International Journal of CAD/CAM *Vol. 13, No. 2, pp. 23~29 (2013)*

INTERNATIONAL JOURNAL OF CAD/CAM
www.ijcc.org

Interactive Design of 3D Geometry Made by Bending Inextensible Sheet Material Containing Slits

Naoya Tsuruta[1], Jun Mitani[1,2], Yoshihiro Kanamori[1] and Yukio Fukui[1]
[1]Department of Computer Science, University of Tsukuba, Japan
[2]JST ERATO, Japan

Abstract – Most manufacturing materials, such as wooden boards, metal sheets, and paper, are planar and nearly inextensible. Bending such materials not only allows us to form objects, but also avoids wastes and reduces gluing costs. However, the resulting shape is limited to developable surfaces. Due to this limitation, this method lacks flexibility in design. This problem can be addressed by adding slits to the materials, thereby allowing the conformation of a single sheet into expressive surfaces with high flexibility. In this paper, we propose an interactive design system for flexible geometry. A rectangular sheet material containing parallel slits is used in this work. The regions between the slits are called strips, and by bending them, we can simulate the desired shape. Both flexibility and geometric constraints are satisfied by a numerical optimization approach. The proposed system optimizes a bending shape of the strips by considering the smoothness of the surface and the user input. We show the effectiveness of our system using evaluation experiments.

***Keywords*: Geometric modeling, Developable surface, Constrained optimization, Sheet material, Containing slits**

1. Introduction

Most well-known materials, such as wooden boards, metal sheet, and paper, are planar and nearly inextensible. Forming an object from these materials by bending a single part avoids waste of the material and reduces the costs for joining parts. However, the shape produced by the bending is limited to the developable surface. The shape thus lacks flexibility in the design. Adding slits to the material is one approach to address this problem. As an actual example, Naruse et al. manually designed and created a chair from a wooden board having parallel slits (Fig. 1).

It is remarkable that an expressive surface can be obtained from a single sheet of material by simply adding slits. However, designing objects made with "a sheet material with slits" has not been performed on a computer. It is because the constraint, which is the unfolded pattern of the shape matching the predefined one without any tearing and wrinkles, is difficult to satisfy during the design process. Naruse et al. had to perform a number of trial and error processes with a sheet of paper before obtaining the chair shown in Fig. 1. In this paper, we present a CAD system for designing an object that can be reconstructed from a sheet material with parallel slits without tearing and gluing. The seating face of the chair consists of a set of *strips*. We refer to the regions in the sheet material between slits as 'strips' hereafter. The whole shape design consists of problems on how each strip of the nearly inextensible material can be shaped while maintaining the constancy of the unfolded pattern. Thus, the length of the strip must be constant and the position of both ends of the strip must be constrained by other parts of the sheet material. Both ends of our target strip are fixed so that it cannot move freely. Therefore, we have to shape the strips under this constraint. Here, we assume this problem as a constrained optimization problem and present an optimization-based shape modeling system. In Section 2, we introduce related work. In Section 3, we show our proposed system. Then, we describe details of the optimization (3.1) and the modeling processes (3.2). We discuss our results in Section 4 by providing several design examples and a fabricated physical object. We finally conclude in Section 5.

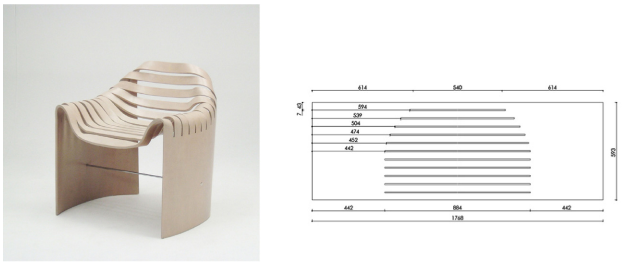


Fig. 1. Chair, designed by Naruse Inokuma Architects, made of a rectangular wooden board having parallel slits [1].

