

スリット入り平板素材を折り曲げてできる形状の 設計支援ツール

鶴田直也^{†1} 三谷 純^{†1,†2}
金森由博^{†1} 福井幸男^{†1}

伸縮しない薄い素材を折り曲げてできる形状は可展面に限られ、自由曲面を含む形を作ることは本質的にできないが、素材に切り込みを入れることを許容すれば形状変形の自由度が増し、より意匠的な形を作ることが可能となる。しかしながら、実際に制作できる形には依然として厳しい幾何的な制約があり、この制約を満たしながら意図した形を設計することは難しい問題である。本研究では、複数のスリット（細長い切りこみ）が入れられた長方形の素材を対象として、対話的に形を操作できるシステムを提案する。システムは、予め設定された折り線の位置と折れ角を最適化手法によって調整し、平面に展開できるという制約を維持しつつ意図した形のデザインを実現する。

A design tool for bending of planar sheet of material with slits

NAOYA TSURUTA,^{†1} JUN MITANI,^{†1,†2}
YOSHIHIRO KANAMORI^{†1} and YUKIO FUKUI^{†1}

Surfaces that can be produced by folding a single planar sheet of material without stretching is inherently limited to developable surface. Allowing the cutting makes greater flexibility and may enable more variety in design. However designing such shape is still difficult problem while it involves geometric constraints. In this paper, we propose an interactive design system for bending of a rectangular material with parallel slits. Our system configures positions of foldings and its angles with an optimization approach, and achieves desired shape while keeping developability.

1. はじめに

紙のような伸縮しない素材の折り曲げによる形状設計は、幾何的な制約を含む難しい問題であり、現在でもその性質や設計手法について研究が行われている。これまで、複雑な折り目を入れることで立体的な形状を作り出す各種の手法が提案されてきた。しかしながら、そもそも折りたたみのみでできる形は制約が非常に厳しく、自由に形状を設計することは難しい。

では、形の自由度を上げるにはどのようなアプローチがあるだろうか。一つは、複数の素材を組み合わせたことが考えられる。これはペーパークラフトのようなアプローチであり、個々の部品は平坦な素材から切り出すことが可能という条件の元で、複数の部品を接合することで得られる最終的な形を設計する。もう一つは、切り込みを入れることである。平板素材に細い切り込み（以降ではスリットと呼ぶ）を入れることで、1枚の素材からでもより複雑な形状を作り出すことが可能になる。ペーパークラフトほどの形の自由度はないが、スリットを入れない場合に比べれば自由度は大きく向上する。スリットを用いる最大のメリットは一枚の素材から形状を設計できることであり、工学的には部材の接合のための工程が不要であるという利点がある。

本研究ではより条件の厳しい後者の切り込みを入れることに着目し、スリットを入れた平板素材の折り曲げによる設計を試みる。その例題として図1に示す椅子を取り上げ、このような形を計算機上で構築することを目的とした。今回対象とするものは、長方形の部材にスリットを入れただけのものであり、素材を有効に活用することができる。多くの既存CADシステムでは立体形状の可展性や展開後の形状を取り扱うことが難しく、この椅子もデザイナーが紙を用いて試行錯誤の繰り返しにより設計したものである。本稿で提案するツールは、最適化アプローチによる可展性とユーザの意図を同時に考慮した対話的な編集が可能であり、出力される立体形状は平面に展開可能かつ長方形に収まるようになっている。

本研究は、1枚の平板素材からできる形という制約条件の元で、実現可能な形を調査するための取り組みでもある。紙を折る芸術である折り紙のように、厳しい制約の中で得られる形に現れる幾何学的な美しさから、芸術的な観点からも研究の意味があると考えている。

^{†1} 筑波大学

University of Tsukuba

^{†2} JST/ERATO

Japan Science and Technology Agency / ERATO



図 1 スリットを入れた素材の折り曲げによる椅子．成瀬・猪熊建築設計事務所¹⁾による設計．
Fig. 1 Chair designed by Naruse Inokuma Architects¹⁾.

