

領域ベース・ストロークベースのアート風画像生成法

ヘンリー・ジョハン[†] 松井 一^{††} 芳賀 俊之^{†††} 土橋 宜典^{††††}
西田 友是[†]

A Method for Creating Region-Based and Stroke-Based Artistic Images

Henry JOHAN[†], Hajime MATSUI^{††}, Toshiyuki HAGA^{†††}, Yoshinori DOBASHI^{††††}, and Tomoyuki NISHITA[†]

あらまし 本稿は入力された画像に対して様々なアート風画像を生成するノンフォトリアリスティックレンダリング手法を提案する。アート風画像には、モザイクやステンドグラスなどの領域分割ベースのものや油絵などのストロークベースのものがあるが、提案法はその両方を生成することを可能にするものである。アート風画像においては、エッジ(色の境界)が強調されることや、色変化の少ない部分が同じ色で描かれることが一般的である。そこで提案法では、まず入力画像をいくつかの領域の集合として近似する。この領域分割は、領域の境界ができるだけ入力画像のエッジと一致するようにして、各領域は1色で塗りつぶすというのであり、ポロノイ図を利用して実現される。続いて、ポロノイ領域をそのまま利用すれば領域分割ベースの画像、色の近い隣接領域をたどるように筆のストロークを描けばストロークベースの画像を生成することができる。

キーワード ノンフォトリアリスティックレンダリング, 領域ベース画像, ストロークベース画像, ポロノイ図

1. はじめに

近年デジタルカメラなどの普及により、デジタル画像を扱う機会が増大し、画像に様々な効果を加えたいという需要も増してきている。そのため、入力画像を油絵風などのアート風にするという手法もまた、重要性を増してきている。Photoshop [17] など市販のソフトウェアには、フィルタを適用することで画像をアート風にする機能があるが、入力画像のエッジがよく表現されていなかったりノイズがのったような結果しか得られなかったりして、必ずしもユーザの望むものと

はいえない。また従来のアート風画像生成法は1つの画風に特化したものが多く、ユーザが気軽に様々な画風を試すということは難しい。そこで本稿は1枚の入力画像から様々なアート風画像を生成する方法を提案する。

提案法は、アート風画像を、貼り絵など領域ベースのもの、油絵などのストロークベースのもの2つに分類して考え、その両方の生成を可能にする。このとき入力画像から直接アート風画像を生成するのではなく、入力画像の特徴(色変化の大きな部分・色の様な部分)を表現するような中間画像を生成する。この中間画像のレンダリング方法を変えることによって、様々なアート風画像を生成する。

2. 関連研究

アート風画像を生成する手法には、画像を入力とするものおよび3次元モデルを入力とするものがあるが、ここでは提案法と同じく画像を入力としたアート風画像生成手法を述べる。

ポロノイ図を利用する方法として、Hoffら[7]によるモザイクやSeoら[14]による貼り絵といった領域ベースのアート風画像を生成する手法が提案された。

[†] 東京大学大学院新領域創成科学研究科, 千葉県
Graduate School of Frontier Sciences, The University of
Tokyo, 5-1-5 Kashiwanoha, Kashiwa, Chiba, 277-8561
Japan

^{††} 東京大学大学院情報理工学系研究科, 東京都
Graduate School of Information Science and Technology,
The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo,
113-0033 Japan

^{†††} (株)NTT データ, 東京都
NTT DATA Corporation, 3-3-3 Toyosu, Koto-ku, Tokyo,
135-6033 Japan

^{††††} 北海道大学大学院工学研究科, 北海道
Graduate School of Engineering, Hokkaido University,
Kita-ku, Kita 13, Nishi 8, Sapporo, 060-8628 Japan

しかし、これらの手法は入力画像のエッジを考慮していないため、その特徴をよく表現しきれていない。Hausner [4] はエッジを考慮したモザイク生成法を提案したが、ユーザがエッジを表す曲線を指定する必要がある。Kim と Pellacini [8] は、Hausner の手法をもとに、入力画像とタイルの色の差も考慮したモザイク生成法を提案した。DeCarlo [2] は、セグメンテーションにより画像をいくつかの色の様な領域に分けその間のエッジを強調することで、画像を抽象化する手法を提案した。

ストロークベースのアート風画像生成手法も数多く提案されている。Salisbury ら [13] はペン・アンド・インク風画像の生成法を提案した。Curtis [1] は流体のシミュレーションを使った水彩画生成法を提案した。画像の輝度の勾配ベクトルに垂直な方向にストロークを描画する手法は Litwinowicz [9] および Hertzmann [5] によって提案された。Hertzmann [5] は太さの異なる複数の筆で描かれる油絵などのストロークベース画像生成法を提案した。

Haerberli [3] は、ストロークにそって母点を配置することで、領域ベース・ストロークベース両方の画像を生成する手法を提案した。しかし、領域ベースの画像の場合、その領域はランダムに生成される。また、ストロークベース画像の場合、ストロークの位置はユーザが指定する必要がある。白石と山口 [15] は、Haerberli の手法を拡張し、ストロークを自動生成する手法を提案した。しかし、ストロークベース画像の生成を目的としており、領域ベース画像は入力画像のエッジを表現することができていない。

