

## 平坦折紙のための重なり順に閉ループを持つ平面集合のレンダリング手法

三谷純

筑波大学大学院システム情報工学科, 科学技術振興機構さきがけ

### 概要

平坦折紙（折りたたみ後の形が平坦になる折紙）には、折りたたみ後の面の重なり順に閉ループを持つものが存在する。CGにおいて一般的に用いられる厚さゼロの平面多角形の集合でこの折紙を表現した場合、この形状を正しく画面に表示することは難しい。本稿では、この問題を解決するための新しいレンダリング手法を提案する。具体的には、面の重なり関係を表す行列とzバッファ法におけるzバッファと似た概念の面IDバッファを準備し、レンダリング時に画素単位で最も上面に位置する面を管理することを行う。スキャンライン法を用いて面IDバッファに対して描画を行い、エッジ抽出を行った結果を画面に出力することで、折りたたみ後の形状表示を行う。また、頂点を挟む2辺の山谷に着目した頂点カラーの設定を行い、イラスト調のレンダリングを行う方法も提案する。さらに、面の重なり関係を表す行列を構築するために、展開図から面の重なり関係を推定する簡易的な手法も提案する。

## Rendering method of plane set with closed-loop in overlap order for flat folding Origami

Jun Mitani

Department of Computer Science, University of Tsukuba, PRESTO, JST

### abstract

In flat folding Origami (Origami which are folded into flat), the cases are exist that the relation of face-overlapping is not ordered (which has closed-loop in overlapping order). It is difficult to display this shape on the screen correctly when this paper folding is expressed by sets of plane polygons of zero thickness generally used in CG. In this paper, we propose a new rendering technique to solve this problem. In our method, we prepare a matrix that represent the overlapping relation between two faces and a face ID buffer which concept is similar with Z buffer in the z-buffer algorithm. With this buffer, the face located in the uppermost is monitored in each pixel at rendering stage. We render the shape on the ID buffer using scanline algorithm, and output the result by extracting the edges. Moreover, we set color to each vertex according to the number of mountain/valley folds connected to the vertex, it makes the result of rendering like human designed illustration. In addition, we propose a simple method that build a overlapping relation from the unfolded pattern.

## 1 はじめに

折紙は一枚の紙を折りたたむことで様々な形を作り出すものであり、伝統的な遊びとして幅広い世代に親しまれている。一方で、折紙の幾何に着目した研究も広く行われ、そこで得られた知見が人工衛星の太陽電池パネルの折りたたみに応用されるなど、工学の実用分野への応用も行われている。近年では、計算機を用いて折紙の研究を行うことも行われるようになっていく。

折紙には、伝承的に折られてきた「鶴」や「兜」のような、特定の形を表現することを目的とした作品が多いが、それ以外にも複数の紙を組み合わせる「ユニット折り」や、ねじり折りを含む形状が繰り返して現れる「平織り」と呼ばれるものなど、幾何的な形状の美しさを楽しむものがある。平織りは、一般に平坦折紙（折りたたみ後の形が平坦になる折紙）であるが、これを計算機で扱う場合には、紙を厚さゼロの平面で表現することが一般的であるから、同一平面上に複数の面が重なった状態となる。これを

正面から見た様子を正しく画面に表示するには、面の重なり順に従って下にあるものから順に上書きを行うペイントアルゴリズム（優先順位アルゴリズムとも呼ばれる）を用いたアプローチが考えられるが、この方法ではねじり折りに代表されるように、面の重なり順にループが含まれる（面の上下関係が数学术語でいう「順序集合」にならない）場合に正しい表示結果を得ることができない。通常の3D物体のレンダリングでは、重なり順がループする面を正しくレンダリングするためにzバッファを用いることが行われるが、平坦折紙では奥行き方向の値（z値）も同一であるため、この方法もうまく機能しないという問題がある。

そこで本稿では、面の重なり関係を保持するための行列と描画面の最上位に位置する面のIDを画素単位で管理するバッファを用いることで、この問題を解決する新しいアルゴリズムを提案する。これにより、面の重なり順にループを持つ折紙を折りたたんだ様子を適切に表示することが可能となる。また、面の各頂点に頂点カラーを設定して面の内部の色を線形補間することでイラスト調のレンダリングを行う方法も提案する。さらに、面の重なり関係を展開図から求める簡易的な方法も提案する。

本稿の2章でねじり折りと平織り (Origami Tessellation) について説明し、3章で関連研究を紹介する。4,5章で本提案手法について述べ、6章で結果を、7章でまとめと展望を述べる。

