WireWarping++: Robust and Flexible Surface Flattening with Length Control

Yunbo Zhang and Charlie C.L. Wang, *Member, IEEE*

***Abstract*— Surface flattening has numerous applications in sheet manufacturing industries, such as garment industry, shoe industry, toy industry, furniture industry and ship industry. Motivated by the requirements of those industries, WireWarping approach presented in [1] is exploited to generate 2D patterns with invariant length of feature and boundary curves. However, strict length constraints on all feature curves sometimes cause large distortions on 2D patterns, especially for those 3D surfaces which are highly non-developable. In this paper, we present a flexible and robust extension of WireWarping by introducing a new type of feature curves named *elastic feature*, which brings flexibility to shape control of the resultant 2D patterns. On these new feature curves, instead of strictly preserving the exact lengths, only the ranges of their lengths are controlled. To achieve this function, a multi-loop shape control optimization framework is proposed to find the optimized 2D shape among all possible flattening results with different length variations on those elastic feature curves, while the lengths of other feature curves are kept unchanged. Besides, we also present a topology processing algorithm on the network of feature curves to eliminate cases that lead to numerical singularity. Experimental results show that the WireWarping++ can successfully flatten surface patches into 2D patterns with more flexible shape control and more robust numerical performance.**

***Note to Practitioners*—The major motivation of this research is to develop a robust and flexible surface flattening technology, which can help sheet manufacturing industries to obtain planar pieces for fabrication of products. For example, in garment industry, after automatically designing a user customized suit in 3D according to the scanned human body (ref. [2], [3]), it is very important to keep the length invariant on the predefined feature curves so as to keep the fitness after sewing the 2D pieces together. Moreover, the boundaries of two neighboring patches should have the same length to prevent unwanted wrinkles when they are sewn together. Other mesh parameterization (and surface flattening) algorithms like [4], [5] and [6] cannot exactly control the lengths of feature curves during the flattening; thus, they are not suitable for the aforementioned applications. WireWarping in [1] can flatten 3D mesh surface into planar pieces while strictly keeping the lengths of all feature curves and boundaries; however, it generates large unstable distortions on those highly non-developable 3D surfaces. The new approach presented in this paper, *WireWarping++*, develops a more flexible flattening technology to obtain better 2D shapes by controlling the length variation in a specified range on the *elastic feature curves*. We have tested the implementation of the WireWarping++ approach on patches of wetsuit sets jeans pants. We have also applied the prototype system to toy and furniture design. In all these applications, WireWarping++ shows significant improvements on the quality of results.**

Manuscript was prepared on May 25, 2010.

All the authors are with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Shatin, NT, Hong Kong. E-mail: cwang@mae.cuhk.edu.hk; Phone: (852) 2609 8052; Fax: (852) 2603 6002.

***Index Terms*— surface flattening, feature curves, length control, shape optimization, multi-loop optimization.**

I. Introduction

THE products in sheet manufacturing industries are fabricated from planar materials. How to determine the corresponding 2D pieces from the designed 3D surfaces is challenging. Ideally, a flattened 2D piece should be an isometric mapping from the 3D surface. However, from differential geometry, we know that only developable surfaces satisfy this property. The shapes of designed products are, however, rarely developable surface. Researchers in the area of surface flattening (or mesh parameterization) usually solve such kind of problems under a constrained optimization framework (e.g., [4], [5], [7]–[19]). Different criteria, such as angle variation, length variation and area variation, are adopted to minimize the difference between the 3D surface and its corresponding 2D piece. In 3D garment industry, it is very important to keep the length invariant on some predefined feature curves on the 3D surface, so as to keep the fitness after sewing the 2D pieces together. Moreover, the boundaries of two neighboring patches should have the same length to prevent unwanted wrinkles when they are sewn together.

WireWarping method in [1] and its least-norm solution presented in [20] give a good solution to such kind of problem by simulating the warping of a given 3D surface into 2D with the boundaries and feature curves serving as tendon. The lengths of boundaries and feature curves can be preserved strictly. To make the shape of a 2D piece similar to its corresponding 3D surface, WireWarping minimizes the variation of surface angles on feature curves during warping. However, when feature curves pass through highly non-developable 3D regions, strictly preventing their lengths causes unstable distortions in 2D (e.g., the quarter-sphere example with two feature curves from [1] – see Fig.1). Moreover, in practical applications, only a few curves among all feature curves whose lengths are quit important for the shape control of final fabricated products and need to be strictly preserved. We denote such feature curves as *rigid feature curves* in the rest of our paper. Except these rigid feature curves, the length of other feature curves can be varied within a controlled range, and these curves are named as *elastic feature curves*. As shown in Fig.1, an improved 2D shape can be obtained by allowing length variations on elastic feature curves using the approach presented in this paper.

During the industrial testing on WireWarping, we found that two special types of topology on the connected feature curves may make the test fail. One case occurs in a network

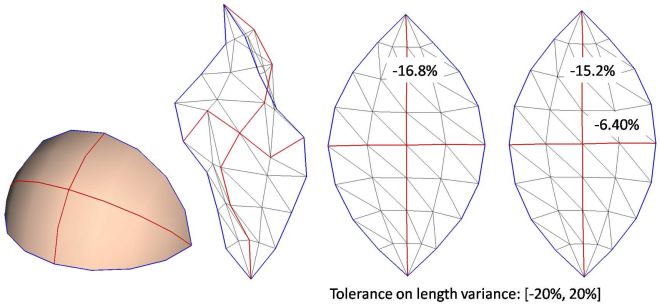


Fig. 1. Surface flattening on a highly non-developable surface – quarter-sphere. (Left) The given mesh model with two orthogonal feature curves defined in red. (Middle-left) The flattening result by WireWarping [1] (i.e., keeping the lengths of two feature curves and the boundaries invariant). (Middle-right) The result obtainted using WireWarping++ approach with the vertical feature curve being *elastic*. (Right) Another result obtainted using WireWarping++ with two elastic feature curves. The sign of the length variance represents the status of shrinkage ($-$) or elongation ($+$). In all these flattening examples, the lengths of boundaries are kept unchanged.

of feature curves with hinged feature curves. As the angle of wire node on the end of such curves is not defined, the numerical system in WireWarping becomes crispy. Another case is that poor flattening results are generated when the 3D surface patch to be flattened has separate boundary loops. These problems are solved by the proposed approach through a topology processing procedure.

The proposed approach is called *WireWarping++*. Comparing it with WireWarping, we improve the flexibility on shape control by introducing the new concept of elastic feature curves with controlled length variation within a range. We exploit a multi-loop optimization framework to find the optimal length variations on the elastic feature curves leading to optimal 2D shapes, while the lengths of rigid feature curves are still strictly preserved. We also propose a topology processing algorithm on the network of features curves embedded in the 3D mesh surface to eliminate the cases that may cause instability to the numerical system.

A. Literature Review

As the fundamental theory of surface flattening, developable surface from differential geometry [21] has been studied for many years. A ruled surface $X(t,v) = \alpha(t) + v\beta(t)$, it is developable if β, $\dot{\beta}$ and $\dot{\alpha}$ are coplanar for all the points on $X(t,v)$. Another characteristic of a developable surface is that the Gaussian curvature of all points on the surface must be zero. Usually a differentiable developable surface belongs to one of the following types: planes, generalized cylinder, conical surfaces (away from the apex), or tangent developable surfaces. Some methods have been investigated based on modeling [22]–[24] or approximating [25]–[27] a model with developable ruled surfaces (or ruled surfaces in other representations, e.g., B-spline or Bézier patches). However, it is impractical to model freeform surfaces using these approaches as they can only model quadrangular surface patches defined on a square parametric domain. Even if trimmed surfaces are considered in [28], to model freeform surfaces, these approaches still have very limited ability.

