

連続 HMM を用いた入力線からの毛筆調レンダリングに関する研究

岡部 雄太[†] 齋藤 豪[†] 中嶋 正之[†]

本稿では、入力となる曲線を自動的に毛筆ストロークとしてレンダリングすることを目指す。レンダリングするために必要なパラメータは多く、それらを入力線から決定する事は単純ではない。そこで本稿では、機械学習の手法を用いてこの問題の解決を試みる。学習に用いる教師データは、実際の毛筆ストロークを撮影し、その動画に対して処理を行なうことにより取得する。その取得したデータを用いて、自然な毛筆ストロークのレンダリングを行うためのモデルを獲得する。本研究で用いる教師データは連続値であり、かつ時系列なデータであることから、学習アルゴリズムとして連続 HMM を選択し、学習させ、毛筆風の描画結果を得た。

A Study of Oriental Brush Rendering using Continuous HMM

YUTA OKABE,[†] SUGURU SAITO[†] and MASAYUKI NAKAJIMA[†]

We aim at carrying out rendering oriental brush strokes to input lines automatically. There are many parameters required for the rendering, and it is not simple to determine them from input lines. To solve this problem, we use a continuous HMM which is one of the machine learning methods. We prepare the captured stroke data as training data, then train a continuous HMM. We can obtain an oriental brush-like painting from input lines by using the trained continuous HMM.

1. はじめに

Non-Photorealistic Rendering における分野として、コンピュータ上での画材の表現が挙げられる。実際の画材の中でも、毛筆は描かれるストロークが変化に富み、非常に興味深い画材のひとつである。しかし、毛筆ストロークの太さの変化は、運筆の曲率、筆の高さ、運筆のスピードなどさまざまな要因が影響し、容易には求められない。

これまでに、様々な仮想 3D 筆モデルが提案されてきた^{(1),(2),(6),(8),(11)}。これらのモデルは対話型のペインティングソフトウェアとして設計されている。ユーザは、ペンタイプディジタイザあるいは触覚型の装置を用いて、3~6 次元のタイプの時系列データの入力し、仮想の毛筆を操作する。このようなモデル化する方法は、ユーザにとって直感的でわかりやすい入力手段を提供できる。しかしながら、ユーザは特別な入力装置操作と、これらを操作する技術を持っていることが必要である。

描画結果の観点から考えると、筆のシミュレーションプロセスよりむしろ、ストロークにおける毛筆のフットプリントの変化が重要である。ここで、フットプリントの変化は、毛筆の房の物理的な特徴および紙と房の間の摩擦によって拘束される状態遷移の一種と考えることができる。

ストロークの軌跡変化に対するフットプリントの変化は単純ではない。しかし、フットプリントの回転や

形状の変化は二つの拘束条件がある。一つは毛筆の物理的な拘束であり、もう一つは高さ、傾きおよびひねりを決定付ける描画技法のルールから来るものである。実際に、芸術家は描くストロークに従って、毛筆の微妙な動きをコントロールする。そこで、典型的な毛筆操作にのみ注目すれば、我々は状態遷移システムを以前の状態と現在の軌跡の接線方向の変化にのみ依存すると見なすことができる。この単純化により、様々な曲線に対して毛筆ストロークを生成するアプリケーションを作成することが可能となる。

我々は、これまでにマウスなどで入力した入力線から自動的に毛筆ストロークを生成する手法を提案してきた⁽²⁾。この手法では、音声認識やバイオインフォマティクス⁽³⁾ における時系列データの解析のための有効な機械学習法として知られている Hidden Markov Model(HMM) を用いている。この HMM によってフットプリントの状態遷移をモデル化し、フットプリントと曲線の軌跡を関連付けることによって、毛筆ストロークを作成していた。しかし、学習に用いる教師データは齋藤らの提案した仮想 3D 筆モデル⁽⁸⁾ から取得していたが、この 3D 筆モデルでは筆をねじる動作が入力できないため、多様なストロークを生成することが難しかった。また、このモデルではフットプリントのけさきが固定されているため、けさきの広がるフットプリントを取得することも不可能であった。

そこで、本稿では、実際の毛筆ストロークから教師データを取得する。これによって、これまで取得が難しかったストロークやフットプリントのけさきの広がりを教師データとして取得することが可能となる。また、これまではフットプリントの状態遷移をモデル化するとき、離散 HMM を用いていたが、本研究で

