Struct Multidisc Optim (2017) 55:1641–1654
DOI 10.1007/s00158-016-1597-9

RESEARCH PAPER

Identifying boundaries of topology optimization results using basic parametric features

Guilian Yi[1,2] · Nam H. Kim[2]

Received: 17 August 2015 / Revised: 7 September 2016 / Accepted: 15 September 2016 / Published online: 15 October 2016

Abstract Topology optimization yields an overall layout of a structure in the form of discrete densities or continuous boundaries. One of important drawbacks, however, is that a serious gap exists between the topology results (e.g., greyscale images with irregular shapes) and parameterized CAD models that are ready for subsequent optimization and manufacturing. Without the corresponding CAD model, topology optimization design is difficult to be interpreted for manufacturing, as well as to be utilized in subsequent applications such as section and shape optimization. It is considered the most significant bottleneck to interpret topology optimization results and to produce a parameterized CAD model that can be used for subsequent optimization. The objective of this paper is to extract geometric features out of topology designs for parameterized CAD models with minimal manual intervention. The active contour method is first used to extract boundary segments from topology geometry. Using the information of roundness and curvature of segments, basic geometric features, such as lines, arcs, circles and fillets, are then identified. An optimization method is used to find parameters of these geometric features by minimizing errors between the boundary of geometric features and corresponding segments. Lastly, using the parameterized CAD model, section optimization is performed for beam-like structures, and surrogate-based shape optimization is employed to determine the final shapes. The entire process is automated with MATLAB and Python scripts in Abaqus, while manual intervention is needed only when defining geometric constraints and design parameters. Three examples are presented to demonstrate effectiveness of the proposed methods.

Keywords Geometric Features Identification · Active Contour Method · Topology Optimization · Shape Optimization · Section Optimization

1 Introduction

As one of the most active research topics in the field of structural optimization, topology optimization yields an overall layout of materials (Sigmund and Maute 2013; Deaton and Grandhi 2014; Rozvany and Lewiński 2014). It can obtain robust results based on well-developed numerical approaches, and has received more and more research attentions recently because of its great potential of industrial applications (Pedersen and Allinger 2006; Zhou et al. 2011). Since many topology optimization approaches are element-based, where the initial design space is discretized by finite elements and the design variables are assumed to be constant within each element, it is efficient in computation and has been applied successfully for solutions of many industrial optimization problems (Schramm and Zhou 2006; Zhou et al. 2011; Zhu et al. 2015). However, one of important drawbacks of topology optimization is that, it is difficult to interpret topological results in terms of basic geometric features, such as straight lines, fillets, arcs and circles, for conventional manufacturing. Even though some useful techniques, such as manufacturing constraints (Zhou et al. 2002; Zegard and Paulino 2016) or feature size controlling (Guest 2009; Lazarov et al. 2016), can

✉ Guilian Yi
guilianyi2012@gmail.com

Nam H. Kim
nkim@ufl.edu

[1] Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul 151742, Republic of Korea

[2] Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611, USA

be applied in topology optimization to improve the manufacturability of engineering designs, a missing link still exists, in the computer-aided design system, between the development of a conceptual solution and the subsequent definition and/or manufacturing of the product layout and geometry. Specifically, without the corresponding computer-aided design (CAD) model of a topology optimization design, it is not only difficult to realize the design using conventional manufacturing methods, but also difficult to have seamless integration with subsequent applications such as section and shape optimization. Therefore, there are needs to identify geometric features out of topology design and to obtain simple and smooth shapes for cost-efficient design and manufacturing.

It is well known that the level-set-method-based topology optimization can provide geometries with smooth and continuous boundaries (Chen et al. 2010; Chen and Chen 2011; van Dijk et al. 2013). Normally the zero contour of the level set function are used to represent the structural boundaries, but some of them appear with a complex geometry with highly nonlinear boundaries. A creative work has been done by Suresh (2013), who directly used the topological sensitivity field as a level set function and seamlessly integrated topology optimization into SolidWorks software. The parameterized CAD model was easily constructed in SolidWorks with the information of the topological sensitivity-based level set function. Recently, Guo et al. (2014) used the level set function to describe the structural geometry, and conducted topology optimization based on the moving morphable components. Only rectangular shapes were considered on the current stage, and a curved geometry had to be approximated by several straight components.

The topology shapes obtained from the level set method have smooth boundaries, but these boundaries can be quite expensive (or even impossible) for the conventional manufacturing processes, which is preferable to use straight lines, fillets, arcs and circles. Besides, other alternative pixel-based topology optimization methods, such as the solid isotropic material with penalization (SIMP) method (Bendsøe 1989; Zhou and Rozvany 1991; Bendsøe and Sigmund 1999), are not able to yield topology results with smooth boundaries. Therefore, it is better to develop a general approach to extract boundary information in terms of basic geometric features, especially for the results provided by element-based topology optimization methods.

Several methods in computer graphics have been proposed for boundary detection and representation, such as the automatic generation of stick models (Garrido-Jurado et al. 2014) and the real-time continuous pose recovery of objects (Tompson et al. 2014), but they cannot find geometry identified with parametric boundaries, which is important and critical for the subsequent optimization. In fact, many approaches were proposed to identify geometric boundaries out of topology optimization results. For example, the image interpretation approach has been used by drawing the shape of the initial structure on top of the topology optimization result (Olhoff et al. 1991; Bendsøe and Rodrigues 1991; Bremicker et al. 1991), or using computer vision techniques to transform grey scaled topology image into realizable design (Chirehdast et al. 1994). These approaches require too many manual interventions in image processing, which makes it inefficient and undesirable. The density contour approach (Kumar and Gossard 1996; Youn and Park 1997; Hsu et al. 2001; Hsu and Hsu 2005) effectively interpreted boundaries by selecting a value of density threshold. The drawback of this approach is that several times of trial and error have to be done before a proper density threshold is chosen. Moreover, disconnected structural parts, rough surface, and/or very thin parts may occur in the density threshold based geometric models of complex structures (Li et al. 2015), which requires repeated model revisions and human interventions. Basic shape templates (Lin and Chao 2000), predefined simple geometric features (Yildiz et al. 2003), and parameterized-feature-based templates (Larsen 2010) are also used to determine the size and location of holes inside topology images, but a limited number of feature templates can be used, while increasing matching accuracy becomes a matter of great concern. Other geometric reconstruction approaches have a focus either on free-form surfaces or on complex surfaces, such as B-spline curves, T-spline curves, Bi-quartic surface splines, and non-uniform rational B-spline (Papalambros and Chirehdast 1990; Change and Tang 2001; Tang and Change 2001; Koguchi and Kikuchi 2006; Chacón et al. 2014). These types of surfaces have an advantage on describing complicated boundary shapes, but a disadvantage on expensive manufacturing cost using conventional subtractive manufacturing methods, such as milling and drillings.

Despite the abovementioned existing geometric boundary detection methods, orientated design optimization methods are developed to cope with the advanced 3D printing technology. Christiansen et al. (2015) presented a deformable simplicial complex-based optimization method to combine topology and shape optimization of 3D structures. The explicit representation of structural shapes used a single mesh for shape representation to implement discrete and continuous optimization sequentially. Sutradhar et al. (2015) proposed a novel multiresolution topology optimization method for designing 3D printed craniofacial implants in clinical surgeons. The prototypes of the reconstructive surgeon-based optimization designs were fabricated using a 3D printer for validation. It is worth noting that design optimization methods orientated towards 3D printers are candidate means to improve the manufacturability of engineering designs. However, the accessibility of 3D printers to the general manufacturing and prototyping facilities is still limited due to a low affordable rate, as well as a risk of wasting the existing traditional manufacturing resources. Nevertheless, not every 3D model is printable.

The limitation of previous methods is that it would be expensive to manufacture those complex surfaces, albeit not impossible. It withdraws our attention from advanced manufacturing technologies to simple and conventional manufacturing techniques. Thus, the contribution of the proposed work is to represent the boundary shapes of topology optimization results using simple geometric features, such as straight line, arc, fillet, etc., which are much easier and cheaper to manufacture using conventional methods. A parametric CAD model corresponding to the topology optimization result is constructed by using these simple features. In the process of design rendering, the section optimization or surrogate-based shape optimization is performed depending on the types of the structures, to fine tune the shapes/boundaries. The same optimization problem formulation (i.e. objective and constraints) in topology optimization is used in the section/shape optimization to retain the original design purpose. Motivated by the idea of reducing manual intervention, the entire process is automated with Matlab and Python script-based Abaqus software, while manual intervention is needed only when defining geometric constraints and design parameters during shape optimization. The flowchart of the proposed process is shown in Fig. 1. Obviously, geometric features identification is the most challenging and time-consuming part of the whole process. Our goal is to interpret geometric features out of a topology design and convert them to a structure ready for manufacturing.

