

筑波大学大学院博士課程

システム情報工学研究科修士論文

正三角格子から作り出される
立体折り紙の設計支援

鳥海遼太

修士（工学）

（コンピュータサイエンス専攻）

指導教員 三谷純

2017年 3月

概要

古くから日本の文化として親しまれている折り紙は、伸縮しない素材を折ることで様々な形状を作り出す技術として、工学分野において研究されるとともに、その幾何学的な操作に関する数理的な研究も広く行われている。また、様々な折りの技法が考案されたことにより、紙による立体造形をアート作品として鑑賞の対象とすることも多い。紙を折って作り出される形状は、その幾何学的な制約の厳しさから、意図した形を自由に作り出すことは難しく、近年では計算機を用いた設計支援が広く行われるようになってきている。

本研究では、従来一般的には設計の対象とされてこなかった正三角形格子パターンに着目し、この格子パターンに折り線を配置することで作り出される立体折り紙の設計支援を行うためのシステムの開発を行った。提案システムでは、格子点に質点を配置し、それらをバネで連結したバネマスモデルを用いた物理シミュレーションを用いることで、対話的な形状操作を実現した。また、設計段階で回転対称性を持たせるなどの制約を付加することで、対称性を持つ構造を容易に作り出せるようにした。また、得られた立体の展開図を出力し、実際に紙を折って制作することで、その形状が実際に作れることを確認した。正三角形格子パターンは、紙を折って作り出すことが容易なうえ、構成要素が合同な三角形であることから、大きなスケールで作る際にも、単一の部材を組み合わせて作ることができるという利点がある。また今後、格子に乗らない折り線を加えるなどの操作を追加することができれば、より自由度の高い形状設計へ拡張可能である。

目次

第1章	序論	1
1.1	はじめに	1
1.2	本研究の目的	1
1.3	本論文の構成	2
第2章	関連研究	3
2.1	折り紙設計	3
2.1.1	平坦折り紙設計	3
	TreeMaker	3
	シルエットからの設計	4
	折り紙テッセレーション	4
	剛体折り紙	4
	ORIPA	5
	折りたたみ形状の提示	5
2.1.2	立体折り紙設計	6
	直線折り	6
	曲線折り	7
2.1.3	仮想的な折り紙シミュレーション	7
	折り紙シミュレーション	7
	Foldinator	7
	折り紙シミュレータ	8
2.2	多面体の設計	8
2.3	物理シミュレーション	9
第3章	提案手法	10
3.1	正三角格子バネマスモデル	10
3.1.1	従来手法における問題点	10
3.1.2	バネマスモデル	11
3.1.3	バネマスモデルにおけるパラメータ設定	12
3.2	バネマスモデルの変形	14
3.2.1	基本的な変形操作	14
3.2.2	回転対称性に基づく変形操作	15
3.3	山谷情報を持った展開図の出力	17
第4章	結果	18
4.1	実装したアプリケーション	18
4.2	紙を用いた作成結果	18
4.2.1	基本的な変形操作	18
4.2.2	回転対称性に基づく変形操作	18
4.3	数値的評価	22

第 5 章 結論.....	23
5.1 まとめ.....	23
5.2 今後の課題.....	23
謝辞.....	25
参考文献.....	26

図目次

図 2-1 川崎定理	3
図 2-2 TreeMaker の実行画面	3
図 2-3 シルエットからの木構造推定	4
図 2-4 Tess の実行画面	4
図 2-5 Freeform Origami による設計例	5
図 2-6 ORIPA の実行画面（展開図と平坦折り予想図）	5
図 2-7 45 度系格子パターンから得られる形状と展開図の例	6
図 2-8 Origamizer による設計例	6
図 2-9 ORI-REVO の実行画面	7
図 2-10 折り紙シミュレーションによる作成例（左が紙飛行機，右が鶴）	8
図 2-11 Foldinator の実行の様子	8
図 2-12 折り紙シミュレータによる作成例	8
図 2-13 Tsuruta らによって列挙されたデルタ多面体の例	9
図 3-1 正三角格子バネマスモデル	11
図 3-2 振動の種類とその様子	13
図 3-3 実装したパラメータにおける単一バネの変形の様子	14
図 3-4 回転対称性に基づく設計操作	15
図 3-5 対応する triangle に応じた頂点の探索	15
図 3-6 axis に所属する頂点群に関する移動の制約	16
図 3-7 山谷判定の様子	17
図 3-8 法線ベクトルを参照する格子三角形の選択	17
図 4-1 実装したアプリケーションのユーザインタフェース	19
図 4-2 基本的な変形操作による設計および作成結果	20
図 4-3 回転対称性に基づく変形操作による設計および作成結果	21

表目次

表 4-1 実装したアプリケーションの機能一覧.....	19
表 4-2 作成したモデルにおけるバネの長さの計測結果.....	22

第1章 序論

本章では、まず研究の背景と目的を述べ、次に本論文の構成を述べる。

1.1 はじめに

日本において、古くより折り紙は芸術・遊びとして広く親しまれている。しかし、単なる文化として捉えられるだけでなく、紙を折るという操作に見られる幾何学的な性質から、数学的な観点からも注目されるようになってきている。また、近年では数学のみならず、工学から建築、芸術、教育など幅広い分野から注目され、研究の対象となっている。

折り紙が幅広い分野において応用される背景として、折り紙を活用した形状設計に関する研究の発展、またそれに伴って確立された様々な設計手法がある。従来は試行覚悟によって行われた新たな折り紙作品の設計において、数学的な理論を用いることでより複雑な形状を設計することが可能となったためである。こうした研究分野は折り紙設計と呼ばれ、近年でその性能を大きく向上させているパーソナルコンピュータはその発展へ大きく寄与している。PC の計算能力によって折り紙設計の手法が日進月歩し様々なソフトウェアが開発される一方で、コミュニケーションや教育といった目的においてもその力は大きな影響を与えている。こうした発展の下、研究対象としてのみならず、折ること自体を楽しむ折り紙、すなわち遊戯折り紙としての側面が再び注目されている。

しかしながら、折り紙設計という分野が発展し、より複雑で美しい作品が創り出される一方で、折り紙について熟練した者とそうでない者との間に差が生じてしまっているという問題がある。特に、完成後の形状が立体となる折り紙においては、折り紙について熟練した者でなければ新たな作品の設計は困難である。

そこで本研究では、立体折り紙における新規作品の設計の補助を目的として研究を行った。

1.2 本研究の目的

本研究の目的は、立体折り紙における新規作品の設計を簡易的に行うことができるような手法を提案することである。しかしながら、立体折り紙に含まれる範囲は非常に大きく、そのすべてを対象としたシステムの構築は現実的ではない。そこで本研究で対象とする立体折り紙は、展開図における折り線が正三角格子上に位置するものとする。展開図が格子パターン上に位置する折り紙は伝承的な作品においても数多く見られるものでもあり、格子パターンを用いることで幾何学的で見た目のよい形状を得ることができるからである。

具体的には、正三角格子パターンから作り出される、構造に対称性を持つ折り紙について、計算機上でユーザーが CG モデルを作り出した形へと変形させることで設計を行い、それを作り出すための山折り・谷折りの情報を持った展開図を出力することで、目的を達成する。紙のモデリングにはバネマスモデルを用いることで、対話的で柔軟な操作を可能とする。これによって、従来は個々の経験に依る部分の大きかった立体折り紙の設計において、ユーザ

一に依存しない気軽に綺麗な折り紙による造形が期待される。

1.3 本論文の構成

本論文は本章を含めて全 5 章で構成される。以下、第 2 章では本研究に関連する研究を紹介する。第 3 章では、本研究の提案手法、第 4 章では結果、第 5 章では結論を述べる。

第2章 関連研究

古くから伝わる伝承折り紙(例えば鶴など)は先人の試行錯誤によって創り出されてきた。しかしながら、試行錯誤によるアプローチでは、複雑な構造を持つ作品の創出は困難であった。そのような背景から、数学的なアプローチでもって折り紙を設計するという研究が発展してきている。

2.1 節では、そのような過程によって生まれた折り紙の設計手法に関する研究を紹介する。2.2 節では、本研究で取り扱う形状に関連する、多面体の設計に関する研究を紹介する。2.3 節では、本研究でも用いた物理シミュレーションについて述べる。

2.1 折り紙設計

数学的な観点から確立された、数学的なアプローチによる様々な折り紙設計の手法について、平坦折りを対象とした設計手法と、非平坦なものを対象とした設計手法を紹介する。

2.1.1 平坦折り紙設計

折り紙作品として広く知られる作品の多くは、平坦に折りたためるものである。ある作品が平坦に折りたためるかどうかは、折り線の構造、すなわち展開図のパターンによって判定される。紙内部の折り線の交点(以後、頂点とする)に注目したとき、その頂点が平坦折り可能であれば、折り線の為す角の1つ置ききの和が 180° であり[1]、山折りと谷折りの数の差が2である[2,3](図2-1)。前者は川崎定理、後者は前川定理として知られており、これらは1頂点に注目した際の局所的な平坦折り可能性を判定する条件である。

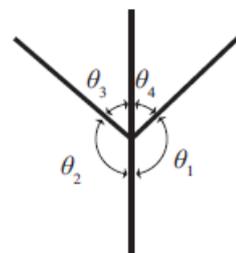


図 2-1 川崎定理

TreeMaker

折り紙作品として創り出したい構造を線分の集合(木構造)とみなし、それらの突起(いわゆる折り紙におけるカド)を作るのに必要となる円形の領域を展開図上に配置する手法に基づく設計アプローチが考えられた。このような手法は circle packing や tree method などと呼ばれ、このアルゴリズムを実装した TreeMaker という折り紙設計ソフトウェアがある[4,5](図2-2)。木構造を入力とし、その構造を折るための展開図が出力されるため、出力として得られるものは基本構造のみとなる。

