

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

帯状三次元座標取得装置を用いた曲線折り形状  
の対話的な生成インタフェースの開発

上村 紳一郎

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

紙のような伸縮しない素材を曲げることで可展面と呼ばれる曲面を作ることができ、それに折りを加えることで、曲線での折りを持つ形状を作ることができる。しかしながら、紙を折ることによって得られる形状の設計においては、直線での折りだけを対象とすることが多く、曲線での折りを扱ったものは少ない。曲線での折りは、平面曲線での折りと、空間曲線での折りに大別できるが、とくに空間曲線での折りについては、その扱いの難しさから、これまであまり形状設計において対象とされてこなかった。理論的側面からは、微分幾何の分野で曲線折りの研究は広く行われており、曲線と折り角の情報が与えられれば、その曲線を折り線とする形状が数学的に導出可能である。しかしながら、そもそも空間曲線を自在に入力することが困難であることから、この理論を実際の紙を折って作る造形に活かすことは、これまでに行われてこなかった。本論文では、ShapeTape という帯状三次元座標取得装置を用いることで、空間曲線を対話的に入力し、この空間曲線を折り線とする形状をリアルタイムに生成、可視化するためのシステムの研究開発を行った。このようにして得られた形状の展開図を出力することで、その形状を実際に作ることができる。この一連の手法により、紙を折り曲げながら行う従来の試行錯誤による形状設計プロセスを大幅に効率化できることを確認した。提案システムでは、使用する紙の形が折り線を中心とした一定幅に限定されるなどの制約や、紙の自己交差を考慮しないなどの制約があるため、これらを改善することで、より実用性を高めることができると期待される。

# 目次

<b>第 1 章</b>	<b>序論</b>	<b>1</b>
1.1	研究背景 . . . . .	1
1.2	研究目的 . . . . .	1
1.3	関連研究 . . . . .	2
1.3.1	直線折りと曲線折り . . . . .	2
1.3.2	曲面を扱った折り紙 . . . . .	2
1.3.3	折り畳み可能な構造 . . . . .	2
1.3.4	可展面 . . . . .	3
1.4	本論文の構成 . . . . .	3
<b>第 2 章</b>	<b>用語の説明</b>	<b>4</b>
2.1	ShapeTape . . . . .	4
2.1.1	ShapeTape の機能 . . . . .	4
2.1.2	ShapeTape の精度 . . . . .	4
2.2	剛体折り構造 . . . . .	5
2.3	可展多面体 . . . . .	6
<b>第 3 章</b>	<b>曲線折りモデル生成のための提案手法</b>	<b>7</b>
3.1	折れ線の生成 . . . . .	7
3.1.1	ShapeTape の出力座標をそのまま使用する手法 . . . . .	7
3.1.2	近似曲線を使用する手法 . . . . .	9
3.2	帯状の可展多面体の生成 . . . . .	10
3.3	曲線折り形状の生成 . . . . .	11
<b>第 4 章</b>	<b>結果</b>	<b>13</b>
4.1	評価 . . . . .	15
4.1.1	評価手法 . . . . .	15
4.1.2	評価に対する考察 . . . . .	21
<b>第 5 章</b>	<b>考察</b>	<b>24</b>
<b>第 6 章</b>	<b>結論と今後の課題</b>	<b>25</b>
6.1	結論 . . . . .	25

6.2	今後の課題	25
6.2.1	新たな帯状の可展多面体の生成手法の発見	25
6.2.2	出力形状を矩形に切り取る機能の開発	25
6.2.3	ポリゴンの交差判定	25
	謝辞	26
	参考文献	27

# 図目次

1.1	折り畳み可能な椅子 画像出典 [18]	1
1.2	曲線折りを生じたパッケージ 画像出典 [19]	1
2.1	ShapeTape と 30cm 定規	4
2.2	上図: ShapeTape を湾曲させた様子 下図: 計算機に出力されたデータを可視化した様子	5
2.3	人工衛星だいち 2 号 画像出典 [20]	5
2.4	ミウラ折りの例 画像出典 [21]	6
3.1	各頂点の概要図。生成フローの (8) に到達し $i = 2$ である場合を示している。赤線は生成した折れ線である。	8
3.2	各頂点の概要図。生成フローの (4) に到達し $c = 3$ である場合を示している。灰線はスプライン曲線、赤線は生成した折れ線である。	9
3.3	左図: 入力折れ線の例 右図: 左図の入力から生成した帯状の多面体の模式図	10
3.4	帯状の可展多面体とそれを構成するベクトル。この図は可展多面体を平面に展開したものである	10
3.5	接線ベクトル $\mathbf{T}_i$ 、主法線ベクトル $\mathbf{N}_i$ 、従法線ベクトル $\mathbf{B}_i$ の関係	11
3.6	左図: 帯状の可展多面体の例 中央図: 左図の可展多面体の一部を右回転・左回転させたもの 右図: 中央図の結果から左図の可展多面体を取り除いたもの	12
4.1	ShapeTape を変形させて曲線折り形状を生成し、ディスプレイに出力している様子	13
4.2	生成した曲線折り形状の展開図をリアルタイムにディスプレイに表示している様子	13
4.3	左図: 提案手法で生成した曲線折り形状 右図: 左図の曲線折り形状からペパクラデザイナー [22] を用いて展開図を生成した様子	14
4.4	図 4.5 左図の曲線折り形状をカッティングプロッタに入力して紙に出力したもの	14
4.5	左図: 提案手法で生成した曲線折り形状をペパクラデザイナーに入力して生成した展開図 右図: Rigid Origami Simulator[23] に左図のデータを入力して出力した結果	15
4.6	評価に用いた ShapeTape の出力データ	16

4.7	ShapeTape の出力その 1 から (1) の方法で生成した折れ線で生成した帯状の可展多面体 (左図) と曲線折り形状 (右図) . . . . .	16
4.8	ShapeTape の出力その 1 から (2) の方法で生成した折れ線で生成した帯状の可展多面体 (左図) と曲線折り形状 (右図) . . . . .	17
4.9	ShapeTape の出力その 1 から (3) の方法で生成した折れ線で生成した帯状の可展多面体 (左図) と曲線折り形状 (右図) . . . . .	17
4.10	ShapeTape の出力その 1 から (4) の方法で生成した折れ線で生成した帯状の可展多面体 (左図) と曲線折り形状 (右図) . . . . .	17
4.11	ShapeTape の出力その 2 から (1) の方法で生成した折れ線で生成した帯状の可展多面体 (左図) と曲線折り形状 (右図) . . . . .	18
4.12	ShapeTape の出力その 2 から (2) の方法で生成した折れ線で生成した帯状の可展多面体 (左図) と曲線折り形状 (右図) . . . . .	18
4.13	ShapeTape の出力その 2 から (3) の方法で生成した折れ線で生成した帯状の可展多面体 (左図) と曲線折り形状 (右図) . . . . .	19
4.14	ShapeTape の出力その 2 から (4) の方法で生成した折れ線で生成した帯状の可展多面体 (左図) と曲線折り形状 (右図) . . . . .	19
4.15	ShapeTape の出力その 3 から (1) の方法で生成した折れ線で生成した帯状の可展多面体 (左図) と曲線折り形状 (右図) . . . . .	20
4.16	ShapeTape の出力その 3 から (2) の方法で生成した折れ線で生成した帯状の可展多面体 (左図) と曲線折り形状 (右図) . . . . .	20
4.17	ShapeTape の出力その 3 から (3) の方法で生成した折れ線で生成した帯状の可展多面体 (左図) と曲線折り形状 (右図) . . . . .	21
4.18	ShapeTape の出力その 3 から (4) の方法で生成した折れ線で生成した帯状の可展多面体 (左図) と曲線折り形状 (右図) . . . . .	21
4.19	左図: 帯状の可展多面体の形状が崩れている場合の法線 (青線) と緑線 (従法線) の例 右図: 法線の向きが急激に変化している場所を拡大したもの . . . . .	22
4.20	左図: 図 4.19 と同じ入力で改善した生成フローで生成した帯状の可展多面体 右図: 左図から生成した曲線折り形状 . . . . .	23

# 第1章 序論

## 1.1 研究背景

曲線折り形状とは紙などの平面状で伸縮しない物体を曲線の折り線で折った折り形状である。図 1.1 は曲線折り形状を用いることで平坦に折り畳むことが可能な椅子である。また、図 1.2 は曲線折りを用いたパッケージである。このように曲線折り形状は立体的で見た目にも良い形状を生成しやすい性質を持つ。

曲線折り形状に関する研究には、入力が幾何学的に表せられる空間曲線であったり [1]、折り目が平面に乗るような空間曲線に限定されていたり [2] と、曲線折りを生成する過程で何かしらの制約があるものがほとんどである。その理由として空間曲線を自在に扱うことが難しく任意の空間曲線を折り線とした曲線折りの設計は今まで行われてこなかったことが挙げられる。



図 1.1: 折り畳み可能な椅子 画像出典 [18]

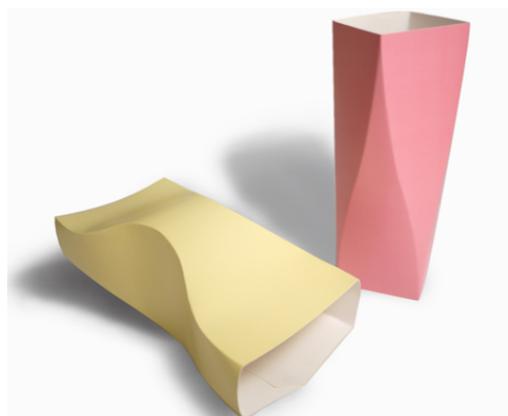


図 1.2: 曲線折りを用いたパッケージ 画像出典 [19]

## 1.2 研究目的

そこで本研究では、任意の空間曲線を入力して曲線折り形状を直感的かつ対話的に生成するインタフェースを開発・提案することで、曲線折り形状に関する研究の一助とする。本研究では ShapeTape という帯状三次元座標取得装置から出力される列状の三次元座標を用いて