*Corresponding author:
Tel: +81-29-853-5388
Fax: +81-29-853-5388
E-mail: tsuruta@npal.cs.tsukuba.ac.jp

2. Related work

Developability is important for products such as papercrafts [2], stuffed toys [3] and garments [4], which are cut out from sheets of paper or cloths. Developable surfaces have valuable characteristics for industrial and architectural designs. Using developable surfaces in architectural design reduces the costs for construction [5], [6]. Another example can be found in the design of a ship hull [7]. Methods that generate a set of developable patches from a closed space curve [8] and from a parametric curved surface [9] are available. These methods construct a whole shape with a set of developable patches. In other words, the developable patches are glued together along their boundaries. Our target shape, which is a chair, is made from a single developable sheet, and adding slits is the only process used.

Peternell proposed a method for reconstructing a developable surface that is fitted to 3D points measured by a 3D scanner [10]. Kilian et al. reconstructed surfaces having folded creases from scanned 3D data using an optimization-based approach [11]. Rohmer et al. proposed a method for reconstructing a folded paper geometry from a 2D pattern and a 3D contour curve [12]. In this paper, we propose an interactive design system. Moreover, we extend the system so that a shape (a chair) is automatically reconstructed from 3D scanned data.

Mitani and Igarashi proposed an interactive design system for origami pieces having curved folds, based on mirror reflection [13]. Bo et al. proposed a system with which a user can interactively bend a sheet of paper [14]. These methods fix one end of the sheet paper in 3D space and allow the other end to be moved. This approach cannot be used for our target shape formed from a sheet material with slits.

3. Proposed system

First, we define the target material as a planar rectangle containing one or more slits. The slits are parallel and of equal length. The shape of the material is symmetric and defined by four parameters, as illustrated in Fig. 2. The designed 3D shape is also symmetric. Therefore, we only consider the half regions of the material in the following. As illustrated in Fig. 3, a strip is divided into M quads and has (M+1) ridge lines. A ridge line E_i has three parameters, $x_{L,i}$, $x_{R,i}$, and θ_i, where $x_{L,i}$, $x_{R,i}$ are the x coordinates of the upper and lower end points of E_i, respectively, when the strip is oriented along the x axis, and θ_i is the dihedral angle of E_i. We name the two end points of E_i as $\upsilon_{L,i}$ and $v_{R,i}$. We assume that the first edge E_0 is fixed. Therefore, the shape of the strip is determined by the parameters associated with the ridge lines. The proposed system calculates the values of these parameters to obtain the intended shape of the strip while keeping the shape of the unfolded sheet constant. We add two more constraints here so that the strip becomes smooth and the user can control its shape. To obtain the final shape of each strip as a solution of an optimization problem, we define a penalty function. We will describe details of the penalty function in the following section. The proposed system solves an optimization problem to minimize the value of the function.

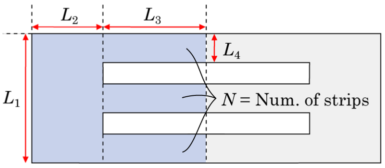


Fig. 2. Developed pattern defined by four numbers from L_1 to L_4 and number of strips N.

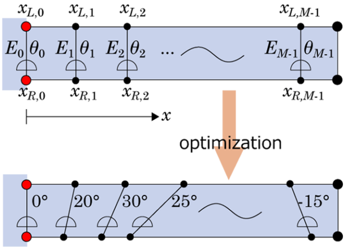


Fig. 3. Ridge lines before after optimization.

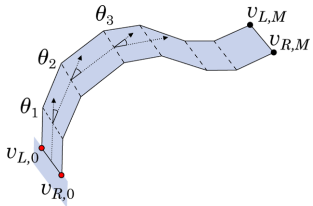


Fig. 4. Calculation of 3D and positions.

