

RESEARCH ARTICLE

Robotic spray painting path planning for complex surface: boundary fitting approach

Hongrui Wu  and Qing Tang* 

School of Mechanical Engineering, Sichuan University, Chengdu, P.R. China

*Corresponding author. E-mail: tangqing_scu@163.com

Received: 11 August 2022; **Revised:** 15 December 2022; **Accepted:** 16 January 2023;

First published online: 14 February 2023

Keywords: path planning, spray painting, offline programming, uniformity optimization, complex surface

Abstract

Currently, most of the robot path planning methods for spray painting are based on slicing the Computer-Aided Design model of the target surface to generate a series of path points. They usually neglect the geometric boundary of the model and the smoothness of the generated path, leading to a non-uniform coating thickness. To ameliorate it, an improved “boundary fitting approach” is proposed. In this method, the upper and lower boundaries of the target surface are firstly detected based on the topology of the stereolithography model. For each pass, several sample points are extracted with a uniform length interval from the intersection points generated by the basic slicing method. The path pass is then described by a fourth-order polynomial curve. It fits the boundary points and sample points for the z - t and y - t relationships, respectively. Based on the spray gun’s motion direction and spray direction, the orientation of each path point is also defined. The parameters of the path pass are optimized by particle swarm optimization to get the optimal uniformity of the resulting coating thickness. Both of the global uniformity and the local uniformity between two adjacent passes are considered. The strength of the proposed approach is validated by comparing the simulation with the basic and other typical algorithms. The results denote that boundary fitting approach could improve the uniformity of coating thickness. It brings about a better performance for the painted workpiece.

1. Introduction

Spray painting is one of the most crucial processes in manufacturing. It plays a pivotal role in boosting the workpiece’s performance, especially the durability in a hostile environment [1]. Currently, human’s role in spray painting processes comes to be taken over by industrial robots, which substantially boost the consistency of path generation and then bring about a much stabler painting performance, especially in large batch production [2]. Hence, it is significant to develop a reliable algorithm to boost the quality and quantity of robotic spray painting.

To evaluate the quality of spray painting, the uniformity of coating thickness could be the commonest indicator [3]. The evenness of the coating thickness, the better the mechanical and thermodynamic properties of the painted workpiece [4]. To acquire a uniform thickness distribution, generating an appropriate painting path is rather critical, which directly impacts the painting result [5]. Typically, there are two categories of methods to generate a robot motion path: online programming and offline programming. Nowadays, the latter is preferred owing to its efficiency of production and feasibility of further optimization. It could automatically generate the optimal path instead of empirically teaching the robot by experienced workers [6]. Therefore, painting path generation based on offline programming has attracted increasing interest from researchers.

Before achieving a uniform thickness distribution, the first issue is enabling the paint to cover the target surface. Hence, the surface’s geometry information should be imported to prepare for path planning. Currently, there are three kinds of models to store the workpiece’s geometry information: the

point cloud model, the parametric model, and the tessellation model. In terms of the point cloud model, Yu *et al.* [7] proposed an edge preservation algorithm and a slicing algorithm for spray painting path planning. Their approach could preserve the model's edge characteristic while processing. Considering the complexity of different point cloud areas, Kong *et al.* [5] presented an approach where different algorithms were used for different regions. An automatic region segmentation algorithm was also proposed in their study. Generally, the point cloud model enjoys high precision when expressing the shape of a real-time object. However, to generate a point cloud model, a professional 3D scanner is needed, which substantially increases the cost. Besides, the data stored in the point cloud model are quite redundant, which hamper further processing.

Getting rid of the high device requirement, the path generation approach based on the parametric or tessellation model only needs the workpiece's Computer-Aided Design (CAD) file. These two kinds of models just differ in the data format that expresses the geometry information. The former uses a precise mathematical description while the latter approximates a complex surface into multiple triangular facets. Utilizing the parametric model, Atkar *et al.* [8] developed an algorithm to generate a group of path passes. It is based on offsetting a selected geodesic called "start curve" extracted from the model. In their approach, a complicated workpiece could be segmented into geometrically simple parts. However, the workpiece had to own an extruded surface to apply their method.

Currently, the tessellated model, especially the stereolithography (STL) model, is more common owing to its explicitness of geometric information and integrity for data processing compared with the parametric one [9]. This model is also adopted in our work. Based on slicing the STL model along a selected direction of its bounding box, Chen and Xi [10] developed an algorithm to generate path points. In their method, a whole workpiece was divided into several simple patches according to the threshold deviation angle between two triangles. Hence, paths were generated independently within each patch for thickness uniformity. Afterwards, an integration algorithm was designed to combine paths between different patches. The steepest-descent algorithm was used to optimize the coating thickness. Their study took the target surface's local geometry into account. However, the integration would be considerably complex when the workpiece was highly curved. Instead of sectioning a workpiece into multiple patches, Guan and Chen [11] presented a path generation approach based on directly slicing the target surface. To ensure a satisfactory coating thickness for the complex surface, they put emphasis on optimizing the shape of the transitional path between two adjacent path points generated by offsetting the corresponding intersection points. The golden section method was used to optimize the spacing distance to find the optimal uniformity. However, in this method, the spray painting parameters to be optimized were too limited to obtain a uniform thickness distribution. Andulkar *et al.* [12] designed a novel method instead of utilizing the slicing algorithm. In their study, the target surface was segmented into a series of triangle sections according to paint's coverage area. Based on the geometry information of the triangles, the path point was generated by offsetting the section's average point along the average normal. Pattern search optimization was adopted to determine the overlap distance and velocity to the optimal uniformity. They smoothed the path points using the two-sided moving average. Considering the geometric boundary of the target surface, Zeng *et al.* [13] developed a path generation algorithm based on model slicing. Particle swarm optimization (PSO) was used for optimization. Their method generated a path that fits the model's boundary. However, the uniformity of coating thickness deteriorated when the boundary factor was considered. The main reason for this consequence was that they treated the boundary as a constraint in the uniformity optimization. The improved spray efficiency and reduced waste of paint were seen as the contribution of boundary fitting.

From these previous studies, it could be seen that most of the path planning methods are based on slicing the CAD model to obtain a series of path points. However, up to date, there has been limited study that considers the geometric boundary of the model, which could exert a strong influence on the resulting path points generated by the slicing algorithm. Lacking further improvement to address this effect, an unsatisfactory thickness distribution could emerge, especially when the target surface is free-formed and complex. Besides, in terms of the optimization, many methods are confined to limited spray painting parameters. It constrains the result to a deficient value. In addition, little attention has been paid to the

smoothness of the generated path, which could also impact the spray painting result. To deal with these problems, this paper proposes an improved path planning algorithm called “boundary fitting method.” By boundary point detection, sample point generation, and curve fitting, it avoids the “boundary effect” and generates a smooth painting path. Meanwhile, an appropriate optimization method is designed to find a better result. Compared with the basic method and other typical methods, the uniformity of coating thickness is enhanced.

This paper has been organized in the following structure. Section 2 explains the basic slicing method and common models to simulate the coating thickness. The improved boundary fitting approach is discussed in Section 3. The optimization for uniform coating thickness is described in Section 4. Section 5 discusses the simulation. Section 6 reaches a conclusion.

