

Apparent Layer Operations を活用した結び目と水引の CG モデル構築

三谷 純 Jun MITANI

五十嵐 健夫 Takeo IGARASHI

概要:本稿では、結び目 (knot) と水引の 3DCG モデルを対話的に、かつ効率よく構築するためのシステムを提案する。提案システムでは、投影面上でのマウス操作で 3D 物体の位置関係を操作する一手法である Apparent Layer Operations を活用する。ユーザはマウスドラッグによって紐の形状を作成し、後からマウスクリックによって重なりのある上下関係を入れ替え、結び目の構造を対話的に変更できる。また、水引の形状を構築する際には、平滑化処理と左右対称化の処理を施し、滑らかで整った形状への補正を自動で行う。

キーワード:空間幾何学 / CG / 結び目 / 紐 / 水引

1. はじめに

結び目理論 (knot theory) の分野では、その幾何学的な形よりも、絡み具合などの位相的な情報に着目して結び目の性質を論じる。しかしながら、絡み具合を視覚的に捉えるためには、なんらかの方法で紐の幾何形状を構築し、それを可視化することが必要となる。結び目を図で示す場合には、2次元平面上に特徴が表されるように紐(曲線パス)を配置し、3本以上の紐が同一箇所では重ならないようにする。2本の曲線が重なる箇所(交点)では、その重なり関係を明示する必要があり、図1に示すような曲線による表現では、下を通る線の一部を描画しないことで、その上下関係を表すことが一般的である。結び目理論の教科書などには、図1のような図が用いられることが多く、鉛筆などで手軽に表現できる利点がある。一方で、図2に示すような太さと陰影を持った紐モデルのCG表現は紐の上下関係を直観的に理解でき、使用する場面によってはこちらの表現の方が好まれることがある。

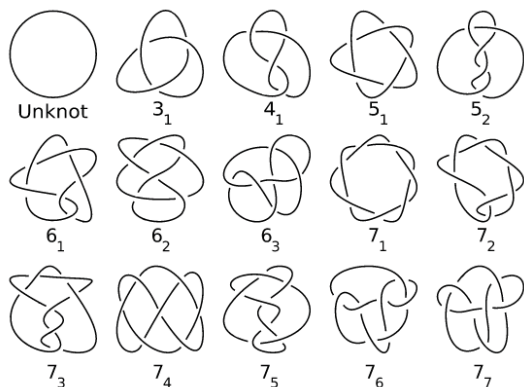


図1 曲線で表された交点数が7以下の結び目(出展 Wikipedia^[1]). 2つの線が交わる箇所では、一方の一部を非表示にすることで前後関係を表す。

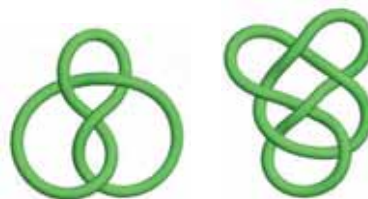


図2 図1の4₁と6₂の結び目をCGで表現した例(本稿で提案するシステムでデザインされた)。

贈答品や祝儀袋に付けられる水引には、「あわじ結び」に代表される古くから伝わる結び方が存在する。その一方で、近年では水引細工として新しい形の創作も活発に行われている。飾りとしての楽しさとともに、結び目理論に通じる数学的な形の面白さもある。図3は、水引で最も一般的な形である「あわじ結び」の例である。結び目と同様に、紐の形およびその重なり順が大事な要素となる。

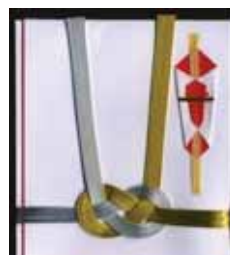


図3 水引(あわじ結び)の例(出展 Wikipedia^[2])。

近年では、汎用的な形状デザインを可能とする3DCGソフトウェアが普及し、様々な形状をCGで表現できるようになってきたが、このような紐や水引の形状をデザインすることは手間がかかる。特に、形が定まった後で、紐の重なり関係を変更することは容易でない。

Igarashiらは、このような問題を解決するために、布や紐などの柔物体の形状を、その重なり状態も含めて、マウス操作で簡単に編集できる手法、Apparent Layer Operations を提案した^[3]。本稿では、この手法を活用し、図2に示す結び目のデザインと、図3に示した水引のCGモデル構築のためのシステムを提案する。

提案システムでは、Igarashiらの手法と同様に、紐の内部モデルを紐の太さと等しい直径を持つ点の列で表し、点間に擬似的な斥力と反発力を働かせることで最終的な形状を決定する。ユーザはこの紐モデルに対して、後から重なりのある前後関係を入れ替えたり、一部をひっぱるなどして、結び目の形状と構造を対話的に変更できる。また、水引は左右対称な形状であることが多いため、入力として与えられた曲線を、対称形状にする修正が自動で行われる。