3.1 Optimization

As the constraints, one end of the strip is located on the symmetry plane, and the other end is fixed where the user specifies. The constraint on the symmetry plane is given because we defined the target shape to be symmetric. The positions of $\upsilon_{L,0}$ and $\upsilon_{R,0}$ are defined by the user. The positions of $\upsilon_{L,i}$ and $\upsilon_{R,i}$ ($1 \le i \le M$) in 3D space are calculated by rotating the quad between $E_{i\text{-}1}$ and E_i by angle θ_i around $E_{i\text{-}1}$. The positions $\upsilon_{L,M}$ and $\upsilon_{R,M}$, which are the last to be obtained, have to be on the symmetry plane, as illustrated in Fig. 4. We use an optimization approach to meet this constraint. Here, we define the following penalty function for each strip. Then, we solve the optimization problem that minimizes the value of the function.

$$F_{penalty} = w_1 F_{no\_gap} + w_2 F_{smooth} + w_3 F_{fair} + w_4 F_{user\_control} \tag{1}$$

We add terms in the function that makes the resulting shape smooth and controllable. The components in $F_{penalty}$ measure the connectivity at the symmetry plane of the strip, smoothness of the strip, fairness of the quads in the strip, and closeness to the user-specified anchor points of the strip, respectively. To prevent the occurrence of a gap at the symmetry plane, we set a larger value of ω_1 in comparison with those of the other weights.

To solve the optimization problem, we use a penalty method with the Levenberg-Marquardt algorithm. The derivative of the above equation is calculated numerically. Since the number of variable end points of the ridge lines in a strip is $2(M-1)$, and the number of variable angles of the ridge lines in a strip is M, the number of whole variables becomes $D = 2(M - 1) + M$. The coefficient matrix used in the optimization is highly sparse, with a size of $D \times D$. We describe the details of each component of the penalty function below.

Position constraint for an end of a strip. The term F_{no_gap} is the sum of the squared distances of $\upsilon_{L,M}$ and $\upsilon_{R,M}$ from the symmetry plane, as illustrated in Fig. 5. Since the whole shape of the designed model is symmetric and obtained by mirroring the half model, the value of this term indicates the size of the gap on a strip at the center of the whole model. Ideally, this value should be zero. We set the symmetry plane on the yz coordinate plane. Then, the term is calculated by the following equation.

$$F_{no\_gap} := (\upsilon_{L,M})_x^{\,2} + (\upsilon_{R,M})_x^{\,2} \tag{2}$$

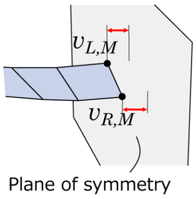


Fig. 5. Evaluation of constraint at end of strip.

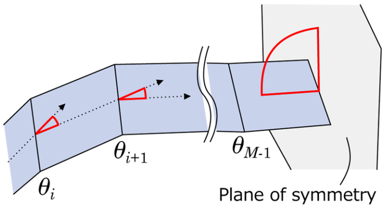


Fig. 6. Evaluation of smoothness.

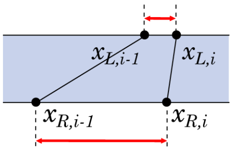


Fig. 7. Evaluation of degeneracy of quads.

Smoothness. The term F_{smooth} is for smoothing a strip. We measure the differences of the angles between neighboring ridge lines. When the differences are small, the strip is smooth. Additionally, we introduce a term, F_{end_angle}, which makes the strip touch the symmetry plane perpendicularly, as illustrated in Fig. 6. This makes the strip smooth at the plane of symmetry (the center of the whole shape). In the following equation, $\vec{\upsilon}_{p\_o\_s}$ is a vector parallel to the symmetry plane. Since the symmetry plane is on the yz coordinate plane, $\vec{\upsilon}_{p\_o\_s} = (0, 1, 0)$.

$$\begin{aligned} F_{smooth} &:= \sum_i (-\theta_{i-1} + 2\theta_i - \theta_{i+1})^2 + F_{end\_angle} \\ F_{end\_angle} &= ((\upsilon_{L,M-1} - \upsilon_{L,M}) \cdot \vec{\upsilon}_{p\_o\_s})^2 \end{aligned} \tag{3}$$

