

結び目、自己相似などを用いたグラフィック処理について

伊藤英則[†], 山田雅之^{††}, 中村剛士[†]

[†]名古屋工業大学知能情報システム学科, ^{††}中京大学情報科学部

あらまし: 本論文は, 1) 結び目理論に基づくひも図形処理として, あやとり, 編み物に関するグラフィック処理について述べる. また, 2) 自己相似理論と繰りこみ群理論を用いた毛筆文字の掠れ生成および滲み処理, および, 似顔絵生成について述べる.

Graphic Processing Methods on Knot theory and Fractal theory

Hidenori ITOH[†], Masashi YAMADA^{††}, Tsuyoshi NAKAMURA[†]

[†]Department of Intelligence and Computer Science, Nagoya Institute of Technology

^{††}School of Computer and Cognitive Sciences, Chukyo University

Abstract: In this paper, 1) cat's cradle string pattern and knitting graphical pattern process are described on knot theory. And 2) a scratched-look expression and a blurred-look Expression on calligraphic characters and a face caricature using fractals theory and Renormalization group theory are described.

1 はじめに

近年, コンピュータグラフィック処理の進展は著しい. 例えば, アニメーション映画やゲームマシンの画面作成などには高度な多数の技術が使われている. このような状況の中で, 何が新規性であるのか, その把握が困難になってきている. 我々が独創性を保ってこの分野の研究を実施続行することは狭き門で困難になってきているように思えるし, また, 返って競争的で興味深いようにも思える. ここではまず, グラフィック情報処理の観点から見て, 特異性をもつ「ひも」処理, すなわち, 「あやとり」と「編み物」について 2, 3 の興味深い結果を得たのでこれを紹介する. つぎに, 東洋の感性で美と解釈されている毛筆文字の掠れと滲み表現について述べる. また, この手法を用いて似顔絵の毛髪表現に現実感を出す手法について紹介

する.

2 結び目理論を用いたひも図形処理

ここでは, ひもで作られる形状の図形表現とその処理について述べる.

ひも形状は切れたり交わることのない 3 次元空間の折れ線で表現できる. このような折れ線について, そのトポロジーを調べ, 分類したり特徴を見つけれたりするのが結び目理論である.

結び目理論では, 図 1 のような図形で 3 次元空間の折れ線を表現する. A, B, C はそれぞれ閉じた折れ線(結び目)を表す図形である. 折れ線の変形はこの図形のライデマイスター移動で表現する. ライデマイスター移動は 3 種類あり, これらを図 2 に示す. 図 1 の A と C は互いに他方へ変形でき, この変形は図 3 に示す

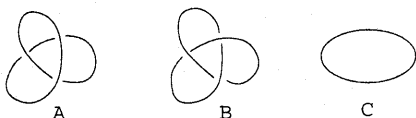


図 1: 3つの結び目

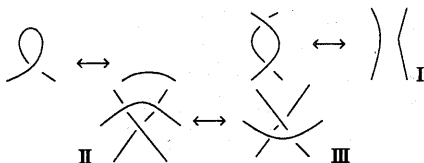


図 2: ライデマイスター移動 I, II, III

ように 2 回のライデマイスター移動で表現できる。一方、図 1 の B はライデマイスター移動をどのように使っても、A や C には変形できないので B は異なる結び目と判定できる。

このような結び目理論の手法は複雑なひも形状を計算機上に表現し、効率よく処理する場合にも有効である。本稿では、あやとりや編物への応用について述べる。

2.1 あやとり

あやとりは世界中で親しまれているあそびであり、人類学的側面、幾何学的側面からの研究も多く行われている。

あやとりを計算機上で表現しようとするといくつかの困難な問題に直面する。そのひとつに、あやとりの操作の表現がある。あやとりの操作は、指を複雑に動かし、ひもを取ったり離したりする。感覚的に操作を覚えていくこともあり、他人に説明するのも難しい。