2. Background

2.1. Basic slicing algorithm

To compactly cover the target surface, path points should be generated by processing corresponding points on the target surface. This is commonly implemented by slicing the model with a set of planes to extract the intersection points. It imitates the slicing process in additive manufacturing but with a much longer spacing between slice planes. In spray painting, most of the path is in a raster pattern, which ensures a better uniformity of coating thickness [14, 15]. Some key parameters of the raster pattern path could be presented by these slice planes. For example, spacing distance between two adjacent passes is expressed by that between two adjacent planes.

By importing the STL model of the target surface and transforming it to fit the XZ plane of the world coordinate system, a group of slice planes could be defined by their Z coordinates. These planes are perpendicular to the XZ plane and parallel to each other. Based on the plane, the intersection points on STL facets’ edges could be calculated by:

$$x_{\text{inter}} = x_1 + \frac{(z_{\text{sli}} - z_1)(x_2 - x_1)}{z_2 - z_1} \quad (1a)$$

$$y_{\text{inter}} = y_1 + \frac{(z_{\text{sli}} - z_1)(y_2 - y_1)}{z_2 - z_1} \quad (1b)$$

$$z_{\text{inter}} = z_{\text{sli}} \quad (1c)$$

where $(x_{\text{inter}}, y_{\text{inter}}, z_{\text{inter}})$ is the coordinate of the resulting intersection point, (x_1, y_1, z_1) and (x_2, y_2, z_2) are coordinates of the edge’s two vertices, and z_{sli} is the Z coordinate of the current slice plane.

In the STL model, a vertex is shared with several adjacent facets. For each vertex, let us calculate the average normal vector among its adjacent facets. This vector is defined as the normal vector of the vertex. Moreover, the normal vector of the intersection point could be constructed by the average normal vector between the sliced edge’s two vertices. By offsetting each intersection point along the normal vector with a certain spray distance, a path pass for spray painting could be generated.

This method works when the slice plane is within the upper and lower boundaries. However, when the plane slices some special places around the boundary, for instance, the highest or lowest vertex of the model, the basic method could only generate limited path points compared with the middle passes. This phenomenon is called “boundary effect.” Besides, the basic method lacks further processing to boost path’s smoothness, making path points discrete and non-uniform. The above problems would cause an unsatisfactory thickness distribution and weaker robot control [16]. To ameliorate these two situations, the boundary fitting demand and path smoothness demand should be met in the proposed method. Figure 1 illustrates the outline of the basic slicing algorithm.

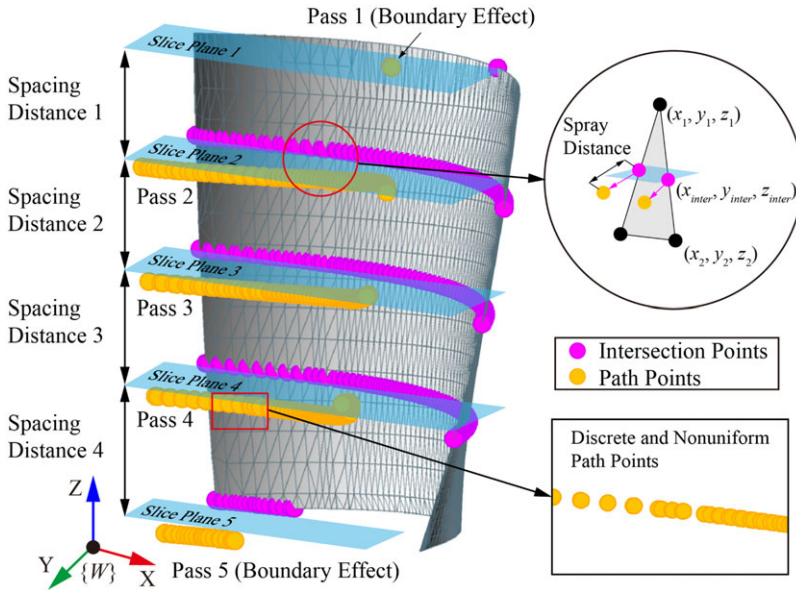


Figure 1. Outline of basic slicing algorithm.

2.2. Thickness modeling

Based on the generated pass, the following work is calculating the resulting coating thickness. This is a preliminary step for further thickness uniformity optimization. To simulate it, two parts should be considered here: the instant model and the dynamic model.

The instant model aims to compute the coating thickness formed by a spray gun targeted at an arbitrary surface in an instant. To model such a scenario, the first step is finding the standard thickness distribution. It is formed by a spray gun targeted at a standard plane with a perpendicular spray direction and a standard spray distance. To implement it, an experimental test should be conducted beforehand to collect related thickness data. Afterwards, a distribution function is adopted to fit the data. In this study, the ellipse dual-β distribution model is selected owing to its flexibility in describing the shape of the thickness distribution [7, 17]. It could be given by:

$$\begin{aligned}
 T_{std}(p) &= T_{std}(x_p, y_p) \\
 &= T_{max} \left(1 - \frac{x_p^2}{a^2}\right)^{\beta_1-1} \left[1 - \frac{y_p^2}{b^2\left(1 - \frac{x_p^2}{a^2}\right)}\right]^{\beta_2-1}, \quad \frac{x_p^2}{a^2} + \frac{y_p^2}{b^2} \leq 1
 \end{aligned}
 \tag{2}$$

where T_{max} is the maximum coating thickness in an instant, it is related to the paint flow rate and paint transfer efficiency of the spray gun. a is the semi-major axis of the elliptical coverage area, and b is the semi-minor axis. They could describe the cross-section contour of the distribution. β_1 is the exponent of the distribution on the Y -axis section and β_2 is that on the X -axis section. They could delineate the contour on the longitudinal section.

To make the model more general, the next step is projecting the distribution function from the standard plane onto an arbitrary surface point. This course is based on differential geometry [18]:

$$T(p_1) = \begin{cases} T_{std}(p) \frac{h^2 \cos \alpha}{gp_1^2 \cos^3 \theta} & , \alpha \leq 90^\circ \\ 0 & , \alpha \geq 90^\circ \end{cases}
 \tag{3}$$

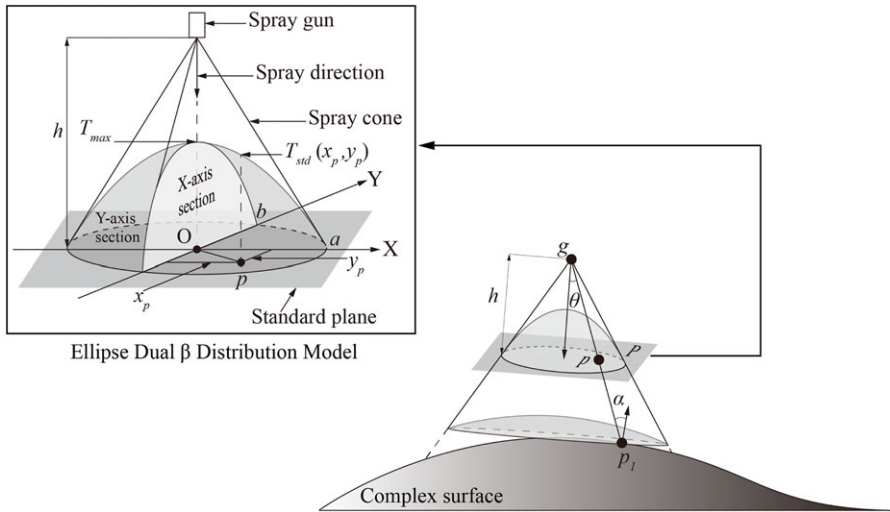


Figure 2. Instant model. The left figure illustrates the standard thickness distribution and the right figure depicts the projection from the standard plane to an arbitrary surface point.

where g is the path point, p_1 is the target point, \vec{gp}_1 is the vector from g to p_1 , P is the standard plane mentioned above, p is the intersection point between \vec{gp}_1 and P , h is the standard spray distance between g and P , which has been determined in the experimental test, θ is the angle between spray direction and \vec{gp}_1 , and α is the angle between the normal vector of p_1 and the opposite direction of \vec{gp}_1 . Hence, an instant model could be constructed as discussed above. The instant model is illustrated in Fig. 2.

