

参照画像を用いたブラシストロークの対話的生成

安東 遼一[†] 鶴野 玲 治^{††}(正会員)[†]九州大学大学院芸術工学府^{††}九州大学大学院芸術工学研究院

あらまし 本研究では、事前に準備した参照画像を用いてブラシストロークを対話的に生成する手法を述べる。本手法では、参照される複数のブラシストロークは自動的に小さな断片画像に分解され、特徴量と共に記録される。生成されるブラシストロークは、これら断片画像を1次元のテクスチャ合成で組み合わせて生成される。提案手法は、既存の物理シミュレーションやピクセル単位の画像合成手法と比べ、計算コストが少なく、様々な筆を再現可能である。更に、流体力学を用いた顔料の移流や、クベルカ-ムンクモデルに基づいた顔料の混合色を計算することで、写実的な画像も生成可能であることを示す。評価実験では、油絵具や水彩画、木炭画、パステル画材が再現され、有効性が示されている。また、口紅や指など画材以外の媒体へも適応可能であることが確認できた。

キーワード：ノンフォトリアリスティック・レンダリング、ブラシストローク、テクスチャ合成

Summary This paper describes an interactive technique for generating brush strokes used in exemplar images. The proposed method decomposes several exemplar images of brush strokes into small fragments that are stitched together through one-dimensional texture synthesis in order to generate the final brush stroke silhouettes. In contrast to previous methods in which the strokes are generated by per-pixel computation or using physically based simulation, our method is not only less computationally expensive but also capable of simulating various kinds of brushes. In addition, we introduce a fluid-dynamics-driven post-process and a Kubelka-Munk synthesis method to make the paint strokes appear more real. Our evaluation experiments show that our method can handle various media such as oil paints, watercolors, charcoal, and pastels. We also obtained interesting results with lipsticks and fingers.

Key words: Non-photorealistic rendering (NPR), brush stroke, texture synthesis

1. はじめに

近年、デジタル画材の需要が高まっている。デジタル画材はアナログ画材とくらべて準備の手間が少なく、やり直しが可能であり、費用を安く抑えられる利点がある。商用ソフトウェアの Adobe[®]Photoshop^①や Corel[®]Painter^②はアナログ画材を再現する機能があり、実際のアナログ画との見極めが難しいほど写実的な表現が可能である。これらのソフトウェアには主にテクスチャベースの手法が採用されている。テクスチャベ-

スの手法では、筆先とキャンバスが接する接触面の画像があらかじめ定義されており、この接触画像を連続的に描くことによって筆の軌跡を再現する。しかし、従来のテクスチャベースの手法は事前に定義された種類の筆しか再現できない。ノンフォトリアリスティックレンダリングの研究分野では、物理ベースのシミュレーションを用いた画材の再現が活発に行われている。^③物理ベースの手法では、画材は流体と筆の弾性体の相互作用をえて複雑な模様を描き出せるが、計算コストが高く一つの手法で多数の画材を扱うのは難しい。

コンピュータグラフィックスではテクスチャ合成を利用した画像処理の研究が進められている。テクスチャ合成は画像の欠損修復や領域拡大をはじめ、超解像度処理や画像中の被写体の変形が可能である。^④本手法ではパッ

“An interactive brush stroke synthesis using exemplar images” by Ryoichi ANDO, (Graduate School, Kyushu University), and Reiji TSURUNO, (Member) (Faculty of Design, Kyushu University).

チベースのテクスチャ合成を転用し、参照画像からブラシストローク生成を対話的に行う手法を提案する。本提案手法のテクスチャ合成で生成されるブラシストロークは、木炭やパステルのような水分の少ない画材では良好な画像が生成できる。しかし、水彩画や水墨画のような水の動きに左右されやすい画材では、違和感のある画像が生成される。本手法では、このような画材には、後処理として低解像度の流体力学を施し、半物理的に水の様子を補完する。また顔料の混色計算には、クベルカ - ムンクモデルを用いて正確な混合色を計算する。提案手法の有効性は、多様な画材の再現実験や計算時間、生成画像の写実度より示される。本論文では以下の4件の技術貢献を述べる。

- ブラシストローク領域の自動抽出と芯線抽出手法
 ブラシストロークから実際の筆領域を自動抽出するための二値化処理と、制限付きドロネー三角分割を用いて抽出領域から芯線を自動的に抽出する(図 1-1)。
- テクスチャ合成を用いたブラシストローク生成
 パッチベースの高速な1次元テクスチャ合成を用いてストロークを生成する。本手法のテクスチャ合成は $O(\log n)$ 時間で動作し、パッチの縫い目は Image Quilting¹⁵⁾を用いて目立たなく連結される(図 1-3)。
- クベルカ - ムンクモデルによる色彩合成
 複数のブラシストロークの混合色計算を、クベルカ - ムンクモデルを用いて計算する(図 1-4)。
- 低解像度流体計算を用いた水の流動表現手法
 水彩画や水墨画に観察される水の動きを、低解像度スタッガード格子上数値流体計算を用いて補完する。

