Fabrication of freeform objects by principal strips

Masahito Takezawa
Yokohama National University

Takuma Imai
Yokohama National University

Kentaro Shida
Yokohama National University

Takashi Maekawa*
Yokohama National University

Figure 1: *Left to right : Computation of orthogonal net of lines of curvature of automobile surfaces. Two-layered principal strips (maximum principal strips on the visible side). Two-layered principal strips (minimum principal strips on the visible side). Woven principal strips.*

Abstract

Current CAD modeling techniques enable the design of objects with aesthetically pleasing smooth freeform surfaces. However, the fabrication of these freeform shapes remains challenging. Our novel method uses orthogonal principal strips to fabricate objects whose boundary consists of freeform surfaces. This approach not only lends an artistic touch to the appearance of objects, but also provides directions for reinforcement, as the surface is mostly bent along the lines of curvature. Moreover, it is unnecessary to adjust the bending of these orthogonal strips during the construction process, which automatically reforms the design shape as if it is memorized, provided the strips possess bending rigidity. Our method relies on semi-isometric mapping, which preserves the length of boundary curves, and approximates angles between boundary curves under local minimization. Applications include the fabrication of paper and sheet metal craft, and architectural models using plastic plates. We applied our technique to several freeform objects to demonstrate the effectiveness of our algorithms.

Concepts: ●**Computing methodologies** → *Parametric curve and surface models;*

Keywords: lines of curvature, principal strips, principal quads, umbilics, reconstruction

---

*E-mail: maekawa@ynu.ac.jp

1 Introduction

Aesthetically pleasing smooth surfaces are widely used in engineering applications, such as in the design of the bodies of automobiles, bullet trains, and ships, which have both functionality and attractive shape requirements. They are also used in the design of consumer products, furniture, and architecture containing aesthetic shapes.

However, the fabrication of these freeform shapes continues to pose new challenges. In this paper we introduce a novel method based on the use of orthogonal principal strips to fabricate such objects of which the boundary consists of freeform surfaces. Principal patches, which were introduced by Martin [1983], are patches whose sides are lines of curvature. These lines are curves on a surface whose tangent at each point is in a principal direction at that point. Because at each (non-umbilical) point there are two principal directions that are orthogonal, the lines of curvature form an orthogonal net of lines.

In engineering mechanics, it is well known that higher stresses occur in higher curvature regions [Fischer-Cripps 2007]. Since surfaces are bent the most along the lines of curvature, they indicate the directions for structural reinforcement. Furthermore, a non-orthogonal structure may be strong in a certain direction but weak in the direction orthogonal to that direction, whereas an orthogonal structure with lines of curvature can resist forces acting in arbitrary directions. Although it is a matter of taste, the lines of curvature generate an artistic pattern on the surface and can be used for paper and metal crafts and in architecture.

Pottmann et al. [2008] approximated the segmentation of the principal model into pieces of right circular cones, and into pieces of Dupin cyclides, which are known as principal strips, for applications in architecture and manufacturing. Although our definition of a principal strip is conceptually similar to that of [Pottmann et al. 2008], our mathematical formulation is different in two ways. First, they restricted the models of the principal strip to circular and conical models, which are convenient for panelization, whereas our formulation is based on the computation of lines of curvature on general freeform surfaces. The other major difference is that the method by Pottmann et al. [2008] relies on global optimization,

whereas our method is based on local optimization, which is much faster.

Our method entails mapping the four sides of each principal patch onto a plane by preserving the exact length and by connecting the sides approximately orthogonal to each other. We refer to the resulting quadrilaterals as principal quads (see Fig. 13 (b)). The distinguishing feature of our method is that the unfolding process is reduced to solving Newton's method with a single variable for each principal patch. Moreover, in the presence of umbilical points, or the involvement of boundary curves (generally not the curvature lines), the resulting patch could be an N-sided patch; however, our unfolding algorithm can easily be extended to accommodate this.

Once all the principal patches are unfolded onto the plane, we connect each of the principal quads one by one by aligning the equilength adjacent edges using translations and rotations. If the alignment occurs along the minimum principal curvature lines, the resulting strip is the maximum principal strip, and vice versa (see Fig. 13 (c), (d)). These strips can be assembled to form the original 3D shape in two ways. A pair of layers - one consisting of the maximum and the other consisting of the minimum principal strips - forms an orthogonal two-layered structure (see Fig. 13 (e), (f)). Alternatively, plain-weaving is employed, in which each maximum principal curvature strip is alternately placed over the minimum principal curvature strip, then underneath the next, and so on (see Fig. 13 (g), (h)). In both cases, each corresponding principal quad is positioned to overlap at the same location to reconstruct the 3D shape, one with maximum and the other with minimum principal curvature.

This paper makes the following contributions:

- We introduce a novel system to reconstruct freeform objects by using orthogonal two-layered principal strips and woven principal strips, which can be used for paper craft, sheet metal craft, and architectural models.

- Our unfolding algorithm reduces to finding the root of a single variable nonlinear equation, which is easy to implement and computationally fast. Moreover, it is able to compute situations in which umbilical points occur, i.e., where the orthogonal net of lines of curvature becomes singular.

- The fabricator does not need to adjust the amount of bending of the strips during construction, as the construction process of the orthogonal strips automatically reforms the design shape as if the shape is memorized provided that the strips possess bending rigidity.

2 Related work

Elber [1995] approximated a given NURBS surface by a set of developable strips within a prescribed tolerance. Each developable surface can then be unfolded, laid flat, and cut from a planar sheet such as paper or metal. However, he used a loose upper bound on the Hausdorf distance, resulting in a larger number of strips than is necessary [Massarwi et al. 2008]. Massarwi et al. [2008] introduced an algorithm to approximate a two-manifold 3D mesh with global error bounds by a set of developable surfaces consisting of generalized cylinders that are represented as a strip of triangles; nevertheless, the method cannot accommodate open models. Mitani and Suzuki [2004] introduced a method for generating unfolded papercraft patterns of rounded toy animal figures represented by a triangular mesh, by approximating the mesh model in terms of a set of continuous triangle strips with no internal vertices. The unfolded patterns were cut out by the cutting machine, and assembled using scotch tape. However, the parameters used in the algorithm were

determined by trial and error, and the method is not able to specify the approximation tolerance to the input mesh model. Shatz et al. [2006] introduced an algorithm for segmenting a mesh into developable approximations that can be used for paper crafting. These contributions [Elber 1995; Massarwi et al. 2008; Mitani and Suzuki 2004; Shatz et al. 2006] are limited in that they neither intend to generate orthogonality in the structure nor do they intend leaving room for adding strength to the structure.

Alliez et al.[2003] introduced anisotropic remeshing, where a curvature tensor field is estimated on a triangulated mesh, and then lines of curvature are traced on the mesh with a local density guided by the principal curvatures. The resulting mesh consists of anisotropic quads aligned to the principal directions, and triangles in isotropic regions. Marinov and Kobbelt [2004] extended the techniques by [Alliez et al. 2003] such that it does not rely on a global parameterization of the mesh and therefore is applicable to arbitrary genus surfaces. Tensor field topology was discussed in [Zhang et al. 2007].

Liu et al. [2006] showed how to approximate surfaces with meshes composed of planar quadrilaterals, so-called PQ meshes. They minimally perturbed the vertices of the input quadrilateral mesh into new positions such that the resulting mesh is a PQ mesh. The planarity condition of a quad face was expressed by enforcing the four angles enclosed by the edges of the quad to sum up to 2π. They used SQP to minimize the energy functionals defined by a fairing term and a term that forces the approximated surface to remain close to the original surface subject to the two planarity constraints.

Pottmann et al. [2008] studied the problem of covering a freeform surface by single curved panels, in particular conical and circular models, which semi-discretize the network of principal curvature lines. A D-strip model, which consists of B-spline D-strips whose control points are subjected to optimization by minimizing the target functional. The functional constituents measure closeness to the input B-spline surface, closeness to the boundary curve, developability of the strips, and fairness. Furthermore, they developed the concept of principal strips, which are optimized towards circularity or conicality from a curve network not too far from principal curvature lines. Optimization towards circular or conical strips makes use of geometric functionals, which penalize deviation from the appropriate angle equalities. In general, the construction of principal patches is difficult except for that of simple surfaces such as surfaces of revolution and Dupin cyclides whose sides are circular arcs. Kilian et al. [2008] introduced an optimization approach for the design and digital reconstruction of surfaces which can be generated by curved folding.

Akleman et al. [2009] introduced a method of creating plainweaving structures based on graph rotation systems over an arbitrary surface. They only rendered the computational models and did not reconstruct the physical models. Garg et al. [2014] presented a computational approach for designing freeform surfaces composed of woven wires arranged in a regular grid. They mapped the properties of wire meshes to the geometric model of Chebyshev nets. However, their approach requires preparation of the 3D scaffold such that the wire mesh can be bent into place.