Elimination of degeneracy of quads. We measure the distances between end points of neighboring ridge lines to prevent crossing or degenerating of a quad into a triangle (Fig. 7). The penalty value of the following equation becomes larger when the distance between the end points of neighboring ridge lines becomes smaller.

$$F_{fair} := \sum_i \left(\frac{1}{x_{L,i} - x_{L,i-1}} + \frac{1}{x_{R,i} - x_{R,i-1}} \right) \tag{4}$$

User-specified anchors. The user controls the shape by specifying two types of anchors: one directly controls the position of each vertex $v_{L,i}$ or $v_{R,i}$, and the other one controls the cross section on the symmetry plane (Fig. 8). The "position-control" anchor measures the squared distance from vertex v to the user-specified position υ'. This controls the shape of a strip locally. The "cross-section" anchor measures the shortest distance from $\upsilon_{L,M}$ and $\upsilon_{R,M}$ to $c_{symmetry}$, which is the cross-

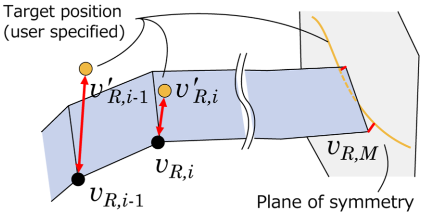


Fig. 8. Evaluation of distances to user-specified anchors.

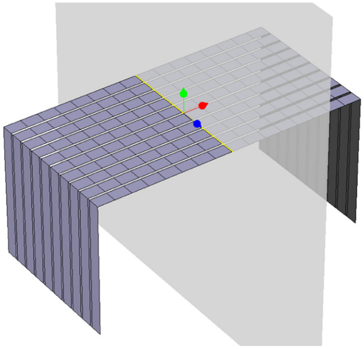


Fig. 9. Initial state.

sectional curve drawn on the symmetry plane by the user. The curve controls the global shape of a strip. The calculation of the distance from a point to a curve requires iterative computation. Therefore, the curve is approximated by small segments and the distance is measured from the nearest segment to reduce the computational cost.

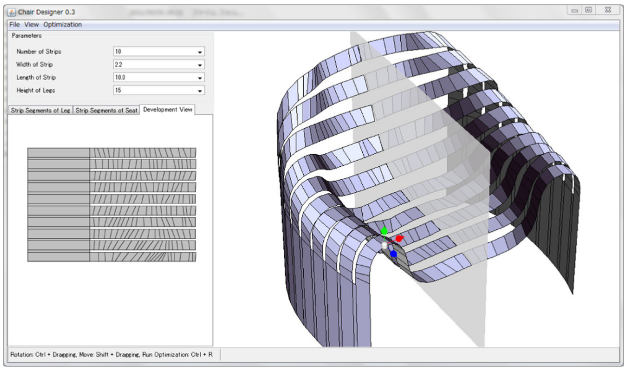
$$F_{user\_control} := \sum_i ((\upsilon'_{L,i} - \upsilon_{L,i})^2 + (\upsilon'_{R,i} - \upsilon_{R,i})^2) \qquad (5)$$
$$+ dist(\upsilon_{L,M}, c_{symmetry})^2 + dist(\upsilon_{R,M}, c_{symmetry})^2$$

In our implementation, we set $(\omega_1, \omega_2, \omega_3, \omega_4)$ =(0.1, 0.04, 0.01, 0.03) as the weights of the terms in Eq. (1). These values are determined by experience. The basic idea is that ω_1 should be a higher value to make gaps at the symmetry plane small. Weight ω_4 affects the controllability.

Before running the optimization, the ridge lines are equally spaced and their angles are zero except for θ_0. As shown in Fig. 9, θ_0 is set to 90 degrees. The part to which the strips are connected (the side panel of the chair) is perpendicular to the ground plane (*xz* plane) and is placed at a distance of L_3 from the symmetry plane.

