

編物デザインを支援するための編目模様生成システムの作成

山田 雅之[†] 伊藤 裕一朗[†]
世木 博久[†] 伊藤 英則[†]

本論文では編物のデザイン表示を支援するためのシステムについて述べる。このシステムは編み手順を表す記号列から編物の出来上がりの模様を計算機を用いて生成する。このシステムは編物における紐の状態を平面図形として表現する。また、この模様生成の過程では、幾何学的移動と遺伝的アルゴリズムに基づく変形処理を適用する。本論文ではこれらの処理方式を述べ、いくつかの例を通してこのシステムで適用される方法の有効性、および生成される図形の妥当性を示す。

An Implementation of a Knit-Pattern Generating System for Supporting Knit Designing

MASASHI YAMADA,[†] YUICHIRO ITOH,[†] HIROHISA SEKI[†]
and HIDENORI ITOH[†]

In this paper, a system for supporting knit designing is described. This system generates a knit pattern from a set of knit symbols. In this system, a string state in a knit is represented as a diagram, and then, geometrical transformations for this diagram are applied. Furthermore, a genetic algorithm is applied to generate good quality knit patterns. The usefulness of these methods is verified by execution examples of this system.

1. はじめに

3次元形状を計算機上で表現し、これを処理することは、グラフィックスの表示や人工現実感など多くの分野の中心的課題の一つであり、また、これには様々な応用がある。例えば、3次元形状をデザインする際の支援ツールは、工業製品や建築物などのデザインには多く使われており、その有用性も認識されている。このようなツールには3次元形状を会話的に、もしくは、自動的に変形し、その処理結果を画面上に表示する機能がしばしば要求される。これらの機能を実現するための手法は、対象とする物体の種類により異なる。特に、3次元空間内で複雑な位相構造をもつ対象を計算機上に表現し、プログラムにより処理することは興味深い。

本論文ではこのような対象として、紐が複雑な位相構造を持つ編物を取り上げる。衣類に代表されるように生活との関係が深い編物は多くの国で伝わる工芸の一つであり、一本の紐を繰り返して変形することにより

様々な模様を作ることで知られている。我国においては編物にかんする変形表示方法は工業規格 (JIS) で定められている¹⁾。

編物では紐を定められた手順で繰り返して変形し、目的の模様を生成する。しかしながら、これらの手順と得られる模様との対応は実際に試行することにより分かるものであり、事前に知ることは困難である。このため編物の模様をデザインするには経験的知識や試行錯誤が必要である。実際は編み手である専門化がこれを行っているのが実状である。よって、編目のデザイン化を容易にするために、計算機でこれを支援することは有用である。しかしながら、これまでに計算機の応用分野として編物を取り上げた例は少なく、編物過程のシミュレーションなどにかんする報告もない。この理由として、計算機上での、1) 複雑に絡み合う紐の状態を表現する方法と、2) この表現の基での処理とその結果の表示の方法にこれまであまり効果的なものがなかったことが挙げられる。このなかで、紐状態を3次元空間の点列により表現する手法が報告されているが^{2),3)}、編物のような複雑な紐状態をこれにより表現することは困難である。また、これを処理する際にも効率に難がある。

[†] 名古屋工業大学工学部知能情報システム学科
Department of AI and Computer Science, Nagoya Institute of Technology

そこで本論文では、紐状態を平面図形として表現し、この図形を処理することにより、1) と 2) における問題を解決することを試みる。そして、この手法に基づき、編物のデザインツールを目指した編目模様生成システムを作成する。

編物の模様を生成するためには、計算機上での模様の定義と、模様を生成するための処理方法が必要である。これらについては模様を、図形における交点数および線分長の最小状態として定義する。また、一般の平面的グラフ図形では考慮する必要はなかったが、紐を表現する図形の処理や表示の際には本来紐が持つ絡みや結びなどの位相的性質を考慮する必要がある。ここで示す図形処理は幾何学移動と遺伝的アルゴリズム(GA)に基づくもので、紐の位相的性質を保存しつつ図形を変換し、模様を生成する。

