芸術科学会論文誌 Vol. 16, No. 3, pp. 40-55 (2017)

微細な物体輪郭を再現する半透明境界拡張テクスチャ合成法

越後谷勇介（学生会員）　**藤本忠博**（正会員）

岩手大学大学院工学研究科

Transparent Boundary Expansion Texture Synthesis for Reconstructing Fine Object Contours

Yusuke Echigoya[1)] (Student Member)　**Tadahiro Fujimoto**[2)] (Member)

Graduate School of Engineering, Iwate University

1) h23j013@eecs.iwate-u.ac.jp　2) fujimoto@cis.iwate-u.ac.jp

アブストラクト

ユーザが描いたペイント画像に合わせてサンプル画像の特徴的な色パターンを出力画像に再現するユーザ制御テクスチャ合成は，通常，サンプル画像上の物体の輪郭（物体と背景の境界）を特別には考慮しない．そのため，背景が不要な場合や異なる背景に対して，物体の画素だけを合成したい場合，ユーザがおおまかに描いたペイント領域の輪郭に合わせて物体の特徴的な輪郭を出力画像上に再現することは難しい．そこで，ユーザ制御テクスチャ合成の基本的な仕組みを利用して，ペイント領域の輪郭を適切に拡張し，サンプル画像上の特徴的な物体輪郭を特別に考慮して再現する境界拡張テクスチャ合成法が提案された．本研究では，従来の境界拡張テクスチャ合成法を発展させ，各画素に不透明度をあらわすα値を与えることで，サンプル画像が持つ特徴的な色パターンとともに，より微細な物体輪郭の再現を可能とする半透明境界拡張テクスチャ合成法を提案する．

Abstract

A user controlled texture synthesis technique reproduces the characteristic color pattern of a sample image on an output image in accordance with a paint image given by a user. This technique does not specially treat the contour of an object, that is, the boundary between an object and its background on the sample image. Thus, when a user wants to use only the pixels of the object for no background or other backgrounds in the synthesis, it is difficult to reproduce the characteristic contour of the object on an output image in accordance with the contour of a paint region roughly painted by the user. Therefore, a boundary expansion texture synthesis technique was proposed to reproduce the characteristic object contour of a sample image by appropriately expanding the contour of a paint region based on the user controlled texture synthesis. In this paper, we propose a transparent boundary expansion texture synthesis technique based on the previous technique. Our new technique reproduces the finer object contour as well as the characteristic color pattern of a sample image by using opacity values, alphas, given to pixels of sample and paint images.

1. はじめに

イメージエディット（image editing）とは，入力画像から目的に応じた様々な画像を出力する画像合成技術の総称であり，コンピュータグラフィックスやコンピュータビジョンの分野で活発に研究されている．テクスチャ合成（texture synthesis）は，入力画像（入力テクスチャ）の特徴を維持する大きさの異なる新たな出力画像（出力テクスチャ）を合成する技術であり，多くのイメージエディット手法の基礎となる重要な技術である[1-10]．ユーザ制御テクスチャ合成[3]は有用な応用技術であり，ユーザが自由にペイント描画した目的画像の配色に合わせて入力画像から出力画像を合成する．この技術では，入力画像に物体と背景が含まれる場合，物体と背景の画素は区別なく同等に扱われ，物体の輪郭は特別には考慮されない．そのため，背景が不要な場合や異なる背景に対して，ユーザが物体だけを意図して描いたペイント領域に入力画像上の物体の画素だけを合成し

たい場合，ユーザがおおまかに描いたペイント領域の輪郭に合わせて物体の特徴的な輪郭を再現した出力画像を生成することは難しい．通常，ペイント領域の輪郭に沿って物体が不自然に途切れてしまう．おおまかに描いたペイント領域に対しても入力画像上の物体の輪郭が出力画像上に適切に再現されることが望まれる．

物体輪郭を考慮した手法としては，Ritter らの“Painting with Texture”[11]と Lukac らの“Painting by Feature”[12]が挙げられる．そして，前者に類似したアプローチにより，その問題点を改善し，特に，複雑で大きな凹凸を持つ輪郭形状を再現することを可能とする境界拡張テクスチャ合成法[13]が提案された．この方法は，ユーザ制御テクスチャ合成に基づき，おおまかに描いたペイント領域に対しても，入力画像上の物体の輪郭，すなわち，物体と背景の境界に合わせてペイント領域の輪郭から外側に向けて合成領域の境界を適切に拡張する．その結果として，物体が不自然に途切れてしまうことなく，特徴的な物体輪郭を出力画像上に再現する．

本研究では，従来の境界拡張テクスチャ合成法を発展させ，各画素に不透明度をあらわすα値を与えることで，より微細な物体輪郭の再現を可能とする半透明境界拡張テクスチャ合成法を提案する．また，将来的にインタラクティブなペイントシステムとしての実現を目指すため，従来手法で処理時間を要した部分を効率化し，処理時間の短縮を図る．本手法では，色パターンと微細な輪郭形状の再現のため，色値に関する相違度とα値に関する相違度により，従来手法とは異なる基準で最適な合成画素を探索する近隣画素群の評価を行う．また，ペイント画像の配色とサンプル画像の色パターンの再現性のトレードオフを調整するペイント・サンプル調整重みを導入し，出力画像の品質向上と処理時間の短縮を図る．さらに，最適な合成画素の探索に用いる k-候補探索の前処理に random walk 法を用いて処理時間を短縮する．また，物体輪郭の再現性の指標となる輪郭形状の再現率を定義し，出力画像上に再現された物体輪郭がどのくらいサンプル画像上の物体輪郭に類似しているかを定量的に評価する．

