

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

三次元形状を折りたたみ可能な立体に  
自動変換する手法の提案

宮本 惠未

修士 (工学)

(コンピュータサイエンス専攻)

指導教員 三谷 純

2018年 3月

## 概要

ものを平坦に折りたたむということは空間を要領よく使う上で効果的な手段である。例えば、梱包材や家具など様々な製品が折りたたみの技術を用いて開発されている。しかし、それらの設計には資材、時間のコストが必要である。これまで折りたたみ可能な立体の設計支援システムが提案されてきたが、それらによって出力される立体は単純な形であることが多く、任意形状の再現は難しい課題である。そこで先行研究によって、より自由度の高い形状を持つ折りたたみ可能な立体の設計システムが提案された。この立体は複数の多面体を接合したものであり、個々の多面体は一方向に押しつぶすようにして折りたたむことができる。先行研究では、個々の多面体の形状を決定するために、ユーザーが天頂面(折りたたみ方向に垂直に配置される凸多角形)と断面線を手動で入力せねばならず、操作が直感的でないという問題がある。そこで本論文では入力された三次元形状を折りたたみ可能な立体へと自動的に変換する手法を提案する。本手法は、参照とする立体形状のデータを入力として受け取り、それを複数のパーツに自動分割したのち、個々のパーツを折りたたみ可能な多面体に自動変換する。その際に、参照とする立体形状をどの程度の精度で近似するかを表すパラメータと、各多面体の折りたたみ方向をユーザが指定する。これにより、先行研究よりも大幅に少ない作業コストで目的を達成できる。本研究では、このような提案手法の有効性を確認するため、プロトタイプシステムの開発を行った。また、得られた立体が平坦に折りたたまれる様子をシステム上で確認できるよう、CGで可視化した。さらに、入力形状が対称性を持つ場合は、対称性を保ったまま折りたたみ可能に変換するための拡張も行った。最終的に、得られた結果から展開図の出力を行い、実際の紙で制作することで提案手法の有効性を確認した。

# 目次

第1章 序論	1
1.1 研究の背景	1
1.1.1 折りたたみ可能な立体の設計	1
1.1.2 先行研究によって提案された折りたたみ可能な立体	2
1.2 研究の目的	5
第2章 関連研究	6
2.1 ポップアップカードの設計	6
2.2 折り紙の設計に関する研究	7
2.3 剛体折りに関する研究	9
第3章 提案手法	12
3.1 折りたたみ可能な立体の構成と変数の定義	12
3.2 既存手法における折りたたみ可能な立体の構成とその設計手法	14
3.2.1 ユーザーの入力	14
3.2.2 折りたたみ可能な多面体パーツの条件	14
3.2.3 多面体パーツの接続	16
3.3 本手法における折りたたみ可能な立体の構成	16
3.3.1 ユーザーの入力	16
3.3.2 多面体パーツの制約	17
3.3.3 多面体パーツの接続	17
3.4 手法の概要	20
3.5 凸パーツへの変換	22
3.6 パラメータの指定	22
3.7 多面体パーツへの変換	25
3.7.1 処理概要	25
3.7.2 天頂面の生成	27
初期形状の生成	27
接続面と平行な辺の挿入	27
天頂面の最適化	27
3.7.3 断面線の生成	31
初期形状の生成と接続面に平行な線の挿入	31

各断面線の最適化 . . . . .	31
最適化の過程で断面線の変形を妨げる項 . . . . .	32
折りたたみ可能にするための項 . . . . .	32
凸制約項 . . . . .	34
3.7.4 隣り合う側面のパネルの衝突を回避するための修正 . . . . .	35
衝突の生じる原因 . . . . .	35
3.8 対称性を考慮した折りたたみ可能な立体の生成 . . . . .	37
3.8.1 凸パーツの分類 . . . . .	37
3.8.2 対称性を考慮した折りたたみ方向の選択 . . . . .	38
3.8.3 折りたたみ方向が対称面に垂直な場合 . . . . .	39
天頂面の生成 . . . . .	39
断面線の生成 . . . . .	40
3.8.4 折りたたみ方向が対称面に平行な場合 . . . . .	42
天頂面の生成 . . . . .	42
天頂面の最適化 . . . . .	42
断面線の生成 . . . . .	43
<b>第 4 章 結果</b> . . . . .	<b>45</b>
4.1 折りたたみ可能な立体の出力例 . . . . .	45
4.2 対称性を考慮した折りたたみ可能な立体 . . . . .	45
<b>第 5 章 考察</b> . . . . .	<b>49</b>
5.1 パラメータの変化による近似誤差と計算時間 . . . . .	49
5.2 断面線の収束 . . . . .	51
5.3 側面のパネルの衝突と断面線の修正 . . . . .	51
5.4 凸制約による影響 . . . . .	54
5.5 対称性を考慮した立体の近似精度 . . . . .	56
5.5.1 折りたたみ方向が及ぼす影響 . . . . .	56
5.5.2 既存手法との比較 . . . . .	57
5.6 今後の課題 . . . . .	59
<b>謝辞</b> . . . . .	<b>61</b>
<b>参考文献</b> . . . . .	<b>62</b>
<b>付録 A</b> . . . . .	<b>65</b>

# 目次

1.1	ペーパークラフトの作品のひとつであるペンギン爆弾 . . . . .	2
1.2	既存手法によって設計された折りたたみ可能な立体 . . . . .	3
1.3	多面体パーツの構成 . . . . .	4
1.4	提案手法によって出力した折りたたみ可能な立体 . . . . .	5
2.1	Mitani らの手法によって生成されたポップアップカードの例 . . . . .	7
2.2	Ruiz らの手法によって生成されたポップアップカードの例 . . . . .	8
2.3	Li らの手法によって生成されたポップアップカードの例 . . . . .	8
2.4	円柱構造の Tachi-Miura Polyhedron の例 . . . . .	10
2.5	厚みのある折りたたみ可能な立体 . . . . .	11
2.6	Li らの手法によって生成された折りたたみ可能な家具の例 . . . . .	11
3.1	折りたたみ可能な立体を構成する要素を示した図 . . . . .	13
3.2	多面体パーツにおける変数を示した図 . . . . .	13
3.3	断面線の折りたたみの様子 . . . . .	15
3.4	折りたたまれる際に隣り合う側面のパネルで干渉が生じる例 . . . . .	15
3.5	凹部を持つ多面体パーツの折りたたみの様子を二次元的に表した図 . . . . .	16
3.6	本手法における断面線の例と折りたたみの様子 . . . . .	18
3.7	凹部を持つ多面体パーツの折りたたみの様子 . . . . .	18
3.8	接続パネルの向きが折りたたみ可能な立体に及ぼす影響を示した図 . . . . .	19
3.9	手法の概要 . . . . .	21
3.10	折りたたみ方向が折りたたみ可能な立体に及ぼす影響を示した図 . . . . .	23
3.11	異なる折りたたみ方向によって生成された多面体パーツの例 . . . . .	24
3.12	折りたたみ方向を示した図 . . . . .	24
3.13	折りたたみ方向を示した UI の例 . . . . .	25
3.14	多面体パーツへの変換方法概要 . . . . .	26
3.15	天頂面の初期形状の生成方法を表した図 . . . . .	28
3.16	接続面と平行な辺の挿入方法 . . . . .	28
3.17	最適化に用いる仮多面体 . . . . .	29
3.18	最適化に用いる多面体の変形パラメータと変形の様子 . . . . .	30
3.19	断面線の生成方法を表した図 . . . . .	31
3.20	断面線の最適化の手順を表した図 . . . . .	33

3.21	凸制約項の計算方法を示した図	34
3.22	隣合う側面のパネル間の衝突の様子を示した図	36
3.23	隣合う側面のパネルで衝突が発生している多面体パーツの断面	36
3.24	対称性を持つ形状の凸パーツの集合と対称面 $\Pi$	38
3.25	対称性を持つ多面体パーツの折りたたみ方向における分類	39
3.26	折りたたみ方向が対称面に対して垂直な場合の仮多面体とパラメータの変更の様子	40
3.27	折りたたみ方向が対称面に対して垂直な場合の断面線の例	41
3.28	対称性を考慮した断面線の最適化における頂点の変化を表した図	41
3.29	天頂面を対称形状に変換する手順を表した図	42
3.30	折りたたみ方向が対称面に対して平行な場合の仮多面体の変形の様子	43
3.31	対称性を考慮した断面線の生成の様子	44
4.1	折りたたみ可能な立体の例	46
4.2	折りたたみ可能な立体の折りたたむ様子	46
4.3	折りたたみ可能な紙模型	47
4.4	対称性を考慮した折りたたみ可能な立体の例 1	48
4.5	対称性を考慮した折りたたみ可能な立体の例 2	48
5.1	最適化前後の断面線の比較	51
5.2	接続面に平行な線を含む断面線の最適化前後の比較	52
5.3	断面線の修正前後の比較	53
5.4	穴の開いた入力形状の例と折りたたみ可能に変換した結果	54
5.5	多面体パーツ間の衝突が生じた立体の例	55
5.6	対称性を考慮した折りたたみ可能な立体の比較	56
5.7	対称性を考慮した折りたたみ可能な立体の折りたたみ方向による形状の変化	58
5.8	既存手法と提案手法との比較	58
5.9	多面体パーツを一度に折りたたむ場合と別々に折りたたむ場合の比較	60
A.1	Armadillo の紙模型を折りたたむ様子	65
A.2	Armadillo の紙模型の展開図	66
A.3	Snowman の紙模型を折りたたむ様子	67
A.4	Snowman の紙模型の展開図	68

# 第1章 序論

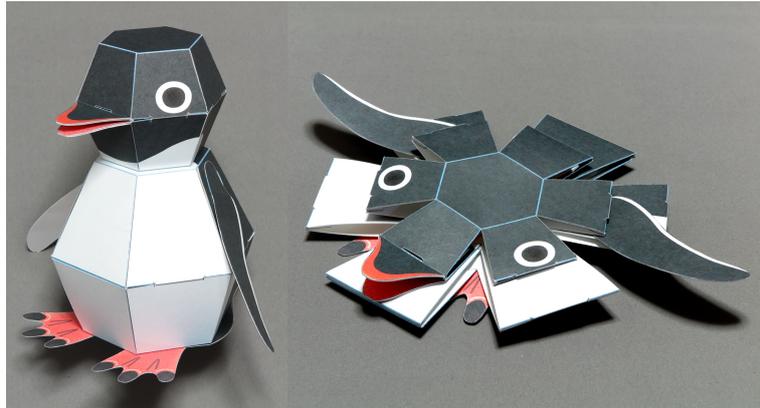
本章では，本研究を行うに到った背景を述べ，次に本論文で用いる語句を説明する．最後に研究の目的を明らかにする．

## 1.1 研究の背景

### 1.1.1 折りたたみ可能な立体の設計

ものを平坦に折りたたむということは空間を要領よく使ううえで効果的な手段である．例えば，ダンボールなどの梱包材は必要な時に簡単に組み立てることができる一方で，使わないときは平坦に押しつぶすことでスペースを有効に使うことができる．また，家具や収納ボックスなど私たちの身の回りの製品にはしばしば折りたたみの技術がみられ，生活を便利にするために用いられている．折り紙工学の分野では様々な折りの構造がロボットや工業製品に応用されている．文献 [14] の人工衛星はミウラ折りと呼ばれる折りの構造を持ち，展開・収納が容易にできるよう設計されている．

折りたたみの技術は空間の有効活用だけでなく傘や扇子，衣服のデザインなどの身近な製品や折り紙，ポップアップカードといったエンターテインメントとして用いられている (文献 [15])．例えば図 1.1 はペンギン爆弾 (出展 [33]) というペーパークラフトで，本手法で扱う折りたたみ可能な立体とよく似た構造をしている．ペンギンの形を模した左の立体は頭の上面に対して垂直に押し下げると縦方向の辺が開き，右のように平坦になる．このように，工学，日用品，玩具などものを折りたたむ技術は様々な分野で取り入れられているが，これらの設計にはプロのデザイナーによる試行錯誤を必要とする．折りたたむことを想定された立体は，折りたたむ過程で形状が変化するため自己交差が生じる可能性がある．また，立体を構成するパネルやヒンジが歪むことなく折りたたまれるよう，剛体としての制約を満たさなければならない．そのような立体の設計にはプロトタイプと製作時間が必要とされる．そのため現在提案されている折りたたみ可能な立体やその設計システムは立方体や回転対称性を持つ単純な形を対象としたものが多く，任意形状の設計は困難な問題であった．このような問題を解決するために加瀬らによってより複雑な形状をもつ，図 1.2 のような折りたたみ可能な立体の設計手法が提案された (文献 [1])．次節ではこの折りたたみ可能な立体の構成について説明する．



Copyright (c) カミカラ オンラインショップ

図 1.1: ペーパークラフトの作品のひとつであるペンギン爆弾

### 1.1.2 先行研究によって提案された折りたたみ可能な立体

はじめに、折りたたみ可能な立体の説明に用いる語句を定義しておく。これらの語句は、図 1.3 で示す。

- **折りたたみ可能な立体:** 折りたたみ可能な多面体の集合 (図 1.3 (a)). 折りたたむと平坦になる。
- **多面体パーツ:** 折りたたみ可能な立体を構成する多面体 (図 1.3 (b)). 一方向に折りたたみ可能である。
- **折りたたみ方向:** 多面体パーツを折りたたむ方向 (図 1.3 (f)). 多面体パーツはこの方向に一直線に折りたたみ可能である。また逆方向に展開が可能である。
- **天頂面:** 折りたたみ可能な立体を構成するパネルの一つで、多面体パーツの上面にあたる。折りたたみ方向に垂直な平面に乗る凸な多角形である。(図 1.3 (c))
- **側面のパネル:** 折りたたみ可能な立体を構成するパネルで、多面体パーツの上面、下面以外のパネル。天頂面の各辺ごとにヒンジで接続された一続きのパネルの集合として定義される。このヒンジは全て天頂面に平行である。折りたたまれる過程で隣り合う側面のパネルが離れる (図 1.3 (e)).
- **側面の断面線:** 折りたたみ可能な立体の側面のパネルの形状を決定する、同一平面上に乗る折れ線 (図 1.3(d)). 断面線の線分はパネル、断面線の点はヒンジに相当する。折りたたみ方向における断面線の高さは全ての断面線において等しいため、天頂面と側面の断面線から底面 (下面) が決定される。

既存手法ではユーザーが天頂面と断面線の頂点を入力し、その後折りたたみ可能となるよう各断面線の頂点位置の修正がシステムによって自動的に行われる。この手法によって折りたたみのメカニズムや三次元モデリングに関する知識を有さない人でも折りたたみ可能な立体の設計が可能となった。しかしながら既存手法では天頂面、側面の断面線という二次元情報から得られる立体形状を把握することが難しく、操作が直感的でないという問題がある。また、それらを構成する各頂点をユーザーが手動で入力するため、ユーザーの望む形状を得るまでに試行錯誤が必要であった。更に、複数の多面体パーツを組み合わせる場合、ユーザーは貼り合わせる面の適切な配置も考慮しなければならない。本手法では天頂面、側面の断面線の生成や多面体パーツの配置を自動的に行うことでこれらの問題の解決を目指す。

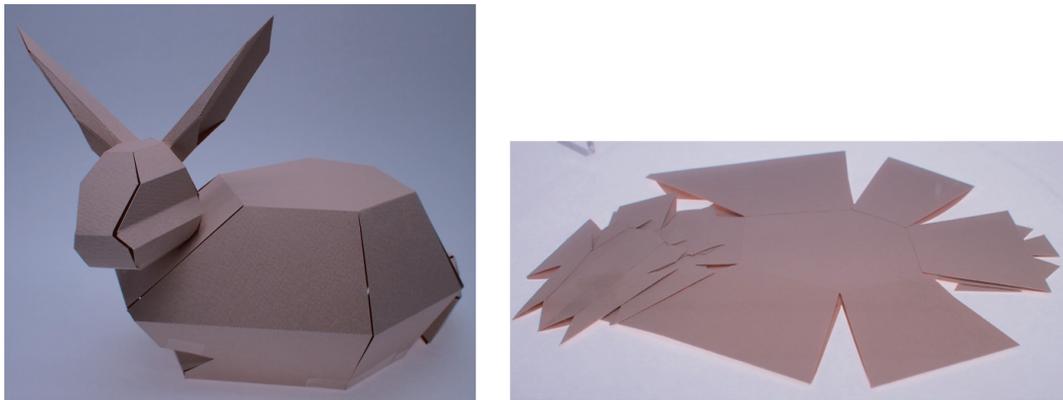


図 1.2: 既存手法によって設計された折りたたみ可能な立体. (左) 折りたたみ前. (右) 折りたたみ後. (引用: 参考文献 [1])

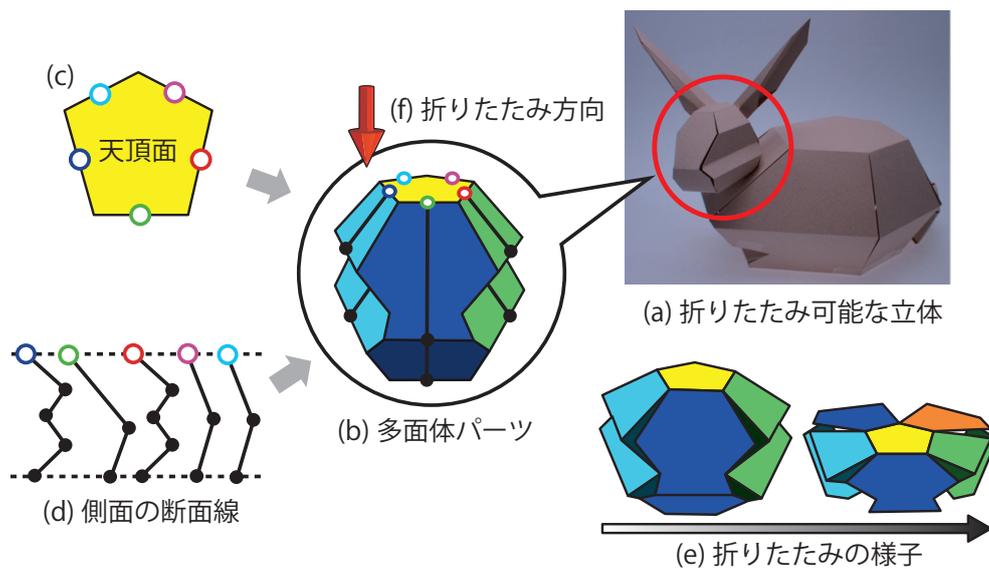


図 1.3: 多面体パーツの構成を示した図。(a) 折りたたみ可能な多面体の集合。(b) 折りたたみ可能な立体を構成する多面体。一方向に折りたたみ可能である。また逆方向に展開が可能である。(c) 折りたたみ可能な立体を構成するパネルの一つで、多面体パーツの上面にあたる。折りたたみ方向に垂直な平面に乗る凸な多角形である。(d) 折りたたみ可能な立体の側面のパネルの形状を決定する、同一平面上に乗る折れ線。(e) 多面体パーツの折りたたみの様子。折りたたまれる過程で隣り合う側面のパネルが離れる。(f) 折りたたみ方向。

## 1.2 研究の目的

本研究の目的は，入力された三次元形状を折りたたみ可能な立体へと自動的に変換する手法を提案することである．本研究で扱う折りたたみ可能な立体は，複数の多面体を接合したものであり，個々の多面体は一方向に押しつぶすようにして折りたたむことができる．本研究で提案する手法は，図 1.4 (a) のような参照とする立体形状のデータを入力として受け取り，それを複数のパーツに自動分割したのち，個々のパーツを折りたたみ可能な多面体に自動変換する．その際に，参照する立体形状をどの程度の精度で近似するかを表すパラメータと，各多面体の折りたたみ方向をユーザが指定する．これにより，先行研究よりも大幅に少ない作業コストで目的を達成できる．本研究では，提案手法の有効性を確認するため，プロトタイプシステムの開発を行った．このシステムによって，図 1.4 (b) のような折りたたみ可能な立体が得られる．また，得られた立体が平坦に折りたたまれる様子をシステム上で確認できるよう，CG で可視化した．図 1.4 (c) はその一例である．さらに，入力形状が対称性を持つ場合は，対称性を保ったまま折りたたみ可能な立体に変換するための拡張も行い，近似精度と最終形状の見映え，組み立てやすさの向上を目指した．

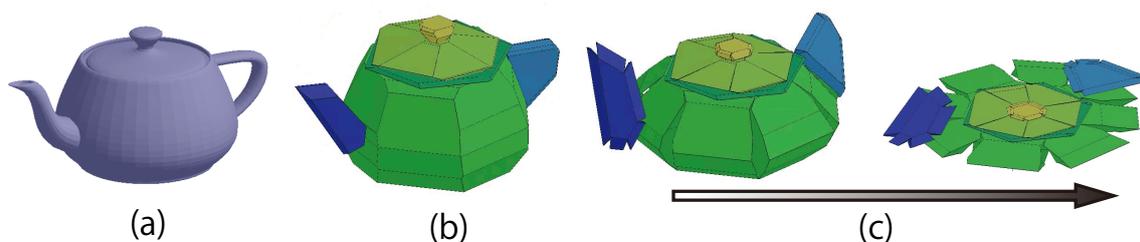


図 1.4: 提案手法によって出力した折りたたみ可能な立体．(a) 入力形状．(b) 提案手法によって出力した折りたたみ可能な立体．(c) 折りたたみの様子．

## 第2章 関連研究

本研究の目的は入力された三次元形状を折りたたみ可能な立体へと自動的に変換する手法を提案することである。折りたたみ可能な立体の設計方法や折りたたみ可能な構造は様々な分野で研究が行われてきた。本章ではそれらの研究について説明し、本研究との関係を明らかにする。

### 2.1 ポップアップカードの設計

ものを平坦に折りたたむための技術は収納や運搬などの実用的な製品に対してだけでなく、アート作品にも応用されてきた。例えば飛び出す絵本やポップアップカードは閉じているときは平坦で、紙を 180 度または 90 度開くと三次元形状が立ち上がる紙のアート作品である。これらは世界中の人に広く親しまれている一方で、設計にはプロのデザイナーによる試行錯誤が必要とされ、完成までに資材や時間コストがかかる。そこでコンピュータグラフィック(以下、CG とする)の分野では折りたたみの様子の可視化や三次元形状処理のアルゴリズムを使った設計システムが提案されてきた。

例えば対話的な設計手法として文献 [2], [3], [4] などが挙げられる。Mitani らによって提案された手法 [2] は 90 度を開いた紙を底面と背面として定義し、背面と平行なグリッド面を移動させ、グリッド面をクリックすることで三次元形状を決定する。この手順を繰り返すことで図 2.1 のようなポップアップカードを作成できる。一方 Iizuka らによって提案された手法 [3] は 180 度開くと三次元形状が展開されるポップアップカードを対象としている。折りたたみ可能なプリミティブが用意されており、ユーザーはインタラクティブにそれらを配置することでポップアップカードを設計する。また、バネマスモデルを用いて折りたたみや展開のシミュレーションが可能であり、ユーザーはコンピュータ上で折りたたみの様子を確認できる。これらの研究によって、プロトタイプの製作や設計に必要な時間が短縮された。

Li らは入力された三次元形状を 90 度を開くと立ち上がるポップアップカードへ自動的に変換する手法 [5] を提案した。Li らは折りたたみ可能性だけでなく、立ち上げた時の形状の安定性を解析し、折りたたみ可能かつ安定した形状へと変換する方法を提案している。この手法によって出力されたポップアップカードの例を図 2.2 に示す。また、Li らは折りたたみ可能なポップアップのプリミティブである V-style pop-ups を解析し、入力された三次元形状を自動的に、このプリミティブの集合で構成される立体へ自動変換する手法を提案した(文献 [6])。Ruiz らは折りたたみ可能なプリミティブを複数定義し、図 2.2 の右図のようにそれらを組み合わせることで入力形状を再現している(文献 [7])。これらの研究は展開された三次元形状の

