A new path planning strategy based on level set function for layered fabrication processes

Xiaoying Hong1,2 · Guiqian Xiao3,1 · Yancheng Zhang3 · Jie Zhou1

Received: 14 June 2021 / Accepted: 15 October 2021 / Published online: 3 November 2021
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2021

Abstract
In this paper, a new equidistant filling theory based on the level set function and a corresponding numerical algorithm based on the dynamic finite difference method were proposed. Firstly, a closed curve is defined as the zero-value contour of the level set function. Then, the level set equation, a partial differential equation, was built to get the level set function, and a dynamic finite difference method was proposed to solve the level set function. Secondly, three types of cross sections, simple polygon, multi-island polygon, and multi-hole polygon, were used to test the equidistant filling effect of the algorithm. The test results show that the proposed theory and algorithm in this paper solved the equidistant filling problem of these cross sections well. In addition, any complex section can be equidistant filled by the algorithm proposed. The calculation cost is significantly related to the number of the offsets, but not directly related to the section complexity. Finally, the remanufacturing of a typical crankshaft hot forging die proves that the algorithm proposed in this paper can effectively produce the remanufacturing repair path of a complex forging die. The surface of the remanufactured die is smooth, and there is no effects such as porosity and slag inclusion after machining. Compared with making a new die, the manufacturing cost can be saved more than 50%, and the efficiency of die manufacturing can be improved more than 60%. This is because the machining allowance of the die after additive manufacturing is very small, and it can be quickly repaired on site.

Keywords Additive manufacturing · Tool path generation · Level set function · Equidistant offset filling

1 Introduction
Manufacturing industries often need to deal with equidistant filling problems of arbitrary sections related with the process, such as additive manufacturing (AM), 3D printing, and CNC machining. Especially, additive manufacturing (AM) or 3D printing, which is based on layer-by-layer manufacturing approach instead of conventional material removal methods, has gained worldwide popularity over the past 30 years [1]. Recently, gained wire and arc additive manufacturing (WAAM) is gaining popularity in aerospace, automotive, and biomedical industries because it can easily produce large components with lower cost and less time compared to other additive manufacture processes [2–4].

In WAAM, the 3D CAD model is sliced into a set of 2D cross section which can be used to generate the welding paths by path planning algorithm. One of the core tasks in WAAM is the welding path generation which decides the motion of the welding torch to fill the 2D cross section. Many path planning strategies have been proposed for AM, such as contour [5–7], raster [8], zigzag [9], spiral [10], fractal space-filling curves [11, 12], and hybrid tool path planning approaches [13].

Among these strategies, the simplest algorithms are the linear fill which offset parallel to a given line, and the contour offset fill which offset parallel to the boundary. The linear filling strategy guides the welding torch to move along a linear line, which means the welding torch need to be turned frequently [13]. These characteristics of the linear filling led to more arc breaks and discontinuous acceleration of the robot, especially for the 6-axis robot, which affects the forming quality and the life of the robot [8, 14, 15]. The spiral filling strategy can solve the problem results from the
linear fill strategy, but it is only suitable for some special models. In WAAM, the additive manufacture models are very complex, and the spiral filling strategy is not universal. The strategy of fractal space-filling curves can reduce the number of arc break and reduce shrinkage during in WAAM process. However, the moving direction of the welding torch often changes, which also leads to poor stability. Another better filling strategy is the contour offset filling strategy which has many advantages such as less arc break, smooth and continuous path, continuous speed and acceleration, and uniform filling. Therefore, this strategy is widely used in CNC machining, additive manufacturing (AM), and 3D printing. However, because of the complexity of different cross-sectional shapes, contour offset filling strategy has high complexity [15–17]. For example, in the manufacturing industry, the cross-sectional shape may be single-island, multi-island, convex, or concave polygon. The traditional contour offset filling strategy needs to realize the operations of polygon intersection, difference, union, and vertex degradation. In addition, a polygon offset may result in several polygons. So, the contour offset filling strategy must deal with the problem of polygon divergence. The complexity of these strategies limit their applications in WAAM.

In order to avoid the complex polygon operation caused by using the offset algorithm to get the contour filling line directly, this paper develops a new path planning algorithm to generate the equidistant contour filling line without complex polygon operations. The proposed algorithm firstly transforms the two-dimensional contour offset problem into a three-dimensional initial value problem of partial differential equation. Secondly, the dynamic finite difference method (DFDM) is used to solve the partial different equation, and then the contour lines also called offsetting contour lines can be obtained by cutting the solution surface of the partial equation at zero level. Finally, three types of cross sections are used to test the effectiveness of the algorithm. The results show that the proposed algorithm has simple structure, high efficiency, no complex polygon operation, and required it can be solved directly by mature finite difference method.