Our work is inspired by the work of Pottmann and his coworkers and their use of principal strips [Pottmann et al. 2008]. While Pottmann et al. [2008] focused more on the panelization of freeform surfaces using surfaces of revolution and Dupin cyclides, our method focuses more on the orthogonality of the principal strips using general freeform surfaces. Moreover, our method relies on local optimization, which improves the computational efficiency. Most importantly, our study introduces novel methods to fabricate freeform objects in two ways using orthogonal principal strips.

3 Construction of orthogonal net

3.1 Lines of curvature

The lines of curvature are computed as the solution of the initial value problem of the coupled nonlinear differential equations [Maekawa et al. 1996]. The solution for a parametric surface $\mathbf{r}(u,v)=(x(u,v), y(u,v), z(u,v))$ generates a sequence of parameter pairs (u_k, v_k), which yield an ordered set of estimated points along the lines of curvature. Joo et al. [2014] introduced a method to compute the differential geometry properties of the lines of curvature, such as curvatures and torsions, and their higher-order derivatives. They completely defined them in terms of surface parameterization using the fact the discrete points are on the lines of curvature. This allows these differential geometric quantities to be evaluated exactly at the estimated points on the line of curvature, avoiding problems with discretization errors that arise when estimating higher-order derivatives by finite differences. Furthermore, Joo et al. [2014] flattened the lines of curvature onto a plane using the fact that curves on isometric surfaces have the same geodesic curvature at the corresponding points [Kreyszig 1968]. In other words, lines of curvature can be flattened onto a plane while preserving their length by integrating the Frenet-Serret formulae where the geodesic curvature of the lines of curvature is used for the curvature in the formulae. Because signs are involved in determining the magnitude of the principal curvatures, the magnitude is best judged without signs. Therefore, we ignore the sign when summing up the magnitude of each principal curvature at the uniform grid points in the parameter space, and the larger curvature is redefined as the maximum principal curvature and vice versa after all the lines of curvature have been computed. Hereafter, red and blue lines represent the maximum and minimum principal curvature, respectively.

3.2 Fairing of freeform surfaces

In our research we assume that the input freeform surfaces are smooth without any bumps. Therefore, in case the input surface is bumpy, we apply smoothing by using a surface fairing algorithm [Dietz 1998], as the lines of curvature are very sensitive to small irregularities in the shape. The fairing algorithm is set up to minimize the following objective function F_{air}

$$F_{air} = \sum_{k=0}^{N} |\mathbf{R}(u_k, v_k) - \mathbf{Q}_k|^2 + \lambda_f F_s \,, \tag{1}$$

where $\mathbf{Q}_k$, $k = 0, \ldots, N$ are the grid points on the input surface that are mapped from the evenly spaced points in the parametric domain. $\mathbf{R}(u_k, v_k)$, $k = 0, \ldots, N$ are the corresponding points on the approximating surface, and F_s is a smoothing term that enforces the fairness of the approximating surface, λ_f is a user-defined parameter that weighs the importance of the fairing factor. If λ_f is small, the resulting surface may not be smooth, yet it will be highly accurate. On the other hand, if λ_f is large, the resulting surface will be smooth, but it may not be highly accurate. Accordingly, it remains difficult to find an optimal λ_f; hence, we interactively increase λ_f until the lines of curvature are sufficiently smooth as shown in Fig. 2.

3.3 Distribution of lines of curvature

Our goal is to cover a 3D surface with a set of evenly-spaced lines of curvature. However, controlling the density of these lines is very difficult, yet this is one of the most important key points in achieving a successful principal strip model. We assume that the parametric speed of isoparametric curves in both the u and v directions
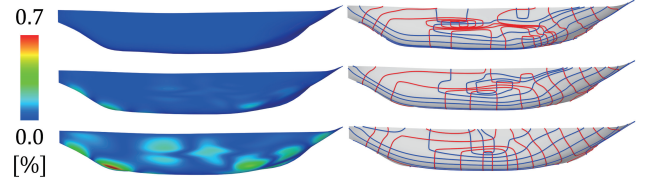

Figure 2: *Fairing process. Left column shows color-coded position errors (normalized by the length of the bounding box diagonal of the model), while right column depicts the lines of curvature of a ship hull. (top row) λ_f =0.0. (middle row) λ_f =0.01. (bottom row) λ_f =0.1.*

of the input surface does not vary significantly. Let us first define two sets of uniform lines $u = i\Delta u$ ($i = 1,\ldots$,M-1) and $v = j\Delta v$ ($j = 1,\ldots$,N-1) in the uv-parametric domain where Δu=1/M and Δv=1/N, respectively. The algorithm proceeds on a row-by-row basis followed by a column-by-column basis. The first seed point is placed at the center of the lowest v, i.e., (0.5, Δv). We compute the lines of curvature as an initial value problem in both directions (forward and backward) for one of the principal curvatures as illustrated in Fig. 3. We recursively divide the line $v = \Delta v$ into two by placing the seed point at the middle of the interval [(0.0, Δv), (0.5, Δv)], if there is no line of curvature passing through the interval, otherwise we do nothing until the divided intervals become sufficiently small. We move up to the next line $v = 2\Delta v$ and repeat the same process until we reach v = 1- Δv. Next, we apply the same algorithm to the other principal curvature. We repeat the same procedures to the u-direction in a column-by-column basis.
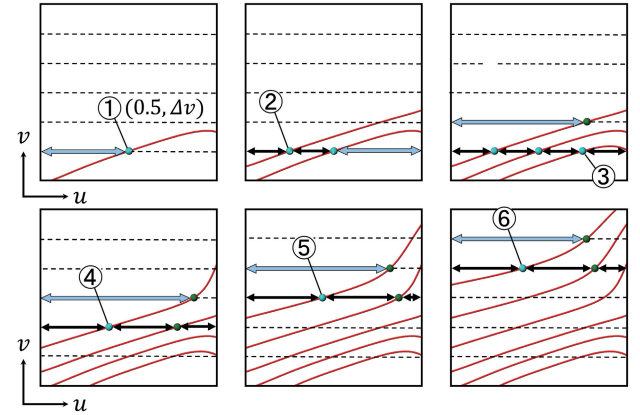

Figure 3: *Algorithm for distributing lines of curvature on the row-by-row basis. Left-to-right, top-to-bottom. Large intervals with blue arrows are subjected to assign a seed point at the middle of the interval, while small intervals with black arrows are already small enough to assign further seed points.*

The resulting lines of curvature have three patterns. The first pattern is a curve that runs from surface boundary to boundary where most of the lines of curvature fall into this category (see blue lines in Fig.4 (a) and (b)). The second pattern forms an internal loop that appears when the surface possesses reflectional symmetry and a pair of lemon-type umbilical points are located at symmetric positions as shown by the red lines in Fig.4 (a). The third pattern forms a nearly closed loop, which is observed when the second pattern loses its symmetry, which is illustrated by the red lines in Fig.4 (b).

We judge the third case by computing three distances between the last three ordered sets of estimated points along the lines of curva-
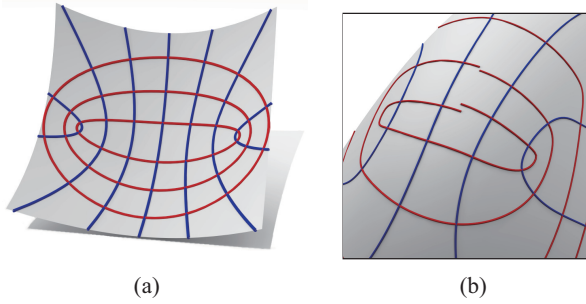
Figure 4: *Three patterns of lines of curvature. (a) Elliptic paraboloid having internal loops (red lines) and ones that run from surface boundary to boundary (blue lines). (b) Nearly closed loop (red lines).*

ture computed from the seed point in both directions (forward and backward), namely,

$$d_i = |\mathbf{r}(u_{N_F+i}, v_{N_F+i}) - \mathbf{r}(u_{N_B+i}, v_{N_B+i})|, \ i = -1, 0, 1 \ , \quad (2)$$

where N_F and N_B are the number of integrated points of the lines of curvature in the forward and backward directions, respectively (see Fig. 5 (a)). We terminate the Runge-Kutta integration if d_{-1}, d_0, and d_1 are all within the prescribed small distance, and d_0 has the smallest distance among the three distances. Then we close the gap between the two points (u_{N_F}, v_{N_F}) and (u_{N_B}, v_{N_B}) in the parameter space in order to generate a successful principal patch by distributing half of the signed gap to each integration point of the lines of curvature in the forward and backward directions along the difference vector between the two points with linearly increasing weights (zero at the seed point, and one at the closing point) such that the closing point is $\left(\frac{u_{N_F}+u_{N_B}}{2}, \frac{v_{N_F}+v_{N_B}}{2}\right)$ as shown in Fig. 5 (b).
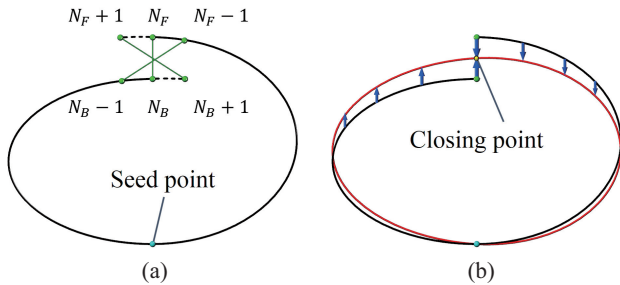

