

# 折り紙の構造把握のための形状構築とCG表示

三 谷 純<sup>†</sup> 鈴 木 宏 正<sup>††</sup>

折り紙の形状を計算機内に構築する際には、紙の厚みを無視した平面の集合で形状を表現することが一般的である。しかし、それを元に折り紙の形状をCG表示した場合、紙が複数枚重なっている部位も厚みを持たない平面として表示されるため、その構造を理解し難いという問題がある。一方で、折り紙の教本に見られる折り方を説明した「折り図」では、形状の特徴を把握しやすいように、紙の厚みを強調したり、角の重なりを意図的にずらして描くなどの工夫が見られる。そこで本稿では、平面の集合で表現された折り紙の形状データに対し、その構造の把握を容易にするための表示を行う手法を提案する。本稿で提案する手法では、レンダリングの前に、形状自体を再構築する点に特徴がある。具体的には、厚み方向への面の再配置、面と面の接続の表現、および頂点位置の変更によるずれの表現を行い、ノンフォトリアスティックレンダリングの表示手法を用いる。これにより、複数の平面多角形が同一平面上で重なっているような、特に折り紙の形状に多く見られる3次元構造を容易に理解できるCG表示を実現できる。

## Model Construction and Rendering for Understanding the Conformation of Origami

JUN MITANI<sup>†</sup> and HIROMASA SUZUKI<sup>††</sup>

It is common to use a set of planar polygons for expressing a figure of Origami to construct and store it in a computer. The main problem when such models are displayed, is that understanding the conformation is difficult. It is because all faces do not have thickness, and duplicated polygons on a same plane are seen as if there is only a single polygon. On the other hand, illustrations on Origami drill-books are drawn with deformations so that they can be easily understood. These deformations are achieved by emphasizing the thickness of paper or sliding a corner of paper that overlaps other corners so that we can understand the duplication of faces. In this paper, we propose a new method for reconstructing and rendering an Origami model that are expressed as a set of planer polygons so that users can understand the conformation easily. In our approach, we move faces to the direction of their normals for the expression of thickness and we add new bridge faces for the expression of connections between faces. Furthermore, we move vertices that overlap other vertices so that the duplication can be recognized. Then we render them like illustrations. CG images generated by this approach are not accurate but quite easy to understand the conformation of Origami.

### 1. はじめに

折り紙は日本に古くから伝わる文化として世界でも広く知られ、教育の場での活用や趣味の1つとして世代を問わず多くの人々に親しまれている。折り紙を数学的な側面からとらえる研究は古くからされているが、近年では計算機環境の発達と普及により、折り紙を計算機で扱う研究も多くされている。

折り紙の形状を計算機内に構築する場合には、紙の

厚みをゼロと仮定して、平面の集合で表現することが一般的である。しかし、紙の厚みは最終的に折り上がる折り紙の形状に大きな影響を与え、複数枚の紙が重なる場所では、実際には紙の歪みや角のずれが発生する。完全に厚みを無視したモデルでは、このような現象が発生しないため、そのままCG表示しても現実味に乏しく、その構造を正しく理解することは難しい。そこで本研究では、平面の集合で表現された折り紙のデータを元に、その構造を理解しやすいような表示を行う手法を提案する。

折り紙を忠実に表現するためには、厚みを持った形状モデルを計算機内に保持し、有限要素解析などによる、高度な物理シミュレーションを応用したモデリング手法が考えられる。しかし、折り紙の作り方や仕組

<sup>†</sup> 独立行政法人理化学研究所

The Institute of Physical and Chemical Research

<sup>††</sup> 東京大学東京大学先端科学技術研究センター

Fine Digital Engineering, RCAST, The University of Tokyo

みを把握するうえで、写真よりも教本に見られるような「折り図」を見た方が分かりやすいことから、構造の理解しやすさという点では、単にモデリングやレンダリングの精度が問題になっているわけではない。これは、ノンフォトリアスティックレンダリングという、意図的に写実的な表示を行わず、特徴を把握しやすいように輪郭や特徴線を強調して表示する手法の有効性が、近年 CG の世界で認められていることからいえる。

ところで、従来のノンフォトリアスティックレンダリングの手法では、形状は変形させずに、その表示手法についてのみ議論されているが、折り紙のデータに多く現れるような同一平面上で重なっている多角形は、そもそもどのように表示しても識別が難しいという問題がある。そこで本研究では「正確さ」ではなく「分かりやすさ」を重視し、構造が理解しやすいように形状の再構築を行い、それに続いてノンフォトリアスティックレンダリングに基づく表示を行う手法を提案する。