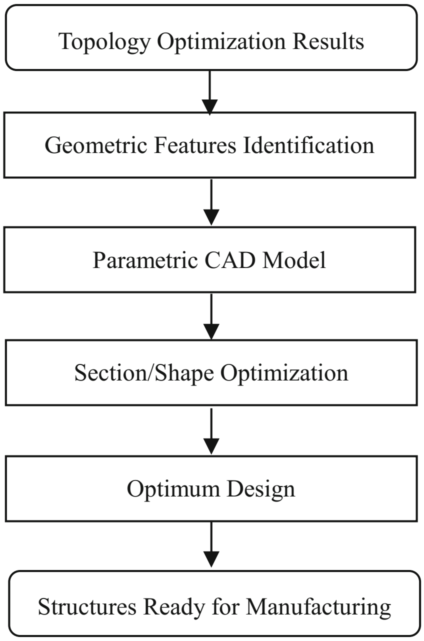

Topology Optimization Results
Geometric Features Identification
Parametric CAD Model
Section/Shape Optimization
Optimum Design
Structures Ready for Manufacturing

Fig. 1 Flowchart of parametric geometric feature identification and optimization from topology optimization result

In this paper the geometric features identification techniques are limited to two-dimensional structures. The rest of the paper is organized as follows. In Section 2, the basic idea of the proposed approach is explained in detail. The technique of extracting beam elements out of beam-like topology optimization results is described in Section 3. The design rendering process is discussed in Section 4. In Section 5, three representative examples are presented to illustrate the effectiveness of the proposed approach, followed by conclusions of this research and directions for future work in Section 6.

2 Geometric features identification

It is important to identify boundary features of topology optimization results for subsequent section/shape optimization and manufacturing. Although B-spline curves/surfaces can be used, they are expensive for conventional manufacturing. In fact, more than 90 % of machine components are composed of basic geometric features, such as straight lines, fillets and circles. Therefore, in this paper, basic geometric features are used to interpret topology optimization results and construct boundaries of geometry. In this section, the basic idea of geometric feature identification approach will be explained in detail.

2.1 Active contour method

Normally, topology optimization results are given in the form of greyscale images or pixel-based density values. The image functions in Matlab can easily transform the images or the density values into image data that can be used for the further processing. The topology results (i.e., topology configurations or pixel-based density values) can be reconstructed in Matlab and the size of meshing is retained. For example, topology optimization result with a 80×50 meshing size will be reconstructed as an image with 80×50 pixels in Matlab. Then, the image data are processed in this section to yield smooth and closed boundary points.

In this section, an active contour model, also called snakes, is used to connect nearby edges and localize the boundaries. Snakes are energy-minimizing splines guided by external constraint forces and influenced by image forces that pull them toward features such as lines and edges. They are often used to obtain a unified account of visual problems, including detection of edges, lines, and subjective contours, as well as motion tracking and stereo matching. Enriched descriptions about snakes algorithm and applications are presented in the literature (Kass et al. 1988; Chan and Vese 2001; Li et al. 2008). A Matlab code based on the snakes algorithm made by Su (2012) is employed here to extract geometry boundaries from topology optimization results. The snakes algorithm provides closed outer/inner boundaries that are composed of coordinates of the points on the boundaries.

As shown in Fig. 2, a greyscale image in Fig. 2a, topology optimization result of a cantilever beam based on 80 by 50 meshing, is imported into Matlab and interpreted into four closed geometric boundaries (the red boundaries) in Fig. 2b by using the snakes algorithm. The closed geometric boundaries are expressed by a sequence of points with coordinates, which are the blue cross points shown in Fig. 2c. All the closed geometric boundaries are given in order and each of them has the same starting and ending point. The red and green cross points in Fig. 2c are the first and second points for the corresponding closed boundary, and they indicate the boundary direction. Therefore, it is easy to separate closed geometric segments automatically by identifying the starting and ending points. The boundary points are called as "snakes points" in the following of this paper and used for the identification of geometric features.

2.2 Detection of circles

Among many geometric features, a circle is a common shape and it is different from straight lines, arcs and fillets. In the proposed geometric feature identification process, circles are identified first using "roundness", which is the measure of how close the shape of a closed boundary to that of a circle.

With the snakes points on a closed boundary, the centroid (x_0, y_0) of the boundary loop can be expressed as

$$(x_0,\ y_0) = \left(\frac{1}{n}\sum_{i=1}^{n} x_i,\ \frac{1}{n}\sum_{i=1}^{n} y_i\right) \tag{1}$$

where n is the number of snakes points in the closed boundary, and x_i and y_i are the coordinates of the i-th snakes point.

Considering a portion of a boundary loop as shown in Fig. 3, a small triangle is formed with two adjacent points and the centroid point. The base length d_i (red line) and height h_i (red dash line) of the triangle can be expressed as

$$d_i = \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \qquad h_i = \frac{|k_i x_0 - y_0 + y_i - k_i x_i|}{\sqrt{1 + k_i^2}} \tag{2}$$

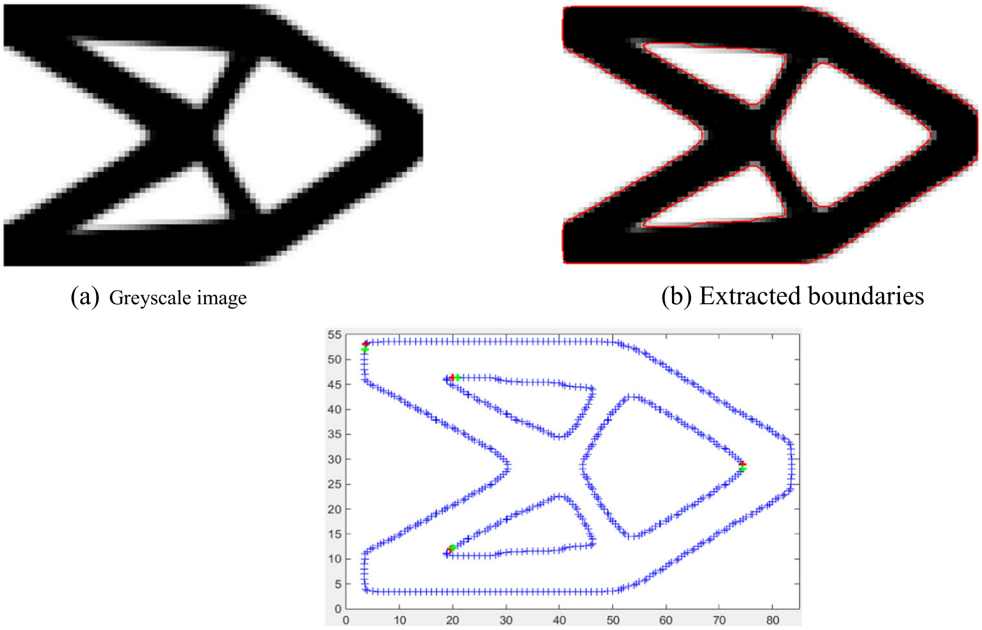
where $k_i = (y_{i+1} - y_i)/(x_{i+1} - x_i)$. Therefore, the perimeter P and area A of this boundary loop can be obtained as

$$P = \sum_{i=1}^{n-1} d_i \qquad A = \sum_{i=1}^{n-1} \frac{1}{2} d_i h_i \tag{3}$$

Thus, the roundness of this boundary loop can be computed by

$$m = \frac{4\pi A}{P^2} \tag{4}$$

where m represents the roundness of a closed geometric boundary loop. A roundness value close to 1.0 indicates that the object is approximately a circle. Figure 4 shows the centroids (red circle points) and roundness values for different geometric boundary loops. The roundness value is approximately 0.95 for a circle, 0.84 for an equilateral pentagon and above 0.5 for a triangle. Therefore, a closed geometric boundary loop is considered as a circle if its roundness is greater than a threshold of 0.9.



