

ユニット折り紙を用いた 3次元メッシュモデルの近似形状構築

田村友和^{†1} 高井昌彰^{†2} 高井那美^{†3}

仮想空間において1枚の紙を用いる一般的な折り紙に関する研究はこれまで数多く行われてきたが、同一形状の折り紙(ユニット)を多数結合させて一つの立体形状を構築していくユニット折り紙を対象とした研究は未だない。本論文では、ユニット折り紙の中で最も代表的な菌部式ユニットに注目した。ユニット折り紙で組み上げる際の制約を考慮した変形を予め3次元メッシュモデルに施した上で、正四面体と正八面体の組み合わせによってモデルをボクセル化し、その後それらの表面にユニットを配置する。これにより任意の3次元メッシュモデルの近似形状をより少ないユニット折り紙で再現することが可能となる。

3D Shape Approximation by Unit-Origami

TOMOKAZU TAMURA,^{†1} YOSHIAKI TAKAI^{†2}
and NAMI TAKAI^{†3}

So far, many researches on virtual Origami modeling have been made. But there has been no research intended for a Unit-Origami that is constructed from many elemental units with the same shape. In this paper, we focus on the most popular Unit-Origami: Sonobe-unit, and we present a Unit-Origami modeling system which can generate an approximated shape of a 3D polygon model. At first, to meet the geometric constraints of Sonobe-unit construction, we deform a given 3D polygon model. Then, we voxelize the deformed model with regular tetrahedra and octahedra. Finally, we arrange Sonobe-units on those voxel surfaces unit by unit. In this way, we can make a Unit-Origami work which approximates an arbitrary 3D polygon model.

1. はじめに

古くから「折り紙」に関する研究はよく行われており、中でも仮想空間内に折り紙を再現する研究はとりわけ盛んである¹⁾²⁾³⁾⁴⁾。それらの研究目的には、折り紙のリアリティ追求や計算コストの軽減、あるいは折り操作のユーザインタフェースの向上などがあげられる。しかし、従来研究の対象は基本的に不切正方形一枚折り(いわゆる一般的な折り紙)であり、複数枚の折り紙を組み合わせて一つの立体形状を作るユニット折り紙を対象とした研究は未だない。

また、ユニット折り紙の組み上げ操作には特有の難しさがあり、基本の正多面体形状の構築でさえも容易ではない。任意の立体形状をユニット折り紙で再現することは習熟者にとっても非常に困難であり、その構築手順を正しく記録することも難しい作業であるといえる。

そこで本研究では、ユニット折り紙モデリングシステムを開発し、仮想空間での対話的な組み上げ操作および組み上げ支援機能を実現する。また、3Dモデルデータを入力として、その近似形状を自動的にユニット折り紙で再現する自動組み上げ機能を実現し、その形状を実際のユニット折り紙で構築するための手順を生成する。

2. ユニット折り紙

ユニット折り紙の最大の特徴は同じ形に折られた複数枚の紙を用いる点にある。同じ形に折られた折り紙をユニットと呼び、それらを組み合わせることによって一つの形を作り上げる。その性質上、一般的には対称性の高い多面体などの形を作ることが多い。「奴さんと袴」などの異なる折り方の折り紙を組み合わせて形を作る方法もあるが、これはユニット折り紙と区別して複合折り紙と呼ばれる。

本研究では、ユニット折り紙の中でも最も代表的な菌部式ユニットを取り上げる。菌部式ユニットは、プラグと呼ばれる他のユニットに挿し込む部分と、ソケットと呼ばれるプラグを受け入れる部分がある。このプラグとソケットを組み合わせることでユニット同士を結合させて、形を作り上げていく(図1)。

^{†1} 北海道大学大学院 情報科学研究科複合情報学専攻
Graduate school of Information Science and Technology, Hokkaido University

^{†2} 北海道大学情報基盤センター
Information Initiative Center, Hokkaido University

^{†3} 北海道情報大学経営情報学部
Hokkaido Information University

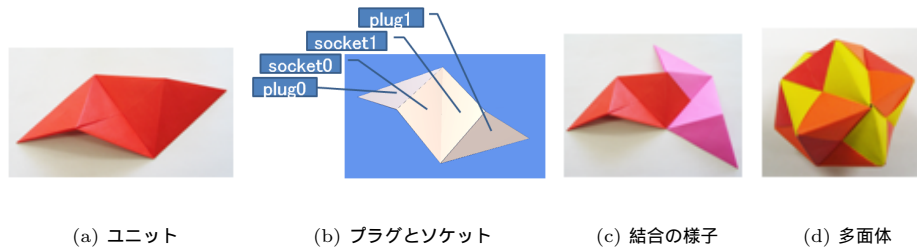


図 1 菌部式ユニット

3. ユニット折り紙モデリングシステム

本システムは、マウス及びキーボードを用いて仮想空間のユニットを操作し、実際にユニット折り紙で実現可能な任意形状を対話的に組み上げることができる。システムはユニットの結合可能な位置候補をユーザに順次提示し、基本的な正多面体形状の組み上げ作業を支援する。