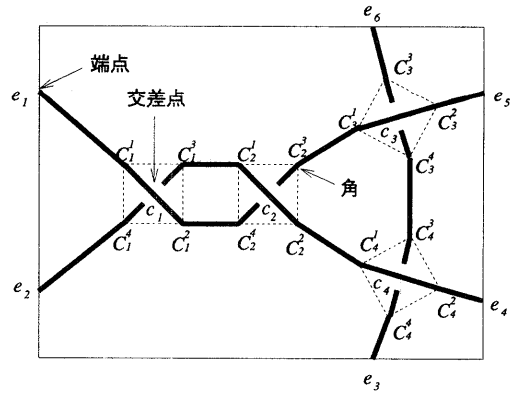
以下 2 章では、まず、紐状態を表現するための図形について述べ、さらに、この図形の計算機上での表現方法について述べる。次に 3 章では、編目模様生成システムの作成方法について述べる。このシステムでは上で述べた図形処理が適用される。4 章では、編物の代表的な例を用いて、ここで示す手法の妥当性とシステムを評価する。

2. 紐図形とその表現

ここでは紐の状態を図 1 (a) に示すような平面図形で表し、これを紐図形と呼ぶ。紐図形では、交差する紐の上下関係を陽に表すため、下に位置する紐を区切れた線分で表す。紐が交差する平面上の点を交差点と呼ぶ。また、紐図形の境界と紐が交わる点を端点と呼ぶ。

[紐図形の表現] 計算機上では紐図形を交差点、および端点の位置と位相に関するデータにより表現する(図 1 (b) 参照)。

交差点と端点の位置データは平面上の座標 (x, y) とする。紐図形において、交差点 c_i (i は交差点の番号) の近傍の四点を、ここでは四角形を用いて表し、それぞれの点を交差点 c_i の角(カド)と呼ぶ(図 1 (a) 参照)。四つの角 C_i^n ($n = 1, 2, 3, 4$) のうち、互いに対角にある角 C_i^1 と角 C_i^2 、角 C_i^3 と角 C_i^4 を結ぶ対角線はそれぞれ交差点の上方に位置する紐と下方に位置する紐に対応する。また各々の角 C_i^n ($n = 1, 2, 3, 4$) は i) 他の交差点の角 C_j^n ($i \neq j, n = 1, 2, 3, 4$) かまたは、ii) 自分自身の交差点の他の角 C_i^m ($n \neq m$)、iii) 端点 e_k (k は端点の番号)、のいずれか一つと連結している(紐によってつながっている)。ここでは交差点の位相データは角の連結先の点とする。また端



(a)

	点	番号	位置(座標)	位相(連結先)
紐図形	交差点	1	(10, 15)	e_1, C_2^3, C_2^1, e_2
		2	(20, 15)	$C_1^3, C_4^1, C_3^1, C_1^2$
		3	(33, 20)	C_2^3, e_5, e_6, C_4^3
		4	(33, 10)	C_2^2, e_4, C_4^3, e_3
紐図形	端点	1	(0, 22)	C_1^1
		2	(0, 8)	C_1^4
		3	(30, 0)	C_4^4
		4	(40, 8)	C_4^2
		5	(40, 22)	C_3^2
		6	(30, 30)	C_3^3

(b)

図 1 (a) 紐図形, (b) 紐図形のデータ構造

Fig. 1 (a) A string diagram, (b) A data structure of a string diagram.

点の位相データも連結先の点とする。例えば図 1 (a) において、1 番の交差点 c_1 の四つの角 $C_1^1, C_1^2, C_1^3, C_1^4$ の連結先はそれぞれ e_1, C_2^4, C_2^1, e_2 であり、端点 e_1 は角 C_1^1 である。

紐の状態を 3 次元空間内の点列により表現する方法があるが、これは各点の空間内の位置を厳密に指定する必要があるため複雑な編物の模様を表現するには不向きである。これに対し、ここでの表現方法は、交差点の平面内の位置と交差点どうしの連結関係を指定するのみでよいことより、複雑な編物の紐状態を比較的容易に表現できる。また編物の模様の表示においては、3 次元内の点列による表現に比べ細部の表現力は劣るが、紐図形は交差点と交差点間の線分のみで線図形であることより比較的簡単な処理で表示できる利点がある。

3. 編物デザイン処理

編物では、いくつかの基本的な編方が用意されており、これらを組み合わせることにより様々な模様を生

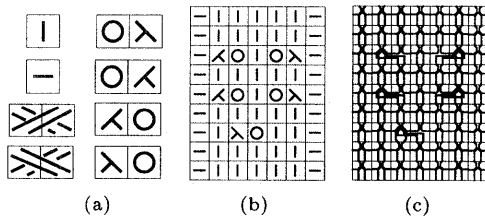


図2 (a) 編目記号, (b) 模様編図, (c) 初期図形
Fig. 2 (a) Stitch symbols, (b) A pattern knit diagram, (c) An initial diagram.

