

折り図作成を支援する手順予測インタフェースと 次の手順候補に対するランク付け手法

鶴田直也[†] 三谷純^{††} 金森由博^{††} 福井幸男^{††}
筑波大学第三学群情報学類[†] 筑波大学大学院システム情報工学研究科^{††}
tsuruta @ npal.cs.tsukuba.ac.jp, {mitani, kanamori, fukui} @ cs.tsukuba.ac.jp

アブストラクト

現在、折り紙作品の折り図を描く際には、一般的なドローツールが用いられている。しかし、近年の作品の複雑化により、折り畳みによって変化する図を一枚ずつ描く手間は、非常に大きいものとなった。本論文では、折り図作成の作業効率を改善する、手順予測を組み込んだインタフェースを提案する。このインタフェースは、与えられた図から、藤田・羽鳥の折り紙公理によって得られる折り線をすべて算出し、その折り線で折った後の形状を予測候補として列挙する。提示された候補の中から、目的の図を選択することで、作図の作業を簡略化できる。また、予測処理により生成される膨大な数の候補から、有用なものをランク付けする手法を示す。実際に本手法を実装した折り図作成支援システムを開発し、その有効性について評価するための実験を行った。

1 はじめに

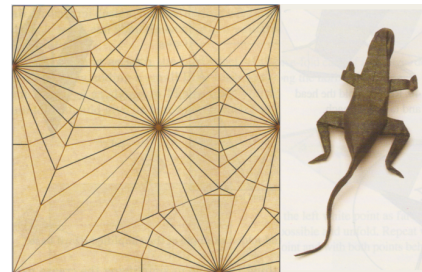
日本の伝統のひとつとして知られる折り紙は、その幾何学的性質から数学の分野で広く研究されてきた。近年では、折り紙作品が持つ構造が注目され、工学をはじめとする様々な分野で応用されている。

折り紙には、最も一般的な、正方形の紙を用いて切込みをいれずに折る「不切正方形一枚折り」をはじめ、複数の紙を組み合わせで作る「ユニット折り」や繰り返しのパターンを折っていく「平折り」など様々な形態がある。特に、「不切正方形一枚折り」においては、特定の構造を持つ作品の設計手法 [1] が確立したことにより、非常に複雑でリアルなものが創作されるようになった。その作品を伝える手段としては、一般的に、一つ一つの手順を図示した「折り図」(図 1(b)) が用いられる。

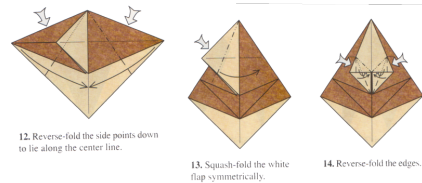
しかし、この折り図の作成作業は簡単なものではない。問題の 1 つは、作品が複雑になったことにより、折り図を一つずつ描く手間が大きくなったということである。伝承作品と呼ばれる「鶴」などの古くからある作品の手順が 10 から 20 程度であるのに対し、近年の複雑な作品は、100 や 200 を越すものまである。それらの複雑な作品は、写真と「展開図」(図 1(a)) を合わせて公開されることが多い。展開図は作品のもつ構造を示したものであるが、一般の人にとって、展開図の情報のみでその作品を折り上げることは困難である。また、一つの折り紙作品には無数の異なる手順が存在する。そのため、折り紙作品を折り図化するとき、他者に伝達する上で効率の良い手順を考えることが難しいということも挙げられる。これについては、「良い手順」には折る側の技術に寄る部分があり、一定の基準で評価することが困難であるため、今回は考慮しない。

本論文では、上で挙げた 2 つの問題のうち前半の折り図を描く手間を軽減する、折り図作成のためのインタフェースを提案する。対象とするのは、「不切正方形一枚折り」かつ、平坦に折り畳める作品である。ただし、紙をずらし

たり潰したりする操作や、紙をねじるように折る操作など、複数の異なる折り線を扱う必要のある作品には対応していない。このシステムには、ある時点の形状から次の折り手順を予測し、提案する機能の実装を行った。この機能は、7 つある藤田・羽鳥の折り紙公理 [2] のうちの 4 つに基づき可能性のある折り線をすべて生成し、その折り線で折った形を候補として列挙する。ユーザは、その中から目的のものを選択することで作図の手間を減らすことができる。これにより、折り図作成作業の効率を改善することを目的とする。



(a) 展開図及び完成写真



(b) 折り図の一部

図 1 Robert J. Lang: Lizard の展開図と折り図の一部
(出典: [1])

2 関連研究

折り紙では、2 辺を重ねる、2 点を重ねるといった操作によって折る位置を決める場合が多い。これらの操作は、藤田・羽鳥の折り紙公理としてまとめられており、以下の

7つからなる．本論文で提案する手順予測機能は，この公理がベースになっている．図2は，それぞれの操作を図示したものである．

公理1 2点 p_1, p_2 が与えられたとき，2点を通るただ一つの折り方がある．

公理2 2点 p_1, p_2 が与えられたとき， p_1 を p_2 に重ねるただ一つの折り方がある．

公理3 2本の直線 l_1, l_2 が与えられたとき， l_1 を l_2 に重ねるような折り方がある．

公理4 1点 p_1 と1本の直線 l_1 が与えられたとき， l_1 に垂直で p_1 を通るただ一つの折り方がある．

公理5 2点 p_1, p_2 と1本の直線 l_1 が与えられたとき， p_1 を l_1 上に重ね， p_2 を通る折り方がある．

公理6 2点 p_1, p_2 と2本の直線 l_1, l_2 が与えられたとき， p_1 を l_1 上に重ね，かつ p_2 を l_2 上に重ねる折り方がある．