なお、本研究は折り紙の表示方法について議論を行うものであるため、折り紙モデルのデータ構造やモデルの構築方法などについては Miyazaki ら<sup>1)</sup>の既存の研究成果を利用した。また、本研究では特に折り紙を対象としているが、本手法は「構造を理解し難い3次元形状に対して、何らかの形状操作を行うことで、分かりやすい表示を行う」という、従来のノンフォトリアスティックレンダリングとは異なる、新しいパラダイムの提案でもある。

本稿の2章では関連する研究を紹介し、3章では本研究で用いた Miyazaki らの提案する折り紙モデルの構造について述べる。4章では、本手法の詳細を述べ、5章で結果をまとめる。最後に6章で結論と今後の課題を述べる。

## 2. 関連研究

日本に古くから伝わる様々な折り紙の手法は、平面を折り曲げることによって様々な形を作るものであり、数学の幾何の分野における研究題材として多く取り上げられている<sup>2),3)</sup>。また、「折り紙の科学国際会議」が1989年と1994年に開催されるなど、折り紙に関する研究は国際的にも広く行われている。

折り紙を計算機で扱う研究として、内田ら<sup>4)</sup>は紙の物理的な制約条件を元に、折り紙の展開図を構成する幾何学的要素から出来上りを推論するプログラムを提案した。このプログラムの中で、面や辺、頂点などの折り紙を構成する要素を定義し、折り操作による面

の二分割を、二分木によるデータ構造を用いて表現するなど、折り紙のモデル化の一手法を提案している。Miyazaki ら<sup>1)</sup>は、計算機を用いて折り紙を対話的に操作する手法を提案した。紙を折る操作によって折り紙の形状が逐次変化する際の、データ更新の手法を提案している。折り紙の形状を計算機内に保持するデータ構造はシンプルであり、折り手順を含めた情報が、二分木を用いて計算機内に保持される。ここで提案されたデータ構造を含むプログラムのソースコードは Web 上で公開されており<sup>5)</sup>、本研究で提案する手法の実装では、これを使用した。このデータ構造は3章で詳しく述べる。横山ら<sup>6)</sup>は、厚さを持った紙の折り曲げのシミュレーションを行い、その結果の「厚さ」と「ずれ」を実際のコピー紙および画用紙と比較している。折りたたまれた紙を計算機で表現する際の、紙の厚さとずれに着目している点は我々と共通するが、それをどのように CG 表示するかについての議論はされていない。また、単純に等分割で紙を折った場合の比較実験しかされていない。Kato ら<sup>7)</sup>は、教本に見られる「折り図」の画像を計算機で解析することで、折り曲げ操作を推定し、それを元に計算機内の折り紙モデルを更新する手法を提案した。我々が、計算機内のモデルから形状表示を行うのとは逆のアプローチである。

ところで、計算機内に構築されたモデルを意図的に写実的でない表示を行うノンフォトリアスティックレンダリングの手法が、近年の CG の分野では多く研究されている。Markosian ら<sup>8)</sup>は、投影時の輪郭を効率的に検索することで、面の数が多いメッシュモデルに対しても、リアルタイムに線画風の表示を行う手法を提案した。Gooch ら<sup>9)</sup>は、白色物体に色を持つ光を当てる手法で絵画調やテクニカルイラスト調の表示が行えることを示し、ノンフォトリアスティックレンダリング用の新しい光源モデルを提案した。Rademacher<sup>10)</sup>は、メッシュモデルを漫画調にレンダリングするために Gooch らの手法を応用している。

これらのレンダリング手法には、人間の手で描画されたような柔らかさみや温かみを表現できるという利点や、細部を省略し特徴を強調することで、全体的な形状を理解しやすいという利点がある。

## 3. 折り紙モデル

本章では Miyazaki らによって提案され、本研究の実装でも使用されている折り紙モデルのデータ構造について述べる。折り紙を構成する要素と用語を表した

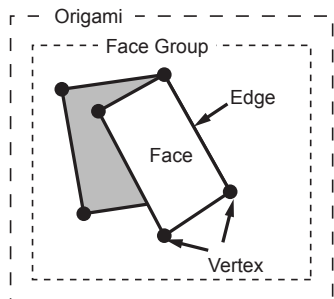


図 1 折り紙を構成する要素  
Fig. 1 Elements of Origami model.

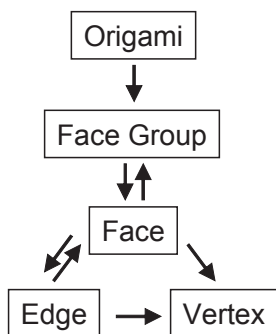


図 2 折り紙モデルのデータ構造  
Fig. 2 Data structure of Origami model.