折れ線を生成し、それを折り線とした曲線折り形状を文献 [3] を参考にして生成する。生成される曲線折り形状とその展開図は計算機を用いてリアルタイムに可視化される。生成される曲線折り形状をカッティングプロッタを用いて紙に出力したり、一般に公開されているソフトウェアを用いて折り畳む様子をアニメーションで表示する方法も紹介する。

## 1.3 関連研究

### 1.3.1 直線折りと曲線折り

折り紙の折り方は直線折りと曲線折りに大別できる。直線折りとは折り線が直線で表される折り方であり、一般的に知られている殆どの折り紙作品にはこの折り方である。それに対して曲線折りとは折り線が曲線で表される折り方であり直線折りと比べて立体的な形状を生成しやすい。曲線折りを扱った研究には平面を曲線折りした際のサーフェイスの設計 [4] や幾何学的に表された曲線からねじれを持った曲線折りを生成する手法 [1] などがある。直線折りは面の回転に置き換えられるので幾何学的に表現しやすく、直線折りを計算機上で再現した技術は、折り紙の対話的操作を実現したソフトウェア [5] や折りパターンから折り畳みをシミュレートするソフトウェア [6] などがあり、ユーザーが対話的に操作できるものが多い。それに対し曲線折りを計算機上で再現した技術は鏡面反射を利用した曲線折りを対話的に生成するソフトウェア [2] やユーザーの入力によってサーフェイスを分割しながら折りを計算機上で再現する手法 [7] などがあるが、どれも空間曲線を入力にした手法は存在しなかった。本研究では空間曲線を入力にした対話的な曲線折り生成インタフェースという新しい手法を提案する。

### 1.3.2 曲面を扱った折り紙

曲面あるいは曲面を近似した面を扱った折り紙の研究には、一枚の紙を折り畳むことで任意の 3D モデルに近似した多面体構造を生成する研究 [8] や紙などの平面の塑性変形をシミュレートした研究 [9]、湾曲エネルギーに基づく可展面の離散化を提案した研究 [10] などがある。これらは平面を入力として曲面を生成しているが本研究は曲線を入力として曲面を生成する。

### 1.3.3 折り畳み可能な構造

折り畳み可能な構造物は、使用しない時は小さく折り畳んで収納し使用する際に展開するというような省スペースかつ搬送性に優れた性質を持っており、工業的に有用なため様々な研究がされてきた。その例として、紙の買い物袋が平坦に折りたためるか議論した研究 [11] や、家具の 3D モデルを省スペースかつ元の 3D モデルと比べて修正量の少ない折り畳み可能な形状を自動で生成する手法 [12]、任意の 3D オブジェクトを切開面を含んだ一方向に折りたためる形状を対話的に生成する手法 [13] が挙げられる。

### 1.3.4 可展面

可展面とは展開すると平面になる形状をいう。可展面はその性質から工業的に扱いやすく有用であり、提案手法で生成する曲線折り形状もこの可展面である。可展面に関する研究には、二本の空間曲線の間を離散的に補間して可展面を生成する手法 [14] や折り畳み可能・不可能な幾何学的形状の分類と非可展開面を可展面にする最適化手法の提案 [15]、ミウラ折りを基にした曲線折りの提案 [16] が挙げられる。

## 1.4 本論文の構成

第1章では研究背景・目的と関連研究を述べた。第2章では本稿で使用する用語の説明、第3章では本研究の提案手法、第4章では結果と評価、第5章は考察、第6章では結論と今後の課題を述べる。

## 第2章 用語の説明

### 2.1 ShapeTape

#### 2.1.1 ShapeTape の機能

ShapeTape とは本研究で提案するインタフェースの入力に用いる装置である。この装置の帯状の部品には等間隔にセンサーが搭載されており、そのセンサーは直方体の部品と帯状の部品の接続部分を原点とした相対的な三次元座標とねじれの情報を出力する(図 2.1 参照)。本研究ではインタフェースの入力である折り線の生成に ShapeTape を用いるので、ShapeTape の出力は三次元座標のみを利用する。ShapeTape の出力する三次元座標を計算機上で可視化すると図 2.2 のようになる。



図 2.1: ShapeTape と 30cm 定規

#### 2.1.2 ShapeTape の精度

ShapeTape の測定精度は文献 [17] で報告されている。これによると、ShapeTape をなるべくねじれないように扱った場合、三次元座標の平均誤差が最大約 80mm 発生しているが、標準偏差は最大約 14mm である。提案手法では、生成した曲線折り形状はリアルタイムに確認できるため、測定した三次元座標のばらつきが小さいならば、平均誤差は多少の大きさであってもインタフェース使用者の希望と大きく外れた形状になってしまう問題は起きにくいと考えられる。

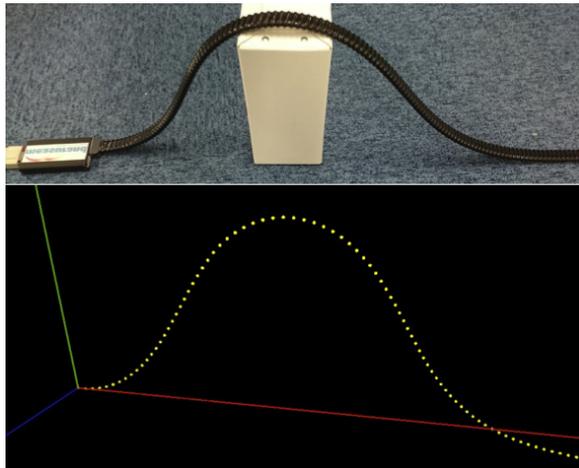


図 2.2: 上図：ShapeTape を湾曲させた様子 下図：計算機に出力されたデータを可視化した様子

## 2.2 剛体折り構造

剛体折りとは折り紙の構造のうち、展開・折り畳みにおける全ての変形が折り線上の回転によって起きている構造をいう。つまり、折り紙の面を固いパネルに、折り線をヒンジと見立てても同じように変形可能な構造である。本研究の提案手法で生成する曲線折り形状は四角形の面で構成された剛体折り構造である。剛体折り構造の例として有名なのは人工衛星の太陽光パネル(図 2.3 参照)やミウラ折り(図 2.4)などが挙げられる。本研究の提案手法で生成する曲線折り形状は剛体折り構造である。



図 2.3: 人工衛星だいち 2 号 画像出典 [20]

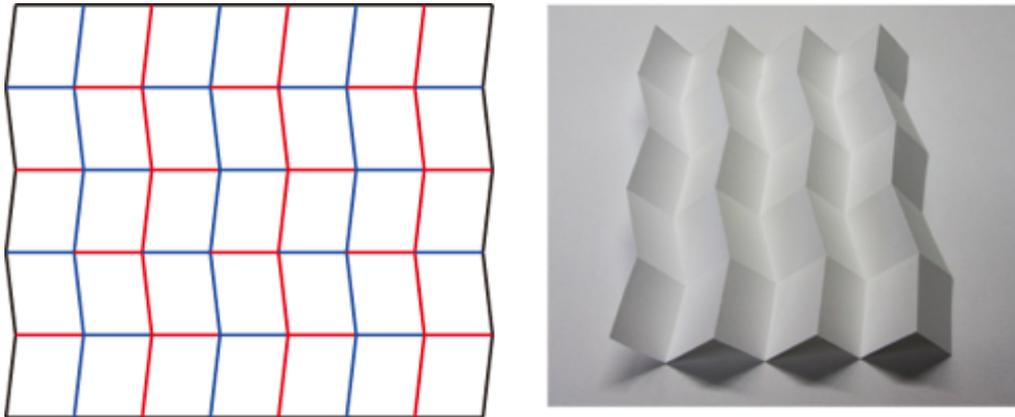


図 2.4: ミウラ折りの例 画像出典 [21]

### 2.3 可展多面体

可展とは伸縮することなしに平面の上に展開できることを指す。多面体とは複数の平面に囲まれた立体を指す。提案手法では曲線折り形状を生成する過程において展開すると長方形になる帯状のポリゴンの集合体を生成する。これを可展多面体と本稿では表現する。帯状のポリゴンの集合体は閉じた立体ではないが、便宜上多面体という単語を使用した。

## 第3章 曲線折りモデル生成のための提案手法

提案手法の入力は折り線・折り角度・出力の曲線折り形状の幅であり出力は離散化されてポリゴンで表現された曲線折り形状である。本研究では ShapeTape の出力する列状の三次元座標群から折れ線を生成し、その折れ線から文献 [3] で提案されている手法を用いて展開すると長方形になる帯状の多面体を生成し、その多面体の各面を折り線を軸として軸回転させることで曲線折り形状を生成する。本研究では入力は滑らかな曲線ではなく曲線に近似した折れ線である。

### 3.1 折れ線の生成

提案手法の折れ線を生成する過程は、節 3.2 にて生成される帯状の可展多面体が形崩れを起こさないように考慮したそれが複数存在する。これについては節 4.1 にて詳しく述べている。ここでは折れ線の生成過程の内基本的なものを二つ述べる。

#### 3.1.1 ShapeTape の出力座標をそのまま使用する手法

ShapeTape の出力する三次元座標から適当なものを間引いてそれから折り線を生成する。この手法は ShapeTape の出力そのものを使用しており計算が容易であるメリットがある。以下に折れ線の頂点  $Q_j$  の生成フローを示す (各頂点の概要は図 3.1 を参照)

