

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

古田 陽介[†] 三 谷 純^{†,††} 福 井 幸 男[†]

本稿ではマウスのみでも直感的、対話的に複雑な折り紙形状をモデリングできる計算モデルとインタフェースを提案する。近年の計算機の高性能化に伴って一般家庭の PC でも 3DCG アニメーションが扱いやすくなったため、これを用いたより分かりやすい折り手順の伝達方法が研究されている。しかし、既存のモデリングソフトで折り手順を示すための 3DCG アニメーションを作ることは、非常に手間のかかる作業である。紙の折り方は複数の種類に分類できるが、これらはある面の回転とそれに付随する隣接面の動きで表現できる場合が多い。そこで本稿で提案する手法では、折り紙の面の形状と動きを計算機で表現するためにバネモデルを用い、さらにマウスで直感的に面の分割や回転操作を行うために、2D ドローツールを参考にした折れ線生成インタフェースや、面を折り曲げる際の回転軸、回転角を決定するための手法を提案する。実際に本手法を計算機上に実装し、これまでモデリングすることが難しかった複雑な形状、折り操作を計算機上で再現できることを確認した。

Modeling and mouse interface for interactive virtual Origami operation.

YOSUKE FURUTA[†], JUN MITANI^{†,††} and YUKIO FUKUI[†]

We propose a new method of modeling and interface for interactive virtual Origami operation that enables a user to handle a complex Origami model intuitively using a common mouse. Although 3DCG animation has become popular in recent years with the progress of performance of common PCs, designing 3DCG animation of folding process of Origami is still difficult and time consuming task. Though there are many ways of folding operation, most of them can be expressed as a set of the rotation of certain face and the movement of the neighboring faces. Based on this thought, the modeling method we propose uses the spring-mass model to express shapes and movements of an Origami model in a computer. The interface we propose uses simple mouse operations. The main operation that generates folding lines works similarly with 2D draw tools. Additionally, we propose new interfaces for specifying the axis and the angle for rotation. We implemented the method on a PC and we found that we can reproduce complex Origami shapes intuitively using a simple mouse.

1. はじめに

日本では多くの場合、折り紙は単なる遊戯と認識されている。しかし折り紙は古くから幾何学との関連性が知られており、学際領域では数学の世界を中心として以前から研究が進められてきた。近年では工学の分野への応用に関する研究も盛んに行われるようになり、自動車のエアバッグの折りたたみ¹⁾やドーム建築物の屋根の収納法²⁾などにその研究成果が応用されている。

また、近年では任意の形状を展開図化するための「設計」と呼ばれる手法が発明され、これを用いて多くの新しい折り紙作品が創作されるようになった。し

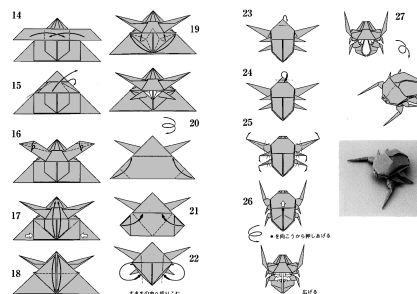


図 1 テナゴコガネの折り図³⁾

かしながらその折り方を他者に伝えるための手段は、旧来からの「折り図 (図 1)」に頼っているのが現状である。折り図で複雑な手順を表現するためには作図に多大な労力が必要となるだけでなく、各イラストの時間的つながりを表現することができないという根本的な問題点がある。現在折り紙が広く普及しているの

[†] 筑波大学 システム情報工学研究科

University of Tsukuba

^{††} 科学技術振興機構 さきがけ

PRESTO, Japan Science and Technology Agency

は、誰でも身近な素材で実際に作ることができるからだといえる。しかしそれをどうやって作るかを他人に伝える点においては、まだ多くの改善の余地が残されている。

そこで近年の計算機の性能向上とともに注目されているのが、3DCG アニメーションを用いた折り手順提示手法である。そのために必要となる折り紙形状のモデリングについても、近年いくつかの手法が発表されてきたが、多種多様な折り紙の形状や紙の対する操作を完全に再現するまでには至っていない。そこで、本研究では折り紙の形状をバネマスモデルで表現することを提案する。これまでの多くの研究では、折り紙の持つ幾何学的性質をもとに形状を計算していたが、この手法では折り操作による複数面の三次元的で連鎖的な移動を計算することが難しく、できたとしても非常に時間がかかることが予想される。しかし、バネモデルを用いることでこのような連鎖的な動きを実時間で計算することができるようになる。

