

平面多角形の集合による「折り紙建築」モデルの表現と 計算機による設計支援

三 谷 純[†] 鈴 木 宏 正[†]

本稿では折り紙建築のモデルを平面多角形の集合で表現し、これを計算機内に保持する手法、および計算機によって設計を支援する手法を提案する。折り紙建築とは3次元形状を紙で表現したものであり、それを2つ折りで平面へ折り畳むという特徴を持つ。また折り畳み後は、再度開くことで、いわゆる「ポップアップカード」のように3次元形状を立ち上げることができる。本稿では特に1枚の紙から、切り込みと折り曲げのみで形を作成し、直角に開いたときに立体が立ち上がるものを対象とする。本手法では、水平および垂直に配置される平面多角形の集合で折り紙建築の形状を表現し、これをユーザが対話的に入力することで、折り紙建築の設計を直感的に行えるようにする。編集操作で新しい面が追加されるごとに、展開図上において多角形の集合演算を用い、1枚の紙から作成されるという制約がつねに満たされることを保証している点が、本手法の大きな特徴である。また、開口部の作成、展開図の生成、形状の妥当性の判定、および開閉アニメーション表示に関する手法についてもまとめた。これらの手法をPC上に実装した折り紙建築モデルの設計支援システムを開発した。このシステムを用いて実際に折り紙建築を設計し、生成された展開図から紙模型を作成することでその有効性を確かめた。

Computer Aided Design for Origamic Architecture Models with Polygonal Representation

JUN MITANI[†] and HIROMASA SUZUKI[†]

Origamic Architecture (OA) is a piece of folded paper from which a three dimensional structure “pops up” when it is opened. It is similar to “Pop Up story book”, but its unique feature is that it is made by only giving cuts to a piece of paper. On the other hand, because of this feature it requires good experience to design an OA. We propose a method to support design of OA with a computer. The OA is modeled using a set of planar polygons. The model must satisfy such conditions that it constitutes a valid, realizable OA. A unique point of our method is applying boolean set operations to the polygons on the unfolded pattern to guarantee that the model can be made with single sheet of paper. And a procedure to check the validity is also presented. We also propose methods to create opening, to generate unfolded pattern, and to display folding animation. We implemented a system based on these methods and demonstrated its usefulness for creating OA. With our system, we allow designers to intuitively design these OA models and easily generate the unfolded patterns.

1. はじめに

折り紙建築(Origamic Architecture, OA)とは3次元形状をいわゆるポップアップカードで表現する手法であり、茶谷正洋氏によって考案された紙工作の手法¹⁾である。紙で3次元形状を表現でき、それを2つ折りで平面へ折り畳むことから、飛び出す絵本や挨拶状など、我々の身近なところでも使用されている。折り紙建築には複数のパーツを組み合わせることで、

180度を開いたときに形が立ち上がるものなど、製作者の創意工夫による様々な種類のものがある^{2),3)}が、本稿では特に一般的に広く普及している形式のものとして、1枚の紙から構成され、切り込みと折り曲げだけを用い、90度を開いたときに3次元形状が立ち上がるものを対象とする。

ここで対象とする折り紙建築は文献4),5)などで様々な形のもものが紹介されている。折り紙建築では立体の側面が表現できないなどの制限があるものの、簡単な工作手順で手軽に立体形状を表現できる優れた特長がある。しかしこれらの折り紙建築は「2つ折りで折り畳める」「1枚の紙から構成される」「90度を開い

[†] 東京大学大学院工学系研究科精密機械工学専攻
Department of Precision Machinery Engineering, The
University of Tokyo

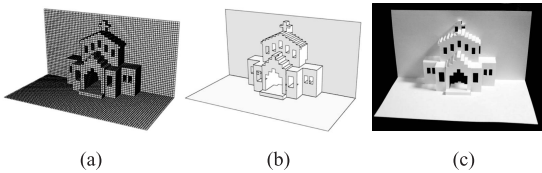


図 1 ボクセルによる折り紙建築の表現 . (a) ボクセルの集合, (b) 折り紙建築の形状の CG, (c) 工作した折り紙建築の写真
Fig. 1 OA with voxel representation. (a) Set of voxels, (b) CG image of OA, (c) Photo.

