

## VRML を利用した折り紙ブラウザ

藤井 友里子 来住 伸子

津田塾大学情報数理科学科

### 概要

折り紙の折り方を本やビデオで伝えることは広く行われているが、分かりにくいなどの短所がある。そこで、Web 上で利用可能な、折り紙の折り方を分かり易く伝えるシステムを作成することにした。このシステムは、折り方のデータを管理するサーバ、折り紙ブラウザ、折り紙エディタなどから構成され、折る過程自体は VRML 2.0 ブラウザを利用した 3 次元アニメーションで表示する。本報告では、このシステムの折り紙ブラウザ、特に、折り方のデータ表現 および VRML 2.0 を利用した 3 次元アニメーションへの変換方法について紹介する。

## Origami Browser Using VRML on the Web

Yuriko Fujii Nobuko Kishi

Department of Mathematics and Computer Science, Tsuda College

### Abstract

Origami, paper folding, is hard to learn from diagrams and videos, because it is necessary to infer 3-D models from 2-D images. To overcome these drawbacks, we are developing a system for learning origami on the Web. The system contains an origami browser, an origami editor, and an origami archiver, which manages a collection of folding methods. It also uses a VRML 2.0 browser as 3-D animation output media. This article describes the origami browser in the system, focusing on how folding methods are stored, and how these methods are converted to 3-D animation in VRML 2.0.

## 1 はじめに

折り紙<sup>1</sup>は、誰もが遊んだことのあるであろう日本の伝統文化の一つである。折り紙は、その作品を見るだけでも楽しいが、一枚の紙から作品を折りあげていくのが魅力である。折り紙の折り方の多くは、本によって伝えられてきた。折り紙の本は、折り方を伝える方法としてポピュラーであるが、本で実際の折り方を記述するのには限界がある。折り図から折り方を理解するのは意外に難しいものである。

本研究では、折り紙の楽しさを広く、わかりやすく伝えたいということから、折り方をインターネット上で3次元グラフィックスで作成、鑑賞するシステムを構築することにした。3次元グラフィックスを使うことによって、本では表現しきれなかった部分まで折り紙を表現することができる。また、折る過程はアニメーションで記述しているので、折り図を見てどう折るのかを考える必要もない。

3次元グラフィックス表現方法としては、VRML2.0を用いている。VRML(Virtual Reality Modeling Language)[1]はインターネット上で3次元シーンおよびオブジェクトを記述するための言語で、パソコン上でも手軽に見ることができる。本論文では、折り紙の折り方をWeb上で観賞するブラウザ部分について述べる。これは、Web上のアプリケーションとして、多くの人が楽しめるということと、折り紙を3次元グラフィックスアニメーションによりわかりやすく紹介できるという特徴を持っている。

以下では、2章で関連研究とそれらの問題点について述べ、3章で、折り紙ブラウザについて述べる。この中で、折り紙、またVRMLファイルの具体例などについてもふれる。4章は今後の課題とまとめである。

## 2 関連研究と問題点

### 2.1 従来の折り方の紹介方法

#### ● 本(折り図)

本では、以下の手順で折り方を説明する。

1. 折る前の紙の状態を図示し、これから折るべき箇所を「折り線」として点線などで示す。

<sup>1</sup>折り紙という言葉は、1.)折るための紙、2.)折ること、3.)折ったあとの完成品の3つの意味を持つ。本論文では、2.)の意味で「折り紙」という言葉を使い、1.)は単に紙、3.)は折り紙の作品と表す。

2. 折る方向を矢印で示す、言葉で補足するなどして折り方を説明する。

3. 折った後の紙の状態を図示するとともに、次に折るべき箇所を示す。

利点：折り紙の本は手軽に手にはいる。

欠点：3次元の作品を2次元表示するので形が分かりにくい。折る途中の過程が記述されていないので折り方を理解しにくい。

#### ● ビデオ

折っている様子を一番わかりやすいと思われる視点から撮影し、言葉による説明などを加えながら折り方を紹介する。

利点：形が分かりやすい。折る途中の過程を動画で確認することができ、わかりやすい。

欠点：折り紙のある決まった視点からしか見ることができない。

### 2.2 インターネット上における折り紙

現在では、Web上に折り紙の完成作品の写真(JPEG, GIF形式など)をのせたり、折り紙とは何かということを紹介するようなページは多数ある[3][4]が、折り方を紹介しようとするものは少ない。そのようなものでも、上で述べた既存の方法を単にWeb上にのせただけ過ぎないものがほとんどである。具体的にはつぎのようなものがある。