公理7 1点 p と2本の直線 l_1, l_2 が与えられたとき， p を l_1 に重ね， l_2 に垂直な折り方がある．

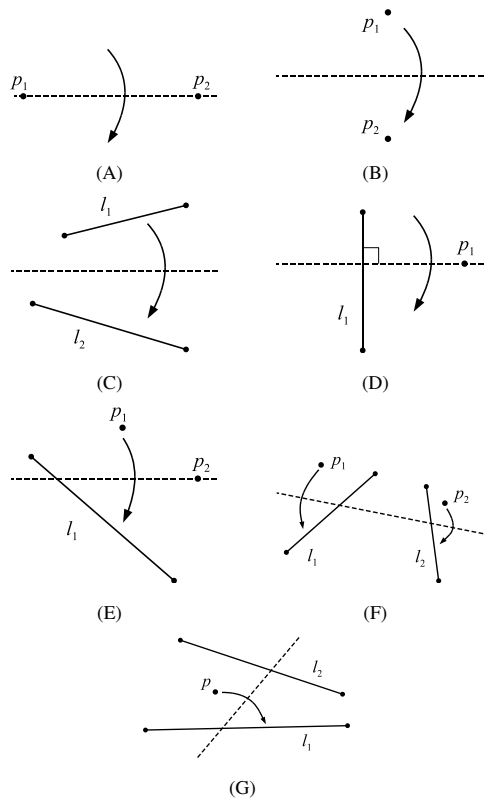


図2 藤田・羽鳥の折り紙公理

折り紙のシミュレータとしては，Miyazaki ら [3] や古田ら [4] の研究がある．これらは，簡単な作品であれば計算機上に再現することができ，作品を折る過程のアニメーションにも利用できる．計算機上に折り紙を構築する手法としては，QRコードを利用して取り込むといったこともされている [5]．紙上での位置情報を記録した QRコードを印刷した紙を使用し，折り工程を1つずつ写真に撮ることで，それを計算機に取り込み復元することが可能である．ただし，内側への折り込みの判定が難しいといった問題もある．

シミュレータとしての機能もあるが，折り図を描くことにより焦点を当てたものが，Foldinator [7] と eGami [8] である．どちらも平坦な折り紙を対象としており，折り畳み方向を示す矢印や折り図らしい紙のずれの表現ができる．しかし，これは先に述べたシミュレータと共通の問題点であるが，両手で行う折り紙の動作を，マウス1つで再現することは難しい．これは，紙の一部を押さえて，ずらすといった動作ができないことを意味する．そういった制約はいくつか存在するが，ある程度，実用的な利用も可能となっている．本論文で提案するインタフェースは，手順予測による支援を加えることで，より効率的な作業を目指す．

研究背景でも述べたように，現在，折り図の作成が困難な作品を公開する場合は，展開図が用いられることが多い．三谷はこの展開図に着目し，角の2等分線や三角形の内心などの，展開図に現れる特徴的な構造を効率的に入力可能なエディタ [6] を開発した．このエディタには，入力された展開図から，折り畳み後の形状を推定して表示する機能も実装されている．形状推定の際の重複削減手法は，3.3節で述べる重複形状の削減で参考にしている．

最後に，折り紙との関連はないが，CADシステムにおいて予測機能を実装した例を取り上げる．Igarashi らは，辺や面を入力した際に，そこから作られる直方体などの候補を提示することで3次元モデルを簡単に作成できるシステム Chateau (図3) [9] を開発した．本研究の手順予測の考えは，このCADシステムから発している．図3は，システムの予測が働いている画面である．下部に表示された候補の中から目的のものを選ぶことで，入力を省略できる．

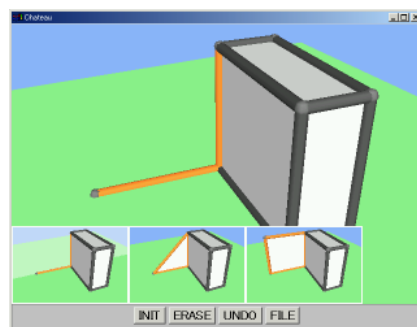


図3 3Dモデリングシステム：Chateau

3 提案手法

本章では，3.1節で折り図に用いるデータ構造を説明した後，続く3.2節で，提案するインタフェースにおける予測候補の生成手法について述べる．そして，生成された候補をランク付けする手法を3.3節で述べる．

3.1 データ構造

1つの図 (Diagram) は，紙の輪郭線または折り線で分割された平面上の複数の面で構成されており，折り図全体はそのリストで表現される (図4)．各面は，重なり順にスタックで保持される．また，1つの図はハーフエッジ構

造 [10] で管理される．ハーフエッジは、辺の両側にあり、属する辺と面、前後のハーフエッジ、反対側のハーフエッジ、始点となる頂点の情報を持つ．前後のハーフエッジを辿ることで面を巡回でき、柔軟な形状操作処理が可能である．

