

絵画調画像を生成するための遺伝アルゴリズム

Genetic Algorithm for Generating Painterly Rendered Images

情報工学専攻 阿部 史

Information and System Engineering ABE Chikashi

1. はじめに

非写実的レンダリングは写真などの画像から非写実的な画像を生成する技術である。非写実的レンダリングのひとつである絵画調画像の生成では、元画像の面影を残しつつ写実性をそぎ落とすことが重要になる。

近年、進化計算に基づく絵画調画像の生成手法が提案されている。本研究では絵画調画像を生成する遺伝アルゴリズムを提案する。絵画の色彩や質感、筆の運びを表現するために素材となる絵画画像をブラシのストロークの形に切り取り、作成したストロークをキャンバスの適当な位置に置くことを一回の描画処理とする。ストロークを画像に置く際には、ストロークの適切な向きを決定するためにストロークの方向場を利用する。提案する遺伝アルゴリズムが世代を重ねるごとに元画像の面影を残しつつ、より絵画らしい特徴を反映する画像を生成することを数値実験により示す。

2. 先行研究

非写実的レンダリングのひとつに絵画調画像の生成というものがある。これは写真などの画像を元にしてその画像の画風を水彩画や油絵風に変換した画像を生成するといった技術である。絵画調画像の生成では、元画像の面影を残しつつ写実性を削ぎ落とすことが重要である。

絵画調画像を生成する代表的な手法のひとつにブラシストロークを用いる手法がある[2]。ブラシストロークを用いて描画することで、筆を使って描いたかのような表現が可能になる。また、ブラシストロークのサイズより小さい部分を写真画像と同じように描画するのは難しいため、写真画像に近づきつつも写実性の削がれた絵画調画像が生成される。

絵画調画像の手法については様々な手法が提案されているが、近年、進化計算法に基づく絵画調画像の

生成手法が提案されている。ここで進化計算法とはメタヒューリスティックアルゴリズムの1つで、生物の進化過程にヒントを得た最適化探索アルゴリズムのことである。進化計算法のひとつである遺伝アルゴリズムは生物の進化メカニズムを模倣して最適解を探索する最適化探索アルゴリズムで、有性生殖をする生物の進化過程のうち、以下の点に着目してモデル化される。

1. 優秀な個体ほど次世代に自分の遺伝子を残せる
2. 2個体の交叉により子を作る
3. 時々突然変異が起こる

絵画調画像を生成する遺伝アルゴリズムの手法のひとつに、既存の絵画画像を用いる手法[3]がある。この手法では、目標画像として写真画像を、素材画像に既存の絵画画像を用意し、既存の絵画画像の一部を切り抜いたパッチとよばれるものを使用する。目標画像と同じサイズの仮想キャンバス画像を遺伝アルゴリズムの個体の染色体とし、突然変異と交叉を行う。仮想キャンバス画像が目標画像にどれだけ近づいたかは適応度関数により評価する。

突然変異では、仮想キャンバス画像にパッチを置く描画処理を行う。その際、パッチを置く位置を決めるために仮想キャンバス画像から候補点をいくつか選択し、候補点の中でパッチと最も色が近い点を配置先とする。このとき、パッチを置くだけではパッチの輪郭が強くなってしまいうため、パッチと仮想キャンバス画像の境界部分では色をなじませる処理を行う。図1は突然変異の流れを簡単に表したものである。

交叉では、目標画像の輪郭部分を復元する処理を行う。具体的には目標画像に対してあらかじめ領域分割を行い、領域分割の情報を利用して、目標画像の同じ領域部分に所属する二つの仮想キャンバス画像の色を交換する。図2は交叉の流れを簡単に表したものである。