## 2 ねじり折りと平織り (Origami Tessellation)

ねじり折りとは、図1左に示すような展開図を、ねじるようにして折る折り方であり、折った後の面の重なり順に閉ループを持つ。このような折り方をした結果を画面に表示するのは困難であり、図1および図2の右側（折りたたみ後の形状）はIllustratorを用い、線を一本ずつ入力したものである。

折紙の一分野として、「平織り (Origami Tessellation: 折紙テッセレーション)」と呼ばれるものがある。これは、基本的には紙にねじりを加える操作を繰り返して規則的なパターンを表現する手法である。図2左に示す展開図を折りたたむ事で、右のような結果を得ることができる。この展開図には規則性があり、複数のパラメータを変更することで、容易にバリエーションに富んだ平織りを生成すること

が可能である。詳細については次章で述べる。

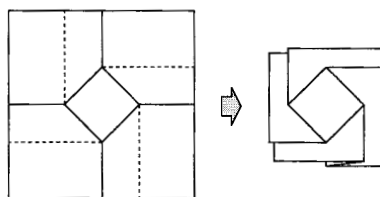


図1: ねじり折り

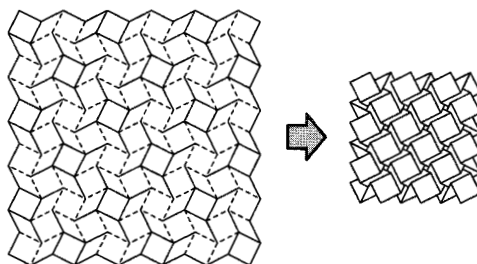


図2: 平織り

## 3 関連研究

### 3.1 折紙と計算機

折紙の研究は古くからされており、その多くは展開図に着目した幾何の分野における考察を行っている [1][2]。また、近年では折紙の設計手法についての研究が行われており、対象とする形状を折り出すための「設計」の概念が確立されつつある [3]。

計算機の普及により、折紙を計算機で扱う研究も増えている。三谷 [4] は折紙の展開図を効率的に入力するエディタの開発を行うことで、展開図データを用いた折紙の研究の効率化を目指している。また、入力された展開図から折りたたみ後の形状（面の重なり順）を推定する手法の提案も行っている [5]。しかし、ここで提案されている手法は面の重なり関係が順序集合であることを前提としており、本稿で対象とするねじり折りのような、面の重なり順に閉ループが存在する折紙に対しては有効に機能しない。

Miyazaki ら [6] は、計算機を用いて折紙をディスプレイに表示しながら対話的に操作する手法を提案した。その実装において面の重なり順を保持したデータ構造を用いているが、やはり対象としている

のは面の重なり順にループが存在しないものだけである。古田ら [7] も、計算機内で仮想的な折紙を対話的に操作するアプリケーションの開発を行っている。これはパネマスモデルを用いることで柔軟なモデル操作を実現している。しかし面の重なり関係は考慮していないため、レンダリング時に、ある面が他の面を突き抜けて表示されるなどの問題がある。

以上述べたように、計算機を用いて折紙を扱う研究が行われるようになってきているが、特に同一平面上で面の重なり順に閉ループが存在する場合には適切な表示方法が確立されていない。

### 3.2 平織りパターンの生成

平織りの展開図は、平面のタイル張りから生成される。その生成手法については文献 [8] に詳しい。図 3 はアルキメデスのタイル張り  $4^4$  に対するタイル張りから平織りの折り目パターンを作り出す工程を示したものである。折り目のパターンを生成するには図 4 に示す、プリーツ角  $\phi$  とプリーツ比  $\alpha/\gamma$  の 2 つのパラメータを用いることができる。折りたたみ後のパターンは、プリーツ角を  $-\phi$  にすることで得られる。現在は、「Tess」という名の平織りのパ

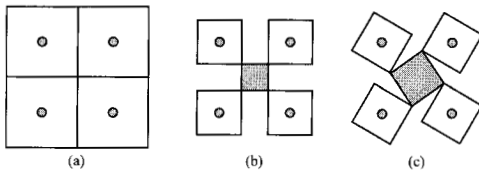


図 3: (a) タイル張り (b) タイル張りの多角形を縮小して新しい「子」多角形を作る (c) 最終のパターンを作るために多角形を回転させる [8]

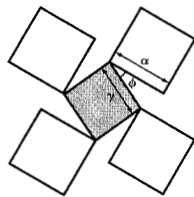


図 4: 平織りを決めるために使われるパラメータ

ターンを生成するためのプログラムが Web 上で公開されている [9]。このプログラムでは、折りたたみ後の様子を表示することが可能であるが、図 5 の

ように透過色でレンダリングしたものであり、紙が複数枚重なる場所は濃い色で表示される。この表示によって、完成形を推測することは可能であるが、実際に折った結果を視覚的に確認するには、図 2 右下のように陰面消去が行われることが望ましい。