2. 関連研究

ノンフォトリアリスティックレンダリング分野では、事前に用意した画像を参照しブラシストロークを生成する方法が多数提案されている。Hsu らは筆画像を任意の曲線に対応させて変形する *Skeletal Stroke* 手法を提案した⁵⁾ *Skeletal Stroke* はカリグラフィーなどのなめらかな画像の変形に適しているが、ラスタライズされた画像では、細部が不自然に引き延ばされてしまう問題がある。Ritter らは、ピクセルベースのテクスチャ合成を用いて筆画像を合成した⁶⁾彼らは Hertzmann⁷⁾らの *Texture-by-Numbers* を改良し、筆とキャンパスの境界の特徴を捉えるエネルギー関数を導入し多様な筆画像を生成した。しかし、この手法は高解像度の筆画像の生成に数秒から

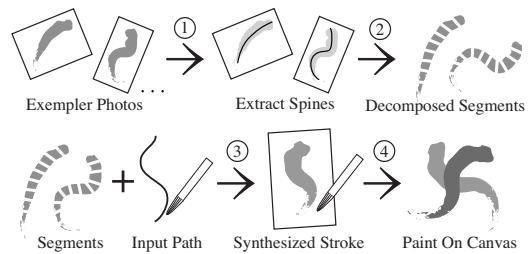


図1 本手法の基本的な流れ
 Fig. 1 Fundamental workflow of our method

数分の計算時間がかかり、対話的な操作に向いていない。テクスチャ合成には GPU を用いた実時間計算手法が存在するか⁸⁾、直接筆画像の合成に転用できない。Bleser らは木炭表面のテクスチャを事前に定義し、これらをキャンパスに描き木炭画を再現した⁹⁾ Ning らは、墨絵ブラシストロークを与えられた輪郭の形状に沿って生成する手法を提案した¹⁸⁾この手法では、入力パスに沿ってテクスチャを配置する手法と比べ、容易にユーザーの意図した形状のストロークを生成可能である。しかし、墨絵以外の画材への適用は示されていない。岡部らは、筆とキャンパスの接触面テクスチャを、入力パスに合わせて変形し、キャンパスに転写する手法を提案した¹⁹⁾この手法では、テクスチャ変形の予測に離散隠れマルコフモデルが用いられ、以前の時刻の変形状態から新たな変形が計算された。Northrup らはユーザーの筆の軌跡から法線方向に伸びる線に沿って三角ポリゴンを形成し、そのポリゴンにテクスチャマッピングを施し水墨画を表現した¹⁰⁾これらブラシの接触面をキャンパスに転写する手法は計算コストが低く写実的な表現が可能だが、事前に定義される筆のテクスチャは人間の手によって注意深く設計される必要がある。Kim らは筆画像の断片から特徴量を計算し、特徴の類似をグラフで記録した¹¹⁾彼らは、このグラフをもとに特徴量の最適な組み合わせを求め筆画像を生成した。本手法のテクスチャ合成はこの手法に置き換え可能だが、事前に特徴グラフを構築する必要がある。

3. 提案手法の概要

最初に、与えられた筆画像のブラシストロークの形状を把握する。これは、参照画像内のブラシストローク領域から芯線を抽出することによって行われる(図 1-1)。ストロークの特徴量計算は、芯線に沿って細断されたセグメント毎に計算される(図 1-2)。各セグメントには隣り合うセグメント上での色情報、曲率、ストローク末端からの距離、切断面の辺の長さを多次元ベクトルとして記録する。ストロークの合成は、ユーザーから与えられた軌跡情報に沿ってセグメントを連結することにより実現

される (図 1-3) . このとき、セグメントは自身の持つ特徴が合成後のストロークの特徴となるべく一致するように配置される . 配置後に発生するセグメント間のつなぎ目は、Image Quilting¹⁵⁾を用いて目立たなく結合される . 水彩画や水墨画の場合は、低解像度流体シミュレーションによって颜料を移流させ、水に流れる効果を演出する . ストローク同士がお互いに重なる場合、クベルカ - ムンクモデルによって混合色を計算する (図 1-4) .

4. 領域抽出と芯線計算

本章では、参照ブラシストローク画像から実際のブラシストロークの領域を抽出するための二値化処理と、ストロークを細かな四角形セグメントに分解するために必要な芯線を計算する手法を述べる .

4.1 二値化

画像の主な自動二値化手法に、P タイル法と判別分析二値化法がある . P タイル法は画素全体の二値化で取り出したい面積のおよその比が分かっている時に有効なアルゴリズムであるが、ブラシストローク面積は筆の種類に依存するので使用できない . 判別分析二値化法は2つのクラス間分散とクラス内分散の比が最大になるように定める . しかし、この手法はキャンパス付近の高い輝度を閾値とすることが多く、安定したストローク領域が抽出できなかった . 実験的に、筆者らはヒストグラム上の2つのピーク点の、高いほうのピーク点のふもとに閾値を与えると、比較的安定した領域が抽出可能なことが分かった . この経験をもとに、閾値 t は次式で算出する

$$t = \arg \max_x \|\mathbf{v}(x) \times \mathbf{v}(\arg \max_x p(x))\| \quad (1)$$

ただし、それぞれ \times は外積、 $0.5 < x < \arg \max_x p(x)$ 、 $p(x)$ は輝度値 x の画素の出現確率で $\sum p(x) = 1$ である . $\mathbf{v}(x)$ はヒストグラム平面中心値 $(0.5, p(0.5))$ から、 $(x, p(x))$ を指す2次元ベクトル関数で、

$$\mathbf{v}(x) = \begin{bmatrix} x - 0.5 \\ p(x) - p(0.5) \end{bmatrix} \quad (2)$$

