Int J Adv Manuf Technol (2016) 87:2873–2887
DOI 10.1007/s00170-016-8669-2

ORIGINAL ARTICLE

Optimal fitting of strain-controlled flattenable mesh surfaces

Yunbo Zhang[1,2] · Charlie C. L. Wang[2,3] · Karthik Ramani[1]

Received: 29 December 2015 / Accepted: 22 March 2016 / Published online: 6 April 2016

Abstract A flattenable mesh surface is a polygonal mesh surface that can be unfolded into a planar patch without stretching any polygon. This paper presents a new method for computing a slightly stretched flattenable mesh surface M from a piecewise-linear surface patch $P \in \Re^3$, where the shape approximation error between M and P is minimized and the strain of stretching on M is controlled. Prior approaches result in either a flattenable surface that could be quite different from the input shape or a (discrete) developable surface has relative simple shape. The techniques investigated in this paper overcome these difficulties. First, we introduce a new surface modeling method to conduct a sequence of nearly isometric deformations to morph a flattenable mesh surface to a new shape which has a better approximation of the input surface. Second, in order to get better initial surfaces for fitting and overcome topological obstacles, a shape perturbation scheme is investigated to obtain the optimal surface fitting result. Last, to improve the scalability of our optimal surface fitting algorithm, a coarse-to-fine fitting framework is exploited so that very dense flattenable mesh surfaces can be modeled and boundaries of the input surfaces can be interpolated.

✉ Yunbo Zhang
will.yunbo.zhang@gmail.com

[1] School of Mechanical Engineering, Purdue University, West Lafayette, IN, USA

[2] Department of Mechanical and Automation Engineering, The Chinese Univerity of Hong Kong, Sha Tin, Hong Kong

[3] Shenzhen Institues of Advanced Technology, Shenzhen, China

Keywords Surface approximation · Flattenable mesh surface · Optimization · Stretching controlled · Coarse-to-fine fitting

1 Introduction

For human centric products fabricated by planner materials such as garments, shoes, and furniture, consumers prefer customization. However, due to lack of support from design and manufacturing automation, customization for human centric products is usually made manually in a trial-and-error manner. Compared with mass produced products, current customized products consume more manufacturing time and labors, therefore, become much higher priced and thereby are not popularized. Moreover, the accuracy of products cannot be guaranteed because of the absent of design and manufacturing automation. There are some efforts [1] made trying to solve this problem by dividing customized products design into four steps: (1) human body scan (Fig. 1a); (2) human body reconstruction and processing (such as Fig. 1b, c); (3) design on/around human body (Fig. 1d–f) and manufacturing (Fig. 1g). These four steps can be thought as the general working flow for automatized customized products design and fabrication. With the popularization of low-cost consumer level depth camera such as Microsoft Kinect and development of computational methods associated such as [2], steps 1 and 2 are no longer barriers to keep consumers from easily obtaining their 3D body shapes. Based on given 3D human model, our previous work [1] presents a computer-aided design framework with various of automated or interactive tools for designing human centric customized products fabricated by planner materials. The designed 3D shape can be fabricated by

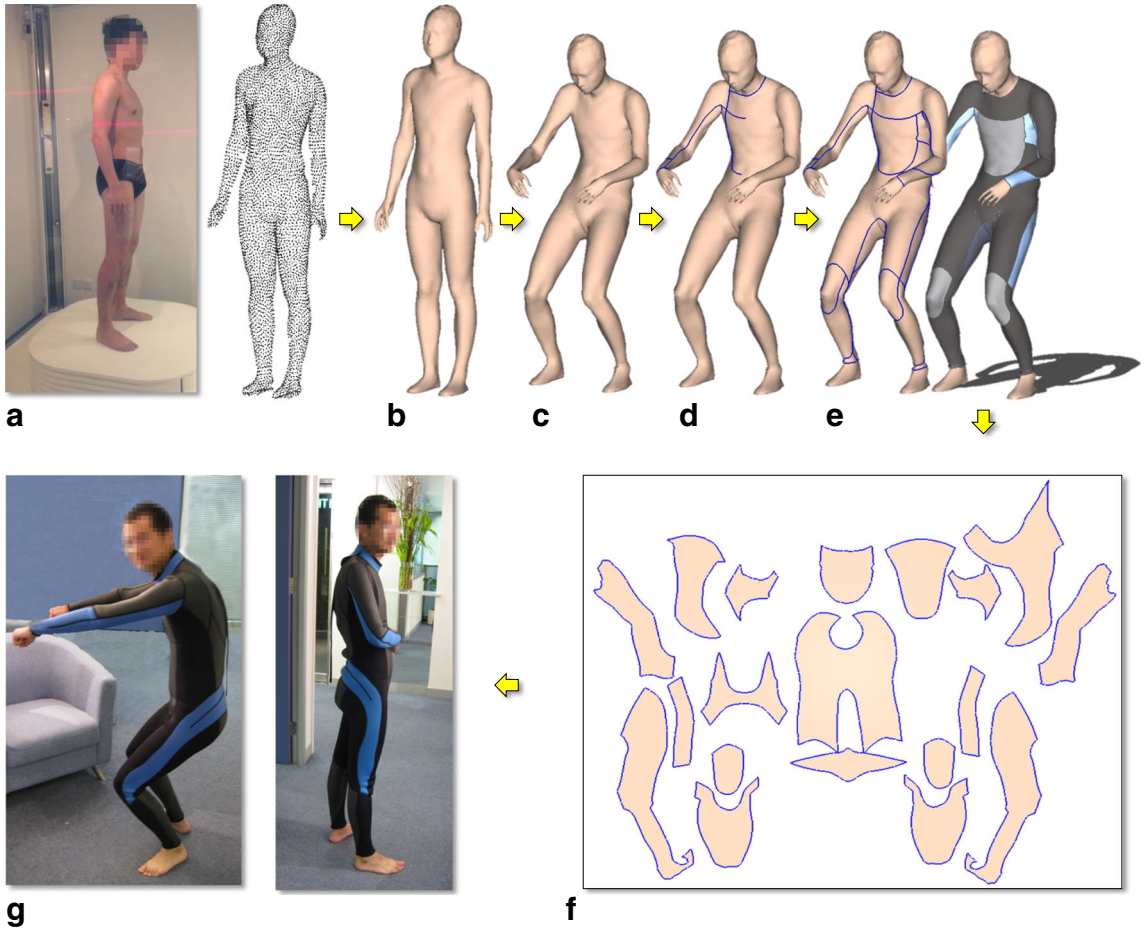

Fig. 1 A typical work flow for customized wetsuit design—human body point cloud data is acquired by 3D scanner (**a**), and a mesh surface is reconstructed (**b**). The mesh model is then deformed into a posture when playing windsurfing. A wetsuit is designed via a CAD system [1] (see (**e**)), processed by our approach and then flattened onto 2D (**f**). Eventually, a wetsuit is fabricated for a windsurfing player according to the 2D shapes (**g**)

warping the 2D pattern back to 3D. From the knowledge of differential geometry [3], only for developable surfaces, there exist 2D shapes which can be warped back to 3D without any distortion, but the design 3D shapes are usually not developable. Therefore, an urgent problem to be solved is how to convert the 3D design into a shape which can be fabricated by planner materials with as little as possible approximation error with 3D design. In practice, the planner materials usually allow certain amount of stretchiness (measured by strain) during the fabrication. The allowance strains can be varied for different materials from less than 1 % for leather in shoe design, within 5 % for cotton fabric in jeans design and even high to around 10 % for Neoprene for wetsuit design.

Problem definition Given a designed 3D product H represented by an assembly of several polygonal meshes (which may be very dense) in disk-like topology, strain-controlled flattenable mesh surfaces are computed to approximate the surface patches on H. For any surface patch $P \in \Re^3$ on H, the computed mesh surface M should have a controlled stretching strain compared with a fully flattenable mesh surface and give a small shape approximation error between P and M. In differential geometry [3], developable surfaces are used to model 3D shape that can be warped from inextensible materials. Nevertheless, the shape of modern products could be complex (e.g., having freeform surfaces, irregular boundaries, etc.), so it is difficult to model them by developable surfaces although developable surfaces can always be warped from planar materials without stretching. The shape optimization methods developed in prior researches for discrete developable surfaces [4] (also called *flattenable mesh surfaces* in [5]) provide some preliminary trials of modeling surfaces in crumpling. However, they have defects in either of the following two aspects: (1) the processed surface could be quite different from the input shape (i.e., the fitting error from a flattenable mesh surface to the input surface patch has not been minimized [5]—see also the example shown in Fig. 2) or (2) only relatively simple surface patches can be processed (e.g., the developable