3.2 Workflow

Fig. 10 shows the GUI of the proposed system. First, the user inputs the parameters L_1 to L_4, which define the developed pattern (Fig. 2). The 3D positions of the ends of the strips are specified by two 2D Bézier curves, i.e., the leg curve and the seat curve, as shown in Fig. 11. The positions of $\upsilon_{L,0}$ and $\upsilon_{R,0}$ are defined by the leg curve and the parameter L_2. The seat curve defines the cross section on the symmetry plane. Both curves are edited by the user. The system runs an optimization to compute a shape that satisfies the constraints. This optimization runs during interactive modeling by the user. The system updates the shape of the designed model just after each operation by the user. The user can add an anchor to a vertex to control the shape (Fig. 12). For precise control, the user can also add multiple anchors. Until the user deletes the anchor, the anchor is used in the optimization process. The optimization runs only on the affected strip when the user edits an anchor point. The computational time for the optimization is reasonably fast. In addition, the user can interactively edit the shape while seeing the effect of the anchor points.



Number of Strips 10
Width of Strip 2.2
Length of Strip 10.0
Height of Legs 15

Rotation: Ctrl + Dragging, Move: Shift + Dragging, Run Optimization: Ctrl + R

Fig. 10. Screenshot of our system.

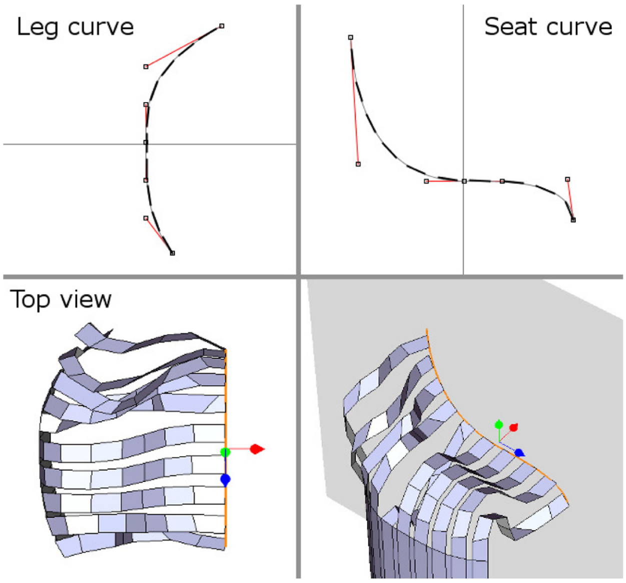


Fig. 11. General design by two curves.

Another user operation can subdivide the strips to make them smoother (Fig. 13). The new ridge lines are inserted with a zero angle between neighboring ridge lines on the strips. The user starts by editing a coarse quad mesh, then divides the quads to make the strips smooth. This process effectively reduces the total computational costs for the optimization. By applying subdivision and optimization alternately, the strips

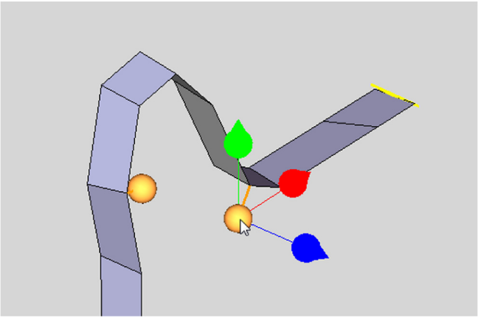


Fig. 12. Adding an anchor point (orange sphere) to a vertex.

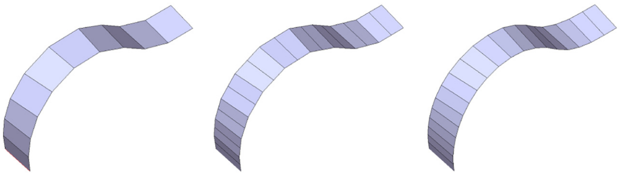


Fig. 13. Subdivision of a strip. From the left: before and after subdividing, and after optimization.