2. CG による紐の表現

CG で紐を扱う場合には、その幾何形状を定義するために、断面形状である円を曲線に沿って掃引することが一般的である。図 4 に示すように、紐の中心を 3 次元曲線で表現し、その曲線をパスとして円を掃引することで、一定の太さを持った紐の幾何形状を構築できる。



図 4 円を曲線に沿って掃引することで紐の立体形状を生成できる(3DCG ソフトウェア Shade12^[4]で生成)

ネクタイや靴の紐などに見られる「結び目」は日常生活の中で身近なものであり、釣りや登山、またはレスキューの場面では、その用途に応じた様々な結び方が用いられている。このような、実世界での結び目は紐の表面の相互作用によって最終的な形状が決定するため、これらを数式で表現することは難しい。とくに固く結ばれた複雑な構造を持つ結び目の幾何形状を、図 3 に示すような円の掃引によって構築する場合は、適切なパスをどのように定義するかが問題となる。掃引を行うためのパス間の距離が近すぎると、紐の形状の一部が他を突き抜けてしまい、遠すぎる場合には紐の表面が接しないため、すき間が生じて不自然な形となる。

KnotPlot^[5]という名称の、結び目の 3 次元モデルを生成するための専用ソフトウェアでは、紐の形状を擬似的な物理シミュレーションで変形させることで、より自然な形を生成することに成功している(図 5)。しかしながら、交点での紐の上下関係を、初期形状構築時に定義しておく必要があり、後から対話的に重なり関係を変更することはできない。



図 5 KnotPlot^[5]の UI.

Igarashi らが提案する Apparent Layer Operations のシステムでは、紐の形を主に次のような操作で編集できるようになっている。

- ・マウスカーソルの軌跡に沿った紐の形状の生成。

- ・マウสดラッグによる紐のひっぱり操作。
- ・マウスクリックによる紐の上下関係の入れ替え。

紐の上下関係の入れ替えは次の方法で実現している。つまり、視点方向から見た紐の輪郭を投影面上に描き、視点方向からみた時に重なり状態にある領域を検出する。その後、マウスクリックが行われた場所が重なり状態にある場合は、その場所での上下関係を入れかえ、隣接領域の奥行きが連続するように最終形状を整える(図 6)。Igarashi らの手法では、3 つ以上の領域が重なり合っている場合にも有効に機能するような工夫が施されているが、提案システムでは高々 2 つの領域の重なりのみが対象となる。

このような操作の組み合わせにより、実際の紐の結び目を模した幾何形状を容易に構築することが可能となる。本稿では、このシステムを用いて、結び目モデルの構築と水引モデルの構築を行う。



図 6 紐が重なる場所(丸で囲まれた箇所)をクリックすると重なり順が入れ替わる。

3. 結び目と水引モデルの構築

3.1. 環の表現

本稿で提案するシステムは、Igarashi らが実装したシステムをベースとしている。紐の操作方法はすでに実現されているが、そのままでは図 1,2 に示すような閉じた環が扱えない。そこで既存の紐モデルの始点と終点を連結し、環を扱えるように拡張した。擬似的な物理シミュレーションの際には、始点と終点は連結しているものとして扱われるため、端点の位置座標だけでなく、接線も連続した形状が得られる。

3.2. 対称化処理

水引の形状は、正面から見た時に左右対称であることが多い(ただし重なり関係は左右で異なる)。提案システムでは紐の形状をマウスカーソルの軌跡で入力するため、そのままでは左右対称な整った形を構築することが困難である。そのため、システムによる自動処理で形状を左右対称にする処理を施す。左右対称化の処理は、次の 3 つのステップから成る。

- (1) 対称な位置に配置される 2 点の対応付けを行う。
- (2) 対象軸の位置を決定する。
- (3) 紐を構成する頂点の位置を移動し、形を変形する。

今回は簡単な実装で実現するために、対象軸は入力された紐形状の外接四角形の中心を通り、y 軸(スクリーンの垂直方向)に平行な直線とした。また、対応関係

の構築の際は、紐が 1 本だけの場合は始点と終点を対応付け、その後、中央に向かって内部頂点を順番に対応づけるものとした。紐が 2 本の場合は、前処理としてそれぞれの紐を構成する頂点の数を統一しておき、端点から 1 つずつ対応付けることとした。一般的な水引には、3 本以上の紐から構成されるものが見あたらなかったため、今回の左右対称化の処理は 1 本または 2 本の紐で構成されるもののみとした。