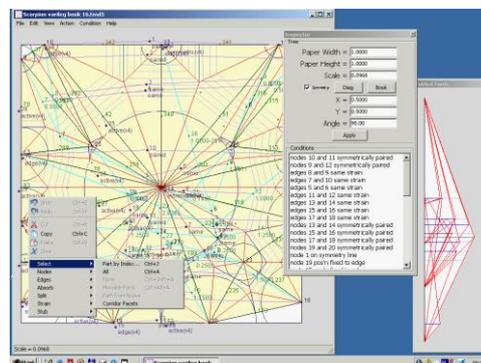


図 2-2 TreeMaker の実行画面

この手法によって出力される展開図は **uniaxial base** と呼ばれる基本形となり、鶴の基本形など伝統的な折り紙作品においても多く見られる基本形である。また、細部の形状についてはユーザーに委ねられる。

シルエットからの設計

前項で述べた **TreeMaker** の入力は木構造であり、折り紙に習熟したユーザーでなければ使用が難しいという点が挙げられる。そこで、より直接的なアプローチとして、折りたたまれた最終形の形状を入力として設計することが考えられる。**Shimanuki** らは、動物などのイラストから骨格構造を抽出し、それを元に折り紙設計を行う手法を提案している[6] (図 2-3)。この骨格構造は、前項でいう木構造に相当する。

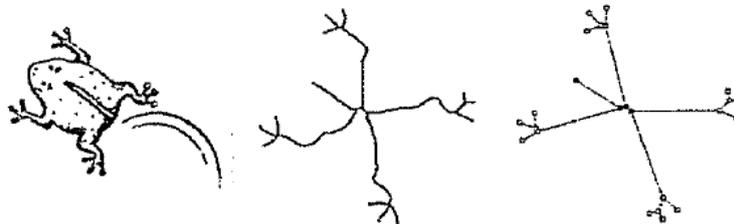


図 2-3 シルエットからの木構造推定

折り紙テッセレーション

幾何学的なパターンを作り出すように紙を折ることを折り紙テッセレーションまたは平織りといい、これは特定の折り線構造を平面上に敷き詰めることによって実現される。幾何学的なパターンから作り出される見た目の美しさから、折り紙の中でも注目される分野のひとつである。このテッセレーションに関連するソフトウェアとして **Tess** というものがある[7] (図 2-4)。これは、タイル形状と中心図形の大きさ、傾きの角度を入力とし、テッセレーションを生成する展開図を出力するものである。

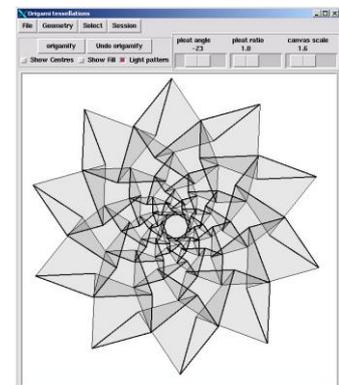


図 2-4 Tess の実行画面

剛体折り紙

剛体折り紙とは、折り線で区切られた領域を剛体のパネルとみなし、それらのパネルがヒンジで結合されたモデルとして展開図を扱うものである。剛体折り紙はその性質から工学的な応用が強く期待される分野であり、ソーラーパネルに応用された例などもある。

剛体折り紙の折りたたみをシミュレーションするソフトウェアとして、**Tachi** による **Rigid Origami Simulator** がある[8]。また、その発展として開発された **Freeform Origami** では、平坦折りが可能であり、また同時に剛体折りが可能という条件を満たしながら、展開図の設

計を行うことができる[9] (図 2-5).

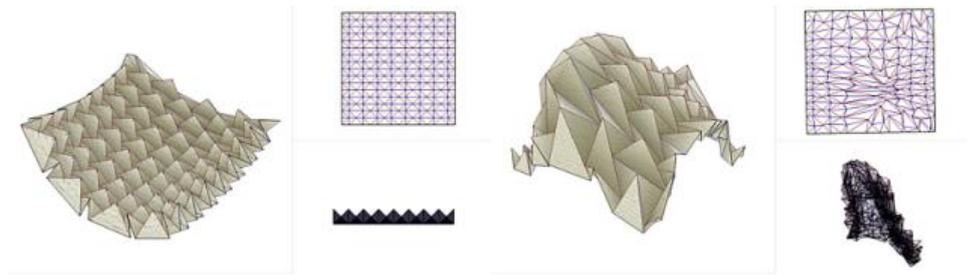


図 2-5 Freeform Origami による設計例

ORIPA

折り紙の展開図を効率的に入力する折り紙の展開図用エディタとして, Mitani らによる ORIPA がある[10,11] (図 2-6). 展開図の入力に特化しており, 角を二等分する線, 指定した点から指定した線分に垂線を引く, 指定した線分群を線対称に複製する, など展開図において典型的にみられるパターンを効率的に描くことが可能である. また, 平坦折り可能な展開図に関して, 折りたたみ後の完成予想図を表示する機能も実装されている.

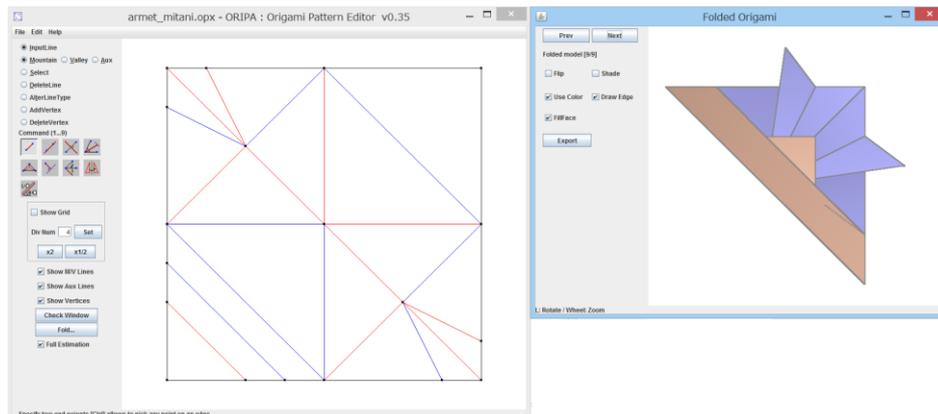


図 2-6 ORIPA の実行画面 (展開図と平坦折り予想図)

折りたたみ形状の提示

折りたい作品の構造を得るのではなく, 折りたい作品の形状を得ることを目的として開発されたソフトウェアがある[12]. これは, ユーザーが目的の形状を入力し, それに類似した数回の基本的な折りたたみのみによって得られる形状を列挙するという手法により実行される.

同様の形状の列挙を用いた研究として, 展開図における折り線が 45 度系格子パターンに重なるものに限定したものがある[13] (図 2-7). 風車やだまし船などの伝統的な折り紙作品の数多くが 45 度系格子パターンに含まれており, その中でも平坦に折りたたむことのできるものを列挙している.

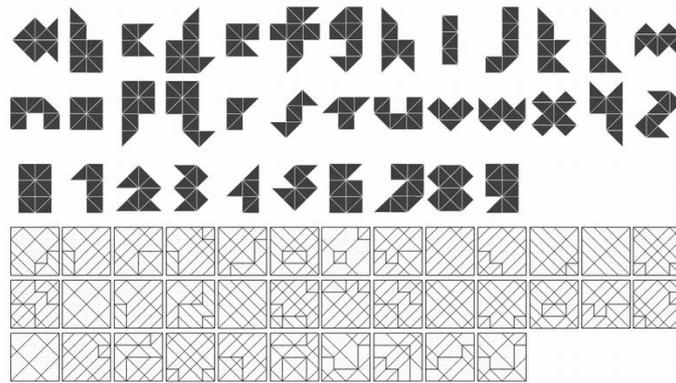


図 2-7 45 度系格子パターンから得られる形状と展開図の例

2.1.2 立体折り紙設計

折り紙設計についての研究の発展に伴って，完成後の形状が立体となる折り紙の設計についての研究が行われるようになった．しかしながら，平坦折りについての数理がほぼ整っているのに対し，立体折りには数学的に未知な部分も未だ存在する．そのため，対象を限定し，特定の構造を持つ形状の設計に特化した手法を提案しているものが多い．ここでは，展開図の構造に注目し，展開図が直線のみで構成されるもの，曲線を含むものに分けて紹介する．

直線折り

直線折りを用いて立体形状を設計する手法として，ある決まった構造パターンを組み合わせるいくつかの手法がある．Tachi の開発した **Origamizer** は，三角形の集合で表現された 3 次元形状を入力として，それを 1 枚の紙で折ることが可能な展開図を出力するソフトウェアである[14]（図 2-8）．**Origamizer** では，各三角形面の上に「ひだ」となる構造を組み込むことで，1 枚の紙による 3 次元形状の表現を実現している．また，**Zhao** らは展開図の一部を入力することで，複数の三角形で構成される軸対称の立体折り紙を対話的に設計する手法を提案している[15]．