たときに形が立ち上がる」という厳しい形状の制約があるため、意図した形を作るとは非常に難しい。そのため、折り紙建築の作成には図形の知識と経験が必要とされ、熟練者の手による試行錯誤によって設計されている。

折り紙建築を計算機で扱う研究は今まであまりなされていないが、茶谷ら⁶⁾は折り紙建築の形状をあらかじめプログラムに組み込むことで、その開閉の様子をCGで表示する例を紹介している。Glassner^{7),8)}はV-Foldと呼ばれるタイプの、180度を開いたときに単純な形が立ち上がるポップアップカードをCGで表示する手法を提案している。しかし、いずれも90度型の折り紙建築を対話的に設計できる手法ではない。

そこで我々は、ボクセルモデルを用いて折り紙建築の形状を保持することで、モデルの構築と3次元表示、および展開図の作成が容易に行え、対話的に形状操作を行える、新しい設計手法を提案した(図1⁹⁾。

ボクセル表現を用いた手法では、簡単なインタフェースで折り紙建築を設計でき、かつアルゴリズムがシンプルであるため、システムの実装も容易であるという利点がある。しかし、ボクセルという立方体の集合で形を表現するため、背面および底面に平行または垂直でない稜線が作成できず、またそのシンプルなデータ構造から、切り起こしを持つ形状を表現できないという制限がある。そこで本研究では、より自由度の高い形状を設計できるように、平面多角形の集合による表現を用いた新しい手法を提案する。

なお、本稿で用いる折り紙建築に関する用語を図2のように定義する。図2に示すような形状は、ボクセルを用いた設計手法では作成できなかったが、本手法ではこのような形状も設計できるようにする。

2. 平面多角形の集合による折り紙建築の表現

本章では平面多角形の集合で折り紙建築を表現する際に、計算機内に形状を保持するための手法を述べる。

2.1 垂直面と水平面

本稿で対象とする折り紙建築は、文献9)で説明され

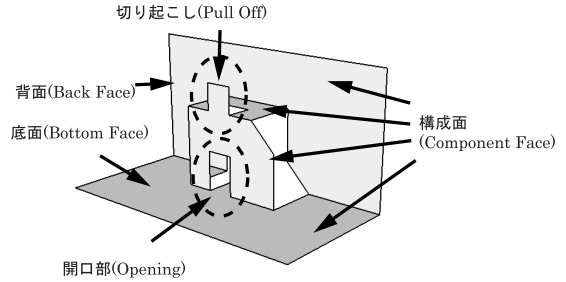


図 2 本稿で用いる用語の定義
Fig. 2 Definition of terms.

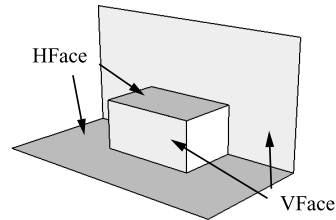


図 3 折り紙建築を構成する VFace と HFace
Fig. 3 VFace and HFace.

ているように、90度を開いた状態では底面に水平な構成面と底面に垂直な構成面の集合から成る。本稿では、それらの構成面をそれぞれ VFace (Vertical Face), HFace (Horizontal Face) として区別する(図3)。

2.2 カード座標(3D)と展開図座標(2D)の関係

対象とする折り紙建築は1枚の紙から作成されるため、180度を開いた状態(展開図)と90度を開いて立ち上げた形状には1対1の対応がある。つまり、1つの展開図から立ち上がる立体形状は一意に決まり、またその逆も成り立つ。それぞれの状態において、3次元の折り紙建築座標と2次元の展開図座標を図4のように底面とVFaceの距離を t_v 、背面とHFaceの距離を t_h と定めると、構成面をなす多角形の各頂点の座標値を次の関係式で相互に変換できる。

「折り紙建築座標(3D) 展開図座標(2D)」変換

$$\begin{cases} x_{(2D)} = x_{(3D)} \\ y_{(2D)} = z_{(3D)} - y_{(3D)} \end{cases} \quad (1)$$

「展開図座標(2D) 折り紙建築座標(3D)」変換
VFaceの場合:

$$\begin{cases} x_{(3D)} = x_{(2D)} \\ y_{(3D)} = t_v \\ z_{(3D)} = y_{(2D)} + t_v \end{cases} \quad (2)$$

HFaceの場合:

