

2次元バーコードを用いた折紙の構造認識とモデル化

三 谷 純†

本稿では、折りたたまれた紙を撮影したデジタル画像から、その折りたたみ構造を認識し、計算機内にモデル化する手法を提案する。提案する手法では、裏と表の両面に2次元バーコードが格子状に複数印刷された紙を用い、撮影された画像に含まれるバーコードの位置関係から紙の折りたたみ構造を推定する。一般に2次元バーコードは文字列情報や固有の識別番号などを格納し「それが何であるか」を示すために用いられるが、本手法ではこれを紙の上に分散配置し、配置された位置に関する情報を格納することで、「紙がどのように折りたたまれているか」を認識することに応用している点に特徴がある。具体的には画像に含まれるバーコードの位置関係から折り線の位置と、どのような折り操作が行われたかを推定し、計算機内のモデルの更新を行う。本手法をPC上に実装し、実際に折りたたまれた紙を撮影した画像から、その構造を認識して計算機内にモデル化できることを確かめた。

Recognition and modeling of Origami using 2D bar codes

JUN MITANI†

This paper proposes a method for recognizing and modeling a folded paper from digital images which photoed the paper. 2D bar codes are printed in grid form on the both sides of the paper. The folding structure is figured out from the positional relation of the codes in a photo. Though the 2D bar codes are generally used for recognizing what the object is, they are used for recognizing how the paper is folded. In the method proposed in this paper, position of a folding line is estimated firstly, the detail information of folding operation is decided and then the model in a computer is updated. The system is implemented on a PC and the validity is confirmed.

1. はじめに

紙を折って形を作成する「折紙」は日本に古くから伝わる文化の1つであり、教育の場での活用や趣味の1つとして幅広い世代に親しまれている。近年では世界的にも広く認知され、その学術的な研究も多くされている。特に最近では計算機環境の発達と普及により、折紙を計算機で扱う研究も多い。

ところで、折紙作品の折り方を他者に伝える方法は古くから存在する「折り図」に依存している。折り図とは、折り工程を図で表現したものであるが、この折り図の作成は労力のかかる作業であり、このことが様々な作品を効率的に他者に伝達することを困難にしている。そこで、折紙の形状を計算機内に構築することができれば、折紙の構造を任意の角度から眺めることができ、またアニメーション表示による折り工程の提示などへ発展させることができると考えられる。

折紙の形状を計算機内に構築する場合には、形状を平面多角形の集合などで表現した幾何情報と、それぞれの多角

形の接続関係を表す位相情報を用いてモデル化することが一般的である。しかし、意図した形状のモデルを計算機内に構築するにはCGやCADの知識が必要となり、初学者には困難な場合が多い。一方で現実世界の紙を手で折って形を作ることは、いわゆる「折紙」として誰でも体験したことがあるほど容易な事である。そこで本稿では、手作業で実際に折りたたんだ紙をデジタルカメラで撮影し、その画像から自動で計算機内にモデルを構築する新しいインタフェースを提案する。これにより、折紙を計算機内にモデル化するための負担を軽減することができる。これを実現するには、画像解析や認識に関する研究の成果を活用することが考えられるが、折りたたまれた紙の写真画像から得られる情報は輪郭線情報だけであり、これらから折りたたみの構造を推測するのは難しい問題である。本研究では、近年広く普及している2次元バーコードを、表と裏の両面に印刷した紙を準備し、これを用いた折紙をデジタルカメラで撮影することで、その構造を容易に認識し、計算機内にモデルを構築する方法を提案する。

本研究は折りたたまれた紙の構造を認識しモデル化するための新しい方法を提案すると共に、2次元バーコードを用いた新しいインタフェースを提案するものである。

本稿の第2章では関連する研究を紹介し、第3章では折

† 筑波大学大学院システム情報工学研究科
Graduate School, University of Tsukuba

りたたまの構造を推定する原理について述べる．第 4 章では本手法の詳細を述べ，第 5 章で結果を，第 6 章でまとめを述べる．

2. 関連研究

2.1 折紙のモデル化

日本に古くから伝わる折紙の手法は，正方形の紙を折り曲げることによって様々な形を作るものであり，幾何の分野における研究題材として多く取り上げられている¹⁾²⁾．また，国内に留まらず海外でも研究が行われており，近年では対象とする形状を計画的に生成する「設計」の概念が折紙にも導入されるようになってきている³⁾．