[†] 東京工業大学 大学院 情報理工学研究所
Graduate School of Information Science Engineering,
Tokyo Institute of Technology

用いる教師データは連続値であるため、連続HMM⁷⁾によってモデル化を行う。これにより、学習するときに生じる量子化誤差を無くすることができる。取得した教師データから連続HMMのパラメータを学習させる。この連続HMMを利用することによって、与えられた曲線の軌跡に対してフットプリントを割り当てる。割り当てられたフットプリントを滑らかにつなぐことによって、毛筆ストロークを作成する。

2. 従来研究

対話型のモデルでは、3D 仮想筆モデルが最も一般的である^{1),2),6),8),11)}。これらのモデルでは、紙と毛筆の房の接する部分の計算によって、フットプリントの形を生成する。ユーザから3から6次元の時系列データを得ることによって、自然な房の動きを生成できる。対話型のモデルでは、このような入力は直感的でユーザにわかりやすい。一方、毛筆ストロークを自動的に生成するようなモデルを考えたい場合、このモデルの持っている自由度が大きいため、制御することが非常に困難である。また、対話型モデルであっても、入力が一般的なマウスである場合は、同様に自由度すべての制御をすることが困難である。

3D 仮想筆モデル以外の毛筆ストロークを生成する手法には、2次元平面で計算や処理を行う手法がある。Skeletal Stroke⁵⁾ はあらかじめ決められたテキストチャを曲線に合わせて曲げる手法である。この手法では、生成されるストロークは参照するテキストチャと同程度の毛筆らしさを持っているが、参照するテキストチャが変わらない限り、生成されるストロークの質感が変化しないという問題がある。また、ストローク中の毛筆の毛の影響を表す柔軟なストロークテキストチャを規則に従って生成する手法も存在する^{4),10)}。その規則では、ストロークの背骨線の制御点に蓄えられた情報を用いてテキストチャを生成している。しかし、各制御点に情報を入力するために、時間がかかってしまう問題がある。

上記以外の毛筆ストロークを生成する手法として、3次元ポリゴンデータからある規則に基づいてストロークを作成する手法がある⁹⁾。この手法では入力として与えられた3次元ポリゴンデータに対して、レイトレーシングを行って、各ピクセルの深さや密度および法線ベクトルの方向などを求める。これらの情報からあらかじめ決められたルールに従ってストロークを生成し、水墨画調の描画を行う手法である。しかし、入力が3次元ポリゴンデータであるため、ユーザの意図する水墨画を生成するためには、モデリングの技術が要求される。

3. 連続HMMによるモデル化

Hidden Markov Model(HMM)とは、状態遷移と各状態での観測ベクトルが確率的に選択される有限状態オートマトンである。観測者は観測ベクトル列は観測可能であるが、状態列は観測不能である。観測ベクトル列から一意的に状態列を復元することができないため、“Hidden”とよばれる。

HMMは、観測ベクトルの型によって離散HMMと連続HMMにわけることができる。我々は連続値の観測ベクトルを扱うため、連続HMMを用いた。連続

HMMにおける状態*i*からの観測ベクトルの出力確率密度関数 b_i は次のように定義される。

$$b_i(\mathbf{o}) = \sum_{j=1}^M c_{ij} \mathcal{N}[\mathbf{o}, \boldsymbol{\mu}_{ij}, \mathbf{U}_{ij}], \quad 1 \leq i \leq N(1)$$

ただし、 \mathbf{o} は観測ベクトルであり、 c_{ij} は状態*i*における*j*番目の重積の係数であり、 \mathcal{N} は混合確率分布の基底分布である正規分布関数を表し、状態*i*の*j*番目の基底分布は平均ベクトル $\boldsymbol{\mu}_{ij}$ と共分散行列 \mathbf{U}_{ij} によって一意に定まる。

混合係数 c_{ij} は正の定数であり、条件 $\sum_{j=1}^M c_{ij} = 1 (1 \leq i \leq N)$ を満たす。それゆえに、確率密度関数 b_i は正数であり、次の条件を満たす。

$$\int_{-\infty}^{\infty} b_i(\mathbf{o}) d\mathbf{o} = 1, \quad 1 \leq i \leq N \quad (2)$$