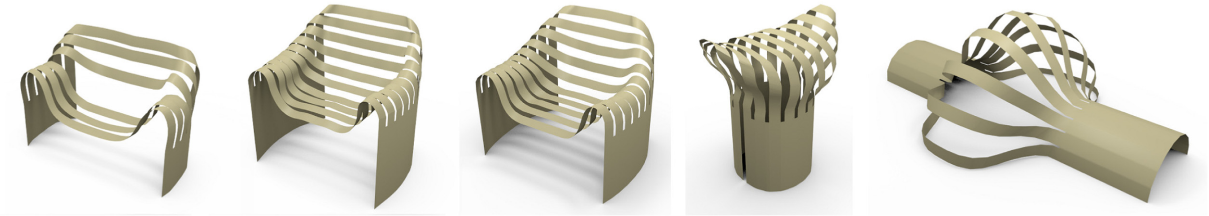


Fig. 14. Examples. Left two: 6-strip chair. Center: optimized chair model with reference to scanned points. Right two: monuments.

become smooth.

4. Results

The prototype system is implemented using Java on a 2.66 GHz PC. Fig. 14 shows some example models designed using the system. Designing each model required approximately 20 minutes. Although composing the rough shape with the coarse quad was easy, precise control of the strip was difficult and took a long time. For designing a chair, a long time was required to make its seat flat since the surface tended to become wavy. Changing the weights of the penalty function would make it easier.

All models except the center model in Fig. 14 were designed interactively. The legs of the rightmost model were set parallel to the ground. The center model was designed with the other approach. We first made a chair-shaped model with a sheet of paper by hand to obtain a shape as shown in Fig. 1. Then, we obtained the 3D coordinates of the points on the paper model using a 3D range scanner. By slightly modifying the penalty function so that the optimized shape came close to the 3D scanned points (details are described in Section 4.1), we obtained the result.

With our approach, the distance from the symmetry plane to the end of a strip was useful as an indicator of the validity of the obtained shape. We calculated the maximum gap distance G in all strips. By dividing G by the length of the developed pattern, $E = G/(L_2 + L_3)$ gives the normalized value of error. The largest error in the examples in Fig. 14 was 0.0015. This means that when we use a 1768 mm long material (the length used for the chair shown in Fig. 1), the gap is approximately 2.65 mm, which is acceptable for manufacturing.

The computational time for the optimization was less than 100 ms per strip. It would be possible for the user to edit the shape with less distraction when the system runs on a more powerful PC. Although subdividing quads in a strip increases the computational cost, the optimization converges fast because the starting shape is already near its solution. The system was fast enough even after the strips were subdivided twice, which made them smooth enough. However, the performance was not good, especially just after the initial state when the shape was far from the final convergence. It took approximately 500 ms per strip in which ten quads are contained. We limited the number of iterations for solving an equation in the optimization process. The user can avoid a slow convergence by changing the anchor positions.

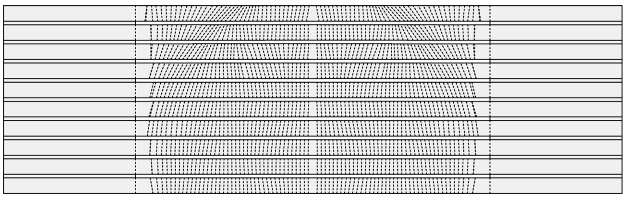


Fig. 15. Unfolded pattern of the center model in Fig. ?.

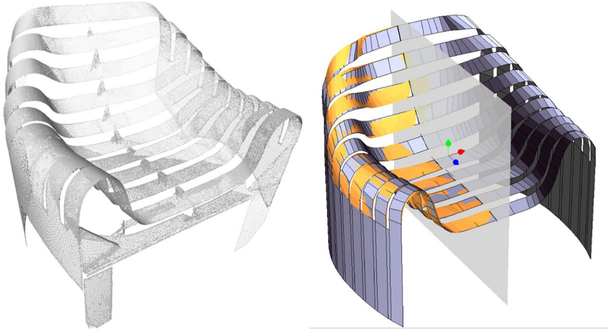


Fig. 16. 3D scanned data and extracted scanned data (orange) overlaid on the edited shape (blue).