- (1) ShapeTape で入力した座標  $x_n (n : 0 \sim 80)$  と折れ線を構成する線分  $l_m$  と各変数  $i$  (初期値は 1),  $a$  (初期値は 1),  $b$  (初期値は 1) を定義
- (2)  $Q_0 = x_0, Q_1 = x_1, x_0$  と  $x_1$  を結ぶ線分を  $l_0$ 、 $l_0$  が向く方向の単位ベクトルをベクトル  $\mathbf{t}_0$  と定義
- (3)  $x_1$  と  $x_{1+i}$  を結ぶ線分を  $l_1$ 、 $l_1$  が向く方向の単位ベクトルをベクトル  $\mathbf{t}_1$  と定義
- (4)  $\mathbf{t}_0$  と  $\mathbf{t}_1$  の内積が 0.99 より小さければ (5) へ、そうでなければ  $i = i + 1$  して (3) へ
- (5)  $Q_2 = x_{1+i}, a = a + i, b = b + 1, i = 1$  とする
- (6)  $x_a$  と  $x_{a+i}$  を結ぶ線分を  $l_{b+1}$ 、 $l_{b+1}$  が向く方向の単位ベクトルをベクトル  $\mathbf{t}_{b+1}$  と定義
- (7)  $\mathbf{t}_b$  と  $\mathbf{t}_{b+1}$  の内積が 0.99 より小さければ (8) へ、そうでなければ  $i = i + 1$  して (6) へ
- (8)  $Q_{b+1}, a = a + i, b = b + 1, i = 1$  とする
- (9) (6)~(8) を  $a + i = 80$  になるまで繰り返す

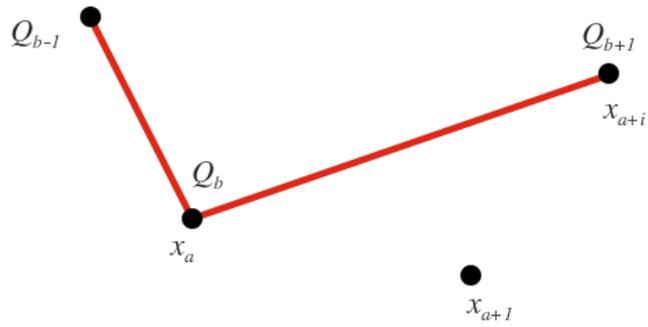


図 3.1: 各頂点の概要図。生成フローの (8) に到達し  $i = 2$  である場合を示している。赤線は生成した折れ線である。

### 3.1.2 近似曲線を使用する手法

ShapeTape の出力する三次元座標のうち、間の座標数が一定になるようにそれを取り、その各々の三次元座標を制御点とした  $N$  次のスプライン補間を行う ( $N \geq 3$ )。  $N$  の値は 3, 5, 7 の場合だと数値振動が起きず滑らかな曲線が生成できる。生成したスプライン曲線を等間隔に分割し、分割点を折れ線の頂点とする。この手法は ShapeTape による入力からパラメタ曲線を介して折り線を生成しているため、折り線となる折れ線の頂点を柔軟に設定できるメリットがある。以下に折れ線の頂点  $Q_k$  の生成フローを示す。

- (1) ShapeTape で入力した座標  $x_n (n : 0 \sim 80)$  と折れ線を構成する線分  $l_m$  と各変数  $i$  (初期値は 1),  $a$  (初期値は 1),  $b$  (初期値は 1) を定義
- (2)  $x_n (n : 0 \sim 80)$  から間隔が一定になるようにそれを取り、それらの座標を  $P_j (j : 0 \sim (80/m) - 1m : \text{間隔})$  と定義する。(例:  $P_0 = x_0, P_1 = x_4, P_2 = x_8, P_3 = x_{12} \dots$ )
- (3)  $P_j$  の x 座標を制御点としたスプライン補間を行い媒介変数  $t$  によって表される導関数  $Sx_j(t)$  を算出する。y 座標と z 座標に対しても同じ計算を行い  $Sy_j(t), Sz_j(t)$  を算出する
- (4) 各の  $Sx_j(t), Sy_j(t), Sz_j(t)$  において  $t$  に  $\frac{1}{c}, \frac{2}{c} \dots \frac{c-1}{c}$  ( $c$ : 分割数) を代入した結果を折り線の頂点  $Qx_k, Qy_k, Qz_k (k : 0 \sim (m * c) + 1)$  とする。(例  $Qx_0 = Sx_0(0), Qx_1 = Sx_0(0.25), Qx_2 = Sx_0(0.5), Qx_3 = Sx_0(0.75), Qx_4 = Sx_1(0)$ )
- (5) 前項に記載したフローに  $Q_k$  を入力して  $a + i = (m * c) + 1$  になるまで計算する

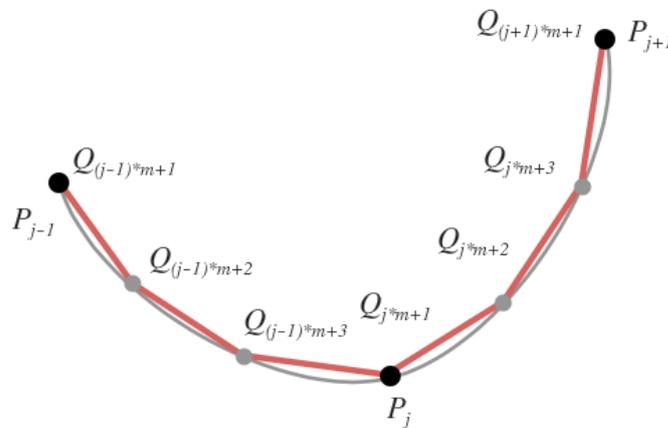


図 3.2: 各頂点の概要図。生成フローの (4) に到達し  $c = 3$  である場合を示している。灰線はスプライン曲線、赤線は生成した折れ線である。

### 3.2 帯状の可展多面体の生成

前節で生成した折れ線を測地線とした帯状の可展多面体を以下に示す手法で生成する。図 3.3 はその模式図である。文献 [3] で提案されている手法を参考にして、折れ線の各線分に接

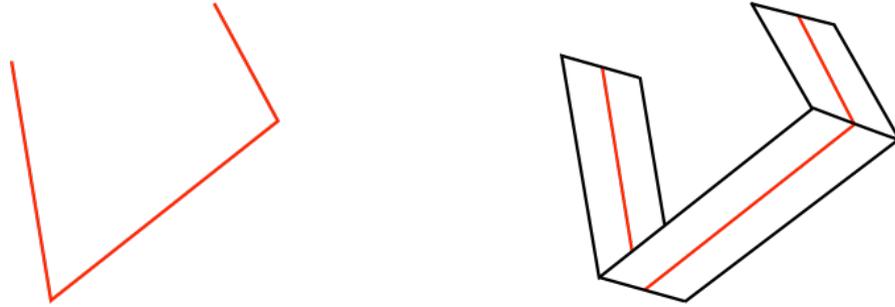


図 3.3: 左図：入力折れ線の例 右図：左図の入力から生成した帯状の多面体の模式図

線ベクトル  $\mathbf{T}$ 、主法線ベクトル  $\mathbf{N}$ 、従法線ベクトル  $\mathbf{B}$ 、境界ベクトル  $\mathbf{r}$  を定義・算出する (図 3.4, 3.5 参照)。境界ベクトルとは帯状の可展多面体を構成するポリゴンの辺のうち、二つのポリゴンの辺である、かつ入力の折れ線でないものをいう。これらのベクトルは三次元ベクトルであり、ベクトル  $\mathbf{r}$  は大きさを持ち、その他のベクトルは単位ベクトルである。これらのベクトルを基に帯状の可展多面体を生成する。

各ベクトルは下記の数式で示した条件を満たす。 $W$  は生成される帯状の可展多面体の幅の長

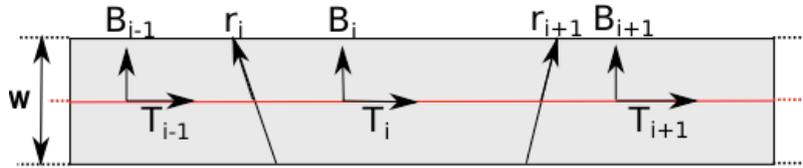


図 3.4: 帯状の可展多面体とそれを構成するベクトル。この図は可展多面体を平面に展開したものである

さ、記号  $\times$  はベクトルの外積を表している。

$$\begin{pmatrix} \mathbf{N}_i^T \\ \mathbf{T}_{i+1}^T - \mathbf{T}_i^T \\ \mathbf{B}_i^T \end{pmatrix} \mathbf{r}_{i+1} = \begin{pmatrix} 0 \\ 0 \\ \frac{W}{2} \end{pmatrix} \quad (3.1)$$

$$\mathbf{N}_{i+1} = \mathbf{r}_{i+1} \times \mathbf{T}_{i+1} \quad (3.2)$$

$$\mathbf{B}_{i+1} = \mathbf{T}_{i+1} \times \mathbf{N}_{i+1} \quad (3.3)$$

以下に各ベクトルの計算の流れを示す。

まずベクトル  $\mathbf{v}_i$  と行列  $\mathbf{A}$  を以下のように定義する。また、 $\mathbf{N}_i(x), \mathbf{N}_i(y), \mathbf{N}_i(z)$  はそれぞれ

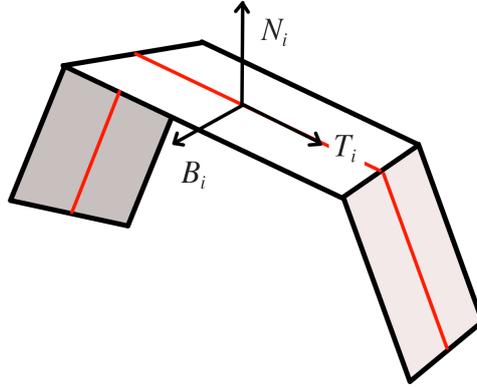


図 3.5: 接線ベクトル  $\mathbf{T}_i$ 、主法線ベクトル  $\mathbf{N}_i$ 、従法線ベクトル  $\mathbf{B}_i$  の関係

$\mathbf{N}_i$  の  $x, y, z$  座標と定義する。  $\mathbf{v}_i, \mathbf{B}_i$  においても同じように  $x, y, z$  座標を定義した。

$$\mathbf{v}_i = \mathbf{T}_{i+1} - \mathbf{T}_i \quad (3.4)$$

$$\mathbf{A} = \begin{pmatrix} \mathbf{N}_i(x) & \mathbf{N}_i(y) & \mathbf{N}_i(z) \\ \mathbf{v}_i(x) & \mathbf{v}_i(y) & \mathbf{v}_i(z) \\ \mathbf{B}_i(x) & \mathbf{B}_i(y) & \mathbf{B}_i(z) \end{pmatrix} \quad (3.5)$$