で与えられる . 本処理は、幾何学的には、ヒストグラムのピーク点とヒストグラム平面中心値 $(0.5, p(0.5))$ 、ふもとの点の3点で構成される三角形の面積が最大になるよ

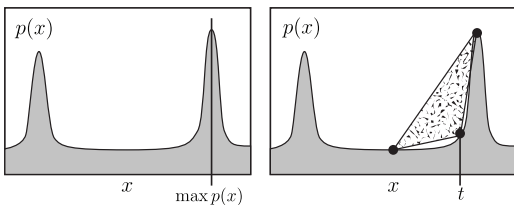


図2 閾値の決定手法

Fig. 2 Determination of binarization threshold

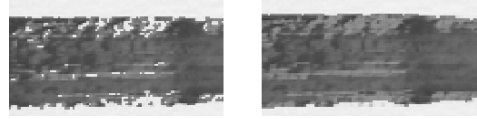


図3 ガウスぼかしてノイズ除去された領域

Fig. 3 Noise removal of domain using gaussian filter



図4 自動分解されたストロークのセグメント

Fig. 4 Automatically decomposed segments

うに、閾値 t を決定していると解釈できる (図 2) . さらに、二値化処理で取り出されたされたストローク領域からノイズを取り除くために、ガウスぼかしを加える . 最終的に取り出される領域は、ぼけた画像を再度適当な閾値 (ex. $t = 0.95$) で二値化したものを使用する (図 3) .

4.2 芯線抽出

ブラシストロークの芯線を抽出するために、制限付きドロネー三角分割 (CDT) を二値化領域に適応する . 制限付きドロネー三角分割は、制限領域内で、三角形ネットワークを、与えられた頂点列で構成する手法である !²⁾ 本手法では、CDT を二値化領域の輪郭を任意の離散間隔で離散化したものに適応する (図 4c) . 芯線は CDT で三角分割された三角メッシュの中心点を繋いで抽出する . 芯線は領域の形状によって分岐することがあるが、芯線は必ず1本である必要がある . 最終的な芯線は、多数生成された芯線の末端の中から2点を、三角形の中心点を繋いだネットワークに沿ってつなげたものを採用する . 末端の2点は、それぞれのネットワーク上の距離が一番遠い2組を選ぶ . このように抽出した芯線は小刻みに震えているので、各頂点位置は近傍の頂点位置の平均をとり平滑化する . このように抽出された芯線の頂点列は、始点から任意の長さでの位置、曲率が計算しやすいよう、2次スプライン曲線で補間する .

5. セグメント分解と特徴量計算

本章では、参照ブラシストローク画像をセグメント単位に分解し、その特徴ベクトルを計算する手法を述べる . 参照画像内のブラシストロークは、芯線とブラシストローク領域を用いて、四角形のセグメントに分解される . 分解は、芯線を一定間隔で再離散化した点から法線上に伸びる切り込みをいれ、この切り込みに沿って切断して行われる . 図 4 に、本手法で分解したセグメント (b) と、参照ブラシストローク (a) を示す . 各分解されたセグメントには、特徴量を事前計算する . 特徴の種類は厳密に

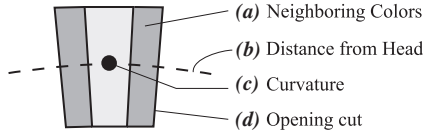


図 5 セグメントで計算される特徴の種類
Fig. 5 Feature selection of a segment

決定されないが、本手法では以下の種類の特徴量を選択した。(a) 前後の隣り合うセグメントの色情報 (b) ブラシストロークの両端からセグメント中心までの距離 (c) セグメント中心点でのパスの曲率 (d) セグメント両端の切断面の辺の長さ。図 5 で、これらの特徴を図示する。色情報 (a) は、任意の離散間隔でセグメント内をサンプリングして取得する。本手法では、縦横 5×10 のサンプルを使用した。PCA 等次元圧縮を施し少ない次元で記録することも可能である。あるセグメント i の色情報を C_i とすると、セグメント i は $c_{\text{head}} = C_{i-1}$ と $c_{\text{tail}} = C_{i+1}$ を特徴として記録する。両端からの距離 (b) は、セグメントの先端からの中心までの距離 d_{head} と終端からの距離 d_{tail} を実数で記録する。曲率 (c) は、セグメント中心で 2 次スプライン曲線を二階微分した値 κ を記録する。しかし、曲率は曲率半径によって大きく変化するので、実際には指数関数で減衰させたもの $\kappa_* = \text{sgn}(\kappa)(1 - \exp(-|\kappa|))$ を利用する。セグメントの切断面の辺の長さ (d) は、前後のセグメントの切断面の辺の長さ $w_{\text{head}}, w_{\text{tail}}$ を実数で記録する。これらを用いて、セグメント i の特徴ベクトル \mathbf{x}_i は、次で表記される $\mathbf{x}_i = \begin{bmatrix} c_{\text{head}}^T & c_{\text{tail}}^T & d_{\text{head}} & d_{\text{tail}} & \kappa_* & w_{\text{head}} & w_{\text{tail}} \end{bmatrix}^T$ 。また異なる成分の特徴が均一に評価されるように、(a-d) はそれぞれ特徴の種類ごとに正規化しておく。具体的には、 $\sum_i (\|c_{\text{head}}\|^2 + \|c_{\text{tail}}\|^2) = 1$, $\sum_i (d_{\text{head}}^2 + d_{\text{tail}}^2) = 1$, $\sum_i (|\kappa_*|^2) = 1$, $\sum_i (w_{\text{head}}^2 + w_{\text{tail}}^2) = 1$ となるように、適当な係数で乗算しておく。よって $\sum_i \|\mathbf{x}_i\|^2 = 4$, \mathbf{x}_i の次元数は $2 \times (5 \times 10) + 5 = 105$ である。