Figure 5: *Closing the gap for nearly closed loop. (a) Three distances between the last three ordered sets of estimated points along the lines of curvature computed from the seed point in both directions (forward and backward). (b) Closing the gaps in parameter domain.*

3.4 Editing of orthogonal net

In Section 3.3 we introduced an algorithm to automatically cover a 3D surface with a set of evenly spaced lines of curvature. However, dense lines of curvature tend to occur in high-curvature regions, whereas sparse lines appear in low-curvature regions. Accordingly, we introduce a graphical user interface (GUI) to enable users to adjust the density of the final orthogonal net of principal curvature lines by applying ʹaddʹ and ʹdeleteʹ functions. Once the user selects one of the principal curvatures, they click a seed point

on the surface, where the lines of curvature are sparse, to add new lines of curvature in both the forward and backward directions. The ʹdeleteʹ function deletes any existing lines of curvature by clicking any points on the lines of curvature the user wishes to delete (see Fig. 6).
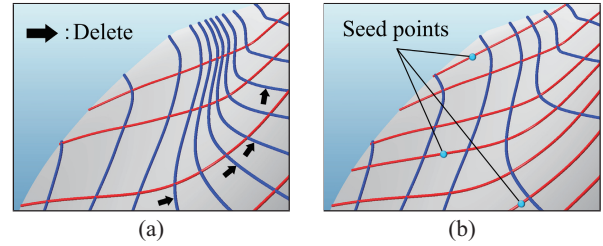

Figure 6: *Adjusting density of lines of curvature. (a) Before editing. (b) After editing.*

4 Unfolding of principal patches

In general, principal patches are not developable, therefore they must be approximated to be unfolded onto a plane. In this section we present the results of our study to unfold principal patches one by one to generate principal quads based on the fact that a line of curvature of a surface can be flattened onto a plane preserving its length using the geodesic curvature of the line of curvature.
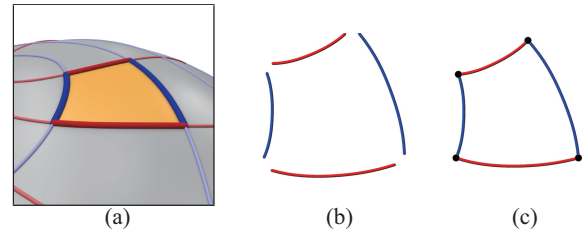
4.1 Four-sided patch

Figure 7: *Unfolding of principal patch. (a) Four sides of principal patch. (b) Unfolded four sides. (c) Principal quad.*

We flatten the four sides of each principal patch onto a plane. At this point, the four edge curves lie arbitrarily on the plane as shown in Fig. 7. Therefore, we transform each curve to form a topologically rectangular shape with the same order in 3D space as illustrated in Fig. 7. However, when the four lines are linked, the four angles between the lines connecting the two end points of the two adjacent curves are not decided. Because we are assuming isometry between the principal patch and the unfolded principal quad, all four angles of the quad are required to approximate right angles. Let us denote the vertices and the corresponding vertex angles of the four curves as $\mathbf{V}_i, \omega_i \ (i = 0, \cdots, 3)$ as illustrated in Fig. 8 (a). If we denote the four vertex angles of the quadrangle as $\Psi_i \ (i = 0, \ldots, 3)$, and denote the angles between the flattened curve and the straight line connecting the two end points of the curve to be $\delta_{i,1}, \delta_{i,2} \ (i = 0, \ldots, 3)$, where the sign is positive when the angle is outside of the quadrangle, and vice versa. We note that $\delta_{i,1}, \delta_{i,2} \ (i = 0, \ldots, 3)$ are inherited from the original principal patches, thus we keep them unchanged. Referring to Fig. 8 (a) and (b), we have

$$\omega_i = \Psi_i + \delta_{i,1} + \delta_{i,2}, \quad i = 0, \cdots, 3 \ . \quad (3)$$

We connect $\mathbf{V_1}$ and $\mathbf{V_3}$, and define $\theta_i, \alpha_i, \beta_i$ (i=0,1) as illustrated in Fig. 8 (c). It seems redundant to define $\theta_i, \alpha_i, \beta_i$ at this moment;

however, they are useful when the algorithm is extended to an N-sided patch. If we fix θ_0, the angles $\theta_1, \alpha_i, \beta_i$ $(i = 0, 1)$ can be determined using the cosine theorem of trigonometry and the fact that angles in the triangle add to π:

$$e^2 = a^2 + d^2 - 2ad\cos\theta_0 , \tag{4}$$

$$\alpha_0 = \cos^{-1}\left(\frac{a^2 + e^2 - d^2}{2ae}\right) , \tag{5}$$

$$\beta_0 = \pi - \theta_0 - \alpha_0 , \tag{6}$$

$$\theta_1 = \cos^{-1}\left(\frac{b^2 + e^2 - c^2}{2be}\right) , \tag{7}$$

$$\alpha_1 = \cos^{-1}\left(\frac{b^2 + c^2 - e^2}{2bc}\right) , \tag{8}$$

$$\beta_1 = \pi - \theta_1 - \alpha_1 . \tag{9}$$

Accordingly, we have

$$\Psi_0 = \theta_0, \ \Psi_1 = \theta_1 + \alpha_0, \ \Psi_2 = \alpha_1, \ \Psi_3 = \beta_0 + \beta_1 . \tag{10}$$

Because the corresponding vertex angles of the principal patches are all equal to $\frac{\pi}{2}$, we can set up an optimization problem to minimize the objective function F_{ang}:

$$F_{ang}(\theta_0) = \sum_{i=0}^{3} \left(\omega_i(\theta_0) - \frac{\pi}{2}\right)^2 , \tag{11}$$

which leads to finding a root of the single variable nonlinear equation with respect to θ_0:

$$\dot{F}_{ang}(\theta_0) = \sum_{i=0}^{3} 2\left(\omega_i(\theta_0) - \frac{\pi}{2}\right)\dot{\omega}_i(\theta_0) = 0 . \tag{12}$$

where $\cdot$ denotes the derivative with respect to θ_0. Equation (12) can easily be solved by Newton's method. We list the first and second derivatives required for Newton's method in Appendix A.1. The initial value for Newton's method is determined as follows: If one of the four angles of the quadrangle Ψ_i is fixed, the rest of the three angles are determined automatically, therefore, we first calculate Ψ_i using (3) assuming $\omega_i = \frac{\pi}{2}$ for $i = 0, \ldots, 3$, and for each Ψ_i, we compute Ψ_j $(j \neq i)$ using equations similar to (4) $\sim$ (9). Finally, we average the four estimated Ψ_0 at $\mathbf{V_0}$ and set $\theta_0 = \Psi_0$. Once θ_0 is found, the remaining angles Ψ_i and ω_i $(i = 0, \ldots, 3)$ are determined from (10) and (3), respectively.
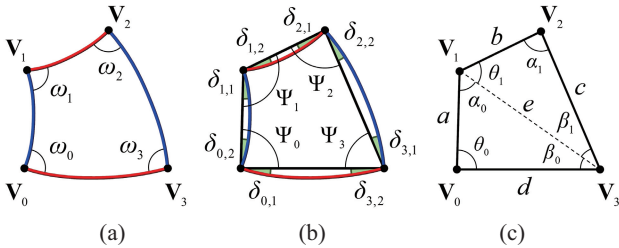

(a)   (b)   (c)

Figure 8: *Four-sided patch. (a) Unfolded four-sided patch, (b) Four-sided polygon generated by connecting the four corners of (a). (c) Definitions of $\theta_i, \alpha_i, \beta_i$.*

In general, surface boundary curves are isoparametric curves and are not lines of curvature. Therefore, if the boundary curves are included in the patch, geodesic curvature for isoparametric lines should be used for the flattening process, and the angles between isoparametric lines and lines of curvature, and the angles between two isoparametric curves must be used instead of $\pi/2$ as well.