2 Theoretical bases

The mathematical theory of the novel contour offsetting method has been developed in this part. Firstly, the two-dimensional contour has been defined as the intersection of a surface and a plane. The construction and solution of the surface are the key problems in this part. The surface is described as an implicit function. Secondly, the differential equation of the surface is proposed. Finally, the evolutionary criteria are proposed to realize the equidistant migration. When the surface is calculated, the problem of contour offset is transformed into the problem of solving the contour line for the implicit function. Level set method, a calculation is put forward by Osher and Sethian interface movement numerical method, it is simple and general, calculation accuracy is high, is easy to deal with the topological merging and disconnect, and can be easily extended to the multidimensional problem, now has been widely applied to fluid calculation, crystal growth, material deposition, path planning and image processing, and other fields [18, 19].

2.1 Mathematical definition of closed curve

It is well known that the filling problem of a cross section belongs to two-dimensional problem. In two spatial dimensions, the lower-dimensional interface is a curve that separates $\mathbb{R}^2$ into separate subdomains with nonzero areas. The boundary of the cross section is closed. For easy processing, the inner contour and outer contour are defined as clockwise and anticlockwise arrangement, respectively. It is easy to propose that a two-dimensional contour can be defined as the intersection of a surface and a plane. The surface can be defined as $\phi(x, y, t)$ called the level set function which was first proposed by Osher and Sethian in 1988 to solve the interface evolution problems in the fields of fluid computing, crystal growth, and image processing [20]. In this paper, the level set function can be used to solve the equidistant offsetting problem. Particularly, the closed curve can be defined by $\phi(x, y, t) = 0$, in which the curve is changing over time. Then, the next problem is to determine the expression or value of the level set function. It is obviously impossible to determine the level set function directly. A desirable method is to determine the differential equation of the level set function. The time differential expression of the equation $\phi(x, y, t) = 0$ is as follows:

$$\frac{\partial \phi}{\partial t} + \nabla \phi \cdot \nabla t = 0$$ (1)

The above equation can be simplified as follows:

$$\frac{\partial \phi}{\partial t} + \nu \| \nabla \phi \| = 0$$ (2)

where $\nu = \frac{\phi_{x} \frac{\partial \phi}{\partial x} + \phi_{y} \frac{\partial \phi}{\partial y}}{\| \nabla \phi \|}$ and $\nabla \phi$ is the gradient of the implicit function which is called level set function. The gradient $\nabla \phi$ is perpendicular to the isocontours of $\phi$ and points in the direction of increasing $\phi$. In equidistant offsetting problem, the $\nu$ can be defined as 1, which is the standard Hamilton–Jacobi equation [21].

2.2 Signed distance function

The implicit equation $\phi(x, y, t) = 0$ is used to describe the closed curves in the previous section. In this section the signed distance functions, which are a subset of the implicit
functions defined in the last chapter, was discussed. A signed distance function has been defined as follows:

$$d(x, y, t) = \min \left( \sqrt{(x - x')^2 + (y - y')^2} \right)$$

(3)

where \(x\) and \(y\) represent the coordinates at any point. \(x'\) and \(y'\) represent the coordinates of any point on the boundary curve \(\phi(x, y, t) = 0\).

The implicit function can be defined as follows:

$$\phi(x, y, t) = \pm d(x, y, t)$$

(4)

As shown in Fig. 1, if point \((x, y)\) is positive inside the boundary, negative outside the boundary, and zero on the boundary. If point \(P'(x', y')\) is the closest point to point \(P(x, y)\), the closest point to any point on the line \(P'P\) is \(P'\).

Then, the following formula can be derived:

$$\nabla \phi = \left( \frac{\partial \left( \sqrt{(x - x')^2 + (y - y')^2} \right)}{\partial x}, \frac{\partial \left( \sqrt{(x - x')^2 + (y - y')^2} \right)}{\partial y} \right)$$

(5)

It is easy to get \(|\nabla \phi| = 1\) from formula (5). The level set Eq. (2) can be simplified as follows:

$$\frac{\partial \phi}{\partial t} + v = 0$$

(6)

where \(v\) is the distance that the boundary motion in unit time.

