

マウス入力による対話的操作を組み込んだ仮想折り紙システム

古田 陽介、三谷 純、福井 幸男

筑波大学 システム情報工学研究科

概要

折り紙は日本の伝統芸能であるが、折り紙の形状を対話的に生成するための研究はまだあまり行われておらず、発展の余地が残された分野である。そこで、折り紙を計算機上で折るためのデータ構造、アルゴリズム、およびインターフェース等を考案し、実装を行った。これにより、今まで計算機上で扱うことが難しかった立体的な折り紙の形状を容易に扱えるようになった。また、本システムでは特別な三次元入力装置を用いずともマウスのみの操作で直感的に形状の生成を行うことができる。

Virtual ORIGAMI System with Interactive Mouse Manipulation

FURUTA Yohsuke, MITANI Jun, FUKUI Yukio

Graduate School of Systems and Information Engineering, University of Tsukuba

Abstract

Although Origami is one of Japanese traditional cultures, studies of interactive shape design are not common, and studying Origami field has room for further development. In this paper, we propose data structures, algorithms and interfaces for folding Origami models on a computer interactively. We implemented a system for evaluations. With our system, users could easily handle a three-dimensional Origami model that was difficult until now. As one of features, users edit shapes of Origami intuitively only with a usual mouse without any special three-dimensional input devices.

1. はじめに

紙に対して単純な折り曲げ操作を繰り返し加えるだけで無限の形状を生み出すことのできる「折り紙」は日本の伝統文化であり、今では「ORIGAMI」として海外にも広く知られている。また、折り紙は幾何学とも深くかかわりがあることが知られ数学の世界を中心として学術的な研究も多くなされている。さらに、近年では作品の創作に「設計」の概念が持ち込まれ、それによってさまざまな新しい形状が生み出されるなど、芸術的な面で大きな発展があった。これらの作品の折り方を他者に伝える方法としては「折り図(図 1)」を用いることが一般的であるが、この折り

図を描くためには多大な労力を必要とするため、このことが他者への作品に関する情報の伝達を困難にしている。

そこで、この問題を解決するために計算機を利用し、3DCG アニメーションで折り手順を提示することに関する研究が以前より行われている。最終的には計算機上で現実世界と同じように紙を折ることができ、それを保存して 3DCG アニメーションとして再生できることが理想である。しかし現状では多種多様な折り紙の形状を完全に再現するまでにはいたっていないことと、モデリングに大変な手間がかかり図を描いた方が早いなど、問題が多数残されている。

折り紙を折るという行為は、面に折り目をつける、複数の平面を同時に動かすという動作の集合に一般化できる。既存の折り紙シミュレータでは複数の面を動かす際に幾何学的な変換手法を用いているが[2]、この手法では同一平面にない複数の面が同時に動く場合のシミュレートを実タイムに行うことができない。そこで本プログラムでは幾何学的変換ではなくバネモデルを応用した物理シミュレーション的手法を用いることにより、これまで難しかった立体的な形状の操作を実タイムに行うことを可能とした。

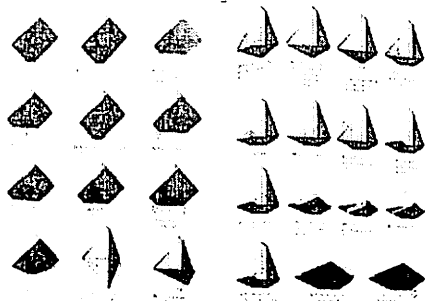


図 1: 折り図の例[1]

また、本プログラムではマウスで折り紙の折り目を任意に生成することが可能である。画面上の任意の場所をドラッグして線を引くとその線に沿って面に折り目をつけることができ、3DCG のモデリングのような複雑な操作を行わなくとも 2D のドローソフトで絵を描くような操作で対話的に形状に手を加えていくことが可能である。さらに、頂点付近や辺、角を等分する場所の付近では微妙なずれを自動的に補正するため、実際に紙を折るよりも正確かつ容易に計算機内の紙を折ることができる。