折り畳みによる紙の構造情報の更新は、次の手順で行われる．図 5(a) は最初の状態を示している．面 F_0 は頂点 V_0 - V_3 からなり、頂点間には辺 E_0 - E_3 と対応するハーフエッジが存在する．まず、折り畳む各面に対して、折り線と辺の交点を算出する．次に、折り線と交差した辺を交点で分割し、分割された辺と交点をつなぐ辺を新しく生成する（図 5(b)）．同時に、それらに属するハーフエッジを生成し、前後と反対側のハーフエッジの接続情報を更新する．図 5(b) では、 E_1 から E_4 と E_5 、 E_3 から E_6 と E_7 が生成されている．また、交点が新しい頂点 V_4 と V_5 として追加されている．そして、交点をつなぐ新しい辺（ E_8 ）の両側のハーフエッジを巡回し、分割された形状の面を新しく生成する（図 5(c)）．この時点で、新しい面 F_1 と F_2 が作られる．最後に、移動する面の頂点を辺と対称の位置に動かしてから、分割された面を取り除き、折り操作に応じた重なり順でスタックに追加する（図 5(d)）．面の表裏は、ハーフエッジの巡回方向が時計回りか否かで判別する．以上が、構造情報の更新処理である．なお、この処理は、予測によって多数の候補を生成する必要があるため、その都度元となる図をコピーしてから行うようにしている．

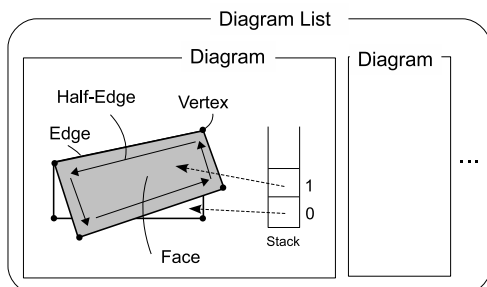


図 4 計算機上での折り図の表現

3.2 手順予測

ここでは、本論文で提案する折り図の手順予測手法について述べる．本システムでは、前述の 7 つの折り紙公理のうち 1-4 番目の公理を実装した．その理由は、5-7 番目の公理は数学的性質の強いものであり、実際に折り紙作品を折る際に利用することは稀だからである．利用される可能性の低い候補は、目的の候補を選ぶ際の妨げになる．そのため、今回は実装を見送った．

手順予測は公理 1-4 を基に、次の順に処理を行う．

1. 現在の紙の状態に折り紙公理を適用する．
2. 生成された折り線をどの方向に折るかを判定する．
3. 手前の面から順に「谷折り」で折り畳んだ形状」と「折り線を付けただけの形状」を得る．
4. 折り畳んだ形状から重複しているものを除外する．

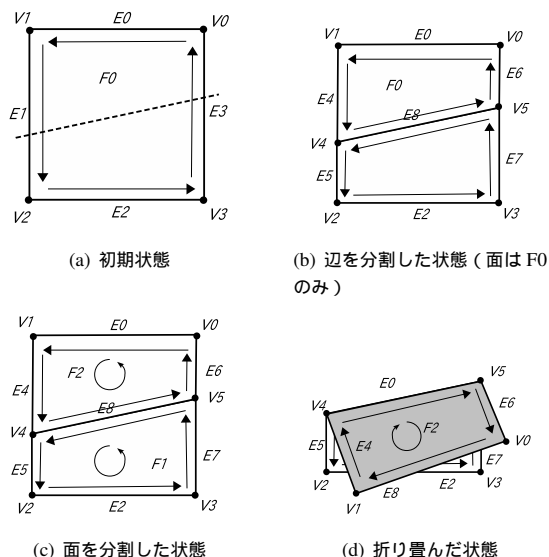


図 5 紙の構造情報の更新

5. 各候補をランク付けし、有用であると思われる順に並べ替えを行う．
6. 生成されたすべての候補をランク順に提示する．

ユーザは、生成された候補の中から目的の図を選択することで、作図作業を省略できる．今回は、谷折りと折り線を付ける操作のみを対象とし、それ以外の折り方や内側に差し込むなどの特殊な操作は除外した．そのため、予測機能に対応できる作品は非常に限定されている．ただし、あらかじめ紙を反転させておくことで、山折りと同等の候補を得ることは可能である．以下、上記手順の詳細を説明する．

3.2.1 折り紙公理の適用

まずは、現在の状態に折り紙公理を適用し、折り線の位置を決定する．この際、図 6 のように、異なる公理から同じ折り線が生成される場合がある．この場合は、同一の候補が生成されるため、折り畳み処理は行わない．

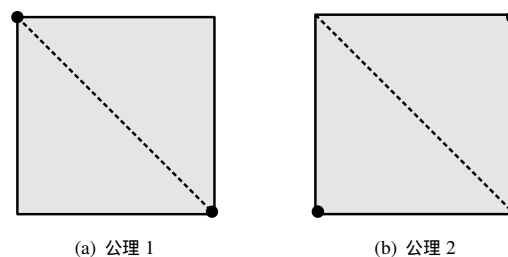


図 6 異なる公理から同じ折り線が生成される例

続いて、手前の面から順番に、生成した折り線で何枚の面が折れるかを調べる．例えば、図 7(a) では、平坦に畳むためには 2 枚の面を同時に折る必要があるが、図 7(b) では、手前の 1 枚の面だけで折り畳むことができる．これは、紙内部の頂点に接続する辺の本数で判断することが可能である．紙内部の頂点において、接続する山折りの数を M 、谷折りの数を V とすると、その差が $M - V = \pm 2$ と