In the area of surface flattening (ref. [7]–[15]) and mesh parameterization (ref. [4], [5], [15]–[19]), many methods have been developed by researchers in the last two decades. An ideal surface flattening gives an isometric mapping between a 3D surface P and its corresponding 2D piece D. Unfortunately, only developable surfaces have this property when being flattened. Therefore, certain criteria (e.g., angles, areas or lengths) should be adopted to evaluate the error between P and D. A more detailed review can be found in [29]. Recently, Liu et al. in [6] described a local/global algorithm for mesh parameterization which tries to preserve the isometric mapping between 3D surface and 2D planar pieces by a local *as-rigid-as-possible* (ARAP) metric. However, this approach does not address the problem of length control on feature curves.

In literature, researchers have proposed some methods to preserve lengths of feature curves. Manning in [30] introduced an isometric tree consisting of feature curves that are mapped onto the plane isometrically. The main drawback of his work is that the network in [30] must be a tree topology and feature curves can only be branches of the tree. The branch curves are flattened one by one without considering the relationship among them. Another flattening algorithm driven by curves is [31], where Bennis et al. mapped isoparametric curves onto plane followed by a relaxation process to position the surface between them. They also employed a progressive algorithm to process complex surfaces; however, the relationship between these isoparametric curves was not well addressed. Azariadis and Aspragathos [8] proposed a method for optimal geodesic curvature preservation in surface flattening with feature curves. Nevertheless, their method is based on an optimization in terms of vertex positions, which is highly nonlinear and can hardly converge to the expected lengths. Besides, the length preserved curve mapping does also relate to the intrinsic form of curves discussed in [32].

Another relevant thread of research is the modeling of developable (or flattenable) surfaces in 3D instead of computing a surface flattening mapping. The authors in [33] processed a given mesh surface by locally fitting a conical surface at every vertex so that the expected normal vectors can be determined. After that, a deformation process is applied to adjust the position of surface vertices to follow the given normal vectors, but the resultant surface can only be approximately developable. The resultant surface approximates a conical surface locally. More generally, Wang and Tang in [34] adopted discrete definition of Gaussian curvature to define the measurements of the developability on given polygonal mesh surfaces. A constrained optimization approach was conducted to deform mesh surfaces to improve their discrete developability. Liu et al. in [35] presented a novel PQ mesh, which can be used to model developable surface in strips. Recently, Wang presented a FL mesh modeling scheme in [36], which can model developable mesh surfaces with more complicated shapes. Of course, if a given mesh surface P is developable, the length of feature curves is not changed during flattening. However, it is never easy to modify any of these approaches so that they can process a surface from non-developable to developable while preserving the length of feature curves. Another interesting work, described in [37], models a developable surface by

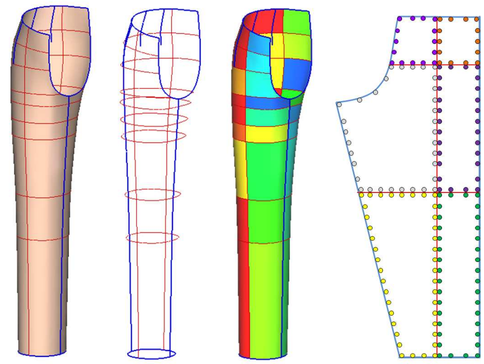


Fig. 2. Illustration of wire-patches: (left) the given piecewise linear surface, (middle-left) the wires, (middle-right) the wire-patches that are displayed in different colors, and (right) wire-nodes on the wire-patches where the wire-nodes belonging to different wire-patches are shown in different colors.

fitting a set of 2D quad pieces to the original surface. Again, this approach can hardly preserve the length of feature curves and has a relatively long computing time.

B. Main Contribution

The main contribution of the work in this paper is summarized as follows.

- A multi-loop surface flattening framework is proposed to optimize the 2D shape of flattened planar pieces while having the ranges of length variation controlled on elastic feature curves and the strictly invariant lengths on rigid feature curves.
- To improve the stability of numerical system in our surface flattening approach, a topology processing algorithm is proposed to eliminate the hinged feature curves and the separate boundary loops on a given 3D surface.

These techniques lead to a new surface flattening approach, *WireWarping++*, the robustness and flexibility of which have been significantly improved. To the best of our knowledge, this is the first surface flattening approach that can provide such kind of robust, flexible and accurate length control.

II. Optimization Based Surface Flattening

In this section, the basic concept of WireWarping is briefed. Then, the multi-loop optimization framework of length-controlled surface flattening is introduced. Afterwards, we present an analysis of shape error function, which is important to the validity and convergence of optimization.

A. Length Preserved WireWarping

For a piecewise linear surface P to be flattened, we define its feature curves by the connected polygonal edges on P. All feature curves together with the boundaries are called *wires* to segment the patch P into several regions called *wire-patches*. For each wire-patch P_i, we record its boundary nodes named as *wire-nodes* shown in Fig.2.

The length preserved WireWarping approach tries to flatten the given surface P onto plane while retaining all the lengths of wires and minimizing the surface angle variation on wires. A constrained optimization as follows is applied to wire-patches,

$$\begin{array}{lll} & min_{\theta_i} \sum_i \frac{1}{2}(\theta_i - \alpha_i)^2 & \\ s.t. & n_p\pi - \sum_{b=1}^{n_p} \theta_{\Gamma_{p(b)}} \equiv 2\pi & (\forall p = 1, ..., m) \\ & \sum_{b=1}^{n_p} l_b \cos\phi_b \equiv 0 & (\forall p = 1, ..., m) \\ & \sum_{b=1}^{n_p} l_b \sin\phi_b \equiv 0 & (\forall p = 1, ..., m) \\ & \sum_{q_k \in v} \theta_k \equiv 2\pi & (\forall v \in \Phi), \end{array} \quad (1)$$

where Φ represents the collection of vertices on feature curves, θ_i is the 2D angle associated with the wire-node q_i, α_i represents q_i's 3D surface angle, and l_b denotes the length of an edge on wires. To simplify the expression, a permutation function $\Gamma_{p(b)}$ is defined for returning the global index of wire-node q_i on the wire-patch P_p with the local index b. The computation is taken in the angle space by setting the angle variations as soft constraints in the objective function. Hard constraints are assigned to prevent self-intersection (the first constraint in Eq.(1)), preserve the closeness of wires on wire-patches (the second and the third constraints), and ensure the compatibility of wire-patches sharing the same vertex (the last constraint in Eq.(1)).

Newton's method is adopted in [1] to solve this constrained optimization problem. However, the magnitude of update vector is not controlled in Newton's method, numerical vibration is easily generated (especially when the variable vector is close to the optimum). In order to make the optimization system more stable, Wang in [20] linearized the constraints in Eq.(1) by changing the variables from θ_i to the angle estimation error $e_i = \theta_i - \beta_i$, where β_i is the current angle on a wire-node q_i and θ_i is the optimal angle to be computed. The constrained optimization is then reformulated into a formula that can be solved by a least-norm solution, which is more numerically robust than conventional Newton's method. Details can be found in [20].

Once the 2D angles on wire-nodes are determined, we can easily locate the vertices on wires by the method in [1]. The positions of the remaining vertices not lying on wires are determined by the shape preserved mesh parameterization method in [38]. The least-norm solution of length preserved WireWarping can robustly and efficiently generate good results in most cases. However, the strict constraints on the invariant length of wires may result in a largely distorted 2D shape. To overcome such a problem, we propose a multi-loop optimization approach to bring flexibility of controlled length variance to those elastic feature curves, thus improving the shape of flattened pieces.