ものが図 1 で、折り紙のデータ構造を模式的に表したものが図 2 である。

折り紙全体を表す Origami には、同一平面上に乗る面をまとめた Face Group が 1 つまたは複数存在する。1 つの Face Group には、同一平面上に乗る面 Face が重なり順に格納される。これらの面からは、自身が属する Face Group を参照可能である。Face には、面を構成する頂点 Vertex と、辺 Edge への参照が面の外周を巡回する順に格納される。Vertex には、その頂点の座標値が保持され、Edge からは自身を含む Face (1 つまたは 2 つ) を参照可能である。

図 1 の例では、Face Group が 1 つしか存在しないが、たとえば直角に折り曲げた面が存在する場合には、それは他の面と同一平面上に乗らないため、別の Face Group に含まれる面として表すことになる。

4. 折り紙モデルの再構築と CG 表示

本研究で対象とする折り紙モデルは 3 章にまとめたように、面の集合で表現される。この面を構成する頂点の座標値をそのまま用いて CG 表示した場合、重なり合う面が同一平面上に配置されることから、その構造を理解し難いという問題がある。そこで、我々は次のような操作を行うことで、折り紙の構造を理解し

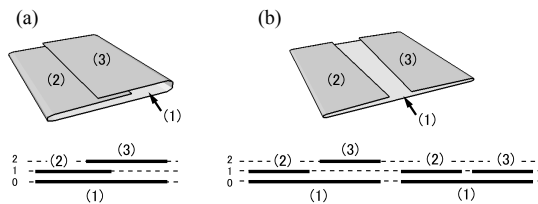


図 3 厚さ方向への面の配置変更  
Fig. 3 Face layout on third dimension.

やすい表示を行う手法を提案する。

- (1) 厚さ方向への面の配置変更
- (2) 同一平面上の面どうしの接続部生成
- (3) 同一平面上にない面どうしの接続部生成
- (4) 重なり頂点の移動
- (5) レンダリング

以降の各節では、上記の内容についてそれぞれの詳細を記す。

4.1 厚さ方向への面の配置変更

紙を厚さゼロの平面で表し、それを CG 表示する場合でも、紙の重なり合いが存在しないのであれば大きな問題はないが、複数の面が折り重なっている場合、面と面の位置関係や、そもそもいくつの面が重なっているのかを把握できない。そこでまず、複数の面が重なる場合には面の位置を紙の厚さ方向(面の法線方向)へ一定量だけ移動させ、それぞれの面を識別できるようにする。

本研究で用いるデータ構造では、同一平面に乗る面を重なり順に取得できるため、単純に考えれば紙の厚さを  $N$  したときに、 $i$  番目の面を  $iN$  だけ厚さ方向に移動させればよい。図 3 (a) の上段は、面 (1) ~ (3) が互いに重なっている例であり、それぞれを順番にずらして配置した場合の断面図を下段に示す。

しかし順番に取得できる面は 1 つずつであるため、図 3 (b) の上段のように、互いに重なる部分を持たない面が存在する場合に問題が生じる。面 (2), (3) は個別に (1) と重なっているため、厚さ方向では同じ位置になるべきであるが、順番に一定量ずつずらす方法では、それぞれの位置が異なってしまう(図 3 (b) 下段左)。そこで、重なり順に取得した面  $F$  の配置場所を決定する際には、すでに配置場所を決定した面で、なおかつ面  $F$  と重なりを持つ面の中で、一番上に配置されているものを探し出し、その配置場所より 1 単位大きな場所に配置する、ということを行う。この方法により、図 3 (b) 下段右に示すような配置を得ること

上段の紙が折り曲げられている様子の図は、本研究で提案する手法によって描画した。

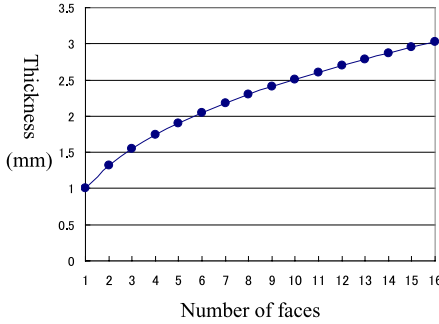


図 4 重なり枚数と全体の厚さの関係

Fig. 4 Relation between the number of sheets and the whole thickness.

