International Journal of Precision Engineering and Manufacturing (2022) 23:921–927
https://doi.org/10.1007/s12541-022-00683-6

Online ISSN 2005-4602
Print ISSN 2234-7593

REGULAR PAPER

Free-Form Surface Flattening Based on Rigid Registration and Energy Optimization

Peng Wei[1,2] · Qiao Kai[1] · Bao Yidong[3] · Zhang Chaoyang[1,2] · Ji Weixi[1,2]

Received: 2 November 2021 / Revised: 22 February 2022 / Accepted: 7 June 2022 / Published online: 21 June 2022

Abstract

Fundamental technology of Computer Aided Design, free-form surface flattening is important for both practical and scientific point of view in mechanical engineering. This paper proposed a flattening algorithm by using a local rigid registration and a global energy optimization. Firstly, each 3D element is aligned to the plane by minimizing the distance between the original 3D element and its corresponding planar element. Then, a global optimization operator is used to stitch and optimize these best-aligned local elements by iteratively minimizing a quadratic energy function composed of linear elastic energy, which makes the internal force of the nodes reach the equilibrium state. The experimental results show that this method is stable and reliable, and can obtain good surface flattening effect under free boundary conditions.

Keywords Surface flattening · Rigid transform · Linear elastic energy · Triangular mesh · Quadrilateral mesh

1 Introduction

Flattening of free-form surfaces has been widely used in the field of mechanical engineering, such as blank estimation in sheet metal forming, computer-aided design for mechanical product and surface reconstruction in reverse engineering. It involves computing a mapping between surface in three-dimensional space and a planar parameter domain. Because of its simplicity and flexibility, surface mesh composed of triangular elements, quadrilateral elements and their combination has become the main expression of three-dimensional models. Both triangular and quadrilateral surface meshes are considered in this surface flattening algorithm.

In recent years, a variety of methods on surface flattening has been developed [1], and the related methods can be divided into linear solution-based method [2–6] and iterative solution-based method by their solving strategies. The most common used method based on linear solution is shape-preserving method [2, 3], which determines the position of each vertex in the flattened mesh by solving a linear system based on convex combination. The main disadvantage of this method is that it requires predefined and convex two-dimensional boundaries. Levy et al. [5] presented a Least- Squares Conformal Mapping (LSCM) method by a least-squares approximation of the discrete Cauchy-Riemann equations to minimize angle deformation. Yavuz [6] uses the dynamic virtual boundary method to reduce the deformation of triangles near the boundary caused by convex combination. While linear solution-based method is easy to use and has high efficiency, it may produce flattening results with local or global overlaps. Iterative solution-based method methods perform the surface flattening by iteratively solving the minimum energy or equilibrium state, which are defined by different mesh properties. Sheffer et al [7] proposed a method based on angle optimization to calculate the parameterization of mesh surface. This method aims to optimize the angular deformation of triangular mesh by solving a non-linear system, which usually requires many calculations. The mass-spring model is often used to construct energy function for surface flattening [8–11]. Zhong et al. [8] divided the three-dimensional surface into several almost developable patches, and then flatten them by solving the energy function via mass-spring model. With the similar mass-spring model based stretching energy, Bing et al. [9] project the

✉ Peng Wei
weipeng@jiangnan.edu.cn

1 School of Mechanical Engineering, Jiangnan University, Wuxi 214122, China

2 Jiangsu Key Laboratory of Advanced Food Manufacturing Equipment and Technology, Wuxi 214122, China

3 School of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

element locally to planar, and then combines the results of a single projection by solving a simplified global matrix. Wang et al. [11] also use a mass-spring model-based energy function to flatten 3D mesh surfaces into 2D patterns, which can obtain result with flexible surface boundaries. A local to global operation [12, 13] is proposed to ensure the minimum distortion between input meshes and flattened meshes by iteratively solving a global energy function, which is superposition of local energy. One-step Inverse Forming Theory in sheet metal forming is introduced to solve the surface flattening problem [14, 15]. However, the one-step inverse forming theory is complex, which leads to the complex parameters and time-consuming of the mesh surface flattening algorithm. Zhuang et al. [16] defined the energy model based on the change of edge length by using Young's modulus. By minimizing this energy model, the flattening results for computer aided garment design are obtained. Zhang et al. [17] proposed a strain constraint method to flatten the triangular surface mesh by morphing the original element to an approximate equidistant triangular mesh, which is a good approximation of the input surface model. Bouaziz et al. [18] adopt shape proximity function and shape projection operator to optimize the geometry processing, which can be used for shape preserving deformation and conformal parameterization of geometric models. While iterative solution-based methods yield better results with natural flattened boundaries, they have high computational cost and will lead to non-convergence results in some cases.

