figs. 3, 23, and 25). Figure 23 shows results on small cropped portions of microscope images with different levels of ellipse content ratio. The probe radius α is chosen as the mode of the radius of the detected fibers ($\alpha = R_{\text{mode}}$) in each test case.

A comparison between the fiber-deficient areas identified by our method and spatial-tessellation-based approaches [7, 12, 13] on the same circles and ellipses detected by pre-processing is shown in Fig. 24. Here, we apply the method described in Ref. [7] that constructs the Delaunay triangulation of the center points of each detected fiber, and identifies triangles with areas greater than a user-defined threshold as fiber-deficient areas. Since such methods ignore size and shape variation in fibers, they are unable to accurately detect regions where elliptical or varying-size circular fiber cross-sections exist (e.g., inside the yellow rectangles in Figs. 24(b) and 24(c)) regardless of threshold choice. In contrast, our method robustly detects fiber-deficient areas (see Fig. 24(d)) as it takes both misaligned fibers and aligned fibers of varying sizes into account.

Another advantage of our method over spatial-tessellation-based methods is the ease of parameter

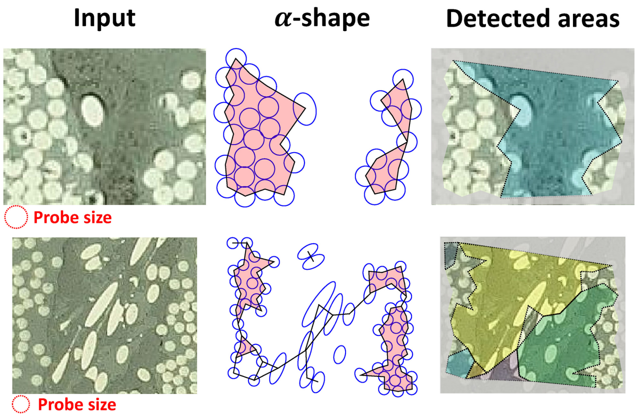

Fig. 23 Experimental results on two real-world microscope images. Separate fiber-deficient areas are distinguished by different colors; gray areas are boundary regions.

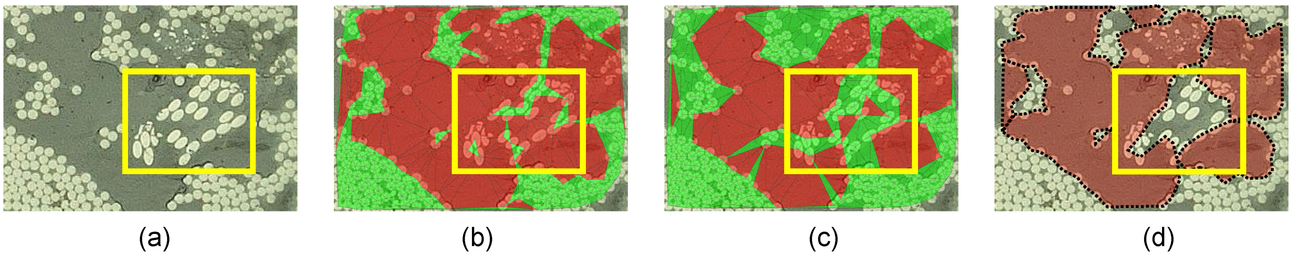

Fig. 24 Comparison between our method and a spatial tessellation (Delaunay) based approach [7]. (a) Input, (b) Delaunay approach with low threshold; (c) Delaunay approach with higher threshold, (d) our result. Yellow squares indicate regions with densely distributed misaligned fibers, which should not be identified as fiber-deficient areas.

value selection. In spatial-tessellation-based methods, since an appropriate threshold value depends on many factors such as image resolution and fiber distribution, and even triangle aspect ratio distribution, it is difficult for users to choose an appropriate threshold a priori. In our method, the α value can be intuitively understood as the size of the moving probe, and we have found that just using the mode of the detected radii R_{mode} or a small multiple thereof typically gives results that material experts find meaningful for analyzing defects.

