Optimal Boundary Triangulations of an Interpolating Ruled Surface

Charlie C. L. Wang
Department of Automation and Computer-Aided Engineering
Chinese University of Hong Kong
Shatin, N.T., Hong Kong, P.R.China
E-mail: cwang@acae.cuhk.edu.hk

Kai Tang*
Department of Mechanical Engineering
Hong Kong University of Science and Technology
Clear Water Bay, KLN, Hong Kong, P.R.China
E-mail: mektang@ust.hk

Abstract

We investigate how to define a triangulated ruled surface interpolating two polygonal directrices that will meet a variety of optimization objectives which originate from many CAD/CAM and geometric modeling applications. This optimal triangulation problem is formulated as a combinatorial search problem whose search space however has the size tightly factorial to the numbers of points on the two directrices. To tackle this bound, we introduce a novel computational tool called multi-layer directed graph and establish an equivalence between the optimal triangulation and the single-source shortest path problem on the graph. Well known graph search algorithms such as the Dijkstra's are then employed to solve the single-source shortest path problem, which effectively solves the optimal triangulation problem in O(mn) time, where n and m are the numbers of vertices on the two directrices respectively. Numerous experimental examples are provided to demonstrate the usefulness of the proposed optimal triangulation problem in a variety of engineering applications.

Keywords: interpolation, ruled surface, weighted graph, global optimum, design and manufacturing.

* corresponding author.

1. Introduction

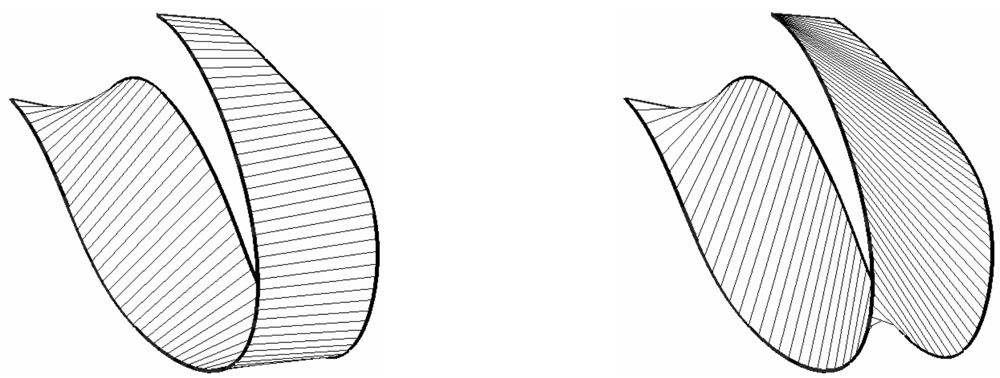
Ruled surfaces are widely used in computer-aided design and manufacturing (CAD/CAM) and computer graphics applications. For example, they are utilized to approximate freeform surfaces so that efficient NC tool paths can be generated [1]. In [2] mould drafts are created on freeform surfaces by approximating isoline surfaces with ruled surfaces. Ruled surfaces are also used in [3] to construct a surface by directionally offsetting 3D curves, where the resulting surface patches are useful elements in some engineering design applications (e.g., sheet metal products with flanges, overflow patches on a forging die, and cutting blades for a trimming die). Ruled surfaces are also the basic surface type for studying surface developability [4-6], which is an important surface property required in garment manufacturing. Mathematically, a ruled surface is the simplest form of surface interpolating two spatial curves: given two 3D C^1 curves $C_1(t)$ and $C_2(t)$ defined on $t \in [0, 1]$, the ruled surface defined on them is the simple linear interpolation between the two corresponding points $C_1(t)$ and $C_2(t)$, i.e.,

$$S(t, w) = (1 - w)C_1(t) + wC_2(t) \qquad (t, w \in [0, 1]), \tag{1}$$

where the line segment $<C_1(t), C_2(t)>$ is referred to be a *ruling*, and the two curves C_1 and C_2 are called the *directrices*, or *rails* sometimes.

Given a pair of directrices C_1 and C_2, depending on their parameterizations, different ruled surfaces can be generated, all interpolating the same two curves. Refer to Fig. 1 for an illustrative example. The different interpolations are best described by a *parameterization mapping* function $\xi(t)$, which can be any mapping from [0,1] to [0,1], as long as it is monotone and C^1 continuous. An interpolating ruled surface of C_1 and C_2 then is defined as

$$S(t, w) = (1 - w)C_1(t) + wC_2(\xi(t)) \qquad (t, w \in [0, 1]). \tag{2}$$



(a) (b)

Fig. 1 Different parameterizations on the same two rails lead to different ruled surfaces

The problem we investigate is finding "optimal" mappings $\xi(t)$ to realize certain optimization objectives. For example, one such optimization objective could be "minimal area", i.e., the corresponding ruled surface has the minimal area among all the ruled surfaces interpolating the same two directrices. Another optimization objective is "maximal developability", which seeks to maximize the number of "twist-free" rulings on a ruled surface, where a ruling is twist-free if all the surface normal vectors along it are parallel to each other – in case all the rulings are twist-free then the ruled surface is developable [4]. Theoretically, this is a variational optimization problem [7]. Let $\Sigma^k([0,1])$ denote the vector space of all the class C^k real-valued functions defined on [0,1]. The problem of optimal mapping $\xi(t)$ then can be cast in the following variational form:

$$J(\xi) = \int_0^1 \int_0^1 F(t, w, C_1(t), C_2(\xi(t)), \xi(t)) dw dt, \quad (3)$$

where the *functional* ξ is defined in $\Sigma^1([0,1])$ and $F(\ldots)$ is a function, usually very complicated and non-rational, depending on the specific optimization objective. Conceivably, it is virtually impossible to find an exact solution to Eq. (3) due to the double integration and the intertwining nature of $F(\ldots)$. Therefore, only approximate and numerical solutions are possible. One natural approach is to restrict $\xi(t)$ to a specific type represented by a set of real-valued coefficients $\{c_1, c_2, \ldots, c_k\}$ for some k (e.g., band-limited B-splines with c_i's as the control points), and convert Eq. (3) to an energy minimization problem with c_i's as optimization variables, which can then be solved by traditional functional optimization techniques such as the conjugate-gradient method [8]. This heuristic numerical approach has seen some success in many disciplines, e.g., imaging processing [9].

In this paper, we propose a rather different approach than the heuristic energy minimization: we formulate the problem as a combinatorial optimization problem and propose efficient algorithmic solutions for it. To be a little bit specific, the two directrices are first approximated by polygonal chains, and a special type of triangulation is then sought to interpolate these two polygons that will realize the given optimization objective in its corresponding discrete form. Since integration has to be involved, any numerical approach must resort to discretization in t and w spaces; thus, our piecewise-linear approximation of the directrices does not lose any data precision compared to numerical solutions. On the other hand, our algorithmic solution guarantees to find the global optimum, unlike numerical solutions such as the conjugate-gradient method which must deal with convergence issue and are never able to ensure the global optimality of the final result. For the record, in [10] we recently developed a technique of optimal triangulation for interpolating two polygonal chains that attempts to maximize the total "developability" of the triangulation; however, the triangulation algorithm in [10] is based on a heuristic method – no global optimum is guaranteed. The main contributions of the work presented in this paper can be summarized by the following three points:

- A large spectrum of optimization objectives for interpolating ruled surfaces are proposed and mathematically formulated in their discrete form; they have direct relevance in a variety of diverse applications.

- A unified O(mn) algorithm is presented that achieves global optimizations for all the objectives defined, where m and n are the numbers of the sampling points on the directrices $C_1(t)$ and $C_2(t)$ respectively.
- A large set of experimental examples are provided.

The rest of the paper is organized as follows. After presenting necessary preliminaries in Section 2, the various optimization objectives are rigorously formulated in Section 3. In section 4, we introduce the single-layer graph and give a detailed account on how to use it to achieve the majority of the optimization objectives defined in Section 3. The concept of multi-layer graph is then introduced in Section 5 which helps solve the optimization problems that a single-layer graph is unable to. A large set of experimental examples are then provided in Section 6 to demonstrate the functionality and usefulness of the proposed interpolation scheme, followed by our conclusion and a discussion on some future potential research topics.

2. Preliminaries

For notational purpose, we first introduce the term of a *strip*, as given below.

Definition 1 A strip defined on $C_1(t)$ and $C_2(t)$ is a closed 3D polygon made of two discrete directrices $P = \{p_1, p_2, \ldots, p_{m-1}, p_m\}$ and $Q = \{q_1, q_2, \ldots, q_{n-1}, q_n\}$, and the two straight lines linking their endpoints (i.e., p_1q_1 and p_mq_n), where P and Q are the polygonal chains approximating $C_1(t)$ and $C_2(t)$, with m and n vertices respectively.

The vertices in P and Q can be sampled on $C_1(t)$ and $C_2(t)$ either uniformly in t or adaptively according to chordal heights. From *descriptive geometry* [11], we define a special type of triangulation on a strip, called *boundary bridge triangulation* (or BBT). In a BBT, there can only be two types of edges: (1) *bank edges*, i.e., the line segments on P and Q themselves (e.g., p_ip_{i+1} and q_jq_{j+1}), and (2) *bridge edges*, those whose two end points fall on different directrices (e.g., p_iq_j). Two bridge edges are called *adjacent* to each other if they belong to a same triangle in the BBT. Based on these terms, the formal definition of a boundary bridge triangulation is given below.

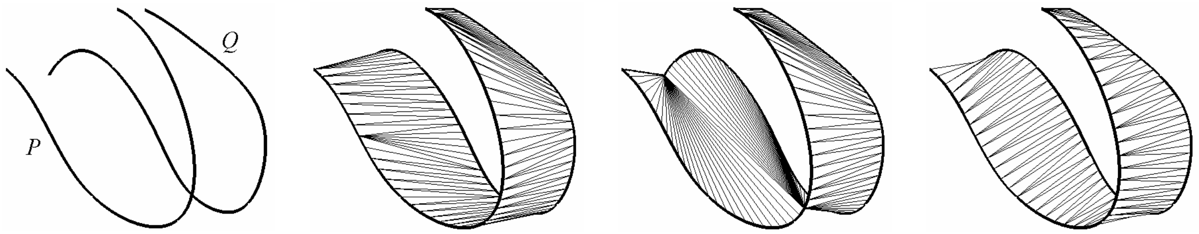
Definition 2 A boundary bridge triangulation (BBT) defined on two directrices P and Q is an ordered collection of triangles $\mathrm{M} = \{T_1, T_2, \ldots, T_\mathrm{N}\}$, which is formed by iteratively applying the following two operators starting from the bridge edge p_1q_1 and ending at p_mq_n:

P-succeed: when this operator is applied to a bridge edge p_iq_j, a new triangle defined by the three edges p_iq_j, $p_{i+1}q_j$ and p_ip_{i+1} is formed;

Q-succeed: this operator constructs a new triangle with three edges p_iq_j, p_iq_{j+1} and q_jq_{j+1}, when applied to a bridge edge p_iq_j.