6. 1 次元テクスチャ合成

本章では、これらの事前計算された情報を用いて、新しいブラシストローク画像を対話的に生成する手順を述べる。新しいブラシストローク画像は、前章で分解されたセグメントを、ユーザーの入力の軌跡に合わせて再結合して生成する。(図 6) 今、結合後並べられたセグメントの番号配列 $X \in \mathbb{N}$ の、 i 番目のセグメントの特徴ベクトルが $\mathbf{x}_{X(i)}$ と表される時、配列 X を次式で求めたい

$$X = \arg \min_X \sum_{i=1}^n E(\mathbf{x}_{X(i)}, \mathbf{x}_{X(i-1)}, \mathbf{x}_{X(i+1)}) \quad (3)$$

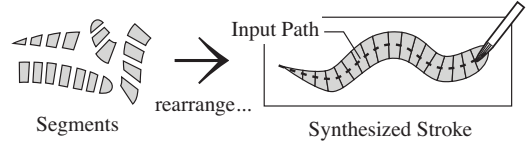


図 6 テクスチャ合成の流れ
Fig. 6 Basic flow of texture synthesis

ただし、それぞれ n は X の格納個数、 $E(\cdot, \cdot, \cdot)$ はセグメントの不自然さを表す視覚コスト関数である。具体的には、セグメントに事前に与えられた特徴ベクトルと結合後の新しい特徴ベクトルの距離を比較する関数である。ただし $\mathbf{x}_{X(i)}$ が $i < 1, n < i$ の時はコストをゼロとする。詳細な数式は次章で述べる。視覚コスト関数は、静的項と動的項で分けられ、次式で表される

$$E(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) = E_s(\mathbf{x}_i) + \sum_{n \in (j,k)} E_d(\mathbf{x}_i, \mathbf{x}_n) \quad (4)$$

ここで、それぞれ $E_s(\cdot)$ は静的項、 $E_d(\cdot, \cdot)$ は動的項である。静的項では特徴のノルムは曲率項と、ブラシストロークのパスの末端からの距離の特徴差のみに影響し、前後のセグメントの特徴ベクトルの影響を受けない。動的項は、前後のセグメントの特徴ベクトルの影響を受ける。今、動的項の表記 $\sum_{j \in (j,k)} E_d(\mathbf{x}_i, \mathbf{x}_n)$ を簡潔に $E_d(\mathbf{x}_i)$ と表すと、 X を求める最小化問題は次式の動的計画法で $O(n^2)$ 時間に解けることが知られている¹³⁾

$$M_i(\mathbf{x}_i) = E_s(\mathbf{x}_i) + \min_{\mathbf{x}_{i-1}} \{E_d(\mathbf{x}_i) + M_{i-1}(\mathbf{x}_{i-1})\} \quad (5)$$

ただし、 M_i は 1 から i までの視覚コスト関数の総和の最小値で、 $M_1(\mathbf{x}_1) = E_s(\mathbf{x}_1)$ 。しかし、 $O(n^2)$ は、本問題を対話的に解くにはコストが高い。そこで本手法では静的特徴の影響が動的要素と比べ、比較的大きいと考える。つまり、特徴ベクトルの分布は不規則に分散しているのではなく、ある程度静的要素に拘束されて集合していると考えられる。すると、最小化問題は次式で近似できる

$$M_i(\mathbf{x}_i) \approx E_s(\mathbf{x}_i) + E_d(\mathbf{x}_i) + \min_{\mathbf{x}_{i-1}} M_{i-1}(\mathbf{x}_{i-1}) \quad (6)$$

式 (5) と比較すると、本手法は最適な組み合わせ X は計算できないが、局所的に最適な X は計算できる。また、特徴の最近傍探索に ANN¹⁴⁾ を用いることで、 $O(\log n)$ 時間で解くことができる。実際には、式 (6) を何度か評価して最適な X の配列を更新していく。最適な特徴を持つセグメントを高速に検索するために、各セグメント i はそれ以外のセグメント j に対し、 $\|\mathbf{x}_i - \mathbf{x}_j\| < \alpha$ を満たすか調べ、これを配列 J_i に記録しておく。ただし、 α は任意の閾値である。 X_i のセグメント検索は、 $J_{X(i-1)}$ と $J_{X(i+1)}$ 内で行われる。テクスチャ合成分野において、配列 J は大域的にジャンプマップと呼ばれ、実時間でテクスチャ生成を行う手法の一つとして知られている²¹⁾式

(6) のテクスチャ合成の計算の流れを Algorithm 1 に示す．最初のループでは初期値となるセグメント列 X を求め，2 回目以降は既に計算された X_{i-1}, X_{i+1} を用いて視覚コスト関数が小さくなるように X_i を更新する．

