Improvement of flattenability using particle swarm optimizer for surface unfolding in bolus shaping

Rui Li¹ · Qingjin Peng¹ · Harry Ingleby² · David Sasaki²

Received: 3 December 2019 / Accepted: 12 August 2020 / Published online: 18 August 2020
© Springer Nature Switzerland AG 2020

Abstract

Bolus, formed by a sheet of material, covers on tumors of patient skins to generate the desired dose distribution in a process of the high-energy radiotherapy. The existing methods of bolus shaping use a manual process to cut the commercial material into the shape, which can cause air gaps between the patient skin surface and bolus in the reduced effect of cancer care treatment. A method of automatic bolus shaping can improve the accuracy and reduce air gaps of bolus shaping, which uses a process of first segmenting 3D patient skin surface models into several patches and then unfolding the patches into 2D patterns for material cutting before folding the cut sheet into the bolus. However, air gaps between the bolus and patient skin surface cannot be directly evaluated to find the bolus accuracy. This paper proposes a method to improve model flattenability and evaluate the air gaps using a particle swarm optimizer (PSO). A 3D surface shape is unfolded only if the air gap and surface flattenability meet the requirement of accuracy. Otherwise, the surface will be segmented into several patches to improve the flattenability and reduce air gaps. The objective and constraint are identified to search for an optimal solution for the 3D surface with a high flattenability. A strategy of the dimension reduction is proposed for the selection of local nodes on a meshed surface to increase the searching efficiency of surface flattenability. A local node selection based PSO (L-PSO) method is developed to search for the optimal solution. The proposed method is verified in two case studies of bolus shaping.

Keywords Surface optimization · Flattenability · Bolus · Particle swarm optimizer (PSO)

1 Introduction

A bolus is used to cover patient skin surfaces during the high-energy radiotherapy for the desired dose distribution [1]. In the clinic, commercial materials are used in bolus shaping by cutting the material into the shape of a targeted body area to cover the area. For some irregular body surfaces such as knee, nose, and elbow, the manual process of bolus shaping will cause air gaps between the human surface and bolus, which reduces the therapeutic effect [2].

Inspiring by applications of digital 3D technologies in the garment manufacturing process, a 3D-to-2D and

2D-to-3D process was developed for the bolus shaping to reduce air gaps and increase shaping efficiency [3, 4]. Human surface data are obtained using a 3D scanner to build a mesh surface that is a kind of discrete surfaces and most of them are non-developable that cannot be unfolded into 2D patches without deformation and distortion. The 3D triangular mesh surface is segmented into several patches to increase flattenability and then to be unfolded into 2D patches for shaping bolus. Commercial materials can then be cut into the bolus shape based on formed 2D patches.

One weakness of the bolus shaping by surface unfolding is that it can only decide the deformation and

✉ Qingjin Peng, Qingjin.peng@umanitoba.ca | ¹Department of Mechanical Engineering, University of Manitoba, Winnipeg, MB, Canada. ²CancerCare Manitoba, Winnipeg, MB, Canada.

distortion between a 3D shape and 2D patches rather than estimate air gaps between the bolus and human surface. A triangular mesh surface can be improved for a flattenable mesh surface that can generate 2D patches without deformation and distortion [5]. The difference between the original surface and improved mesh surface can be used to estimate air gaps between an original surface and its refolded surface.

This paper proposes a method to improve the flattenability of a non-developable surface and estimate air gaps in bolus shaping. If the air gaps and flattenability meet the requirement of accuracy, the improved surface will be unfolded directly. Otherwise, the original surface will be segmented into several patches to improve the flattenability. The process continues until the air gaps and flattenability meet the requirement of accuracy.

The surface improvement method proposed in this paper is based on the particle swarm optimizer (PSO) method. An objective function is built by considering four requirements and one constraint including the minimum deformation and distortion, maximum smoothness, maintaining the original surface area, minimum air gaps, and constraint of nodes moved in the positive direction of the normal vector of an outer surface. To reduce dimensions of the optimization, a local node selection strategy based the PSO (L-PSO) method is developed to achieve the objective function.

Following parts of this paper are structured as follows. Section 2 reviews related methods of the surface flattenability improvement and optimization algorithms. Section 3 describes the proposed method for surface improvement. Details of the optimization search process are discussed in Sect. 4. Section 5 is case studies, followed by the summary and further work in Sect. 6.

2 Literature review

2.1 Surface flattenability improvement

A developable surface is ruled by $X(t, v) = \alpha(t) + v\beta(t)$ from differential geometry. If β , $\dot{\beta}$ and $\dot{\alpha}$ are coplanar for all points on a ruled surface, the surface is developable [6]. In other words, if the Gaussian curvature of each point in a surface is zero, the surface is a developable surface. Developable surfaces, such as planes and generalized cylinders, can be unfolded without any deformation and distortion.

However, most of the surfaces in nature are non-developable surfaces. When unfolding this kind of surfaces directly, it will cause deformation and distortion. A flattenable mesh surface is also called a discrete developable surface by adopting the definition of the Gaussian curvature in the discrete differential geometry.

For increasing surface flattenability, the main idea is to convert a discrete non-developable mesh surface into the flattenable mesh surface or discrete developable surface. Liu et al. presented PQ meshes by using the constrained optimization framework. However, only simple shapes can be modeled into developable surfaces by using PQ meshes [7]. Liu et al. proposed a dynamic programming method to generate free-form developable surfaces by optimizing a tangent-variation objective globally and then proposed a simple sketching interface [8]. Zeng et al. generated an optimal quasi-developable mesh patch from an efficient least-squares solution using a measure defined in an extended Gaussian image [9]. Lee et al. generated feature curves from a 3D point clouds model and then used the feature curves to build developable surfaces that are similar to the 3D point clouds model [10]. Chu et al. proposed a geometric design method using developable Bézier patches to generate a developable strip. An aggregate of triangular and quadrilateral patches is generated by interpolating a strip specified by two space curves [11]. Fernández-Jambrina proposed a linear algorithm to construct B-spline control nets for spline developable where five free parameters were applied to control vertices [12]. Developable surfaces were constructed with Bézier patches using Aumann's algorithm to form a polynomial edge of regression [13].