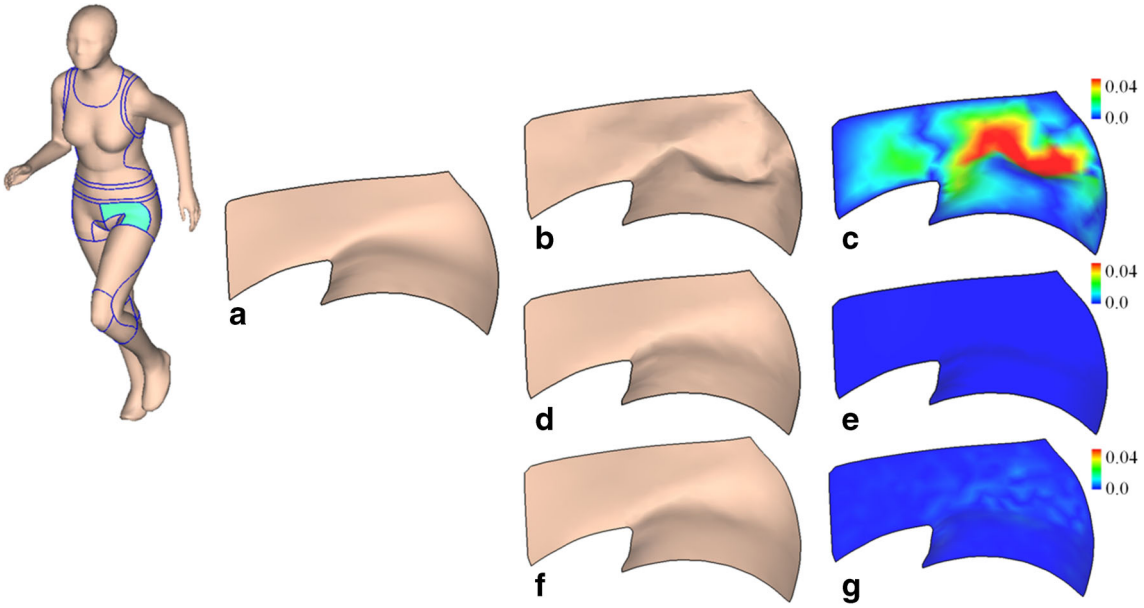

Fig. 2 For a given surface (**a**) which is highly curved, the resultant surface generated by the flattenable mesh processing technique [6] can be fully developable. However, the processed flattenable mesh surface has a large shape variation to the input (see the shape approximation error shown in (**c**)). An resultant flattenable mesh surface after fitting with 10 % maximum strain having a better fitting result to the input is shown in (**d**), which has a much smaller shape approximation error (see (**e**)). The final result after relaxation with more tight strain limit (7.5 % in maximum) is shown in (**f**), but the shape approximation (see (**g**)) is still better than the one in (**c**)

surface strips in [4]). Moreover, fully inextensible planar materials (e.g., paper) are rarely used in commercial production as they provide less flexibility for modeling highly curved shapes. Slightly extensible planar materials are more widely used, which allow a certain level of in-plate stretch during fabrication (e.g., less than 5 % in textile industry). This paper investigates techniques in modeling the complex shape of products that are fabricated from slightly extensible planar materials. This leads to a new geometric modeling framework for designing such products more efficiently and accurately, which will benefit various industrial applications (e.g., ship and airplane building and design automation of user-customized clothes). The techniques exploited here include a surface fitting method for flattenable mesh surfaces, a shape perturbation-based scheme for obtaining optimal fitting results, and a coarse-to-fine fitting framework (Fig. 3).

1.1 Previous work

Related previous work can be reviewed in the aspects of developable surface modeling, mesh processing for discrete developable surface, and mesh parameterization and flattening. The research work proposed in this project relates to the study of developable surface in differential geometry [3], where the common forms of developable surface are generalized cylinders, conical surfaces (away from the apex), and tangent developable surfaces. The developability of a surface can be characterized by Gaussian curvature which is the product of the maximum and minimum normal curvatures at a given point. In general, a surface is developable if and only if the Gaussian curvature of every point on it is zero. When computing in a discrete form (i.e., piecewise linear surfaces), such a surface is named as flattenable mesh surface that has zero Gaussian curvature on non-boundary

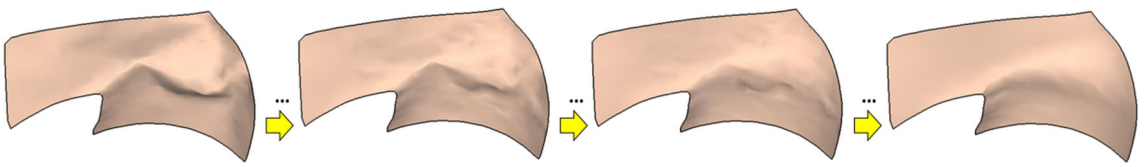

Fig. 3 Progressive results of surface fitting—our surface fitting approach can iteratively morph the shape of a flattenable mesh surface to approximate the shape of an input surface by isometric deformations: (most *left*) the flattenable mesh surface shown in Fig. 2b before fitting, (most *right*) the fitting result has shown in Fig. 2c

vertices [5]. Some prior approaches [7–11] directly construct models by continuous developable ruled surfaces (or ruled surfaces in other representations—e.g., B-Spline or Bèzier surface patches). Although conical dislocations in crumpling have been studied in [12], it is still not convenient to model the freeform surfaces of modern products (e.g., the sports suit shown in Fig. 2) by continuous developable surfaces. Another limitation of these approaches is that they can only model surface patches with 4-sided boundaries as the surfaces are usually defined on a squared parametric domain. Another thread of research reformulates the problem and makes it computable in a discrete form (e.g., by piecewise linear surfaces). Decaudin et al. [13] process a given mesh surface by locally fitting a conical surface at every vertex and then optimizing the positions of vertices so that their normal vectors satisfy the requirement of conical surface. More generally, the mesh surface is directly optimized in [14] to minimize the objective in terms of discrete Gaussian curvature, which is derived from a prior work in [15]. However, as the zero value on discrete Gaussian curvature is formulated as the soft constraints in these approaches, the resultant surfaces are neither really developable (i.e., the Gaussian curvatures are not zero) nor variants from flattenable mesh surfaces with controlled strain. This problem is somewhat solved in [6] which formulates the discrete developablility as hard constraints in a numerical optimization framework. However, the processed flattenable mesh surface M gives a very poor approximation to the input surface P (see Fig. 2), which is a problem to be solved in this research project. Liu et al. in [4] presented a novel PQ mesh, which can be used to model developable surfaces in strips. The discrete developable surface constructed by [4] is still simple—having a shape similar to ruled surfaces. Similar problem occurs in the approach of [16]. A recent interesting work in [17] models a discrete developable surface by fitting a set of 2D quadrilateral pieces to the original surface. The surface generated by this approach can have creases. However, in some applications (e.g., clothes design and manufacturing), the processed flattenable mesh surface M is required to interpolate the boundary of the input surface S. The curved folding method in [17] has difficulty in satisfying this boundary interpolation requirement. When fitting to a highly curved surface P (e.g., the patch shown in Fig. 2), it is difficult to find the ruling directions on P and therefore is hard to conduct the quad dominant decomposition by the algorithm of [17]. Originally motivated by the application of texture mapping, there are many algorithms were developed in the computer graphics community (e.g., [18–22]) to find an optimal mesh parameterization for a given 3D piecewise linear surface. In the computer-aided design area, surface flattening is studied in [23–27] to determine the shape of a planar pattern that can be used to fabricate the 3D surface patches on a designed product. In all of these researches, certain criteria (e.g., angles, areas, and lengths) are employed to evaluate the error between the 3D and the planar surfaces. A more comprehensive review can be found in [28]. However, the given 3D surface patch and the computed 2D pattern can rarely be deformed between each other in an isometric way as the stretching distortion is only minimized but not eliminated. Converting an input surface P into a flattenable mesh surface M while interpolating the surface boundary, ∂P is a better solution to improve the manufacturability of a designed product H containing S.

1.2 Contributions and overview

The major contribution of our work is a novel surface modeling method to generate a slightly stretched flattenable mesh surface from a piecewise linear surface, which presents the following properties.

- **Optimal surface fitting:** The shape approximation error between the flattenable mesh surface M and the input surface patch P is minimized, which is benefitted by our new approach of isometric surface fitting and the novel shape perturbation scheme. The isometric surface fitting optimize the 3D shape of M, and the shape perturbation scheme further enlarge the optimization space by allowing variation on the corresponding planar pattern of M.
- **Constrained stretching strain:** Nearly isometric deformation is conducted to iteratively morph a flattenable mesh surface M into a shape approximating the input surface patch, where the strain of stretching on M is well controlled to ensure resulting a surface that can be fabricated by slightly extensible planar materials.
- **Scalability:** A coarse-to-fine fitting framework is developed in this work to process surfaces with very dense meshes. None of the existing multi-scale mesh editing techniques (e.g., [29–32]) can be directly applied here due to the fact that we need to maintain the flattenability during surface fitting while ensuring the compatibility between boundaries of the refined mesh surface and the input surface.

These properties enrich the geometric modeling ability of flattenable mesh surfaces, which will help shorten the design cycle of products fabricated by slightly extensible planar materials. The rest of this paper is organized as follows. Our method for computing the optimal shape approximation by flattenable mesh surfaces is presented in Section 2. Section 3 gives a multi-scale surface fitting framework which can greatly improve the scalability of this shape approximation algorithm so that the given surface with very dense polygonal meshes can be processed and the boundaries can be interpolated. Experimental results

are given in Section 4 to demonstrate the functionality of our approach, and our paper ends with the conclusion section.

2 Optimal shape approximation

By using the flattenable mesh surface processing method [5], we can process the input mesh surface P into a flattenable mesh surface M by solving a constrained optimization problem as

$$\arg \min_{p \in int(P)} w_1 J_{pos} + w_2 J_{fair} \quad s.t.\ \theta(\mathbf{v}_p) \equiv 2\pi \tag{1}$$

where $int(\cdots)$ is the set of interior vertices on P, $\theta(\mathbf{v}_p) \equiv 2\pi$ is the hard constraint to ensure the flattenability of vertex $\mathbf{v}_p$, J_{pos} is a term attracting $\mathbf{v}_p$ to its position on S and J_{fair} is a term to improve the fairness on surface by using Laplacian operator. In [6], the functional minimization is separated into two steps of the least-norm-based position estimation and the least-square-based surface shape update, where the algorithm is faster than the method in [5]. However, as illustrated in Fig. 2b, the resultant surface from [6] (and also [5]) presents a shape quite different from P. Our isometric surface fitting approach will solve this problem. Stimulated by the method in [33], the shape derivation of M from the input surface P is evaluated by a metric

$$E(M, P) = \frac{1}{A_{total}} \sum_{p \in M} A_{vor}(\mathbf{v}_p) \|\mathbf{v}_p \mathbf{c}_p\| \tag{2}$$

with $\mathbf{c}_p$ being the closest point of $\mathbf{v}_p$ on the input surface P, which can be efficiently evaluated by the method in [34]. $A_{vor}(\mathbf{v}_p)$ denotes the voronoi area of a vertex $\mathbf{v}_p$ (ref. [35]), and A_{total} is the surface area of M. Note that all the colorful map of shape approximation errors in this paper (e.g., Fig.2c) are generated by visualizing the distance between any vertex $\mathbf{v}_p$ on M and the input surface P (i.e., $\|\mathbf{v}_p \mathbf{c}_p\|$).