Algorithm 1 SYNTHESIZE_STROKE(n)

```

1:  $X \leftarrow$  Allocate Blank Segment Array Size  $n$ 
2: // First Loop
3: for all  $i \in X$  do
4:    $X_i \leftarrow \arg \min_j E(\mathbf{x}_j, \mathbf{x}_{X(i-1)}, \mathbf{0})$ 
5: end for
6: // Refinement Loop
7: for several times do
8:   for all  $i \in X$  do
9:      $X_i \leftarrow \arg \min_j E(\mathbf{x}_j, \mathbf{x}_{X(i-1)}, \mathbf{x}_{X(i+1)})$ 
10:  end for
11: end for
12: return  $X$ 
    
```

7. 視覚コスト関数

本章では視覚コスト関数 E の詳細を述べる．視覚コスト関数 E は，結合されたセグメントの局所位置 i が自然に見えるとき 0 に近づき，不自然に見えるとき大きな実数を取る．本手法では次式のように決定した

$$E(\mathbf{x}_i, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}) = E_s(\mathbf{x}_i) + E_d(\mathbf{x}_i, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}) \quad (7)$$

$$E_s(\mathbf{x}_i) = G(p_{\text{head}}, \sigma)(p_{\text{head}} - d_{\text{head}}^i)^2 + G(p_{\text{tail}}, \sigma)(p_{\text{tail}} - d_{\text{tail}}^i)^2 + (p_{\text{curvature}} - r_{\text{sk}}^i)^2 \quad (8)$$

$$E_d(\mathbf{x}_i, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}) = \|c_{\text{head}}^i - C_{i-1}\|^2 + \|c_{\text{tail}}^i - C_{i+1}\|^2 + \quad (9)$$

$$(w_{\text{head}}^i - w_{\text{tail}}^{i-1})^2 + (w_{\text{tail}}^i - w_{\text{head}}^{i+1})^2 \quad (10)$$

ただし， p_{head} ， p_{tail} ， $p_{\text{curvature}}$ はそれぞれ，セグメント点での，ユーザーの入力バスの先頭からの距離，末端からの距離，曲率である． $G(x, \sigma)$ はガウス関数で， σ は 1 ~ 2cm 前後の定数である．特徴量右肩の添え字 i は特徴ベクトル \mathbf{x}_i の要素を示す．静的項では，式 (8) は入力バスのセグメント点での特徴と，セグメントの持つ特徴のノルムを取っている．式 (8) の第一項，二項では，末端からの距離に応じてガウス関数に沿ってノルムの影響を減少させている．これは，ストローク中央付近のセグメントは末端からの距離の影響を受けにくいと考えられるからである．しかし，これら項は式を非線形にするので，ANN による近傍探索ができない．そこで，両末端からの距離が 3σ 以下のセグメントは，全探索を行い，中心部では式 (8) の一項 ~ 二項 ≈ 0 と仮定して ANN 近傍探索を行って高速化する．動的項では，式 (9) の第一項，

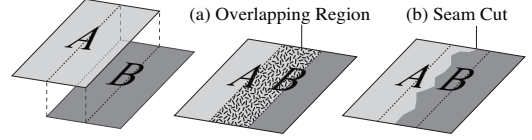


図7 Image Quilting を用いたセグメントの接合

Fig. 7 Seam cut using Image Quilting

二項では隣接するセグメントの色情報の変化のノルムを計算している．第三項では隣接するセグメントの断面の辺の長さの違いを計算している．これら動的項のコストが小さいとき，視覚的にセグメントはなめらかに並べられる．視覚コスト関数は決定的でない．例えば，色情報の値に 1 より大きな定数をかけて，よりなめらかな合成を行ったり，筆の傾きや筆圧情報を加えてペンタブレットを活用することもできる．

8. セグメントのなめらかな接合

単純に配置されたセグメントの境界には直線の不自然な接合線ができる．本章では Image Quilting¹⁵⁾を用いてセグメント間の境界をなめらかに接合する手法を述べる．本手法では，隣り合ったセグメント領域を離散間隔の半分ほど拡張し，意図的に重なり部分を生成する．(図 7a) セグメントが図のように重なっている場合，切れ目 P は，重複面を上から下に，(図 7b) 次式が満たされるよう走らせる．

$$P = \arg \min_P \sum_i \{A(P_i) - B(P_i)\}^2 \quad (11)$$

ここで，それぞれ P_i は重複面での画素位置ベクトル $P_i = (i, j)$ ， A, B は重なってるセグメントの色情報である．これを満たす P は動的計画法を用いて計算できる

$$E(P_i) = \{A(P_i) - B(P_i)\}^2 + \min E(P_{i-1}) \quad (12)$$

ただし， $\|P_i - P_{i-1}\| < \epsilon$ ． ϵ は小さな距離で，1 ~ 3mm である．切れ目を高解像度画像で計算するとコストが高いので， A, B はそれぞれ 10×20 解像度にダウンサンプルされた．

9. クベルカ-ムンクモデルを用いた混合色計算

本章では，Curtis ら¹⁶⁾のクベルカ-ムンクモデル (KM モデル) を用いて，複数のブラシストロークの混合色を求める手法を述べる．KM モデルを用いると，アルファ合成や乗算合成と比較して，現実に近い顔料の混合色を再現できる．KM モデルでは，各顔料層には分光透過率，分光反射率が与えられる．これらは，RGB 成分毎の分光吸収係数 $K = (K_r, K_g, K_b)$ と，分光散乱係数 $S = (S_r, S_g, S_b)$ を用いて計算する．下地が黒色のキャンバスに描かれたブラシストロークの RGB 色を R^b ，下

