

Research Article

Research on the Flattening Method of an Energy-Based Fabric Deformation Model in Garment Design

Yuxiang Zhu ^{1,2}, Gangqiang Li,^{1,2} Yanjun Peng,³ Mingmin Zhang,⁴ Yingran Ma,³ and Yuanhong Wang³

¹College of Information Engineering, Huanghuai University, Zhumadian, Henan 463000, China

²Henan Provincial Key Laboratory of Smart Lighting, Huanghuai University, Zhumadian, Henan 463000, China

³College of Computer Science and Technology, Shandong University of Science and Technology, Qingdao, Shandong 266590, China

⁴College of Computer Science and Engineering, Zhejiang University, Hangzhou, Zhejiang 310007, China

Correspondence should be addressed to Yuxiang Zhu; zhuyuxiangcn@163.com

Received 30 October 2021; Revised 28 December 2021; Accepted 1 April 2022; Published 16 April 2022

Academic Editor: Feng Li

Copyright © 2022 Yuxiang Zhu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Surface flattening plays an important role in the whole process of garment design. We proposed a novel method by using three-dimensional triangle mesh flattening in this study. First, the three-dimensional triangle mesh is flattened to a two-dimensional plane to approximate the original surface. The initial flattening results are then used as preliminary guesses for subsequent optimizations. Considering that the deformation energy in the real woven fabric is related to tensile or shear deformation, a simplified fabric deformation model based on energy is proposed to update the energy distribution to determine the best two-dimensional pattern. An innovative unified axis system process is proposed to obtain the deformation energy, and energy relaxation in local flattening is proposed to release the distortion of flattening. Finally, the experimental results show that complex surfaces such as garments could achieve better flattening results. Compared with other energy-based methods in garment design, our proposed methods are more flexible and practical.

1. Introduction

Three-dimensional computer-aided design (CAD) tools have developed rapidly in the fashion design industry [1, 2] and achieved better results in recent years. However, what garment industry need is a two-dimensional design rather than three-dimensional model. So, surface flattening [3–7] plays an important role in the whole garment design process. Almost all garments are nondevelopable [8], and the flattening of a nondevelopable is a complicated process and local distortions are inevitable. While, the existing surface flattening methods for triangulated surfaces have several limitations. A major problem is the existence of gaps and overlaps in the resulting mesh.

For this phenomenon, a flattening method of energy-based fabric deformation model is proposed in this study. The area representation method is more likely to prevent the

triangle area from being reduced to zero during flattening than models that only consider edge deformation energy. Therefore, the fabric energy acquisition method based on area representation is adopted. Traditional energy-based fabric deformation models [9] need to be converted to weft and warp axes before energy can be obtained, which can be cumbersome. In this article, a simplified model of fabric deformation based on energy is proposed. A simplified energy-based fabric deformation model simplifies the calculations; at the same time, the energy relaxation in local flattening eliminates local deformation.

The properties of the materials need to be considered when flattening, so energy-based or physics-based methods are generally used in the flattening process. In the field of computer graphics, flattening is mainly used to expand a three-dimensional grid into a two-dimensional grid to facilitate texture mapping. This process is also called grid

parameterization. To obtain good texture mapping quality, the mesh should be prevented from being deformed as much as possible, and the flattening method based on the problem of minimizing the deformation is often used. There are many flattening methods [10–15], and we briefly summarize the development of flattening methods over the past few decades.

Levy [16] proposed a quasi-conformal parameterization method based on least squares approximation. This method uses an objective function to minimize the angular deformation. First, the complex surface is decomposed into a series of developable surface elements, and then these developed surfaces are decomposed into another series of developable surface elements. After the elements are expanded, they are combined to obtain the final parameterized result. When the elements are expanded, the original orientation of the triangles can be maintained, so there will be no triangle inversion. However, when the boundaries of the elements intersect in the texture space, they may overlap. In this case, we could subdivide the face element until there is no overlap. Wang et al. [17] proposed a new wire warping method for flattening surfaces using feature curves retained by length. In addition, the progressive deformation scheme for local shape control and the global deformation scheme for highly curved surfaces were introduced. Since the invariant length of feature curves were used to generate the 2D patterns, these strictly controlled lengths were inappropriate when highly nondevelopable 3D surfaces were required. Zhang et al. [18] improved Wang's wire warping algorithm by replacing the constant length of the feature curve with an elastic feature curve. We could use this new feature curve to control the shape of a flattening 2D pattern.

Energy-based or physics-based methods are often used in surface flattening. There are many methods to obtain energy [19, 20]. In [21], Afzal proposed a neutrosophic statistical approach to analyze the resistance of conducting material for big data. In [22], Afzal adopted neutrosophic statistical methods; they were more informative, flexible, and adequate than classical statistics for analyzing the measured values of data. McCartney et al. [23] proposed a flattening algorithm to obtain a planar development by minimizing strain energy. The algorithm uses an energy model to deform the edges of the triangle mesh. In addition, darts or gussets can be handled in a triangular mesh to assist in the fitting process. Based on McCartney's algorithm, Wang et al. [24] introduced a spring-mass system from a geometric point of

view, in which the forces generated by the elastic deformation energy stored in the spring-mass system could be used to move these points. The Lagrange equation was applied to release energy and the penalty function was used to prevent overlap. In addition, the energy distribution is displayed using the interpolation function, which is determined by the surface cutting line. Li et al. [25] introduced cross-springs to reduce the deformation of the final surface. The central triangle improves the levelling efficiency and proposes a local correction method to overcome the overlapping problem. Zhong et al. [26] introduced a novel method of surface flattening by opening the curved configuration of each winged triangle pair. The 2D final pattern is generated on the collision plane, and the strain control mechanism is introduced to preserve the area and size of the original 3D surface.

2. The Flattening Process of the Energy-Based Fabric Deformation Model

2.1. A Simplified Energy-Based Fabric Deformation Model. Fabric deformation produces energy. Because of shearing, bending, gravity, and tension, a piece of fabric is thought to have an energy component. Each energy component is represented by one term in the energy equation [24]:

$$E_{\text{total}} = E_{\text{gravitational}} + E_{\text{tensile}} + E_{\text{shear}} + E_{\text{bending}}, \quad (1)$$

where E_{total} is the energy of the entire fabric, $E_{\text{gravitational}}$ is the potential energy of the fabric, E_{tensile} represents the energy produced by tensile energy or compression of the fabric, E_{shear} represents the energy produced by shear, and E_{bending} is the energy produced by bending of the fabric.