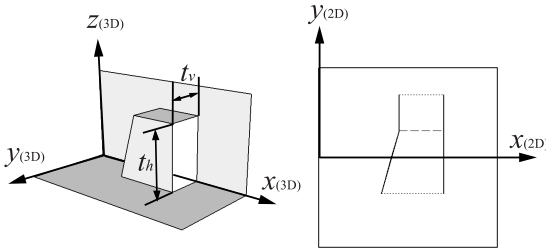


図 4 折り紙建築座標 (3D) と展開図座標 (2D)

Fig. 4 Model coordinate (3D) and pattern coordinate (2D).

$$\begin{cases} x_{(3D)} = x_{(2D)} \\ y_{(3D)} = -y_{(2D)} + t_h \\ z_{(3D)} = t_h \end{cases} \quad (3)$$

2.3 データ構造

計算機内に折り紙建築の形状を保持するには、保持されたデータから各構成面の輪郭をなす多角形の各頂点の座標値を復元できればよい。式 (1) ~ (3) の関係式から、各頂点の折り紙建築座標 (3 次元) での座標値と展開図座標 (2 次元) での座標値は相互に変換可能なので、どちらか一方の座標値だけを保持すればよい。次章で述べるが、本手法では展開図座標での各構成面の集合演算を行うため、各頂点の座標値は展開図座標で保持することとする。この場合、1 つの構成面を表現するのに、構成面の輪郭をなす頂点列の 2 次元座標値とともに、構成面が VFace と HFace のどちらであるかを示すフラグ、および式 (2), (3) 中の t の値 (VFace の場合は $t = t_v$, HFace の場合は $t = t_h$) が分かればよい。構成面を表すデータ構造 OAFace と、それらの集合によって表される折り紙建築のデータ構造 OrigamicArchitecture は図 5 のような擬似コードで定義できる。

2.4 展開図作成可能条件

本稿で対象とする折り紙建築は 1 枚の紙から作成されるため、各構成面の展開図上の多角形 (図 5 中の OAFace.polygon2D) は互いに重複せず、またそれらの和は工作用紙全体と一致する。これは、図 6 のように n 個の構成面 (底面と背面を含む) から成る折り紙建築の i 番目の構成面の、展開図上の平面多角形を F_i , 折り紙建築に用いる工作用紙全体を S とすると、式 (4) のように定義でき、本手法で扱う折り紙建築の形状はつねにこの式を満たす必要がある。以降、この条件を展開図作成可能条件と呼ぶこととする。

$$\begin{aligned} F_i \cap^* F_j &= \phi \quad (i \neq j) \\ F_1 \cup^* F_2 \cup^* \dots \cup^* F_n &= S \end{aligned} \quad (4)$$

```
enum faceType {VFACE, HFACE};

class Polygon2D {
    list<Point2D> points;
};

class OAFace {
    double value_of_t;
    faceType type_of_this_face;
    Polygon2D polygon_on_unfolded_pattern;
};

class OrigamicArchitecture {
    list<OAFace> faces;
};
```

図 5 データ構造の定義

Fig. 5 Definition of the data structure.

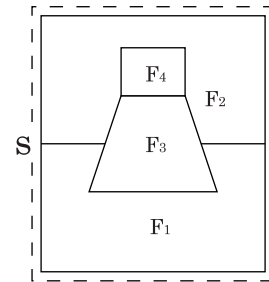


図 6 展開図を構成する平面多角形

Fig. 6 Polygons that construct OA pattern.

式中の \cap^* と \cup^* は、積と和を表す正規化集合演算子¹⁰⁾であり、 ϕ は空集合を現す。

2.5 立ち上げ可能条件

前節で述べた展開可能条件を満たすことで、1 枚の紙から作成できることが保証されるが、実際に工作して 90 度に開いたときに形が立ち上がらない場合がある。図 7 は折り紙建築として適切でないものの例である。

図 7(a) には宙に浮いた部分が存在する。(b) は下方と上方から立ち上がっている部分が分離しているため、台紙を開いたときに形が引き起こされないという問題がある。そこで、台紙を 90 度に開いたときに構成面が立ち上がる条件を立ち上げ可能条件と呼び、対象とする折り紙建築はこの条件を満たさなければならないものとする。この条件を満たすか否かを判定するアルゴリズムは 3.6 節で述べる。

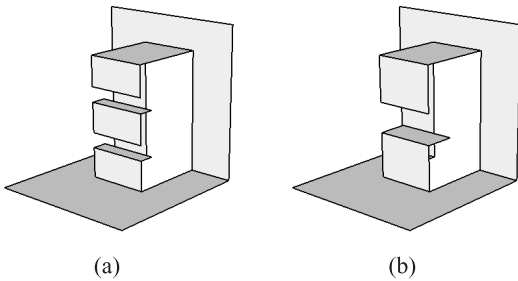


図 7 実現不可能な開口部を持つ場合
Fig. 7 Invalid cases.