2.1 Isometric surface fitting

Starting from a flattenable mesh surface M_0 (e.g., the one computed by [6]), we are going to use isometric deformations to iteratively morph its shape into M_i ($i = 0, 1, \ldots$), and then minimize the shape approximation error between M_i and P. Specifically, after each modification, the lengths of all edges on M_i before the deformation should be preserved on the new mesh surface M_{i+1}. The isometric constraint on any edge $\mathbf{v}_i \mathbf{v}_j$ can be defined as (ref. [36])

$$C(\mathbf{v}_i, \mathbf{v}_j) = \|\mathbf{v}_i - \mathbf{v}_j\|^2 / l_{ij} - l_{ij} \equiv 0 \tag{3}$$

with l_{ij} being their undeformed length. To reduce the accumulated round-off errors on the length constraints after several deformations, we store a pair of 3D/2D representations for the flattenable mesh surface M_i under processing—i.e., every vertex on M_i will store its coordinates both in 2D (for the planar shape before fabrication) and 3D (for the current shape). The corresponding planar patch D of a flattenable mesh surface M can be obtained by either using an unfolding algorithm [25] which may accumulate round-off errors during unfolding or using the global method [20] to distribute the round-off errors to the whole patch.

2.1.1 Nearly isometric deformation

Every flattenable mesh surface can be considered as a piece of cloth to be deformed by some forces. In cloth simulation (ref. [36, 37]), these forces include external loadings from gravity, wind flow, collided rigid objects, frictions, etc. and the internal forces of stretch, bend, and shear. When using inextensible cloth simulation [36] as a toolbox to govern isometric deformations on flattenable mesh surfaces, internal forces on the physical model are kept since stretch and shear will help enforce the isometric property and - different bending stiffness actually indicate the levels of expected smoothness on the surface. In the surface fitting scenario of our approach, the external loadings in cloth simulation will be replaced by the forces which drive particles to move towards the input surface P and special treatment must be given to the surface boundary ∂P. - Different from [36] that conducts quadrilateral meshes, we work on triangular meshes here since the length of diagonals in a quadrangle is not constrained if we only add length constraints on polygonal edges as Eq. 3. The constraints on invariant edge lengths are slightly released to provide more flexibility in the surface fitting and mimic the behavior of slightly extensible materials. Specifically, the fast projection loop stops when the strains on all edges are less than 10 %. Applying too strict length preservation constraints here will lead to very poor fitting results. Details will be discussed below.

2.1.2 Surface fitting

The isometric surface fitting is based on the routine of inextensible cloth simulation to preserve the isometric deformation during the surface evolution. The surface fitting procedure is expected to minimize the shape approximation error defined in Eq. 2 on the deformed mesh surface M_i. More than that, the boundary vertices of M_i are demanded to

be coincident to the boundary vertices on the input surface P. Two types of external forces are defined.

$$\mathbf{f}_p^{fit} = w_1((\mathbf{v}_p - \mathbf{c}_p) \cdot \mathbf{n}_{c_p})\mathbf{n}_{c_p} \quad (4)$$

$$\mathbf{f}_b^{bnd} = w_2(\mathbf{v}_b^P - \mathbf{v}_b) \quad (5)$$

$\mathbf{f}_p^{fit}$ is employed for driving an interior vertex $\mathbf{v}_p$ of M_i onto the surface P, where $\mathbf{c}_p$ is the closest point of $\mathbf{v}_p$ on P and $\mathbf{n}_{c_p}$ is the unit normal vector of the surface at $\mathbf{c}_p$. According to the analysis in [38], fitting onto the tangent plane of the closest point (i.e., $\forall \mathbf{x}, (\mathbf{x} - \mathbf{c}_p) \cdot \mathbf{n}_{c_p} = 0$) can effectively avoid the local sticking problem which often happens when directly fitting onto the closest point $\mathbf{c}_p$. $\mathbf{f}_b^{bnd}$ is used for letting a boundary vertex $\mathbf{v}_b$ to be coincident to its corresponding vertex $\mathbf{v}_b^P \in P$. To enforce the boundary interpolation constraints, w_2 should have more weight than w_1. We always choose $w_1 = 0.01$ and $w_2 = 1.0$ in our implementation. Incorporating these external forces into the inextensible cloth simulation routine can efficiently drive the shape of a flattenable mesh surface M_0 into a surface minimizing the shape approximation error defined in Eq. 2. The inner loop of inextensible cloth simulation is an iteration called fast projection, which enforces the strains on polygonal edges under a user specified threshold ε. A smaller value is given for ε, the isometric property is preserved more strictly. However, applying an over-strong enforcement on the extension could become an obstacle to prevent the deformation of mesh surfaces for surface fitting. As demonstrated in Fig. 4, the shape of a flattenable mesh surface cannot be effectively changed by the forces defined in Eq. 4 when a very small threhold is used (e.g., 0.4 %). Increasing the magnitude of the forces or the weight will bring very high stiffness to the numerical system, thereforem, hurt its numerical stability (ref. [37]). We solve this problem by choosing a relative large tolerance (e.g., 10 %) in the fast projection, and the strains on triangular edges are enforces by adding a relaxation step after the surface fitting (details will be given in Section 2.3).

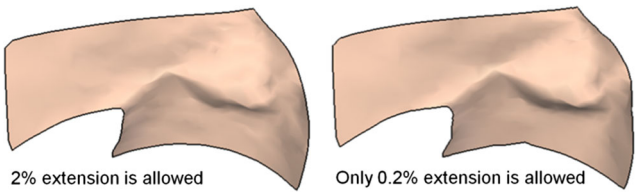

Fig. 4 Applying too strict length-preservation constraints in the isometric surface fitting will prevent the deformation of a flattenable mesh surface towards the shape of input surface. Considering about the patch shown in Fig. 2d where has maximal 10 % extension on edges is allowed during the surface fitting, the surface fitting result becomes poor when reducing the allowed length variation to 4 % (*left*). Further reducing the allowed extension on edges to 0.4 % (*right*) will let the shape almost stuck at the shape before surface fitting (i.e., similar to Fig. 2b)

2.1.3 Problem of numerical singularity

The procedure of using inextensible cloth simulation to deform a flattenable mesh surface by the external forces defined in Eqs. 4 and 5 actually equals to a constrained optimization procedure, where the shape similarity is the objective function and the enforcement of invariant edge lengths is the hard constraints. The solution is found by the method of Lagrange multipliers. The update vector for the multipliers is first determined, and then the update vector for the vertex positions can be obtained. However, when the number of constraints is greater than the *degree-of-freedom* (DOF) on a dynamic mesh surface, the linear system for determining the update vector for multipliers will become singular. Therefore, the computation of inextensible cloth simulation will diverge. Our surface fitting setup introduced above does not have this problem of numerical singularity. Study for the connectivity of a triangular mesh gives the following remark (Proof is given in Appendix).

Remark 1 For a triangular mesh surface having disk-like topology, the number of vertices V (including V_I interior vertices – $V_I > 0$ and V_B boundary vertices) and the number of edges E (including E_I interior edges and E_B interior edges) satisfy (1) $3V > E$ and (2) $3V_I < E_I$.

In our surface fitting, V vertices each has 3-DOF give $3V$-DOF for the dynamic surface. E triangular edges satisfying the invariant length constraints in Eq. 3 lead to E constraints. According to Remark 1, $3V > E$—i.e., the degree-of-freedom is is greater than the number of constraints on the dynamic surface. If we enforce the boundary interpolation in a hard way by substituting the coordinates of boundary vertices on P, the dynamic mesh will only present $3V_I$-DOF but has E_I constraints. Since $3V_I < E_I$, the problem of numerical singularity occurs. Therefore, we do not enforce the boundary interpolation constraints in the surface fitting step as they can be easily solved in the relaxation step for accurate strain control (Section 2.3).