提案するシステムを用いることにより，新しいデザインの創発や実際にものづくりを行う際の支援が期待される．続く 2 章では関連した研究について紹介する．3 章では提案手法について述べ，システムを用いて得られた結果を 4 章に記載する．最後に 5 章でまとめを述べて本稿の締めくくりとする．

2. 関連研究

薄い板状の素材から，素材を伸縮させることなく立体的な形状を作るとき可展面が現れる．可展面は直線の軌跡によって描かれる線織面に含まれる面であり，平面，円柱，錐面，接線曲面に分類される．したがって折り曲げによって作られる可展性をもつ立体形状は，複数の可展面パッチの複合によって表現できる．

冒頭で述べた複数の部品を使うことを前提としたアプローチとしては，ペーパークラフトを題材としたもの^{2),3)} や，実際の建築構造を考慮したオフセットが得られるような構造^{4),5)}，あるいは空間中に閉曲線をデザインすることで，その内部に可展面を生成する手法⁶⁾などが挙げられる．しかし，ここでは 1 枚の素材から形を作るアプローチについて詳細を述べる．

後者のアプローチは，切り貼りを許容するアプローチに比べて形の制約が厳しく，設計はさらに難しい．Kilian らは，実際に人の手で折った形を 3 次元計測器で測定し，その 3D モデルを可展面の集合で近似的に再構築している⁷⁾．しかしながら，複雑な最適化を要することからリアルタイムでの計算は行えず，対話的な設計は行えない．

現在のところ，可展面で構成された立体形状を対話的に設計できる手法は少ない．Mitani は可展面で表現されたオブジェクトを，空間中の平面で対称に折り曲げることで形を編集する手法を提案している⁸⁾．これは常に可展性を保持しながら対話的に設計できるが，平面で折り曲げることから，曲線が平面曲線に限定されているといえる．より単純なものとしては，紙の帯状部分（以後ストリップと呼ぶ）を対話的に折り曲げて形を作る手法がある⁹⁾．これはストリップの中心を通る曲線を入力とし，与えられた曲線を一定の間隔で区切って 3 次微分を計算することでストリップを生成している．しかしこの手法は，スリットが素材の輪郭と接している形状，つまり「切り込み」を入れた形状は可能であるが，図 1 のようなスリットが素材内部にある切り抜いた形状には対応できない．

提案手法の中心となるのは，可展性を維持したストリップの対話的な編集である．折り曲げたストリップ形状の表現，および妥当な形状を求める最適化計算には，文献 4) および文献 5) で用いられている PQ メッシュ（平面四辺形の集合）による表現およびその最適化手法を応用している．

3. 提案システム

提案システム（図 2）ではユーザ入力によって制約条件を変化させながら，最適化処理によって制約を満たす形状を計算する．本章では，最初に椅子の幾何モデル（3.1 節）について説明し，最適化処理で計算する制約を述べる（3.2 節）．その後，ユーザ入力と作業プロセス（3.3 節）について述べる．

3.1 椅子の幾何モデル

本稿では次のような素材を対象とする．

- 対象とする形状は左右対称とする．
- ストリップ部分の幅と長さは一定とする．
- スリットの幅と長さは一定かつスリットは等間隔に配置される．

左右対称なものに限定しているため，以降では片側のみを取り扱う．展開図は図 3 のように全体の幅 ($L1$)，連結部の長さ ($L2 =$ 脚部の高さ)，ストリップ部分の長さ ($L3$)，各ストリップの幅 ($L4$)，ストリップの本数 (N) から定義される．内側のストリップの位置はこれらの値から均等に配置される．

ストリップ部分の折り曲げは図 4 のように実現する．以下では，1 本のストリップに注目し説明する．ストリップは n 分割されており，分割位置には折り目 E_i が入っている．各折り目 E_i は，ストリップ中での位置を表す $(x_{L,i}, x_{R,i})$ のセットと折り角度 θ_i をもつ．この

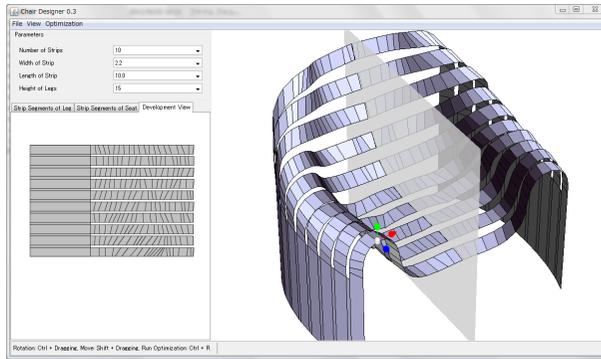


図 2 提案システムの実行画面
Fig. 2 Screenshot of our proposed system.

