

# 3D 折り紙アニメーションのためのモデリングソフトウェアの開発

システム情報工学研究科 1年 200620987 古田 陽介

指導教員： 三谷 純 福井 幸男

2006年11月2日

## 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るようになった。また折り紙は古くから幾何学との関連性が知られており、学問の世界では数学を中心として以前から研究が進められてきた。さらに、指先を使って紙を折ることは脳の活性化にもつながることから折り紙は数学の教材としても注目を集めている。

し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。複雑な手順を折り図で表現するためにはその作図に多大な労力を必要とすること、折り図では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ことが複雑化する折り紙の折り手順を他者に伝達することを困難にしている。

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。VRML や Web3D などの技術と組み合わせることでインターネットを通じた情報配布などへの応用も期待されているが、その形状を計算機内に作成するためには CG や CAD の知識、高価な 3D モデリングソフトが必要となる場合が多いという欠点がある。またそのモデリングの過程も一般の折り紙を折る手順とは全く異なるものであるため、折り紙作家は折り紙に関する知識だけではなくモデリングソフトの扱いについても精通しなければならず敷居の高いものとなっている。折り紙のモデリングに特化しシンプルな操作でモデリングを行う手法の研究も行われているが、現状では多種多様な折り紙の形状や紙に対する操作を完全に再現するには至っていない。

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インタフェースの問題である。一般に折り紙を折る際には両手を使用するが、10本の指の動きを計算機に取り込むためには特殊な装置が必要である場合が多く、そのため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場合が多い。そのため、既存の多くの 3D モデリン

グソフトはマウスによる二次元入力から三次元的な操作を行うために異なる視点からの正投影図を複数表示するインタフェースを用いているが、これは直感的であるとは言えず 3D モデリングソフトの扱いづらさを増長している。

そこで本研究では一般的な入力装置でも直感的に三次元折り紙モデルの生成を行えるような計算モデルとインタフェースを提案し、実際に計算機上に実装を行った。

## 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。

計算機上で折り紙を扱う研究としては、内田らによる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[5]。また、井田らは折り紙の持つさまざまな幾何学的性質を検証したり証明するためのシステムを計算機上に構築している[6]。

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[7]。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[8]。

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[9]。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。本研究では面の相互作用による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。

## 3. 提案手法

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

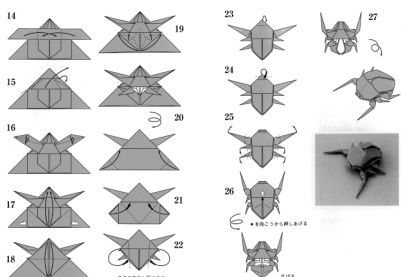


図1：  
テナゴコガネの  
折り図[3]

せであることが多い。しかしこれらの動作は両手が使えることが暗黙の前提となっており、複数の指の三次元的な動きを入力することができない計算機環境でその動作をシミュレートすることは難しい。

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。そこで本論文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、それに「隣接面を付随して動かす」仕組み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を提案する。三次元の面の表面に線を描くことや、ある軸を中心として回転させることは二次元で表現することが可能であるため、提案手法を用いればマウスやキーボードのみしかない一般的なPCでも折り操作をシミュレートすることが可能となる。

また、本論文ではマウスで直感的に面の分割や回転操作を行うための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についても提案する。

### 3.1. 付随する面の位置計算

折り曲がり(湾曲)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が、互いに連結した複数の剛体の三次元的な運動を実時間で求めることは一般的に困難な問題である。また平面幾何とともに説明されることが多い折り紙の形状であるが、現実の折り紙には紙の反りなどを多用した数式で表現することが難しい形状や折り方が多数存在している。

そこで本プログラムでは、ある面に着目したときにその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2b)、各頂点に接続されている個々のバネの弾性力の合計から微小時間経過後の頂点の位置を求めることとする。このアルゴリズムを用いることで、数式では表現することが困難な形状、操作も単純な計算の繰り返して高速に求めることができる。

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[10]。

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

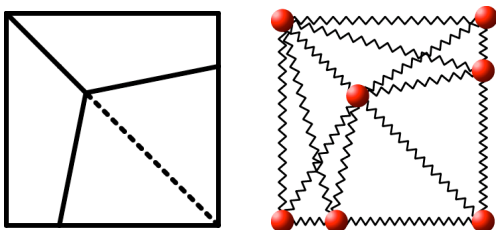


図2：展開図(a)とバネモデルによる形状表現(b)