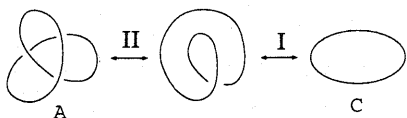


図 3: ライデマイスター移動による変形表現

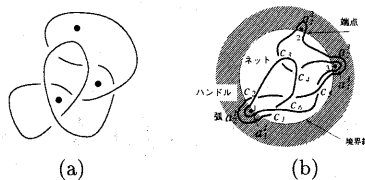


図 4: 点を含むひも図形と 2つの領域

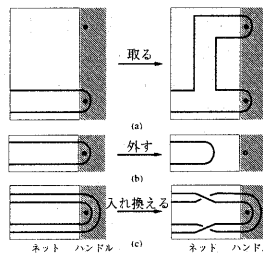


図 5: 基本操作

ここでは、このようなあやとりの操作を単純な 3 つの操作の組み合わせと考え、図形を使って表現する試みを紹介する [7]。

あやとりの図形表現

あやとりのひもは単純な輪であり、あやとりの形は指とひもとの位置関係で決まるので、図形中に指を点で表す。一般に、点を含むひもの図形(図 4(a) 参照)、は二つの領域に分けることができる(図 4(b) 参照)。ひとつは点を含み交点を含まない領域(図中網掛け)で、もう一つは交点を含み点を含まない領域である。前者をハンドル部、後者をネット部と呼ぶ。あやとりの図形をこのような二つの領域に分けて考えると都合がよい。

基本操作と畳み込み表現

ここでは、3 つの基本操作、i) 取る、ii) 外す、iii) 入れ換えるを考える。これを図形で表現したものを図 5 に示す。

基本操作に対応する図形をあやとりの図形のネット部に逐次追加してできる図形を畳み込み図形と呼ぶ。以下では、あやとりの変形過程がこの畳み込み図形で表現できることを示そう。まず、例として、あやとりの代表的な変形過程

であるナバホどりを畳み込み図形で表現する。

ナバホどりは図6(a)の (k_0) のように二本のひもが指に掛かっている状態から、変形を繰り返して、同図(b)の形を作る取り方である。ナバホどりはi)取る、ii)入れ換える、iii)取る、iv)入れ換える、v)外す、vi)外すの一連の基本操作の繰り返しであり、この過程は基本操作に対応する図形をネット部に逐次追加して表現できる(図6(a)参照)。

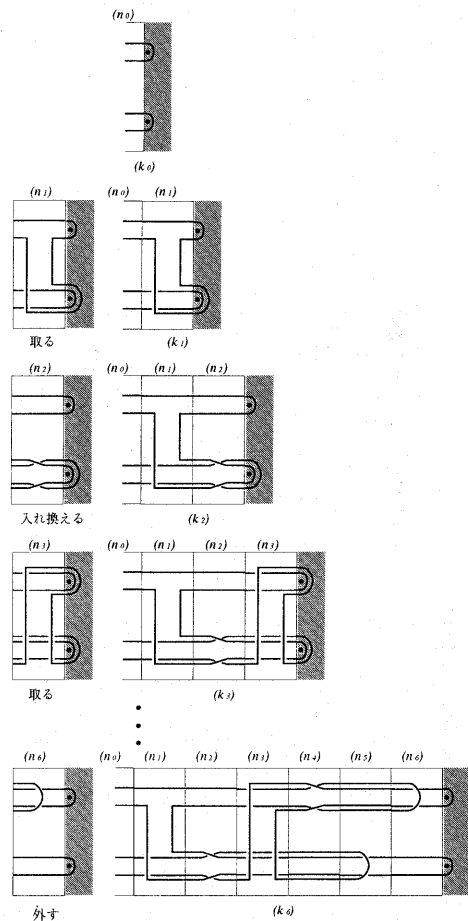
- i) k_0 のハンドルを基本操作「取る」に対応する図形に置き換えることにより k_1 が得られる。このとき k_0 のネット n_0 は k_1 にそのまま保存される。
- ii) k_1 のハンドルを基本操作「入れ換える」に対応する図形に置き換えることにより k_2 が得られる。このとき k_1 のネット n_1 は k_2 にそのまま保存される。
- iii) k_2 のハンドルを基本操作「取る」に対応する図形に置き換えることにより k_3 が得られる。このとき k_2 のネット n_1, n_2 は k_3 にそのまま保存される。
- iv)~vi) 同様に、基本操作「入れ換える」、「外す」、「外す」に対応する図形を順次ハンドルと置き換える。