本プログラムでは多くの人々に使ってもらえるように入力装置にマウス(一部キーボード)を用いることを前提として設計した。マウスでは三次元的な奥行きを表現できないため、一般的な 3D モデリングソフトでは異なる視点からの正投影図を複数表示している。しかし折り紙を折ることについていえば、その行為は「紙を折る」という単純な動作の繰り返しであるためモデリングの際にはその「紙を折る」という操作が行えればよいと考えることができる。そこで本プログラムでは画面上に軸を設けその軸上で面を動かすというインターフェースを採用し、その結果複数の正投影図を画面に表示しなくても、マウスで立体的な操作が可能となった。なお、入力デバイスに関するプログラムはシミュレーション部とは独立しており、それを拡張することによって将来他の入力装置から本プログラムを操作することも可能である。

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行われている。代表的なものとしては Robert による研究[1]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[3]などの展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。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。

Kato らは「折り図」の画像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する手法を提案した[4]。この手法は図中に含まれる矢印や折れ線の情報を活用している。三谷ら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た[5]。

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[2]。

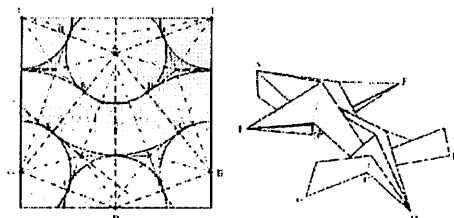


図 2: 折り紙設計の例[1]

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変換手法では操作できる面は同一平面上にあるものに限られ、それに該当しない面を含めて大域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。本研究では大域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。

3. 折り紙シミュレータ

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。

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と二頂点間の距離を、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。仮想空間内の 3D ポリゴンで構成された折り紙モデルは複数の面クラスを元に構成されている。また、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

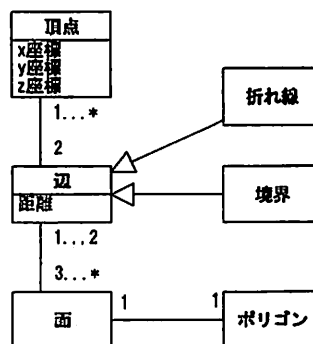


図 3: クラスのUML表記

いるものは折れ線クラス、それ以外は境界クラスとする(図3)。

ユーザーは、仮想空間が投影された二次元のスクリーンを介して仮想空間内のポリゴンを操作する。そして内部では次に説明する他の面を追従させて動かすためのプロセスが走っており、面を動かすとそれに応じて他の面がリアルタイムに再配置される。また、スクリーン上の二次元座標(スクリーン座標)と仮想空間上の三次元座標は相互に変換可能である。

3.1. バネモデル

一般的に、計算機に接続されているマウスは一つであり、そのためマウスで一度に操作できる面の数は一つのみである。しかし、折り紙の場合は折り重なっている多数の面が同時に動くため、一度のマウス操作で複数の面を矛盾なく同時に動かす仕組みが必要となる。

曲面の存在を無視すれば、折り紙を構成する各面は剛体であるとみなすことができる。それは同じ面内で頂点間の距離は常に一定であるとも言え換えることができ、この制約条件からある面の移動後の他の面の移動先を計算によって求めることが可能である。その計算は、距離が一定、または成す角が一定といった条件から解析的に求めることもできるが、計算機を用いてリアルタイムに解くには適さない。そこで本プログラム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、接続されている個々バネの弾性力の合計を現時刻の頂点に加えることによって次の時刻の頂点の位置を求めることとした(図4)。

一般的に、ある頂点 i にかかる弾性力 F_i は(1)式で表すことができる[6]。

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

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が計算でき、面の変形が計算できる。運動方程式の数値計算はオイラー法により行う。具