4.1 Approximation to scanned data

We perform the following steps to make the strips close to the scanned 3D model shown in Fig. 16 left.

1. Clean up the scanned data by eliminating noise.
2. Manually extract the points corresponding to each strip and output them individually.

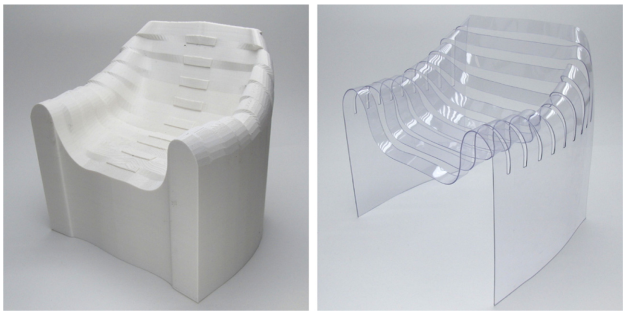


Fig. 17. 3D printed mold and the formed polyvinyl chloride board. Size(W/H/D): 247×233×179 mm.

3. Display the strips in the system (Fig. 16 right).
4. Roughly adjust the position of the strips manually.
5. Optimize with an additional constraint.

We combine an additional penalty function shown in Eq. (6), such that *nearest*(υ_i) is the closest point to v_i in the scanned points, into Eq. (1) to take consider the closeness of a strip to point clouds. We could obtain a shape resembling the original chair by adding this term. However, the chair shown in Fig. 1 seems to contain deformation of the material, and the points clouds may contain an error created while scanning. It was impossible to match them completely because our system models an ideal shape.

$$F_{prox} := \sum_i ((nearest(\upsilon_{L,i}) - \upsilon_{L,i})^2 + (nearest(\upsilon_{R,i}) - \upsilon_{R,i})^2) \qquad (6)$$

4.2 Fabrication

We verified the proposed system by making a physical model. First we created a solid mold model from the side panels and strips by adding faces in the front, back and bottom. We printed out the mold on a 3D printer, then we pressed a 1 mm thick polyvinyl chloride sheet with slits onto the mold. We heated the sheet with a hair dryer to bend it. Fig. 17 shows the 3D printed mold and the formed sheet. The unfolded pattern to cover the mold was slightly shorter. It seems that the shortage was due to the fact that we did not consider the thickness of the material when we printed out the mold.

However, we confirmed that we could reconstruct the model designed on our system with a flat sheet material.

5. Conclusion and Limitations

In this paper, we have proposed a method for designing a shape that can be made from a planar material containing slits. The positions and angles of the ridge lines on the strips are calculated by an optimization. The user can control the final shape that satisfies the geometric constraints. To evaluate our system, we printed out a mold using a 3D printer, cut out the unfolded pattern, and formed the designed shape by compressing the material onto the mold. We confirmed that the shape could be reconstructed with a sheet material.

The domain of the shapes that we can design on our current system is very narrow. Changing the shape and position of slits is a way to allow more variation. Asymmetric shapes could be obtained without mirroring. The user interface could be improved so that the user can control the final shape more precisely. Optimizing all strips together with considering the gap-sizes could generate smooth changes between strips. Combining a physical simulation of strength or balance makes a practical system for designing self-supporting and robust shapes.

References

[1] Naruse, Y. and Inokuma, J. Naruse Inokuma Architects, http://www.narukuma.com/main.html.

[2] Mitani, J. and Suzuki, H. (2004). Making papercraft toys from meshes using strip-based approximate unfolding, *ACM Trans. Graph.*, 23, 259-263.

[3] Julius, D., Kraevoy, V. and Sheffer, A. (2005). D-Charts: Quasi-Developable Mesh Segmentation, *Computer Graph. Forum*, 24(3), 581-590.