2. 関連研究

2.1 テクスチャ合成

テクスチャ合成の方法は局所的逐次合成法[1-7]と大域的反復合成法[8-10]に大別される．局所的逐次合成法は，入力画像と出力画像の間で局所的に近隣画素群の類似性を評価し，逐次的に出力画像の合成領域を拡大する．この方法は，画素（ピクセル）単位で合成を行うピクセルベース手法[1-4]と隣接する画素群からなるパッチ単位で合成を行うパッチベース手法[5-7]に分類される．大域的反復合成法[8-10]は，出力画像の全画素を同時に反復的に更新し，出力画像全体を大域的に最適な画像へ収束させる．

本研究で提案する手法は，局所的逐次合成法の中でも画素単位で合成を行うピクセルベース手法に基づく．ピクセルベース手法は，通常，走査線順で出力画像の画素を合成する．出力画像上の処理画素に隣接する合成済みの近隣画素群に対して，最も類似する色パターンの近隣画素群を持つ入力画像上の画素を探索し，その色を処理画素に合成する．最もシンプルな全探索法（full search）は入力画像上の全ての画素を探索するが，探索を効率化する候補探索法（coherence search）[3]，その候補の数を増やす k-候補探索法（k-coherence search）[4]もある．また，近似的に最類似の近隣画素群を高速に探索する ANN 探索（Approximate Nearest Neighbor Search）の方法として，PatchMatch 法[14]や random walk 法[15]が提案されている．

テクスチャ合成の応用である事例ベーステクスチャ合成（example-based texture synthesis）は，事例として与えたサンプル画像を利用して出力画像を生成する技術であり，様々な手法が提案されている．ユーザ制御テクスチャ合成（user controlled texture synthesis）[3]は，ユーザが描いた目的画像の配色に合わせて入力画像から出力画像を合成する．Image Analogies[16]は，事例として与えた元画像とその変換画像から，入力画像の雰囲気を変える画像変換フィルタを実現する．その応用である Texture-by-numbers[15,16]は，サンプル画像を構成要素の種類によってラベリング色でセグメント分割した後，ラベリング色で描いたペイント画像に合わせてサンプル画像から出力画像を生成する．その他にも，テクスチャ合成に基づく様々な高度なイメージエディット技術が提案されている[9,14,17]．

ユーザが描いた画像を利用する事例ベーステクスチャ合成では，描かれた図形の輪郭に沿ってサンプル画像上の特徴的な物体輪郭を再現することは重要である．物体輪郭を考慮した手法として，Ritter らの手法[11]と Lukac らの手法[12]がある．前者では，近隣画素群の評価に前景（物体やテクスチャパターン）が映る「前景画素」と背景が映る「背景画素」の区別を導入し，ユーザのペイント描画に合わせてサンプルテクスチャの輪郭を考慮した出力画像をインタラクティブに生成する．後者では，１次元の特徴を持つ前景輪郭と２次元の特徴を持つ前景内部を分けて扱い，それぞれを描画するブラシ（brush）ツールとフィル（fill）ツールによってサンプル画像に基づく出力画像をインタラクティブにペイント描画する．

2.2 境界拡張テクスチャ合成法

境界拡張テクスチャ合成法[13]は，Ritter らの手法に類似したアプローチにより，物体輪郭の形状を考慮した類似度評価により合成画素の探索を行う．この手法は，ユーザ制御テクスチャ合成に基づき，サンプル画像（入力画像）の特徴的な物体輪郭に合わせて，ユーザが描いたペイント画像（目的画像）上のペイント領域の外側まで合成領域の境界を適切に拡張させ，サンプル画像の色パターンと輪郭形状をともに出力画像上に再現する．詳細は 3.2 節で述べるが，特に Ritter らの手法との違いの観点から，その特長を以下に示す．

(1) 物体（前景）と背景の境界を考慮した最適な合成画素の探索のため，近隣画素群の評価に前景画素と背景画素の区別を導入し，サンプル画像と出力画像・ペイント画像の対応画素

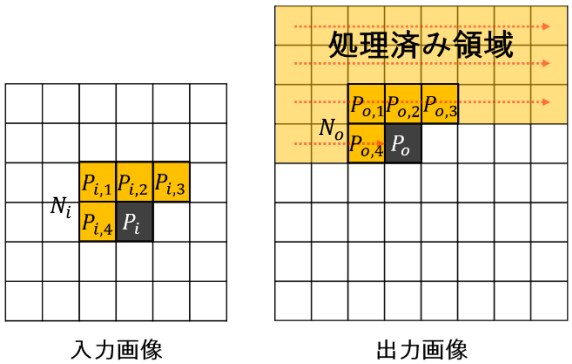


図 1. ピクセルベース手法によるテクスチャ合成

における前景画素と背景画素の組み合わせを考慮した相違度を与える．特に，前景画素ー背景画素の組に対する調整可能な重みにより複雑な境界形状に安定して対応する．

(2) 出力画像にサンプル画像の類似した境界を対応させるため，「境界距離」を用いて近隣画素群を評価する．

(3) 複雑で大きな凹凸を持つ境界形状の再現のため，キャンバス上の処理画素を合成キューに格納して管理し，ペイント領域の外側に向かって合成領域の境界を拡張することを可能とする．そして，適切な「境界画素」の決定により拡張を停止する．

(4) ユーザ制御テクスチャ合成を利用し，ユーザが意図した配色を持つ出力画像を生成する．

2.3 本手法と従来手法の関連

本研究の半透明境界拡張テクスチャ合成法（本手法）は，新たにα値を導入して微細な物体輪郭の再現を可能とする点で，従来の境界拡張テクスチャ合成法（従来手法）を発展させたものである．本手法は次の点で従来手法を発展させている．