本稿では、連続HMMによって曲線の接線方向の変化とフットプリントの形状を関連付ける。HMMの状態*i*から状態*j*への状態遷移確率 a_{ij} と状態*i*からの出力ベクトルの確率分布 b_i は毛筆のストロークデータによって学習される。入力された曲線の時刻*t*における接線方向の変化量 $\mathbf{o}(t)$ に対して、最尤状態遷移系列 S は次のように推定される。

$$S = \{k(t) : \arg \max_{k(1), \dots, k(n)} \prod_{t=1}^n a_{k(t)k(t+1)} b_{k(t)}(\mathbf{o}(t))\} \quad (3)$$

ただし、 $k(t)$ は時刻*t*における状態を表す。各状態ではフットプリントの形状の情報を計算によって求めることができるため、最尤状態遷移系列 S からフットプリントの形状を決定することができる。本手法全体の流れを図1に示す。

3.1 ストロークタイプごとのHMM

一つのHMMで曲線全体をモデル化する場合、様々なタイプの曲線をモデル化するためには、多くの状態が必要になる。HMMの中に多くの状態が存在するとさらに多くの状態遷移が起こるため、十分な精度の状態遷移確率を計算するためには、非常に長い計算時間が必要となり、さらには計算が収束しないという問題が生じる。

このような学習上の問題のために、曲線を直線と右回りと左回りの3種類に分類し、その曲線ごとにHMMを作成した。それゆえに、各HMMの状態数は少なく抑えられている。詳細に関しては3.3で述べる。

曲線の小区間ごとにすべてのHMMによって最尤状態系列を推定する。その中で最も確率の高い状態系列を選択する。詳細に関しては3.4で述べる。

3.2 ストロークデータの取得

HMMの教師データに用いるフットプリント系列は実際の毛筆の軌跡から取得する。以下では、実際の毛筆の軌跡から教師データを作成する過程を述べる。

ガラスの上に紙を置き、膠を混ぜた水を使用して、毛筆で直線と右回り曲線と左回り曲線の3種類の様々なストロークをその紙に描く。膠は墨特有の粘りを表現するために使用した。図2に示すように、3種類の曲線ごとにそれぞれ2種類のストロークを作成した。図2の(a),(b),(c)と(d),(e),(f)では筆の進行方向に対する房の向きが異なるため、太さの異なるスト

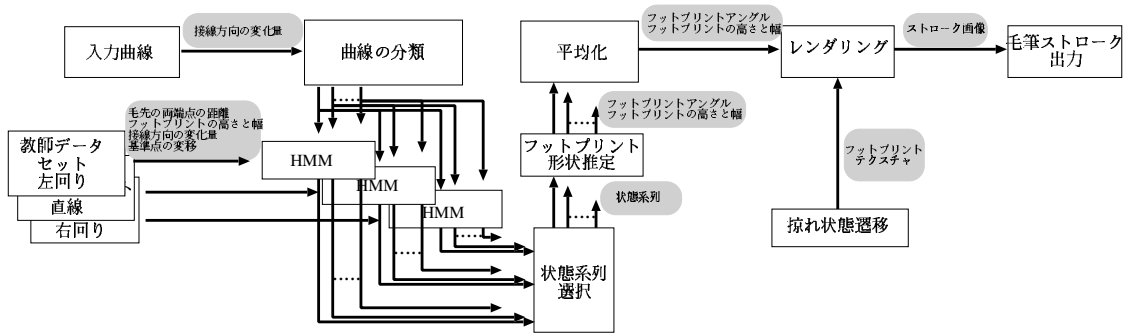


図 1 本手法全体の流れ

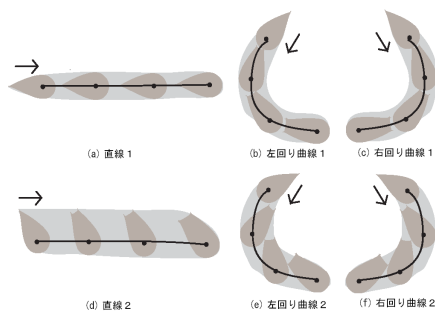


図 2 入力する毛筆ストローク

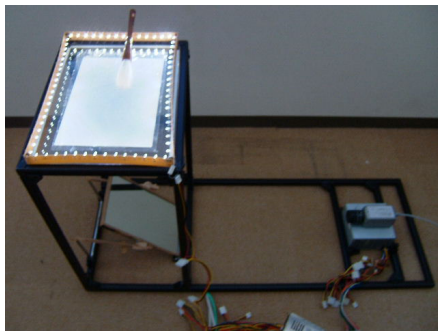


図 3 実験装置

ロークが描かれる。(d)のような描き方は、紙と房との摩擦に逆らって描いているため、齋藤らの3D筆モデル⁸⁾では表現できなかったストロークである。その様子をガラスの下から CCD カメラを用いて1秒間に15フレームずつ撮影する。そのときの毛筆の影を毛筆のフットプリントと見なす。精度良く影を抽出するために、ガラスの辺上に白色 LED を一定間隔で設置して、毛筆に向けて照射した。そのときに用いた実験装置を図 3 に示す。