2.2 Shape perturbation for optimal fitting

A tough problem in the above surface evolution mechanism is how to find a good initial guess (i.e., M_0) for the fitting process. The difficulty comes from two aspects:

- Once the shape of the planar patch D is determined by the flattenable mesh surface M_0 computed by [6], the shape of D will not be changed in the 3D surface evolution. This limits the domain of finding an optimal solution.
- Moving vertices for optimization may be prevented by topological obstacles. See Fig. 5 for an example,

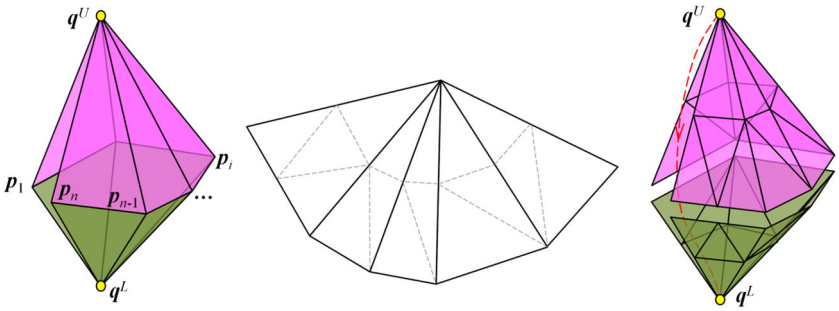

Fig. 5 When isometric deformation is conducted to optimize the shape of a flattenable mesh surface, the computation may not converge because of topological obstacles. (*Left*) If the optimal moving direction of a vertex $\mathbf{q}^U$ is pointing towards $\mathbf{q}^L$ and all $\mathbf{p}_i$s are not allowed to move, the movement of $\mathbf{q}^U$ is prevented since the lengths of edges, $\|\mathbf{q}^U\mathbf{p}_i\|$, should not be changed during isometric deformation. Here, the patches in pink and green are isometric to each other, and they are both isometric to the planar patch (shown in the middle). (*Right*) If the same shape is represented by a finer mesh, the optimization will not be prevented—the red dash curve could be the trajectory of moving $\mathbf{q}^U$ to $\mathbf{q}^L$ during the optimization. The refined mesh in 2D is shown by *gray dash lines* (in the middle)

where the movement from $\mathbf{q}^U$ to the position of $\mathbf{q}^L$ is stuck by the isometric constraints on edges $\mathbf{q}^U\mathbf{p}_i$. This prevents the convergence of computation for surface fitting.

We develop a shape perturbation method to overcome this difficulty of finding a good initial guess. Before applying the method in [6] to convert P into a flattenable mesh surface, we change the shape of P randomly by perturbation

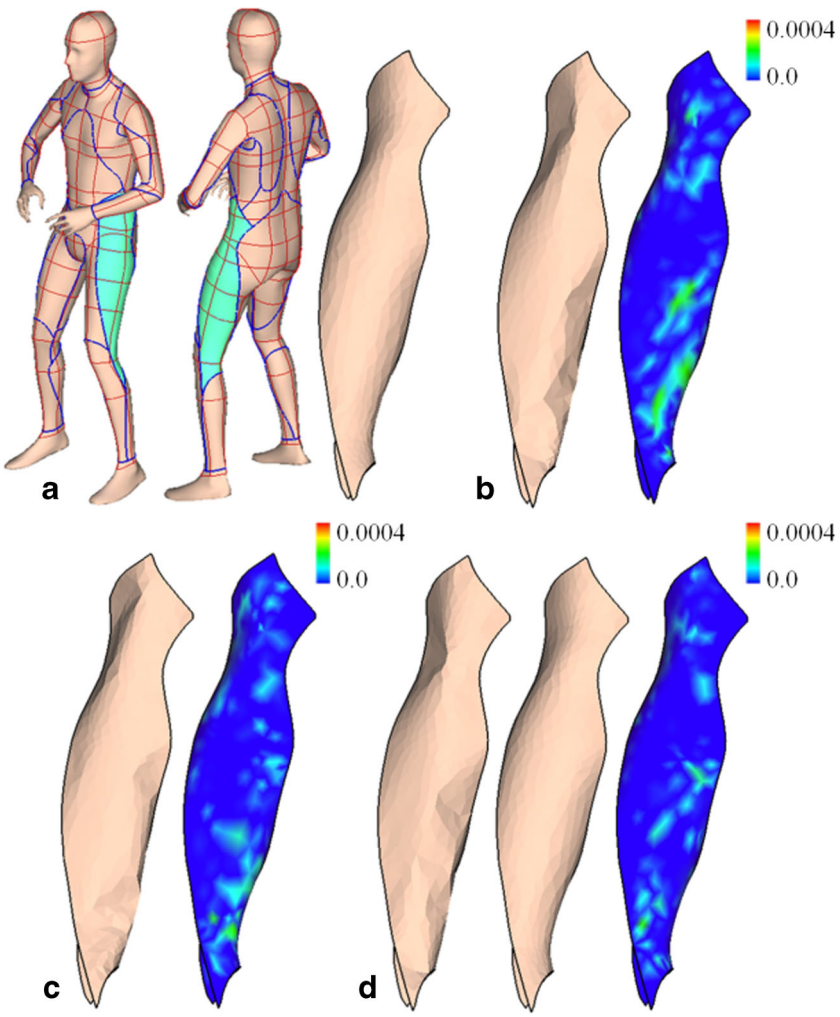

Fig. 6 Different flattenable mesh surfaces varied from the flattenable mesh surface obtained by [6] may give different shape approximation errors on the isometric fitting results: **a** the surface patch P on a user-customized sport-suit (see more details about this application in [1]), **b** The flattenable mesh surface M generated by [6] and its fitting results (the distribution of shape approximation error is visualized by the color mesh), **c** a flattenable mesh surface M^j varied from M via shape perturbation and its fitting result which has a smaller error in $E(M^j, P)$

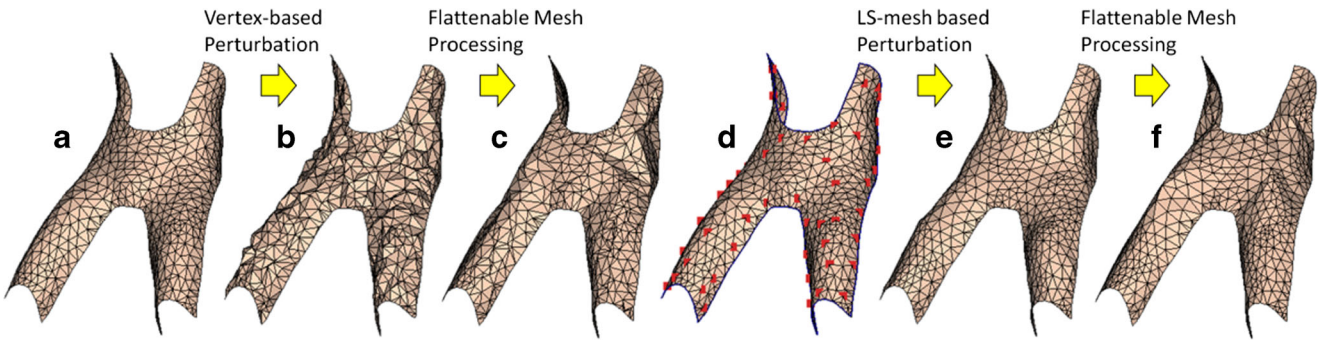

Fig. 7 An illustration of shape perturbation methods: **a** the given mesh surface, **b** the surface generated by applying perturbation on all interior vertices, **c** the result of flattenable mesh processing [6] on the patch in (**b**), **d** the anchor points (in red) are generated by the farthest point sampling [39], **e** the shape perturbation result by the least-square mesh (LS-mesh) based scheme, and **f** the result of flattenable mesh processing on the patch in (**e**). It is easy to find that (**e**) and (**f**) have much more regular meshes than (**b**) and (**c**)

into several variations—i.e., P^1, P^2, …, P^{N^*}, all sharing the same boundary curve, ∂P. Then, they will be converted into m flattenable mesh surfaces M^1, M^2, …, M^{N^*}. After that, all of them will be fit onto the input P via near isometric deformations. m fitting results will be obtained, and the one with the smallest error in $E(M, P)$ (as defined in Eq. 2) is considered as the best result. Figure 6 gives an illustration that different flattenable mesh surfaces varied from M will lead to different shape approximation errors on the fitting results. Simply moving all interior vertices on P at random will result in a mesh surface with many unwanted crumpling regions (see Fig. 7a, b), and the shapes of triangles will become poor (i.e., with caps and needles) which may make the numerical computation on them unstable. Although the flattenable mesh processing techniques in [5, 6] have already integrated the Laplacian term in their formulation, the quality of mesh surfaces after such processing is still poor (see Fig. 7c). In order to solve this problem, we propose a Least-square mesh-based perturbation scheme. Specifically, a few vertices are selected among the interior vertices on S as anchor points (the red ones in Fig. 7d) by the farthest point sampling [39]. Then, the positions of these anchor points are randomly moved with magnitudes smaller than a specified value. The positions of all the remaining vertices are determined by finding the least square solution of a linear system that encodes the relationship between the anchor points and the remaining vertices (details about LS-mesh can be found in [40]). As illustrated in Fig. 7e, f, the shape perturbation conducted in this way gives smoother mesh surfaces and the mesh has better quality. The result shown in Fig. 6c is generated by the LS-mesh-based shape perturbation.