(1) 色パターンと輪郭形状の再現性の制御　役割が異なる２つの相違度（色パターンの再現のための色値に関する相違度，輪郭形状の再現のためのα値に関する相違度）を用い，境界拡張の度合いの変化を伴う色パターンと輪郭形状の再現性を制御可能とする．

(2) ペイント画像とサンプル画像の影響の制御　ペイント・サンプル調整重みの導入により，ペイント画像の配色とサンプル画像の色パターンの再現性のトレードオフを制御可能とし，出力画像の品質の向上を図る．

(3) 処理速度の向上　合成画素を探索する k-候補探索の前処理で k-候補を求めるため，全探索ではなく，効率的な random walk 法を用いる．輪郭形状の再現のためのα値に関する相違度の導入により，境界拡張により絶えず再計算を要する境界距離を不要とする．ペイント・サンプル調整重みにより，近隣画素群のサイズを縮小化して処理時間を短縮する．

(4) 物体輪郭の再現性の評価　出力画像上に再現された物体輪郭に対して，サンプル画像上の物体輪郭と形状が一致しているかを比較する輪郭形状の再現率による定量的な評価を行う．

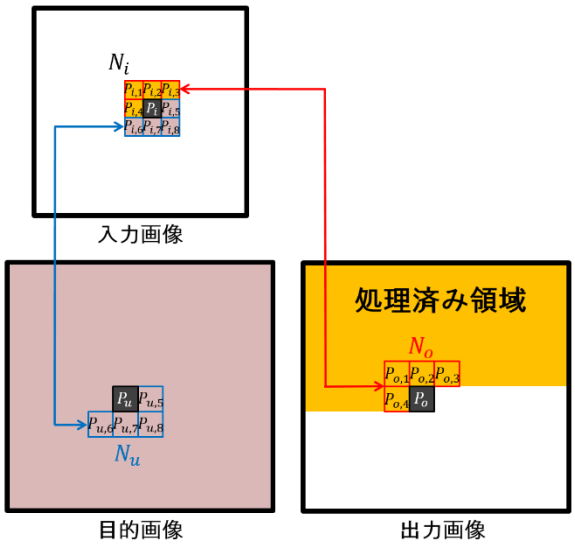


図 2. ユーザ制御テクスチャ合成

3. 本研究に必要な既存技術

3.1 テクスチャ合成の基本アルゴリズム

本研究で用いるピクセルベースのテクスチャ合成では，図 1 のように，通常，走査線順で出力画像の画素を合成する．出力画像上の処理画素 P_o について，合成済みの M 個の画素からなる左上 L 型の近隣画素群 N_o に対して最類似の色パターンを持つ入力画像上の画素 P_i の近隣画素群 N_i を探索し，画素 P_i の色を画素 P_o に合成する．２つの近隣画素群の類似性は，対応する位置の画素どうしが持つ色値の差の二乗の和（Sum of Squared Difference : SSD）による相違度 F を用いて評価する．

$$F = \sum_{m=1}^{M} \left| \boldsymbol{C}_{i,m} - \boldsymbol{C}_{o,m} \right|^2$$

$$= \sum_{m=1}^{M} \left(R_{i,m} - R_{o,m}\right)^2 + \left(G_{i,m} - G_{o,m}\right)^2 + \left(B_{i,m} - B_{o,m}\right)^2 \tag{1}$$

$\boldsymbol{C}_{i,m} = [R_{i,m}, G_{i,m}, B_{i,m}]$ と $\boldsymbol{C}_{o,m} = [R_{o,m}, G_{o,m}, B_{o,m}]$ は近隣画素群 N_i と N_o 内の対応する画素 $P_{i,m}$ と $P_{o,m}$ の色であり，相違度 F が小さいほど類似性が高い．最もシンプルな全探索法は，処理画素 P_o に対して入力画像上の全ての画素 P_i を評価し，最も相違度 F が小さい画素を選択する．探索を効率化する候補探索法[3]は，処理画素 P_o の近隣画素群 N_o 内の各画素に合成済みの入力画像の画素を利用して，入力画像上の隣接する画素群がなるべくまとまって出力画像上に合成されるように，画素 P_o に合成する候補となる入力画像の画素 P_i を限定する．さらに，前処理で入力画像上の各近隣画素群について類似した k-1 個の近隣画素群を探索して記憶しておくことで，候補探索法の候補の数を k 倍に増やす k-候補探索法[4]がある．

ユーザ制御テクスチャ合成[3]は，ユーザが描いた目的画像の配色に合わせて，入力画像から目的画像と同じ解像度の出力画像を合成する．図 2 のように，出力画像上の処理画素 P_o に隣

接する合成済み画素からなる左上 L 型の近隣画素群 N_o に加えて，目的画像上の右下 L 型の近隣画素群 N_u も用い，それらを合わせた正方型の近隣画素群と最類似の色パターンを持つ入力画像上の画素 P_i の近隣画素群 N_i を探索し，画素 P_i の色を画素 P_o に合成する．近隣画素群の相違度は式(1)で求める．図 2 の目的画像の画素 P_u は出力画像の処理画素 P_o と同じ画素位置であり，式(1)では，目的画像上の近隣画素群 N_u 内の画素 $P_{u,m}$ の色を $\boldsymbol{C}_{o,m}$ とする．図 2 では $M = 8$ である．

3.2 境界拡張テクスチャ合成法の概要

境界拡張テクスチャ合成法[13]は，ユーザ制御テクスチャ合成に基づき，サンプル画像（入力画像）の特徴的な物体輪郭を再現するため，ユーザが描いたペイント画像（目的画像）上のペイント領域の外側まで合成領域の境界を適切に拡張させる．最類似の近隣画素群の探索には k-候補探索を用い，境界距離に関する相違度と色と前背景画素の区別に関する相違度により，最適な合成画素を決定する．また，合成キューと境界画素の利用により，適切な合成領域の境界拡張を行う．以下，その概要を説明する（詳細は文献[13]を参照）．