Wang and Tang defined the discrete definition of Gaussian curvature [14] to increase the flattenability of a given mesh surface using a constrained optimization method [15]. They presented a Flattenable Laplacian (FL) mesh modeling scheme to generate a stable computation process for complicated shapes. Zhang et al. extended Wang's method using an optimal fitting surface algorithm to reduce errors in the process of optimizing discrete non-developable surfaces for flattenable mesh surfaces [16].

In summary, for using surface unfolding to shape the bolus, the original surface is unfolded to generate 2D shapes that are then used to build the bolus to cover the human surface. If the shape of an original surface is changed to increase flattenability, the difference between the original surface and improved surface should be evaluated to meet the requirement. This paper proposes a method using the idea of moving nodes on the original surface to improve flattenability of the original surface and estimate the difference between the original surface and improved surface as air gaps.

2.2 Particle swarm optimizer

Conventional optimization methods, such as the gradient descent method, Newton's method and Quasi-Newton method, are widely used in different problems. The gradient descent method is an easy and simple optimization

method for calculating the derivative of the objective function. Newton's method uses the first few items in a Taylor series of the objective function to search the maximum or minimum value of an optimization problem, which needs to calculate the first and second derivatives of the objective function. The Quasi-Newton method was proposed to reduce the computational complexity by using a positive definite matrix to approximate the Hessian matrix. The gradient descent method, Newton's method, and the Quasi-Newton method need to calculate the derivative of the objective function, which is complicated for some optimization problems [17, 18]. In addition, these methods only have good performance when the initial input is close to the maximum or minimum result.

Modern optimization algorithms, such as neural networks, genetic algorithms and PSO, have the good performance in solving optimization problems without calculating derivatives of the objective function and have the low initial sensitivity.

Particle swarm optimizer is an evolutionary computation algorithm [19] inspired by the social behavior of bird flocking to generate an optimal solution in complex search spaces [20]. Based on advantages of the easy implementation and fewer parameters required for adjustment, PSO is widely used in the functional optimization, neural network training, fuzzy system control, and other applications of evolutionary algorithms.

Yang et al. proposed a new discrete particle swarm optimization algorithm based on the quantum individual, which is simpler than the existing algorithm. The simulation experiment and its application in the code division multiple access also prove its high efficiency [21]. Sun et al. proposed a characterization method for the high-loss piezoelectric composite material based on the particle swarm optimization algorithm [22]. Shahbeig et al. proposed a hybrid algorithm to determine the development of the most relevant involved genes in breast cancer. The minimal subset of genes involved in breast cancer was found based on combined Teaching–Learning–Based Optimization (TLBO) and the proposed mutation fuzzy adaptive particle swarm optimization algorithm. The results show that the proposed technique is able to achieve 91.88% accuracy [23]. Tang et al. used an improved particle swarm

optimization to search the optimal solution in dynamic flexible flow shop scheduling problems [24].

Researchers also proposed other methods to improve the PSO method. Zhang et al. proposed an improved PSO algorithm for the parameter identification of nonlinear dynamic hysteresis models. Research shows that this improved method reduces the influence of randomness [25]. Liang and Suganthan proposed a hybrid strategy by adding a local search technique in a dynamic multi-swarm PSO to increase the ability of the local search [26]. Fan and Zahara [27] used PSO for unconstrained optimization by integrating the simplex search method into iterations. Coelho and Mariani [28] proposed a novel chaotic PSO combined with an implicit filtering local search method to solve economic dispatch problems. To improve the local search ability of PSO, Li et al. presented a hybrid global optimization strategy to combine PSO with a modified Broyden–Fletcher–Goldfarb–Shanno (BFGS) method. In addition, to improve the global search ability of PSO, a reposition technique was proposed to maintain the diversity of particles using territory techniques [29]. Luo et al. proposed a hybrid intelligent model combined with Support Vector Regression (SVR) and PSO for an optimal control of parameters in prediction of the phytoplankton dynamics [30].

In summary, PSO provides the good performance to optimize data without calculating derivatives of the objective function. In addition, the initial input of PSO does not have much influence on the final result and convergence rate. Thus, an optimization method based on PSO is proposed in this research to improve an original 3D surface to increase its flattenability. A local nodes selection strategy is combined with PSO to increase the optimization efficiency.

3 Proposed methods

A flowchart of the proposed method is shown in Fig. 1. An original 3D surface shape is processed for improvement to increase flattenability. Air gaps are evaluated using the difference between an original surface and the improved surface. If the air gaps and flattenability meet the requirement

Fig. 1 A flowchart of the proposed method

of accuracy, the improved surface will be unfolded into 2D patches directly. Otherwise, the improved surface will be segmented into several patches. Each patch of the 3D surface will be segmented until the air gaps and flattenability meet the requirement of accuracy. Details are discussed as follows.

3.1 Surface improvement

A 3D mesh surface M can be defined as triples V, E, F where V is a set of vertices $V = \{v_i | v_i \in R^3, 1 \leq i \leq m\}$, m denotes the number of vertices. E is a set of edges $E = \{e_{ij} = (v_i, v_j) | v_i, v_j \in V, i \neq j\}$, and F is a set of faces $F = \{f_{ijk} = (v_i, v_j, v_k) | v_i, v_j, v_k \in V, i \neq j, j \neq k, i \neq k\}$.

Searching improved surface M_{Imp} from M can be considered as a constrained optimization problem. The improved surface M_{Imp} should approximate to the shape of original triangular mesh surface M . The improved positions v_p of a vertex with original position v_p^0 is an ideal position

Fig. 2 Relationship between an original mesh surface and improved mesh surface

Fig. 3 Deformation of different inner triangular meshes

satisfying the optimization constraint. The relationship between v_p and v_p^0 can be presented as: $v_p = v_p^0 + d_p$. Relationships between an original mesh surface and the improved mesh surface are shown in Fig. 2 using a triangular mesh.

To reduce air gaps between the bolus and original surface in bolus shaping, when using an improved surface to instead the original surface, the shape of the improved surface should be similar to the original surface. The improved surface is used to form a 3D bolus to cover the human skin surface. Thus, all the vertices in the original mesh surface should be moved in the positive direction of normal vectors of an outer surface. The moving distance can be calculated as air gaps between the original surface and bolus. As both the original surface and bolus are continuous surfaces, the original surface should be converted into a discrete mesh surface to be as smooth as a continuous surface.