(a) Greyscale image

(b) Extracted boundaries

(c) Snakes points

Fig. 2 Image interpretation by using the active contour method

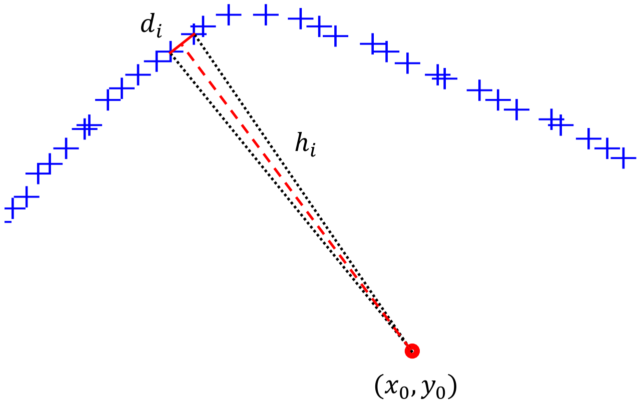


Fig. 3 A portion of the boundary

2.3 Identification of straight lines and fillets

After circles are identified by using roundness, other closed boundary loops are assumed to be composed of straight lines and fillets. Since the boundaries are represented by snakes points, slopes between points are used to determine if the portion of the boundary loop is a straight line or fillet. When the change in slopes is less than a threshold, the set of points is considered as a straight line. It is also assume that two straight lines are connected by a corner fillet.

Firstly, a set of *key points* are determined. The key points are approximately located where features are changed from straight line/fillet to fillet/straight line. They are shown as black circles in Fig. 5. Beginning with the starting point of a boundary loop, four slopes of the lines between the first point and the following four points are calculated. If these slopes change beyond a threshold, it is considered that the previous feature (i.e., straight line or fillet) may end at the point where a large slope change occurs, and the first point is marked as *a key point*. Otherwise, no key point can be found among the current five points and these points are considered to locate on the same feature. With the newly located key point, or the fifth point (when no key point is found), the slopes of next five points will be examined in the same way, until the corresponding boundary loop ends. The number and locations of key points can be tuned through the slope change threshold.

Secondly, a set of *control points* are selected. In order to ensure that the locations of straight lines and fillets capture the movements of the snakes points, two equal-distance points between two consecutive key points are selected from the snakes points. If two key points are very close, one middle point is selected instead. We call these selected points *control points*. Following the example shown in Fig. 5, the control points, as well as the key points, are shown in Fig. 6a.

Thirdly, straight lines are identified by evaluating the difference in slope between adjacent line segments. All key points and control points are connected by line segments in the order, and the slopes of the line segments are computed. The vertical lines can be identified through the coordinates of adjacent points, and the horizontal and other lines can be identified by comparing the slopes of adjacent line segments. A small slope difference of two adjacent line segments indicates that the corresponding points of these line segments are on the same straight line and the first and last points are set to be the two ending points of this straight line. The points at corners normally yield line segments with large slope differences, and they should be separated from the straight line points, to construct the fillets in the next step. But the straight lines share the same ending points with the fillets. As shown in Fig. 6b, straight lines are expressed with black lines passing through several black point in the middle and two red points at the ends. The fillet points are marked in red.

Lastly, fillets are identified between two straight lines. It is assumed that a corner is expressed with a fillet tangent to two straight lines. The corner points (red points) in Fig. 6b and the tangent property with adjacent lines are used to determine a fillet at a corner. The exact location and radius of a fillet are determined such that the fillet is tangent to two adjacent lines and has the minimum error with points in the corner portion. Figure 7 shows the geometric relationship of a fillet and two straight lines (red lines) at a corner boundary (blue cross points). The relationship is formulated in (5) to calculate fillet center and tangent points for a given fillet radius. In Fig. 7, n_1 and n_2 are, respectively, the numbers of two tangent points' locations on a closed geometric feature. The points between the n_1-th and n_2-th points are used to approximate a fillet tangent to two straight lines.

$$\begin{cases} \sqrt{(x_1 - x_c)^2 + (y_1 - y_c)^2} - r = 0 \\ \sqrt{(x_2 - x_c)^2 + (y_2 - y_c)^2} - r = 0 \\ (x_1 - x_c) + k_1(y_1 - y_c) = 0 \\ (x_2 - x_c) + k_2(y_2 - y_c) = 0 \\ k_1 x_1 + b_1 - y_1 = 0 \\ k_2 x_2 + b_2 - y_2 = 0 \end{cases} \tag{5}$$

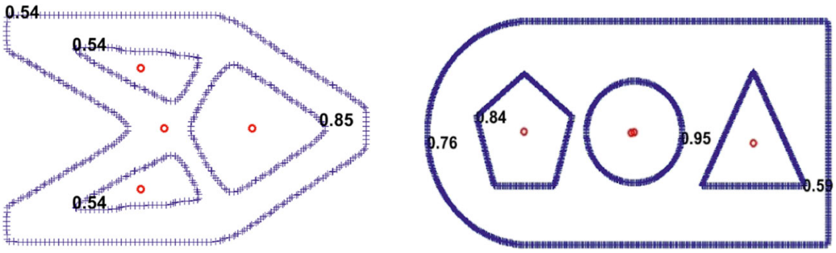


Fig. 4 Roundness values of different shapes

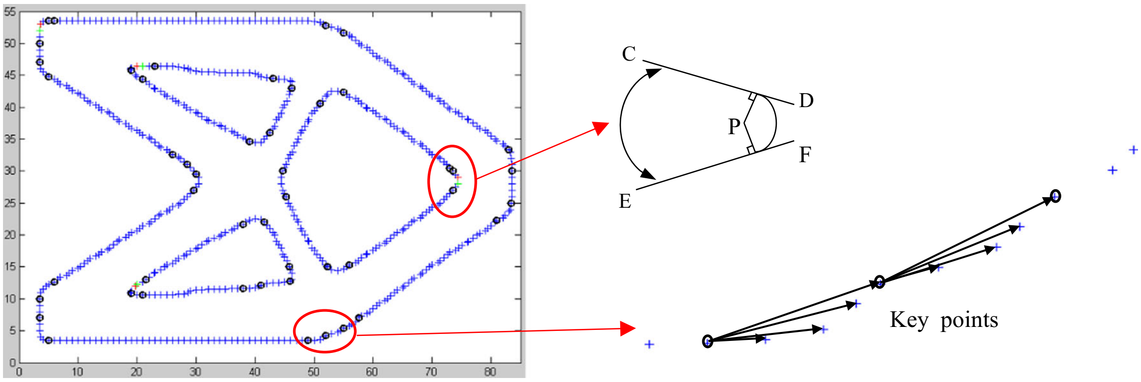


Fig. 5 Selection of key points using slope change

In (5), the subscript "c" stands for the center. "1" and "2" stand for the straight line 1 and straight line 2, respectively. (x_c, y_c) denotes coordinates of the fillet center point. r is the fillet radius. (x_1, y_1) and (x_2, y_2) are two tangent points, k_1 and k_2 are the corresponding slopes of the two lines, and b_1 and b_2 are the y-intercepts of the two lines.

Several parameters are involved in the representation of a corner portion with a fillet, but the fillet radius is the most important parameter, because the fillet should capture the snakes points at the corner portion as close as possible. Here, optimization techniques, which is often used to provide promising solutions for feature extraction and object detection (Bouaziz et al. 2012; He et al. 2014), can be employed to fine tune the important parameters of the fillet. Thus, an optimization problem in (6) is formulated to minimize the maximum distance from the fillet to the snakes points.

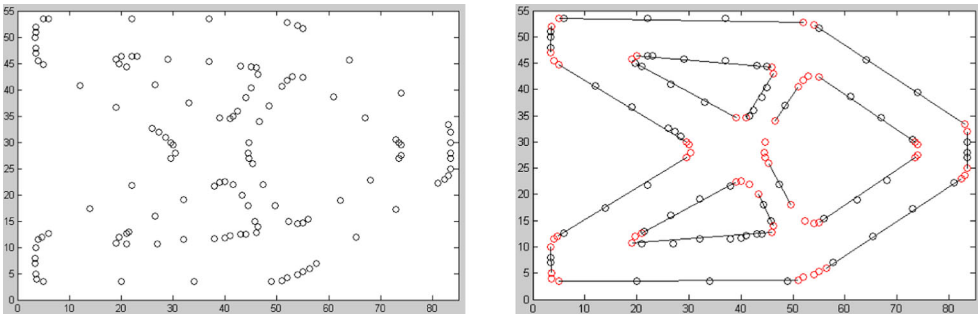
$$\begin{aligned} &\text{Find} \quad r \\ &\text{Min} \quad f(r) = \max_{i \in [n_1, n_2]} \left(\left| \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} - r \right| \right) \\ &\text{s.t.} \quad r > 0 \end{aligned} \tag{6}$$

Once the optimal fillet radius is obtained from (6), the fillet center point and two tangent points can be determined by solving (5). Figure 8a and b show the parameterized CAD models before and after fillet optimization with the snakes points. The blue points are the initial boundaries and the red lines are the identified boundaries. The tangent points and fillet center points are marked with red stars. It indicates that, the optimization process for fillets obviously reduces the errors between the identified boundaries and snakes boundaries; thus the final parameterized shapes can represent the initial topological shapes better. Arcs can be detected in the same way as fillets.