また、入力された 3D モデルデータをシステムが解析し、ユニットを適切に組み合わせてその近似形状を自動的に構築することができる。構築後に個々のユニットを対話的に操作することも可能である。さらに、実際のユニット折り紙のために、全体の組み上げ手順リストを出力することもできる。

3.1 ユニットのデータ構造

本システムではユニット折り紙を表現するためにデータ構造を二つ定義する。1つは個々のユニットを表すためのデータ構造で、もう一つはユニットが複数結合した状態を表すためのデータ構造である。このユニットが結合した状態をユニオンと呼ぶ。システム内部には個々のユニットに対応したユニットデータクラスが定義される。このクラスは自身のユニットの番号と属しているユニオンの番号を持ち、また頂点に関するデータとしてユニオンの中でどの頂点と対応しているかという情報を保持している。その他にプラグとソケットに関する情報として、結合しているユニットの番号とプラグあるいはソケットの番号を保持しており、どのユニットとどのように結合しているかがわかるようになっている。同様にユニオンデータクラスが定義される。これは自身のユニオンの番号とユニオンを構成するユニットの番号を保持している。他に頂点に関する情報として、各頂点の座標とユニオンを構成する辺や面がどの頂点からなるかという情報も持っている。なお、ユニットが一つも属さないユニ

オンやユニオンに属していないユニットは存在せず、ユニットとユニオンの間には必ず対応関係がある。

3.2 ユニットの組み上げ

3.2.1 面の配置の計算

ユニットの組み合わせ方は無数に存在し、同じユニットの組み合わせ方で組み上げたとしても、個々のユニットの形状が一意に定まるとは限らない。そのため、運動学を解いてユニットの各頂点を求めることは難しい。そこで本研究では、古田らの提案した手法⁵⁾を用いた。この手法は、面を構成する頂点に網羅的にバネを張り、小さな面の歪みは許容しつつ、面が大きく歪んでしまった場合にはバネの釣り合おうとする力を利用して頂点を徐々に適切な位置へ移動させることにより、複雑な面の配置を実現している。

この手法ではある頂点にかかる力 F_i を一般的なバネモデルを簡略化した式 (1) で表す。

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

式 (1) の運動方程式を式 (2)、式 (3) を用いてオイラー法で数値的に解くことにより頂点の時間変化を計算することができる。

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

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

ここで、 L_{ij} はバネの自然長、 r_{ij} は頂点 j に対する頂点 i の相対位置、 d は減衰定数、 m は頂点の質量である。

本研究では、各辺以外に菌部式ユニット特有の折れ曲がった形状を維持するために図 2 で示す場所に、3つのバネを追加した。

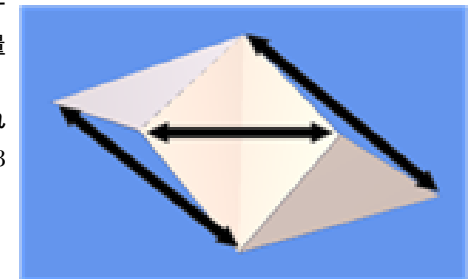


図 2 追加したバネの位置

3.2.2 ユニットの結合と綴じ

ユニットの組み上げに必要な処理には二つのタイプがある。第 1 のタイプは二つのユニオンを一つのユニオンに結合する処理 (図 3) であり、第 2 のタイプは同じユニオンに属するユニットのプラグとソケットを組み合わせて綴じる処理 (図 4) である。さらに綴じる処