なる [11]。したがって、紙が平坦に折り畳めるとき、紙内部の頂点には必ず偶数本の辺が接続している。辺の数が奇数ならば平坦には折り畳めないため、次の面を調べる。

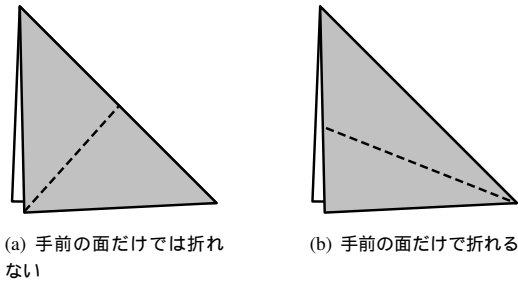


図7 折り線の位置による折り畳み面数の違い

折り畳む枚数が決定した時点で面の分割を行う。分割処理が行われると、各面につき2つの面が新しく生成され、分割元となった面が取り除かれる。谷折りをする場合は、どちらかの面を折り線に対して反対側に移動しなくてはならない。そのため、新しく生成された面は一時的にスタックに保持され、面の移動処理を行った後、元の図に追加される。折り線を付けるだけの形状を生成する場合は、新しく生成された面はそのまま追加され、候補が1つ生成される。頂点の移動処理は行わないため、3.3節の重複確認処理に移る。

3.2.2 折り畳み方向の決定

谷折りをする場合は、どちらかの面を移動させる必要があり、そのために折り畳み方向を決定しなければならない。折り畳み方によっては、生成される候補の数が異なる可能性がある。例えば、図7(b)において下から折り上げる場合のパターンは、図8(a)と図8(b)の2通りであるが、上の部分を折り下げの場合のパターンは図8(c)の1通りである。

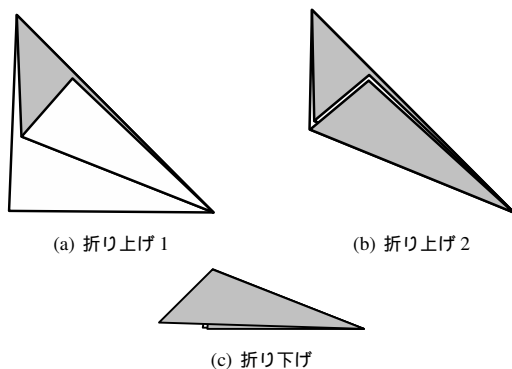


図8 折る方向による候補数の違い

この判定処理は、未探索の面があるかないかによって異なる。以下、それぞれの手法を説明する。

3.2.3 未探索の面がある場合

面の探索途中で、まだ未探索の面が残っている場合は次の手順で折る方向を決定する。例を示すと、図9のようになる。

1. 未探索の面と接続している面を調べる。

2. その面のある側を固定側とする。
3. 残りの面を移動面とし、折り畳み処理を行う。

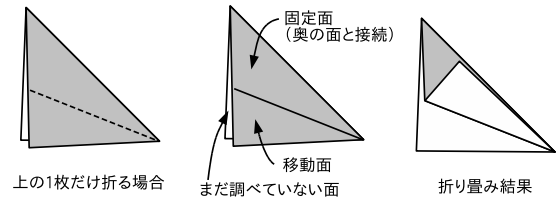


図9 未探索の面がある場合の折り畳み方向の決定

3.2.4 すべての面が探索済みの場合

すべての面の探索が終わっているとき、折り線の上に重なる面は存在しない。そこで、次の方法で折る方向を決める(図10)。

1. すべての面を折り線の左右に振り分ける。
2. 左右の面群の重心から折り線までの距離を比較する。
3. 距離の短い方を移動面として、折り畳み処理を行う。

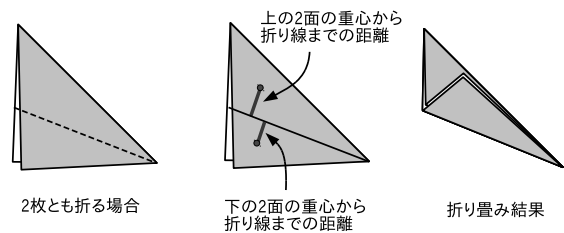


図10 すべての面が探索済みの場合の折り畳み方向の決定

3.2.5 折り畳み処理

前節までの処理で折り畳み方向が確定している。ここでは、一時的に保持しておいた分割された面に対して折り畳み処理を行う。まず、折り畳まない側にある面をスタックから取り出し、元の図に追加する。そして、折り畳む側の面に含まれる頂点のうち、折り線上にない頂点のみを、折り線に対して反対側へ移動する。移動処理を行った面を後から元の図に追加することで、谷折りをした形状が得られる。

今回実装した予測機能は、手前の面から順に谷折りを行う。したがって、生成される候補は図8(a)や図8(b)のようになり、奥の面だけ折る形状(図11(a))や内側に折り込む形状(図11(b))は得られない。