$\mathbf{A}$  の逆行列を式 3.1 の両辺の左側から掛けることで以下の式 3.6 を算出する。

$$\mathbf{r}_{i+1} = \mathbf{A}^{-1} \begin{pmatrix} 0 \\ 0 \\ \frac{W}{2} \end{pmatrix} \quad (3.6)$$

これにより求められた  $\mathbf{r}_{i+1}$  を式 3.2 に代入して  $\mathbf{N}_{i+1}$  を求め、  $\mathbf{N}_{i+1}$  を式 3.3 に代入することで  $\mathbf{B}_{i+1}$  を求める。これらを繰り返すことでベクトル  $\mathbf{N}, \mathbf{B}, \mathbf{r}$  を全て算出する。

### 3.3 曲線折り形状の生成

前項で求めた帯状の可展多面体を、入力折れ線を軸として軸回転させる。提案手法では、ベクトル  $\mathbf{T}$  の向きを前、ベクトル  $\mathbf{N}$  の向きを上とした場合の入力折れ線より右側の帯状の可展多面体を回転させる。右回転させた後の面を構成する境界ベクトルを  $\mathbf{r}'_i$ 、法線ベクトルを  $\mathbf{N}'_i$  と定義する。同様にして左回転させた後の面を構成する境界ベクトルと法線ベクトルをそれぞれ  $\mathbf{r}''_i, \mathbf{N}''_i$  と定義する (図 3.6 参照)。

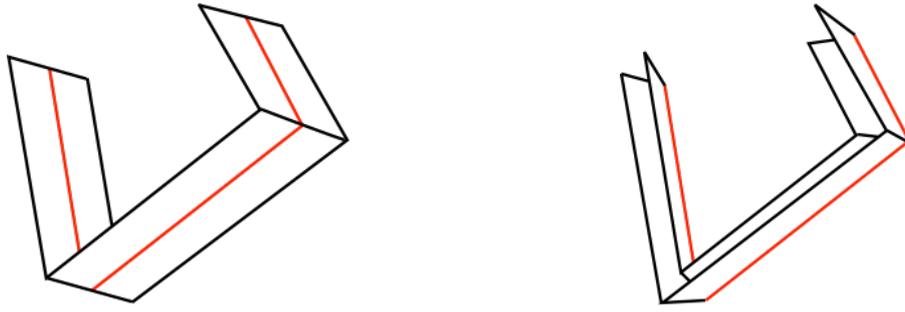


図 3.6: 左図: 帯状の可展多面体の例 中央図: 左図の可展多面体の一部を右回転・左回転させたもの 右図: 中央図の結果から左図の可展多面体を取り除いたもの

各面を回転させた後、隣り合う面が交差している場合ははみ出た部分を切り取り、離れている場合はお互いの面を拡張して交差した部分で切り取る。 $\mathbf{r}'_i$  と  $\mathbf{r}''_i$  は隣合った面の法線ベクトルの外積から求める (式 3.7,3.8)。

$$\mathbf{r}'_i = \mathbf{N}'_{i-1} \times \mathbf{N}'_i \text{ if } (\mathbf{r}'_{i-1} \cdot \mathbf{r}'_i < 0) \mathbf{r}'_i = -\mathbf{r}'_i \quad (3.7)$$

$$\mathbf{r}''_i = \mathbf{N}''_{i-1} \times \mathbf{N}''_i \text{ if } (\mathbf{r}''_{i-1} \cdot \mathbf{r}''_i < 0) \mathbf{r}''_i = -\mathbf{r}''_i \quad (3.8)$$

求めた  $\mathbf{r}'_i$  と  $\mathbf{r}''_i$  より回転した各面の頂点を算出し曲線折り形状が完成する。

## 第4章 結果

提案手法では、曲線折り形状は計算機によって計算されディスプレイにリアルタイムに表示される(図4.1参照)。インタフェース操作中はキーボードのテンキーを用いることで対話的に曲線折りの折り角度と出力の曲線折り形状の幅を指定できる。