Different choices of α result in different detected areas (see Fig. 25). Higher α values keep significant fiber-deficient areas and ignore tiny ones, useful when we only need to locate large defects such as inter-layer or inter-strip fiber-deficient areas. Lower α detects both significant and tiny fiber-deficient areas, which is preferred when we want to compile thorough statistics of fiber distribution in the part. For high-quality industrial FRP composite parts (typically 50%–70% nominal fiber volume fraction), depending on the application, α values ranging from R_{mode} to $3R_{\text{mode}}$ provide good inspection results.

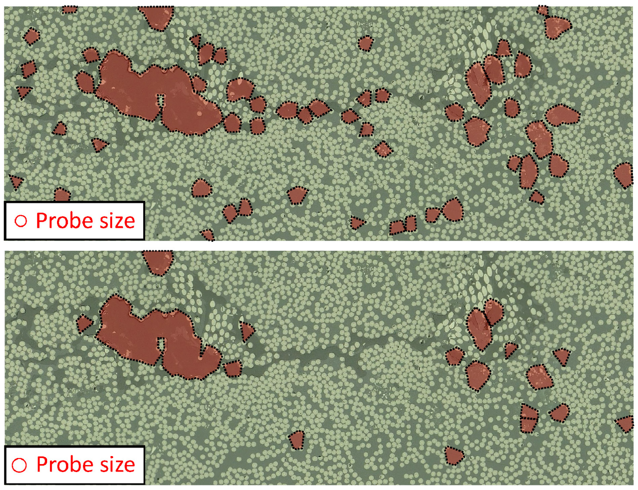

Fig. 25 Effect of changing α. Above: 55 detected fiber-deficient areas when $\alpha = 2R_{\text{mode}}$. Below: 18 detected areas when $\alpha = 2.5R_{\text{mode}}$.

To perform a complexity analysis, let n be the number of input sites (detected fiber cross-sections, including expanded and misaligned ones). It takes $\mathcal{O}(n \log n)$ worst case time and $\mathcal{O}(n)$ average time [31] to construct the ordinary Voronoi diagram of points and thus of the centers of detected fiber cross-sections. The cell expansion process of each large circle or ellipse takes time linear in the number of neighboring sites of the expanded cell. Since there might be $(n-1)$ large circles and ellipses, and each expanded cell may interact with all the other $(n-1)$ sites, the cell-expansion process takes $\mathcal{O}(n^2)$ time in the worst case. In most cases, the cell-expansion process takes $\mathcal{O}(n)$ time because in general Voronoi cells have few, $\mathcal{O}(1)$, neighboring sites. Constructing the dual triangulation from the Voronoi diagram only takes $\mathcal{O}(1)$ time because the dual is already captured in the Triangle software's data structure. From the dual triangulation, we compare the size/shape of each of the $\mathcal{O}(n)$ edges and cells to the size of the α-probe, and preserve those that are not traversable by the α-probe as the α-shape. Therefore, the total time complexity of our algorithm is $\mathcal{O}(n)$ for typical real-world inputs, even if $\mathcal{O}(n^2)$ in the worst case.

We have implemented our algorithm in C++, and tested our code on a PC with an Intel Core i7-9700K CPU with 16 GB RAM. To test the efficiency of our implementation, we randomly selected real-world microscope images and cropped them to give smaller images containing different numbers of fibers. Run time (exclusive of the fiber detection process) for different cropped images is shown in Table 1. We can observe that, although the calculation complexity depends on the number of expanded and misaligned fibers, the run time is roughly linear with regard to the number of detected fibers in these real-world inputs.

Table 1 Run time (excluding circle/ellipse detection) for real-world inputs with different numbers of detected fibers

Number of detected fibers	Number of expanded fibers	Number of misaligned fibers	Run time (s)
10,087	85	0	0.25
50,048	40	5	2.17
100,040	92	3	3.12
500,156	1418	20	25.67
1,000,144	4482	144	68.63

However, when the input includes a large number of misaligned fibers, the algorithm takes significantly more time, a limitation of our approach.

To test the sensitivity of our algorithm to the number of expanded or misaligned fibers, we artificially increased their content ratios by randomly picking regular detected circles and reshaping them into large circles or ellipses. For an example with 100,040 detected fibers, the algorithm's run time for different numbers of artificially expanded circles and ellipses is shown in Tables 2 and 3 respectively. Compared to the number of large circles, the number of ellipses has more impact on the total run time, due to the more complicated calculation of circumcircles and shortest distances for ellipses.