Most of the surface flattening algorithms are only for triangular meshes, in order to obtain the flattening results for surfaces composed of triangular elements or quadrilateral elements, a free-form surface flattening method is proposed based on local rigid registration and global energy optimization, mainly consist of two steps: (1) Local rigid registration for single element: each surface element is best align to its correspondence planar element by minimizing their distance; (2) Global stitching operation for all elements: linear elastic finite element energy is used to stitching the transformed element to ensure the validity of the mesh connectivity. The proposed method is mainly based on rigid registration and elastic energy optimization, can be abbreviated as RR/EO method. The rest of this paper is organizing as follows. Section 2 gives the detailed steps of surface flattening, including local alignment of single element and global "stitch" operation. Several experimental results of surface flattening are given in Sect. 3, and the conclusions are finally discussed in Sect. 4.

2 Free-Form Surface Flattening

The purpose of surface flattening is to minimize the parametric deformation between the original surface and the flattened surface as much as possible. To this end, a local to global surface flattening strategy is adopted in this paper. Each surface element (Triangular or Quadrilateral) is transformed to its corresponding planar parametric element by finding the best rotation and translation transform. In order to stich the transformed elements together, the linear elastic finite element energy is used to find an equilibrium state of internal force. The steps of the proposed RR/EO flattening algorithm are summarized in Algorithm 1.

Algorithm 1. Surface flattening

Input: A surface mesh M with vertices V, elements E.

Obtain an initial planar mesh P_0 with vertices V_0, elements E_0 by linear flattening method.

t = 0, **repeat**

Step1: Local transformation of surface element

  for every element E in mesh M **do**

    Align the element E to its corresponding element E_t by minimizing the distance between them, obtaining a planar transformed element E''.

  end for

Step2: Global optimization of surface mesh

  for every element E'' **do**

    Calculate the internal force of element E_t by using the displacement between element E_t and E''.

    The internal force of node is obtained by superimposing the internal force of its adjacent elements at the node.

  end for

  Establish global energy by using the linear elastic energy of mesh and the internal force of node.

  Calculate the node displacement by minimizing the global energy function.

  Update P_t with calculated displacements $\boldsymbol{q}_t$ to give P_{t+1}:

  $P_{t+1} = P_t + \boldsymbol{q}_t$

$t = t+1$

until $\|\boldsymbol{q}_t\| < \sigma$ or $t > t_{\max}$

Output: The flattened surface mesh P_t

2.1 Local Alignment of Single Element

Most surfaces are composed of triangular elements, quadrilateral elements and their combination, while most surface flattening algorithms are only for surfaces composed of triangular meshes. This leads to the need for additional split operation for quadrilateral elements, and different element splitting strategies will affect the flattening results. Therefore, the proposed method in this paper needs to be applied to triangular and quadrilateral elements. At the same time, the given element needs to be transformed to make it as close as possible to its corresponding planar parameterization element.

Given a surface element $E = (v_1, \dots, v_{ne})$, $v_i = (x_i, y_i, z_i) \in \mathrm{R}^3$ is the nodes of the given element, ne is the number of node in element E, and its corresponding planar element is $E_t = (p_1, \dots, p_{ne})$, $p_i = (x_i, y_i) \in \mathrm{R}^2$. The goal is to find a transformation that best superposes the given element E with elements E_t. The first step is to transform the three-dimensional element E to the OXY plane and obtain a planar element $E' = (v'_1, \dots, v'_{ne})$. Then, the planar element E' is aligned to its corresponding planar element E_t by minimizing the Euclidean distance between planar element E' and E_t. The mean square objective function to be minimized is:

$$f(\boldsymbol{R}, \boldsymbol{T}) = {}^1/_{ne} \sum_{i=1}^{ne} \left| p_i - \boldsymbol{R} v'_i - \boldsymbol{T} \right|^2 \tag{1}$$