In contrast to the instant model, the dynamic model is targeted at finding the coating thickness formed by a spray gun moving along a certain pass in a certain time interval. Dynamic model during time interval $[0, t]$ could be configured by integrating instant model T over time.

$$T_{dyn} = \int_0^t T dt \tag{4a}$$

To calculate it, a continuous path is divided into enough discrete segments where each segment is a line segment bounded by two adjacent path points [19]. Hence, an instant time could be approximated as a time interval as follows:

$$dt = \frac{|\vec{g_i g_{i+1}}|}{v} \tag{4b}$$

where $|\vec{g_i g_{i+1}}|$ is the length of the path segment bounded by path points g_i and g_{i+1} , and v is the average spray gun motion speed in this segment. In terms of p_1 , the cumulative thickness $T_{dyn}(p_1)$ could be computed by summing all the products between the instant thickness $T_{i+1}(p_1)$ gained in the discrete path segment $g_i g_{i+1}$ and the corresponding time interval.

$$T_{dyn}(p_1) = \sum_{i=1}^{n-1} T_{i+1}(p_1) \frac{|\vec{g_i g_{i+1}}|}{v} \tag{4c}$$

where i and n are the index and the total number of the path points, respectively. This model is depicted in Fig. 3.

3. Path generation in boundary fitting approach

To deal with problems mentioned in Section 2.1, an improved algorithm called ‘‘boundary fitting approach’’ is proposed. It smoothly generates a pass using a polynomial curve that fits the geometric

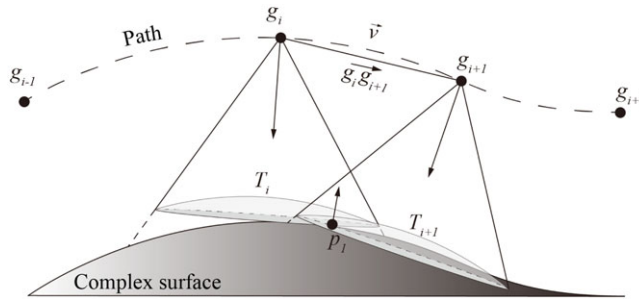


Figure 3. Dynamic model.

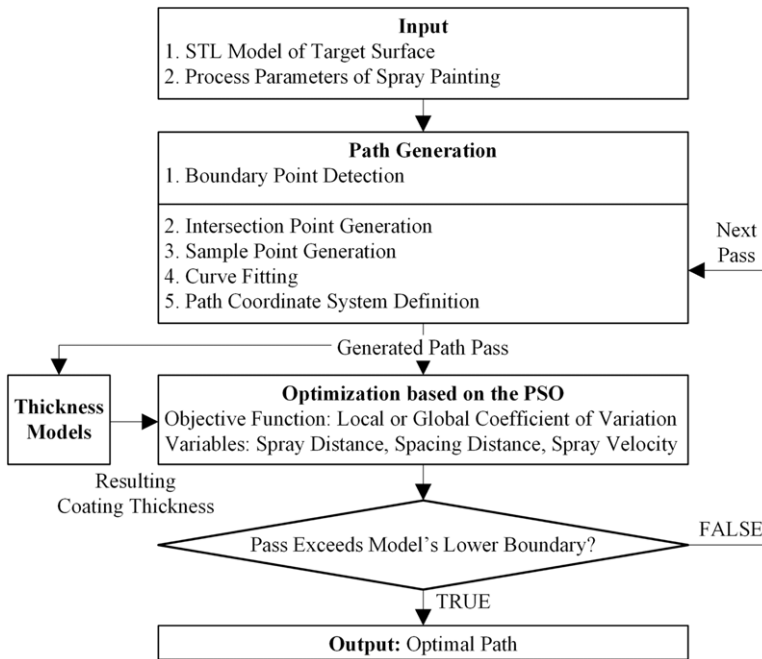


Figure 4. Overview of boundary fitting approach.

boundary and sample points of the intersection points. The path generation starts from the pass near the upper boundary. When the pass exceeds the lower boundary, the generation ends. To obtain the optimal global uniformity of coating thickness, the local uniformity between two adjacent passes is optimized based on PSO. Hence, key parameters related to coating thickness are adjusted pass by pass [12]. The resulting coating thickness is obtained by thickness modeling mentioned in Section 2.2. Figure 4 overviews the proposed approach.

3.1. Boundary point detection

To meet the demand for boundary fitting, the first step is identifying the boundary points of the target surface, which are evident for human beings but a little tricky for computers. In the STL model, an edge is defined as a line segment bounded by two vertices on a certain facet. According to the topology of a meshed model, each boundary edge is exclusive to only one facet while other edges are shared

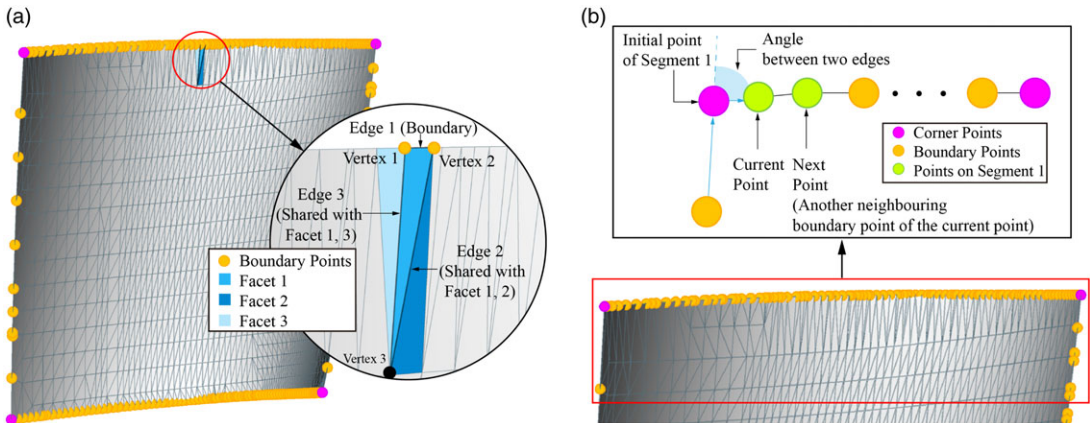


Figure 5. Boundary point detection. (a) Boundary point extraction. (b) Segmentation.

with different facets. Based on this principle, the whole loop of boundary edges could be searched and boundary points on these edges are extracted. A boundary point is shared with two boundary edges and the other two boundary points on these edges are its neighboring boundary points. Figure 5(a) depicts this course.

Generally, a boundary loop could be divided into four independent segments by four corner points. To identify the corner points, let us connect each boundary point's two boundary edges and compare the angle between these two edges as shown in Fig. 5(b). The point with one of the largest four angles is a corner point. Afterwards, to initialize the segmentation of boundary points, let us randomly choose one corner point as the initial point on the initial segment and then randomly choose one of its neighboring boundary points as the second point. Afterwards, by excluding the former point on the segment, the remaining neighboring boundary point of the current point is selected as the next point. This step repeats until another corner point is searched. The next segment begins with the new corner point and a new step continues in the same way. This loop ends when the initial corner point is searched again. Figure 5(b) also illustrates this course.