折りたたみ可能性や再現性にのみ焦点を当てているが、アーティストの設計した飛び出す絵本は展開する物体の動きにアニメーションを取り入れたものもある。Ruizらはアニメーションのついたポップアップカードの自動生成手法を提案した(文献[8])。ユーザーは三次元メッシュモデルとスキニング、リギングといったアニメーションのデータを入力する。その後システムは自動でアニメーションのついたポップアップカードの展開図を出力する。Ruizらは紙の開閉時に生じるプリミティブの動きに着目し、それらを用いて平行移動、回転といったモーションを定義した。システムは入力されたアニメーションを近似可能なモーションを持つプリミティブをポップアップの紙面上に配置する。その後折りたたみや展開の過程で生じる衝突を回避するよう最適化によってプリミティブの配置を修正する。この手法によってアニメーションのついたポップアップカードがほぼ自動的に出力可能である。これらの研究によって生成される立体は90度または180度展開される形状に限定されている。一方、本手法で用いる多面体パーツは天頂面を一方方向に引っ張ることで立ち上がり、更に各多面体パーツは異なる折りたたみ方向を持つため、立体の構成が異なる。また、ポップアップカードの自動生成手法では折りたたみ可能なプリミティブを用いて入力形状を近似することでロバストな折りたたみ可能性を実現している一方で、曲面を再現することが難しい。例えば建物や車など平面で構成された立体はうまく近似できるが、球や動物のように滑らかな面を持つ立体は再現することが難しい。本手法ではプリミティブを当てはめるのではなく、各多面体パーツの折りたたみ可能性を保ったまま、できるだけ参照モデルに近い形状となるよう計算するというアプローチをとることでより詳細な表現を可能とする。また、本手法で扱う立体は折りたたまれる過程で側面の辺が切開するため、閉じた多面体であるにも関わらず平坦に折りたたむことが可能である。

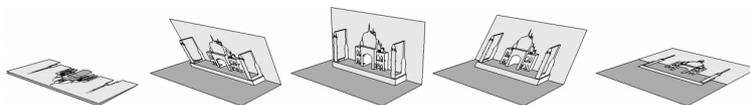


図 2.1: Mitani らの手法によって生成されたポップアップカードの展開の様子。(引用: 参考文献 [2])

## 2.2 折り紙の設計に関する研究

折り紙は紙を折りたたんで様々な形を表現する日本の伝統的な遊戯であり、世界中で広く親しまれている。折り紙というと多くの日本人は正方形の紙を用いてつくる鶴やカブトといったものを思い浮かべるだろう。しかし、近年では数学的な側面から研究が行われるようになり、より複雑な形を設計するためのシステムが提案されるようになった(文献[9], [10], [11])。これらはいずれも一枚の紙によって表現可能な三次元形状の設計と、展開図の出力が可能である。

このような設計システムによって新しい折り紙作品が創作されるようになった。しかし、そ

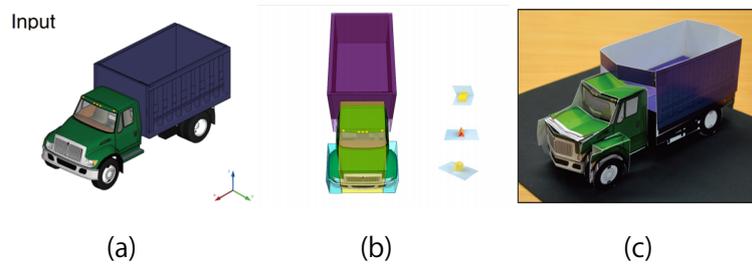


図 2.2: Ruiz らの手法によって生成されたポップアップカードの例. (a) 入力形状. (b) 折りたたみ可能なプリミティブの割り当ての様子. (c) 出力されたポップアップカード. (引用: 参考文献 [7])

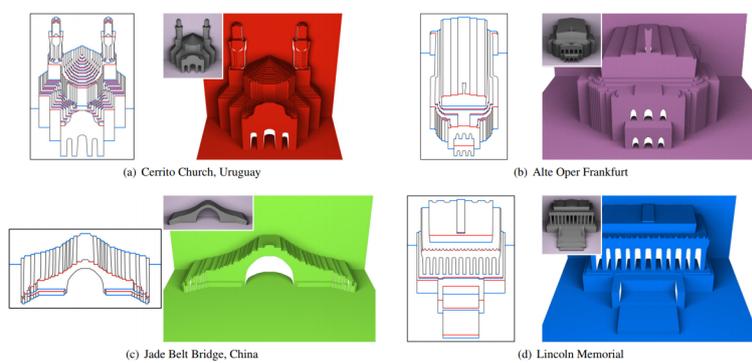


図 2.3: Li らの手法によって生成されたポップアップカードの例. (引用: 参考文献 [5])

の折りたたむ手順を示すための方法は折り紙の形状変移を図で描いた折り図が主流であった。そこで文献 [12] はバネマスモデルを用いて紙を表現し、対話的に折り紙を操作するための手法を提案した。

このようにコンピュータを用いることで折り紙の表現の幅が広がり、それに伴い様々な分野で折り紙の研究が行われるようになった。これらの手法は主に一枚の紙から三次元形状を表現、設計するということを目標としているが、本手法では入力された形状を平坦に折りたたみ可能な形に変換することが目的であるため、これらの手法を直接応用することが難しい。

## 2.3 剛体折りに関する研究

剛体折り可能な立体とは剛体のパネルと山谷の折れ線によって構成され、パネルが変形することなく折りたたむことができる立体である。本研究で扱う立体は剛体パネルとヒンジによって構成されるものとし、折りたたみ可能であるとは剛体折り可能であることを意味する。本節ではこれまで提案されてきた剛体折り可能な立体やその設計手法について述べる。

近年ではCGを用いた設計システムの発展により、剛体折りの構造やその設計手法が工学や数学の分野でも研究がされるようになった。また、剛体折り可能な立体はさまざまな製品に応用することが可能である。例えば、[13] は紙袋が剛体パネルとヒンジで構成されると考えた時の折りたたみ可能性について議論している。人工衛星のソーラーパネル(文献 [14])は剛体折り可能な形状の1つであるミウラ折り(文献 [16])を元に設計された。ミウラ折りは規則的な山折りと谷折りの展開図を持つ1自由度の折り構造であり、すばやい展開や折りたたみが可能である。工学分野だけでなく、服飾の分野でも剛体折りを用いたデザインが取り入れられており、文献 [15] は金属のパネルを素材としたケープを提案している。このような規則的な折り構造はテッセレーションと呼ばれる。剛体折り可能な立体にはしばしばこのテッセレーションが用いられる。例えば図 2.4 は Tachi らによって提案された剛体折り可能な立体であり、規則的な折り構造を持つ(文献 [17])。その他にも様々な剛体折り可能なテッセレーションのパターンやその設計手法が提案されている(文献 [18], [19])。また、設計や構造の提案だけでなく、入力された展開図から剛体折りの様子をアニメーションで示すためのシミュレータが提案されている(文献 [20], [21])。

これらの剛体折り可能な立体は紙のような薄い板を材料としている場合が多い。しかし剛体パネルが厚みのある素材の場合、干渉が発生し折りたたむことができない。厚みを考慮した折りたたみ可能な立体の設計手法としては、文献 [22], [23] が挙げられる。これらの研究は剛体折りの展開図を厚みのある素材を用いた場合でも交差を生じず折りたたみ可能となるような展開図へと変換することを目的としている。図 2.5 はその一例である。テッセレーションを用いた剛体折り構造は展開や折りたたみが容易に可能であるが、一方で規則的な折りパターンによって構成されるため、意図した形状を設計することが難しい。

より複雑な構造を持つ剛体折り可能な立体の設計手法としては、入力された家具の三次元モデルを折りたたみ可能な形状へ自動的に変換する手法を提案した、文献 [24],[25] が挙げられる。ユーザーは三次元形状と折りたたむ方向を入力する。システムは入力モデルを複数の

ユニットへと分解し，ユニット間の衝突関係をあらわすグラフを構築する．それらの情報から衝突の発生しない折りたたみ方や折りたたむ順番を決定する．この手法によって図 2.6 のように家具を折りたたみ可能に変換することが可能である．Shao らの手法 (文献 [27]) や Koo らの手法 (文献 [26]) はユーザーが入力した二次元のスケッチから直方体の集合で近似した折りたたみ可能な三次元モデルを出力する手法を提案している．これらの手法は直方体の集合によって構成される形状に限定されるため，形状の自由度が小さいという問題点がある．本手法で扱うような閉じた立体を設計することができない．

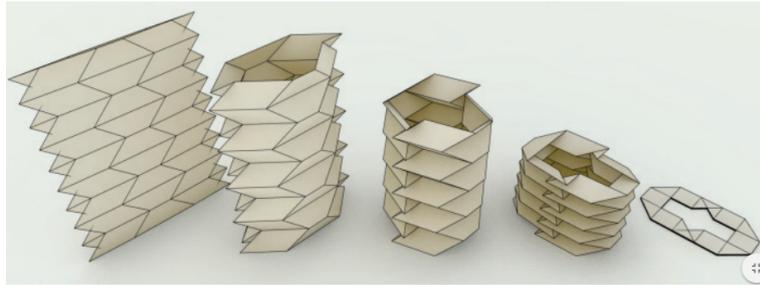


図 2.4: 円柱構造の Tachi-Miura Polyhedron の例. (引用: 参考文献 [17])

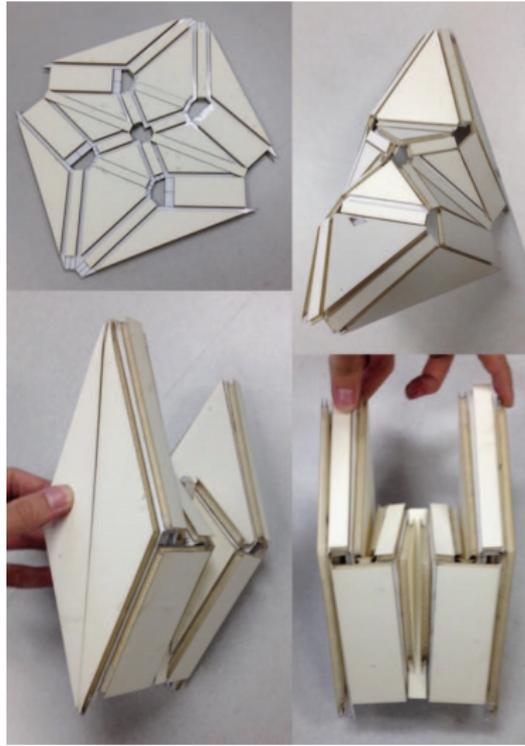


図 2.5: 厚みのある折りたたみ可能な立体. (引用: 参考文献 [23])

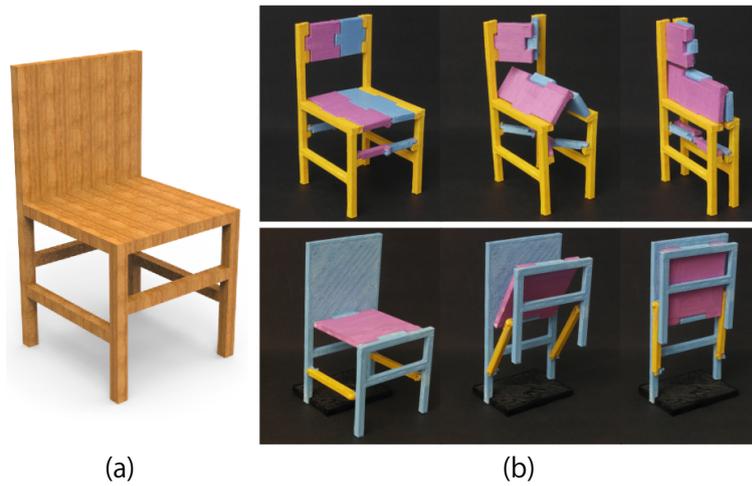


図 2.6: Li らの手法によって生成された折りたたみ可能な家具の例. (a) 入力形状. (b) 出力された折りたたみ可能な家具の折りたたみの様子. (引用: 参考文献 [24])

## 第3章 提案手法

本章では折りたたみ可能な立体を出力する手法について詳細を述べる。はじめに、本手法で用いる折りたたみ可能な立体の構成を述べ、変数の定義を行う。次に先行研究によって提案された折りたたみ可能な立体の詳細と、本手法における折りたたみ可能な立体の違いを明らかにする。次に折りたたみ可能な多面体パーツの生成方法を述べる。最後に対称性を考慮した折りたたみ可能な立体の生成方法について述べる。

### 3.1 折りたたみ可能な立体の構成と変数の定義

図 3.1 は本手法を説明する際に用いる構成要素の名称をまとめたものである。左図の色分けされた凸形状の多面体は凸パーツといい、入力された三次元モデルを分割し、凸多角形に変換したものである。凸パーツ間の面は接続面といい、分割したメッシュの境界上に位置する頂点の最小二乗平面である。右の色分けされた多面体は多面体パーツといい、凸パーツを折りたたみ可能に変換したものである。この多面体パーツを各面で貼り合わせた立体が折りたたみ可能な立体となる。

多面体パーツの一例を図 3.2 に示す。多面体パーツは互いに平行な天頂面と底面、そして天頂面の各辺に対応する断面線によってその形状が決定される。各断面線に対応する辺に対して平行な方向へ無限遠まで押し出し、水平に隣接するパネルで切り取ることで折りたたみ方向に平行に連結した側面のパネルが形成される。側面のパネルは折りたたみの過程で外側に折りたたまれる。天頂面は折りたたむ方向に垂直で、水平に隣接する側面のパネル同士の衝突を避けるため凸形状でなければならない。底面の形状は側面のパネルを切り取ることで一意に決定される。天頂面の頂点の数を  $N$ 、そして天頂面の各辺を  $E_j (j = 0, 1, \dots, N - 1)$  とする。そして各側面の断面を  $\Pi_j$  とする。 $\Pi$  は辺  $E_j$  の中点を通り、折りたたみ方向に対して平行である。断面線  $P_j$  は頂点  $\mathbf{p}_i^j (i = 0, 1, 2, \dots, M_j - 1)$  によって構成される。ここで  $M_j$  は断面線  $P_j$  に含まれる頂点の数である。また、断面線  $P_j$  は断面  $\Pi_j$  上の  $uv$  平面に定義され、その頂点は  $\mathbf{p}_i^j = (u_i^j, v_i^j)$  と表される。 $\mathbf{p}_0^j$  と  $\mathbf{p}_{M_j-1}^j$  はそれぞれ天頂面、底面と接続している。 $\mathbf{p}_0^j$  は  $v$  軸上のみを動くものとし、 $\mathbf{p}_{M_j-1}^j$  は  $u$  軸上に固定されているものとする。

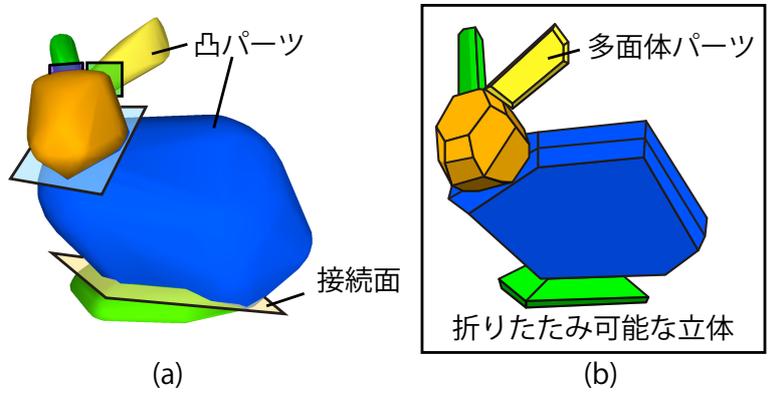


図 3.1: 折りたたみ可能な立体を構成する要素を示した図. (a) 入力形状を分割し, 凸パーツへと変換した様子. 凸パーツ間の面は接続面である. (b) 折りたたみ可能な立体の例. 多面体パーツによって構成される.

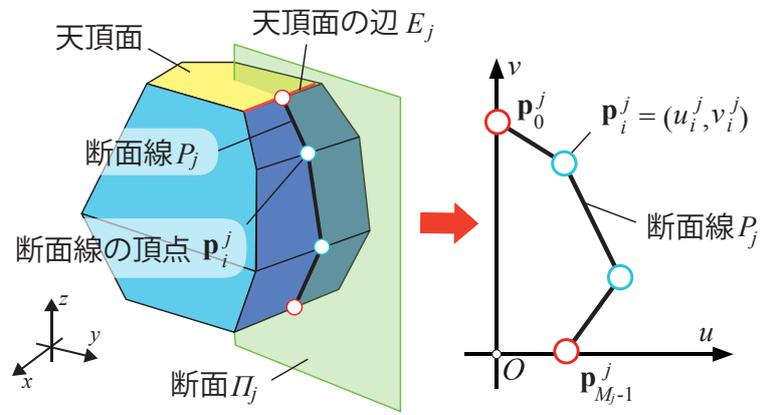


図 3.2: 多面体パーツにおける変数を示した図. 天頂面の各辺を  $E_j (j = 0, 1, 2 \dots N - 1)$  とし, 各辺  $E_j$  に対応する断面線を  $P_j (j = 0, 1, 2 \dots N - 1)$  とする. ここで  $N$  は断面線の頂点数である. 断面線  $P_j$  の頂点数を  $M_j$  とした時, 頂点は  $\mathbf{p}_i^j (i = 0, 1, 2 \dots M_j - 1)$  とする. 断面線  $P_j$  は  $E_j$  に垂直な断面  $\Pi_j$  上に定義され,  $u, v$  座標を用いて  $\mathbf{p}_i^j = (u_i^j, v_i^j)$  と表す.

## 3.2 既存手法における折りたたみ可能な立体の構成とその設計手法

本手法は加瀬らによって提案された折りたたみ可能な立体の設計手法 (文献 [1]) に基づいている。はじめに加瀬らによって提案された折りたたみ可能な立体の詳細と、その設計手法について説明する。

### 3.2.1 ユーザーの入力

ユーザーはスクリーン上をクリックすることで天頂面と側面の断面線の頂点を一つずつ入力する。折りたたみ方向が  $z$  軸と平行であるとしたとき、天頂面は  $x, y$  平面上に反時計回りに描かれる。側面を決定する断面線は、天頂面の辺の数  $N$  だけ定義され、ユーザーは  $N$  個の断面線の頂点を入力する。天頂面と底面を互いに平行にするため、すべての断面線の始点  $\mathbf{p}_0^j$  の  $v$  座標  $v_0^j$  は等しく終点は  $u$  軸に固定されている。これはユーザーが入力した 1 つ目の断面線を基準に決定される。

### 3.2.2 折りたたみ可能な多面体パーツの条件

第 3.1 節で述べたように、折りたたみ可能な立体は複数の多面体パーツによって構成され、多面体パーツは天頂面と側面の断面線によってその形状が決定される。断面線は図 3.3 のように隣接辺との連続性を保ちつつ、その長さが変化しないように折りたたまれ、折りたたみ後には  $u$  軸に対して平行になる。各頂点位置での折れ角は 0 度または 180 度になるため、折りたたみ方は  $2^{M-1}$  通りある。しかし、折りたたみ可能であるためには全ての断面線が以下の式を満たさなければならない。

$$u'_i \geq u_i, \quad (3.1)$$

$$\epsilon = (u_{M-1} - u_0) - \sum_i \delta_i l_i \approx 0, \quad (3.2)$$

ここで式 3.1 の  $u_i$  と  $u'_i$  は折りたたみ前後の頂点  $\mathbf{p}_i$  における  $u$  座標である。式 3.1 が全ての頂点について満たされていれば、折りたたみの過程で水平に隣りあう側面のパネルが干渉せずに折りたたまれる。この条件を満たさない場合、図 3.4 のように折りたたみの過程で頂点が内側に移動するため、隣接するパネルとの衝突が生じる。式 3.2 の  $l_i (i = 0, 1, 2, \dots, M-2)$  は頂点  $\mathbf{p}_i$  と  $\mathbf{p}_{i+1}$  をつなぐ線分  $S_i$  の長さを表す。 $\epsilon$  は折りたたみ前後での  $\mathbf{p}_0$  の  $u$  座標の差分を表す。これを以降 *gap* と呼ぶ。 $\delta_i$  は折りたたみ後に辺  $E_i$  のなすパネルの法線が天頂面側を向いている時に 1、底面側を向いている時に -1 をとるものとする。折りたたみの過程で始点  $\mathbf{p}_0^j$  と終点  $\mathbf{p}_{M_j-1}^j$  の  $u$  座標が一定であるとした時、 $\epsilon = 0$  であれば、折りたたみの前後で各線分の長さが一定であるため、歪みが生じることなく折りたたみ可能であるといえる。しかし、 $\epsilon = 0$  を満たす断面線が存在することは稀である。そこで既存手法では許容誤差  $\tau$  を定義し  $\epsilon \leq \tau$  となるような折りたたみ方の選択と最適化による断面線の修正を行っていた。

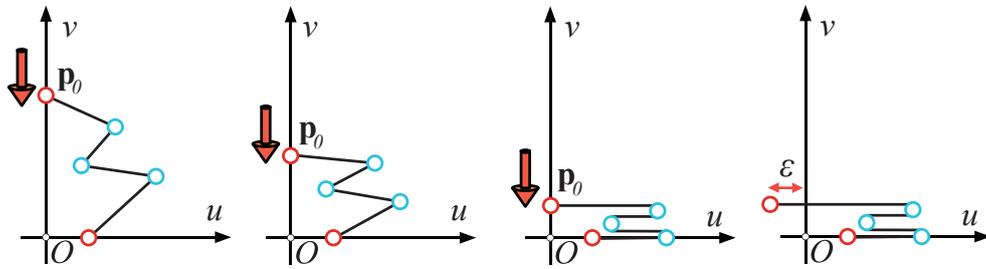


図 3.3: 断面線の折りたたみの様子. 右は  $\epsilon > 0$  となるため, 折りたたむことができない.