2.3 Relaxation for accurate strain control

As the constraints about maximal strains on the triangular edges have been released to 10 % during the isometric surface fitting, we need an additional relaxation step for

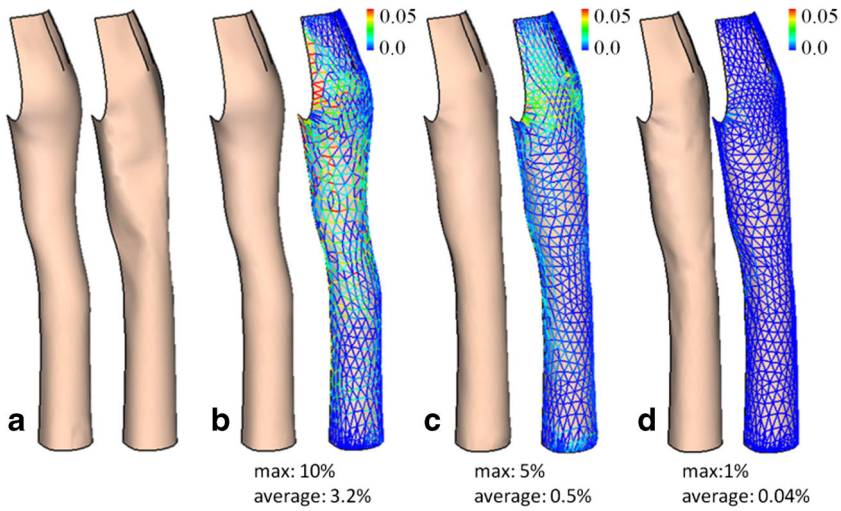

Fig. 8 The relaxation step gives accurate strain control the resultant mesh surfaces: **a** the input surface P to be approximated and the processed result by FL mesh, **b** the nearly flattenable mesh surface after fitting with maximum strain E_{max} at 10 % and average strain E_{ave} at 3.2 %, **c** the mesh surface after the relaxation gives the E_{max} at 5 % and E_{ave} at 0.5 %, and **d** the mesh surface after being processed to reach the E_{max} at 1 % and E_{ave} at 0.04 %. The strains on triangular edges are visualized by colors

achieve more accurate strain control on the resultant mesh. The relaxation step can be defined as an unconstrained optimization

$$\arg \min_{p \in int(M)} \sum (\|\mathbf{v}_i \mathbf{v}_j\| - l_{ij})^2, \tag{6}$$

where $int(\cdots)$ is the set of interior vertices on M, $\mathbf{v}_i \mathbf{v}_j$ is an edge on M, and l_{ij} denotes the length of this edge in the corresponding planar patch D. This unconstrained method can effectively solved by the Newton's method equipped with line search [41]. The Newton iteration of this relaxation step is stopped when the maximum strain of all interior edges on M is less than a user specified threshold. Figure 8 shows the results with different thresholds for the termination of iterations.

3 Multi-scale surface fitting

In this section, the scalability of the above surface fitting algorithm will be improved by a coarse-to-fine shape approximation strategy. A given surface patch P with a very dense polygonal mesh will first be coarsened into a mesh surface P_s with smaller number of vertices (e.g., less than 1000). After that, m flattenable mesh surfaces at the coarse level will be obtained from P_s by the shape perturbation method proposed above. The one, $M*$, having minimal shape approximation error in Eq. 2 will be selected and further refined by splitting every triangle on $M*$ into four triangles, moving the vertices to a position closer to P (for the fitting purpose) while smoothing the mesh surface. To be scalable, local computation schemes are used for fitting and smoothing. The refinement is conducted on both the 3D and 2D representations. However, applying such a refinement step on M_i may result in a mesh surface that is not flattenable; in other words, the 2D and 3D representations of the same mesh are not isometric to each other. A spring-mass system similar to [25] that can be evaluated in a local manner will be conducted to relax the difference between the 2D and 3D lengths of triangular edges on M_i by moving the 3D positions of vertices. These refinement and relaxation steps will be applied repeatedly to the mesh surface until a user specified level is reached (e.g., when having a similar

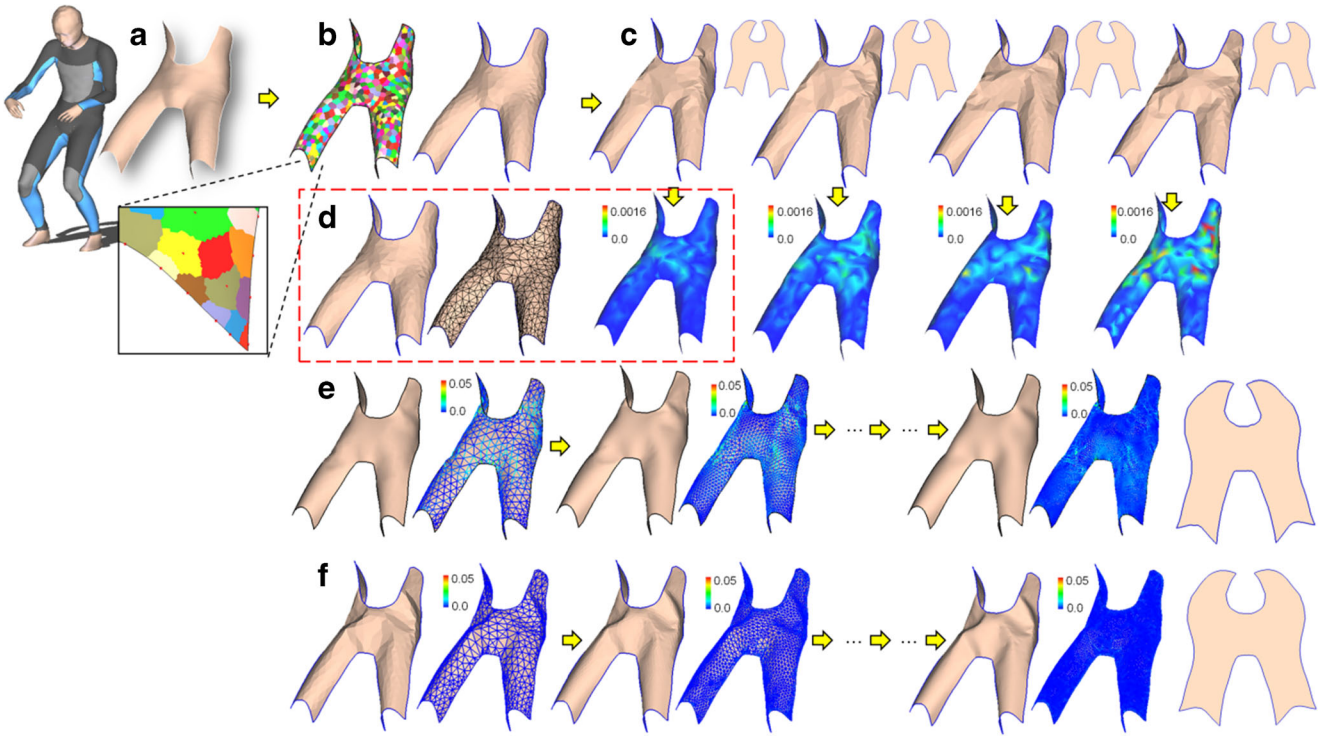

Fig. 9 An illustration of our multi-scale surface fitting framework. **a** The input surface from a user-customized wetsuit has more than 100k vertices, **b** The coarsened mesh surface based on CVD—from the zoom view, we can find that the site points (*red dots*) of voronoi diagram adjacent to boundary are located on the boundary. **c** The examples of flattenable mesh surfaces that are generated by the shape perturbation scheme, where the color maps show the shape approximation error (as defined in Eq. 2) on surface patches. **d** The flattenable mesh surface with the minimal shape approximation error (*circled by the red dash lines*) is selected for the further refinement and update. **e** The refinement is taken on the selected flattenable mesh surface, where the color map shows the strains on edges. **f** For the comparison purpose, results of the variational subdivision scheme of FL mesh proposed in [5] are given. The maximum strain is set to 5 % for the final result

number of vertices compared with that of the given mesh surface P). The maximum strain of resultant mesh surface is controlled below a user specified threshold throughout the whole procedure. In order to interpolate the boundary of P, special refinement schemes will be developed for the triangles on the boundary of the refined mesh surface. An illustration of our multi-scale surface fitting framework is given in Fig. 9.

3.1 Coarsening, local update and relaxation

To obtain simplified surfaces with good mesh quality, *Centroidal Voronoi Diagram* (CVD) [42, 43] will be computed on the given surface P and then the site points of CVD are used as the vertices of P_s. Following the strategy of [42], the CVD can be computed by alternatively applying the two steps of (1) region classification and (2) site selection. After randomly selecting k points on the surface as sites, the region classification step assigns the triangles on the given surface into different voronoi regions. The site selection step updates the position of sites inside their voronoi region. To speed up the computation, a local update scheme suggested in [42] is employed to re-classify the triangles after the sites are moved. The dual triangulation of a CVD on the input surface is the coarsening result. When a surface with dense mesh is given, the coarsening gives very regular triangles as result. Different from [42], we are processing (i.e., remeshing) an open surface. Specifically, we need to locate the sites of some voronoi regions on the boundary of the given surface P to be coarsened, therefore the coarsened triangular mesh P_s has boundary vertices located on ∂P. Such result can be obtained by enforcing only boundary vertices are selected as sites in the regions containing boundary edges. Although this works, a more sophisticated approach is used to improve the quality of flattenable mesh surfaces. Details will be presented in Section 3.2. The refinement procedure is similar to the variational subdivision framework proposed in [44]. For a pair of patches in 3D and 2D (M^j, D^j) at the jth level, a topological splitting operator is first conducted to introduce new vertices to increase the number of degree of freedom by inserting new vertices in the middle of every edges not on the boundary. Note that the M^j and D^j pair shares the same topological graph on their meshes, therefore, their connectivities are always changed together. After the triangle subdivision, the geometry of refined mesh must also be updated by the following steps—all conducted locally.

– *Fitting and smoothing*: For each interior vertex $\mathbf{v}_i$ on M^j, a vector $\mathbf{h}_i = \mathbf{c}_{v_i} - \mathbf{v}_i$ indicates its updating direction with $\mathbf{c}_{v_i}$ being it closest point on M^j. To generate a smooth surface, the update vector $\mathbf{h}_i$ assigned to all interior vertices are filtered by a Laplacian operator

$$\mathbf{h}_i = \frac{1}{|N(\mathbf{v}_i)|} \sum_{\mathbf{v}_k \in N(\mathbf{v}_i)} \mathbf{h}_k \tag{7}$$

for three to five times with $N(\mathbf{v}_i)$ being the 1-ring neighbors of $\mathbf{v}_i$ and $|\cdots|$ getting the number of elements. Then, the positions of all interior vertices are updated by

$$\mathbf{v}_i = \mathbf{v}_i + \frac{w_{fit}}{|N(\mathbf{v}_i)|}\left(\frac{\mathbf{h}_i}{\|\mathbf{h}_i\|} \sum_{\mathbf{v}_k \in N(\mathbf{v}_i)} \|\mathbf{v}_i\mathbf{v}_k\| + \sum_{\mathbf{v}_k \in N(\mathbf{v}_i)} \mathbf{v}_i\mathbf{v}_k\right), \tag{8}$$

which tends to move the vertices closer to the input surface P—therefore, have a better shape approximation.