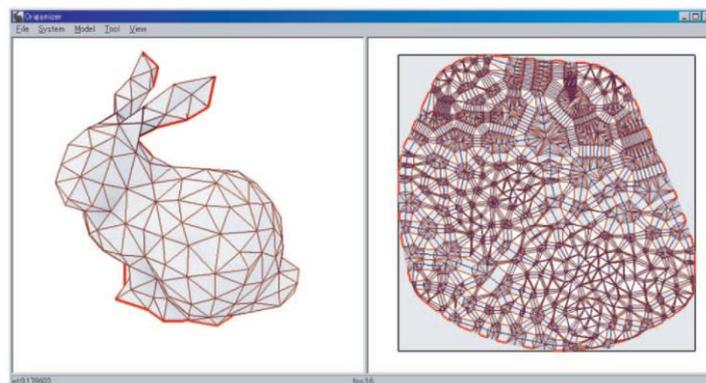


図 2-8 Origamizer による設計例

曲線折り

紙という特性を活かした、曲線を折り線とする折り紙作品も古くから存在する。Mitani は断面となる折れ線を入力とし、軸対称な立体折り紙を設計する手法を提案し、それを実装した ORI-REVO を公開している[16] (図 2-9)。これは、前項で述べたような、「ひだ」を加えることで 1 枚の紙による展開図を設計するものである。また、可展面で表現された立体を、空間中の平面で鏡映反転することで、曲線で構成された折り紙を生成するソフトウェアも開発した[17]。

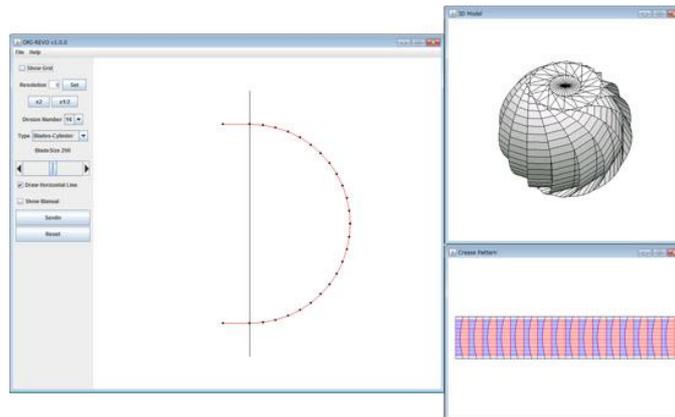


図 2-9 ORI-REVO の実行画面

2.1.3 仮想的な折り紙シミュレーション

前項までで紹介した折り紙設計の研究は、計算機を活用することで実現したものである。同様に計算機を活用した折り紙に関する研究として、折り紙をシミュレーションするというものがある。

折り紙シミュレーション

Miyazaki らは、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ている[18] (図 2-10)。これは、仮想空間上で紙の頂点をマウスカーソルで掴んで移動させるという操作を繰り返しことで、仮想的に折り紙のシミュレーションを行うものである。これは曲面の生成も可能であるが、同一平面上に無い複数面の動きを計算できないため、面の相互作用による 3 次元形状への影響を無視できない。

Foldinator

Szinger らによって開発されたソフトウェアでは、折り紙モデルと同時に、それを折る手順を示す折り図を出力することが可能である[19,20] (図 2-11)。仮想的な紙に対して様々な折り操作を加えることができ、また各工程において折り方や折り方向を示す記号や注釈を加えることで、作品と折り図の出力が可能となっている。

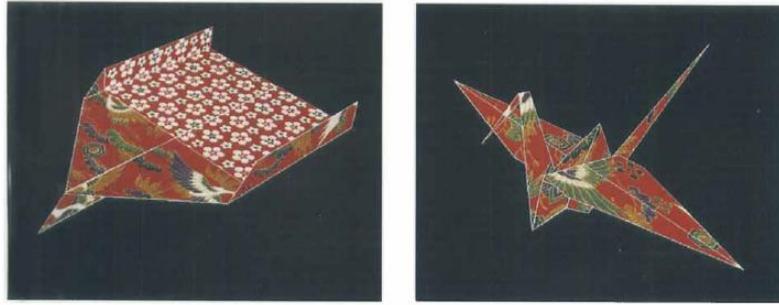


図 2-10 折り紙シミュレーションによる作成例（左が紙飛行機，右が鶴）

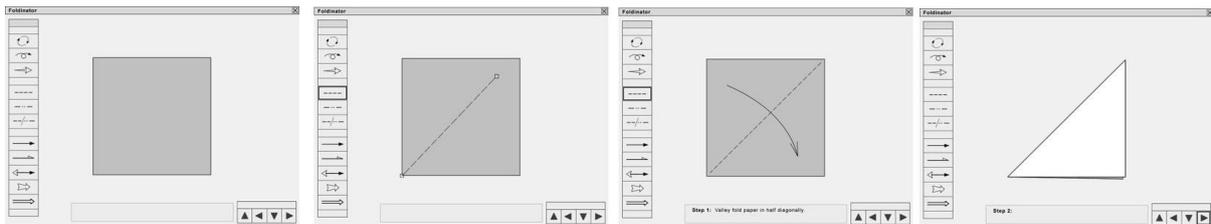


図 2-11 Foldinator の実行の様子

折り紙シミュレータ

古田らによって開発されたソフトウェアでは，対話的に折り紙形状をモデリングできる計算モデルを実装している[21]（図 2-12）．仮想的な紙に対して折り操作を加えることで折り紙をシミュレートするだけでなく，折り手順をアニメーションで表示することで，折り紙の作成を支援することも可能となっている．また，本手法でも用いたバネマスモデルによる紙のモデリングを行っており，紙の振れや複数面の相互作用による複雑な挙動をシミュレーションすることができる．

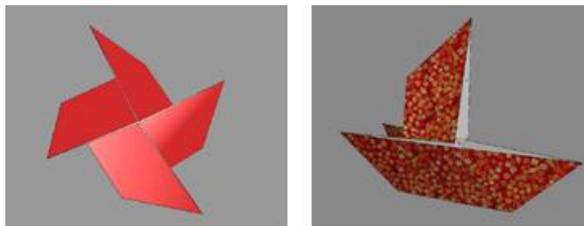


図 2-12 折り紙シミュレータによる作成例

2.2 多面体の設計

汎用的な 3D モデルを作成するソフトウェアは数多く存在するが，多面体を対象としているものは多くない．多面体を対象としたモデリングツールとして，Stella がある[22]．正多面体など代表的なモデルがあらかじめ用意されており，それら多面体同士の連結などの操作を加えることができる．

また、正三角形のみによって構成される多面体（デルタ多面体）についての研究として、Tsuruta らによるものがある[23]（図 2-13）。これは、多面体グラフのうちデルタ多面体として実現可能なものを数え上げるものである。これによって列挙されるデルタ多面体は、複数枚の折り紙を組み合わせてによって構成されるユニット折り紙によって実現することが可能である。

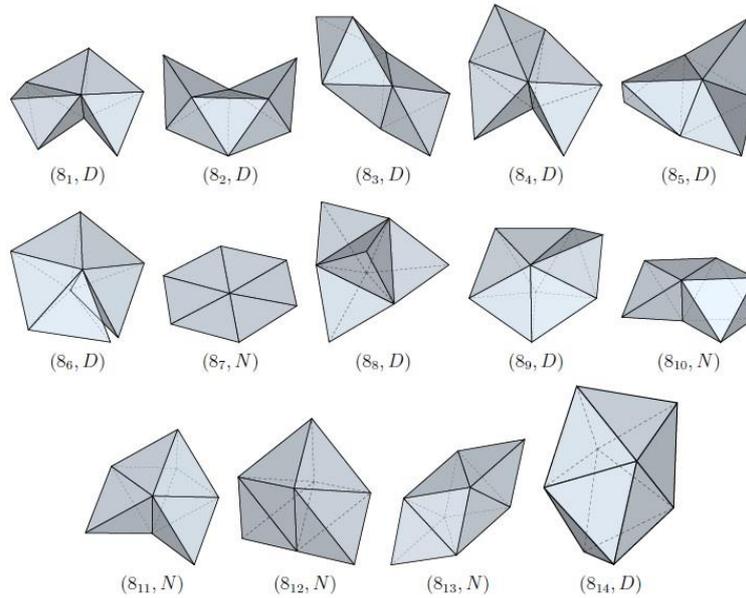


図 2-13 Tsuruta らによって列挙されたデルタ多面体の例

2.3 物理シミュレーション

本研究では、モデリングの際に物理シミュレーションの手法を取り入れている。CG における物理シミュレーションを用いた事例としては、布の動きを計算するクロスシミュレータや、流体シミュレータなどが市販のモデリングソフトに搭載されている[24]。

また、近年では物理演算機能をライブラリとして公開している例がある。代表的なものとしては、PhysX や HAVOK FX, Open Dynamics Engine などがある[25,26,27]。特に前者二つは GPU の利用によって、多数のオブジェクトに対しても高速に計算することが可能であり、近年のゲームにおいて取り入れているものもある。しかしながら、大きな計算コストを必要とすることや、結果を人為的に制御することが困難であるといった問題点がある。また、ゲーム利用を前提として、体積のあるオブジェクトが比較的疎に配置されている状態を想定するなど、折り紙のように体積のほとんどない薄い面が密に配置されているような状態をシミュレーションするには適していない。

第3章 提案手法

紙は物理的な物体であることから、折り紙の素材として用いる場合には、大きく伸縮しない、分割されない（切れ目が入らない）などの制約が重要となる。そこで本研究では、そのような紙の制約や変形をモデル化し、計算機上で紙が時間的に連続して変形する様子をシミュレーションするためのアルゴリズムを提案する。

本手法では、展開図における折り線が正三角格子上に存在する折り紙を対象とする。まず操作を加える前のベースとなる紙のモデリングについて 3.1 節で述べる。次に、モデルの変形操作について 3.2 節で述べる。最後に、完成したモデルを紙で折るための展開図の出力について 3.3 節で述べる。