B. Multi-loop Optimization Framework

We present the key component of WireWarping++ here. The surface flattening results are computed under a multi-loop optimization framework. A shape error function is defined in the outer-loop to find an optimal length variation on the elastic feature curves. The selection of shape error function is quite important to numerical stability and convergence in the outer-loop optimization, and, it is discussed in the next subsection.

From the knowledge of differential geometry, only developable surfaces can retain the isometric property between any two points on it. For a given patch P which is far from developable, a length preserved WireWarping flattening may cause a large distortion on the flattened 2D pattern if the feature curves lie on those highly non-developable regions. In this case, the stretch energy can hardly be released as the feature curves are strictly constrained. Other flattening algorithms focusing on minimizing angle error (or surface stretch) can improve the shape in these cases but has no length control on feature curves and boundaries. Therefore, we propose a method to optimize the shape with controlled length variation in a defined range on the elastic feature curves, while still preserving the invariant lengths on rigid feature curves.

We introduce a new variable vector $\mathbf{h}$, where the dimension $n_{\mathbf{h}}$ is the number of elastic feature curves. $h_i \in \mathbf{h}$ represents the ratio of length variation on the i-th elastic feature curve F_i^e. The length variation range for F_i^e is specified by a positive coefficient ϵ_i as $h_i \in [-\epsilon_i, \epsilon_i]$. If we use l_i^0 to denote the original length of feature curves, and l_i to denote the varied length, h_i can be represented as

$$h_i = \frac{l_i - l_i^0}{l_i^0} \quad (\forall i = 1, ..., n),$$

with $-\epsilon_i \leq h_i \leq \epsilon_i$. In other words, we have

$$l_i = (1 + h_i) l_i^0. \tag{2}$$

By a fixed vector $\mathbf{h}$, we can substitute the new lengths, l_is, into the least-norm solution of WireWarping to obtain a 2D pattern according to the varied lengths of elastic feature curves. Note that, the lengths of rigid feature curves are kept invariant. As the flattening result ψ of this approach depends on $\mathbf{h}$, we can consider ψ as a function of $\mathbf{h}$, and use $\psi(\mathbf{h})$ to denote it.

We define a shape error function $E(\psi(\mathbf{h}))$ based on the WireWarping flattening $\psi(\mathbf{h})$.How to find a good flattening now becomes a constrained optimization problem as

$$\begin{array}{c} \min_{\mathbf{h}} \{E(\psi(\mathbf{h}))\} \\ s.t. \; -\epsilon_i \leq h_i \leq \epsilon_i. \end{array} \tag{3}$$

For each inequality, $-\epsilon_i \leq h_i \leq \epsilon_i$, we can rewrite it into two inequalities as $h_i + \epsilon_i \geq 0$ and $-h_i + \epsilon_i \geq 0$. Therefore, we have $2n_{\mathbf{h}}$ inequality constraints in total, and the active set method (ref. [39]) is adopted to introduce them into the optimization. In short, the inequality constraints are partitioned into an active set and an inactive set – only the constraints in the active set are added into the numerical system as equality constraints, which can be solved by quasi-Newton method (ref. [40]).

When using the quasi-Newton method to solve a constrained optimization problem, the objective function $E(\psi(\mathbf{h}))$ should be 2nd order continuous with respect to the variable $\mathbf{h}$. It should also be noted that, the selection of a shape error metric $E(\cdots)$ will affect the optimized 2D shape and the convergence of numerical computation. This problem is discussed in more detail in the next subsection.

In summary, the surface flattening algorithm, which allows controlled length variation on elastic feature curves, is in fact under a multi-loop optimization framework. In the inner-loop, a WireWarping flattening in terms of length variation $\mathbf{h}$ is computed by the least-norm approach [20]. In the outer-loop, a quasi-Newton method iteratively minimizes $E(\mathbf{h})$ under the active constraints, which ensures that the optimized $\mathbf{h}$ is not beyond the tolerance of length variation. The optimization starts from an initial guess $\mathbf{h}^0$, and ends when a minimized $E(\mathbf{h})$ is achieved. In the quasi-Newton approach, an approximate Hessian matrix is updated in each iteration by using the *Broyden Fletcher Goldfarb Shanmo* (BFGS) method. In addition, we conduct a line-search algorithm with constraints as penalty terms [39] to make the outer-loop optimization more numerically stable. Pseudo-code of the optimization algorithm is shown in Algorithm 1.

Algorithm 1 Multi-loop Optimization for Flattening

Initialize $\mathbf{h}^0$ of a given mesh patch;
Compute $E(\psi(\mathbf{h}))$ and $\nabla_{\mathbf{h}} E$, and let the Hessian matrix be I;
Let $\delta_{\mathbf{h}}^0 = -\nabla_{\mathbf{h}} E(\mathbf{h}^0)$, and empty the set of active constraints Ω_{act};
for $i = 1$ to m **do**
  Find the update vector $\delta_{\mathbf{h}}^i$ of $\mathbf{h}$ and the Lagrange multiplier λ_c^i by the quasi-Newton method;
  Using the *penalty line-search* algorithm [39] to find a proper scalar $\alpha \in [0, 1]$ to let $E(\psi(\mathbf{h}^{i-1} + \alpha\delta_{\mathbf{h}}^i)) < E(\psi(\mathbf{h}^{i-1}))$;
  $\mathbf{h}^i = \mathbf{h}^{i-1} + \alpha\delta_{\mathbf{h}}^i$;
  if $\|\delta_{\mathbf{h}}^i\| < \tau_1$ or $\|E(\psi(\mathbf{h}^i)) - E(\psi(\mathbf{h}^{i-1}))\| < \tau_2$ **then**
    return;
  end if
  Update the set of active constraints by checking whether every inequality constraint is satisfied at $\mathbf{h}^i$;
  Update the approximate Hessian matrix by the *BFGS* method;
end for
Get the final 2D pattern by using the length preserved WireWarping with the optimal $\mathbf{h}$;
return;

Note that the active set method cannot completely ensure that the resultant value of h_i falls in the range of $[-\epsilon_i, \epsilon_i]$. Therefore, a post verification step is employed to see if the resultant h_i is in the range. If not, h_i must be projected back into the interval to guarantee length control.

C. Shape Error Function

The shape error function $E(\psi(\mathbf{h}))$ measures the distortion on flattened planar patterns generated by WireWarping with a given length variation $\mathbf{h}$ on the elastic feature curves. Several functions, such as angle variation, edge length variation, area difference and signed area difference, are studied. However, none of them is sensitive enough to the global shape distortion given on the planar pattern compared with the 3D shape of the given surface patch. In other words, the converging speed of optimization is slow.