The above geometric feature detection procedures are implemented automatically in Matlab. The identified geometric features can be translated into a geometric model in any CAD software. A shape or size optimization will be used to fine tune the structure based on the CAD model.

2.4 Boundary detection of the L-shaped beam

Following the above procedures of the proposed method, the boundary extraction of the L-shaped beam problem is conducted as well, which is shown in Fig. 9. The L-shaped beam problem is well-known for stress-based topology optimization (Brampton et al. 2015). The optimized L-shaped beam obtained by using SIMP method is shown in Fig. 9a, in which the boundaries are completely zigzagged after applying the density threshold. The zigzagging boundaries, thin members, and multiple holes make the boundary extraction of this L-shaped beam structure difficult and complex.



(a) Control points

(b) Straight lines

Fig. 6 Identification of straight lines

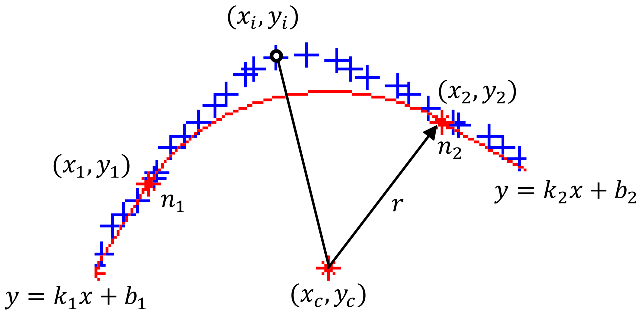


Fig. 7 Geometric relationship of a fillet and two straight lines

The snakes boundaries (red lines) on top of the initial topology result boundaries are shown in Fig. 9b. It can be seen that the snakes algorithm tracks the zigzagging boundaries very well. The whole structure is identified as six boundary loops by the snakes boundaries, including one external boundary and five internal boundaries. As each boundary loop is a closed boundary, straight lines and fillets are used to identify the boundaries for each loop separately in the following procedure. In order to better represent the zigzagging boundaries with straight lines, optimization of minimizing the distance between a straight line and the corresponding initial boundary section is necessarily performed for all straight lines. Figure 9c shows the extracted boundaries on top of the snakes boundaries before the fillet optimization, where the straight lines and fillets (including the fillet centers and start-and-end points) are shown in red, and the snakes boundaries are shown in blue. After performing the optimization for all the fillets, the final extracted boundaries are shown in Fig. 9d.

Compared with the L-shaped beam obtained by using the level set method in the work of Brampton et al. (2015), the L-shaped beam presented here has much more complex geometric features and characteristics (i.e., zigzagging boundaries and thin members). After using the proposed method for geometric feature detection, the degree of smoothing of the identified boundaries are comparable to that of the level-set boundaries. Furthermore, it is for certain that the boundaries composed of straight lines and fillets can be manufactured using conventional manufacturing methods easily.

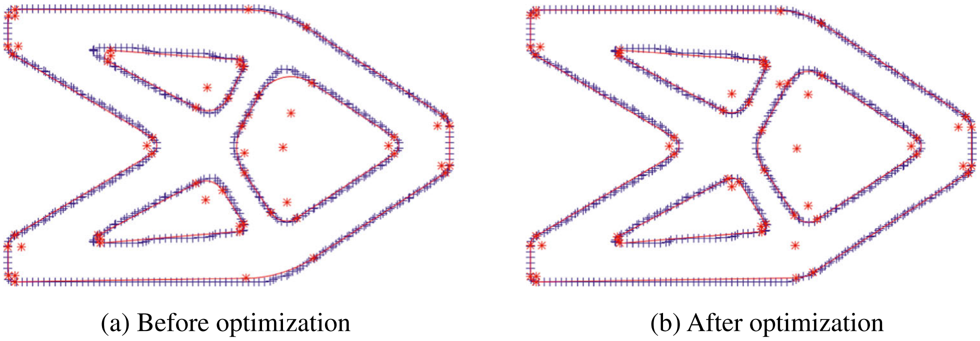
3 Identification of beam-like structures

The proposed geometric feature identification approach can be applied to any two-dimensional (2D) topology optimization result. Since the beam-like structure is different from solid-like structures, a special issue of beam identification is discussed here. Based on the parameterized CAD model provided by the geometric feature identification approach, the following procedures are implemented to model beam-like structures.

Step 1, find the parallel straight lines between different geometric features;
Step 2, find the middle line between two parallel lines;
Step 3, merge multiple intersection points into one;
Step 4, form the beam-like structure with middle lines.

Specifically, for example, the topology optimization result of a cantilever beam structure yields a parameterized CAD model shown in Fig. 10a, where four closed geometric boundary loops are named as Loop 1, Loop 2, Loop 3, and Loop 4, and each straight line (also line segment) is labeled within the corresponding boundary loop. These labels and loop numbers are used to identify straight lines from different boundary loops. Starting from the inner boundary loops, we calculate the absolute slope differences between a straight line in the current boundary loop and each straight line in the other three boundary loops, as well as the distances from the middle point of the line segment in the current loop to the middle point of each line segment in the other loops. Two straight lines from different boundary loops that yield the smallest absolute slope difference and middle-point distance are considered as parallel lines. After all parallel lines are addressed for each inner boundary loop, the middle line between two parallel line segments is determined by locating the median of a trapezoid formed by the two parallel lines. The parallel line segments and middle line segments are, respectively, shown in red and green in Fig. 10b. With only these green middle lines, three new beam loops can be defined, and the sequence of the

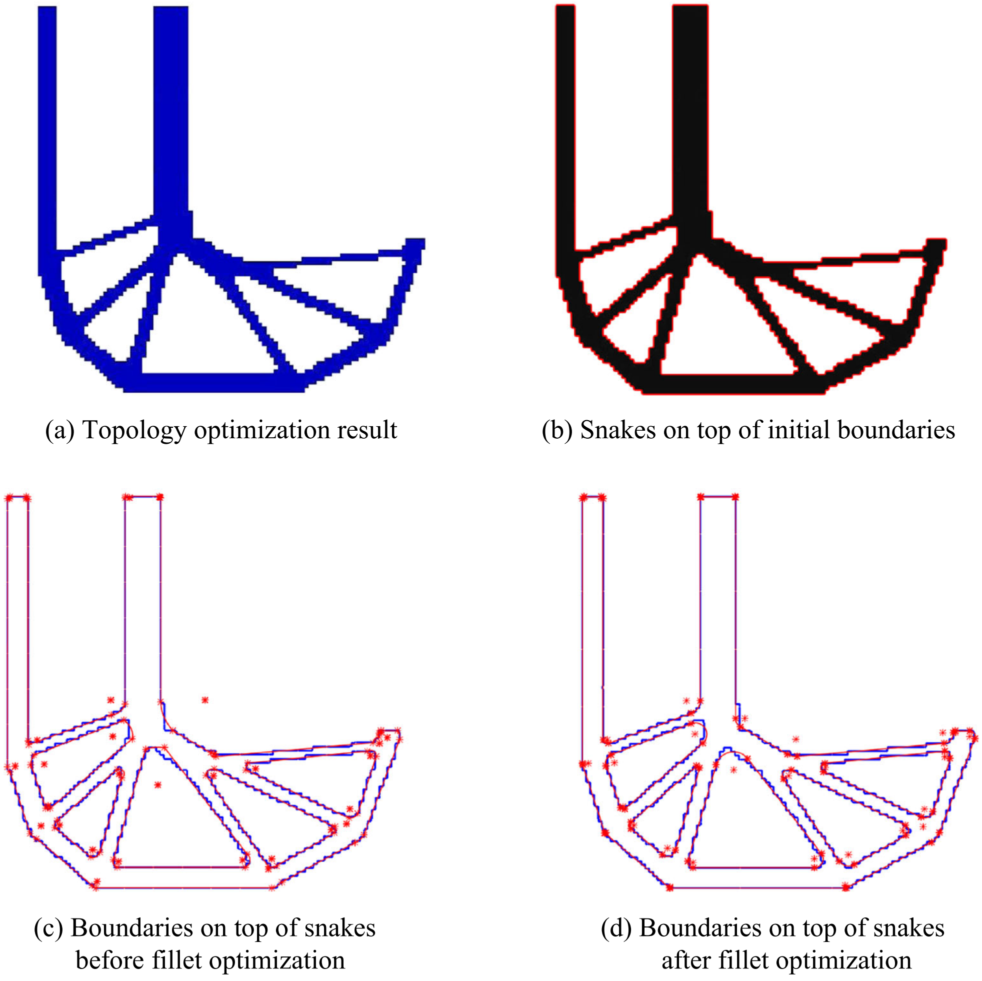
Fig. 8 Before and after optimization for fillets