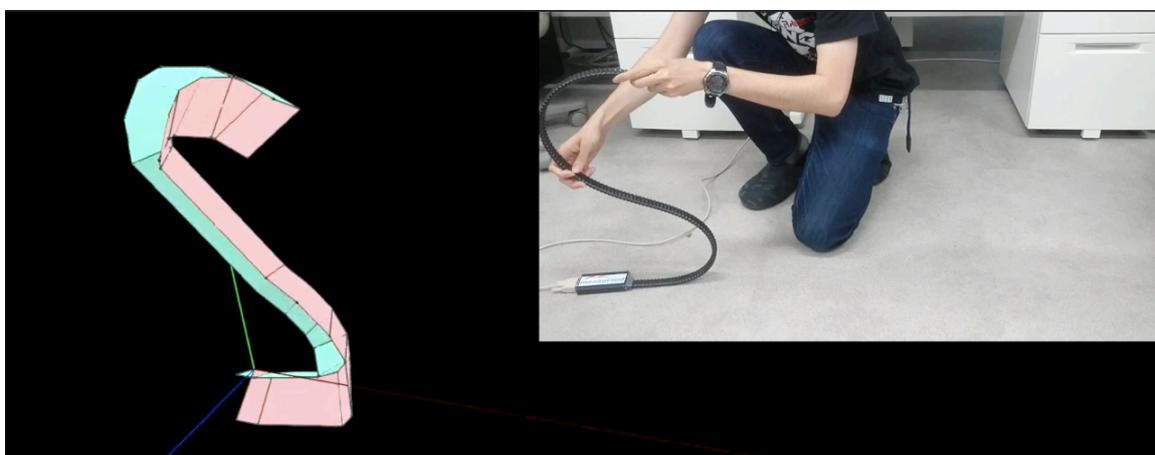


図 4.1: ShapeTape を変形させて曲線折り形状を生成し、ディスプレイに出力している様子

提案手法で生成した曲線折り形状の展開図はリアルタイムにディスプレイに出力可能である(図4.2参照)。

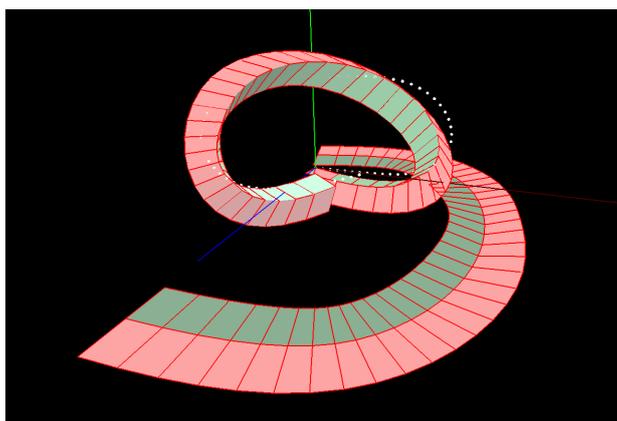


図 4.2: 生成した曲線折り形状の展開図をリアルタイムにディスプレイに表示している様子

また、生成した曲線折り形状はファイルに保存でき、一般に公開されているソフトウェアにそれを入力することで展開図を得ることも出来る(図 4.3 参照)。

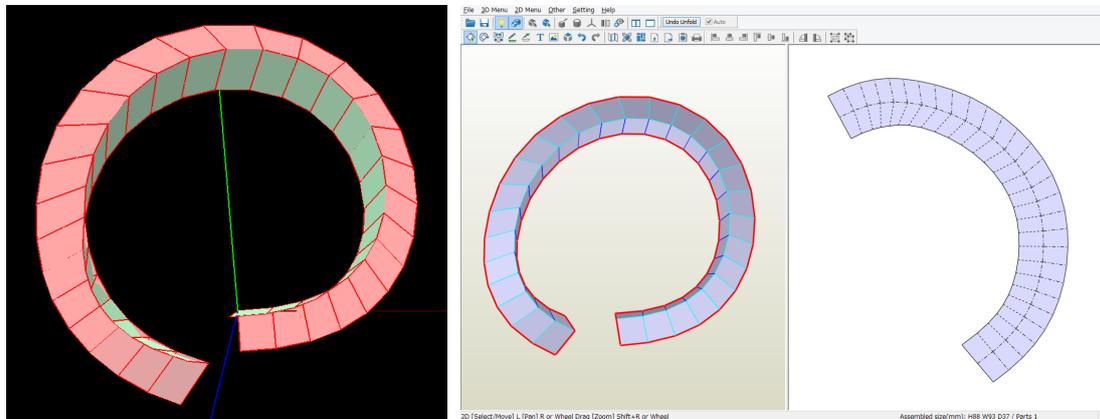


図 4.3: 左図: 提案手法で生成した曲線折り形状 右図: 左図の曲線折り形状からペパクラデザイナー [22] を用いて展開図を生成した様子

提案手法で生成した曲線折り形状のデータは、カッティングプロッタに入力することで曲線折り形状を紙に出力し実際に手で折り畳んだりして観察できる(図 4.4 参照)。



図 4.4: 図 4.5 左図の曲線折り形状をカッティングプロッタに入力して紙に出力したもの

一般に公開されているソフトウェアを用いることで生成した曲線折り形状がどのように展開・折り畳みがなされるかをアニメーションで観察できる。

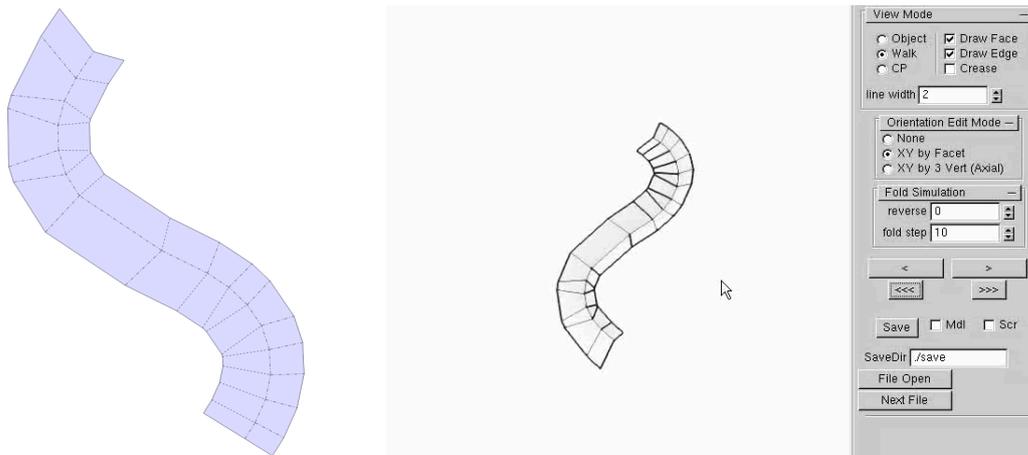


図 4.5: 左図: 提案手法で生成した曲線折り形状をペパクラデザイナーに入力して生成した展開図

右図: Rigid Origami Simulator[23] に左図のデータを入力して出力した結果

## 4.1 評価

結果より、実際に ShapeTape を用いて曲線折り形状を生成していると、ポリゴンが極端に大きかったりポリゴン同士の交差がある、つまり現実世界では再現できないような出力が生成されている時があることが判明した。この問題の原因を突き止め、美しい出力が得られるためにはどのような条件が必要なのかを探るために入力折れ線の違いによって出力がどのように異なるかを評価した。

### 4.1.1 評価手法

入力に用いる ShapeTape の出力データを 3 通り用意し (図 4.6)、同一の出力データから 4 通りの方法で折れ線を生成し曲線折りを生成して計 12 通りの曲線折り形状を生成した。評価は目視で行った。

折れ線は下記の 4 通りの方法で生成した。

- (1) ShapeTape の出力の三次元座標全てで折れ線を生成する
  - (2) 項 3.1.1 の方法で折れ線を生成する
  - (3) 項 3.1.2 の (1) から (4) までの方法で折れ線を生成する
  - (4) 項 3.1.2 の方法で折れ線を生成する
- (3)(4) に関しては項 3.1.2 の (2) において  $x_n$  から 16 ずつの間隔でとり、(4) において分割数  $c = 6$  とする。

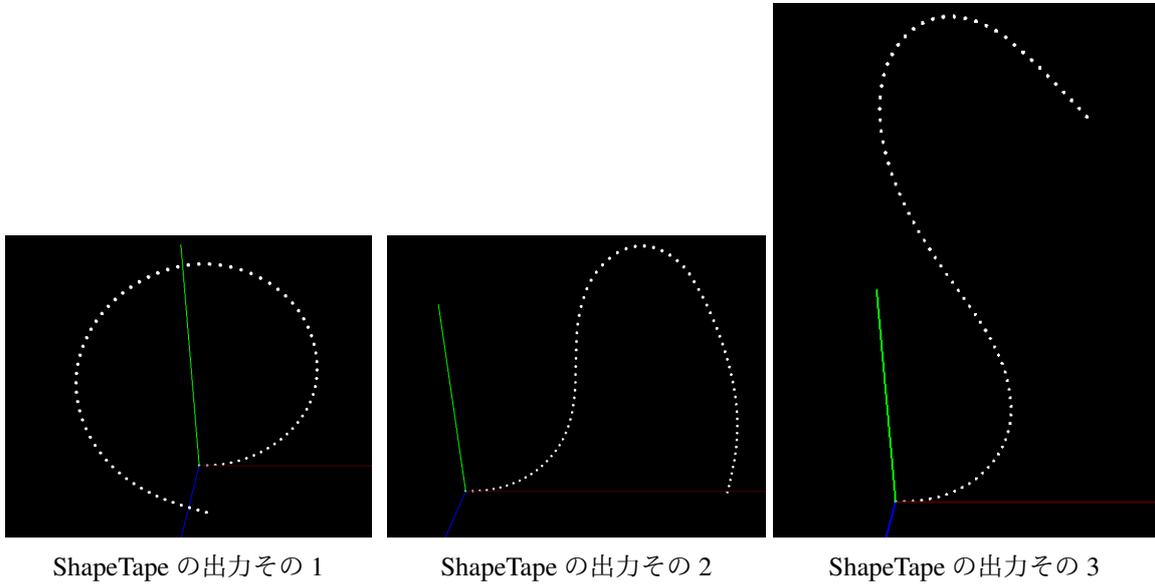


図 4.6: 評価に用いた ShapeTape の出力データ

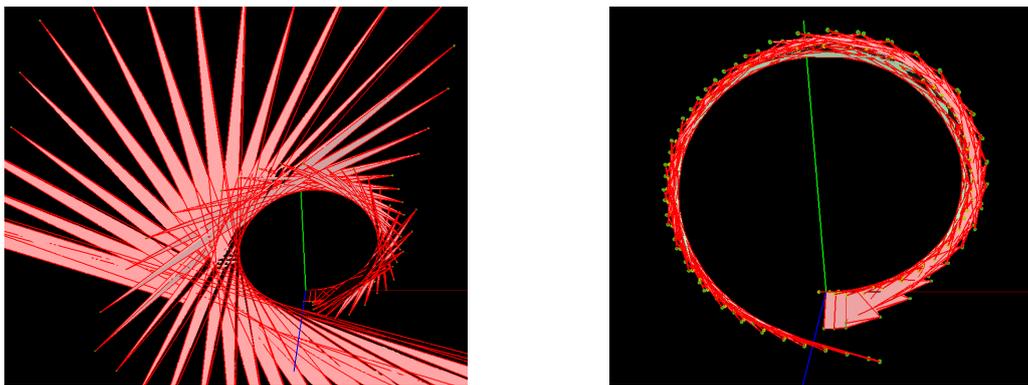


図 4.7: ShapeTape の出力その 1 から (1) の方法で生成した折れ線で生成した帯状の可展多面体 (左図) と曲線折り形状 (右図)

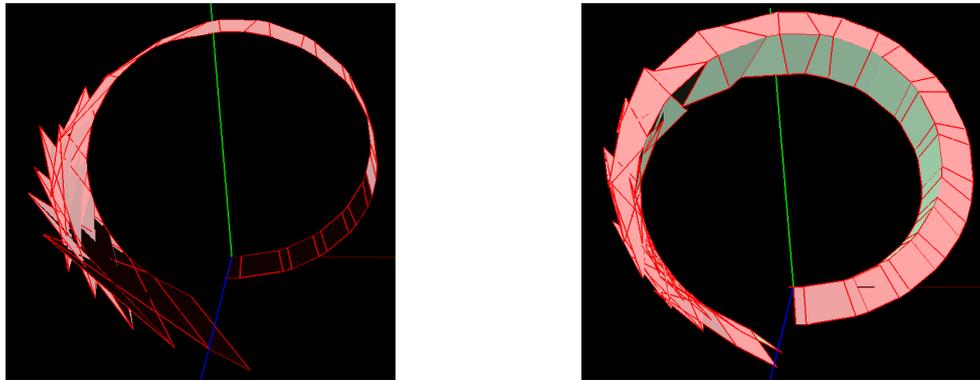


図 4.8: ShapeTape の出力その 1 から (2) の方法で生成した折れ線で生成した帯状の可展多面体 (左図) と曲線折り形状 (右図)

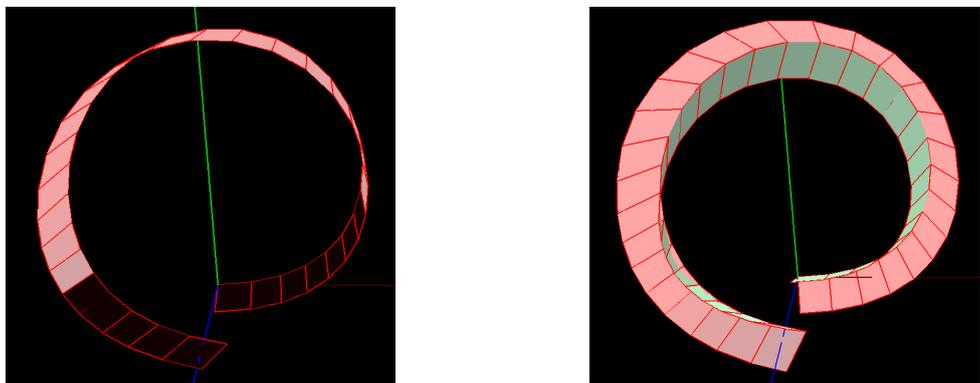


図 4.9: ShapeTape の出力その 1 から (3) の方法で生成した折れ線で生成した帯状の可展多面体 (左図) と曲線折り形状 (右図)

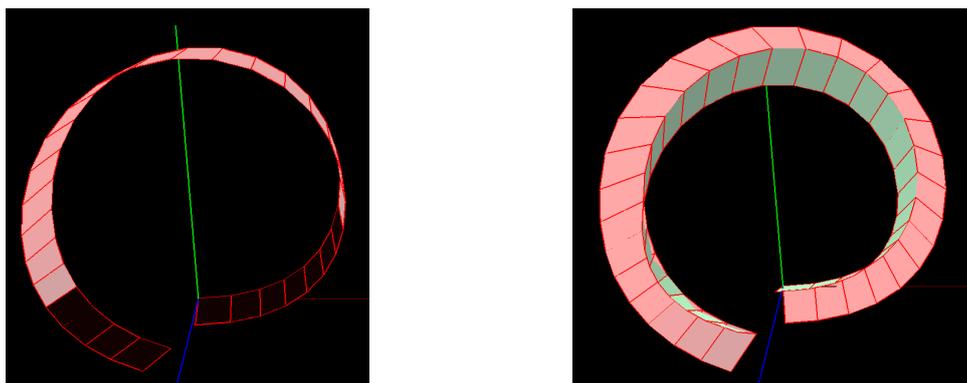


図 4.10: ShapeTape の出力その 1 から (4) の方法で生成した折れ線で生成した帯状の可展多面体 (左図) と曲線折り形状 (右図)