3.1 正三角格子バネマスモデル

3.1.1 従来手法における問題点

平坦折り紙については、2.1 節で述べたように様々な手法による折り紙設計が実現されている。そのアプローチは大きく分けて以下の 4 つに大別できる。

- (1). 展開図に折り線を入力
- (2). テッセレーションの元となるパターンを入力
- (3). 完成後の形状列挙からの選択
- (4). 完成後の形状の入力

平坦折り紙と比較した場合、立体折り紙は展開図と完成後の形状との関係性の理解がより困難であり、(1)のアプローチを立体折り紙において応用することは現実的でない。また(3)については、平坦折り紙以上に膨大な候補が生まれることからこれも現実的でない。したがって、(2)もしくは(4)のアプローチを検討するのが妥当となる。ここでは、(4)の完成後の形状の入力という面から考える。

完成後の形状の入力について、木構造を用いた入力方法は直観的な操作が難しいことから本研究の目的において妥当ではない。本研究の目的の 1 つでもある、不慣れたユーザーにも扱うことのできる直観的な操作という観点から、紙をモデリングしそれを操作することで、目的の形状を直接得るというアプローチを考える。紙の持つ制約や変形を考慮したモデル化には様々なアプローチが存在する。代表的な折り紙システムとしては、第 2 章において述べたように Miyazaki らの折り紙シミュレーションや Tachi らの Freeform Origami などがある。

Miyazaki らによる折り紙シミュレーションでは、紙の頂点をマウス操作によって移動させた際に、折れ線が自動生成されて面がマウス操作に追従するようになっている。しかし、この自動生成された折れ線は対象面と同一平面に存在する面のみにはしか生成されず、面同士の相互作用から複雑な挙動を示す折り紙を正しくシミュレーションできない。

Tachi らによる Freeform Origami では、剛体折り紙モデルに基づいたアルゴリズムとな

っており、複数面が同時に動くような場合も扱うことができる。しかし、面を剛体として扱い数学的に厳密であるため、現実の紙のもつ自由度を一切考慮していない。そのため、折る過程において振れが発生するような変形操作を扱う際には、あらかじめ振れが発生しない状態まで面を分割する必要がある。

これらの問題が発生する原因は、折り紙における制約として、局所平坦折りが可能であるという数学的性質のみを対象としていることに起因する。そこで本研究では、現実の紙と同様に柔軟性を持つモデル、すなわちバネマスモデルに基づく計算手法を提案する。このモデルでは、前述した折り紙システムにおけるアルゴリズムと比較した場合に、数学的な厳密さにおいて劣るため、ある程度の誤差が発生してしまうという問題点が存在する。しかし、紙という素材特有の柔軟性から、微小な誤差であれば十分に許容が可能な範囲であり、折り紙のシミュレーションとしては十分適当であると考えられる。また、バネマスモデルを用いた折り紙システムの例として、古田らによるものがあるが[21]、これは実際の折り紙と同様に紙のモデルをシステム上で折っていくことで目的のモデルを得るものであり、直観的に目的のモデルを得るという点で提案手法とは異なるものである。

3.1.2 バネマスモデル

バネマスモデルとは、複数の質点 (Mass) とそれらを結合するバネを用いて対象をモデリングする手法である。折り紙においては、折れ線または稜線で囲まれた各面において、各頂点を質点とし、それらの頂点間にバネを張ることで面の形状を維持するものとなる。本手法で対象とする折り紙においては、図 3-1 に示すように正三角格子の格子点が頂点に相当し、それらの頂点間にバネを張るものとする。

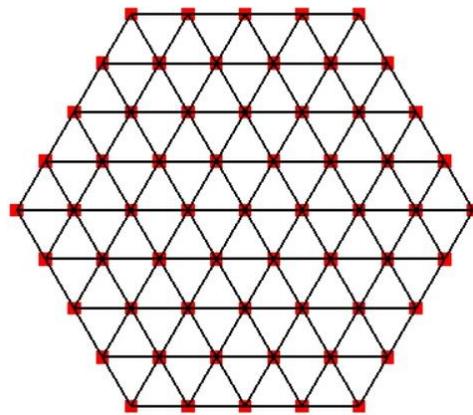


図 3-1 正三角格子バネマスモデル

各頂点は三次元の座標値を保持し、各バネの自然長は両端の頂点の二次元の展開図上での距離に等しい。このとき、ある頂点 i に働く力 F_i は一般的に式(3-1)のように表すことができる。

$$F_i = \sum_j \left\{ k \left(1 - \frac{L_{ij}}{|r_{ij}|} \right) r_{ij} + D v_{ij} \right\} - mg \quad (3-1)$$

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、重力加速度を示す。この運動方程式を解くことで頂点位置の時間変化が計算でき、各頂点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基づいて行う。具体的には、式(3-2),(3-3)によって Δt 後の頂点 i の位置 r_i と速度 v_i を計算する。

$$v_i(t + \Delta t) = v_i(t) + \frac{F_i(t)}{m} \Delta t \quad (3-2)$$

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

上記アルゴリズムは条件分岐の少ない単純なものであり、計算機上で高速に処理を行えることから、ユーザーが対話的に操作を行うのに十分であると考えられる。

またこのとき、ユーザーによって構造体全体である六角形の一辺に存在する格子点数（単位格子正三角形の頂点数）を指定することで、創り出したい折り紙作品の規模に応じた大きさのバネマスモデルを生成する。

3.1.3 バネマスモデルにおけるパラメータ設定

前項の式(3-1)において、設定すべきパラメータはバネ定数 k 、ダンパ定数 D 、および頂点の質量 m である。これらのパラメータを、紙を表現するにあたって適切な数値に設定する必要がある。バネマスモデルにおける頂点の挙動は複数のバネに接続されていることにより振動を示すが、このとき求められる条件は以下の通りである。

- (1). 振動は必ず減衰する
- (2). 現実的な時間で収束する
- (3). 変形操作の際に違和感の無い挙動を示す

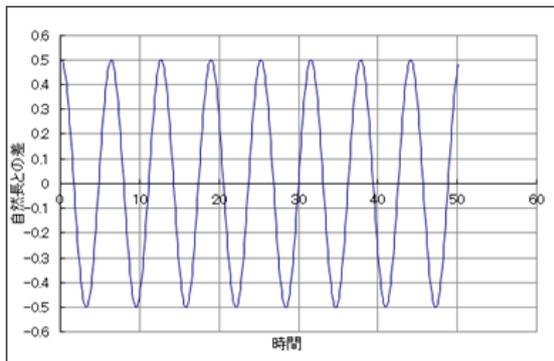
このうち(1)については、提案手法ではダンパを設定しており、速度に比例する抵抗力が発生することから、振動の減衰が保証されている。しかし、対話的な操作でモデルの変形を行うことから、(2)に示すように変形操作にあたって現実的な時間で収束することが求められる。また、本手法においてはバネマスモデルをユーザー操作によって変形させることで作り出したい形状を得るのが目的であるが、その際に挙動が紙らしくない、すなわちバネの集合体のような挙動を示す場合、ユーザー操作が困難なものになってしまう。そのため、これらの条件を満足するパラメータの設定にあたって、単一のバネにおける収束の程度、およびその所要時間、挙動について実験を行った。

バネダンパモデルについては、一般的にその運動方程式を式(3-4)のように表すことができる。

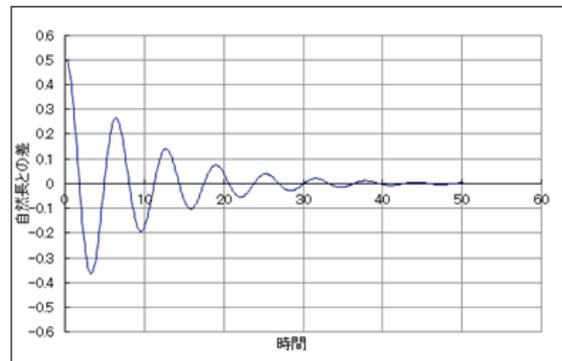
$$\frac{d^2x}{dt^2} + 2\gamma\omega_0 \frac{dx}{dt} + \omega_0^2 x = 0 \quad (3-4)$$

この式において、 γ と ω_0 の関係性から次のように振動の様子を分類することができる。また、それぞれの挙動の概要を図 3-2 に示す。

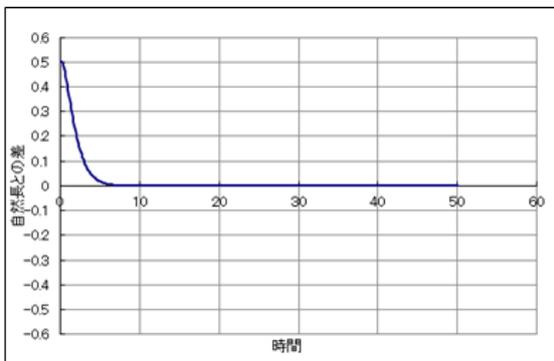
- (a). $\gamma = 0$: 単振動
- (b). $0 < \gamma < \omega_0$: 減衰振動
- (c). $\gamma = \omega_0$: 臨界減衰
- (d). $\gamma > \omega_0$: 過減衰