(a) Before optimization

(b) After optimization

Fig. 9 Geometric feature identification of the L-shaped beam

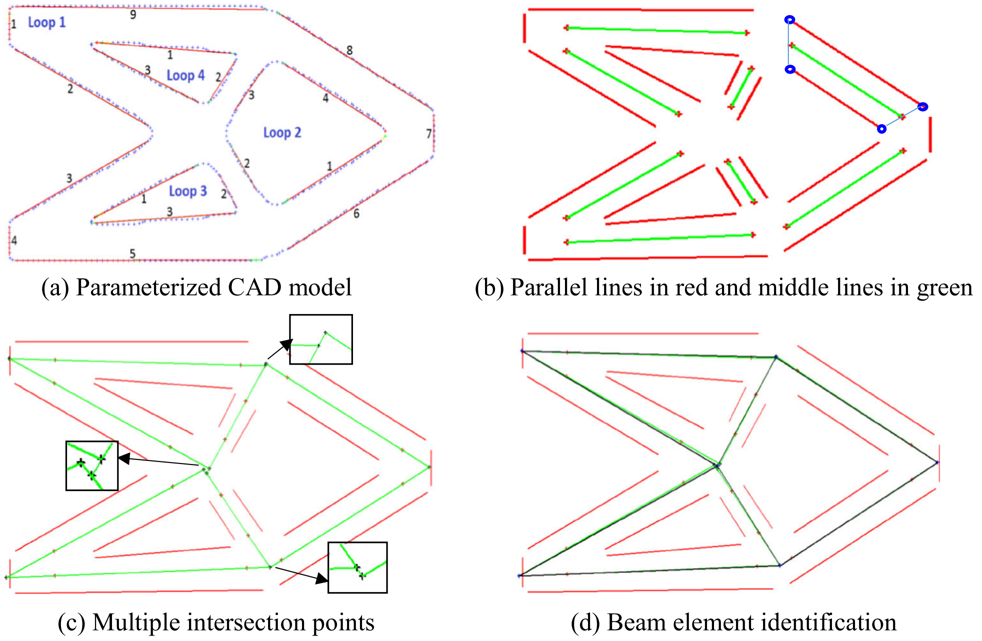

(a) Topology optimization result

(b) Snakes on top of initial boundaries

(c) Boundaries on top of snakes before fillet optimization

(d) Boundaries on top of snakes after fillet optimization

Fig. 10 An example of beam-like structure identification



(a) Parameterized CAD model

(b) Parallel lines in red and middle lines in green

(c) Multiple intersection points

(d) Beam element identification

middle lines in a beam loop follows the sequence of the corresponding straight lines in the inner boundary loop. Two different beam loops that share the same middle line should have the same straight line information (i.e., coordinates of ending points) of this middle line. Within a beam loop, every two middle lines in sequence meet each other and yield an intersection point by extending them. However, since the locations of these end points of middle lines are approximated from the topology optimization result, it is most likely that multiple intersection points may occur at the junction point where more than two middle lines cross, which can be seen in Fig. 10c. In order to merge multiple intersection points into one at the junction point of multiple middle lines, the average point of these intersection points is computed, and it is taken as the new intersection point of all the surrounding middle lines. As a result, middle lines are updated with the new intersection points, without shifting too much from the original locations, and they yield a beam-like structure as shown in Fig. 10d.

4 Design rendering

Although the geometric features are extracted from topology optimization results, they may not be considered an optimum geometry because the structural response can slightly be different from that of topology design. Therefore, a follow-up section or shape optimization is performed to fine tune the structural geometry for desired performances, which is called a design rendering process in this paper. However, the CAD-based shape optimization requires more information (i.e., mechanical performance and structural dimensions) than boundary geometries. All geometric features must be constrained such that when design variables change, all dimensions are updated properly.

It is worth to highlight that it is not allowed to make topological changes during shape optimization. The optimal designs are highly dependent on the shape and topologies that are obtained from topology optimization. Especially for engineering problems, the optimized structural performance can be heavily affected by subsequent topology changes due to geometry evolving in shape optimization. In some cases, the optimality of the solution from topology optimization can be utterly ruined. Thus it is of great importance to keep the beauty of topology design in the subsequent shape optimization. In order to prevent topology changes, member size control method is often performed in shape optimization (Seo et al. 2010; Le et al. 2011). In this paper, limits on the dimension changes are included in the constraints, and we determine the upper and lower bounds for the design variables manually when imposing full constraints on the geometry.

4.1 Fully constrained geometry

The state-of-the-art technology in shape optimization in industry is based on geometric features (dimensions, radius, etc.) on fully constrained geometry. Many engineers spend time to convert topology optimization results into geometric model with proper geometric constraints. This has been a major bottleneck in design automation. In the viewpoint of automating shape optimization, constraining geometry is difficult, and we consider this is out of the scope of this paper. Therefore, except for constraining geometry manually, all of the other process shown in Fig. 1 can be automated.

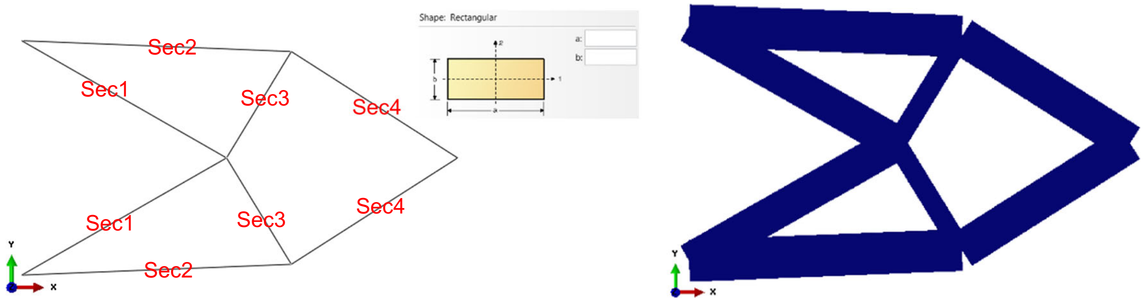
In this paper, Abaqus software is used for geometric features reconstruction and finite element simulation. With the geometric parameters of circles, straight lines, arcs and fillets, the geometric features can be constructed in Abaqus automatically by using Python scripts. Abaqus uses constraints to create relationships between dimensions and parameters. Constraining geometric features removes degrees of freedom in geometry. The fully constrained geometry can be used in shape optimization. It is noted that the process of constraining geometry depends on designer's intention, and there is no unique way of constraining geometry. Therefore, this process requires manual intervention. In addition, selecting shape design parameters also requires manual intervention. Except for this process, however, the proposed method is fully automated.

4.2 Design optimization

CAD-based design optimization, either section optimization or shape optimization, is implemented to fine tune the topology design using the extracted geometric features. The same optimization formulation used for topology optimization is used for the section or shape optimization. During the follow-up optimization, no new geometry features will be introduced, but the initial features will be modified to find the optimum design.

