

3次元紐図形表現方法を用いた編物パターン処理について

伊藤 裕一朗[†] 山田 雅之[†] 宮崎 剛^{†,☆}
世木 博久[†] 伊藤 英則[†]

本論文では、編物の紐状態を3次元紐図形としてとらえ、これを表現し、紐図形の模様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できあがり模様の3次元イメージを計算機を用いて生成し、これを表示する。この紐図形生成の過程では、ライデマイスター移動処理と力学的移動処理に基づく変形処理を逐次的に紐の交差点に対して施す。また、複数の交差点移動を効率的に行うために適応型移動係数を提案し、その効率を評価した。さらに、いくつかの例を通してこのシステムで提案した方法の有用性、および生成される図形の妥当性を示す。

Processing for Knitting Patterns Using a Representation Method for 3D String Diagrams

YUICHIRO ITOH,[†] MASASHI YAMADA,[†] TSUYOSHI MIYAZAKI,^{†,☆}
HIROHISA SEKI[†] and HIDENORI ITOH[†]

In this paper, string states of knitting patterns are represented by three-dimensional string diagrams, and a system for supporting knitting pattern displaying is described. This system generates and displays a three-dimensional knitting pattern from a table of notations which represents a knitting process. To generate a string diagram, this system repeatedly performs a transformation processing based on both Reidemeister movements and a string-tension movement, to crossing points of the string diagram. Also, an adaptive move coefficient is presented to execute the string-tension movement efficiently, and its effect is evaluated. The usefulness of the proposed methods and the validity of generated diagrams are verified by some examples.

1. はじめに

本論文では、3次元空間内に存在し複雑に絡み合った紐状物体の処理効率を考慮した表現方法を提案し、これを編物表示処理に適用する方法について述べる。

紐状物体を計算機上で表現するには、i) ビット表現と、ii) 記号表現がある。前者は紐状物体の表現能力の点では優れているが、処理効率の点では後者に劣る。後者の記号による表現方法には、ii-a) 結び目理論で用いられる多項式表現方法¹⁾、ii-b) 紐の連結状態をセルオートマトンを用いて表現する方法^{2)~4)}、および、ii-c) 紐の交差点と交差点の連結状態を記号化した表現方法^{5)~10)}がある。ここで、方法 ii-a) は紐どうしが互いに交差する交差点ごとの絡み具合から紐状物体のトポロジカルな普遍的性質を描出する表現方法であり、紐状態の普遍的性質に関与しない部分の差異は表

現しない。方法 ii-b) は、方法 i) と同様に紐の連結状態をセルの連結状態としてとらえているので紐状物体の位置情報を忠実に表現できるが、処理効率の点で難がある。

これらを背景として、山田らは方法 ii-c) を提案した。すなわち、交差点に着目して交差点の紐の上下関係とその交差点の2次元座標および他の交差点との連結状態を記号化し、この処理方法を示した。しかし、これまでの方法では紐図形を3次元的に表現することはできなかった。

本論文では紐の太さを考慮に入れ、ii-c) の表現方法を3次元表現に拡張した表現方法と紐にかかる張力のベクトル和に基づく3次元編目模様生成システムの処理方法について述べる。ここで提案するシステムは日本工業規格(JIS)による編目記号¹¹⁾からの編み上がり模様を3次元表示し、厚み、面積、および、その模様を作成するのに必要とされる紐の長さの総和を試算することを可能とする。

編物は、いくつかの基本的な紐の絡みによる編目が単位化され記号化されており、これらを組み合わせる

[†] 名古屋工業大学

Nagoya Institute of Technology

[☆] 現在、日本電気(株)

Presently with NEC Corporation

ことにより様々な模様を生成できる。ただし、編目記号が定められているものには家庭用編み機編み、棒針編み、かぎ針編み、アフガン編みがあるが、本論文では、このうちで一般的な家庭用編み機編みと棒針編みを対象とする。また、家庭用編み機編みと棒針編みのために 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。

2. 交差点と連結点に着目した紐図形の表現方法

3次元空間内の紐状物体を図1に示す。これを紐図形と呼ぶ。ここで、紐は半径 r を持つ円筒形とし、紐図形内の交差点 C とはある方向 \ast から紐図形を見下ろしたときに紐が交差する点とする(図2参照)。交差点は図3の(a), (b)の2つの型に分けられる。(a)は上の紐と下の紐が接して絡まる交差点でありこれをA型交差点、(b)は上の紐と下の紐が離れていて絡まらない交差点でありこれをB型交差点と呼ぶ。交差点を座標で表現するとき $C(x, y, (z_+; z_-))$ と記す。

ここに、 z_+ は交差点の上側の紐の z 座標、 z_- は下側の紐の z 座標であり、 $z_+ - z_- > 2r$ のときはB型交差点であり、 $z_+ - z_- = 2r$ のときA型交差点である。

なお、これらの交差点は p^1, p^2, p^3 および p^4 と連結しているものとし、これらの4点を交差点の連結点と呼ぶ。さらに、紐図形の境界に紐が到達している中心点を端点 E とする。3次元空間内の端点 E を座標で表現するときは $E(x, y, z)$ と記す。紐図形は複数の交差点 C と複数の端点 E およびこれらの連結より構成される。