The best shape error function found in our tests is the *as-rigid-as-possible* (ARAP) metric defined in [6]. The function measures the distortions on all triangles as

$$E = \frac{1}{2} \sum_{t=1}^{T} \sum_{i=0}^{2} \cot(\theta_t^i) \|(\mathbf{u}_t^i - \mathbf{u}_t^{i+1}) - L_t(\mathbf{v}_t^i - \mathbf{v}_t^{i+1})\|^2, \tag{4}$$

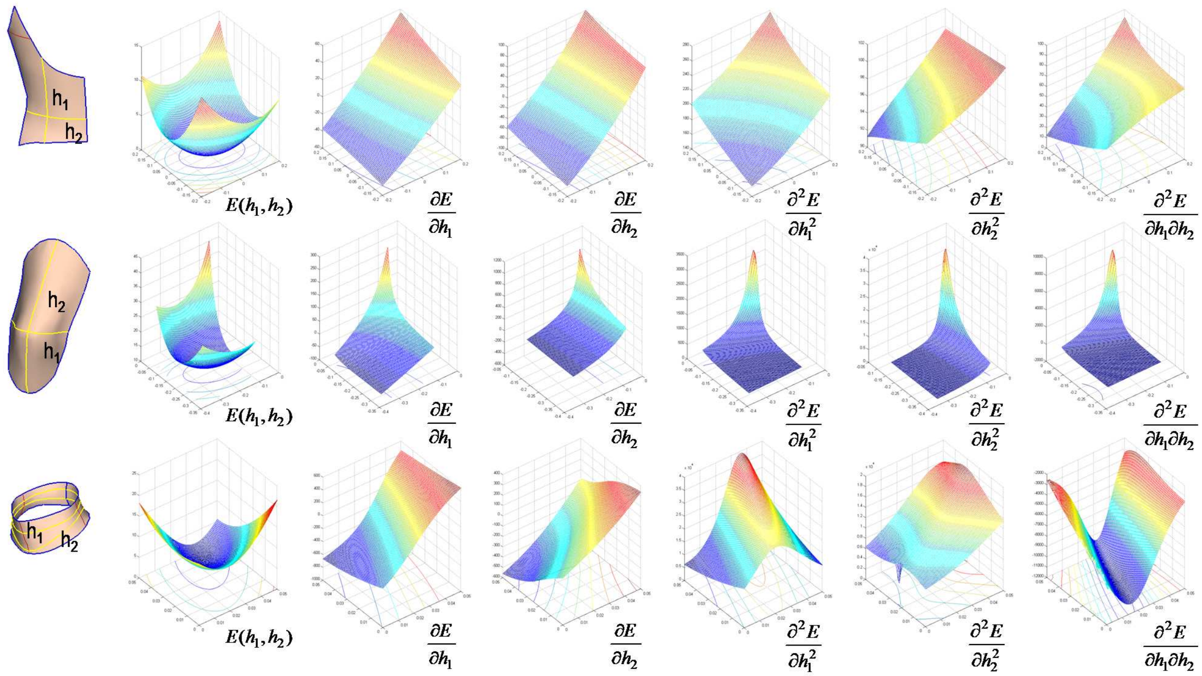


Fig. 3. The continuity study of shape error function E defined in Eq.(4) on different example patches from a wetsuit: (top) a chest patch, (middle) a knee patch, and (bottom) a collar patch, where the chest patch is nearly developable and the remaining two are highly non-developable.

where $\mathbf{v}_t^i$ and $\mathbf{u}_t^i$ are the 3D and the 2D coordinates of the i-th vertex in triangle t, and θ_t^i is the angle opposite to the edge $\mathbf{v}_t^i\mathbf{v}_t^{i+1}$ in 3D. $L_t(\cdots)$ is a rigid transformation matrix to map the triangle t onto plane. Note that, not only the vertices on feature curves but also other vertices are evaluated here.

Another matter of our optimization approach is how to determine the initial guess $\mathbf{h}^0$. We apply the *as-rigid-as-possible* mesh parameterization in [6] to obtain a flattening D for a given patch P. The length variations between D and P on the elastic feature curves are then employed as the initial value $\mathbf{h}^0$.

Now we need to verify the continuity of the shape error function employed in our approach to see if its second order differentiation (in terms of length variation vector $\mathbf{h}$) around the initial guess is continuous. Although it is hard to prove that analytically, the trial tests conducted on all kinds of surface patches verify that the shape error function presented in Eq.(4) satisfies this requirement. The continuity analysis on several examples is given in Fig.3. To conclude, the function in Eq.(4) reflects the shape error quite well and is suitable for the optimization.

III. Topology Processing

Using WireWarping++ method to flatten mesh surfaces has certain requirements on the topology of the network of wires (i.e., the feature curves and boundaries). There are two types of topology that make the numerical system of WireWarping++ unstable: 1) the network with hinged feature curves and 2) the patch with separate boundary loops. To enable flattening of surface patches with such a topology, we develop the following two algorithms to process the topology of feature curves network Υ.

A. Processing on Hinged Feature Curves

The definition of hinged wires is first given below, and then the processing method used to eliminate them is presented.

Definition 1 For a feature curve F defined in the network Υ, if any portion of the curve has its left and right regions belonging to the same wire-patch, such a portion of F is defined as a *hinged wire*.

Vertices located on the hinged wires are called *hinged wire-nodes*. When using WireWarping++ to compute flattened surfaces, the surface angles on wire-nodes formed by adjacent edges on the wires are computed in the inner loop of optimization. However, such surface angles are not defined on the hinged wire-nodes, thus crashing the numerical solver. Note that hinged wires are different from darts, which are parts of boundaries of the given patch P and have well defined surface angle. Hinged wires must be processed to vanish the hinged wire-nodes.

Definition 2 For the network of feature curves Υ defined on a given surface patch P, if there is a vertex $\mathbf{v} \in P$ that has only one adjacent edge on Υ, the vertex $\mathbf{v} \in P$ is defined as a *tail node*.

The elimination of hinged wires can be achieved by extending the hinged wires starting from the tail nodes. The steps of topology processing on hinged features are detailed as follows.

1) Firstly, we detect all the tail nodes on the network of feature curves, Υ. To eliminate the hinged wires, we need to find a surface curve path, which starts from the tail nodes and finally intersects another feature curve or boundary. The regions on the left and right sides of a hinged wire are then separated into two wire-patches.

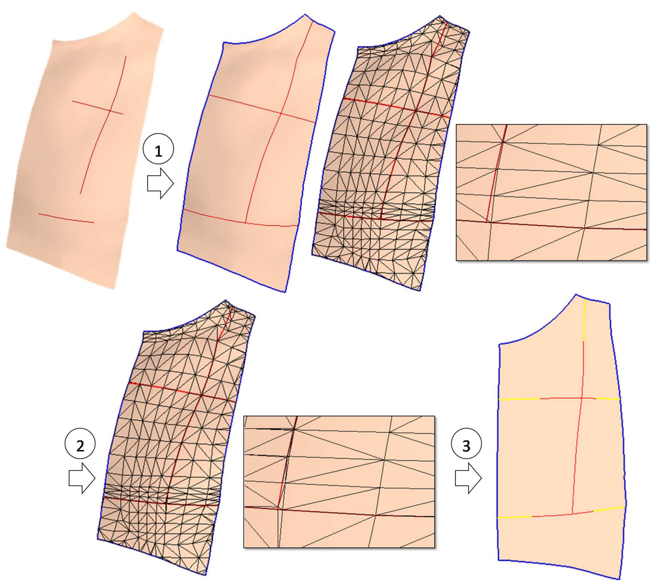


Fig. 4. Topology processing on hinged feature curves. Step 1: extending the hinged curves along their endpoint tangent directions by discrete surface geodesic curves. Step 2: apply the CDT to convert surface curves into triangle's edges. Step 3: flatten the surface by WireWarping++ with the newly added feature curves (in yellow) being "super" elastic.

The surface path, that extends the hinged feature curve along its tangent vector at the tail node and goes along the geodesic direction on P, is a good choice. For a piecewise linear surface P, the piecewise linear geodesic curve on it along a given direction can be incrementally computed by ensuring that the curve always has the equal left and right surface angles on P. Details can be found in [41] and [42]. Fig.4 shows an example of such an extension.

2) Secondly, we need to modify the topology of the original mesh surface according to the extended geodesic curve paths found in the previous step. A *Constrained Delaunay Triangulation* (CDT) is employed to carry out the triangulation to make the newly added surface curves into triangle edges (see step 2 in Fig.4).
3) Finally, we set every newly added feature curve F_j as a super elastic feature curve with uncontrolled variant length. The range of length variation h_j is not controlled by the range $[-\epsilon_j, \epsilon_j]$ any more. Instead, we simply set $1 + h_j > 0$ to ensure that the length of an elastic feature curve is positive.