地が白色のものを R^w としたとき、成分 K, S は次式で求める

$$S_i = \frac{1}{b} \cdot \coth^{-1} \left(\frac{b^2 - (a - R_i^w)(a - 1)}{b(1 - R_i^w)} \right) \quad (13)$$

$$K_i = S_i(a - 1), b = \sqrt{a^2 - 1} \quad (14)$$

$$a = \frac{1}{2} \left(R_i^w + \frac{R_i^b - R_i^w + 1}{R_i^b} \right) \quad (15)$$

ただし、 R^b, R^w はユーザーの入力によって与えられる値である。合成したブラシストロークのRGB成分毎の光透過率 $T \in \mathbb{R}^3$ 、光反射率 $R \in \mathbb{R}^3$ は次式で求める

$$R_i = \sinh bS_i x / c, \quad T_i = b/c \quad (16)$$

$$c = a \sinh bS_i x + b \cosh bS_i x \quad (17)$$

$$x = \max(0, 1 - \frac{1}{3t}(R_r^w + R_g^w + R_b^w)) \quad (18)$$

ただし、それぞれ x は顔料の厚みで、 t は式(1)で求めた閾値である。本手法では、顔料の厚みにブラシストローク画像の濃度を用いた。合成ブラシストロークをキャンバスに転写するとき、キャンバスの分光透過率 T_c 、分光反射率 R_c は次式を用いて更新される

$$R_c \leftarrow R + \frac{T^2 R_c}{1 - R R_c}, \quad T_c \leftarrow \frac{T T_c}{1 - R R_c} \quad (19)$$

ただし、添え字 i は省略されている。レンダリングには、キャンバスの分光反射率 R_c を使用する。

10. 水の流れの表現

本章では、水彩画や水墨画の水の流動を低解像度の格子法数値流体計算を用いて表現する手法を述べる。本論文では、Baxterら¹⁷⁾の手法を用いて、2次元キャンバスで生成される流体速度場を計算した。格子は、約5mm×5mm間隔のスタッガード格子を用いた。水の流れるは、次式非圧縮ストークス方程式で記述される

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla) \mathbf{u} - \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f} \quad (20)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (21)$$

ただし、 $\mathbf{u}, p, \nu, \mathbf{f}$ はそれぞれ、流速、圧力、拡散係数、外力である。外力は、筆先とキャンバスの接触面での流速を \mathbf{u} に代入して再現する。低解像度格子は数値拡散を伴うので、拡散項 $\nu \nabla^2 \mathbf{u}$ は無視される。流速は、移流、非圧縮投影の2段階で計算される。ある地点 p の時間 t の流速を $\mathbf{u}_t(p)$ としたとき、移流後の流速 $\mathbf{u}_*(p)$ は1次精度セミ・ラグランジュ法で求める

$$\mathbf{u}_*(p) = \mathbf{u}_t(p - \Delta t \mathbf{u}_t(p)) \quad (22)$$

右辺は格子間を線形補間あるいは高次スプライン補間して求める。流速の粘性は補間精度で調整を行う。次ステップの流速場 $\mathbf{u}_{t+\Delta t}$ は、移流後の流速場から圧力勾配を引いて求める

$$\mathbf{u}_{t+\Delta t} = \mathbf{u}_* - \nabla p \quad (23)$$

圧力場は次式のポアソン方程式にて算出する

$$\nabla^2 p = \nabla \cdot \mathbf{u}_* \quad (24)$$

ただし、水と空気の境界は $p = 0$ のディリクレ境界条件を用いる。水と空気の境界は、Volume-of-fluid法(VOF法)を用いて追跡する。キャンバスには、水の含有確率分布 f が定義され、水と空気の境界は $f = 0.5$ で表現される。 f は、ユーザーの筆先とキャンバスの接触点では、 $f = 1$ で、流速に沿って次式のように移流する

$$\frac{\partial f}{\partial t} = -(\mathbf{u} \cdot \nabla) f \quad (25)$$

キャンバスの乾燥を表現するために、 f は、時間が進むにつれ一定の割合で減衰させる。顔料の水の流れは、キャンバス上の分光吸収係数 K 、分光散乱係数 S 、厚み x を式(25)と同様に移流させて表現する。分光反射率 R_c は、移流された K, S 、厚み x から式(14,16-17)を用いて随時更新する。

11. 結 果

提案手法によって生成されたブラシストロークの例を図8に示す。左から、それぞれ指・口紅・水墨である。これらは、それぞれ3つの参照ストローク画像を合成して生成された。提案手法では、図のように指や口紅などの画材以外の画像からも興味深いストロークを生成可能である。参照画像は、解像度がおおよそ800×800前後のものが使用された。2mm間隔でセグメントを離散化した場合、かかった事前計算時間は、二値化から特徴ベクトル計算まで1秒以内だった。なおこれら計算は2.8GHz Intel Quad-Core Xeon 計算機でシングルスレッドで行われた。250個の離散化セグメントを取り出した場合、15cm長までのストロークは50ms以内で生成できた。図9に、配列 X を一度に3回更新した場合のテクスチャ合成の詳細な計算時間を示す。なお、Image Quilting 処理やKMモデルを用いた混合色計算は、ユーザーが筆の軌跡の入力を終えたとき(ex. タブレットのペンを、画面から離れた瞬間)に行う。プレビューは、各セグメントをグラフィックハードウェア上で高速にテクスチャマッピングし