また折り紙を折ることを計算機上でシミュレートする上でもう一つ問題となるのが、入力装置の問題である。一般に折り紙を折る際には両手を使用するが、10本の指の動きを取り込める装置はあまり一般的ではなく、多くの場合はマウスやキーボードによる限られた情報しか得ることができない。実際に紙を折る時は、折り目を付ける動作と面を折る動作を同時に行うことが一般的である。しかし本研究では折り操作をマウスなどでシミュレートするために、その二つの動作を別々に行うこととし、さらにそれらを直感的におこなうためのインタフェースを提案する。ある面上の折り目やそれを軸とした面の回転は二次元で表現することができるが、それらを自然に統合することで、ユーザはマウスなどでも直感的で自由な折り操作を行えるようにした。

本研究は折り紙の形状を計算するためのモデルと、折り操作を計算機上の仮想空間内で行うためのインタフェースを提案するものである。本稿の第2章では関連する研究を紹介し、第3章では提案手法の詳細について述べ、第4章では結果、第5章では考察と今後の展望を述べる。

2. 関連研究

折り紙についての研究として、複雑な形状を折り出すための設計手法に関するものが近年活発に発表されている。代表的なものとしては Lang による研究⁴⁾があげられる(図2)。

計算機上で折り紙を扱う研究としては、内田らの展

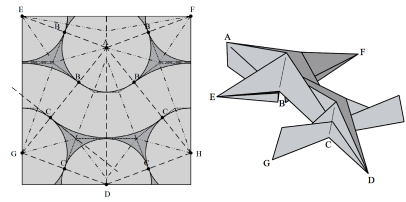


図2 折り紙設計の例⁴⁾

円のパッキング問題を解くことで正方形の紙から任意の長さの角を折り出すことができる

開図から折り上がる形状を推定するアルゴリズムに関するものがある⁵⁾。設計技術の進歩によって複雑な形状が折り上がるようになったが、それに伴い展開図も複雑化している。そのような展開図に対しても適切な折り上がり形状を短時間に予測できるような手法について現在も研究が行われている。

Miyazaki らは計算機を用いて折り紙を対話的に操作する手法と、それ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⁶⁾。この研究で目指しているものは本研究と非常に近いが、この論文で用いられている幾何学的制約に基づく形状計算手法では同一平面上にない複数の面の動きを計算できず、面が相互に作用しながら三次元的に形状が変化する場合には適応できない。また、この研究で得られたプログラムは折り手順を記録し任意に再生することが可能であるが、先の制約によって手順そのものが実際のものとは異なっている場合が多い。本研究では三次元的な形状変化にも対応し、そのため実際の折り手順に近い状態をシミュレーションすることが可能である。詳細は第3章2節で述べる。

三谷らは、折り紙の構造をわかりやすく表示するための計算機を用いたデフォルメ手法を提案している⁷⁾。この手法を適用するためにはあらかじめ面の重なり順を求めておく必要があるため、本研究に直接適用することはできないが、折り紙の形状を分かりやすく伝えるという目的のためには非常に有効であるといえる。

計算機上に折り紙形状を取り込む研究としては、Kato らによる「折り図」の画像をもとにした手法が挙げられる⁸⁾。Kato らの手法では、折り図画像に含まれる矢印や折れ線の情報を計算機で解析することによって折り操作を推定し、それをもとに計算機内の折り紙モデルを更新している。また、三谷は二次元バーコードを印刷した紙を折ったものをカメラで撮影し、その写真をもとに折り畳み構造を推定しモデルを構築する手法を提案している⁹⁾。しかしこれらは他の媒体を計算機で解析するという手順を踏まなければならないため、直感的かつインタラクティブであるとはい

がたい。本論文で提案する手法では、ユーザは計算機上のモデルを直接操作することができるため、より直感的かつインタラクティブに操作することが可能である。詳細は第3章3節で述べる。

バネモデルを用いた形状計算手法の中には商用3Dモデリングソフトの布シミュレータに用いられているものがある¹⁰⁾。しかしこれは対象が布であるため面の平面性をほとんど考慮していない点と、ユーザが対話的に折り筋を入力したり面を動かしたりするインタフェースがない点などにより、折り紙をモデリングする用途に適しているとはいえない。