3. 計算機による折り紙建築の設計支援

前章では折り紙建築の形状を計算機内に保持するための手法と、形状データが満たすべき条件を示した。本章では、条件を満たした形状をユーザが容易に作成できるように、計算機によって折り紙建築の設計を支援する手法を述べる。

3.1 ユーザインタフェース

従来の人の手による折り紙建築の設計は、展開図を直接編集することで行われているため、実際に組み立てを行うまで、90度を開いたときに立ち上がる形状を確認できず、試行錯誤に頼らざるをえなかった。そこで本稿では、折り紙建築の立体形状を計算機の画面で確認しながら、完成形状を対話的に編集することで、より直感的に設計を行う手法を提案する。

折り紙建築は水平または垂直な構成面から成るため、これらを3次元空間上の意図した場所に配置できれば折り紙建築の設計を行うことができる。本稿では、次のようなインタフェースで折り紙建築の設計を行う手法を提案する。

- (1) 底面と背面のみで構成される折り紙建築を初期状態とする(図8(a))。
- (2) 垂直(水平)な基準面の位置を前後(上下)に移動させ、新規に作成する構成面 VFace(HFace) の位置を確定する(図8(b)は垂直な場合を示す)。
- (3) (2)で決定した基準面上の格子点を選択し、構成面の形(多角形)を入力する(図8(c))。
- (4) 構成面を指定された場所に配置し、完成イメージを3次元表示する。
- (5) (2)~(4)を繰り返す。

3.2 新規構成面の追加によるデータ更新

前節のインタフェースで折り紙建築の編集が可能であるが、2.4節で述べた「展開図作成可能条件」を満たす形になることをユーザがつねに意識しながら編集

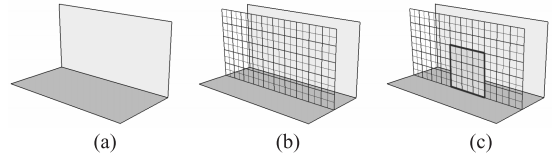


図 8 折り紙建築を設計するインタフェース
Fig. 8 Interface for OA design.

を進めるのは容易でない。そこで、ユーザが新しい構成面を追加したときに、展開図作成可能条件を満たすように、システムが自動で全体の形状を修正することとする。これにより、ユーザは折り紙建築の形状設計に専念できる。

具体的には、前節で述べたインタフェースのステップ(3)と(4)の間に、次の処理を行うこととする。

- (1) ユーザが新しく追加した3次元空間内の構成面について、式(1)を用いて展開図上での多角形を求め、新しく OAFace を生成する。
- (2) 新しく生成された OAFace と、すでに存在する展開図上の各構成面 $OAFace'(i)$ について、次の多角形の集合演算を行い、既存の構成面の更新を行う。 i は各構成面を識別するインデックスである。式中の演算子 $(-*)$ は、正則化集合演算の減算¹⁰⁾を意味する。

$$OAFace'(i).Polygon2D = OAFace'(i).Polygon2D - * OAFace.Polygon2D$$
(5)
- (3) OAFace とすべての $OAFace'(i)$ について、式(2)、(3)を用いて展開図座標から3次元座標を求め、3次元表示を行う。

新規構成面を追加する前の時点で、折り紙建築の形状が展開図作成可能条件を満たしていれば、上記の処理を行った後も、この条件は満たされる。設計の初期状態である、底面と背面のみを持つ折り紙建築は展開図作成可能条件を満たしているため、それ以降に行われる編集操作でも、つねに展開図作成可能条件は満たされることになる。

この手法では、新規に追加される構成面の形は変更されず、それ以前に作成された構成面の形が修正されるため、ユーザの直感的な操作を妨げない利点がある。

3.3 開口部の作成

前節で述べたデータ更新が自動で行われる場合「開口部を作成する操作」には、それと等価な「構成面を作成する操作」が存在する。たとえば図9(a)の四角で囲まれた領域に開口部を設け、(c)のような形状を作成する操作は、(b)の四角で示される水平な構成面を新規に作成する操作に置き換えることができる。