Thus far, the four independent boundary segments could be distinguished. Among them, the segment with the highest average Z coordinate is defined as the upper boundary, whereas the segment with the lowest one is defined as the lower boundary.

3.2. Sample point generation

Based on the intersection points mentioned in Section 2.1, several typical points should be selected from them to reflect the general geometric information and facilitate further processing. The term "key points" is used here to describe such points. They are selected at a uniform length interval between the leftmost and the rightmost intersection point of a pass.

As shown in Fig. 6, the search starts from the leftmost intersection point, which is selected as the first key point. During each step, the length of the line segment adds up one by one. When the accumulative length of segments exceeds the length interval $invl$, the two vertices of the current segment remain to be selected. If the current accumulative length is closer to $invl$, the right vertex is selected as the next key point. Otherwise, the left vertex is selected. In the next step, the accumulative length is cleared and the searching continues from the new key point. At the end of this loop, the rightmost intersection point is selected as the last key point. By offsetting each key point along its normal vector with a value of spray distance, a corresponding sample point is formed, whose spray direction is opposite to the normal vector:

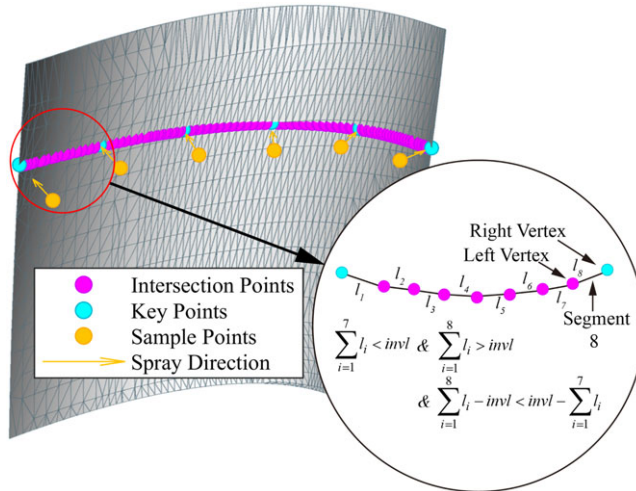


Figure 6. Sample point generation. Key points are searched with a uniform length interval. The right vertex of Segment 8 is selected as the key point because the accumulative length until Segment 8 just exceeds the length interval and is closer to it than the former one.

$$\begin{bmatrix} x_{\text{smp}} \\ y_{\text{smp}} \\ z_{\text{smp}} \\ 1 \end{bmatrix} = \begin{bmatrix} I_{3 \times 3} & h_1 \cdot \vec{n}_{\text{key}} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{\text{key}} \\ y_{\text{key}} \\ z_{\text{key}} \\ 1 \end{bmatrix} \tag{5a}$$

$$\vec{n}_{\text{smp}} = -\vec{n}_{\text{key}} \tag{5b}$$

where $(x_{\text{key}}, y_{\text{key}}, z_{\text{key}})$ and \vec{n}_{key} are the coordinate and normal vector of the key point, $(x_{\text{smp}}, y_{\text{smp}}, z_{\text{smp}})$ and \vec{n}_{smp} are those of the sample point, $I_{3 \times 3}$ is a 3×3 identity matrix, and h_1 is spray distance. Figure 6 depicts the generation of sample points from key points. By offsetting different key points with different spray distances, various shapes of the pass could be generated based on the resulting sample points.

3.3. Curve fitting

Fitting boundary points and sample points extracted above, the pass could be expressed by a polynomial curve [20]. Generally, a space curve could be defined by a parametric equation:

$$\vec{r} = (x(t), y(t), z(t)) \tag{6a}$$

Firstly, let x be a free parameter:

$$x(t) = t \tag{6b}$$

Regarding the y - t relationship, a fourth-order polynomial is used to fit the sample points. It could be approximated by:

$$y(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 \tag{6c}$$

where c is the coefficient of the y - t polynomial.

In terms of the z - t relationship, another fourth-order polynomial should be introduced to meet the demand for boundary fitting. Therefore, coordinates of the upper and lower boundary points are utilized in the curve fitting. Firstly, let us assume that these two groups of points are translated with the same

distance along the z axis to the place near the slice plane:

$$z_{\text{upr}} - (\bar{z}_{\text{upr}} - z_{\text{sli}}) = a_0 + a_1x_{\text{upr}} + a_2x_{\text{upr}}^2 + a_3x_{\text{upr}}^3 + a_4x_{\text{upr}}^4 \tag{6d}$$

$$z_{\text{lwr}} + (z_{\text{sli}} - \bar{z}_{\text{lwr}}) = b_0 + b_1x_{\text{lwr}} + b_2x_{\text{lwr}}^2 + b_3x_{\text{lwr}}^3 + b_4x_{\text{lwr}}^4 \tag{6e}$$

where x_{upr} , x_{lwr} , and $z_{\text{upr}}, z_{\text{lwr}}$ are each X and Z coordinates of the upper and lower boundary points, \bar{z}_{upr} and \bar{z}_{lwr} are the average Z coordinates of the upper and lower boundary points, z_{sli} is the Z coordinates of the slice plane indicating the rough Z -axis position of the pass, a is the coefficient of the translated upper boundary polynomial, and b is the coefficient of the lower one. To offer more possibilities of the generated path, the first or the last z_{sli} could exceed the maximum or minimum Z coordinate of the model. In this case, its $y-t$ relationship is constructed by fitting corresponding upper or lower boundary points instead of sample points.

Besides, the shape of the curve should vary gradually from pass to pass. It ensures an even overlap and brings about a more uniform thickness distribution. To fulfill it, a compound polynomial is constructed by a linear interpolation between two coefficients with the same order from the two translated polynomials:

$$d_i = \frac{z_{\text{sli}} - \bar{z}_{\text{lwr}}}{\bar{z}_{\text{upr}} - \bar{z}_{\text{lwr}}} a_i + \frac{\bar{z}_{\text{upr}} - z_{\text{sli}}}{\bar{z}_{\text{upr}} - \bar{z}_{\text{lwr}}} b_i, \quad i \in \{0, 1, 2, 3, 4\} \tag{6f}$$

$$z(t) = d_0 + d_1t + d_2t^2 + d_3t^3 + d_4t^4 \tag{6g}$$

where i is the index of the polynomial coefficient, and d is the compound coefficient. Eq. (6g) denotes the final $z-t$ polynomial. In the curve fitting, the fourth-order polynomial is selected to describe the path. As shown in Eq. (6f), it could facilitate further construction by processing its five explicit and dominating coefficients. Besides, the fourth order is sufficient to approximate various path passes. Meanwhile, it could also avoid Runge’s phenomenon where the generated path could be highly oscillatory when the degree comes to be higher. Figure 7 depicts the $z-t$ and $y-t$ polynomial curves.

Thus far, with an arbitrary t , the coordinates of path points could be obtained based on Eq. (6a). Owing to the varying curvature of the path curve, points should be interpolated with a uniform length interval as explained in Section 3.2, instead of directly plugging a series of uniform t into the polynomial. It ensures a uniform distribution of the path points.