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。

(1)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、(2)、(3)式により  $\Delta t$  秒後の頂点  $i$  の位置  $r_i$  と速度  $v_i$  を計算できる。

$$v_i(t + \Delta t) = v_i(t) + d \frac{F_i(t)}{m} \Delta t \quad (2)$$

$$r_i(t + \Delta t) = r_i(t) + v_i(t) \Delta t \quad (3)$$

ここで  $d$  は時間係数であり、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. 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きと始点、終点、視点の三点で定義される平面で形状を分割することによって、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」ことができるようになっている(図3)。

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、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可能である。

### 3.3.2 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」を軸として自動的に選択することによって直感的に回転操作が行えるようになっている。

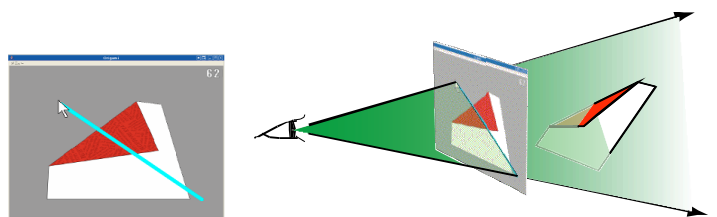


図3：折れ線の入力

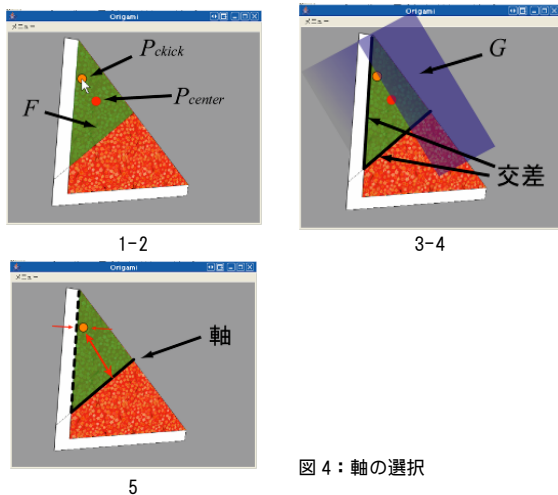


図 4：軸の選択

1. 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。
2. 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
3.  $P_{click}$  と  $P_{center}$  を通り、 $F$  に垂直な面  $G$  を定義
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

なお、本アルゴリズムでは軸の自動選択を行うことができるが、面を構成している辺以外を軸とする場合に備え手動で軸を指定することもできる。

### 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て面を回転させる(図 5)。 $G$  上には軸を中心とした円を表示し、面が軸を中心として回転することを視覚的に分かりやすくしている。これらの処理

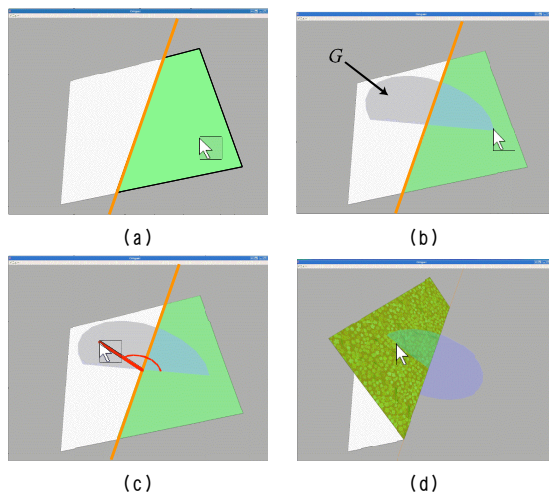


図 5：面回転

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うことができる。

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しまい回転角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なっている。

## 4. 結果

本プログラムを用いることによって、図 6 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) x2  
RAM: Samsung PC2100 DDR-SDRAM 3.5GB  
GPU: 3Dlabs WildcatIII 6210 (VRAM: 416MB)

モデリングに要した時間は鳳凰、カタツムリの場合で約15分、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 7)。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、隣接していない面同士の相互作用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 6 の鳳凰、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衝突判定を行って適切に移動させる処理を追加する必要があるだろう。

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, vol82, Issue3, pp.157-161, (2005)

- [3] 桃谷好英, 誠文堂新光社: おりがみ虫の世界, (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, 32(12), pp.1566-1573, (1991)
- [6] 井田 哲雄: Computational Origami Project; <http://www.score.is.tsukuba.ac.jp/~ida/Ida2004/CompOrigami.htm> (2001)
- [7] 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).
- [8] 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)
- [9] 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)
- [10] 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)