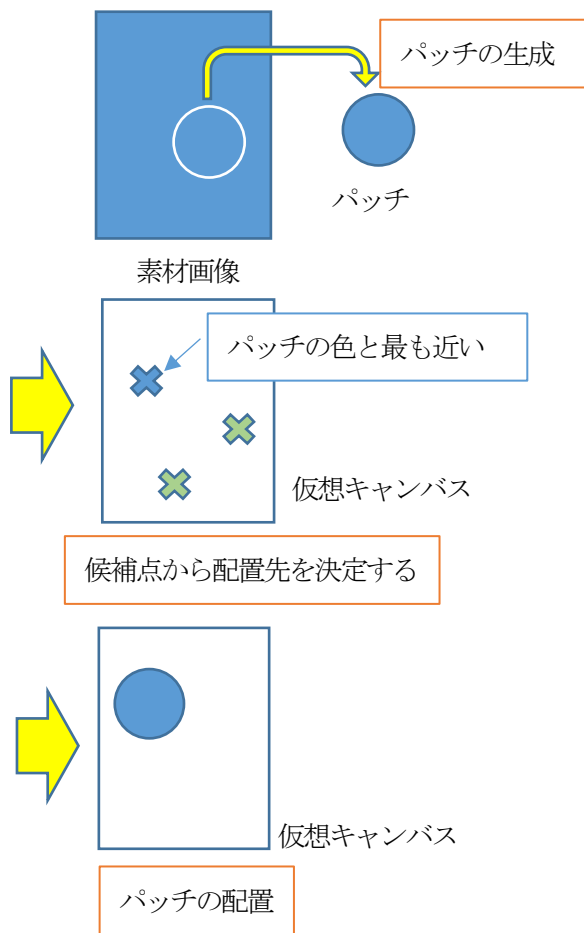


図1 突然変異の流れ

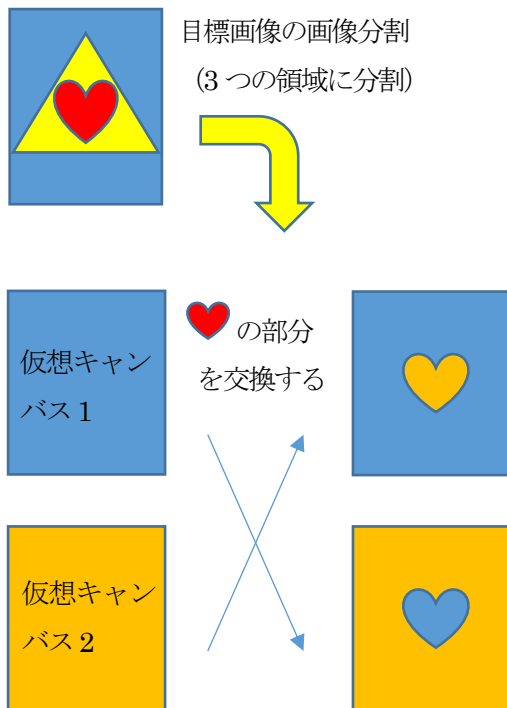


図2 交叉の流れ

適応度関数は以下のように定義する：

$$F = \sum_{y=1}^H \sum_{x=1}^W \sqrt{\sum_{c=1}^C (I(x,y,c) - T(x,y,c))^2}$$

ここで、 $I(x,y,c)$ は仮想キャンバス画像の画素の階調値、 $T(x,y,c)$ は目標画像の画素の階調値であり、 W と H は画像の横幅と縦幅、 C は色成分の数を表す。適応度関数の値は仮想キャンバスの色が目標画像の色に近づくほど小さくなる。

3. 提案手法

2節で紹介した手法[3]ではパッチが円形であるため、遺伝アルゴリズムの世代交代の回数が少ない場合にはパッチの形が目立ってしまい、絵画として不自然になることがある。本研究ではパッチの形を変更することで、より絵画らしい画像の生成を目指す。2.1節で紹介したように、ブラシストロークを用いた手法はブラシストロークを用いて描画することで、筆を使って描いたかのような表現が可能になる。

提案手法では文献[2]を参考にして、遺伝アルゴリズムで使用するパッチを円形からブラシストロークに変更する。図2のようにストローク状のパッチは円形とは異なり、パッチを生成する際に素材画像を切り抜く時の向きや仮想キャンバス画像にパッチを置くときの向きによって生成されるものが大きく変わる。例えば、素材画像から切り抜く際に素材画像の筆の流れに沿って切り抜くことができるなら素材画像の味を残すことができると考えられる。生成したパッチを置く際

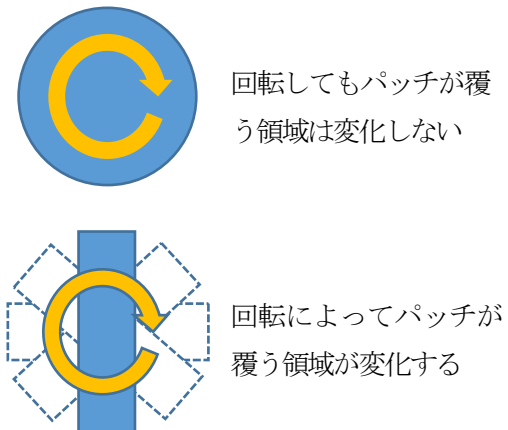


図3 パッチの形の違いによる回転の影響

には、ある物体の輪郭に沿った向きにパッチを置くことができれば、目標画像をより正確に表現できると考えられる。提案手法では、パッチを置く向きを決定するために、ストロークの方向場を利用する[1, 4]。ストロークの方向場 $\mathbf{v}(x, y) = [u(x, y), v(x, y)]$ はエネルギー汎関数：

$$E(\mathbf{v}) = \iint |\nabla I^\perp|^2 |\mathbf{v} - \nabla I^\perp|^2 + \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) dx dy$$