In short, the process uses four requirements and one constraint to search the improved surface with a high flattenability for bolus shaping as follows. (1) Minimum deformation and distortion, (2) Maximum smoothness, (3) Maintaining the original surface area, (4) Minimum air gaps, and one constraint of moving nodes on the original surface in the positive direction of normal vectors of an outer surface.

3.2 Flattenable mesh surface and flattenability

Based on definition of the flattenable mesh surface [5], the summed inner angle $\theta(v_p)$ of an inner vertex v_p in a triangular mesh surface is used to measure flattenability as shown in Fig. 3. For a flattenable inner vertex, the summed inner angle $\theta(v_p) = 2\pi$. Triangles around v_p can be unfolded into 2D patches without distortion and deformation. If the summed inner angle $\theta(v_p) \neq 2\pi$, 2D patches

will have distortion and deformation when unfolding the triangles around v_p . If $\theta(v_p) > 2\pi$, unfolded 2D patches will generate overlaps when unfolding triangles around v_p without distortion and deformation. If $\theta(v_p) < 2\pi$, the unfolded 2D patches will generate gaps when unfolding triangles around v_p without distortion and deformation.

$\omega(v_p)$ at v_p shown in Eq. (1) is defined to measure the flattenability in v_p .

$$\omega(v_p) = |\theta(v_p) - 2\pi| \quad (1)$$

A smaller $\omega(v_p)$ at vertex v_p in a triangular mesh surface means that the surface can be unfolded into 2D patches with less distortion and deformation. Flattenability of a surface is calculated by Eq. (2).

$$FLA = 10 \times 0.5 \sum |\theta(v_p) - 2\pi| \quad (2)$$

Therefore, for a triangular mesh 3D surface M , if summed inner angles for all inner vertices are equal to 2π , surface M can be unfolded into 2D patches without deformation and distortion for each triangle. Thus, for an improved 3D surface M_{inp} , if its summed inner angles are closer to 2π , surface M_{inp} can be unfolded into 2D patches with the less deformation and distortion. To generate an improved surface for good flattenability, the surface should satisfy Eq. (3).

$$\min \sum_{v_p \in V_{in}} |\theta(v_p) - 2\pi| \quad (3)$$

3.3 Laplacian mesh surface

The Laplacian mesh is widely used to remove rough features from irregularly triangular mesh surfaces for maintaining desirable geometric features [31]. For a mesh surface expressing with triples V, E, F , v_p denotes the position of vertex v_p in a 3D coordinate as follows.

$$v_p - \frac{1}{|N(v_p)|} \sum_{j \in N(v_p)} v_p = 0 \quad (4)$$

where $N(v_p)$ is a set of 1-ring neighboring vertices of v_p , v_p is an interior vertex ($v_p \in V_{in}$), and $|N(v_p)|$ denotes the number of elements in the set of 1-ring neighboring vertices of v_p . The linear system can be rewritten in the matrix as Eq. (5)

$$\begin{aligned} Lx &= 0 \\ Ly &= 0 \\ Lz &= 0 \end{aligned} \quad (5)$$

where x, y and z are $n \times 1$ vectors that contain x, y and z coordinates of the vertices. Matrix L is a Laplacian operator as shown in Eq. (6).

$$L_{ij} = \begin{cases} 1 & i = j \\ -\frac{1}{|N(v_p)|} & j \in N(v_p) \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

When fixing boundary vertices, coordinates of inner vertices on a Laplacian mesh patch are decided by the membrane functional minimization $\int_{\Omega} \|v_{\mu}\|^2 + \|v_v\|^2 d\mu dv$ [31]. The optimized mesh surface from Eq. (5) is a smooth mesh surface which is uniquely defined by the original surface and position of boundary vertices. However, a Laplacian mesh surface is not a flattenable mesh surface. To generate a smooth mesh surface, the optimization function derived from the Laplacian mesh is shown in Eq. (7).

$$\min \sum_{v_p \in V_{in}} \|Lv_p\|^2 \quad (7)$$

3.4 Area difference and air gaps

In order to cover the human body surface with the minimum air gap, the area of the unfolded surface or improved surface should be as same as the area of the original surface to form the bolus. Area accuracy is applied to evaluate the area difference between a 3D shape and the improved 3D shape [32]. To evaluate the difference between the original surface and improved surface, area accuracy ε_A is defined in Eq. (8).

$$\varepsilon_A = \frac{\sum |A_f - A_f^0|}{\sum A_f^0} \quad (8)$$

where A_f^0 is the area of triangle f in original surface M , A_f is the area of triangle f in improved surface M_{inp} . To generate the improved surface similar to the original surface area, areas of triangles in two surfaces should satisfy Eq. (9).

$$\min \frac{\sum |A_f - A_f^0|}{\sum A_f^0} \quad (9)$$

Air gaps between the improved surface and original surface can be calculated as the distance between new positions and original positions of vertices. To generate the improved surface with smaller air gaps, the distance between new positions and original positions of vertices should satisfy Eq. (10).

$$\min \sum_{v_p \in V_{in}} \|v_p - v_p^0\|^2 \quad (10)$$

3.5 Objective function of improved surface

As the bolus is a piece of material covering on the skin surface of patients, when improving a surface to shape the bolus, nodes in the surface should be moved in the positive direction of normal vectors of an outer surface. Thus, the original coordinate of each node is transferred to a new coordinate where the positive direction of the z-axis is the normal vector of the node. A constraint is therefore added in the optimization model to ensure that all the nodes are moved in the positive direction of the z-axis. The computational scheme for the improved surface with four defined requirements and one constraint is formulated into a constrained multi-objective optimization problem as shown in Eq. (11).

$$\begin{aligned} & \min \sum_{v_p \in V_{in}} \|v_p - v_p^0\|^2 \\ & \min \sum_{v_p \in V_{in}} |\theta(v_p) - 2\pi| \\ & \min \sum_{v_p \in V_{in}} \|Lv_p\|^2 \\ & \min \frac{\sum |A_f - A_f^0|}{\sum A_f^0} \end{aligned} \quad (11)$$

$$\begin{aligned} & \text{s.t. } \|v_p - v_p^0\|^2 = 0 \quad v_p \notin V_{in} \\ & v'_{pz} - v^0_{pz} > 0 \end{aligned}$$

where V_{in} is a set of inner vertices that are allowed to move during the optimization search, and the boundary vertices are fixed. v'_{pz} and v^0_{pz} are direction of normal vectors for v_p and v_p^0 , respectively. The normal vector of a vertex v_p is estimated by Eq. (12)

$$N_{v_p} = \frac{\sum_{f_k \in F(v_p)} A_{f_k} N_{f_k}}{\left\| \sum_{f_k \in F(v_p)} A_{f_k} N_{f_k} \right\|} \quad (12)$$