このようにしてできた図6(a)の (k_6) がナバホどりの畳み込み表現である。この表現は変形過程が1つの図形で表現でき、また、ライドマイスター移動(図2)を使って出来上がりの形(図6(b))に変形できる。

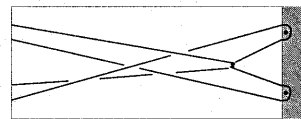
われわれは畳み込み表現がどのようなあやとりを表現できるか調査した。図7に示すあやとり(ほうき、星、蝶)の畳み込み表現を図8に示す。全てのあやとりが畳み込み表現可能かどうかは、残念ながら証明が困難なため明らかにはできないが、調査した代表的なあやとりについては畳み込み表現できた。なお、1つのあやとりにも複数の変形過程が存在するため、畳み込み表現は一意ではない。

2.2 編物デザイン

編物は一本のひもから様々な模様を作る工芸である。編物には、棒針編、かぎ針編、機械編

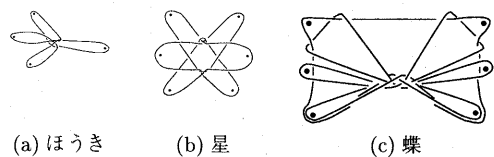


(a) ナバホどり過程の畳み込み表現



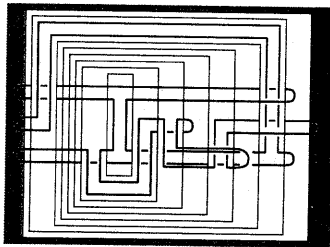
(b) ナバホどりが作る形

図6: ナバホどりの畳み込み表現とその形

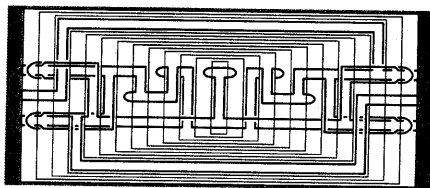


(a) ほうき (b) 星 (c) 蝶

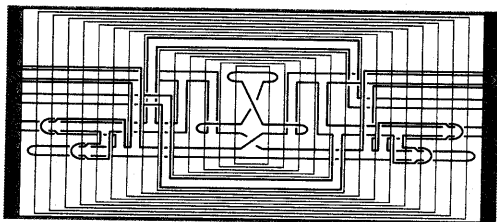
図7: いろいろなアヤトリ



(a) ほうきの畳み込み表現



(b) 星の畳み込み表現



(c) 蝶の畳み込み表現

図 8: いろいろなアヤトリの畳み込み表現

表目 1 ↓	裏目 2 ↑	かけ目 3 ○	右上2目一度 4-1 ↘ 4-2 ↗	左上2目一度 5-1 ↗ 5-2 ↘	中上3目一度 6-1 ↑ 6-2 ↓
右上3目一度 7-1 ↗ 7-2 ↘	左上3目一度 8-1 ↘ 8-2 ↗	右増目 9-1 ↑ 9-2 ↓	左増目 10-1 ↓ 10-2 ↑	編出し増目	
11 V	11-a ○	11-b ○	11-c ○	右上交差 12-1 X 12-2 X	左上交差 13-1 X 13-2 X
右目を通す交差 14 X	左目を通す交差 15 X	滑目 16 V	滑目 17 A	引上げ目 18 ↑	おじり目 19-1 ○ 19-2 ○
巻目 21 ⊗	かぶせ目 22-a ↓ 22-b ↑				

図 9: 編目と JIS 記号

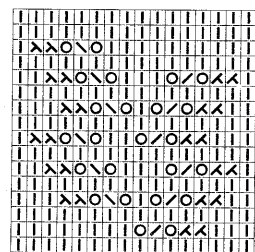
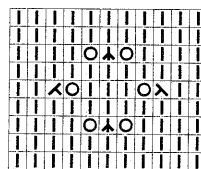