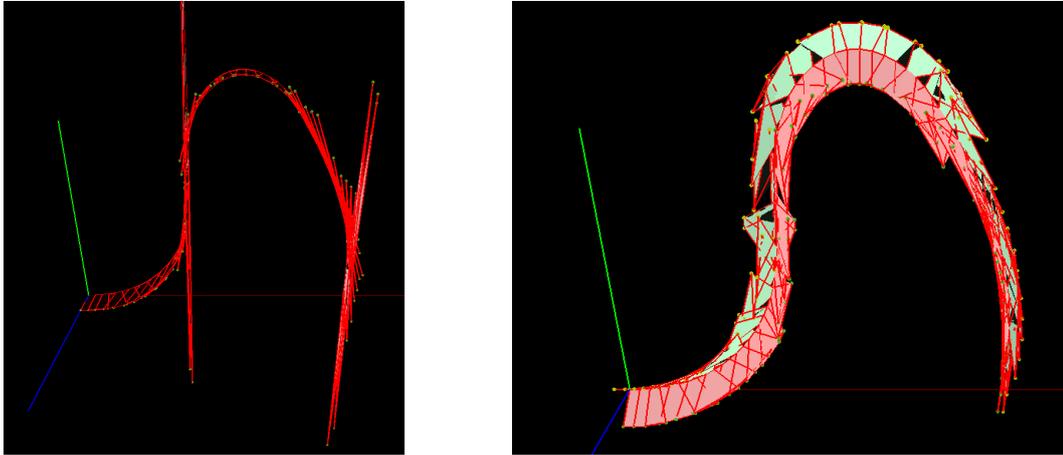


図 4.11: ShapeTape の出力その 2 から (1) の方法で生成した折れ線で生成した帯状の可展多面体 (左図) と曲線折り形状 (右図)

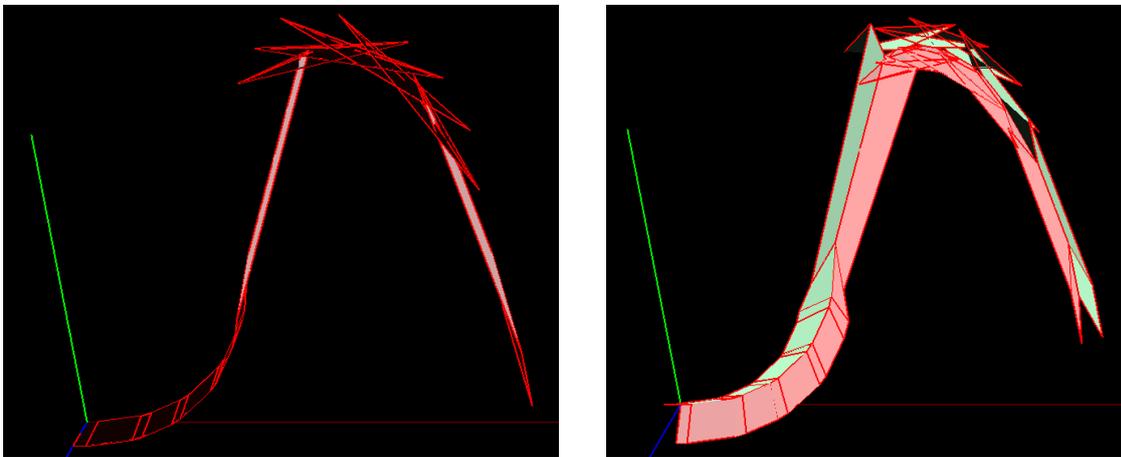


図 4.12: ShapeTape の出力その 2 から (2) の方法で生成した折れ線で生成した帯状の可展多面体 (左図) と曲線折り形状 (右図)

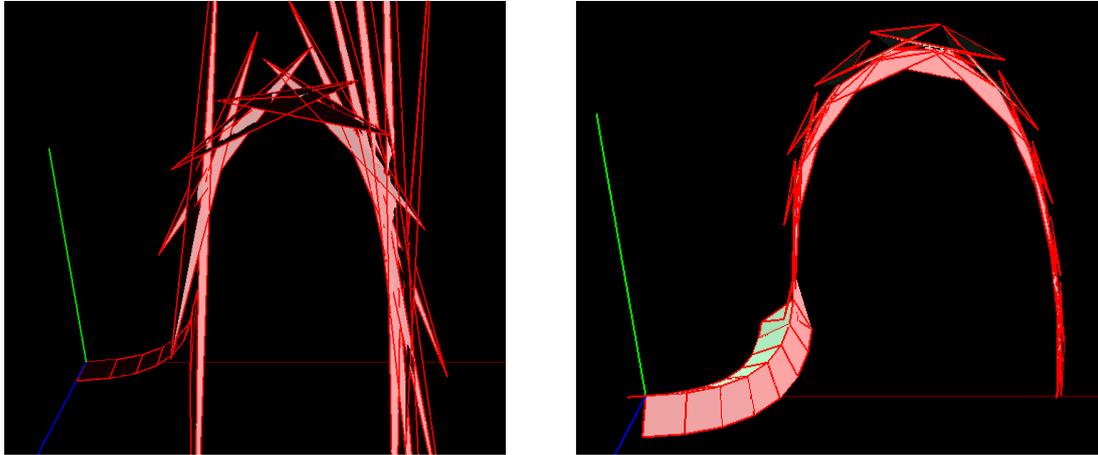


図 4.13: ShapeTape の出力その 2 から (3) の方法で生成した折れ線で生成した帯状の可展多面体 (左図) と曲線折り形状 (右図)

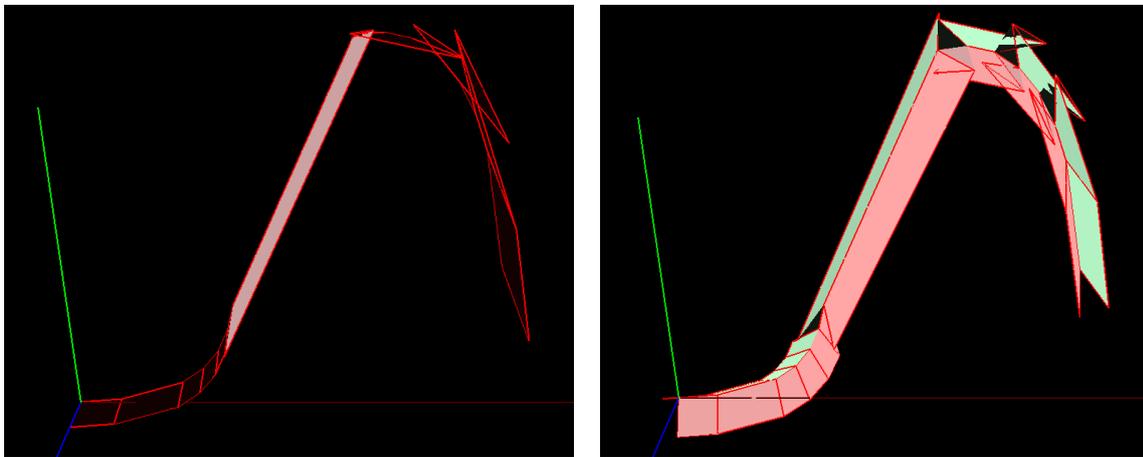


図 4.14: ShapeTape の出力その 2 から (4) の方法で生成した折れ線で生成した帯状の可展多面体 (左図) と曲線折り形状 (右図)

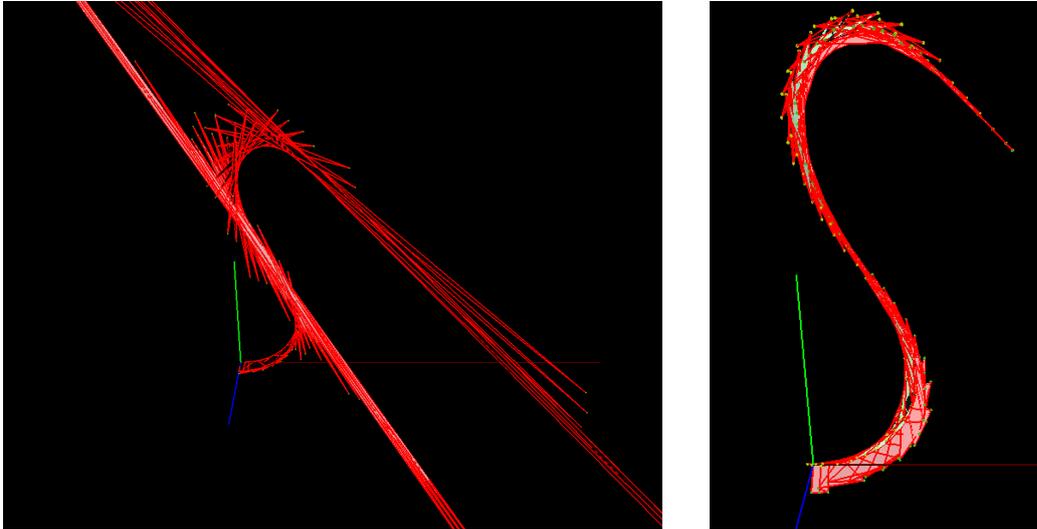


図 4.15: ShapeTape の出力その 3 から (1) の方法で生成した折れ線で生成した帯状の可展多面体 (左図) と曲線折り形状 (右図)