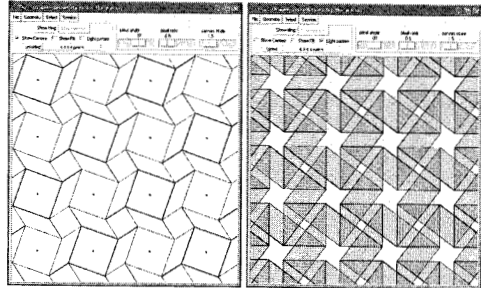


図 5: Tess の画面。展開図 (左) と折りたたんだ様子の透過色表示 (右)

## 4 面の重なり関係

### 4.1 面の重なり関係の行列表現

三谷 [5] によって、面の重なり順をスタックを用いて決定する方法が提案されているが、この手法では、面の重なり順にループがない場合 (面の重なり関係が順序集合である場合) のみを対象としている。そのため、今回の目的のために用いることはできない。

面の重なり順にループが存在しない場合は、各面に順番を定めることができるが、ループを含む場合はそのようにはできない。そこで、行列を用いて面と面の重なり関係を表現する。折紙に含まれる面の数を  $N$  としたとき、面  $F_1 \sim F_N$  の重なり関係を表す  $N \times N$  行列 (以降では簡単のために OR 行列 (Overlap Relation 行列) と呼ぶこととする) の各要素  $m_{ij}$  には、次の 4 つの状態を表す記号のうちの 1 つを設定するものとする。

- U (Upper)  $F_i$  が  $F_j$  より上にある
- L (Lower)  $F_i$  が  $F_j$  より下にある
- - (Undefined)  $F_i$  と  $F_j$  には上下関係が無い
- ? (Unknown)  $F_i$  と  $F_j$  は重なっているがどちらが上にあるか未定

たとえば図 6 のような三つ折りの場合、この OR 行列は次のようになる。図の例では、面の重なり関係は一意に決まるが、複数の可能性がある場合

には, "?"(Unknown) である要素が含まれることになる。

$$\begin{pmatrix} - & U & U \\ L & - & L \\ L & U & - \end{pmatrix} \quad (1)$$

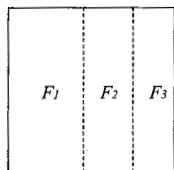


図 6: 三つ折りの展開図

## 4.2 面の重なり関係の決定

展開図から面の重なり関係を決定することは難しい問題であり, 任意の展開図から面の重なり関係を算出するアルゴリズムは知られていない. 本稿では, 面の重なり関係を求める簡易的な方法を提案する.

この方法では, 一般的な形状では完全には求まらないことがあるが, 今回対象とする平織りのような単純な構造の折紙では, ほぼ問題なく一意に値が求まることが実験から確認されている. 具体的には, 以下に述べる手順を行う.

### 折りたたみ後の面の位置を算出する

まずはじめに, 対象を平坦に折りたたまれるものとして, 展開図を折りたたんだ後に, 各面がどこに位置するかを算出する.

まず, 起点となる任意の面を選択し, それに隣接する面を再帰的に巡回する. 隣接面に移動する毎に, 2 面間を介する稜線によって (山折, 谷折の別によらずに) 面を折り返す (つまり平面上で折り線を介して反転させる) ことを行う. この反転操作は, 各面に対して高々 1 回行うものとする.

これにより, 折りたたみ後の面の位置を求めることができる. 起点となる面から奇数回の移動で到達する場合は面の向きが裏に, 偶数回の移動で到達する場合は面の向きが表になる.

### 面の対ごとに重なり関係の有無を調べる

すべての 2 面の組み合わせ ( $N(N-1)/2$  通り) について, その 2 面が折りたたみ後に重なり関係にあるか否かを調べる. これは, 単純に平面上の 2 つの多角形の重なりの有無を判定することで実現される. 面  $F_i$  と面  $F_j (i \neq j)$  が重なり関係に無い場合は, OR 行列の  $m_{ij}$  の値を "U" (Undefined) に設定し, 重なり関係にある場合は値 "?" (Unknown) に設定する.

### 隣接する面間の上下関係を決定する

折り線の山谷および面の向きの情報から, 互いに隣接する 2 面間の上下関係を決定することができる. 面  $F_i$  と面  $F_j (i \neq j)$  が辺  $e$  を介して隣接する場合, 次のように上下関係が定まる. なお,  $F_i > F_j$  という表記は  $F_i$  が  $F_j$  よりも上に位置することを意味する. この判定に基づいて得られる上限関係の情報を OR 行列に追加する.