ここで、交差点 C が連結点 p^1, p^2, p^3 および p^4 を持つときその表現を以下とする。

$$\{C_\phi(x, y, (z_+; z_-)), p^1(x^1, y^1, z^1), p^2(x^2, y^2, z^2), p^3(x^3, y^3, z^3), p^4(x^4, y^4, z^4)\}. \quad (1)$$

ここに、 p^i は他の交差点 $\ast\ast$ または端点である。 $p^i(x^i, y^i, z^i)$ は p^i の3次元空間座標表現である。紐 $(p^1 - p^2)$ が紐 $(p^3 - p^4)$ の上にあるときは $\phi = +$ 、その逆のときは $\phi = -$ とする。なお、交差点 C がA型交差点であるかB型交差点であるかは、各 p^i か

\ast 以降の議論で編物の広がりを見易く xy 平面とするため、ここでは z 軸方向とする

$\ast\ast$ 厳密には後述する交差点の近傍点である。

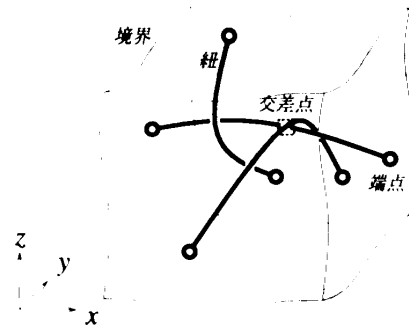


図1 紐図形

Fig. 1 A string diagram.

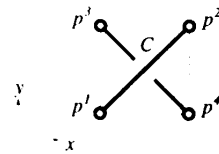


図2 z 軸方向から見下ろした交差点

Fig. 2 Crossing point from z axis direction.

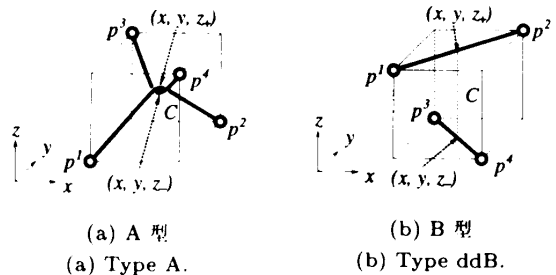


図3 3次元空間上の交差点 C

Fig. 3 Crossing point of 3D.

ら交差点 C までの紐が直線であると仮定すれば、 p^i の座標値に依存しているので判定可能である。

3. 交差点移動方法

ここでは、紐図形に存在する交差点の移動処理方法について述べる。

3.1 ライデマイスター移動による紐図形の簡単化
文献5)~10)と同様にライデマイスター移動¹⁾を z 軸方向から見た xy 平面上の紐図形に施す(図4参照)。ここに、ライデマイスター移動 I, IIでは交差点を減少させ、また、ライデマイスター移動 IIIでは、紐のたわみをなくし紐長を縮小させる。なお、これらの3つのタイプの移動は、A型交差点およびB型交差点を区別せずに施す。

3.2 ベクトル和による交差点移動

ここで述べる3次元ベクトル和による交差点の移動はA型交差点のみに施す。任意のA型交差点 C と連

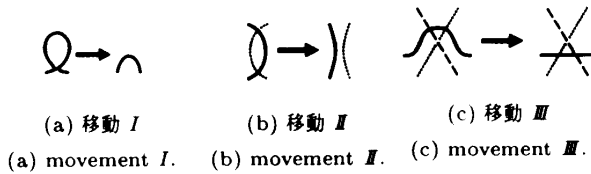


図4 ライデマイスター移動 I, II, III
Fig. 4 Reidemeister movements I, II, III.

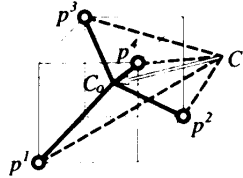


図5 ベクトル和が0となる点 C_0
Fig. 5 C_0 has 0 vector summation.

結する4連結点 p^1, p^2, p^3 および p^4 への C から向かう単位ベクトルの和 V_C を式 (2) とする。

$$V_C = \sum_{i=1}^4 \frac{p^i - C}{|p^i - C|} \quad (2)$$

ここに、 C, p^i はそれぞれ C, p^i のベクトル表現である。

3次元空間内で交差点 C は可動で、4連結点 $p^i (i = 1, \dots, 4)$ が固定しているとき、交差点 C と4連結点 $p^i (i = 1, \dots, 4)$ との距離の和が最小となる点をとくに交差点 C_0 とする (図5参照)。このとき、 C_0 に関して式 (3) が成り立つ¹²⁾。

$$V_{C_0} = 0. \quad (3)$$

ここで、交差点を3次元物体としてとらえ、紐の太さ (半径 r) を考慮するため、交差点の4近傍点 (c^1, c^2, c^3, c^4) を付録に示す方法で定め、以降では式 (2) の代わりに4近傍点 (c^1, c^2, c^3, c^4) を用いた式 (4) を用いる。

$$V_C = \sum_{i=1}^4 \frac{p^i - c^i}{|p^i - c^i|} \quad (4)$$

ここに、 c^i は c^i のベクトル表現である。次に、交差点 C を移動し、式 (4) のベクトル和 V_C の大きさを小さくして、安定した紐図形を逐次的に得る方法について述べる。

