

# 帯における結び目構造の入力のための 抽象化モデルによる入力・表示インタフェース

熊谷 一生<sup>†</sup> 松田 浩一<sup>†</sup>

## An interface using abstract ribbon model for knot structure

KAZUKI KUMAGAI<sup>†</sup> KOICHI MATSUDA<sup>†</sup>

### 1. はじめに

近年、エンターテインメントの分野などを中心に三次元コンピュータグラフィックス(3DCG)を用いる機会が増えてきている。3DCG の利用の増加に伴い、様々な形状の三次元モデルをモデリングする必要性が増えてきているがモデリングが困難な形状も存在する。

モデリングが困難な形状のひとつとして結び目を持つ形状が上げられる。結び目を持つ形状は構造が入り組んでおり、自己遮蔽が多くなっている。そのため、既存のモデリングソフトを用いた場合には多くの視点変更を行いながらの作業を必要とし、高度な技術を要する。

これまで紐における構造の入力をスケッチベースで行い、結び目をモデリングする手法が提案されている[1]。この手法では紐の軌跡をなぞるようにして、紐の形状や前後関係をスケッチベースで入力している。また、同時に物理シミュレーションを行い、紐の結び目をモデリングしながらのインタラクティブな形状の編集を可能にしている。しかし、帯のように幅を持つ形状にはこの手法は適用できない。

本稿では、帯におけるモデリングを実現するために、構造のみ(帯の軌跡、前後関係、裏表)の情報を入力する構造情報入力のステップと、三次元情報を入力するための表現情報入力のステップの二つに分けた。特に本稿では構造情報入力に焦点を当て、二次元的な操作のみによって帯の構造入力を可能にする抽象化モデルの提案を行う。また、入力した構造の確認と表現情報入力の前段階として、入力した構造の三次元的な簡易表示方法について提案を行う。

### 2. 帯における結び目

#### 2.1 帯の向き

帯において結び目を作成する際の帯の向きについて述べる。帯の向きは大きく分けて次の二つに分類される(図1)。

- (1) 幅を持つ向き
- (2) 幅を持たない向き

このうち、帯の結び目を作成する場合において帯は主に幅を持つ向きが正面を向く。幅を持つ向きが正面を向く理由としては帯を折り返す時や交差するときに幅を持つ面同士を合わせるためである。

このことから、常に幅を持つ向きが常に正面を向くと、結び目の作成が容易になると考えた。また、常に正面を向けることによって帯の構造を二次元的に捉えることができる。

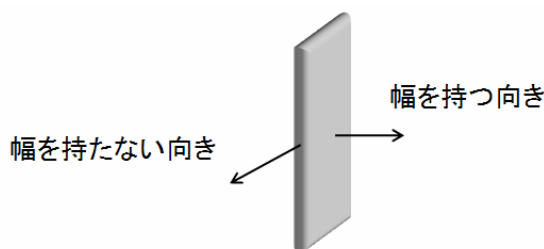


図1 帯の向き

#### 2.2 帯を結ぶ際に必要な要素

帯の結び目を作成する際に必要な要素についてネクタイの結び方(図2)を例として述べる。帯を結ぶ際に必要な要素は次の四つからなる。

- (1) 移動
- (2) 折り返し
- (3) 前後関係の指定
- (4) 縛る・緩める

<sup>†</sup> 岩手県立大学 ソフトウェア情報学部  
Iwate Prefectural University

以上の四つの要素を挙げた理由について述べる。

はじめに、図 2(a)から形状を編集するための前提として帯の移動が行える必要がある。帯を這わせて結び目を構築していくことはもちろんのこと、あとから帯の配置に関して微調整を行えることも必要である。

つぎに、図 2 (b)から帯を折り返す必要があることが分かる。折り返しに関しては山折と谷折が存在し、区別して折り返しを行わなければならない。

また、図 2 (c)から帯を他の帯の間に通す際に、他の帯の手前を通すか奥を通すかを一意に指定する必要があることが分かる。

最後に、図 2(c)から帯を通すために、帯を通す先で帯の間隔が十分に開いていないと帯を通すことができないことが分かる。よって帯を緩めて帯の間隔をあげる動作が必要になる。また、形状を整えるためにも緩める動作と逆の縛る動作も必要となる。