3. 提案手法

3.1 概要とデータ構造

本プログラムを実行すると、画面内の仮想空間内に紙を模した正方形のポリゴンが表示される。これに対してマウスで折り目の生成や面の回転などの操作を繰り返し加えることで、紙の折り操作をシミュレートする。折り紙モデルの各面は常に同じ形を維持するよう対角線状にバネが張られており、面を動かした際に隣接する面が歪むことを防いでいる(図3、図4)。なお、バネの配線はプログラムを初期化した際と、ユーザによって折り目が生成され、折り紙モデルを構成する面の個数や頂点数が変化した際に実行される。ユーザはマウス操作で面を動かすことができ、その操作と同時にバネモデルによる形状計算を行い画面をリアルタイムに更新するため、ユーザはあたかも計算機上でリアルタイムに折り操作を行っているかのような印象を得ることができる。

プログラムは内部に複数の頂点クラスと辺クラス、面クラスを持っており、頂点クラスは三次元座標を、辺クラスは始点、終点をあらわす二つの頂点クラスへの参照を、面クラスは辺で囲まれた閉領域を表し辺への参照を三個以上保持している。辺クラスは二つの面から参照されているかどうかで分類することができ、二つの面から参照されているものは折れ線クラス、それ以外は境界クラスとする。計算機内の折り紙モデルは複数の面クラスで構成されており、それらをもとに仮想空間内に3Dポリゴンモデルとして表現される。なお、本論文では穴の開いた面、凸型ではない面の生成については対象としない。

3.2 立体的な折り紙の形状維持

折り紙の面は辺による閉領域で定義され、折り操作は面の移動と回転で表現することが可能である。しかし場合によっては一度の折り操作で複数の面が連鎖的に動くことがあり、隣接する面も適切に移動させな

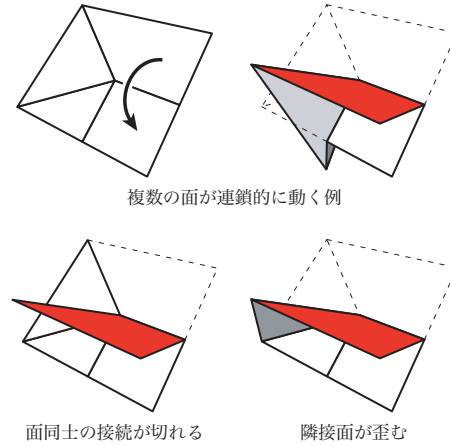


図3 面の千切れ・歪み

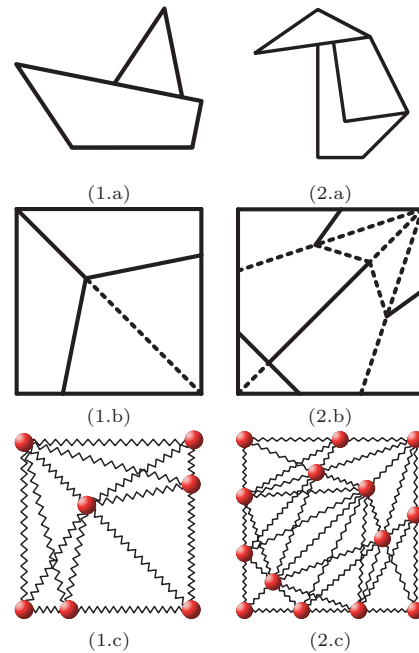


図4 船(1.a)とペンギン(2.a)の展開図(b)とバネモデルによる形状表現(c)

ければ、面が千切れたり歪んだりしてしまうことがある(図3)。折り紙の面を剛体であると捉え、幾何学的な拘束条件から面の運動を求めることも可能であるが¹¹⁾、実時間内で計算結果を得ることは一般的に困難である。

そこで本研究では、面を構成するすべての頂点間に網羅的にバネが張られているというモデルを採用し(図4)、面が歪んでしまった場合にはバネが釣り合おうとする力を繰り返し計算することで頂点を適切な位

置へと移動させることとした。具体的なバネ計算のアルゴリズムは以下の通りである。

一般的なバネモデルでは、ある頂点 i にかかる力 F_i を (1) 式で表すことができる¹²⁾。

$$F_i = \sum_j \left\{ k \left(1 - \frac{L_{ij}}{|r_{ij}|} \right) r_{ij} + Dv_{ij} \right\} - mg \quad (1)$$