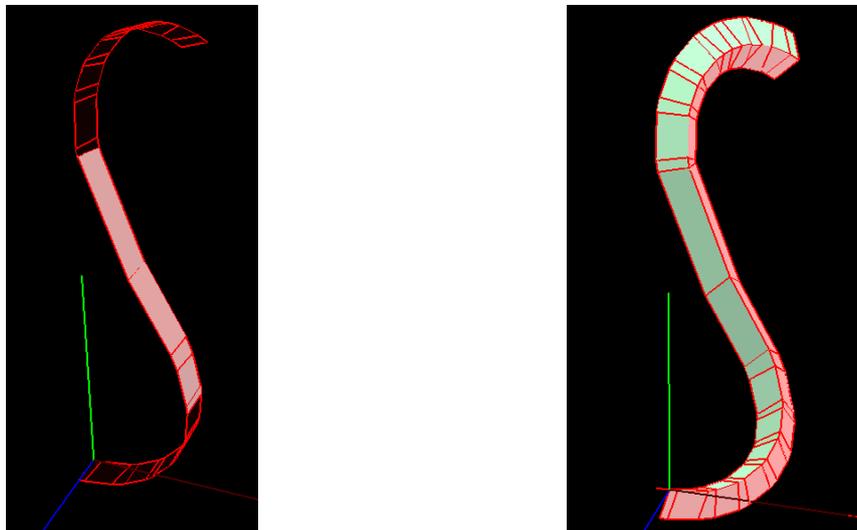


図 4.16: ShapeTape の出力その 3 から (2) の方法で生成した折れ線で生成した帯状の可展多面体 (左図) と曲線折り形状 (右図)

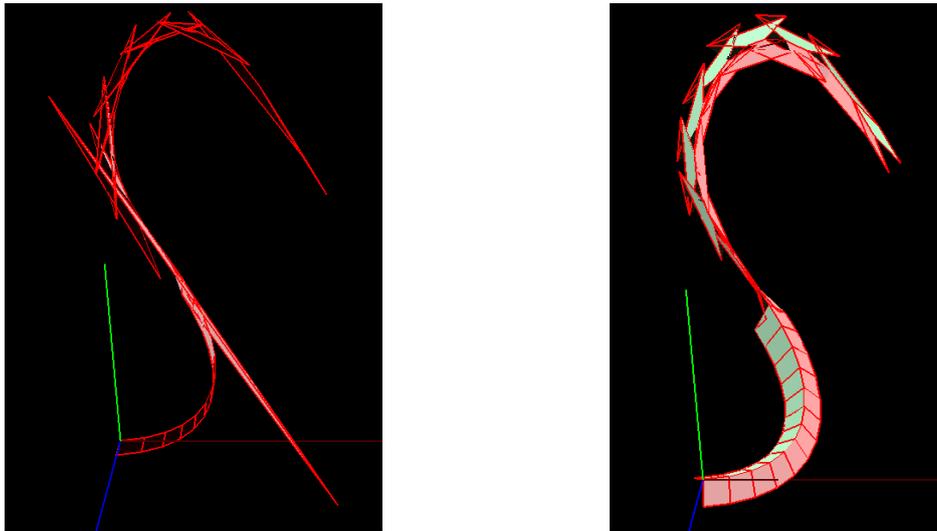


図 4.17: ShapeTape の出力その 3 から (3) の方法で生成した折れ線で生成した帯状の可展多面体 (左図) と曲線折り形状 (右図)

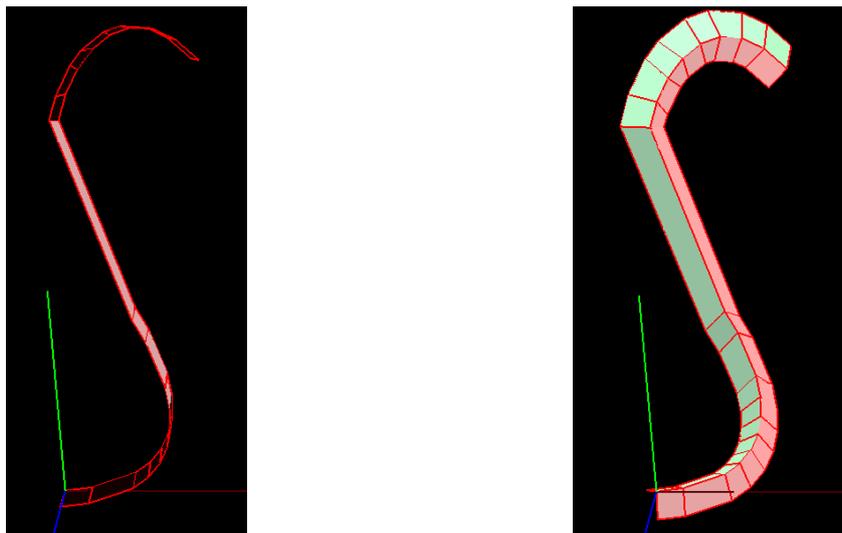


図 4.18: ShapeTape の出力その 3 から (4) の方法で生成した折れ線で生成した帯状の可展多面体 (左図) と曲線折り形状 (右図)

#### 4.1.2 評価に対する考察

図 4.7 と図 4.8、または図 4.7 と図 4.9 を比較すると、折れ線を構成する線分が少ない方が辺の交差の無い美しい出力を得やすい傾向があると思われる。また、図 4.12 と図 4.13 を見ると折れ線が湾曲する向きが急激に変化している部分で形が崩れていることが分かる。

これらの結果から、良い出力結果を得るためには以下の条件が満たされていると良いと仮定した。

1. 折れ線を構成する線分の数が少ない
2. 入力折れ線を曲線とみなした時、曲率が変化する向きが急激に変化しない

これらの仮定を基に出力の改善を以下に示すように試みた。

まず仮定2. に関して、曲率変化が大きい場所ではどのような特徴があるかを観察した。曲率変化の大きい場所は法線ベクトルの向きの変化が大きい場所でもあるので、折れ線の各線分の法線を可視化して観察した(図4.19参照)

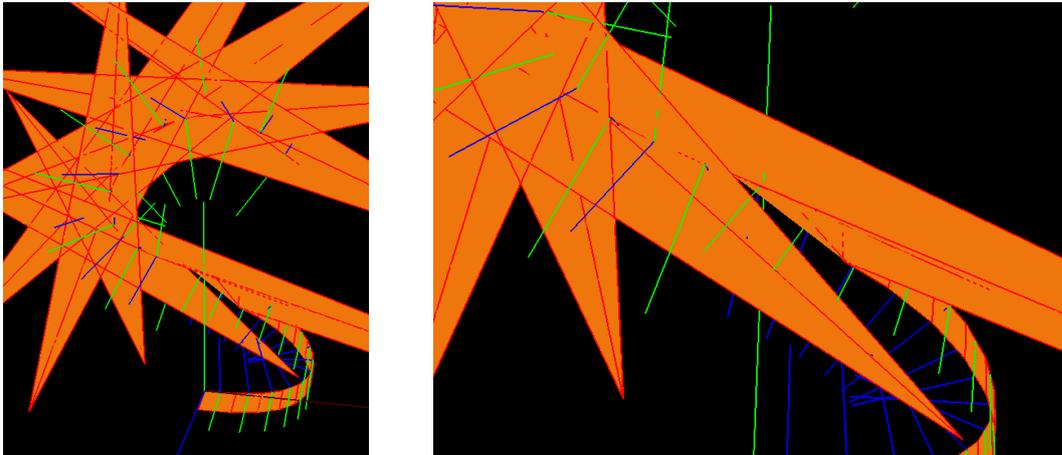


図 4.19: 左図: 帯状の可展多面体の形状が崩れている場合の法線(青線)と緑線(従法線)の例  
右図: 法線の向きが急激に変化している場所を拡大したもの

これらの図を見ると、法線の向きが大きく変化している部分ではポリゴンが極端に細長くなってしまっているのが分かる。これは節3.2におけるベクトル  $\mathbf{r}$  とベクトル  $\mathbf{t}$  の成す角度が  $0^\circ$  または  $180^\circ$  に極端に近くなっているためである。この問題を解決するために、ShapeTape からの入力から折れ線を完全に生成してから帯状の可展多面体を生成するという今までの過程を、以下に示すように折れ線と帯状の可展多面体を同時進行で生成する過程に変更して出力の改善を試みた。下記のベクトル  $\mathbf{T}, \mathbf{N}, \mathbf{B}, \mathbf{r}$  と  $\mathbf{W}$  は節3.2にて定義されているものと同じである。

- (1) ShapeTape で入力した座標  $x_n (n : 0 \sim 80)$  と折れ線を構成する頂点  $Q_j$ 、線分  $l_m$ 、各変数  $i$ (初期値は1),  $a$ (初期値は1),  $b$ (初期値は1) を定義
- (2)  $Q_0 = x_0, Q_1 = x_1, x_0$  と  $x_1$  を結ぶ線分を  $l_0$ 、 $l_0$  が向く方向の単位ベクトルをベクトル  $\mathbf{T}_0$  と定義
- (3)  $x_1$  と  $x_{1+i}$  を結ぶ線分を  $l_1$ 、 $l_1$  が向く方向の単位ベクトルをベクトル  $\mathbf{T}_1$  と定義
- (4)  $\mathbf{B}_0 = \mathbf{t}_0 \times \mathbf{t}_1, \mathbf{r}_0 = \mathbf{B}_0 * \frac{\mathbf{W}}{2}, \mathbf{N}_0 = \mathbf{B}_0 \times \mathbf{T}_0$  とする
- (5) 式3.4,3.5,3.6に各ベクトルを代入してベクトル  $\mathbf{r}_1$  を算出する
- (6)  $\mathbf{T}_0$  と  $\mathbf{T}_1$  の内積が0.99より小さい、かつ  $\mathbf{T}_0$  と  $\mathbf{r}_1$  の成す角度が  $0^\circ \sim 30^\circ, 150^\circ \sim 180^\circ$  であ