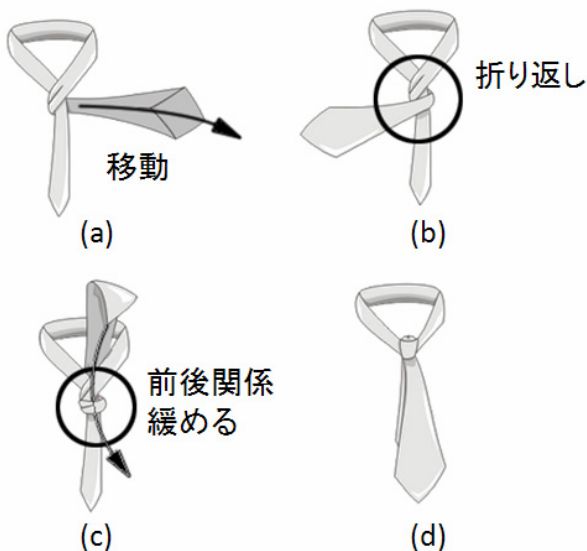


図2 ネクタイの結び方

### 3. 構造入力のための帯抽象化モデル

#### 3.1 帯抽象化モデル

帯の向きに関する制約と、帯を結ぶために必要な要素をもとに抽象化モデルを提案する。提案する抽象化モデルと実物との対応を図 3 に示す。抽象化モデルは折り返し点や端点に相当するピンと帯本体に相当するアームの二つからなる。帯における折り返し点や端点の位置は、帯の結び目を作成する上で重要な役割を担っている。よって、折り返し点や端点に相当するピンを移動可能にし、形状の編集を行えるようにした。また、帯本体に相当するアームはピンの移動に付随して

伸縮する。

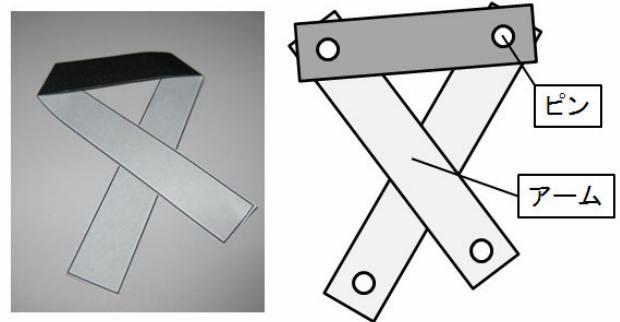


図3 実物 (左) と抽象化モデル (右) との対応

#### 3.2 帯抽象化モデルにおける必要動作の実現

帯を結ぶための動作として挙げた「移動」、「前後関係の指定」、「折り返し」、「縛る・緩める」という四つの動作を以下のように実現した。

##### 3.2.1 移動

移動の動作を実物の帯で行った場合、帯の長さに余剰や不足が発生する。余剰や不足を解消するために実物の帯では帯の端点に向けて余剰分を追い出したり、不足分を補ったりする。この動作は帯の構造全体に影響を及ぼす煩雑な作業となる。

よって、提案した抽象化モデルでは帯の長さの余剰や不足が発生しないように帯に相当するアームが伸縮する (図 4)。また、帯の移動は移動させたピンに直接に接続しているアームにしか行われない。このため、帯の移動によって他の部分も意図せずに変形されてしまうなどの副作用を及ぼさず、局所的な形状の編集が可能となる。

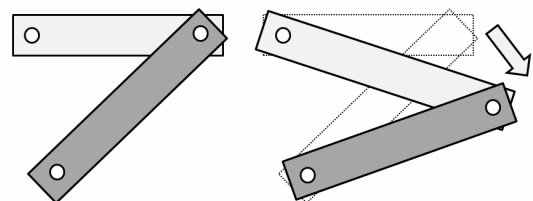


図4 帯の移動

##### 3.2.2 前後関係の指定

実際の帯で他の帯と交差させる場合、任意の奥行きを与えて帯を交差させる。しかし、奥行き座標値のインタラクティブな入力ディスプレイなどの二次元的な表示に限られる環境においてもっとも困難な課題の一

つである。

そこで、提案する抽象化モデルでは奥行き座標値を持たず、アーム毎の前後関係のみを入力することとした。入力方法としては帯と交差した場合にマウスの左ボタンが解放されているときは交差した帯の手前側を通し、マウスの左ボタンが押下されているときは交差した帯の奥を通すこととした（図5）。