Table 2 Run time under different number of large circles within input containing 100,040 detected fibers

Number of large circles	Number of ellipses	Run time (s)
0	0	2.51
1,000	0	2.77
2,000	0	3.01
5,000	0	3.70
10,000	0	4.76

Table 3 Run time with different numbers of ellipses within input containing 100,040 detected fibers

Number of large circles	Number of ellipses	Run time (s)
0	0	2.51
0	10	3.84
0	20	4.28
0	50	6.78
0	100	10.94
0	500	53.96

8 Conclusions and future work

This paper has presented a new approach and algorithm to automatically detect fiber-deficient areas in microscope images of 3D printed FRP parts. It successfully handles real-world microscope images containing more than 1,000,000 fiber cross sections in 70 s, whether the fibers are aligned or misaligned, in general position or otherwise. We exploit the particular characteristics of fibers in 3D printed FRP parts in our design to considerably improve the efficiency over general-purpose geometric construction algorithms. Although the computation time is sensitive to the unpredictable number of expanded and misaligned fibers, in our real-world examples, it shows a roughly linear relationship to the number of fiber cross sections.

One potential approach to address this limitation would be to use an approximation when calculating circumcircles and shortest distances relating to expanded circles and ellipses, for example, approximating an ellipse by a set of boundary points [32], or a set of circles [18, 33]. Such approximate methods avoid complex exact calculations, but may lead to accuracy problems and bring additional complexity to the automatic detection process (e.g., in choosing the sampling/approximation density).

Since large fiber-deficient areas usually indicate inter-layer or inter-strip gaps, one area for future research is to automatically recognize boundaries between strips (as well as the separate individual strips) by connecting appropriate detected fiber-deficient areas (see Fig. 26). Recognition of individual strips will enable us to better evaluate the manufacturing process by determining the amount of intra-strip or inter-strip defects. For instance, substantial intra-strip defects usually indicate imperfections in the raw material, whereas

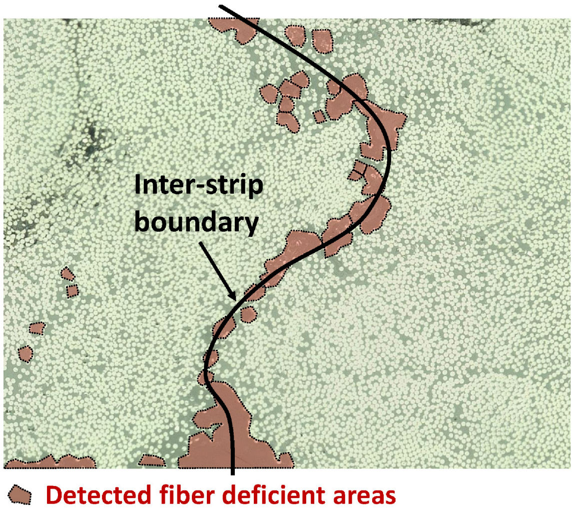

Fig. 26 Adjoining fiber-deficient areas along an inter-strip boundary.

substantial inter-strip defects usually indicate an excessive bead width in the 3DP setup.

Our algorithm works not only for circles and ellipses, but is also applicable to all convex shapes. Following the same process, we could extend this algorithm to construct Voronoi diagrams and α-shapes of other convex shapes by calculating circumcircles and shortest distances among them.

Acknowledgements

We thank Arevo Inc. and Danning Zhang for valuable discussions and providing the cross-sectional microscope images for testing, Sara Shonkwiler for feedback on the manuscript, the Voronoi Diagram Research Center at Hanyang University for their freely available Voronoi visualization software BetaConcept [29], and the anonymous reviewers for helpful suggestions for improving the manuscript.

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

References

[1] Yang, C. C.; Tian, X. Y.; Liu, T. F.; Cao, Y.; Li, D. C. 3D printing for continuous fiber reinforced thermoplastic composites: Mechanism and performance. *Rapid Prototyping Journal* Vol. 23, No. 1, 209–215, 2017.