折紙を計算機で扱う研究として，内田ら⁴⁾は紙の物理的な制約条件を元に，折紙の展開図を構成する幾何学的要素から出来上がりを推論するプログラムを提案した．Miyazaki ら⁵⁾は，計算機を用いて折紙を対話的に操作する手法を提案した．紙を折る操作によって折紙の形状が逐次変化する際の，データ更新の手法を提案している．折紙の形状を計算機内に保持するデータ構造はシンプルであり，折り手順を含めた情報が，二分木を用いて保持される．ここで提案されたデータ構造を含むプログラムのソースコードは Web 上で公開されており⁶⁾，本研究で提案するシステムでは実装の一部にこれを使用した．Kato ら⁷⁾は，教本に含まれる「折り図」の画像を計算機で解析することで，折り操作を推定し，それを元に計算機内の折紙モデルを更新する手法を提案した．この手法は図中に含まれる矢印や折れ線の情報を活用しているため，写真画像からのモデル構築には適用することができない．Ju ら⁸⁾は計算機とのインタフェースに関する研究の中で，無線タグを埋め込んだ紙を使用した折紙の教示を例題として扱った．無線タグの空間位置を元に折紙の形状を計算機が把握できるようになっている．この手法は本研究の目指すところと共通する点が多いが，無線タグと読み取り機器という大がかりな装置が必要となる点で一般的には扱いにくいという問題がある．三谷ら⁹⁾は計算機内に構築された折紙モデルに対し，その構造を理解しやすくするために，厚みや頂点のずれを追加して表示する手法を提案した．本研究では計算機内のモデルを表示する際にこの手法を用いる．

2.2 2次元バーコードの利用

一般にバーコードは，それが付随する個体の識別 ID を格納するため，または個体に関する情報を格納するために使用される．近年では，従来の 1 次元バーコードよりも多くの情報を格納できるものとして 2 次元バーコードが開発され，様々なものが存在する¹⁰⁾．これらは，アパレル業界や医療機関，工場などでの物品管理用途などに幅広く使用されている¹¹⁾．また，ロボットの環境認識への応用¹²⁾など新しい用途への活用も研究されている．インタフェースの分野においても，計算機に物体を認識させる上で広く活用されている．

ところで，2 次元バーコードの 1 つである QR コードは，早くから規格化され仕様が公開されている (ISO/IEC18004 および JIS-X-0510) ことと，高速な認識に適していることから，現在では携帯電話のカメラに標準で読み取り機能が備わっていることが多い．QR コードを読み取るための API は市販されており，新規システムへも容易に利用できる環境が整っている．そこで本稿で提案するシステムでは，折りたたまの認識に用いる 2 次元バーコードに QR コードを使用し，読み取り機能の実装には 13) の API を用いることとした．

3. 折りたたみ構造のモデル化の原理

3.1 状態の表現と折り操作

本研究では計算機内に折りたたまれた紙の構造を保持するために，Miyazaki らの提案する折紙モデルを使用する．このモデルでは，Stage と呼ばれる構造体が折紙の状態を表すために使用される．1 回の折り操作の情報と，現在の Stage の情報から，折り操作を行った後の状態を表す新しい Stage が構築される．初期状態を $Stage[0]$ ， n 回目の折り操作を $Fold[n]$ ， n 回の折り操作を行った後の状態を $Stage[n]$ と表すと，これらの関係は図 1 のようになる．

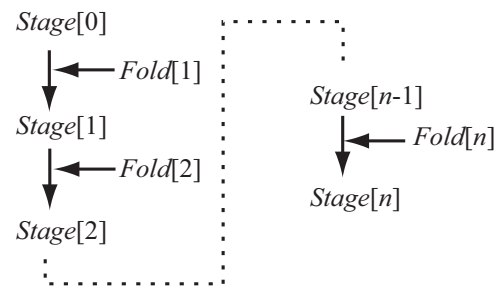


図 1 折り操作 Fold による Stage の更新
Fig. 1 Updateing of Stages by Fold operations.

本稿で扱う折紙の状態には，実際に人の手によって折られたものと，計算機の中に構築するモデルの 2 つが存在する．これらを区別するために，以降ではそれぞれを $Stage^{Real}[n]$ と $Stage^{Model}[n]$ と表すこととする．

なお，本手法では平坦に折りたたまれた紙だけを対象とし，紙飛行機や折り鶴の翼を広げた状態のような立体的な構造を持つものは対象としない．そのため，折り操作 $Fold[n]$ には，折り線の位置に関する情報と，それが山折か谷折りのどちらであるかの情報，および紙が複数枚重なっている箇所 (図 8 の例を参照) については何枚目が折り操作の対象となっているかの情報が含まれていればよい．折り角は常に 180 度であるので，角度に関する情報は必要ない．

3.2 処理の流れ

本稿では図 2 に示すように，折る操作を一回行う毎にデジタルカメラで撮影し，その都度計算機内のモデルを更新することで，複数回の折り操作で折りたたまれた紙のモデ

ルを計算機内に構築するシステムを提案する．

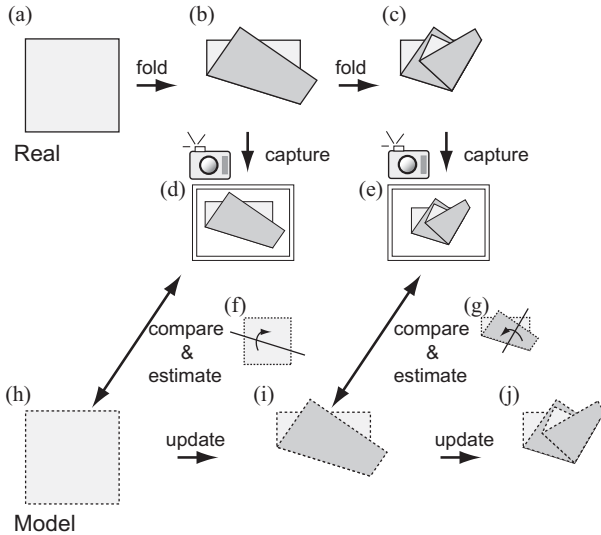


図 2 処理の流れ
Fig. 2 Flow.