ができる。

ところで、折り紙を 2 つ折りする操作では、全体の大きさが約半分になる一方で、厚さが 2 倍に増えるため、紙の厚さ  $N$  が一定であると、重なりが多くなると極端に厚みが生じて違和感が生ずる問題がある。文献 6) の研究では、紙の厚みを次式で表現すると実際の測定値に近い値が得られることが示されている。

$$x = d \cdot a^{i-1} \tag{1}$$

式 (1) において、 $x$  は紙 1 枚の厚さ、 $d$  はデフォルトの紙の厚さ、 $a$  は折り曲げた紙の間に存在する紙の枚数、 $i$  は実際値とのずれを調整するパラメータである。文献 6) では、 $i = 0.8$  のときに実測値に近い結果が得られたと記述されている。本研究においては「それらしく見える」ことを重視し、試行の結果 1 辺が 10 cm の折り紙を使用した場合に、 $a =$  (同一平面上に乗っている面の数)、 $d = 1.0$  (mm)、 $i = 0.4$  として、紙 1 枚の厚さ  $x$  を決定することとした。この場合の、枚数と厚さの関係は図 4 に示すグラフのようになり、枚数が多くなるにつれ全体の厚みの増分が小さくなる。

#### 4.2 同一平面上の面どうしの接続部生成

紙の厚みを考慮し、面どうしを離して配置することで面の重なりを把握できるようになるが、それぞれがどのような接続関係にあるかを理解することはできない。そこで互いに接続関係にある面に対して、その関係が分かるように継ぎ手を表現するための面の追加を行う。

##### 4.2.1 接続辺の表現

たとえば 2 つの面が重なっている場合、図 5 (a) のように、面の間に隙間を設けることで面の重なりを把握できる。さらに図 5 (b) のように、接続関係にある辺どうしの上に継ぎ手を表現する長方形の面を追加すると、2 つの面の接続関係を把握できる。しかし、図 5 (c) のように複数の継ぎ手が重なる場合、接続の構

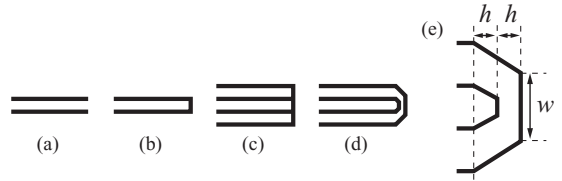


図 5 複数の面が重なる場合の接続辺の表現

Fig. 5 Representation of bridges between topologically connected faces.

```

全ての辺 E について、
E.connectionHeight = 0 を設定する
重なり順に面 F を取得し、次の処理を行う {
  F に含まれる各辺 E について、次の処理を行う {
    maxHeight = 0
    E を介して F に接続する面 F' を取得する。
    もし F' が存在し、
    なおかつ F' の方が F よりも下にあったら、
    F と F' の間に位置する面 F'' の
    各辺 E' について次の処理を行う {
      もし E' と E が互いに重なる場所にあり、
      なおかつ E' を介して 2 つの面が
      接続している場合は次の処理を行う {
        maxHeight =
          max(maxHeight, E'.connectionHeight)
      }
    }
    E.connectionHeight = maxHeight + 1
  }
}

```

図 6 接続辺を台形で表現した場合の台形高さの決定アルゴリズム  
Fig. 6 Algorithm that determines the height of trapezoids for edges.

造が分からない。そこで図 5 (d) のように、接続部分が台形となるような面で継ぎ手を生成することとする。

ここで問題となるのは、台形の高さの決定方法である。本研究では、図 5 (e) に示すように、一番内側の台形の高さを  $h$  とした場合、その辺をまたいで接続する辺の継ぎ手の高さは  $2h$  となるようにした。つまり、間にある継ぎ手の高さの最大値が  $nh$  であった場合、 $(n + 1)h$  となるようにする。

この値は図 6 に示すアルゴリズムで決定できる。なお、アルゴリズム中の connectionHeight が高さを  $nh$  で表したときの  $n$  に対応する  $h$  は任意の定数を使用できるが、本研究の実装では、4.1 節で求めた 1 枚の紙の厚さの 2 分の 1 を使用した。また、台形の上辺の幅 (図 5 (e) 中の  $w$ ) は、(底辺の長さ)  $\times 0.6$  とした。

##### 4.2.2 接続点の表現

紙を複数回折ることによって生じる折れ線どうしの交わり箇所では、紙の内部に頂点が生じる。前項で述べた手法で接続辺に台形の面を生成すると、この頂

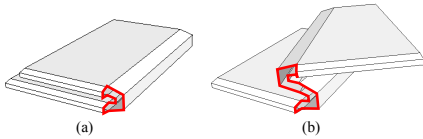


図 7 内部頂点に生じる穴  
Fig. 7 Holes that appear at inner vertices.