3.3 交差点の逐次的移動

紐図形は複数の交差点の集まりであるので、隣合う2つの交差点 C, C' は互いに相手の連結点となっている。このことから、交差点 C を移動することは、他の交差点 C' の連結点が移動することになり、一度に

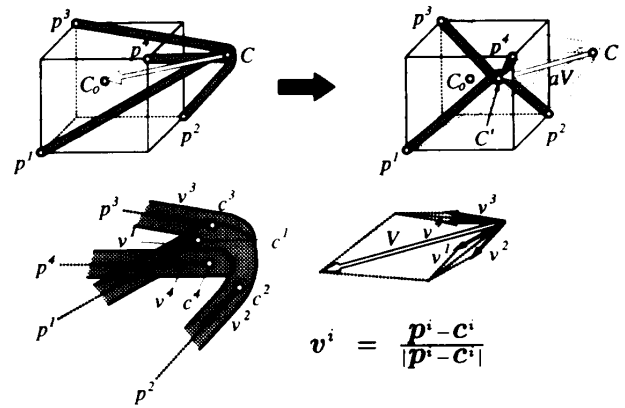


図6 ベクトル和による交差点の移動
Fig. 6 Crossing point, moved to vector summation.

C をベクトル和 V_C が0となる点 C_0 へ移動すると、互いに他のベクトル和に干渉する。このため、各交差点のベクトル和の大きさが減少する方向に、逐次的に移動する方法を提案する。すなわち、交差点 C に対して、4連結点 p^1, p^2, p^3 および p^4 の方向から力が働いていると仮定して、それらの和 V_C の方向に交差点 C を移動させ、これを逐次的に繰り返すことにより安定した位置に移動する方法をとる。この移動を全交差点に対し繰返し適用し、紐長を漸時短くしていき、安定した状態へと変形する。ここでは、紐長が縮小していく過程で、移動すべき交差点がなくなる状態を安定状態と呼ぶ。

なお、これらの移動過程においては、紐長の縮小化および交差点の移動により紐が移動することから、B型交差点がA型交差点となる場合、または逆にA型交差点がB型交差点となる場合があるが、これらは前述したように判定可能である。

3.3.1 交差点の移動係数

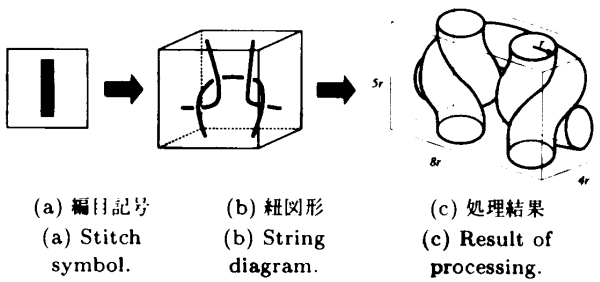
逐次的な移動を効率的に行うため移動係数 a を導入する。いま、ある状態の紐図形にある交差点 C の移動先を C' とする。このとき、式 (5) が成り立つ。

$$C' = C + aV_C. \quad (5)$$

すなわち、交差点 C をベクトル V_C の方向に距離 $a|V_C|$ 移動する (図6参照)。

3.3.2 移動係数固定方式

この方式は毎回の移動に用いる移動係数 a を固定にする方式である。図7(a)に代表的な「表目」の編目記号を示す。図7(b)はこれを紐図形として表現したものであり、図7(c)にはこの紐図形に処理を繰り返したときの結果を示す。図7(b)の紐図形に対して、逐次的交差点移動を繰り返したときの移動回数と紐長



(a) 編目記号 (a) Stitch symbol. (b) 紐図形 (b) String diagram. (c) 処理結果 (c) Result of processing.

図7 交差点を持つ紐図形の例 (表目)

Fig. 7 An example of the string diagram including crossing points (knit-stitch).

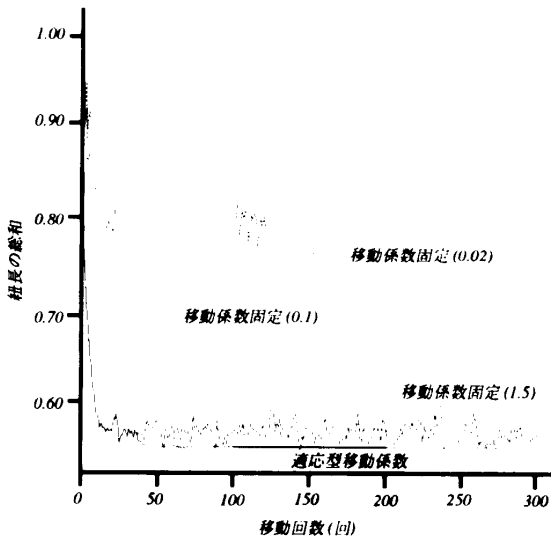


図8 表目における移動係数固定型方式と適応型方式の比較
Fig. 8 Comparison between the fixed move coefficient and the adaptive move coefficient in knit-stitch.