図 2 に示した処理の流れを以下にまとめる．

- (1) 折る前の紙 (a) を計算機内にモデル (h) として保持する．
- (2) 折り操作を 1 回行った様子 (b) をデジタルカメラで撮影し (d), それと (h) を比較してどのような折り操作が行われたかを推定する (f)．
- (3) 推定 (f) に基づいて計算機内のモデル (h) を更新し, モデル (i) を生成する．
- (4) 引き続き, 2 回目の折り操作を行った様子 (c) をデジタルカメラで撮影し (e), それを (i) と比較してどのような折り操作が行われたかを推定する (g)．
- (5) 推定 (g) に基づいて計算機内のモデル (i) を更新し, モデル (j) を生成する．

上記の (4)(5) の処理は (2)(3) の処理の繰り返しであるが, 必要に応じて繰り返しを増やすことで, 回数の多い折り操作によって作成される形に対応する．つまり, $Stage^{Real}[n+1]$ の写真と $Stage^{Model}[n]$ から $Fold[n+1]$ を推測し, $Stage^{Model}[n]$ と $Fold[n+1]$ によって $Stage^{Model}[n+1]$ を構築することが, 本研究で提案する手法である．

3.3 折り線推定の原理

図 3(a) のように紙を折り線 l で谷折りし, 図 (b) の状態になった場合, この操作を計算機内のモデルに反映するためには折り線 l の位置と折る向き (谷折りまたは山折り) の情報が必要である．図 (c) のように, モデル上の頂点 P が移動した後の点 P' の位置がわかる場合, この 2 点の垂直二等分線を折り線と推定できる．しかし, $Stage^{Real}[n+1]$ の写真と $Stage^{Model}[n]$ の間で対応する頂点を何の工夫も無しに画像処理だけで見つけ出すのは容易でない．頂点位置に色の付いたシールを貼るなどのマーキングによって, 対

応付けを容易にすることが可能であるが, 折り操作を進める度に頂点の数は増えるので (図 (a) では頂点数 4 であるが図 (b) では 6) 現実的な対応は難しい．

ところで, 頂点の位置でなくても図 (c) の点 Q と点 Q' ように, モデル上の特定の点の位置がわかるようにマークを付けておけば, この 2 点の垂直二等分線から折り線を推定することができる．ただし, この点 Q はカメラが可視で折り操作によって位置が変わる場所 (図 3 の例では直線 l の下側) に存在する必要があるため, 事前に特定の位置に配置しておくことと任意の折り方に対応できないという問題がある．

そこで本研究では, 事前に紙の上に複数の QR コードを分散配置し, その QR コードを目印に利用する．状況に応じて目印とするコードを切り替えることで上述の問題を解決する．QR コードには, QR コード自身が存在する (紙上の) 位置に関する情報を格納しておくことで, 色の付いたシールを貼るような煩雑な手間を無くすることができる．QR コードの配置方法や折れ線の推定方法などについては次章で詳細を述べる．

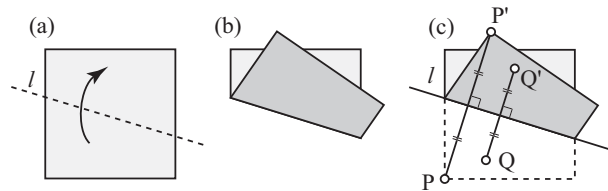


図 3 折り線位置推定の原理
Fig. 3 Estimation of the position of a fold line.

4. 手法の詳細

4.1 QR コードの配置

図 4 のように紙の両面に QR コードを格子状に配置する．それぞれの QR コードには, その位置と裏表の別を表すための 5 桁の数字をエンコードする．数字の 1 桁目は表裏の別 (0:面, 1:裏) とし, 2 桁目と 3 桁目で何列目であるか, 4 桁目と 5 桁目で何行目であるかをそれぞれ 2 桁の整数で表現する．例えば表面の 6 列 7 行目の QR コードには「00607」という数列を格納し, 裏側の同じ場所には「10607」を表す QR コードを配置する．これにより, 写真画像中に含まれる QR コードをデコードすることで, その紙上での位置と面の表裏の別を識別できる．

紙の横幅と縦幅がそれぞれ W, H であり, QR コードが横方向に U , 縦方向に V だけ等間隔に配置され, デコードされた QR コードが i 列 j 行目であった場合, この QR コードの中心は紙の左上を原点として $(\frac{W}{U}(i+0.5), \frac{H}{V}(j+0.5))$ の位置にあることがわかる．紙の上に配置する QR コードの数 (または大きさ) は, 写真画像からの認識成功率と, 折りたたみ形状を推定する精度のトレードオフで決定される