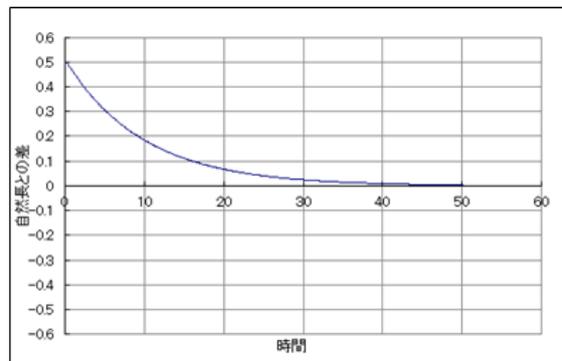
(a). 単振動の様子



(b). 減衰振動の様子



(c). 臨界減衰の様子



(d). 過減衰の様子

図 3-2 振動の種類とその様子

このうち、(a)単振動は減衰せず値が収束しないことから提案手法において適切ではない。また、(d)過減衰は収束が遅く対話的な操作に適していない。したがって、(b)減衰振動、もしくは(c)臨界振動のどちらかであり、かつ条件(2)、(3)を満たすようなパラメータを設定すれば良い。

図 3-2 に示した様子は単一のバネにおける結果であり、提案手法において用いるバネマスモデルは複数のバネが相互に接続されたものとは異なる。そこで、実際にこの両者となるようパラメータを調整し、その挙動を確認した。

(b)減衰振動において、 γ の値を 0 から ω_0 の範囲で変化させた結果、 γ が小さく単振動に近い

挙動の場合、バネの影響が大きく違和感の無い挙動についての条件(3)を満足できなかった。一方で、 γ を大きくすることでより素早く収束することを確認した。

(c)臨界減衰の場合、収束までの間バネは常に一定方向に変形しており、それによって一定方向への力が発生し続けるため、それによって隣接するバネもまた変形し、それらが積み重なることでバネマスモデル全体が常に変形し続けるような結果となった。

以上の結果から、バネ定数、ダンパ定数、および頂点の質量のパラメータは、個々のバネが減衰振動となるような範囲、かつ比較的 γ が大きい値であれば条件を満足できることを確認した。これらから、バネ定数、ダンパ定数、および頂点の質量をそれぞれ10,20,30と設定した。また、自然長は1とし、10ミリ秒毎にすべてのバネ、質点における力、速度、位置の計算を行うものとした。このパラメータにおける単一バネの時間変化の様子を図3-3に示す。自然長に対し、その長さの50%だけバネを伸ばした状態から収束までの様子を測定し、1.37秒経過時点で誤差1%未満まで収束していることを確認した。正三角格子バネマスモデルにおいて、完全に収束するのを待たずとも次の変形操作が可能なこと、自然長に対して50%以上変形させるような操作は想定されないことから、バネマスモデルにおいて求められる条件は満足できたと考えられる。

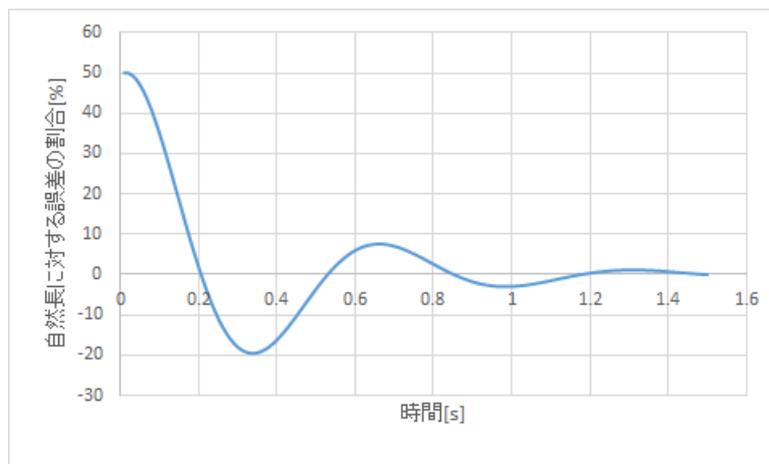


図 3-3 実装したパラメータにおける単一バネの変形の様子

3.2 バネマスモデルの変形

3.2.1 基本的な変形操作

前節で述べたバネマスモデルを計算機上に表示し、それをユーザーによる操作によって折り紙として創り出したい形状へと変形させる。マウスなどを用いた操作によって頂点の位置を移動させ、それに伴うバネの変形によって発生する力の影響を考慮した構造体全体の変形を逐次描画することで、対話的な操作を可能とする。

本研究において実装した基本操作は、次の通りである。

- (1) マウス右ボタン : ドラッグによる視点の移動
- (2) マウススクロールボタン : 任意頂点の固定, またその解除の指定
- (3) マウス左ボタン : クリックで頂点の選択, ドラッグによって選択頂点の移動
- (4) キーボード z/x キー : ズームイン/ズームアウト

上記の操作によって, 比較的頂点数の少ない単純なモデルについては, 創り出したい形状へと変形させることが可能である.

3.2.2 回転対称性に基づく変形操作

前項までの基本的操作のみでも, 比較的頂点数の少ない単純なモデルを取り扱うことは十分に可能である. しかし, 頂点数の増えた大きなモデルの場合, ユーザーが操作する頂点が極端に増加し, 操作が煩雑になるという問題点がある. そこで, 本手法で対象とする正三角格子バネマスモデルのもつ回転対称性に基づいた変形操作を提案した.

図 3-4 に示すように, 構造体全体の 1/6 にあたる正三角形領域を前項において述べた基本的操作によって変形させ, それを残りの 5/6 の領域へ対称的に反映させることで, 容易に幾何学的で見た目のよい形状を得ることが可能となる.

最初に, 1/6 ごとの正三角形領域に位置する頂点群を 6 つ, またそれらの境界にあたる頂点群を 6 つ判別するためのデータ構造を用意する. なお, 以降初めにユーザーが操作する部分にあたる最初の正三角形領域を **triangle1**, およびそれを反映させる正三角形領域を **triangle2**, またそれを反映する正三角形領域を **triangle3**… (以下同様に **triangle6** まで続く), およびそれらを総称して **triangle** と呼ぶ. また, それらの境界にあたる頂点群を同様に **axis1,axis2**…, それらを総称して **axis** と呼ぶ. このとき, **triangle1** および各 **axis** に位置する頂点群は, 最初の平坦な状態であれば幾何学的に容易に判別できる. その後, **triangle1** の各頂点から **axis1** までの最短距離のベクトルを計算し, その最近点からそのベクトルと同じだけ移動した座標に位置する頂点を探索する (図 3-5). こうして探索された **triangle1** の各頂点と対応する頂点群が **triangle2** となり, **triangle2** と **axis2** について同様の操作を行うことで **triangle3** を探索できる. これを繰り返すことで全ての **triangle** が定まる.

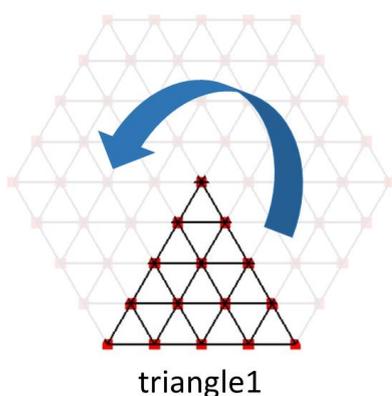


図 3-4 回転対称性に基づく設計操作

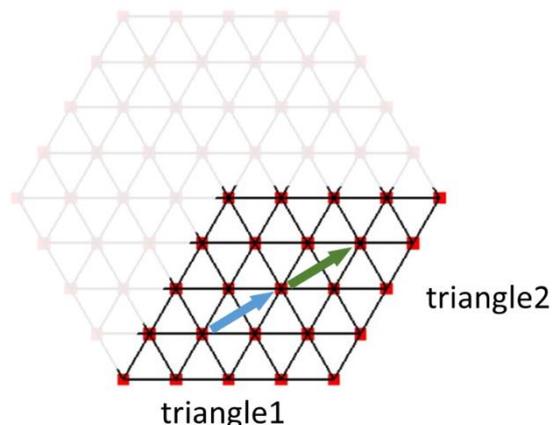


図 3-5 対応する **triangle** に応じた頂点の探索

次に、前項で述べた基本的な変形操作を用いて、triangle1 の各頂点を操作する。その後、triangle1 の各頂点の初期位置と、現在の位置とのベクトルを計算し、初期モデル（平坦な状態）と完成後モデルの triangle1 部分との差分ベクトルを求める。それを式(3-5)に示す回転行列に従って axis1 の周りに 180 度回転させることで、axis1 について triangle1 と対称な triangle2 が求まる。

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \cos\theta + n_x^2(1 - \cos\theta) & n_x n_y(1 - \cos\theta) - n_z \sin\theta & n_x n_z(1 - \cos\theta) + n_y \sin\theta \\ n_x n_y(1 - \cos\theta) + n_z \sin\theta & \cos\theta + n_y^2(1 - \cos\theta) & n_y n_z(1 - \cos\theta) - n_x \sin\theta \\ n_x n_z(1 - \cos\theta) - n_y \sin\theta & n_y n_z(1 - \cos\theta) + n_x \sin\theta & \cos\theta + n_z^2(1 - \cos\theta) \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (3-5)$$

ここで x', y', z' はそれぞれ triangle2 の初期位置ベクトルに加える移動ベクトルであり、 x, y, z は triangle1 の初期位置からの差分ベクトル、 n_x, n_y, n_z は回転の軸上の単位ベクトルであり、本手法においては各 axis 上の単位ベクトルとなる。同様の操作を triangle2 から triangle6 まで繰り返すことで、ユーザーの操作によって変形した triangle1 を各 axis に対して 180 度回転させた、幾何学的で見た目のよいモデルを得ることができる。