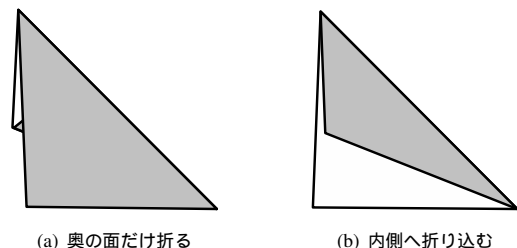


図11 生成されないパターン

このメリットは、面の貫通処理が必要でなくなることが

挙げられる．コンピュータ上で紙を扱う場合，図 12(c) のように，面の貫通が発生することがある．手前から順に面を折っていけば，この状態は発生しない．

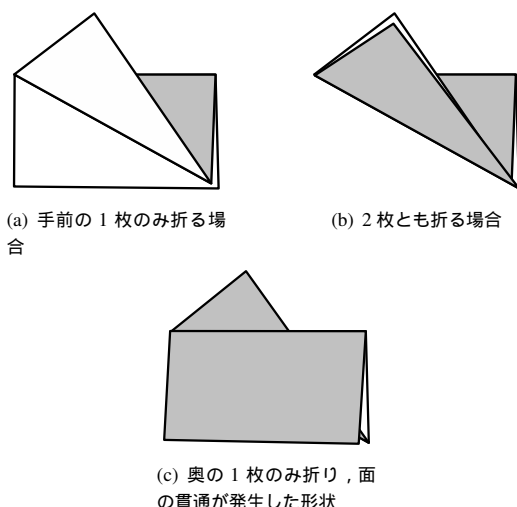


図 12 面の貫通が発生し得る状況

3.3 ランク付け手法

前述の手順で候補を生成すると，膨大な数の候補が生成される．大量の候補は，目的の候補を選択する際の妨げとなる．そこで，回転・反転した形状を重複とみなして削除し，残りの形状に対して有用と思われる順にランク付けを行うこととした．これにより，目的の候補の選択を容易にする．

3.3.1 重複した形状の削除

重複削減は以下の 3 段階の処理によって行われる．

まず，対象の形状の面の数を比べる．今回の手法では，折り線をつけるだけの操作も候補として出力される．そのため，最も単純な例を挙げれば，図 13 のように，頂点数は同じでも面の数が異なるパターンが存在する．よって，最初に面数を比較し，同じであれば重複している可能性があるとする．

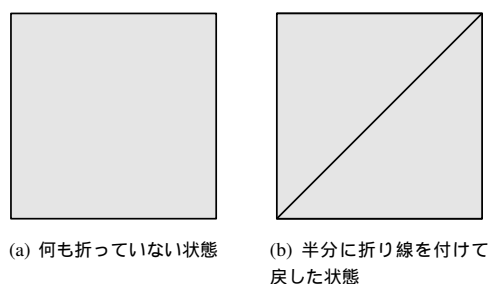


図 13 頂点数が等しく面数が異なる形状

次に，重心から各頂点への距離の総和を求め，この値が等しいかどうかを調べる．この値は，図 14 に示すように回転や反転した形状においても等しくなる．また，頂点数と位置によって決まるので，前述の図 13 ではこの値は等しくなる．

重心からの距離の総和を比較しただけでは，重複してい

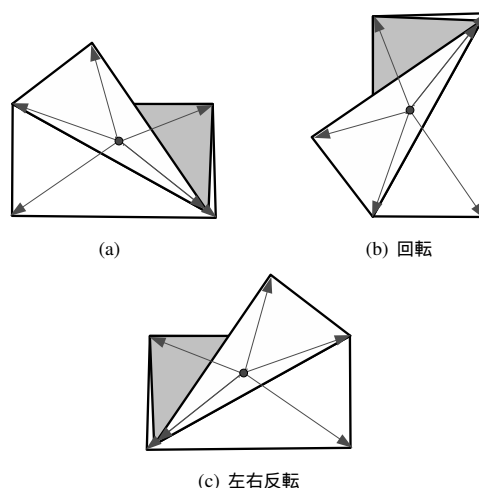


図 14 重心から各頂点への距離

ると見なされてしまうパターンが存在する．それが図 15 に示すような場合である．この図は，三角形に 2 回折った状態からどちらかの角を半分に折ったものである．矢印の右にある 2 つの形状は，図 14 に示した重心から各頂点への距離の総和が等しいため，重複と判定されてしまう．

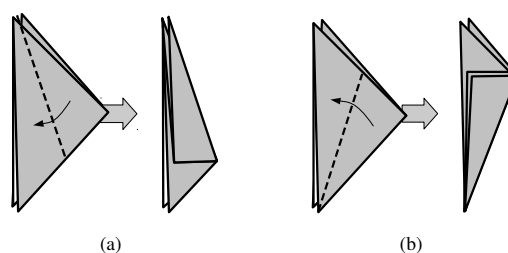


図 15 同じ値を持つ形状の例

このような場合に正しく判定を行うため，さらに面の接続を調べて，その情報を比較する．図 15 に示した 2 つの形状を展開すると，それぞれ，図 16(a),(b) のようになる．各面に付けられた数字は，その面が展開前の形状において，奥から何番目に位置していたかを表している．例えば，左の形状 (a) では，面 1 が面 8 と面 2 に接続していることがわかる．この関係を，行列で表すと表 1 と表 2 のようになる．この行列を比較すると，2 つの形状が異なるものであると判定できる．