提案法は Haerberli の手法と同様、領域ベース・ストロークベース画像を生成することができる。Haerberli の手法に比べて最も異なるのは、提案法は入力画像を自動的にいくつかの多角形領域で近似し、これを基にして、領域ベースおよびストロークベースのアート風画像を生成する点である。

3. 提案手法の概要

提案法では、まず入力画像を近似するための中間画像を生成する。中間画像は入力画像をいくつかの多角形の領域に分割するもので、各領域は入力画像の色の様な部分を表し、領域の境界が入力画像のエッジを表すようにする。領域ベースのアート風画像生成は、領域をそのまま利用したり、領域を組み合わせるとより大きな領域を利用したりできると表現の幅が広がる。ス

トロークベースの画像生成は、色の近い領域をたどることでストローク生成ができる。そのためには各多角形領域が凸であると領域をたどるのが容易である。このような条件を考慮して、中間画像としてはポロノイ図を利用する。各ポロノイ領域はその母点と同じ位置にある入力画像の画素の色で塗りつぶされる。

中間画像を生成するために、母点を規則的に配置するか、または、入力画像の特徴を考慮して配置し、初期ポロノイ図を生成する (4. 節)。次に母点を移動することで、ポロノイ図の画像と入力画像との画素の差を小さくする (5. 節)。続いて、ポロノイ領域をそのまま利用することで領域ベースの画像 (6. 節)、色の近いポロノイ領域をたどって、ストロークを描くことでストロークベースの画像 (7. 節) を生成する。

4. 母点の初期配置

ユーザは母点を等間隔に配置する方法、あるいは、自動的に入力画像の色変化の大きな部分 (エッジ) に多くの母点を配置する方法を選択できる。

4.1 等間隔の母点配置

母点をハニカム格子状に配置する。ユーザは 2 つのパラメータ (母点間の距離、格子の中心からのずれの大きさ) を指定できる。

4.2 アダプティブな母点配置

エッジは入力画像の特徴を表している可能性が高いので、母点をエッジ付近に多く配置し、色変化の少ない部分にはあまり配置しないようにする。このアダプティブな配置を行うために、各画素 p に対して、次の 3 つの値を計算する。

- *Occupancy*, $Occ(p)$: 画素 p に母点を置けるかどうかを表す。その値は、配置可能または配置不可のいずれかである。初期状態では全画素配置可能である。
- *Importance*, $Imp(p)$: 画素 p がエッジ上に存在するかどうかを表し、画素 p の重要度を表す。各画素における色の勾配ベクトルの大きさを初期化される。色を考慮した勾配ベクトルの計算には Ruzon と Tomasi [12] によって提案された手法を用いる。
- *Distance*, $Dist(p)$: 画素 p から一番近いエッジまでの距離を表す。色の勾配ベクトルの大きさがある閾値より大きな画素をエッジとし、チェスボードディスタンス [11] を利用して計算する。

ユーザは 3 つのパラメータを指定する。 (d_{sites} : 母点間の距離, d_{min} : 母点間の最小距離, t_{imp} : エッジとみなす画素における重要度の閾値)

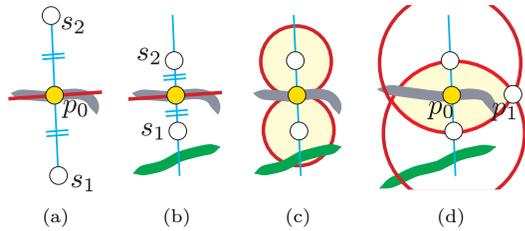


図 1 アダプティブな母点の配置, (a) ボロノイエッジによる入力画像のエッジの近似, (b) 他のエッジを考慮した母点の配置, (c) 近くの画素の状態を配置不可に変更, (d) 近くの画素の重要度を低減

Fig. 1 Adaptive sites placement. (a) Approximate an edge in the input image using a Voronoi edge. (b) Consider other nearby edges when placing sites. (c) Forbid placing sites in nearby pixels. (d) Reduce the importance values of nearby pixels.

まず $Imp(p)$ が一番大きな画素 p_0 を選ぶ。このとき p_0 が入力画像のエッジ上に存在すると考え、勾配方向に等距離に母点 s_1, s_2 を配置すれば、ボロノイ辺で入力画像のエッジを近似することができる (図 1(a)).