In the process of flattening, a single triangle is flattened at a time. The triangle of the 3D shape is designed to be fixed, and its corresponding 2D triangle is variable during the flattening process. Therefore, the energy model will calculate the energy generated by transforming the triangle of the 2D shape to its corresponding 3D triangle. The summation of the shear energy and tensile strain energy indicates the total internal energy of a triangle [24]. Therefore, this fabric model only considers the variation associated with shear and tensile stress. At the same time, it is assumed that the energy generated by the bending is a constant. Tensile energy E_{tensile} is described as follows [24]:

$$E_{\text{tensile}} = \iint \frac{1}{2} K_{su} (S_u - 1)^2 dudv + \iint \frac{1}{2} K_{sv} (S_v - 1)^2 dudv = \frac{1}{2} A \{ K_{su} (S_u - 1)^2 + K_{sv} (S_v - 1)^2 \}, \quad (2)$$

where A is the area of the 2D triangle, K_{su} and K_{sv} are the tensile constants for the U and V directions, respectively, and S_u and S_v (see Section 4.1) are strains. Shear energy E_{shear} is described as follows [24]:

$$E_{\text{shear}} = \iint \left(\frac{1}{2} K_r \phi_v^2 \right) dudv = \frac{1}{2} A K_r \phi_v^2. \quad (3)$$

Similar to (2), A is the area of the 2D triangle, K_r is the shear constant, and ϕ_v (see Section 4.1) is the strain.

To obtain the energy based on fabric deformation, a transition stage of converting 2D pattern coordinates to weft and warp coordinates is necessary in the traditional model. In addition, tedious calculations could take a considerable amount of time. To reduce the tedious work, we propose a novel simplified model that instead uses the variable R_{ratio} during this stage. Based on this simplified energy model, we propose an innovative unified axis system in Section 4.1 that will simplify computation and innovate energy relaxation in Section 4.2, which will increase the convergence speed. While energy data come from uncertain energy, we adopt some neutrosophic statistical methods [27, 28].

2.2. Flattening Process. Given a triangular mesh $\Omega(K, P)$, where $K = \{K_1, K_2, \dots, K_n\}$ is the spatial vertex set, $K_i \in \mathfrak{R}^3$, $P = \{T_1, T_2, \dots, T_n\}$ is the topological connection information of the mesh, and $T = \{v_i, v_j, v_k\}$ is the triangle of the surface mesh. The goal of mesh flattening is to obtain another triangular mesh $\Omega'(K', P)$, where $K' = \{K'_1, K'_2, \dots, K'_n\}$ is a 2D plane point set, $K'_i \in \mathfrak{R}^2$, and K'_i corresponds to K_i . The area of each triangle and the length of each side remain unchanged after flattening. In addition, the entire algorithm flow is shown in Figure 1.

To be flattened, triangle list V , active triangle list A , flattened triangle list F , and initial flattening mesh $\Theta(K'', P)$ are introduced in an initial flattening algorithm.

Initial flattening algorithm (its aim is to obtain the initial flattening mesh Θ of Ω) is as follows:

- (i) Step 1: Initially, add all triangles in Ω to V , and set A and F as empty.
- (ii) Step 2: Seed triangle T_s is searched in list V and then T_s is flattened. Then, T_s is added to the active list A and removed from the available list V . The selection of seed triangle is detailed in Section 3.1.
- (iii) Step 3: Perform a search in list V to find all the triangles that share an edge with T_s and insert them into the tail of list A .
- (iv) Step 4: If the list A is empty, stop processing. Otherwise, obtain the next triangle T from the rest of list A .
- (v) Step 5: Insert triangle T into the tail of list F and then mark T as flattened. A search is performed in list V to find all the triangles that share an edge with T and are not marked as flattened, and then they are inserted into the tail of list A . Return to Step 4.

To optimize vertex V_k and optimize triangle set T_{set} , initial vertex set K'' is introduced in the optimal flattening algorithm.

Optimal flattening algorithm (its aim is to obtain the optimal flattening mesh Ω' of Ω) is as follows:

- (i) Step 1: Obtain the first vertex V_k from K'' .
- (ii) Step 2: A search is performed in list F to find the triangle set T_{set} that contains vertex V_k .

- (iii) Step 3: Each 2D triangle of triangle set T_{set} proceeds with its corresponding 3D triangle of Ω in the unified axis system and then obtains the energy of vertex V_k .
- (iv) Step 4: Calculate the minimum value of the energy at the original position of vertex V_k and its eight directions position. Thereafter, vertex V_k is moved to the minimum energy position.
- (v) Step 5: If the vertices in K'' are all traversed, go to Step 6. Otherwise, take the next vertex V_k . Return to Step 2.
- (vi) Step 6: Iterate Steps 1–5, until all vertex positions are unchanged.

3. Key Procedure for Initial Flattening

3.1. Selection of Seed Triangle. Before flattening, selecting the seed triangle is an important process. The order in which triangles are flattened depends on the selected seed triangle, so selecting the seed triangle can effectively reduce the number of iterations and running time. Li et al. [25] used integer indices to mark all vertices in the mesh. Also, Liu et al. [29] applied the integer index to label all triangular faces in the mesh. However, we use a novel labelling standard that relies on the triangular edges. In addition, the selection algorithm is detailed as follows:

- (1) Set the triangle mesh edge index on the boundary to 0.
- (2) Search all unlabelled triangle mesh edges that have the same vertices as the labelled edge and increase the unlabelled edge index number by 1.
- (3) Repeat Step 2 until all triangular mesh edges are fully labelled.
- (4) Select the largest sum of the three edge indices as the seed triangle. If there is more than one, choose one of them.

3.2. Unconstrained Triangle Flattening. In the phase of surface flattening, there are two kinds of flattening methods: the unconstrained triangle flattening method and the constrained triangle flattening method. When one edge of a triangle has been flattened, the third vertex can be located using the unconstrained triangle flattening method. As shown in Figure 2, $T(v_0, v_1, v_2)$ is a 3D triangle and the vertices are arranged in reverse order. Edge v_0v_1 has been flattened to edge $v'_0v'_1$ and then vertex v_2 is flattened. The angle between the vector $\overrightarrow{v_0v_1}$ and the vector $\overrightarrow{v_0v_2}$ is calculated as follows:

$$\theta = \arccos(\overrightarrow{v_0v_1}, \overrightarrow{v_0v_2}). \quad (4)$$