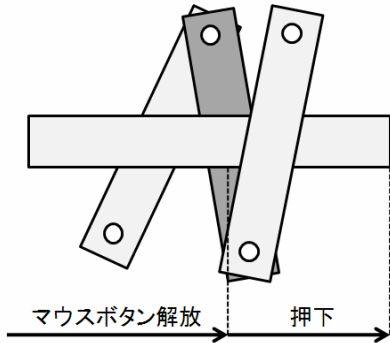


図5 前後入力方法

### 3.2.3 折り返し

帯を折り返す際は山折と谷折が存在し、同じ折り返しでも構造がまったく異なる。

折り返しの山折と谷折を区別できるように抽象化モデルではアームの重なり方に変化をつけることによって区別できるようにした（図6）。帯の結び目は巻きつける動作が主となる。よって螺旋を描くように折り返しの向きを自動で設定し巻きつけを発生させる。また、折り返しの向きを変更したい場合は前述した前後入力の指定と同じ動作によって変更する。折り返しを構成する二つのアームの前後関係を変えることによって山折と谷折を変更することができる。

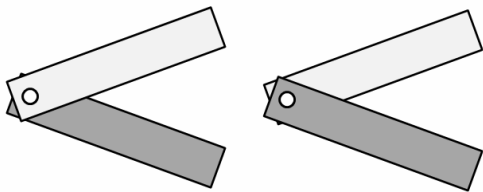


図6 折り返しの山折と谷折の表現

### 3.2.4 縛る・緩める

実際の帯を縛る場合は縛りたい場所の帯を引っ張り、また、緩める場合は緩めたい場所を指で押し広げて操作を行いたい範囲内の帯を同時に移動する。さらに、移動させたくない部分の帯に関しては指を用いて固定

しなければならない。このような同時に複数本の指を用いて行っている動作を、一般のコンピュータのような単一のポインティングデバイスしか存在しない環境で行うことは困難である。

抽象化モデルにおいては、選択範囲中の複数のアームの長さを同時に伸縮することによって縛る動作と緩める動作を疑似的に実現した（図7、図8）。縛る・緩める動作を行うには、初めに操作の対象にする範囲を選択する。そして、選択した範囲内のアームが一様に伸長、または収縮することによって縛る・緩める動作を実現する。

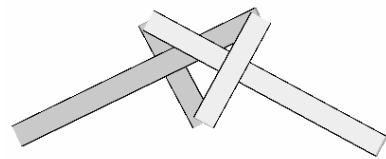


図7 縛る・緩める動作を行う前の構造

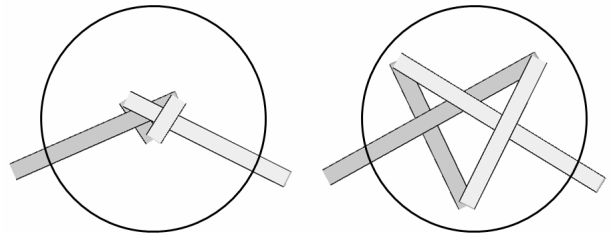


図8 円の範囲中を縛った場合（左）と緩めた場合（右）

## 4. 構造表現のための物理シミュレーション

抽象化モデルを用いた形状の入力画面においては構造の入力は容易にできるが、構造を把握するには直感的な表示ではない。抽象化モデルを用いて入力した形状には、前後関係についての情報は持つが奥行き座標値に関する情報がなく、抽象化モデルは入力に特化しているため現実的な形状になっていない。

そこで、構造のみを有した帯形状に、丸み、奥行きを持たせる力を働かせたうえで物理シミュレーションを行い、三次元形状を作成した。三次元形状の作成は三次元モデルの初期配置と物理シミュレーションの二つのステップからなる。

### 4.1 三次元モデルの初期配置

まず、入力した構造を基に、帯の三次元モデルを配置する（図9）。配置を行う際に前後関係を考慮し、手前を通る帯と奥を通る帯とで初期奥行き座標に微小な差を設ける。ここで差を微小にした理由は、手前と奥との奥行き座標に大きな差を設けると、三次元モデ