基本的に s_1 と s_2 の距離が d_{sites} となるようにするが、図 1(b) のように別のエッジがある場合は、母点をより p_0 の近くに配置する必要がある。

p_0 の勾配方向にあって p_0 からの距離が $d_{sites}/2$ 以下の画素の中で、 $Dist(p)$ が極大になる 2 点を選び、 $Dist(p)$ が小さいほうの点を s_1 、 p_0 から s_1 までの距離を r とする。 p_0 に関して s_1 と点対称な点を s_2 とする。 s_1, s_2 に母点を配置することが可能ならば配置し、そうでなければ r の値を小さくして配置を試みる。

母点が密になりすぎるのを防ぐため、母点 s_1, s_2 を配置した後、 $C(s, h)$ を点 s を中心とする半径 h の円として、 $C(s_1, r)$ と $C(s_2, r)$ が包含する領域に属する画素に母点を配置することを禁止し (図 1(c)), $C(s_1, 2r)$ と $C(s_2, 2r)$ との共通な領域に属する画素の $Imp(p)$ を次式にしたがって減少させる (図 1(d)).

$$Imp(p) = Imp(p) \times \frac{|p - p_0|}{|p_1 - p_0|}. \quad (1)$$

以上を $Imp(p)$ の値が t_{imp} よりも大きい画素がなくなくなるまで繰り返すことでアダプティブな母点配置が行われる。ただし、この配置では色変化の少ない部分に母点を置くことができないので、そのような部分には等間隔の母点配置を行う。

4.3 局所的なパラメータ設定

母点の初期配置は、ユーザが等間隔の配置またはア

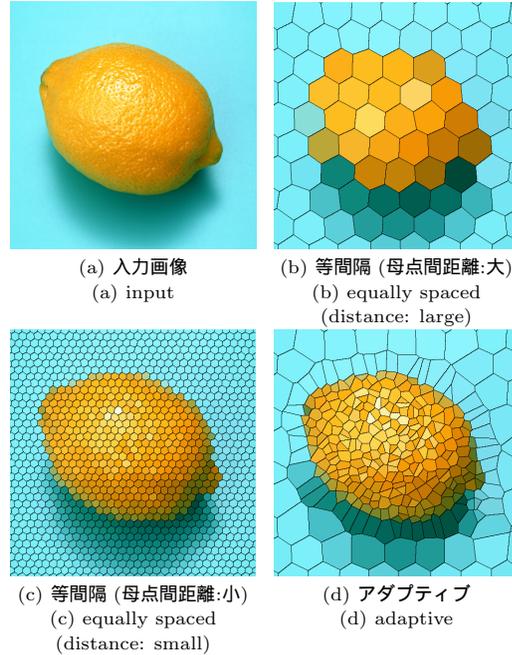


図 2 母点の初期配置
Fig. 2 Initial sites positions.

ダプティブな配置を選択しパラメータを調整することで自動的に行われる。しかし、よりユーザの望む結果に近づけるため、ユーザが範囲を指定してその範囲ごとにパラメータを設定することも可能にした。

4.4 母点配置手法の比較

図 2 に母点の初期配置の例を示す。(b), (c) は等間隔の配置によって生成されている。(b) においてはレモンを認識するのが難しく、(c) においては背景の色の一様な領域まで多くの領域によって表現されている。(d) はアダプティブな配置によって生成されており、エッジ付近は小さな領域で表現され、色の一様な部分は大きな領域で表現されている。これは、色変化の大きな部分は細かい筆で描き、色の一様な部分は大きな筆で塗りつぶすという画風に対応させることができる。

5. ボロノイ図の最適化

ボロノイ図の最適化では、母点を動かすことで入力画像をよりよく近似するようなボロノイ図を生成するものである。

5.1 最適化手法

母点は一度に 1 画素しか動かさないことにして、母点を動かさない場合および隣接する 8 つの画素に動かした場合の 9 つの場合についてそれぞれボロノイ図を

計算し、入力画像とポロノイ図の誤差 E_{color} (式 2) が最も減少する画素に母点を移動する。

$$E_{color} = \sum_x \sum_y color_diff(P_{x,y}^V, P_{x,y}^I). \quad (2)$$

P^V, P^I はそれぞれポロノイ図と入力画像の画素, x, y は画素の座標, $color_diff(p, q)$ は La^*b^* 色空間における画素 p, q の色の差を表している. ポロノイ領域内の画素の色はその母点と同じ場所にある入力画像の画素の色として決定される. ポロノイ領域内の画素の平均色を用いる方法も考えられるが, 次のような不具合が生じる可能性がある. 最適化を行う際, ポロノイ領域が入力画像における色の異なる隣り合う領域をまたぐ可能性があり, このとき平均色を計算したら, 新しい色が得られ, 入力画像に存在しない色の領域が得られてしまう可能性がある. また入力画像に存在しない色が得られた場合, 誤差の計算にも正確さが失われ, その結果エッジをよく近似できない. そのため, 平均色ではなく, 母点と同じ場所にある入力画像の画素の色を使用する. 計算時間を短縮するため, 2つのステップにより最適化を行う.

ステップ 1(5.2 節) では, 母点を移動してもそのポロノイ領域の形は変わらないと仮定して, すべての母点について同時に最適な移動方向を計算することで入力画像の大域的な特徴をとらえる. E_{color} の変化が閾値以下になるまでステップ 1 を繰り返し実行する.