Vector τ is obtained by rotating vector $\overrightarrow{v'_0v'_1}$ counterclockwise by angle θ . Then, vertex v'_2 is located using the following expression:

$$v'_2 = v'_0 + |\overrightarrow{v_0v_2}| \tau. \quad (5)$$

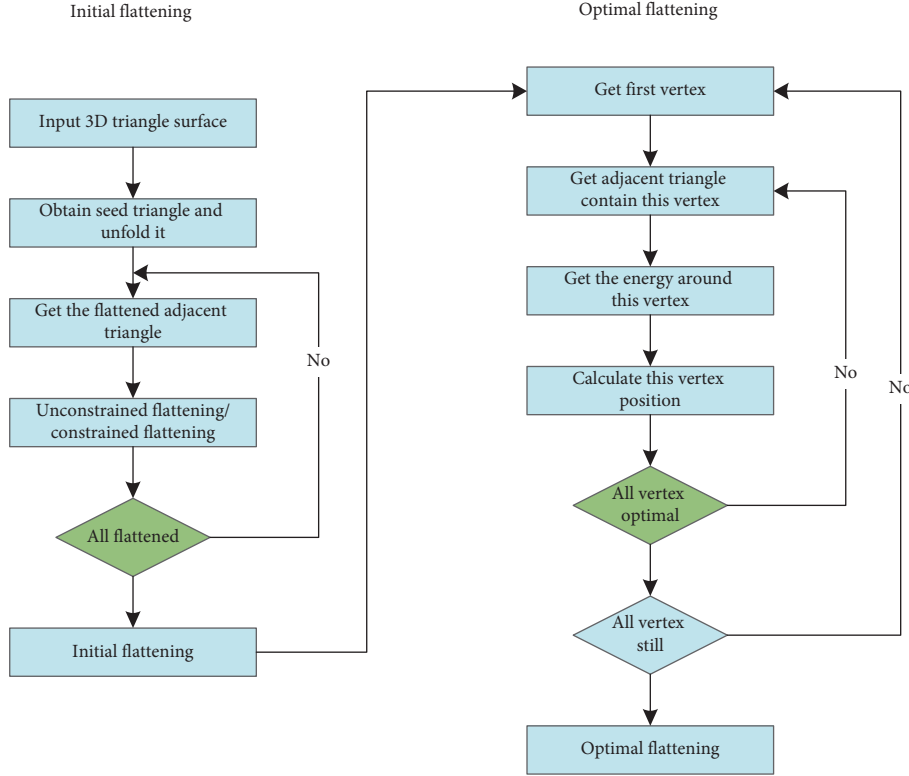


FIGURE 1: Algorithm flow chart.

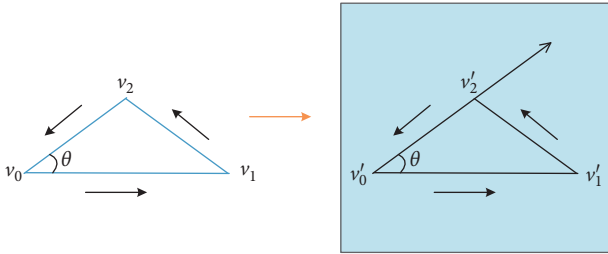


FIGURE 2: Unconstrained triangle flattening.

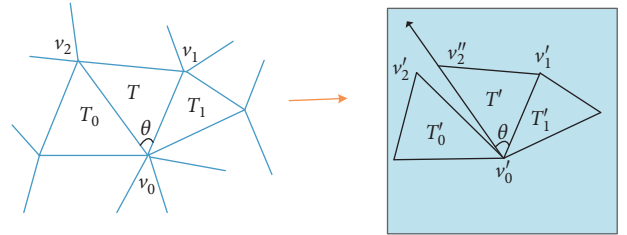


FIGURE 3: Novel constrained triangle flattening.

As a result of flattening, the length of the triangle edge has no change without producing elastic potential energy, which is known as unconstrained triangle flattening.

3.3. Constrained Triangle Flattening. Figures 3 and 4 show two different constrained flattening methods, Figure 4 is a classic traditional method [25] of calculating the intersection point of two circles with two central points and two radii. These circles are centered at v'_0 and v'_1 , and their radii are r_{02} and r_{12} , respectively, as shown in Figure 4. However, this method will produce two intersections. A point within the flattening range needs to be discarded. We use another method to find the third point, as shown in Figure 3. Triangles T_0 and T_1 have been flattened, and our aim is to flatten triangle $T(v_0, v_1, v_2)$. We choose edge $v'_0v'_1$, and vertex v_2 is flattened to v''_2 by using the unconstrained method in Section 3.2. Then we calculate their average coordinate as the flattened vertex, that is

$v''_2 = 1/2(v'_2 + v''_0)$. The side length of the flattened triangle $T'(v'_0, v'_1, v''_2)$ has been changed with respect to T , which will produce elastic potential energy. Therefore, after the initial flattening, optimized flattening is required to release the elastic potential energy.

4. Optimized Flattening

4.1. Process of Obtaining a Unified Axis System. The detailed process of obtaining a unified axis system is shown in

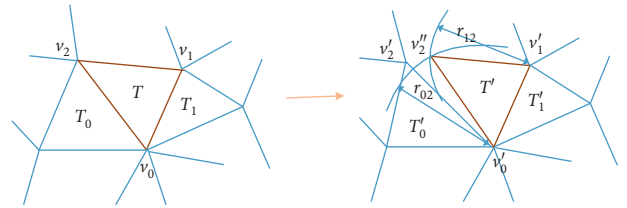


FIGURE 4: Traditional constrained triangle flattening.

Figure 5. P is a 2D triangle and P' is its corresponding 3D triangle. First, the 2D woven triangle without distortion is placed on a uv axis system. The corresponding 3D deformed triangle is then superimposed on the system under the same conditions. The 2D fabric undeformed triangle P without energy, and its corresponding 3D fabric triangle P' with energy is generated after geometric deformation. The strain variables S_u and S_v , as well as the shear variable ϕ_v , transform position v_1 to v'_1 and v_2 to v'_2 , respectively.

As shown in the right part of Figure 5, v_0 is located at the origin of the coordinates. The three-side length of the 2D triangle P is known, and the coordinates of v_1 and v_2 can be obtained using (6) and (7), where θ is the angle between axis u and edge v_0v_2 . ∂ is the angle between edge v_0v_1 and edge v_0v_2 . By setting the value of R_{ratio} as 1/2, the flattening could achieve a better effect. The v'_1 and v'_2 coordinates of the 3D triangles can be obtained in the same way.

$$R_{\text{ratio}} = \frac{|v_0v_2|\sin\theta}{|v_0v_2|\sin\theta + |v_0v_1|\sin(\partial - \theta)}, \quad (6)$$

$$\partial = \arccos\left(\frac{|v_0v_1|^2 + |v_0v_2|^2 - |v_1v_2|^2}{2|v_0v_1||v_0v_2|}\right). \quad (7)$$

The unifying process completes the specification geometric transformation that a representative rectangle must undergo. This geometric transformation can also deform the triangle P to P' . The total energy required for the distortion can be calculated by determining the energy to distort the representative weave element. Thereafter, considering it in its entirety, we utilized the same transformation for the triangle. The distortion of the representative woven element can be represented by an affine transformation. The scaling factor and shear angle can be calculated from affine

transformation [30]. The general representation of an affine transformation is as follows:

$$[u'v'1] = [uv1] \begin{pmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & 0 \end{pmatrix}. \quad (8)$$

For a particular affine transformation, the affine transformation matrix will now be referred to as M , where this matrix is comprised as follows:

$$[u'v'1] = [uv1]M. \quad (9)$$

The shape of a triangular fabric is deformed first by stretching and then by shearing; M is therefore represented by a matrix:

$$M = \begin{pmatrix} S_u & 0 & 0 \\ 0 & S_v & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ \sin\phi_v & \cos\phi_v & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} S_u & 0 & 0 \\ \sin\phi_v & S_v \cos\phi_v & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (10)$$