3.4. Path coordinate system definition

Up to now, a series of consecutive path points with certain positions have been generated. However, orientation, another significant parameter to define a robot path, has not been expressed. In robotics, a common way to describe these two parameters is attaching a coordinate system called $\{Path\}$ to each path point. In this scenario, the origin of $\{Path\}$ is the position of the path point, the Y axis of $\{Path\}$ is along the spray gun’s motion direction, the Z axis of $\{Path\}$ is along the spray direction, and the X axis is orthogonal to both of the Y axis and Z axis. In this case, the motion direction is defined as the direction of a connecting line between the current path point and the next one. Meanwhile, the spray direction is mapped from the opposite normal vector of the closest intersection point to the path point.

In the definition of the coordinate system, the orthogonality between the motion direction and the spray direction, the two independent axes, is required to be ensured. It could be fulfilled by rotating the spray direction to be perpendicular to the motion direction. Let $\vec{p}_1\vec{p}_2$ represent the vector from path point p_1 to its adjacent path point p_2 , \vec{n}_1 represents the original spray direction of p_1 , $\vec{i}_{\{path\}}$, $\vec{j}_{\{path\}}$, $\vec{k}_{\{path\}}$ are the standard bases of the X axis, Y axis, and Z axis in $\{Path\}$ attached to p_1 , and θ is the angle between the original spray direction and the adjusted one:

$$\theta = \arccos\left(\vec{n}_1 \cdot \vec{j}_{\{path\}}\right) - \frac{\pi}{2} \tag{7a}$$

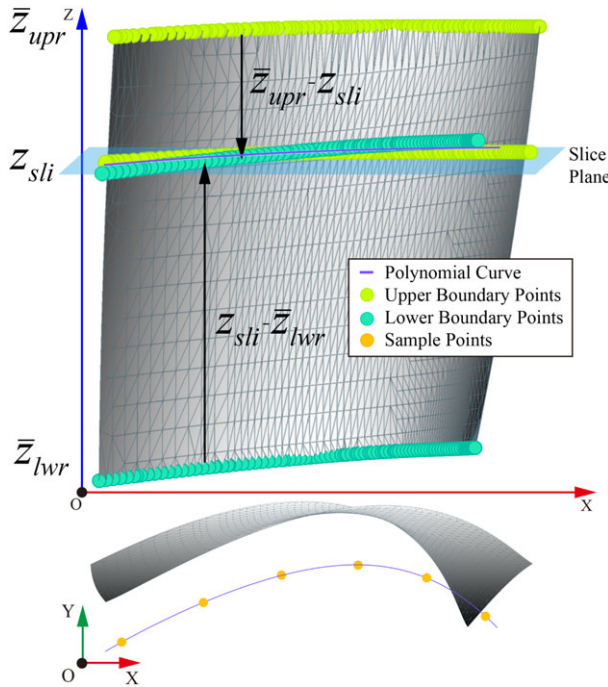


Figure 7. z - t and y - t polynomial curves. $x(t) = t$.

hence $\vec{k}_{\{path\}}$ is given by:

$$\vec{k}_{\{path\}} = \frac{R_{\vec{i}_{\{Path\}}}(\theta) \vec{n}_1}{|\vec{n}_1|} \tag{7b}$$

where $R_{\vec{i}_{\{Path\}}}(\theta)$ is the desired rotation matrix that rotates \vec{n}_1 about $\vec{i}_{\{path\}}$ to be orthogonal to $\vec{j}_{\{path\}}$. It could be obtained by:

$$R_{\vec{i}_{\{Path\}}}(\theta) = I_{3 \times 3} + \sin\theta \vec{i}_{\{Path\}} + (1 - \cos\theta) \vec{i}_{\{Path\}}^2 \tag{7c}$$

Therefore, $\{Path\}$ could be completely defined by a 4×4 transformation matrix [21]:

$$T = \begin{bmatrix} \vec{i}_{\{path\}} & \vec{j}_{\{path\}} & \vec{k}_{\{path\}} & P_{\{path\}} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7d}$$

where $P_{\{path\}}$ is the origin of $\{Path\}$. These coordinates are expressed from $\{W\}$, the world coordinate system. Figure 8 illustrates the definition of $\{Path\}$.

4. Optimization for thickness uniformity

4.1. Influencing factors

With parameters of the generated path and the model to calculate the resulting coating thickness, an optimization could be carried out to obtain the optimal uniformity of coating thickness. In spray painting, a variety of parameters could impact uniformity. Among them, spray distance, spray speed, and spacing distance are three of the most dominating factors related to path generation.

As discussed in Section 3.2, spray distance is introduced there to express the offset distance from the key point to the sample point along the normal vector. Hence, spray distance affects the shape of the pass, which fits the sample points. Besides, a higher spray distance would bring about a lower coating

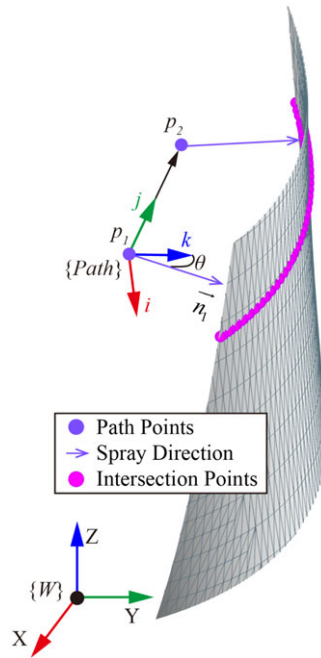


Figure 8. Definition of {Path}.

thickness and a wider coverage area, and vice versa. To gain a uniform coating thickness, it is workable to adjust spray distance for each key point.

Spray speed directly affects the dynamic model of the thickness simulation as shown in Eq. (4c). The higher the spray speed, the lower the coating thickness, and vice versa. In this study, the spray gun is assumed to move at a constant speed along a certain pass.

Differing from the above two parameters, spacing distance impacts the uniformity indirectly. A standard thickness distribution is in a volcano shape where the thickness value is higher in the middle while lower on the two tails. It makes the thickness uneven. To alleviate it, coating thickness distributions resulting from two adjacent passes should overlap each other as shown in Fig. 9. It would accumulate a higher coating thickness on the side of the coverage area, making the compositive thickness more uniform. In this scenario, spacing distance impacts the degree of overlap. Therefore, it also has an influence on the thickness uniformity.

4.2. Objective function

To evaluate the uniformity of coating thickness, the coefficient of variation is selected to be the indicator [11]:

$$c_v = \frac{\sigma_{\text{thk}}}{\mu_{\text{thk}}} \times 100\% \tag{8a}$$

where c_v is the coefficient of variation, σ_{thk} and μ_{thk} are the standard deviation and the mean of the coating thickness, respectively. As c_v is a dimensionless number, it is feasible to evaluate the uniformity among thickness datasets with different units and different desired values. If a specific desired thickness T_{desired} is determined, μ_{thk} is replaced by this value and σ_{thk} is revised to be the standard deviation from T_{desired} .