ステップ 2(5.3 節) では, 各母点について順に, それを各方向に動かした場合のポロノイ図を正確に計算することで局所的な調整を行う. 全ての母点が処理され, かつ誤差 E_{color} が最も減少する方向に各母点を動かすので, 母点を処理する順番は重要ではない. ステップ 1 と同様, E_{color} の変化が閾値以下になるまで行う.

最適化を行う際, ポロノイ図を高速に生成するため, Hoff ら [7] が提案したグラフィックスハードウェアを用いたポロノイ図の生成手法を利用する. この手法は母点が頂点となる円錐を描画することによってポロノイ図を生成する.

5.2 ポロノイ図の大域的な最適化

ポロノイ図の最適化による入力画像の近似処理を高速にするために, 近似的な領域形状の計算を行い, すべての母点を一斉に動かすということを繰り返す. そして, ポロノイ図が入力画像の特徴を大まかに表すようにする.

このステップでは, 「ある母点が移動しても(周りの

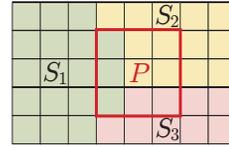


図 3 E_{color} の変化量の計算

Fig. 3 The calculation for the change of E_{color} .

母点の移動は無視して), 対応するポロノイ領域の形状は変化しないで, 母点と同じだけ平行移動する」と仮定する. また, 1 回の母点の移動を隣接する 8 方向の画素のみに限定しているため, E_{color} の変化の計算が低コストで行えるようになる.

これを図 3 を用いて説明する. 図 3 の, P を注目する画素, S_1, S_2, S_3 を P の近傍の母点とする. また, 母点 S_i に対し, 方向 D_j に移動した際の E_{color} の変化量 $dE_{(i,j)}$ を準備する.

P は S_2 の領域に含まれており, また, P の上, 右上, 右の画素も S_2 の領域に含まれる画素である. よって, S_2 が下, 左下, 左に移動する場合には, P は S_2 の領域に含まれたままの画素であり, S_2 がそれ以外の方向に移動する場合には, P は S_2 の領域から外れる画素である. そこで, これらの情報から, S_2 が保持する各方向に対応した dE を更新する. P の下, 右下の画素は, S_3 の領域に含まれる画素である. よって, S_3 が上, 左上に移動する場合には, P は S_3 に対応した領域に新たに含まれる画素である. ゆえに, S_3 の上方向と左上方向の移動に対応した dE の値を更新する.

他の場合についても同様であり, 全画素について, それと隣接する 8 つの画素に対して, 上記の計算を行うと, 結果として, すべての (i, j) について, つまり, 母点 S_i の方向 D_j への移動について, $dE_{(i,j)}$ の値が計算される. そして, 各母点について, dE が最小となる方向に移動する. 得られたポロノイ図の入力画像との誤差 E_{color} を計算し, その変化が閾値以下になるまでこのステップを繰り返す.

5.3 ポロノイ図の局所的な最適化

本節で説明するステップでは, 一度に移動させる母点は一つのみとする. そして, 各母点について, 最適な母点の位置を探索するために, 隣接する 8 画素へ移動した際の色誤差 (E_{color}) の変化量を算出し, 誤差の最も減少する方向へ母点を移動させる. しかし, 移動方向毎にポロノイ図の計算, および, フレームバッファの読み出しを行うと時間を要する. 一般的に, グラフィックスハードウェアを用いた手法では, バッファ

の読み出しが計算時間の点でボトルネックとなり易い。

そこで、フレームバッファの主メモリへの読み出し回数を減らし、かつ、母点に対応した円錐の描画回数を減らせる方法を提案する。ポロノイ画像は、各母点を頂点とする円錐を描画することで得られるが、移動方向毎にすべての母点に対する円錐を描画すると時間を要する。そこで、あらかじめ、移動させる母点を除くすべての母点に対する奥行きを記憶しておき、次に、移動させる母点の円錐を描画すると、移動後のポロノイ図が抽出される。このとき、フレームバッファにおける色の論理演算を利用して、複数の移動方向に対応するポロノイ図を効率的に描画する。

具体的に以下のような方法でポロノイ図を描画する。色の論理演算とは、フレームバッファの画素に値を書き込むとき、画素の元の値と書き込まれた値の間で論理演算を行い、その結果を画素の新しい値とする機能であり、高速に実行される。各母点には一意のIDを与えて、ポロノイ図描画時には、そのIDに対応する円錐の色とする。実験の結果から母点の数が1万個以下であるので、ある母点 s を動かすとき、フレームバッファのRGB 24ビットのうち15ビットだけを使って s 以外の母点のつくるポロノイ図を描画することができる。続いて、論理演算 OR を使用し、デプスバッファへの書き込みを無効にして、残り9ビットの各ビットに s の移動後の各ポロノイ領域を描画する。以上によりカラーバッファだけを使って必要なポロノイ図の情報を得ることができる。

この情報を用いて E_{color} の最も減少する方に母点を動かす。移動により E_{color} が減少する可能性のある母点について行い、 E_{color} の変化が閾値以下になるまで繰り返す。