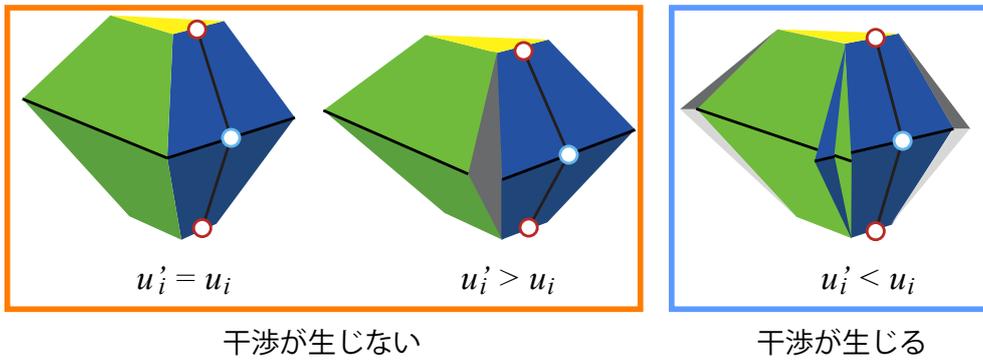


図 3.4: 折りたたまれる際に干渉が生じる例

### 3.2.3 多面体パーツの接続

多面体パーツの形状は天頂面と側面の断面線に依存し、自由度が低い。例えば二股に分かれるような形は再現できない。そこで既存手法では複数の多面体パーツをパネルを貼り合わせることで接続し、より複雑な形状を再現している。接続はユーザーがシステムとの対話的な操作によって行う。また、多面体パーツの回転、等比拡大、平行移動が可能である。

多面体パーツを複数組み合わせることで複雑な形状を表現することが可能であるが、隣接するパーツ間で干渉が発生する場合がある。特に図 3.5 の緑の領域で多面体パーツを接続するとパーツ同士が干渉する確率が高い。そこで干渉を防ぐため、折りたたみ後に多面体パーツ同士が上下に重なるよう配置する。したがって図 3.5 で示すように、各多面体パーツは折りたたみ後に天頂面側、または底面側から見えるパネル以外では接続しない。図 3.5 の青い線、赤い線はそれぞれ折りたたみ後に天頂面側から見えるパネル、底面側から見えるパネルであり、緑の線分は天頂面、底面両方の側から見えないパネルである。

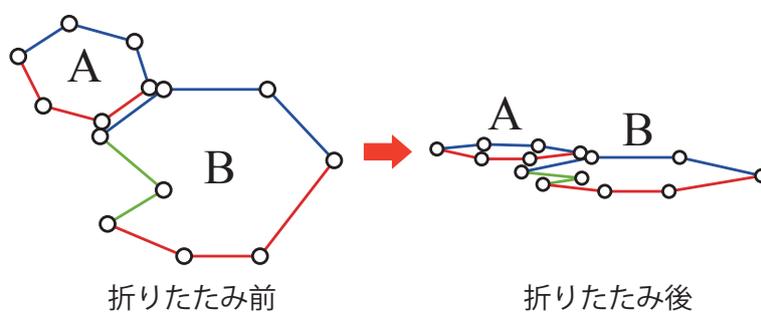


図 3.5: 凹部を持つ多面体パーツの折りたたみの様子を二次元的に表した図

## 3.3 本手法における折りたたみ可能な立体の構成

本手法で用いる折りたたみ可能な立体は既存手法と同様に天頂面と側面の断面線から生成された多面体パーツを複数組み合わせることで最終的な形状を得る。しかし、本研究の目的はユーザーが入力した参照モデルを折りたたみ可能な立体に自動的に変換することであり、既存手法とは入力方法や多面体パーツの接続方法が異なる。本節では既存手法と異なる点について明らかにする。

### 3.3.1 ユーザーの入力

本手法では天頂面、断面線はシステムによって生成されるため、天頂面、断面線の頂点情報を入力する必要はない。そのかわり折りたたみ可能に変換したい三次元メッシュモデル、分割後の各パーツの折りたたむ方向、多面体パーツの近似精度を調節するパラメータを入力す

る。この近似精度を調節するパラメータとは天頂面の頂点の最小数  $N_{min}$  と断面線の頂点の最大数  $M_{max}$  である。

### 3.3.2 多面体パーツの制約

手動で設計を行う既存手法と異なり、本手法は折りたたみ可能性と形状の再現性を満たしつつ、自動的に折りたたみ可能な立体を生成しなければならない。そのため形状の自由度が高いと衝突の生じない折りたたみ可能な形状の探索が難しくなる。そこで、本手法では折りたたみ可能な立体の形状に制約を加えることで、この問題の解決を図る。本節ではこの制約について説明する。

本手法で扱う側面の断面線には谷折りがなく、ただ一つの山折りを持つものとする。そしてこの山折りとなる頂点のことを折り頂点と呼ぶ。それ以外の頂点は折りたたまれると平坦になる。また、折りたたみ前の断面線が成す多角形は凸形状であるものとする。このような断面線の制約を凸制約と呼ぶ。本手法の対象とする断面線を図 3.6 に示す。この制約によって得られるメリットは以下である。

1. 始点、終点を除くすべての頂点は折りたたみの過程で  $u$  座標の正の方向に動くため、式 3.1 が常に満たされる。したがって、断面線の折りたたみ可能性を考慮する場合、式 3.2 のみ留意すればよい。
2. 本手法における多面体パーツは既存手法と異なり凸形状である。そのため、複数の多面体パーツを組み合わせる際、図 3.5 のような凹部を持つ多面体パーツにおける干渉の問題を考慮しなくてよい。
3. 既存手法における折りたたみ方は頂点を山折りにするか、谷折りにするか組み合わせであるため、 $2^{M-1}$  通りであったが、今回の制約によって折りたたみ方を  $M-2$  通りに削減することが可能となった。

また、この制約によって多面体パーツを折りたたむことが容易になる。例えば図 3.7 のような凹部を持つ多面体パーツを天頂面に対して垂直に押しつぶすと、凹部の側面のパネルは内側に動き、隣り合うパネルと衝突する場合がある。そのような衝突を避けるためには凹部のパネルを手で外側に引き出しながら折りたたまなければならない。複数の多面体パーツを組み合わせた場合、このような折りたたみはとても面倒な作業である。

### 3.3.3 多面体パーツの接続

本手法では入力形状の分割と多面体パーツへの変換、そしてパーツ間の接続はシステムによって行われる。そのためユーザーが接続パネルの配置を考慮する必要がない。しかしながら各多面体の接続パネルの配置や向きは最終的に得られる全体形状に大きく影響するため適切に配置しなければならない。例えば図 3.8 の青い多面体パーツの姿勢は (b) と (c) で大きく異

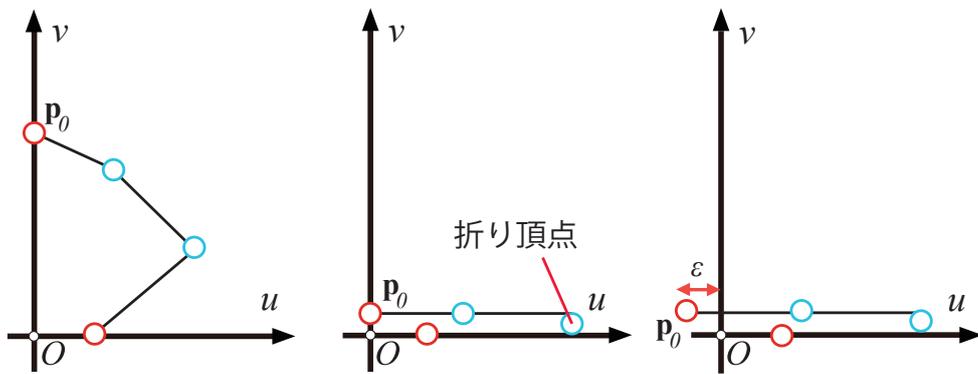


図 3.6: 本手法における断面線の例と折りたたみの様子. 右は  $\epsilon > 0$  となるため, 折りたたむことができない断面線.

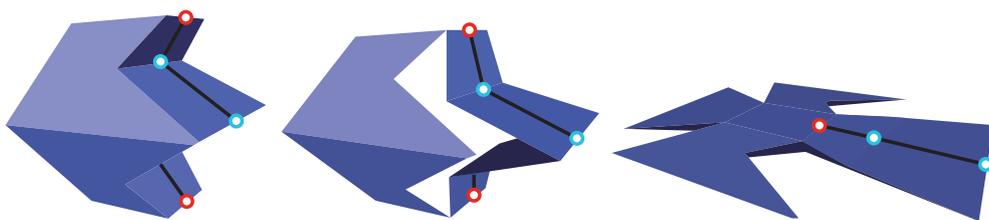


図 3.7: 凹部を持つ多面体パーツの折りたたみの様子

なる。これは接続パネルの向きが影響している。つまり多面体パーツを生成するときは折りたたみ可能性や近似精度だけでなく、適切な接続パネルの配置についても考慮しなければならない。そこで各多面体を分割した境界に接続パネルの向きの基準となる面、接続面を配置する。多面体パーツを生成する際は側面のパネルがこの接続面と平行になるよう天頂面、側面の断面線を生成する。接続パネルの向きは天頂面の辺と対応する断面線の線分によって決定されるため、接続面と平行な線を天頂面と断面線に挿入する。その方法についてはそれぞれ第 3.7.2 節，第 3.7.3 節で説明する。

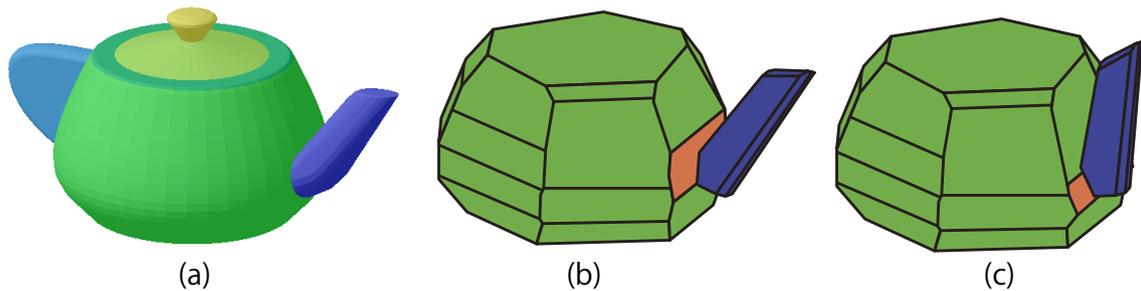


図 3.8: 接続パネルの向きが折りたたみ可能な立体に及ぼす影響を示した図。(a) 入力形状。(b) 接続パネルの面の向きが接続面と等しく、適切な場合。(c) 接続パネルの向きが接続面と異なり、不適切な場合。

### 3.4 手法の概要

本節では提案システムが入力形状を折りたたみ可能な立体に変換するまでの一連の処理について説明する。図 3.9 で示されているように、本手法は 4 つのステップによって構成されている。

1. 入力された三次元メッシュモデルを自動的に分割し凸パーツに変換する。(第 3.5 節)
2. ユーザーが各凸パーツの折りたたむ方向と近似精度を調節するパラメータを指定する。(第 3.6 節)
3. 天頂面と側面の断面線をシステムが生成し、凸パーツを多面体パーツへと変換する。(第 3.7 節)
4. すべての多面体パーツを接続し、折りたたみ可能な立体の三次元モデルデータを出力する。

本手法では折りたたみ方向の選択と各多面体パーツの近似精度の設定以外はすべてシステムが自動で処理を行う。以降では折りたたみ可能な立体を出力するまでの各ステップについて詳細を述べる。

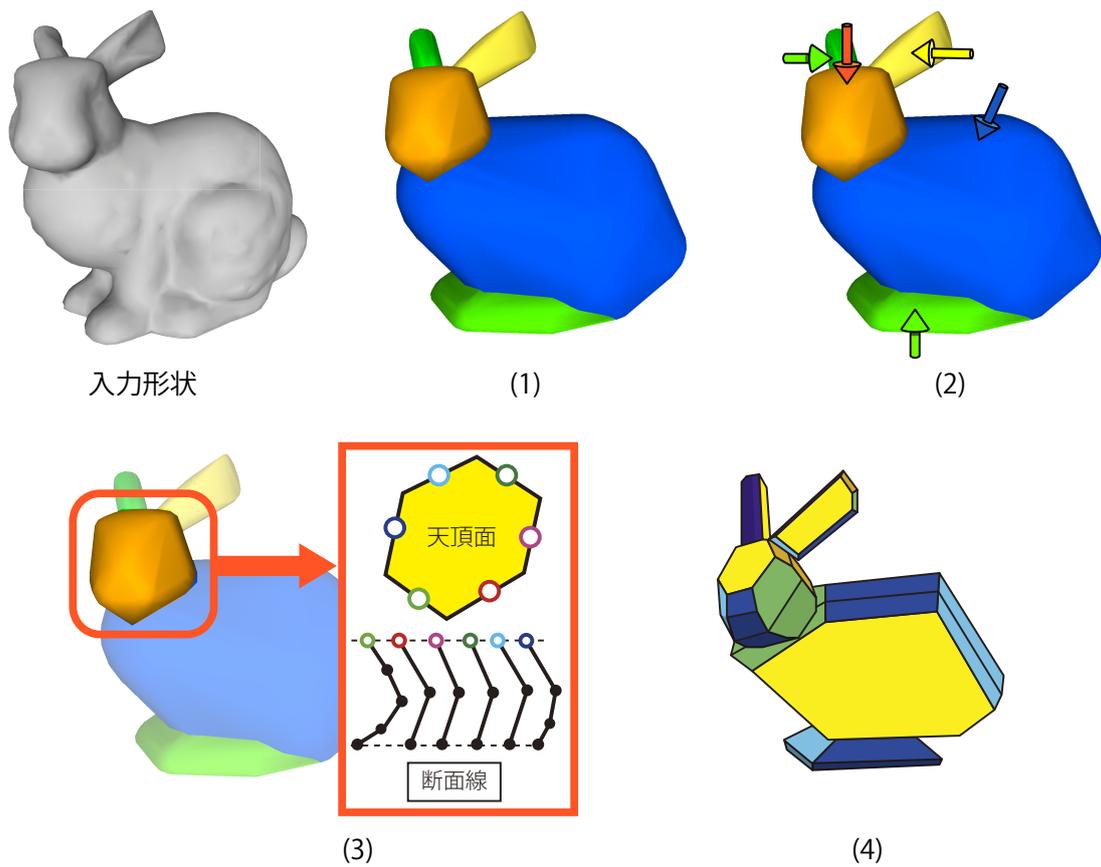


図 3.9: 手法概要を示した図. (1) 入力された三次元メッシュモデルを自動的に分割し凸形状に変換. (2) ユーザが各凸パーツの折りたたむ方向と近似精度を調節するパラメータを指定. (3) 天頂面と側面の断面線を自動的に生成し, 凸パーツを多面体パーツへと変換. (4) すべての多面体パーツを接続し, 折りたたみ可能な立体の三次元モデルデータを出力.

### 3.5 凸パーツへの変換

本手法では複数の多面体パーツを組み合わせることで複雑な形状を再現する。そのため入力形状の分割方法によって最終的な形状が大きく左右される。しかしユーザーにとってどのように分割すればよりよい結果が得られるかを考慮し、分割する領域を手作業で指定することは負担となる。そこで分割を自動で行うこととした。本手法では入力形状の特徴を保ったまま領域分割を行う Shapira らの手法 [28] を用いて入力形状の分割を行った。この手法は C++ で実装された形状処理のライブラリである CGAL(引用 [31]) に実装されており、容易に導入が可能で、計算も高速である。分割後、分割されたメッシュの境界に位置する頂点の最小二乗平面を計算する。この最小二乗平面が接続面となる。

第 3.3 節で述べたように本手法で用いる断面線は山折りとなるただ一つの頂点を持つ。結果として得られる多面体パーツは凸形状であり、天頂面は凸多角形でなければならない。そのような天頂面、断面線を生成するためにはその元となる各パーツをあらかじめ凸形状に変換しておいたほうが都合がよい。そこで本手法では分割された各パーツを Quickhull アルゴリズム (文献 [30]) によって凸形状へと変換する。

### 3.6 パラメータの指定

ユーザーは出力される立体を操作するパラメータとして、多面体パーツの折りたたむ方向と天頂面、断面線の頂点数を選択できる。特に折りたたみ方向は多面体パーツの形状を左右する重要なパラメータである。天頂面は折りたたみ方向に垂直な凸多角形であり、折りたたみ方向によってその形状が決定される。そして側面の断面線は天頂面の各辺に対応するため、折りたたみ方向は断面線の形状にも影響を与えることとなる。具体的な例を用いて説明する。図 3.10 は (a) の凸パーツから異なる折りたたみ方向を入力して生成された 2 種類の折りたたみ可能な立体を示している。上段の赤い多面体パーツは上下方向が折りたたみ方向として選択されたため、頭の上部に天頂面が配置されている。そして両耳は天頂面と接続している。一方で下段の折りたたみ方向は前後方向である。この場合、天頂面は最前面の多角形である。両耳の多面体パーツは側面のパネルと接続されている。結果として上段と下段では両耳の姿勢が異なっている。つまり折りたたみ方向は各多面体パーツの形状を左右し、最終的に得られる形状全体に影響を与える。

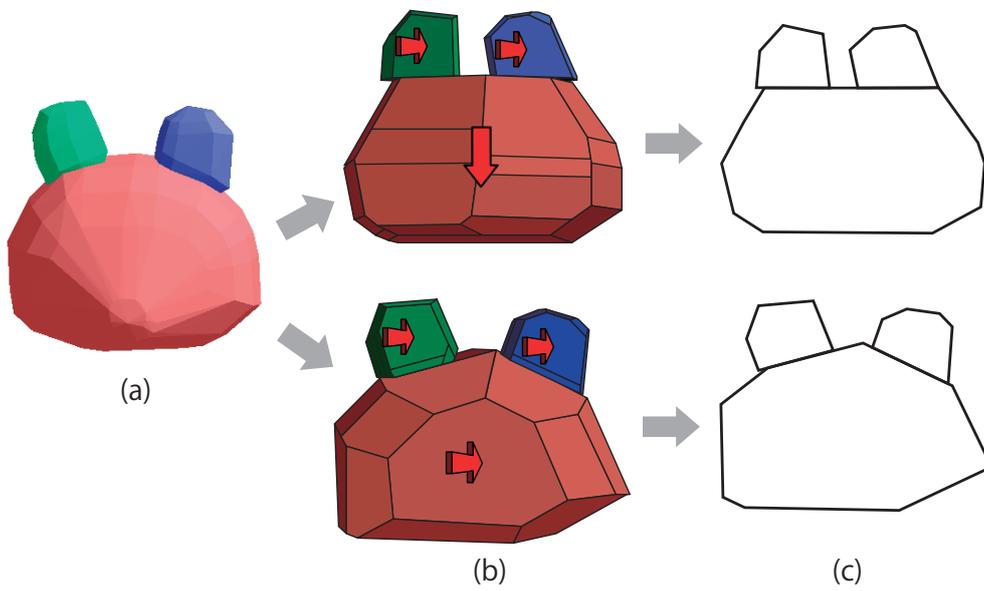


図 3.10: 折りたたみ方向が折りたたみ可能な立体に及ぼす影響を示した図. (a) 凸パーツ. (b) 多面体パーツを接続した図. 赤い矢印は折りたたみ方向を表す. (c) 多面体パーツの輪郭線.

入力形状を適切に近似する折りたたみ方向を選択するためには、多面体パーツの近似精度、接続後の各多面体パーツの姿勢、折りたたみの過程における衝突の有無などを考慮せねばならず、とても複雑な課題である。そのため、本手法では折りたたみ方向はユーザーが決定することとしたが、最適な折りたたみ方向の選択はユーザーにとっても難しい課題であり、試行錯誤が必要とされる。そこで、折りたたみ方向をある程度限定するため、ユーザーは図 3.12 のように凸パーツの第一主成分に対し、垂直な方向のみを選択可能とした。ここで赤い矢印は第一主成分、青い矢印が選択可能な折りたたみ方向である。側面のパネルは折りたたみ後に外側に広がるため、図 3.11 のように折りたたみ方向における側面のパネルの辺が長くなると、多面体パーツ間で干渉の生じる確率が高くなる。第一主成分に垂直な方向を選択することで、折りたたみ方向における側面のパネルの辺が短くなるため、結果として干渉の生じる確率が低くなる。

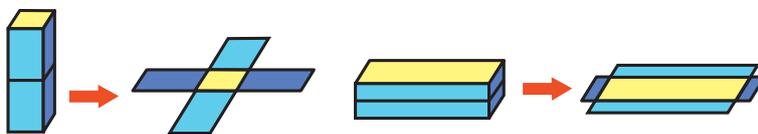


図 3.11: 異なる折りたたみ方向によって生成された多面体パーツの例。黄色いパネルは天頂面を表す。折りたたみ方向における側面のパネルの辺が長いほど展開するパネルの面積も大きくなる

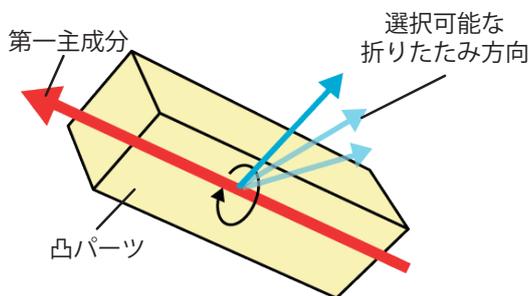


図 3.12: 折りたたみ方向を示した図。黄色い立体は凸パーツ，赤い矢印は第一主成分，青い矢印は選択可能な折りたたみ方向を表す。

既存手法では天頂面、断面線の頂点数はユーザーが決定していたが、本手法では多面体パーツの生成を自動で行うため、適切な天頂面、断面線の頂点数は未知である。頂点数を増やすほど詳細な表現が可能だが、一方で計算時間の増加を引き起こし、折りたたみにくい複雑な形状となる可能性がある。そこで頂点数の選択はユーザーにゆだねるものとした。ユーザーは天頂面の頂点数の最小数  $N_{min}$  と断面線の頂点数の最大数  $M_{max}$  を変更できる。システム上では図 3.13 のように、折りたたみ方向は凸パーツの重心を通る矢印として描かれている。ユーザーはマウスピッキングによって凸パーツを選択でき、スライダーを動かすと凸パー

ツの第一主成分を軸として折りたたみ方向のベクトルが回転する。  $N_{min}$ ,  $M_{max}$  は各パーツのリストに名前と共に表示されたスピンボックスによって変更することができる。

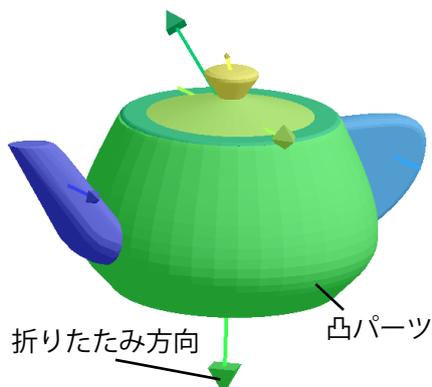


図 3.13: 折りたたみ方向を示した UI の例。凸パーツを横切る矢印はシステム上に表示される折りたたみ方向である。

## 3.7 多面体パーツへの変換

本節では提案システムにおいて一番重要な多面体パーツへの変換方法について説明する。この処理は前節で述べたパラメータの入力後、すべて自動で行われる。多面体パーツは天頂面、側面の断面線によって形状が決定されるが、天頂面は凸形状であり、接続面に平行な辺を持つ。一方断面線は折りたたみ可能でなければならず、その形状や考慮しなければならない制約は天頂面と異なる。したがって同時に最適な形状を計算することが難しい。そこで天頂面と断面線の生成処理は別々に行う。また、側面の断面線は天頂面の各辺に対応しており、その位置や結果的に得られる側面のパネルの向きは天頂面の形状に影響される。そのため本手法では天頂面の生成処理後に天頂面の各辺に対応する側面の断面線を生成する。

### 3.7.1 処理概要

多面体パーツへの変換処理は以下の手順で行う。また、この様子を図 3.14 に示す。

1. 天頂面の生成 (第 3.7.2 節)
  - 初期形状の生成
  - 接続面と平行な辺の挿入
  - 最適化
2. 側面の断面線の生成 (第 3.7.3 節)

- 初期形状の生成
- 接続面と平行な辺の挿入
- 最適化

3. 隣接する側面のパネル間で衝突が発生する場合，断面線を修正 (第 3.7.4 節)

4. 天頂面と側面の断面線から多面体パーツを生成

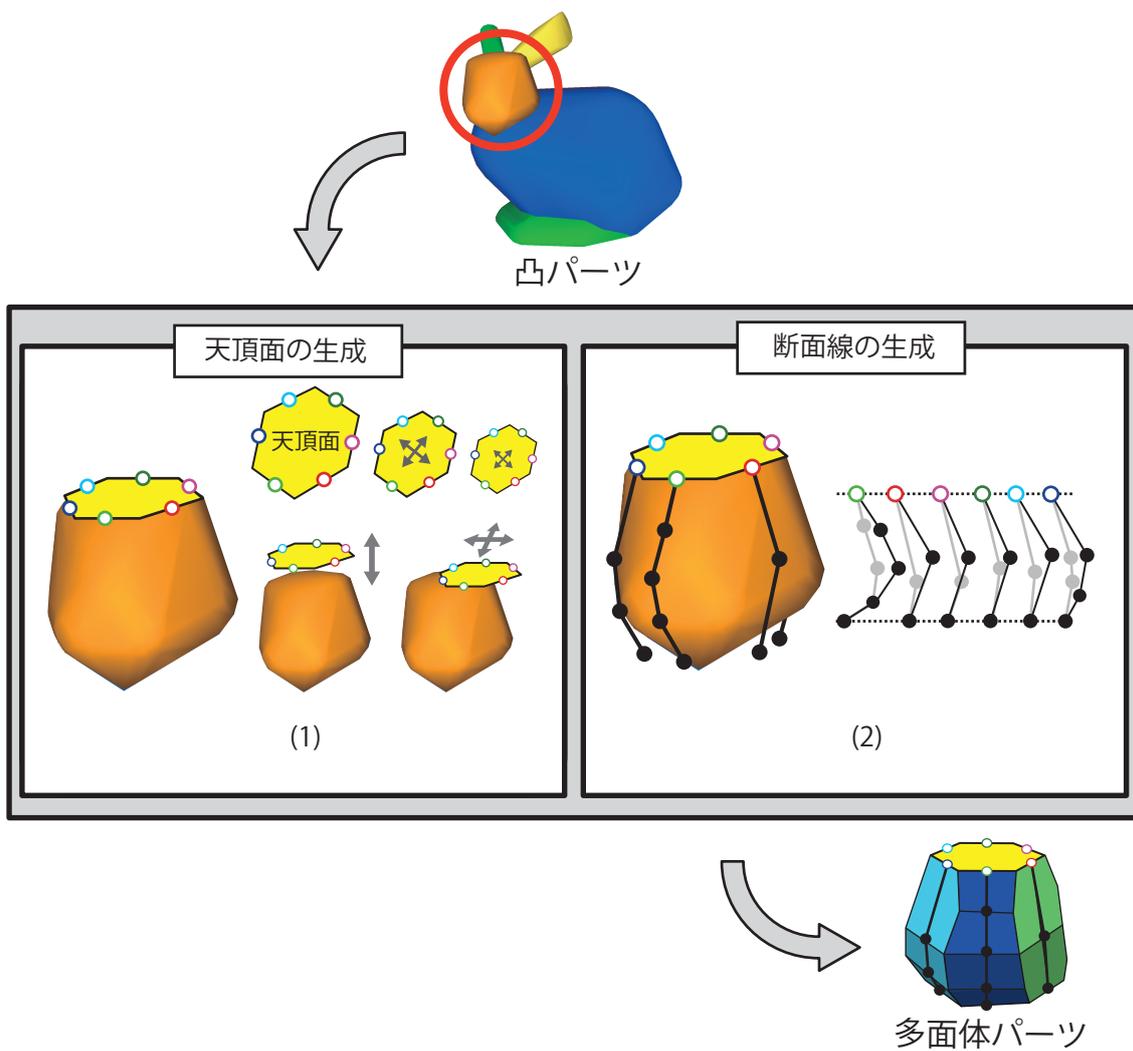


図 3.14: 多面体パーツへの変換方法概要. (1) 天頂面を生成し，大きさと位置を最適化する. (2) 断面線をひとつずつ生成し，最適化する.