対応関係と対称軸が定まった後の位置の更新については、Mitra らの対称化処理⁶⁾で用いられている、以下の方式を用いた。つまり、2 点 p, q が対応関係にあり、それぞれ d_p および d_q だけ移動させた点 p', q' が、鏡映変換 T (提案システムでは上述の通りの、 y 軸に平行な直線での反転) によって互いに位置を交換する関係であるとすると (図 7)、 $\|d_p\|^2 + \|d_q\|^2$ を最小とする d_p および d_q は次式で表される。

$$d_p = \frac{T^{-1}(q) - p}{2}, \quad d_q = \frac{T(p) - q}{2}$$

奥行き方向は無視し、投影面上での d_p および d_q の値を各点の移動方向と定める。この方向に働く力を、最終形状を決定する擬似的な物理シミュレーション処理時に、目的の位置へ移動する力成分に加えることとした。

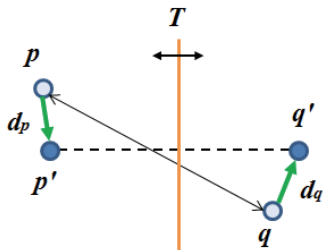


図 7 2 点 p, q の対称位置への移動

3.3. 平滑化処理の適用

水引に用いられる紐は硬く張りがあり、柔軟性に乏しい。この特徴を表すため、紐を構成する点列に対して、次式による平滑化処理 (ラブラシアンスムージング) の適用を行う。

$$P_i \leftarrow \frac{1}{4}P_{i-1} + \frac{1}{2}P_i + \frac{1}{4}P_{i+1}$$

上式では、点の座標 p_i は、自分自身の座標と、隣接する前後の点 p_{i-1} および p_{i+1} の重み付線形和で更新される。この平滑化処理を繰り返し適用することで、曲線の形は図 8 のように変形する。



図 8 平滑化処理の適用による形の変化

3.4. 基本形状の置き換え

結び目で扱う紐と異なり、水引には複数の細いこより紐が平面上に複数並んだ、横幅のある平坦な紐が使用される。そのため、中心線を掃引する断面形状を複数の小さい円が並んだものとする。たとえば断面の円を 5 つにすることで、図 9 に示すような形状を得る。

掃引の際には、断面が常に投影面と平行になるようにすることで、水引の紐が水平な状態で 3 次元モデルを構築できる。水引は平板に似た形と見なすことができ、3 次元モデル構築の際には奥行き方向の座標を圧縮して、すき間をなくすことを行う。

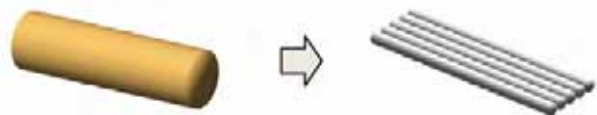


図 9 基本形状の置き換え

4. 結果

本稿で提案したシステムを用いてデザインした結び目を図 10 に示す。図 1 と同じ結び目を表現したものであるが、滑らかな紐の幾何形状の構築に成功しており、陰影のついたレンダリングにより、交点における紐の重なり関係を認識できる。

図 11 はマウスのドラッグによるストロークの軌跡で入力された初期形状 (図左) に対して、対称化および平滑化処理を施した例 (図右) である。対称化処理および平滑化処理を複数回交互に適用することで、左右対称で滑らかな曲線を得ることができた。

図 12 は、基本形状の置き換えによって、紐の形状から水引の形状へ変換してデザインした作例である。一部を拡大したものが図 13 であり、2 次元の図ではなく、3 次元の幾何構造を持っていることを確認できる。

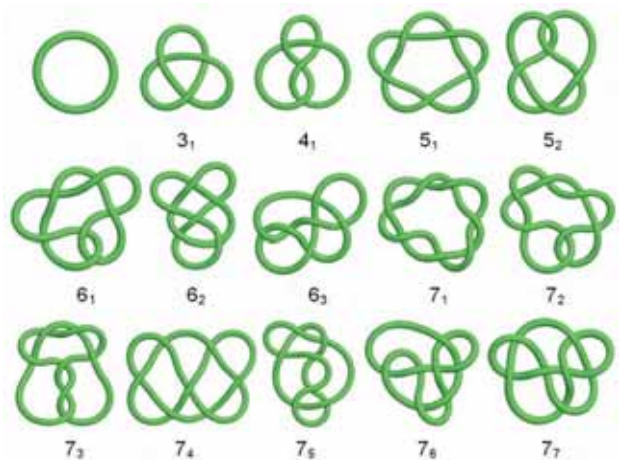


図 10 図 1 と同じ結び目を CG で表現した例。本稿で提案するシステムでデザインされた。

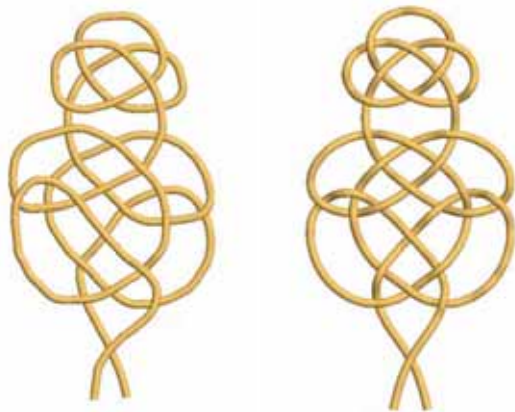


図 11 対称化, 平滑化および重なり関係の変更による形状修正. 修正前(左)と修正後(右)