In a BBT, each triangle is formed by two bridge edges and one bank edge. It is trivial to see that there are exactly $m+n-2$ bank edges in any M. Since every bank edge contributes to exactly one triangle, and vice versa, there are exactly a total of $m+n-2$ triangles in any M. Moreover, all the bridge edges in an M satisfy a partial ordering relationship, called *no-crossing relationship*: for any two bridge edges p_iq_j and p_kq_l, either $i \le k$ and $j \le l$ or $k \le i$ and $l \le j$.

One can interpret a BBT M as a discrete approximation of the parameterization map $\xi(t)$ as in Eq. (2). A bridge edge p_iq_j is exactly a ruling on the ruled surface, with $p_i = C_1(t_i)$ and $q_j = C_2(\xi(t_i))$, for some t_i. Since all the vertices on both P and Q are fixed, by choosing different vertices on Q to form bridge edges with vertex p_i, one effectively realizes different $\xi(t_i)$. Taking into consideration of the no-crossing relationship between the bridge edges, the set of all the BBTs then constitute discrete approximations of $\xi(t)$. These approximations approach to a continuum $\xi(t)$ when the number of sampling points on C_1 and C_2 tends to infinity (i.e., $m \to \infty$ and $n \to \infty$). In Fig. 2, we show several different BBTs, all on a same pair of P and Q.



(a) **(b)** **(c)** **(d)**

Fig. 2 Different BBTs on the same directrices *P* and *Q* in (a)

It is important to point out that, for two arbitrary directrices P and Q with m and n vertices respectively, there is a total of $\binom{m+n-2}{m-1} = \binom{m+n-2}{n-1} = \frac{(m+n-2)!}{(m-1)!(n-1)!}$ distinct boundary bridge triangulations (refer to the proof given in [10]). This is an extremely formidable number. To help appreciate its magnitude, for just a small $n = m = 100$, this number is larger than 10^{49}! Our task is to find one particular BBT from this huge pool that will optimize certain given scalar objective (e.g., the total amount of bending angle between the triangles in the triangulation). Obviously, exhaustive search is neither plausible nor practical. Our main contribution is that this search is reduced into a single-source shortest path problem on a weighted graph, so that the optimal order of triangulation operators can be determined with the help of well-established shortest path algorithms [12]. The idea of using weighted graph for optimal triangulation was explored by other authors before, e.g., the work of [13] for minimal area triangulation. However, it is by no means a trivial effort to extend their work to a more general optimal triangulation with complicated optimization objectives. Moreover, for some of the optimization objectives stipulated in this paper, such as the minimization of total bending energy, the weights on the edges in the graph are no longer static – they become dynamic and path dependent; as a result, traditional single-source shortest path algorithms (e.g., the Dijkstra's algorithm) can not be directly used. As we will show, delicate and novel techniques need to be developed to deal with this path-dependence, so that traditional single-source shortest path algorithms such as the Dijkstra's become applicable again. But before that, though, we need to provide a carefully defined set of optimization objectives, in the next section.

3. Classification of Types of Optimization

A total of six types optimization objectives are defined and discussed.

3.1 Minimal surface area

The first optimization objective, also perhaps the simplest, is the minimization of the total surface area.

Definition 3 For a given BBT M, the surface area function A(M) is the summation area of all the triangles in M.

Minimal surfaces are defined as surfaces with zero mean curvature [4]. Pertaining to our setting, minimal surfaces may also be characterized as surfaces of minimal surface area for given boundary conditions, which is a problem in the calculus of variations known as Plateau's problem. By fixing the representation form of surfaces, the Plateau's problem is reduced to a sub-problem (e.g., the Plateau-Bézier problem [14], which focuses on finding a Bézier surface with minimal area from among all Bézier surfaces with a prescribed border). Similarly, the Plateau-Ruled-Surface problem refers to determining a ruled surface with minimal surface area interpolating the given directrices. The functional that can be adjusted in a Plateau-Ruled-Surface problem is exactly the parameterization map $\xi(t)$. In the discrete form, this translates to finding a bridge boundary triangulation with the minimal surface area function value, thus the following objective.

Objective 1 *Minimal surface area*: Given two discrete directrices P and Q, find a boundary bridge triangulation M with minimal A(M) among all the possible BBTs on P and Q.

3.2 Maximal Developability

The next optimization objective relates to the developability of a ruled surface. A ruled surface is not developable in general. However, if the rulings move along the directrices in such a way that the tangent plane to the surface remains the same at each ruling, the surface is then developable. This is known as the *common tangent plane condition*, which leads to the concept of *normal twist* on a BBT.

Definition 4 The two normal vectors at vertices p_i and q_j of a bridge edge p_iq_j on a BBT is defined as:

$$n_{p_i} = \frac{p_i q_j \times t_{p_i}}{\left\| p_i q_j \times t_{p_i} \right\|} \text{ and } n_{q_j} = \frac{p_i q_j \times t_{q_j}}{\left\| p_i q_j \times t_{q_j} \right\|} \tag{4}$$

where t_{p_i} and t_{q_j} are tangents on P and Q at p_i and q_j.

Definition 5 The *normal twist* of a bridge edge p_iq_j is defined as $Tw(p_iq_j) = 1 - n_{p_i} \cdot n_{q_j}$; the *total normal twist* – N_T(M) – on a boundary bridge triangulation M is then defined to be the summation of normal twists of all the bridge edges in M.

Since the normal twist on a bridge edge is non-negative, the N_T(M) of any M is also non-negative. When N_T(M) = 0, which means that the normal twist on every bridge edge is zero, we say that this M satisfies the *common tangent plane condition* everywhere. The scalar N_T(M) can then be adopted as a measurement of the *developability* of M, which leads to our second optimization objective.

Objective 2 *Minimal total twist*: finding a boundary bridge triangulation M that minimizes the total normal twist N_T(M) for given directrices P and Q.

In [15], Frey proposed a relatively weak condition for measuring the developability of a boundary triangulation: every interior edge must be locally convex in a developable boundary triangulation. Since our bridge boundary triangulation is a special type of boundary triangulation, this proposition also applies. The local convexity is defined below.

Definition 6 A bridge edge p_iq_j is said to satisfy the *local convexity* property if it lies on the convex hull of the six points $\{p_{i-1}, p_i, p_{i+1}, q_{j-1}, q_j, q_{j+1}\}$, otherwise it is said to be *concave*.

The local convexity provides another means for measuring the developability of a ruled surface. When the sampling density tends to infinitesimal (i.e., the numbers of sampling points m and n turn to ∞), the local convexity at a bridge edge becomes the common tangent plane condition at that ruling. The following maximization is thus in order.

Objective 3 *Maximal convexity*: finding a boundary bridge triangulation M that maximizes the number of locally convex bride edges for given directrices P and Q.

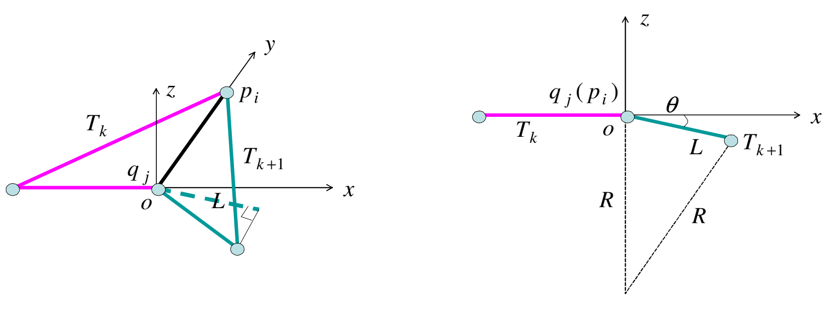


Fig. 3 Bending energy calculation on a bridge edge

3.3 Minimal bending energy

The strain energy of a ruled surface gives an integral measure of the curvature of the surface. Since a BBT is discrete, its strain energy is represented in the form of *bending energy*. Suppose that in an M, the bridge edge p_iq_j is shared by two adjacent triangles T_k and T_{k+1} with T_k lying in the x-y plane and p_iq_j coincident with the y-axis. Figure 3 illustrates the bending between T_k and T_{k+1} along p_iq_j. Assuming that the bending angle is very small, the energy due to this bending is (ref. [10])

$$U = \int_0^L \frac{EI(s)}{2R^2} ds \tag{5}$$

which can be further simplified into a form

$$U = K \frac{A \sin^2 \theta}{L^2} \tag{6}$$

where K is a coefficient determined by the thickness of the sheet and the Young's modulus (for detail derivations, see [10]). Since only relative value is needed for comparison purpose, we simply set K to *one*. For a boundary bridge triangulation, there are exactly m+n-1 bridge edges; so the total bending energy on a given boundary bridge triangulation M can be computed by

$$U(M) = \sum_{k=1}^{m+n-1} U_k \tag{7}$$

with U_k representing the bending energy on the k^{th} bridge edge. The first (k = 1) and last (k = m+n-1) bridge edge on a BBT are assumed to be in natural state, i.e., free of bending. We hence set up the following objective.

Objective 4 *Minimal bending*: For given directrices P and Q, find a boundary bridge triangulation M that minimizes the total bending energy U(M).

3.4 Minimal mean curvature variation

The shape quality of a surface is sometimes measured by the variation of curvatures on it [16], and a fair surface is defined as one with little curvature variation. On a BBT, curvature only exists cross bridge edges since the 2nd derivative of $S(t, w)$ with respect to w is zero (i.e., the curvature along bridge edges is zero). For any bridge edge e, the mean curvature vector defined on it can be computed by (ref. [17])

$$H_e = (2\|e\| \cos\frac{\theta_e}{2}) n_e \tag{8}$$

where θ_e is the dihedral angle of the edge e, $\|e\|$ is the length of bridge, and n_e is unit normal vector on the bridge edge. The unit normal vector on e can be calculated by $n_e = \frac{n_l + n_r}{\|n_l + n_r\|}$ with n_l and n_r being the unit normal vectors of its left and right adjacent triangles respectively. Thus, the norm of difference vector on the mean curvature vectors of two adjacent bridge edges can be adopted to measure the fairness of a BBT.