In case the lines of curvature pass near umbilical points or the patch includes surface boundary curves (non-curvature line), the resulting patch will generally not be a four-sided patch; instead, it is more likely to be an N-sided patch. In the following sections, we show that the method for processing four-sided patches can be extended to N-sided patches.

4.2 Two- and three-sided patches

In case the lines of curvature pass near a lemon-type umbilical point, a two-sided principal patch is generated as shown in Fig. 9. In such cases, we flatten the two curves and split them in the middle to form a four-sided patch, since the lines connecting the two end points of the two curves, in general,

Figure 9: *Two- and three-sided patches.*

do not have the same length. The angle between the two split curves is π. Note that in the process of unfolding the patches which share these two curves, we use the same split two curves for the quads for compatibility reasons. A three-sided patch often occurs near the surface boundary curves, and because the three edges of the flattened curves determine a unique triangle, there is no need for optimization.
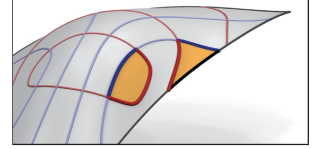
4.3 N-sided patch ($N \geq 5$)

Similar to the four-sided patch case, we denote the N vertex angles of the N-sided polygon as Ψ_i, $(i = 0, \cdots, N - 1)$, and the two angles between each flattened curve and the straight line connecting the two end points of the curve to be $\delta_{i,1}, \delta_{i,2}, i = 0, \cdots, N - 1$, where the sign is positive when the angle is outside of the quadrangle, and vice versa. Our task is to determine the angles $\omega_i(i = 0, \cdots, N - 1)$.

We construct N-2 triangles by connecting $\mathbf{V}_i$ $(i = 1, \cdots, N - 3)$ to $\mathbf{V}_{N-1}$ as illustrated in Fig. 10. We define $\theta_i, \alpha_i, \beta_i$, $(i = 0, \cdots, N - 3)$ as in Fig. 10, where α_i, β_i can be expressed as functions of $\boldsymbol{\theta} = (\theta_0, \cdots, \theta_{N-4})$ using the cosine theorem of trigonometry and the fact that the angles in the triangle add to π. Note that the angles in the N-2-th triangle $\theta_{N-3}, \alpha_{N-3}, \beta_{N-3}$ can be automatically computed from the angles of the first $N - 3$ triangles. The vertex angles of the N-sided polygon Ψ_i $(i = 0, \cdots, N - 1)$ can be obtained as follows:
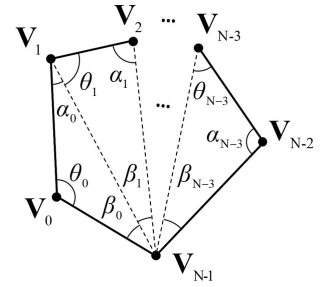

Figure 10: *N-sided polygon.*

$$\Psi_0 = \theta_0 , \tag{13}$$

$$\Psi_i = \theta_i + \alpha_{i-1}, \quad i = 1, \cdots, N - 3 , \tag{14}$$

$$\Psi_{N-2} = \alpha_{N-3} , \tag{15}$$

$$\Psi_{N-1} = \sum_{j=0}^{N-3} \beta_j , \tag{16}$$

and hence

$$\omega_i \;=\; \Psi_i + \delta_{i,1} + \delta_{i,2}, \quad i = 0, \cdots, N-1 . \quad (17)$$

It is observed that concave polygons occur along the boundaries. We accommodated such cases by introducing signs in θ_i. Suppose vertices $\mathbf{V}_i$ are numbered in increasing numerical sequence in a clockwise direction, θ_i is positive, if the line segment $\mathbf{V}_i\mathbf{V}_{i+1}$ is in the clockwise direction of $\mathbf{V}_i\mathbf{V}_{N-1}$ with respect to $\mathbf{V}_i$, and vice versa. The concave vertex occurs at $\mathbf{V}_i$ if $\Psi_i > \pi$, which can be further divided into the following two cases:

If $\theta_i < 0$, we reset α_i and β_i as follows:

$$\alpha_i \;=\; 2\pi - \alpha_i , \quad (18)$$
$$\beta_i \;=\; -\beta_i . \quad (19)$$

Similarly if $\theta_i > \pi$, we reset

$$\alpha_i \;=\; -\alpha_i , \quad (20)$$
$$\beta_i \;=\; -\beta_i . \quad (21)$$

Fig. 11 (a) illustrates the case when $\theta_1 < 0$, and $\mathbf{V}_2$ is a concave vertex, whereas Fig. 11 (b) shows the case when $\theta_0 > \pi$, and $\mathbf{V}_0$ is a concave vertex.
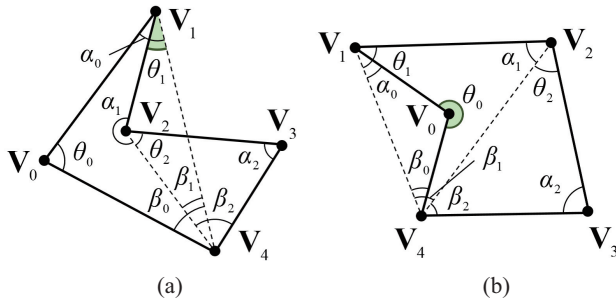

Figure 11: *Concave polygons: (a) Pentagon with $\theta_1 < 0$. (b) Pentagon with $\theta_0 > \pi$.*

The objective function for minimization can be written as in the four-sided patch case:

$$F_{ang}(\boldsymbol{\theta}) = \sum_{i=0}^{N-1} \left(\omega_i(\boldsymbol{\theta}) - \frac{\pi}{2} \right)^2 . \quad (22)$$

The objective function (22) can be minimized by finding the roots of $\nabla F_{ang}(\boldsymbol{\theta}) = \mathbf{0}$. The iterative scheme for Newton's method can be described as follows:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - [\mathbf{H}F_{ang}(\boldsymbol{\theta}_k)]^{-1}\nabla F_{ang}(\boldsymbol{\theta}_k), \quad (23)$$

where $[\mathbf{H}F_{ang}(\boldsymbol{\theta}_k)]^{-1}$ is the inverse of a Hessian matrix, and the subscripts k and $k+1$ denote the k-th and $k+1$-th iterations. The initial values for Newton's scheme $\boldsymbol{\theta}_0$ can be determined similarly to the four-sided patch case.

4.4 Errors in angles

In this section we examine angle errors induced in the process of unfolding principal patches onto the plane using a model of the nose of a bullet train. Fig. 12 shows patch based color-coded errors mapped onto the bullet train consisting of 51, 105, and 305 principal

patches, respectively. This indicates that the maximum angle error decreases as the number of principal patches increases.
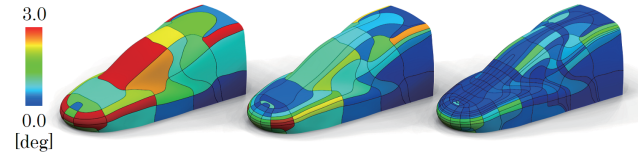

Figure 12: *Color-coded angle errors mapped onto the nose of a bullet train with 51 patches (left), 105 patches (middle), and 305 patches (right).*

5 Fabrication of objects

Once all the principal patches are unfolded onto the plane, we connect the principal quads one by one by aligning the equi-length adjacent edges and by applying translations and rotations. Fig. 13 (a) illustrates an ellipsoid $\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$ of a=80mm, b=60mm, c=40mm, which contains four lemon-type umbilical at $\left(\pm a\sqrt{\frac{a^2-b^2}{a^2-c^2}}, \; 0, \; \pm c\sqrt{\frac{b^2-c^2}{a^2-c^2}} \right)$ consisting of four two-sided and 178 four-sided principal patches. We can unfold these principal patches onto the plane to generate principal quads as shown in Fig. 13 (b). If we align the quads along the minimum principal curvature, the resulting strip is the maximum principal strip as shown in Fig. 13 (c), whereas if we align quads along the maximum principal curvature, the resulting strip is the minimum principal strip as shown in Fig. 13 (d).

Using this orthogonal set of strips, we are able to reconstruct the original 3D shape of the input object in two ways. The first is to form two-layered principal strips, whereas the other involves fabricating the woven principal strips.

5.1 Two-layered principal strips

The two-layered principal strip structure consists of a layer of maximum principal strips and a layer of minimum principal strips as shown in Fig. 13 (e) and (f). We emphasize the lines of curvature by using yellow and golden yellow colors for the maximum principal strips, whereas blue and light blue colors are used for the minimum principal strips. We adhere a pair of quads that have exactly the same shape, but different colors.