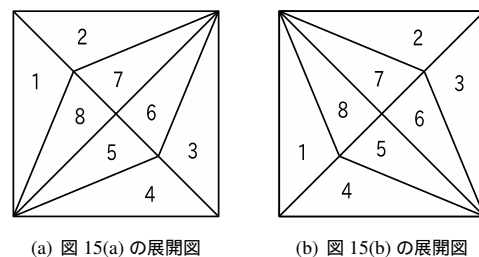


図 16 各形状の展開図

表 1 図 16(a) の面接続行列

	1	2	3	4	5	6	7	8
1		o						o
2	o						o	
3				o		o		
4			o		o			
5				o		o		o
6			o		o		o	
7		o				o		o
8	o				o		o	

表 2 図 16(b) の面接続行列

	1	2	3	4	5	6	7	8
1				o				o
2			o				o	
3		o				o		
4	o				o			
5				o		o		o
6			o		o		o	
7		o				o		o
8	o				o		o	

3.3.2 角度による評価

目的の図が候補リストの最後の方にある場合、それを探すことに時間がかかってしまう。そこで、生成された候補の並べ替えを行い、候補の選択が容易になるようにした。

並べ替えの基準には、折り線の角度を用いる。折り線の角度 θ は、水平線を基準として $0 \leq \theta \leq 180$ 度で与えられるとする。折り紙では、折り線や辺が成す角として、90度の半分の 45 度や 4 分の 1 の角度である 22.5 度といった値が現れることが多い。これは、折り紙の手順の多くが、紙の辺と辺を合わせるといった、折り紙公理の繰り返しになっていることによる。よって、折り線がそれらの角度である候補のランクを高く設定した。

また、多くの折り紙作品は左右または上下対象であり、対称軸が 0 度や 90 度、対角線にとられる。そのため、一手前と同じように反対側を折るという手順が頻繁に現れる。そのため、最後に折った折り線の角度を保持しておき、その角度および反転したときの角度の場合もランクを高くするようにした。これは、折り線の角度を $0 \leq \theta \leq 180$ までとしたとき、図 17 の矢印の角度に対して、45 度で反転したものと、それぞれを 90 度で反転したものの合計 4 パターンのランクを高くするということである。

具体的には、先頭から次の順に並べ替えて表示する。

- 直前の折り線に関連する角度
- 0 度, 90 度, 180 度
- 45 度, 135 度
- 22.5 度, 67.5 度, 112.5 度, 157.5 度
- それ以外の角度

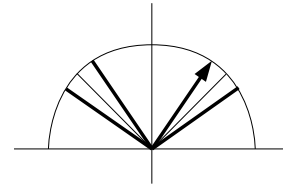


図 17 同時にランクを上げる角度

4 結果と考察

第 3 章で提案した手法を実装したシステムによって得られた結果と、それに対する考察を述べる。実装は Java と Java3D ライブラリを用いて行った。実行環境は、Windows Vista Business, Intel Core2Duo E6750 2.66Ghz, 2048MB RAM を搭載した PC である。まず、このシステムの実行画面を図 18 に示す。

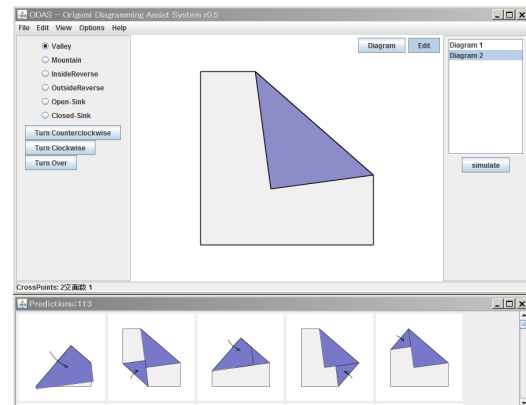


図 18 実装したシステムの実行画面

上にあるのがメインウィンドウであり、中央の大きな図が現在の紙の状態を表示している。頂点をマウスクリックで選び、移動させたい位置で再度クリックすると折り線の入力ができる。また、頂点をドラッグして移動することで、紙のずれを表現した折り図らしい描画が可能になる。左のパネルは、山折りや谷折りなどの入力する折り操作の選択と回転・反転操作が行える。1 ステップ毎の図は、右側のリストで選ぶことができる。下のウィンドウに表示された複数の図は、予測候補である。ユーザは、目的の図があればそれを選択でき、無ければ自由に折り線を入力して、折り図の作成を進めていく。折る方向を示す矢印は、各面の重心や辺の長さを元に、自動的に追加される。

続いて、いくつかの作品を例として、本システムで折り図を作成したときの予測機能の動作を評価する。最初に、比較的簡単な例として「かぶと」の折り図を図 19 に示す。ここで示す折り図は、本システムで作成した図を、各ステップ毎にベクトル画像ファイル (EPS 形式) として出力したものである。一部の図では折り線や矢印の表示が適切でないが、これは、折られる面の数と位置や折り線との距離を考慮していないためである。また、予測機能の動作結果を表 3 に示す。2 列目の採用した図番号とは、選択した候補が、すべての候補の中で先頭から何番目に位置して