Definition 7 For a boundary bridge triangulation M with a sequence of ordered bridge edges {e_1, e_2, …, e_i, e_{i+1}, …, e_{m+n-1}}, the mean curvature difference between e_i and e_{i+1} is defined by

$$\partial H_{e_i} = \left\| H_{e_i} - H_{e_{i+1}} \right\|, \tag{9}$$

and the *total mean curvature variation* on M is given by

$$H(M) = \sum_{i=2}^{m+n-2} \partial H_{e_i} \ . \tag{10}$$

Similar to the bending case, there is no mean curvature defined on the first and last bridge edge. Therefore, the fairness of a BBT M can be measured by the integral *H*(M), which leads to the following optimization objective.

Objective 5 *Minimal mean curvature variation*: For given directrices *P* and *Q*, find a boundary bridge triangulation M that minimizes the total mean curvature variation *H*(M).

3.5 Minimal normal variation

A boundary bridge triangulation can also be utilized as an approximation of a blending surface. Suppose that two surfaces M_a and M_b have polygonal boundaries B_a and B_b respectively, and the Hausdroff distance between B_a and B_b is small. By taking B_a and B_b as the directrices, a BBT fills the gap between M_a and M_b, thus blending the two into a single polygonal mesh. When two surfaces are blended, the blending BBT is expected to follow the original normal vectors along B_a and B_b as much as possible. We introduce the following terms on a BBT to gauge this conformity.

Definition 8 The *normal variance* of a bridge edge p_iq_j is defined as

$$N_v(p_iq_j) = (1 - n_{p_i} \cdot \overline{n}_{p_i}) + (1 - n_{q_j} \cdot \overline{n}_{q_j}) \tag{11}$$

where n_{p_i} and n_{q_j} are discrete unit surface normal vectors as given in Eq. (4), $\overline{n}_{p_i}$ is the unit normal vector to surface M_a at p_i, and $\overline{n}_{q_j}$ the one to surface M_b at q_j; the *total normal variation* of M is then the summation of the normal variances over all the bridge edges e_i, as

$$V(\mathrm{M}) = \sum_i N_v(e_i)\,. \tag{12}$$

The corresponding optimization objective is then the following.

Objective 6 *Minimal normal variation*: For given directrices *P* and *Q*, find a boundary bridge triangulation M that minimizes the total normal variation *V*(M).

3.6 Coupled optimization

The optimization objectives so far prescribed are individual and independent of each other. They can also be combined to form a *coupled optimization*. This is particularly appealing in the case of Objective 3 whose corresponding function values are integers: two BBTs M and M', which both maximize the number of locally convex bridge edges, can have very different total bending energy *U*(M) and *U*(M'). Explicitly, there are two optimization objectives in a coupled optimization problem

– the primary objective and the secondary objective – and the final goal is to *meet* the primary objective while at the same time *try* to be as close as possible to the secondary objective. For the particular coupled optimization problem that we are interested in this paper, the primary optimization objective is Objective 3, i.e., to maximize the number of locally convex bridge edges, while two secondary optimization objectives are considered: 1) Objective 4, i.e., try to minimize the total bending energy U(M), and 2) Objective 5, attempt to minimize the mean curvature variation H(M).

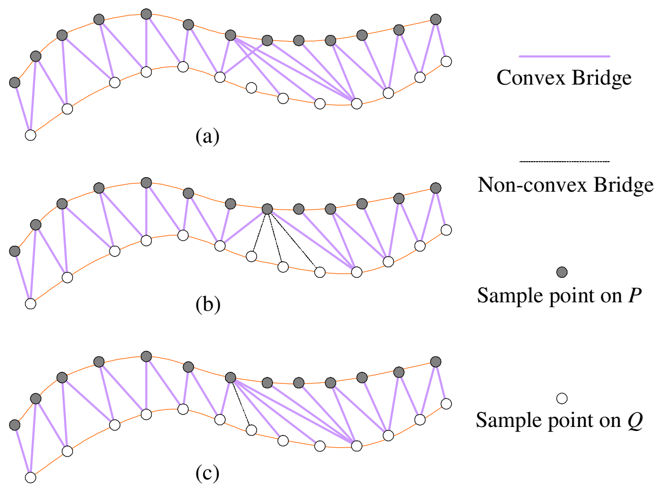


Fig. 4 Failure of local optimum approach in finding a global optimum

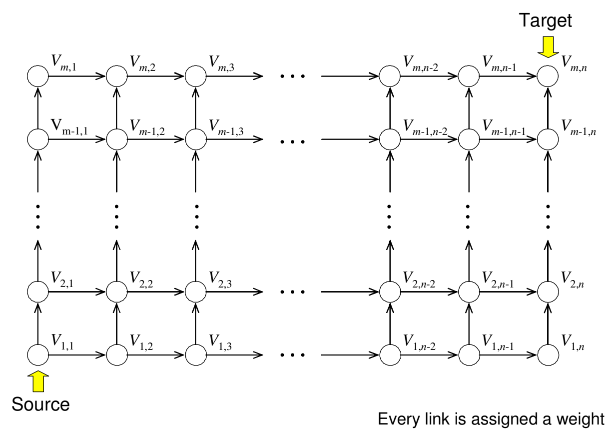


Fig. 5 An example single layer graph constructed from *P* and *Q* with *m* and *n* points respectively

4. Optimization Based on Single Layer Graphs

Having rigorously defined all the optimization objectives on a BBT, we now proceed to present our algorithmic solutions for achieving them. By Definition 2, a boundary bridge triangulation is generated on given directrices P and Q by applying P-succeed and Q-succeed operators iteratively. Thus, the problem is to find a right sequence of P-succeed and Q-succeed operators for a given optimization objective. As already alluded earlier in the beginning of the paper, a local optimum approach specifically for Objective 3 (i.e., maximizing the number of locally convex bridge edges) was developed by us in [10]. That approach is strictly local because the search is strictly sequential: starting from the default bridge edge p_1q_1, when determining the next bridge edge after the current active bridge edge p_iq_j (which together form a new triangle), the local costs made by $p_iq_j - p_{i+1}q_j$ and $p_iq_j - p_iq_{j+1}$ are compared, and the one with the smaller cost will be picked to be the next bridge edge; no back-track thus is performed. As a result, global optimum can be easily missed. For example, for the configuration (the directrices and the locally convex bridge edges) given in Fig. 4(a) and with the optimization objective being Objective 3, the local optimum approach of [10] would output a BBT shown in Fig.4(b). However, it is not hard to find another BBT (e.g., Fig. 4(c)) that has more locally convex bridge edges. This manifestation of inability of local optimum approaches applies to all the optimization objectives laid out in Section 3. Therefore, we need to develop search algorithms that must be of global nature.

Our basic idea is to convert the triangulation problem into a single-source shortest path problem on a weighted graph; the Dijkstra's algorithm can then be utilized to obtain the shortest path which uniquely determines an ordered sequence of *P*- and *Q*-succeed operators that generates a global optimum. The type of graph introduced in this section is referred to as the *Single Layer Graph* (SLG), while the other type of graph called *Multi-Layer Graph* will be discussed in the next section. A single layer graph Γ corresponding to the directrices *P* and *Q* is constructed by following the rules below:

Single_Layer_Graph_Construction:

- For every bridge edge p_iq_j, it has a corresponding node $V_{i,j}$ in Γ;
- A directed link $<V_{i,j}, V_{i+1,j}>$ is defined for every pair of "horizontally" neighboring nodes pointing from $V_{i,j}$ to $V_{i+1,j}$ with $i = 1, 2, \cdots, m-1$; and similarly a "vertical" directed link $<V_{i,j}$, $V_{i,j+1}>$ is defined for every pair of "vertically" neighboring nodes pointing from $V_{i,j}$ to $V_{i+1,j}$; and
- Every directed edge is assigned a weight.

Figure 5 gives an example of a single layer graph. Traveling on Γ, any path $\hbar$ from $V_{0,0}$ to $V_{m,n}$ indicates a unique BBT on the given strip, and vice versa. Every link in $\hbar$ can be viewed as an operator applied on the current bridge edge to form a new triangle. The horizontal links pertain to *P*-succeeds while the vertical links correspond to *Q*-succeed operators. Thus, the path $\hbar$ in fact gives an ordered sequence of operators which generates a valid BBT for the given directrices.

By taking $V_{1,1}$ as source and $V_{m,n}$ as target, a shortest path $\hbar^*$ linking them can be determined by using the well-known Dijkstra's algorithm. This shortest path has the smallest summation of the weights of the links among all the possible path linking $V_{1,1}$ and $V_{m,n}$ in Γ. By setting appropriate weights to the links in Γ according to different optimization objectives, a shortest path in Γ effectively realizes an globally optimal BBT for the strip. The optimization objectives that can be realized by this single level graph scheme are Objective 1 (minimal area), Objective 2 (minimal total twist), Objective 3 (maximal convexity), and Objective 6 (minimal normal variation), which we describe in details one by one next.

4.1 Triangulation with minimal area

To generate a BBT for Objective 1, we invoke procedure Single_Layer_Graph_Construction to build a single layer graph Γ comprising *mn* nodes and (2*mn*-*m*-*n*) links. The weight assigned to a link is the area of the triangle formed by the two incident nodes of the link; that is, the link from $V_{i,j}$ to $V_{i+1,j}$ is assigned the weight equal to the area of $\Delta p_i q_j p_{i+1}$, and the weight on the directed link $<V_{i,j}$, $V_{i,j+1}>$ is set to be the area of $\Delta p_i q_j q_{j+1}$. After that, the Dijkstra's algorithm is applied to Γ to determine a shortest path from $V_{1,1}$ to $V_{m,n}$, which is a sequence of operators that generate a BBT with the minimal total area.

4.2 Triangulation with minimal total twist

Following Definition 5, the normal twist on a bridge edge $p_i q_j$ is measured by $Tw(p_i q_j) = 1 - n_{p_i} \cdot n_{q_j}$, where the two discrete surface normal vectors n_{p_i} and n_{q_j} are determined by Eq. (4). To evaluate n_{p_i} and n_{q_j}, the tangents at p_i on *P* and at q_j on *Q* are requested. If *P* and *Q* are sampled on two C^1 continuous parametric curves, the tangents on them at p_i and q_j will be adopted for t_{p_i} and t_{q_j} respectively. On the other hand, if only *P* and *Q* are available, to enhance data precision, we approximate the tangents at p_i and q_j by fitting a quadratic curve $C(t) = a_0 + a_1 t + a_2 t^2$ locally on the discrete points. Let $C(0) = p_{i-1}$, $C(1) = p_{i+1}$, and $C(\alpha) = p_i$, the $C(t)$ can be determined as

$$\begin{aligned} a_0 &= p_{i-1} \\ a_1 &= -\frac{1+\alpha}{\alpha} p_{i-1} + \frac{1}{\alpha(1-\alpha)} p_i + \frac{\alpha}{\alpha-1} p_{i+1} \\ a_2 &= \frac{1}{\alpha} p_{i-1} + \frac{1}{\alpha(\alpha-1)} p_i + \frac{1}{1-\alpha} p_{i+1} \end{aligned} \tag{13}$$

where $\alpha = \dfrac{\|p_i p_{i-1}\|}{\|p_i p_{i-1}\| + \|p_i p_{i+1}\|}$, i.e., by taking the chordal length parameterization on p_{i-1}, p_i, and p_i (assuming all the sample points are distinct). From $C(t)$, t_{p_i} can be determined by

$$t_{p_i} = C'(\alpha) = a_1 + 2\alpha a_2 . \tag{14}$$

For the tangents at the two ending points of *P*, we simply let $t_{p_0} = p_1 - p_0$ and $t_{p_m} = p_m - p_{m-1}$; when $\|p_i p_{i-1}\| = 0$ or $\|p_i p_{i+1}\| = 0$, to avoid singularity, $\alpha = \frac{1}{2}$ is chosen. The tangent of every point on *Q* is determined in the same manner.