点部分に隙間が生じてしまう問題がある。たとえば図 7 (a) は紙を 4 つ折りにしたものであるが、図中の太線で囲まれた領域に穴が生じる。さらにこの頂点で別の折れ線が交わる場合には図 7 (b) のような複雑な形の穴が生じる。そのため、定型的な方法をあらかじめ準備し、それによって対応することは難しい。

ポリゴンモデル上に存在する穴を埋める手法は、レンジデータからモデルを構築する際に必要となるため「Hole Filling」の問題として多くの研究がされている<sup>11)</sup>が、実装はそれほど容易ではない。ところで、図 7 では面の厚みを大きめにとり、接続辺を大きくすることで頂点部分を強調しているが、実際に生じる穴の形状が全体の外観に占める割合はきわめて小さい。そのため、本研究では考えうる最も容易な対応法として、穴の外周を輪郭とする多角形面（非凸で平面でない場合がある）を生成し、そのレンダリングは OpenGL ライブラリ<sup>12)</sup>に委ねることとした。この手法は、きわめて単純なアプローチであるため、得られる結果の形は良くないが、反対側が筒抜けで見えてしまうことは防ぐことができ、後に述べるノンフォトリアリスティックレンダリングでは、輪郭と境界線のみを表示するため、表示上、特に大きな問題とはならない。

4.3 同一平面上にない面どうしの接続部生成

ある辺を介して互いに接続する面 A と A' が異なる平面上にあるとき、その断面方向から見た様子は図 8 (a) のようになる。これらが複数の面の重なりである場合、4.1 節で述べたように厚みを考慮して面の配置場所を変更するため、図 8 (b) のように互いに交差したり、辺と辺の間に隙間が生じたりする問題が生じる。

このような問題に対応するために、図 8 (b) 中の B と B' のように互いに接続関係にある面が干渉する場合には、互いの面でトリムを行い、A と A' のように隙間ができる場合には、互いの辺の間に新しい継ぎ手を生成し、その隙間を埋めるようにする（図 8 (c)）。図 8 では、2 つの面が重なっている場合の例を示しているが、3 つ以上の面が重なる場合にも同様の処理を行う。

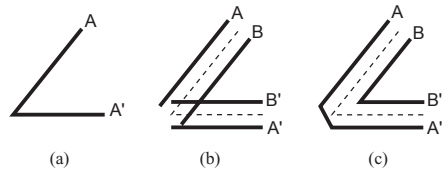


図 8 同一平面上にない面どうしの接続部生成  
Fig. 8 Creation of bridges between faces not lying on the same plane.

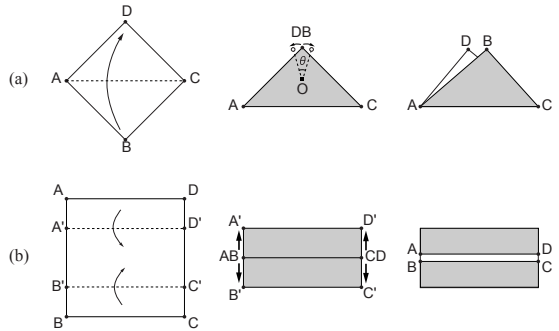


図 9 重なり頂点の移動  
Fig. 9 How to move the positions of duplicated vertices.

4.4 重なり頂点の移動

同一形状の面が互いに重なっている場合、面の法線方向から眺めたときには面の重なりを把握できない。また、複数の辺が同一の位置に存在する場合も構造の把握が難しくなる。これらの問題は、異なる頂点と同じ位置に存在することが原因となっている。

ところで、折り紙の教本に見られる「折り図」では、重なる位置にくる頂点を意図的にずらすことで、複数の面が重なっている様子を表現することが多い。このようにすることで複数の面の存在を視覚的に確認できる。また、複数の辺が重なる場合は、わざと隙間を作ることで折り方を把握しやすくすることも多い。そこで、真上（面の法線方向）から見たときに同一位置にくる頂点がある場合、意図的に頂点の位置をずらすことを行う。本研究では、頂点が重なって見える場合を「3次元空間内では異なる位置にあるが）面の法線方向から見たときの頂点の重なり」と「3次元空間内で同じ位置にある頂点の重なり」の2通りに分け、それぞれ次のような方法で頂点を移動させることとする。

面の法線方向から見たときの頂点の重なり

同じ FaceGroup に属し、面の法線方向から見たときに重なり合う頂点を、FaceGroup 内の面の外接四角形の重心を中心とした回転によって移動させる。

たとえば図 9 (a) のように、正方形 ABCD を対角線 AC で 2 つ折りにした場合、頂点 D と B が同じ位置に重なる。これを、法線方向から見たときに異なる