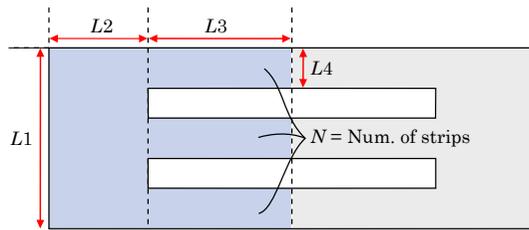


図 3 椅子の展開図. $L1$ から $L4$ までの連続値およびストリップの数 N によって定義される.
Fig. 3 Development pattern of a chair which is defined by parameters named $L1$ to $L4$ and the number of strip N .

折り目の位置と角度を最適化処理によって修正しつつ目的の形を実現する. 3次元空間での各頂点の座標は, 一方の端を固定しておき, 折り線の位置と角度を順番に計算していくことで求まる(図5). ここで計算した立体形状が平面に展開できるための条件は, 「終端(自由端)が対称面上に乗る」ということである. 次のセクションではこの可展性を考慮した最適化処理について述べる.

3.2 最適化

各頂点を微小距離ずつ動かし, 反復計算によって平面に展開できる形状を求める. 平面に展開可能という条件を満たす形状は無数に存在するが, その中からより「良い」形状を求めるために, ストリップが滑らかに変化すること, 境界面で鏡像とスムーズに連結すること,

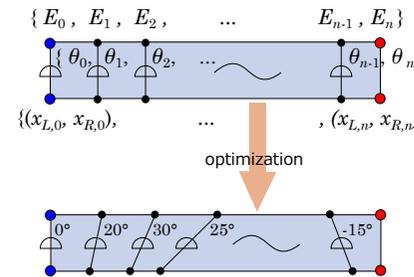


図 4 ストリップを分割する折り目と最適化計算後のイメージ
Fig. 4 Foldings in a strip and image after optimization.

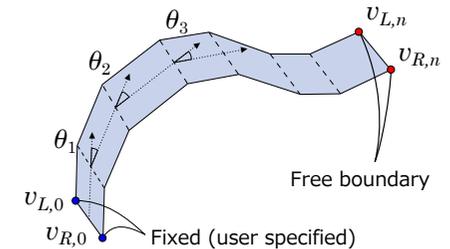


図 5 3次元座標の計算
Fig. 5 Calculation of coordinates in the 3D space.

折り目で区切られた四辺形が均一であること, そしてユーザの意図をなるべく反映させることを考慮した計算を行った. これら 4 項目について設定した, ペナルティ関数を以下で述べる. なお, 展開図上での折り目の位置は x_L, x_R , 3次元空間での位置は v_L, v_R で表す.

滑らかさ(図6) ストリップが鋭く折れ曲がらないように, それぞれの折り目での曲率を計算し, その総和をエラー関数とする. 離散的な曲線における曲率は次の式で計算できる¹⁰⁾.

$$F_{smooth} := \sum_i (\theta_i)^2 \quad (1)$$

対称性(図7) ストリップの終端が対称面上に乗るといのが可展性に必要な条件であるが, これだけでは対称面で角度のついた形状が生成されてしまう. そこで対称面と垂直に接するという条件を与えることとした. 対称性 $F_{symmetry}$ は終端と対称面との距離 $F_{endpoint}$ と終端の角度 $F_{endangle}$ の和で表す.

$$F_{symmetry} := F_{endpoint} + F_{endangle} \quad (2)$$

$$F_{endpoint} = (v_{L,n} + v_{R,n})^2$$

$$F_{endangle} = ((v_{L,n-1} - v_{L,n}) \cdot \vec{v}_{plain_of_symmetry})^2$$

ただし, 対称面を $x = 0$ としているため, $F_{endpoint}$ では終端の頂点の x 座標をそのまま利用している. また同じ理由から $\vec{v}_{plain_of_symmetry} = (1, 0, 0)$ である.

フェアネス(図8) 折り目同士が極端に近づく, その付近で素材が尖ったように曲がってしまう. これを回避するために, 折り目と折り目の距離の逆数を関数として与え, 頂点同

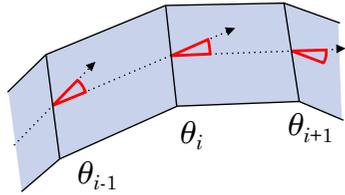


図 6 滑らかさの計算
Fig. 6 Evaluation of smoothness.

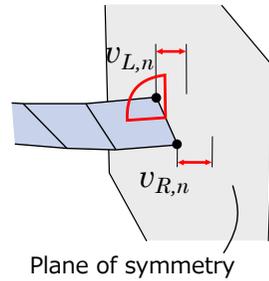


図 7 対称性の計算
Fig. 7 Evaluation of symmetry.