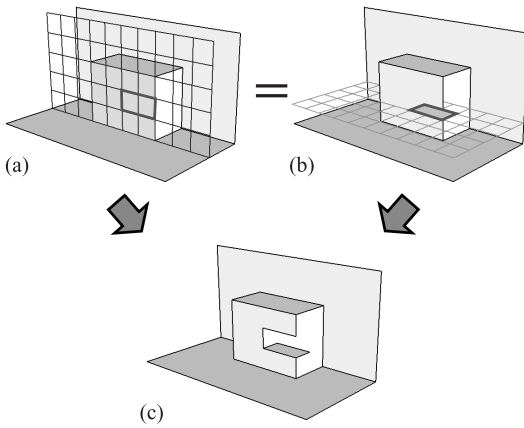


図 9 開口部の作成
Fig. 9 Creation of opening.

したがって、開口部を生成するためには、それと等価な構成面の生成を行えばよい。これにより、構成面の作成と開口部の作成を、システム内部では共通のアルゴリズムで行うことができる。

しかし、開口部を生成するために構成面の作成を行うのはユーザにとって直感的ではない。そこで、この処理はシステム側で行うこととする。つまりユーザが開口部とする多角形を入力したときには、システムがこれと等価な構成面の生成処理を内部的に実行する。

垂直面と水平面に開口部を生成する命令をそれぞれ CreateVHole, CreateHHole, 垂直および水平な構成面を生成する命令をそれぞれ CreateHFace, CreateVFace とし、各命令が入力として多角形の輪郭を表す点列 $P[]$ を受け取るとすると、この入力に用いる点列の各座標値を適切に変換することで、次のように開口部の生成命令を構成面の生成命令に変換できる。

$$\text{CreateVHole}(P_v[]) = \text{CreateHFace}(P'_v[])$$

$$\text{CreateHHole}(P_h[]) = \text{CreateVFace}(P'_h[])$$

ここで、入力された多角形の輪郭を表す点列 $P_v[]$ と $P_h[]$ の i 番目の要素の座標値を $(x_v[i], y_v[i], z_v[i]), (x_h[i], y_h[i], z_h[i])$ と表した場合、変換後の点列 $P'_v[]$ と $P'_h[]$ の各要素は次のように表現できる。

$$\begin{aligned} P'_v[i] &= (x_v[i], y_v[i] - z_v[i] + \min Z, \min Z) \\ P'_h[i] &= (x_h[i], \min Y, z_h[i] - y_h[i] + \min Y) \end{aligned} \quad (6)$$

なお、 $P_v[]$ の各点は底面に垂直な多角形面上にあるため $y_v[i]$ は定数となり、 $P_h[]$ の各点は底面に水平な多角形面上にあるため $z_h[i]$ は定数となる。また、 $\min Z$ は点列 $P_v[]$ の z 座標値の最小値、 $\min Y$ は点列 $P_h[]$ の y 座標値の最小値である。

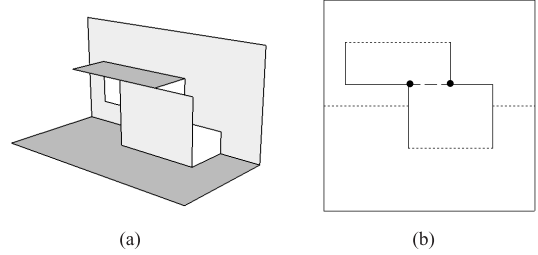


図 10 互いに一部を共有する線分
Fig. 10 Line segments sharing a part each other.