where $\boldsymbol{R}$ is the 2×2 rotation matrix and $\boldsymbol{T}$ is the 2×1 translation matrix. This minimizing problem can be solved by many non-iterative optimization methods [19, 20], such as singular value decomposition (SVD) method, quaternion method, etc. Here, the singular value decomposition method is used to get the rigid transformation matrix $\boldsymbol{R}$ and $\boldsymbol{T}$. The solution of Eq. (1) can be converted into the SVD decomposition of the following matrix:

$$\boldsymbol{H} = \sum_{i=1}^{ne} (v'_i - \overline{v})(p_i - \overline{p})^{\mathrm{T}} \tag{2}$$

where $\overline{v}$ and $\overline{p}$ are the center of element E' and E_t respectively, calculated by: $\overline{v} = 1/ne \sum_{i=1}^{ne} v'_i$, $\overline{p} = 1/ne \sum_{i=1}^{ne} p_i$. In particular, using SVD, $\boldsymbol{H}$ can be written as

$$\boldsymbol{H} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathrm{T}} \tag{3}$$

The best fit rotation matrix $\boldsymbol{R}$ and translation matrix $\boldsymbol{T}$ can be obtained by

$$\begin{cases} \boldsymbol{R} = \boldsymbol{V}\boldsymbol{U}^{\mathrm{T}} \\ \boldsymbol{T} = \overline{p} - \boldsymbol{R}\overline{v} \end{cases} \tag{4}$$

Figure 1 shows the result of local alignment of triangle element. The original triangle $v_1v_2v_3$ and its corresponding triangle $p_1p_2p_3$ are plotted in Fig. 1a, b marked by blue solid lines, while the transformed triangle $v''_1\, v''_2\, v''_3$ obtained by alignment operation is shown in Fig. 1b with red dotted lines. Figure 2 is a local alignment example of quadrilateral element. In this example, the original quadrilateral $v_1v_2v_3v_4$ is transformed to $v''_1\, v''_2\, v''_3\, v''_4$ marked by red dotted lines, shown in Fig. 2b.

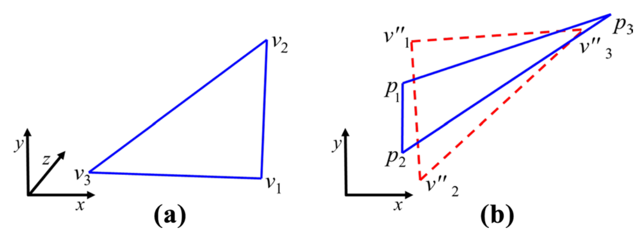

Fig. 1 Local alignment of triangular element: **a** Original Triangle; **b** Best fit result

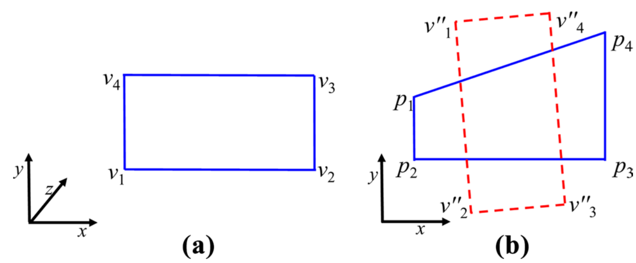

Fig. 2 Local alignment of quadrilateral element: **a** Original quadrilateral; **b** Best fit result

2.2 Global Stitching of Planar Mesh

Due to the shape difference between the 3D element and its corresponding 2D element, the locally transformed 2D mesh is disconnected. Then, a global optimization is used to "stitching" the transformed elements together to a 2D mesh, which is derived by minimizing a quadric energy function. The energy function consists of two terms, including the energy caused by nodal loads and the linear elastic strain energy. The global stitching can be seen as an elastic deformation process from the planar element E_t to its corresponding transformed element E'', and the internal force of the nodes will inevitably be generated during the deformation process.