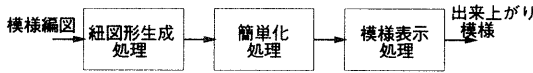


図3 編目模様生成システムの処理過程
Fig. 3 A process of a knit-pattern generating system.

成できる。これらの編方やこれらを適用する手順については日本工業規格 (JIS) で定められた表記法がある¹⁾。

編目記号 (図2(a)参照) は各編方を表記するための JIS 記号であり, 模様は編目記号を組み合わせてできる模様編図 (図2(b)参照) により表記できる。ただし, 編物は幾つかの種類に分類されており, 編目記号が定められているものには家庭用編機編, 棒針編, かぎ針編, アフガン編がある。本論文ではこのなかで最も一般的な家庭用編機編と棒針編を対象とする。また, 家庭用編機編と棒針編のために JIS では 24 種類の編目記号が定められており, これらの編目記号により表記される模様のみをここでは対象とする。

ユーザは編物の設計図である模様編図をエディタ上でデザインし, これをシステムに入力する。システムは入力の模様編図から紐図形を生成し (紐図形生成処理), 次にこの紐図形に対し単純化と呼ぶ変形を施す (単純化処理)。さらに単純化処理により得られた紐図形に対し, 後処理として紐の質感を出すための処理を施し出来上がり模様として表示する (模様表示処理)。このシステムの処理過程の概念を図3に示す。

3.1 紐図形生成処理

システムはまず模様編図から出来上がりの模様を生成するため, 模様編図を紐図形に変換する。これには編目記号と紐図形を対応づける編目データベース (図4参照) を用いる。システムは編目データベースを参照しながら各編目記号を対応する紐図形に置き換え, 模様編図から紐図形を生成する。以後, 紐図形生成処理で得られた紐図形を初期図形と呼ぶ (図2(c)参照)。

3.2 単純化処理

ここでは初期図形を変形することにより出来上がり

stitch symbol	stitch string diagram	stitch symbol	stitch string diagram
□		○↘	
—		○↙	
		○↗	

図4 編目データベース
Fig. 4 A stitch database.

の模様を生成する方法について述べる。なお, 編物の模様は紐のたわみが少ないことより, 出来上がりの模様は以下の条件 1)~3) を満たす紐図形とする。

- 1) 交差点数は最小である。
- 2) 紐の長さ総和は最小である (紐の長さ総和は以下で定義する)。
- 3) 初期図形と出来上がりの模様がそれぞれ表現する 3次元空間内の紐はその位相的性質が互いに等価である。

初期図形を 1)~3) を満たす図形に変形することを単純化と呼ぶ。著者らはすでに単純化のための手法を提案しアヤトリ図形生成に適用しその有効性を確認した^{4),5)}。この単純化手法では, 三つの図形変形, i) ライデマイスター移動, ii) 交差点移動および iii) 遺伝的アルゴリズムを適用する。

3.2.1 ライデマイスター移動

ライデマイスター移動 I, II, III⁶⁾ (図5参照) は位相幾何学の結び目理論と呼ばれる分野で提案された紐図形変形である。ここでは紐図形の交差点数を最小化するために, これらライデマイスター移動のうち交差点数を減少させるか, または, 増加させない変形を用いる。つまり, ライデマイスター移動 I, II は交差点数を減少させる変形 r_1 と r_2 のみを適用し, また, 減少も増加もしないことより III は両方向の変形 r_3, l_3 を適用する。

さらに, 交差点がある程度接近しているパターンの中にライデマイスター移動 III を適用した方が処理効率がよいことより⁵⁾, ここではライデマイスター移動 III に次のような制限を設ける。

[制限付きライデマイスター移動 III] ライデマイスター移動 III は図5に示すパターン P3 に適用できる。このパターンは三つの交差点 c_i, c_j, c_k からなり, ここではこれらの交差点の接近度 S を次式 (1) により定義する。

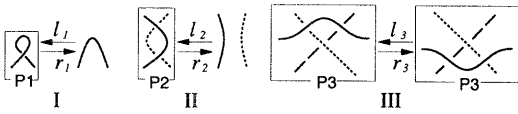
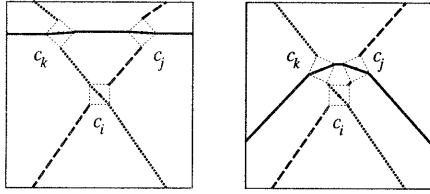


図5 ライデマイスター移動 I, II, III
Fig. 5 Reidemeister movements I, II, III.