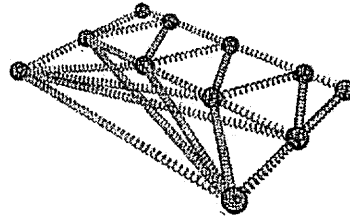
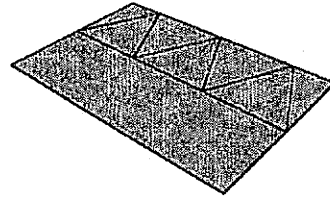


図4：面を構成するすべての頂点間にバネを張る

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 は衰減定数であり、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。

3.2. 軸の選択と面の回転角の決定

本プログラムでは紙を折るという操作を、辺を軸として面を回転させるモデルで表現している。また、軸に垂直で軸上のある点を通る平面とマウス座標をマッピングさせ、その移動量から面の回転角を求めている(図5)。それによって二次元座標しか入力できないマウスのみでも直感的に三次元的な操作が可能となっている。

操作の対象となる面は、スクリーン上をマウスボタンでクリックすることによって決定される。もしクリックした位置にポリゴンが投影されていれば、そのポリゴンに対応する面を操作対象として選択するとともに、クリックしたスクリーン座標に対応する三次元座標を取得し、それに応じて軸を決定する。このとき軸に選ばれるのは、直感的に「面の中心から見て、クリックした位置の反対にある折れ線」として理解される辺とする。現実世界で紙を折ろうとするときは、折り目から遠い、紙の縁付近をつまんで作業することが多い。そのため本プログラムではその動作を参考とした。

面の回転は、選択した後マウスをドラッグするインターフェースで実行される。スクリーン上でマウスボ

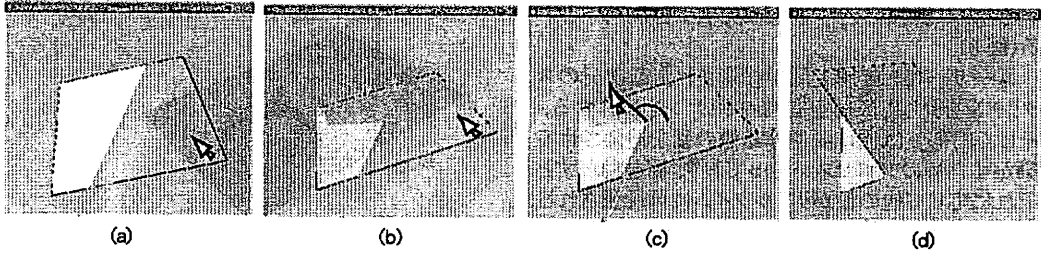


図5：面回転

- 面をクリックして選択する。選択された面のうちクリックした点の反対側にある折れ線を軸として選択する。
- クリックした点を通り、軸に垂直な円盤を表示する。
- 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。
- 軸を中心として、面を回転させる。

タンを押すことによって面と軸を選択した後、軸に垂直で、マウスでポイントした座標に対応する三次元座標を含む平面を定義する。その平面上にマウスドラッグの始点、終点に対応する三次元座標を定義し、それに軸との交点を加えた三点から角度を求め、2 軸と回転軸との内積から角度の正負を決定し、面の回転を実行する。

なお、軸と視線が垂直に近い角度で交わるとマウスでの操作が困難になるため、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。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(図 5b)。