ここで k 、 L_{ij} 、 r_{ij} 、 D 、 v_{ij} 、 m 、 g は、それぞれバネ定数、バネの自然長、頂点 j に対する頂点 i の相対位置、ダンパー定数、頂点 j に対する頂点 i の相対速度、頂点の質量、重力加速度である。なお、本プログラムでは上式を簡略化し $k = 1$ 、 $D = 0$ 、 $m = 1$ 、 $g = 0$ とした (2) 式を使用している。

$$F_i = \sum_j \left(1 - \frac{L_{ij}}{|r_{ij}|} \right) r_{ij} \quad (2)$$

(2) の運動方程式を数値的に解くことで頂点の位置の時間変化を計算でき、面の位置と形状が定まる。運動方程式の数値計算はオイラー法により行う。具体的には、(3)、(4) 式により Δt 秒後の頂点 i の位置 r_i と速度 v_i を計算できる。

$$v_i(t + \Delta t) = v_i(t) + d \frac{F_i(t)}{m} \Delta t \quad (3)$$

$$r_i(t + \Delta t) = r_i(t) + v_i(t) \Delta t \quad (4)$$

ここで d は減衰定数であり、0 以上 1 以下の値を指定する。本プログラムでは $d = 0.2$ とし、また $\Delta t = 1$ とした。

上記のアルゴリズムによって得られる結果はある瞬間における形状であり、その変化を実時間アニメーションとして提示するためには画面書き換え時間内に計算を終了させなければならない。その点、このアルゴリズムによる形状の変化は連続的であるため、値の収束を待たずに途中で計算を打ち切っても、近似値をユーザに提示することが可能である。さらに上記のアルゴリズムは条件分岐がほとんどない非常に単純なものであるため、計算機を用いた高速化は比較的容易である。そのため本研究のようなシミュレーションには非常に適している。

ちなみに、バネが張られているのは同一の面内にある頂点同士のみであり、違う面に存在している頂点の間にはバネを張る必要はない (図 4 の c)。なぜなら本手法では歪んだ面を含む個々の面の形状のみに注目し、これらを歪のない形状へと連続的に変化させていくことで全体の形状を得ているためであり、その過程において他の面との制約を考慮する必要はないからである。

3.3 マウスオペレーション

現実の折り紙において紙を折る操作というのは、「角を合わせる」動作と「圧力をかけて折り目をつける」動作の組み合わせであることが多い。しかしこれらの動作は両手を使えることが暗黙の前提となっており、片手だけでこの動作を行うことは難しい。10 本の指の動きを計算機に取り込むための装置はまだあまり普及しておらず、ほとんどの環境ではマウスによる二次元相対座標のみという、3D モデリングにおいては十分とはいえない入力値しか得ることができない。

紙の折り方には多くの種類が存在しているが、それらはある面の回転とそれに付随する隣接面の動きで表現できる場合が多い。また、あらかじめ空間上に軸を設定しておけば、回転は二次元変換で表現することが可能である。そこで本プログラムでは紙を折る操作を「紙に折り目をつける」操作と「折り目に沿って面を回転させる」操作に分割し、それらを組み合わせることで現実の折り操作をシミュレートすることとした。あらかじめ必要な折れ線をいくつか生成した後、そのうちのひとつを実際に折る軸として、明示的に回転させると、残りの面もバネモデルに従って移動するようになっている。また折れ線の生成や軸となる折れ線の選択と回転角の決定は一連のマウス操作に対応した形で自動的に行われるため、ユーザはそれらを意識することなく直感的に面を操作することが可能である。以下でこれらの操作について詳細を述べる。

3.3.1 折れ線の入力

折り紙を折ることを対話的に行うためには、ユーザが自由に、かつ少ない手順で折り目を付けられることが望ましい。そこで本プログラムでは紙に折り目をつける操作 (平面に折り筋を入力する操作) に 2D のドローツールを参考としたインタフェースを採用した。一般的なドローツールではキャンバス上をマウスでド