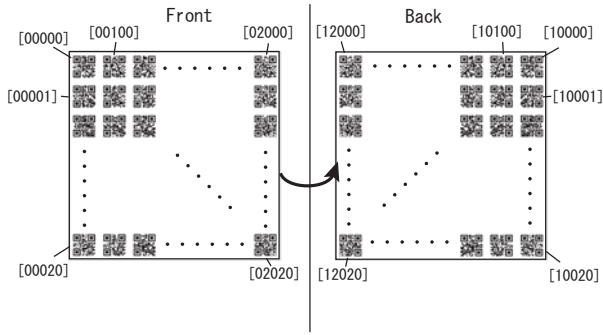


図4 QRコードの配置(正方形の紙に縦横20ずつ配置した場合)
Fig. 4 Arrangement of QR codes (placing 20 × 20 codes on a square sheet).

が、次章で述べる実験結果では20cm四方の紙に20×20のコードを配置することで実用的な成果を出すことができた。

4.2 撮影

QRコードを配置した紙を通常の折紙のように折り、折った様子をデジタルカメラで撮影する。ただし、本手法では折る前と折った後のQRコードの位置の変化に基づいて折り線の位置を推定するため、QRコードの位置関係の変化が把握できる方向から撮影する必要がある。つまり、図5(a)の点線で奥に折るような場合、(b)のように一方の面が他方の面に完全に隠された状態ではなく、(c)のように両方が見える側から撮影する。なお、図3のような折り方の場合は、一方が他方を完全に隠すことはないため、どちらから撮影しても構わない。

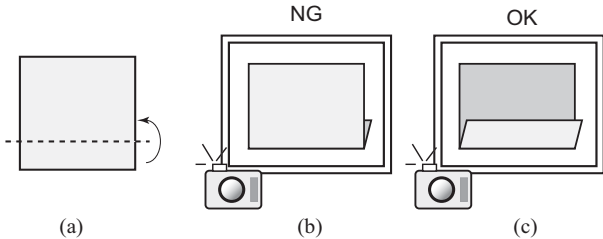


図5 撮影
Fig. 5 Taking a photo.

4.3 座標変換

本稿で提案する手法では、 $Stage^{Real}[n+1]$ の写真と計算機内に構築されているモデル $Stage^{Model}[n]$ を比較して折り操作を推定するため、まず両者で座標系を統一する必要がある。ここでは、写真的座標系(写真座標)から計算機内のモデルの座標系(モデル座標)への変換を行い、両者をモデル座標で比較することとする。カメラレンズの収差による歪みが無いという仮定の下、変換行列を A とすると、 A は拡大率 s のスケール変換と角度 θ による回転および (d_x, d_y) の移動量による平行移動を伴う次のような 3×3 の同次変換行列で表現できる。

$$A = \begin{bmatrix} s \cos \theta & -s \sin \theta & d_x \\ s \sin \theta & s \cos \theta & d_y \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

上記行列の未知数は4であるため、2つのQRコードの写真座標での位置とモデル座標での位置がわかれば A を導出できる。基準とする2つのQRコードは図6の点 X と点 Y のように、 $Stage^{Real}[n+1]$ の写真画像に含まれ、なおかつ $Stage^{Model}[n]$ の中でその点での法線の向きが同一(この場合は手前に向かう方向)であればよい(図中の点 Z は $Stage^{Model}[n]$ の中で裏側に存在し、法線の向きが奥に向いているため点 X と点 Z のペアは基準に使用できない)。このような条件を満たす任意の2つのQRコードから変換行列を算出し、写真に写ったQRコードをモデル座標に変換する。ただし、誤差を小さくするために互いに最も離れている2点を選択することとする。

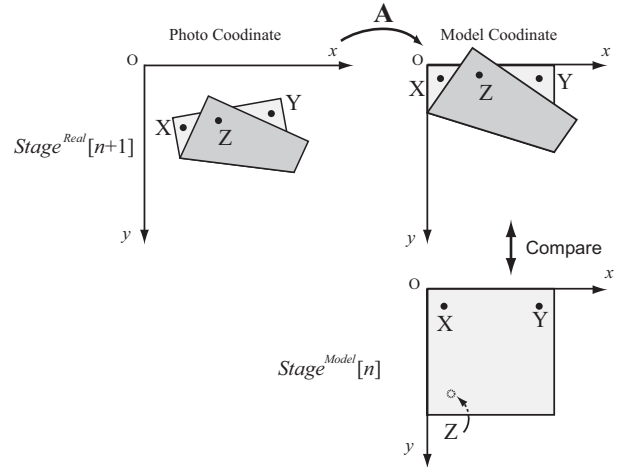


図6 座標変換
Fig. 6 Coordinate transformation.