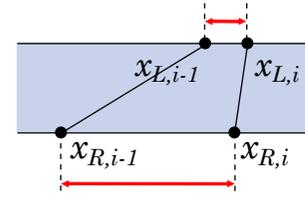


図 8 フェアネスの評価
Fig. 8 Evaluation of fairness.

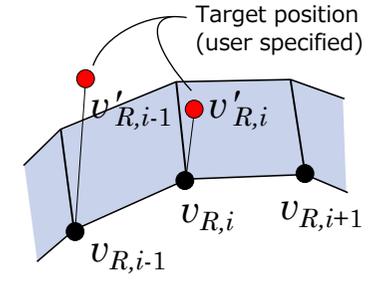


図 9 アンカーまでの距離
Fig. 9 Distance from vertex to its anchor.

士が近づきすぎないようにする。

$$F_{fair} := \sum_i \left(\frac{1}{x_{L,i} - x_{L,i-1}} + \frac{1}{x_{R,i} - x_{R,i-1}} \right) \quad (3)$$

ユーザによる形状指定 (図 9) 最後はユーザの意図を反映させるための項である。3次元空間で頂点の目標座標を指定している場合、その座標との2乗距離を計算する。ある頂点に対応するアンカー点を v' で表す。

$$F_{anchorpoint} := \sum_i \left((v'_{L,i} - v_{L,i})^2 + (v'_{R,i} - v_{R,i})^2 \right) \quad (4)$$

上記4項目を計算した後、それらの重み付き総和 $F_{penalty}$ を計算し、ペナルティ法を用いて最小化する。反復計算には Levenberg-Marquartz アルゴリズムを使用した勾配法を用いる。それぞれの係数はユーザが編集集中に変更できるが、正しく展開できるようにするためには $F_{symmetry}$ の重みを高く設定することが必要である。重みを変更しつつ動作させた結果、妥当な設定の一例は $(w_1, w_2, w_3, w_4) = (1.0, 3.0, 1.0, 2.0)$ であった。ただし、最終的な仕上げ段階では w_2 をさらに上げた方がよい結果が得られる。

$$F_{penalty} = w_1 F_{smooth} + w_2 F_{symmetry} + w_3 F_{fair} + w_4 F_{anchorpoint} \quad (5)$$

3.3 ユーザ入力

ユーザが入力するものは次の項目である。展開後の形を指定する数値 (図 3) ストリップの数、幅と長さ、脚部の長さの各数値を入力する。展開図全体の幅とスリットの間隔はストリップの開始位置を指定する際に計算される。

ストリップの開始位置 (脚部の位置) ストリップの開始位置を指定するために簡易な入力インタフェースを用意した。ユーザはベジェ曲線の制御点を編集することで、これを指定することができる。またベジェ曲線の制御点が修正される度に最適化処理が実行される。このベジェ曲線の長さが展開図での幅であり、全体の幅 $L1$ とストリップの幅 $L4$ 、本数 N から、スリットの幅 $W = (L1 - L4 \times N) / (N - 1)$ が求まる。ベジェ曲線は2つの3次ベジェ曲線を連結したものを使用しており、2つの曲線の端点にあたる3つの点から求まる法線により、ストリップの方向が決定される。

各頂点へのアンカー (目標位置) 位置を指定したい頂点を選択し、目標位置に移動させることでアンカーの指定ができる。一度指定したアンカー点は保持されるため、複数の頂点に対して同時にアンカーを指定することもできる。このアンカー点の指定中は、選択しているストリップのみに最適化処理が実行されるようにした。

細分割の実行 折り線の数を増やすことで、より詳細な編集が可能になる。新しい折り線は、元の各折り線の間中に角度0度として配置される (図 10)。ただし、細分割処理によって密になったストリップは計算時間が長くなってしまふ。そのため、粗い段階で大まかな形状を決めておき、その後実行することが望ましい。

4. 結果と考察

提案システムを用いて生成した形状を図 11 に示す。実装は Java を用いて、描画に JOGL、行列演算に EJML を使用した。最適化を用いるアプローチでは、生成した形状が必ず展開できることは保証されない。提案手法では、対称面に終端がどれだけ近いかによって形状の正しさが評価できる。そこで、すべてのストリップのうち、終端の頂点と対称面との最大距