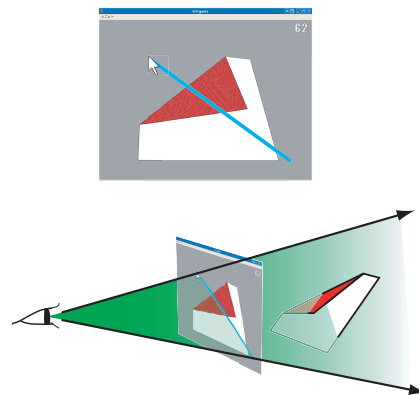


図 5 折れ線の入力

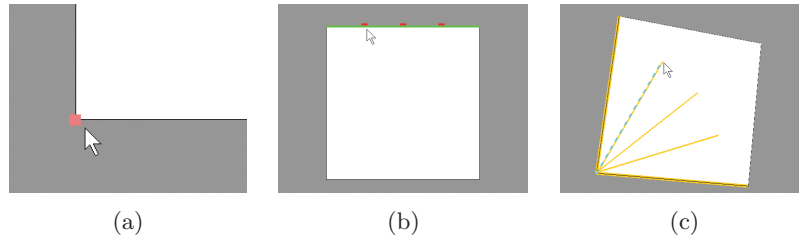


図 6 カーソル座標の近似

- カーソル座標と頂点座標の距離が閾値以下の場合はカーソル座標を頂点座標へ一致させる
- カーソル座標と辺を n 分割する点との距離が閾値以下の場合はガイド（辺上の点）を表示し、カーソル座標をその位置へ一致させる
- 始点が頂点でありかつ現在のマウス座標が始点の角を n 等分する点に近い場合はガイド（ n 本の点線）を表示し、カーソル座標をその位置へ一致させる

ラッグすると軌跡が線となって描かれるが、本プログラムでも画面上の任意の場所をドラッグして線を引くことによって、ユーザは面のどの場所にも自由にかつ対話的に折り目を「描く」ことが可能となっている（図 5）。その手順は次のとおりである。

- マウスでスクリーン上をドラッグし、ドラッグの始点と現在のマウス座標を結ぶ直線をスクリーン上に描画する。
- 始点、終点に対応する仮想空間上の三次元座標を求め、視点から両点を通り奥へ無限に伸びる二本の半直線を仮想空間上に生成する。
- 二本の半直線の間に挟まれた三角状の領域を定義し、その領域とすべての辺との交差判定を行う。
- 交点に新たに頂点を配置し、その頂点を結ぶように面分割を行う。
- 分割後の面に対してバネを張りなおす。

重なり合った面に対して折り目を描いた場合は、それらすべての面に新たに折れ線が追加される。また頂点付近や辺、角を等分する場所の付近では自動的にスナップするため（図 6）、実際に紙を折るよりも正確で、容易に計算機内の紙に折り目をつけることができる。なお、本論文では面を完全に分割しない折れ線（図 7）の生成については対象外とした。これにより、生成される面はすべて凸多角形となる。

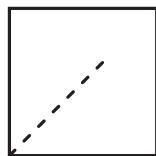


図 7 完全に面を分割しない折れ線の例

3.3.2 回転面と軸の選択

経験的に、人はドアを開くときにテコの原理による

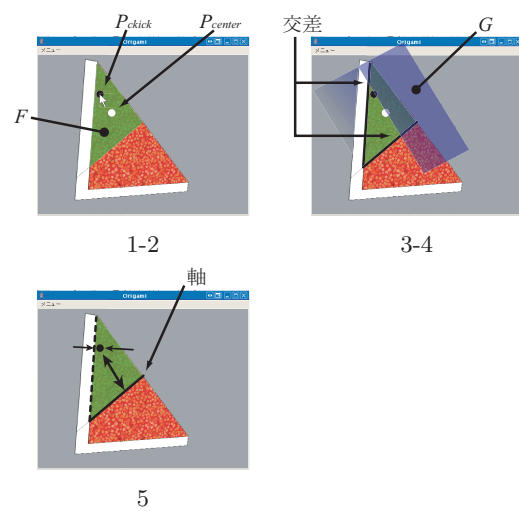


図 8 回転面と軸の選択

労力軽減を期待して蝶番からできるだけ遠い場所に力点を置くことが多い。本プログラムではその動作を参考として、面を選択すると同時にポインタ位置から人間のイメージする軸の位置を予測し、自動的に選択することとした（図 8）。しかし、現実の動作をそのまま逆転させ、ポインタから一番遠いところにある折れ線を軸とするのは適当ではない場合がある（図 9.b）。そこで本プログラムでは以下のような手順で「面の中心から見て、クリックした位置の反対側にある折れ線」を求め、それを軸とすることで人間のイメージする軸を効果的に予測する。

- 仮想空間上の紙をクリックし、ポインタの位置に対応する面のうち一番視点に近い面 F を取得する
- クリックした位置の三次元座標 P_{click} と面 F の中心（本プログラムでは頂点の平均）座標 P_{center} を求める
- P_{click} と P_{center} を通り、 F に垂直な面 G を