3.4 展開図の生成

本手法では、各構成面について展開図座標における位置情報を保持しているため、展開図の生成は各面の輪郭線を展開図座標に基づいて用紙に出力することで容易に行える。しかし、山折り線・谷折り線・切断線によって異なる線種を用いる場合、展開図の出力の前に輪郭線を構成するそれぞれの線分がどのタイプに属するのか判定する必要がある。

折り紙建築は展開図座標および折り紙建築座標における x 軸を中心とした回転で台紙の開閉が行われるため、 x 軸に平行でない線分は折れ線になることはなく、必ず切断線となる。線分が x 軸に平行である場合、その線種は切断線、谷折り線、山折り線のいずれにもなりうる。

この線種の判定は、その線分の一部または全部を共有して展開図上で隣接する構成面との、折り紙建築座標における位置関係から判定する。折り紙建築座標において、互いに線分を共有しない場合、この線分は切断線であり、凸に共有していれば山折り線、凹に共有していれば谷折り線である。図 10 (a) のように、隣接する構成面と一部のみを共有する線分の場合には、互いの線分を他方の線分の端点 (図 10 (b) 中の黒丸) で分割を行い、分割された線分それぞれについて上記の判定を行えばよい。

3.5 CG アニメーションによる開閉シミュレーション

文献 9) と同様に、各構成面の頂点の折り紙建築座標での座標値 (x, y, z) を、次式で変換することで、折り紙建築の開閉途中の形状を表現できる。

$$\begin{cases} X = x \\ Y = y - z \cos \theta \quad (0^\circ \leq \theta \leq 180^\circ) \\ Z = z \sin \theta \end{cases} \quad (7)$$

図 11 は 4 章であげる例題を、式 (7) の θ の値を徐々に変化させた CG アニメーションの例である。

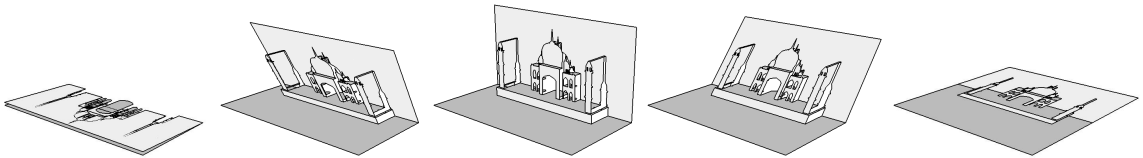


図 11 折り畳み途中の形状表示

Fig. 11 Folding and unfolding animation.

3.6 立ち上げ可能条件の判定

作成した折り紙建築モデルが、2.5 節で述べた立ち上げ可能条件を満たしているか否かを判定するアルゴリズムとして、展開図上の構成面にフラグ立てを行いながら巡回することで妥当性をチェックする手法を提案する。具体的には、展開図を作成後に、次のようなアルゴリズムによって立ち上げが可能であるかを判定できる。

- (1) 展開座標上の各構成面について、以下の方法で巡回を行い、巡回済みのフラグ立てを行う。
 - (a) 背面をなす構成面からスタートし、展開図上の山折り線、または谷折り線を介して隣接する構成面を再帰的に巡回する。ただし、構成面のタイプが VFace から HFace に移るときには谷折り線、HFace から VFace に移るときには山折り線を介していなくてはならない。
 - (b) 底面をなす多角形から開始し、展開図上の山折り線、または谷折り線を介して隣接する多角形を再帰的に巡回する。ただし、構成面のタイプが VFace から HFace に移るときには山折り線、HFace から VFace に移るときには谷折り線を介していなくてはならない。
- (2) すべての VFace について (1) の (a) と (b) のフラグのどちらか一方、または両方が立っていないものが存在したら立ち上げ可能条件を満たさず、折り紙建築として適切でない形状である。

上記のアルゴリズムを適用した例を図 12 に示す。展開図のうち、左側が (1) の巡回を行ったもの、右側が (2) の巡回を行ったものである。(a), (b) は、フラグの立っていない VFace (図中×印) が存在するため、適切でない形状であると判定される。

4. 結果

本手法を PC 上に実装し、実際に折り紙建築の設計を行った。その結果を図 13 に示す。(1) は座標軸に平行でない稜線と切り起こし、および開口部を含む例題であり、(2) はインドのタージマハルを意識して作成したものである。それぞれ、(a) は本手法で作成し