After fixing all hinged wires, the WireWarping++ can be applied to flatten the surface patch. The flattening result of a patch after fixing all hinged wires is shown in Fig.4.

B. Connecting Separate Boundary Loops

WireWarping++ method works well on surfaces with a disk-like topology. However, it has problems when flattening a surface with separated boundary loops.

Definition 3 A boundary loop of a surface patch P is a set of connected boundary vertices where each pair of neighboring vertices is connected by a boundary edge.

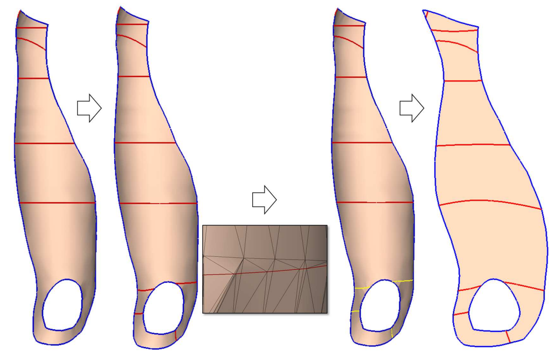


Fig. 5. Topology processing on separate boundary loops, where the newly added feature curves are set as "super" elastic.

Definition 4 Two boundary loops are defined as separate if there is no feature curve linking them.

As there is no feature curve (i.e., wire) linking them, the two separate boundary loops (wires) are decoupled in the numerical system of WireWarping, which leads to a degenerate result. We develop an automatic connecting method which is akin to the virtual cutting scheme proposed in [43]. The algorithm consists of three steps:

1) After detecting all separate boundary loops, we select one with no feature curve connected, L_j. The shortest path from the vertices on L_j to the vertices on other boundary loops passing through the edges of triangles is determined by the multi-source Dijkstra's algorithm [44]. The zipzag shortest path is further smoothed by an iterative refinement procedure to approximate a geodesic curve [41].
2) If the above curve starts from $\mathbf{v}_s$ on L_j, we find the farthest vertex from $\mathbf{v}_s$ on L_j, $\mathbf{v}_f$. Then, the shortest path from $\mathbf{v}_f$ to the boundary vertices on other boundary loops is generated by the single source Dijkstra's algorithm [44] and refined by [41]. To get better results, two more such curves can be added starting from the vertices between $\mathbf{v}_s$-$\mathbf{v}_f$ or those between $\mathbf{v}_f$-$\mathbf{v}_s$. Therefore, of total of four curves are added.
3) Repeat the above two steps until all boundary loops join with more than one feature curves. After that, the newly added feature curves are triangulated into edges using CDT, and are assigned as super elastic feature curves.

When applying WireWarping++ algorithm to such processed surface patches with separate boundary loops, the satisfactory flattening results can be obtained. Figure 5 shows an example of flattening a surface patch with separate boundary loops.

IV. Experimental Results and Applications

We have implemented the proposed algorithm into a prototype program by C++. SuperLU [45] wrapped by OpenNL is conducted as the numerical computation kernel. All the examples presented in this paper are tested on a PC with Intel 2.4GHz Quad-Core CPU and 2GB RAM running Windows Vista operating system.

A. Metrics

Several metrics are adopted to verify the distortion of flattening results.

Edge-Length Variation The length variation of each edge e on the feature curve is measured by

$$E_L = \frac{\|l_e^0 - l_e\|}{l_e^0}, \tag{5}$$

where l_e^0 is the length of the edge e in 3D, and l_e is its length in 2D.

Global Aspect Ratio The aspect ratio E_r is proposed by Azariadis and Sapidis in [46] to measure the distortion on the results of surface flattening. The ideal value of E_r is one, which is shown only on isometric mappings. In our test results, we display the color map of the aspect ratio on every face.

ARAP Energy We also compute the as-rigid-as-possible energy defined in [6] as

$$E_A = \frac{1}{2}\sum_{i=0}^{2} \cot(\theta_t^i)\|(\mathbf{u}_t^i - \mathbf{u}_t^{i+1}) - L_t(\mathbf{v}_t^i - \mathbf{v}_t^{i+1})\|^2, \tag{6}$$

on every triangle face. The value of this function measures the stretch from 3D surface to 2D pattern. We also display a color map in terms of E_A to illustrate the distortion per face in our results.

B. Experimental Results

The first example tested here is a model of jeans pants from the garment industry. After designing a 3D model for a user as the shape shown in the top-left corner of Fig.6, the corresponding 2D patterns for fabrication need to be computed. Our results are compared with those generated by the state-of-the-art in literature (i.e., the results generated by the angle based flattening (ABF++) [5], the as-rigid-as-possible parameterization (ARAP) [6], and WireWarping [1]). Note that ABF++ does not preserve the scale of a flattening, so we scale the flattening result by the longest edge length in 3D. From Fig.6, it is obvious that ABF++ and ARAP cannot preserve the lengths on feature curves and boundaries. This is a significant disadvantage when applying them in the sheet manufacturing industries. Although the WireWarping method can preserve the lengths on all feature curves and boundaries, it gives large distortions on some triangles (see the color maps of E_r and E_A in the second row of Fig.6). The WireWarping++ approach proposed in this paper gives superior results than WireWarping on the aspect ratio E_r and the as-rigid-as-possible stretch metric E_A. The shape of the jeans pants fabricated from the patterns generated by WireWarping++ is also better (see Figs.6 and 7).

The second example is a collar pattern of a wetsuit, the shape of which is like the skin of a human body. Therefore, it is highly non-developable. The flattening result generated by the WireWarping approach [1], [20] gives large distortions with ”S” shape boundaries (see the first row of Fig.8). This is unacceptable by the fashion industry as it will generate many unwanted bumps on the wetsuit fabricated from such a planar pattern. After specifying two feature curves as elastic feature curves, we first flatten the 3D collar patch with a length variation tolerance $\pm 3.0\%$. The resultant 2D pattern shown in the second row of Fig.8 is much better. In addition, the color maps of the aspect ratio E_r and the as-rigid-as-possible energy E_A also verify the improvement of flattening results on distortion. If we further broaden the allowed length variation range from $\pm 3.0\%$ to $\pm 5.0\%$, the flattening result will be even better (see the last row of Fig.8). The actual length variations on the two elastic feature curves are 3.4% and 4.3% respectively.

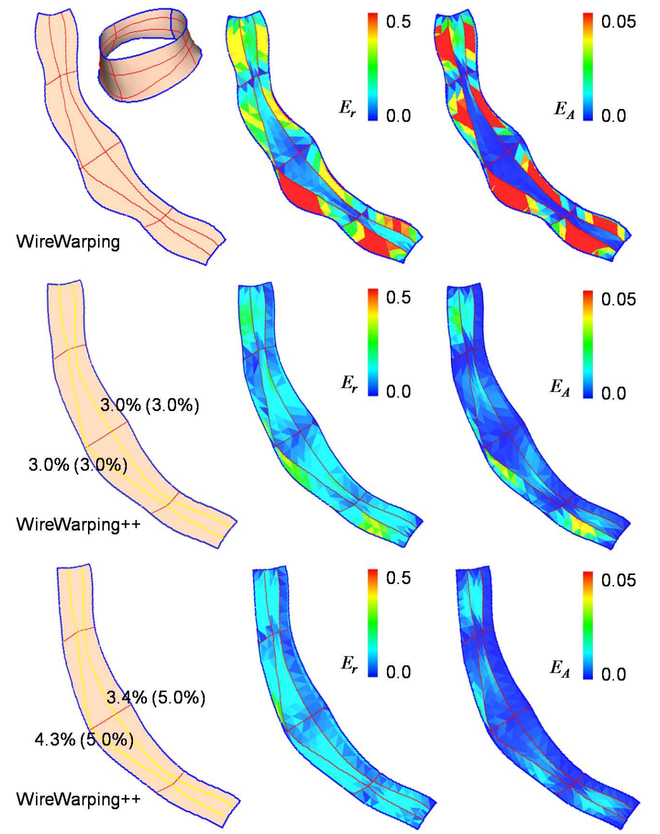


Fig. 8. Surface flattening of a collar pattern of a wetsuit: (top row) the result obtained from WireWarping, the color maps of the aspect ratio E_r and the as-rigid-as-possible energy E_A; (middle row) the result obtained from WireWarping++ with a length variation tolerance of 3.0%; and (bottom row) the flattening result generated by WireWarping++ with a length variation tolerance of 5.0%.