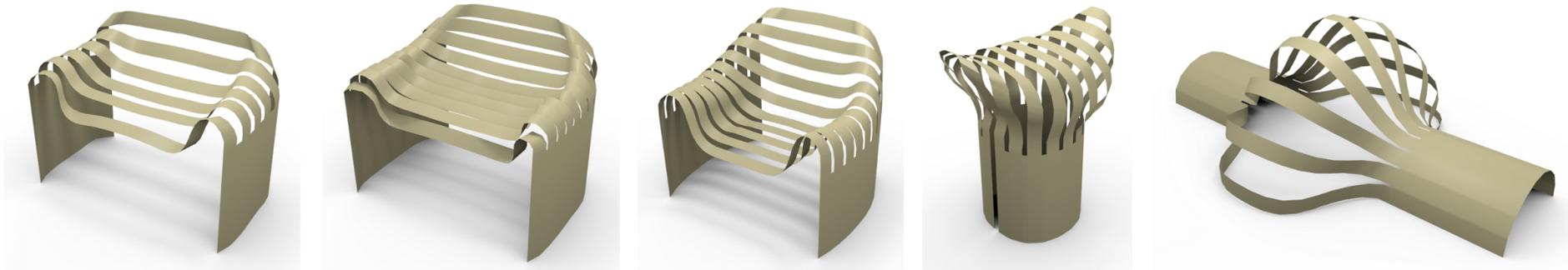


図 11 作例．ストリップ 6 本とストリップ 10 本の椅子モデル，測定点群に近づけるように最適化を加えたモデル（中央），モニュメントのデザイン（右 2 つ）．
Fig.11 Examples. 6-strip chair and 10-strip chair (left two), an optimized chair model with reference to scanned points (center), monuments (right two).

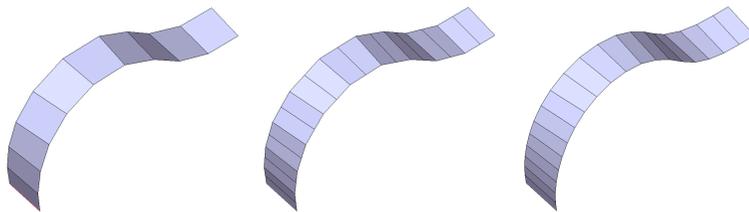


図 10 ストリップの細分割．左から分割前，分割後，最適化後．
Fig.10 Subdivision of a strip. From left, strip before subdivision, after subdivision, and after subdivision followed by optimization.

離 G を展開図全体の長さ $(L2 + L3)$ で割ることで誤差 $E = G / (L2 + L3)$ を計算した．その結果は，図 11 の例で最大 0.0015 であった．

この結果を得るための計算時間は，ストリップ 1 本あたり 9 分割（8 個の折り目，これを初期値とした）の場合で約 500ms，細分割後の 18 分割（17 個の折り目）で約 8s であった．ただし，最適化計算は初期状態に大きく依存するため，粗い分割の場合でも，作り始めの形状が大きく変化する段階では 5s 程度の時間を要した．逆に，大きな変化がない場合はより高速に動作した．実行速度の改善には，計算の高速化も必要ではあるが，計算回数を調整するインターフェイスをつけることで対応できると考えている．これはつまり，最適化処理がユーザの操作の度に行われるため，編集の途中段階では反復計算をより早い回数で打ち切ることが可能だからである．

図 11 の例題を生成するのに要した作業時間は 20 分程度であった．大まかな形を指定するのは容易だが，ストリップの波打や細かい位置修正は難しいことがわかった．例えば椅子の形であれば，座部を水平にしたいという要求がある．このような特定の部分に対する制約やそれを指定するインターフェイスがあれば，より効率的な作業が行えると思われる．また，本稿で参考にした図 1 の椅子に似せた形状を作ろうと試みたが，前述の理由により手作業での位置指定では大変な作業となった．そこで，紙で作られた図 1 のモデルを 3 次元計測器でスキャンし，得られた点群に近くなるように最適化を行った．

4.1 測定点群への近似

具体的な作業手順は次の通りである．

- (1) 測定した点群を単純化し，また孤立した面などを除去する．
- (2) ストリップ部分のみを個別のファイルに出力する．
- (3) それらのファイルをまとめて読み込み，ツール上で重ねて表示する（図 12）．
- (4) 大まかに手で位置を合わせた後，最適化を行う．