Similar to the case of minimal area optimization, we build a single lever graph Γ based on *P* and *Q*. When deciding the weights, for a node $V_{i,j}$, all the (directed) links that end at $V_{i,j}$ are assigned the same weight of $Tw(p_i q_j)$. The shortest path determined by the Dijkstra's algorithm then gives a sequence of operators that generate a triangulation with the globally minimal total twist.

4.3 Triangulation with maximal convexity

The mechanism of using a single layer graph for achieving Objective 3 is exactly the same as the previous two cases, with the only difference in the weight assignment. For node $V_{i,j}$ in Γ, if the corresponding bridge edge $p_i q_j$ is locally convex, then the two (directed) links ending at $V_{i,j}$ are assigned a weight of 0; otherwise, these two links have a weight of 1. Figure 6 shows an example of graph Γ. Therefore, if a fully developable BBT M exists on *P* and *Q* (i.e., all the bridge edges in M are locally convex), the corresponding path of M in Γ has zero total weight. Accordingly, a shortest path from $V_{1,1}$ to $V_{m,n}$ in the constructed Γ designates an M that will maximize the number of locally convex bridge edges among any BBTs of *P* and *Q*.

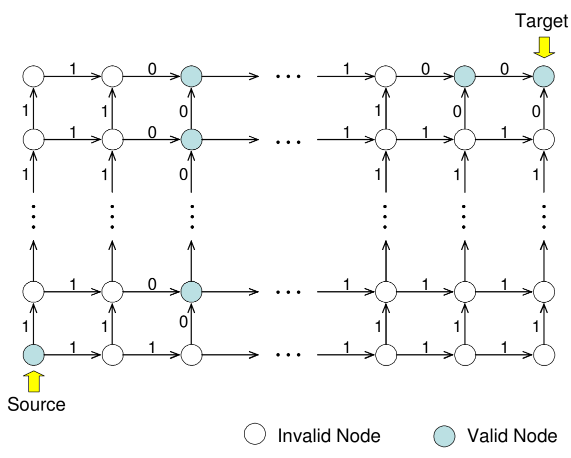


Fig. 6 An example single layer graph for developable triangulation

4.4 Triangulation with minimal normal variation

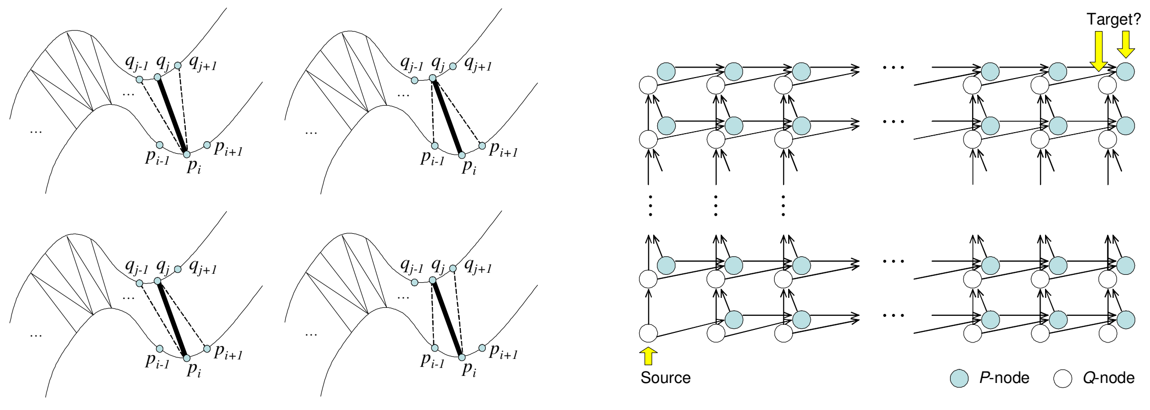
The optimization objective this time is given by Eq. (12). Again, everything else being the same, the only difference is in the weight assignment. For node $V_{i,j}$ in Γ, the two links ending at it are assigned the weight of $N_v(p_i q_j)$, as defined by Eq. (11). A shortest path from $V_{1,1}$ to $V_{m,n}$ in Γ corresponds to a BBT that minimizes the total normal variation.

5. Optimization Based on Multiple Layer Graphs

The single layer graph method employed for solving the optimization problems presented in the previous section bears a distinct character: the weight on a link in the graph is completely determined by its two nodes (i.e., the two bridge edges). For instance, in the case of Objective 1, the weight of the link from $V_{i,j}$ to $V_{i+1,j}$ is the area of the triangle made of the two edges p_iq_j and $p_{i+1}q_j$, isolated from any other edges. However, this isolation no longer exists for Objective 4 (minimal bending) and Objective 5 (minimal mean curvature variation). This is because, in these two cases, the weight on a link is path-dependent – it depends not only on the two nodes of the link but also on the previous node in the current path leading to the current node. As a result, the single layer graph becomes insufficient. Our solution is to use a multiple layer graph. As a matter of fact, the number of layers needed depends on the specific objectives, which we entail next.

5.1 Triangulation with minimal bending energy

When using Eq. (6) to evaluate the bending energy at a bridge edge p_iq_j, one needs not only the operator that will generate the next bridge edge, but also the previous operator which has resulted p_iq_j. More specifically, as shown in Fig.7(a), there are four possible amounts of bending energy associated with p_iq_j, all depending on which two of the four pertinent triangles to be chosen on the final BBT: (1) $\Delta p_{i-1}p_iq_j$ and $\Delta p_ip_{i+1}q_j$, (2) $\Delta p_{i-1}p_iq_j$ and $\Delta p_iq_{j+1}q_j$, (3) $\Delta p_iq_jq_{j-1}$ and $\Delta p_ip_{i+1}q_j$, and (4) $\Delta p_iq_jq_{j-1}$ and $\Delta p_iq_{j+1}q_j$. The weights (which are the amounts of the associated bending energy) on edges <$V_{i,j}$, $V_{i+1,j}$> and <$V_{i,j}$, $V_{i,j+1}$> are not static – they are path-dependent, i.e., depending on the current path of search that arrives at node $V_{i,j}$.



(a) **(b)**

Fig. 7 Building the *dual layer graph* for global minimum bending triangulation: (a) four configurations of triangles neighboring a bridge edge, and (b) dual layer graph

To cater to this dynamic nature of weights, and still be able to utilize the Dijkstra's algorithm, we introduce a two-layer graph Ω called a *dual layer graph* (DLG). In this graph, every bridge edge has two corresponding nodes – one, called *P*-node, indicates that this bridge edge is generated by a *P*-

succeed operation, and the other, called *Q*-node, tells that a *Q*-succeed operator was used to generate this bridge. Based on this dual node configuration, the non-unique weight problem is elegantly resolved. Specifically, when applying a *P*-succeed (respectively *Q*-succeed) operator on a node in Ω, regardless *P*- or *Q*-node, the graph edge should point to a *P*-node (respectively *Q*-node); and, for any graph node *V*, knowing whether it is a *P*- or *Q*-node, the weights on the two outgoing graph edges of *V* are uniquely determined by Eq. (6). Figure 7(b) offers a pictorial illustration of graph Ω: all the vertical graph edges indicate *Q*-succeed operators and all the horizontal graph edges denote *P*-succeeds.

The configuration at the first bridge edge p_1q_1 is unique – it is generated by neither *P*- nor *Q*-succeed, so only one node is constructed to represent it in Ω. The optimal triangulation problem is still a single-source problem. Using the Dijkstra algorithm on Ω, the minimum-weight paths from the source node to all the other nodes in the graph can then be determined. For the two dual graph nodes of the ending bridge edge p_mq_n, we choose the one whose path from the source node has the less weight – the path from the source to this node then determines a sequent of *P*- or *Q*-succeed operators that generates a BBT of the strip between *P* and *Q* with guaranteed (globally) minimal total bending energy.

5.2 Triangulation with minimal mean curvature variation

For the minimal mean curvature variation problem, the weight on a link in the graph is assigned the mean curvature difference ∂H_{e_i} defined on the two adjacent bridge edges e_i and e_{i+1} (see Eq. (9)). The mean curvature vector defined on a bridge edge (from Eq.(8)) itself needs a certain configuration of triangles around it. Therefore, compared to the bending energy case, more information is needed here for calculating the weights and a mere dual layer graph no longer suffices. Instead, a new type of multi-layer graph called *quadruple layer graph* (QLG) is required. In a quadruple layer graph Ξ, for an arbitrary bridge edge p_iq_j, four (quadruple) nodes are defined for it:

1) *QQ*-node $V_{i,j}^{QQ}$: the bridge p_iq_j is preceded by *Q*-succeed and followed by *Q*-succeed;

2) *PP*-node $V_{i,j}^{PP}$: the bridge p_iq_j is preceded by *P*-succeed and followed by *P*-succeed;

3) *QP*-node $V_{i,j}^{QP}$: the bridge p_iq_j is preceded by *Q*-succeed and followed by *P*-succeed; and

4) *PQ*-node $V_{i,j}^{PQ}$: the bridge p_iq_j is preceded by *P*-succeed and followed by *Q*-succeed.

The *QQ*-node represents the configuration around p_iq_j as indicated by the upper-left part in Fig. 7(a); the *PP*-node symbolizes the configuration of the upper-right part in Fig. 7(a); and the *QP*- and *PQ*-node stand for the configurations corresponding to the lower-left and lower-right part in Fig. 7(a) respectively. An example of quadruple layer graph is depicted in Fig. 8. Note that only *Q*-succeed can be applied to a bridge p_iq_j when $i = m$ and only *P*-succeed can be operated on p_iq_j when $j = n$; thus the last row of Ξ has neither *PP*- nor *QP*-nodes while at the last column only *PP*- and *QP*-nodes will be allowed. By the same reasoning, for the first row (i.e., $i=1$) in a QLG, only *QQ*- and *QP*- nodes are included; and similarly, the first column contains only *PP*- and *PQ*- nodes. The nodes for the boundary edges p_1q_1 and p_mq_n should be specially treated (since there is only one triangle linking each of them). Adopting the natural condition, the mean curvature vector is set to zero at both of them, and only one graph node is needed and used to represent each of the two edges.