の相関を図8*に示す。

図8は移動係数固定方式で $a = 1.5$, $a = 0.1$ および $a = 0.02$ のときのこれらの相関を表す。 $a = 1.5$ の場合は、交差点どうしの相互干渉が継続し、いつまでも安定状態を得ない。また $a = 0.1$ および $a = 0.02$ の場合は、安定状態を得るのに時間を要す。

3.3.3 適応型移動係数方式

3.3.2のような移動係数固定方式の問題点を解決するため、移動係数を可変とする方式を示す。これをここでは適応型移動係数方式と呼ぶ。

いま、 t 回目の交差点移動のときの移動係数 a を a^t とする。 a^t の増加率を τ_{inc} 、減少率を τ_{dec} とする。ただし、 a^t の増加、減少は式(6)による紐図形における紐の長さの総和の減少効率 S^{t-1} によって定める。

すなわち、 D^t を t 回目の交差点移動における紐長

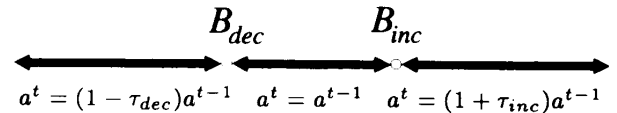


図9 適応型移動係数 a^t の決定方法
Fig. 9 Deciding method of an adaptive move coefficient a^t .

の減少値とし、 M^t を t 回目の交差点移動で各交差点が移動した距離の合計とする。このとき、 S^{t-1} は、i) $t-n$ 回から $t-1$ 回までの各 D^{t-i} ($1 \leq i \leq n$) の和と、ii) $t-n$ 回から $t-1$ 回までの各 M^{t-i} ($1 \leq i \leq n$) の和との比とし、式(6)で表現する。

$$S_n^{t-1} = \frac{\sum_{i=1}^n D^{t-i}}{\sum_{i=1}^n M^{t-i}}, \text{ ただし } (t > n). \quad (6)$$

図9に示すように、 S_n^{t-1} がしきい値 B_{inc} よりも大きければ、すなわち順調に紐の長さが減少しているとき $a^t = (1 + \tau_{inc})a^{t-1}$ 、 S_n^{t-1} がしきい値 B_{dec} よりも小さければ、すなわち紐の長さの減少が停滞しているとき $a^t = (1 - \tau_{dec})a^{t-1}$ とする。また、それ以外のときは $a^t = a^{t-1}$ 、初めの n 回は $a^t = a^0$ ($t = 1, \dots, n$) とする。さらに、 a^t に下限値 a_{min} を設け、減少時に $(1 - \tau_{dec})a^{t-1}$ が a_{min} より小さければ、 $a^t = a_{min}$ とする。

図8に適応型移動係数方式による処理効果を示す。ここでは以下の数値を用いた。

$$\tau_{dec} = 0.05, \tau_{inc} = 0.05, B_{dec} = 0.01, \\ B_{inc} = 0.5, a_{min} = 0.02, a^0 = 1.5, n = 5.$$

なお、これらの数値は紐長の減少効率の良い場合における実験値である。

4. 紐図形の安定化アルゴリズムとその処理

ここでは、前述したライデマイスター移動とベクトル和による交差点移動を用いて、安定した紐図形を得る処理について述べる。

図10(a)に編み方を表記するための編目記号の一部を示す。これらの編目記号を組合せたものを模様編み図と呼び、その一例を図10(b)に示す。

入力として与える紐図形を初期図形と呼び(たとえば図10(c)は図10(b)に対応する初期図形である)、まず、初期図形として端点の座標、交差点の初期座標、および、交差点の近傍点または端点の連結関係を与えてこれを以下のアルゴリズムで処理する。

* ここでは、自然感を持つ編みを得るため x 幅を $8r$, y 幅を $5r$ および z 幅を $4r$ として求めた。

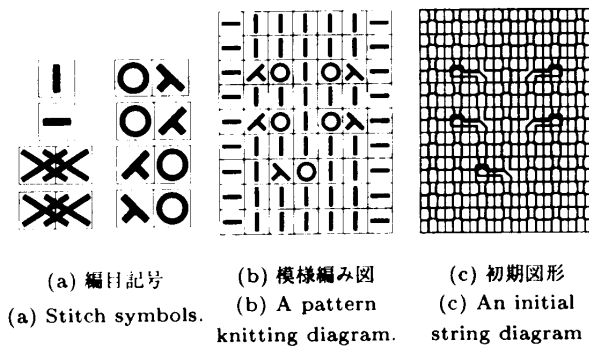


図 10 編目記号, 模様編み図, 初期図形

Fig. 10 Stitch symbols, A pattern knitting diagram, An initial string diagram.

4.1 紐図形安定化アルゴリズム

紐図形安定化アルゴリズムを以下に示す. ここに, パターン I, II, IIIとは, それぞれライドマイスター移動 I, II, IIIを適用できる紐図形の部分図形である (図4 参照). また, 最大ステップ (MAXSTEP) 回だけ交差点移動を繰り返す. なお, 以下に示すアルゴリズム中のパターン IIIの接近度 W とはパターン IIIを構成する3つの交差点間の距離の和である. また, TH はライドマイスター移動 IIIを適用するか否かを決定する際の W のしきい値である. 交差点 C_i は i 番目の交差点, C_i は C_i のベクトル表記である.

t = 1;

```
while (t ≤ MAXSTEP) {
    ライデマイスター移動 ();
    ベクトル和交差点移動 (t);
    t = t + 1;
}
```