位置になるように、同じ FaceGroup に属する面の外接四角形の重心  $O$  を中心に、互いに一定角だけ回転させてずらすこととする。同じ位置に  $N$  個の頂点が重なっている場合、法線方向から見て下から  $i$  番目の頂点は  $(i - N/2)\theta$  だけ時計回りに回転させる。これにより、元の位置を中央に、角  $\theta$  の間隔でずれた位置にそれぞれの頂点を配置することができる。なお、本研究では実装の際に、この角度  $\theta$  を  $4^\circ$  とした。

### 3 次元空間内で同じ位置にある頂点の重なり

3 次元空間内で同じ位置にある頂点が存在する場合、それぞれの頂点が属する面に含まれる辺の中で、他の面と接続する辺に下ろした垂線の方角へ移動させる。

たとえば図 9 (b) のように、正方形 ABCD の辺 AD と辺 BC が一致するような折り方をした場合、頂点 A と B, D と C がそれぞれ同じ位置になる。これらを、異なる位置になるように、面 A'ADD' に含まれる頂点 A と D は、他の面 (面 A'B'C'D') と接続する辺 A'D' に下ろした垂線の方角へ移動させる。同様に、面 BB'C'C に含まれる頂点 B と C は、他の面 (面 A'B'C'D') と接続する辺 B'C' に下ろした垂線の方角へ移動させる。他の面と接続する辺が複数ある場合は、それぞれへ垂線を下ろした方向ベクトルの和をとることとする。なお、本研究では実装の際に、折り紙の 1 辺の長さを 10 cm としたときに、この移動量を 2.5 mm とした。図 9 (b) のように、互いに離れる方向に移動する場合には 5 mm の隙間が生じることになる。

### 4.5 レンダリング

前節までの手法で、折り紙の構造を容易に理解するための形状モデルを計算機内に構築することができる。このモデルを CG 表示することで、ユーザに折り紙の立体形状を提示する。本研究で扱う対象は折り紙であり、また、教本のようにグレースケールでもその構造が理解しやすい表示を行いたいため、以下に示す単純なアルゴリズムでレンダリングを行うこととした。

- 面の描画は、RGB 値が同値のグレースケールで行い、それぞれの面を単色で描画する。なお、表示色は面の法線ベクトル  $\vec{N}$  と視線ベクトル  $\vec{E}$  の内積をもとに、RGB の各値 (0.0 ~ 1.0) を次のように決定する。
  - 表面 (図 10 中の Front Face):  $0.7\vec{N} \cdot \vec{E}$
  - 裏面 (図 10 中の Back Face):  $0.5(1 + \vec{N} \cdot \vec{E})$
- 形状モデルの境界となる辺 (図 10 中の Boundary Edge) を太い黒の実線で描画する。
- 表示の際に輪郭となる辺 (図 10 中の Silhouette Edge) を太い黒の実線で描画する。特定の辺が

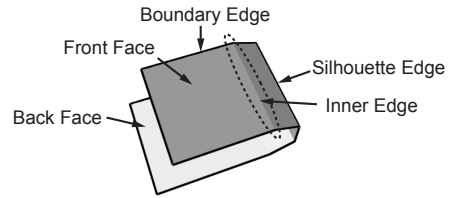


図 10 レンダリング  
Fig. 10 Rendering.

表示の際に輪郭となるか否かは、辺に隣接する 2 つの面の法線を調べ、それぞれの向きが視点から見て反転しているか否か (法線ベクトルと視線ベクトルの内積の符号) で判別できる。ところで、4.2.2 項で述べたように、面の折れ線どうしが交わる箇所では、穴埋めのための面の生成を厳密に行っていないため、つねに描画対象からはずすこととした。

- 形状モデル内の、上記以外の辺 (図 10 中の Inner Edge) は描画を行わない。

## 5. 結 果

本研究で提案する手法を実装した結果を図 11 に示す。図中 (a) は厚みを考慮しない従来の表示方法であり、(b) は本手法によって再構築されたモデルである。ただし頂点の移動は行っていない。(c) と (d) は再構築されたモデルに対して 4.5 節で述べたレンダリングを行ったものであるが、(d) は (b) に対して 4.4 節で述べた頂点の移動によるずれの表現を追加している。

モデル A, B, C は、従来の手法では構造把握が困難な例として作成したものであり、D, E は Miyazaki らが Web 上で公開しているサンプルデータの「兜」と「飛行機」の例である。E 以外はすべての面が同一平面上に乗っているため、1 つの Face Group で表現されている。