図 10: JIS 記号配列

などの種類があるが、ここでは棒針編のみを対象とする。

編物の模様はいくつかの部分的なひもの絡みパターンの繰り返しであり、この部分パターンを編目という。編目は日本工業規格 (JIS) の記号で表記でき [8], 棒針編には 22 種類の記号が用意されている (図 9 参照)。模様全体はこの記号を組み合わせた JIS 記号配列で表せる (図 10 参照)。通常、編物をする場合、JIS 記号配列をみながら模様をつくるが、初心者にはこれがどのような模様を表しているか予想しにくいし、自由に模様をデザインすることも難しい。

ここでは JIS 記号配列を編み上がり模様へ変換する方法について述べる [4]。

模様のトポロジカルな構造

模様を構成する編目のパターンは予め列挙できる。図 11 には代表的な編目の JIS 記号と

編目記号	パターン	編目記号	パターン
Ⅰ		Ⅱ	
Ⅲ		Ⅳ	

図 11: 編目の JIS 記号とパターン

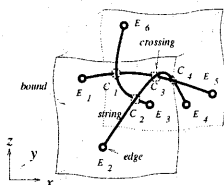


図 12: ひもの構造

ひものパターンを示す. 1つの記号に1つのパターンが対応するとは限らず, 隣接する二つの記号が明らかでないと, パターンが決定できない場合もある.

模様全体のひも構造は JIS 記号配列に従って編目のパターンを連結すれば求めることができる.

編み上がり形状の生成

JIS 記号配列から模様のトポロジカルな構造が得られたら, 変形を施し編み上がりの形状を生成する.

ここでは, 複雑なひも形状を効率よく処理するため, ひも形状を交点集合により表現する. 交点とは 3次元空間内のひも形状 (図 12 参照) を xy 平面へ射影したときに 2本の線分が交わる点であり, ひも形状は互いに連結する交点の集合で表現できる. さらに, 各交点を図 13 の (a), (b) の 2つの型に分ける. (a) は上の線分と下の線分が接して絡まる交点でありこれを A 型交点, (b) は上の線分と下の線分が離れていて絡まらない交点でありこれを B 型交点と呼ぶ. このように交点単位で表現し, A 型か B 型かを区別することで, 後の処理を効率よく行え, 良好な形状を得ることができる.

ひも形状は様々に変形できるが, 編み上がり

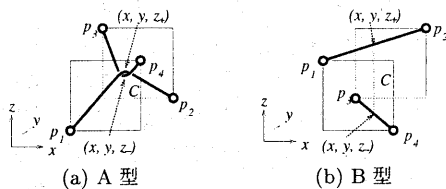


図 13: 3次元空間上の交点 c

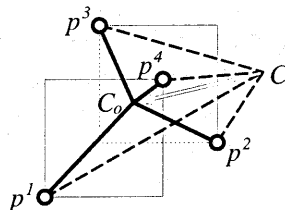


図 14: 交点移動

形状はその中の一状態と考える. 実際の編物が編目を密にあんでいくことに着目し, 編み上がり形状は (i) 交点数が最小で, かつ, (ii) 交点に関する力学的なベクトル和が最小な状態と仮定し, このような形状を求める. (i) についてはライデマイスター移動を利用し, (ii) については, 次に述べるような交点移動を行う.

ベクトル和による交点移動

任意の A 型交点 c について, その位置を C で表し, 4つの連結点 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 (1) とする.

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

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次元空間内で交点 c は可動で, 4連結点 p^i が固定しているとき, V_c の大きさが最小となる交点位置を C_0 とすると (図 14 参照), C, C_0, V_c の間に式 (2) が成り立つ.

$$C_0 = aV_c + C. \quad (2)$$

ここで, a を移動係数と呼ぶ. a に適当な値を与え, 交点 c を C_0 の方向へ aV_c だけ移動することをベクトル和交点移動と呼ぶ.