For beam-like structures, section optimization is applied. The assignment of beam orientation and cross section properties should be defined before optimization, and the cross section parameters will be selected as design variables. For solid-like structures, surrogate-based shape optimization is implemented for design rendering. As the dimensions change, the structural shape and mesh change, too. It causes numerical noise when the finite difference method is used to calculate sensitivity information. In order to address the issue of mesh-related numerical noise, the surrogate-based optimization approach is employed to perform shape optimization.

In design optimization, it is unnecessary to change all parameters; a limited number of dimensions can be selected as design variables. The lower and upper bounds for design variables need to be selected such that the geometry can be well



(a) Extracted cantilever beam and beam profile (b) Optimal widths for each beam section

Fig. 11 Section optimization of the cantilever beam

defined within these bounds. Since the selection totally depends on the experiences of designers, this parameterization process has to be done based on trial and error in Abaqus/GUI after fully constraining geometry.

Once the design optimization model is confirmed, all of the pre- and post- processes involved with Abaqus are implemented using Python scripts. The optimization problem is solved by the "fmincon" function in Matlab. The optimization for the CAD model is implemented by using Matlab to execute Python scripts in Abaqus. Thus the whole process of this research is implemented automatically with the least human intervention.

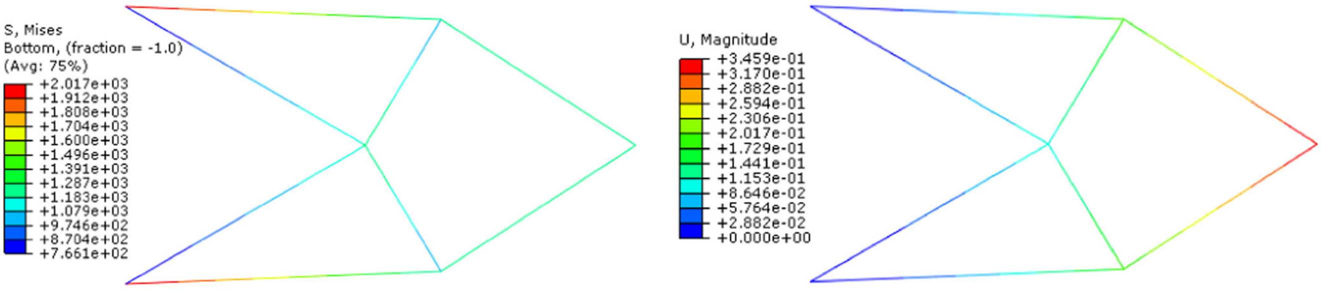
5 Numerical examples

In order to demonstrate the validity and capability of the proposed framework for geometric feature identification, three numerical examples are tested in this section. The first example is a cantilever beam structure identified using beam-like structures, the second example is the same cantilever beam identified using solid-like structures, and the third example is a clamped plate. Beam-element-based section optimization is performed in the first example, and surrogate-based shape optimization is considered for the second and third example.

5.1 Identification of beam-like structure and section optimization

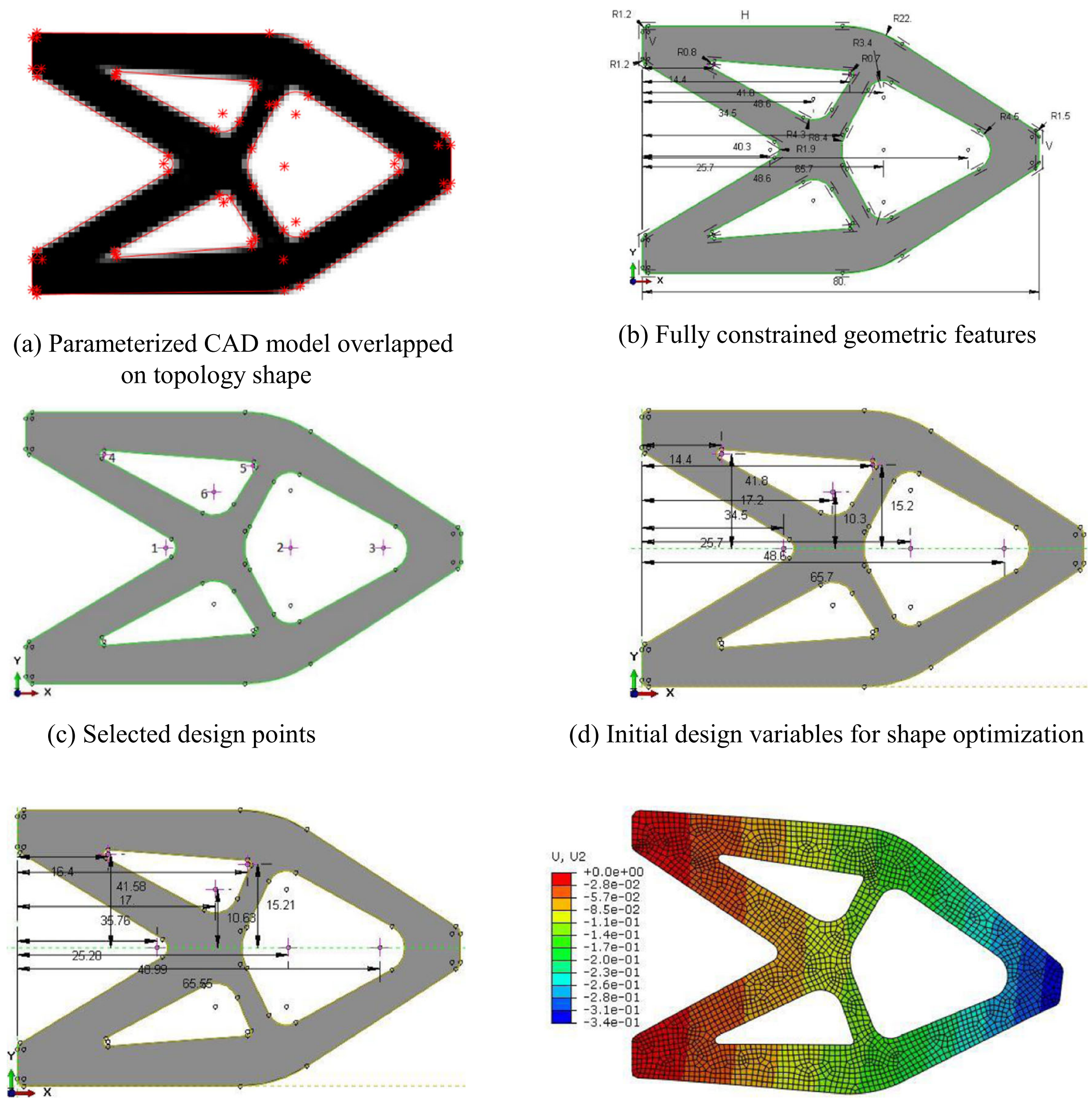
A cantilever beam is fixed on the left and a concentrated vertical load is applied at the middle of the right edge. Detailed structural dimensions and material properties are referred to the fourth example in the paper by Yi and Sui (2015). Initially, the topology optimization problem was formulated to minimize compliance with a volume fraction limit of 0.5. Computed by the 99-line topology optimization code (Sigmund 2001), the optimal topology shape of the cantilever beam structure was obtained and shown in Fig. 2a, which yielded the minimum compliance of 3139.53 /Pa and a volume fraction value of 0.5.

After implementing the geometric features identification techniques, the topology optimization result is interpreted into a beam-like structure, as shown in Fig. 11a. A section optimization of minimizing compliance with volume fraction constraint of 0.5 is formulated for this structure. Since the cantilever beam is symmetric, four section areas are selected as design variables. A rectangular section profile is defined for each of them. Since the thickness of the beam is 1.0 mm in topology optimization, the design variables are actually the widths of the rectangular four sections. A lower bound vector of lb = [0.1, 0.1, 0.1, 0.1] (mm) and an upper bound vector of ub = [10, 15, 10, 15] (mm) are given for the design variables.