## 6. 結論と今後の展望

複数の頂点の重なりを持つ図 11 の A, B, C は、従来の表示方法 (a) では、その構造を把握することが非常に難しいが、本研究で提案する手法により、その構造を理解しやすい表示 (d) を行えた。これらは立体形状として 3DCG 表示されているため、PC 上では任意の角度から眺めて形状を確認できる。D, E のような複雑な形状においては、構造を完全に理解することは難しいが、面が複数回折りたたまれて形が作られていることを認識できる。E の形状は厚紙で作ったようにも見てとれるが、これは式 (1) で定めた厚さの値によって調整可能である。結果の図 11 より、本研究で

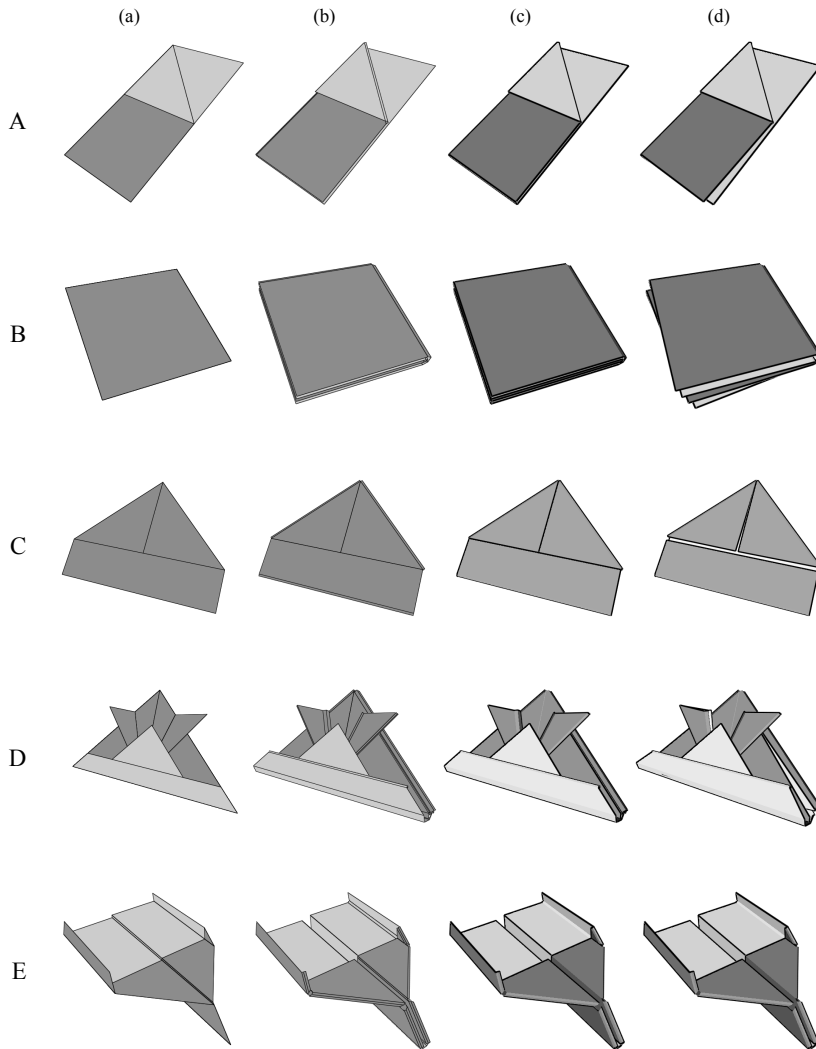


図 11 結果  
Fig. 11 Results.

提案する手法により，従来は構造を把握することが困難であった折り紙モデルを，その構造を理解しやすいようなCG表示を行うことができたといえる．

本研究では，一貫して実装の容易な手法を提案してきたが，4.2.2 項で述べたように，折り線が交わる箇所にてできる穴埋めを厳密に行ったり，レンダリングの手法を改良したりなど，さらに改善が可能な点は多く残っている．また，図 12 に示すような構造で，点 A と D，B と C がそれぞれが同一の場所になったときに，面の接続を表現するために追加する継ぎ手の面どうしが干渉するという問題がある（この問題は，結果の D の中央上部に見られる）．このような特定の条件下で発生する個別の問題について，どのように対応してゆくかは今後の課題である．

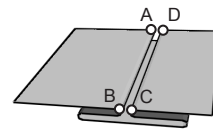


図 12 辺を近づけたときに接続辺に生成した面が干渉するケース  
Fig. 12 An example where faces intersect when two edges approach.