This unit structure can be stacked to form a multi-layered shell structure. Up to now, we have not taken into account the material thickness; however, this assumption may not be true if the number of layers becomes large. Suppose the strip thickness is t, then the lines of curvature must be computed on the offset surface $\hat{\mathbf{r}}(u,v)=\mathbf{r}(u,v) + t\mathbf{N}(u,v)$, where $\mathbf{N}(u,v)$ is the unit surface normal vector. It is well known that the lines of curvature of offset surfaces correspond to each other [Willmore 1959]. In other words, if the lines of curvature are computed on the reference surface $\mathbf{r}(u_k, v_k)$, then the corresponding lines of curvature on the offset surfaces are $\hat{\mathbf{r}}(u_k, v_k)$. However, the geodesic curvature along the lines of curvature on offsets must be computed by applying the algorithm by [Joo et al. 2014] to $\hat{\mathbf{r}}(u,v)$.

5.2 Woven principal strips

Woven principal strips are constructed by using an alternating approach to pass each maximum principal strip over the minimum principal strip, then under the next, and so on, and vice
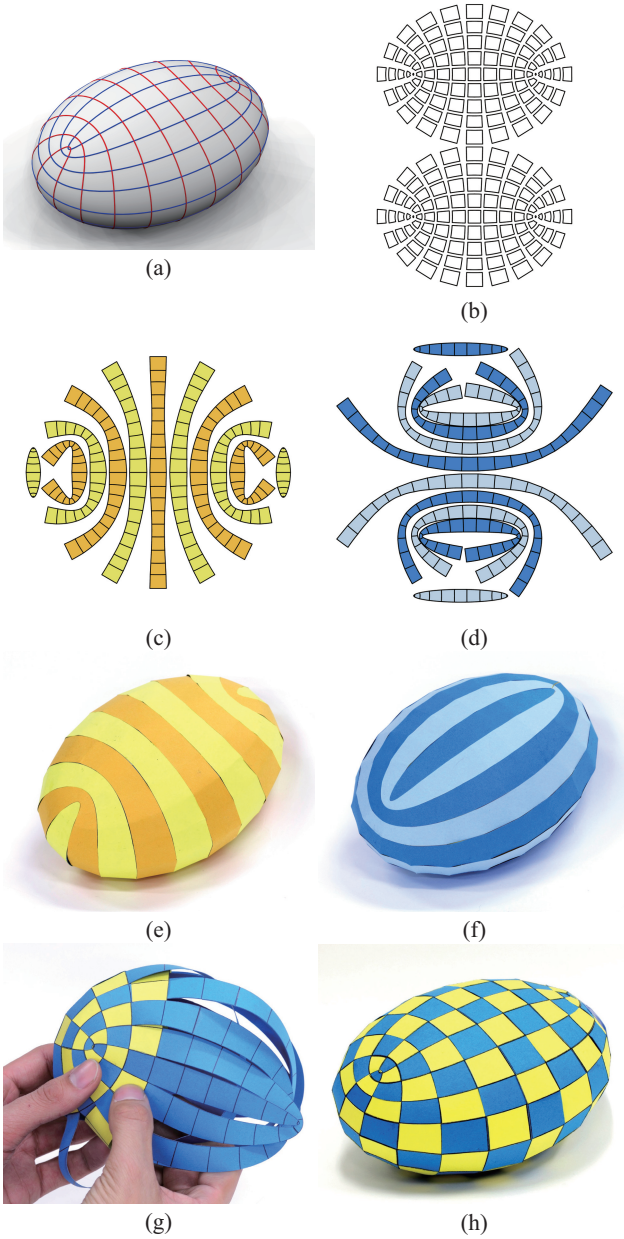
versa. A rigid structure is ensured by adhering a pair of principal quads. Fig. 13 (g) illustrates the process of weaving the strips, and the resulting woven principal strips is depicted in Fig. 13 (h). The algorithm performs well for thin strips; however, as the strip becomes thicker, it starts to overshoot at the borders due to the bending rigidity as shown in Fig. 14, and hence the strip needs to be elongated as in the figure below (white portions). According to the beam theory, the length of the neutral axis of the strip spanned by the central angle α with the radius of curvature ρ is $\rho\alpha$, and hence the length of the fiber at the surface of the strip of thickness t is $(\rho + \frac{t}{2})\alpha$ (see Fig. 14). Therefore, the strain in the surface fiber is $\epsilon = \frac{(\rho + \frac{t}{2} - \rho)\alpha}{\rho\alpha} = \frac{t}{2\rho}$. Referring to Fig. 14 the amount of elongation necessary for the quad for the plain-weaving procedure can be estimated as $\delta = 4\rho(\alpha - \sin\alpha) = \frac{2t}{\epsilon}(\alpha - \sin\alpha)$. In this paper we use $\alpha = \frac{\pi}{4}$, and the strain ϵ corresponding to the yield stress, which depends on the material used.
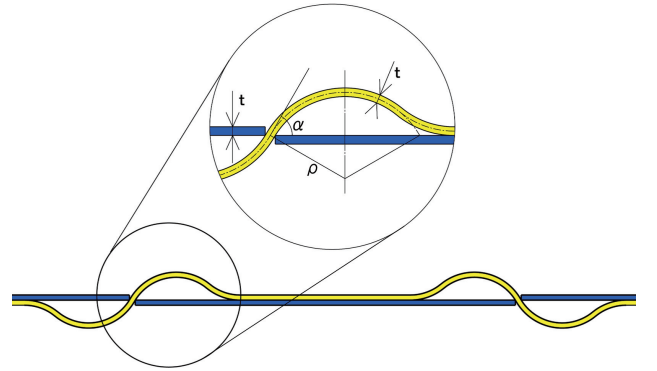

Figure 14: *Amount of elongation necessary in plain-weaving for a quad with thickness t.*

5.3 Accuracy evaluation of fabricated objects

In this section, we examine the accuracy of the fabricated models by first comparing the differences in the surface areas between the unfolded quads and their corresponding CAD models and then evaluating the distances between the laser-scanned point clouds of the objects and their corresponding CAD models. Fig. 15 shows patch-based color-coded surface-area errors mapped onto an ellipsoid and the nose of a bullet train consisting of 182 and 177 principal patches, respectively. The errors are normalized by the areas of the corresponding principal patches. Table 1 lists the entire surface area errors E_{Area} of the two models. Although there are a few principal quads that have a surface-area error near 10% (see Fig. 15), the entire surface-area errors for the ellipsoid and bullet train are merely 1.15% and 0.666%, respectively (see Table 1). Large errors occur in regions having large Gaussian curvatures (see Fig. 16) and at the three-sided patches where the three edges of the flattened curves automatically determine a unique triangle.

Next, we examine the accuracy of fabricated models by laser-scanning them and comparing the point clouds with their corresponding CAD models. We use a 3D laser scanner (Roland PICZA LPX-60) to inspect the accuracy of the fabricated objects. Since the point clouds of the laser-scanned fabricated models and CAD models are constructed in different coordinate systems, rigid-body
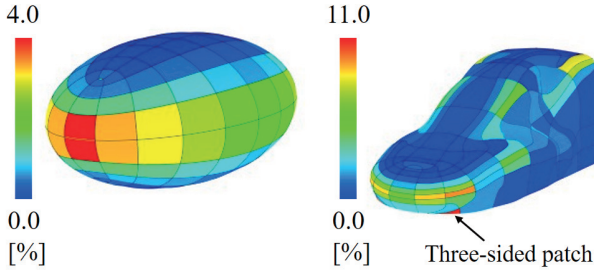

Figure 13: *(a)Ellipsoid with evenly distributed lines of curvature. (b) Unfolded principal quads. (c)Maximum principal strips. (d) Minimum principal strips. (e) Two-layered principal strips with maximum principal strips on the visible side. (f) Two-layered principal strips with minimum principal strips on the visible side. (g) Plain-weaving of the principal strips. (h) Woven principal strips.*

Figure 15: *Color-coded surface-area errors mapped onto an ellipsoid consisting of 182 patches and the nose of a bullet train consisting of 177 patches.*
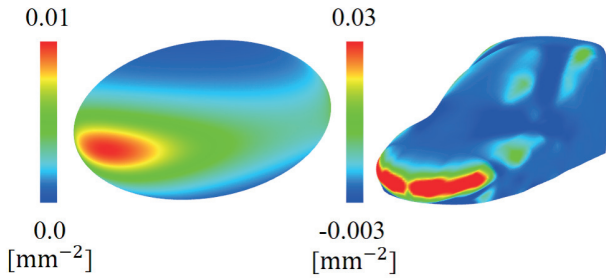

Figure 16: *Gaussian curvature maps of the ellipsoid and bullet train models.*

translations and rotations must be iteratively applied to the point clouds in order to bring them into the closest correspondence with the CAD models [Hirano et al. 2009]. This iterative procedure is called localization. Once the point cloud and the corresponding CAD model are localized, we compute the distance between the point cloud and the CAD model, as shown in Fig. 17. The maximum and average distance errors normalized by the length of the bounding box diagonal of the model, $E_D Max$ and $E_D Ave$, are listed in Table 1. The results for the surface-area and distance errors clearly indicate that the fabrication of freeform objects by principal strips is quite accurate, which justify that our unfolding algorithm based on local minimization is sufficiently accurate for 3D fabrication.