(a) Stress contour (b) Displacement contour

Fig. 12 Stress and displacement contours of the optimal design



(a) Parameterized CAD model overlapped on topology shape

(b) Fully constrained geometric features

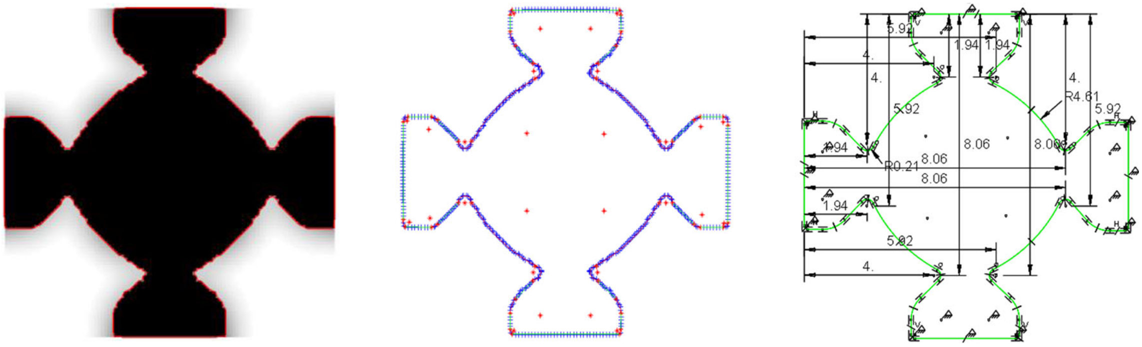
(c) Selected design points

(d) Initial design variables for shape optimization

(e) Optimized design after shape optimization

(f) Displacement contour of the final design

Fig. 13 Shape identification and optimization for the cantilever beam

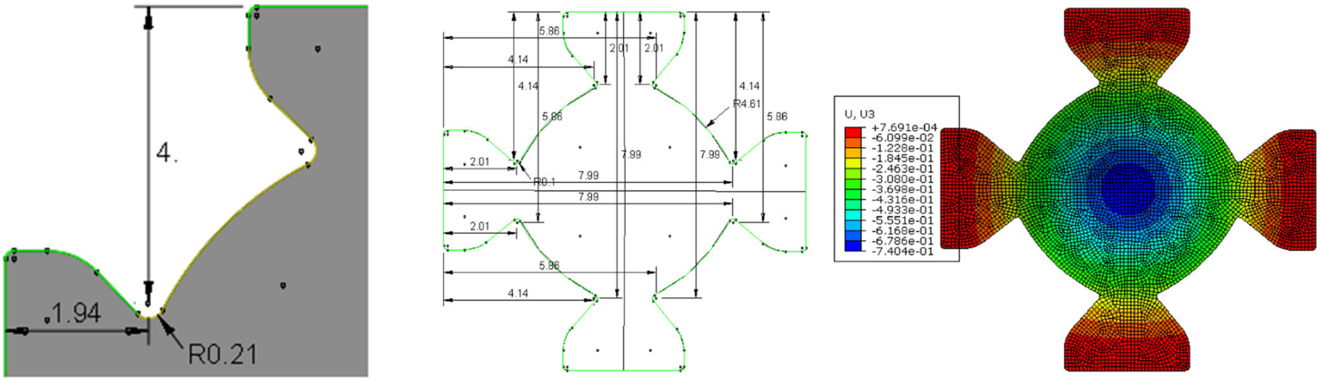


(a) Topology shape

(b) Identified boundaries

(c) Fully constrained CAD model

Fig. 14 Plate shape identification



(a) Design variables (b) Optimal shape (c) Displacement contour

Fig. 15 Plate shape optimization

The optimum design is found at **x*** = [6.50, 7.60, 3.67, 6.83] (mm), which are the widths of each section shown in Fig. 11b. The volume fraction of the final design is exactly 0.5, and the minimum compliance is 3112.65 /Pa, reduced by 0.86 % compared to the minimum compliance of 3139.53 /Pa obtained from the 99-line topology optimization code. The stress and displacement contours of this beam-like structure are shown in Fig. 12.

5.2 Shape identification and optimization for the cantilever beam

A topology optimization problem of minimizing mass with a displacement constraint limit of 0.34 mm imposed on the loading point was formulated for the same cantilever beam described in the previous example. Computed by the 120-line code (Yi and Sui 2015), the optimum topology shape was almost the same as that obtained from the compliance minimization problem as both produced the same displacement under the same loading and boundary conditions. The minimum mass of the optimized topology shape was 1990.17 kg, and the displacement at the loading point along the loading direction was equal to 0.34 mm.

The extracted and parameterized boundaries overlapped on the topology shape are shown in Fig. 13a, where the geometric features are expressed with straight lines and fillets. The fully constrained geometry with dimensions and constraints is shown in Fig. 13b, where manual intervention is required. The same optimization problem as that formulated in topology optimization is considered during shape optimization. Since the topology shape is symmetric along the horizontal line in the middle, the location of 6 points (shown in Fig. 13c) are selected for design variables. It is assumed that points 1, 2, and 3 can only move along x-axis, and points 4, 5, and 6 can move along x- and y-coordinate directions. The other three points symmetric to points 4, 5, and 6 are changed accordingly. The rest of the dimensions remain fixed. Thus, all 9 design variables are defined in the shape optimization. Figure 13d shows the design variables in the initial fully constrained CAD model. The Kriging surrogate model is used to express the relationships between the 9 design variables and structural mass / displacement at the loading point. The optimal design is obtained and shown in Fig. 13e. The final mass is 1919.42 kg, reduced by 3.55 % compared to the topology optimization result. The displacement constraint imposed at the loading point is satisfied, and the displacement contour is shown in Fig. 13f.

5.3 Plate shape identification and shape optimization

A square plate is fixed on all four edges and a concentrated load is imposed at the center of the plate in the normal direction. The displacement at the middle of the plate along the loading direction is required to be less than 0.74 m. According to the topology optimization result provided by Yi (2014) through Python scripts and Abaqus, the total mass of the optimal structure was 5.32 kg, and the displacement constraint was fully satisfied at 0.74 m.

Figure 14a shows the extracted boundaries (in red) overlapped on the topology shape (in black and grey). The parameterized CAD boundaries overlapped on the Snakes points are shown in Fig. 14b, where the geometric features are expressed with straight lines, fillets and arcs. The CAD model is reconstructed in Abaqus and the fully constrained geometry is shown in Fig. 14c.

Since the geometry is fully symmetric and the dimensions are related to each other, x- and y-cordinate of the fillet center point, and fillet radius, are considered as design variables x_1, x_2 and x_3, which are shown in Fig. 15a. The shape optimization with a displacement constraint is performed on the identified geometry. The Kriging surrogate model is utilized to approximate the relationships between three design variables and the mass / displacement at the center. The optimal design is found at **x*** = [2.01, 4.14, 0.1] (m), where the total mass is 5.00 kg, reduced by 6.02% compared to the topology optimization result. The displacement at the center point is 0.74 m, and Fig. 15b gives all dimensions of the optimum shape. The displacement contour of the optimal shape is shown in Fig. 15c.

6 Conclusions

The proposed approach based on the active contour method and optimization techniques allows representing the boundary of topology optimization results in terms of basic geometric features. The process of extracting boundary geometry is automated and the extracted geometry is imported to Abaqus finite element software to perform size or shape optimization. The later process is semi-automatic in a sense that the user has to select design variables and define optimization problem, but this intervention is inevitable. This research presents a design framework of integrating topology and section/shape optimization to produce best structural designs that are ready for manufacturing. Representative 2D topology optimization benchmark problems (i.e., a truss structure, a beam structure, L-shaped beam and a plate structure) are used to verify the proposed approach. The results demonstrate that the proposed method of geometric feature identification can interpret topology optimization results appropriately.

Geometric features used in this research are simple and good for conventional manufacturing process, but they are still not enough to represent complex geometries and 3D structures. Thus, in the future, more advanced geometric features will be exploited, such as quadratic curves, ellipse, extrusion, sweeping, and sphere. Scaling models and techniques will also be developed for automatic selection of relevant geometric features. Another perspective would be to parameterize boundary conditions to make process fully automatic.

Acknowledgments We gratefully acknowledge the financial assistance from the China Scholarship Council (2011–2013) and the Brain Korea 21 Plus project (2015–2016).