- る、かつベクトル  $\mathbf{r}_0$  と  $\mathbf{r}_1$  が交差していなければ (7) へ、そうでなければ  $i = i + 1$  して (3) へ
- (7)  $Q_2 = x_{1+i}, a = a + i, b = b + 1, i = 1$  とする
- (8)  $x_a$  と  $x_{a+i}$  を結ぶ線分を  $l_{b+1}$ 、 $l_{b+1}$  が向く方向の単位ベクトルをベクトルを  $\mathbf{T}_{b+1}$  と定義
- (9) 式 3.4,3.5,3.6 に各ベクトルを代入してベクトル  $\mathbf{r}_{b+1}$  を算出する
- (10)  $\mathbf{T}_b$  と  $\mathbf{T}_{b+1}$  の内積が 0.99 より小さい、かつ  $\mathbf{T}_b$  と  $\mathbf{r}_{b+1}$  の成す角度が  $0^\circ \sim 30^\circ, 150^\circ \sim 180^\circ$  である、かつベクトル  $\mathbf{r}_b$  と  $\mathbf{r}_{b+1}$  が交差していなければ (11) へ、そうでなければ  $i = i + 1$  して (8) へ
- (11)  $Q_{b+1}, a = a + i, b = b + 1, i = 1$  とする
- (12) (8)~(11) を  $a + i = (\mathbf{Q}_j$  の頂点数) になるまで繰り返す
- この生成フローで生成した帯状の可展多面体と曲線折り形状は以下の図のようになった。

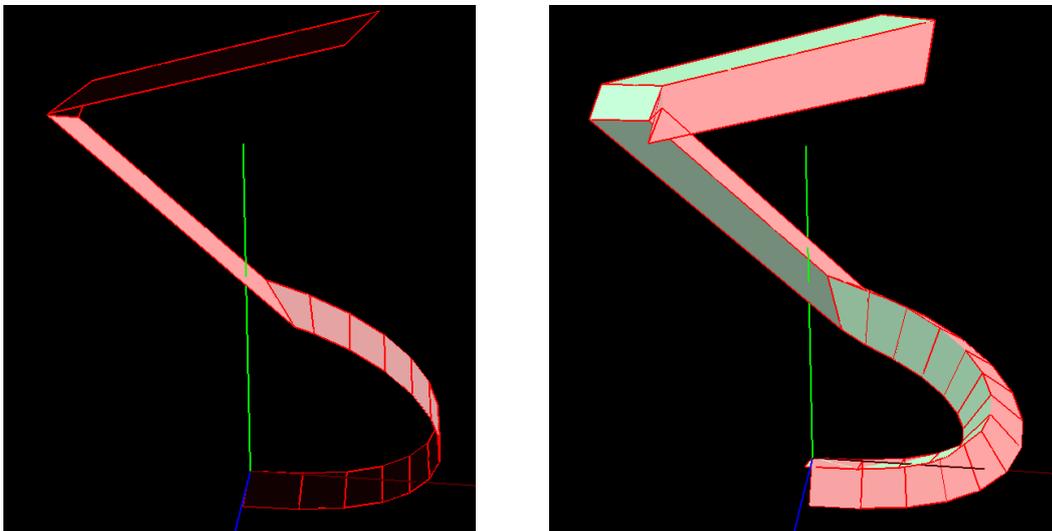


図 4.20: 左図: 図 4.19 と同じ入力で改善した生成フローで生成した帯状の可展多面体 右図: 左図から生成した曲線折り形状

## 第5章 考察

節4.1より、生成した曲線折り形状の折り線の折れ線を構成する頂点は少ない方が美しい出力が得られやすいことが判明した。また、図4.20のように湾曲する向きが急激に変化する場所では折れ線を構成する頂点は少ない方が出力の形状が崩れにくいことも判明した。これらより、提案手法においてはなるべく折り線の折れ線の頂点数は少なくなるように処理すべきだと考察した。しかし、折れ線の頂点数が少なくなるとそれが空間曲線に似なくなってしまう問題があり、頂点を減らすべき場所は減らし増やせる場合は増やすなどの対策を取る必要がある。この結果より、折り線の折れ線の線分の数が大きく減らざるを得ない結果になってしまったが、形が大きく崩れることなく曲線折り形状が得られた。

## 第6章 結論と今後の課題

### 6.1 結論

本研究では ShapeTape を用いて曲線折り形状を対話的かつ直感的に生成するインタフェースを開発した。提案インタフェースで生成する曲線折り形状は各パラメタを対話的に操作できる。また、そのインタフェースの出力が崩れないための条件を探るために出力の評価をした。

### 6.2 今後の課題

#### 6.2.1 新たな帯状の可展多面体の生成手法の発見

節 4.1 において様々な過程で出力形状が崩れない帯状の可展多面体を生成することを試みた。全ての入力ですべて美しい出力を得ることは出来ないことが判明した。そこで、館の手法 [3] でない新たな帯状の可展多面体の生成手法を発見又は開発する必要がある。

#### 6.2.2 出力形状を矩形に切り取る機能の開発

提案手法で生成した曲線折り形状は展開するとリボンのように細長い形状になり「曲線折り形状は平面を一本の曲線を折り目にして折った形状」ということが伝わりにくい問題がある。そこで生成した曲線折り形状の展開図を矩形に切り取る、あるいは切り取ったように見せるインタフェースを開発することで、生成物をより曲線折り形状らしいものにする。

#### 6.2.3 ポリゴンの交差判定

提案手法では出力の曲線折り形状のポリゴン同士の交差判定は行っていないため、交差が発生した場合は計算機上では出力出来ても紙などに出力すると最後まで折りたためない問題が発生する。現状ではディスプレイへの出力を目視して交差が発生しない様ユーザーが配慮する必要がある。そこで、自動でポリゴンの交差を検知してユーザーに知らせる機能を実装する。

## 謝辞

本論文の執筆にあたり、三谷純先生には多大なご指導とご助言を頂きました。深く感謝いたします。また、非数値処理アルゴリズム研究室の担当教員である遠藤先生、金森先生にはプログラミングについて多くのご助力をいただきました。研究室の同期、後輩らにも助力を求めた際も多くのご助言を頂きました。最後に普段から自分を支えてくれる家族、友人に心からの感謝をここに申し上げます。

## 参考文献

- [1] Marcelo A. Dias, Levi H. Dudte, L. Mahadevan, and Christian D. Santangelo, "Geometric Mechanics of Curved Crease Origami", *Physical Review Letter* 109, 114301, 2012.
- [2] Mitani, Jun, and Takeo Igarashi, "Interactive design of planar curved folding by reflection.", *Pacific Graphics* (short paper), 2011.
- [3] Tomohiro Tachi, "One-DOF Rigid Foldable Structures from Space Curves ", in *Proceedings of the IABSE-IASS Symposium 2011*, London, UK, September 20-23, 2011.
- [4] Kilian, et al. "Curved Folding", *Proceedings of ACM SIGGRAPH 2008: Volume 27 Issue 3*, August 2008.
- [5] 宮崎, 安田, 鳥脇, 横井. "仮想空間における折り紙の対話型操作の実現", *情報処理学会論文誌*: Vol 34 Issue 9, pp.1994-2001, 1993.
- [6] Tomohiro Tachi. "Simulation of Rigid Origami ," *Origami4*, pp. 175-187, 2009.
- [7] Kergosien, Y.L. "Bending and creasing virtual paper", *IEEE computer graphics and applications*: Volume 14 issue 1, January 1994.
- [8] Tomohiro Tachi, Student Member. "Origamizing Polyhedral Surfaces", *IEEE transactions on visualization and computer graphics*: Volume 16 Issue 2, 298-311, 2009.
- [9] Rahul Narain, Tobias Pfaff and James F. O' Brien, "Folding and Crumpling Adaptive Sheets", *ACM Transactions on Graphics (TOG): SIGGRAPH 2013 Conference Proceedings*: Volume 32 issue 4 No. 51, 2013.
- [10] Justin Solomon, Etienne Vouga, Max Wardetzky, Eitan Grinspun, "Flexible Developable Surfaces", *Computer Graphics Forum*: Volume 31 Issue 5, August 2012.
- [11] Devin J. Balkcom, Erik D. Demaine, Martin L. Demaine, John A. Ochsendorf, Zhong You. "Folding paper shopping bags.", *Origami 4*: 315, 2009.
- [12] Honghua Li, Ruizhen Hu, Ibraheem Alhashim, Hao Zhang. "Foldabilizing furniture." *ACM Trans. Graph* 34.4 2015.

- [13] 加瀬 悠人, ”折りたたみ可能な切開辺を含む立体形状の設計”, 筑波大学大学院, 修士論文, 2016.
- [14] Yong-Jin Liu, Kai Tang, Wen-Yong Gong and Tie-Ru Wu, ”Industrial design using interpolatory discrete developable surfaces”, *Computer-Aided Design: Volume 43, Issue 9*, pp 1089-1098, 2011.
- [15] Francesco Gioia, David Dureisseix, René Motro and Bernard Maurin ”Design and Analysis of a Foldable/Unfoldable Corrugated Architectural Curved Envelop”, *Journal of Mechanical Design: 134(3) 031003*, 2012.
- [16] Joseph M. Gattas and Zhong You, ”Miura-Base Rigid Origami: Parametrizations of Curved-Crease Geometries”, *Journal of Mechanical Design: 136(12) 121404*, 2014.
- [17] 上村 紳一郎, ”帯状三次元座標取得装置を用いた曲面モデリング”, 筑波大学, 学位論文, 2014.
- [18] Flux furniture, <http://www.fluxfurniture.com/>(2017/1/11 アクセス)
- [19] つくるぐ, <http://tukurug.blog111.fc2.com/blog-entry-75.html>(2016/12/24 アクセス)
- [20] JAXA, [http://www.jaxa.jp/projects/sat/alos2/index\\_j.html](http://www.jaxa.jp/projects/sat/alos2/index_j.html),(2016/12/24 アクセス)
- [21] 折り紙研究ノート, <http://mitani.cs.tsukuba.ac.jp/origami/>(2017/2/1 アクセス)
- [22] ペパクラデザイナー, <http://www.tamasoft.co.jp/pepakura/>,(2017/1/5 アクセス)
- [23] TT's page Software, <http://www.tsg.ne.jp/TT/software/>,(2017/1/5 アクセス)