模様は複数の交点の集まりなので隣合う 2つの交点 c, c' は互いに相手の連結点となってい

る。このことから、交点 c を移動することは、他の交点 d の連結点が移動することになり、互いに他のベクトル和に干渉する。このため、各交点をベクトル和の大きさが減少する方向に逐次的に移動し、これを全ての交点に繰り返す。なお、B 型交点についてはベクトル和を最小にする交点位置が幾何学的に求まるので、ただちにその点へ移動する。これらの移動過程においては、交点の移動により線分が移動することから、B 型交点が A 型交点となる場合、または逆に A 型交点が B 型交点となる場合があるが、これらは判定可能である。

表示方法と出力例

編物における立体感を強調して表現するため、求められた交点の連結、座標データから、交点間の線分を円筒形で、その継ぎ目を球体の一部で表してからレンダリングする。

図 10 の JIS 記号配列から得られた編み上がり模様の出力結果を図 15 に示す。

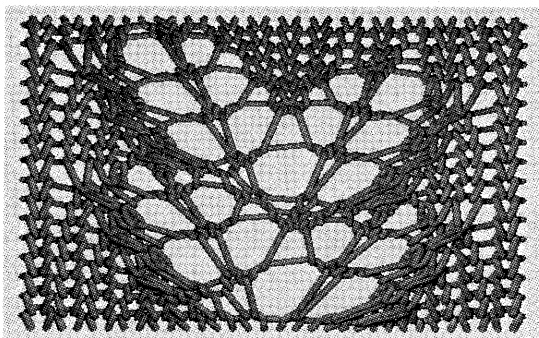
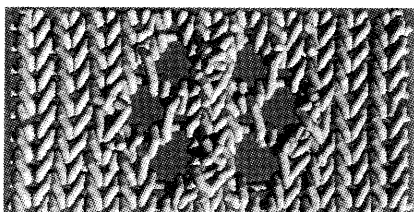


図 15: 出力例

3 毛筆文字生成

ここで提案するシステムの構成を図 16 に示す。本システムは入力部、認識部、画変形部、

かすれ・にじみ生成部、出力部、分類データベース、標準画データベースの 7 つの部分から構成される [2, 1, 11]。かすれ・にじみ生成部については、くりこみ群モデルまたはフラクタルモデルによる処理を行う。入力画の認識に用いる分類データベースには JIS 第 1 第 2 水準の楷書をそのストロークの類似度により分類した 21 グループ・46 種のストロークを蓄積する [2, 1, 11] (表 1 参照)。

システムの基本的処理の流れとしては、ユーザによって電子ペンにより入力されるストローク毎に以下の (1) から (5) までの処理を行い、毛筆画に変換し出力する。システムは、これをストローク毎に繰り返し画面上に重ねて出力することにより毛筆文字を完成する。

1. システム・ウィンドウ上に画面入力された、あるストロークから入力画データ (座標、筆圧、筆速) を得る。
2. 入力画データを基に認識部でストロークの認識を行い、分類データベースから $ID(gr_no, p_no)$ を得る。
3. 画変形部で $ID(gr_no, p_no)$ に対応する出力画データを標準画データベースから検索し、入力画データを基に出力画に変形する。
4. かすれ・にじみ生成部でくりこみ方式またはフラクタル方式を用いてかすれとにじみを生成し、これを出力画に施す。
5. 毛筆出力画を画面上に表示する。

以下、くりこみ群モデルおよびフラクタルモデルによるかすれとにじみの表現方法について述べる。なお、両モデルにより生成した毛筆文字例を図 17 に示す。

3.1 くりこみ群モデル

くりこみ方式では、筆をモデル化し、その筆モデル内部の墨量の変化をくりこみ群を用いて表現する。モデル化された筆から仮定の紙上に墨が染み出すことでかすれとにじみを表現する。くりこみ群 [13] の基本例を図 18 に示す。左側の 4 つの格子点に存在する "o" の数が 3 個以上のとき右側の 1 つの格子点にくりこまれ、2 個以下のときくりこまれない例である。なお、