[2] Van der Klift, F.; Koga, Y.; Todoroki, A.; Ueda, M.; Hirano, Y.; Matsuzaki, R. 3D printing of continuous carbon fibre reinforced thermo-plastic (CFRTP) tensile test specimens. *Open Journal of Composite Materials* Vol. 6, No. 1, 18–27, 2016.

[3] Zhang, J.; Fox, B. L.; Gao, D.; Stevenson, A. W. Inspection of drop-weight impact damage in woven CFRP laminates fabricated by different processes. *Journal of Composite Materials* Vol. 43, No. 19, 1939–1946, 2009.

[4] Parandoush, P.; Zhou, C.; Lin, D. 3D printing of ultrahigh strength continuous carbon fiber composites. *Advanced Engineering Materials* Vol. 21, No. 2, 1800622, 2019.

[5] Sugiyama, K.; Matsuzaki, R.; Malakhov, A. V.; Polilov, A. N.; Ueda, M.; Todoroki, A.; Hirano, Y. 3D printing of optimized composites with variable fiber volume fraction and stiffness using continuous fiber. *Composites Science and Technology* Vol. 186, 107905, 2020.

[6] Harish, S.; Michael, D. P.; Bensely, A.; Lal, D. M.; Rajadurai, A. Mechanical property evaluation of natural fiber coir composite. *Materials Characterization* Vol. 60, No. 1, 44–49, 2009.

[7] Ghayoor, H.; Marsden, C. C.; Hoa, S. V.; Melro, A. R. Numerical analysis of resin-rich areas and their effects on failure initiation of composites. *Composites Part A: Applied Science and Manufacturing* Vol. 117, 125–133, 2019.

[8] Hayes, B. S.; Gammon, L. M. *Optical Microscopy of Fiber-Reinforced Composites.* ASM International, 2010.

[9] Edelsbrunner, H.; Kirkpatrick, D.; Seidel, R. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory* Vol. 29, No. 4, 551–559, 1983.

[10] Cantwell, W. J.; Morton, J. The significance of damage and defects and their detection in composite materials: A review. *The Journal of Strain Analysis for Engineering Design* Vol. 27, No. 1, 29–42, 1992.

[11] Schneider, C. A.; Rasband, W. S.; Eliceiri, K. W. NIH Image to ImageJ: 25 years of image analysis. *Nature Methods* Vol. 9, No. 7, 671–675, 2012.

[12] Gommer, F.; Endruweit, A.; Long, A. C. Quantification of micro-scale variability in fibre bundles. *Composites Part A: Applied Science and Manufacturing* Vol. 87, 131–137, 2016.

[13] Yang, H.; Colton, J. S. Quantitative image processing analysis of composite materials. *Polymer Composites* Vol. 15, No. 1, 46–54, 1994.

[14] Li, X.; Shonkwiler, S.; McMains, S. Fiber recognition in composite materials. In: Proceedings of the IEEE International Conference on Image Processing, 2623–2627, 2021.

[15] Edelsbrunner, H.; Mücke, E. P. Three-dimensional alpha shapes. *ACM Transactions on Graphics* Vol. 13, No. 1, 43–72, 1994.

[16] Kim, D. S.; Cho, Y.; Sugihara, K.; Ryu, J.; Kim, D. Three-dimensional beta-shapes and beta-complexes via quasi-triangulation. *Computer-Aided Design* Vol. 42, No. 10, 911–929, 2010.

[17] Park, C. H.; Lee, S. Y.; Hwang, D. S.; Shin, D. W.; Cho, D. H.; Lee, K. H.; Kim, T.-W.; Kim, T.-W.; Lee, M.; Kim, D.-S.; et al. Nanocrack-regulated self-humidifying membranes. *Nature* Vol. 532, No. 7600, 480–483, 2016.

[18] Choi, S.; Ryu, J.; Lee, M.; Cha, J.; Kim, H.; Song, C.; Kim, D.-S. Support-free hollowing with spheroids and efficient 3D printing utilizing circular printing motions based on Voronoi diagrams. *Additive Manufacturing* Vol. 35, 101254, 2020.

[19] Kim, D. S.; Kim, D.; Sugihara, K. Voronoi diagram of a circle set from Voronoi diagram of a point set: I. Topology. *Computer Aided Geometric Design* Vol. 18, No. 6, 541–562, 2001.