For this special change, coordinate v_1 transforms to coordinate v'_1 and coordinate v_2 transforms to v'_2 :

$$(v_{11}, v_{12}) \longrightarrow (v'_{11}, v'_{12}), (v_{21}, v_{22}) \longrightarrow (v'_{21}, v'_{22}). \quad (11)$$

The equation can be rewritten again as shown below:

$$\begin{pmatrix} 0 & 0 & 1 \\ v'_{11} & v'_{12} & 1 \\ v_{21} & v'_{22} & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ v_{11} & v_{12} & 1 \\ v_{21} & v_{22} & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & 1 \end{pmatrix}, \quad (12)$$

or $U' = UA$, and this results in:

$$A = \frac{1}{v_{11}v_{22} - v_{21}v_{12}} \times \begin{pmatrix} (v_{22}v'_{11} - v_{12}v'_{21}) & (v_{22}v'_{12} - v_{12}v'_{22}) & 0 \\ (v_{11}v'_{21} - v_{21}v'_{11}) & (v_{11}v'_{22} - v_{21}v'_{12}) & 0 \\ 0 & 0 & (v_{11}v_{22} - v_{21}v_{12}) \end{pmatrix}. \quad (13)$$

By comparing (10) and (13), the strain variables S_u and S_v , and the shear variable ϕ_v are as follows:

$$S_u = \frac{(v_{22}v'_{11} - v_{12}v'_{21})}{(v_{11}v_{22} - v_{21}v_{12})}, \quad (14)$$

$$\phi_v = \tan^{-1} \frac{(v_{11}v'_{21} - v_{21}v'_{11})}{(v_{11}v'_{22} - v_{21}v'_{12})}, \quad (15)$$

$$S_v = \frac{(v_{11}v'_{21} - v_{21}v'_{11})}{(v_{11}v_{22} - v_{21}v_{12})\sin\phi_v}. \quad (16)$$

The steps for the energy of the triangular fabric deformation are given as follows:

- (1) Obtain 3D coordinates of the triangle
- (2) Obtain the 2D coordinates of the triangle
- (3) Unify the 3D and 2D triangles into the (U, V) axes with equations (6) and (7)
- (4) According to the two unified triangular coordinates, acquire the strain variables S_u and S_v and the shear variable ϕ_v by applying equations (14)–(16)
- (5) Use equations (2) and (3) to obtain tensile energy E_{tensile} and shear energy E_{shear} respectively

4.2. Energy Relaxation Process in Partial Flattening. Energy relaxation plays an important role through the flattening process, which determines the efficiency of

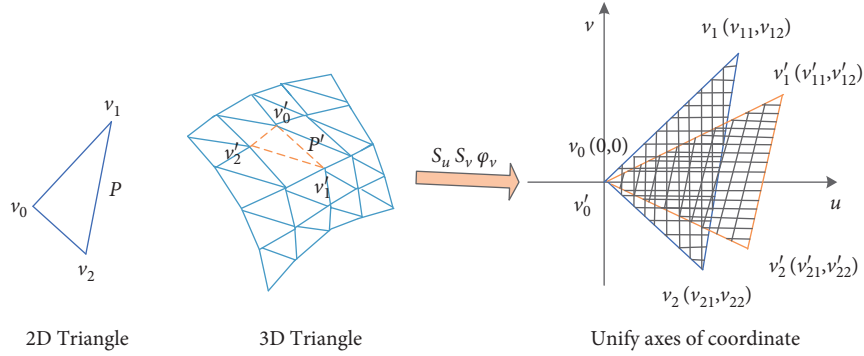


FIGURE 5: Process of unify axis coordinate.

flattening and the effect of flattening. In previous studies, most works have been conducted to determine how to use the energy released to achieve a satisfactory flattening result. Wang et al. [24] used the spring force obtained by the Lagrange equation to move the vertex, and a better result could be achieved after many iterations, but the obvious drawback is the time-consuming process. However, we propose an innovative method of simplifying fabric deformation based on energy in Section 2.1. As this method adds four orthogonal directions in the specific node, it shows the weighted optimization efficiency and optimization effect. The original position of the node is considered when calculating the next position of the node; the details of the process are described below.

As shown in Figure 6, we assumed that there are 2D vertices P and adjacent vertices P_1, P_2, \dots, P_5 . The triangles formed with P are T_0, T_1, \dots, T_4 . The energy of the vertex P is obtained by the following expression, where $E_{\text{energy}}(i)$ is the energy of the i th triangle connected to P and n is the number of triangles connected to vertex P .

$$E_P = \sum_{i=0}^{n-1} E_{\text{energy}}(i). \quad (17)$$

The eight directions of optimized flattening are $\vec{n}, n+1, \dots, n+7$, and the movement increment is Δd . The next position of P will slip to the minimum energy value direction. The minimum energy value E_{\min} can be obtained as follows:

$$E_{\min} = \min\{E_P, E \rightarrow_{op+n} \rightarrow, E \rightarrow_{op+n+1} \rightarrow, E \rightarrow_{op+n+2} \rightarrow, \dots, E \rightarrow_{op+n+7} \rightarrow\}. \quad (18)$$

5. Realization of the Virtual Try-On Process

The realization of the virtual stitching process is an important step of the virtual try-on. Before virtual garment is flattened, the virtual human body needs to be tried-on.

5.1. Virtual Stitching. When user selects the stitching edge, if the start point and the end point of the two corresponding edges are selected in the same direction (clockwise or counterclockwise), the corresponding relationship between the points on the two edges is correct, as shown in Figure 7(a). If different directions are chosen, that is, one side selects the start and end points in a clockwise direction, and the other side selects the start and end points in a counterclockwise direction, the corresponding relationship between the points on the two sides will be wrong, as shown in Figure 7(b).

The specific representation of stitching is shown in Figure 7(c). The brown dots represent discrete mass points of the garment piece, and Dis represents the distance between the two mass points. Among them, point A is applied with a stitching force in the direction of \vec{AB} , point B is applied with a stitching force in the direction of \vec{BA} , point A and point B

will gradually approach each other. The stitching force is defined as a linear function corresponding to the distance between stitching points. For two stitching points A and B , the stitching force can be calculated as follows:

$$\vec{F}_{\text{stitching force}} = -k \bullet \vec{l}. \quad (19)$$

Among them, k is the stitching force coefficient, which is related to the stitching performance of the fabric. Generally, a larger stitching force coefficient is used for fabrics that are more difficult to deform; \vec{l} represents the distance direction vector from the stitching point A to B . The closer the distance between the stitching points, the smaller the stitching force.