サンプル画像とペイント画像　ユーザが描きたい物体を含むサンプル画像を用意し，物体が映る画素を前景画素，それ以外の画素を背景画素として分離する．続いて，その物体の配色を考慮した色を用いてユーザがキャンバス上にペイントツールで自由に物体を描いたペイント画像を用意する．ペイントした領域内の画素が前景画素，それ以外の画素が背景画素となる．ユーザ制御テクスチャ合成により，サンプル画像は入力画像，ペイント画像は目的画像として扱われる．ペイント画像上のペイント領域の輪郭に合わせてサンプル画像上の特徴的な物体輪郭を再現した出力画像が生成される．

境界距離　合成の前処理として，サンプル画像とペイント画像の全ての前景画素について，背景画素に達するまでの 8 方向の距離を求める．これらを「境界距離」と定義し，距離の差を相違度として類似度評価に利用する．ペイント画像上の境界距離は合成領域の境界拡張で変化するため，合成処理の最中にも更新する．

画素の処理順序　合成処理では，ピクセルベースの画素単位の合成を行う．ペイント領域の境界拡張のため，ペイント領域の中心付近，具体的には，ユーザが指定した開始位置からペイント領域の境界に向けて内側から外側へ合成処理を行う．キャンバス上の画素の処理順序を記憶する合成キューを用意し，合成処理の開始時にペイント領域内の全画素を上記の順序で合成キューに格納する．格納順に処理画素を取り出して合成処理を行い，合成キューが空になった時点で全体の処理を終了する．

最適な合成画素の探索　キャンバス上の処理画素に合成するサンプル画像上の最適な画素の探索には k-候補探索[4]を用いる．合成の前処理として，サンプル画像の全ての画素 P_i に対して，式(1)と同様の評価により，サンプル画像内の他の画素

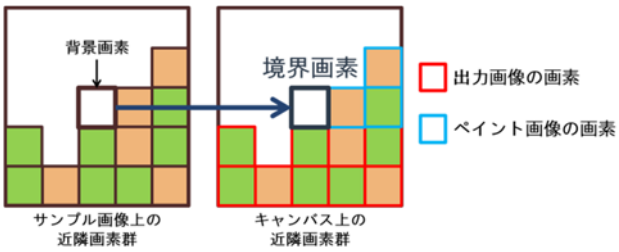


図 3．境界画素の合成

のうちで画素 P_i（P_i^1 と記す）と近隣画素群の類似性が高い k-1 個の画素 $P_i^2, \cdots, P_i^k$ を探索し，k-候補として記憶する．処理画素 P_o の合成画素の探索では，近隣画素群 N_o 内の各処理済み画素 $P_{o,m}$ に合成された画素 $Q_{i,m}$ に関する候補画素 $R_{i,m}$（$R_{i,m}^1$ と記す）と $R_{i,m}^2, \cdots, R_{i,m}^k$ からなる k-候補のみを探索の対象とし，処理画素 P_o と近隣画素群の類似性が最も高いものを選ぶ[4]．

近隣画素群の類似性の評価　ペイント画像の配色に合わせた出力画像を生成するため，サンプル画像を入力画像，ペイント画像を目的画像として近隣画素群を評価し，ユーザ制御テクスチャ合成を行う．相違度の算出には，式(1)による色に関する相違度に加えて，境界距離，ならびに，前景画素と背景画素の区別を利用する．境界距離に関する相違度 F_d，色と前背景画素の区別に関する相違度 F_c から，それらの比率を調整する重み a を用いて，最終的な相違度は以下のようになる．

$$F_{all} = a\,F_d + (1 - a)\,F_c \tag{2}$$

相違度が最小の候補画素が背景画素となる場合，処理画素に背景画素を合成する．これを「境界画素」と定義する（図 3 参照）．これはサンプル画像上の物体輪郭を適切に再現するために重要である．

合成領域の境界拡張　ペイント領域の外側に向けて合成領域の境界を適切に拡張するため，キャンバス上の現在の処理画素への合成を終えた時点で，その処理画素の上下左右の 4 近傍の画素をチェックする．それらのうち，ペイント領域外にあり，未処理で合成キューに未格納のものを合成キューに格納する．処理画素に境界画素（背景画素）が合成された場合（図 3 参照），その 4 近傍の画素の格納は行わず，合成領域の拡張を止める．これにより，複雑で大きな凹凸を持つ境界形状でも，サンプル画像の前景画素と背景画素の境界に合わせて適切に合成領域が拡張される．

大きな境界拡張の結果，合成済みの境界画素群が出力画像上でひび割れのような隙間を生じることがある．そこで，処理画素の 4 近傍の画素を合成キューに格納するかを判定する際，合成済みの境界画素は再格納し，全ての合成処理の後，近隣画素群の再評価による合成を行う．これにより，通常，前景画素が合成されることで隙間が埋められ，ひび割れが補完される．

3.3 Random walk 法

近似的に最類似の近隣画素群を高速探索する ANN 探索を行

うPatchMatch法[14]は，画像Aの各画素 P_a と類似した近隣画素群を持つ画像Bの画素 $P_b(P_a)$ を探索する．画像Aの全画素をランダムに初期化（initialization）した後，画像A上で走査線順に全画素に対して伝播（propagation）とランダム探索（random search）を行うことで $P_b(P_a)$ を更新する処理を反復する．