### 3.7.2 天頂面の生成

指定された折りたたみ方向に従い，システムはまず天頂面の初期形状を生成する．その後天頂面の大きさと位置を最適化する．本節では天頂面の生成処理について詳細を述べる．

#### 初期形状の生成

天頂面の初期形状は図 3.15 で示すように以下の手順で生成される．

1. 折りたたみ方向に垂直な平面に凸パーツの頂点を投影し，二次元空間の点群へと変換する．
2. 二次元の点群から凸多角形を生成する．
3. ユーザーの指定したパラメータ  $N_{min}$  まで凸多角形の頂点を削減する．
4. 底面から  $0.9 \times Height$  の位置の断面に内接するよう天頂面の大きさを調節する．

ここで， $Height$  は折りたたみ方向における凸パーツの長さである．凸多角形の頂点の削減には Dyken らによって提案されたアルゴリズム [29] を利用した．天頂面の単純化後に断面線の挿入を行うため，天頂面の頂点数は単純化後より多くなり，接続面の数によって増数が決まる．そこで天頂面の頂点数を操作するパラメータを天頂面の頂点の最小数  $N_{min}$  としている．

#### 接続面と平行な辺の挿入

第 3.3.3 節で述べたように，適切な接続パネルを生成するためには接続面に平行な辺を天頂面に挿入する必要がある．本手法では図 3.16 のように，接続面で天頂面を切り取ることで実現する．ただし，天頂面が接続面にほぼ平行な場合，天頂面または底面を接続パネルとする．また，天頂面のある辺が接続面にほぼ平行な場合，その辺に対応する側面のパネルを接続パネルとする．

#### 天頂面の最適化

多面体パーツの形状は側面のパネルの影響を受けるため，天頂面の最適化では側面のパネルの形状も考慮しなければならない．そこで，天頂面と 3 つの頂点を持つ断面線を用いて大まかに入力形状を近似した多面体を生成する．以後，これを仮多面体と呼ぶ (図 3.17 (c))．この仮多面体を凸パーツに近似するよう最適化を用いて変形する．仮多面体の形状が最も凸パーツに近い形状になった時の天頂面を採用する．

図 3.17 で示すように，多面体を構成する断面線は 3 つの頂点と 2 つの辺を持つ．各断面線  $P_j$  の頂点を  $\mathbf{p}_i^j = (w_i^j, v_i^j)$  ( $i = 0, 1, 2$ ) とする．そして各断面線の  $v$  座標は等しく，多角形  $\Lambda_0$ ,  $\Lambda_1$ ,  $\Lambda_2$  が形成される．これは天頂面と同じ形で，それぞれ  $\mathbf{p}_0^j$ ,  $\mathbf{p}_1^j$ ,  $\mathbf{p}_2^j$  を含む．

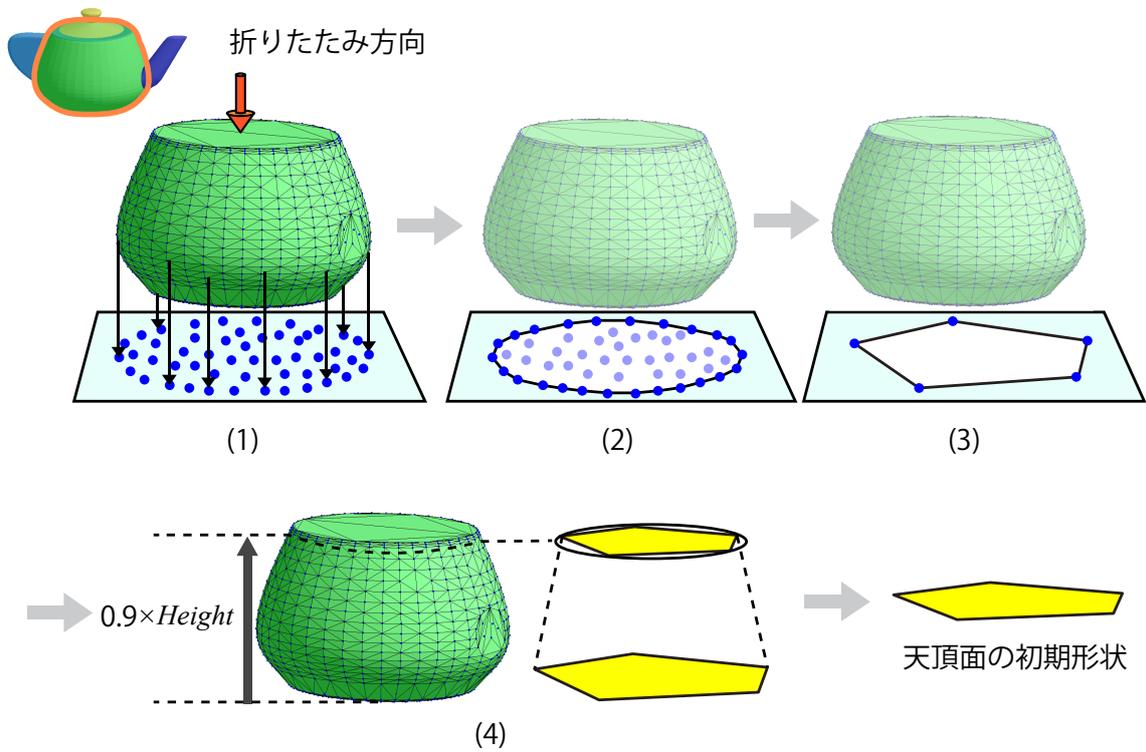


図 3.15: 天頂面の初期形状の生成方法を表した図. (1) 三次元メッシュモデルの頂点を天頂面と平行な二次元平面に投影. (2) 投影した二次元の頂点から凸包を生成する. (3) 頂点数を  $N_{min}$  まで削減する. (4) 底面から  $0.9 \times Height$  の位置の断面に内接するよう天頂面の大きさを調節する.

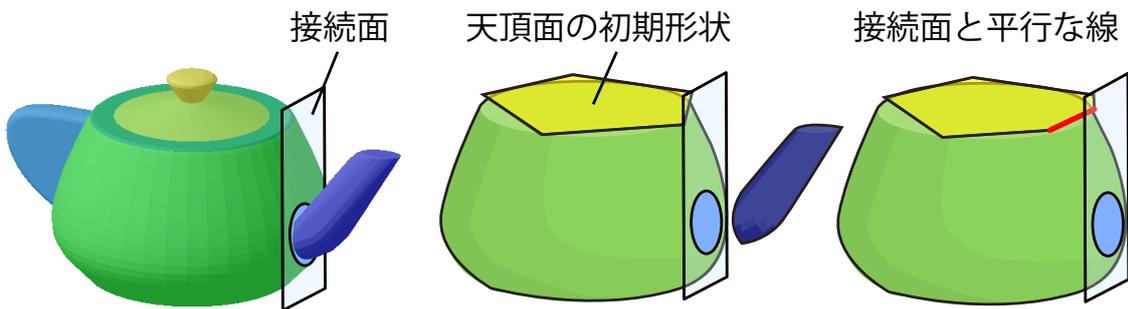


図 3.16: 接続面と平行な辺の挿入方法

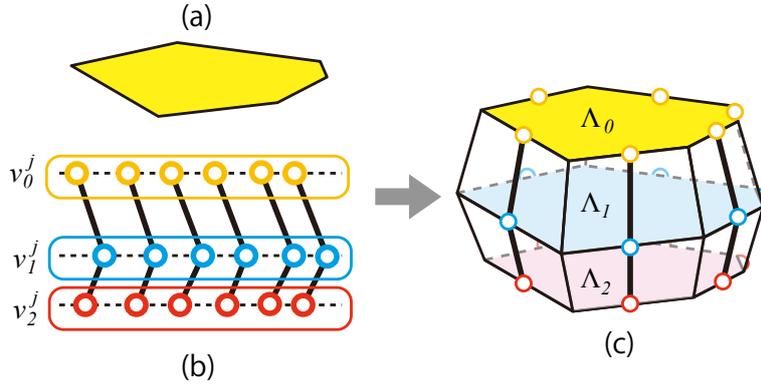


図 3.17: (a) 最適化前の天頂面. (b) 三つの頂点を持つ断面線. (c) 仮多面体.

折りたたみ方向を  $z$  軸の負の方向とした時 多面体を変形するパラメータは以下の 8 つである.

1. 各面  $\Lambda_0, \Lambda_1, \Lambda_2$  のスケール値  $s_0, s_1, s_2$  (3 変数)
2. 各面  $\Lambda_0, \Lambda_1, \Lambda_2$  の  $v$  座標  $v_0, v_1, v_2$  (3 変数)
3. 多面体の重心の  $x, y$  座標 (2 変数)

パラメータによる多面体の変化の様子を図 3.18 に示す. ここで, 多面体の重心の  $z$  座標は  $v$  座標によって操作されるため, 考慮しない.

$V_A$  を凸パーツの頂点群,  $V_B$  を仮多面体の頂点群とする. 以下のエネルギー関数を最小化することで仮多面体の最適な形状を得る.

$$E = \sum_{a \in V_A} \min_{b \in V_B} \text{dist}(a_s, b) + \sum_{b \in V_B} \min_{a \in V_A} \text{dist}(b_t, a) \quad (3.3)$$

$$s.t. \quad v_0^j > v_1^j > v_2^j, \quad (3.4)$$

$$0 < s_i \leq 1, \quad (3.5)$$

$$H_{min} \leq v_2^j, \quad v_0^j \leq H_{max}, \quad (3.6)$$

式 3.3 は多面体と凸パーツの最短距離の和を表し, 二つの体積の排他的論理和を近似することを目的としている. 式 3.4 は断面線の頂点の  $v$  軸方向における一貫性を保つために設けている. また, 式 3.5 の  $0 < s_i$  は各面が反転するのを防ぎ,  $s_i \leq 1$  は各面  $\Lambda_i$  が初期形状より小さくなるよう設定している. 式 3.6 の  $H_{min}$  と  $H_{max}$  は折りたたみ方向における天頂面と底面の高さの最大, 最小値であり, この制約は多面体を  $H_{min}$  と  $H_{max}$  の範囲内に収めるために設定している. ここでは, 天頂面の適切な位置や大きさのみに焦点をおいているため, 断面線の折りたたみ可能性を考慮しない.

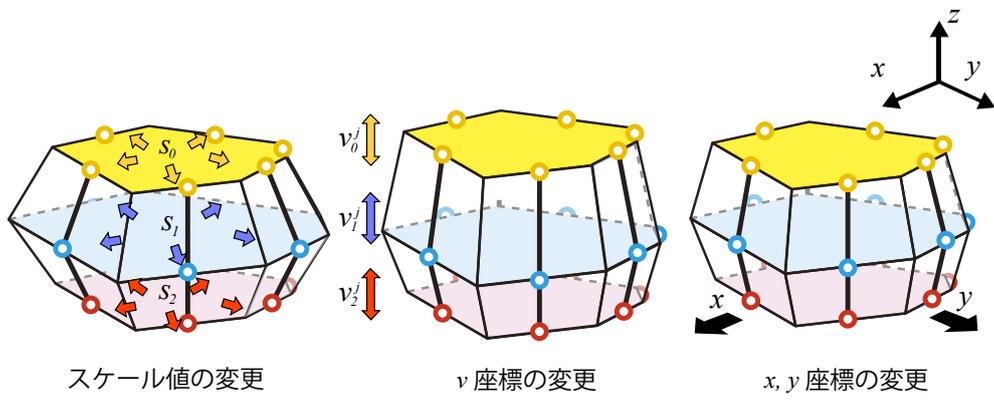


図 3.18: 最適化に用いる多面体の変形パラメータと変形の様子

### 3.7.3 断面線の生成

第 3.7.2 節の処理によって適切な天頂面が得られた。次に本システムは天頂面の各辺ごとに断面線を生成する。本節では折りたたみ可能な断面線の生成方法について詳細を述べる。

#### 初期形状の生成と接続面に平行な線の挿入

断面線の初期形状は以下の手順で生成する。

1. 断面  $\Pi_j$  と凸パーツの辺との交点を頂点とした折れ線を生成する (図 3.19 (1)).
2. 断面線のなす多角形が凸形状となるよう、不要な頂点は削除する (図 3.19 (2)).
3. 接続面の頂点を投影する (図 3.19 (3)).
4. 投影した点群の端点を折れ線と連結する (図 3.19 (4)). 投影した頂点が天頂面、底面の範囲内に存在しない場合、接続面は挿入せず、天頂面または底面と接続する。

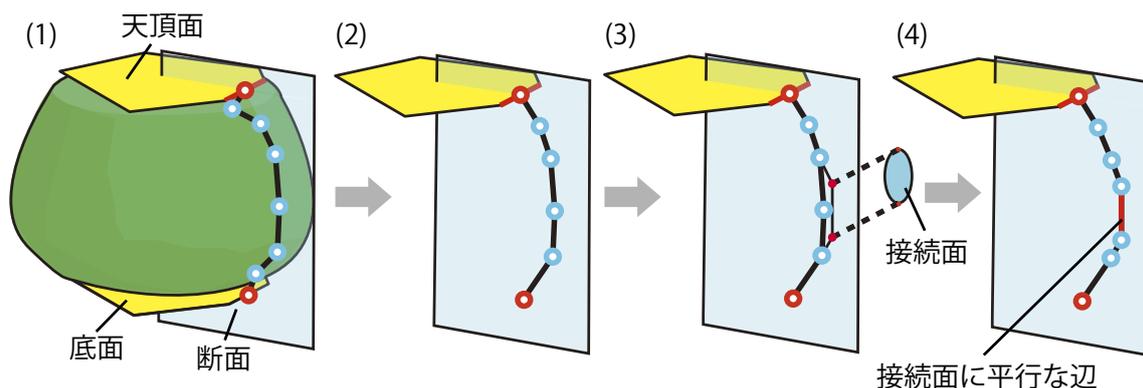


図 3.19: 断面線の生成方法を示した図. (1) 断面  $\Pi_j$  と凸パーツの交点を抽出する. (2) 断面線のなす多角形が凸形状となるよう、不要な頂点は削除する. (3) 接続面上の点を断面  $\Pi_j$  に投影する. (4) 投影した点と折れ線の頂点を連結する.

#### 各断面線の最適化

本節では各断面線を折りたたみ可能となるよう個別に最適化する方法について述べる。単純化と最適化の処理を施した断面線を複数生成し、その中から  $\epsilon$  が最小のものを選択するという流れで行う。図 3.20 は断面線の最適化の流れを示したものである。そして具体的な手順を以下に示す。

1. 断面線  $P_j$  を異なる頂点数で単純化し，新たな断面線を複数生成する．単純化後の断面線を  $\hat{P}_j^k (k = 3, 4, 5 \dots M_{max})$  とする．ここで  $k$  は単純化後の断面線の頂点数である．単純化において，断面線の端点  $\mathbf{p}_0$  と  $\mathbf{p}_{M-1}$ ，接続面と平行な線分に含まれる頂点は固定する．
2.  $\hat{P}_j^k$  について最も適切な折り頂点  $\mathbf{p}_h (h = 1, 2, \dots, k-2)$  を選択する． $\mathbf{p}_h$  の位置を式 3.7 を用いて  $v_{h+1} < v_h < v_{h-1}$  という制約の下，最適化する．そして  $\epsilon$  が最小のときの  $\mathbf{p}_h$  を  $\hat{P}_j^k$  の折り頂点とする．
3.  $\hat{P}_j^k$  を構成する頂点のうち， $\mathbf{p}_0, \mathbf{p}_{M-1}$  以外を式 3.7 を用いて式 3.11 という制約の下，最適化する．最適化において，接続面と平行な線分に含まれる頂点は接続面と平行な方向にのみ動くことができる．最後に最小の  $\epsilon$  を持つ  $\hat{P}_j^k$  を採用する．

断面線の単純化には Douglas-Peucker のアルゴリズムを用いた．このアルゴリズムは単純化後の頂点数を指定できないが，本手法では任意の頂点以下になるまで頂点を削減するよう変更を加えている．ステップ 2 において折り頂点の位置のみ修正することで，断面線全体の変形を緩和することを目的としている．

折りたたみ可能性や再現性，凸制約などを考慮に入れたエネルギー関数を式 3.7 のように定義し，各頂点の  $u, v$  座標をパラメータとして式 3.7 を最小化することで各断面線を最適化する．

$$E = w_s E_s + w_g E_g + w_c E_c \quad (3.7)$$

ここで  $E_s, E_g, E_c$  は以降で説明するエネルギー項である． $w_s, w_g, w_c$  は重みである．本手法では  $w_s = 0.7, w_g = 1.0, w_c = 0.9$  を用いた．

#### 最適化の過程で断面線の変形を妨げる項

最適化の過程で変形された断面線と初期の断面線の形状が異なることを防ぐため，エネルギー関数に以下の項を加える．

$$E_s = d_H(\mathbf{p}_i, \hat{\mathbf{p}}_{i'}) \quad (3.8)$$

ここで  $d_H(\mathcal{A}, \mathcal{B})$  は頂点の集合  $\mathcal{A}$  と  $\mathcal{B}$  のハウズドルフ距離である．そして  $\mathbf{p}_i (i = 0, 1, 2, \dots, M-1)$ ， $\hat{\mathbf{p}}_{i'} (i' = 0, 1, 2, \dots, k-1)$  は単純化前後の頂点座標である．

#### 折りたたみ可能にするための項

第 3.3 で述べたように，断面線は  $\epsilon$  がほぼ 0 である場合，折りたたみ可能であるといえる．ここで許容誤差  $\tau$  を定義し， $\epsilon$  が  $\tau$  以下の場合，折りたたみ可能とみなす．加瀬らの手法と同

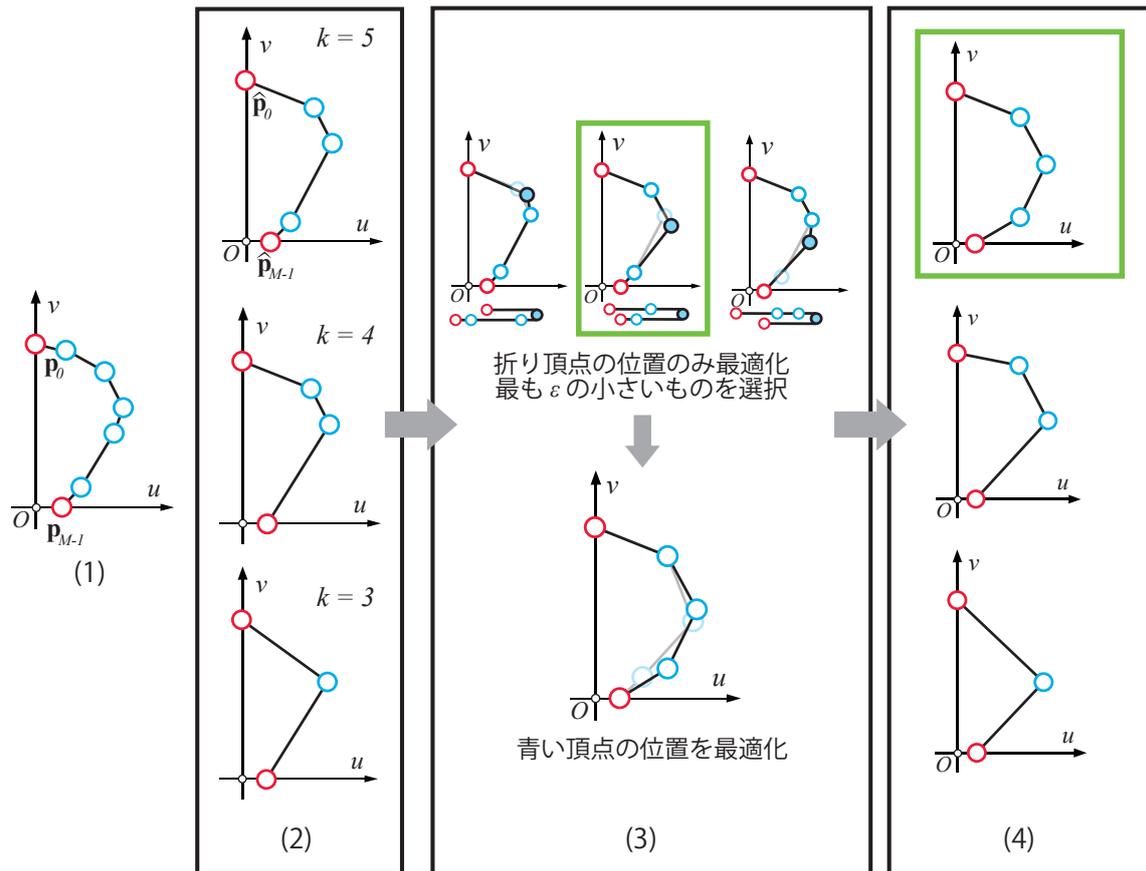


図 3.20: 断面線の最適化の手順を表した図. (1) 初期形状. (2) 断面線を単純化する. 単純化後の頂点数を  $k$  とした時, 断面線を  $\hat{P}_j^k (k = 3, 4, 5 \dots M_{max})$  とする. (3) 折りたたみ可能となるよう, 断面線の頂点位置を最適化を用いて修正する. 折り頂点を  $\mathbf{p}_h (h = 1, 2, 3 \dots k - 2)$  とした時,  $\mathbf{p}_h$  の位置を修正し,  $\epsilon$  が最小の時の  $\mathbf{p}_h$  を折り頂点とする. その後始点, 終点以外の頂点の位置を最適化を用いて修正する. (4) 最適化を用いて修正した断面線  $\hat{P}_j^k$  のうち,  $\epsilon$  が最小のものを採用する.