5.2. Virtual Try-On. The complete virtual stitching process is shown in Figure 8. First, seam lines are explicitly specified by choosing pairs of panel boundary edges. Then, the designed coats are assembled and linked by seaming lines to simulate clothing behavior on the 3D mannequin. By applying elastic force on the seam lines, the garment patterns can be connected to each other during the stitching process, as shown in Figure 8(a). After stitching, the virtual human completes the 3D virtual garment try-on, as shown in Figure 8(b).

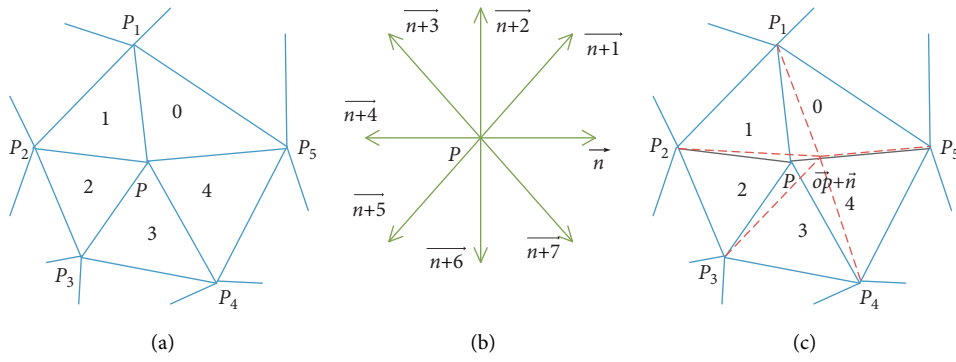


FIGURE 6: Energy relaxation procedure.

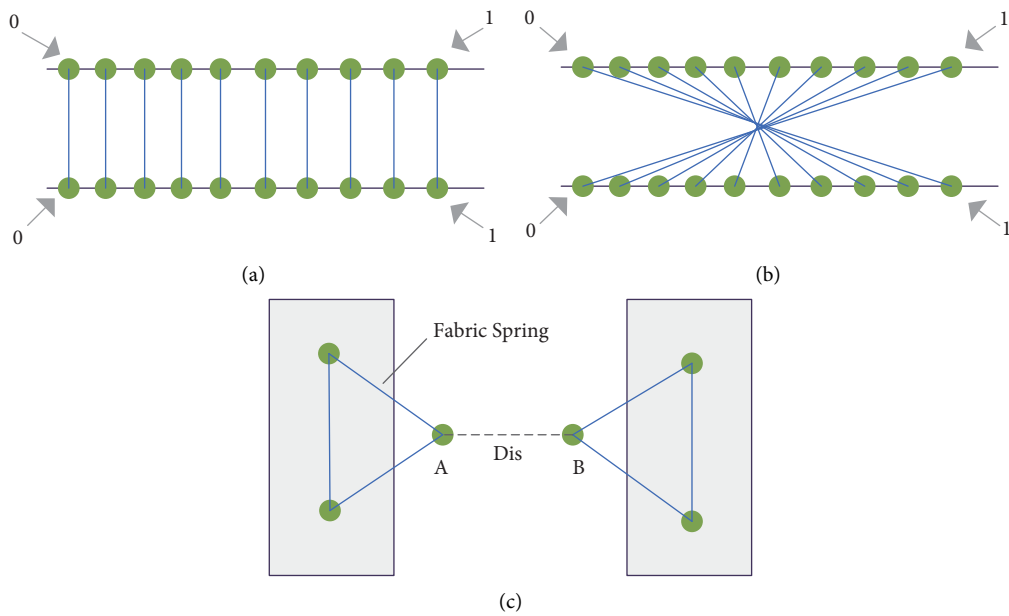


FIGURE 7: Diagram of virtual stitching information: (a)correct correspondence, (b)incorrect correspondence, and (c)diagram of stitching.

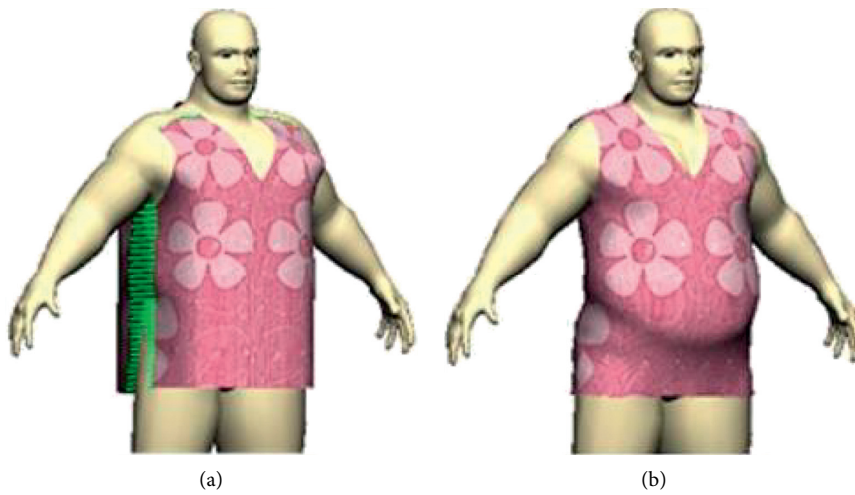


FIGURE 8: Diagram of virtual dressing process: (a) virtual stitching in progress and (b) virtual stitching completed.

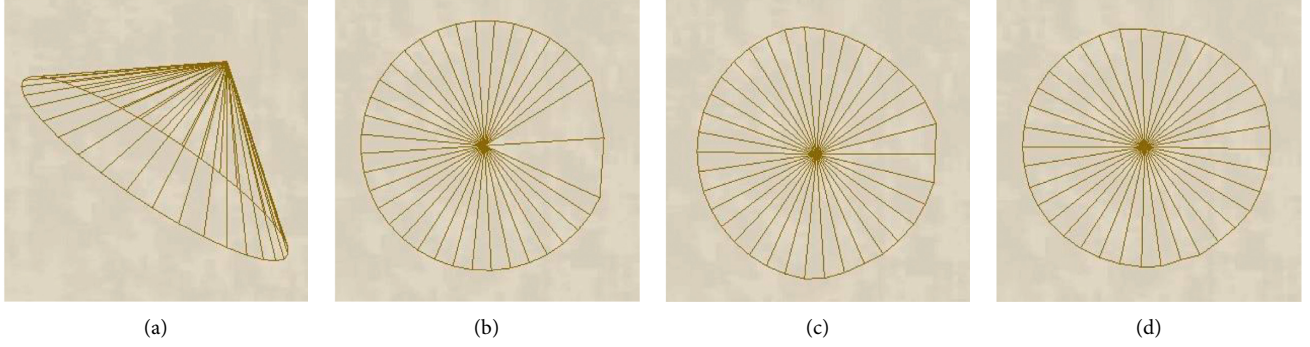


FIGURE 9: Optimized flattening for the 36-sided cone model: (a) 3D original Mesh, (b) initial flattening, (c) optimized iteration after 170 times, and (d) final flattening result.