ランダム探索は，伝播の終了時の画素 $P_b(P_a)$ を中心とした探索範囲の半径を反復的に狭めながら，各反復時の探索範囲内でランダムに選択した画素 $P_b{}'$ と画素 $P_b(P_a)$ の画素 P_a に対する近隣画素群の類似性を比較し，画素 $P_b{}'$ の類似性が高ければ $P_b(P_a)$ を $P_b{}'$ で更新する．この反復は半径が1画素となるまで行う．ランダム探索を改良したrandom walk法[15]は，より効率的に近隣画素群の類似性が増すように画素 $P_b(P_a)$ を更新する．ランダム探索は反復的に狭める探索範囲の中心を伝播終了時の画素 $P_b(P_a)$ に固定する．一方，random walk法は，その中心を反復により更新される画素 $P_b(P_a)$ に動的に変更し，類似性の高い近隣画素群が得られる確率を高める．

4. 半透明境界拡張テクスチャ合成法

4.1 概要

本研究では，従来の境界拡張テクスチャ合成法[13]に対して，各画素に不透明度をあらわすα値を導入し，微細な輪郭の再現を可能とする半透明境界拡張テクスチャ合成法を提案する．本手法は，従来手法と同様に，ユーザ制御テクスチャ合成を利用し，ユーザが描いたペイント画像に合わせてサンプル画像が持つ特徴的な色パターンとともに微細な輪郭形状を出力画像上に再現する．この実現のため，近隣画素群の評価において，色パターンの再現のための色値に関する相違度，ならびに，輪郭形状の再現のためのα値に関する相違度を用いる．なお，このα値に関する相違度は，輪郭形状の再現のために導入されたものではあるが，この相違度が単独で特徴的な輪郭形状の再現のために働くのではなく，色値に関する相違度の働きと組み合わされることで，その効果が生み出される．つまり，２つの相違度が直接的に色パターンと輪郭形状の再現性をトレードオフの関係で制御するのではない．また，出力画像の品質向上のため，ユーザがペイント画像に与えた配色の再現性とサンプル画像の色パターンの再現性のトレードオフを調整するペイント・サンプル調整重みを導入する．これは，近隣画素群のサイズを小さくしても良好な出力画像が得られることに貢献し，処理時間の短縮も実現する．さらに，k-候補探索のための前処理でk-候補を求める方法として，類似した近隣画素群の探索を高速に行うrandom walk法を用いることで，さらなる処理時間の短縮がなされる．また，出力画像上に再現された物体輪郭がどのくらいサンプル画像上の物体輪郭に類似しているかを定量的に評価するための輪郭形状の再現率を定義する．

4.2 サンプル画像とペイント画像

従来手法と同様に，ユーザが描きたい物体を含むサンプル画像，その配色を考慮した色を用いてユーザがキャンバス上に物体を描いたペイント画像を用意する．ただし，従来手法と異なり，両画像とも，各画素には色値 $[R, G, B]$ に加えて不透明度 α を与える．なお，α値は $0 \leq \alpha \leq 1$ であるため，以下，色値も $0 \leq R, G, B \leq 1$ とする．また，物体が映る画素（$\alpha > 0$）を前景画素，それ以外の画素（$\alpha = 0$）を背景画素とする．サンプル画像のα値は，αマッティングのソフトウェア[18]等によりα値を持たない通常の画像から自動的に得ることができる．また，不透明度を与えた描画が可能であるペイントツール等によりα値を持つペイント画像を描くことができる．

4.3 境界距離

本手法では，α値を利用した評価により境界形状を再現するため，従来手法における境界距離を用いた評価は行わない．これにより，合成処理の最中に必要であったペイント画像の境界距離の更新は不要となり，処理時間の短縮が図られる．

4.4 画素の処理順序

従来手法ではペイント領域の中心付近から境界に向けて合成処理が行われた．しかし，実験により出力画像の品質が画素の処理順序にあまり影響されないことが分かったため，効率化のため，本手法では走査線順に処理を行う．しかし，境界拡張によりペイント領域外の画素も処理する必要があるため，従来手法と同様，開始時にペイント領域内の全画素を走査線順に合成キューに格納し，合成処理の最中に境界拡張に応じて新たに処理が必要となった画素を追加で格納していく．格納された順に処理画素を取り出して合成処理を行い，合成キューが空になった時点で全体の処理を終了する．

4.5 最適な合成画素の探索

処理画素に合成するサンプル画像上の最適な画素の探索には，従来手法と同様にk-候補探索を用いるが，前処理でk-候補を求める方法が異なる．従来手法は，画素 P_i^1 以外の全ての画素を全探索し，多くの処理時間を要した．一方，本手法は，3.3節で述べたrandom walk法により，処理時間を大幅に短縮する．画素 P_i^1 に対するk-1回のrandom walk法の適用で効率的に画素 $P_i^2, \cdots, P_i^k$ を得る．k-候補 $P_i^t, t = 2, \cdots, k$，を得るためのrandom walk法の実行では，反復的に探索範囲の半径を狭める探索において，反復1回目では，画素 P_i^1 を中心とした探索範囲内でランダムに選択した画素をk-候補 P_i^t の暫定候補とし，反復2回目の探索範囲の中心とする．反復2回目以降は，その時点の暫定候補を中心とした探索範囲内で画素をランダムに選択し，近隣画素群の類似性がより高ければ，新たな暫定候補とする．探索範囲の半径は，反復1回目ではサンプル画像の解像度の$1/2$とし，反復ごとに$1/2$倍する．半径が1画素以下となった時点の暫定候補を最終的にk-候補 P_i^t とする．