3 Numerical algorithms

The formulas and equations deduced in the previous part are all based on analytical theory. In practice, the section which needs to be filled is very complex, so it is impossible to solve the partial differential equations by analytical method. In addition, due to the complexity of some cross sections, the boundaries of these cross section may not be described analytically, and these boundaries can only be represented by a set of points arranged clockwise or counterclockwise.

Therefore, the following sections discuss the numerical algorithm required to solve the equidistant offset filling according to Eq. (6) in practical applications.

3.1 Auxiliary algorithms

3.1.1 Judgment algorithm of point in polygon

It is necessary to identify whether a point is within the polygon because of the introduction of the signed distance function. There are many algorithms to identify whether a point is within the polygon, such as area summation algorithm, angle summation algorithm, and ray algorithm. If the sum of the triangle area composed of the target point and each edge of the polygon is equal to area of the polygon, the target point is inside the polygon in area summation algorithm. If the sum of angle between the target point and all the edges is equal to 360 degrees, the target point is inside the polygon in angle summation algorithm. If the number of the points which intersect the edges of the polygon is odd, the target point is inside the polygon in ray algorithm.

In this paper, the last algorithm has been used. The detailed steps are as follows: compare the \(y\) coordinate of the target point with each point of the polygon to get a list of intersection points between the horizontal line where the target point is and the polygon edge. If the number of the intersection points on both sides of the target point is odd, the target point is inside the polygon; otherwise, it is outside the polygon. In the example shown in Fig. 2, there are seven intersections on the left and one intersection on the right of the target point. They are all odd numbers, so the point is in the polygon.

3.1.2 Judgment algorithm of polygon direction

It is necessary to judge whether the polygon is clockwise or counterclockwise when the zero solution contour line of the implicit function is obtained, and the inner and outer contour judgment algorithm is needed at this time. A general method to judge the direction of a complex polygon is curve integral, according to Green’s formula:

Fig. 1 Implicit function \(\phi\), its boundaries, and distance function. a Cross section and boundary. b surface and boundary
where $\partial D^+$ is the counterclockwise boundary of region $D$. Green’s formula reveals the relationship between double integrals in a plane region and plane curve integrals. If we take the integrand $Q = y$ and $P = 0$, then the following equation is derived according to Green’s formula:

$$
\int_D \left( \frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) dxdy = \oint_{\partial D^+} (Pdx + Qdy)
$$

(7)

where $s$ represents the area of region $D$. If the integral path is counterclockwise, the integral is positive. Otherwise, the integral is negative. Therefore, by calculating the positive and negative values of the curve integral, we can determine whether the polygon boundary is clockwise or counterclockwise. For the discrete polygon boundary, the discrete formula can be used to calculate the curve integral:

$$
s = \oint_{\partial D^+} xdy \approx \frac{1}{2} \sum_{i=1}^{n} \left( x_{i+1} + x_i \right) \left( y_{i+1} - y_i \right)
$$

(9)

where $n$ represents the number of discrete points of the polygon and $x_i$ and $y_i$ are coordinates of the $i$–th point on the boundary of the polygon. The advantage is that no matter how complex the polygon is, it can judge the sequence and inverse relationship of the polygon under the linear complexity, and the programming is relatively simple by using Green’s formula.

### 3.1.3 Judgment algorithm of inner or outer boundary

A cross section may be composed of several groups of polygons, so how to automatically judge the inside and outside of a polygon is very important. A simple method is to calculate the number of times a contour is included. If a contour is included even times, it is an outer contour; otherwise, it is an inner contour.

As shown in Fig. 3A, C and F are outer contours, which have been included even number of times. B, E, and D are
inner contours, which have been included odd number of times. Whether a target polygon is included in another polygon can be done as follows: traverse all vertices of the target polygon to determine whether all vertices are included in another polygon. If all vertices are included in another polygon, then the polygon is included.