手続き ライデマイスター移動 ()

```
{
    /* ライデマイスター移動に関しては交差点
       の型は考慮しない */
    while (紐図形にパターン I, IIが存在する) {
        ライデマイスター移動 I, IIをすべての
        パターン I, IIに施す;
    }
```

```
for (紐図形中のパターン IIIの集合) {
    接近度 W を求める;
    if (W < TH)
        ライデマイスター移動 IIIを
        このパターン IIIに施す;
}
```

```
}
```

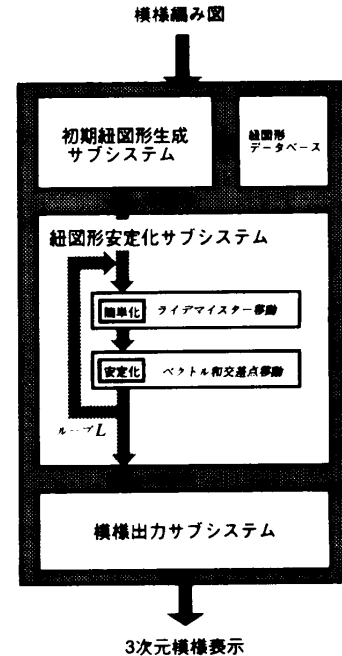


図 11 システム構成と処理の流れ

Fig. 11 System scheme and its processing flow.

手続き ベクトル和交差点移動 (入力: t)

```
{
    for ( $C_i \in$  紐図形中の交差点の集合) {
        交差点  $C_i$  の型を連結点の座標から判定する.
        if (交差点  $C_i$  が A 型交差点) {
            交差点  $C_i$  から連結点へのベクトル和  $V_{C_i}$  を求める;
            if ( $t > n$ ) {
                if ( $S_n^t > B_{inc}$ )
                     $a^t = (1 + \tau_{inc})a^{t-1}$ ;
                else if ( $S_n^t < B_{dec}$ )
                     $a^t = (1 - \tau_{dec})a^{t-1}$ ;
                else
                     $a^t = a^{t-1}$ ;
            }
            else
                 $a^t = a^0$ ;
             $C_i = C_i + a^t V_{C_i}$ ;
            交差点  $C_i$  の新しい近傍点を求める;
        }
    }
}
```

4.2 紐図形安定化処理

与えられた初期図形に対し、ライデマイスター移動とベクトル和による交差点移動を繰り返すことにより、安定な紐図形を得る。これを紐図形安定化処理と呼び、図 11 中のループ L を一定回数繰り返す。ループの前半ではライデマイスター移動を行い、後半では交差点のベクトル和による交差点移動を行う。

5. 3次元編目模様生成システム

ここでは、紐図形処理手法の応用として作成した編物パターン表示支援ツールとして利用できる“3次元編目模様生成システム”について述べる。

ユーザは編物の設計図である模様編み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。システムは入力した模様編み図から編み上がり模様のリアルな表示を出力する。

システムは図 11 に示すように 3つのサブシステムから構成される。以下、各サブシステムについて述べる。

5.1 初期紐図形生成サブシステム

ここでは、入力された模様編み図から紐図形の初期図形 (図 10 (c) 参照) を生成する。これには、編目記

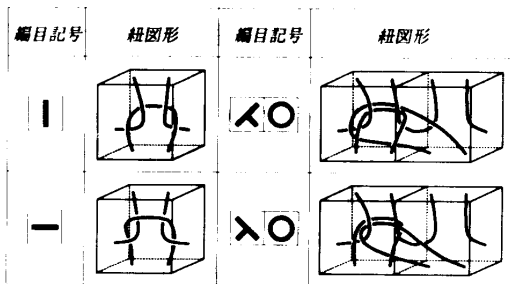


図 12 紐図形データベース (例)

Fig. 12 An example of stitch pattern database.

号と紐図形を対応づける紐図形データベース (図 12) を用いる。システムは紐図形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、模様編み図から紐図形を生成する。

5.2 紐図形安定化サブシステム

ここでは、初期紐図形生成サブシステムによって生成された初期図形に対し安定化処理を施し、安定した交差点の連結および位置を求める。

5.3 模様表示サブシステム

編物における立体感を強調して表現するため、紐図形安定化サブシステムにより求められた最適な交差点の連結、座標データから、レイトレーシング¹³⁾手法により編み上がり模様を表示する。ここでは、交差点間の線分を円筒形で、その継ぎ目を球体の一部として表している。

6. 出力例と評価

図 13 (a)、図 14 (a) および図 15 (a) に例示する模様編み図を 3次元編目模様生成システムの入力としたときの出力結果を示す。

i) 図 13 (a) を入力として紐の半径を r とし、 x 軸幅を $48r (= 6 \times 8r)$ 、 y 軸幅を $30r (= 6 \times 5r)$ 、 z 軸幅を $4r$ のときの出力を図 13 (b) に、また紐の半径を $0.7r$ とし、 x 軸幅、 y 軸幅および z 軸幅は図 13 (b) と同様にした場合の出力を図 13 (c) に示す。

ii) 図 14 (b) に図 14 (a) を入力としたときのシステムの出力を示す。図 14 (c) は図 14 (b) の図形の y 軸方向からの表示であり、また同様に、

iii) 図 15 (b) に図 15 (a) を入力としたときのシステムの出力を示す。図 15 (c) は図 15 (b) の図形の y 軸方向からの表示である。