理については、順々に結合していくプラグとソケットが隣り合っている場合と(図4(a)), 隣り合っていない場合(図4(b))とに分けられる。

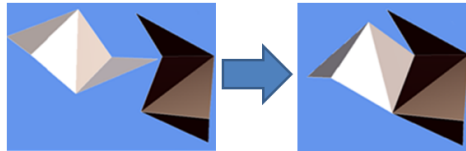
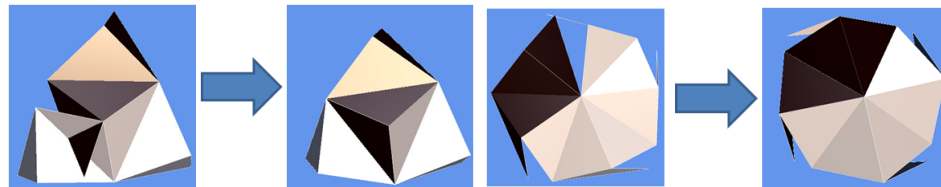


図3 結合させる処理の例



(a) プラグとソケットが隣り合っている場合

(b) プラグとソケットが隣り合っていない場合

図4 繋げる処理の例

結合の処理を行う場合には、結合させたいユニットのプラグとソケットを構成する座標がそれぞれ等しくなるように、そのユニットを含むユニオンの各頂点を並進および回転させる。その後、ユニオンの頂点に関するデータや辺、面のデータを書き換え、パネを張り替える処理を行う。

繋げる処理では、ユニオンの頂点に関するデータや辺と面のデータを更新しパネを張り替える処理を行う。その後、先に述べたパネモデルによって頂点座標の修正を行い全てのユニットの面の歪みを修正する。

3.3 組み上げ支援

菌部式ユニットで組み上げることのできる形状のうち対称性の高いいくつかの基本的な形状について、システムがその構築を支援する機能を実装した。これはユーザがあるプラグを選択した時に、基本的な形状を組み上げるには次にどのソケットに挿すべきなのかということを、そのソケットを点滅させることによって示すものである。その際点滅させるソケットは、選択したプラグの周囲の状況によって以下のように定める。

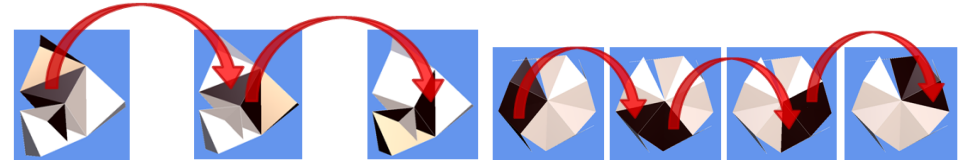
条件(1) 図5(a)のようにユニットを辿り、2個のユニットを辿ることができた場合には図6(a)のソケットを示す。

条件(2) 図5(b)のようにユニットを辿り、2個から4個のユニットを辿ることができた場合には図6(b)の位置に相当するソケットを示す。

条件(3) 先の両方の条件を満たす場合は図6(a)のソケットを優先する。

条件(4) 先の条件(1)と(2)のどちらも満たさなかった場合は、別のユニオンにあるユニットが結合していないソケット全てを示す。

なお、条件(2)の場合で示したソケットに初めて結合させる時には、たどることができたユニットの数を保存しておく。以降は、ユニットをその数だけたどることができたときのみ結合させるソケットを示し、構築する形状を基本的なものへ誘導する。



(a) 隣り合っているプラグとソケットを辿る場合

(b) 隣り合っていないプラグとソケットを辿る場合

図5 結合状況の解析



(a) 条件(1)及び(3)の場合



(b) 条件(2)の場合

図6 点滅させるソケットの位置

3.4 近似形状の自動組み上げ

3Dモデルの近似形状をユニットで組み上げる準備として、入力モデルの収縮および正四面体と正八面体によるボクセル化を行う。著者らの先の発表⁶⁾では、入力モデルの収縮処理を行わず、平行六面体によるボクセル化のみであるため、近似形状を少ないユニットで精度

よく再構築することに問題があった。

菌部式ユニットは組み合わせによって基本的に正三角形を形成する性質がある。このため、あらかじめ入力モデルを収縮させてからユニットによる構築を行い、その形状を入力モデルに近づける。またこの性質を利用してボクセル表面の正三角形にユニットを隙間なく配置できるようにする。

3.4.1 収縮

前処理として入力モデルを収縮させるために、各頂点の頂点法線ベクトル n を求め、その逆ベクトル n^{-1} の方向へ各頂点を移動させる。その際、ユニットで作られる三角錐の高さだけ移動させる。しかし、この距離が大きすぎる場合には、反対側の面を突き抜ける可能性がある(図7(b))。そこで、あらかじめ法線ベクトルの逆ベクトルがモデルと衝突する距離から、各頂点を移動させることのできる上限を設定しておく(図7)。その際、事前に各ポリゴンの位置とインデックスに対応関係を用意しておくことで、 n^{-1} と衝突判定を行い距離を計算するポリゴンの数を大幅に減らしている。