5.4 計算対象領域の制限

ステップ2では母点を1つずつ動かす際に毎回ポロノイ図を描画する必要があり、特に E_{color} の計算のためにグラフィックスメモリから主メモリにデータを読み込む操作が大きなボトルネックとなっている。しかし移動前後のポロノイ図で変化があるのは、移動する母点の周りの限られた領域だけである。したがって、その領域を含むような矩形領域内にだけポロノイ図を描画し読み込むことで、描画する円錐数およびデータの転送量を減らすことができる。以下、この矩形領域を計算対象領域と呼ぶ(図4(a))。

ある母点 s を動かすと仮定する。図4(b)を参考にし、母点 s しかない場合、そのポロノイ領域は ABCD

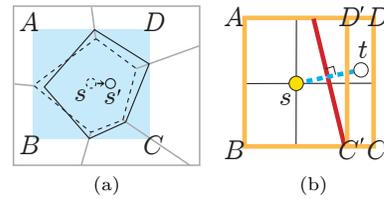


図4 (a) 母点 s を s' に動かしたときの必要な計算対象領域は矩形領域 $ABCD$, (b) 母点 s に関する計算対象領域は、他の母点 t との垂直二等分線により $ABCD$ から $ABC'D'$ に制限可能

Fig. 4 (a) Rectangular area $ABCD$ is the necessary region for computation when site s is moved to s' . (b) Taking into account the perpendicular bisector between sites s and t , the computation region of site s can be restricted from $ABCD$ to $ABC'D'$.

である。しかし、他に母点 t がある場合、母点 s のポロノイ領域は s と t を結ぶ直線(点線)の垂直二等分線(実線)により制限される。図4(b)の場合、母点 s のポロノイ領域は垂直二等分線の左側の領域になる。これを利用して、母点 $s = (s_x, s_y)$ の計算対象領域を次のように求める。(計算対象領域の左下の座標を (x_1, y_1) , 右上の座標を (x_2, y_2) とする。)

まず、計算対象領域を画像全体として初期化する ($x_1 = 1, y_1 = 1, x_2 =$ 画像の幅, $y_2 =$ 画像の高さ)。次に、 s 以外の各母点 $t = (t_x, t_y)$ を利用して逐次的に計算対象領域を狭めていく。 $t_x > s_x + 1$ ならば、 s の移動後の各点について、 t との垂直二等分線を求め、それぞれ直線 $y = y_1$ および $y = y_2$ との交点を計算する。それらの交点の中から一番大きな x 座標 x'_2 を選び、 x_2 を $\min(x_2, x'_2)$ に更新する。 $t_x = s_x, t_x = s_x + 1$ の場合、 s が上か下に、右に1画素移動した際、 t との垂直二等分線は直線 $y = y_1$ および $y = y_2$ と交点を持たないので、 x_2 を更新しない。 $t_x < s_x - 1$ ならば、同様にして x_1 を更新する。 y_1, y_2 に関しても同様に更新する。

6. 領域ベースの画像

領域ベースのアート風画像の生成においては、中間画像内のポロノイ領域をそのまま利用することができる。

6.1 ステンドグラス風

各ポロノイ領域を1枚のガラス部分に対応させて、以下のようにレンダリングを行う。

- ポロノイ辺(境界線)の太さを、それに隣接する2つのポロノイ領域の色の誤差が大きいほど、太くし、

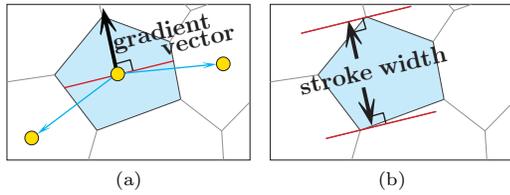


図 5 (a) ストロークをのばす方向, (b) ストロークの幅
Fig. 5 (a) Stroke direction and (b) stroke width.

入力画像中の輪郭や特徴をより強調する.

- 実際のガラスでは歪み等の原因で、部分毎に明るさや透過率の変化がある. そのような材質感を擬似的に表現するため、ポロノイ領域の各頂点で乱数を加味して輝度を計算して塗りつぶす.

6.2 貼り絵風

ひとつのポロノイ領域を 1 枚の紙片と対応させ, Seo [14] の手法に基づいて, 紙をモデリングする. 紙は表と裏の 2 層からなるとみなし, 表はポロノイ領域の色, 裏は白色とする. それぞれの層に独立に中点変位法を適用することで, 破った紙の境界のぎざぎざ感や, 裏の白い部分が見える効果を表現することができる. これに, 紙の材質を表すように Perlin ノイズ [10] を利用して生成したテクスチャを加え描画する.