を最小化することで求められる。ここで、 μ は正則化パラメータ、 $\nabla I = (\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y})$ は画像の勾配、 ∇I^\perp は画像の勾配 ∇I に垂直なベクトルである。

エネルギー汎関数 $E(\mathbf{v})$ を最小化するためには、変分法で得られる Euler 方程式：

$$\mu \nabla^2 u - \left(u + \frac{\partial I}{\partial y} \right) \left(\left(\frac{\partial I}{\partial x} \right)^2 + \left(\frac{\partial I}{\partial y} \right)^2 \right) = 0$$

$$\mu \nabla^2 v - \left(v - \frac{\partial I}{\partial x} \right) \left(\left(\frac{\partial I}{\partial x} \right)^2 + \left(\frac{\partial I}{\partial y} \right)^2 \right) = 0$$

を解けばよい。この Euler 方程式を文献[4]をもとに拡散方程式として以下のように書き直す：

$$u_t(x, y, t) = \mu \nabla^2 u(x, y, t) - b(x, y)u(x, y, t) + c^1(x, y)$$

$$v_t(x, y, t) = \mu \nabla^2 v(x, y, t) - b(x, y)v(x, y, t) + c^2(x, y)$$

$$b(x, y) = \left(\frac{\partial I(x, y)}{\partial x} \right)^2 + \left(\frac{\partial I(x, y)}{\partial y} \right)^2$$

$$c^1(x, y) = -b(x, y) \frac{\partial I(x, y)}{\partial y}$$

$$c^2(x, y) = b(x, y) \frac{\partial I(x, y)}{\partial x}$$

ここで u_t , v_t , $\nabla^2 u$, $\nabla^2 v$ を以下のように近似する：

$$u_t = \frac{1}{\Delta t} (u_{i,j}^{n+1} - u_{i,j}^n), \quad v_t = \frac{1}{\Delta t} (v_{i,j}^{n+1} - v_{i,j}^n)$$

$$\nabla^2 u = \frac{1}{\Delta x \Delta y} (u_{i+1,j} + u_{i,j+1} + u_{i-1,j} + u_{i,j-1} - 4u_{i,j})$$

$$\nabla^2 v = \frac{1}{\Delta x \Delta y} (v_{i+1,j} + v_{i,j+1} + v_{i-1,j} + v_{i,j-1} - 4v_{i,j})$$

このとき拡散方程式は以下のように変形される：

$$u_{i,j}^{n+1} = (1 - b_{i,j} \Delta t) u_{i,j}^n + \frac{\mu \Delta t}{\Delta x \Delta y} (u_{i+1,j}^n + u_{i,j+1}^n + u_{i-1,j}^n + u_{i,j-1}^n - 4u_{i,j}^n) + c_{i,j}^1 \Delta t$$

$$v_{i,j}^{n+1} = (1 - b_{i,j} \Delta t) v_{i,j}^n + \frac{\mu \Delta t}{\Delta x \Delta y} (v_{i+1,j}^n + v_{i,j+1}^n + v_{i-1,j}^n + v_{i,j-1}^n - 4v_{i,j}^n) + c_{i,j}^2 \Delta t$$

以上の漸化式を用いて反復処理をすることによって u と v を求めることができる。

4. 実験結果

提案手法を用いて実際に絵画調画像を生成し、既存手法と比較する。また目標画像は変えず、素材画像を変えた場合の結果についても考察する。

図4は絵画調画像を生成する際に目標画像となる写真画像、図5は素材画像となる絵画画像である。提案手法を用いて生成した画像と使用したストローク状のパッチの例を図6に示す。図7は図4から生成したストロークの方向場であり、図4の赤枠で囲った部分を拡大表示したものである。図7の方向場は、図6に示したストローク状のパッチを置く向きを決める際に利用されている。

図8は、2.2節で述べた既存手法[3]を用いて生成した画像である。図6と図8はどちらも10000世代の結果である。既存手法ではより多く世代を重ねることでより望ましい絵画調画像が得られる。生成画像を見てみると円形に切り抜かれたパッチが置かれているのが確認できる。空の部分などは白色や青色の部分うまく切り抜けたパッチが置かれた場合は自然な描写となっているが、緑や黄色の部分に切り抜かれた場合その色が円形に残ることで不自然に目立っている箇所もある。