最適化は式 5 に以下の項を加えて行った．

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

ここで， $nearest(v_i)$ は測定点群に含まれる頂点のうち v_i に最も近い頂点である．この項を加えることにより，より元の椅子に近いモデルを生成することができた．ただし，提案手法で作られるのはあくまで理想的なモデルである．実際の制作物には面のねじれが含まれてい

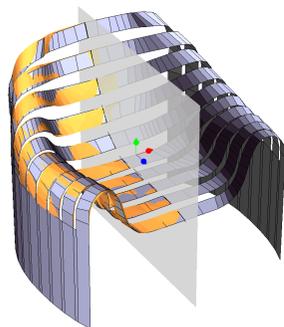


図 12 編集中の形と測定点群を重ねて描画
Fig. 12 Scanned data (orange) overlaid on the edited shape (blue).

と考えられ、測定した点群にはさらに測定時の誤差も含まれる。そのため、点群と完全に一致させることは不可能である。

5. まとめと今後の展望

本稿では、スリットを入れた平板素材を折り曲げてできる形の設計について述べた。最適化計算によって折れ曲がる箇所の位置と角度を計算することで、展開可能な目的の形を達成している。提案したツールを用いることで、計算機上でこれまで実現することが困難だった形状を構築することが可能となる。

提案手法で扱うことができる展開図は大きく限定されたものである。よりバリエーションのある形を対象とするためには、スリットの位置や幅をずらすことが考えられる。本稿で提案した幾何モデルで任意の多角形の素材とスリットを取り扱うためには、展開図の形とスリットの位置、そして折れ曲がる部分を指定するためのインターフェースが必要である。

本稿では椅子を題材としてシステムを構築したが、現在のところ実際にものづくりをするまでには至っていない。建築家の手によって作られた図1の椅子は、最初に型を作り、その型に素材を押し付けるようにして形成した。出力として得られた形状からさらに型を生成することで、より実用的な方向に進められると考えている。また、そうした場合には強度やバランスなどの解析を同時に行い、最適化計算に組み込むことで、新しいCADシステムとしての貢献ができると思われる。

参考文献

- 1) Naruse, Y. and Inokuma, J.: NARUSE INOKUMA ARCHITECTS, <http://www.narukuma.com/main.html>.
- 2) Mitani, J. and Suzuki, H.: Making papercraft toys from meshes using strip-based approximate unfolding, *ACM Trans. Graph.*, Vol.23, pp.259–263 (2004).
- 3) Massarwi, F., Gotsman, C. and Elber, G.: Paper-craft from 3D polygonal models using generalized cylinders, *Comput. Aided Geom. Des.*, Vol.25, pp.576–591 (2008).
- 4) Liu, Y., Pottmann, H., Wallner, J., Yang, Y.-L. and Wang, W.: Geometric modeling with conical meshes and developable surfaces, *ACM Trans. Graph.*, Vol.25, pp. 681–689 (2006).
- 5) Pottmann, H., Schiftner, A., Bo, P., Schmiedhofer, H., Wang, W., Baldassini, N. and Wallner, J.: Freeform surfaces from single curved panels, *ACM Trans. Graph.*, Vol.27, pp.76:1–76:10 (2008).
- 6) Rose, K., Sheffer, A., Wither, J., Cani, M.-P. and Thibert, B.: Developable surfaces from arbitrary sketched boundaries, *Proceedings of the fifth Eurographics symposium on Geometry processing*, Aire-la-Ville, Switzerland, Switzerland, Eurographics Association, pp.163–172 (2007).
- 7) Kilian, M., Flöry, S., Chen, Z., Mitra, N.J., Sheffer, A. and Pottmann, H.: Curved folding, *ACM Trans. Graph.*, Vol.27, pp.75:1–75:9 (2008).
- 8) Mitani, J. and Igarashi, T.: Interactive Design of Planar Curved Folding by Reflection, *Pacific Conference on Computer Graphics and Applications - Short Papers*, Kaohsiung, Taiwan, pp.77–81 (2011).
- 9) Bo, P. and Wang, W.: Geodesic-Controlled Developable Surfaces for Modeling Paper Bending, *Comput. Graph. Forum*, Vol.26, No.3, pp.365–374 (2007).
- 10) Grinspun, E., Desbrun, M., Polthier, K., Schröder, P. and Stern, A.: Discrete Differential Geometry: An Applied Introduction, Siggraph 2006 Course, <http://ddg.cs.columbia.edu/SIGGRAPH06/Intro.pdf>.