#### ● 折り紙の本と同じ折り図、または写真

#### ● Java、GIF、MPEGなどによるアニメーション

#### ● OOG[2]による3次元グラフィックスアニメーション

これは、筆者がVRMLが標準化される前に作成したものである。OOG(Object Oriented Graphics Language)のフォーマットで折り紙をデータ化し、キーフレームアニメーションで折り方を紹介している。このデータを見るためには、専用のブラウザ(Geonview)が必要となる。

この方法の利点は、3次元グラフィックスで表現するので視点を自由に交えられることである。また、アニメーションで折り方を表示するのでわかりやすい。欠点は、データ量が多くなること、一つの作品をデータ化するまでに多くの時間と労力がかかってしまうことである。

● VRML1.0による3次元グラフィックス

これは、上のOGLのデータをVRML1.0のデータに変換したものである。VRML1.0を用いたことによって、特別なソフトウェアを追加しなくても、多くの人が見られるようになった。しかし、VRML1.0ではアニメーション機能を持たないため、折り方の紹介はユーザーのクリックによるコマ送りとなっている。

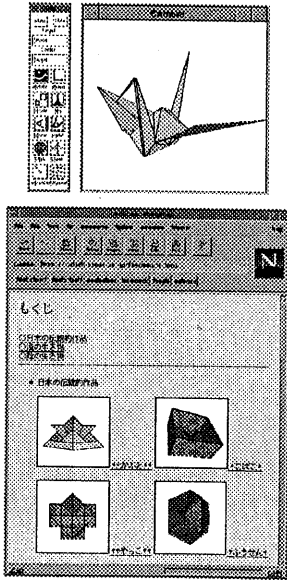


図1：OGLによる折り紙

### 3 折り紙システム

#### 3.1 システム概要

前章で述べた問題点を解決するために、VRML2.0を利用した折り紙システムを提案する。このシステムは、図2に示すように、折り紙を作成するためのEditorと、鑑賞するためのBrowserの二つの部分から成る。折り紙エディタは、折り紙をデータ化し、登録サーバによってそのデータをシステムに蓄積する。折り紙ブラウザは、システムに登録された作品のデータを取り出し、折り方をアニメーションとして見る手段を提供する。二つの部分は、折り方の情報を共有している。

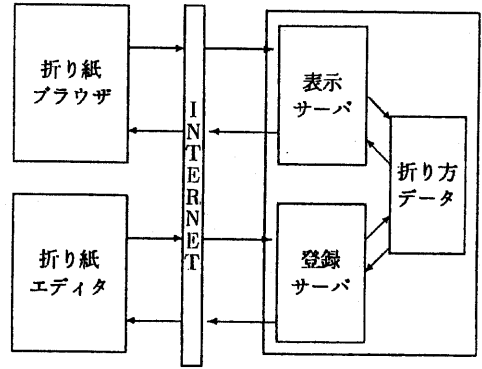


図2：システム概念図

#### 3.2 折り紙ブラウザ

折り紙の作品を見るには、まず、見たい作品をブラウザ上でクリックする。その情報を受け取ると、表示サーバは、折り方データから該当するデータを取り出し、コンバータによってVRML2.0に変換し、ブラウザに送る。ブラウザはVRMLファイルを受け取ると、VRMLブラウザを立ち上げ、折り紙のデータをロードする。

#### 3.3 折り紙とは

一口に折り紙といってもその種類はさまざまであるが、このシステムでは次のような折り紙は対象としない。

- ユニット折り紙  
触手を持った折り紙ユニットをいくつかつづけておいて、それらを互いに結び付けて組み立ててつくるもの。この場合、複数のものを組み合わせて初めて一つの作品となる。(例：座布団、手裏剣)
- 切り折り紙  
一枚の紙から作品を折っていくが、その過程、あるいは仕上げの段階でハサミ、のりなどを使う。

すなわち、本システムでは、糊やハサミは使わず、一枚の正方形の紙を折ることだけで仕上げられる作品を対象としている。また、そのなかでも、現在では平面のみで構成される作品に限っている。具体的には、「風船」などのように「ふくらます」、「ぼうし」のように「ひっぱって広げる」という工程を経て完成される作品は対象外としている。

### 3.4 折り紙の表現方法

折り方の情報とは、折れ線のない正方形の紙が完成品になるまでの情報である。このシステムは、折り方の情報を状態  $S_i$  と折り線  $L_i$  の組の並びで表現する。ここで状態  $S_i$  とは、平面の集合、折り線  $L_i$  とは、折り目にあたる線分の集合とする。

現在の紙の状態:  $S_i$  に折り線の情報:  $L_i$  を与えると、現在の状態に折り線を加えた状態:  $S'_i$  が作成される。この状態に、どちら側かを折るかを与え、次の状態:  $S_{i+1}$  を出力する。