ここに述べたシステムの交差点移動処理時間は図 16

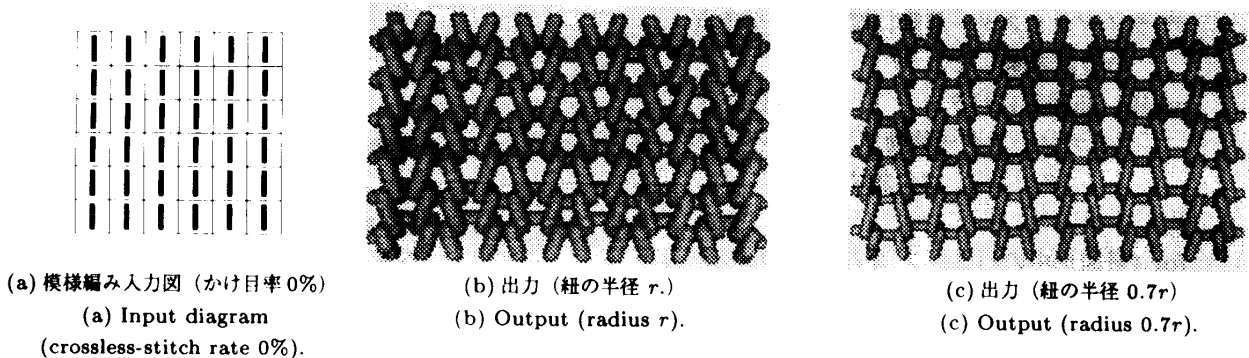


図 13 出力例 (1)

Fig. 13 An output example (1).

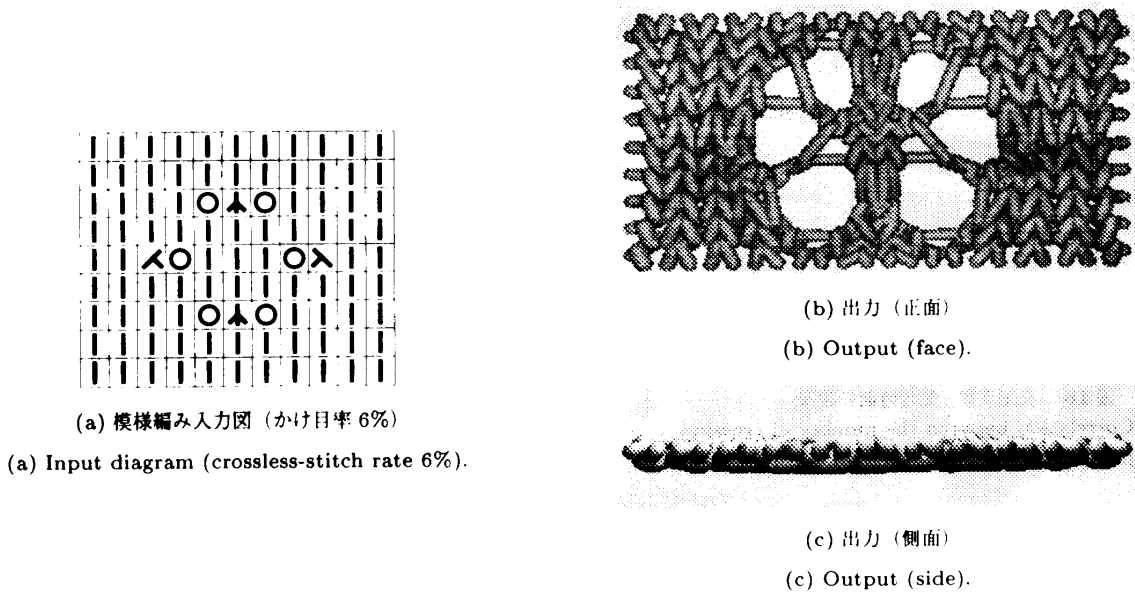


図 14 出力例 (2)
 Fig. 14 An output example (2).