Once all the nodes have been created in Ξ for all the bridge edges, we need to establish correct links between the nodes as well as to assign appropriate weights to the links. From the *QQ*-node (and *PQ*-node) of the bridge p_iq_j, two links are established to point to the *QP*- and *QQ*- node of the bridge edge p_iq_{j+1}, that is, the direct links $<V_{i,j}^{QQ}, V_{i,j+1}^{QP}>$, $<V_{i,j}^{QQ}, V_{i,j+1}^{QQ}>$, $<V_{i,j}^{PQ}, V_{i,j+1}^{QP}>$, and $<V_{i,j}^{PQ}, V_{i,j+1}^{QQ}>$ are established. The reason for having these four links is that the two adjacent nodes $V_{i,j}$ and $V_{i,j+1}$ must agree with each other – if a node V of p_iq_j specifies that its ensuing operator is *Q*-succeed, which means that the next bridge edge to take is p_iq_{j+1}, then V must point to a node V' of p_iq_{j+1} whose preceding operator is *Q*-succeed too. By the same token, another four direct links – $<V_{i,j}^{PP}, V_{i+1,j}^{PP}>$, $<V_{i,j}^{PP}, V_{i+1,j}^{PQ}>$, $<V_{i,j}^{QP}, V_{i+1,j}^{PP}>$, and $<V_{i,j}^{QP}, V_{i+1,j}^{PQ}>$ – are created to reflect the *P*-succeed and *P*-Precede relationship between edges p_iq_j and of $p_{i+1}q_j$. Since every node has knowledge of both its preceding and succeeding operators, the mean curvature vector associated with the node is fully determined by Eq. (8); the weight of a link is then readily decided by taking the norm of the difference vector between the mean curvature vectors at the two linked nodes (using Eq. (9)). As for the two special unitary nodes $V_{1,1}$ and $V_{m,n}$, since their mean curvature vectors vanish by assumption of natural condition, links should be established between them and all their adjacent edges; or explicitly, we have $<V_{1,1}, V_{i+1,j}^{PP}>$, $<V_{1,1}, V_{i+1,j}^{PQ}>$, $<V_{1,1}, V_{i,j+1}^{QP}>$, and $<V_{1,1}, V_{i,j+1}^{QQ}>$ for $V_{1,1}$, and $<V_{m-1,n}^{PP}, V_{m,n}>$, $<V_{m-1,n}^{QP}, V_{m,n}>$, $<V_{m,n-1}^{PQ}, V_{m,n}>$, and $<V_{m,n-1}^{QQ}, V_{m,n}>$ for $V_{m,n}$. One example of quadruple layer graph is shown in Fig. 8.

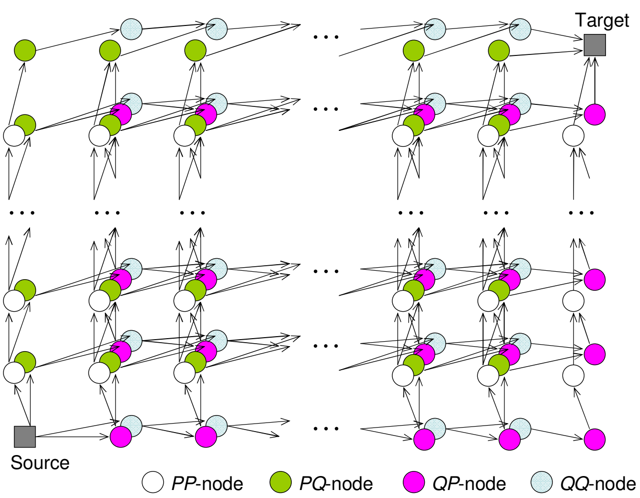


Fig. 8 Building the *quadruple layer graph* for the BBT with minimal mean curvature variation

By applying the Dijkstra's algorithm to the thus prescribed quadruple layer graph Ξ, the shortest path from $V_{1,1}$ to $V_{m,n}$ can be determined – this is the operation sequence that generates a boundary bridge triangulation with minimal mean curvature variation.

5.3 Triangulation with coupled optimization objectives

By now, all the individual optimization objectives have been addressed. The common solution is to establish a one-to-one correspondence between the BBTs and paths in a weighted graph (single- or multi-layers) and utilize efficient shortest-path algorithms to find a shortest path. The cost or weight assigned to a link in the graph is exclusively decided by the individual objective. In order to be able to use the same idea for coupled optimizations, we need to device a suitable weight assignment scheme that will cater to both the primary optimization objective and the secondary. We demonstrate how this can be done by solving two specific coupled optimization problems, as follows.

Maximal Convexity+ Minimal Bending Energy

To keep the number of locally convex bridges maximal while at the same time try to reduce the bending energy as much as possible, we first construct a *dual layer graph* Ω just like the case of minimizing the total bending energy as described in Section 5.1. Let $W_{\max}$ denote the *maximum total weight* of any paths in Ω, i.e., it is the total weight of the *longest path* in Ω from source $V_{1,1}$ to the target $V_{m,n}$ (this can be readily obtained by negating the original weights on the links and then applying the Dijkstra's algorithm). Next, for every link edge e in Ω, whose weight U_e is the bending energy due to Eq. (6), its weight is scaled down from U_e to $0.9(U_e / W_{max}) + 1.1$. We then examine every node in the graph, no matter whether it is a P- or a Q-node: if the bridge edge of this node is locally convex, the weights of all the links – two of them – pointing to this node are set to *zero*. After these two types of modifications on the weighs in Ω, a shortest path from $V_{1,1}$ to $V_{m,n}$ gives a BBT that will maximize the number of locally convex bridge edges, and at the same time minimize the summation of the bending energy on the concave bridges in the triangulation. In Appendix a formal proof is given for this assertion. Thus, through the manipulation of weights, we have successfully achieved the primary optimization and also the constrained secondary optimization.

Maximal Convexity + Minimal Mean Curvature Variation

The treatment for this coupled optimization is identical to that of the first, except that this time the graph is a quadruple layer graph Ξ, the original weights on the links in the graph are the mean curvature variations according to Eq. (8) and (9) (following the manner of Section 5.2), and the maximum total weight $W_{\max}$ is the maximal total mean curvature variation of any BBTs. A shortest path from $V_{1,1}$ to $V_{m,n}$ in the weight-adjusted Ξ then gives a BBT that will maximize the number of locally convex bridge edges, and at the same time minimize the total of the mean curvature variations occurring at the locally concave bridges in the triangulation.

6. Experimental Results and Applications

The first example, which we have briefly visited in Fig. 2, illustrates different strip triangulations with different optimization objectives on two simple discrete directrices; the various optimization objectives tested in this example are the minimal area (Fig. 9(b)), minimal twist (Fig. 9(c)), maximal number of locally convex edges (Fig. 9(d)), minimal bending energy (Fig. 9(e)), and minimal mean curvature variation (Fig. 9(f)). Their related computational statistics are listed in Table 1. From the table it is easily seen that when compared with each other, each triangulation method always achieves its intended optimization objective. Both the objectives of the minimal twist and the maximal convexity aim at achieving maximal developability of an interpolating ruled surface; however, the former is based on the original common tangent plane condition [4], while the latter is based on the local convexity proposition [15] for the discrete case. Conceivably, when the sampling is dense enough, these two would generate similar results, i.e., Fig. 9(c) vs. Fig. 9(d). Figure 10 depicts the paths that indicate the operation orders of the two triangulations, where the background is a matrix called the validity map – if the bridge edge p_iq_j is locally convex, a black box with width h is displayed at coordinate (ih, jh); otherwise, the region is left white. As revealed in Fig.10, the paths corresponding to Fig. 9(c) and Fig. 9(d) have little difference.