4.4 折り線位置の推定

前出の変換を用いて $Stage^{Real}[n+1]$ の写真画像に含まれる各QRコードをモデル座標に変換し、 $Stage^{Model}[n]$ での各座標値と比較する。このときに、誤差によって値が多少異なるものが存在するが、特に大きく異なるQRコード(例えば図6の点 Z)が存在する場合、これは折り操作によって位置が変わったものであると判断できる。このようなコードのうち、最も移動量の大きいものについて、 $Stage^{Model}[n]$ での座標と $Stage^{Real}[n+1]$ の写真画像から算出した座標を図3に示す点 Q と点 Q' として用いることで、折り線の位置を推定する。なお、本システムを実装する際には、折り操作による位置の変化ではなく誤差によるものだと判断する閾値は紙の幅の5%程度に設定した。写真画像に含まれる全てのQRコードが折り操作によって位置が変化しなかったと判断された場合は4.7節で述べる「二等分折り」がされたものと判断する。

4.5 折り方向の推定

折り線の位置が決まっても、折り方には図7の(a)と(b)のように谷折りと山折りの2通りが存在する。この折り方向を決定するために、まず計算機モデルに対して両方の折り方を試し、それぞれの折り方を行ったときに外部から見えるQRコード群を算出する。このQRコード群と、実際に写真から抽出されたQRコード群を比較し、両者に含まれるQRコードの数が多い折り方を正しい折り方として採用する。

なお、このようにして複数の折り方から最適なものを決定する方法を、以降では簡単のために「可視コード群比較法」と呼ぶこととする。

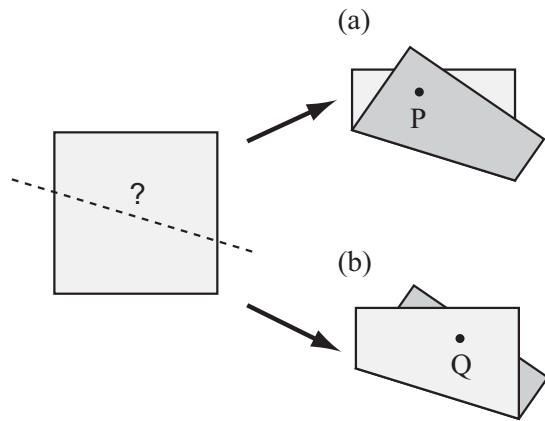


図7 谷折りと山折り
Fig. 7 Folding direction: valley and ridge.

4.6 一重折りと多重折りの判別

図7のように、折り曲げられる対象となる面が1つである場合、前節で述べた折り方向が決定すれば、折り方は一意に定まるが、図8のように折り曲げの対象となり得る面が複数存在する場合、折る対象となる面と対象とならない面を明らかにする必要がある。図7(b)のように、1枚の面だけを折る場合を一重折りと呼び、(c)のように複数の面を同時に折る方法を多重折りと呼ぶ。図7(a)の場合は(b)、(c)の2通りしか存在しないが、場合の数は折ることが可能な面の数に依存する。

このように折る対象となる面が複数ある場合は、取り得る全ての場合を計算機モデルに対して試し、前節と同様に「可視コード群比較法」で最適な折り方を決定する。

4.7 二等分折りへの対応

図9のように、折り線によって元の形が二等分されるような折り方を「二等分折り」と呼ぶこととする。この場合、折りたたまれた形をどちら側から見ても、相対的に場所が変化したQRコードが存在しないため、4.4節で述べた方法では折り線の位置を推定することができない。そこで、4.4節の方法で折り線位置が推定できなかった場合は、二等分折りがなされたものと判断し、可能な全ての二等分折りを

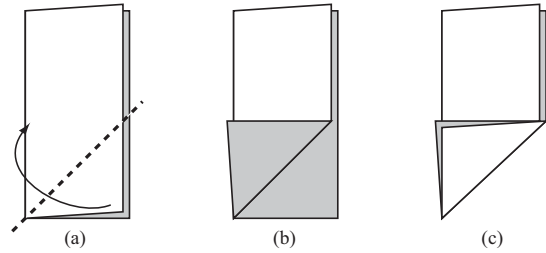


図8 一重折りと多重折り
Fig. 8 Single folding and multiple folding.

計算機モデルに対して試し、「可視コード群比較法」で最適な折り方を決定する。なお、二等分折りが行われる場合は、少なくとも1つの頂点が他の頂点に重なるように移動するので、全ての頂点の組み合わせから得られる二等分線について、それぞれを折り線であると仮定した試行を行う。この方法だと、二等分折りでないケースも含まれるが、可能な二等分折りは必ず含まれ、また実装が容易である利点がある。

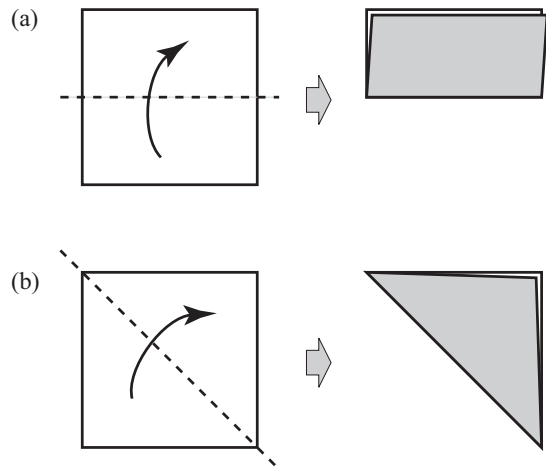


図9 二等分折り
Fig. 9 Median folding.