複雑な形状については，それぞれの折り過程の中間モデルを連続して表示することでアニメーションによる教示も可能である．ただし，アニメーション表示の際には折り畳み途中で生じる可能性のある面と面の干渉なども考慮する必要があるだろう．さらに本研究で扱った形状は平面の集合で表現されるものに限られるが，

曲面を含む折り紙の扱いについても今後の課題である。しかし従来の折り紙の研究のほとんどが平面の集合で表現されることを前提としているため、曲面を扱う際にはその形状表現の方法についても研究する必要がある。

また、Rademacher<sup>10)</sup> はキーフレームの手法を用いて視点に応じて形状を変化させる手法を提案しているが、その例にならって、視点に応じて最も構造が分かりやすいように形状を変化させることも、研究の価値があるであろう。また、折り鶴に見られるような、折り終わった後の「膨らみ」の表現も大きな課題である。さらに、対象を折り紙に限定せずに「分かりにくい構造のものを、形状変形を用いて理解しやすく表示する」という課題は、今後さらなる研究が必要なテーマである。

謝辞 本研究では、Web 上で公開されている宮崎慎也氏の折り紙シミュレータのプログラムコード<sup>5)</sup> を使用させていただいた。ここに感謝の意を表す。

### 参 考 文 献

- 1) Miyazaki, S., Yasuda, T., Yokoi, S. and Toriwaki, J.: An Origami Playing Simulator in the Virtual Space, *The Journal of Visualization and Computer Animation*, Vol.7, No.1, pp.25-42 (1996).
- 2) 川崎敏和：バラと折り紙と数学と，森北出版株式会社 (1998).
- 3) 深川英俊：折紙の数学，森北出版株式会社 (2002).
- 4) 内田 忠，伊藤英則：折り紙過程の知識表現とその処理プログラムの作成，情報処理学会論文誌，Vol.32, No.12, pp.1566-1573 (1991).
- 5) 宮崎慎也：折り紙シミュレータのページ。  
<http://www.sccs.chukyo-u.ac.jp/miyazaki/origami/indexj.html>
- 6) 横山卓弘，高井昌彰：厚さを持った折り紙シミュレーションとその評価，グラフィックスと CAD，101-4, pp.19-24 (2000).
- 7) Kato, J., Watanabe, T., Hase, H. and Nakayama, T.: Understanding Illustrations of Origami Drill Books, *情報処理学会論文誌*，Vol.41, No.6, pp.1857-1873 (2000).
- 8) Markosian, L., Kowalski, M.A., Trychin, S.J., Bourdev, L.D., Goldstein, D. and Hughes, J.F.: Real-Time Nonphotorealistic Rendering, *Computer Graphics (SIGGRAPH 97)*, pp.415-420 (1997).
- 9) Gooch, A., Gooch, B., Shirley, P. and Cohen, E.: A Non-Photorealistic Lighting Model For Automatic Technical Illustration, *Computer Graphics (SIGGRAPH 98)*, pp.447-452 (1998).
- 10) Rademacher, P.: View-Dependent Geometry, *Computer Graphics (SIGGRAPH 99)*, pp.439-446 (1999).
- 11) Wang, J. and Oliveira, M.M.: A Hole Filling Strategy for Reconstruction of Smooth Surfaces in Range Images, *XVI Brazilian Symposium on Computer Graphics and Image Processing*, pp.11-18 (2003).
- 12) Board, O.A.R.: OpenGL リファレンスマニュアル第 2 版，ピアソン・エデュケーション (1999).

(平成 16 年 7 月 20 日受付)

(平成 16 年 11 月 1 日採録)



三谷 純 (正会員)

1975 年生。2000 年東京大学大学院工学系研究科情報工学専攻修士課程修了，同博士課程進学。2004 年東京大学大学院工学系研究科博士課程修了・工学博士，独立行政法人理化学研究所基礎科学特別研究員，現在に至る。コンピュータグラフィックスおよび計算機を用いたペーパークラフトの設計支援に関する研究に従事。精密工学会，日本図学会各会員。



鈴木 宏正 (正会員)

1957 年生。1986 年東京大学大学院工学系研究科博士課程修了・工学博士。1987 年東京大学助手 (教養学部)，1988 年同講師，1989 年同助教授。1994 年同大学院工学系研究科精密機械工学専攻助教授，2003 年同教授，2004 年同先端科学技術研究センター教授，現在に至る。この間，機械設計 CAD のための製品情報モデリングや形状モデリングに関する研究に従事。2000 年より (株) エリジオン取締役を兼業。精密工学会，ACM，IEEE 等の会員。