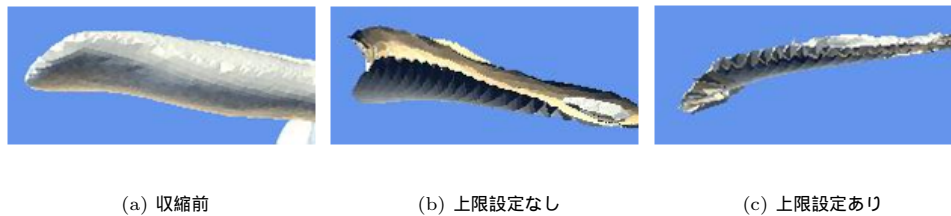


図7 入力モデルの収縮処理

3.4.2 ボクセル化

次に収縮させたモデルを正四面体と正八面体でボクセル化する。そのために、まず正四面体二つと正八面体一つで構成される平行六面体(図8)によってボクセル化を行う。立方体を平行六面体に変形する行列 A の逆行列を収縮させたモデルに適用し、変形させる。その後、立方体によるボクセル化を行い、ボクセルに行列 A を適用することで立方体のボクセルを平行六面体のボクセルに変換する。

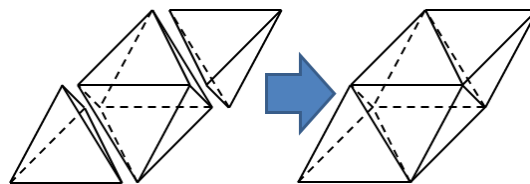


図8 正四面体と正八面体からなる平行六面体

この平行六面体のボクセルを正四面体二つと正八面体一つに分解し、モデルを全く含まな

い物を排除する。こうして、正四面体と正八面体によるボクセルを得る。なお、このボクセル空間の解像度を変更することで、近似形状構築に使用するユニットの総数を大まかに調整することができる。

以上のボクセル化の流れを図9に示す。

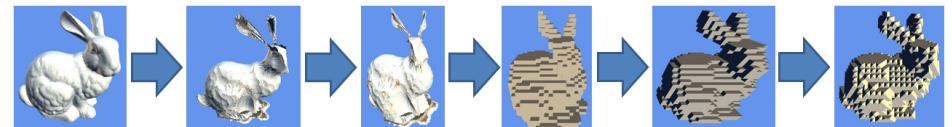


図9 正四面体と正八面体によるボクセル化の流れ

3.4.3 ユニットの配置

正四面体と正八面体によるボクセル化の後、ユニットを配置する場所を決定するためのボクセル表面上の正三角形の辺の位置と、ユニットの結合のさせ方を判断するための辺同士の位置関係を調べる。3つのユニットをお互いに結合させようとユニットの中心の折れ目は正三角形を形成する。この正三角形の辺をそれぞれ e_0, e_1, e_2 とすると、 e_0 に配置したユニットは e_2 に配置したユニットに結合し、同様に e_1 に配置したユニットは e_0 に、 e_2 に配置したユニットは e_1 に配置したユニットに結合している(図10)。このように、ユニットは反時計回りに順々に結合していくので、各ユニットの結合先の情報としてそれぞれに対応した辺の情報を保持する。

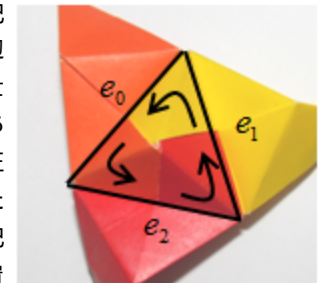


図10 辺の位置関係

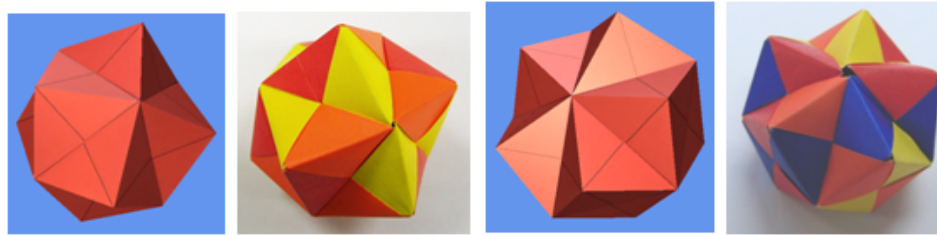
最終的にユニットによる近似形状の構築を行うには、各辺が持っている位置情報に合わせてユニットの位置を定め、位置関係の情報から結合すべきユニットとソケットを特定し、3.2に示した手法を用いてユニットを順次組み上げていく。