#### 3.4.1 例

例として、下図のように、正方形の紙を向い合う頂点を結ぶ線を折り線として、三角形に折ることを考える。

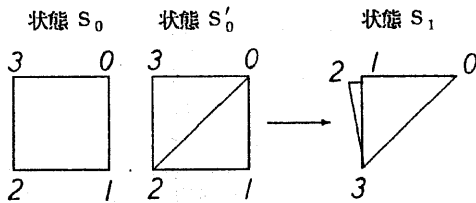


図3: 折り紙の一工程

“紙”は3次元空間内の平面で表現している。また、折り線は直線で表現している。折り線が一本入った一枚の紙は、直線が一本入った一枚の平面ではなく、折り線部分で連結された二枚の平面で表現している。従って、紙が折られていくと、グラフィックスの世界では、“紙”を構成する平面の数が増えていく。

すべての紙の状態は、平面の集合であるので、それを構成する頂点数、面の数、頂点の座標、各頂点をどの順に結んで面を構成するかで表現できる。図3の状態  $S_0$  は次のようになる。

#### 状態 $S_0$ の内部表現

4 1	#頂点数 面数
1 1 0	#頂点0の座標
1 -1 0	#頂点1
-1 -1 0	#頂点2
-1 1 0	#頂点3
4 0 1 2 3	#面の情報

一行目は、頂点数の数と面の数の並びである。二行目から五行目までは、各頂点の座標である。六行目の面の情

報は、最初の数字は、いくつの頂点を結ぶかを表し、次の数字からは、頂点番号をつなぐ順に記述したものである。

ユーザが指定した折り線は、折り線の両端の頂点番号として以下のように表現される。

#### 折り線情報 $L_0$ の内部表現

1	#折り線の数
0 2	#両端の頂点番号

折り線の両端が頂点でない場合(辺の真ん中など)は、折り線情報をつくる前に、新しい頂点として  $S_0$  に付け加えておく。紙の状態  $S_0$  と折り線情報  $L_0$  から、折り線の記述された、状態  $S'_0$  が作成される。

#### 状態 $S'_0$ の内部表現

4 2	
1 1 0	#頂点0
1 -1 0	#頂点1
-1 -1 0	#頂点2
-1 1 0	#頂点3
3 0 1 2	
3 2 3 0	

最後にどちらの面が回転するかの情報を与えると、面上の頂点に対して適切な回転変換が行われ、次の状態  $S_1$  が作成される。

#### 状態 $S_1$ の内部表現

4 2	
1 1 0	#頂点0
-1 1 0	#回転によって変換された頂点1
-1 -1 0	#頂点2
-1 1 0	#頂点3
3 0 1 2	
3 2 3 0	

### 3.5 VRML について

VRML は、インターネット上に3次元仮想空間を実現するための仕様である。現在の最新の仕様は VRML2.0 であり、これは、96年8月4日にリリースされた。それまでの VRML1.0 では、静的な3次元世界しか構築できなかったのに対して、VRML2.0 では動的な3次元仮想空間が表現可能になり、より複雑な世界を記述できるようになった。VRMLのフォーマットで記述された

オブジェクトを見るためには、専用のブラウザが必要である。

### 3.6 VRML2.0でのアニメーション

アニメーションの実現方法を二つあげる。

- Interpolator ノードによる補間

VRMLでは、ジオメトリの形状の他、付帯情報などもすべてノードと呼ばれるオブジェクトの集まりとして記述する。それぞれのノードの特性はフィールドと呼ばれるパラメータによって決定される。VRML2.0には動きを扱うノードとして Interpolator ノードがある。Interpolator ノードは、線形キーフレームアニメーションのために設計されている。Interpolator ノードに、「key」とそれと同じ数だけの「keyvalue」を与えると、それぞれの値に応じた補間が行われ、 $f(\text{key})=\text{keyvalue}$  となるような区分線形関数  $f(t)$  が定義される。VRML2.0では6種類の Interpolator ノードが用意されている。

このシステムでは、折り紙を折るということをし折り線 (=平面と平面の接線) まわりの平面の回転としてとらえている。その実現のためには、OrientationInterpolator (方向補間処理) ノードを用いている。

- キーフレーム

折り方が複雑になってくると、同じ平面上の頂点が違う方向へ回転することもあるので、上のような方法では表すことができないことが考えられる。この場合は、折る途中の一コマコマをそれぞれ一つの VRML ファイルで記述し、それを切替える方法をとる。VRML2.0では、ファイルを切替えることのできるノードとして、Switch ノードが用意されているが、これは、ユーザのクリックによって次のファイルをロードするので、アニメーションとはならない。そこで、自動的に次のファイルをロードするノードを拡張 VRML ノードとして記述する必要がある。