6. Experiments and Discussion

This article introduces an energy-based fabric deformation algorithm, which is implemented by the object-oriented Visual C++ language. It is implemented on a PC with a Core i5 CPU and 4 GB RAM. A series of triangular mesh object surfaces are presented to measure the performance of our algorithm. As classical geometrically shaped parts of a torus, the garment shapes of coats and pants are selected as our example.

We used the typical 36-sided cone model as an example to test our optimized flattening method. The 3D model is shown in Figure 9(a). The initial flattening result is shown in Figure 9(b) and Figure 9(c) shows the optimized flattening results after 170 iterations. The final flattening result is shown in Figure 9(d).

Three criteria are used to evaluate the surface accuracy after flattening, including area accuracy, shape accuracy, and energy accuracy. The area accuracy and shape accuracy were proposed by Wang et al. [24]. The energy accuracy is presented in this study based on an energy-based fabric deformation model.

Area accuracy:

$$E_{\text{area}} = \frac{\sum_{i=0}^{n-1} |a_i - a'_i|}{\sum_{i=0}^{n-1} a_i}, \quad (20)$$

where E_{area} is the accuracy of the area after flattening the entire surface; a_i is the i th triangle area before flattening and a'_i is its corresponding triangle after flattening; and E_{energy} is the number of triangles.

Shape accuracy:

$$E_{\text{edge}} = \frac{\sum_{i=0}^{m-1} |e_i - e'_i|}{\sum_{i=0}^{m-1} e_i}, \quad (21)$$

where E_{edge} is the accuracy of the shape after flattening the entire surface; n is the i th edge length before flattening and e'_i is the length of its corresponding edge after flattening; and m is the number of triangular mesh edges.

Energy accuracy:

$$E_{\text{energy}} = \sum_{i=0}^{n-1} E_{\text{energy}}(i), \quad (22)$$

where E_{energy} is the energy accuracy of the entire surface after flattening; $E_{\text{energy}}(i)$ is the i th triangle energy after flattening, and the corresponding triangle energy before flattening is zero; and n is the number of triangles on the entire surface.

Table 1 shows the 36-sided cone measurement accuracy data using the three kinds of surface accuracy methods. As we can see from Table 1, the three measurement accuracies will decrease during the optimized flattening process, and the minimum value is achieved in the final process. The value of area accuracy, shape accuracy, and energy accuracy in the final process is 0.397%, 1.053%, and 5.1, respectively.

We compare our optimized method with the classical mass-spring energy relaxation method, and Figures 10(a)–10(d) show the flattening result of the partial torus. Figure 10(a) is the original 3D mesh model of a part of the torus. As shown in Figure 10(b), the initial flattening result has great distortion. Figure 10(c) is the result of optimization based on the fabric deformation model proposed in this study, and the result of mass-spring energy relaxation is shown in Figure 10(d). From the appearance point of view, the optimization result in Figure 10(c) is slightly better than that in Figure 10(d).

The performance accuracy is listed in Table 2. In our experiment, the area accuracy, shape accuracy, and energy accuracy after mass-spring energy relaxation are 1.201%, 1.533%, and 20.2 Nmm, respectively. The value of the area accuracy, shape accuracy, and energy accuracy after the woven fabric-based method is 1.073%, 1.342%, and 10.3Nmm, respectively.

The following two groups of experiments are applied in the garment design process. Figure 11(a) shows the 3D triangular mesh surface of a skirt, with 3414 triangular faces and 1851 vertices. Figure 11(b) shows the flattening result after optimization.

Figure 12(a) shows the 3D triangular mesh surface of a coat, with 3492 triangular faces and 1965 vertices. Figure 12(b) is the result of its optimized flattening.

To illustrate the flattening effect of our method, we added two group experiments of pant and vest in the garment design process. Figure 13(a) shows the 3D triangular mesh surface of a pant, with 3852 triangular faces and 1963 vertices. Figure 13(b) shows the flattening result after optimization.

TABLE 1: Accuracy statistic of the 36-sided cone.

	Initial	Optimized (1)	Optimized (2)	Final
Area (%)	0.897	0.682	0.564	0.397
Shape (%)	1.642	1.435	1.211	1.053
Energy (Nmm)	23.1	12.2	7.9	5.1

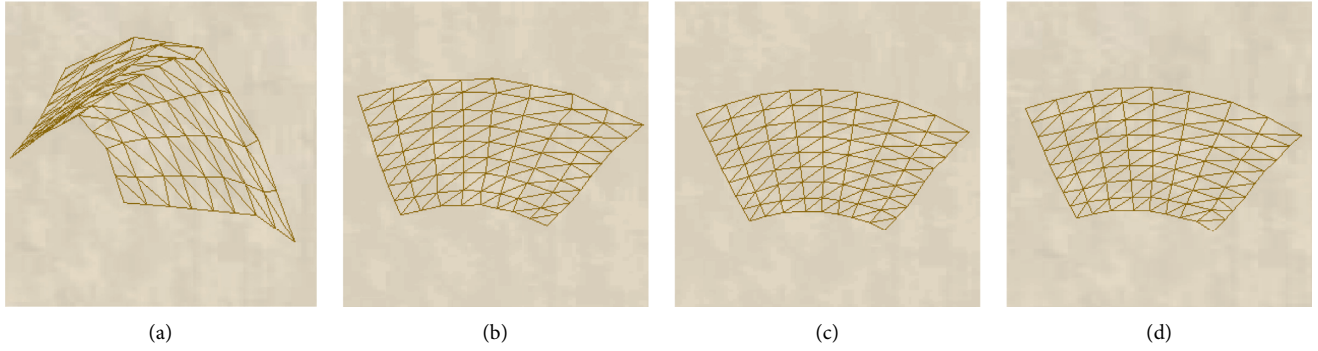


FIGURE 10: Comparison of optimization with the woven fabric method and mass-spring method: (a) original 3D mesh, (b) initial flattening, (c) optimized with woven fabric method, and (d) optimized with mass-spring method.

TABLE 2: Accuracy statistics of a torus.