Suppose a 2D surface element $E_t = (p_1, \dots, p_{ne})$, $p_i = (x_i, y_i) \in \mathrm{R}^2$ and its corresponding transformed surface element $E'' = (v''_1, \dots, v''_{ne})$, $v''_i = (x''_i, y''_i) \in \mathrm{R}^2$. The displacement vectors of the surface mesh in the global coordinate system are $\boldsymbol{q}^{\mathrm{e}} = \{\Delta x_1, \Delta y_1, \dots, \Delta x_{ne}, \Delta y_{ne}\}$. The elements of displacement vector $\boldsymbol{q}^{\mathrm{e}}$ can be calculated as follows: $\Delta x_i = x''_i - x_i$, $\Delta y_i = y''_i - y_i, i = 1, \dots, ne$. Considering that the transformed element E'' is obtained by the linear elastic deformation of its corresponding planar surface element E_t, the stress of the element is

$$\sigma = \boldsymbol{D}\boldsymbol{B}\boldsymbol{q}^{\mathrm{e}^{\mathrm{T}}} \tag{5}$$

where $\boldsymbol{B}$ is strain matrix and $\boldsymbol{D}$ is linear elastic matrix. Therefore, the internal force of the nodes in the global coordinate system of the surface mesh is defined as follows

$$F_{\text{in}}^{\text{e}} = \int_{\text{e}} B^{\text{T}} \sigma d\text{v} = k^{\text{e}} q^{\text{eT}} \tag{6}$$

The matrix k^e is the element stiffness matrix of the quadrilateral element and triangular element which can be calculate by the three-node triangular element and planar four-node isoparametric element [21]. After calculating the node internal forces of each element by formula (6), suppose that $L(i) = \{l_1, \ldots, l_{m(i)}\}$ denotes $m(i)$ surface elements adjacent to the node v_i. The internal force of node v_i is obtained by superposition of its adjacent elements

$$F_{\text{in}}^{i} = \sum_{i \in L(i)} F_{in}^{\text{e}L(i)} \tag{7}$$

$F_{\text{in}}^{eL(i)}$ represents the internal force of element e$L(i)$ at node v_i. The internal force F_{in} of the global surface mesh is obtained by the integration of F_{in}^{i} of each node.

According to the linear elastic finite element theory, the linear elastic strain energy is defined as follows:

$$E_{\text{B}}(q) = qKq^{\text{T}} \tag{8}$$

The energy caused by the external load of the node is:

$$E_{\text{F}}(q) = qF_{\text{in}} \tag{9}$$

Among them, K is a $2\text{n} \times 2\text{n}$ global stiffness matrix composed of element stiffness matrix k^e; $q = \{\Delta x_1, \Delta y_1, \ldots, \Delta x_{\text{n}}, \Delta y_n\}$ is a $1 \times 2\text{n}$ vector of the node displacements, and n is the number of nodes in the surface mesh. The displacements of all nodes are obtained by minimizing the following sum of all energy terms:

$$E(q) = {}^1\!/_2 qKq^{\text{T}} + \theta q F_{\text{in}} \tag{10}$$

where θ is the weight of nodal load, generally set it to 0.6–08. The above energy function is quadratic, which can be solved by a sparse linear system:

$$Kq^{\text{T}} = \theta F_{\text{in}} \tag{11}$$

In each iteration of the proposed surface flattening algorithm, the position of all nodes is updated as: $P_{t+1} = P_t + q_t$, and new node positions are used as the initial planar mesh of the next iteration. In order to make the iteration converge, the terminal condition for the iterative procedure is determined as:

$$\|q_t\| = {}^1\!/_{2n} \sum_{i=1}^{2n} (q_t^i)^2 \leq \sigma \quad \text{or} \quad t \geq t_{\max}$$

where σ is a given precision and $t_{\max}$ is a given maximal number of iterations.

The surface flattening method proposed in this paper requires an initial parameterization to start the iterative solution. The basic requirement of initial parameterization is that it has a valid mesh connectivity without too much parameterization distortion, and be fast to generate. Therefore, the shape-preserving surface parameterization method proposed by Floater [2] can be used to quickly obtain the plane mesh with fixed convex boundary. An appropriate pre-defined outer boundary can be obtained mainly by the following steps: (1) Calculate the average normal of the surface, and then project the boundary of the surface onto the plane according to the normal direction; (2) Remove unnecessary points in the boundary by the Douglas-Peucker method; (3) Remove all concave points in the boundary iteratively until the boundary is convex.