様に,  $\tau = 0.001 \times \bar{l}$  とする.  $\bar{l}$  は断面線の線分の平均の長さである.  $\epsilon$  を最小化するため, 以下の項を加える

$$E_g = \left| u_{M-1} - u_0 - \sum_{i=0}^{h-1} l_i + \sum_{i=h}^{M-1} l_i \right| \quad (3.9)$$

ここで  $h$  は折り頂点の番号である.

### 凸制約項

断面線が凸制約を満たしたまま最適化されるよう, 以下の項を定義する.

$$E_c = \sum_{i=1}^{M-2} d_i = \begin{cases} 0 & \text{if } \theta_i \leq \pi \\ \text{dist}(\hat{\mathbf{p}}_{i-1}\hat{\mathbf{p}}_{i+1}, \hat{\mathbf{p}}_i) & \text{otherwise} \end{cases} \quad (3.10)$$

ここで  $\text{dist}(E, \mathbf{p})$  は辺  $E$  と点  $\mathbf{p}$  とのユークリッド距離をあらわす.  $\theta_i$  はベクトル  $\hat{\mathbf{p}}_{i+1} - \hat{\mathbf{p}}_i$  とベクトル  $\hat{\mathbf{p}}_{i-1} - \hat{\mathbf{p}}_i$  の反時計回りの角度を表す. この様子を図 3.21 に示す.

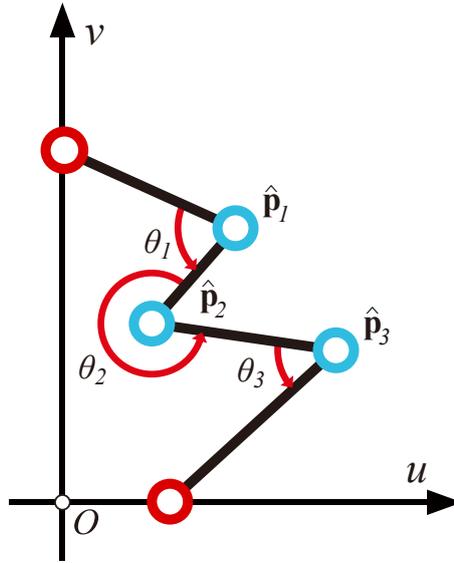


図 3.21: 凸制約項の計算方法を示した図.  $\theta_1, \theta_2, \theta_3$  は頂点  $\hat{\mathbf{p}}_1, \hat{\mathbf{p}}_2, \hat{\mathbf{p}}_3$  の反時計回りの角度を表している.  $\theta_2$  が  $\pi$  以上であるため, 線分  $\hat{\mathbf{p}}_1, \hat{\mathbf{p}}_3$  と点  $\hat{\mathbf{p}}_2$  との距離が凸制約項の値となる.

また, 頂点の  $v$  座標の一貫性を保つため (前後の頂点位置が反転し, 断面線が交差することを防ぐため) 以下の制約の元最適化を行う.

$$v_{i+1} < v_i < v_{i-1}, \quad (3.11)$$

折りたたみ可能であることが最も重要であるため、 $\epsilon$ が $\tau$ 以下にならない場合、 $w_s$ と $w_c$ の値を小さくし、 $\epsilon$ が $\tau$ 以下になるまで最適化を繰り返す。

最適化には数値微分を用いた最適化を使用した。また、初期形状は凸制約と $v$ 座標の一貫性が満たされているものとする。最適化の途中である頂点 $v_i$ が式 3.11 を満たさない場合、頂点位置を修正した上で最適化を繰り返す。例えば $v_i > v_{i-1}$ の時、 $v_i$ の位置を $v_{i-1}$ より少し小さい値にする。接続面と平行な線分の向きを保つため、線分を構成する点は線分のベクトル方向にのみ動く。線分の端点の位置を単位ベクトルの移動量のスカラ値によってパラメータ化し、 $u, v$ 座標の代わりにスカラ値を最適化することによってこのような処理を実現する。

### 3.7.4 隣り合う側面のパネルの衝突を回避するための修正

第 3.7.3 節では側面の断面線の一つずつ個別に最適化した。そして折りたたみ可能な断面線を得ることができた。しかしながら、隣り合う側面のパネルは図 3.22 の左図のように衝突する場合がある。このような側面のパネルによって構成された多面体パーツは折りたたむことができない。そこで、多面体パーツを適切に出力するため、隣合う断面線の頂点の位置関係を考慮し、断面線全体の頂点位置を修正する。

#### 衝突の生じる原因

はじめに、側面のパネルが衝突する原因を図 3.22 を用いて説明する。図 3.22 の (b) は (a) の多面体パーツの $v = v_j^i$ における断面図である。衝突の発生を説明する上で重要なのは二つの線 $L_{j+1}^i$ と $L_{j-1}^i$ である。 $L_{j+1}^i$ は $P_{j+1}$ のなす側面のパネルを切断した時にできる線であり、 $c_{j+1}^j$ は $P_{j+1}$ 上の $\mathbf{p}_j^i$ と同じ $v$ 座標を持つ点である。 $L_{j+1}^i$ は天頂面の辺 $E_{j+1}$ と平行であり $m_j$ は天頂面の辺 $E_j$ の中点である。また、 $C_{j+1}^{j-1}$ は $L_{j+1}^i$ と $L_{j-1}^i$ の交点である。図 3.22 の左図のように $\mathbf{p}_j^i$ が $C_{j+1}^{j-1}$ より外側にある場合、衝突が発生する。つまり、衝突の発生する条件を式で表すと以下ようになる。

$$\|\mathbf{p}_j^i - m_j\| > \|C_{j+1}^{j-1} - m_j\| \quad (3.12)$$

多面体パーツを適切に生成するため、すべての頂点が式 3.12 を満たすよう、以下の項を追加したエネルギー関数を定義し、最小化することで断面線を修正する。

$$E_f = \sum_{j=0}^{N-1} \sum_{i=1}^{M-2} \max(\|I_{j+1}^i - \mathbf{p}_j^i\|, \|I_{j-1}^i - \mathbf{p}_j^i\|), \quad (3.13)$$

ここで、 $I_{j+1}^i$ 、 $I_{j-1}^i$ は図 3.23 で示すように、線分 $\mathbf{p}_j^i m_j$ と線 $L_{j+1}^j$ 、 $L_{j-1}^j$ との交点である。そして式 3.7 を以下のように変更する。

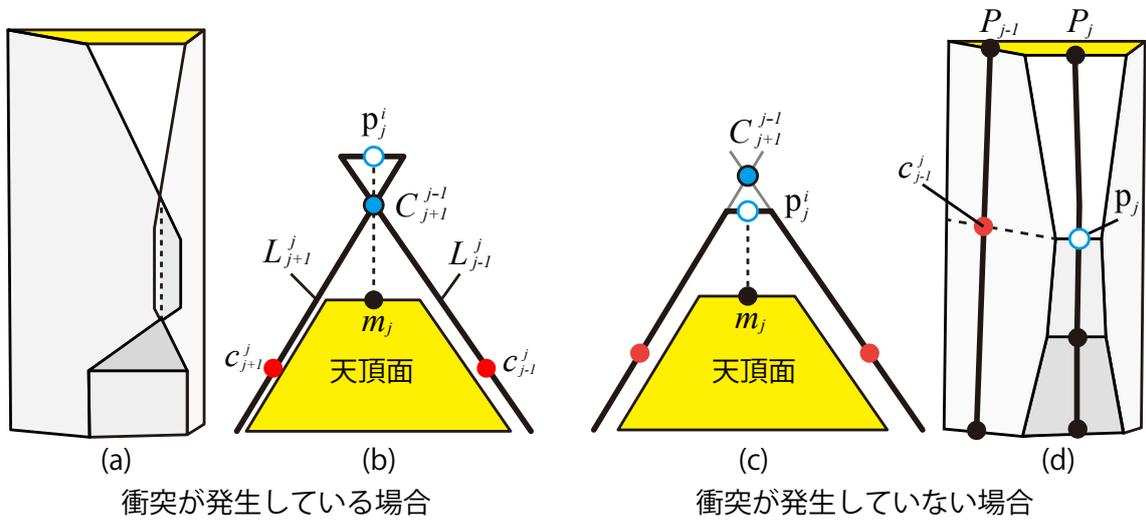


図 3.22: 隣合う側面のパネル間の衝突の様子を示した図. (a) 隣り合う側面のパネル間で衝突が発生した多面体パーツ. (b) (a) の多面体パーツの  $v = v_j^i$  における断図. (c) (d) の多面体パーツの  $v = v_j^i$  における断図. (d) 隣り合う側面のパネル間で衝突が発生していない多面体パーツ.

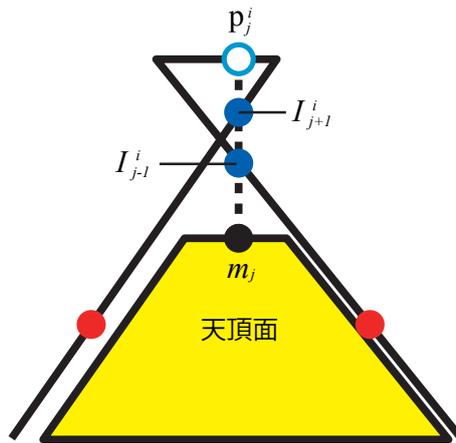


図 3.23: 隣合う側面のパネルで衝突が発生している多面体パーツの断面

$$E = w_s E_s + w_g E_g + w_c E_c + w_f E_f, \quad (3.14)$$

重みには  $w_s = 0.01, w_g = 1.0, w_c = 0.01, w_f = 1.0$  を用いた。そして式 3.14 を用いて断面線の頂点位置を修正する。  $\epsilon \leq \tau$  と式 3.12 をすべての頂点が満たすまで最適化を繰り返す。

以上で天頂面、側面の断面線の生成処理が完了したため、多面体パーツの三次元形状を出力することが可能となる。各多面体パーツを回転、平行移動することで隣接するパーツと接続される。多面体パーツの生成例は第 4 章で示す。

### 3.8 対称性を考慮した折りたたみ可能な立体の生成

私たちの身の回りには対称性を持つ形状が数多くある。例えば人間、動物などの生き物や車、マグカップ、などの日常的に使う製品も多くのもが左右対称な形状である。第 3.7.4 節までの処理によって折りたたみ可能な立体を生成できるが、入力形状が左右対称な形であったとしても、生成される立体は左右対称ではない。対称性を考慮することで折りたたみ可能な立体の近似精度や最終形状の見栄えや組み立てやすさが向上すると考えられる。そこで入力形状が左右対称な形状である場合、対称性を保ったまま折りたたみ可能な立体を出力する。本手法では図 3.24 の (a) のように、ある面に対して対称な立体を出力するものとする。そしてこのような面を対称面と呼ぶ。対称形状を保ったまま変換するかどうかはユーザーが決定する。システム上にチェックボックスを配置し、チェックしたか否かで判断する。

#### 3.8.1 凸パーツの分類

折りたたみ可能な立体は多面体パーツによって構成される。したがって立体全体が左右対称となるよう、多面体パーツを生成しなければならない。図 3.24 は Armadillo という三次元メッシュモデルの凸パーツと対称面  $\Pi$  である。Armadillo を構成する凸パーツのうち、手、足、耳は右耳、左耳のように対となる凸パーツが存在するが、それ自身は  $\Pi$  について対称な形ではない。一方、頭、胴体、腰、尾は  $\Pi$  について対称な形である。手、足、耳についてはこれまでと同様の方法で折りたたみ可能に変換し、その後  $\Pi$  に対して反転することで反対側の多面体パーツも生成することが可能である。しかし、頭、胴体などは面  $\Pi$  について対称な多面体パーツが得られるよう天頂面や断面線の形状を決定しなければならない。

本システムでは対称面はユーザーによって配置されるものとする。システム上に表示される面を平行移動、回転することで配置する。対称面が配置された後、対称性持つ凸パーツとを持たないパーツの分類、凸パーツの左右の対応付けは自動で行われる。対称面と交差する凸パーツは対称性を持つものとし、それ以外を対称性を持たない凸パーツとする。対称性を持たない凸パーツの左右の対応付けは以下の手順で行う。

1. 各凸パーツを対称面に対して左右に分類する。

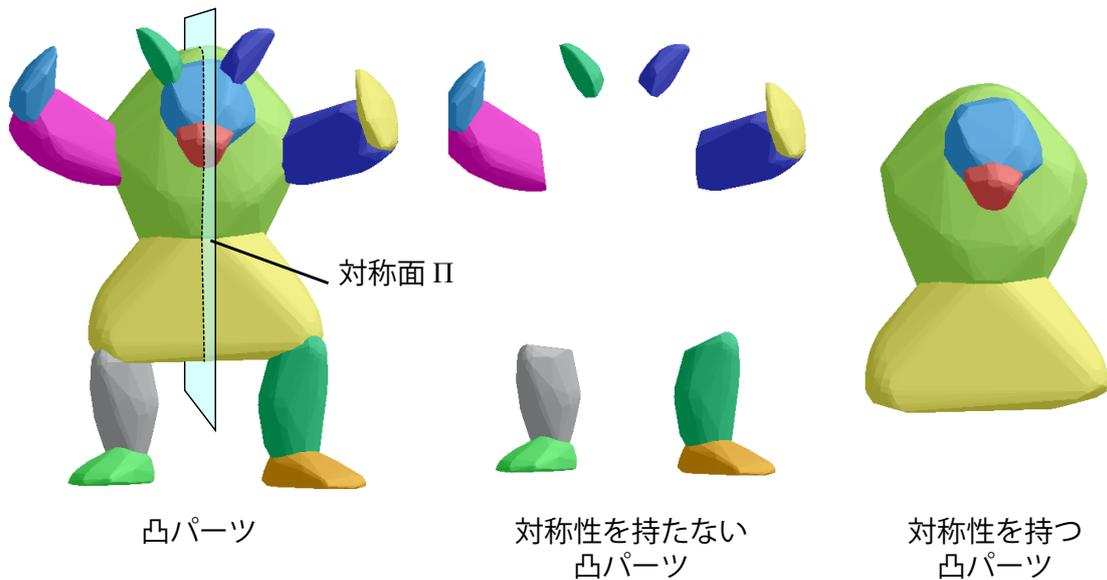


図 3.24: 対称性を持つ形状の凸パーツの集合と対称面  $\Pi$

2. 片方の凸パーツを対称面に対して反転し，左右の凸パーツ間のハウズドルフ距離を計測する。
3. 距離の最も小さいパーツ同士を左右対称なパーツとする。

### 3.8.2 対称性を考慮した折りたたみ方向の選択

多面体パーツの形状は天頂面と側面の断面線によって決定されるため，多面体パーツを対称な形状とするには天頂面や断面線が面  $\Pi$  について対称でなければならない。そして天頂面，断面線の配置は折りたたみ方向によって決定されるため，対称性を持つ凸パーツの場合，折りたたみ方向は以下の2つに限定される。

- 対称面に垂直な方向 (図 3.25 左)
- 対称面に平行な方向 (図 3.25 右)

対称面に垂直な方向を選択した場合，各断面線が  $\Pi$  について対称な形状となり，天頂面と底面は同じ形である。対称面に平行な方向を選択した場合，天頂面が  $\Pi$  について対称な形状となる。以降では折りたたみ方向が対称面に対して垂直，平行な場合それぞれについて多面体パーツの生成方法を述べる。

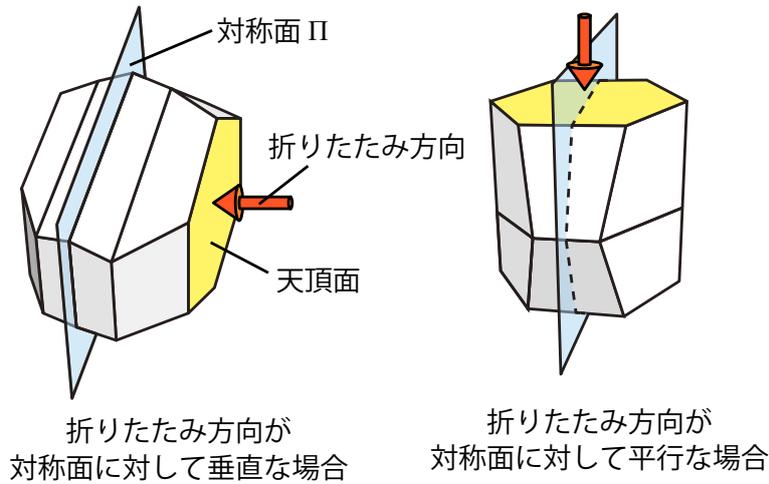


図 3.25: 対称性を持つ多面体パーツの折りたたみ方向における分類

### 3.8.3 折りたたみ方向が対称面に垂直な場合

#### 天頂面の生成

第 3.8.2 節で述べたように、折りたたみ方向が対称面に垂直な場合、側面の断面線が対称形状であり、天頂面と底面は同じ形状である。天頂面の初期形状の決定については第 3.7.2 節と同様に行うが、天頂面の最適化では仮多面体が最適化の過程で常に対称な形であるよう、パラメータに以下のような制限を加える。

- $\Lambda_1$  は対称面上に乗る。
- 天頂面  $\Lambda_0$  と底面  $\Lambda_2$  の形状は等しい。
- 天頂面  $\Lambda_0$  と底面  $\Lambda_2$  の対称面  $\Lambda_1$  との距離は等しい。

そして制約を考慮した仮多面体の変形パラメータは以下である。

1. 面  $\Lambda_0$  と  $\Lambda_2$  のスケール値  $s_{02}$ ,  $\Lambda_1$  のスケール値  $s_1$  (2 変数)
2. 面  $\Lambda_0$  と  $\Lambda_2$  の  $\Lambda_1$  との距離  $d$  (1 変数)
3. 多面体の重心の  $x, y$  座標 (2 変数)

仮多面体の変形の様子を図 3.26 に示す。

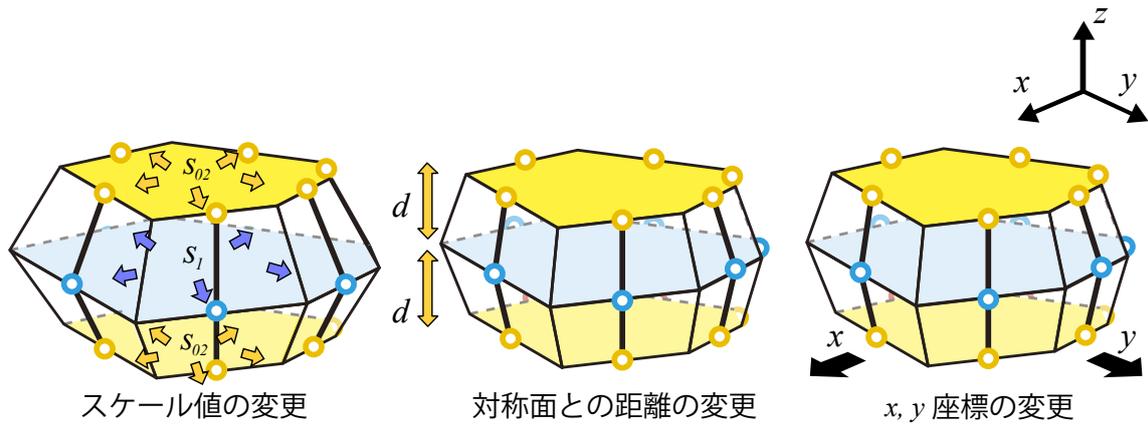


図 3.26: 折りたたみ方向が対称面に対して垂直な場合の仮多面体とパラメータの変更の様子

### 断面線の生成

第 3.8.2 節で述べたように、折りたたみ方向が対称面に垂直な場合、側面の断面線が対称形状となる。そこで本手法では図 3.27 の中図のように対称面上に折り頂点を配置し、天頂面側の頂点を対称面に対して反転することで対称性をもつ断面線を得る。このような断面線は始点と終点の  $u$  座標が等しく、始点から折り頂点までの折れ線の長さと折り頂点から終点までの折れ線の長さが等しい。折りたたみの前後で始点の  $u$  座標が変化せず  $\epsilon = 0$  となるため、ゆがみが生じない。したがって常に折りたたみ可能であり、最適化する必要はない。しかし、隣り合う断面線の配置によっては隣接する側面のパネルが衝突する可能性がある。そこで第 3.7.4 節と同様に、式 3.13 を最小化することで断面線の頂点位置を修正する。断面線が対称性を保ったまま修正されるよう、最適化の過程で断面線の頂点は対称性を保ったまま動くよう制限する。例えば図 3.28 は最適化における頂点の動きを表した図である。図 3.28 の中図の頂点  $\mathbf{p}_1$  は右上方向に動き、頂点  $\mathbf{p}_1$  と対になる頂点  $\mathbf{p}_3$  は右下に動いている。また、折り頂点  $\mathbf{p}_2$  は常に対称面上にあるため、 $u$  軸方向にのみ動く。このような処理によって断面線の対称性が保たれる。

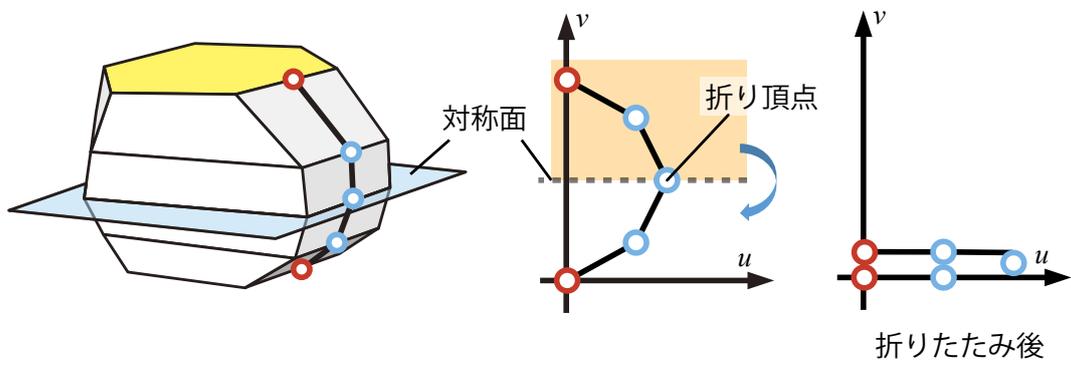


図 3.27: 折りたたみ方向が対称面に対して垂直な場合の断面線の例. 対称面に対して頂点位置を反転する.