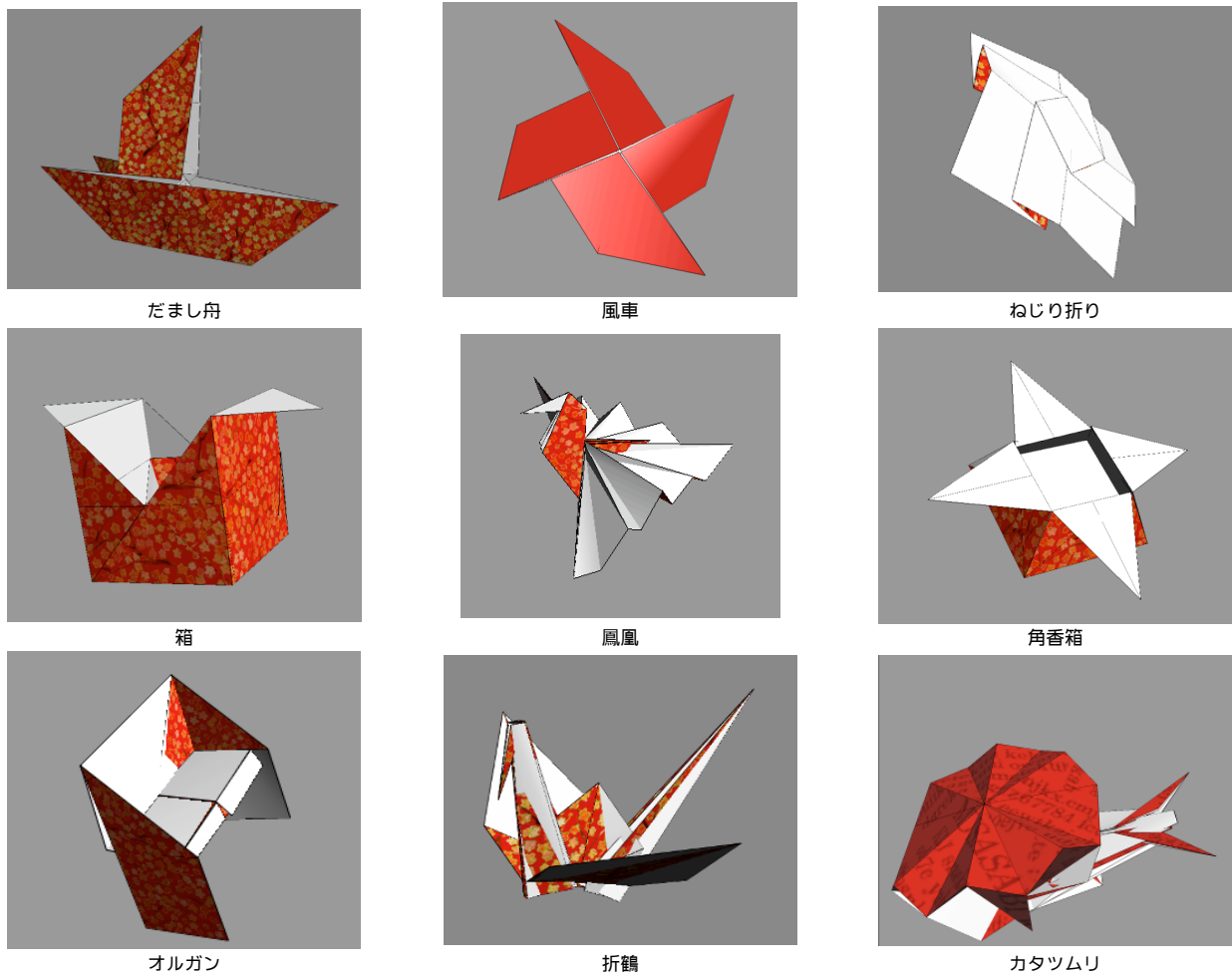


図 6 : 作例

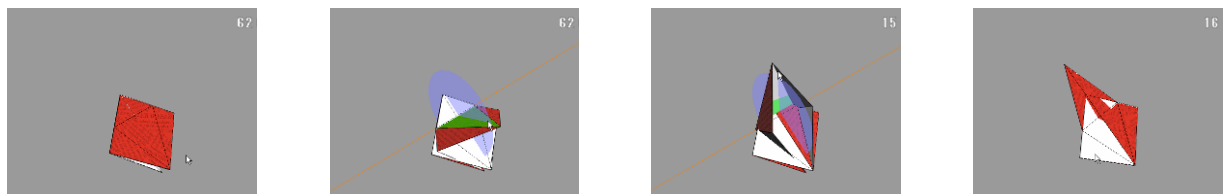


図 7 : 花弁折りを本プログラムと既存のプログラム[8]で行った場合の比較