撮影した画像からフットプリントを抽出するために、背景差分法を適用し毛筆の影の部分を取り出す。切り出した様子を図 4 に示す。切り出したフットプリントから様々な特徴点を測定し、曲線の種類ごとに HMM の教師データを作成する。

まず、得られたフットプリントの主軸を求める。求

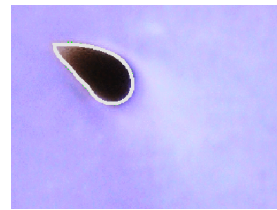


図 4 背景差分法の適用例

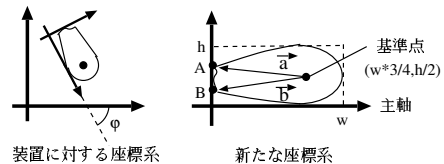


図 5 データの測定

めた主軸及びそれに直交する軸を新たな座標軸として、フットプリントを図 5 のように配置する。時刻 t におけるフットプリントの基準点を図 5 のように設定し、フットプリントの高さ h と幅 w を求め、それぞれ $h(t)$ と $w(t)$ で表す。

また、時刻 t における毛先の両端点 $A(t)$ と $B(t)$ の座標を測定し、その 2 点間の距離を $d(t)$ で表す。フットプリントの輪郭の曲率が一定以上であり、かつ $A(t-1)$ 、 $B(t-1)$ に近い点の中から、高さ方向が最大の点と最小の点を選び、その座標をそれぞれ点 $A(t)$ 、 $B(t)$ として自動的に定める。そのような候補点が求められない場合は手動で点を定める。さらに、時刻 t における主軸の水平方向からの傾きを $\varphi(t)$ とし、時刻 $t-1$ の基準点から時刻 t の基準点へのベクトルを $k(t)$ で表す。また、新しい座標軸上での $k(t)$ を $k'(t)$ で表す。つまり、 $k'(t)$ は $k(t)$ を $-\varphi(t)$ 回転させたベクトルである。また、時刻 t でのストロークの接線方向の変化量 $\psi(t)$ を $k(t)$ と $k(t+1)$ のなす角を計算することにより求める。

各フレームから、フットプリント特徴ベクトルとして次のように定義されるベクトル $v(t)$ を求める。

$$v(t) = (d(t), h(t), w(t), \psi(t), k'(t)) \quad (4)$$

このベクトル $v(t)$ を撮影したすべてのストロークから計測し、ストロークの種類ごとに用意する。次に

—XS!—このベクトル列を使用して、曲線の種類ごとに HMM を学習させる。

3.3 学習過程

前節で作成したストロークデータを使用して HMM のパラメータを学習させる。すなわち、前節のベクトル $v(t)$ を曲線の種類ごとに HMM の観測ベクトルとして割り当てる。割り当てられた観測ベクトル列に対して、Baum Welch アルゴリズム⁷⁾ を適用することによって、曲線の種類ごとに HMM のパラメータを学習させる。すなわち、状態の遷移確率や各状態からの出力確率の確率分布のパラメータ、つまり混合係数 $\{c_{ij} | 1 \leq i \leq N, 1 \leq j \leq M\}$ と平均ベクトル $\{\mu_{ij} | 1 \leq i \leq N, 1 \leq j \leq M\}$ と共分散行列 $\{U_{ij} | 1 \leq i \leq N, 1 \leq j \leq M\}$ が学習によって決定する。ただし、 N は HMM の状態数であり、 M は出力確率分布の混合数を表す。

3.4 HMM の使用方法

本稿では、すべての曲線は 3 種類の曲線から構成されていることを仮定している。それゆえに、入力線から小区間を取り出し、その小区間の曲線に対して、Viterbi アルゴリズム⁷⁾ を 3 種類の HMM に適用して、最尤状態系列をそれぞれ導く。その中で最も確率の高い状態系列がその小区間に対する最も尤もらしい状態系列であると見なし、その状態系列からフットプリント列を定める。このような操作を小区間の開始点をサンプリングポイントを 1 点ずつずらしていきながら行い、入力線全体に対してフットプリント列を決定する。