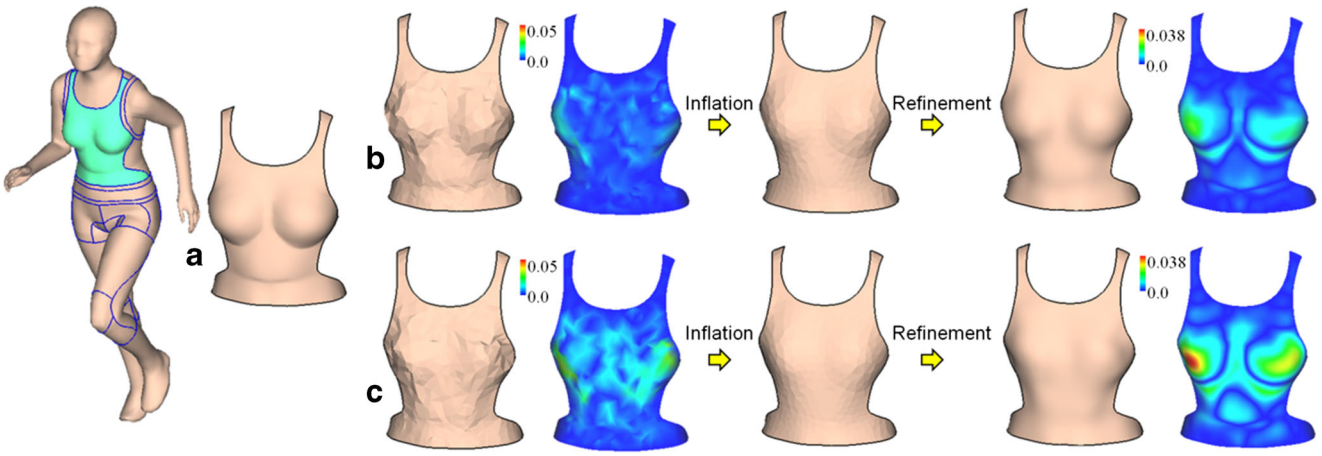

Fig. 10 For the highly curved surfaces, using a V-spring-based inflation can improve the shape of resultant patch in surface fitting. **a** The front piece of a sports vest is highly curved. **b** One flattenable mesh surface patch generated by shape perturbation and its results after repeatedly applied inflation and refinement. **c** Another flattenable mesh surface patch with larger shape approximation error at the coarse level and its corresponding results after inflation and refinement. Ten percent strain is allowed on the fitting result

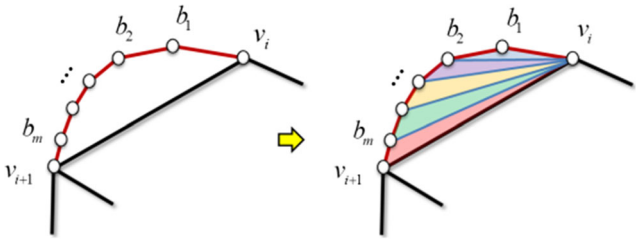

Fig. 11 Gap between boundary edge on the refined mesh surface M_F^n and boundary edges on input mesh surface P, where $\mathbf{v}_i$ and $\mathbf{v}_{i+1}$ are two neighboring vertices on both P and the boundary of M_F^n

$w_{fit} = 0.1e^{-j}$ is a weight to adjust the importance of fitting, which is progressively reduced from 0.1 during the refinement with j increasing.

- *Remeshing* (Optional): After that, the connectivity of M^j and D^j is optimized by applying the area-equalizing remeshing method [30]. The lengths of edges used as criteria for splitting and collapsing are evaluated on the planar patch D^j as the lengths in 3D may retain residual strains. As working on open surfaces with boundary interpolation constraints, the remeshing operators are prevented on the edges containing boundary vertices and the boundary vertices are not moved.
- *Inflation* (Optional): Besides remeshing, we find that for some models with highly curved shape like the one in Fig. 10, the smoothness may be still poor after applying the fitting and smoothing operator defined above. This problem can be solved by the inflation deformation proposed in [45] using V-springs.
- *Relaxation*: After moving the interior vertices in above steps, the distance between two neighboring vertices $\mathbf{v}_i$ and $\mathbf{v}_k$ could be different from their distance l_{ik} in 2D. In other words, the deformation between M^j and D^j is not isometric. To preserve the isometric property, we adopt a spring-mass system similar to [25] to move the interior vertices on M^j by minimizing the energy function

$$\sum_{(\mathbf{v}_i,\mathbf{v}_k)\in M^j} (\|\mathbf{v}_i\mathbf{v}_k\| - l_{ik})^2.$$

To efficiently compute the positions of interior vertices by relaxing the residual strains on edges, the power of parallel computing on the *Graphics Processing Unit* (GPU) is utilized here.

The remeshing and the inflation steps are optional and only applied to the first two to three levels of the refinement.

3.2 Boundary constraints of interpolation

The above refinement and local update operations do not process the boundary edges so that the refined mesh surface does not interpolate the boundary of the input surface P. There are gaps between the refined flattenable mesh surface M_F^n and the boundary of P, which are separated by the vertices on the boundary of M_F^n. Without loss of generality, as shown in Fig. 11, $\mathbf{v}_i$ and $\mathbf{v}_{i+1}$ are two neighboring boundary vertices on M_F^n and there are totally m other vertices, $\mathbf{b}_j$ $(j = 1, ..., m)$, between them on the boundary curve ∂P to be interpolated. A triangulation algorithm [46] is used here to link the boundary curve ∂P between $\mathbf{v}_i$ and $\mathbf{v}_{i+1}$ (i.e., $\{\mathbf{v}_i, \mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_m, \mathbf{v}_i\}$) and the edge $\mathbf{v}_i\mathbf{v}_{i+1}$ by triangles. For given a loop of vertices, $\{\mathbf{v}_i, \mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_m, \mathbf{v}_i\}$ as boundary, there are many possible ways to triangulate them into boundary-triangular mesh surfaces. To select an optimal one among all the possible triangulation, Barequet and Sharir in [46] developed a dynamic programming-based method to obtain a triangulation with minimal value on an particular objective function. As the triangulation may result in faces the normal vectors of which are significantly different from the normal of triangle on M_F^n next to the edge $\mathbf{v}_i\mathbf{v}_{i+1}$ (see Fig.12), we wish to minimize the area of this newly triangulated region. Therefore, a minimal area triangulation is generated by [46] to

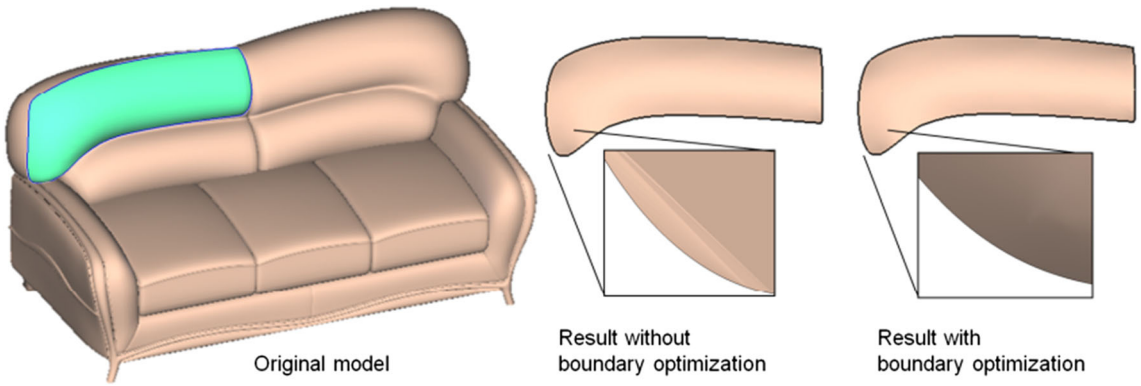

Fig. 12 For a patch from a sofa model (*left*), the final result starting from a model generated by our boundary optimization algorithm (*middle*) gives better smoothness near the boundary compared to the result starting from the one generated by a heuristic method (*right*)

fill the gap. After filling all the gaps, the resultant flattenable surface, M_F^*, interpolating the given boundary ∂P is obtained.

Remark 2 The boundary triangulation that fills the gap between the refined surface M_F^n and the boundary of the given surface M will not change the flattenability of resultant mesh surface M_F^*.

This is because that all the vertices on the newly created triangles are boundary vertices and the triangulation will not convert any boundary vertex to an interior one. The corresponding planar patch, D_F^*, can be obtained by flattening the newly created triangles onto plane and stitching them to the boundary one by one.

3.2.1 Optimal boundary coarsening

The minimal area triangulation can help to reduce the artifact generated on the resultant surface; however, the artifact can be further reduced if we design a special algorithm to generate a coarsened boundary B xby making it as similar as possible to ∂P. Basically, we need an algorithm like the line segment algorithm used in computer vision (ref. [47]). However, the line segments generated here must have their endpoints located on the given boundary curve ∂P. For a boundary curve ∂P with l vertices as $\{\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_l\}$, starting from the longest segment $\mathbf{b}_s\mathbf{b}_{s+1}$, we can iteratively generate a simplified set of line segments for B by the following steps.