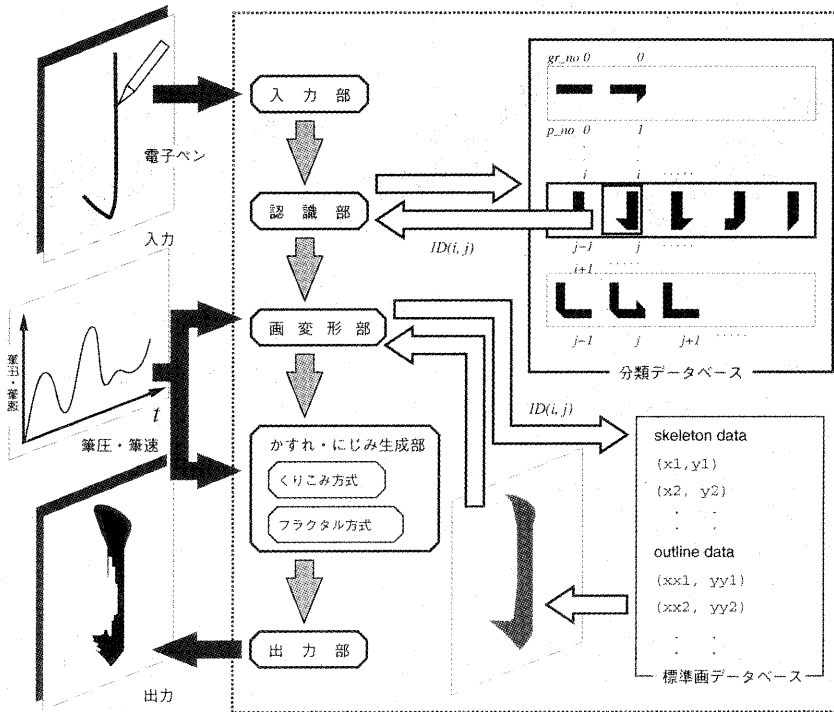


図 16: システム構成

左側の4つの格子点の中央に右側の1つの格子点が配置される。

本システムでは、この概念を拡張しくりこみ変換 [6] f を以下のように定義する。前述と同様、4格子点を1格子点にくりこむ。ここでは、各格子点に重み w_i を付け、4格子点の重み w_i ($i = 1, 2, 3, 4$) からくりこみ後の格子点の重み w_5 を式 (3) で決定する¹。その概念を図19に示す。ここでは、このくりこみ変換 f を筆内で墨粒が染み込むモデルに適用する。

$$\begin{aligned} w_5 &= f(w_1, w_2, w_3, w_4) \\ &= w_1 + w_2 + w_3 + w_4 \end{aligned} \quad (3)$$

筆のモデルを図20に示す。

- 筆を n 層からなる階層構造とする。最下位層を L^1 とし、最下位層から上方に i 番

¹ ここでは、処理の単純化のため変換 f を単に加算とした。

目の層を L^i ($i = 1, 2, \dots, n$) とする (図20 (a))。

- 各層 L^i ($i = 1, 2, \dots, n$) を格子状に分割し、 L^i 上の格子点を n_{jk}^i ($j = 1, \dots, j_{\max}, k = 1, \dots, k_{\max}$) とする (図20 (b))。
- ここでは、最下位層 L^1 は64個 (8×8) の格子点を持つ。また、 L^i 上の縦、横の格子幅は下位層 L^{i+1} 上の縦、横の格子幅のそれぞれ2倍とする (図20 (b))。すなわち、 $j'_{\max} = 2 \times j_{\max}$, $k'_{\max} = 2 \times k_{\max}$ 。
- L^i に存在する各々の格子点 n_{jk}^i の重み w_{jk}^i で墨量を表現する。
- 電子ペンの移動位置をここで述べた多層からなる格子点筆モデルが移動する。

(1) ストロークを部分区間に分割 (2) かすれ線の描画

(3) フラクタル処理

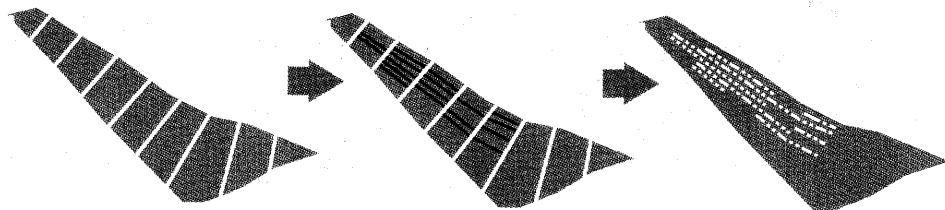


図 21: フラクタル処理によるかすれ表現

3.2 フラクタルモデル

フラクタル方式では、ストローク上にかすれ線と呼ぶ線を複数本描き、そのかすれ線にフラクタル計算法 [6, 9] を用いて白点をプロットし、それを描画していくことで擬似的にかすれを表現する。(図 21 参照)。以下に基本アルゴリズムを示す。