References

Bendsøe MP (1989) Optimal shape design as a material distribution problem. Struct Optim 1(4):193–202

Bendsøe MP, Rodrigues HC (1991) Integrated topology and boundary shape optimization of 2-D Solid. Comput Methods Appl Mech Eng 87(1):15–34

Bendsøe MP, Sigmund O (1999) Material interpolation schemes in topology optimization. Arch Appl Mech 69(9–10):635–654

Bouaziz S, Deuss M, Schwartzburg Y, Weise T, Pauly M (2012) Shape-up: shaping discrete geometry with projections. Comput Graph Forum 31(5):1657–1667. doi:10.1111/j.1467-8659.2012.03171.x

Brampton CJ, Dunning PD, Kim HA (2015) Topology optimization for stress using level set method. 56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Kissimmee

Bremicker M, Chirehdast M, Kikuchi N, Papalambros PY (1991) Integrated topology and shape optimization in structural design. Mech Struct Mach 19(4):551–587

Chacón JM, Bellido JC, Donoso A (2014) Integration of topology optimized designs into CAD/CAM via an IGES translator. Struct Multidiscip Optim 50(6):1115–1125. doi:10.1007/s00158-014-1099-6

Chan TF, Vese LA (2001) Active contours without edges. IEEE T Image Process 10(2):266–277

Change KH, Tang PS (2001) Integration of design and manufacturing for structural shape optimization. Adv Eng Softw 32(7):555–567

Chen S, Chen W (2011) A new level-set based approach to shape and topology optimization under geometric uncertainty. Struct Multidiscip Optim 44(1):1–18

Chen S, Chen W, Lee S (2010) Level-set based robust shape and topology optimization under random field uncertainties. Struct Multidiscip Optim 41(4):507–524

Chirehdast M, Gea H-C, Kikuchi N, Papalambros PY (1994) Structural configuration examples of an integrated optimal design process. J Mech Des 116(4):997–1004

Christiansen AN, Bærentzen JA, Nobel-Jørgensen M, Aage N, Sigmund O (2015) Combined shape and topology optimization of 3D structures. Comput Graph 46(C):25–35. doi:10.1016/j.cag.2014.09.021

Deaton JD, Grandhi RV (2014) A survey of structural and multidisciplinary continuum topology optimization: post 2000. Struct Multidiscip Optim 49(1):1–38

Garrido-Jurado S, Muñoz-Salinas R, Madrid-Cuevas FJ, Marín-Jiménez MJ (2014) Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern Recogn 47(6):2280–2292

Guest JK (2009) Imposing maximum length scale on topology optimization. Struct Multidiscip Optim 37(5):463–473

Guo X, Zhang W, Zhong W (2014) Doing topology optimization explicitly and geometrically – a new moving morphable components based framework. J Appl Mech 81(8):1–12

He K, Sigal L, Sclaroff S (2014) Parameterizing object detectors in the continuous pose space. Computer Vision – ECCV 2014, Volume 8692 of the series Lecture Notes in Computer Science, 450–465. doi: 10.1007/978-3-319-10593-2_30

Hsu MH, Hsu YL (2005) Interpreting three-dimensional structural topology optimization results. Comput Struct 83(4–5):327–337

Hsu YL, Hsu MS, Chen CT (2001) Interpreting results from topology optimization using density contours. Comput Struct 79(10):1049–1058

Kass M, Witkin A, Terzopoulos D (1988) Snakes: active contour models. Int J Comput Vision 1(4):321–331

Koguchi A, Kikuchi N (2006) A surface reconstruction algorithm for topology optimization. Eng Comput 22(1):1–10

Kumar AV, Gossard DC (1996) Synthesis of optimal shape and topology of structures. J Mech Des 118(1):68–74

Larsen SH (2010) Recognizing parametric geometry from topology optimization results. Scholars Archive at Brigham Young University. Retrieved from http://scholarsarchive.byu.edu/etd/2072/

Lazarov BS, Wang FW, Sigmund O (2016) Length scale and manufacturability in density-based topology optimization. Arch Appl Mech 86(1):189–218. doi:10.1007/s00419-015-1106-4

Le C, Bruns T, Tortorelli D (2011) A gradient-based, parameter-free approach to shape optimization. Comput Methods Appl Mech Eng 200(9–12):985–996

Li C, Kao CY, Gore JC, Ding Z (2008) Minimizing of region-scalable fitting energy for image segmentation. IEEE T Image Process 17(10):1940–1949

Li C, Kim IY, Jeswiet J (2015) Conceptual and detailed design of an automotive engine cradle by using topology, shape, and size optimization. Struct Multidiscip Optim 51(2):547–564. doi:10.1007/s00158-014-1151-6

Lin CY, Chao LS (2000) Automated image interpretation for integrated topology and shape optimization. Struct Multidiscip Optim 20(2): 125–137

Olhoff N, Bendsøe MP, Rasmussen J (1991) On CAD-integrated structural topology and design optimization. Comput Methods Appl Mech Eng 89(1–3):259–279

Papalambros PY, Chirehdast M (1990) An integrated environment for structural configuration design. J Eng Des 1(1):73–96. doi:10.1080/09544829008901645

Pedersen C, Allinger P (2006) Industrial implementation and applications of topology optimization and future needs. In: Bendsøe MP, Olhoff N, Sigmund O: IUTAM Symposium on Topological Design Optimization of Structures, Machines and Materials: Status and Perspectives, 137 (Part 6), Springer Netherlands, 229–238

Rozvany GIN, Lewiński T (2014) Topology optimization in structural and continuum mechanics. CISM International Centre for Mechanical Sciences 549, Springer-Verlag, Wien

Schramm U, Zhou M (2006) Recent developments in the commercial implementation of topology optimization Needs. In: Bendsøe MP, Olhoff N, Sigmund O: IUTAM Symposium on Topological Design Optimization of Structures, Machines and Materials: Status and Perspectives, 137 (Part 6), Springer Netherlands, 239–248

Seo YD, Kim HJ, Youn SK (2010) Shape optimization and its extension to topological design based on isogeometric analysis. Int J Solids Struct 47(11–12):1618–1640

Sigmund O (2001) A 99 line topology optimization code written in Matlab. Struct Multidiscip Optim 21(2):120–127

Sigmund O, Maute K (2013) Topology optimization approaches. Struct Multidiscip Optim 48(6):1031–1055

Su DC (2012) Active contour without edge. On the File Exchange on Matlab Central Website. Retrieved from http://www.mathworks.com/matlabcentral/fileexchange/34548-active-contour-without-edge

Suresh K (2013) Efficient generation of large-scale Pareto-optimal topologies.Struct Multidiscip Optim 47(1):49–61

Sutradhar A, Park J, Carrau D, Nguyen TH, Miller MJ, Paulino GH (2015) Designing patient-specific 3D printed craniofacial implants using a novel topology optimization method. Medical & Biological Engineering & Computing, 1–13. doi: 10.1007/s11517-015-1418-0

Tang PS, Change KH (2001) Integration of topology and shape optimization for design of structural components. Struct Multidiscip Optim 22(1):65–82

Tompson J, Stein M, Lecun Y, Perlin K (2014) Real-time continuous pose recovery of human hands using convolutional networks. ACM T Graphic 33(5), article No.: 169

van Dijk NP, Maute K, Langelaar M, van Keulen F (2013) Level-set methods for structural topology optimization: a review. Struct Multidiscip Optim 48(3):437–472

Yi GL (2014) Expanded SIMP method for structural topology optimization by transplanting ICM method. Dissertation, Beijing University of Technology

Yi GL, Sui YK (2015) Different effects of economic and structural performance indexes on model construction of structural topology optimization. Acta Mech Sinica 31(5):777–788

Yildiz AR, Ozturk N, Kaya N, Ozturk F (2003) Integrated optimal topology design and shape optimization using neural networks. Struct Multidiscip Optim 25(4):251–260

Youn SK, Park SH (1997) A Study on the shape extraction process in the structural topology optimization using homogenized material. Comput Struct 62(3):527–538

Zegard T, Paulino GH (2016) Bridging topology optimization and additive manufacturing. Struct Multidiscip Optim 53(1):175–192. doi:10.1007/s00158-015-1274-4

Zhou M, Rozvany GIN (1991) The COC algorithm, Part II: topological, geometry and generalized shape optimization. Comput Methods Appl Mech Eng 89(1–3):197–224

Zhou M, Fleury R, Shyy YK, Thomas H, Brennan J (2002) Progress in topology optimization with manufacturing constraints, 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization. Multidisciplinary Analysis Optimization Conferences, Atlanta. doi:10.2514/6.2002-5614

Zhou M, Fleury R, Patten S, Stannard N, Mylett D, Gardner S (2011) Topology optimization-practical aspects for industrial application. Proceeding in the 9th World Congress on Structural and Multidisciplinary Optimization, Shizuoka

Zhu JH, Zhang WH, Xia L (2015) Topology optimization in aircraft and aerospace structures design. Arch Comput Meth Eng, 1–28. doi: 10.1007/s11831-015-9151-2