4.8 折れ線を元にした計算機モデルの更新

前節までに述べた手法により、折り操作の情報 $Fold[n+1]$ を決定できるため、これと $Stage^{Model}[n]$ を元に $Stage^{Model}[n+1]$ を生成して計算機内のモデルを更新する。本システムではデータの保持に Miyazaki らの提案を採用しているため、データの更新方法についても同様に Miyazaki らの提案手法を用いる。

5. システムの実装結果

本稿で提案するシステムを CPU Pentium Mobile Processor 2.0GHz, RAM 1.0GB の PC 上に実装し検証を行った。QRコードはバージョン1(21セル四方・最もサイズの小さなバージョン)、誤り訂正能力レベルH(最も高いレベル)を用い、20cm四方の紙に縦横20ずつの計400を配置

した。この紙に対して折り操作を行う度に、 2048×1536 ピクセルの解像度でデジタル撮影した。

図 10(a) は検証に使用した紙を撮影したものであり、(b) は本システムのアプリケーションウィンドウで初期状態の折紙モデルを表示した様子である。折紙モデルの表示には、文献 9) で提案されている厚みを強調する手法を用い、構造を把握しやすいようにしている。

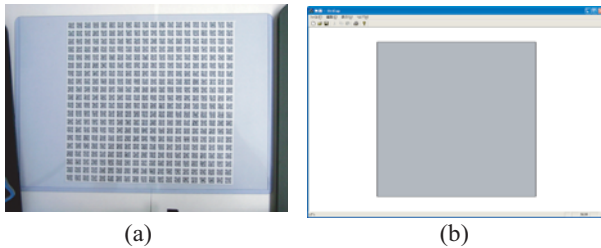


図 10 初期状態の写真とモデル表示

Fig. 10 A photograph and a model display of the initial state.

5.1 一般的な折り方

本稿で提案する折り線の推定手法が妥当であることを検証するために、一般的な折り方（二等分折りでなく、かつ多重折りの場合分けを考慮する必要のない折り方）を 2 回行った結果を図 11 に示す。(a) は撮影された写真の画像である。なお、システムではデジタルカメラで撮影したものをそのまま入力として使用しているが、(a) はわかりやすくするために必要な箇所だけをトリムしている。結果の (b) は、本手法で得られたモデルをアプリケーションウィンドウに表示したものである（以降の図も同様である）。実際の紙の形状を計算機のモデルに反映できたことが確認できる。

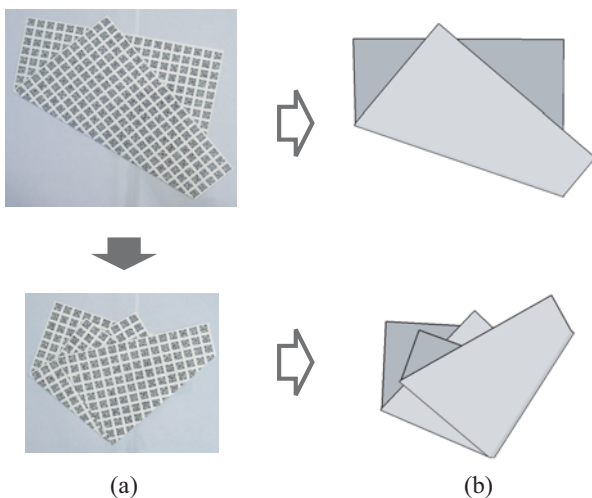


図 11 一般的な折り方

Fig. 11 General case.

5.2 二等分折り

長方形に二等分折りし、さらにそれをもう一度二等分折りした結果を図 12 に示す。

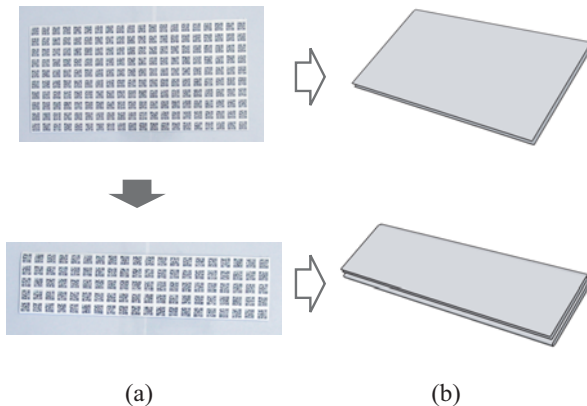


図 12 二等分折り

Fig. 12 Median folding.