3 Experimental Results

The proposed surface flattening algorithm have applied to flatten a variety of mechanical models, running on a machine with Core i7 2.6 GHz CPU and 8 GB memory, and here are some of them. Figure 3 shows the main steps of RR/EO surface flattening method. Figure 3a is the original surface mesh. With the automatically created outer boundary, the initial parameterization mesh shown in Fig. 3b is obtained by using Floater's shape-preserving method [2]. The transformed elements with local alignment operation are shown in Fig. 3c. Figure 3d is the final surface flattening result after three iterations. From the example shown in Fig. 3, it can be seen that the initial flattened mesh by shape-preserving method meets the requirements of the proposed iterative surface flattening algorithm, so the shape-preserving method is used to generate the initial solution in the follow-up.

To demonstrate the effectiveness of the proposed method, the RR/EO method is compared with Local/Global method proposed in [12]. Figure 4 shows the flattening results of a sheet metal part. The original surface mesh of a sheet metal part is shown in Fig. 4a and b is the initial parameterization mesh. The flattened results using RR/EO method and Local/Global method are shown in Fig. 4c and d.

Figure 5 is a surface flattening example of rock arm. It is necessary to fill the internal holes of rock arm surface when using shape-preserving surface parameterization method. Figure 5a displays the original rock arm model with 9312 triangular elements and 4952 nodes. Figure 5b shows the flattened result using RR/EO method after four iterations, while the flattened result by Local/Global method is plotted in Fig. 5c. As can be seen from Fig. 5b, the mesh distortion between the flattening result and the original model is small.

Different with most other surface flattening algorithms, the proposed method can directly flatten surfaces consisting of triangular and quadrilateral meshes. In sheet metal forming simulation, it is usually necessary to flatten the model to predict the size of blank. Then, two flattening examples of automobile body parts composed of triangular and quadrilateral

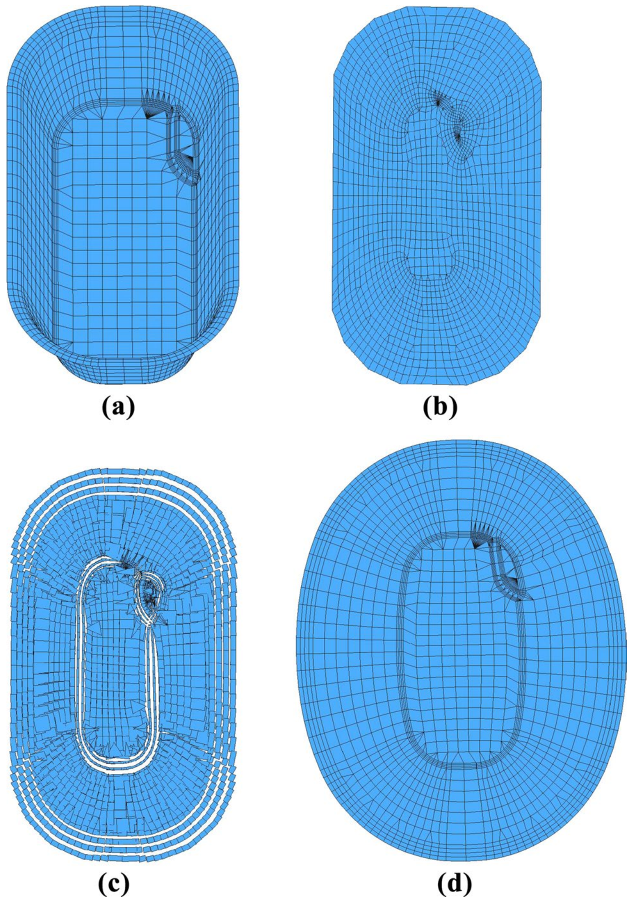

Fig. 3 Surface flattening of an oil drain: **a** Original surface; **b** Initial flattened mesh; **c** Local alignment; **d** Flattened result