いたかを示したものである．この数字が小さいほど，図の選択が容易であり，それに必要な時間が短いと言える．

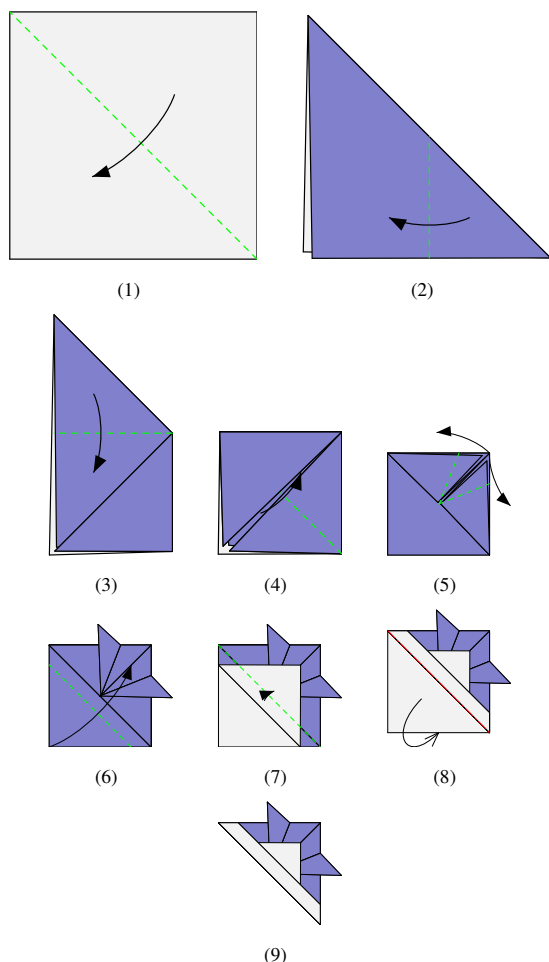


図 19 「かぶと」の折り図

表 3 「かぶと」における予測機能の動作結果

手順	採用した図番号 () 内はソート前)	候補数	処理時間 (s)
1	3(2)	4	0.016
2	3(2)	8	0.015
3	1(10)	44	0.031
4	8(6)	23	0.023
5	-	44	0.016
6	-	595	0.546
7	3(106)	1050	1.050
8	-	1074	1.123

全 8 ステップのうち，予測候補が存在するのは 5 ステップであった．また，最初に提示された候補は図 20 に示す 4 つであった．重複形状が取り除かれて，ユニークな形状のみが表示されていることがわかる．

ランク付けによる並べ替えが効果的に働いているのは手順 3 や手順 7 であるが，手順 4 のように並べ替えが逆に働いてしまう場合もあった．ランク付けの指標として

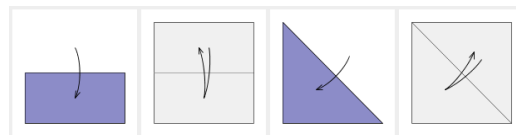


図 20 最初に表示される 4 つの候補

角度のみしか用いていないので，角度が切り替わる部分ではあまり有効でなくなるといえる．

手順 7 の場合は，その前の 2 回の手入力の影響が大きいことも述べておく．手入力では折り線の位置や角度に誤差が発生するため，その後生成される候補にも違いが出てくる．何度か試した結果，候補数で 100 程度の差がでることもあった．

最後に山折りする手順があるが，今回は谷折りしか実装していないため候補は存在しない．ただし，反転してから予測を行えば，候補は生成される．その際，候補数 1161 の中で 1 番目（ソート前は 147 番目）に表示された．「かぶと」の場合は，全体を見れば上手く動作したと言える．

別の例として，「かぶと」より複雑な「鶴」の折り図の一部（図 21）と動作結果（表 4）を示す．鶴の場合は，全 16 ステップ中 7 ステップで予測が使用可能であった．ただし，手順 10 の場合は候補を探すより自分で入力した方が早いと思われ，それを除くと，実際に有効だと判断できるのは 6 ステップである．かぶとでは使われない「中割り折り」という技法が多く用いられるため，候補が利用できない場面が多かった．

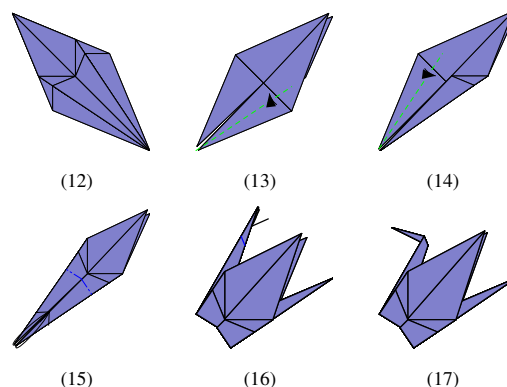


図 21 「鶴」の折り図の一部（手順 12 から手順 17 まで）

以上の動作結果より，問題点としては，手数が増えたときに候補数が膨れ上がることが挙げられる．これは，面の分割に伴って，頂点数が増え辺が細かく分割されることにより，それらの組み合わせが膨大になることが原因である．「鶴」の手順 10 や手順 11 のように，候補数が 1000 近くあると同じランクの候補が増えてしまい，候補の選択が困難になる．紙の重なりが多くなる手順後半では，計算時間の問題も大きくなっている．1 入力毎に数秒の待ち時間が発生しては，操作のインタラクティブ性を大きく損ねる．これに対しては，候補を表示するウィンドウを消すことで，予測機能を使わないことが可能である．ただし，その判断はユーザーが行うものであり，機能のオンオフを切