5.3 一重折りと多重折り

一重折りと多重折りの存在する例題として、「ヨット」を折った結果を図 13 に示す。1 回目の折操作は三角形への二等分折りであるが、2 回目の折り操作では、折り操作の対象となる面が 1 枚なのか 2 枚なのかを識別する必要がある。

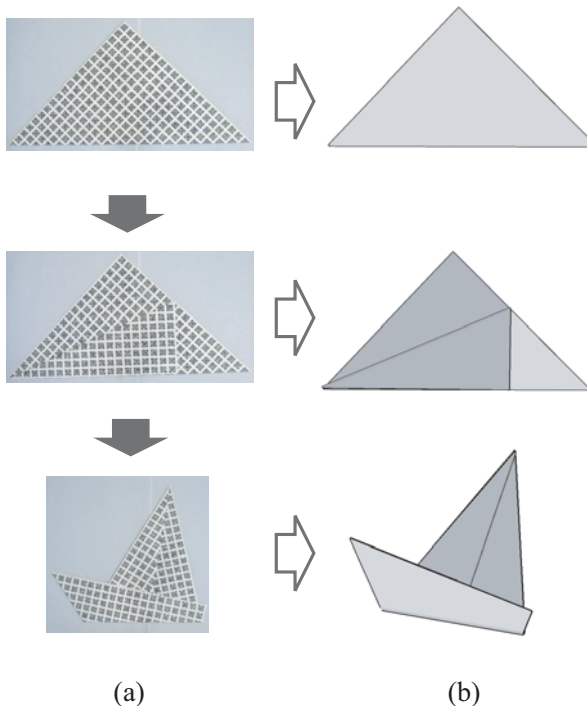


図 13 一重折りと多重折りの判別が必要な例（ヨット）

Fig. 13 An example that requires single folding and multiple folding (Yacht).

5.4 複合的な例

より複雑な例として「セミ」を折った結果を図 14 に示す。最後まで折るには 9 回の折り操作が必要になるが、紙幅の都合上途中の一部を省略している。最終的にできあがった形を正しく計算機に取り込むことができた。

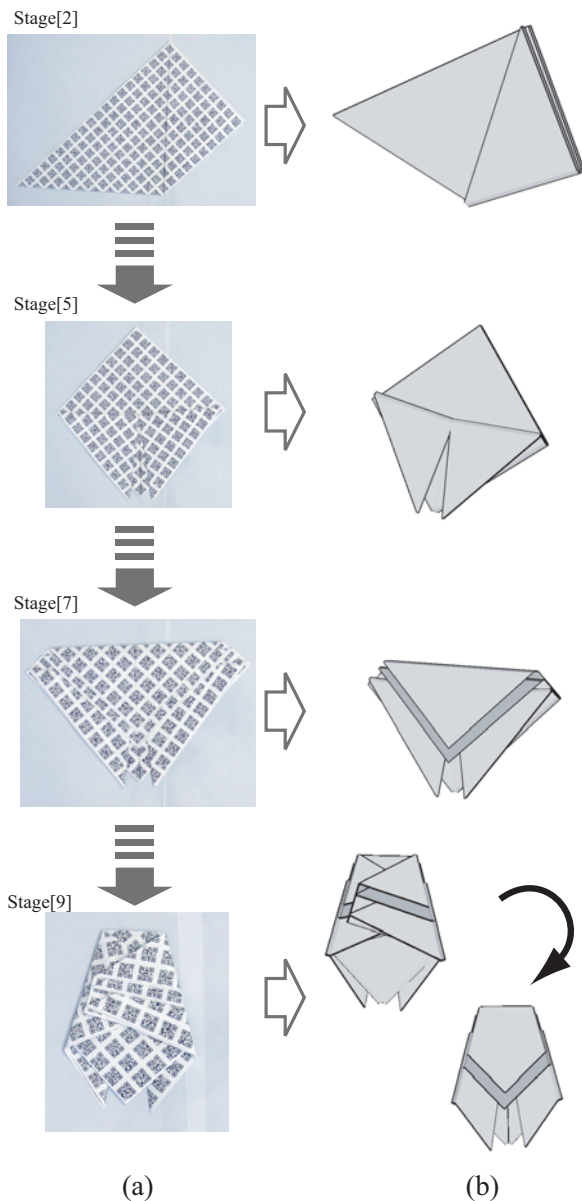


図 14 複合的な例(セミ)
Fig. 14 A composite example (cicada).

5.5 評価

QR コードの読み取りには既存の API を使用し、400 個のコードが含まれる図 10(a) の写真を処理するのに要した時間は 20 秒であった。また、そのときに認識できたコードの数は 391 (全体の約 97%) であった。使用した API には読み取り時に設定可能なパラメータが複数あり、その設定によって読み取り時間と精度は異なったものになるが、今回は

精度を優先することとした。図 14 下段の最も QR コードが少ない写真は約 1 秒で処理することができ、目視で全体が写っていることが確認できる 17 個のコード中 16 個 (94%) を正しく読み取ることができた。

読み取ったコードから折紙モデルを構築する処理は、図 13 の最終ステップで 0.15 秒、図 14 の最終ステップで 3.34 秒であった。今回は特に高速化の工夫を施していないため、例えば「可視コード群比較法」での可能性の無い折り方を効率よく省くなどの改善で、より高速にモデルを構築することが可能であると考えられる。