In spray painting, the overall coating thickness is accumulated pass by pass and the global uniformity of the target surface is a result of the local uniformity between each neighboring pass pair. To consider the relationship between the global uniformity and the local one, the objective function here is constructed

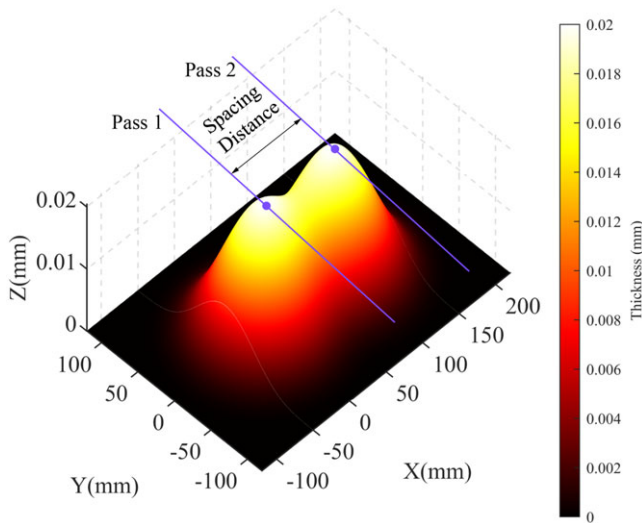


Figure 9. Two overlapped thickness distributions.

as a combination of them. For the first pass and the last pass, the global uniformity is selected as the objective function. It considers the thickness of the whole surface points. In terms of the pass within them, the corresponding objective function is the local uniformity, which only includes the surface points between the current pass and the last neighboring one. Hence, the final objective function is given by

$$\min f = \begin{cases} c_{v_{\text{global}}}, & \text{the first or the last pass} \\ c_{v_{\text{local}}}, & \text{otherwise.} \end{cases} \quad (8b)$$

Considering the parameters mentioned in Section 4.1, the optimization problem could be defined as:

$$\min f(H, v, d) \quad (8c)$$

where H is an array stores spray distance of each sample point on the current pass, v is the spray speed of the current pass, and d is the spacing distance between the current pass and the last pass. Figure 10 outlines the factors in the optimization.

4.3. Optimization method: PSO

Even though the problem has been defined in Eq. (8a), the objective function $\min f$ is non-continuous and not specific. Therefore, the original optimization method requires the derivative of the objective function is not workable in this case. To deal with it, a global search algorithm needs to be adopted, which only requires calculating the value of the objective function during running. Among global search algorithms, the PSO [22, 23] is preferred in spray painting problems owing to its convergence efficiency and feasibility for a large number of optimization variables [24]. Thus, it is selected to solve the optimization problem here.

PSO is a bio-inspired algorithm. It heuristically imitates a flock of birds when they forage in groups. In PSO, a particle is a position consisting of the optimization variables. In terms of the initialization of PSO, the algorithm would randomly form a swarm of particles within the search space. The number of particles in the swarm is defined as “size of swarm.” For each particle, the position with the minimal value of the objective function so far is called “particle’s best-known position.” For the whole swarm, the position with the minimal value so far among all of the particles is called “swarm’s best-known position.”

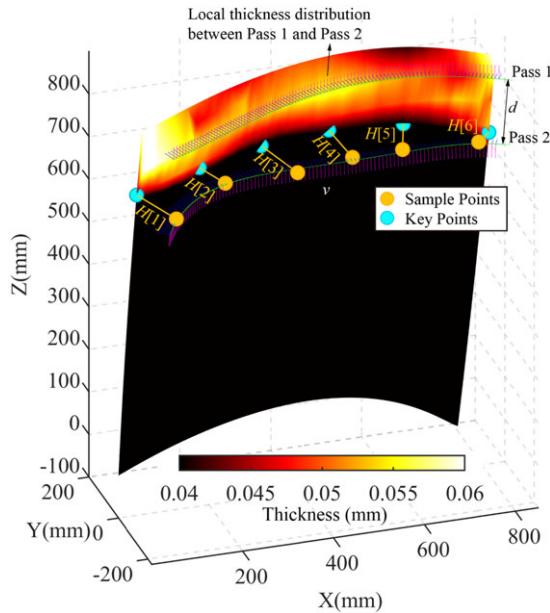


Figure 10. Factors in the optimization. The current pass is Pass 2. H is an array stores spray distance of each sample point on the current pass, v is the spray speed of the current pass, and d is the spacing distance between the current pass and the last pass.

At each step of PSO, a particle would move to a new position toward a certain direction. This direction is constructed by the vector toward the particle's best-known position, the vector toward the swarm's best-known position, and the vector from the last position to the current position. The influence degrees of these three factors are expressed by the cognitive coefficient (between [0,4]), social coefficient (between [0,4]), and inertia weight (between [0.1,1.1]). By this means, the position of each particles is iteratively updated. Finally, these particles would gradually converge around where the global minimum of the objective function is.

5. Simulation and discussion

To validate the performance of the proposed method, a simulation was conducted by comparing boundary fitting approach with the basic one explained in Section 2.1 and other typical algorithms. Zeng *et al.* [13]'s method and Guan and Chen [11]'s method were reproduced and applied there. Related programs were executed in MATLAB R2021b on a PC with two Intel Core i7-1165G7 2.80 GHz processors.

To ensure the adaptability, STL models of a concave surface and a convex surface in the shape of a turbine blade were imported as the target surfaces. They shared the same vertices while normal vectors were opposite to each others. The number of the vertices was 1219. Parameters of the thickness model required in Eq. (2) are given in Table I. Parameters of the spray painting process are given in Table II. In terms of path generation, spacing distance was set between [150,200] mm. The Z coordinate difference between the initial pass and the model's highest vertex was within [0,100] mm. The number of key points and path points on each pass was 6 and 100, respectively.

The results are shown in Figs. 11, 12, and Table III. In terms of the uniformity of coating thickness, Table III presents that the coefficient of variation c_v , resulting from boundary fitting approach is lower than the other methods. Meanwhile, Figs. 11 and 12 illustrate that the thickness distribution resulting from the proposed approach is also more uniform. Hence, the results verify that boundary fitting approach could improve the uniformity of coating thickness for robotic spray painting. The main reason

Table I. Parameters of the thickness calculation model.

$T_{\max}(\text{mm})$						
Concave	Convex	β_1	β_2	$a(\text{mm})$	$b(\text{mm})$	$h(\text{mm})$
0.20	0.13	2	2	150	30	100

Table II. Parameters of spray painting.

Desired thickness (mm)	Spray speed (mm/s)	Spray distance (mm)
0.05	[100,200]	[50,150]