提案手法と既存手法を比較すると、提案手法の方が目標画像に近く、建物の輪郭や雲の形もはっきりしている。パッチを円形からストローク状に変更し、パッチを置く際にストロークの方向場を利用することで、輪郭に沿ってパッチを置くことができ、より自然な表現ができています。こちらも空の部分などに黄色や緑の部分が残っているがストローク状のパッチにすることで違和感を少なくし、パッチの形が残る部分を既存手



図4 目標画像



図5 素材画像A

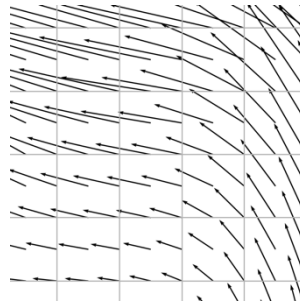


図7 図4から生成したストロークの方向場



図8 既存手法

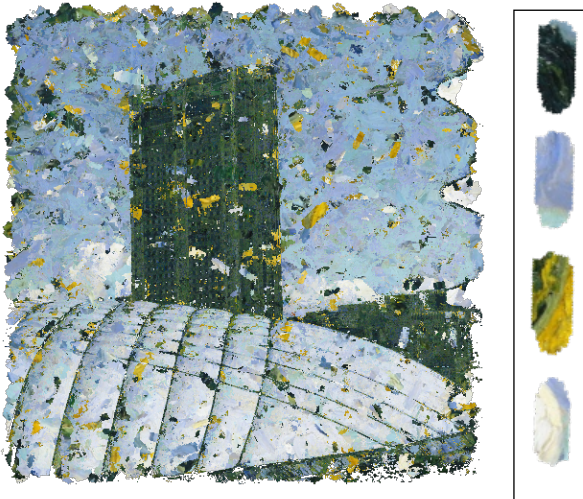


図6 提案手法で生成した画像(左)と切り取ったストローク状のパッチの例(右)



図9 素材画像B



図10 素材画像C

法より小さくすることで目立たなくさせている。

次に、様々な素材画像を用いて提案手法による生成実験を行った。目標画像は図4に固定し、素材画像を図9、図10に変更した場合の生成画像がそれぞれ図11と図12である。

図11は全体的に暗い印象を受ける。これは図11の素材画像である図9が全体的に暗く、青や灰色に近い色の部分がパッチとしてビルやドーム、空などに配置されたためと考えられる。図12は全体的に明るく淡い印象を受ける。これは図12の素材画像である図10が全体的に明るく、白色や水色に近い色の部分がパッチとしてビルやドーム、空などに配置されたためと考えられる。ビルは図6や図11のものとは比べて淡く白いものとなりぼやけた印象となったが、一方で空やドームの部分は陰影やグラデーションを細かく表現でき、素材となる絵画の持ち味を活かした結果となっていると言える。

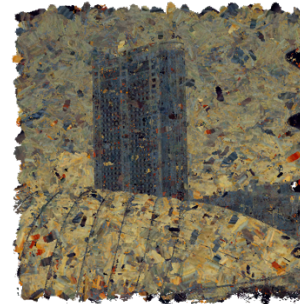


図11 素材画像Bを用いた生成画像

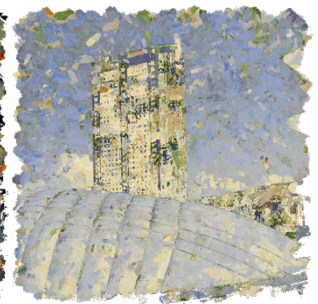


図12 素材画像Cを用いた生成画像

参考文献

- [1] U. K. Chakraborty, H. W. Kang and P. P. Wang: Image-Based Painterly Rendering by Evolutionary Algorithm, *New Mathematics and Natural Computation*, Vol. 3, No. 2, pp. 239-257, 2007.
- [2] P. Haeberli: Paint by numbers: Abstract image representations, *Computer Graphics*, Vol. 24, pp.207-214, 1990.
- [3] 中山恵太, 白川真一, 矢田紀子, 長尾智晴: 既存の絵画画像を用いた絵画風画像の進化的生成, 進化計算学会論文誌, Vol. 3, No. 2, pp. 12-21, 2012.
- [4] C. Xu and J. L. Prince: Snakes, Shapes, and Gradient Vector Flow, *IEEE Transactions on Image Processing*, Vol. 7, No. 3, pp. 359-369, 1998.