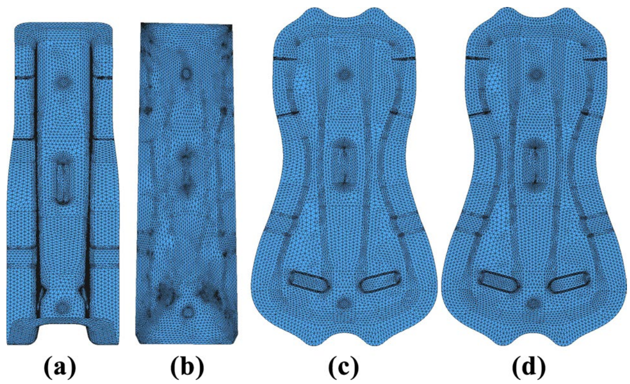

Fig. 4 Surface flattening of a sheet metal part: **a** Original surface; **b** Initial flattened mesh; **c** RR/EO method; **d** Local/Global method

meshes are given in the following. Figure 6a–c are the original surface, initial guess mesh and flattened result of auto-body panel respectively. Figure 7a shows the original surface of a fender model, and the flattened meshes generated by RR/EO method is displayed in Fig. 7b. Since the surfaces are hybrid

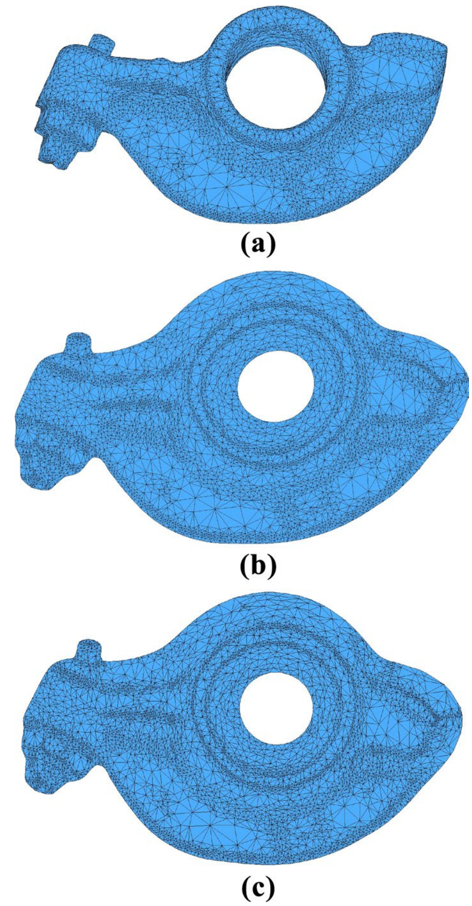

Fig. 5 Surface flattening of a rock arm: **a** Original surface; **b** RR/EO method; **c** Local/Global method

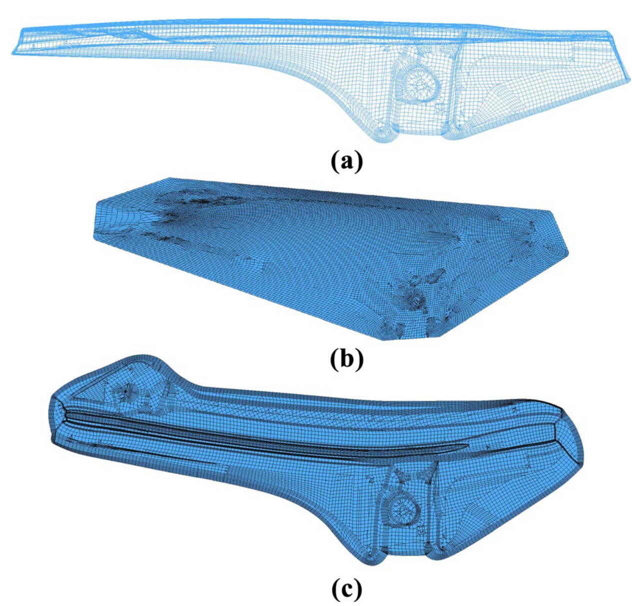

Fig. 6 Surface flattening of an auto-body panel: **a** Original surface; **b** Initial flattened mesh; **c** Flattened result