4. システムの実行結果

はじめに、マウス及びキーボードによってユーザが対話的に入力を行い、菌部式ユニットで実際に組むことのできるいくつかの形状を本システムで組み上げた。その結果を図11に示す(右の図は実際のユニット折り紙)。形状A(図11(a))は使用するユニットの枚数が少なく、対称性も非常に高いため菌部式ユニットで作られる形状としては最も基本的なも

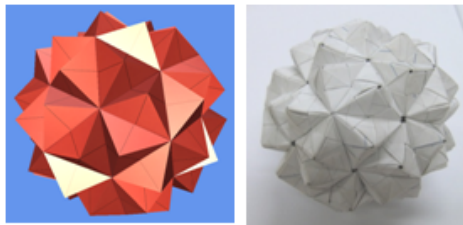
のである。形状 B (図 11(b)) はユニットの枚数は少ないが、形状 A よりも対称性が低い。形状 C (図 11(c)) は他の形状よりも使用するユニットの枚数が多く、複雑な形状をしている。どの形状についても本システムで構築できることが分かる。

なお、形状 A については 12 枚すべてのユニットを組み上げ支援にのって組み上げることができる。形状 B については対称性が下がるため、システムが支援することはできない。形状 C は複雑な形状ではあるが組み上げの一部については支援をすることができる。



(a) 形状 A (ユニット数 12, 組み上げ時間 1 分)

(b) 形状 B (ユニット数 15, 組み上げ時間 1 分半)

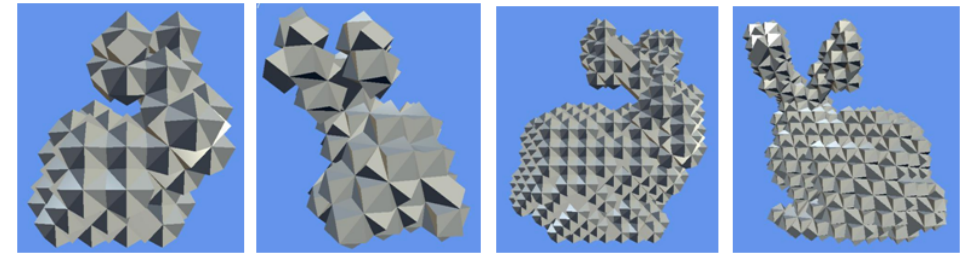


(c) 形状 C (ユニット数 90, 組み上げ時間 10 分)

図 11 マウス入力による対話的な構築結果

次に、システムにスタンフォードバニー及びユタティーポットを入力し、その形状を菌部式ユニットで自動的に構築した。その結果を図 12 と図 14 に示す。また、入力モデルを収縮せずに平行六面体でボクセル化した場合のスタンフォードバニーの構築結果を図 13 に示す。スタンフォードバニーの自動組み上げについては総数 350 個程度のユニットをがあれば、

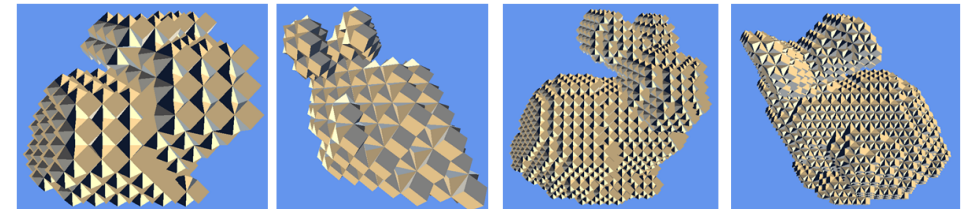
耳などの大きな特徴が再現できていることが分かる (図 12(a))。また、2,400 程度のユニットを用いれば入力モデルの形状をよく再現できていることが分かる (図 12(b))。図 12 と図 13 の結果を比較すると、本稿で示したボクセル化により、入力モデルの近似形状をより少ないユニットで構築できていることが分かる。



(a) ユニット数 : 357 個

(b) ユニット数 : 2,364 個

図 12 ユニット折り紙によるバニーの構築
(収縮処理あり, 正四面体と正八面体によるボクセル化の結果)