1. 画変形部で出力毛筆画を部分区間に分割する。分割数については出力画の長さによる。
2. 各部分区間毎に線分を複数本描く。この線分をかすれ線分と呼ぶ。各部分区間のかすれ線分数は、筆速、筆圧、墨量からファジィ推論を用いて求められたかすれ度合により決定する。このかすれ線分を連結してかすれ線を形成する。
3. 各かすれ線にフラクタル処理を施し、かすれを生成する。フラクタル処理とは、かすれ線にアフィン変換を用いて白点（ここでは、かすれ点と呼ぶ。）を以下に述べるアルゴリズムによりプロットしていく。なお、かすれ線は図 22 に示すように $\{ \text{点 } P_i(x_i, y_i) : i = n-1, n, \dots, m \}$ を通るものとする。以降、区間 $P_{i-1}P_i$ を第 i 区間と呼ぶ。
 - (a) 第 j 区間上に任意な $P(x, y)$ をとる。
 - (b) かすれ点をプロットする区間 i を選択する。ここで第 i 区間が選択される確率は第 i 区間におけるかすれ度合に依存する。
 - (c) アフィン変換 w_i を用いて第 i 区間上に $P'(x', y')$ を決定し、かすれ点をプロットする。
 - (d) $P'(x', y')$ を新たに $P(x, y)$ とし (b), (c)

を繰り返す (図 22 参照)。

この方式で用いたアフィン変換 $w_i (i = n, n+1, \dots, m)$ は条件式 (4) を満たす。

$$\frac{L_j(x, y)}{L_m} = \frac{l_i(x', y')}{l_i} \quad (4)$$

ただし、

$$\begin{aligned} l_i(x', y') &= \sqrt{(x' - x_{i-1})^2 + (y' - y_{i-1})^2}, \\ l_{n-1} &= 0, \\ l_i &= \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}, \\ L_j(x, y) &= \sum_{k=n-1}^{j-1} l_k + l_j(x, y), \\ L_m &= \sum_{i=n-1}^m l_i. \end{aligned}$$

4 似顔絵の毛髪表現

FACE(Face Caricature Equipment)[5] は、1 枚の入力顔画像をもとに、様々な画像処理手法を用いて顔形状を抽出し、顔パーツ (目、鼻、口、毛髪等) の個人特徴を的確に捉えた似顔絵を自動生成する。とくに、毛髪の表現については似顔絵生成における重要な課題の 1 つとされており、それは FACE においても同様である。[12]

図 23 は FACE における毛髪表現処理部の構成と処理のながれを示すものである。処理のながれとしては、まず FACE の顔パーツ抽出処理によって入力顔画像から毛髪領域を特定する。つぎに、特定された毛髪領域からファジィ・ガウスフィルタリングを用いたエッジ抽出処理