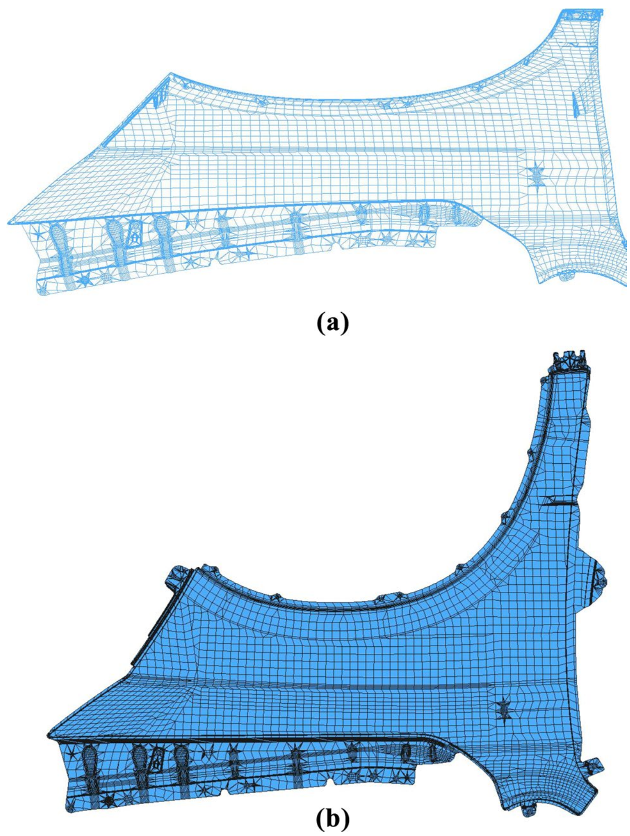

Fig. 7 Surface flattening of a fender: **a** Original surface; **b** Flattened result

meshes consisting of triangles and quadrilaterals, it is necessary to subdivide a single quadrilateral into two triangles for operating by Local/Global method.

In order to measure the flattening distortion, the angle and area distortions are calculated as follows:

$$D^{angle} = {}^{1}\!/_{3n\mathrm{Elem}} \sum_{k=1}^{n\mathrm{Elem}} \sum_{i=1}^{ne} {}^{1}\!/_{(\beta_i^k)^2} (\alpha_i^k - \beta_i^k)^2 \tag{12}$$

$$D^{area} = \sum_{k=1}^{n\mathrm{Elem}} \omega_k {}^{1}\!/_{A(E_k)^2} (A(E_k'') - A(E_k))^2 \tag{13}$$

where α_i^k and β_i^k denote the angle of elements of flattened surface and original surface respectively; nElem is the number of elements; $A(E_k'')$ is the area of element in flattened surface and the area of its corresponding element in original surface is $A(E_k)$; ω is a weight defined by $\omega_k = A(E_k) / \sum A(E_k)$. Table 1 shows the statistics of flattening distortion and computational time between the proposed method and Local/Global method proposed in [12]. As can be seen from Table 1, the proposed RR/EO method can achieve better results with lower area distortion and is relatively stable for irregular meshes because of its physical background.

4 Conclusions

Surface flattening has been widely used in sheet metal forming simulation and computer aided design. In this paper, a surface flattening algorithm based on local rigid registration and global elastic energy optimization is proposed. The method uses a local-to-global strategy to transform a single element to a parametric space, and the final flattened surface is obtained by using a linear elastic energy optimization model to stitch the elements after local transformation. Experiments show that this method is suitable for surface flattening with arbitrary shape, and the flattening results have the advantages of free boundary and less distortion. Unlike other methods based on geometric feature optimization, this method has definite physical meaning and achieves a good balance in solving time and flattening results. At the same time, unlike most methods only for triangular meshes, the proposed method is suitable for surfaces composed of triangular and quadrilateral meshes. For models with large number nodes, the time consumption of the proposed RR/EO algorithm is not fast enough. In the future, it will be considered to flatten the surface in patches, which can improve the efficiency and speed of flattening.

Table 1 Comparisons of flattening distortion and running time

Model	No. of nodes	No. of elements	Local/Global			RR/EO method		
			Angle	Area	Run time (s)	Angle	Area	Run time (s)
Figure 3	1985	2010	0.0032	3.5671	1.52	0.0022	1.9303	1.04
Figure 4	7567	14,802	0.0054	1.6564	19.23	0.0067	1.2263	15.34
Figure 5	4952	9312	0.0690	1.9835	5.87	0.0586	1.4787	3.55
Figure 6	18,571	19,511	0.0061	8.0141	18.36	0.0011	5.5903	25.94
Figure 7	12,247	11,177	0.0254	4.7733	26.28	0.0360	2.7754	20.35

Acknowledgements This work was funded by the Project of the National Natural Science Foundation of China (No.11402264) and Open Fund of Jiangsu Key Laboratory of Precision and Micro Manufacturing Technology. These supports are gratefully acknowledged.