[4] Decaudin, P., Julius, D., Wither, J., Boissieux, L., Sheffer, A. and Cani, M. P. (2006). Virtual Garments: A Fully Geometric Approach for Clothing Design, *Computer Graphics Forum (Eurographics' 06 proc.)*, 25(3) (2006).

[5] Liu, Y., Pottmann, H., Wallner, J., Yang, Y. L. and Wang, W. (2006). Geometric modeling with conical meshes and developable surfaces, *ACM Trans. Graph.*, 25, 681-689.

[6] Pottmann, H., Schiftner, A., Bo, P., Schmiedhofer, H., Wang, W., Baldassini, N. and Wallner, J. (2008). Freeform surfaces from single curved panels, *ACM Trans. Graph.*, 27, 76:1-76:10.

[7] Perez, F. and Suarez, J. A. (2007). Quasi-developable B-spline surfaces in ship hull design, *Computer-Aided Design*, 39(10), 853-862.

[8] Rose, K., Sheffer, A., Wither, J., Cani, M. P. and Thibert, B. (2007). Developable surfaces from arbitrary sketched boundaries, *Proceedings of the fifth Eurographics symposium on Geometry processing*, Aire-la-Ville, Switzerland, Switzerland, Eurographics Association, 163-172.

[9] Subag, J. and Elber, G. (2006). Piecewise developable surface approximation of general NURBS surfaces, with global error bounds, *Proceedings of the 4th international conference on Geometric Modeling and Processing*, Berlin, Heidelberg, Springer-Verlag, 143-156.

[10] Peternell, M. (2004). Developable surface fitting to point clouds, *Comput. Aided Geom. Des.*, 21, 785-803.

[11] Kilian, M., Flory, S., Chen, Z., Mitra, N. J., Sheffer, A. and Pottmann, H. (2008). Curved folding, *ACM Trans. Graph.*, 27, 75:1-75:9.

[12] Gumerov, N. A., Zandifar, A., Duraiswami, R. and Davis, L. S. (2006). 3D Structure Recovery and Unwarping of Surfaces Applicable to Planes, *Int. J. Comput. Vision*, 66(3), 261-281.

[13] Mitani, J. and Igarashi, T. (2011). Interactive Design of Planar Curved Folding by Reflection, Pacific Conference on Computer Graphics and Applications - Short Papers, Kaohsiung, Taiwan, 77-81.

[14] Bo, P. and Wang, W. (2007). Geodesic-Controlled Developable Surfaces for Modeling Paper Bending, *Comput. Graph. Forum*, 26(3), 365-374.

Naoya Tsuruta is currently pursuing his Ph.D. in Engineering at the University of Tsukuba, Japan. He received his B.S. in Computer Science and M.S. in Engineering from University of Tsukuba in 2010 and 2012, respectively. His research interests include geometric modeling, optimization, and interactive CAD systems for the geometry of origami.

Yoshihiro Kanamori received his B.S., M.S., and Ph.D. degrees in computer science from the University of Tokyo, Japan, in 2003, 2005, and 2009, respectively. Since April 2009, he has been an assistant professor at the University of Tsukuba. His research interests center in computer graphics including rendering and visual simulations. He now studies image editing techniques for reproducing real-world phenomena.

Jun Mitani received his Ph.D. in Engineering from the University of Tokyo in 2004. He had been working at RIKEN and University of Tsukuba since 2004 and 2005, respectively. He has been an associate professor in the Department of Computer Science at University of Tsukuba since 2009. He has been studying geometric modeling in the field of computer graphics.

Yukio Fukui received his Ph.D. in Engineering from the University of Tokyo in 1993. He had been working at the National Institute of Bioscience and Human-Technology, M.I.T.I. and the Institute of Information Sciences and Electronics at University of Tsukuba since 1993 and 1998, respectively. He has been a professor in the Department of Computer Science at University of Tsukuba since 2004.

Naoya Tsuruta

Jun Mitani

Yoshihiro Kanamori

Yukio Fukui