現在、このシステムでは前者の方法で実現している。

#### 3.6.1 例

先ほどの 3.5.1 節であげた、正方形の紙を三角形に折る例を VRML ファイルで実現した例を示す。実際は、

三角形の面を二つ記述し、片方の面を回転させる記述が必要であるが、紙面の関係から、回転する面の方だけを示してある。

この例では、ファイルがブラウザに読み込まれると同時に、紙が折られていく様子のアニメーションが始まり、ユーザがクリックするまで同じ動作を繰り返すようになっている。

まず、この例に含まれているノードの定義を示す。

- Transform

グループ化ノードの一つ。グループ化ノードは階層的な座標変換グラフを作成するために使用され、children フィールドに子ノードを持つ。

- Shape

ジオメトリを記述するときに必要なノード。実際の形状ノードはこのノードの geometry フィールドに記述する。

- IndexedFaceSet, IndexedLineSet

頂点の座標と頂点をどう結ぶかを指定して、図形を描くためのもの

- Coordinate

上記の二つのノードで使われる座標を指定するためのもの

- TimeSensor ノード

ジオメトリや位置に無関係な時計で、Interpolator ノードのような時間に基づくノードをストップ/スタートさせるために使われる。

- TouchSensor ノード

あるジオメトリに関して、ユーザのポインティングの動きを感知する。

- OrientationInterpolator ノード

あるイベントを受け取ると、補間関数の数値計算を実行する。計算の結果は value\_changed 変数 (eventOut) に保持される。

次に、VRML2.0の基本概念をあげる。

- イベント

eventIn 定義は、他のノードから送られてくるデータメッセージを受け取るために使われ、受け取ったノードの状態を変化させる。eventOut 定義は、ソースノード内で変化した状態を目的ノードへ送

出するために使用される。この例では、最後の三行の set\_fraction、set\_rotation、set\_stop-Time が eventIn、fraction\_changed、value\_changed が eventOut である。

#### ● ROUTE

イベントを発生するノードとイベントを受け取るノード間の接続のこと。この文法によって、同じ型のイベントをノード間でやり取りすることができる。この例では、最後の三行が ROUTE 記述である。

TimeSensor ノードは、変数 loop を TRUE に設定することによって、ブラウザがファイルを読み込むと同時に自分を活性化する。そして、アニメーションを始めるためのイベントを発生する。このイベントは、ROUTE によって OrientationInterpolator に受け渡され、OrientationInterpolator ノードは補関数の計算を実行し始める。その計算結果は、ROUTE によって Transform ノードに送られる。これにより、Transform 内のノードが回転変換され、その様子がアニメーションとして出力される。

```
#VRML V2.0 utf8
DEF TRA Transform{
  children[
    DEF TIM TimeSensor {
      loop TRUE
      cycleInterval 5
    }
    DEF TOU TouchSensor { }
    DEF ORI OrientationInterpolator{
      key [ 0, 0.5, 1 ]
      keyValue [1 1 0 0, 1 1 0 1.57, 1 1 0 3.14]
    }
    Shape{
      geometry IndexedFaceSet {
        coord Coordinate{
          point [
            1 1 0,
            1 -1 0,
            -1 -1 0,
            -1 1 0
          ]
        }
        coordIndex [
          0, 2, 3, -1,
          3, 2, 0, -1,
        ]
      }
    }
  ]
}
Shape{
```

```
geometry IndexedLineSet {
  coord Coordinate{
    point [
      1 1 0,
      1 -1 0,
      -1 -1 0,
      -1 1 0
    ]
  }
  coordIndex [
    2, 3, 0, -1,
  ]
}
}
}
ROUTE TIM.fraction_changed TO ORI.set_fraction
ROUTE ORI.value_changed TO TRA.set_rotation
ROUTE TOU.touchTime TO TIM.set_stopTime
```

#### 4 今後の課題とまとめ

従来の折り紙の折り方の説明の方法をあげ、その問題点を指摘した上で、改善策として3次元グラフィックスアニメーションによる紹介方法を提案した。現在、VRML2.0によって3次元データ化した折り方を、アニメーションとして出力するブラウザを評価中である。インターネットのインタラクティブ性を生かして、誰でもがWeb上で折り紙の作品をつくることできるためには、折り紙エディタが必要である。そのためには、ユーザの折り方の指定方法と実際の折り方をどう対応させるかが重要となる。また、最終的には、複数のユーザどうしが、それぞれの作品データを共有、交換することができるようなシステムを目指している。

#### 参考文献

- [1] "The Virtual Reality Modeling Language: Cover page", <http://websapce.sgi.com/moving-worlds/>
- [2] "Geonview Home Page", <http://www.geom.umn.edu/software/geonview/>
- [3] "Joseph Wu's Origami page", <http://www.datt.co.jp/Origami>
- [4] "Origami Tanteidan Home Page", <http://www.ask.or.jp/origami/t/index.html>