4.6 近隣画素群の類似性の評価

従来手法と同様，本手法でも，サンプル画像を入力画像，ペイント画像を目的画像として近隣画素群を評価し，ユーザ制御

テクスチャ合成を行う（図 2 参照）．ただし，本手法では，従来手法における境界距離に関する相違度は用いない．そして，微細な物体輪郭の再現のため，従来手法の色と前背景画素の区別に関する相違度に代わり，色値に関する相違度とα値に関する相違度を用いる．なお，色値に関する相違度には前背景画素の区別も含まれるが，従来手法とは異なる相違度を与える．また，出力画像の品質向上のため，色値に関する相違度にペイント画像の配色の再現性とサンプル画像の色パターンの再現性を調整するペイント・サンプル調整重みを導入する．

キャンバス上の現在の処理画素を囲む正方型の近隣画素群内で，処理済み画素はサンプル画像から出力画像に合成した色値とα値，未処理の画素はペイント画像の色値とα値を持つものとする．そして，キャンバス上の近隣画素群をサンプル画像上で探索する近隣画素群と比較して類似性を評価する相違度を計算する．物体が映る前景画素では $\alpha > 0$ であり，物体の内部では $\alpha = 1$，輪郭（境界）の部分では $0 < \alpha < 1$ である．背景画素では $\alpha = 0$ である．近隣画素群が M 個の画素を含むのものとすると，α値に関する相違度 F_α は次式となる．

$$F_\alpha = \sum_{m=1}^{M} \left|\alpha_{i,m} - \alpha_{o,m}\right|^2 \tag{3}$$

$\alpha_{i,m}$ はサンプル画像の画素 $P_{i,m}$ のα値，$\alpha_{o,m}$ はキャンバス上の出力画像・ペイント画像の画素 $P_{o,m}$ のα値である．相違度 F_α は，以下で述べる色値に関する相違度 F_c の働きと組み合わされることで，微細な輪郭形状を最適に再現することに貢献する．

色値に関する相違度 F_c は，式(1)に基づき，次式となる．

$$F_c = \sum_{m=1}^{M} F_m \tag{4}$$

値 F_m は m 番目の対応する画素間の相違度であり，サンプル画像と出力画像（処理済み画素の場合）・ペイント画像（未処理の画素の場合）の間の対応する画素が前景画素と背景画素のいずれの組み合わせかによって，表 1 に示す値が与えられる．まず，(a) 両方とも前景画素の場合，式(1)と同様，通常の色値による評価を行う．表 1 中の $\boldsymbol{C}_{i,m}$ はサンプル画像の画素 $P_{i,m}$ の色値，$\boldsymbol{C}_{o,m}$ はキャンバス上の出力画像・ペイント画像の画素 $P_{o,m}$ の色値である．β_m は出力画像上でのペイント画像の配色の再現性とサンプル画像の色パターンの再現性のトレードオフを調整するペイント・サンプル調整重みであり，以下のように２つの固定値 β_S，$\beta_P \geq 0$ を与える．キャンバス上の画素 $P_{o,m}$ が処理済み画素であり，$\boldsymbol{C}_{o,m}$ がサンプル画像から合成された出力画像の色値である場合，$\beta_m = \beta_S$ とする．画素 $P_{o,m}$ が未処理の画素であり，$\boldsymbol{C}_{o,m}$ がペイント画像の色値である場合，$\beta_m = \beta_P$ とする．よって，β_S を大きくすると出力画像上にサンプル画像の色パターンのほうが強めに再現され，β_P を大きくするとペイント画像の配色のほうが強めに再現される．

表 1．対応画素間の相違度 F_m

		サンプル画像	
		前景	背景
出力画像・ペイント画像	前景	$\beta_m \left\|\boldsymbol{C}_{i,m} - \boldsymbol{C}_{o,m}\right\|^2$	F_{max}
	背景	F_{max}	F_{max}

β_S と β_P には任意の値を与えることができるが，後ほど詳しく述べるように，小さな近隣画素群を用いて短い処理時間で良好な出力画像を得るためには，$\beta_S \geq \beta_P$ を満たしながら β_S と β_P の値を調整するのがよいということが実験によって経験的に分かった．そこで，5 節の実験では，常に $\beta_S = 1$ としながら β_P の値を $1 \geq \beta_P > 0$ の範囲で調整している．この場合，出力画像上には，β_P が 0 に近いほどサンプル画像の色パターン，1 に近いほどペイント画像の配色が強めに再現される．次に，(b) 両方とも背景画素の場合と (c) 一方が前景画素で他方が背景画素の場合を考える．従来手法では，(b) の場合には高評価，(c) の場合には低評価の相違度を与えることで，色に関する相違度 F_c が色パターンの再現に加えて前背景画素の区別による輪郭形状の再現にも直接的に関与していた．一方，本手法では，従来手法とは異なり，(b) と (c) の両方の場合，すなわち，少なくとも一方が背景画素の場合には，$|\boldsymbol{C}_{i,m} - \boldsymbol{C}_{o,m}|^2$ の最大値 $F_{max} = 3 \cdot 1^2$ を与えて低評価とする．これにより，キャンバス上の近隣画素群 N_o 内の背景画素に対してはサンプル画像上で探索する近隣画素群 N_i 内の対応する画素が前景画素か背景画素かによらずに最大値 F_{max} が与えられるため，類似性の評価は N_o 内の前景画素に関する相違度だけで決まる．このとき，その前景画素に N_i 内の背景画素が対応すると F_{max} が与えられてしまうため，前景画素が対応する N_i が選択される傾向が高くなり，その前景画素どうしの比較には (a) の色値による評価が行われることになる．結果として，最類似の近隣画素群 N_i の決定において (a) の色値による評価が支配的となることから，相違度 F_c は色パターンの再現にほぼ専念することになる．一方，従来手法とは異なり，本手法では輪郭形状の再現には前述のα値に関する相違度 F_α が関わる．役割が異なる２つの相違度を用い，それぞれの効果を後述の係数 w_{ac} によって調整する仕組みにより，境界拡張の度合いの変化を伴いながら，色パターンと輪郭形状の再現性が制御可能となる．