- Using $\mathbf{b}_s$ as the starting point of a line segment, a vertex $\mathbf{b}_e$ is to be determined which let $\|\mathbf{b}_s\mathbf{b}_e\|$ as close as possible to $\frac{1}{\sqrt{n}}\|\partial P\|$ and the summed area of triangles

$$A_{s:e} = \sum_{s<i<e} Area(\Delta \mathbf{b}_s\mathbf{b}_i\mathbf{b}_{i+1}) \tag{9}$$

  less than a threshold ϵ. $Area(...)$ gives the area of a triangle and $\|\partial P\|$ denotes the length of surface boundary ∂P. Here, we actually triangulate the line segments $\{\mathbf{b}_s, \mathbf{b}_{s+1}, ..., \mathbf{b}_e\}$ into triangles: $\Delta\mathbf{b}_s\mathbf{b}_{s+1}\mathbf{b}_{s+2}$, $\Delta\mathbf{b}_s\mathbf{b}_{s+2}\mathbf{b}_{s+3}$, ..., $\Delta\mathbf{b}_s\mathbf{b}_{e-1}\mathbf{b}_e$. We could use $\epsilon = 100\bar{A}$ as our threshold with $\bar{A}$ being the average triangle area on M, and n is the number of vertices that is expected on the coarsened mesh surface.
- After determining the vertex $\mathbf{b}_e$ by above method, the line segments is inserted into B as a new edge. Then a new search is started from $\mathbf{b}_e$.
- These steps are repeatedly run until the vertex, from which the computation begins, is reached.

The vertices in B will then serve as site points of voronoi regions at the boundary, and these sites are fixed in the CVD-based coarsening. As a result, we can generate a coarse mesh M^0 by using vertices in B as boundary vertices. The gap filling result on the optimal boundaries can be found in the right of Fig. 12.

4 Results and discussion

We have implemented the proposed algorithm into a prototype program by C++. All the examples presented in this paper are tested on a laptop PC with Intel core i7 2.67GH CPU and 2GB RAM running Windows 7 system. We have tested our approach by several models from various industrial applications. Besides the examples that have been shown in previous Figs. 2, 6, 8, 9, 10, and 12, another test on the shoe design is shown in Fig.13. The computational statistics are listed in Table 1. We compare our method with FL mesh processing [6] in Figs. 2, 8, and 9. Our method shows much better shape approximation in terms of both visual effect and shape approximation error (see color map in Fig. 2). The good scalability is verified on both applying our method directly on models with small number of faces (Figs. 2, 6, and 8), and models with large number faces (Figs. 9, 10, 12, and 13) under the multi-level framework. For different types of materials, different allowed strains are adopted. For materials with high elasticity like nylon and neoprene, we set maximum allowance strain as 5 % (Figs. 6 and 9), 7.5 % (Fig. 2), or even 10 % for some highly curved shape (Fig. 10). For those materials with small elasticity, like textile (Fig. 8) or leather (Figs. 12 and 13), we set only 1 % allowance strain. Table 1 also lists out the average strain which measures the total stretching of the result. The processing time is dominated by the shape perturbation. In shape perturbation, N^* flattenable mesh surfaces are generated. In principle, the greater N^* is the more chance

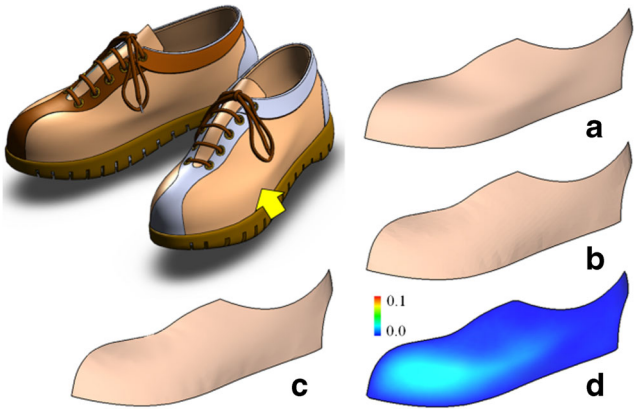

Fig. 13 The result on a shoe product: **a** the designed surface patch, **b** the result at coarse level after fitting and relaxation with 1 % maximum strain, **c** the result at fine level after fitting and relaxation with 1 % maximum strain, and **d** color map to illustrate the shape approximation error on surface shown in (**c**)

Table 1 Computational statistics

model	# Input face	# Result face[a]	Maximum strain	Average strain	Perturbation time (min)[b]	Total time (min)
Fig. 2	1008	–	7.5 %	1.2 %	1.05 (10)	1.55
Fig. 6	1105	–	5 %	0.61 %	1.06 (10)	1.56
Fig. 8	1382	–	1 %	0.04 %	1.4 (10)	3.24
Fig. 9	212k	184k	5 %	0.3 %	1.2 (10)	2.6
Fig. 10	333k	201k	10 %	3 %	12.6 (100)	13.9
Fig. 12	480k	260k	1 %	0.13 %	18.6 (100)	24
Fig. 13	217k	183k	1 %	0.07 %	1.62 (10)	4.62

[a]Some models are processed directly, thus have no change on result face number
[b]The values in the *bracket* indicate the number of models generated by shape perturbation

a optimal fitting result will be obtained and obviously the more time will be taken. We set different m or different tested models. For most of the cases, small N^* (i.e., 10) is good enough, but for some models (Figs. 10 and 12), larger N^* (i.e., 100) is adopted to guarantee an optimal fitting results.

4.1 Discussion

In literature, several existing approaches compute the optimal planar pattern D_P from a given 3D surface P by minimizing the length variation on edges (e.g., [48]) or the non-rigid deformation on triangles (e.g., [22]). When the given surface P is not flattenable, the deformation between D_P and P is not isometric.

Remark 3 There is no guarantee that a planar patch D_P computed from a surface $P \in \Re^3$ by minimizing the distortion between D_P and P can be isometrically warped back into a 3D shape interpolating the boundary of P.

Considering about two points $\mathbf{b}_s$ and $\mathbf{b}_e$ on ∂P, their corresponding points on the boundary of D_P are $\mathbf{d}_s$ and $\mathbf{d}_e$. As the deformation between D_P and P is not isometric, there is no guarantee about the value difference between $\|\mathbf{d}_s\mathbf{d}_e\|$ and $\|\mathbf{b}_s\mathbf{b}_e\|$. When $\|\mathbf{d}_s\mathbf{d}_e\| < \|\mathbf{b}_s\mathbf{b}_e\|$, there is no way to warp D_P to a shape interpolating $\mathbf{b}_s$ and $\mathbf{b}_e$ on ∂P since no distance between two points in a surface can be shorter than the Euclidean distance between them. Therefore, the 3D $\Rightarrow$ 2D flattening approaches cannot be used to solve the flattenable surface fitting problem when the boundary interpolation is required. Different from the surface flattening techniques, we compute the flattenable surface in 3D at coarse level directly by using the nonlinear optimization. However, there is no consensus whether such numerical computation will result in a flattenable mesh surface. Given a triangular mesh M_B interpolating the boundary, ∂P, of a surface patch P with disk-like topology, M_B is called *boundary-triangular mesh* if the vertices of M_B are on ∂P. A surface represented by boundary-triangular mesh is flattenable. This is because that there is no interior vertex on a boundary-triangular mesh surface. For a boundary ∂P with more than three edges, the boundary triangulation is not unique (ref. [49–51]). It is also proved in [51], for the boundary curve with one loop, there always exist a boundary triangulation. We thus have the following remark.

Remark 4 For a given boundary curve with one loop, solutions of the fitting problem by flattenable mesh surface exist.

Any boundary triangulation is a solution for this fitting problem (although may be not an optimal solution). Starting from a boundary-triangular mesh surface M_B, we can morph the surface M_B into other flattenable mesh surfaces interpolating ∂P. The flattenablity of a deformed surface can be guaranteed by preserving the isometric mapping between the dual representation of a surface in both 3D and 2D. All of these flattenable mesh surfaces satisfy the condition that the geodesic distance between any two boundary vertices is greater than the Euclidean distance between them. The solution space of our surface fitting problem is spanned by these deformed flattenable mesh surfaces.

5 Conclusion

We present a new method for computing a slightly stretched flattenable mesh surface M, which approximates the shape of the given mesh surface $P \in \Re^3$. Different from prior approaches that result in either a flattenable surface that could be quite different from the input shape or a (discrete) developable surface has relative simple shape, the techniques investigated in this paper overcome these

difficulties in three aspects. First, a new surface modeling method is introduced to conduct a sequence of nearly isometric deformations to morph a flattenable mesh surface to a new shape which has a better approximation of the input surface. Second, a shape perturbation scheme is investigated to obtain the optimal surface fitting result which can get better initial surfaces for fitting and overcome topological obstacles. Lastly, a coarse-to-fine fitting framework is exploited so that very dense flattenable mesh surfaces can be efficiently modeled and boundaries of the input surfaces can be interpolated. Experimental tests from different industrial applications have been shown to demonstrate the function of techniques developed in this paper.

Acknowledgments The research presented in this paper was partially supported by the Hong Kong Research Grants Council (RGC) General Research Fund (GRF): CUHK/417508, CUHK/417109, Shenzhen Science Plan, Grant No.: JCY J20120903092425971, the NSF CMMI award # 1547134 and the Donald W. Feddersen Professorship from the School of Mechanical Engineering at Purdue University. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views or opinions of the funding agency.

Appendix

The proof of Remark 1 is given below.