We also test the approach on an upper body of a wetsuit (see Fig.9). After selecting a few feature curves as the elastic ones with tolerance $\pm 5.0\%$, the flattened 2D patterns with a low aspect ratio error E_r and low as-rigid-as-possible energy E_A are generated by the WireWarping++ method proposed in this paper.

A computer-aided design system for modeling high quality user customized wetsuit has been developed. Surface flattening is one of the most important functions in the system without which the final patterns used for fabrication can never be computed accurately. The length control on feature curves and boundaries has been proved to be a very good method to control the quality of the final fabricated wetsuit according

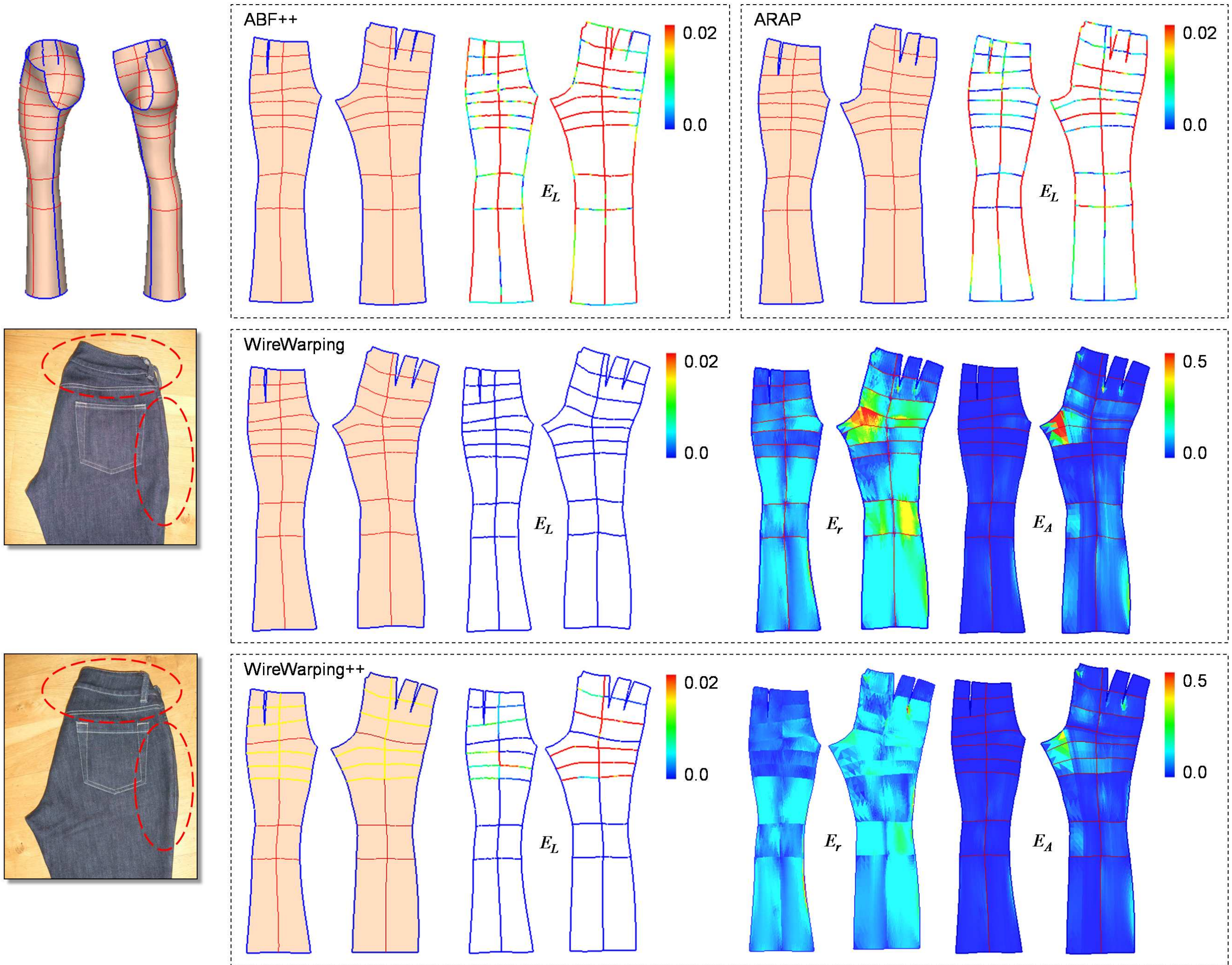


Fig. 6. Surface flattening of a pair of jeans pants. (Top row) The left-most one is the 3D given surface with feature lines and cutting lines (boundaries) defined. The flattening results obtained from the angle based flattening approach (ABF++ in [5]) is shown in the middle, and the result generated by the as-rigid-as-possible (ARAP) parameterization [6] is shown in the right. The colors on the curves indicate the length variation on the feature curves and boundaries. (Middle row) The surface flattening results generated by WireWarping [1], where the lengths on all feature curves are invariant. The color maps show the distribution of the aspect ratio E_r (in the left) and the as-rigid-as-possible energy E_A on all triangles. (Bottom row) The resultant patterns generated by WireWarping++ are shown, where length variations only occur on the specified elastic feature curves (in yellow), and both the aspect ratio E_r and the as-rigid-as-possible energy E_A on the 2D patterns are reduced compared with the results obtained from WireWarping. The maximum length variation range is set to 5.0%. The jeans pants fabricated from the 2D patterns generated by WireWarping and WireWarping++ are also shown in the middle and the bottom rows respectively.

to its 3D design. The interface of our CAD system and a fabricated wetsuit are shown in Fig.10.

C. Other Applications

Besides the garment industry, the technique proposed in this paper in fact can also be applied to other industries as long as their products are fabricated by assembling 2D pieces into a 3D shape (e.g., fabric toys and furniture covered by leather). Examples of using our approach in these industries are shown in Fig.11.

V. Conclusion

In this paper, we present a surface flattening technology, WireWarping++, with a flexible and robust length control. A new type of feature curves named *elastic feature curves* are introduced to achieve the flexibility of shape control. To obtain the optimal 2D shape with length variation on elastic feature curves, we propose a multi-loop optimization frame. In the inner loop, the 3D surface is flattened by a least-norm WireWarping with a certain length variation, while the outer loop minimizes a shape error function to estimate the shape error of each flattening. To control the length variation, we specify a maximum length variation range on each elastic feature curve, and set it as a constraint in the outer loop of optimization. Compared with the original WireWarping approach in [1], the 2D shape of the flattened patches has significant improvement. To improve numerical stability of WireWarping++, we conduct topology processing on the net-

Fig. 7. Photos of the jeans pants fabricated from the patterns generated by WireWarping versus those generated by WireWarping++. The one generated by WireWarping++ fits the back waist band and the back yoke much better – this is the comment made by a fashion specialist.