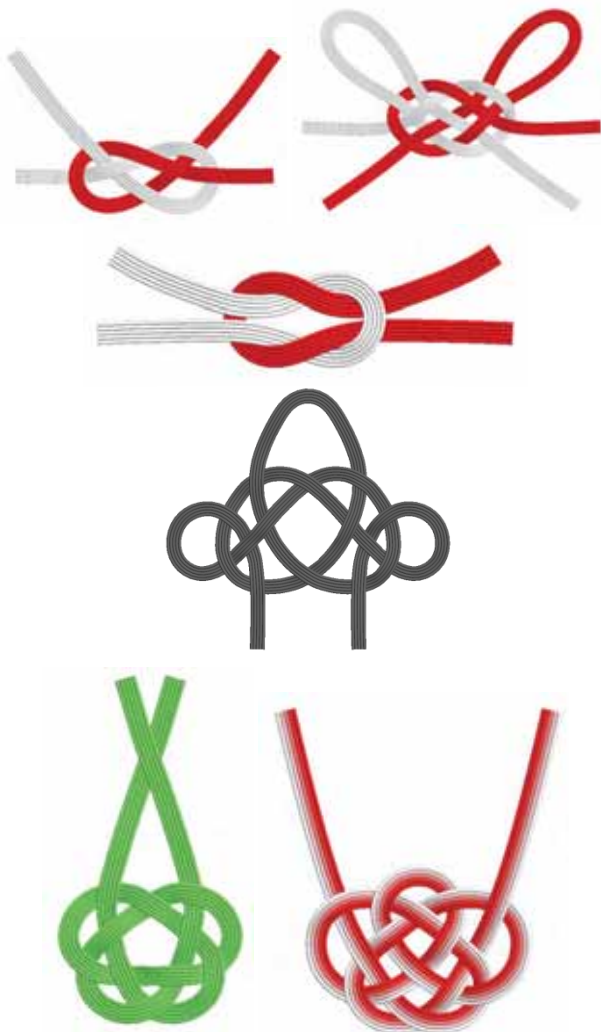


図 12. 提案システムで形状を構築した水引の例



図 13 水引モデルの拡大図

5. まとめ

Apparent Layer Operationsを活用し, 結び目と水引のCG表現のための形状を, 対話的にかつ効率的にデザインするためのシステムを提案した. 水引のモデルのデザイン時には, 左右対称性を考慮することで, 形の整った出力を得ることができた. 図 14 のように, 対象軸が y 軸に平行でない場合には, 対象軸を自動で決定するなどの改善の余地がある. 結びかたの手順の提示や, 難易度の推定, 対話的な着色機能などを付加することで, 創作水引を支援するためのツールとして使用できる可能性がある.

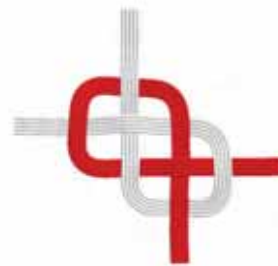


図 14 対称軸が投影面上の y 軸に平行でない例

参考文献

- [1] Wikipedia 結び目理論, <http://ja.wikipedia.org/wiki/結び目理論>, (2011/10/16 アクセス)
- [2] Wikipedia 水引, <http://ja.wikipedia.org/wiki/水引>, (2011/10/16 アクセス)
- [3] Takeo Igarashi, and Jun Mitani, Apparent Layer Operations for the Manipulation of Deformable Objects, ACM Transactions on Graphics, Volume 29, Issue 4, Article No.110, 2010.
- [4] 3DCG 作成ソフト | Shade | イーフロンティア, <http://shade.e-frontier.co.jp/>, (2011/10/16 アクセス)
- [5] Rob Scharein, KnotPlot, <http://www.knotplot.com/>, (2011/10/16 アクセス)
- [6] Niloy J. Mitra, Leonidas J. Guibas, Mark Pauly, Symmetrization, ACM Transactions on Graphics, Volume 26, Issue 3, Article No.63, 2007.

著者紹介

みにに じゅん : 筑波大学システム情報系/JST ERATO, 305-8573, 茨城県つくば市天王台 1-1-1

いがらし たけお : 東京大学情報理工学系研究科/JST ERATO, 113-0033, 東京都文京区本郷 7-3-1