サンプル画像上の物体の特徴を捉えた色パターンを出力画像上に再現するためには，一般に，その特徴を十分に捉えるだけの大きさを持つ近隣画素群を用いる必要がある．ユーザ制御テクスチャ合成の仕組みでは，前述のように，色値に関する相違度 F_c の評価において，キャンバス上の近隣画素群内の処理済み画素には合成されたサンプル画像の色値，未処理の画素にはペイント画像の色値が評価に用いられる．通常，ユーザがおおまかに与えるペイント画像上の配色はサンプル画像上の実際の物体の色パターンとは一致しない．合成処理の開始時にはキャンバス上の全ての画素が未処理であり，合成処理の進行に伴って処理済み画素が増えていく．このとき，近隣画素群のサイズ

が小さく，その中の処理済み画素が少ないときには，合成されたサンプル画像の色値が少ないために物体の特徴を十分に捉えることが難しく，画素数の割合からも未処理の画素が持つペイント画像の色値が評価において支配的となる．その結果，出力画像はペイント画像に近いものとなる．よって，サンプル画像上の色パターンを出力画像上に的確に再現するためには，近隣画素群内に物体の特徴を捉えられる程度に多くの処理済み画素が必要であり，近隣画素群のサイズを大きくすることが一策である．しかし，近隣画素群のサイズの増加は処理時間を増加させる．そこで，本手法では，別の策により，小さな近隣画素群内で処理済み画素が少ない場合でも，その狭い範囲内でそれらの画素の色値が物体の色パターンの局所的な特徴を的確に捉えるために十分に働くようにする．具体的には，相違度 F_c の評価において，処理済み画素が持つサンプル画像の色値と未処理の画素が持つペイント画像の色値を同等に扱うのではなく，後者よりも前者に大きな重みを与えて評価する．これにより，キャンバス上の近隣画素群の中心にある現在の処理画素に対して，処理済み画素が持つ色値との間でサンプル画像上で適切なつながりを持つような色値を与える画素が k-候補から選択され，結果的にサンプル画像上の物体の特徴を捉えた色パターンが出力画像上に再現される．本手法では，先述のように，ペイント・サンプル調整重み β_m に与える２つの固定値 β_S と β_P を $\beta_S \geq \beta_P$ を満たしながら調整することとし，β_S に対して β_P を小さくすることで上記を実現する．これにより，近隣画素群のサイズを小さくしても良好な出力画像が得られることを可能とし，結果として処理時間の短縮も実現される．

最終的な相違度は，式(3)と式(4)を用いて，以下のようになる．

$$F_{all} = w_{\alpha c} \cdot 3F_\alpha + (1 - w_{\alpha c}) \cdot F_c \tag{5}$$

係数 $w_{\alpha c}$，$0 \leq w_{\alpha c} \leq 1$，はα値に関する相違度 F_α と色値に関する相違度 F_c の比率を調整する重みである．また，色値をあらわす $\boldsymbol{C} = [R, G, B]$ の各要素が $0 \leq R, G, B \leq 1$，α値が $0 \leq \alpha \leq 1$ であるため，固定値 β_S，β_P にもよるが，おおよその見積もりとして F_c は F_α の３倍の値となる．そのため，それらの評価が対等になるように F_α を３倍にする．そして，k-候補探索によるサンプル画像上の候補画素のうち，相違度 F_{all} が最小となるものを選択して処理画素に合成する．従来手法と同様，相違度が最小となる候補画素が背景画素（$\alpha = 0$）の場合，背景画素を境界画素として合成する（図３参照）．すなわち，その処理画素には色を与えず，背景画素であるという属性だけを与える．境界画素の合成は，サンプル画像上の物体輪郭に合わせてペイント領域の境界を適切に拡張するために重要である．

4.7 合成領域の境界拡張

ペイント領域内の画素の処理順序に違いはあるが，合成領域の境界拡張のため，従来手法と同様の方法で合成キューにより処理画素を管理する．また，従来手法と同様，合成済みの境界画素の再評価によるひび割れの補完も行う．

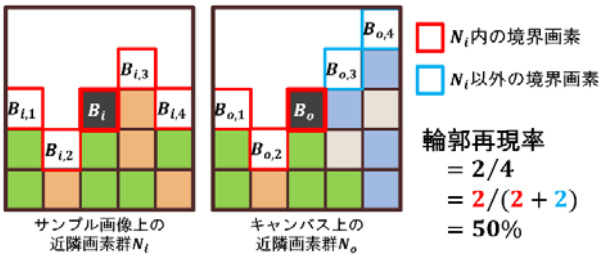


図４．輪郭再現性の評価の例

4.8 輪郭形状の再現性の評価

サンプル画像上の特徴的な物体輪郭が出力画像上に再現される度合い，すなわち，輪郭形状の再現性は，サンプル画像上で物体輪郭を構成する連続した背景画素群が出力画像上にそのまま連続した境界画素群として合成されて物体輪郭を構成しているかを調べることで評価する．合成処理が終了した出力画像が全体で K 個の境界画素を含むものとすると，輪郭形状の再現率 E_c を次式で定義する．

$$E_c = \sum_{k=1}^{K} \frac{H_k}{L_k} \tag{6}$$

再現率 E_c は，出力画像上の境界画素ごとに，その近隣画素群内の境界画素がサンプル画像上の背景画素からどれくらい正確に合成されたものであるかを評価する．値 L_k と H_k は k 番目の境界画素に関するものであり，以下の値とする．キャンバス上の画素 B_o にサンプル画像上の背景画素 B_i が合成された結果として，画素 B_o が出力画像上の境界画素になったとする．境界画素 B_o の近隣画素群（以下，近隣）を N_o，背景画素 B_i の近隣を N_i とする．このとき，合成処理が終了した出力画像上で近隣 N_o 内の画素 B_o 以外の境界画素の個数を L_k とし，その各境界画素について，近隣 N_i 内の同じ位置にある背景画素が合成されたものであるかを調べる．そして，この条件を満たす境界画素の個数を H_k とする．