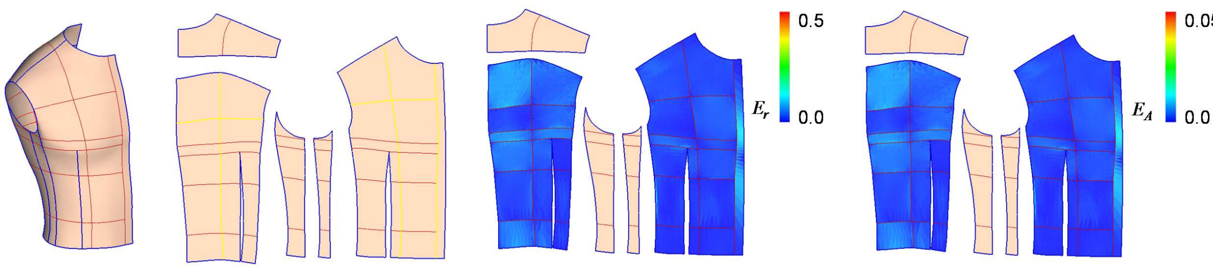


Fig. 9. An example of flattening an upper body of a wetsuit. (From the left to right) The 3D surface of an upper body with feature curves and cutting curves defined, the flattening results, the color maps of the aspect ratio E_r, and the color maps of the ARAP energy E_A

work of feature curves to eliminate hinged feature curves and we add super elastic feature curves to connect separate boundary loops. The experimental results in this paper verify the performance of this new approach. Our future research focuses on how to conduct the length control on feature curves to generate patterns for compression garments.

Acknowledgment

This work was partially supported by the Hong Kong Innovation and Technology Fund (ITF) ITS/026/07 and Shun Hing Institute of Advanced Engineering (SHIAE) Research Grant (CUHK/8115022), and the authors would like to thank TPC (HK) Ltd. for providing some testing data.

References

[1] C.C.L. Wang, “WireWarping: A fast surface flattening approach with length-preserved feature curves,” *Computer-Aided Design*, vol. 40, no. 3, pp. 381–395, 2008.
[2] C.C.L. Wang, Y. Wang, and M.M.F. Yuen, “Design automation for customized apparel products,” *Computer-Aided Design*, vol. 37, no. 7, pp. 675–691, 2005.
[3] C.C.L. Wang, K.C. Hui, and K.M. Tong, “Volume parameterization for design automation of customized free-form products,” *IEEE Transactions on Automation Science and Engineering*, vol. 4, no. 1, pp. 11–21, 2007.
[4] B. Lévy, S. Petitjean, N. Ray, and J. Maillot, “Least squares conformal maps for automatic texture atlas generation,” in *SIGGRAPH ’02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 2002, pp. 362–371, ACM.
[5] 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.
[6] L. Liu, L. Zhang, Y. Xu, C. Gotsman, and S.J. Gortler, “A local/global approach to mesh parameterization,” *Eurographics Symposium on Geometry Processing*, 2008.
[7] P. Azariadis and N. Aspragathos, “Design of plane developments of doubly curved surfaces,” *Computer-Aided Design*, vol. 29, no. 10, pp. 675–685, 1997.
[8] P.N. Azariadis and N.A. Aspragathos, “Geodesic curvature preservation in surface flattening through constrained global optimization,” *Computer-Aided Design*, vol. 33, no. 8, pp. 581–591, 2001.
[9] M. Aono, D.E. Breen, and M.J. Wozny, “Modeling methods for the design of 3d broadcloth composite parts,” *Computer-Aided Design*, vol. 33, no. 13, pp. 989–1007, 2001.
[10] M. Aono, D.E. Breen, and M.J. Wozny, “Fitting a woven-cloth model to a curved surface: mapping algorithms,” *Computer-Aided Design*, vol. 26, no. 4, pp. 278–292, 1994.
[11] J. McCartney, B. K. Hinds, and B. L. Seow, “Flattening of triangulated surfaces incorporating darts and gussets,” *Computer-Aided Design*, vol. 31, no. 4, pp. 249–260, 1999.
[12] J. McCartney, B.K. Hinds, and K.W. Chong, “Pattern flattening for orthotropic materials,” *Computer-Aided Design*, vol. 37, no. 6, pp. 631–644, 2005.
[13] C.C.L. Wang, S. Smith, and M.M.F. Yuen, “Surface flattening based on energy model,” *Computer-Aided Design*, vol. 34, 2002.
[14] C.C.L. Wang, K. Tang, and B.M.L. Yeung, “Freeform surface flattening based on fitting a woven mesh model,” *Computer-Aided Design*, vol. 37, no. 8, pp. 799–814, 2005.
[15] Y. Zhong and B. Xu, “A physically based method for triangulated surface flattening,” *Computer-Aided Design*, vol. 38, no. 10, pp. 1062–1073, 2006.
[16] M. Desbrun, M. Meyer, and P. Alliez, “Intrinsic parameterizations of surface meshes,” *Computer Graphics Forum*, vol. 21, no. 3, pp. 209–218, 2002.

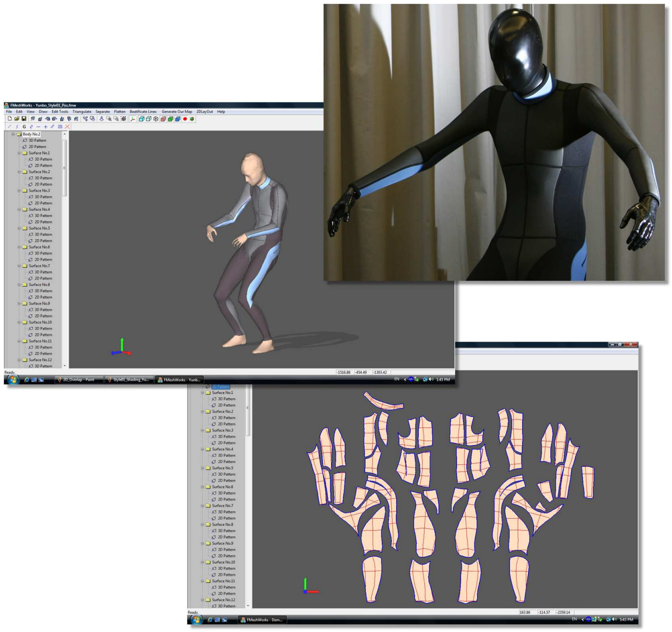


Fig. 10. The interface of our CAD system for wetsuit design and the final wetsuit fabricated from the patterns generated by this system.

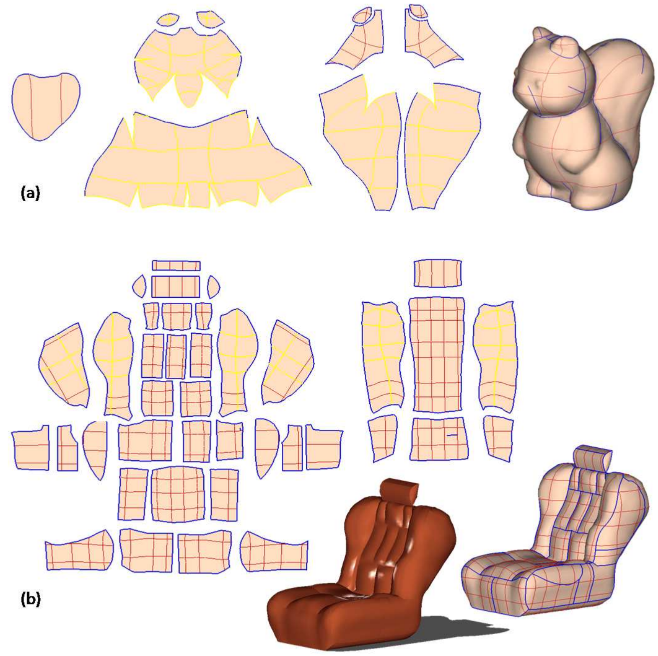


Fig. 11. The applications of our technique in (a) toy industry and (b) furniture industry. Again, the yellow feature curves are assigned as elastic.