7. ストロークベースの画像

ストロークベースのアート風画像の生成においては, 中間画像内の色の近いポロノイ領域をたどり, その中心点を制御点とする 3 次 B スプライン曲線に沿ってストロークを描画する. 領域をたどる方向を決めるために, まず領域 v の色を考慮する勾配ベクトル $G(v)$ を, 近傍の領域 u_i を利用して以下のように定義する.

$$G(v) = \sum_{u_i} \frac{|L(u_i) - L(v)|}{|C_{u_i} - C_v|} \cdot \frac{C_{u_i} - C_v}{|C_{u_i} - C_v|}. \quad (3)$$

ただし, $L(w)$ および C_w はそれぞれポロノイ領域 w の色 ($L_a^*b^*$ 色空間) および中心点の位置ベクトルを表す.

ストローク生成に必要な制御点は以下のようにして求める. まず勾配の一番大きな領域を選び, その中心点を制御点とする. 続いて, 選ばれた領域に隣接するポロノイ領域のうち, 勾配ベクトルに垂直な 2 つの方向にある領域を選び, その中心点を新たな制御点とするという操作を繰り返す (図 5(a)). ただし, 最初に選ばれた領域との色差が大きくなる場合, ストロークが長くなりすぎる場合, ストロークが大きく曲がる場合



図 6 油絵のストローク描画用の (a) マスクおよび (b) テクスチャ, (c), (d) 木炭画のためのテクスチャ

Fig. 6 (a) Mask and (b) texture for drawing strokes for oil painting. (c), (d) are textures for charcoal drawing.

は, 中心点を制御点とはしない. これらの条件はユーザ指定のパラメータにより制御される.

最初のストロークを生成した後, まだ処理されていないポロノイ領域のうち, もっとも勾配の大きい領域を選び, その領域の中心点を次のストロークの出発点とする. 以降, ストロークを生成する度に, 上述の方法を用いてストロークの出発点を決定する. 生成されたストロークはスタックに積み, すべての領域が処理されるまでストローク生成を繰り返す. 制御点におけるストロークの幅は, その制御点が属するポロノイ領域の勾配ベクトル方向に測った幅とする (図 5(b)).

ストローク生成の際, 出発点となるポロノイ領域をランダムに選ばないので, 離れ島のような領域 (孤立した領域, すなわち隣接する領域が全て処理済み) が多く発生するという事はない. 少数の離れ島の領域が発生してしまう可能性はあるが, 離れ島の領域が発生しても, その領域はある程度の大きさを持っているので, 生成されるストロークは極端に短くならず, 結果画像には影響しない.

ストロークをすべて求めた後, スタックの上から順にストロークを描画する. 上述のように, 勾配の大きい領域 (エッジ) を通るストロークが先に生成されるので, これらのストロークはスタックの下の方にあり, そのためエッジが後から描画され, 結果画像でエッジを表現することができる.

7.1 油絵風

ストロークの色は, 生成の際最初に選ばれた領域の色とする. Hertzmann [6] の手法を利用して, ストロークを三角形ストリップで近似し, ストロークの形状を表すマスク (図 6(a)) と油絵の表面の凹凸を表すテクスチャ (図 6(b)) を用いて描画する. ただし, 実際の幅で生成されたストロークを描画するだけでは塗り残しが生じる可能性があるため, まず, 2 倍の幅で下絵を描画する. そして, 塗り残しがあるかどうかを調べ, なければ下絵の描画を終了する. 塗り残しがある場合, 幅を 3 倍にして下絵を描画して塗り残しを調べる. こ

表 1 画像をポロノイ図で近似するための計算時間
Table 1 Computation times for approximating images using Voronoi diagrams.

例	サイズ (pixel)	母点数	計算時間 (sec)
図 7	512 × 512	1366	5.6
図 9 上	512 × 512	1015	3.2
図 9 下	640 × 480	679	4.5

のように、ストロークを少しずつ太くして下絵を描画すれば、塗り残しのない結果画像が得られる。また、下絵の上に実際の幅でストロークを描いているので、結果画像には不自然な部分は生じない。

7.2 木炭画風

木炭画は白黒画像であるが、場合によっては木炭をこすることにより様々な濃淡を表現することができる。これを実現するため、ポロノイ領域の輝度に応じてストロークの輝度を決め描画する。このとき図 6(c) および (d) のように輝度および詳細模様の異なる 255 枚のテクスチャを利用する。

8. 結果および考察

図 7 および 9 に提案法を用いて生成された様々なアート風画像を示す。すべての画像について、アダプティブな母点の初期配置を行った (母点間の距離 $d_{sites} = 32$, 母点間の最小距離 $d_{min} = 8$)。入力画像をポロノイ図で近似するのに要した時間を表 1 に示す。ポロノイ図からのアート風画像生成はすべて 1 秒以下で計算できた。CPU は Pentium4 3.0GHz, グラフィックスカードは nVIDIA GeForce FX 5900 Ultra を使用している。

8.1 高速化の効果

ポロノイ図最適化の際に計算対象領域の制限 (5.4 節) を全く行わない場合、図 7 の入力画像をポロノイ図で近似するのに 7 分以上かかってしまうが、計算対象領域の制限を行うことで 5.6 秒で計算できた。計算対象領域の平均的なサイズは 36×36 であった。計算対象領域の制限に大きな効果があることが分かる。