The following is an intuitive explanation. The smoothed surface is tessellated into an orthogonal net of lines of curvature in a global manner yielding a set of quadrilateral principal patches. At this point, the error is zero except for the numerical integration error for computing the lines of curvature. We then map the four sides of each principal patch onto a plane by preserving the exact length and by connecting the sides approximately orthogonal to each other in a local manner. Even though the flattening process of principal patches is computed locally, the original principal patches are segmented globally, and hence we believe that the reconstruction to 3D will generate a surface close to the original one.

6 Results

In this section, we demonstrate the effectiveness of our algorithm by applying it to four models: an automobile consisting of hood, front and rear wheelhouses, roof, a bullet train, a hyperbolic paraboloid, and a Dupin cyclide. All the computations were performed on a PC with a core i7-5820-K 3.30 GHz processor and 32 GB of RAM.
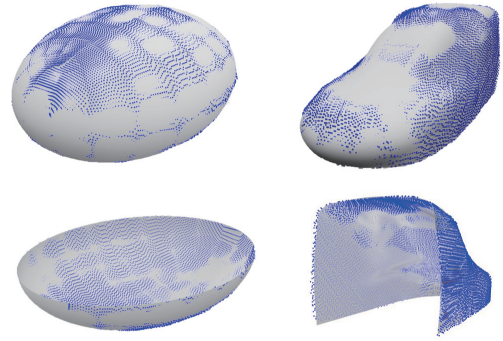
Figure 17: *Localized laser-scanned point cloud of the ellipsoid and bullet train models with respect to their CAD models.*

Table 1: *Surface-area errors normalized by the corresponding (faired) input surface area and the distance errors normalized by the length of the bounding box diagonal of the model.*

Model	# of principal patches	E_{Area} [%]	$E_D Max$ [%]	$E_D Ave$ [%]
Ellipsoid	182	1.15	1.12	0.188
Bullet train	177	0.666	1.92	0.323

6.1 Computational time

The computational time for the unfolding process with respect to the number of principal patches is plotted in Fig. 18. We can observe that the computational time is proportional to the number of principal patches. This result is obvious because the unfolding process depends on solving Newton's method for a single variable for each principal patch, which on average converges within five iterations; therefore, the process can be considered as requiring constant time. We note here that we need to solve Newton's method for (N-3) variables for an N-sided patch; however, the occurrence of such patches is small. Although the surface models and computers are different, the run times of our unfolding process are in milliseconds, whereas the computational time of [Pottmann et al. 2008], which uses global minimization, is in seconds. Recently, Tang [2016] further reduced the computational time so that it is fast enough for interactive modeling by adopting the method of [Tang et al. 2014] for solving the constraint equations related to developability. However, their speed is still in the order of seconds, while our method is in milliseconds.
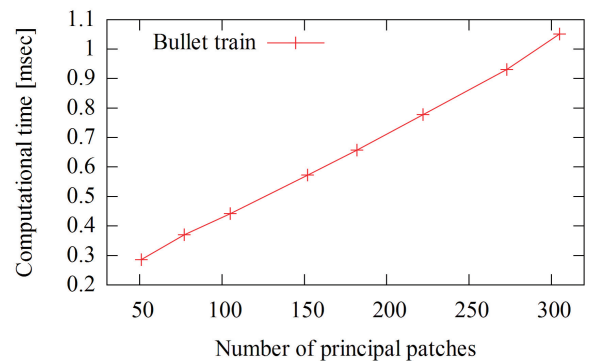

Figure 18: *Computational time for the unfolding process with respect to the number of principal patches.*
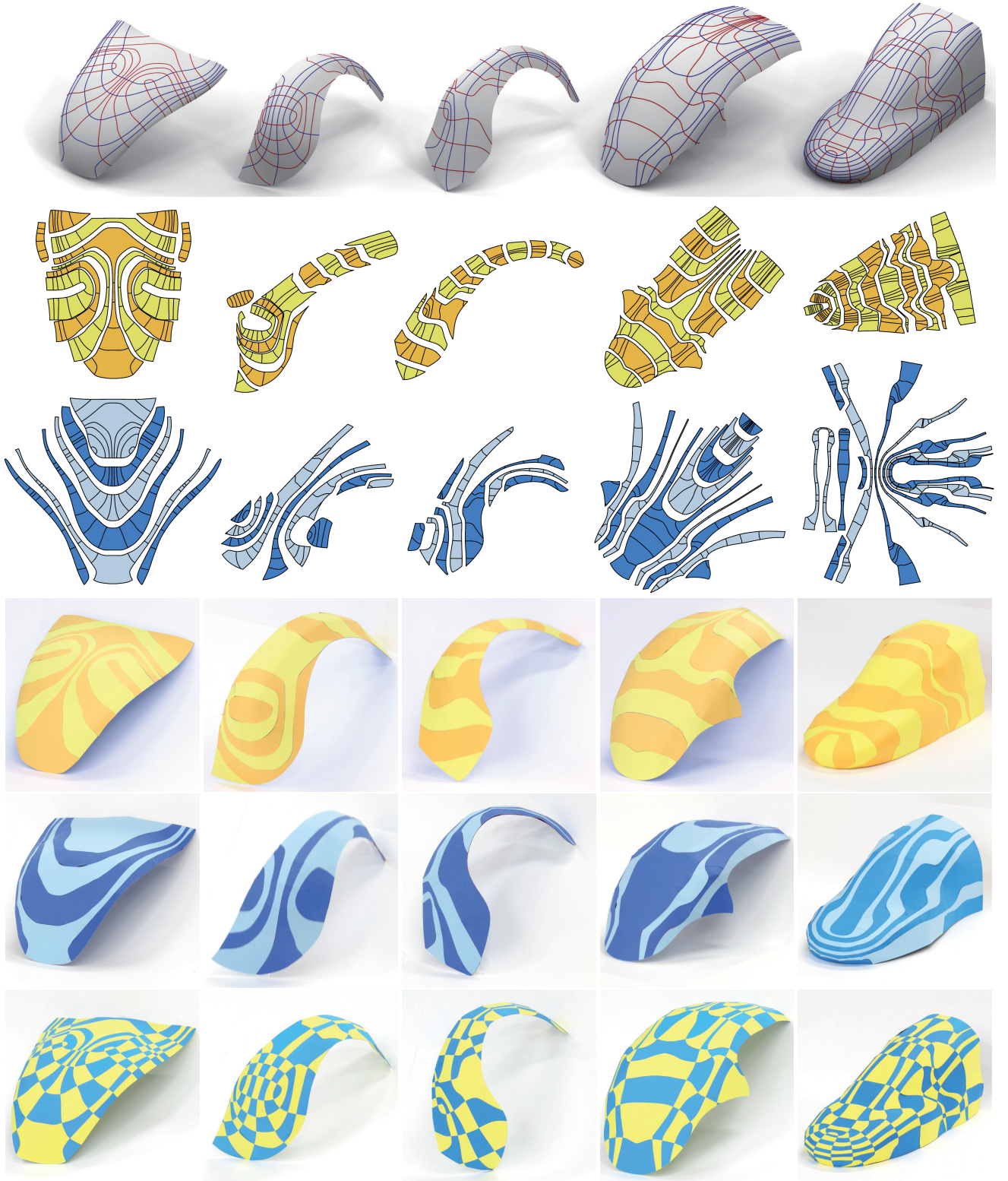
Figure 19: *Fabrication of freeform objects by principal strips for five different models. Left-to-right: Hood, front wheelhouse, rear wheelhouse, roof, and the nose of the bullet train. Top-to-bottom: Faired model with lines of curvature, unfolded maximum principal strips, unfolded minimum principal strips, photos of reconstructed two-layered principal strips (maximum principal strips on the visible side), photos of reconstructed two-layered principal strips (minimum principal strips on the visible side), and photos of woven principal strips.*