[20] Kim, D. S.; Kim, D.; Sugihara, K. Voronoi diagram of a circle set from Voronoi diagram of a point set: II. Geometry. *Computer Aided Geometric Design* Vol. 18, No. 6, 563–585, 2001.

[21] Jin, L.; Kim, D.; Mu, L. S.; Kim, D. S.; Hu, S. M. A sweepline algorithm for Euclidean Voronoi diagram of circles. *Computer-Aided Design* Vol. 38, No. 3, 260–272, 2006.

[22] Lee, M.; Sugihara, K.; Kim, D. S. Topology-oriented incremental algorithm for the robust construction of the Voronoi diagrams of disks. *ACM Transactions on Mathematical Software* Vol. 43, No. 2, Article No. 14, 2016.

[23] Kim, D.; Kim, D. S. Region-expansion for the Voronoi diagram of 3D spheres. *Computer-Aided Design* Vol. 38, No. 5, 417–430, 2006.

[24] Hu, Z. Y.; Li, X.; Krishnamurthy, A.; Hanniel, I.; McMains, S. Voronoi cells of non-general position spheres using the GPU. *Computer-Aided Design and Applications* Vol. 14, No. 5, 572–581, 2017.

[25] Li, X.; Krishnamurthy, A.; Hanniel, I.; McMains, S. Edge topology construction of Voronoi diagrams of spheres in non-general position. *Computers & Graphics* Vol. 82, 332–342, 2019.

[26] Emiris, I. Z.; Tzoumas, G. M. Exact and efficient evaluation of the InCircle predicate for parametric ellipses and smooth convex objects. *Computer-Aided Design* Vol. 40, No. 6, 691–700, 2008.

[27] Emiris, I. Z.; Tsigaridas, E. P.; Tzoumas, G. M. Exact Voronoi diagram of smooth convex pseudo-circles: General predicates, and implementation for ellipses. *Computer Aided Geometric Design* Vol. 30, No. 8, 760–777, 2013.

[28] Shewchuk, J. R. Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator. In: *Applied Computational Geometry Towards Geometric Engineering. Lecture Notes in Computer Science, Vol. 1148*. Lin, M. C., Manocha, D. Eds. Springer Berlin Heidelberg, 203–222, 1996.

[29] Kim, J. K.; Cho, Y.; Kim, D.; Kim, D. S. Voronoi diagrams, quasi-triangulations, and beta-complexes for disks in R2: The theory and implementation in BetaConcept. *Journal of Computational Design and Engineering* Vol. 1, No. 2, 79–87, 2014.

[30] Zheng, X. Y.; Palffy-Muhoray, P. Distance of closest approach of two arbitrary hard ellipses in two dimensions. *Physical Review E* Vol. 75, No. 6, 061709, 2007.

[31] Ohya, T.; Iri, M.; Murota, K. Improvements of the incremental method for the Voronoi diagram with computational comparison of various algorithms. *Journal of the Operations Research Society of Japan* Vol. 27, No. 4, 306–337, 1984.

[32] Sugihara, K. Approximation of generalized Voronoi diagrams by ordinary Voronoi diagrams. *CVGIP: Graphical Models and Image Processing* Vol. 55, No. 6, 522–531, 1993.

[33] Lee, M.; Fang, Q.; Cho, Y.; Ryu, J.; Liu, L. G.; Kim, D. S. Support-free hollowing for 3D printing via Voronoi diagram of ellipses. *Computer-Aided Design* Vol. 101, 23–36, 2018.

Xiang Li is currently a Ph.D. candidate in the University of California, Berkeley, where he received his master degree. He has his bachelor degrees from both Shanghai Jiao Tong University and the University of Cincinnati. His research interests include computer aided design, computer vision, and computational materials science.

Sara McMains is a professor of mechanical engineering at the University of California, Berkeley. She received her B.A. degree in computer science from Harvard and her M.S. and Ph.D. degrees in computer science both from the University of California, Berkeley. Her research interests include computational geometry, computer graphics and visualization, geometric and solid modeling, CAD/CAM, geometric design for manufacturing feedback, geometric dimensioning and tolerancing, additive manufacturing, computer vision, and machine learning.