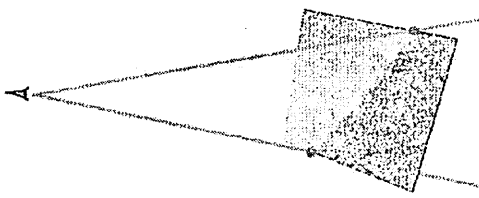


図 6：視点、始点、終点を含む三角状の領域とすべての辺で交差判定を行う

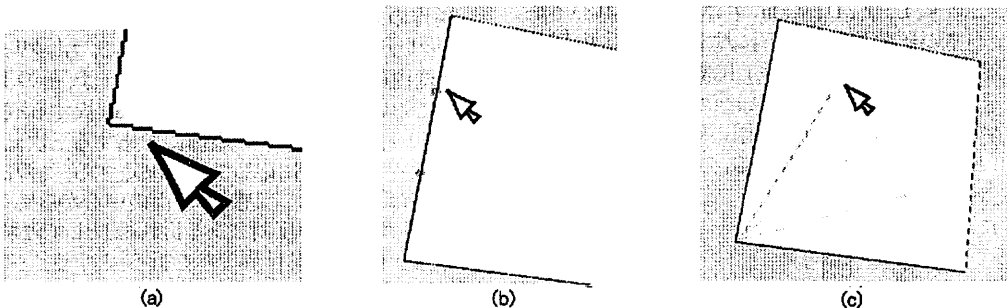


図7：カーソル座標の近似

- 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カーソル座標をその位置へ一致させる

3.3. 折り目の入力

折り紙を折ることを対話的に行うためには、折り目はユーザーが自由に付けられなければならない。一般的なドローツールではキャンバス上をマウスでドラッグすると軌跡が線となって描かれるが、本プログラムでもそのインターフェースを参考とした。

マウスでスクリーン上をドラッグすると、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。マウスボタンから指を離すと、視点から両点に対応する三次元座標を通る二本の半直線を生成し、その間に挟まれた三角状の領域をとすべての辺との交差判定を行い、適切に面分割を実行する(図 6)。

なお、マウス座標と頂点の座標の間の距離があらかじめ設定された閾値以下の場合はマウス座標を頂点座標へと誘導させる。同様にマウスが辺の上に来た場合、その場所が辺を n 分割($n=2,4,3$; 順序は優先度)する点に近い場合はガイドを表示しマウス座標をその位置へ、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($n=2,4,3$)する点に近い場合はマウス座標をその位置へ一致させる(図 7)。

3.4. 押し潰し

本プログラムを用いると立体的な造形を簡単な操作で生成できるが、複数の面を同一平面として扱う仕組みがないため操作中に意図せず形状が立体的になってしまうケースが頻繁に発生する。このようなケースでは面の分割の際に誤差が発生してしまうため、ユーザーが任意のタイミングですべての頂点の座標を仮想空間の奥行き方向に圧縮させ、厚みをゼロにする機能を実装した(図 8)。

なお、この処理では頂点間の距離に関する制約を考慮せず単純に z 値を減少させるだけなので、この処理によって変形された形状に対しては再度バネ計算を繰り返し行っている。

4. 結果と考察

本プログラムを用いることによって、図 9 のような形状を折ることができる。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。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もできるだろう。

しかしリアルタイム性を重視した結果、本プログラムでは面同士の衝突判定アルゴリズムを搭載することができなかった。そのため面同士がすり抜けてしまったり交差してしまうケースが発生している。それを解決するためのアルゴリズムの考案が今後の課題である。

2006 年 2 月 14-19 日にかけて、名古屋市東区の市民ギャラリー矢田にて一般市民を対象にして

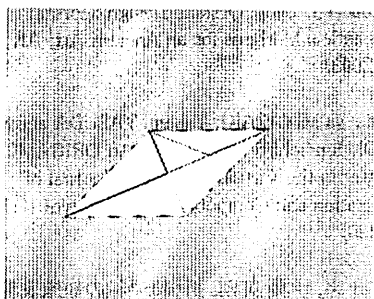
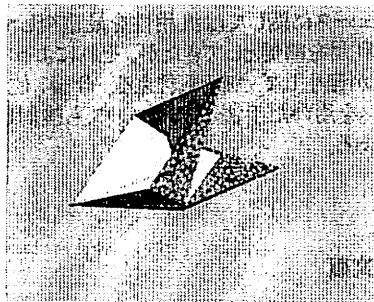
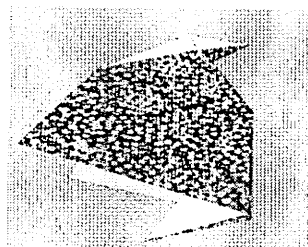
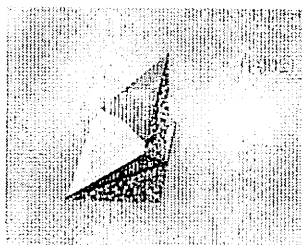