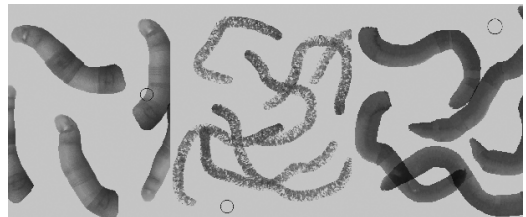


図8 提案手法を用いたブラシストローク生成の例
Fig. 8 Generated brush stroke images

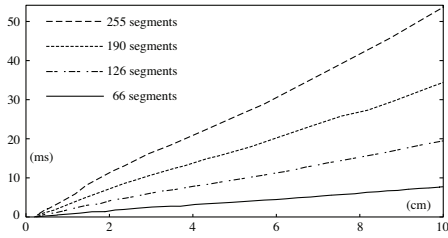


図9 提案手法のテクスチャ合成時間

Fig. 9 Computational timing of texture synthesis



図10 生成されたブラシストローク(右)とその参照ラベル(左)

Fig. 10 Generated brush strokes (right) and its referenced labels (left)

て行う。生成されたブラシストロークの、参照画像のラベル付けをおこなったものを図10に示す。曲率項の特徴のみでストローク合成を行ったものと、全ての特徴を用いてストロークを合成した例を図11に示す。ストローク中に走る複数の細い白い亀裂は Image Quilting で計算された境界線で、もともと隣り合わなかったセグメント同士が隣り合った場合の接合面を示す。セグメントの選択は、それぞれの特徴が複雑に影響し合うので、特徴がストロークの視覚的に優れた合成に具体的にどのように寄与するか測定するのは困難であった。しかし、本実験では、考慮する特徴の種類と数によって、セグメントの並び方が多様に変化することが分かった。例えば、曲率項のみを評価したストローク合成では、前後のセグメントがなめらかに接合しなかった(図11上)。しかし、他の特徴も組み込むと、多くの箇所ではなめらかに接合することが分かった(図11下)。経験的に、動的特徴量を増やすと、セグメントはよりなめらかに接合し、静的特徴量を増やすと、セグメントは様々に変化することが分かった。接合面については、多くのブラシで境界をなめらかに繋ぐことができたが、図10の水墨画のように、不自然さが残ったものもあった。このような不自然さは、参照画像を増やしても取り除くことはできなかった。しかしこのような境界は、ポアソン画像処理²²⁾を施すと、目立ちにくく接合できると考えられる。数値流体力学を用

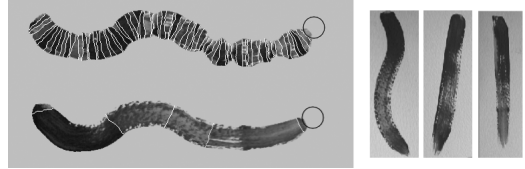


図11 曲率のみで合成した例(上)全ての特徴で合成した例(下)参照画像(右)

Fig. 11 Strokes stitched with curvature (up) and with full spectrum feature (bottom) and its referenced images (right)



図12 数値流体計算による水の表現

Fig. 12 Fluid flow using computational fluid dynamics



図13 本手法で描かれた絵画

Fig. 13 Paintings painted with our method

いた顔料の流れの表現を、図12に示す。流体計算を用いると、顔料のにじみや、かすれの表現が可能である。ただし、流体による顔料の移流は、ユーザーが筆の入力を終わるまで開始できないので、対話性には若干の時差が生じることがある。図13では本手法で描かれた様々な画風の絵画を例示する。これらでは、木炭のきめの細かいタッチや、ぶらしのはけ、水のにじみが表現できた。

12. 議 論

本論文では、ユーザーが画像を与えるだけでブラシストロークを自動的に解析する手法を提案したが、自動的に解析できない事柄が存在する。例えば、参照画像のストロークの形状は、どちらが先頭でどちらが末端であるか自動判別するのは難しい。ただし、これはユーザーにあらかじめ画像のブラシの形状をおおまかに指定することで解決できる。参照の画像が写真撮影された場合、画像にレンズの歪みを含むことがある。この場合、線形な歪みはアフィン変換で取り除くことができるが、魚眼効

果のような非線形歪みを取り除くのは難しい。また、本手法は機械学習ベースの手法なので、生成画像の品質は参照画像に大きく影響する。しかし、画像数が多いとセグメントの数が増え、計算時間も長くなってしまふ。経験的に、参照画像は3~5枚ほどがバランスのよい結果だった。生成ストローク画像には、部分的にセグメントの繰り返しパターンが観測されることがある。この繰り返しは、以前使用したセグメントに別途視覚コストを与えることで抑えられるが、その場合視覚コスト関数全体が非線形になるので、計算時間が増えてしまうことがある。本論文では Curtis らの KM モデルにて混色を計算したが、この手法は濁りが発生することが知られている。この濁りは、KM モデルを直接的に実装することで、軽減可能であることが報告されている²⁰⁾