また、この操作によって得られる形状は、各 axis を基準に triangle1 が 180 度反転していた形状であるが、式(3-6)に示す回転行列を用いて、各 axis の位置する面を基準に鏡映反転させた形状を得ることができる。

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \cos\theta + n_x^2(1 - \cos\theta) & n_x n_y(1 - \cos\theta) - n_z \sin\theta & n_x n_z(1 - \cos\theta) + n_y \sin\theta \\ n_x n_y(1 - \cos\theta) + n_z \sin\theta & \cos\theta + n_y^2(1 - \cos\theta) & n_y n_z(1 - \cos\theta) - n_x \sin\theta \\ n_x n_z(1 - \cos\theta) - n_y \sin\theta & n_y n_z(1 - \cos\theta) + n_x \sin\theta & \cos\theta + n_z^2(1 - \cos\theta) \end{pmatrix} \begin{pmatrix} x \\ y \\ -z \end{pmatrix} \quad (3-6)$$

しかし、これらの機能の実装にしたがって axis に所属する頂点群の移動に制約を設ける必要がある。これは、図 3-6 に示すように axis に所属する頂点群は、axis を挟んで位置する triangle 間の境界面上にしか位置できない、というものである。この制約を満たすことで、紙としての構造を破綻させない、すなわち紙に切れ目を生じさせないまま、回転対称性に基づく操作を実行することが可能となる。

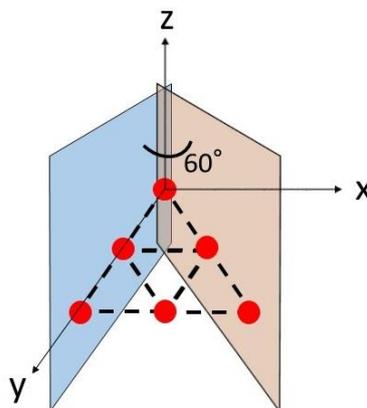


図 3-6 axis に所属する頂点群に関する移動の制約

3.3 山谷情報を持った展開図の出力

前節までの操作によって、創り出したい立体形状のバネマスモデルを取得できた。それを実際に紙で折るためには、山折り、谷折りの情報を持った展開図が必要となる。本節では、その山谷情報を持った展開図の出力について述べる。

山谷の判別については、折り線に相当するバネと、それを挟むように配置された頂点の位置関係から判別を行う。図 3-7 に示すように、あるバネを一边としそのバネを挟むように配置された 2 つの格子正三角形において、それぞれの重心をつなぐベクトル \vec{g} と、片方の格子正三角形の法線ベクトル \vec{n} の 2 つのベクトルの角度 θ を考える。図 3-7 右に示すように、 $\theta > 90^\circ$ であれば山折り、 $\theta < 90^\circ$ であれば谷折り、 $\theta = 90^\circ$ であれば平坦（折られていない状態）であるとし、山折りとなるバネを赤色、谷折りとなるバネを青色、平坦であるバネを黒色で表示する。これによって、山谷の情報を持った展開図を出力することができ、それをを用いて実際に紙で折ることが可能となる。

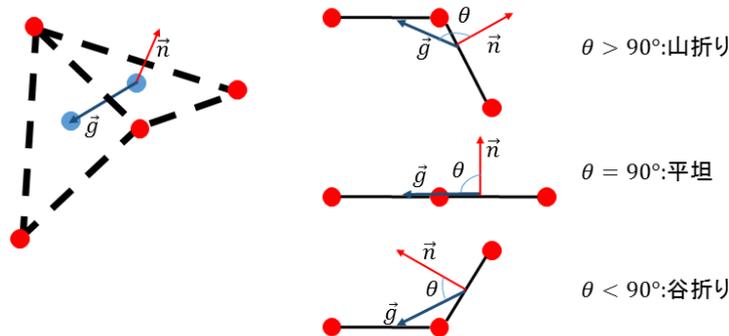


図 3-7 山谷判定の様子

また、あるバネを挟む格子三角形は 2 つ存在するが、このとき法線ベクトルを得る単位三角形の選択方法を統一しなければならない。すなわち、紙の裏表を判定する必要がある。これにあたっては、まず変形操作前の平坦な状態を考える。図 3-6 のように、対象となるバネの両端の質点 p_0, p_1 について、 p_0 から p_1 へ伸びるベクトル \vec{s} と、 p_1 から p_2, p_3 へそれぞれ伸びるベクトル $\vec{p_1p_2}, \vec{p_1p_3}$ との外積を計算すると、 $\vec{s} \times \vec{p_1p_2}, \vec{s} \times \vec{p_1p_3}$ の z 成分は必ずどちらかが正、もう片方が負の値となる。ここで、外積の z 成分が正の値を取る方の格子三角形における法線ベクトルを用いることで、バネの山谷判定が可能となる。

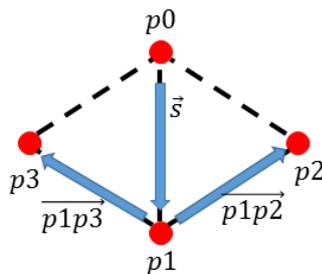


図 3-8 法線ベクトルを参照する格子三角形の選択

第4章 結果

本手法を基にしたアプリケーションを実装し，そこで生成されたモデルについて試作を行った．本章では，その結果について述べる．

4.1 実装したアプリケーション

図 4-1 に実装したアプリケーションのユーザインタフェースを示す．右のウィンドウに表示されたモデルを第 3 章で述べた操作を用いて変形させる正三角格子バネマスモデルが表示され，左のウィンドウにその他各種操作を実行するボタンを配置した．左ウィンドウ中に配置された各ボタンによる機能を表 4-1 に示す．なお，システムの実装には C++を用いて，ライブラリとして OpenGL, GLUT, GLUI を利用した．

4.2 紙を用いた作成結果

3 章で述べた提案手法を用いて複数の折り紙作品を設計し，またそれらが実際の紙でも問題なく折れることを確認した．

4.2.1 基本的な変形操作

3.2.1 項で述べた基本的な変形操作のみを用いて折り紙作品の設計，および作成を行った．そのバネマスモデル，展開図，実際に紙で折った作品を図 4-2 に示す．3.2.1 項で述べたように，比較的頂点数の少ない単純なモデルであれば，これらの機能で十分に設計が可能であることを確認した．(a), (b), (c) はそれぞれ $P_{num}=2, 3, 3$ のバネマスモデルを用いて設計したものである．また，それぞれ左図が設計したバネマスモデル，中央図がその展開図，右図がその展開図を用いて紙で作成した結果である．

4.2.2 回転対称性に基づく変形操作

3.2.2 項で述べた，回転対称性に基づく変形操作を用いて折り紙作品の設計，および作成を行った．そのバネマスモデル，展開図，実際に紙で折った作品を図 4-3 に示す．基本的な変形操作のみでの作成結果と比較して，より複雑な形状の折り紙作品を設計，および作成できていることを確認した．(a), (b) はそれぞれ $P_{num}=5, 6$ のバネマスモデルを用いて設計したものである．また，それぞれ左上段が設計したバネマスモデルを真上から描画した図，左下段が設計したバネマスモデルを真横から描画した図，右上段がその展開図，右下段がその展開図を用いて紙で作成した結果である．

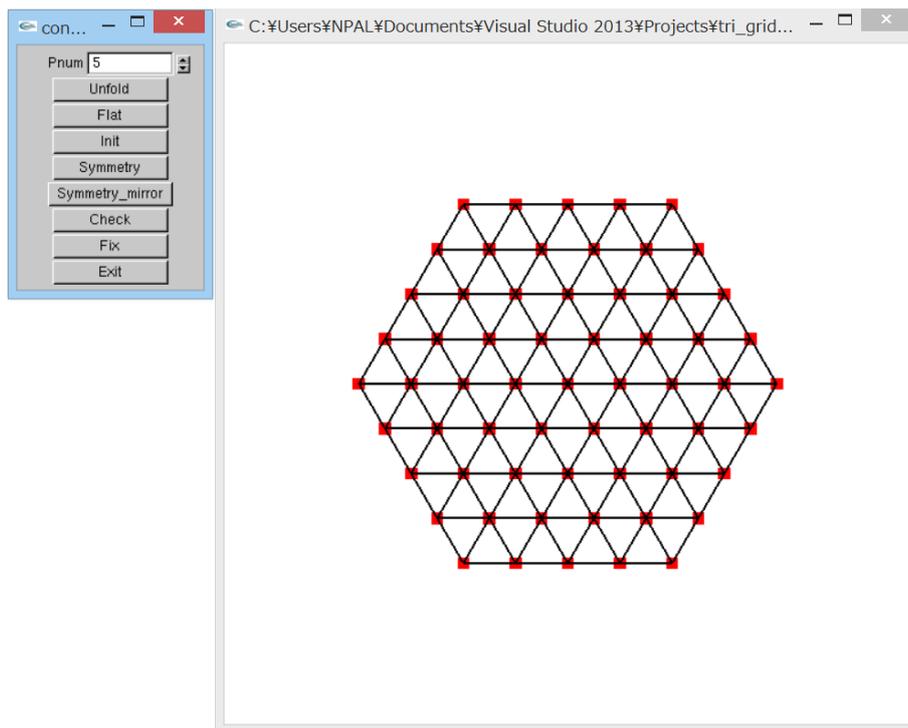
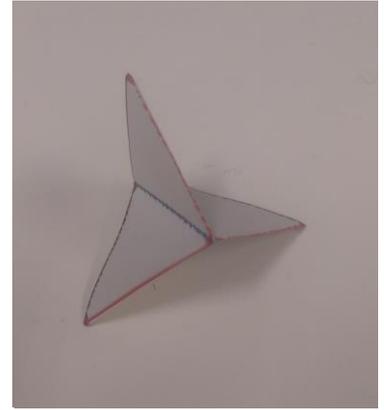
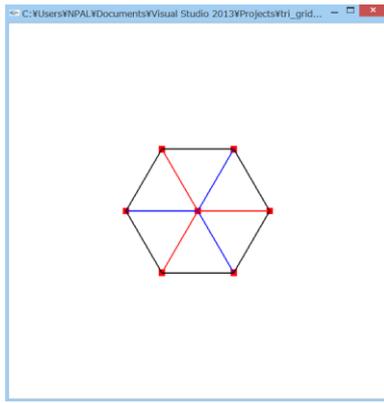
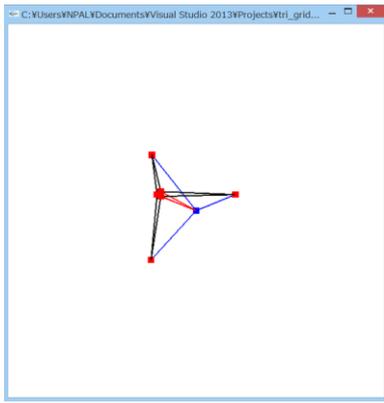