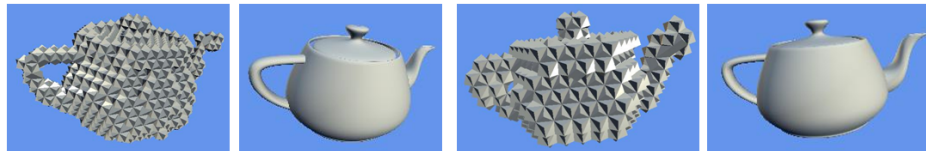
(a) ユニット数 : 1,008 個

(b) ユニット数 : 9,492 個

図 13 ユニット折り紙によるバニーの構築
(収縮処理なし, 平行六面体によるボクセル化の結果)

また、ユタティーポットの自動組み上げ結果については 5,000 個程度のユニットを使った場合であれば、注ぎ口や取っ手などの特徴をよく再現できていることが分かる。しかし、使用ユニットを 1,800 個程度まで減らした場合、注ぎ口や上部の突起は再現できているもの

の、取っ手の部分は穴がつぶれてしまい再現できているとは言い難い結果となっている。これは取っ手の部分が細く、モデルを収縮させても十分な効果が得られないためであると考えられる(図14)。



(a) ユニット数 : 4,827 個

(b) ユニット数 : 1,826 個

図14 ユニット折り紙によるティーポットの構築

なお、以下の環境において自動構築に要する時間はいずれの例でも3秒程度であった。

Model	Dell Precision 490
CPU	Intel(R) Xenon(R) CPU 5160 3.00GHz 2.99GHz
RAM	12.0GB
GPU	NVIDIA Quadro FX 3500

本システムが3Dモデルを自動構築した際には、図15のようなユニットの結合の仕方を一挙手ずつ記したテキストファイルを出力する。ここで例えば、1行目の「Unit-1.0 -> Unit-0.0」は「1番ユニットの0番プラグを0番ユニットの0番ソケットに結合させる」と言う意味である。図12(a)の場合では、手順の総数は714である。

```
0 : Unit-1.0 -> Unit-0.0  
1 : Unit-2.0 -> Unit-1.0  
2 : Unit-0.0 -> Unit-2.0  
3 : Unit-1.1 -> Unit-3.0  
4 : Unit-4.0 -> Unit-1.1  
5 : Unit-3.0 -> Unit-4.0  
6 : Unit-5.0 -> Unit-0.1  
7 : Unit-2.1 -> Unit-5.1  
8 : Unit-6.0 -> Unit-5.0  
9 : Unit-0.1 -> Unit-6.0  
10 : Unit-1.2 -> Unit-1.2
```

図15 出力された構築手順

5. おわりに

本稿では、ユニット折り紙を仮想空間に再現し、3Dモデルデータを入力としてその近似形状を自動的に構築し、さらにはその構築手順を出力するモデリングシステムについて述べた。入力3Dモデルに対して収縮の前処理を施し、正四面体と正八面体の組み合わせによるボクセル化を行うことで、近似形状をより少ないユニット数で再構築することができた。

今後、システムが提示した結果を実物のユニット折り紙で組み上げるためのユーザ支援機能として、例えば、ユニットを結合すると隠れる部分に結合手順の番号を印刷する機能を実装することにより実用性の高いシステムを目指す。

参考文献

- 1) 三谷純, 鈴木宏正 “折り紙の構造把握のための形状構築とCG表示”, 情報処理学会論文誌, Vol.46, No.1, pp.247-254, 2005.
- 2) 横山卓弘, 高井昌彰 “厚さを持った折り紙シミュレーションとその評価”, 情報処理学会研究報告, Vol.2000, No.115, 2000-CG-101, pp.19-24, 2000.
- 3) 宮崎慎也, 安田孝美, 横井茂樹, 鳥脇純一郎: 仮想空間における折り紙の対話操作の実現, 情報処理学会論文誌, Vol.34, No.9, pp1994-2001(1993)
- 4) Tomohiro TACHI: Smooth Origami Animation by Crease Line Adjustment; SIG-GRAPH 2006 Posters, (2006)
- 5) 古田陽介, 三谷純, 福井幸雄 “マウス入力による対話的操作を組み込んだ仮想折り紙システム”, 情報処理学会研究報告, Vol.2006, No.76, 2006-CG-123, pp.13-28, 2006.
- 6) 田村友和, 高井昌彰, 高井那美 “近似形状の組み上げ機能を有するユニット折り紙モデリングシステム”, 画像電子学会 Visual Computing 研究会 / 情報処理学会 グラフィクスとCAD研究会 合同シンポジウム 2010 論文集, No.24, pp.1-6, 2010