Fig. 4 Normal vector of the triangle

Fig. 5 Candidate transformation process

where $F(v_p)$ is a one-ring neighbor face of v_p , A_{f_k} is the area of triangle f_k , N_{f_k} is the normal vector of triangle f_k , calculated by Eq. (13) as shown in Fig. 4.

$$N_{f_k} = \frac{e_{ij+1} \times e_{ij}}{\|e_{ij+1} \times e_{ij}\|} = \frac{(v_i - v_{j+1}) \times (v_i - v_j)}{\|(v_i - v_{j+1}) \times (v_i - v_j)\|} \quad (13)$$

where e_{ij+1} is the edge vector from v_{j+1} to v_p , e_{ij} is the edge vector from v_j to v_p .

A spatial coordinate transformation is applied to transfer the original coordinate of each node to a new coordinate where the positive direction of the z-axis is the normal vector of the node. The transformation process is shown in Fig. 5 and Eq. (14).

$$\begin{bmatrix} v'_{px} \\ v'_{py} \\ v'_{pz} \end{bmatrix} = R(\alpha)R(\gamma) \begin{bmatrix} v_{px} \\ v_{py} \\ v_{pz} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_{px} \\ v_{py} \\ v_{pz} \end{bmatrix} \quad (14)$$

where α and γ are calculated by Eqs. (15) and (16), respectively.

$$\alpha = -\arccos \left(\frac{N_{vpx}}{\sqrt{N_{vpx}^2 + N_{vpy}^2 + N_{vpz}^2}} \right) \quad (15)$$

$$\gamma = -\arccos \left(\frac{N_{vpy}}{\sqrt{N_{vpx}^2 + N_{vpy}^2}} \right) \quad (16)$$

Equation (11) can be transferred into a constrained optimization problem as Eq. (17).

$$\begin{aligned} J(X) = & \mu_1 \sum_{v_p \in V_{in}} |\theta(v_p) - 2\pi| + \mu_2 \sum_{v_p \in V_{in}} \|L\mathbf{v}_i\|^2 + \mu_3 \sum_{v_p \in V_{in}} \|\mathbf{v}_p - \mathbf{v}_p^0\|^2 + \mu_4 \frac{\sum |A - A_0|}{\sum A_0} \\ \text{s.t. } & \|\mathbf{v}_p - \mathbf{v}_p^0\|^2 = 0 \\ & v_p \notin V_{in} v_{pz} - v_{pz}^0 > 0 \end{aligned} \quad (17)$$

Four coefficients μ_1 , μ_2 , μ_3 and μ_4 are weights of each part in the optimization model and assigned by users. In this paper $\mu_1=10$, $\mu_2=1.0$, $\mu_3=0.1$ and $\mu_4=10$ are used for all examples. The value of μ_3 must be much smaller than μ_2 for the system stability [4]. Additionally, for increasing the surface flattenability, μ_1 should be larger than μ_2 and μ_3 . The order of magnitudes for area accuracy is less than other requirements, because the area accuracy is measured in percentage. Thus, μ_4 should be larger than μ_2 and μ_3 .

4 Optimization search process

4.1 Local node selection strategy

As a large number of nodes in a triangular meshed surface, the optimization search process is normally a high-dimensional computational problem. To improve the efficiency of the optimization process, a local node selection strategy is proposed in the optimization process to decide initial pre-optimized nodes. The local vertices selection strategy is the evaluation of nodes that contribute large $\omega(v_p)$ for the surface.

$\omega(v_p)$ calculated by Eq. (1) is used to search the contribution of each node in the surface improvement process. These nodes with the large value of $\omega(v_p)$ are selected as high contribution nodes.

Fig. 6 Node movements in different conditions. **a** Node movements with $\theta(v) - 2\pi < 0$; **b** Node movements with $\theta(v) - 2\pi > 0$

Based on the requirement of bolus covering on the human surface, all the nodes in the original surface should be moved only to the positive direction of the z-axis. If a node meets $\theta(v) - 2\pi > 0$, the node will be moved to the positive direction of the z-axis to improve the flattenability as shown in Fig. 6(b). While if a node $\theta(v) - 2\pi < 0$, the node will be moved to the negative direction of the z-axis to improve flattenability. One-ring neighbor nodes for a node with $\theta(v) - 2\pi < 0$ can be moved to the positive direction of the z-axis to improve the flattenability as shown in Fig. 6a.

Based on the local vertices selection strategy, nodes with the high contribution of improving flattenability are moved at each step. The process of the local vertices selection strategy is shown in Algorithm 1 as follows.

- (1) $\omega(v_p)$ is calculated using Eq. (1) for each node in the original surface. Nodes with the top 30% large values of $\omega(v_p)$ are selected as reference nodes.
- (2) Reference nodes with $\theta(v) - 2\pi > 0$ are selected as pre-optimized nodes.
- (3) Reference nodes with $\theta(v) - 2\pi < 0$ are left in the reference nodes set, and one-ring neighbor nodes of them are selected as candidate nodes. If nodes in the candidate nodes set are not in the reference nodes set, they are selected as pre-optimized nodes to move at each step in iteration.
- (4) After several iterations, some nodes have been moved to the positive direction of the z-axis. When updating initial input nodes by the local

Algorithm 1: local node selection strategy

```

1: for  $i=1$  to number of vertices
2:    $\omega(v_i) = |\theta(v_i) - 2\pi|$ 
4: end for
5:  $\text{Top} \leftarrow$  Top 30% large values of  $\omega(v_i)$ 
6: for  $i=1$  to number of vertices
7:   if  $\omega(v_i) > \text{Top}$  then
8:     if  $\theta(v_i) - 2\pi > 0$  then
9:        $v_i$  selected as pre-optimized nodes
10:    else
11:       $v_i$  is left in reference nodes set
12:       $v_i$  is selected as pre-optimized nodes and moved to the negative direction of the z-axis
12:      one-ring neighbor nodes of  $v_i$  are selected as candidate nodes
13:    end if
14:  end if
14: end for
15: for  $i=1$  to number of vertices in candidate nodes set
16:   if candidate nodes are not in the reference nodes set then
17:      $v_i$  selected as pre-optimized nodes
18:   end if
19: end for
20: return pre-optimized nodes

```

nodes selection strategy, reference nodes with $\theta(v) - 2\pi < 0$ are left in the reference nodes set. If they have been moved to the positive direction of the z-axis in the iteration process, they are selected as pre-optimized nodes and may be moved to the negative direction of the z-axis with a distance that they have been moved to the positive direction of the z-axis in the previous iteration process.