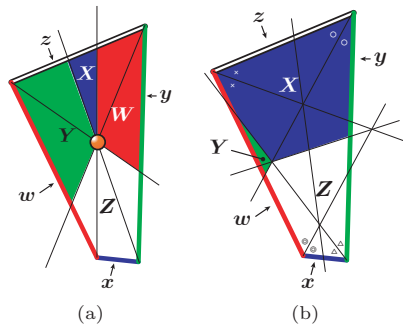


図 9 クリックした領域と選択される軸の関係。この形状の場合、マウスポインタから一番遠いところにある折れ線を軸とするアルゴリズム (b) では辺 w を選択可能な領域が存在しない。

定義

4. F を構成するすべての折れ線について、 G と交差判定を行う
5. 交差していた折れ線のうち、 P_{click} から最も遠いものを軸とする

本アルゴリズムにおいて、ある折れ線と、その折れ線が軸として選択されるクリック位置との関係を図示すると図 9.a のようになり、例えば P_{click} が領域 X 内であれば、軸として選択されるのは折れ線 x となる。ちなみにマウスポインタから一番遠いところにある折れ線を軸とするアルゴリズムでも同様に関係を図示すると図 9.b のようになり、人間の意思に関わらず軸として選択できない折れ線 (図中では w) が存在してしまうため、このアルゴリズムは適当ではないといえる。

なお、本アルゴリズムでは選択面の中に軸となる折れ線が含まれていることを想定しているが、面を構成している辺以外を軸としたい場合もあると考え、本プログラムでは手動による軸の指定もおこなえるようになっている。

3.3.3 回転角の決定

面の回転は、仮想空間上の面をマウスで選択した後ドラッグするというインタフェースで実行される。面を選択し軸を決定した後、軸に垂直で上記の P_{click} を通るような平面 G を定義し、そのままマウスをドラッグさせると、 G 上でマウスに対応する三次元座標を通るように回転角を決定し、面を回転させる (図 10)。この処理によって仮想空間内にある操作対象の面と画面上のポインタの位置が感覚的に一致するようになるため、ユーザは直感的に回転操作を行えるようになっている。

なお軸と視線が垂直に近い角度で交わっている場合、カーソルの微小な移動が回転角に大きな影響を与えて

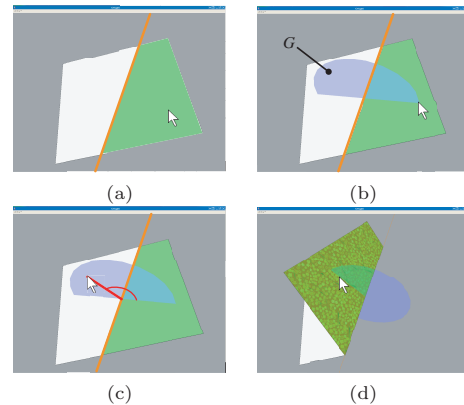


図 10 面回転

- a. 面をクリックして選択し、軸を決定する。
- b. クリックした点を通り軸に垂直な平面 G を定義して、 G 上に円盤を表示する。
- c. マウスをドラッグすると、カーソルの位置に応じて回転角を決定する。
- d. 軸を中心として、面を回転させる。

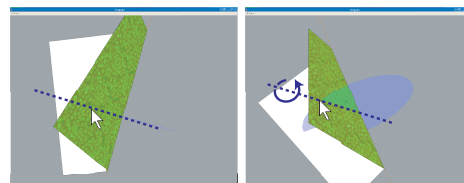


図 11 視点移動

しまい、回転の制御が困難になる。そのような場合は一時的に視点を移動させ軸と視線が垂直にならないようにする。このときの視点の移動は上記の P_{center} 、 P_{start} を通る直線を軸とする回転移動とすることで、マウスの位置と仮想空間上の紙の位置がずれないようにになっている (図 11)。

また、マウスボタンを押した際には軸に垂直でマウスボタンを押した位置を通る円を表示させ、面がその軸を中心として回転することを直感的に意識しやすいようにしている。

3.4 押しつぶし

本プログラムでは回転操作を主体としてモデリングを行うが、 180° ではない中途半端な回転を繰り返すことによって形状が意図せず立体的になってしまうケースが発生する。現実の折り紙でも紙の塑性のよって同様に形状が立体的になってしまう場合があり、そのような場合には圧力を加えて形状を平面に戻すことがある。