Model	Surface fairing			Unfolding of principal strips				
	λ_f	$E_D Ave$ [%]	$E_D Max$ [%]	# of principal patches	Comp. time [msec]	$E_{Ang} Ave$ [deg]	$E_{Ang} Max$ [deg]	E_{Area} [%]
Hood	0.01	0.0237	0.134	118	0.517	0.399	2.95	0.664
Front wheelhouse	0.05	0.260	2.54	73	0.552	0.540	2.11	0.928
Rear wheelhouse	0.05	0.0435	0.256	58	0.407	0.418	2.63	0.906
Roof	0.04	0.142	0.992	100	0.506	0.362	3.47	1.13
Bullet train	0.004	0.155	1.16	177	0.662	0.665	3.70	0.666
Ellipsoid	-	-	-	182	0.622	1.02	3.49	1.15
Hyperbolic paraboloid	-	-	-	112	0.778	0.114	0.751	0.124
Dupin cyclide	-	-	-	224	0.820	1.60	3.58	1.87

Table 2: *Computational results. We use the following abbreviations: $E_{Dist}Ave$ and $E_{Dist}Max$ are the average and maximum distance errors, respectively, resulting from the fairing process, $E_{Ang}Ave$ and $E_{Ang}Max$ are the average and maximum angle errors, respectively, induced during the unfolding process. E_{Area} is the surface-area error with respect to the corresponding surface area of the faired input surface.*

6.2 Applications

Papercraft

Fig. 19 depicts the results of our method applied to various aesthetically pleasing smooth engineering surfaces of genus zero, namely, surfaces of an automobile such as the hood, front wheelhouse, rear wheelhouse, roof, and the nose of the bullet train. The first row shows the faired surface models with evenly distributed lines of curvature. The second and third rows depict the unfolded principal quads aligned along minimum/maximum principal lines of curvature yielding maximum/minimum principal strips, respectively. The fourth and fifth rows show the two-layered maximum/minimum principal strips on the visible side, respectively. The bottom row shows the woven principal strips.

Fig. 20 shows a ring Dupin cyclide, which is a surface of genus one, having parameters a=100, b=95, and μ=60. The parameters are defined in [Pratt 1990]. Fig. 20 (a) depicts the model with lines of curvature, (b) is a photo of woven principal strips, (c) is an exterior view of the architectural design using the Dupin cyclide.

The woven principal strips of the models clearly show that the net of the reconstructed principal strips is orthogonal, and provide an artistic touch to the appearance of the objects. It is possible to use these orthogonal strips as guidelines for reinforcement (see Figs. 20 (c) and 23 (b)), because the surfaces are bent the most along these principal strips. Furthermore, N-sided patches effectively handle the singularity of the orthogonal net due to the existence of the umbilical points.

In Table 2, we summarize the distance errors E_D, which are normalized by the length of the bounding box diagonal of the model generated during the fairing process, the angle errors E_{Ang}, the surface-area errors with respect to the corresponding surface areas of the faired input surfaces E_{Area} induced during the unfolding process, and the run times. Since the lengths of the sides of the quads are exact and the errors in the angles and areas are small, we can say that our unfolding algorithm based on local optimization is quite accurate.

Plastic plates and sheet metals

Thus far we have fabricated objects with paper, but in this section, we extend our method to other materials, namely plastics, aluminum, and brass. Fig. 21 shows a hyperbolic paraboloid reconstructed by woven plastic principal strips for architectural design. As the thickness of the plastic is 0.35 mm, which is not negligibly thin, the stretch is taken into account for every principal quad.
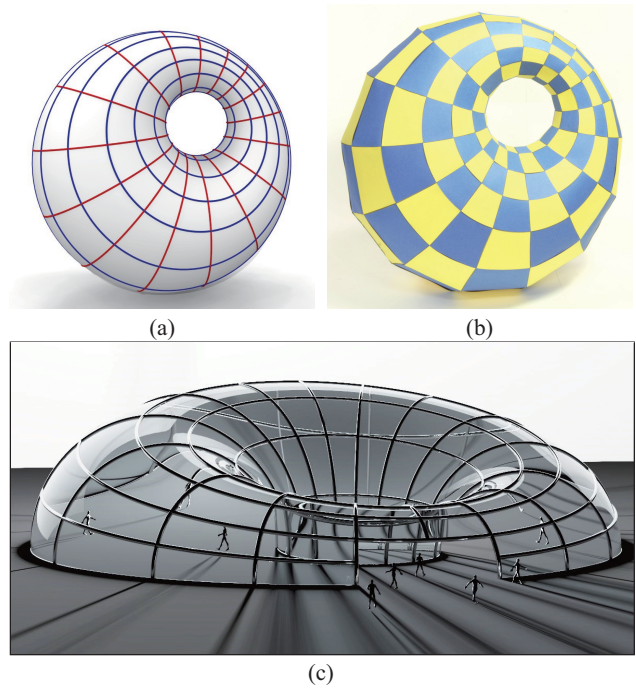

(a)　　　　　(b)

(c)

Figure 20: *A Dupin cyclide. (a) The model with lines of curvature. (b) Photo of fabricated Dupin cyclide. (c) Exterior view of the architectural design.*

Figure 21: *Woven principal strips for plastic plate. Hyperbolic paraboloid is used for architectural design with plastic plates.*

Fig. 22 depicts an example of sheet metal craft consisting of aluminum and brass sheets for the fabrication of the hood model in Fig. 19. The thickness of both sheet metals are 0.1 mm, and hence the stretch is taken into account.


Figure 22: *Woven principal strips for sheet metals. Hood of the automobile is fabricated by sheet metals of principal strips.*
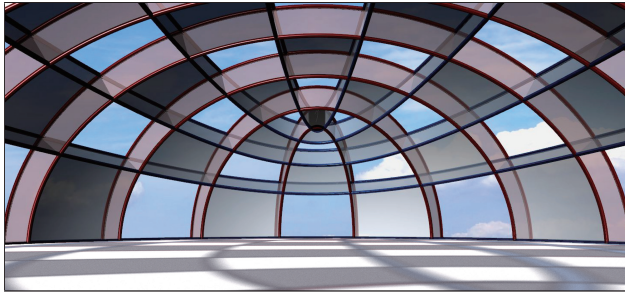
Reinforcement along lines of curvature

One of the advantageous properties of orthogonal principal strips is that they indicate directions for reinforcement of the structure as the surfaces are bent the most along the lines of curvature. Because the principal strips are almost straight in the hyperbolic paraboloid model (see Fig. 21), carbon fiber reinforced plastics (CFRPs) based on a two-layered structure can be effectively used for fabrication.

If the stiffeners or frames are added along the lines of curvature, as illustrated in Fig. 23 (a) and (b), the resulting structure would be reinforced.
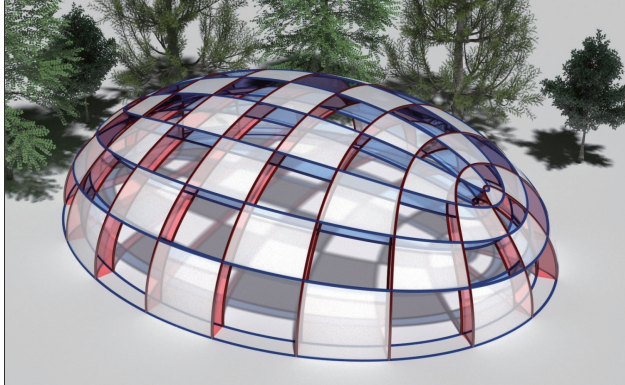
(a)

(b)

Figure 23: *Reinforcement along lines of curvature. (a) Exterior view of the architectural design using ellipsoid. (b) Interior view showing the stiffener/frame along lines of curvature.*

7 Conclusion

We have introduced novel methods to reconstruct objects of which the boundary consists of freeform surfaces by using orthogonal two-layered principal strips and woven principal strips. The unfolding algorithm reduces to finding the root of a single variable non-linear equation, which is easy to implement and computationally efficient. Moreover, it is capable of processing cases with umbilical points where the orthogonal net of lines of curvature becomes singular.

As demonstrated by way of examples, the new concept of using orthogonal principal strips provides an artistic touch to the appearance of objects, and shows guide lines for possible reinforcement. Our unfolding algorithm is computationally efficient, as it depends on solving Newton's method for a single variable for each principal patch. Nevertheless, our algorithm accurately reconstructs input freeform objects. Moreover, the fabricator does not need to adjust the amount of bending of the strips during construction, as the construction process of the orthogonal strips automatically reforms the design shape as if the shape is memorized. We have applied our method to various freeform surfaces and materials and demonstrated the effectiveness of our algorithm.

Limitations and future work

The nature of lines of curvature is such that they tend to be dense in high-curvature and sparse in low-curvature regions. Therefore, the current algorithm is unable to accommodate surfaces with high variations in curvature. In future we plan to apply our method to forming sheets of CFRPs.