(a) $S \geq a$ (b) $S < a$

図6 制限付きライデマイスター移動 III
Fig. 6 Restricted Reidemeister movement III.

$$S = d_{i,j} + d_{j,k} + d_{k,i}, \quad (1)$$

ここで、 $d_{p,q}$ は交差点 c_p と c_q ($p, q = i, j, k$) との距離を表す。さらにある閾値 TH を設定し、以下のようにライデマイスター移動 III を制限する。

ライデマイスター移動 III は条件： $S < TH$ を満たす P3 にのみ適用可能である。

すなわち、閾値が $TH = \infty$ であれば、すべての P3 に対しライデマイスター移動 III が適用可能であり、また、閾値が $TH = 0$ であれば、ライデマイスター移動 III は適用され得ない。ある定数 a を閾値とすることにより、図 6 (a) の P3 にはライデマイスター移動 III は適用できないが、図 6 (b) の P3 には適用できることになる。

ライデマイスター移動は交差点、および端点の位相の変形であるが、紐図形が表現する空間内の紐の位相的性質を変化させない⁶⁾。すなわち初期図形にライデマイスター移動を適用して得られる紐図形は出来上がりの模様であるための条件 3) を満足する。

3.2.2 長さ縮小処理

ここではまず紐の長さ総和を定義する。

[紐の長さ総和] 角 C_i^n とこれに連結する他の交差点の角または端点との距離を d_i^n とする。このとき交差点 c_i と連結する紐の長さ d_{c_i} を以下とする。

$$d_{c_i} = \sum_{n=1}^4 d_i^n. \quad (2)$$

また、端点 e_k とこれに連結する角または他の端点との距離を d_{e_k} とし、さらに図形中に存在する交差点と端点の個数をそれぞれ N_c, N_e とする。このとき、紐の長さの総和 L を以下とする。

$$L = \frac{1}{2} \left(\sum_{i=1}^{N_c} d_{c_i} + \sum_{k=1}^{N_e} d_{e_k} \right). \quad (3)$$

紐の長さ総和を最小化するため、簡単化処理では以下に定義する長さ縮小処理を紐図形に対し施す。

[長さ縮小処理] 以下に示す交差点移動を紐図形中の任意の交差点に繰り返し適用し、紐の長さ総和を減少させる。

[交差点移動] 交差点 c_i の四つの角 C_i^n ($n = 1 \sim 4$) が連結している 4 点を g_n とする (図 7 参照)。 d_{c_i} を減少させるため以下 (1), (2), (3) のように交差点 c_i を移動する。

- (1) 4 点 g_1, g_2, g_3, g_4 が凸四角形の頂点となるとき、この凸四角形の二つの対角線の交わる点へ c_i を移動し、さらに (3) を行う (図 7 (a) 参照)。
- (2) 4 点 g_1, g_2, g_3, g_4 が凹四角形の頂点となるとき、この凹四角形の凹頂点に c_i を移動し、さらに (3) を行う (図 7 (b) 参照)。
- (3) より d_{c_i} が小さくなるように交差点 c_i を回転させる。

d_{c_i} が減少するとき、明らかに紐の長さ総和 L も減少する。また交差点移動は紐図形が表現する空間内の紐の位相的性質を変化させないことから、得られる図形は上述の条件 3) を満足する。

3.2.3 GA に基づく簡単化処理

長さ縮小処理は紐の長さ総和 L を小さくするには有効な手段であるが、局所的な変形を用いた逐次処理であり、山登り方法的に L が最小である図形を探索することから一般的には L を十分最小化することができない。これを補うため、本論文では遺伝的アルゴリズム (GA)⁷⁾ を適用する。