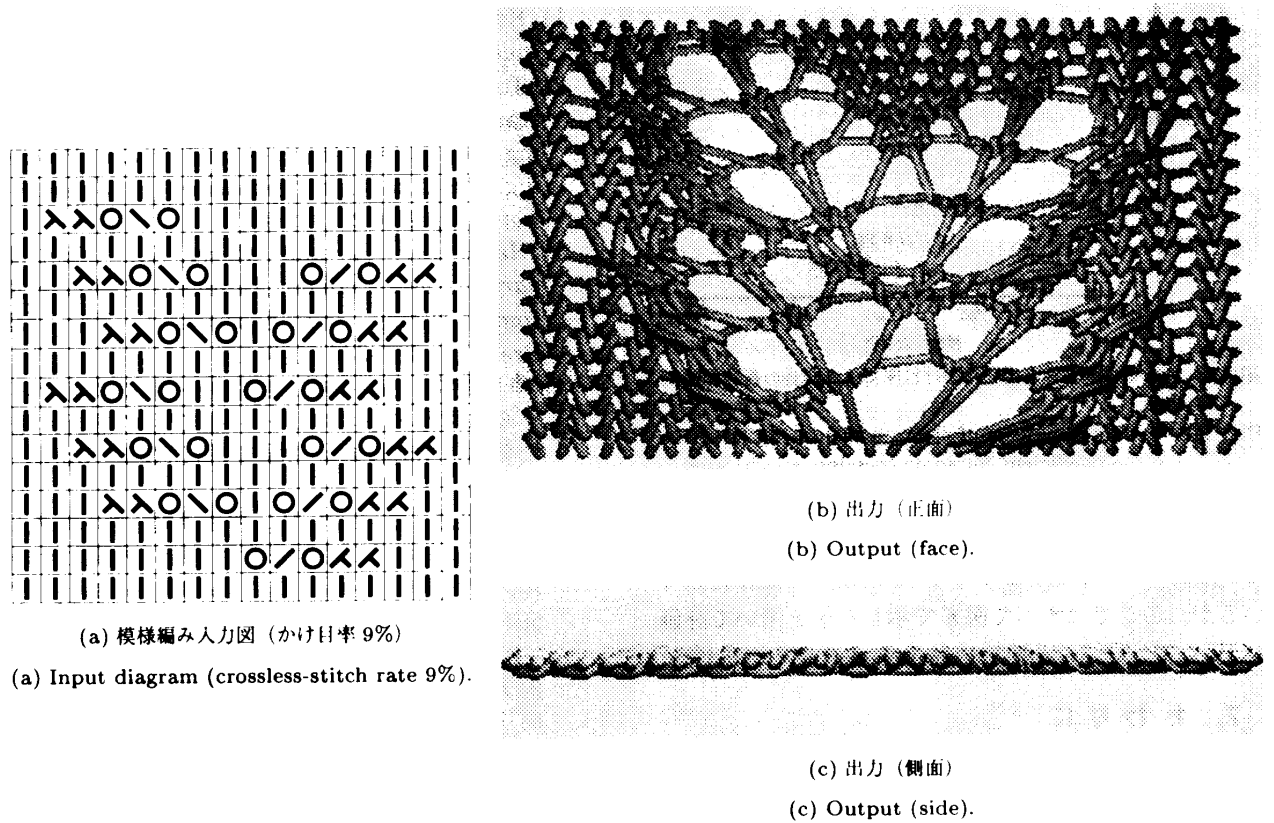


図 15 出力例 (3)
 Fig. 15 An output example (3).

に示すように、与えられる模様編み図の交差点数にほぼ比例する。また、編物では複雑な模様を表現するときはかけ目（編目記号では○で記す）率が増加するのが一般的であり、このときは紐長の減少率が大きくなる。ここで与えた例におけるかけ目率はそれぞれ図 13

では 0%，図 14 では 6%，図 15 では 9% である。本論文で処理対象とした家庭用編み機編みと棒針編みでは、ある程度の編物の強度を保つためにかけ目率は最大 10% 程度が一般的である¹⁴⁾。図 17 にかけ目率 0%，5%，10% における紐長の減少効果と交差点移動

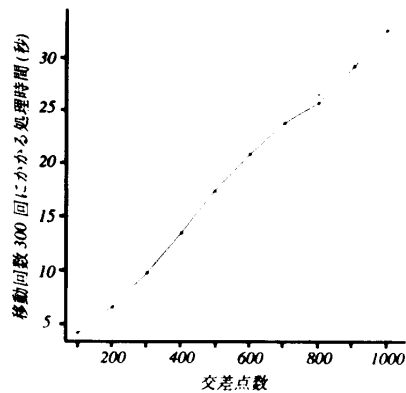


図 16 交差点数と処理時間の相関

Fig. 16 Correlation between the number of crossing points and the processing time.

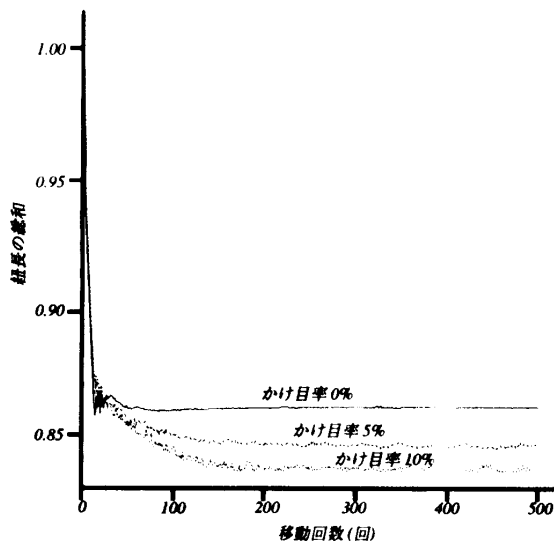


図 17 適応型移動係数を用いたかけ目率による紐長の総和の減少効率

Fig. 17 Reduction effects of string length using adaptive move coefficient for some crossless-stitch rates.