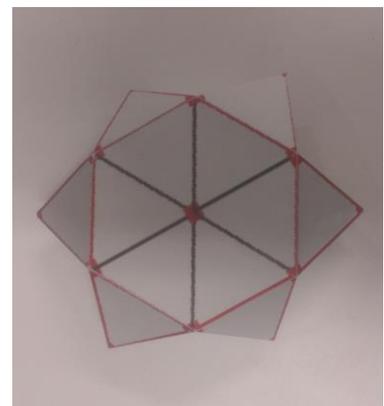
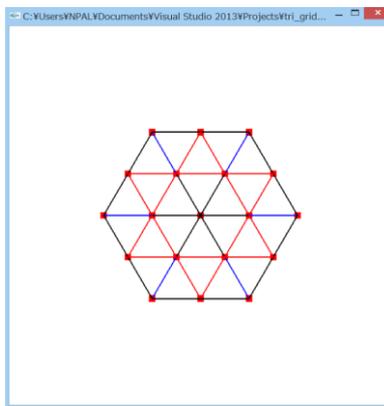
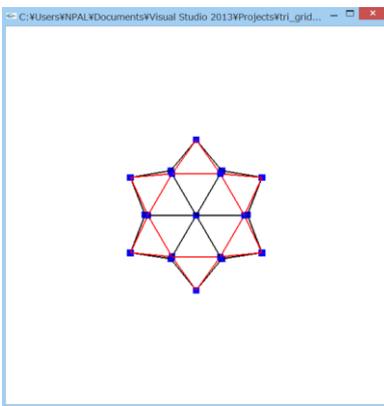
図 4-1 実装したアプリケーションのユーザインタフェース

表 4-1 実装したアプリケーションの機能一覧

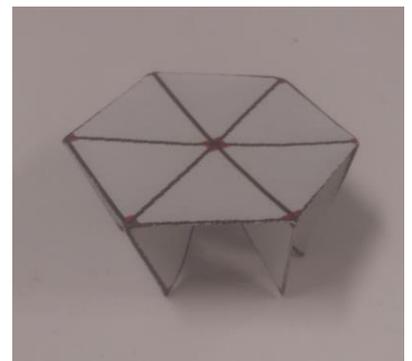
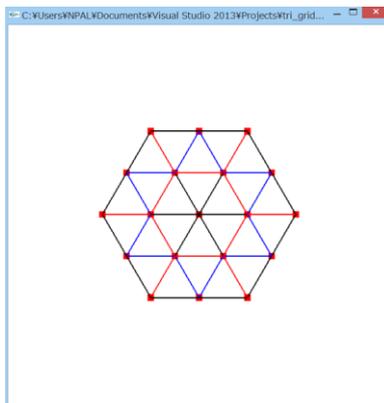
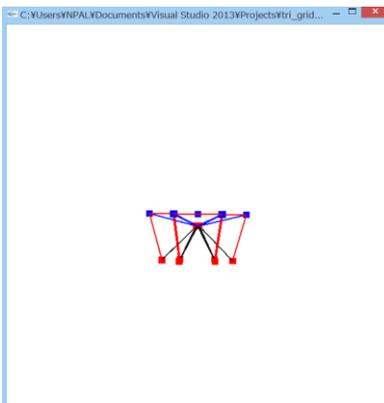
名称	機能
Pnum	バネマスモデルの大きさを指定 構造全体の六角形の一辺に含まれる頂点数を入力
Unfold	全てのバネの山谷判定を実行 山折りを赤色，谷折りを青色，それ以外を黒色で表示
Flat	山谷判定の結果を保持したまま，頂点位置を初期状態へ戻す
Init	全データを初期状態へ戻す
Symmetry	回転対称性に基づく変形操作(3.2.2 項)の実行
Symmetry_mirror	回転対称性に基づく変形操作(3.2.2 項，鏡映反転)の実行
Check	全てのバネの長さの自然長との誤差の確認 誤差 1.0%未満の場合は青色，誤差 0.1%未満の場合は赤色で表示
Fix	実行毎に全頂点を固定/非固定
Exit	アプリケーションの終了



(a) $P_{num}=2$ の設計および作成結果

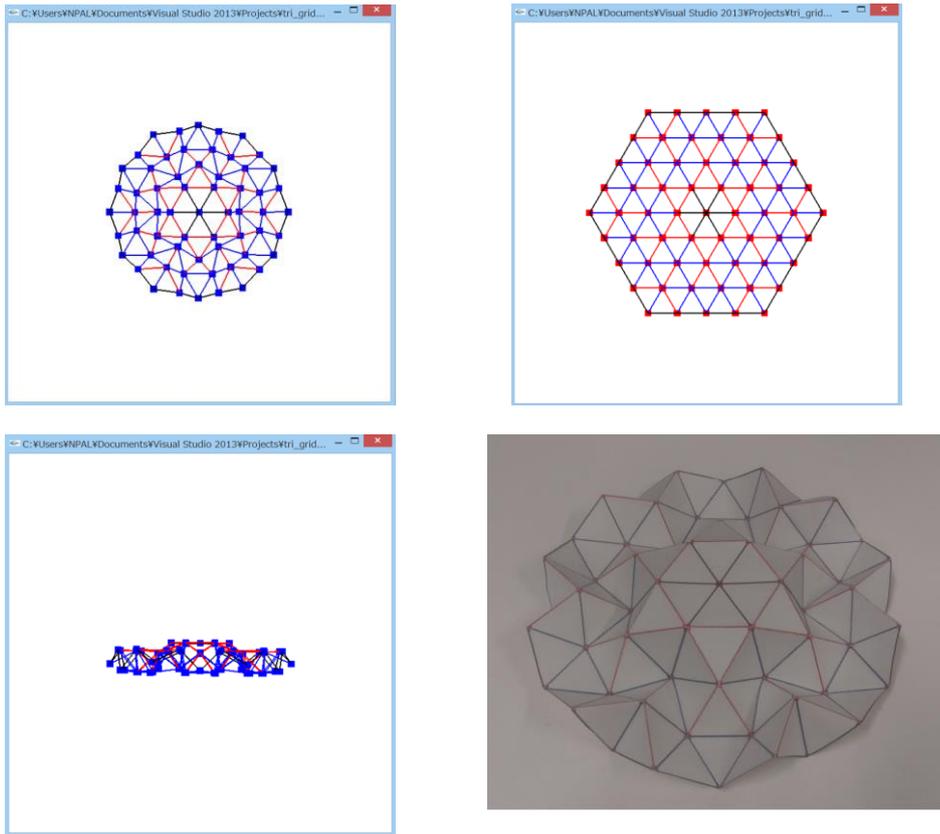


(b) $P_{num}=3$ の設計および作成結果

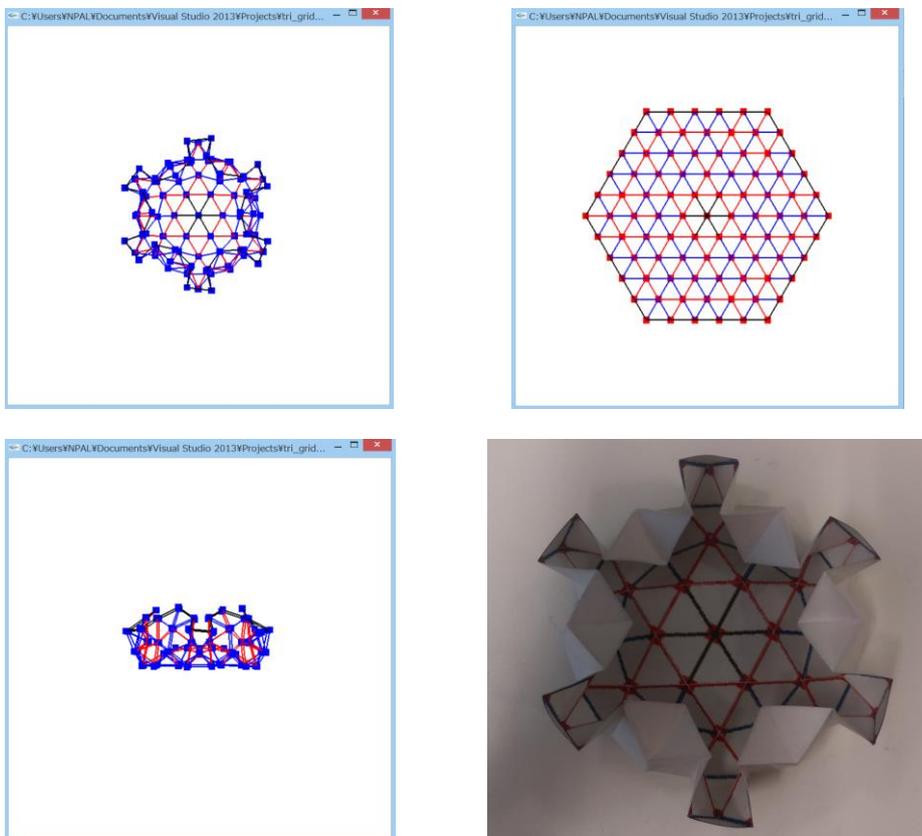


(c) $P_{num}=3$ の設計および作成結果

図 4-2 基本的な変形操作による設計および作成結果



(a) Pnum=5 の設計および作成結果



(b) Pnum=6 の設計および作成結果

図 4-3 回転対称性に基づく変形操作による設計および作成結果

4.3 数値的評価

変形させたバネマスモデルにおけるバネの収束の程度を計測した（表 4-2）。いずれも，所望のモデルへと変形させた後，約 30 秒程度経過時点での，モデルに含まれる全てのバネの長さの平均値，および長さが最大，最小であったバネの数値である。