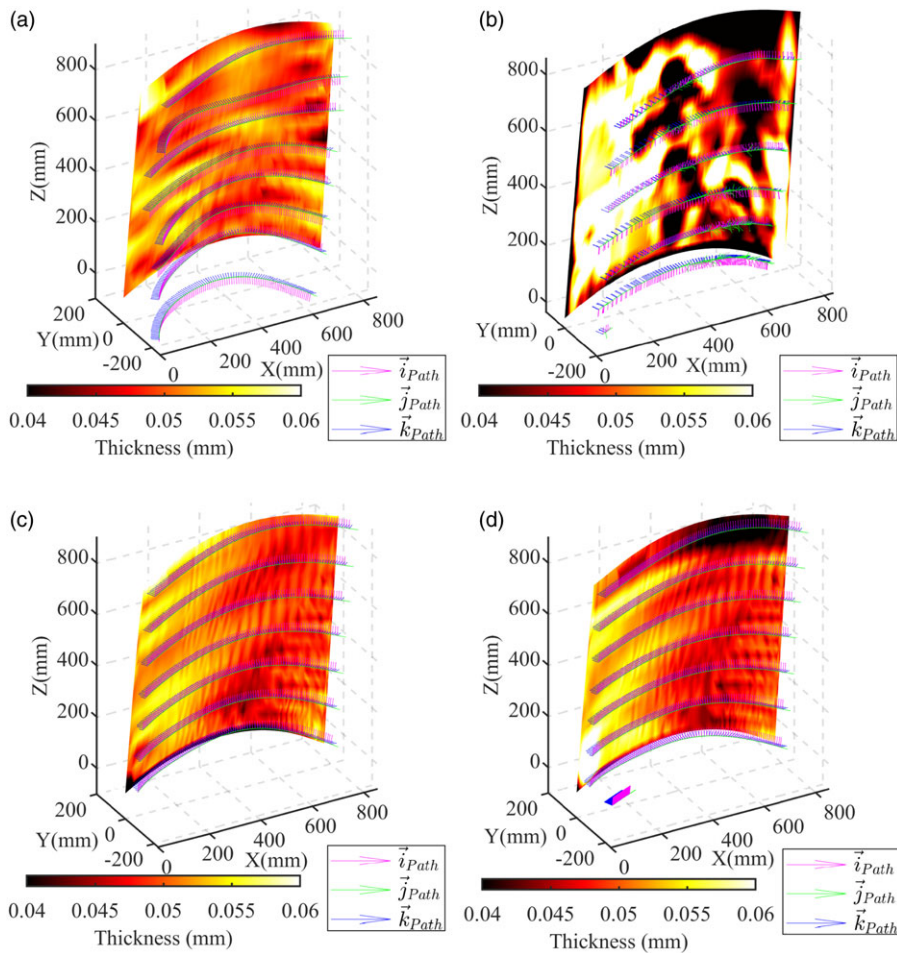


Figure 11. Generated path and resulting thickness for concave surface. (a) Boundary fitting approach. (b) Basic approach. (c) Zeng et al. [13]'s approach. (d) Guan and Chen [11]'s approach.

for the improvement is that boundary fitting approach enjoys richer flexibility in optimization than the other three methods. By modulating each sample point's spray distance, the generated pass could be more varied. Therefore, a more satisfactory thickness distribution is likely to be found. In comparison, the other methods could only adjust the spray distance of a whole pass. Besides, the pass generated by

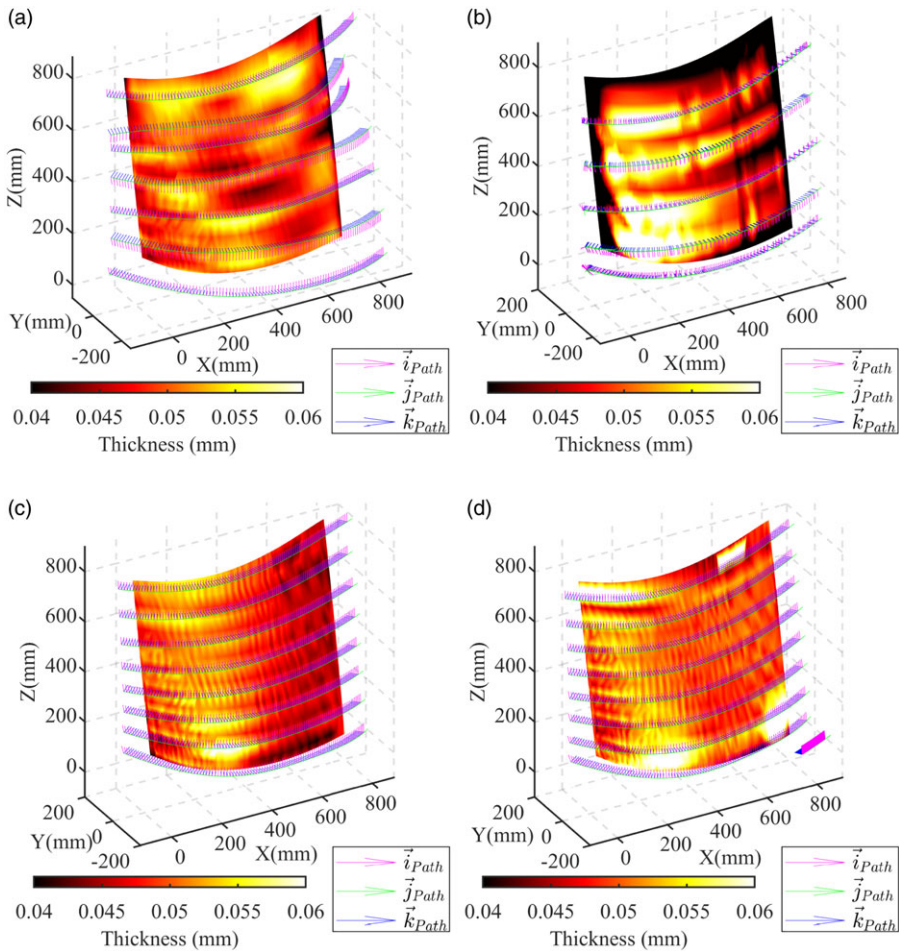


Figure 12. Generated path and resulting thickness for convex surface. (a) Boundary fitting approach. (b) Basic approach. (c) Zeng et al. [13]'s approach. (d) Guan and Chen [11]'s approach.

boundary fitting approach could even exceed the target surface's boundary. It also brings about more possibilities for optimization.

In terms of the boundary fitting demand, Figs. 11 and 12 highlight that the path generated by the proposed approach avoids the boundary effect. The pass around the boundary is complete. Nevertheless, the basic method and Guan and Chen's method only generate few path points around the boundary, as shown in Figs. 11(b) and 12(b). In terms of the path smoothing demand, Figs. 11 and 12 depict that the path points generated by the proposed approach are smooth and uniform. It facilitates the robot control. However, the points generated by the basic method are discrete and fluctuating. Furthermore, for both of the concave surface and the convex surface, the results show the same tendency. It validates the adaptability of boundary fitting approach.

As discussed in Section 4.3, PSO is adopted to find the optimal uniformity of coating thickness. To judge its optimization performance, other global search algorithms were also tested in the simulation to draw a comparison. Simulated annealing, genetic algorithm, and pattern search were selected there. They were executed to solve the same optimization problem in boundary fitting approach with the same maximum number of iterations. Both concave surface and convex surface are considered. Table IV gives the parameters of each method. The results are shown in Table V. It can be seen that PSO enjoys the

Table III. Results of uniformity indicators.

Model	Approach	Average thickness (mm)	c_v (%)
Concave surface	Boundary fitting	0.0506	6.59
	The basic	0.0479	27.28
	Zeng <i>et al.</i> 's	0.0494	10.79
	Guan and Chen's	0.0492	9.56
Convex surface	Boundary fitting	0.0499	7.04
	The basic	0.0471	26.06
	Zeng <i>et al.</i> 's	0.0495	7.60
	Guan and Chen's	0.0513	9.55

Table IV. Parameters of optimization methods.

Parameter	Value	Parameter	Value
Size of swarm	20	Initial temperature	100
Cognitive coefficient	1.49	Reannealing interval	100
Social coefficient	1.49	Temperature update coefficient	0.99
Inertia weight	1	Maximum iteration	100
Maximum iteration	100		
(a)		(b)	
Parameter	Value	Parameter	Value
Population size	20	Initial mesh size	1
Crossover rate	0.7	Mesh contraction factor	0.5
Mutation rate	0.01	Mesh expansion factor	2.0
Maximum iteration	100	Maximum iteration	100
(c)		(d)	

(a) PSO. (b) Simulated annealing. (c) Genetic algorithm. (d) Pattern search.

Table V. Results of the optimization method comparison.