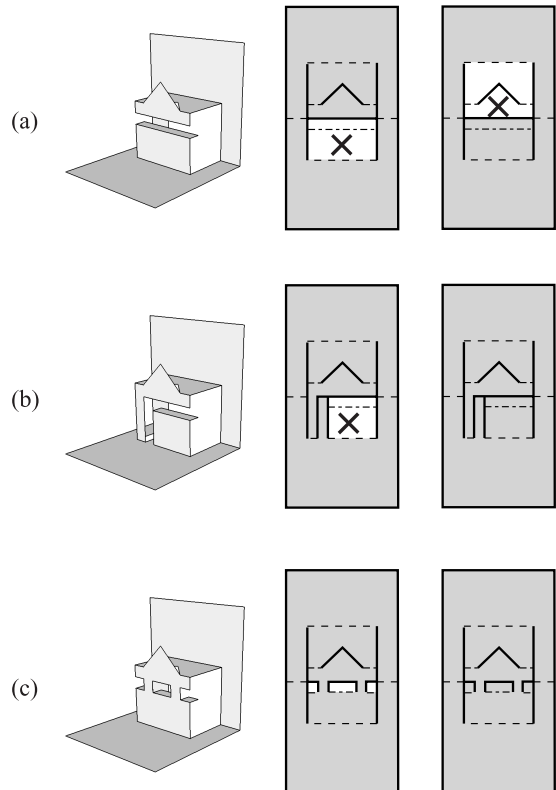


図 12 立ち上がり可能性の判定

Fig. 12 Judgment for the condition for Pop-Up.

た折り紙建築の CG 画像、(b) は展開図、(c) は実際に工作を行った写真である。モデルの制作は対話的に行うことができ、(1) の作品は 2 分程度、(2) の作品には 30 分程度の設計作業時間を要した。

また、作品に下絵画像を張り付けられるように拡張したものを、一般の大学生 4 名に使用してもらい、その評価を行った。あらかじめ下絵を台紙に張ることで、それを参考に形を作ることができる。今回は使用方法をしばらく練習した後、図 14 に示すような作品をそれぞれ数時間程度の作業で作ることができた。特に折り紙建築の知識がなくても、本手法を用いること

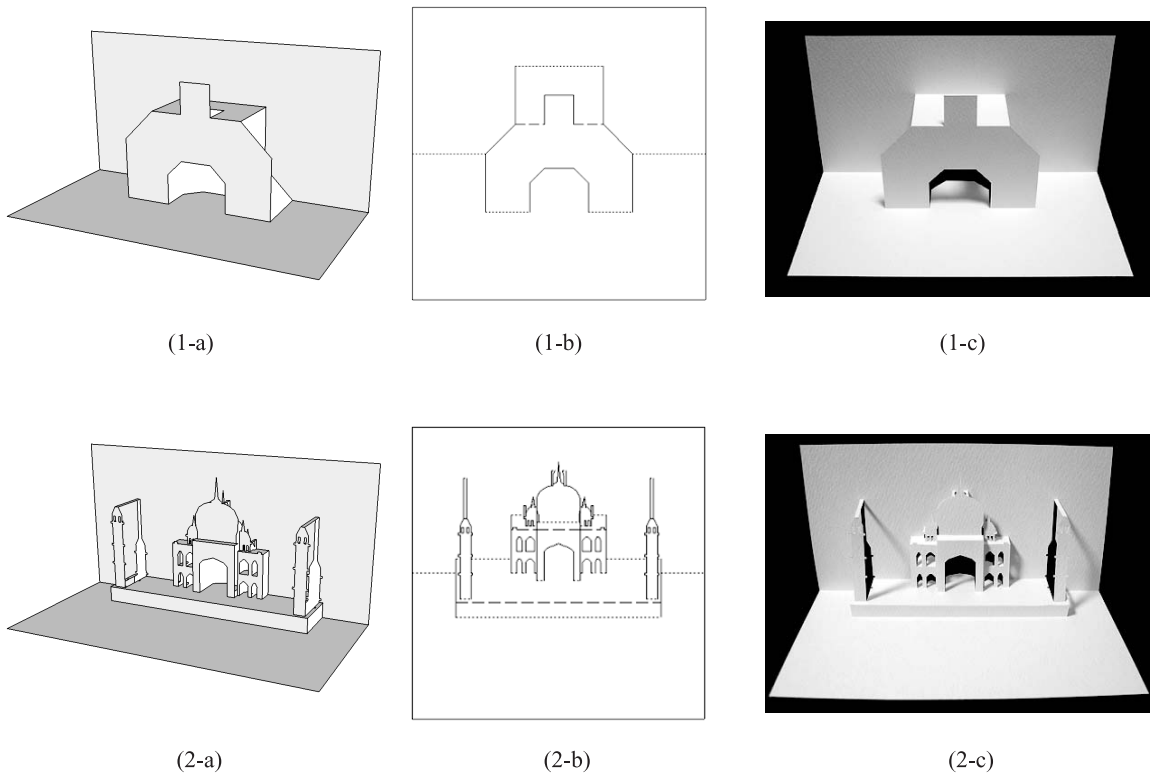


図 13 作品例 . (a) CG 画像, (b) 展開図, (c) 作成した折り紙建築の写真
Fig. 13 Examples. (a) CG image, (b) Unfolded pattern, (c) Photo.