### 3.2 Dynamic finite difference method

Partial differential Eq. (6) is the core of equidistant offset filling. It is well known that the numerical solution of partial differential equation (PDE) can be solved by the dynamic finite difference method (DFDM). The level set function \( \phi(x, y, t) \) is discretized into a sequence in the time dimension and into a two-dimensional grid in the space dimension. Then, \( \phi(x, y, t) \) is defined at every grid point on Cartesian grid. Updating \( \phi \) in time consists of finding new values of \( \phi \) at every grid point after some time increment \( \Delta t \). A simple first-order accurate method for the time discretization of Eq. (6) is the forward Euler method given by

\[
\frac{\phi(x_i, y_j, t_{k+1}) - \phi(x_i, y_j, t_k)}{\Delta t} + v(x_i, y_j, t_k) = 0 \tag{10}
\]

where \( t_{k+1} = t_k + \Delta t, x_{i+1} = x_i + \Delta x, \) and \( y_{j+1} = y_j + \Delta y \). \( \Delta x, \Delta y \) are increments in time, \( x, \) and \( y \) respectively; the smaller these values are, the higher the solution precision is. In addition, \( i = 1, 2, ..., M, j = 1, 2, ..., N, k = 1, 2, ..., W \). In addition, \( L = \Delta t \cdot v \) is called the fill spacing. The initial value of the level function \( \phi \) can be calculated by signed distance function. In particular, considering the equidistant offset, the points on the contour are offset at the same velocity, so \( v \) is a constant independent of time and position. Then, the solution algorithm can be simply described as follows:

\[
\left\{ \begin{array}{l}
\phi(x_i, y_j, t_{k+1}) = v \Delta t + \phi(x_i, y_j, t_k) \\
\phi(x_i, y_j, t_k) = f(x_i, y_j)
\end{array} \right. \tag{11}
\]

where \( f(x_i, y_j) \) is the discrete signed distance function in which the auxiliary algorithms should be used to calculate the signed minimum distance. The judgment algorithm of inner or outer boundary and the judgment algorithm of polygon direction are used together to calculate the inside and outside of the boundary, and the judgment algorithm of point in polygon is used to determine the positive and negative. In general, the detailed algorithm flow for solving Eq. (6) or Eq. (11) is as follows:

As shown in Fig. 4, the algorithm is a dynamic FDM algorithm, which updates the boundary and initializes the level set function every time when the filling line with specified spacing is obtained. This is due to the discretization of the boundary, spatial dimension, and time dimension; the \( |\nabla \phi| \) is not exactly equal to one. If all the filling lines are calculated statically, only once solving the partial differential equation, the filling lines may be distorted. In other words, the filling line obtained by the static method has poor precision and cannot achieve high precision equidistant filling lines.

### 3.3 Numerical tests

The traditional offsetting algorithm based on geometric processing is easy to get a set of equidistant filling lines for simple convex polygon. However, it is very difficult to obtain equidistant offset filling lines for complex cross sections, such as multi-island, multi-hole, and concave polygon. This section discusses the equidistant offset filling algorithm for complex cross section by using the algorithm described in this article.

#### 3.3.1 Simple polygon

In order to test the effectiveness of the algorithm, two simple polygons are selected for the offset fill test. The offset fill effect of square and circle is shown in Fig. 5.
Two-dimensional space is discretized into a 400×400 grid in these cases, and the offset distances of the two polygons in Fig. 5 are 1 mm and 1 mm, respectively.

When the iteration satisfies $v = t_0 = L$, the $\phi$ function are re-initialized, and the iteration starts again until the polygon is completely filled. The value of level set function $\phi$ has only one extreme value and is different after each initialization. This is because for a convex polygon section with only one boundary, the polygon does not split during the offset process; that is, multiple polygons will not be generated by one polygon offset.

The filling effect of the two simple polygons is shown in Fig. 6. The algorithm introduced in this paper can quickly deal with the offset filling of simple polygon. When the level set function $\phi$ does not intersect with the plane described by $z=0$, the filling ends.

### 3.3.2 Multi-island polygon

As shown in Fig. 7, the algorithm designed in this paper can also be applied to the equidistant offset filling calculation for multi-island cross section with three outer boundary and one inner boundary. In the case of multi-island cross section, the level set function has multiple extreme values, and the level set function varies greatly. So, the dynamic finite difference method is very necessary to solve the level set equation.

In these cases, two-dimensional space is discretized into a 1000×1000 grid, and the offset distance is defined as 2 mm. As shown in Fig. 8, the number of polygons obtained by the offsetting is changed in the process of equidistant offsetting. Fortunately, in this algorithm, no special operation is required whether polygons are added, subtracted, or disappeared, and the algorithm can automatically evolve the polygon by taking the contour of $\phi = 0$.