13. む す び

本論文では、ユーザーに与えられたブラシストロークの参照画像から、任意の形状のブラシストロークを生成する手法を提案した。提案手法では、ブラシストロークの形状は、自動二値化と制約付きドロネー三角分割を用いて抽出された。そして、芯線に沿ってストロークを輪切りに切断し、セグメント単位に分解した。各セグメントには色情報、ストロークの両末端からの距離、曲率、両端の切断面での辺の長さが計算され、特徴量として記録された。ブラシストロークの生成は、これらセグメントを入力パスに沿って自然に見えるよう1次元テクスチャ合成を用いて配置し、Image Quilting を用いてなめらかに接合することにより実現された。またキャンバスへ転写では、ケルカ-ムンクモデルを用いて混合色を計算し、数値流体力学計算を用いて水彩画のような流動を表現した。結果より、提案手法は様々な画材が表現可能であることがわかった。

参 考 文 献

- 1) Adobe Photoshop
http://www.adobe.com/products/photoshop
- 2) Corel Painter - http://www.corel.com/painter
- 3) Bruce Gooch, Amy Gooch: "Non-Photorealistic Rendering", A K Peters/CRC Press 2001. ISBN: 9781568811338.
- 4) Li-Yi Wei, Sylvain Lefebvre, Vivek Kwatra, Greg Turk: "State of the Art in Example-based Texture Synthesis", EUROGRAPHICS 2009 STAR.
- 5) Siu Chi Hsu, Irene H. H. Lee: "Drawing and animation using skeletal strokes." Proceedings of the 21st annual conference on Computer graphics and interactive techniques. 1994. pp109-118.
- 6) Ritter L., Li W., Agrawala M., Curless B., Salesin. D: "Painting with texture." 2006. In 17th Eurographics Symposium on Rendering, Nicosia, Cyprus.
- 7) Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin: "Image analogies."

- In Proceedings of the 28th annual conference on Computer graphics and interactive techniques (SIGGRAPH '01). 2006. ACM, New York, NY, USA, pp327-340.
- 8) Sylvain Lefebvre and Hugues Hoppe: "Parallel controllable texture synthesis." ACM Trans. Graph. 24, 3 (July 2005), pp777-786.
- 9) Teresa W. Bleser, John L. Sibert, and J. Patrick McGee: "Charcoal sketching: returning control to the artist." ACM Trans. Graph. 7, 1 (January 1988), pp76-81.
- 10) J. D. Northrup and Lee Markosian: "Artistic silhouettes: a hybrid approach." In Proceedings of the 1st international symposium on Non-photorealistic animation and rendering (NPAR '00). ACM, New York, NY, USA, pp31-37.
- 11) Kim, M. and Shin, H. J: "An Example-based Approach to Synthesize Artistic Strokes using Graphs." Computer Graphics Forum, 2010. Volume 29, Issue 7. pp2145-2152.
- 12) L. P. Chew. 1987: "Constrained Delaunay triangulations." In Proceedings of the third annual symposium on Computational geometry (SCG '87), D. Soule (Ed.). ACM, New York, NY, USA, pp215-222.
- 13) Jian Sun, Lu Yuan, Jiaya Jia, and Heung-Yeung Shum: "Image completion with structure propagation." ACM Trans. Graph. 24, 3 (July 2005), pp861-868.
- 14) Sunil Arya, David M. Mount, Nathan S. Netanyahu, Ruth Silverman, and Angela Y. Wu: "An optimal algorithm for approximate nearest neighbor searching fixed dimensions." J. ACM 45, 6 (November 1998), pp891-923.
- 15) Alexei A. Efros and William T. Freeman: "Image quilting for texture synthesis and transfer." In Proceedings of the 28th annual conference on Computer graphics and interactive techniques (SIGGRAPH '01). ACM, New York, NY, USA, pp341-346.
- 16) Cassidy J. Curtis, Sean E. Anderson, Joshua E. Seims, Kurt W. Fleischer, and David H. Salesin: "Computer-generated watercolor." In Proceedings of the 24th annual conference on Computer graphics and interactive techniques (SIGGRAPH '97). ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, pp421-430.
- 17) William Baxter, Yuanxin Liu, and Ming C. Lin: "A viscous paint model for interactive applications." Research Articles. Comput. Animat. Virtual Worlds 15, 3-4 (July 2004), pp433-441.
- 18) Xie Ning, Hamid Laga, Suguru Saito and Masayuki Nakajima: "Contour-driven Sumi-e rendering of real photos." Computers & Graphics. Vol.35 No.1 pp.122 - 134 (2011)
- 19) 岡部雄太, 前田 大介, 齋藤豪, 中嶋正之: 「離散 HMM を用いた線の毛筆調レンダリング」電子情報通信学会論文誌 D, 電子情報通信学会, Vol. J90-D, No. 1, pp. 106-114, (2007)
- 20) 齋藤 豪, 中嶋 正之: 「Kubelka-Munk の理論を用いたデジタルペインティングのための絵の具モデル」電子情報通信学会論文誌 D-II Vol. J82-D-II, No. 3, pp.399-406 (1999)
- 21) Steve Zelinka, Michael Garland. "Towards real-time texture synthesis with the jump map." In Proceedings of the 13th Eurographics workshop on Rendering (EGRW '02). Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2002. pp99-104.
- 22) Patrick Pérez, Michel Gangnet, and Andrew Blake. "Poisson image editing." ACM Trans. Graph. 22, 3 (July 2003), pp313-318.

(2011年2月4日受付)

(2011年4月19日再受付)