Method	Model	Running time (s)	Average thickness (mm)	c_v (%)
PSO	Concave surface	4349.64	0.0506	6.59
	Convex surface	3770.95	0.0499	7.04
Simulated annealing	Concave surface	442.88	0.0495	13.53
	Convex surface	418.65	0.0496	12.56
Genetic algorithm	Concave surface	2052.55	0.0486	10.86
	Convex surface	2239.37	0.0499	12.00
Pattern search	Concave surface	3233.69	0.0494	7.57
	Convex surface	3195.37	0.0498	7.65

best performance in terms of the coating thickness uniformity. The coefficients of variation are only 6.59% and 7.04% for the concave surface and convex surface, respectively. However, its running time is somewhat lengthy compared with other methods. It indicates that further research should be conducted to improve the efficiency of the optimization algorithm.

6. Conclusion

To improve the uniformity of coating thickness, this paper proposes an improved path planning algorithm called “boundary fitting approach” for robotic spray painting. It differs from the basic slicing method in considering the geometric boundary of the target surface as well as the smoothness of the generated path. The ellipse dual- β distribution model is used here to simulate the resulting coating thickness. The path is expressed by a fourth-order polynomial curve that fits the boundary points and sample points. The boundary points are detected according to the topology of the STL model and the sample points are extracted from the intersection points with a uniform length interval. The orientation of each path point is also defined by the spray gun’s motion direction and spray direction. Based on PSO, the parameters of the path are optimized pass by pass. By constructing a compound objective function, both of the global uniformity and local uniformity are considered in the optimization. Comparing the optimized thickness distribution with other methods, it indicates that the proposed approach meets the boundary fitting demand and significantly improves the uniformity of coating thickness.

Notwithstanding the progress, several issues remain to be tackled. Firstly, the efficiency of the optimization method needs further improvement. To concentrate on the optimal coating uniformity, other indexes to estimate the quality of spray painting were not considered in this paper, such as paint waste, cycle time, and latent surface failure. Hence, there is still a fruitful area for further work to validate and improve this algorithm.

Author contributions. Hongrui Wu and Qing Tang designed the algorithm, developed the program, and wrote the article.

Financial support. This work was supported by Sichuan Science and Technology Program under Grant 2020YFG0116 and Grant 2021YFG0074.

Conflicts of interest. The authors declare no conflicts of interest exist.

Ethical approval. Not applicable.

References

- [1] B. Zhang, J. Wu, L. Wang and Z. Yu, “Accurate dynamic modeling and control parameters design of an industrial hybrid spray-painting robot,” *Robot. Comput.-Integr. Manuf.* **63**, 101923 (2020).
- [2] J. Wu, X. Wang, B. Zhang and T. Huang, “Multi-objective optimal design of a novel 6-dof spray-painting robot,” *Robotica* **39**(12), 2268–2282 (2021).
- [3] P. N. Atkar, D. C. Conner, A. Greenfield, H. Choset and A. A. Rizzi, Uniform coverage of simple surfaces embedded in r^3 for auto-body painting, (2004).
- [4] S. H. Deng, Z. H. Cai, D. D. Fang, H. L. Liao and G. Montavon, “Application of robot offline programming in thermal spraying,” *Surf. Coat. Technol.* **206**(19), 3875–3882 (2012).
- [5] M. Kong, A. Li and F. Gao, “A Trajectory Planning Algorithm of Spray RObot for Workpieces with Complex Geometric Features Based on 3D Point Cloud,” *2021 IEEE 5th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Vol. 5, (2021) pp. 13–20.
- [6] D. Gleeson, S. Jakobsson, R. Salman, N. Sandgren, F. Edelvik, J. S. Carlson and B. Lennartson, “Robot Spray Painting Trajectory Optimization,” *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, (2020) pp. 1135–1140.
- [7] X. Yu, Z. Cheng, Y. Zhang and L. Ou, “Point cloud modeling and slicing algorithm for trajectory planning of spray painting robot,” *Robotica* **39**(12), 2246–2267 (2021).
- [8] P. N. Atkar, A. Greenfield, D. C. Conner, H. Choset and A. A. Rizzi, “Uniform coverage of automotive surface patches,” *Int. J. Robot. Res.* **24**(11), 883–898 (2016).
- [9] J. N. P. Heping Chen, T. Fuhlbrigge and X. Li, “A review of cad-based robot path planning for spray painting,” *Ind. Robot.* **36**(1), 45–50 (2009).
- [10] H. Chen and N. Xi, “Automated tool trajectory planning of industrial robots for painting composite surfaces,” *Int. J. Adv. Manuf. Technol.* **35**(7–8), 680–696 (2006).
- [11] L. Guan and L. Chen, “Trajectory planning method based on transitional segment optimization of spray painting robot on complex-free surface,” *Ind. Robot.* **46**(1), 31–43 (2019).
- [12] M. V. Andulkar, S. S. Chiddarwar and A. S. Marathe, “Novel integrated offline trajectory generation approach for robot assisted spray painting operation,” *J. Manuf. Syst.* **37**, 201–216 (2015).

- [13] Y. Zeng, Y. Yu, X. Zhao, Y. Liu, J. Liu and D. Liu, “Trajectory planning of spray gun with variable posture for irregular plane based on boundary constraint,” *IEEE Access* **9**, 52902–52912 (2021).
- [14] J.-H. Kao and F. B. Prinz, “Optimal Motion Planning for Deposition in Layered Manufacturing,” *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers, vol. 80364, (2021) p. V006T06A018.
- [15] D. Gleeson, S. Jakobsson, R. Salman, F. Ekstedt, N. Sandgren, F. Edelvik, J. S. Carlson and B. Lennartson, “Generating optimized trajectories for robotic spray painting,” *IEEE Trans. Autom. Sci. Eng.* **19**(3), 1380–1391 (2022).
- [16] D. Hegels, T. Wiederkehr and H. Müller, “Simulation based iterative post-optimization of paths of robot guided thermal spraying,” *Robot. Comput.-Integr. Manuf.* **35**, 1–15 (2015).
- [17] Y. Zhang, Y. Huang, F. Gao and W. Wang, “New model for air spray gun of robotic spray-painting,” *Chin. J. Mech. Eng.* **2006**(11), 226–233 (2006).
- [18] D. C. Conner, P. N. Atkar, A. A. Rizzi and H. Choset, “Development of Deposition Models for Paint Application on Surfaces Embedded in \mathbb{R}^3 for Use in Automated Path Planning,” *IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, vol. 2, (2002) pp. 1844–1849.
- [19] M. A. S. Arkan and T. Balkan, “Process modeling, simulation, and paint thickness measurement for robotic spray painting,” *J. Robot. Syst.* **17**(9), 479–494 (2000).
- [20] L. Noakes and T. Popiel, “Geometry for robot path planning,” *Robotica* **25**(6), 691–701 (2007).
- [21] X. Liu, Y. Xu, B. Zhou and F. Fang, “Automatic Programming System of Painting Robots for Unstructured Workpieces Spraying,” *2021 36th Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, (2021) pp. 696–701.
- [22] D. Wang, D. Tan and L. Liu, “Particle swarm optimization algorithm: An overview,” *Soft Comput.* **22**(2), 387–408 (2018).
- [23] S. Bouraine and O. Azouaoui, “Safe motion planning based on a new encoding technique for tree expansion using particle swarm optimization,” *Robotica* **39**(5), 885–927 (2021).
- [24] B. Guo, W. Chen, J. Zhang and Q. Wei, “Optimization of Auto Lamp Spraying Trajectory Based on Improved PSO Algorithm,” *2021 China Automation Congress (CAC)*, (2021) pp. 7487–7492.