References

1. Sheffer, A., Praun, E., & Rose, K. (2006). Mesh parameterization methods and their applications. *Foundation and Trends in Computer Graphics and Vision, 2*(2), 105–171.
2. Floater, M. S. (1997). Parameterization and smooth approximation of surface triangulations. *Computer Aided Geometry Design, 14*, 231–250.
3. Floater, M. S. (2003). Mean value coordinates. *Computer Aided Geometric Design, 20*(1), 19–27.
4. Sorkine, O., Cohen-Or, D., Goldenthal, R., et al. (2002). Bounded-distortion piecewise mesh parameterization. In *IEEE Visualization* (pp. 355–362)
5. Lévy, B., Petitjean, S., Ray, N., et al. (2002). Least squares conformal maps for automatic texture atlas generation. *ACM Transactions on Graphics, 21*(3), 362–371.
6. Yavuz, E., Yazici, R., Kasapbasi, M. C., et al. (2019). Improving initial flattening of convex-shaped free-form mesh surface patches using a dynamic virtual boundary. *Computer Systems Science and Engineering, 34*(6), 339–355.
7. Sheffer, A., Lévy, B., Mogilnitsky, M., et al. (2005). ABF++: Fast and robust angle based flattening. *ACM Transactions on Graphics, 24*(2), 311–330.
8. Zhong, Y., & Xu, B. (2006). A physically based method for triangulated surface flattening. *Computer Aided Design, 38*(10), 1062–1073.
9. Bing, Y., Yue, Y., Ran, Z., et al. (2018). Triangulated surface flattening based on the physical shell model. *Journal of Mechanical Science & Technology, 32*(5), 2163–2171.
10. Liu, Q., Xi, J., & Wu, Z. (2013). An energy-based surface flattening method for flat pattern development of sheet metal components. *International Journal of Advanced Manufacturing Technology, 68*(5), 1155–1166.
11. Wang, C., Smith, S. F., & Yuen, M. (2002). Surface flattening based on energy model. *Computer-Aided Design, 34*(11), 823–833.
12. Liu, L. G., Zhang, L., Xu, Y., et al. (2008). A local/global approach to mesh parameterization. *Computer Graph Forum, 27*(5), 1495–1504.
13. Wang, Z., Luo, Z. X., Zhang, J. L., et al. (2016). ARAP++: An extension of the local/global approach to mesh parameterization. *Frontiers of Information Technology & Electronic Engineering, 17*(6), 501–515.
14. Li, B., Zhang, X., Zhou, P., et al. (2010). Mesh parameterization based on one-step inverse forming. *Computer Aided Design, 42*(7), 633–640.
15. Zhu, X. F., Hu, P., & Ma, Z. D. (2013). A new surface parameterization method based on one-step inverse forming for isogeometric analysis-suited geometry. *International Journal of Advanced Manufacturing Technology, 65*(9), 1215–1227.
16. Zhuang, M., & Zhang, X. (2013). A novel apparel surface flattening algorithm. *International Journal of Clothing Science and Technology, 25*(4), 300–316.
17. Zhang, Y., Wang, C. C. L., & Ramani, K. (2016). Optimal fitting of strain-controlled flattenable mesh surfaces. *International Journal of Advanced Manufacturing Technology, 87*, 9–12.
18. Bouaziz, S., Deuss, M., Schwartzburg, Y., et al. (2012). Shape-Up: Shaping Discrete Geometry with Projections. *Computer Graphics Forum, 31*(5), 1657–1667.
19. Besl, P. J., & Mckay, N. D. (1992). A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 14*(2), 239–256.
20. He, Y., Liang, B., Yang, J., et al. (2017). An iterative closest points algorithm for registration of 3D laser scanner point clouds with geometric features. *Sensors, 17*(8), 1862–1869.
21. Dhatt, G., Lefrançois, E., & Touzot, G. (2012). *Finite element method* (1st ed.). John Wiley & Sons.