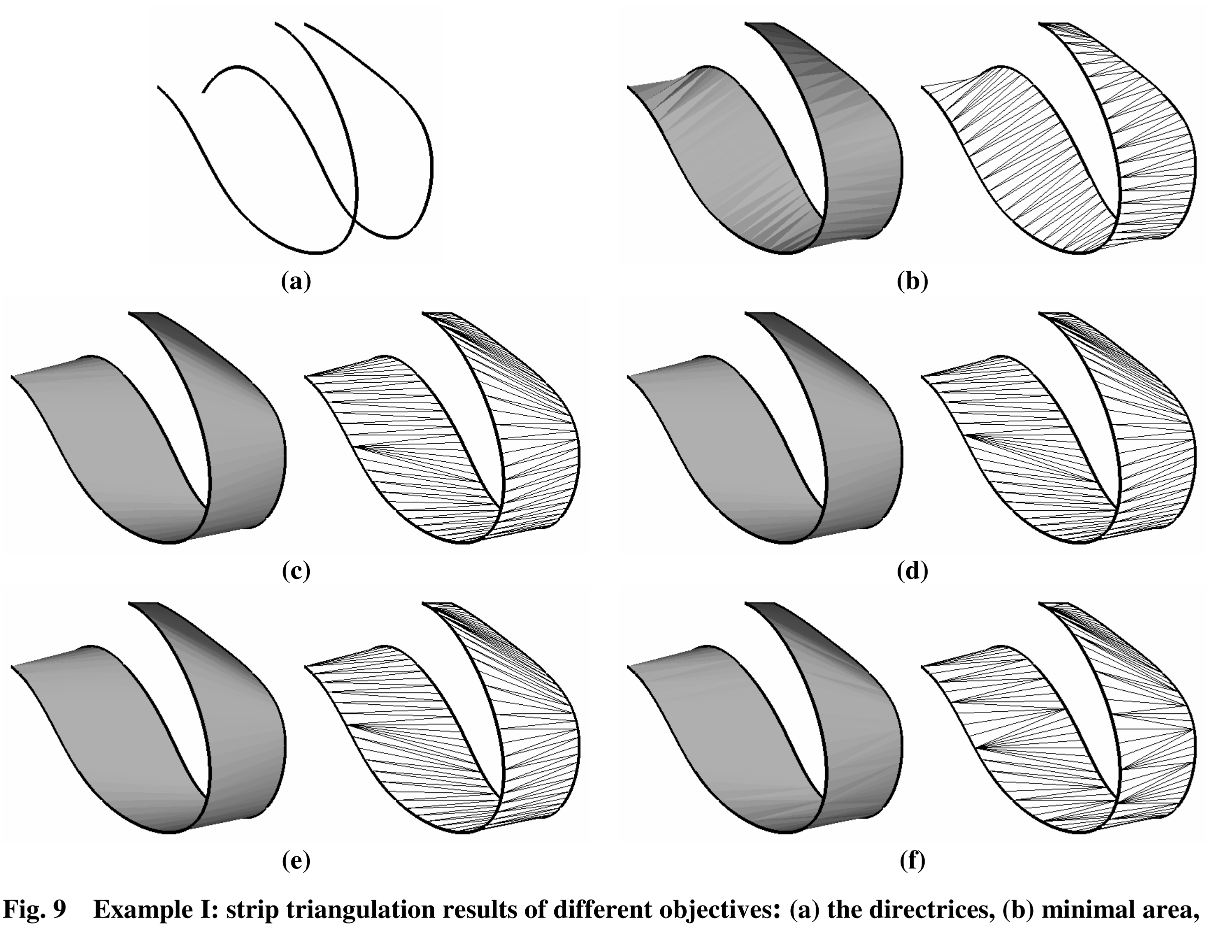
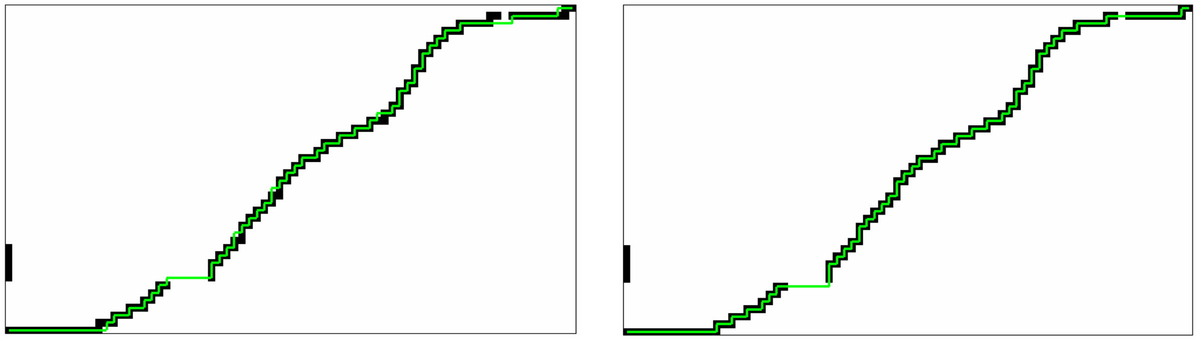
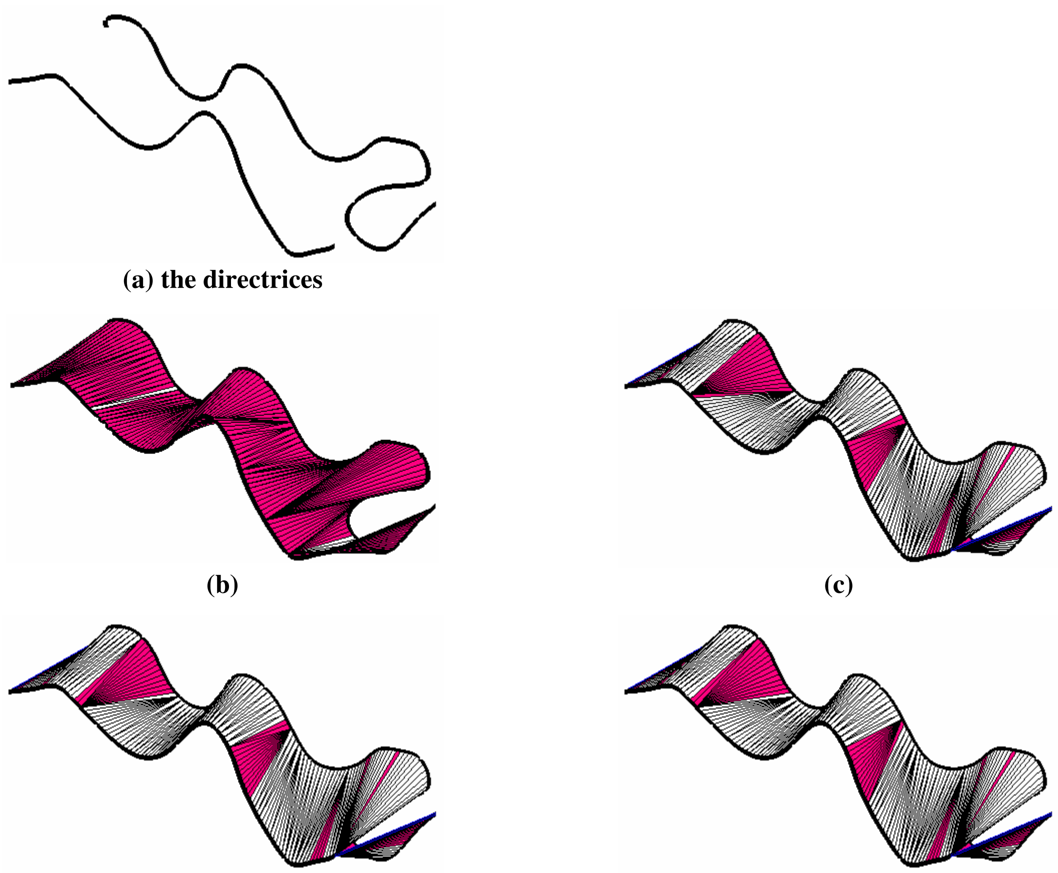


Fig. 9 Example I: strip triangulation results of different objectives: (a) the directrices, (b) minimal area, (c) minimal twist, (d) maximal convexity, (e) minimal bending, and (f) minimal mean curvature variation



(a) (b)

Fig. 10 Example I: comparison of paths on the validity map: (a) path of minimal twist BBT, and (b) path of maximal convexity BBT



(a) the directrices

(b) (c)

(d) (e)

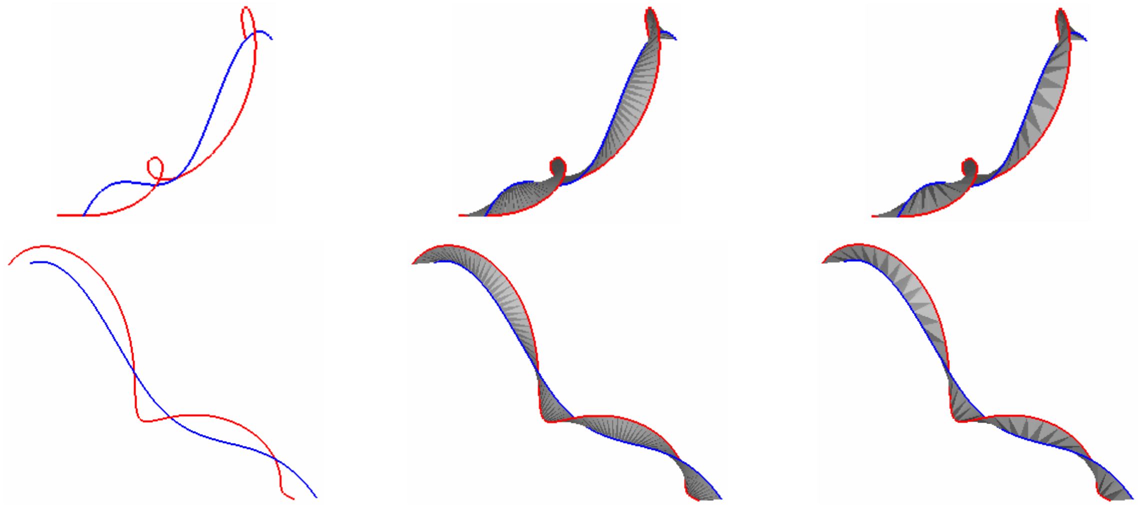
Fig. 11 Example II: strip triangulation with coupled objectives: (a) the directrices, (b) minimal area, (c) maximal convexity, (d) maximal convexity + minimal bending energy, and (e) maximal convexity + minimal mean curvature variation

Example II is provided to illustrate triangulation results with coupled optimization objectives. For the strip given in Fig.11(a), a boundary bridge triangulation with minimal area is generated as given in Fig. 11(b), and a BBT with the maximal number of locally convex edges is shown in Fig. 11(c), where the red regions are *non-developable* (i.e., bounded by concave edges). In the BBT in Fig. 11(c), there are 180 locally convex bridge edges, out of a total of (n+m-1) = 267 bridge edges. When the objective of maximizing number of locally convex edges is coupled with that of minimizing the bending energy, the resultant BBT, shown in Fig. 11(d), still maintains a total of 180 locally convex bridges while minimizing the bending energy on the remaining 87 concave bridge edges. Likewise, in Fig. 11(e) we display a triangulation resulted from the coupled optimization of maximal convexity + minimal mean curvature variation. From the statistics in Table 1, we can easily find that, comparing to the triangulations with other objectives, the BBT with minimal area gains a fine improvement – with more than 2% area reduced; however, the minimal area BBT gives worse result on the costs relating to other objectives (e.g., its bending energy is about 20 times of other BBTs'). This indicates that, when the directrices *P* and *Q* differ greatly from each other, we should carefully choose optimization objectives to determine appropriate boundary bridge triangulations.