表 4-2 作成したモデルにおけるバネの長さの計測結果

Pnum	平均値	最大値	最小値
2 (図 4-2,(a))	1.000000	1.000000	1.000000
3 (図 4-2,(b))	1.000000	1.000002	0.999998
3 (図 4-2,(c))	1.000000	1.000000	1.000000
5 (図 4-3,(a))	0.999939	1.001491	0.998654
6 (図 4-2,(b))	0.998956	1.011021	0.986982
6 (図 4-2,(b)) 約 2 分経過後	0.999904	1.004709	0.991812

頂点数の少ないモデルについては，誤差は無視できる程度まで収束していることが確認できた。頂点数の多いモデルについては誤差が 1%を超えるバネも存在するが，さらに数十秒経過した後には誤差が 0.1%未満に収まる程度に収束していることを確認できた。折り紙の素材は紙であるため，この程度の誤差であれば十分に許容できる範囲であり，提案手法の有用性を確認できた。

第5章 結論

5.1 まとめ

本論文では、立体折り紙における新規作品の設計支援について提案し、アプリケーションの出力結果から実際に新たな立体折り紙の作品の設計、作成を行った。本手法では、展開図における折り線が正三角格子上に位置する立体折り紙について、バネマスモデルを用いて紙をモデリングし、直観的な操作によって創り出したい形状を得られることを確認した。また、回転対称性に基づく設計手法によって、幾何学的で見た目のよい形状を容易に得られ、それらが確かに紙で作成できることを確認した。

5.2 今後の課題

今後の課題として、下記のことが挙げられる。(後で詳細を加筆)

- 基本変形操作の追加
アプリケーションにおいて実装した基本的な変形操作に新たな操作機能を追加することで、より簡単な完成形状の取得や、より複雑なモデルの作成が期待できる。例として、バネを挟んで対称な頂点の同時移動、複数頂点の同時並行移動、特定範囲の複製、異なる2頂点の接続などが挙げられる。また、正三角格子に乗らない折り線の追加機能などを追加することで、より自由度の高い形状設計が可能になると考えられる。
- 折り操作の角度への制限の追加
現状では折り操作の角度に制限を設けておらず、バネマスモデルにおいて衝突判定などを実装していないことから、180度を越えた折り操作が可能となってしまっている。そのため、180度を越えると山谷判定が逆転してしまうという問題点がある。各バネにおける折り操作の上限を180度と設定することで、山谷判定の逆転を解消できるだけでなく、局所的な衝突を回避することができると考えられる。しかし、アルゴリズムの計算時間などの観点から単純な衝突判定を本手法に実装することは現実的でないため、大域的な衝突を防ぐアルゴリズムを検討する必要がある。
- より頂点数の多いモデルでの試作、評価
頂点数の多いモデルにおける操作性を評価する必要がある。また、頂点数が増えるにつれバネの収束において誤差が顕著なものとなるが、実際に紙で折る際にどの程度の誤差であれば許容できるのか、紙での作成結果とバネマスモデルの構造の整合性から評価を行う必要がある。また、頂点数が増えるとモデルの視認性が極端に悪化し、構造の把握が困難になるという問題点がある。これについては、正三角格子毎にポリゴンを生成し、陰面消去を実行する機能の実装などによって解決できると考えられる。

- ユーザーテストによる機能・UI の改善

本研究の目的は立体折り紙の設計を容易に行うことであり，立体折り紙の設計経験の無いユーザーに本アプリケーションを使用してもらうことによる評価が必要である．また，バネマスモデルのパラメータについても改善の余地は十分にあるため，ユーザーテストを通じてモデル変形の容易さと収束までの時間を考慮する必要がある．

謝辞

本研究を進めるにあたり，三谷純先生，金森由博先生，遠藤結城先生には数多くのご助言やご指導をいただきました。特に三谷純先生，遠藤結城先生には，他大学・他専攻から修士として所属してきたためプログラミングにおいて苦勞していた私に，研究の方針や進め方，コーディングなど細部に至るまで丁寧なご指導をしていただきましたこと，心より感謝申し上げます。また，非数値処理アルゴリズム研究室の皆さまにも大変お世話になりました。最後に，修士としての2年のみならず，学生生活を通して様々な面から支援してくださった両親に心から感謝申し上げます。

参考文献

- [1] Toshikazu Kawasaki. On the relation between mountain-creases and valley-creases of a flat origami. In *Proceedings of the First International Meeting on Origami Science and Technology*, pp. 229–237, 1989.
- [2] Marshall Bern and Barry Hayes. The complexity of flat origami. In *Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*. Vol96, pp. 175–183, 1996.
- [3] Thomas Hull. On the mathematics of flat origamis. *Congressus Numerantium*, Vol. 100, pp. 215–224, 1994.
- [4] Robert J. Lang. *Origami Design Secrets: Mathematical Methods for an Ancient Art*. A K Perers Ltd, illustrates edition, 2003.
- [5] Robert J. Lang. Treemaker.
<http://www.langorigami.com/science/treemaker/treemaker5.php4>, 2006.
- [6] Hiroshi Shimanuki, Jien Kato, and Toyohide Watanabe. Constituting origami models from sketches. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. Vol. 1. IEEE, 2004.
- [7] Alex Bateman. Tess: origami tessellation software.
<http://www.papermosaics.co.uk/software.html>.
- [8] Tomohiro Tachi. Simulation of rigid origami. In *Origami 4: Fourth International Meeting of Origami Science, Mathematics, and Education*, pp. 175–187. A K Peters, 2009.
- [9] Tomohiro Tachi. Freeform rigid-foldable structure using bidirectionally flat-foldable planar quadrilateral mesh. In *Advances in Architectural Geometry 2010*, pp. 87–102, 2010.
- [10] 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, pp.3309-3317, 2007.
- [11] 三谷 純. 折紙展開図エディタ ORIPA
<http://mitani.cs.tsukuba.ac.jp/pukiwiki-origa/>
- [12] Naoya Tsuruta, et al. Simple Flat Origami Exploration System with Random Folds. *The 6th International Meeting on Origami in Science, Mathematics and Education*, 2014.
- [13] Yohei Yamamoto, Jun Mitani. The Number of Different Flat-foldable Crease Patterns in 45 Degree Grid System. *10th AEARU Workshop on Computer Science and Web Technology*, 2015.
- [14] Tomohiro Tachi. Origamizing polyhedral surfaces. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 16, pp. 298–311, 2010.
- [15] Y. Zhao, Yoshihiro Kanamori and J. Mitani. Geometry of axisymmetric 3D origami consisting of triangle facets. In *Proc. of the 17th International Conference on Geometry and Graphics*. 2016.

- [16] Jun Mitani. Recognition, modeling and rendering method for origami using 2d bar codes. In *Origami 4: Fourth International Meeting of Origami Science, Mathematics, and Education*, pp. 251-258. 2009,
- [17] Jun Mitani and Takeo Igarashi. Interactive design of planar curved folding by reflection. In *Pacific Conference on Computer Graphics and Applications - Short Papers*, pp. 21-23. 2011.
- [18] Shinya Miyazaki, et al. An Origami Playing Simulator in the Virtual Space. *The Journal of Visualization and Computer Animation* 7.1, pp.25-42, 1996.
- [19] John Szinger. The Foldinator Modeler and Document Generator. In *Proceedings of 3rd International Meeting of Origami Science, Mathematics, and Education*, 2001.
- [20] John Szinger, Zing Man. The Foldinator Project.
<http://www.zingman.com/origami/foldinator.html>
- [21] 古田陽介, 木本晴夫, 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. *情報処理学会論文誌*, Vol.48, No.12, pp.3658-3669, 2007.
- [22] Robert Webb. Stella: Polyhedron navigator.
<http://www.software3d.com/Stella.php>.
- [23] Naoya Tsuruta, et al. Random Realization of Polyhedral Graphs as Deltahedra. *Journal of Geometry Graphics*, Vol.19, No.2, pp.227-236, 2016.
- [24] Autodesk, Maya 8 Unlimited, 2006.
- [25] AGEIA, 2006.
- [26] HAVOK, NVIDIA, 2005.
- [27] Russell Smith, <http://ode.org/> , 2007.