端点の位置、および交差点と端点の位相が固定されているとき、紐の長さ総和 L は図形中の各交差点の位置により定まる。例えば図 8 (a) と (b) は端点の位置、および交差点と端点の位相は同一であるが、交差点の位置が互いに異なり、紐の長さ総和 L は (a) の方が小さい。この例のように、紐の長さ総和 L は互いに連結している交差点の位置関係、すなわち各交差点位置の組合せに依存するため、 L の最小化のためには L を最小とする交差点位置の組合せを求める必要がある。しかしながら、各交差点が平面上の任意の位置を取り、さらに交差点数が大きい紐図形の場合、交差点位置の組合せの数も非常に大きいため、 L を最小にするような各々の交差点の位置を見つけることは困難である。GA はこのような組合せの数の大きい最適化問

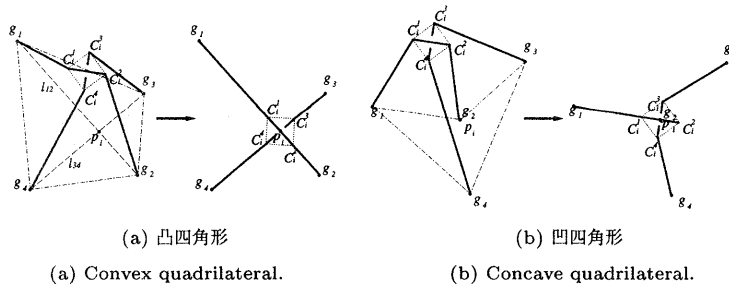


図7 交差点移動

Fig. 7 Crossing movements.

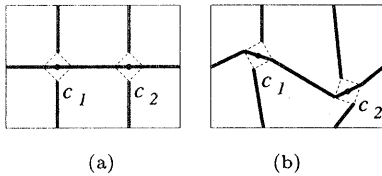


図8 交差点位置と紐の長さ総和

Fig. 8 Crossing positions and total string length.

題の近似解法として有効であり、また、確率的な処理候補探索アルゴリズムであることから局所解から抜け出すことができるという特徴を持つ。ここで示す単純化処理は紐の長さ総和最小化のために長さ縮小処理とGAを適用する。これによりGAが不得手とする局所探索を長さ縮小処理を用いて補い、さらに、GAの確率的な処理候補探索により局所解から抜け出す機能を持つことができる。

[単純化処理]

単純化処理は、まず、対象となる紐図形(初期図形)の各交差点位置を乱数値に置き換え、交差点位置の組合せを m 個作る。

- 1: $(x_1, y_1)^1, (x_2, y_2)^1, \dots, (x_{N_c}, y_{N_c})^1.$
- 2: $(x_1, y_1)^2, (x_2, y_2)^2, \dots, (x_{N_c}, y_{N_c})^2.$
- ...
- m : $(x_1, y_1)^m, (x_2, y_2)^m, \dots, (x_{N_c}, y_{N_c})^m.$

ここでは個々の組(1~ m)を個体と呼び、 $(x_i, y_i)^k$ は個体 k における交差点 i の座標を表す。以後、これらの個体は紐図形の交差点位置の候補となる。

次に単純化処理は交差点移動とライデマイスター移動からなる処理CRP(Crossing movements and Reidemeister movements Process)により(図9参照)、紐図形の紐の長さ総和と交差点数を最小化する。

CRPは、まず紐図形のパターンP1またはパターンP2である箇所に対しライデマイスター移動IまたはIIを施し、交差点を削除する。次に、紐の長さ総

```

begin
t:=0;
while t ≤ T do
while (紐図形にパターン P1 または P2 が存在する) do
ライデマイスター移動 I または II を
これら P1 または P2 に施す;
全個体に対し長さ縮小処理を施す;
if (紐図形にパターン P3 が存在する) then
最良個体の交差点位置に基づきパター
ン P3 の接近度 S を求める;
if (S < TH) then
ライデマイスター移動 III をこの P3 に施す;
t:=t+1;
end
    
```

図9 CRP
Fig. 9 CRP.