システムでは 1 つの折り操作に対して 1 つの写真を読み込む必要があるが、複数の写真を連続的に読み込むことで複数回の折り操作を一括で処理できるため、ユーザーは読み込むファイルを最初に 1 度指定するだけでよい。従来のマウスとキーボードを使用した折紙モデルの操作に比べ、極めて簡単に折紙の形を計算機内に構築することが可能である。実際に紙を折りながら写真を撮影することが若干の手間ではあるが、7 章で述べるように動画像から QR コードを抽出できるようにすればリアルタイムでの取り込みへ改善可能であると考えられる。

6. まとめ

本稿では QR コードを紙の両面に印刷し、それを撮影したデジタル画像から折紙のモデルを計算機内に構築する新しいインターフェースを提案した。提案する手法を PC 上に実装し、検証することで簡単な折紙であれば、本手法が有効に機能することを確認した。本システムは、一般的な PC とプリンタ、およびデジタルカメラがあれば十分である。特別な装置を必要としない、安価に構築可能な環境で、現実世界での紙を折る操作で計算機内の仮想的な折紙モデルを更新できることを示せた。

ただし、本手法では平坦に折りたたまれるケースのみを対象としているため、紙飛行機や紙風船のような立体的な構造を持つ形には対応できないという制約がある。さらに、図 15 に示すような「折り込み」操作を扱えないという制約も存在する。

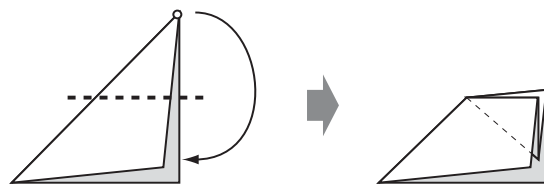


図 15 折り込み
Fig. 15 Tack-In folding.

7. 展望

本システムでは立体的な構造を持つ折紙は扱えないが、例えば複数のカメラから撮影することで問題が解決できるか

もしない「折り込み」操作に対しては、原理的に写真だけからの認識は困難であるため、人の手による対話的な指示を受け入れる柔軟性を追加する必要があるだろう。また、静止画ではなく動画を継続的にキャプチャすることで、実際の折り操作とリアルタイムに連動して折紙モデルを更新させるインタフェースを実現することも可能であろう。

本稿では、計算機内でモデル化する対象に折紙を用いたが、これ以外にも一定の拘束条件のある物体（例えばリンク構造を持つ物体など）に QR コードを分散配置することで、その物体がどのような状態にあるのか認識することにも応用可能と考えられる。

謝辞 本システムの実装において、QR コード読み込みにソフトアドバンス株式会社の「QR コードソリューションキット」を、折紙の構造保持のために Web 上で公開されている宮崎慎也氏のプログラムコードを使用させていただいた。また、本研究は日本学術振興会科学研究費補助金（若手研究 B，課題番号 17700131）により助成を受けて行ったものである。ここに感謝の意を表する。

参 考 文 献

- 1) 川崎敏和: バラと折り紙と数学と, 森北出版株式会社 (1998).
- 2) 深川英俊: 折紙の数学, 森北出版株式会社 (2002).
- 3) Lang, R. J.: *Origami Design Secrets: Mathematical Methods for an Ancient Art*, AK Peters, Ltd. (2003).
- 4) 内田忠, 伊藤英則: 折り紙過程の知識表現とその処理プログラムの作成, 情報処理学会論文誌, Vol. 32, No. 12, pp. 1566-1573 (1991).
- 5) Miyazaki, S., Yasuda, T., Yokoi, S. and Toriwaki, J.: An Origami Playing Simulator in the Virtual Space, *The Journal of Visualization and Computer Animation*, Vol. 7, No. 1, pp. 25-42 (1996).
- 6) 宮崎慎也: 折り紙シミュレーション. <http://www.om.sccs.chukyo-u.ac.jp/main/research/origami/indexj.html>.
- 7) Kato, J., Watanabe, T., Hase, H. and Nakayama, T.: Understanding Illustrations of Origami Drill Books, 情報処理学会論文誌, Vol. 41, No. 6, pp. 1857-1873 (2000).
- 8) Ju, W., Bonanni, L., Fletcher, R., Hurwitz, R., Judd, T., Post, R., Reynolds, M. and Yoon, J.: Origami Desk: Integrating Technological Innovation and Human-centric Design., *Proc. of Designing Interactive Systems 2002*, pp. 399-409 (2002).
- 9) 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).
- 10) 岩間司, 白江久純, 西浦稔修, 鈴木こおじ, 上釜和人: 2次元バーコードを用いた郵便情報システムに関する調査研究, 郵政研究月報, No. 8, pp. 24-38 (2000).
- 11) メトロロジック・ジャパン株式会社: 2次元バーコードの応用事例. <http://www.metrologic.co.jp/apply/index.html>.
- 12) 太田順, 新井民夫: 二次元コードを用いたサービスロボット<バーコードを利用したロボット作業環境の整備>, 月刊バーコード, Vol. 12, No. 5, pp. 12-16 (1999).

- 13) ソフトアドバンス株式会社: QR Code Solution. <http://www.softadvance.co.jp/qr/index.html>.