4.2 Particle swarm optimization

Particle swarm optimization is a stochastic evolution method based on swarm intelligence. In PSO, an individual in the population is regarded as a particle without mass and volume, but velocity and position with dimension D in a searching space.

The position of each particle represents a feasible solution in the searching space. The particle moves at a constant speed in the searching space, and the flight process of the particle is the searching process for the feasible solution. The particle position is usually measured by a given objective function. The speed of a particle can be adjusted based on its best local and global positions compared to other particles. After multiple iterations, the best

particle position can be found as the solution of an optimization problem. The evolution mechanism of a single particle in the canonical PSO can be described as follows.

In a D -dimensional searching space, a particle population with NP particles has a vector $\mathbf{X}$ to represent position and a velocity vector $\mathbf{v}$ to represent its moving direction and speed. The i -th particle in the population in the t -th iteration can be expressed by its position and velocity, where the position vector is $X_i(t) = [x_i^1(t), x_i^2(t), \dots, x_i^D(t)]^T$ and the velocity vector is $v_i(t) = [v_i^1(t), v_i^2(t), \dots, v_i^D(t)]^T$. After t times iteration, the best position of the i -th particle (the best local position) can be found as $pbest_i(t) = p_i(t) = (p_i^1(t), p_i^2(t), \dots, p_i^D(t))$. The best global position recorded as $gbest(t)$ is calculated by Eq. (18).

$$gbest(t) = \min\{pbest_1(t), pbest_2(t), \dots, pbest_{NP}(t)\} \quad (18)$$

At the $t+1$ th step in the process of iteration, the positional parameter $x_i^j(t)$ and velocity parameter $v_i^j(t)$ of the i -th particle in the j -th dimension can be updated based on the information of the previous generation and current information as shown in Eqs. (19) and (20), respectively.

$$x_i^j(t+1) = x_i^j(t) + v_i^j(t+1) \quad (19)$$

$$v_i^j(t+1) = wv_i^j(t) + c_1r_1(pbest_i^j(t) - x_i^j(t)) + c_2r_2(gbest^j(t) - x_i^j(t)) \quad (20)$$

Algorithm 2: PSO

```

1:  $t \leftarrow 1$ 
2:  $j \leftarrow 1$ 
3:  $i \leftarrow 1$ 
4: input Initial population NP, Initial position  $\mathbf{X}$ , Initial velocity  $\mathbf{V}$ , Iteration number T, The dimension of
   searching space D
5: while  $t \leq T$  (iteration number) do
6:   while  $i \leq \text{NP}$  (initial population) do
7:     for  $j=1$  to D (The dimension of searching space)
8:        $\varpi_i^j(t) \leftarrow w\varpi_i^j(t-1) + c_1r_1(\text{pbest}_i^j(t-1) - x_i^j(t-1)) + c_2r_2(\text{gbest}^j(t-1) - x_i^j(t-1))$ 
9:        $x_i^j(t) \leftarrow x_i^j(t-1) + \varpi_i^j(t)$ 
10:    end for
11:     $i \leftarrow i+1$ 
12:  end while
13:  Update the best local position  $\text{pbest}(t) \leftarrow \{\text{pbest}_1(t), \text{pbest}_2(t), \dots, \text{pbest}_{\text{NP}}(t)\}$ 
14:  Update the best global position  $\text{gbest}(t) \leftarrow \min \{\text{pbest}_1(t), \text{pbest}_2(t), \dots, \text{pbest}_{\text{NP}}(t)\}$ 
15:   $t \leftarrow t+1$ 
16: end while
17: output  $\text{gbest}(t)$ 

```

where c_1 and c_2 are acceleration factors, r_1 and r_2 are random numbers with a uniform distribution, w is an inertia coefficient. Processing steps of PSO are as follows.

- (1) Defining population size NP and the maximum number of iterations T.
- (2) Setting an initial position of the particle: $x_i^j = x_{\min} + \text{rand}(0, 1) \times (x_{\max} - x_{\min})$ and the initial velocity of the particle: $\varpi_i^j = \varpi_{\min} + \text{rand}(0, 1) \times (\varpi_{\max} - \varpi_{\min})$.
- (3) If $t < T$, go to Step 4, else, the search stops.
- (4) Calculating the objective function $f(x_i(t))$ for each particle in population, updating $\text{pbest}_i(t)$ for each particle and $\text{gbest}(t)$.
- (5) Calculating the updated position vector and velocity vector for particles using Eqs. (19) and (20), respectively. Go to Step 3).

4.3 L-PSO

The L-PSO process is shown in Algorithm 3 to improve flattenability of a surface. The main steps of the proposed optimization process are as follows.

Initial pre-optimized nodes are selected by the local node selection strategy as inputs of the PSO method. Initial inputs also include the initial population NP, initial position $\mathbf{X}$, initial velocity $\mathbf{V}$, and initial iteration number T for the PSO process. After completing iteration number T, gbest is the best position in the current iteration process. If $f(\text{gbest})$ is smaller than $f(\text{Gbest}^{k-1})$, where Gbest^{k-1} is the best global position of the $k-1$ time of choosing pre-optimized nodes, $\text{Gbest}^k = \text{gbest}$; else, the iteration number will be increased to $T+1$. If $f(\text{Gbest}^k) - f(\text{Gbest}^{k-1}) < \epsilon$, the optimization process stops and Gbest^k is the finally improved surface, else, initial pre-optimized nodes are updated based on the current surface using the local nodes selection strategy.