和 L を減少させるため、長さ縮小処理を用いて各々の個体の交差点位置を逐次変更する。これにより各々の個体は L をある程度小さくする交差点位置の組合せとなる。 m 個の個体に長さ縮小処理を施した後、今度は制限付きライデマイスター移動IIIを紐図形に施す。このとき、 m 個の個体の中で紐の長さ総和を最も小さくする個体(以降、最良個体と呼ぶ)の交差点位置に基づきパターンP3の接近度 S を求める。CRPはあらかじめ定めた期間 T の間、上記の処理を繰り返す。

さらに単純化処理では、紐の長さ総和をより小さくする交差点位置を求めるため、各個体の交差点位置をGAの遺伝的操作を用いて組み換え、新たに m 個の個体を生成する。ここでは以下に示す三つの遺伝的操作、選択、交叉、突然変異を順に繰り返し適用し m 個の個体を生成する。

選択: 式(4)により各個体の適応度 f を求める。

$$f(L) = A/L^2, \tag{4}$$

ここで L は紐の長さ総和とし、 A は定数とする。式(4)より紐の長さ総和を小さくする個体ほど適応度は高い。さらに、個体集団から一対の個体を選ぶ。このとき適応度の高い個体ほど高い確率で選択される。

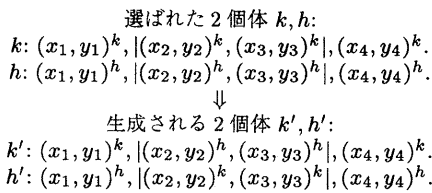


図10 交叉: $(x_i, y_i)^k$ と $(x_i, y_i)^h$ を入れ換える

Fig. 10 Crossover: $(x_i, y_i)^k$ and $(x_i, y_i)^h$ are exchanged.

交叉: 選択された一対の個体の交差点位置を組み換え、新しい個体を生成する。なお、ここでは交差点位置を組み換えるために二点交叉を用いる。この交叉は、まず個体上の二点をランダムに選び、これら二点間にある交差点位置を入れ換える (図10 参照)。

突然変異: 交叉により生成された個体のある交差点位置の値を一定の確率で乱数値に置き換える。

簡単化処理は上で述べた CRP と遺伝的操作を交互に繰り返し、初期図形を簡単化する。以下 step 1 から step 3 に簡単化処理をまとめる。

step 1 m 個の個体を生成する。各個体の交差点位置は乱数により与えられるものとする。世代数は0とする。

step 2 期間 T の間、CRP を施す。

step 3 世代数が N のときの紐図形を簡単化処理の出力とする。ここで紐図形の交差点位置は最良個体の交差点位置とする。世代数が N 未満のとき、遺伝的操作を用いて新たに m 個の個体を生成し、世代数を1増やし、step 2 へ行く。

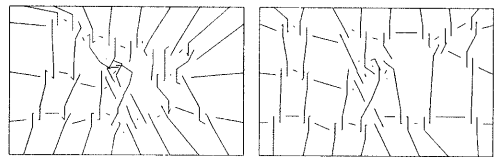
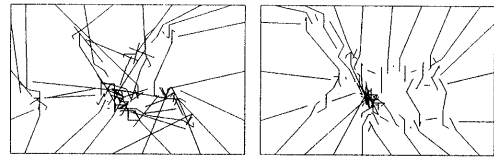
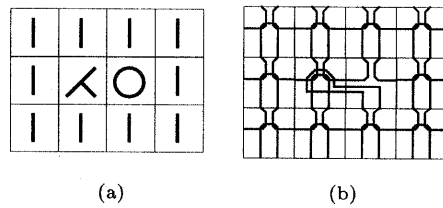
3.3 模様表示処理

編物における紐の質感を表現するため、システムは簡単化処理により得られた紐図形の線分に肉づけをし、それを表示する。

4. 実行例

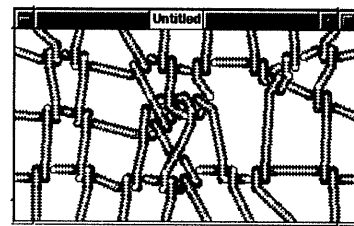
まず、システムの処理過程例を図11に示す。図11(a)に示す模様編図をシステムに入力した場合、システムはまず紐図形生成処理により図11(b)に示す初期図形を生成し、さらに、この初期図形に対して簡単化処理を施す。簡単化処理では、まず初期図形の交差点位置をランダムに設定し0世代目の個体を生成する。さらにライデマイスター移動、交差点移動、遺伝的操作を繰り返し施し紐図形を簡単化する。

図11(c)には簡単化処理過程における各世代の最良個体の交差点位置に基づく紐図形を示す。システムは簡単化処理の出力である $N (= 64)$ 世代目の紐図形に対し模様表示処理を施し、図11(d)の出来上がり模様



Generation 8 length = 830 Generation 64 length = 681

(c)



(d)

図11 システムの処理過程例 (a) 模様編図, (b) 初期図形, (c) 簡単化処理過程, (d) 出来上がり模様

Fig. 11 An execution process of the knit-pattern generating system (a) A pattern knit diagram, (b) An initial diagram, (c) A process of simplifying process, (d) A final pattern.