### 3.3.3 Multi-hole polygon

Another complex situation is the multi-hole cross section as shown in Fig. 9 and Fig. 10. There are five inner boundaries and one outer boundary, which result in the level set function having multiple extreme values. In this case,
two-dimensional space is discretized into $1000 \times 1000$ grid, and the offset distance is defined as 1 mm.

The level set function varies with the boundary, and the polygon boundary changes are much more complex than those in the previous multi-island case. The algorithm presented in this paper can easily handle the filling problem of complex cross section, no matter how complex the cross sections are. In addition, the complexity of the algorithm
is related to the number of offset and independent of the shape of the cross section.

As is shown in Fig. 10, the equidistant filling line shrinks from the outside to the inside, which is important for arc fuse additive remanufacturing processes because the welding torch can only be deflected toward the center or it may hit the base material.

4 Additive manufacture experiment

In this section, a typical hot forging die is used for additive remanufacturing to analyze the effectiveness of the algorithm described above. The welding process parameters are as follows: voltage is 24.5 V, current is 200A, wire feeding speed is 600 mm/min, and welding speed
is 450 mm/min. Under these parameters, the width and height of the weld are 9 mm and 3 mm, respectively. Therefore, the slicing height is 3 mm and the offsetting distance is 4.5 mm. A typical cross section of crankshaft die is shown in Fig. 11. Although the section is thin, multiple offsets are required to obtain an equidistant filling line.

The section is thin, so it is difficult to fill it evenly by traditional filling algorithm. This section has only one outer boundary and one inner boundary. When the offsetting distance is 9 mm, the boundary will split into five outer boundaries. When offsetting distance is 13.5 mm, the boundary continues to divide into nine outer boundaries. Finally, when the filling is completed, some boundaries disappear, leaving only five small outer boundaries.

The offsetting filling and welding filling effects of this section are shown in Fig. 12. The section is relatively smooth after welding. Although there are small pits in local areas, these small pits are filled out as the current layer is covered by the next layer throughout the additive manufacturing process. Because the size of the section cannot be divided by the width of the weld, no matter what algorithm is used, there will be overfilling or underfilling in local areas. Fortunately, these areas are very small and different layer positions are different, so they can be filled automatically in the additive manufacturing process.
Fig. 9 The multi-hole cross-sectional offset curves and level set functions at different offset distances. a Offsetting distance 1.5 mm, b offsetting distance 3 mm, c offsetting distance 4.5 mm, d offsetting distance 6 mm, e offsetting distance 7.5 mm, f offsetting distance 9 mm, g offsetting distance 10.5 mm, h offsetting distance 12 mm, i offsetting distance 12.5 mm
As shown in Fig. 13, the process of wire arc additive manufacturing includes extracting target model, slicing, equidistant offset filling, welding, and machining. The target model is a 3D model obtained by scanning or modeling. In this paper, a target model of crankshaft hot forging die has been shown in Fig. 13a. This target model is very complex, which has many complex surfaces and curves. The equidistant offset filling is the key task of this process, which directly affects the welding accuracy and quality. As shown in Fig. 13b, c, the welding surface is smooth and the shape accuracy is high. At the edge of the die base, there is a welding tumor. This is due to excessive welding heat input and outflow of weld metal, but it does not affect the welding quality. As shown in Fig. 13d, the surface of the die is smooth, and there is no defects such as porosity and slag inclusion after machining. Compared with manufacturing a new die, the cost can be saved more than 50%, and the efficiency of die manufacturing can be improved more than 60%. On the one hand, making a new die needs to machine the mold cavity, which not only wastes a lot of material but also wastes a lot of machining time. On the other hand, the machining allowance of the die after additive remanufacturing is very small, and it can be quickly repaired on site. In addition, because the material used in the new die is a homogeneous structure, its service life is much lower than that of the remanufactured die with gradient structure.

5 Conclusions

(1) In this paper, a new equidistant offset filling theory based on level set function has been proposed, and the corresponding numerical solving algorithm based on dynamic finite difference method has been developed. Firstly, the plane closed curve is defined as the zero-value contour of implicit function. Then, the level set equation, a partial differential equation, has been developed to get this implicit function. Finally, a dynamic
finite difference method has been proposed to solve the level set function.

(2) Three types of cross sections which are simple polygon, multi-island polygon, and multi-hole polygon are used to test the equidistant offset filling effect of the proposed algorithm. The test results show that the algorithm in this paper can well deal with the equidistant filling problem of these sections. In addition, any complex section can be equidistant filled by the algorithm proposed in this paper, and the calculation cost is significantly related to the number of offsets but not directly related to the complexity of the cross section.