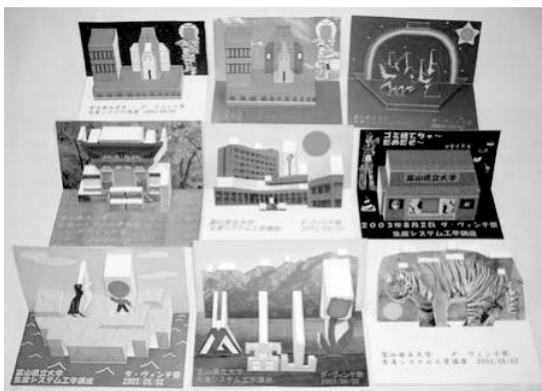


図 14 大学生の作品
Fig. 14 Practical use in a university.

で個人の趣向に応じた様々な形の折り紙建築を容易に作成できることを確認できた。

5. 今後の展望

本稿で提案した手法により、自由度の高い折り紙建築形状を対話的な操作で設計できるようになった。本手法では、折れ線によって入力できる多角形面の集合で形状を表現しているため、自由曲線を入力できない

という課題が残されている。制御点を入力することで曲線を生成し、その近似多角形を使用する手法も考えられるが、後から曲線の修正を行う場合を考慮するとデータの持たせ方を工夫する必要があるであろう。また、実際に工作するときのことを考えると、紙の物理特性によって、90度を開いたときの強度を考慮したり、紙の厚みによる折りたたみ時の歪みなどを考慮したりした設計支援の可能性が考えられる。

参考文献

- 1) 茶谷正洋：折り紙建築虎の巻，彰国社（1985）.
- 2) 中沢圭子：折り紙建築グリーティングカード集，彰国社（1994）.
- 3) 茶谷正洋，中沢圭子：折り紙建築京の旅，彰国社（1994）.
- 4) 茶谷正洋：折り紙建築型紙集，彰国社（1984）.
- 5) 茶谷正洋：折り紙建築型紙集-2，彰国社（1986）.
- 6) 茶谷正洋，中村俊文夫，安藤直見：実例パソコン折り紙建築と折り紙，講談社（1987）.
- 7) Glassner, A.: Andrew Glassner's Notebook, *IEEE Computer Graphics and Applications*, Vol.22, No.1, pp.79-86 (2002).
- 8) Glassner, A.: Andrew Glassner's Notebook, *IEEE Computer Graphics and Applications*,

Vol.22, No.2, pp.74-85 (2002).

- 9) 三谷 純, 鈴木宏正, 宇野 弘: 計算機によるボクセルを用いた「折り紙建築」モデルの設計手法, 情報処理学会誌, Vol.44, No.5, pp.1372-1373 (2003).
- 10) Requicha, A.: Representation of Rigid Solids: Theory, Methods, and Systems, *ACM Computing Surveys*, Vol.12, No.4, pp.437-464 (1980).
(平成 15 年 9 月 1 日受付)
(平成 15 年 12 月 2 日採録)



三谷 純 (学生会員)

1975 年生. 2000 年東京大学大学院工学系研究科情報工学専攻修士課程修了, 同博士課程進学. 2003 年東京大学大学院工学系研究科精密機械工学専攻転専攻, 現在に至る. コ

ンピュータグラフィックスおよび計算機を用いたペーパーラフトの設計支援に関する研究に従事. 精密工学会, 日本図学会各会員.



鈴木 宏正 (正会員)

1957 年生. 1986 年東京大学大学院工学系研究科博士課程修了・工学博士. 1987 年東京大学助手 (教養学部), 1988 年同講師, 1989 年同助教授. 1994 年同大学院工学系研究科精密機械工学専攻助教授, 2003 年同教授, 現在に至る. この間, 機械設計 CAD のための製品情報モデリングや形状モデリングに関する研究に従事. 2000 年より (株) エリジオン取締役を兼業. 精密工学会, ACM, IEEE 等の会員.