Algorithm 3: L-PSO

```

1:  $k \leftarrow 1$ 
2:  $t \leftarrow 1$ 
3:  $i \leftarrow 1$ 
4: do
5:   input Pre-optimized nodes, Initial population NP, Initial position  $X$ , Initial velocity  $V$ , and Initial iteration number T
6:   do
7:     while  $t \leq T$  (Initial iteration number) do
8:       while  $i \leq NP$  (initial population) do
9:          $w_i^j(t) \leftarrow w_i^j(t-1) + c_1 r_1 (pbest_i^j(t-1) - x_i^j(t-1)) + c_2 r_2 (gbest^j(t-1) - x_i^j(t-1))$ 
10:         $x_i^j(t) \leftarrow x_i^j(t-1) + w_i^j(t)$ 
11:         $i \leftarrow i+1$ 
12:      end while
13:      Update the best local position  $pbest(t) \leftarrow \{pbest_1(t), pbest_2(t), \dots, pbest_{NP}(t)\}$ 
14:      Update the best global position  $gbest(t) \leftarrow \min\{pbest_1(t), pbest_2(t), \dots, pbest_{NP}(t)\}$ 
15:       $t \leftarrow t+1$ 
16:    end while
17:    if  $f(gbest) < f(Gbest^{k-1})$  then
18:       $Gbest^k \leftarrow gbest$ 
19:    else
20:       $T = T+1$  (increase the iteration number)
21:    end if
22:    until  $f(gbest) < f(Gbest^{k-1})$ 
23:  end do
24:  until  $f(Gbest^k) - f(Gbest^{k-1}) < \epsilon$ 
25: end do
26: output  $Gbest^k$ 

```

5 Case studies

5.1 A simple free-form surface

The free-form surface contains 89 nodes, 247 edges and 159 faces. Its initial 3D model and 3D mesh model are shown in Fig. 7a and b. Flattenability of the original surface calculated by Eq. (2) is 0.6548. The largest value of $\omega(v_p)$ among all the vertices is 0.2899. Figure 7c shows a color graph of flattenability of the surface where blue and yellow areas show the vertex having the large value of $\omega(v_p)$.

The initial population NP and initial iteration number T are set as 100 and 10,000, respectively. According to Algorithm 3, the iteration number can be increased based on the need of the solution search in the iteration process. Thus, the value of initial iteration number T will not affect the accuracy and efficiency of the optimization process. After generating the improved surface, the total iteration

number is 30,000. Data of the original surface, improved surface and an intermediate surface generated by iteration number 10,000 are listed in Table 1.

After the optimization search, the surface flattenability is improved to 5.0261. Its improved 3D mesh model is shown in Fig. 7g-i. Values of $\omega(v_p)$ for most of the triangles are near to zero. The maximum air gap is 3.6055 mm. After the optimization process, the surface flattenability is improved from 0.6548 to 5.0261. The surface smooth is decreased from 26.880 to 26.822. The area of the surface is increased from 2871.8 mm² to 2889.2 mm². The optimization process improves the flattenability significantly, but it only changes the smooth of the surface and surface area a bit. Figure 7j shows the frequency distribution histogram of air gaps in the improved surface, where most of the nodes have air gaps between 0 and 0.4 mm. Figure 7k shows the position of air gaps where red areas show vertexes with large air gaps.

Fig. 7 Original surface and improved surface of the simple free-form surface

Table 1 Results of original free-form surface and improved free-form surface

	Number of nodes	Number of edges	Number of faces	Number of iterations	Flattenability	Maximum value of $\omega(v_p)$	Smooth	Surface area (mm ²)	Air gaps (mm)
Original surface	89	247	159	0	0.6548	0.2899	26.880	2871.8	0
Intermediate surface	89	247	159	10,000	1.8422	0.2478	26.944	2942.8	3.7450
Improved surface	89	247	159	30,000	5.0261	0.1571	26.822	2889.2	3.6055

5.2 A nose model

The nose model is an irregular surface of the human body [33, 34]. It contains 548 nodes, 1575 edges and 1028 faces

in this case study. The original 3D model and triangular meshed 3D nose model are shown in Figs. 8a and b. Its flattenability is 0.1212. The largest value of $\omega(v_p)$ among all the vertices is 0.3151. Figure 8c shows the color graph

Fig. 8 Original surface and improved surface of the nose model

of $\omega(v_p)$ in each vertex using blue and yellow areas for the vertex having the large value of $\omega(v_p)$. The tip and bridge of the nose model have a large value of $\omega(v_p)$.

The initial population NP and initial iteration number T are set as 100 and 10,000, respectively. After generating the improved surface, the total iteration number is 70,000. Data of the original surface, improved surface and two intermediate surfaces generated by iteration numbers 20,000 and 40,000 are listed in Table 2. After the

optimization search, flattenability of the nose surface is improved to 0.6487 as shown in Fig. 8j, k and i. The maximum air gap is 0.3295 mm. Figure 8m shows the frequency distribution histogram of air gaps in the nose model, where most of the nodes have air gaps between 0 and 0.1 mm. Figure 8m shows the position of air gaps where red areas show the vertex with the large air gap.

Although the air gaps are smaller than the acceptable maximum value (1 mm), the surface flattenability has not

Table 2 Results of the original nose model and improved nose model

	Number of nodes	Number of edges	Number of faces	Number of iterations	Flattenability	Maximum value of $\omega(v_p)$	Smooth	Surface area mm ²	Air-gaps mm
Original surface	Whole surface	548	1575	1028	0	0.1212	15.368	2692.8	0
	Patch1	198	525	328	0	4.0481	11.013	807.41	0
	Patch2	274	704	431	0	2.6449	8.829	889.27	0
	Patch3	264	716	453	0	1.1318	8.056	996.13	0
Intermediate surface 1	Whole surface	548	1575	1028	20,000	0.2512	15.404	2694.4	0.3251
	Patch1	198	525	328	10,000	4.7159	11.085	808.57	0.6809
	Patch2	274	704	431	10,000	3.4073	11.141	889.88	0.5058
	Patch3	264	716	453	10,000	2.0001	9.326	997.27	0.4973
Intermediate surface 2	Whole surface	548	1575	1028	40,000	0.5030	15.413	2692.9	0.3299
	Patch1	198	525	328	30,000	5.0362	11.203	809.35	0.7036
	Patch2	274	704	431	30,000	4.4934	11.162	889.07	0.5749
	Patch3	264	716	453	30,000	3.3017	9.346	996.89	0.4957
Improved surface	Whole surface	548	1575	1028	70,000	0.6487	15.451	2690.6	0.3295
	Patch1	198	525	328	50,000	5.2935	11.110	808.71	0.7036
	Patch2	274	704	431	50,000	5.1577	11.150	888.50	0.5638
	Patch3	264	716	453	50,000	3.4570	9.359	996.74	0.4954

Fig. 9 Surface segmentation process

met the requirement. Based on the solution of the previous research, the surface segmentation can increase surface flattenability [35]. Thus, the segmenting surface is applied to divide the nose model into 3 patches. The cutting line is formed by finding positions with large values of $\omega(v_p)$ as shown in Fig. 9a. The segmented results are shown in Fig. 9b-d. The number of nodes, edges and faces, and flattenabilities of three parts are listed in Table 2.