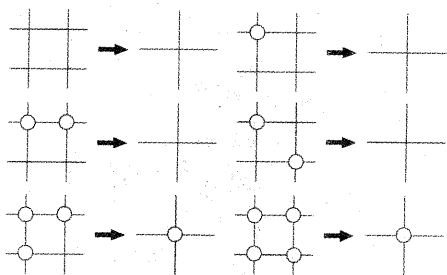


図 18: くりこみ例

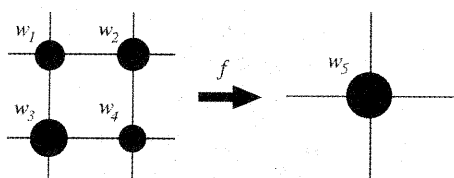


図 19: くりこみ変換 f の概念

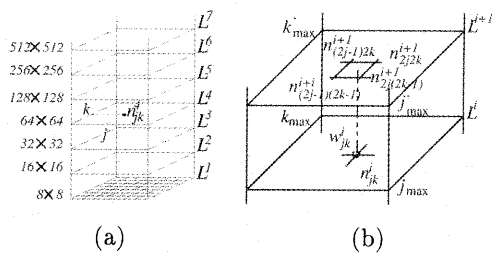


図 20: 筆モデル

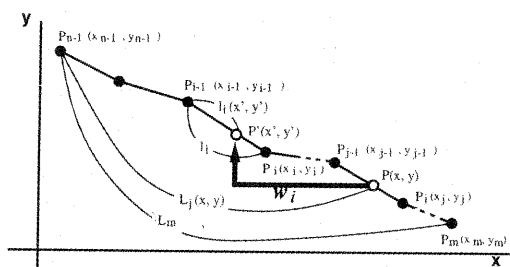


図 22: かすれ線上のアフィン変換

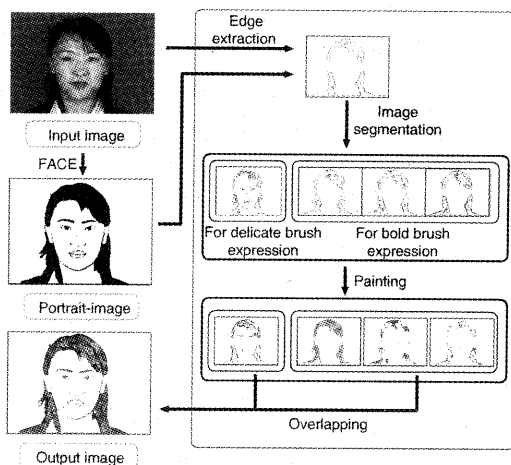


図 23: 毛髪表現処理部の構成と処理の流れ



図 24: 入力顔画像例



図 25: 出力顔画像例

相似理論と繰りこみ群理論を用いた毛筆文字の掠れ生成および滲み処理, および, 似顔絵生成について述べた. 今後は, さらに新しい具体的な処理対象を発見してその処理を研究して行く.

参考文献

- [1] 中村 剛士, 黒田 崇, 伊藤 英則, and 世木 博久. 筆記速度のファジィ評価方法を導入した毛筆文字生成システムについて. 日本ファジィ学会誌, 7(2):371-379, 1995.
- [2] 中村 剛士, 松下 政親, 世木 博久, and 伊藤 英則. フラクタルを用いた毛筆文字のかすれ表現について. 日本ファジィ学会誌, 8(3):558-566, 1996.
- [3] 中村 剛士, 真野 淳治, 世木 博久, and 伊藤 英則. 毛筆フォントの掠れ・滲み処理システムについて. 情報処理学会論文誌, 38(5):1008-1015, 1997.
- [4] 伊藤 裕一郎, 山田 雅之, 世木 博久, and 伊藤 英則. 編物における紐状態表現と編み上がり模様生成の一手法. 情報処理学会論文誌, 39(1):60-69, 1998.
- [5] 大西 孝二, 谷本 尚之, and 西澤 賢一. 似顔絵の自動生成システム “FACE”. *Journal of Technology Transfer*, 23:27-29, 2000.
- [6] 高安 秀樹. フラクタル. 朝倉書店, 1986.
- [7] 山田 雅之, Rahmat Budiart, 伊藤 英則, and 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, 35(3):497-504, 1994.
- [8] 日本工業規格協会. 編目記号. JIS 1 0201, 日本工業規格, 1978.
- [9] M. Barnsley. *Fractals Everywhere*. Academic Press, 1988.
- [10] T. Law, H. Seki, and H. Itoh. Image filtering, edge detection, and edge tracing using fuzzy reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(5):481-491, 1995.
- [11] T. Nakamura, M. Matsushita, H. Itoh, and H. Seki. Fuzzy-based writing system for acquiring good writing skill of brush characters based on the analysis of writing speed. In *The 3rd Pacific Rim International Conference on Artificial Intelligence*, volume 2, pages 822-827, 1994.
- [12] T. Noma, T. Nakamura, H. Itoh, K. Ohnishi, K. Nishizawa, H. Komanome, and L. He. A proposal of image conversion to calligraphic painting style. *Software Engineering, Artificial Intelligence, Networking & Parallel/Distributed Computing*, pages 656-662, 2001.
- [13] K. G. Wilson and J. Kogut. Renormalization group and ϵ -expansion. *Phys. Repts*, 12c(2):75-200, 1974.