また、ポロノイ図の最適化を 2 段階 (大局 (5.2 節) と局所 (5.3 節) 的な最適化) で行うことによって効率よく入力画像の近似処理ができた。比較のため、最適化の際、局所的な最適化しか行わない場合、図 7 の画像を近似するのに 12.4 秒かかる。実験の結果、2 段階の最適化を行うことによって、2.0 ~ 2.5 倍の高速化を実現した。

8.2 既存の手法およびソフトウェアとの比較

結果画像において、生成されたストロークは最大でも 1100 本程度であった。Hertzmann [5] や白石と山口 [15] の手法において 1 万のオーダーのストロークが生成されているのに比べると、ストローク数は比較的少ないことが分かる。それにもかかわらずよい結果が得られているのは、入力画像から直接ストロークを生成するのではなく、入力画像の特徴をよく表現するような中間画像を利用したため、効率よくストローク生成を行うことができたからであると考えられる。画風に関して、従来手法とは多少異なるが、提案手法のレンダリング結果はストローク描画の際に用いられるマスクおよびテクスチャによって決まる。これらを変更することによって、様々な画風を表現することができる。

アート風画像を生成することのできる既存のソフトウェアはいくつかある。Painter [16] はストロークベースの画風に特化しており、ユーザが画像全体をストロークで塗りつぶす必要があるため、1 枚の入力画像から様々な画風を気軽に生成するという本稿の目的とは対象が異なる。Photoshop [17] や GIMP [18] は入力画像に対してフィルタを適用することでアート風画像を生成できる。

図 8 は Photoshop や GIMP を使って生成されたアート風画像である。入力画像は図 7 と同じである。領域ベースの画像を生成しようとした場合、Photoshop では入力画像のエッジが考慮されていないので、何の画像であるのか認識できなくなってしまうが、GIMP は比較的エッジを表現することができている。また Photoshop, GIMP 共に領域の大きさを局所的に変更することができず、領域が一定サイズのものしか生成できない。提案法では入力画像との色の誤差が小さくなるようにポロノイ図を最適化しているため、入力画像のエッジをよく表現することができ、アダプティブな母点の初期配置により、入力画像の特徴に応じて領域のサイズを自動的に決定することができる。

ストロークベースの画像を生成しようとした場合、Photoshop や GIMP で簡単な操作をいくつか実行しただけでは、ストロークが見えなかったり、ストロークの方向が一定の方向のものだけになってしまったりする。そのため、得られたストロークベースの画像は人間が描いたものとは極めて異なる。提案法ではストロークの方向を勾配を使って決定しているため、より人間が描いたものに近いストロークを生成できている。



入力画像
input



中間画像 (ボロノイ図)
intermediate image (Voronoi diagrams)



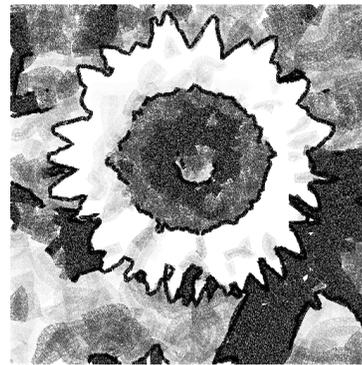
ステンドグラス風
stained glass



貼り絵風
colored paper mosaics



油絵風
oil painting



木炭画風
charcoal drawing

図 7 結果画像 (ひまわり)
Fig.7 Results (sun flower).



ステンドグラス風 (Photoshop)
stained glass



油絵風 (Photoshop)
oil painting



モザイク風 (GIMP)
mosaic



油絵風 (GIMP)
oil painting

図 8 Photoshop, GIMP によるアート風画像
Fig.8 Artistic images of sun flower created using Photoshop and GIMP.



図 9 結果画像 (上:犬, 下:野菜)
Fig. 9 Results (top: dog, bottom: vegetables).

9. ま と め

中間画像としてポロノイ図を用いることにより, 1 枚の入力画像から領域ベースおよびストロークベースのアート風画像を生成可能な方法を提案した. 最初に入力画像に母点を配置してポロノイ図を生成し, 母点を動かすことでポロノイ図が入力画像をよりよく近似するように最適化する. 提案法は入力画像の大局的な特徴をとらえ, そして局所的な特徴をとらえるように 2 段階の最適化を行うことで, 効率よく入力画像を近似できた. 次に, ポロノイ領域をそのまま利用することで領域ベース, 色の近い領域をたどるようにストロークを描画することでストロークベースのアート風画像を生成できる.

入力画像の色変化の大きな部分により多くの母点を配置するというアダプティブな母点の初期配置を導入することで, 入力画像の特徴をよりよく表現することを可能にした. ポロノイ図を最適化の際, 計算対象領域を制限することで, 計算時間の高速化を達成した. 実験の結果, 入力画像の特徴をよく表現したアート風

画像を生成できた. アート風画像は短時間で生成できるため, ユーザは気軽に様々な画風を生成することが可能である.