図 8：意図せず立体的になってしまった形状（上）に対して押しつぶし動作を連続して加えた結果（下）

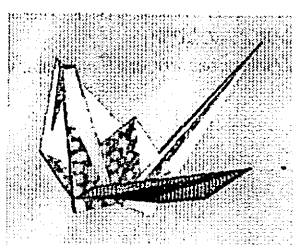
本プログラムを実演する機会があった(図 10)。その場で数人に操作をしていただいたが、慣れの問題もあると思われるがあまり芳しい評価をいただけなかった。これはインターフェースに関する問題であると考えられる。それに関連して、本プログラムでは両手でつかんで引っ張ったりする操作をシミュレートすることができないため、製作に際して両手の使用が必須となる形状を生成することができない。インタ



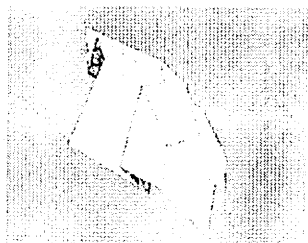
中折り、カブせ折り



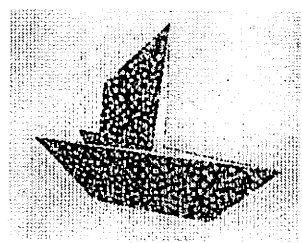
折鶴の基本形（途中）



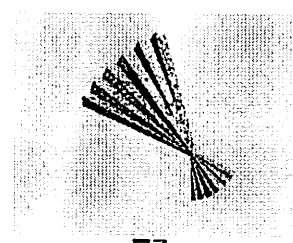
折鶴



ねじり折り



だまし角



扇子

図 9：折りあがり形状のサンプル

ーフェースに関しては汎用性を最重視しマウスのみでの利用を前提としてきたが、それによってこれらの問題が発生しているとも言える。今後、他の入力デバイスをを用いた手法などについても検討していく必要がある。

参考文献

- [1] Robert J. Lang: Origami Design Secrets: Mathematical Methods for an Ancient Art; AK Peters, Ltd. (2003)
- [2] 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(1), pp.25-42 (1996)
- [3] 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, 32(12), pp.1566-1573, (1991)
- [4] Jien KATO, Toyohide WATANABE, Hiroyuki HASE, Takeshi NAKAYAMA: Understanding Illustrations of Origami Drill Boo; 情報処理学会論文誌, Vol.41, No.6, pp.1857-1873 (2000).
- [5] 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)
- [6] 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)

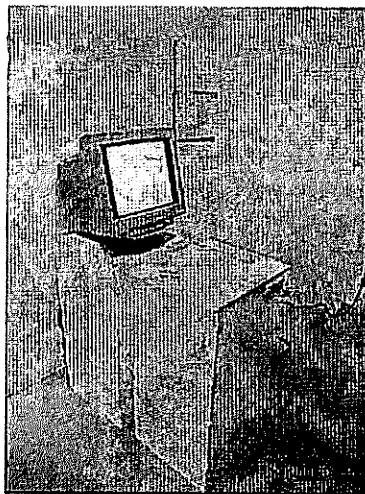


図 10: 展示風景

ユーザーによる意図しない操作を避けるため、必要最小限のスイッチのみで構成されたキーボードを製作して使用している