以下にその手順を示す。

- (1) 入力線を等間隔でサンプリングし、その点を $D = \{d_1 \cdots d_n\}$ で表す。その点での接線の傾きの変化を計算し、 $\Theta = \{\theta_1 \cdots \theta_n\}$ で表す。
- (2) $j = 1$ に初期化し、小区間の点列の数を定数 C に代入する。ただし、 $n < C$ の場合は、 $C = n$ とする。
- (3) 小区間での接線の傾きの変化量 $\zeta_j = \{\theta_j, \theta_{j+1}, \dots, \theta_{j+C-1}\}$ を定める。
- (4) ζ_j に対して、Viterbi アルゴリズムを適用し、3 種類の HMM 各々についての最も確からしい状態遷移系列を求める。このとき ζ_j は 1 次元の数値である。一方、観測ベクトルの出力確率分布の入力は 3 次元である。そのため、3.3 節で作成した出力確率分布を正射影することによって、接線の傾きの変化に対する 1 次元の出力確率分布を作成し、それを Viterbi アルゴリズムで用いた。Viterbi アルゴリズムによって求められた 3 つの最尤状態系列の中で、最も確率の高い状態系列を S_j で表す。
- (5) 状態列 S_j に対するフットプリントの横幅や高さ及び基準点の変位を点群 D に記録する。フットプリントの情報は状態が l の場合、 $\sum_{m=1}^M c_{lm} \mu_{lm}$ によって計算する。
- (6) $n \leq j + C - 1$ の場合、ステップ 7 を行う。それ以外はステップ 3 を再び実行する。
- (7) 各 d_i に対して、複数のフットプリントの情報が割り当てられている場合は、それらを平均する。

3.5 曲線の分類結果

HMM で学習された結果の信頼性を確かめるために、入力線の各サンプリング点での曲線の分類結果を示す。HMM の状態数 N を 5 とし、確率密度関数の混合

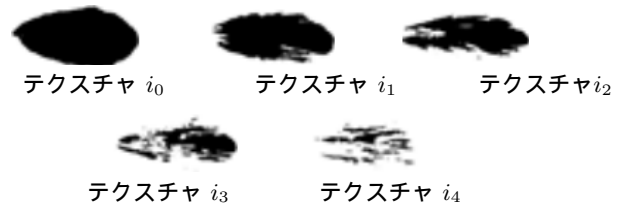


図 7 フットプリントテクスチャ

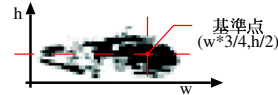


図 8 フットプリントテクスチャの基準点

数 M が 4 であるとき、Baum Welch アルゴリズムを用いて、直線と右回り曲線と左回り曲線の 3 種類の HMM を作成した。マウスで引いた曲線を 8 点ずつの小区間に分割した。各小区間に対して、Viterbi アルゴリズムを用いて、小区間の接線の傾きの変化が各 HMM に受理される確率を計算し、最も確率の高い結果をその小区間の分類結果とする。図 6 は入力された曲線の分類結果を注釈で示している。注釈から後の点はすべて同じ曲線として分類されていることを示している。図 6 の (a) から (c) は一つの曲線で表されており、(d) と (e) は複数の曲線が組み合わさっている。どちらの場合も正確に曲線の分類が行われていて、後者の場合は曲線のつなぎ目の部分が感知できていることがわかる。これより、HMM は正しく学習されていることがわかる。

4. レンダリング

HMM によって導かれた最も確率の高いフットプリント系列を用いて、毛筆ストロークをレンダリングする手法について述べる。

4.1 フットプリントテクスチャ

掠れは毛筆表現をする上で非常に重要な特徴の一つである。それを表現するために、掠れ方の異なるフットプリントのテクスチャ $I = \{i_0 \cdots i_m\}$ を用意した。用意したフットプリント画像は、図 7 に示す。テクスチャはすべて同じ大きさであり、基準点も $(\frac{3}{4}w, \frac{1}{2}h)$ に固定されている。その様子を図 8 に示す。