Acknowledgements

A.1 Derivatives required in Newton's method for a four-sided patch

First derivatives:

$$\dot{\omega}_0 = \dot{\Psi}_0 = 1 \tag{A-1}$$

$$\dot{\omega}_1 = \dot{\Psi}_1 = \dot{\theta}_1 + \dot{\alpha}_0 \tag{A-2}$$

$$\dot{\omega}_2 = \dot{\Psi}_2 = \dot{\alpha}_1 \tag{A-3}$$

$$\dot{\omega}_3 = \dot{\Psi}_3 = -(1 + \dot{\alpha}_0 + \dot{\theta}_1 + \dot{\alpha}_1) \tag{A-4}$$

$$\dot{\alpha}_0 = -\frac{\dot{\cos}\alpha_0}{\sqrt{\sin^2\alpha_0}} \tag{A-5}$$

$$\dot{\theta}_1 = -\frac{\dot{\cos}\theta_1}{\sqrt{\sin^2\theta_1}} \tag{A-6}$$

$$\dot{\alpha}_1 = -\frac{\dot{\cos}\alpha_1}{\sqrt{\sin^2\alpha_1}} \tag{A-7}$$

$$\dot{e} = ade^{-1}\sin\theta_0 \tag{A-8}$$

$$\dot{\cos}\alpha_0 = -\frac{\dot{e}(a^2 - e^2 - d^2)}{2ae^2} \tag{A-9}$$

$$\dot{\cos}\theta_1 = -\frac{\dot{e}(b^2 - e^2 - c^2)}{2be^2} \tag{A-10}$$

$$\dot{\cos}\alpha_1 = -\frac{e\dot{e}}{bc} = -\frac{ad}{bc}\sin\theta_0 \tag{A-11}$$

Second derivatives:

$$\ddot{\omega}_0 = \ddot{\Psi}_0 = 0 \qquad \text{(A-12)}$$

$$\ddot{\omega}_1 = \ddot{\Psi}_1 = \ddot{\theta}_1 + \ddot{\alpha}_0 \qquad \text{(A-13)}$$

$$\ddot{\omega}_2 = \ddot{\Psi}_2 = \ddot{\alpha}_1 \qquad \text{(A-14)}$$

$$\ddot{\omega}_3 = \ddot{\Psi}_3 = -(\ddot{\alpha}_0 + \ddot{\theta}_1 + \ddot{\alpha}_1) \qquad \text{(A-15)}$$

$$\ddot{\alpha}_0 = -\frac{\sin^2 \alpha_0 \ddot{\cos}\alpha_0 + \cos \alpha_0 \dot{\cos}^2 \alpha_0}{\sin^2 \alpha_0 \sqrt{\sin^2 \alpha_0}} \qquad \text{(A-16)}$$

$$\ddot{\theta}_1 = -\frac{\sin^2 \theta_1 \ddot{\cos}\theta_1 + \cos \theta_1 \dot{\cos}^2 \theta_1}{\sin^2 \theta_1 \sqrt{\sin^2 \theta_1}} \qquad \text{(A-17)}$$

$$\ddot{\alpha}_1 = -\frac{\sin^2 \alpha_1 \ddot{\cos}\alpha_1 + \cos \alpha_1 \dot{\cos}^2 \alpha_1}{\sin^2 \alpha_1 \sqrt{\sin^2 \alpha_1}} \qquad \text{(A-18)}$$

$$\ddot{e} = ade^{-1}(\cos \theta_0 - e^{-1}\dot{e}\sin \theta_0) \qquad \text{(A-19)}$$

$$\ddot{\cos}\alpha_0 = \frac{(2\dot{e}^2 - e\ddot{e})(a^2 - d^2) + e^3\ddot{e}}{2ae^3} \qquad \text{(A-20)}$$

$$\ddot{\cos}\theta_1 = \frac{(2\dot{e}^2 - e\ddot{e})(b^2 - c^2) + e^3\ddot{e}}{2be^3} \qquad \text{(A-21)}$$

$$\ddot{\cos}\alpha_1 = -\frac{\dot{e}^2 + e\ddot{e}}{bc} = -\frac{ad}{bc}\cos \theta_0 \qquad \text{(A-22)}$$

References

AKLEMAN, E., CHEN, J., XING, Q., AND GROSS, J. L. 2009. Cyclic plain-weaving on polygonal mesh surfaces with graph rotation systems. *ACM Transactions on Graphics 28*, 3, 78.

ALLIEZ, P., COHEN-STEINER, D., DEVILLERS, O., LÉVY, B., AND DESBRUN, M. 2003. Anisotropic polygonal remeshing. *ACM Transactions on Graphics 22*, 3, 485–493.

DIETZ, U. 1998. Fair surface reconstruction from point clouds. In *Proceedings of the international conference on Mathematical methods for curves and surfaces II Lillehammer, 1997*, Vanderbilt University, 79–86.

ELBER, G. 1995. Model fabrication using surface layout projection. *Computer-Aided Design 27*, 4, 283–291.

FISCHER-CRIPPS, A. C. 2007. *Introduction to contact mechanics*. Springer.

GARG, A., SAGEMAN-FURNAS, A. O., DENG, B., YUE, Y., GRINSPUN, E., PAULY, M., AND WARDETZKY, M. 2014. Wire mesh design. *ACM Transactions on Graphics 33*, 4, 66.

HIRANO, D., FUNAYAMA, Y., AND MAEKAWA, T. 2009. 3D shape reconstruction from 2D images. *Computer-Aided Design and Applications 6*, 5, 701–710.

JOO, H. K., YAZAKI, T., TAKEZAWA, M., AND MAEKAWA, T. 2014. Differential geometry properties of lines of curvature of parametric surfaces and their visualization. *Graphical Models 76*, 4, 224–238.

KILIAN, M., FLÖRY, S., MITRA, N. J., AND POTTMANN, H. 2008. Curved folding. *ACM Transactions on Graphics 27*, 3, 75.

KREYSZIG, E. 1968. *Differential Geometry*. Dover Publications, Inc., NY.

LIU, Y., POTTMANN, H., WALLNER, J., YANG, Y. L., AND WANG, W. 2006. Geometric modeling with conical meshes and developable surfaces. *ACM Transactions on Graphics 25*, 3, 681–689.

MAEKAWA, T., WOLTER, F. E., AND PATRIKALAKIS, N. M. 1996. Umbilics and lines of curvature for shape interrogation. *Computer Aided Geometric Design 13*, 2, 133–161.

MARINOV, M., AND KOBBELT, L. 2004. Direct anisotropic quad-dominant remeshing. In *Proceeding of the Computer Graphics and Applications, 12th Pacific Conference*, IEEE Computer Society, 207–216.

MARTIN, R. R. 1983. Principal patches - a new class of surface patch based on differential geometry. In *Eurographics '83*, P. J. W. T. Hagen, Ed., 47–55.

MASSARWI, F., GOTSMAN, C., AND ELBER, G. 2008. Paper-craft from 3D polygonal models using generalized cylinders. *Computer Aided Geometric Design 25*, 8, 576–591.

MATSUO, K., AND MATSUOKA, K. 2010. Development of new system for developing curved shell plates of ships. *Transactions of the Japan Society of Mechanical Engineers. C 76*, 771, 2797–2802. In Japanese.

MITANI, J., AND SUZUKI, H. 2004. Making papercraft toys from meshes using strip-based approximate unfolding. *ACM Transactions on Graphics 23*, 3, 259–263.

PATRIKALAKIS, N. M., AND MAEKAWA, T. 2002. *Shape Interrogation for Computer Aided Design and Manufacturing*. Springer-Verlag, Heidelberg.

POTTMANN, H., SCHIFTNER, A., WIEN, P. B. T., SCHMIEDHOFER, H., WANG, W., BALDASSINI, N., AND WALLNER, J. 2008. Freeform surfaces from single curved panels. *ACM Transactions on Graphics 27*, 3, 76.

PRATT, M. 1990. Cyclides in computer aided geometric design. *Computer Aided Geometric Design 7*, 221–242.

SHATZ, I., TAL, A., AND LEIFMAN, G. 2006. Paper craft models from meshes. *The Visual Computer 22*, 9, 825–834.

TANG, C., SUN, X., GOMES, A., WALLNE, J., AND POTTMANN, H. 2014. Form-finding with polyhedral meshes made simple. *ACM Transactions on Graphics 33*, 4, 70.

TANG, C., BO, P., WALLNER, J., AND POTTMANN, H. 2016. Interactive design of developable surfaces. *ACM Transactions on Graphics 35*, 2, 12.

WILLMORE, T. J. 1959. *An Introduction to Differential geometry*. Clarendon Press, Oxford.

ZHANG, E., HAYS, J., AND TURK, G. 2007. Interactive tensor field design and visualization on surfaces. *IEEE Transactions on Visualization and Computer Graphics 13*, 1, 94–107.