An improved surface is formed using the same initial population and iteration number. The total iteration number is 50,000. Data of the original surface, improved surface and two intermediate surfaces generated by iteration number 10,000 and 30,000 are listed in Table 2. After the optimization search, flattenabilities of three patches are improved to 5.2935, 5.1577 and 3.4570, respectively. The improved surfaces are shown in Figs. 10j, 11j and 12j,

respectively. The maximum air gaps of three parts are 0.7036 mm, 0.5638 mm, and 0.4954 mm, respectively. Figures 10m, 11m and 12m show the frequency distribution histogram of air gaps in each part, respectively. Most of the nodes have air gaps between 0 and 0.1 mm in three parts. In addition, Figs. 10n, 11n and 12n show the position of air gaps where red areas show the vertices with the large air gaps. The flattenability and air gaps can finally meet the requirement of surface unfolding for bolus shaping.

The flattenability is closely related to the number of vertexes. If a surface has many vertexes, its flattenability cannot be improved to the expectation. In addition, the constraint of nodes in the original surface moved in the positive direction of normal vectors of an outer surface also restricts the further improvement of the flattenability.

Fig. 10 Original surface and improved surface of segmentation part 1 of the nose model

Fig. 11 Original surface and improved surface of segmentation part 2 of the nose model

Fig. 12 Original surface and improved surface of segmentation part 3 of the nose model

6 Conclusions and future work

A flattenability improvement method for bolus shaping was proposed in this paper to estimate air gaps. Four requirements and one constraint were identified to build the objective function for improving the surface. A local node selection strategy was proposed to search vertices having the most contribution for flattenability, which reduces dimensions of the optimization to increase the search efficiency. An original surface was optimized by an L-PSO optimization process. The maximum air gap was measured using the difference between the original mesh surface and improved surface. If the air gap and flattenability meet the requirement of accuracy, the improved surface can be unfolded directly. Otherwise, the improved surface will be segmented into several patches to further reduce the air gap and improve the flattenability. The proposed method was validated in two case studies for improvement of the surface flattenability.

Comparing with the existing methods for improving the flattenability of surfaces, the proposed method considers the area difference between an original surface and the improved surface to measure performance of bolus shaping and estimate air gaps. It adds a constraint of nodes moved in the positive direction of normal vectors of an outer surface. An L-PSO optimization process is applied to search the best solution. Results of case studies show that the proposed method has a good performance in improving surface flattenability and calculating air gaps. Contributions of this paper are as follows.

- (1) An objective function is proposed by considering four objectives and one constraint including the minimum deformation and distortion, maximum smoothness, maintaining the original surface area, minimum air gaps, and constraint of nodes moved in the positive direction of normal vectors of an outer surface.
- (2) A local node selection strategy is designed to select nodes in a 3D mesh surface with the most contribution to improve flattenability.
- (3) A local node selection strategy based PSO (L-PSO) method is proposed to reduce dimensions of the optimization search.

Using PSO in the surface improvement still has some limitations such as the reduced convergence rate when results close to the objective. The efficiency of the optimization process needs improvement. PSO will be combined with other methods to further increase efficiency of the optimization search in future research. In addition,

the proposed bolus shaping method will be tested using commercial materials and applied in cancer treatment.

Acknowledgements The authors wish to acknowledge that this research has been supported by the Discovery Grants from the Natural Sciences and Engineering Research Council (NSERC) of Canada, and the Graduate Enhancement of Tri-Council Stipends (GETS) program from the University of Manitoba.

Compliance with ethical standards

Conflicts of interest The authors declare that they have no conflict of interest.

References

1. Vyas V, Palmer L, Mudge R, Jiang R, Fleck A, Schaly B et al (2013) On bolus for megavoltage photon and electron radiation therapy. *Med Dosim* 38(3):268–273. <https://doi.org/10.1016/j.meddos.2013.02.007>
2. Kong M, Holloway L (2007) An investigation of central axis depth dose distribution perturbation due to an air gap between patient and bolus for electron beams. *Aust Phys Eng Sci Med* 30(2):111. <https://doi.org/10.1007/BF03178415>
3. Liu YJ, Zhang DL, Yuen MMF (2010) A survey on CAD methods in 3D garment design. *Comput Ind* 61(6):576–593. <https://doi.org/10.1016/j.compind.2010.03.007>
4. Huang HQ, Mok PY, Kwok YL, Au JS (2012) Block pattern generation: from parameterizing human bodies to fit feature-aligned and flattenable 3D garments. *Comput Ind* 63(7):680–691. <https://doi.org/10.1016/j.compind.2012.04.001>
5. Wang CC (2008) Towards flattenable mesh surfaces. *CAD Comput Aided Des* 40(1):109–122. <https://doi.org/10.1016/j.cad.2007.06.001>
6. Do Carmo MP (2016) *Differential geometry of curves and surfaces: revised and updated*, 2nd edn. Courier Dover Publications, USA
7. Liu Y, Pottmann H, Wallner J, Yang YL, Wang W (2006, July) Geometric modeling with conical meshes and developable surfaces. In *ACM transactions on graphics (TOG)*. ACM 25(3):681–689. <https://doi.org/10.1145/1141911.1141941>
8. Liu YJ, Tang K, Gong WY, Wu TR (2011) Industrial design using interpolatory discrete developable surfaces. *CAD Comput Aided Des* 43(9):1089–1098. <https://doi.org/10.1016/j.cad.2011.06.001>
9. Zeng L, Liu YJ, Chen M, Yuen MMF (2012) Least squares quasi-developable mesh approximation. *Comput Aided Geom Des* 29(7):565–578. <https://doi.org/10.1016/j.cagd.2012.03.009>
10. Lee KW, Bo P (2016) Feature curve extraction from point clouds via developable strip intersection. *J Comput Des Eng* 3(2):102–111. <https://doi.org/10.1016/j.jcde.2015.07.001>
11. Chu CH, Wang CC, Tsai CR (2008) Computer aided geometric design of strip using developable Bézier patches. *Comput Ind* 59(6):601–611. <https://doi.org/10.1016/j.compind.2008.03.001>
12. Fernández-Jambrina L (2007) B-spline control nets for developable surfaces. *Comput Aided Geom Des* 24(4):189–199. <https://doi.org/10.1016/j.cagd.2007.03.001>
13. Fernández-Jambrina L (2017) Bézier developable surfaces. *Comput Aided Geom Des* 55:15–28. <https://doi.org/10.1016/j.cagd.2017.02.001>
14. Meyer M, Desbrun M, Schröder P, Barr AH (2003) Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and mathematics III*. Springer, Berlin, Heidelberg, pp 35–57. https://doi.org/10.1007/978-3-662-05105-4_2