4.2 テクスチャの遷移

各点での適切な掠れテクスチャを選択するために、筆の掠れ方を表現する遷移規則を作成する。遷移規則は図 9 に示す。毛筆に含まれる墨汁の量は紙に描けば描くほど減っていくため、一定のサンプリング点を過ぎたときに、テクスチャが掠れる方向に遷移するような規則になっている。墨汁が少なくなった場合、墨汁は紙に吸収されないため、墨汁が減ることはほとんど無い。その現象を表現するために、墨汁が少なくなったとき、一定の確率でテクスチャを掠れる方向やその逆方向に遷移させる。

4.3 描画手順

遷移規則によって定まったフットプリントのテクスチャは、入力ストロークの各サンプリング点に 3.4 で述べた手法で、割り当てられたフットプリントの情報

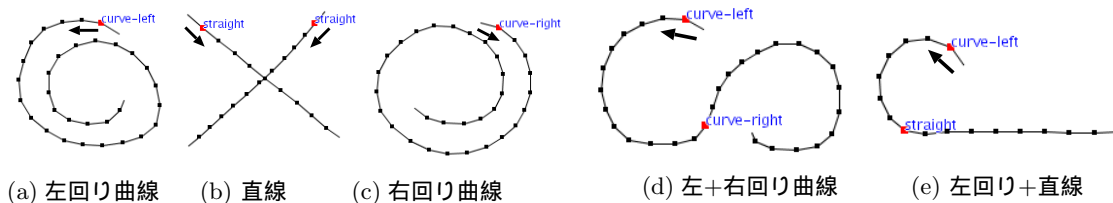


図 6 入力線の分類結果

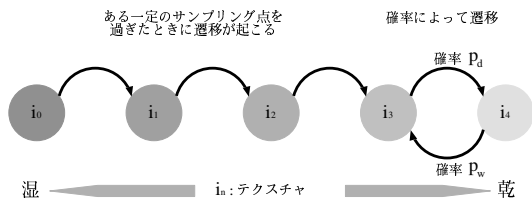


図 9 フットプリントテクスチャの遷移規則

にあわせてアフィン変換が成される．そして，入力線のサンプリング点にテクスチャの基準点を合わせて配置する．さらに，配置されたテクスチャ上の各対応点を B スプライン曲線を用いて滑らかに結ぶ．B スプライン上の α 値は次の手順で決定する．

テクスチャ上を結ぶすべての B スプラインに対して，次の操作を行う．

- (1) フットプリントの初期テクスチャを i_c と i_n に代入する．ただし， i_c は現在のサンプリング点におけるテクスチャを表しており， i_n は次のサンプリング点におけるテクスチャを表している．
- (2) サンプリング点に対するカウンター j を 0 に初期化する．
- (3) α 値に $255 - m$ を代入する．ただし， m は i_c が i_n の対応点でのピクセル値を表している．選択するピクセル値はランダムで選択する．そして，その対応点を結ぶ B スプライン曲線を代入された α 値で描画する．
- (4) $i_c \neq i_n$ ならば， i_n を i_c に代入する．
- (5) j がサンプリング点の終点に達していなければ， j を 1 増加させる．達していた場合，一つの曲線に対する描画操作は終了．
- (6) 節 4.2 で述べたテクスチャの遷移操作を行い，結果のテクスチャを i_n に代入する．
- (7) ステップ 3 へ．

5. 結果画像

結果画像は次のようになる．入力線はすべてマウスで描かれたものである．ユーザはストロークを書き始めるときに，どのタイミングで毛筆を墨汁に浸すのか選択することができる様に設計した．

図 10 に示す入力線から毛筆ストロークを作成した結果を図 11, 12 に示す．結果画像からわかるように，毛筆ストロークの太さの変化や掠れなどが表現できていることがわかる．また，掠れている部分では，テクスチャが複雑にねじれていることがわかる．これはフットプリントの情報に従ってテクスチャがスケールと回転をしていて，かつそれらを B スプライン曲線で描いているためである．また，図 11, 12 の画

像は同じ入力線から作られたにもかかわらず，趣の異なる画像になっている．これは，連続 HMM の学習に用いるデータセットを変えたため，ストロークの描き方が異なっているからである．図 11 には教師データとして図 2 における (a),(b),(c) を用いて，図 11 には教師データとして (a),(e),(f) を用いた．その他の結果画像を図 13, 14 に示した．図 13 には教師データとして図 2 における (a),(e),(f) を用いて，図 14 には教師データとして (b),(c),(d) を用いた．