- 面  $F_i$  が表を向き,  $e$  が山折ならば  $F_i > F_j$
- 面  $F_i$  が表を向き,  $e$  が谷折ならば  $F_i < F_j$
- 面  $F_i$  が裏を向き,  $e$  が山折ならば  $F_i < F_j$
- 面  $F_i$  が裏を向き,  $e$  が谷折ならば  $F_i > F_j$

### その他の面間の上下関係を決定する

上記で述べた方法では, 互いに隣接関係に無い面間の上下関係が定まらない. そこで, 次の定理に基づいて, 残りの面間の上下関係を決定する

定理:  $F_i < F_j$  であり,  $F_j$  の輪郭を構成する頂点  $V$  が  $F_i$  上にある場合,  $V$  を輪郭に持つすべての面は  $F_i$  よりも上に位置する.

以上の方法で, 平織りのような単純な構造に対してはほぼ一意に値の定まった OR 行列を得ることができる.

## 5 レンダリング

### 5.1 線画レンダリング

前節で述べた方法で得た, 面の重なり関係を表す OR 行列に基づいて, 折りたたみ後の形状のレンダリングを行う.

## 面 ID での塗りつぶし

基本的な概念は、zバッファ法に用いられるものであり、描画領域と同じ大きさを持つ面 ID バッファを事前に準備する。この面 ID バッファに、zバッファ法で用いられるスキャンライン法によって、各画素単位で面の ID を格納する。ここで、面の ID を上書きで格納するか否かを判定する際に、OR 行列の各要素を参照し、値が"U"(Upper) または"-"(Undefined) の場合のみ格納することとする。

## エッジ抽出

面の輪郭線を描画するために面 ID バッファに対してエッジ強調に用いられる Sobel フィルタを用いる。Sobel フィルタのマスクは次に示すとおりであり、これによって得られたエッジを画面に描画する。

$$h_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, h_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (2)$$

## 5.2 イラスト調レンダリング

平坦に折りたたまれた折紙の各面は、計算機内ではそれぞれが同じ平面上に配置され、当然のことながら法線も同じ向きであるから、大域照明を用いない通常のレンダリング方法ではすべての面が同一色となる。しかし、これらに着色を施し、図 8 右下のような結果を得られるのであれば、より形状を把握しやすくなると考えられる。本稿では経験に基づくアルゴリズムで妥当な結果を得ることを目指した。

一般的な経験則として、谷折りになる箇所は光が当たらないので暗く、山折りになる箇所は明るい。そこで、面の輪郭を構成する各頂点に着目し、その頂点を構成する 2 つの辺の山谷の別を見て、頂点の明るさ  $C_b$  を次の式によって決定する。

$$C_b = (\text{山折りの数} - \text{谷折りの数} + 2) / 4 \quad (3)$$

山折りの数、谷折りの数ともに取りうる値は 0 から 2 の間なので、 $0 \leq C_b \leq 1$  である。

この  $C_b$  だけで頂点の色を決定すると、面が単色で塗りつぶされる場合が多く、不自然な印象を受けるため、頂点の位置 (座標値) に応じて色に変化をもたせることを行う。これには物理的な根拠はなく、グラデーションがあると綺麗に見える、という経験則にのみ則ったものである。今回は、次のようにし

て頂点カラー  $V_{color}$  を決定した。

$$t = \min(1, 0.9(0.75 + 0.25C_b) + 0.15d) \quad (4)$$

$$V_{color} = RGB(t, t, 0.97t) \quad (5)$$

式 (4) の  $d$  の値は、対象とする面の外接四角形の左上隅からの距離を外接四角形の対角線の長さで割った値である。式中の定数は重み付けに用いている値であるため、レンダリング結果が望ましくなるように変更可能である。上記のように頂点の色を設定し、面の内部の画素については、頂点カラーを線形補間することで色を算出する。

## 6 結果

本稿で提案する手法を Java 言語を用いて PC 上にて実装したレンダリング結果を図 7, 8, 9 に示す。入力には ORIPA 形式 [10] の展開図データを用い、提案手法で面の折り重なり関係を推定し、512x512 ピクセルの領域にレンダリングを行った。図 7, 8 は、それぞれ「ねじり折り」と「平織り」を折りたたんだ後の形状であり、左上が Java2D を用いて透過色で表示したもの、右上が提案手法でエッジのみ表示したもの、左下が提案手法で頂点カラーを設定し、面の塗りつぶしを行ったもの、そして右下がエッジの表示と面の塗りつぶしを行ったものである。図 9 は参考までに、折鶴を本提案手法でレンダリングした結果である。