Table 1 Computational Statistics

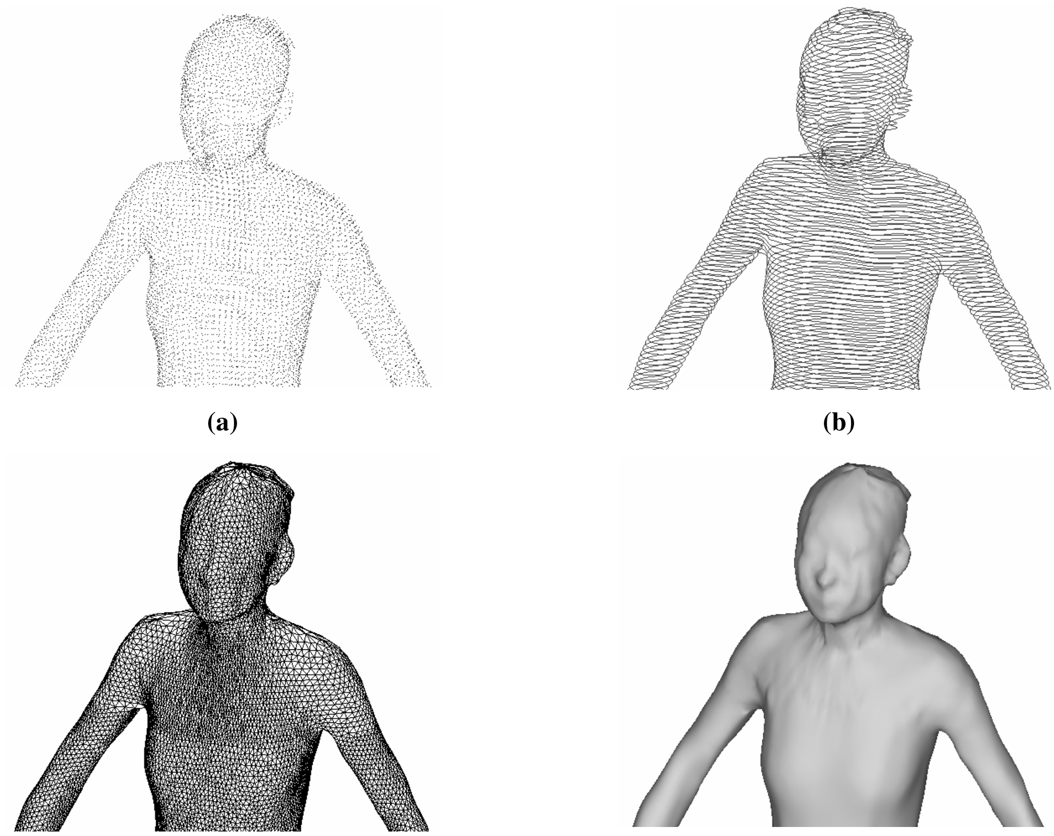
Example	Objective	Fig	Total Area	Total Normal Twist	Convex Edge Ratio	Bending Energy	Mean Curvature variation	Normal Variation
I	Minimal area	9b	**40.56**	16.62	12/119	65.47	49.65	N/A
	Minimal twist	9c	40.94	**5.28**	105/119	4.07	14.94	N/A
	Maximal convexity	9d	40.94	5.29	**112/119**	3.97	14.50	N/A
	Minimal bending	9e	40.94	5.30	108/119	**3.92**	14.79	N/A
	Minimal mean curvature variation	9f	40.93	5.93	74/119	7.37	**7.66**	N/A
II	Minimal area	11b	**87.13**	124.08	8/267	1168.67	218.29	N/A
	Maximal convexity	11c	89.22	77.00	**180/267**	60.86	53.98	N/A
	Maximal convexity + Minimal bending	11d	89.23	78.57	**180/267**	**58.45**	51.85	N/A
	Maximal convexity + Minimal mean curvature variation	11e	89.22	77.42	**180/267**	71.81	**51.10**	N/A
III	Minimal area	12b	**12.10**	20.54	5/501	49.03	32.34	N/A
	Minimal bending	12c	12.15	23.32	10/501	**22.84**	29.19	N/A
IV	Maximal convexity	14b	0.15	10.73	**153/385**	35.34	2.41	N/A
	Maximal convexity + Minimal bending	14c	0.15	9.24	**153/385**	**24.50**	2.14	N/A
V	Minimal bending	15e	41.85	3.02	103/221	**44.18**	29.54	7.39
	Minimal normal variation	15f	41.90	16.56	16/221	292.37	90.57	**4.35**

The rest of the examples of the experiments demonstrate the application of optimal triangulations in various fields. The first one, Example III, deals with design of a ribbon which is useful for the design of DNA and proteins [18], where a ribbon can be modeled by specifying its two directrices. As shown in Fig. 12, when two directrices of a ribbon are given (Fig. 12(a)), we can generate a interpolating triangular surface with minimal area as in Fig. 12(c), and we can also construct a surface with minimal mean curvature variation (see Fig. 12(d)). The comparison of cost functions about different objectives is listed in Table 1. As confirmed by the table, in this particular example, the total bending energy on a minimal bending triangulation is less than half of that on a minimal area triangulation.



(a) (b) (c)

Fig. 12 Example III: strip triangulation for ribbon design: (a) the directrices, (b) minimal area triangulation, and (c) minimal bending triangulation



(a) (b)

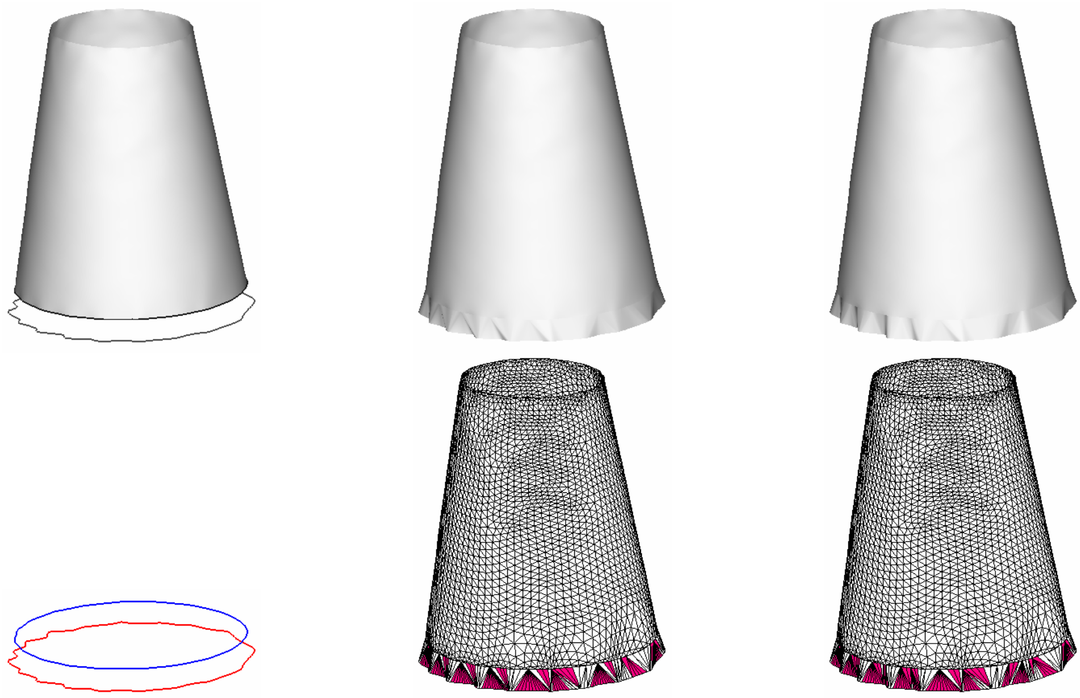
(c) (d)

Fig. 13 Contour-based surface reconstruction in human body modeling: (a) the point cloud, (b) contours generated, (c) surface by "sewing" the contours, and (d) shaded result

The second application example is contour-based surface reconstruction. Using any data points sectioning technique (e.g., one in [19]), a human model, which is originally represented by a 3D point cloud (Fig. 13(a)), can be sliced into many parallel polygonal contours (Fig. 13(b)). By interpolating consecutive neighboring pairs of the contours, a surface model can be reconstructed. Figure 13(c) displays one based on the minimal area optimization objective. Of course, other objectives could also be selected to generate varieties of optimal surfaces – in [20], a detail review of this tiling problem could be found.

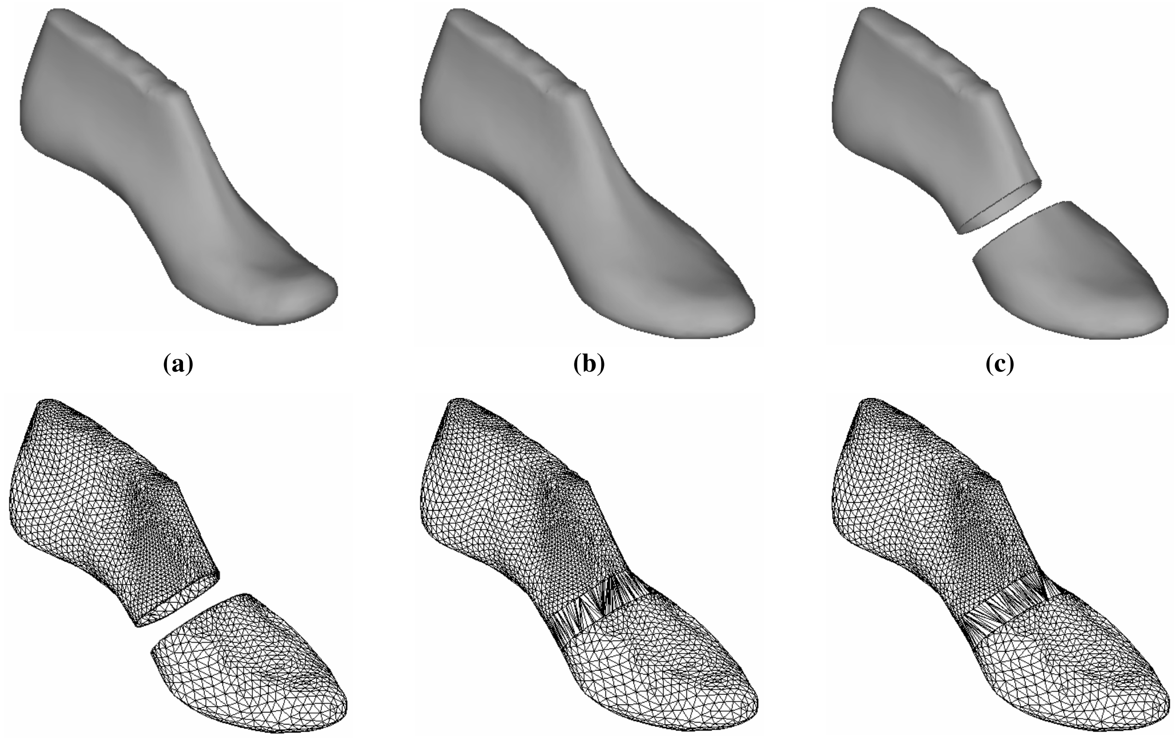
The third application example is surface wrinkle design, which is often needed in clothing or shoe design. For a given skirt (Fig.14(a)), wrinkles are required to be added at its bottom boundary. A "pattern" curve that reflects the general shape of wrinkles is first specified (i.e., the red curve in Fig. 14(a)), which forms a narrow strip with the bottom boundary of the skirt. A wrinkle surface then is constructed by triangulating this strip. Figure 14(b) shows a triangulation with a single optimization objective – maximizing the number of locally convex edges (i.e., the developability), and Fig. 14(c) depicts the result with a coupled objective: maximal convexity + minimal bending. As seen in the figure, Fig. 14(c) gives a smoother surface.