	Initial	Woven fabric	Mass-spring
Area (%)	1.932	1.073	1.201
Shape (%)	2.53	1.342	1.533
Energy (Nmm)	62.1	10.3	20.2

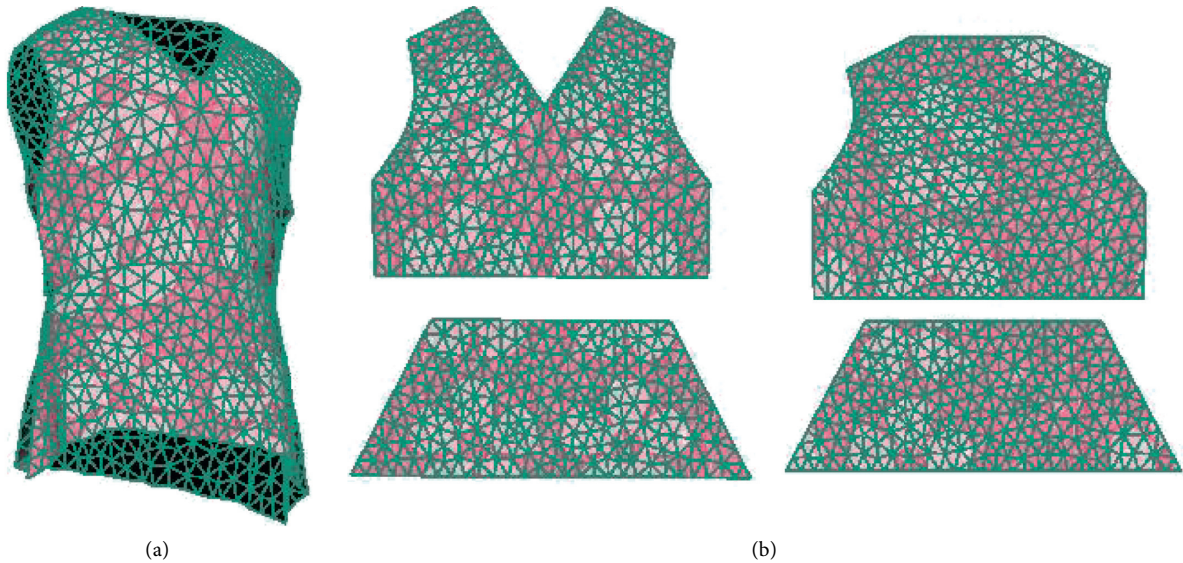


FIGURE 11: Energy-based flattening method for a skirt: (a) original 3D mesh of a skirt and (b) final 2D flattening pattern of a skirt.



FIGURE 12: Energy-based flattening method for a coat: (a) original 3D mesh of a coat and (b) final 2D flattening of a coat.

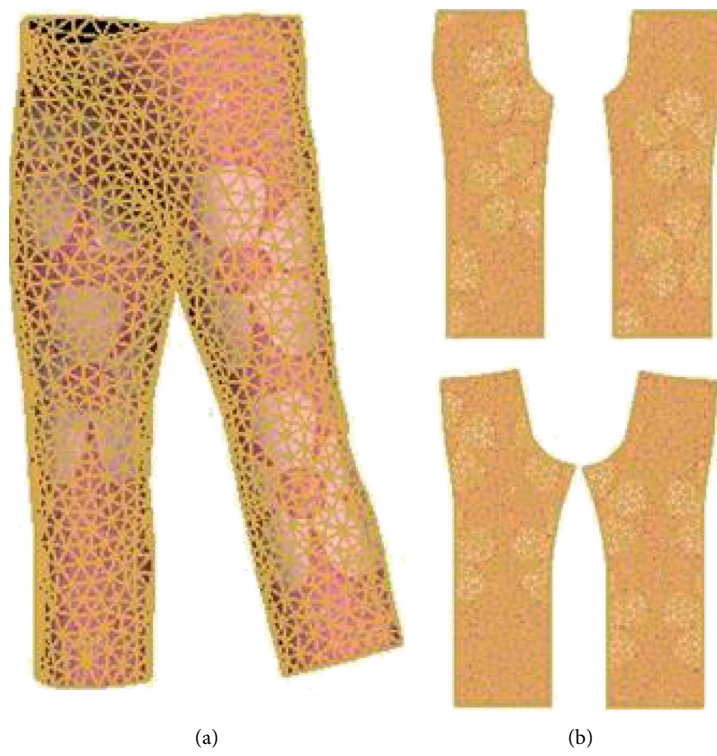


FIGURE 13: Energy-based flattening method for a pant: (a) original 3D mesh of a pant and (b) final 2D flattening pattern of a pant.

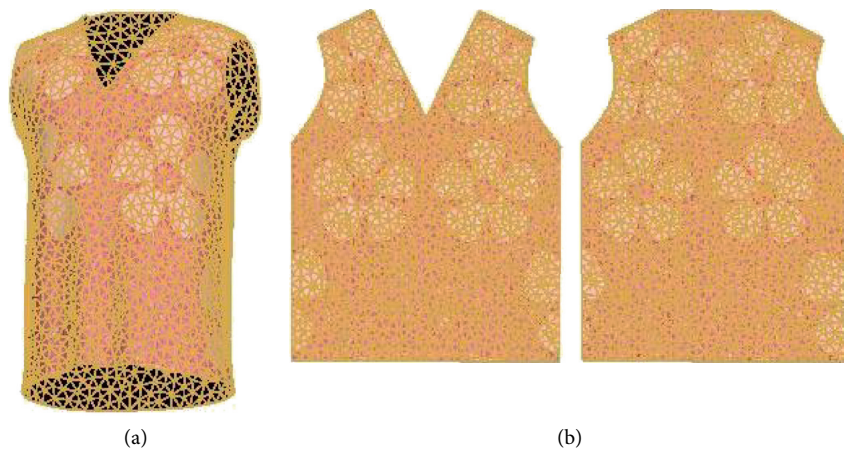


FIGURE 14: Energy-based flattening method for a vest: (a) original 3D mesh of a vest and (b) final 2D flattening pattern of a vest.

Figure 14(a) shows the 3D triangular mesh surface of a vest, with 3752 triangular faces and 1857 vertices. Figure 14(b) is the result of its optimized flattening.

7. Conclusion and Future Work

In this article, an optimized flattening method based on the fabric energy deformation model is proposed. First, the 3D triangle mesh is flattened. The selection of seed triangles can effectively reduce the number of iterations and runs. A new markup algorithm using the edges of triangles is adopted, and the constraint triangle flattening method generates the elastic potential energy and then optimizes the initial flattening results. Second, in the process of optimizing flattening, a new unified axis system process is proposed to obtain the deformation energy, and an energy relaxation method is used to release energy to eliminate the local deformation caused by initial flattening. Finally, based on the model of fabric deformation, which is determined by the energy, a new energy precision is proposed in this study to evaluate the surface accuracy after flattening. The experimental results show that the complex surface of clothing can obtain a better flattening effect. Our approach is more flexible and practical in clothing design than other energy-based methods. In the future, we will consider different materials and other physical parameters of the material in the optimized flattening process for better flattening results.

Data Availability

The research data used to support the findings of this study are available from the corresponding author upon request.

Disclosure

The authors have presented the manuscript in “2017 International Conference on Virtual Reality and Visualization (ICVRV).”

Conflicts of Interest

No conflicts of interest exist in the submission of this manuscript.