表4 「鶴」における予測機能の動作結果

手順	採用した図番号 (() 内はソート前)	候補数	処理時間 (s)
1	3(2)	4	0.015
2	4(1)	8	0.016
3	6(10)	32	0.046
4	-	23	0.047
5	-	25	0.016
6	-	82	0.094
7	-	126	0.116
8	-	126	0.096
9	-	84	0.078
10	67(69)	98	0.126
11	10(135)	384	0.655
12	-	524	1.654
13	1(3)	518	1.544
14	1(3)	679	2.309
15	-	1499	21.777
16	-	2009	38.579

り替えることが煩わしい操作となってしまう。

このような問題を軽減するためには、候補数を減らすことが考えられる。公理の適用や折り畳み処理は、比較的簡単な演算によって行われており、候補一つあたりの計算処理を高速化することが難しいためである。折り紙では、「かぶと」の手順 6-7 のように紙の一部分を連続して折ることが多い(手順 2-5 も片方ずつ考えれば連続していると見なせる)。そこで、折り畳む部分を、一手前に折った部分と隣接する部分のみとして候補を生成すれば、候補数を減らせると考えられる。また、使用する公理を少なくしても候補数を減らすことができる。かぶとで採用した候補は、公理 1 または公理 2 からのみ生成されており、公理 3 と公理 4 の候補は使われなかった。鶴で採用した候補は、公理 1-3 から生成されており、公理 4 から生成された候補は使われなかった。例に挙げたのは 2 作品だけであるが、公理 4 の利用場面は少ないのではないと思われる。他にもいくつかの作品を折って調査する必要があるが、候補の利用頻度が低ければ、公理自体を使わないことで候補数の削減が可能である。

5 結論

本論文では、手順予測機能に重点を置いて、折り図作成を支援する手法について述べた。折り操作の入力部分は、関連研究で挙げたものを参考にすれば、改善の余地が十分にある。今回実装していない特殊な折り方の実装がその一つとして挙げられる。しかし、第 4 章で示した作品程度の複雑さであれば、コンピュータ上で折り図を描くことが可能であった。

提案したインタフェースは、折り紙公理に基づき、現在の紙の状態から折り操作を行って得られる形状を、予測候

補として提示するものである。これは、折り始めでは機能したが、手数が増えるにつれて候補数が増加し、候補の選択が困難な状況が発生した。紙の重なりが多くなる手順後半では、計算時間の問題も大きくなっていく。入力支援として実用可能なレベルとしては、計算が実時間で終了することが必須である。数百ステップを必要とする作品においては、実時間で計算を行うことはできないため、予測機能の利用は困難である。

今後の課題としては、探索範囲と利用する公理を絞ることで、候補数を減らすことが考えられる。候補数を常に数十で抑えることができれば計算時間の問題は解決され、対応できる作品の範囲を広げることが可能である。また、ユーザテストを行う必要がある。現在の実装では、折り紙作品の創作家が作る複雑な作品に利用することは難しい。その一方で、クリックするだけで折り紙ができるという点を見れば、図学教育におけるツールとしての可能性もあると思われる。これらを考慮し幅広い視点で評価を進め、さらなる操作性の改善に努めたい。

参考文献

- [1] Robert J. Lang: Origami Design Secrets: Mathematical Methods for an Ancient Art, AK Peters, Ltd. (2003).
- [2] Hatori Koshiro “K’s Origami: Origami Construction” <http://origami.ousaan.com/library/conste.html>
- [3] Miyazaki Shinya, Yasuda Takami, Yokoi Shigeki, Toriwaki Junichiro, “An Origami Playing Simulator in the Virtual Space”, The Journal of Visualization and Computer Animation, vol.7(1), pp.25-42, 1996.
- [4] 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.
- [5] 三谷純, “2 次元バーコードを用いた紙の折りたたみ構造の認識とそのモデル化”, 情報処理学会論文誌, Vol.48, No.8, pp.2859-2867, 2007.
- [6] 三谷純, “折紙の展開図専用エディタ (ORIPA) の開発および展開図からの折りたたみ形状の推定”, 情報処理学会論文誌, Vol.48, No.9, pp.3309-3317, 2007.
- [7] John Szinger, “The Foldinator Origami Modeler and Document Generator.” 3OSME, 2001.
- [8] Jack Fastag, “eGami: Virtual Paperfolding and Diagramming.” 4OSME, 2006.
- [9] Takeo Igarashi, John F. Hughes, “A Suggestive Interface for 3D Drawing” ACM UIST’01, Orlando, Florida, November 11-14, 2001, pp.173-181.
- [10] K. Weiler: Edge-Based Data Structures for Solid Modeling in Curved-Surface Environments, IEEE Computer Graphics and Applications, 1985
- [11] Hull, T.: On the Mathematics of Flat Origamis, Congressus Numerantium, Vol. 100 (1994), pp. 215-224.