ルの突き抜けが発生するためである。また、帯の幅方向の突き抜けを防ぐために、三次元モデルの初期配置では帯が幅方向に縮んだ状態で配置する。

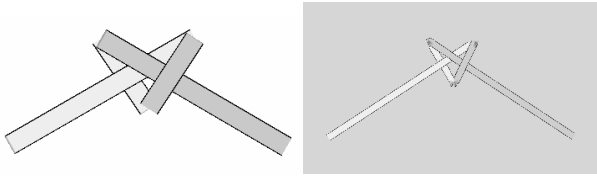


図9 入力した構造 (左) と初期配置 (右)

#### 4.2 物理シミュレーション

物理シミュレーションにはクロスシミュレーションにおいて広く用いられているマスバネモデルを用いた。形状に丸みを持たせるために強力なヒンジ復元力を加えた。ヒンジ復元力とは折れている帯を平らな状態に戻そうとする力である。ヒンジ復元力により帯の急な角度が是正される。物理シミュレーション結果を図10に示す。図10(a)の初期配置に対して図10(d)の三次元モデルが得られる。

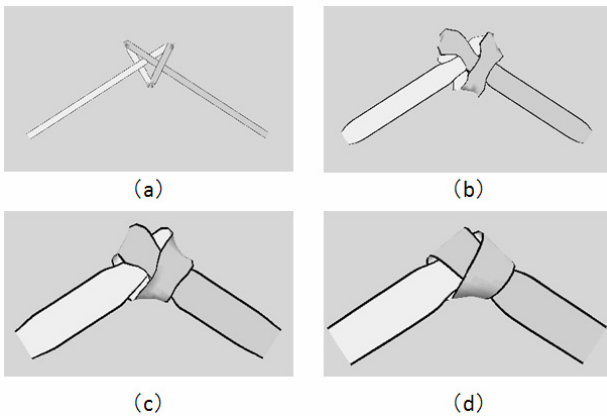


図10 物理シミュレーション過程

### 5. 結果

以上の提案手法によりネクタイのプレーンノット結びにおける、構造の入力と構造の三次元化を行った。入力した構造を図11に、入力した構造から得られた三次元モデルを図12に示す。

提案した抽象化モデルによりネクタイの結び目の構造を入力することができた。構造の入力は二次元的な操作しか必要としないため、多数の視点変更などの高度なモデリング技術がなくとも構造を入力し三次元化することが可能となった。

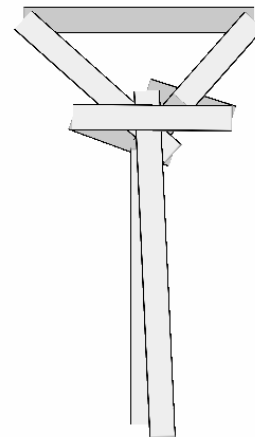


図11 入力したネクタイの構造

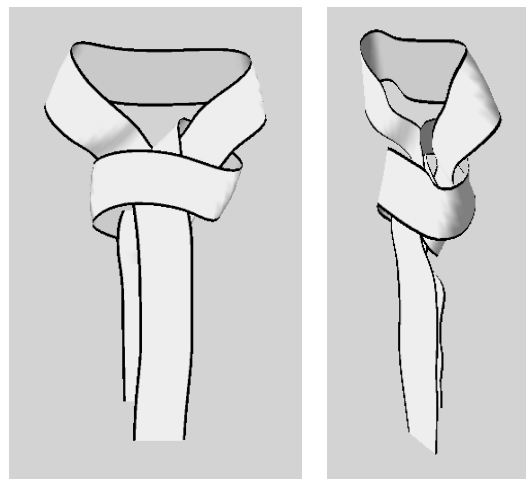


図12 ネクタイの構造の物理シミュレーション結果

### 6. おわりに

本稿では帯における結び目構造の入力のために、結び目を作る際に必要な動作や結び目の特徴について説明し、抽象化モデルを提案した。また、入力した結び目構造を人間が直観的に理解できるように、入力した構造の三次元的な簡易表示を行った。

今後の課題としては、表現情報の入力のために、三次元表示を行ったモデルを編集する機能を付加する必要があると考える。また、入力した構造の情報から帯の結び方を提示し、結び方を覚えるための学習支援に応用できると考える。

### 参考文献

- 1) Puripant Ruchikachorn: An interactive system for knot design, 東京大学大学院 情報理工学系研究科 修士論文, 2008