文 献

- [1] C. Curtis, S. Anderson, J. Seims, K. Fleischer, and D. H. Salesin. Computer-generated watercolor. *Proc. of SIGGRAPH 97*, 421–430, 1997.
- [2] D. DeCarlo and A. Santella. Stylization and abstraction of photographs. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2002)*, **21**(3):769–776, 2002.
- [3] P. E. Haeberli. Paint by numbers: Abstract image representations. *Computer Graphics (Proc. of SIGGRAPH 90)*, **24**(4):207–214, 1990.
- [4] A. Hausner. Simulating decorative mosaics. *Proc. of SIGGRAPH 2001*, 573–580, 2001.
- [5] A. Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. *Proc. of SIGGRAPH 98*, 453–460, 1998.
- [6] A. Hertzmann. Fast paint texture. *Proc. of Non-Photorealistic Animation and Rendering 2002*, 91–96, 2002.
- [7] K. E. Hoff III, T. Culver, J. Keyser, M. Lin, and D. Manocha. Fast computation of generalized voronoi

- diagrams using graphics hardware. *Proc. of SIGGRAPH 99*, 277–286, 1999.
- [8] J. Kim and F. Pellacini. Jigsaw image mosaics. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2002)*, **21**(3):657–664, 2002.
- [9] P. Litwinowicz, Processing images and video for an impressionist effect. *Proc. of SIGGRAPH 97*, 407–414, 1997.
- [10] K. Perlin. Improving noise. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2002)*, **21**(3):681–682, 2002.
- [11] J. Russ. *The Image Processing Handbook*. CRC Press, 1992.
- [12] M. A. Ruzon and C. Tomasi. Color edge detection with the compass operator. *Proc. of IEEE Conference on Computer Vision and Pattern Recognition 99*, 160–166, 1999.
- [13] M. P. Salisbury, S. E. Anderson, R. Barzel, and D. H. Salesin. Interactive pen-and-ink illustration. *Proc. of SIGGRAPH 94*, 101–108, 1994.
- [14] S. Seo, Y. Park, and S. Kim. Colored-paper mosaic rendering. *SIGGRAPH 2001 Technical Sketches*, 157, 2001.
- [15] M. Shiraiishi and Y. Yamaguchi. An algorithm for automatic painterly rendering based on local source image approximation. *Proc. of Non-Photorealistic Animation and Rendering 2000*, 53–58, 2000.
- [16] Corel Corporation. Painter.
- [17] Adobe Systems. Adobe Photoshop.
- [18] GIMP. <http://www.gimp.org/>
(平成 x 年 xx 月 xx 日受付)



ヘンリー・ジョハン

平 11 東大・理卒．平 13 同大学院理学系研究科修士課程了．平 16 同大学院情報理工学系研究科博士課程了．平 16 同大学院新領域創成科学研究科客員共同研究員．主として，コンピュータグラフィックス，特に形状補間，形状モデリング，ノンフォトリアリスティックレンダリングに関する研究に従事．情報処理学会，芸術科学会各会員．



松井 一

平 15 東大・理卒．平 15 同大学院情報理工学系研究科修士課程に在学中．主として，コンピュータグラフィックス，特にノンフォトリアリスティックレンダリングに関する研究に従事．



芳賀 俊之

平 12 東大・理卒．平 14 同大学院理学系研究科修士課程了．同年，NTT データ（株）入社．主として，コンピュータグラフィックス，特にノンフォトリアリスティックレンダリングに関する研究に従事．



土橋 宜典

平 4 広島大・工卒．平 9 同大学院工学研究科了．同年，広島市立大学情報科学部助手．平 12 北海道大学大学院工学研究科助教授．工博．主として，コンピュータグラフィックスに関する研究に従事．情報処理学会，映像情報メディア学会各会員．



西田 友是（正員）

昭 46 広島大・工卒．昭 48 同大学院工学研究科了．同年，マツダ（株）入社．昭 54 福山大学工学部講師．昭 59 同助教授．平 2 同教授．平 10 東京大学理学部教授．平 11 同大学院新領域創成科学研究科教授．工博．主として，コンピュータグラフィックスに関する研究に従事．情報処理学会，画像電子学会，ACM，IEEE 各会員．

Abstract

This paper proposes a non-photorealistic rendering method for creating various artistic images from an input image. There are two kinds of artistic images, region-based images, such as mosaic and stained glass, and stroke-based images, such as oil painting. The proposed method can create both kinds of images. In artistic images, it is common that edges (boundaries between regions with different colors) are emphasized and nearby regions having similar colors are drawn with one color. Therefore, the proposed method first approximates the input image using several regions such that the boundaries of the regions match the edges of the input image and fills each region with one color. Voronoi diagrams are used to approximate the input image. Next, region-based images can be created by rendering the Voronoi regions, while stroke-based images can be created by drawing strokes along the Voronoi regions whose colors are similar.

Keywords: Non-photorealistic rendering, region-based images, stroke-based images, Voronoi diagrams.