Proof For a triangle adjacent to the boundary of a mesh surface, if there is only one boundary edge in the triangle, such boundary edge is defined as type-I; if there are two boundary edges in the triangle, these boundary edges are defined as type II. The number of type I boundary edges is denoted by E_{B1}, and the number of type II boundary edges is E_{B2}. The Euler characteristic for mesh surface with disc-like topology is 1—i.e., $V-E+F=1$. From the definitions of internal edge, type I and type II boundary edges, we have the fact that every internal edge has two faces adjacent to it, every type I boundary edge has one face adjacent, and two type II boundary edges share one face (i.e., each has 1/2 face adjacent). Considering every triangular face has three edges, we get the following edge–face relationship: $2E_I + E_{B1} + \frac{1}{2}E_{B2} = 3F$. Another important relationship is $V_B = E_{B1} + E_{B2}$. Grouping the conditions together, we can derive the following relationship.

$$\begin{cases} V - E + F = 1 \\ E = E_I + E_{B1} + E_{B2} \\ V = V_B + V_I \\ 3F = 2E_I + E_{B1} + \frac{1}{2}E_{B2} \\ V_B = E_{B1} + E_{B2} \end{cases} \Rightarrow 3V_I = E_I - (E_{B1} + \frac{1}{2}E_{B2} - 3) \tag{10}$$

In general, the boundary curve has more than three edges (i.e., $E_{B1} + \frac{1}{2}E_{B2} - 3 > 0$); therefore, we have $3V_I < E_I$. As $V = V_I + V_B$ and $V_B = E_{B1} + E_{B2}$, we can derive the following formulas:

$$\begin{cases} 3V_I = E_I - (E_{B1} + \frac{1}{2}E_{B2} - 3) \\ V_B = E_{B1} + E_{B2} \\ V = V_I + V_B \\ E = E_I + E_{B1} + E_{B2} \end{cases} \Rightarrow 3V = E + (E_{B1} + \frac{3}{2}E_{B2} + 3). \tag{11}$$

As $E_{B1} \geq 0$ and $E_{B2} \geq 0$, we can have $3V > E$ based on the above formula. Therefore, the remark is proved. □

References

1. Wang CCL, Zhang Y, Sheung H (2010) From designing products to fabricating them from planar materials. IEEE Comput Graph Appl 30:74–85
2. Kwok T-H, Yeung K-Y, Wang CC, Kwok T, Yeung K, Wang C (2014) Volumetric template fitting for human body reconstruction from incomplete data. J Manuf Syst 4(33):678–689
3. do Carmo MP (1976) Differential geometry of curves and surfaces. Prentice-Hall, Englewood Cliffs
4. Liu Y, Pottmann H, Wallner J, Yang YL, Wang W (2006) Geometric modeling with conical meshes and developable surfaces. ACM Trans Graph 25:681–689
5. Wang CCL (2008) Towards flattenable mesh surfaces. Comput Aided Des 40:109–122
6. Wang CCL (2008) A least-norm approach to flattenable mesh surface processing. In: Proceedings of IEEE International Conference on Shape Modeling and Applications 2008, pp 131–138
7. Leopoldseder S, Pottmann H (2008) Approximation of developable surfaces with cone spline surfaces. Comput Aided Des 30:571–582
8. Pottmann H, Wallner J (1999) Approximation algorithms for developable surfaces. Computer Aided Geometric Design 16:539–556
9. Chen HY, Lee IK, Leopoldseder S, Pottmann H, Randrup T, Wallner J (1999) On surface approximation using developable surfaces. Graphical Models and Image Processing 61:110–124
10. Pottmann H, Wallner J (2001) Computational line geometry. Springer, Berlin
11. Chu CH, Sequin C (2002) Developable bezier patches: properties and design. Comput Aided Des 34:511–527
12. Cerda E, Chaieb S, Melo F, Mahadevan L (1999) Conical dislocations in crumpling. Nature 401:46–49
13. Decaudin P, Julius D, Wither J, Boissieux L, Sheffer A, Cani MP (2006) Virtual garments: a fully geometric approach for clothing design. Computer Graphics Forum 25:625–634
14. Tang K, Chen M (2009) Quasi-developable mesh surface interpolation via mesh deformation. IEEE Trans Vis Comput Graph 15:518–528
15. Wang CCL, Tang K (2004) Achieving developability of a polygonal surface by minimum deformation: a study of global and local optimization approaches. Vis Comput 20:521–539
16. Bo P, Wang W (2007) Geodesic-controlled developable surfaces for modeling paper bending. Computer Graphics Forum 26:365–374
17. Kilian M, Flöry S, Chen Z, Mitra N, Sheffer A, Pottmann H (2008) Curved folding. ACM Trans Graph 27:75:1–75:9
18. Levy B, Petitjean S, Ray N, Maillot J (2002) Least squares conformal maps for automatic texture atlas generation. In: Proceedings of SIGGRAPH '02, pp 362–371

19. Desbrun M, Meyer M, Alliez P (2002) Intrinsic parameterizations of surface meshes. Computer Graphics Forum 21:209–218
20. Sheffer A, Levy B, Mogilnitsky M, Bogomyakov A (2005) Abf++: fast and robust angle based flattening. ACM Trans Graph 24:311–330
21. Karni Z, Gotsman C, Gortler SJ (2005) Free-boundary linear parameterization of 3d meshes in the presence of constraints. In: Proceedings of Shape Modeling and Applications 2005, pp 268–277
22. Liu L, Zhang L, Xu Y, Gotsman C, Gortler SJ (2008) A local/global approach to mesh parameterization. Computer Graphics Forum 27:1495–1504
23. Azariadis P, Aspragathos N (1997) Design of plane developments of doubly curved surfaces. Comput Aided Des 29:675–685
24. Aono M, Breen DE, Wozny MJ (2001) Modeling methods for the design of 3d broadcloth composite parts. Comput Aided Des 33:989–1007
25. Wang CCL, Smith SSF, Yuen MMF (2002) Surface flattening based on energy model. Comput Aided Des 34:823–833
26. McCartney J, Hinds BK, Chong KW (2005) Pattern flattening for orthotropic materials. Comput Aided Des 37:631–644
27. Wang CCL, Tang K, Yeung BML (2005) Freeform surface flattening based on fitting a woven mesh model. Comput Aided Des 37:799–814
28. Floater MS, Hormann K (2005) Surface parameterization: a tutorial and survey. In: Advances in Multiresolution for Geometric Modelling, pp 157–186
29. Litke N, Levin A, Schröder P (2001) Fitting subdivision surfaces. In: Proceedings of the Conference on Visualization'01, pp 319–324
30. Botsch M, Kobbelt L (2004) A remeshing approach to multiresolution modeling. In: Proceedings of the 2004 Symposium on Geometry processing, pp 185–192
31. Marinov M, Kobbelt L (2004) Optimization techniques for approximation with subdivision surfaces. In: Proceedings of the Ninth ACM Symposium on Solid Modeling and Applications, pp 113-122
32. Shi L, Yu Y, Bell N, Feng WW (2006) A fast multigrid algorithm for mesh deformation. ACM Trans Graph 25:1108–1117
33. Roy M, Foufou S, Truchetet F (2002) Generic attribute deviation metric for assessing mesh simplification algorithm quality. In: Proceedings of the IEEE International Conference on Image Processing, pp 817–820
34. Larsen E, Gottschalk S, Lin MC, Manocha D (2000) Fast proximity queries with swept sphere volumes. In: Proceedings of International Conference on Robotics and Automation, pp 3719–3726
35. Meyer M, Desbrun M, Schröder P, Barr AH (2002) Discrete differential-geometry operators for triangulated 2-manifolds. In: Proceedings of vismath '02
36. Goldenthal R, Harmon D, Fattal R, Bercovier M, Grinspun E (2007) Efficient simulation of inextensible cloth. ACM Trans Graph 26(3). doi:10.1145/1276377.1276438
37. Baraff D, Witkin A (1998) Large steps in cloth simulation. In: Proceedings of SIGGRAPH 98, pp 43–54
38. Liu Y, Pottmann H, Wang W (2006) Constrained 3d shape reconstruction using a combination of surface fitting and registration. Comput Aided Des 38:572–583
39. Moenning C, Dodgson NA (2003) Fast marching farthest point sampling. In: Technical report
40. Sorkine O, Cohen-Or D (2004) Least-squares meshes. In: Proceedings of the Shape Modeling International 2004, pp 191–199
41. Frandsen PE, Jonasson K, Nielsen HB, Tingleff O (2004) Unconstrained optimization online course notes
42. Valette S, Chassery J, Prost R (2008) Generic remeshing of 3d triangular meshes with metric-dependent discrete voronoi diagrams. IEEE Trans Vis Comput Graph 14:369–381
43. Liu Y, Wang W, Levy B, Sun F, Yan D-M, Lu L, Yang C (2009) On centroidal voronoi tessellation—energy smoothness and fast computation. ACM Trans Graph 28 101:1–17
44. Kobbelt L, Schröder P (1998) A multiresolution framework for variational subdivision. ACM Trans Graph 17:209–237
45. Yamada A, Furuhata T, Shimada K, Hou K (1999) A discrete spring model for generating fair curves and surfaces. In: Proceedings of the 7th Pacific Conference on Computer Graphics and Applications, pp 270–279
46. Barequet G, Sharir M (1993) Filling gaps in the boundary of a polyhedron. Computer Aided Geometric Design 12:207–229
47. Sonka M, Hlavac V, Boyle R (1998) Image processing, analysis and machine vision. Chapman & Hall
48. Wang CCL (2008) Flattenable mesh surface fitting on boundary curves. ASME J Comput Inf Sci Eng 8 (2) 021006:1–10
49. Rose K, Sheffer A, Wither J, Cani M-P, Thibert B (2007) Developable surfaces from arbitrary sketched boundaries. In: Proceedings of the fifth Eurographics symposium on Geometry processing, pp 163–172
50. Wang CCL, Tang K (2005) Optimal boundary triangulations of an interpolating ruled surface. ASME J Comput Inf Sci Eng 5(4):291–301
51. Frey WH (2004) Modeling buckled developable surfaces by triangulation. Comput Aided Des 36(4):299–313