Acknowledgments

The authors thank Prof. Yanjun Peng at the Shandong University of Science and Technology for his valuable comments, Dr. Yingran Ma at the Shandong University of Science and Technology for his data collection, Mrs. Yuanhong Wang at the Shandong University of Science and Technology for her literature collection, and Assoc Prof. Mingmin Zhang at the Zhejiang University for discussion. This work was supported by the Key Science and Technology Research of Henan Province under Grant nos. 212102210516 and 212102210255, the National Natural Science Foundation of China under Grant no. 61976126, and the Natural Science Foundation of Shandong Province under Grant no. ZR2019MF003.

References

- [1] N. Umetani, D. M. Kaufman, T. Igarashi, and E. Grinspun, “Sensitive couture for interactive garment modelling and editing,” *ACM Transactions on Graphics*, vol. 30, pp. 1–11, 2011.
- [2] A. Bartle, A. Sheffer, V. G. Kim, D. M. Kaufman, and N. Vining, “Physics-driven pattern adjustment for direct 3D garment editing,” *ACM Transactions on Graphics*, vol. 35, no. 4, pp. 1–11, 2016.
- [3] S. Lu, P. Y. Mok, and X. Jin, “A new design concept: 3D to 2D textile pattern design for garments,” *Computer-Aided Design*, vol. 89, pp. 35–49, 2017.
- [4] H. Q. Huang, P. Y. Mok, Y. L. Kwok, and J. S. Au, “Block pattern generation: from parameterizing human bodies to fit feature-aligned and flattenable 3D garments,” *Computers in Industry*, vol. 63, pp. 680–691, 2012.
- [5] Y. J. Liu, D. L. Zhang, and M. M. F. Yuen, “A survey on CAD methods in 3D garment design,” *Computers in Industry*, vol. 61, pp. 576–593, 2010.
- [6] C. K. Au and Y. X. Ma, “Garment pattern definition, development and application with associative feature approach,” *Computers in Industry*, vol. 61, no. 6, pp. 524–531, 2010.
- [7] S. W. Hsiao, R. Q. Chen, and C. Y. Lin, “Surface flattening assisted with 3D mannequin based on minimum energy,” *World Academy of Science, Engineering and Technology*, vol. 8, pp. 1444–1449, 2014.
- [8] R. Goldman, “Curvature formulas for implicit curves and surfaces,” *Computer Aided Geometric Design*, vol. 22, pp. 632–658, 2005.
- [9] J. McCartney, B. Hinds, and K. W. Chong, “Pattern flattening for orthotropic materials,” *Computer Aided Design*, vol. 37, pp. 631–644, 2005.
- [10] A. Sheffer, B. Lévy, M. Mogilnitsky, and A. Bogomyakov, “ABF++: fast and robust angle based flattening,” *ACM Transactions on Graphics*, vol. 24, no. 2, pp. 311–330, 2005.
- [11] Y. C. Yang and W. Y. Zhang, “Prototype garment pattern flattening based on individual 3D virtual dummy,” *International Journal of Clothing Science and Technology*, vol. 19, pp. 334–348, 2007.
- [12] Y. Yang, W. Zhang, and C. Shan, “Investigating the development of digital patterns for customized apparel,” *International Journal of Clothing Science and Technology*, vol. 19, pp. 167–177, 2007.
- [13] A. Jung, S. Hahmann, D. Rohmer, A. Begault, L. Boissieux, and M. P. Cani, “Sketching folds: developable surfaces from non-planar silhouettes,” *ACM Transactions on Graphics*, vol. 34, pp. 1–12, 2015.
- [14] C. C. L. Wang, “Towards flattenable mesh surfaces,” *Computer Aided Design*, vol. 40, pp. 109–122, 2008.
- [15] J. Hoschek, “Approximation of surfaces of revolution by developable surfaces,” *Computer Aided Design*, vol. 30, pp. 757–763, 1998.
- [16] B. Lévy, S. Petitjean, N. Ray, and J. Maillot, “Least squares conformal maps for automatic texture atlas generation,” *ACM Transactions on Graphics*, vol. 21, pp. 362–371, 2002.
- [17] C. C. L. W. Wang, “A fast surface flattening approach with length-preserved feature curves,” *Computer Aided Design*, vol. 40, pp. 381–395, 2008.
- [18] Y. B. Zhang and C. C. L. Wang, “WireWarping++: robust and flexible surface flattening with length control,” *IEEE Transactions on Automation Science and Engineering*, vol. 8, pp. 205–215, 2011.

- [19] M. A. U. Haq, S. R. Gadde, M. Albassam, and M. Aslam, "Marshall-olkin power lomax distribution for modeling of wind speed data," *Energy Reports*, vol. 6, pp. 1118–1123, 2021.
- [20] M. Aslam and A. Algarni, "Analyzing the solar energy data using a new Anderson-Darling test under indeterminacy," *International Journal of Photoenergy*, vol. 2020, Article ID 6662389, 6 pages, 2020.
- [21] U. Afzal, H. Alrweili, N. Ahamd, and M. Aslam, "Neutrosophic statistical analysis of resistance depending on the temperature variance of conducting material," *Scientific Reports*, vol. 11, no. 1, Article ID 23939, 2021.
- [22] U. Afzal, N. Ahmad, Q. Zafar, and M. Aslam, "Fabrication of a surface type humidity sensor based on methyl green thin film, with the analysis of capacitance and resistance through neutrosophic statistics," *RSC Advances*, vol. 11, pp. 38674–38682, 2021.
- [23] J. McCartney, B. K. Hinds, and B. L. Seow, "The flattening of triangulated surfaces incorporating darts and gussets," *Computer Aided Design*, vol. 31, pp. 249–260, 1999.
- [24] C. C. L. Wang, S. S. F. Smith, and M. M. F. Yuen, "Surface flattening based on energy model," *Computer Aided Design*, vol. 34, pp. 823–833, 2002.
- [25] J. Li, D. Zhang, G. Lu, Y. Peng, X. Wen, and Y. Sakaguti, "Flattening triangulated surfaces using a mass-spring model," *The International Journal of Advanced Manufacturing Technology*, vol. 25, no. 1-2, pp. 108–117, 2005.
- [26] Y. Zhong and B. Xu, "A physically based method for triangulated surface flattening," *Computer-Aided Design*, vol. 38, no. 10, pp. 1062–1073, 2006.
- [27] M. Aslam, "Analyzing wind power data using analysis of means under neutrosophic statistics," *Soft Computing*, vol. 25, no. 10, pp. 7087–7093, 2021.
- [28] M. Aslam, M. Azam, and M. Albassam, "Sampling plan using process loss index using multiple dependent state sampling under neutrosophic statistics," *IEEE Access*, vol. 7, pp. 38568–38576, 2019.
- [29] Q. S. Liu, J. T. Xi, and Z. Q. Wu, "An energy-based surface flattening method for flat pattern development of sheet metal components," *International Journal of Advanced Manufacturing Technology*, vol. 68, pp. 1155–1166, 2013.
- [30] P. J. Schneider and D. H. Eberly, *Geometric Tools for Computer Graphics*, pp. 98–99, Morgan Kaufmann Publisher, Burlington, MA, USA, 2002.