が得られる。

その他‘透かし編み’と‘なわ編み’についての処理結果をそれぞれ図12(a)と(b)に示す。

図11において、初期図形と出来上がりの模様の交差点数はともに50であり、簡単化処理による交差点数の減少はない例である。同様に図12の例も簡単化処理による交差点数の減少はない(図12(a)と(b)の模様の交差点数はそれぞれ262と248である)。なお、参考文献4)と5)でアヤトリの重ね合わせ図形とライデマイスター移動I, IIによる交差点数減少処理について述べたが、編物における初期図形も重ね合わせ図形の一つとも見なせる。しかしながら、編物では、紐

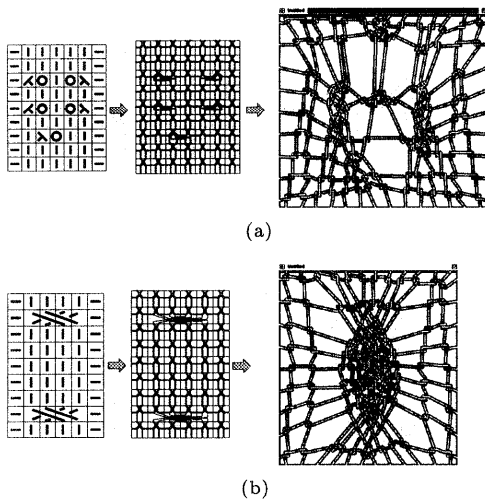


図 12 生成される模様例 (a) 透かし編み, (b) なわ編み
 Fig. 12 Generated patterns (a) Transparent knit,
 (b) Rope knit.

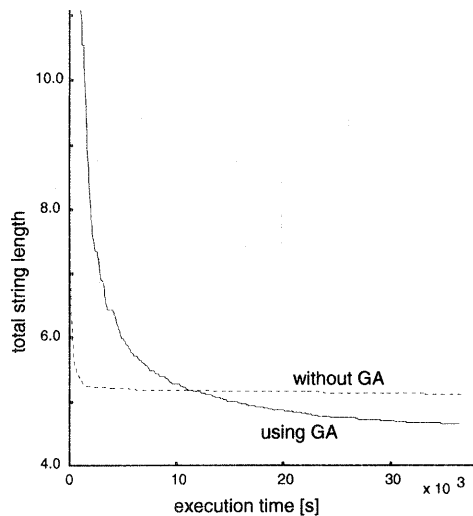


図 13 GA の効果
 Fig. 13 An effect of GA.

が必要以上にたわみを持っていることを表現することがない限り、交差点数はすでに最小化されているのが一般的である。

図 13 は GA の効果を示す。横軸は実行時間であり縦軸は紐の長さ総和である。一方の曲線 (using GA) は長さ縮小処理と遺伝的操作による長さの最小化過程の様子であり、もう一方の曲線 (without GA) は遺伝的操作を用いず長さ縮小処理のみによる長さ最小化過程の様子である。ここで、前者は個体数 m を 100 とし、後者は単一の個体に対する処理とした。前者は、

最小化の速度は遅いが、GA の広域的探索機能の効果により比較的長さ総和の小さい紐図形を見つけ出すことが図 13 より確認できる。

5. おわりに

本論文では、編物の紐状態を紐の交差する点に着目し紐図形として取り扱い、これにより編物の模様を表現し、さらに幾何学移動と GA を用いた処理により模様を表示する方法を示した。紐図形を用いて模様を表現することにより、その処理は交差点位置と交差点間の連結に対するいくつかの変換を繰り返すのみであり、3次元点列による表現^{2),3)}の場合と比べ、複雑なアルゴリズムが必要ないという特徴がある。また、GA などの最適化手法もこの表現のもとでは容易に適用できる利点がある。