そこで本プログラムでも仮想空間上の紙に対して任意のタイミングで仮想的な「押しつぶし」操作を行える

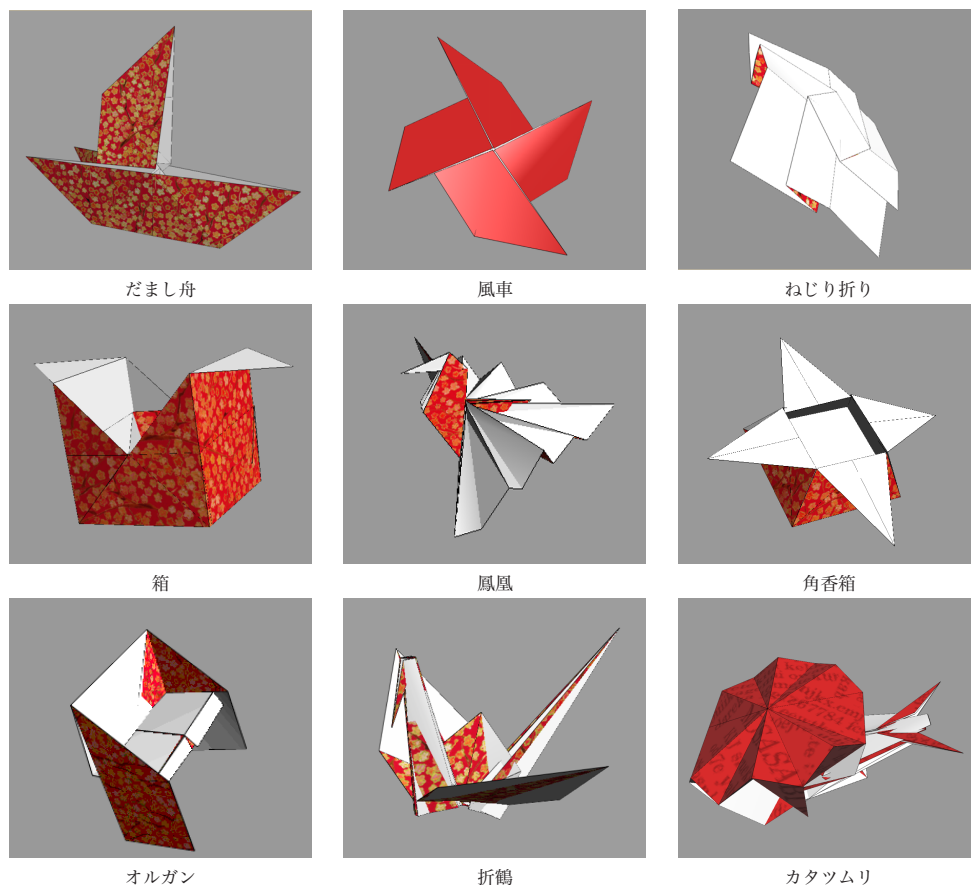


図 13 作例

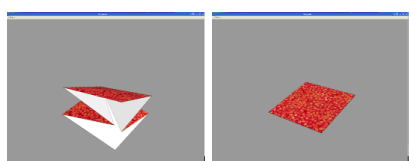


図 12 押しつぶしの実行結果

ようにし、紙を平面に保つことを可能とした (図 12)。

実際のプログラム中では各頂点の z 座標を制約に関係なく単純に減少させているだけなので、場合によっては個々の面が変形してしまうことがある。そのためこの処理によって平坦に変形された形状に対しては再度バネ計算を繰り返し行うことによって正しい形状に復元することとしている。

4. 結 果

本プログラムを用いることによって、図 13 に示すような形状を折ることができた。このとき使用した環境は以下の通りである。

CPU : AMD AthlonMP 2800+ (2133MHz) ×2
RAM : PC-2100 DDR-SDRAM (3.5GB)
GPU : 3Dlabs WildcatIII 6210 (VRAM: 416MB)

モデリングに要した時間は折鶴の場合は約 7 分ほどであり、他の形状もおよそそれ以下の時間で製作できた。また、現実の紙を折る過程において複雑な動きを見せるケースでも本プログラムでは多くの場合でそれを再現することができた (図 14)。その動きを適切な方法で保存することによって今ある折り図の代わりにできると考えられる。

5. 考察と今後の展望

本論文による提案手法を用いることでこれまで計算機上で表現することが難しかった複雑な形状、動きを表現することができるようになったが、個々の面同士の間合いや衝突を考慮していないため意図しない結果になることがある。たとえば図 13 のオルガン、折鶴、カタツムリのようにごく近い場所で面が多数重