The next application example shows the usefulness of our triangulation as a blending tool. In shoe design, a common modeling method is to use parts of multiple existing shoe designs and patch them together to form a new design. For instance, the rear part of a shoe last (Fig. 15(a)) could be combined with the front part of another shoe last (Fig. 15(b)) to create a new design. These two parts need to be blended at their interfacing boundary curves so that a complete surface last can be formed. Figure 15(d) and 15(e) show two BBT blending results, one with minimal bending energy and the other with minimal normal variation. As expected, the latter gives a better performance in terms of the smoothness in transition between the two parts (see Table 1).



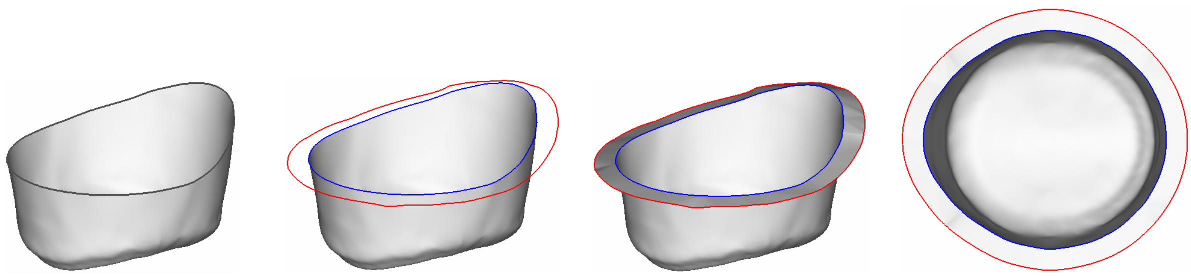
(a) **(b)** **©**

Fig. 14 Example IV: surface wrinkle design: (a) the skirt and the directrices to specify surface wrinkles, (b) wrinkle strip generated with the maximal convexity objective, and (c) wrinkle strip generated with the coupled objective of maximal convexity + minimal bending



(a) (b) (c) (d) (d) (e)

Fig. 15 Example V: strip blending in shoe design: (a) shoe last A, (b) shoe last B, (c) the rear part of A + the front part of B, (d) mesh representation of (c), (d) the blending strip with minimal bending energy, and (e) the blending strip with minimal normal variation.



(a) (b) (c) (d)

Fig. 16 Example VI: strip triangulation for design of a flange: (a) the sheet metal part to add a flange, (b) the directrices for optimal triangulation, (c) the flange as a BBT with minimal bending, and (d) top view

The last example shows the application of the presented optimal triangulation technique in defining a flange off a sheet metal part. As demonstrated in Fig. 16, the boundary curve of the part is offset (along the direction of the surface normal vector) to generate another directrix (the red curve) which together with the boundary curve define a strip. This strip is then triangulated using a chosen optimization objective, i.e., with the minimal bending energy as shown in Fig. 16(c) and 16(d).

7. Summary and future research

The primary goal of this paper is to develop an efficient algorithm for constructing an optimal triangulated ruled surface that interpolates two discrete directrices. We first provide a spectrum of rigorously defined optimization objectives for the construction that have application to a variety of practical problems. We then formulate this optimal triangulation problem as a combinatorial optimization problem whose search space nevertheless has a size that is factorially proportional to (m+n), with m and n being the numbers of vertices on the two directrices respectively. Our main contribution is the establishment of a one-to-one correspondence between the optimal triangulation problem and the single-source shortest-path problem on a weighted graph whose nodes and edges are both capped by the upperbound O(mn). Well-known single-source shortest-path algorithms such as the Dijkstra's can then be employed to find a shortest-path on the graph. Since the graphs developed in our approach are all *Directed-Acyclic-Graphs* (DAGs), the formulated optimal triangulation problem is efficiently solved in O(mn) time. (The Dijkstra's algorithm runs in O(|V|+|E|) time on DAGs, with |V| and |E| being the numbers of nodes and edges in the graph respectively, see [12].) Besides being efficient, the presented optimization algorithm is also straightforward to implement and robust – the conversion to the directed weighted graph is straightforward and the Dijkstra's algorithm is well-known to be robust and fast.

We are interested in some further extensions to the current. In the aspect of design, a more positive objective will be on how to modify the two given directrices P and Q, but within certain specified tolerance range, so that the resultant BBT is the optimum among all the possible designs of P and Q within the tolerance range.

References

[1] Han, Z., Yang, D.C.H., and Chuang, J.-J., 2001, "Isophote-based Ruled Surface Approximation of Free-form Surfaces and its Application in NC Machining", International Journal of Production Research, **39**(9), pp.1911-1930.

[2] Tokuyama, Y., and Seockhoon, B., 1999, "An Approximate Method for Generating Draft on a Free-form Surface", The Visual Computer, **15**(1), pp.1-8.

[3] Shin, H., and Cho, S.K., 2002, "Directional Offset of a 3D Curve", *Proceedings of the Seventh ACM symposium on Solid modeling and applications*, pp.329-335, ACM.

[4] DoCarmo M., 1976, *Differential Geometry of Curves and Surfaces*, Prentice-Hall, Englewood Cliffs, NJ.

[5] Pottmann, H., and Wallner, J., 1999, "Approximation Algorithms for Developable Surfaces", Computer Aided Geometric Design, **16**(6), pp.539-556.

[6] Peternell, M., 2004, "Recognition and Reconstruction of Developable Surfaces from Point", *Proceedings of Geometric Modeling and Processing 2004*, IEEE Computer Society, pp.301-310.

[7] Smith, D.R., 1998, *Variational Methods in Optimization*, Dover Publications, Inc., ISBN 0-486-40455-2.

[8] Belegundu, A.D., and Chandrupatla, T.R., 1999, *Optimization Concepts and Applications in Engineering*, Prentice-Hall, Upper Saddle River, N.J.

[9] Kass, M., Witkin, A., and Terzopoulos, D., 1988, "Snakes: Active Contour Models", International Journal of Computer Vision, **1**, pp 321-332.

[10] Tang, K., and Wang, C.C.L., 2005, “Modeling Developable Folds on a Strip”, Journal of Computing and Information Science in Engineering, ASME Transactions, **5**(1).

[11] Watts, E.F., and Rule, J.T., 1946, *Descriptive Geometry*, Prentice-Hall, New York.

[12] Cormen, T.H., Leiserson, C.E., Rivest, R.L., and Stein, C., 2001, *Introduction to Algorithms* (2nd ed.), MIT Press.

[13] Fuchs, H., Kedem, Z.M., Uselton, S.P., 1977, “Optimal Surface Reconstruction from Planar Contours”, Communications of the ACM, **20**(10), pp.693-702.

[14] Monterde, J., 2004, “Bézier Surfaces of Minimal Area: the Dirichlet Approach”, Computer Aided Geometric Design, **21**, pp.117–136.

[15] Frey, W.H., 2004, “Modeling Buckled Developable Surfaces by Triangulation”, Computer-Aided Design, **36**(4), pp.299-313.

[16] Moreton, H.P., and Sequin, C.H., 1992, “Functional Optimization for Fair Surface Design”, *SIGGRAPH 92 Proceedings*, pp.167-176.

[17] Hildebrandt, K., and Polthier, K., 2004, “Anisotropic Filtering of Non-linear Surface Features”, Computer Graphics Forum, **23**(3).

[18] Au, C.K, and Woo, T.C., 2004, “Ribbons: Their Geometry and Topology”, Computer-Aided Design & Applications, **1**, pp.197-206, CAD'04 Conference, Pattaya Beach, Thailand, May 24-28.

[19] Wang, C.C.L., Chang, T.K.K., and Yuen, M.M.F., 2003, "From Laser-scanned Data to Feature Human Model: a System Based on Fuzzy Logic Concept", Computer-Aided Design, **35**(3), pp.241-253.

[20] Stollnitz, E.J., DeRose, T.D., and Salesin, D.H., 1996, *Wavelets for Computer Graphics: Theory and Applications*, San Francisco, Calif.: Morgan Kaufmann Publishers.

Appendix

For the coupled objective of maximal convexity + minimal bending energy, after modifying the weights on a *Dual Layer Graph*, how can we still guarantee that the resultant triangulation still has the maximum number of locally convex edges while at the same time it also minimizes the total bending energy on the concave edges? A formal proof is given below.

Lemma A. After rescaling the weight on each bridge edge from U_e to $0.9(U_e / W_{\max}) + 1.1$, the shortest path found on the corresponding DLG from $\mathrm{V}_{1,1}$ to $\mathrm{V}_{m,n}$ gives a BBT that not only has the maximal number of locally convex bridge edges but also at the same time minimizes the total bending energy on the concave edges.

Proof. Let us consider two arbitrary paths from $V_{1,1}$ to $V_{m,n}$: Path-I has n_1 locally convex edges and the summation of bending energy on its concave edges is U_1, and Path-II has n_2 locally convex edges and U_2 is the summation of the bending energy on its concave edges. There are totally $m+n-1$

bridges in a valid BBT, so there are exactly $m+n-2$ links on both passes. The weight on Path-I after rescaling is

$$W_1 = 0.9(U_1 / W_{max}) + 1.1(m+n-2-n_1) ,$$

and the adjusted weight on Path-II is

$$W_2 = 0.9(U_2 / W_{max}) + 1.1(m+n-2-n_2) .$$

Consider the following situations:

1) If $n_1 < n_2$ and $U_1 \le U_2$, we have $1.1(m+n-2-n_1) > 1.1(m+n-2-n_2)$ and their difference is greater than one – since n_1 and n_2 are integers; also, we have $0.9(U_1 / W_{max}) \le 0.9(U_2 / W_{max})$ but with $\left|0.9(U_1 / W_{max}) - 0.9(U_2 / W_{max})\right| < 1$. Therefore, we get $w_1 > w_2$ – Path-II is shorter than Path-I. If $n_1 < n_2$ and $U_1 > U_2$, since $1.1(m+n-2-n_1) > 1.1(m+n-2-n_2)$ and $0.9(U_1 / W_{max}) > 0.9(U_2 / W_{max})$, Path-II is still shorter. Either way, Path-II is chosen.
2) For $n_1 > n_2$, we have $1.1(m+n-2-n_1) < 1.1(m+n-2-n_2)$ and the absolute difference between is greater than one; regardless $U_1 \le U_2$ or $U_1 > U_2$, because $\left|0.9(U_1 / W_{max}) - 0.9(U_2 / W_{max})\right| < 1$, the weights on the two paths satisfy $w_1 < w_2$ – that is, Path-I is shorter and hence is selected.
3) Lastly, suppose $n_1 = n_2$. If $U_1 < U_2$, we have $w_1 < w_2$; otherwise, $w_1 > w_2$. Either way, the path with less bending energy on the concave edges will be selected.

Q.E.D.