15. Wang CC, Tang K (2004) Achieving developability of a polygonal surface by minimum deformation: a study of global and local optimization approaches. *Vis Comput* 20(8–9):521–539. <https://doi.org/10.1007/s00371-004-0256-0>
16. Zhang Y, Wang CC, Ramani K (2016) Optimal fitting of strain-controlled flattenable mesh surfaces. *Int J Adv Manuf Technol* 87(9–12):2873–2887. <https://doi.org/10.1007/s00170-016-8669-2>
17. Schraudolph NN, Yu J, Günter S (2007, March) A stochastic quasi-Newton method for online convex optimization. In *Artificial intelligence and statistics*, pp 436–443.
18. Zheng W, Bo P, Liu Y, Wang W (2012) Fast B-spline curve fitting by L-BFGS. *Comput Aided Geom Des* 29(7):448–462. <https://doi.org/10.1016/j.cagd.2012.03.004>
19. Kennedy J, Eberhart R (1995) Particle swarm optimization (PSO). In *Proceedings of IEEE international conference on neural networks*, Perth, Australia, pp 1942–1948.
20. Marini F, Walczak B (2015) Particle swarm optimization (PSO). A tutorial. *Chemometr Intell Lab Syst* 149:153–165. <https://doi.org/10.1016/j.chemolab.2015.08.020>
21. Yang S, Wang M (2004) A quantum particle swarm optimization. In: *Proceedings of the 2004 congress on evolutionary computation* (IEEE Cat. No. 04TH8753). IEEE, vol. 1, pp 320–324. <https://doi.org/10.1109/CEC.2004.1330874>
22. Sun X, Li S, Dun X, Li D, Li T, Guo R, Yang M (2019) A novel characterization method of piezoelectric composite material based on particle swarm optimization algorithm. *Appl Math Model* 66:322–331. <https://doi.org/10.1016/j.apm.2018.08.015>
23. Shahbeig S, Helfroush MS, Rahideh A (2017) A fuzzy multi-objective hybrid TLBO–PSO approach to select the associated genes with breast cancer. *Signal Processing* 131:58–65. <https://doi.org/10.1016/j.sigpro.2016.07.035>
24. Tang D, Dai M, Salido MA, Giret A (2016) Energy-efficient dynamic scheduling for a flexible flow shop using an improved particle swarm optimization. *Comput Ind* 81:82–95. <https://doi.org/10.1016/j.compind.2015.10.001>
25. Zhang J, Xia P (2017) An improved PSO algorithm for parameter identification of nonlinear dynamic hysteretic models. *J Sound Vib* 389:153–167. <https://doi.org/10.1016/j.jsv.2016.11.006>
26. Liang JJ, Suganthan PN (2005) Dynamic multi-swarm particle swarm optimizer. In: *Proceedings 2005 IEEE swarm intelligence symposium*, 2005. SIS 2005. IEEE, pp. 124–129. <https://doi.org/10.1109/CEC.2005.1554727>
27. Fan SKS, Zahara E (2007) A hybrid simplex search and particle swarm optimization for unconstrained optimization. *Eur J Oper Res* 181(2):527–548. <https://doi.org/10.1016/j.ejor.2006.06.034>
28. dos Santos CL, Mariani VC (2009) A novel chaotic particle swarm optimization approach using Hénon map and implicit filtering local search for economic load dispatch. *Chaos Solitons Fractals* 39(2):510–518. <https://doi.org/10.1016/j.chaos.2007.01.093>
29. Li S, Tan M, Tsang IW, Kwok JTY (2011) A hybrid PSO-BFGS strategy for global optimization of multimodal functions. *IEEE Trans Cybern* 41(4):1003–1014. <https://doi.org/10.1109/TSMCB.2010.2103055>
30. Lou I, Xie Z, Ung WK, Mok KM (2017) Integrating support vector regression with particle swarm optimization for numerical modeling for algal blooms of freshwater. In: *Advances in monitoring and modelling algal blooms in freshwater reservoirs*. Springer, Dordrecht, pp. 125–141. https://doi.org/10.1007/978-94-024-0933-8_8
31. Desbrun M, Meyer M, Schröder P, Barr AH (1999) Implicit fairing of irregular meshes using diffusion and curvature flow. In: *Proceedings of the 26th annual conference on computer graphics and interactive techniques*, pp 317–324. ACM Press/Addison-Wesley Publishing Co, New York/Boston.
32. Wang CC, Smith SS, Yuen MM (2002) Surface flattening based on energy model. *CAD Comput Aided Des* 34(11):823–833. [https://doi.org/10.1016/S0010-4485\(01\)00150-6](https://doi.org/10.1016/S0010-4485(01)00150-6)
33. Burleson S, Baker J, Hsia AT, Xu Z (2015) Use of 3D printers to create a patient-specific 3D bolus for external beam therapy. *J Appl Clin Med Phys* 16(3):166–178. <https://doi.org/10.1120/jacmp.v16i3.5247>
34. Park JW, Oh SA, Yea JW, Kang MK (2017) Fabrication of malleable three-dimensional-printed customized bolus using three-dimensional scanner. *PLoS ONE* 12(5):e0177562. <https://doi.org/10.1371/journal.pone.0177562>
35. Li R, Peng Q, Ingleby H, Sasaki D (2018) Methods and analysis of 3D shape unfolding and folding for radiotherapy. In: *ASME 2018 international design engineering technical conferences and computers and information in engineering conference*. American Society of Mechanical Engineers Digital Collection. <https://doi.org/10.1115/DETC2018-85464>

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