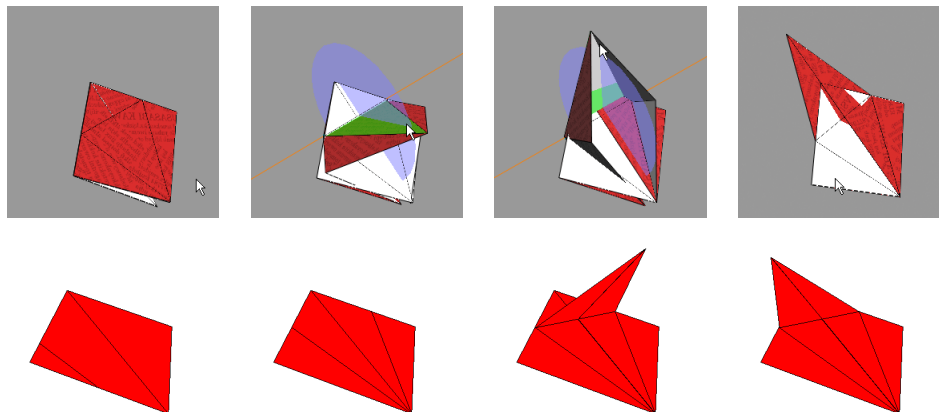


図 14 花卉折りを本プログラム（上段）と既存のプログラム⁶⁾（下段）で行った場合の比較

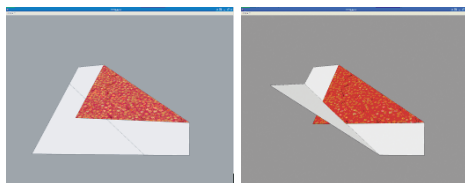


図 15 重なりを考慮して同時に動かすことができない

なり合っている形状の場合、面が交差してしまいレンダリング結果が不正になっている。また図 15 のように重なり合った面を同時に動かした場合、マウスで直接操作している面以外をその面に追従させて動かすことができない。それらを解決するためには、面同士の位置関係の変化を調べ、衝突していたら適切な位置へと移動させる処理を加える必要があるだろう。何重にも重なった面を同時に動かす場合においても高速に処理を行えるような手法を模索中である。

本論文で提案する手法を用いて折り操作を効果的に分割することで、通常両手が必要となる折り紙の操作をマウスのみでも折ることができた。しかし紙を引っ張るような本質的に両手が必須の操作はそのようなアプローチでは解決できないため、仮想空間上に紙をピン止めする機能の追加など更なる改良が必要であるといえる。

参 考 文 献

- 1) Cristoffer Cromvik: Airbag Folding based on Origami Mathematics; The Fourth International Conference on Origami in Science, Mathematics, and Education (4OSME), (2006)
- 2) K.Goppert: Adaptive Tragwerke-Wandelbare Dachkonstruktionen für Sportbauten.; Bautechnik, Vol.82, No.3, pp.157-161, (2005)

- 3) 桃谷好英: おりがみ虫の世界; 誠文堂新光社, (2002)
- 4) Robert J. Lang: Origami Design Secrets: Mathematical Methods for an Ancient Art; AK Peters, Ltd., (2003)
- 5) 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)
- 6) Shin-ya MIYAZAKI, Takami YASUDA and Shigeki YOKOI, Jun-ichiro TORIWAKI: An ORIGAMI Playing Simulator in the Virtual Space; The Journal of Visualization and Computer Animation, Vol.7, No.1, pp.25-42, (1996)
- 7) 三谷純, 鈴木宏正: 折り紙の構造把握のための形状構築と CG 表示; 情報処理学会論文誌, Vol.46, No.1, pp.247-254, (2005)
- 8) Jien KATO, Toyohide WATANABE, Hiroyuki HASE, Takeshi NAKAYAMA: Understanding Illustrations of Origami Drill Book; 情報処理学会論文誌, Vol.41, No.6, pp.1857-1873, (2000).
- 9) 三谷 純: 二次元バーコードを用いた紙の折りたたみ構造の認識とモデル化; 情報処理学会研究報告, 2005-CVIM-150, pp. 115-122, (2005)
- 10) Autodesk: Maya 8 Unlimited, (2006)
- 11) Tomohiro TACHI: Smooth Origami Animation by Crease Line Adjustment; SIGGRAPH 2006 Posters, (2006)
- 12) 幸島 明男: Web 上で操作可能な仮想柔物体の変形操作システムの試作; <http://www.consorts.org/sashima/vr.pdf>, (2004)