回数との関係を示す。なお、かけ目率によって、前述したパラメータ τ , B , a および n に最適値が存在しているが、ここではすべて図 8 で示した値を用いて評価した。

7. おわりに

本論文では紐図形を 3 次元図形として表現し、処理するための手法を提案した。この手法では紐図形を交差点の集合として表現し、交差点および連結関係に対し移動を繰り返すことで安定した紐状態を得ることができる。また、この手法を実装した編物パターン表示を支援できるシステムを作成した。このシステムは、入力された模様編み図を、紐図形データベースを用いて紐図形に変換し、この紐図形に対し処理を施すことにより編み上がり模様の 3 次元イメージを表示する。

編目単位 24 種の隣合う n 個の組合せは 24^n 種類存在するが、ここで述べた手法は、これらを初期図形として与えたものはすべて 3 次元表示可能である。また、他の編目パターンにおいても同様にして処理できることより紙面の都合で代表的な例の処理を示すにとどめた。

なお、今後の課題としては、

- i) 紐の交差する部分の力のかかり具合から半径の縮みを考慮して表現すること、
- ii) 紐が幾重にも重なる場合の自然感を表現すること、
- iii) 編目模様のかげ目率や複雑さに対応させた最適なパラメータを設定した適応型移動係数による表示を行うこと、さらに、
- iv) 編物の広がりここでは xy 平面としたが、凹凸物体を被覆する場合の編目パターンを表現することなどがある。

参考文献

- 1) Kauffman, L.: *On Knots*, *Annals of Mathematics Studies*, Princeton Univ. Press (1987).
- 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*, pp.219-231 (1990).
- 3) Hasslacher, B. and Meyer, D.: *Knot Invariants and Cellular Automata*, *Cellular Automata Theory and Experiment*, The MIT Press, pp.328-344 (1991).
- 4) 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 PRU83-90, pp.15-22 (1990).
- 5) 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).
- 6) 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).
- 7) 伊藤裕一郎, 山田雅之, 世木博久, 伊藤英則: 遺伝的アルゴリズムを用いた 3 次元編み物表示, 第 50 回情報処理学会全国大会予稿集 (1995).
- 8) Yamada, M., Budiart, R., Itoh, H. and Seki, H.: A String Diagram Transformation Process using a Genetic Algorithm—A Cat's Cradle Diagram Generating Method, *Proc. of PRI-CAI'94*, Vol.1, pp.429-434 (1994).
- 9) Yamada, M., Budiart, R., Itoh, H. and Seki, H.: A Cat's Cradle String Diagram Dis-

play Method Based on a Genetic Algorithm, *FORMA* (for Science on Form, T.S. (ed.)), Vol.9, No.1, pp.11-28 (1994).

- 10) Yamada, M., Itoh, H. and Seki, H.: An Implementation of a Knit Designing System Based on a Genetic Algorithm, *Proc. of ICARCV'94*, Vol.1, pp. 1277-1281 (1994).
- 11) 日本規格協会：編目記号, JIS L 0201, 日本工業規格 (1978).
- 12) 岩田至康 (編)：幾何学大辞典, 2 基本定理と問題—空間—, 槇書店, pp.234-235 (1974).
- 13) 千葉則茂, 村岡一信：Cによるレイトレーシング, サイエンス社 (1991).
- 14) 瀬戸信昭 (編)：春夏の模様 200, 日本ヴォーグ社 (1994).

付録 交差点の近傍点の求め方

紐 I と紐 II が点 α から点 β まで接しており, その中点を交差点 C とする (図 18 参照). また, 紐 I は点 p^1, p^2 の方向に伸びていて, 紐 II は p^3, p^4 の方向に伸びているものとする. ここで, C を通り紐 II を断面が円となる面で切断し, その中心を O とする (p^3 の方向とは正確には他の交差点 C' または端点 E' が存在する方向である). このとき, p^1 へ伸びる紐 I

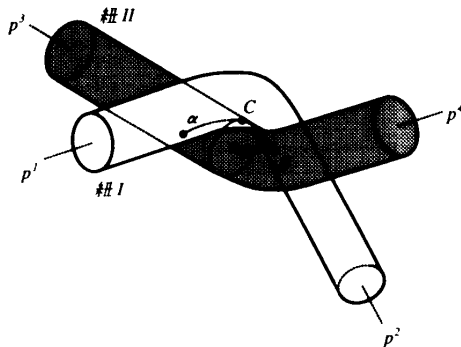


図 18 A型交差点 C
Fig. 18 Type A crossing point C.

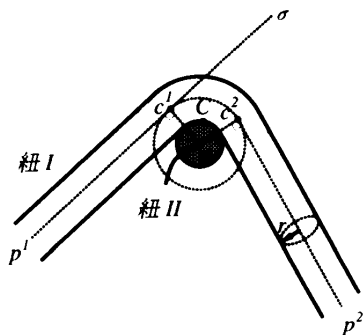


図 19 A型交差点 C における近傍点 c^i
Fig. 19 Adjacent points c^i of type A crossing point C.

の中心線 σ へ中心 O からの垂線の交点を c^1 とする (図 19 参照). 同様にして, c^2 を決定する. さらに, 紐 II についても同様に c^3, c^4 を決定する. ここで決定する c^1, c^2, c^3 および c^4 を交差点 C の 4 近傍点と呼ぶ.

(平成 7 年 3 月 23 日受付)
(平成 7 年 11 月 2 日採録)



伊藤裕一郎 (学生会員)

1995 年名古屋工業大学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.



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.



宮崎 剛 (正会員)

1995 年名古屋工業大学工学部知能情報システム学科卒業. 同年, 日本電気株式会社に入社, 現在に至る. 在学中は, 画像処理と画像認識の研究に従事.



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. 1985~1989 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。