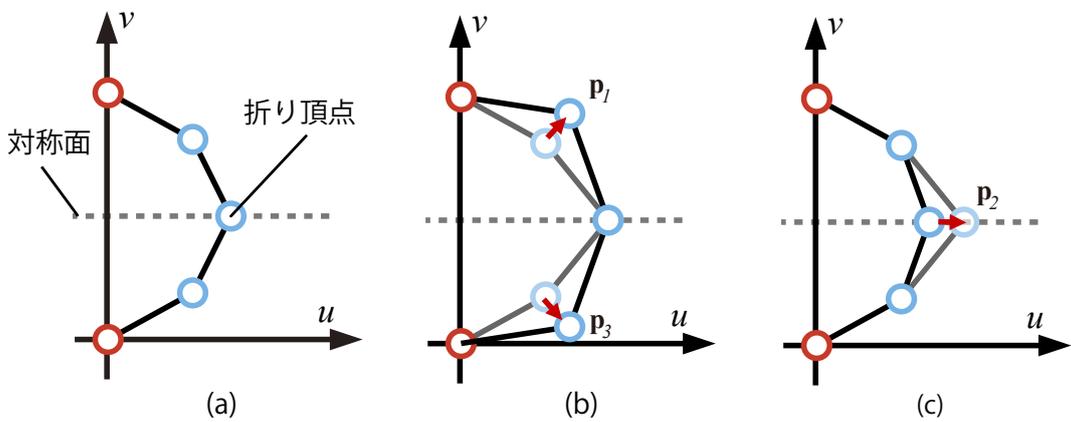


図 3.28: 対称性を考慮した断面線の最適化における頂点の変化を表した図. (a) 最適化前の形状. (b) 頂点  $p_1$ ,  $p_3$  の変化の様子. (c) 折り頂点  $p_2$  の変化の様子.

### 3.8.4 折りたたみ方向が対称面に平行な場合

#### 天頂面の生成

折りたたみ方向が面  $\Pi$  に平行な場合、天頂面が面  $\Pi$  に対して対称な形状となる。そこで第 3.7.2 節と同様に天頂面の初期形状を生成した後、天頂面を以下の手順で対称な形状へ変形する。また、その様子を図 3.29 に示す。

1. 天頂面の頂点を面  $\Pi$  について左右に分類する
2. 頂点数の多い方を残し、反対側へ反転する
3. 頂点をつなぎなおし、多角形を生成する
4. 対称面をまたぐ頂点のうち、辺の長さが各辺の平均の長さ  $\bar{l} * 0.1$  より小さい場合、辺の端点を対称面上に移動して 1 つに結合する

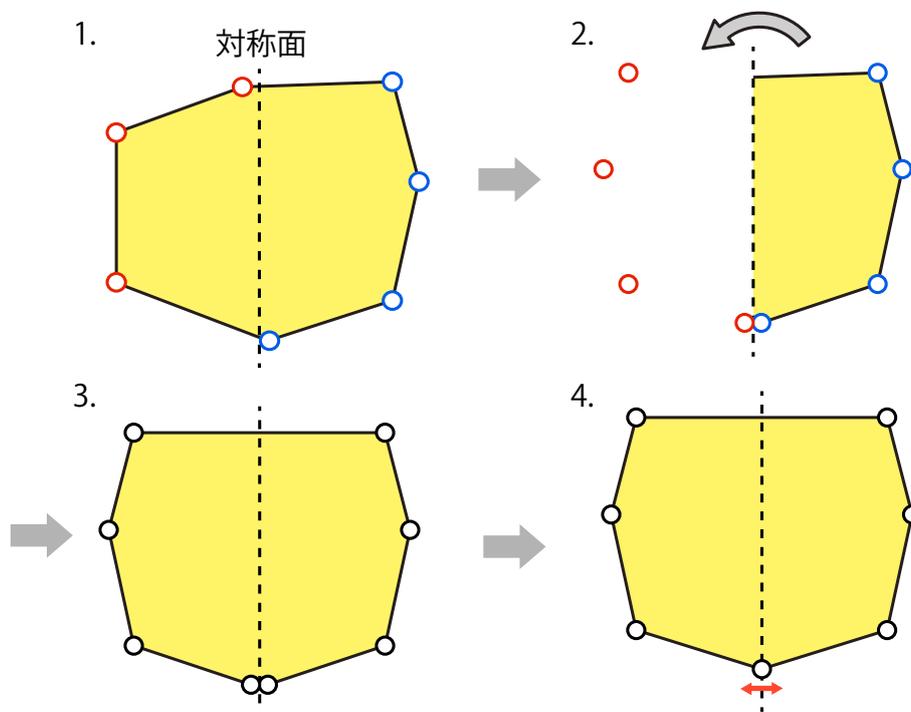


図 3.29: 天頂面を対称形状に変換する手順を表した図.

#### 天頂面の最適化

折りたたみ方向が対称面に対して平行な場合、天頂面は対称形状であるが、各断面線についての制約はない。そこで天頂面の最適化は第 3.7.2 節と同様に行う。しかし、対称性を損な

わないようにするため、仮多面体の重心の移動は対称面の上限定する。この様子を図 3.30 に示す。

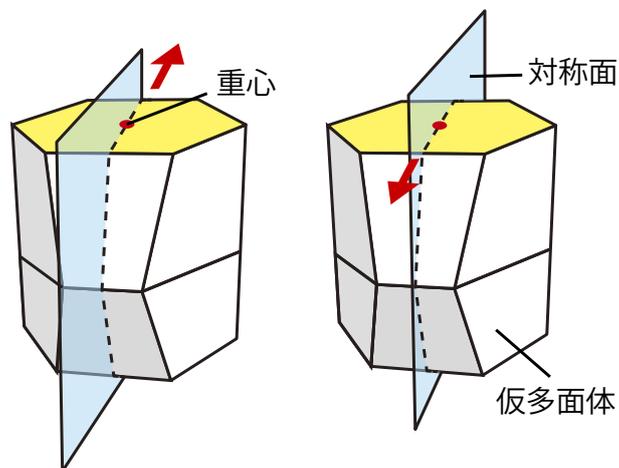


図 3.30: 折りたたみ方向が対称面に対して平行な場合の仮多面体の変形の様子

### 断面線の生成

断面線は天頂面の各辺に対応し、天頂面は対称形状であるため、各辺には対となる辺が存在する。例えば図 3.31 の天頂面の辺  $E_1$ ,  $E_2$  はそれぞれ断面線  $E_5$ ,  $E_4$  に対応する。そこで図 3.31 のように片側だけ断面線を生成し、対となる辺へ複製することで天頂面の各辺に対応する断面線すべてを得ることができる。 $P_1$ ,  $P_2$  は初期形状の生成、最適化ともに第 3.7.3 節と同様に行う。また、側面のパネルの衝突を回避するため、 $P_0$ ,  $P_1$ ,  $P_2$ ,  $P_3$  の断面線を式 3.13 を最小化することで修正し、 $P_5$  と  $P_4$  はそれぞれ  $P_1$  と  $P_2$  を複製する。

以上の処理で面  $\Pi$  について対称な多面体パーツを生成する。多面体パーツを回転、平行移動することで隣接するパーツと接続し、折りたたみ可能な立体の全体形状を得る。

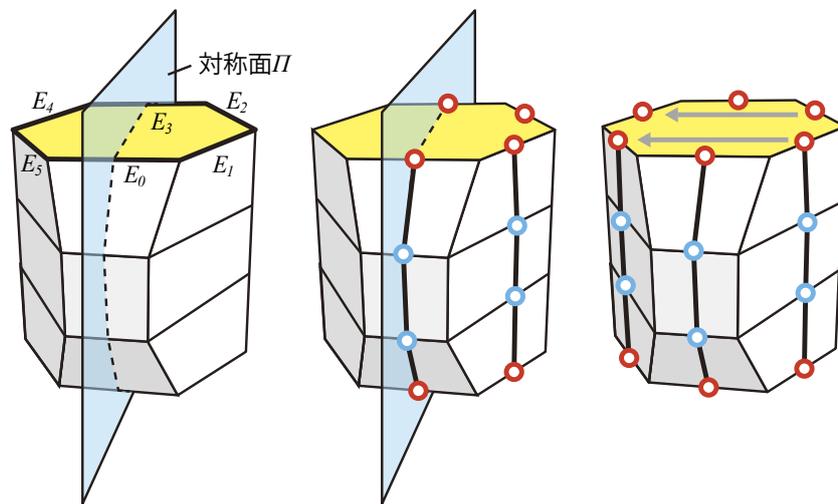


図 3.31: 対称性を考慮した断面線の生成の様子

## 第4章 結果

本手法の有効性を示すため、C++とCGのAPIである、OpenGLを用いて提案手法をシステムに実装し、折りたたみ可能な立体を生成した。使用したPCのスペックはIntel Core i7, 8GBのRAMである。また、入力形状を凸パーツへ分割するため、C++のライブラリであるCGAL(文献 [31])を利用した。本章では実装したシステムによって出力した折りたたみ可能な立体とシステム上で表示される折りたたみのアニメーションを図で示す。また、提案システムによって出力された折りたたみ可能な立体の紙模型を作成し、実際に折りたたむことを確認した。最後に紙模型の折りたたみの様子を図で示す。

### 4.1 折りたたみ可能な立体の出力例

図 4.1 は提案手法によって生成された折りたたみ可能な立体の三次元モデルである。そして図 4.2 は各折りたたみ可能な立体の折りたたむ様子を示している。図のように入力形状から出力された多面体パーツは折りたたまれると平坦になり、折りたたみ可能な立体を出力できていることがわかる。図 4.3 は出力した折りたたみ可能な立体の紙模型とその折りたたむ様子である。紙模型の出力のため、提案システムで出力した三次元モデルからペパクラデザイナー [32] を用いて展開図を作成した。その展開図に Adobe Illustrator を用いてのりしろを付け加え、接着剤を用いて模型を組み立てた。これらの結果から、提案手法によって生成された立体は実際に折りたたみ可能であることが確認できた。

### 4.2 対称性を考慮した折りたたみ可能な立体

図 4.4, 図 4.5 に対称性を考慮した折りたたみ可能な立体と提案システム上で表示される折りたたみのアニメーションを示す。多面体パーツと同じ色で描かれた矢印は各パーツの折りたたむ方向を示している。このようにこれらの立体は平坦に折りたたむことが可能である。

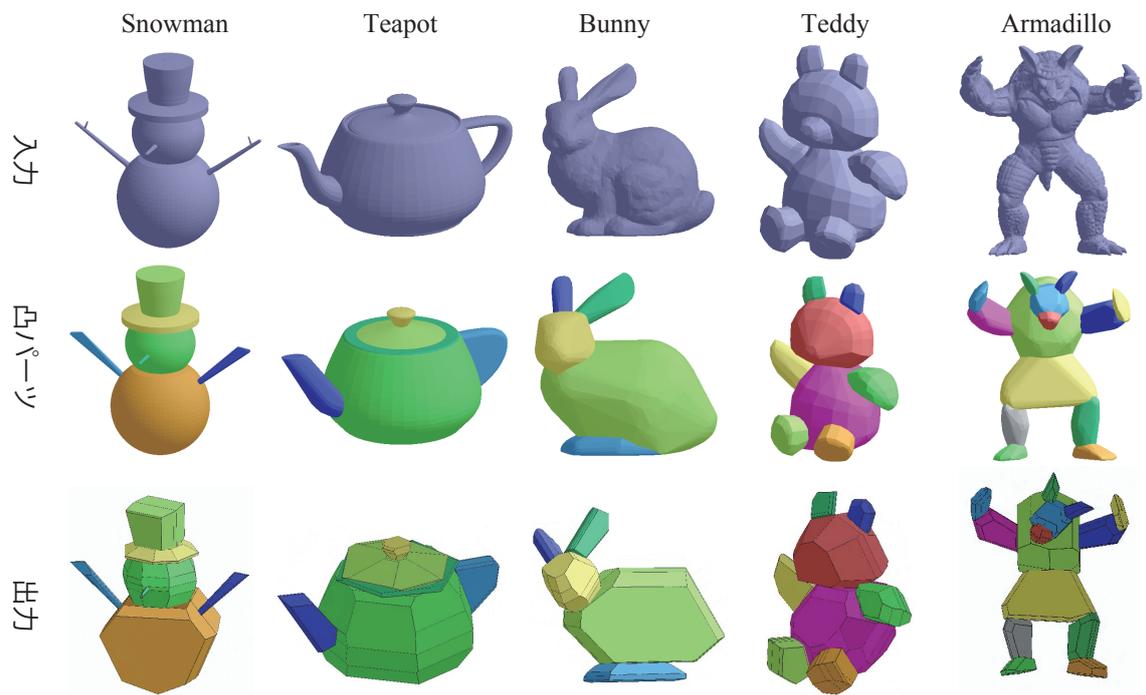


図 4.1: 折りたたみ可能な立体の例

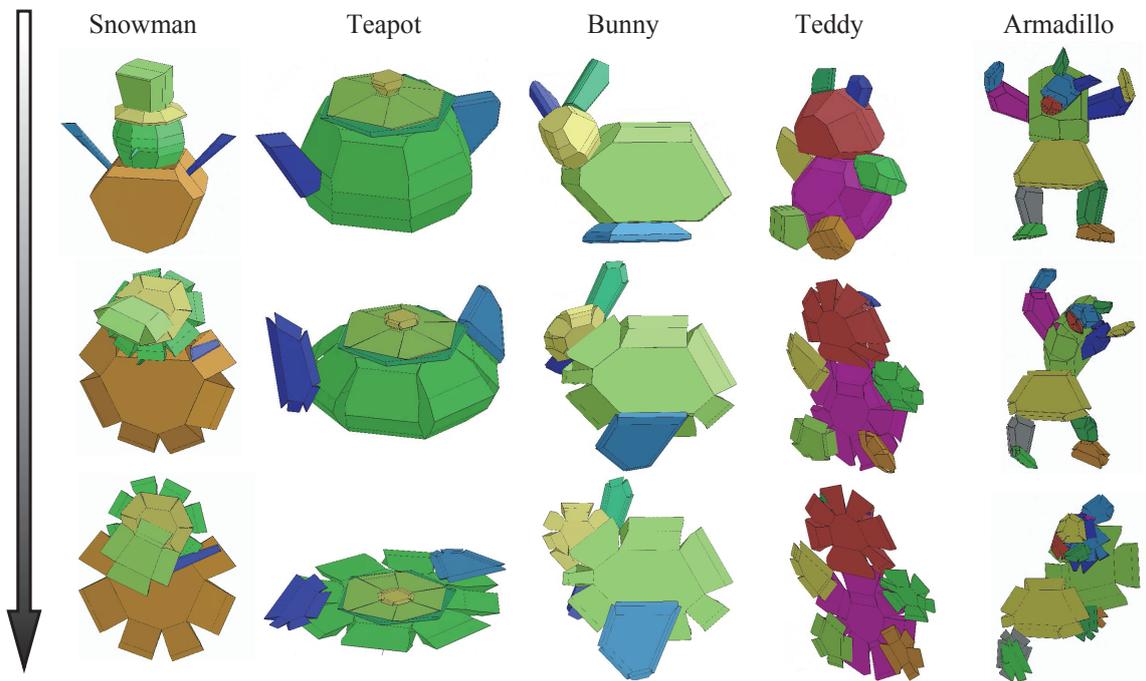


図 4.2: 折りたたみ可能な立体の折りたたむ様子

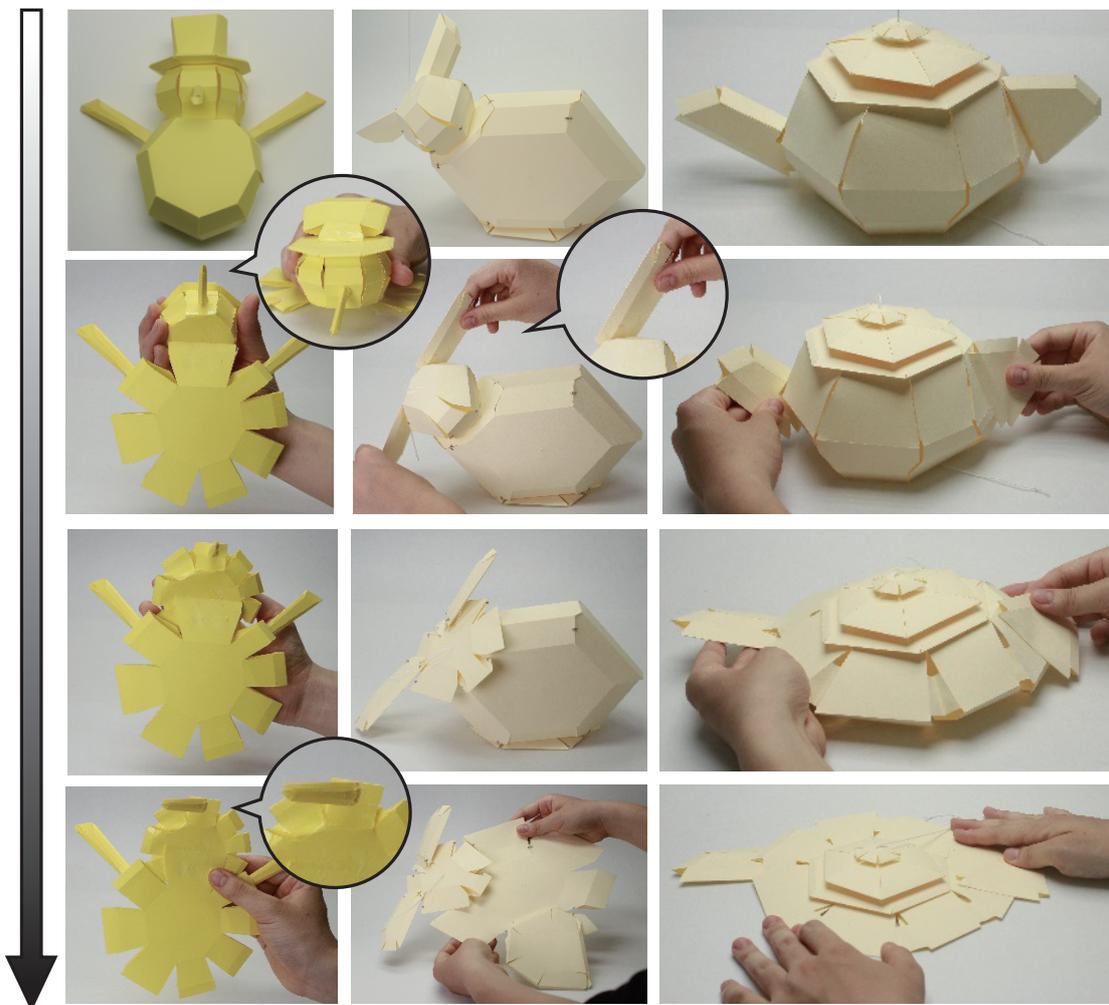


図 4.3: 折りたたみ可能な紙模型

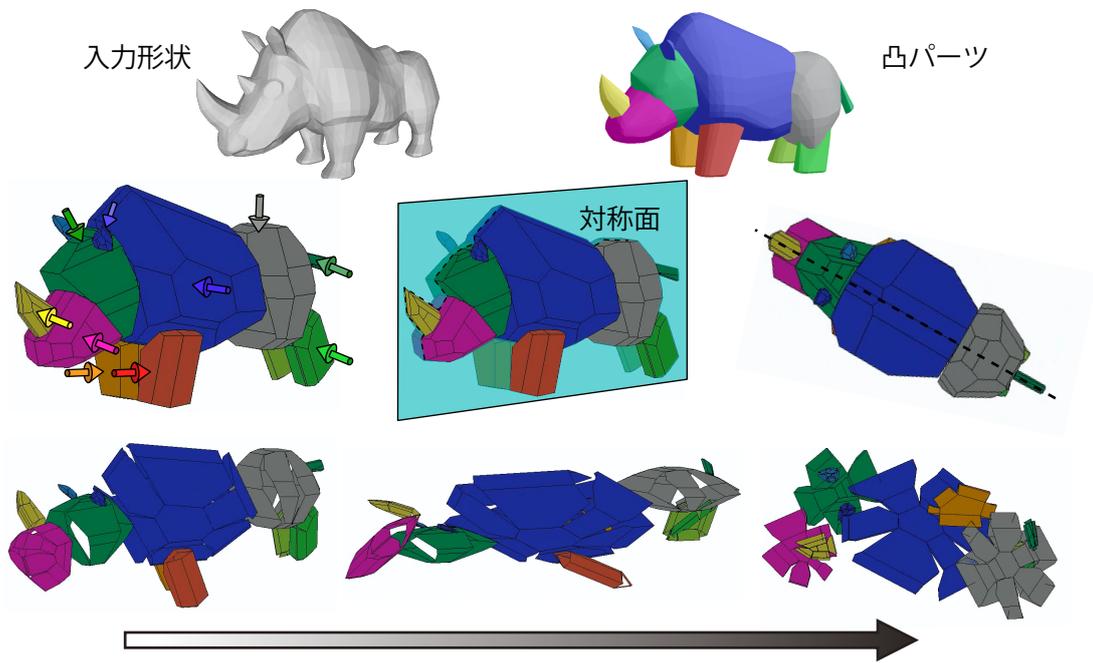


図 4.4: 対称性を考慮した折りたたみ可能な立体の例. 二段目左図の矢印は折りたたみ方向を表す.