6. 結 論

本稿では，実際の毛筆ストロークから取得したデータから連続 HMM のパラメータを学習させた．その HMM を用いることにより，入力線に対して，実際のデータに基づいた毛筆ストロークを生成することができた．さらに，描き方の異なるストロークを用いることによって，趣の異なる毛筆ストロークを生成できることを示した．

本手法では，実際の毛筆のストロークから学習データを取得している．そのため，ユーザの毛筆の描き方と物理的な特徴を反映した毛筆ストロークを再現することができる．また，モデルが与えられていれば，マウスで描いたような曲線以外の一般的な曲線に対して本手法を適用することができる．

今後の課題としては，にじみ等の毛筆特有の表現を実装し，毛筆としての表現力を高めていきたい．現在は曲線を直線と右回り曲線と左回り曲線に分類して学習していたが，角などの曲率が急激に変化するストロークはこの分類だけでは表現できないため，学習させるストロークの種類を増やしていく必要がある．

参 考 文 献

- 1) Bill Baxter, Vincent Scheib, and Ming C. Lin. Dab: interactive haptic painting with 3d virtual brushes. In *SIGGRAPH2001 conference proceedings*, pp. 461-468. ACM, ACM Press, 2001.
- 2) Nelson S.-H. Chu and Chiew-Lan Tai. An efficient brush model for physically-based 3d painting. In *Proceedings of Pacific Graphics 2002*, pp. 403-421, 2002.
- 3) Richard Durbin, Sean R. Eddy, Andrew Krogh, and Greame Mitchison. *Biological sequence analysis*. Cambridge University, 1998.
- 4) Qunghian Guo and Tosiyasu L. Kunii. "ni-jimi" rendering algorithm for creating quality



図 10 図 11,12 の入力線



図 11 結果画像



図 12 結果画像

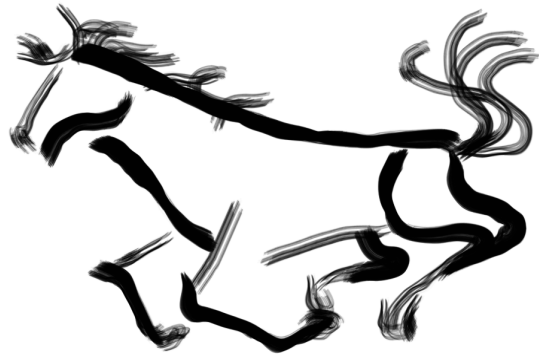


図 13 “馬”の描画結果



図 14 “蛙”の描画結果

- black ink paintings. In *proceedings Computer Graphics International 2003*, pp. 152–159. The Society for Art and Science, IEEE press, 2003.
- 5) Siu Chi Hsu and Irene H. H. Lee. Drawing and animation using skeletal strokes. In *SIGGRAPH 94 Conference Proceedings*, pp. 109–118. ACM, ACM Press, 1994.
 - 6) Jintae Lee. Simulating oriental black-ink painting. *IEEE Computer Graphics and Applications*, Vol. 19, No. 3, pp. 74–81, 1999.
 - 7) Lawrence Rabiner and Biing-Hwang Juang. *Fundamentals of Speech Recognition*. PTR Prentice-Hall, 1993.

- 8) Suguru Saito and Masayuki Nakajima. 3d physics-based brush model for painting. In *ACM SIGGRAPH 99 Conference abstracts and applications*, p. 226. ACM, ACM Press, 1999.
- 9) Youetsu Sato, Tadahiro Fujimoto, Kazunobu Muraoka, and Norishige Chiba. Suibokuga-like rendering for three-dimensional geometric models. In *NICOGRAPH International 2004 Conference Proceedings*, pp. 107–112. The Society for Art and Science, 2004.
- 10) Steve Strassmann. Hairy brushes. *ACM SIGGRAPH Computer Graphics*, Vol. 20, No. 4, pp. 225–232, 1986.
- 11) Songhua Xu, Francis C.M. Lau, Feng Tang, and Yunhe Pan. Advanced design for a realistic virtual brush. *Computer Graphics Forum*, Vol. 22, No. 3, pp. 533–542, 2003.
- 12) 前田大介, 齋藤豪, 中嶋正之. 線の毛筆調レンダリング. 情報処理学会 第 65 回全国大会, pp. 207–208. 情報処理学会, 2003.