[17] Z. Karni, C. Gotsman, and S.J. Gortler, "Free-boundary linear parameterization of 3d meshes in the presence of constraints," in *International Conference on Shape Modeling and Applications*. 2005, pp. 268–277, IEEE Computer Society.

[18] Y. Lee, H. S. Kim, and S. Lee, "Mesh parameterization with a virtual boundary," *Computers and Graphics (Pergamon)*, vol. 26, no. 5, pp. 677–686, 2002.

[19] A. Sheffer and E. De Sturler, "Parameterization of faceted surfaces for meshing using angle-based flattening," *Engineering with Computers*, vol. 17, no. 3, pp. 326–337, 2001.

[20] C.C.L. Wang, "A note on least-norm solution of global wirewarping," *Computer-Aided Design*, vol. 41, no. 9, pp. 695–698, 2009.

[21] M.P. do Carmo, *Differential Geometry of Curves and Surfaces*, Prentice-Hall, Englewood Cliffs, NJ., 1976.

[22] S. Leopoldseder and H. Pottmann, "Approximation of developable surfaces with cone spline surfaces," *Computer-Aided Design*, vol. 30, no. 7, pp. 571–582, 1998.

[23] H. Pottmann and J. Wallner, "Approximation algorithms for developable surfaces," *Computer Aided Geometric Design*, vol. 16, no. 6, pp. 539–556, 1999.

[24] C.H. Chu and C.H. Séquin, "Developable bézier patches: Properties and design," *Computer-Aided Design*, vol. 34, no. 7, pp. 511–527, 2002.

[25] H.-Y. Chen, I.-K. Lee, S. Leopoldseder, H. Pottmann, T. Randrup, and J. Wallner, "On surface approximation using developable surfaces," *Graphical Models and Image Processing*, vol. 61, no. 2, pp. 110–124, 1999.

[26] M. Peternell and T. Steiner, "Reconstruction of piecewise planar objects from point clouds," *Computer-Aided Design*, vol. 36, no. 4, pp. 333–342, 2004.

[27] M. Peternell, "Recognition and reconstruction of developable surfaces from point clouds," in *GMP '04: Proceedings of the Geometric Modeling and Processing 2004*, Washington, DC, USA, 2004, p. 301, IEEE Computer Society.

[28] C.C.L. Wang, Y. Wang, and M.M.F. Yuen, "On increasing the developability of a trimmed nurbs surface," *Engineering with Computers*, vol. 20, no. 1, pp. 54–64, 2004.

[29] M.S. Floater and K. Hormann, "Surface parameterization: A tutorial and survey," *Advances in Multiresolution for Geometric Modelling*, pp. 157–186, 2005.

[30] J. Manning, "Computerized pattern cutting methods based on an isometric tree," *Computer-Aided Design*, vol. 12, no. 1, pp. 43–47, 1980.

[31] C. Bennis, J.M. Vezien, and G. Iglesias, "Piecewise surface flattening for non-distorted texture mapping," *Computer Graphics*, vol. 25, no. 4, pp. 237–246, 1991.

[32] A. W. Nutbourne, P. M. McLellan, and R. M. L. Kensit, "Curvature profiles for plane curves," *Computer-Aided Design*, vol. 4, no. 4, pp. 176–184, 1972.

[33] P. Decaudin, D. Julius, J. Wither, L. Boissieux, A. Sheffer, and M. Cani, "Virtual garments: A fully geometric approach for clothing design," *Computer Graphics Forum*, vol. 25, no. 3, pp. 625–634, 2006.

[34] C.C.L. Wang and K. Tang, "Achieving developability of a polygonal surface by minimum deformation: A study of global and local optimization approaches," *The Visual Computer*, vol. 20, no. 8-9, pp. 521–539, 2004.

[35] Y. Liu, H. Pottmann, J. Wallner, Y.-L. Yang, and W. Wang, "Geometric modeling with conical meshes and developable surfaces," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 681–689, 2006.

[36] C.C.L. Wang, "Towards flattenable mesh surfaces," *Computer-Aided Design*, 2007.

[37] M. Kilian, S. Flöry, Z. Chen, N.J. Mitra, A. Sheffer, and H. Pottmann, "Curved folding," *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*, pp. 1–9, 2008.

[38] M.S. Floater, "Parametrization and smooth approximation of surface triangulations," *Comput. Aided Geom. Des.*, vol. 14, no. 3, pp. 231–250, 1997.

[39] K. Madsen, H. Nielsen, and O. Tingleff, *Optimization with constraints*, Lecture Notes, 2004.

[40] J. Nocedal and S.J. Wright, *Numerical Optimization*, Springer-Verlag, 1999.

[41] C.C.L. Wang, "Cybertape: an interactive measurement tool on polyhedral surface," *Computers & Graphics*, vol. 28, no. 5, pp. 731 – 745, 2004.

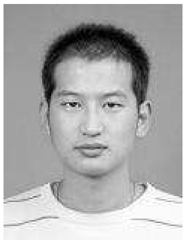
[42] G.V.Ravi Kumar, P. Srinivasan, V. Devaraja Holla, K. G. Shastry, and B. G. Prakash, "Geodesic curve computations on surfaces," *Computer Aided Geometric Design*, vol. 20, no. 2, pp. 119–133, 2003.

[43] C.C.L. Wang, "Computing length-preserved free boundary for quasi-developable mesh segmentation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 1, pp. 25–36, 2008.

[44] T.H. Cormen, C. Stein, R.L. Rivest, and C.E. Leiserson, *Introduction to Algorithms*, McGraw-Hill Higher Education, 2001.

[45] J.W. Demmel, S.C. Eisenstat, J.R. Gilbert, X.S. Li, and J.W.H. Liu, "A supernodal approach to sparse partial pivoting," *SIAM J. Matrix Anal. Appl.*, vol. 20, no. 3, pp. 720–755, 1999.

[46] P.N. Azariadis and N.S. Sapidis, "Planar development of free-form surfaces: Quality evaluation and visual inspection," *Computing (Vienna/New York)*, vol. 72, no. 1-2, pp. 13–27, 2004.



Yunbo Zhang is a Ph.D Student of Department of Mechanical and Automation Engineering at The Chinese University of Hong Kong. His research interests include CAD&CAM, physical based geometric modeling, and computer graphics. Zhang received his Mphil in digital design and manufacture from National Engineering Research Center for CAD, Huazhong University of Science and Technology.

Charlie C.L. Wang is currently an Associate Professor at the Department of Mechanical and Automation Engineering, the Chinese University of Hong Kong, where he began his academic career in 2003. He gained his B.Eng. (1998) in Mechatronics Engineering from Huazhong University of Science and Technology, M.Phil. (2000) and Ph.D. (2002) in Mechanical Engineering from the Hong Kong University of Science and Technology. He is a member of IEEE and ASME, and an executive committee member of Technical Committee on Computer-Aided Product and Process Development (CAPPD) of ASME. Dr. Wang has received a few awards including the ASME CIE Young Engineer Award (2009), the CUHK Young Researcher Award (2009), the CUHK Vice-Chancellor's Exemplary Teaching Award (2008), and the Best Paper Awards of ASME CIE Conferences (in 2008 and 2001). His current research interests include geometric modeling in computer-aided design and manufacturing, biomedical engineering and computer graphics, as well as computational physics in virtual reality.