(3) The remanufacturing of a typical crankshaft hot forging die proves that the algorithm proposed in this paper can effectively produce the remanufacturing repair path of this kind of forging die. The surface of the die is smooth, and there is no defects such as porosity and slag inclusion after machining. Compared with making a new die, the manufacturing cost can be saved more than 50%, and the efficiency of die manufacturing can be improved more than 60%. This is due to the machining allowance of the die after additive manufacturing is very small, and it can be quickly repaired on site.

Author contribution Xiaoying Hong put forward the relevant algorithm and ideas and wrote the article. Guiqian Xiao wrote the software and developed the hardware and carried out experiments. YanCheng Zhang...
did the welding process parameter experiment. Jie Zhou guided and reviewed the whole process.

**Funding** This research is supported by National Natural Science Foundation of China (No. 52075058) and by Postdoctoral Science Foundation of Chongqing Natural Science Foundation (No. cstc2020jcyj-bshX0006).

**Data availability** All data generated or analyzed during this study are included in this published article.

**Declarations**

**Ethics approval** Not applicable.

**Consent to participate** Not applicable.

**Competing Interests** The authors declare no competing interests.

**References**

1. Ding D, Pan Z, Cuiuri D, Li H (2015) A practical path planning methodology for wire and arc additive manufacturing of thin-walled structures. Robot Comput Integr Manuf 34(8): 8–19
2. Singh S, Sharma SK, Rathod DW (2021) A review on process planning strategies and challenges of WAAM. Mater Today: Proc 2021(4)
3. Yaseer A, Chen H (2021) A review of path planning for wire arc additive manufacturing (WAAM). J Adv Manuf Syst
4. Karunakaran KP, Suryakumar S, Pushpa V, Akula S (2010) Low cost integration of additive and subtractive processes for hybrid layered manufacturing. Robot Comput Integr Manuf 26(5):490–499
5. Farouki RT, Koenig T, Tarabanis KA, Korein JU, Batchelder JS (1995) Path planning with offset curves for layered fabrication processes. J Manuf Syst 14(5):355–368
6. Yang Y, Loh HT, Fuh J, Wang Y (2002) Equidistant path generation for improving scanning efficiency in layered manufacturing. Rapid Prototyp J 8(1):30–37
7. Li H, Dong Z, Vickers GW (1994) Optimal toolpath pattern identification for single island, sculptured part rough machining using fuzzy pattern analysis. Comput Aided Des 26(11):787–795
8. Dunlavey MR (1983) Efficient polygon-filling algorithms for raster displays. ACM Trans Graph 2(4):264–273
9. Park SC, Choi BK (2000) Tool-path planning for direction-parallel area milling. Comput Aided Des 32(1):17–25
10. Wang H, Jang P, Stori JA (2001) A metric-based approach to two-dimensional (2D) tool-path optimization for high-speed machining. Am Soc Mech Eng
11. Kulkarni P, Marsan A, Dutta D (2000) A review of process planning techniques in layered manufacturing. Rapid Prototyp J 6(1):18–35
12. Konrad C (2011) Two-constraint domain decomposition with space filling curves. Parallel Comput 37(4–5):203–216
13. Dwivedi R, Kovacevic R (2004) Automated torch path planning using polygon subdivision for solid freeform fabrication based on welding. J Manuf Syst 23(4):278–291
14. Rajan VT, Srinivasan V, Tarabanis KA (2001) The optimal zigzag direction for filling a two dimensional region. Rapid Prototyp J 7(5):231–241
15. Park SC, Choi BK (2000) Tool-path planning for direction-parallel area milling. Comput Aided Des 32(1):17–25
16. Sang CP, Yun CC, Choi BK (2003) Contour-parallel offset machining without tool-retractions. Comput Aided Des 35(9):841–849
17. Mao L, Gao F, Yuan Z, Chen X, Benke WU (2017) Filling algorithm of two-dimensional contour in selective laser melting molding. J Hefei Univ Technol
18. Sethian JA (1999) Level set methods and fast marching method. J Comput Phys 118 2
19. Osher S, Sethian JA (1988) Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. J Comput Phys 79(1):12–49
20. Crandall M, Lions PL (1983) Viscosity solutions of Hamilton-Jacobi equations. Trans Am Math Soc 277(1):1–42

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

I would like to declare on behalf of my co-authors that the work was an original research that has not been published previously.