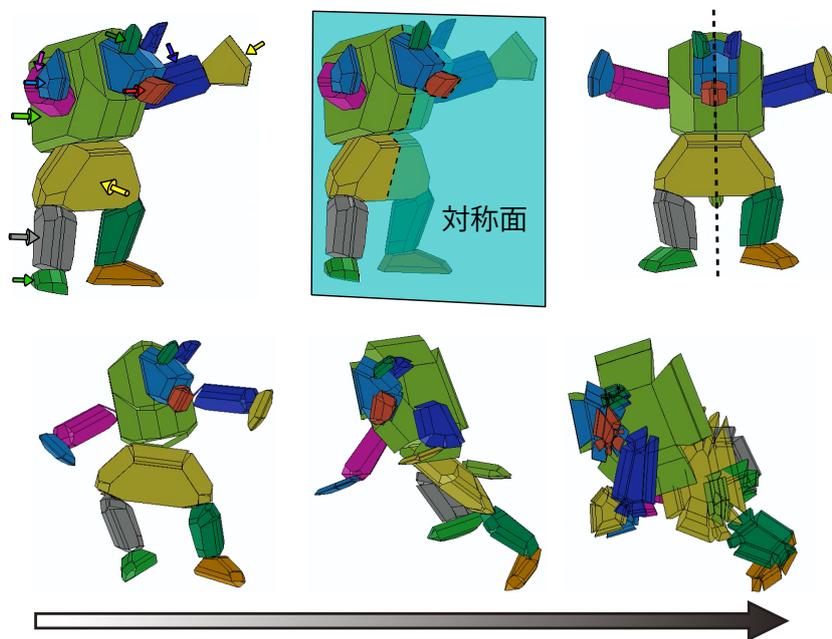


図 4.5: 対称性を考慮した折りたたみ可能な立体の例 2

## 第5章 考察

本章では提案手法を用いて出力した立体の近似誤差や計算時間を比較し、本手法の有効性や課題について考察する。

### 5.1 パラメータの変化による近似誤差と計算時間

表 5.1 に図 4.1 で示した立体を生成するのに要した計算時間を示す。表 5.1 の中で、最も計算時間が長いのは Armadillo である。これはパーツの数が最も多いことが原因である。Teddy と Teapot を比較すると、Teddy の方がパーツ数が多いにもかかわらず、パーツの変換に必要な計算時間は Teapot のほうが長い。Teddy よりも Teapot のメッシュの数が多いからである。これらのことから、計算時間はメッシュの数とパーツの数に影響されることがわかる。

表 5.2 は図 4.1 の Bunny の凸パーツと折りたたみ可能な立体の近似誤差を表したものである。近似誤差はハウズドルフ距離をバウンディングボックスの最大の辺の長さで割ったものである。上から二段目は断面線の頂点数の最大値  $M_{max}$  を大きくした時の誤差を示している。 $M_{max}$  の値を大きくすると、誤差も小さくなっていることがわかる。一方四段目のように  $M_{max}$  の値を小さくすると誤差が増加している。三段目は天頂面  $N_{min}$  の値を大きくした場合を示しており、全体的な誤差が減少している。五段目のように  $N_{min}$  を小さくすると誤差は若干大きくなる。

これらの結果から、断面線、天頂面の増減によって近似精度を操作することが可能であることが示されている。また、 $M_{max}$  のほうが  $N_{min}$  よりも近似精度に影響を与える。これは天頂面、底面に比べ、側面のパネルが占める表面積のほうが大きいためだと考えられる。

パラメータの変化と計算時間の関係を表す表 5.3 を見てみると、 $M_{max}$ 、 $N_{min}$  の値を大きくすると、計算時間が増加していることがわかる。パラメータの値を大きくすることで近似精度は向上するが、同時に計算時間の増加を引き起こす。特に  $M_{max}$  の値を大きくすると、計算時間が著しく増加している。近似精度と計算時間の関係を考慮するに、 $N_{min}$ 、 $M_{max}$  は共に 10 以下がよいと考えられる。

表 5.1: 折りたたみ可能な立体の計算時間 (sec)

	メッシュ数 (パーツ数)	凸パーツへの変換	多面体パーツ への変換	総計
Snowman	13,450 (6)	771	234	1,005
Teapot	6,543 (6)	148	193	341
Bunny	4,968 (5)	13	194	207
Teddy	3,192 (8)	27	174	201
Armadillo	17,296 (15)	63	449	512

表 5.2: Bunny のモデルの近似誤差

		ハウズドルフ距離 (天頂面の頂点数)					
$N_{min}$	$M_{max}$	左耳	右耳	頭	体	足	平均
5	6	0.048 (5)	0.038 (5)	0.056 (5)	0.092 (5)	0.098 (5)	0.066
5	10	0.046 (5)	0.034 (5)	0.063 (5)	0.087 (5)	0.097 (5)	0.065
8	6	0.049 (8)	0.052 (8)	0.039 (8)	0.093 (8)	0.081 (8)	0.063
5	3	0.044 (5)	0.047 (5)	0.062 (6)	0.097 (5)	0.101 (5)	0.070
4	6	0.051 (4)	0.032 (4)	0.055 (5)	0.100 (5)	0.092 (4)	0.066

表 5.3: Bunny の頭部の計算時間

$N_{min}$	$M_{max}$	計算時間 (sec)
5	6	15.886
5	10	49.501
8	6	26.599
10	6	34.235
5	3	4.965
4	6	12.268

## 5.2 断面線の収束

最適化前後の断面線の例を図 5.1 に示す。図 5.1 の青い線は断面線の初期状態，灰色の線は単純化した状態，オレンジ色の線は最適化した状態を示しており， $M_{max} = 5$  である。図のように  $\epsilon$  は許容誤差  $\tau$  よりも小さくなっており，折りたたみ可能に収束していることがわかる。また，図 5.2 は接続面と平行な線分を含む断面線の最適化前後の様子を示している。水色の点は接続面と平行な線分の端点を表している。 $\epsilon$  は  $\tau$  よりも小さくなっており，さらに接続面と平行な線分の向きが変化することなく，折りたたみ可能に収束していることがわかる。

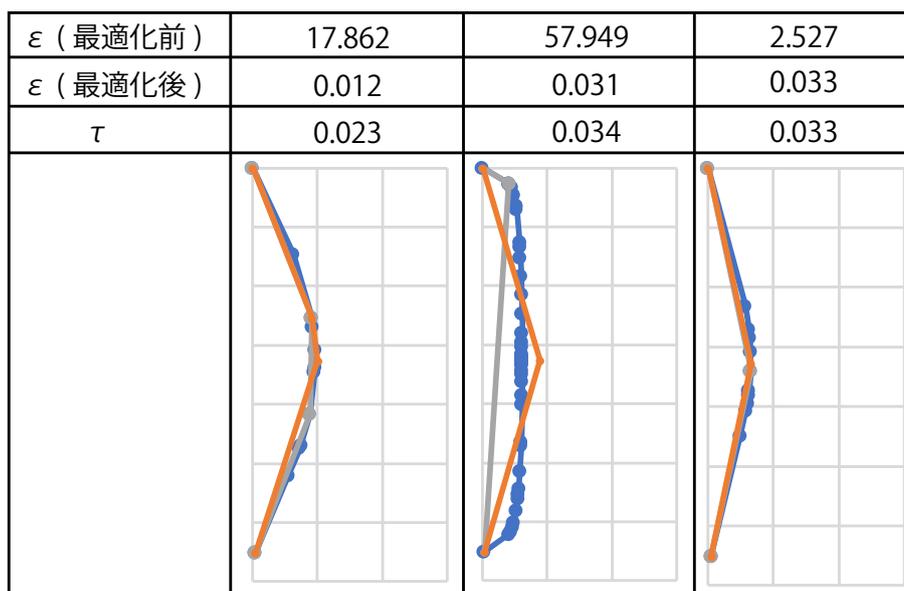


図 5.1: 最適化前後の断面線の比較。青い線は断面と凸パーツの交点から抽出した断面線の初期形状，灰色の線は単純化した断面線，オレンジ色の線は最適化した断面線。

## 5.3 側面のパネルの衝突と断面線の修正

図 5.3 に側面のパネルの衝突が生じた多面体パーツと断面線，そしてその修正後の例を示す。図のように，(1)，(2)，(3) の多面体パーツは衝突を回避するよう断面線が修正されたことが確かめられた。断面線 (a)～(e) は多面体パーツ (1) の断面線である。(a)～(e) すべての断面線の  $\epsilon$  の値が  $\tau$  以下になっており，折りたたみ可能に収束していることがわかる。断面線 (e) は接続面と平行な線分を含むが，修正の前後で線分の向きが変わることなく修正されている。衝突の生じた多面体パーツを観察すると，側面のパネル間で衝突が生じるのは天頂面の隣り合う辺との成す角が小さい場合である。赤い丸で囲んだ天頂面の辺は対応するパネル間で衝突が生じている。それらの辺は隣り合う辺との向きがほぼ等しく，なす角が小さい。したがって，天頂面の頂点数を大きくすると，滑らかな面の表現が可能だが，側面のパネルの衝突が

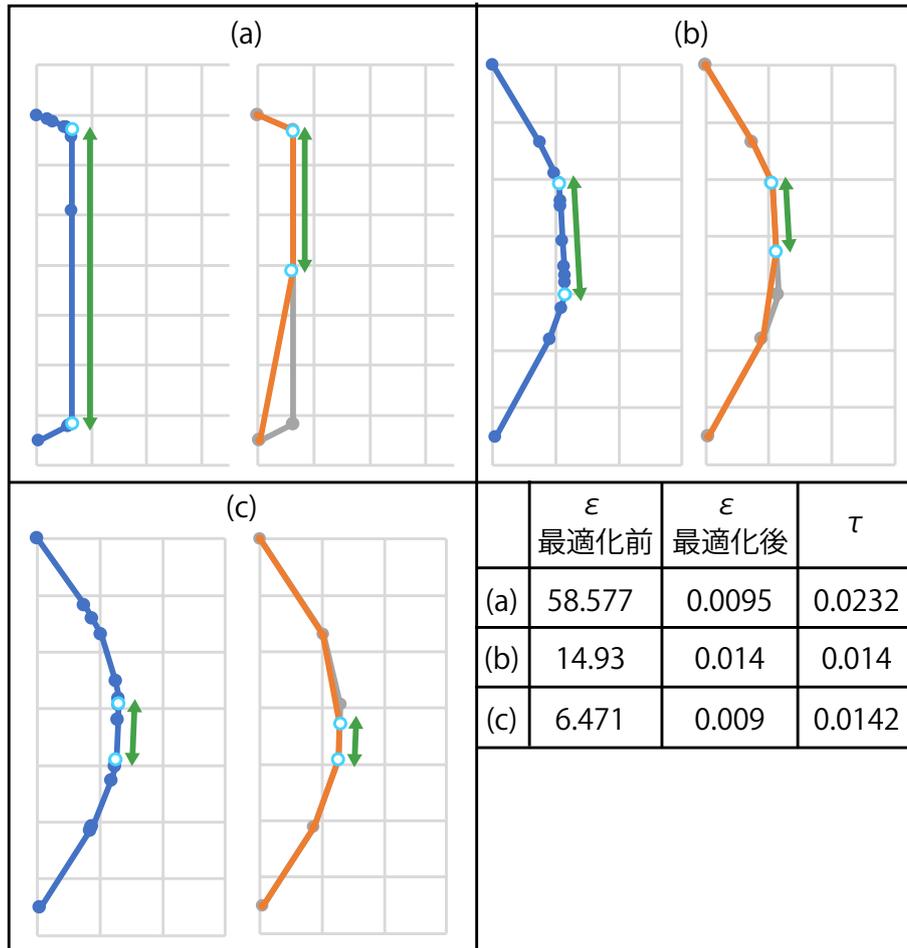


図 5.2: 接続面に平行な線を含む断面線の最適化前後の比較。青い線は断面と凸パーツの交点から抽出した断面線の初期形状，灰色の線は単純化した断面線，オレンジ色の線は最適化した断面線。水色の点は接続パネルと平行な線分の端点を表す。緑の矢印は接続パネルと平行なベクトルである。

起こりやすくなると考えられる。一方 (4) は二箇所でパネルの衝突が生じており、片方の衝突は修正されているが、もう片方は修正されていない。これは隣合う断面線が両方とも接続面に平行な線分を含んでおり、線分の端点は頂点の移動が制限されているため、衝突の生じない位置に修正することができないためである。

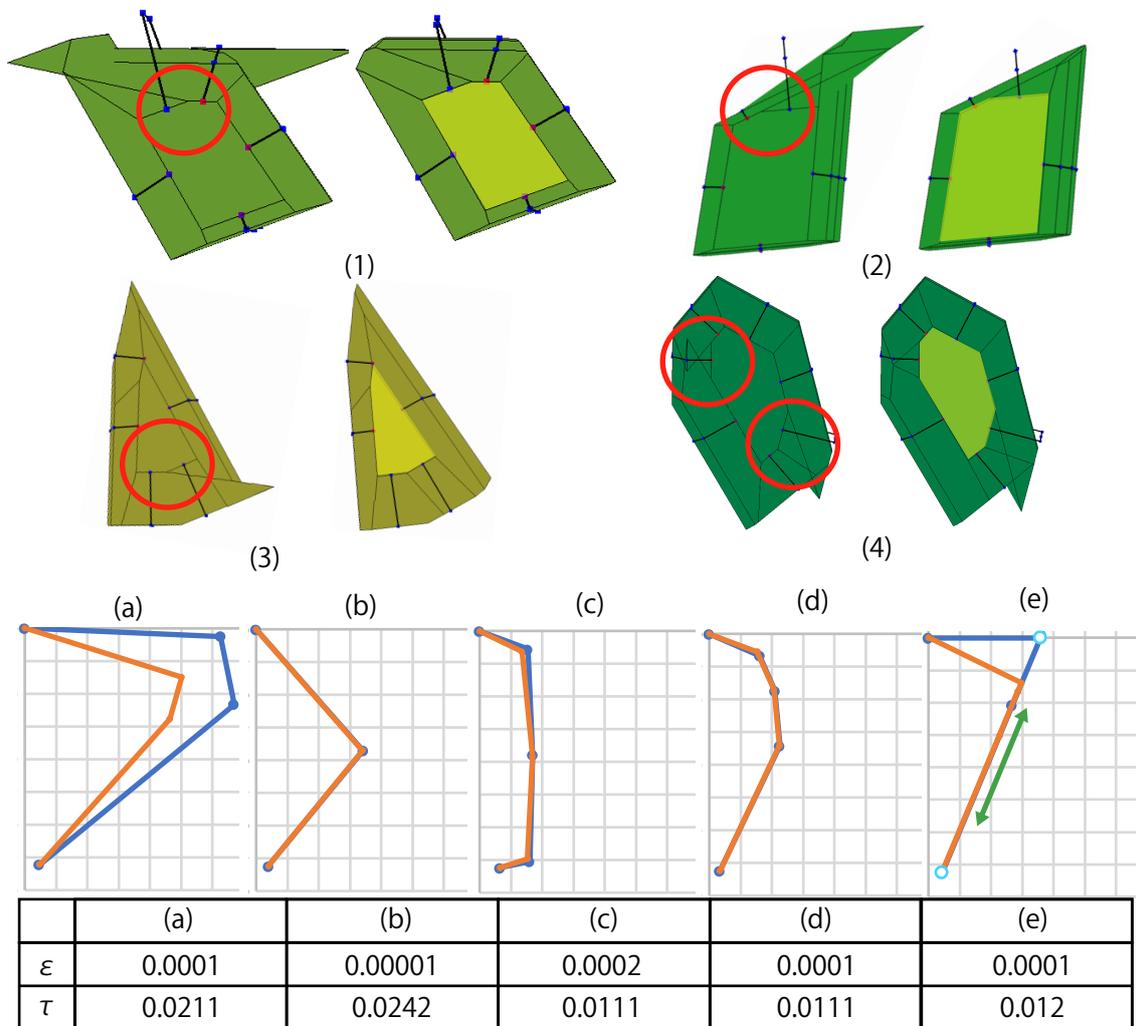


図 5.3: 断面線の修正前後の比較。青い線は修正前の断面線，オレンジ色の線は修正後の断面線。水色の点は接続パネルと平行な線分の端点を表す。緑の矢印は接続パネルと平行なベクトルである。黄色いパネルは天頂面である。

## 5.4 凸制約による影響

凸制約によって入力形状の詳細が損なわれる場合がある。たとえば、図 4.1 の Armadillo の腰や Bunny の頭部は適切な分割がなされなかったことや凸パーツへの変換によってその細部が損なわれている。図 5.4 の左図は穴を持つ立方体である。これを折りたたみ可能に変換すると、右図のように穴のない形に変換されてしまう。これも凸パーツへの変換や不適切な分割によるものである。また、このような穴の開いた形状を分割し、多面体パーツに変換した例とその折りたたみの様子を図 5.5 に示す。変換後も穴の開いたトポロジを保っているように見えるが、赤い丸で囲った部分は衝突が生じている。また、折りたたみの際はパーツ A と C が離れトポロジが損なわれている。これは多面体パーツ A, B, C は互いに接続しており、接続関係の木構造にサイクルが生じてしまったことが原因である。多面体パーツ A を基準に折りたたむ場合、B は A の折りたたみに追従し、B は C の折りたたみに追従し、C と A は連結しているので A は C の折りたたみに追従するという事態が生じ、折りたたみの一貫性を保つことができなくなる。したがって穴の開いた形状は多面体パーツごとに折りたたむことは可能だが、多面体パーツ間の衝突が発生するため、折りたたむことができない。

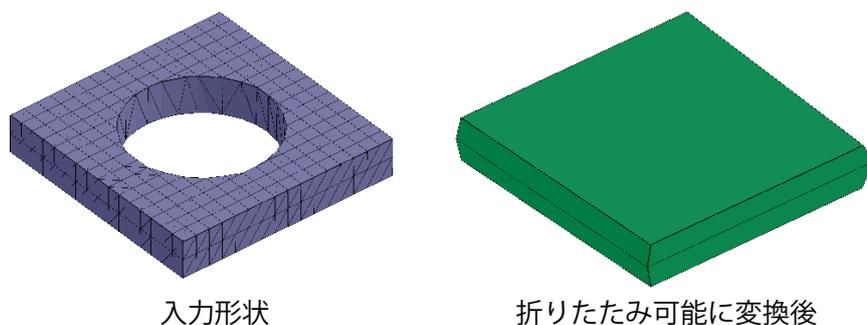


図 5.4: 穴の開いた入力形状の例と折りたたみ可能に変換した結果

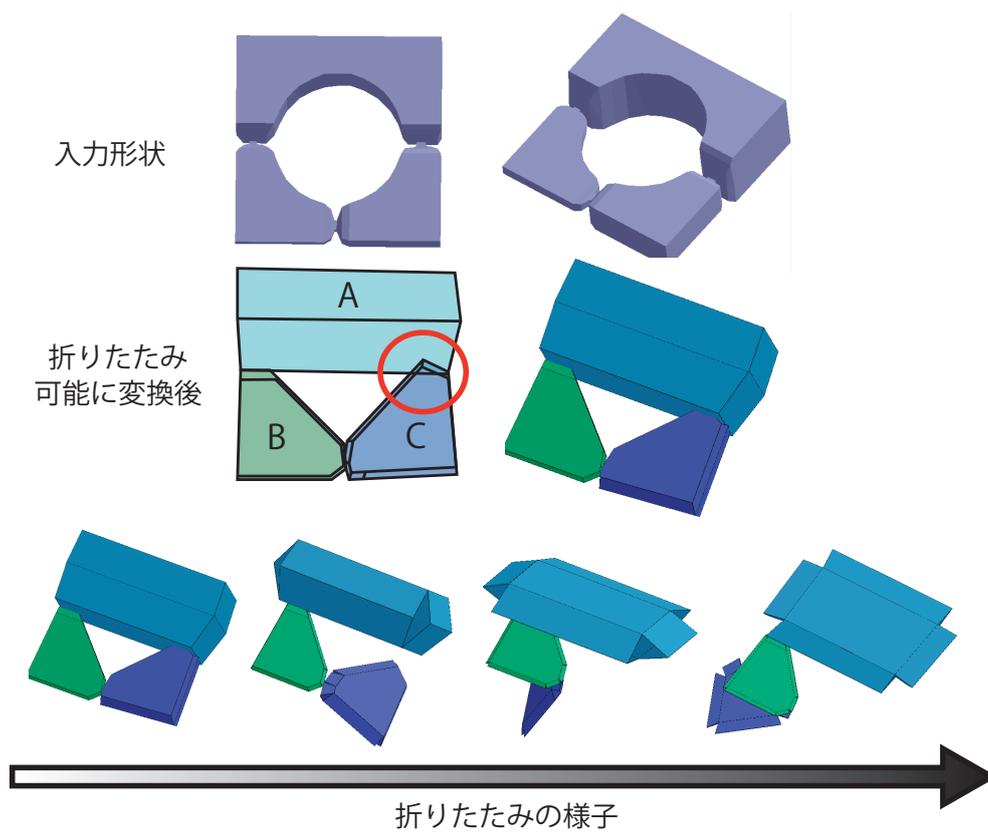


図 5.5: 多面体パーツ間の衝突が生じた立体の例

## 5.5 対称性を考慮した立体の近似精度

対称、非対称形状の結果を比較した様子を図 5.6 に示す。図 5.6 の (a) は対称性を考慮した折りたたみ可能な立体で、(b) は考慮せず出力したものである。(b) に比べ、(a) のほうが均整のとれた形状をしていることが見て取れる。特に青や青紫の多面体パーツは (b) の場合、左右で形が異なるため、(a) よりも形が崩れているような印象を与える。定量的な比較を行うため、各多面体パーツの凸パーツとの近似誤差と計算時間を計測した。その結果を表 5.4 に示す。表をみると、誤差の平均に大きな差はない。一方計算時間をみてみると、多面体パーツの生成に必要な時間は対称形状である方が全体的に短いことがわかる。これは左耳、左足などのパーツは反対側のパーツを反転、複製することで生成され、計算時間が実質 0 であるということ、断面線の最適化を省くことができることが理由である。これらの結果から、対称性を考慮した多面体パーツの近似精度における定量的な有効性を示すことはできないものの、左右対称な立体の生成が可能となり、多面体パーツの変換に必要な計算時間も削減できる。

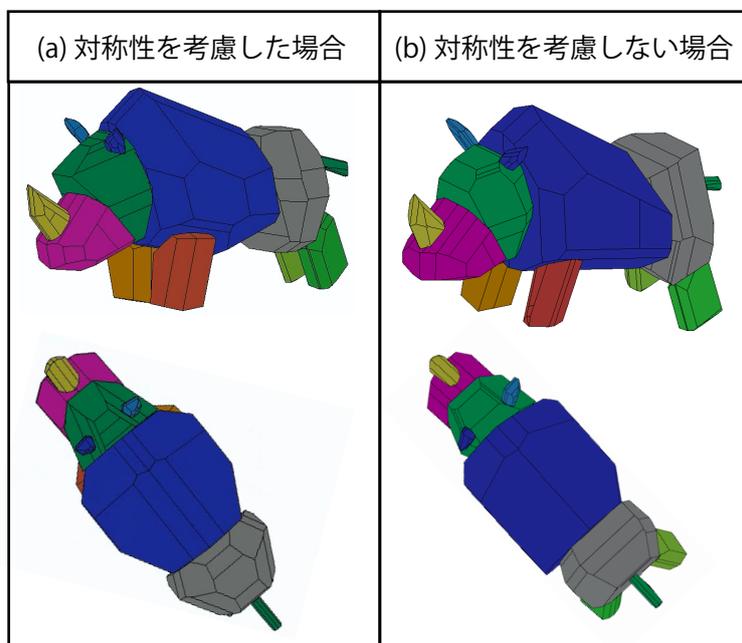


図 5.6: 対称性を考慮した折りたたみ可能な立体の比較

### 5.5.1 折りたたみ方向が及ぼす影響

折りたたみ方向は対称面に対して垂直または平行な方向を選択することが可能である。この 2 種類の折りたたみ方向によって生成される多面体パーツを図 5.7 に示す。図の (a) は対称面に対して平行な折りたたみ方を選択しており、(b) は折りたたみ方向に対して垂直な折りたたみ方向を選択している。Rhinoceros の場合、(b) のほうが (a) に比べ近似精度が高く、計算

表 5.4: Rhinoceros の近似誤差と計算時間

モデル	ハウズドルフ距離		計算時間 (sec)	
	対称	非対称	対称	非対称
尾	0.037	0.060	1.18	2.434
右耳	0.047	0.048	2.960	3.682
左耳	0.043	0.034	-	2.324
角	0.047	0.034	1.28	2.262
後右足	0.097	0.103	2.12	4.914
後左足	0.100	0.103	-	7.816
前右足	0.113	0.065	2.634	2.246
前左足	0.104	0.129	-	6.418
あご	0.087	0.092	3.593	6.552
腰	0.180	0.163	8.751	28.143
頭	0.156	0.172	10.595	21.609
体	0.212	0.259	4.423	19.126
平均	0.102	0.105	3.128	8.961

時間も少ない。これは角の曲面部分に天頂面が割り当てられていることが原因である。一方 Teapot の場合、(a) の方が (b) に比べ近似精度が高い。これはふたの部分が薄い板状の立体であり、かつ軸対称性をもつ円形状であることが原因である。一方計算時間を比較すると、(a) よりも (b) の方が計算時間は短い。これは (b) の場合、断面線を最適化する必要がないためである。これらのことから、対称面に対して垂直な方向を選択した方が計算時間を省くことが可能であり、およそその多面体パーツはより良い近似精度であるが、薄い板状の凸パーツや軸対称性を持つ場合、対称面に対して平行な折りたたみ方向を選択した方がよい。

### 5.5.2 既存手法との比較

図 5.8 は加瀬らによる手法 (文献 [1]) と提案手法を用いて作成した折りたたみ可能な立体の例である。中図の立体は入力形状 (左) から折りたたみ方向や近似精度を調整するパラメータをユーザーが指定した後にシステムが自動的に出力した結果であり、右図の立体は入力形状にできるだけ似せるよう、加瀬らの手法を用いて設計した。設計に必要な時間はそれぞれ約 20 分、50 分程度である。加瀬らの手法は設計の自由度が大きいいため、ある程度の時間で尤もらしい結果が得られるが、接続パネルの配置がうまくいかない場合、多面体パーツを設計し直す必要がある。一方本手法では接続パネルの配置が自動で行われ、パラメータは頂点数を操作するパラメータ、折りたたみ方向のみであるため、ユーザーの負担が少ない。しかし、入力形状を自動的に折りたたみ可能な立体へ変換するため、参照形状を必要とする。

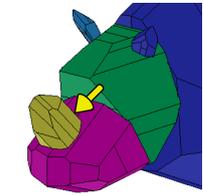
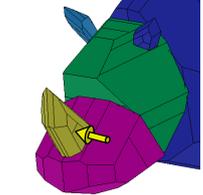
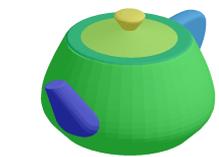
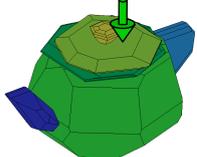
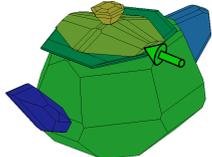
	凸パーツ	(a)	(b)
			
ハウズドルフ距離		0.115	0.047
計算時間 (sec)		1.934	0.952
			
ハウズドルフ距離		0.070	0.089
計算時間 (sec)		3.306	2.367

図 5.7: 対称性を考慮した折りたたみ可能な立体の折りたたみ方向による形状の変化. (a) 対称面に対して平行な折りたたみ方向. (b) 対称面に対して垂直な折りたたみ方向. 近似誤差はハウズドルフ距離をバウンディングボックスの最大の辺の長さで割ったものである.