## 7 まとめと展望

面の重なり関係を表す行列と面 ID バッファを用い、エッジ抽出を用いることで、その結果を適切に表示できることを示した。さらに、面の重なり関係を表す行列の簡易的な構築方法を示した。また、頂点カラーを設定する新しいアルゴリズムを提案し、これによりイラスト調の見た目のよい結果を得ることができた。今後の展望として、頂点カラーの設定方法に物理的な根拠を示すことが望まれる。得られる結果の画像はラスタ画像であるが、その後の編集などのためには、ベクタ形式でのエッジの出力ができることが望ましい。また、面の重なり関係の推定に今回用いた手法は簡易的なものであり、あらゆるケースに対応できるものではないので今後のさらなる改善が必要である。

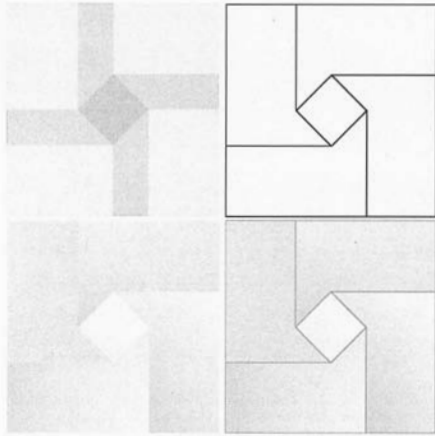


図 7: ねじり折りの例

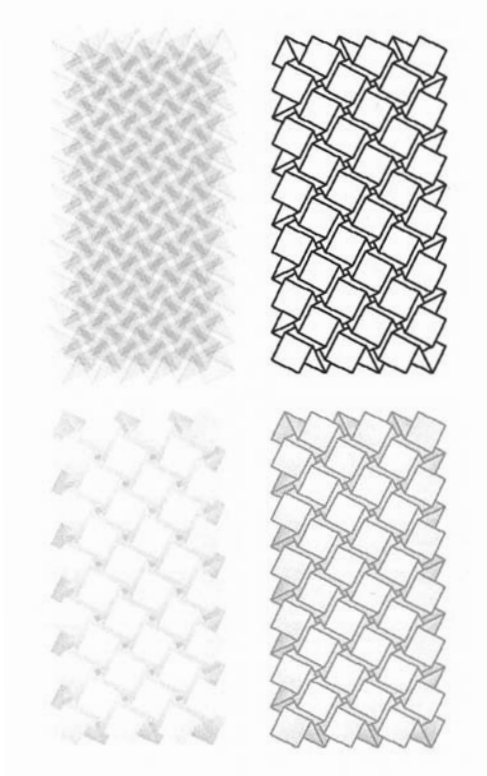


図 8: 平織りの例

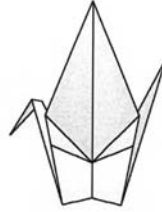


図 9: 折鶴の例

## 参考文献

- [1] 川崎敏和: パラと折り紙と数学と, 森北出版株式会社 (1998).
- [2] 深川英俊: 折紙の数学, 森北出版株式会社 (2002).
- [3] Lang, R. J.: *Origami Design Secrets: Mathematical Methods for an Ancient Art*, AK Peters, Ltd. (2003).
- [4] 三谷純: 折紙の展開図専用エディタ (ORIPA) の開発および展開図からの折りたたみ形状の推定, 情報処理学会論文誌, Vol. 48, No. 9 (2007).
- [5] 三谷純: 平坦折り折紙から再現される形態数の数え上げ手法, 日本図学会, Vol. 41, No. 1, pp. 27-33 (2007).
- [6] Miyazaki, S., Yasuda, T., Yokoi, S. and Toriwaki, J.: An Origami Playing Simulator in the Virtual Space, *The Journal of Visualization and Computer Animation*.
- [7] 古田陽介, 三谷純, 福井幸男: マウスによる仮想折り紙の対話的操作のための計算モデルとインタフェース, *インタラクション*, Vol. 2007, No. 4, pp. 137-144 (2007).
- [8] Hull, T., 川崎敏和: 折り紙の数理と科学, 森北出版株式会社 (2005).
- [9] Bateman, A.: Paper Mosaics. <http://www.papermosaics.co.uk/>.
- [10] 三谷純: 折紙展開図エディタ ORIPA. <http://mitani.cs.tsukuba.ac.jp/pukiwiki-origami/>.