これらの表現および処理方法に基づき編目模様生成システムを作成した。これは、編み手が模様をデザインする際、編目記号で表記できる模様編図を入力し、編み上がり時に得られる模様を確認できる編物デザインツールを目指すものであり、ここで定義した編目データベースを用いて模様編図から紐図形を生成し、次にこれを単純化し、後処理を施して出来上がりの模様を表示する。ここでは各処理方法について述べ、さらに、諸実行結果より本手法が妥当な模様を生成できること、および、GA の適用が本システムに有効であることを確認した。しかしながら、本手法は紐を四方から引っ張ったときにできる模様のみを対象としたため、紐にたるみがある模様は取り扱えない。また、与えられた初期図形の交差点数や紐の長さ総和が単純化処理により最小化されたか否かを、実行過程の任意の段階で判定する方法を確立することは困難であるため、単純化処理の有効性をここでは厳密には評価していない。さらに、単純化処理により得られた紐図形が計算幾何学的または位相幾何学的にどのような意味を持つのかについて明らかにしていない。そして、完成度の高い編物デザインツールとしての条件を満たすためには、ここではあまり触れなかった処理速度の改善、および、編物の熟練者による生成模様の評価が不可欠であり、これらを今後の検討課題とする必要がある。

以上、本論文では空間内で複雑な位相構造をもつ例として編物を取り上げ、これを表現、処理する方法について述べた。なお、ここで示した手法は、編物に限らず他の紐物体さらには空間グラフのようなより複雑な位相構造を持つ図形の変形処理にも応用でき得る。

参 考 文 献

- 1) 日本規格協会：編目記号，日本工業規格 JIS L 0201-1978 (1978).
- 2) Toriwaki, J., Yokoi, S. and Saito, T.: Understanding Forms by Man and Computer Using Computer Graphics and Image Processing, *Proc. 2nd Int. Symposium for Science on Form* (Ishizaka, S. (ed.)), pp.219-231 (1990).
- 3) 齊藤豊文，横井茂樹，鳥脇純一郎：3次元デジタル線図形のトポロジー—結び目の解析，電子情報通信学会技術研究報告，PRU 90-83 (1990).
- 4) 山田雅之，Budiart, R., 伊藤英則，世木博久：アヤトリにおけるひも図形変形過程の表現とその処理，情報処理学会論文誌，Vol.35, No.3, pp.497-504 (1994).
- 5) 山田雅之，杉山 貴，世木博久，伊藤英則：遺伝的アルゴリズムに基づく紐図形処理—アヤトリ図形生成方法—，情報処理学会論文誌，Vol.35, No.10, pp.2059-2068 (1994).
- 6) Kauffman, L. H.: On Knots, *Annals of Mathematics Studies*, Vol.155, Princeton Univ. Press (1987).
- 7) Goldberg, D. E.: *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley (1989).

(平成 6 年 6 月 20 日受付)

(平成 7 年 9 月 6 日採録)



山田 雅之 (正会員)

1992 年名古屋工業大学工学部電気情報工学科卒業。1994 年同大学院工学研究科博士前期課程修了。同年同大学工学部知能情報システム学科助手。人工知能学会会員。



伊藤裕一朗 (学生会員)

1995 年名古屋工業大学工学部知能情報システム学科卒業。現在，名古屋工業大学大学院工学研究科博士前期課程に在学中。3次元物体の表現，処理に関する研究に従事。



世木 博久 (正会員)

1979 年東京大学工学部計数工学科卒業。1981 年同大学院工学系研究科修士課程修了。同年 4 月より三菱電機(株)中央研究所に勤務。1985 年～1989 年(財)新世代コンピュータ技術開発機構に出向。1992 年 4 月より名古屋工業大学工学部知能情報システム学科助教授。工学博士。論理プログラミング，演繹データベース等に興味を持つ。電子情報通信学会，人工知能学会，ACM，IEEE Computer Society 各会員。



伊藤 英則 (正会員)

1974 年名古屋大学大学院工学研究科博士課程電気・電子専攻満了。工学博士号取得。同年日本電信電話公社入社，横須賀研究所勤務。1985 年(財)新世代コンピュータ技術開発機構出向。1989 年より名古屋工業大学教授，現在知能情報システム学科所属。これまでに，数理言語理論とオートマトン，計算機ネットワーク通信 OS，知識ベースシステムなどの研究と開発に従事。電子情報通信学会，人工知能学会，ファジィ学会各会員。