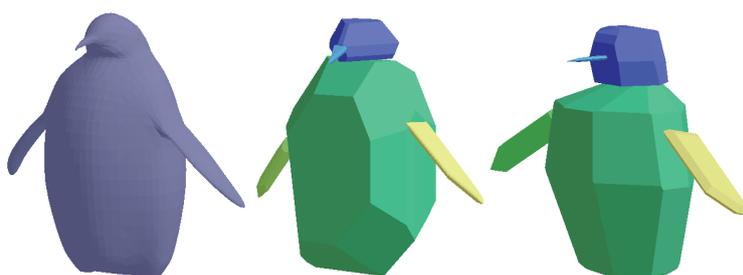


図 5.8: 既存手法と提案手法との比較. (左) 入力形状. (中) 本手法を用いて生成した折りたたみ可能な立体. (右) 既存手法を用いて手動で生成した折りたたみ可能な立体.

## 5.6 今後の課題

- 本手法で出力されるのは折りたたみ可能な立体の三次元モデルであり、展開図の生成と、のりしろの編集などは既存のツールを用いている。そのため、ユーザーは多面体パーツを組み立てる際にどのパネルをどのように接続すべきか、折りたたみのアニメーションを確認しなければならない。これはユーザーにとって負担となる。この問題を改善するため展開図を出力し、さらに接続部分を示すことが必要である。
- 本手法で生成した折りたたみ可能な立体は結果で示したように、実際の紙を用いて作成することが可能である。しかし、コンピュータ上では折りたたみ可能であったとしても、実際に模型として出力すると、組み立てや折りたたみが困難な場合がある。例えば断面線の二頂点間の距離が非常に短いと結果として得られるパネルの面積が非常に小さくなる。このようなパネルが接続パネルとなる場合、実際に多面体パーツを組み立てることが困難である。この問題を解決するためには、制作時の大きさを参考にして各辺やパネルの大きさに制約を加える必要がある。
- 第 5.5 節で述べたように凸制約によって再現可能な形状が制限される。より自由な表現を可能とするため、凸制約を取り払う必要がある。凸制約を設けることの利点は隣り合う多面体パーツの干渉を回避できること、多面体パーツの最適化の問題をよりシンプルにできることである。これらの問題を十分に考慮することで凸制約を取り払うことが可能だと考えられる。
- 本手法で出力した折りたたみ可能な立体は紙などの薄い素材であれば、実際に出力することが可能である。しかし、厚みのある木材などのパネルは折りたたむことができない。それは折りたたむ際に隣り合うパネルの端が衝突するためである。また、折りたたむことは容易であるが、立ち上げた際の形状を保つことが非常に難しい。様々な素材に対応するため、折りたたみ可能な立体のスケールや厚みを考慮する必要がある。
- 本手法は折りたたみ方向と近似精度を調節するパラメータのみをユーザーが入力するものとし、それ以外はシステムがすべて自動で行う。これはユーザーの負担を軽減できる一方、設計できる形状の自由度が低いという問題がある。例えば本手法は参照モデルを必要とする。また、多面体パーツは自動的にされるため、ユーザーが編集することはできない。より自由な設計を可能とするため、多面体パーツの生成後に形状を修正できるよう、対話的なインターフェースの実装が必要である。
- 折りたたみ可能な立体は多面体パーツそれぞれが折りたたみ可能であっても、パーツ間の衝突が生じる場合がある。また、図 5.9 のように、折りたたむ順番によっては衝突を回避できる。たとえば図 5.9 の上段のように Bunny の耳を一度に折りたたむと、耳同士が衝突し折りたたむことができない。しかし下段のように耳を別々に折りたたむと衝突が生じない。衝突を回避するため、折りたたむ順番や多面体パーツの形状を最適化する必要がある。

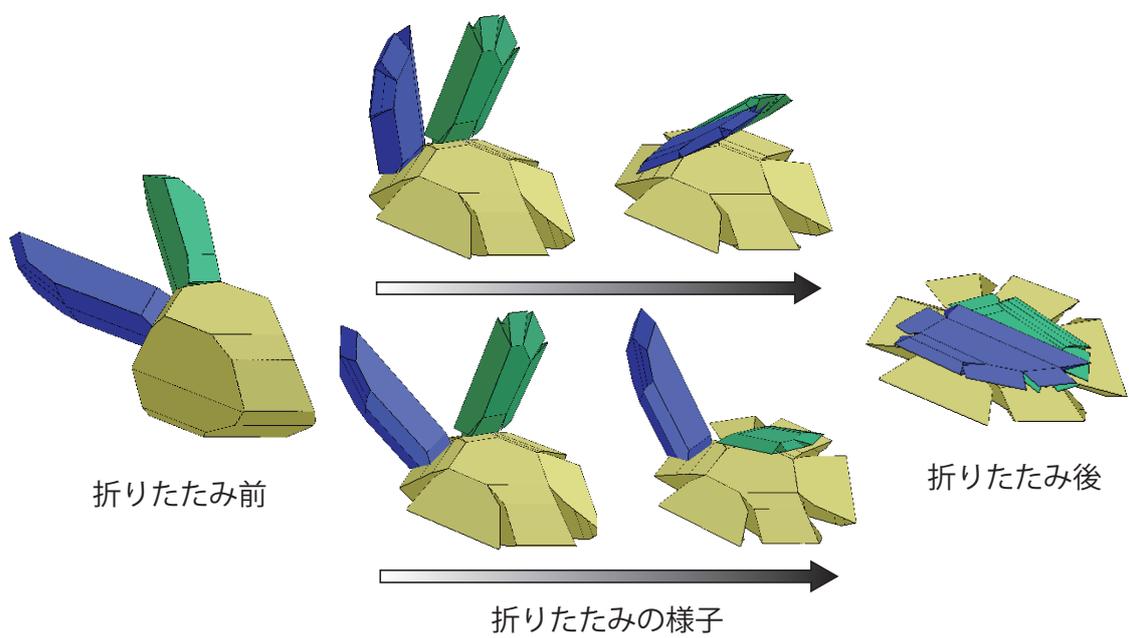


図 5.9: 多面体パーツを一度に折りたたむ場合と別々に折りたたむ場合の比較

## 謝辞

本研究を進めるにあたり，協力した頂いたすべての方々に感謝いたします．特に研究や論文の執筆についてご指導いただいた，三谷純教授，金森由博准教授，遠藤結城助教授に深謝いたします．常日頃から支えて下さった研究室の皆様，友人，家族へ心からの感謝の気持ちと御礼を申し上げたく，謝辞にかえさせていただきます．

## 参考文献

- [1] Yuto Kase, Yoshihiro Kanamori and Jun Mitani. A method for designing flat-foldable 3D polygonal models. *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 5B, pp. DETC2015-46566, 2015.
- [2] Jun Mitani and Hiromasa Suzuki. Computer aided design for Origamic Architecture models with polygonal representation. *Proceedings of the Computer Graphics International*, pp. 93-99, July 2004.
- [3] Satoshi Iizuka, Yuki Endo, Jun Mitani, Yoshihiro Kanamori and Yukio Fukui. An interactive design system for pop-up cards with a physical simulation. *The Visual Computer (Proceeding of Computer Graphics International 2011)*, Vol. 27, No. 6, pp. 605-612, April 2011.
- [4] Sosuke Okamura and Takeo Igarashi. An Interface for Assisting the Design and Production of Pop-Up Card. *Proceedings of the 10th International Symposium on Smart Graphics*, No. 11, pp. 68-78, 2009.
- [5] Xian-Ying Li, Chao-Hui Shen, Shi-Sheng Huang, Tao Ju and Shi-Min Hu. Popup: Automatic Paper Architectures from 3D Models. *ACM Transactions on Graphics*, Vol. 29, No. 4, pp. 111:1-9. July 2010.
- [6] Xian-Ying Li, Tao Ju, Yan Gu and Shi-Min Hu. A Geometric Study of V-style Pop-ups: Theories and Algorithms. *ACM Transactions on Graphics*, Vol. 30, No. 4, pp. 98:1–10. July 2011.
- [7] Conrado R. Ruiz, Sang N. Le, Jinze Yu and Kok-Lim Low. Multi-style Paper Pop-up Designs from 3D Models. *Computer Graphics Forum*, Vol. 33, No. 2, pp. 487-496. May 2014.
- [8] Conrado R. Ruiz, Sang N. Le and Kok-Lim Low. Generating animated paper pop-ups from the motion of articulated characters. *The Visual Computer*, Vol. 31, No. 6, pp. 925-935. June 2015.
- [9] Jun Mitani. ORI-REVO:A Design Tool for 3D Origami of Revolution. [http://mitani.cs.tsukuba.ac.jp/ori\\_revo](http://mitani.cs.tsukuba.ac.jp/ori_revo), (Dec. 12, 2017).

- [10] Erik D. Demaine and Tomohiro Tachi. Origamizer: A Practical Algorithm for Folding Any Polyhedron. Proceedings of the 33rd International Symposium on Computational Geometry, July 2017.
- [11] Robert J. Lang. Treemaker. <http://www.langorigami.com/science/computational/treemaker/treemaker.php>, (Dec. 16, 2017)
- [12] 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.
- [13] Devin J. Balkcom, Erik D. Demaine, Martin L. Demaine, John A. Ochsendorf and Zhong You. Folding Paper Shopping Bags. In Abstracts from the 14th Annual Fall Workshop on Computational Geometry, pp. 14-15, November, 2004.
- [14] ミウラ折りラボミウラ折り公式サイト. <http://www.miuraori.biz>, (Dec. 16, 2017).
- [15] Tine De Ruysser. Wearable Metal Origami. Origami 6, pp. 613-624, 2016.
- [16] Koryo Miura Method of Packaging and Deployment of Large Membranes in Space. The Institute of Space and Astronautical Science report, pp. 1-9, 1985.
- [17] Tomohiro Tachi and Koryo Miura. Rigid-foldable cylinders and cells. Journal of the International Association for Shell and Spatial Structures, Vol. 53, pp. 217-226, December 2012.
- [18] Tomohiro Tachi. Geometric Considerations for the Design of Rigid Origami Structures. Proceedings of the International Association for Shell and Spatial Structures (IASS) Symposium, January 2010.
- [19] Sicong Liu, Yan Chen and Guoxing Lu. The Rigid Origami Patterns for Flat Surface. ASME. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 6B, August 2013.
- [20] Yves Klett. Realtime Rigid Folding Algorithm for Quadrilateral-Based 1-DOF. Proceedings of the ASME Design Engineering Technical Conference, Vol. 6, August 2013.
- [21] Tomohiro Tachi. Simulation of Rigid Origami. Origami 4: Fourth International Meeting of Origami Science, Mathematics, and Education, Vol. 4, August 2009.
- [22] Tomohiro Tachi. Rigid-Foldable Thick Origami. Origami5, pp. 253-263, 2011.
- [23] Jason S. Ku and Erik D. Demaine. Folding Flat Crease Patterns with Thick Materials. ASME. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2015.

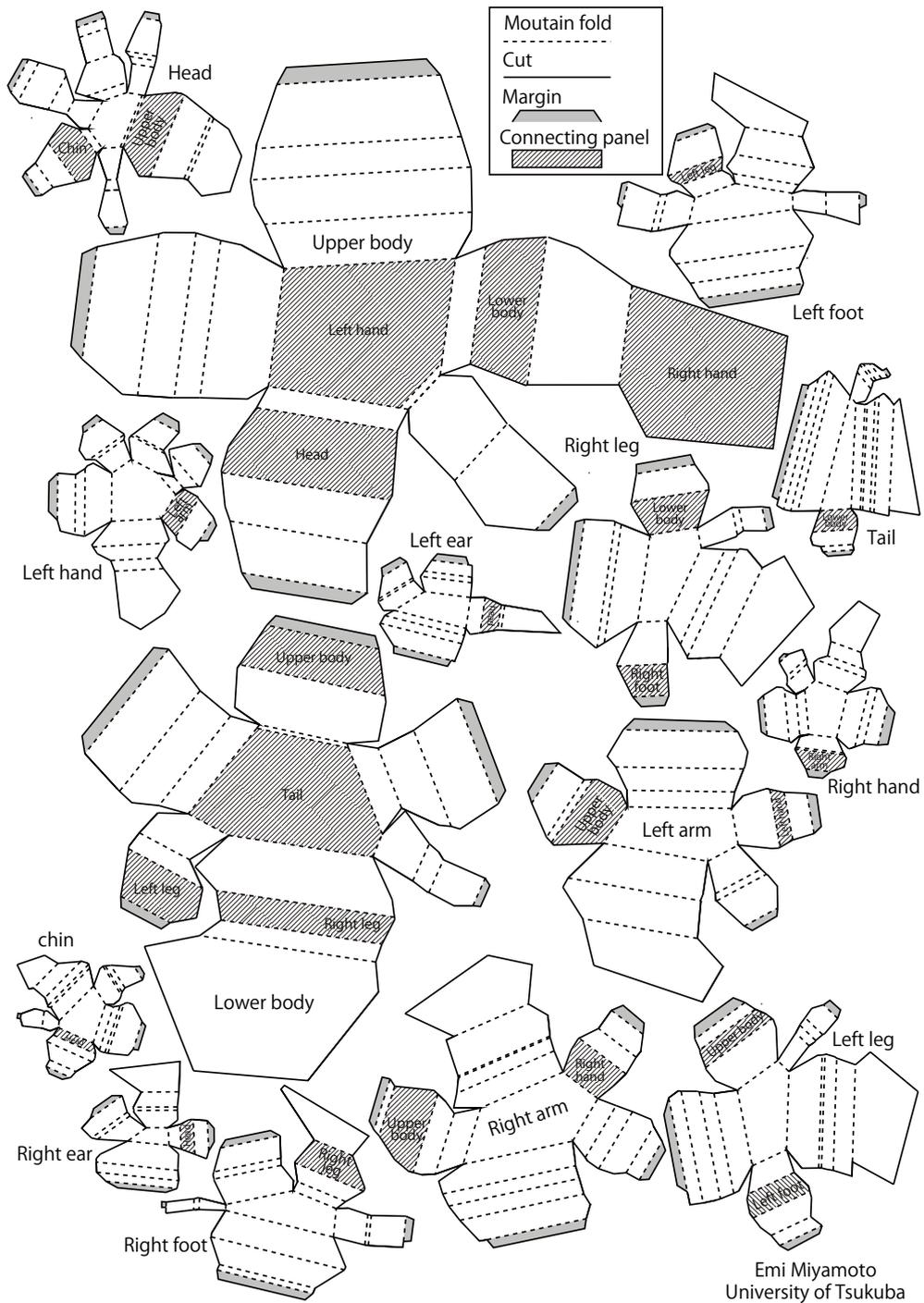
- [24] Honghua Li, Ruizhen Hu, Ibraheem Alhashim and Hao Zhang. Foldabilizing Furniture. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2015)*, Vol. 34, No. 4, 2015.
- [25] Mohamed Ibrahima and Dong-Ming Yanb. Fold and Fit: Space Conserving Shape Editing. *Computers and Graphics 2017*, Vol. 70, pp. 316-326, October 2017.
- [26] Bongjin Koo, Wilmot Li, JiaXian Yao, Maneesh Agrawala and Niloy J. Mitra. Creating Works-Like Prototypes of Mechanical Objects. *ACM Transactions on Graphics (Special issue of SIGGRAPH Asia 2014)*, 2014.
- [27] Tianjia Shao, Wilmot Li, Kun Zhou, Weiwei Xu, Baining Guo and Niloy J. Mitra. Interpreting Concept Sketches. *ACM Transactions on Graphics*, Vol. 32, No. 4, pp. 56:1–56:10, July 2013.
- [28] Shapira Lior, Shamir Ariel and Cohen-Or Daniel. Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer*, Vol. 24, No. 4, pp. 249-259, March 2008.
- [29] Christopher Dyken, Morten Dahlen and Thomas Sevaldrud. Simultaneous curve simplification. *Journal of Geographical Systems*, Vol 11(3), pp. 273-289, September 2009.
- [30] C. Bradford Barber, David P. Dobkin and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, Vol 22, No. 4, pp. 469-483, December 1996.
- [31] The Computational Geometry Algorithms Library. <https://www.cgal.org>, (Dec. 08, 2017).
- [32] 多摩ソフトウェア有限公司. ペパクラデザイナー. <http://www.tamasoft.co.jp/pepakura>, (Dec. 15, 2017).
- [33] カミカラ. ペンギン爆弾. <http://kamikara.cart.fc2.com/ca4/36/p-r-s>, (Jan. 6, 2018).

## 付録 A

出力した紙模型の作成例と折りたたみの様子，そして展開図を示す。



図 A.1: Armadillo の紙模型を折りたたむ様子



Emi Miyamoto  
University of Tsukuba

図 A.2: Armadillo の紙模型の展開図

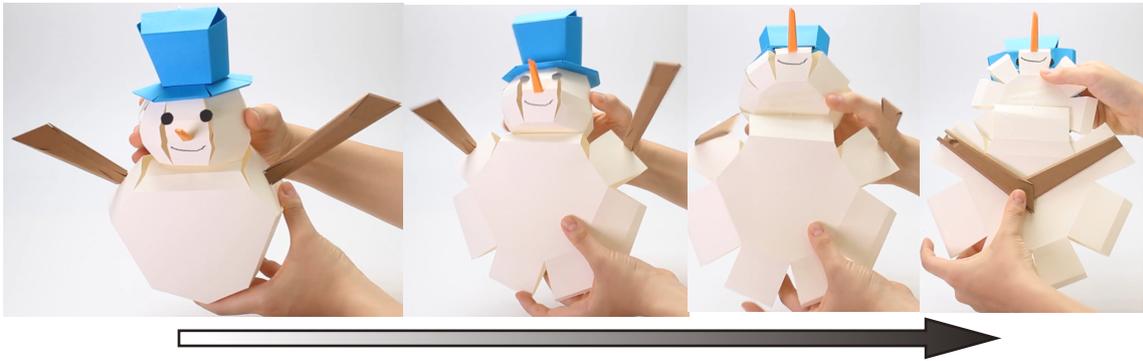


図 A.3: Snowman の紙模型を折りたたむ様子

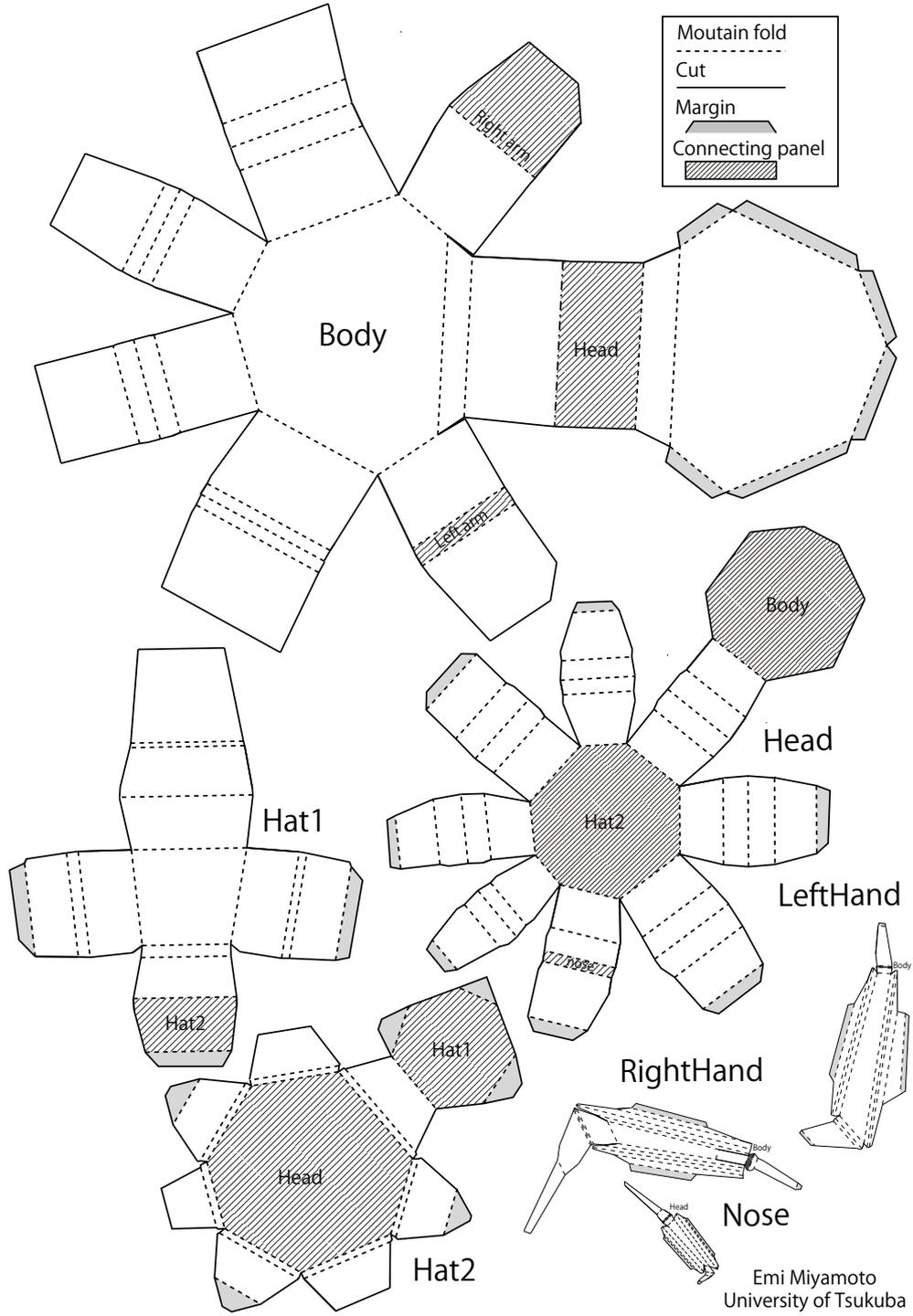


図 A.4: Snowman の紙模型の展開図