例えば，図４の例では $L_k = 4$ である．近隣 N_o 内の境界画素 $B_{o,1}$ と $B_{o,2}$ は近隣 N_i 内の背景画素 $B_{i,1}$ と $B_{i,2}$ が合成されたものであり，一方，境界画素 $B_{o,3}$ と $B_{o,4}$ は近隣 N_i 以外の近隣内の背景画素が合成されたものとする．この場合には $H_k = 2$ となる．

再現率 E_c は $0 \leq E_c \leq 1$ となり，再現性が高いほど 1 に近い値となる．

5. 実験

従来手法と本手法との比較実験，ならびに，特徴的な微細な物体輪郭を半透明領域として含む様々なサンプル画像に対して本手法を適用した実験を行った．使用したサンプル画像は，αマッティングにより得られるαマット画像，または，ペイントツール（GIMP 2, Paint Tool SAI）を使って元の画像から物体領域を切り抜いて作成した画像である．使用した PC は，CPU: Intel Core i7-4790K，メモリ: 8.00 GB，OS: Windows 8.1 Pro である．

全ての実験で重み $\beta_S = 1$ である．

5.1 従来手法との比較

従来手法との比較実験の結果を図 5，図 6 に示す．図 5 の実験ではサンプル画像とペイント画像のいずれも半透明な領域を含む．αマット画像はペイント画像のα値を示し，白くなるほど高いα値をあらわす．α値を考慮しない従来手法では，ペイント画像上のα値が低い半透明な領域に対してもサンプル画像上のα値が高い画素が合成され，ペイント画像を十分に反映しない出力画像となっている．一方，本手法では，ペイント画像とサンプル画像のα値を考慮した適切な合成が実現されている．また，図 6 は重み β_m の効果を検証する本手法と従来手法との比較実験の結果である．従来手法では，近隣画素群サイズが大きい場合（15x15 画素）はペイント画像の配色とサンプル画像の色パターンがバランスよく再現されているが，サイズが小さい場合（7x7 画素）はサンプル画像の色パターンが十分に現れない．一方，本手法では，ペイント・サンプル調整重み β_m に与える値を $\beta_S > \beta_P$ とすることでサンプル画像から合成された出力画像の色値の評価が優先された結果，近隣画素群サイズが小さい場合（7x7 画素）にも比較的良好な出力画像が得られた．近隣画素群サイズが大きな場合（15x15 画素）の従来手法の結果と比較すると，評価領域が小さいために多少の色パターンや輪郭の崩れが現れたものの，処理時間の大幅な短縮が実現されている．

5.2 係数 $w_{\alpha c}$ の効果

図 7，図 8 は，本手法において，式(5)の係数 $w_{\alpha c}$ の値を変えて実験した結果である．いずれも半透明な物体輪郭を含むサンプル画像を用いており，ペイント画像にも半透明な領域を与えている．なお，それぞれの画像（左）をαマット画像（中）と色画像（右）に分離したものも示している．

図 7 の実験で用いたペイント画像は，左右に異なる２色を用いて，輪状に描かれたペイント領域の内外の境界部分で内側と外側に向かってα値が徐々に低くなるようにペイントされている．サンプル画像の物体輪郭に見られる微細な毛が，色の分布とともに，係数 $w_{\alpha c}$ の値に応じて異なる現れ方で出力画像上に再現されている．$w_{\alpha c} = 1.00$ の場合は，境界拡張があまり起こらず，ペイント画像のα値の分布に従う合成が行われており，ペイント領域の境界の半透明領域に沿ってサンプル画像の微細な輪郭形状は再現されているが，一方，ペイント画像の色の分布には従わず，サンプル画像の色パターンも再現されていない．逆に，$w_{\alpha c} = 0.00$ の場合は，サンプル画像の色パターンが再現され，ペイント画像の色の分布に起因する合成が行われているものの，ペイント画像のα値の分布から離れた大きな境界拡張が起こっている．しかし，$w_{\alpha c} = 1.00$ の場合に比べて，拡張された境界部分の輪郭形状はサンプル画像上のものに類似しており，輪郭形状の再現性が高いことが分かる．それらの中間の $0.00 < w_{\alpha c} < 1.00$ の場合は，その値に応じて，ペイント画像とサンプル画像が持つα値と色値の両方の分布を反映させた輪郭形状と色パターンの再現，ならびに，境界拡張がなされている．係数 $w_{\alpha c}$ の値を調整することで，輪郭形状と色パターンの再現性，境界拡張の度合いが制御され，出力画像の特徴が変化する．図 8 も図 7 と同様の条件で行った実験結果であり，出力画像には同様の特徴が見られている．

上記の係数 $w_{\alpha c}$ による効果は，式(5)における形状（以下の議論は「輪郭形状」に限定されないため，「形状」とする）の再現のためのα値に関する相違度 F_α と色パターンの再現のための色値に関する相違度 F_c の働きから，次のように説明できる．以下，簡単化のため，近隣画素群を近隣と呼ぶ．サンプル画像の画素をサンプル画素，その近隣をサンプル近隣と呼び，出力画像・ペイント画像をキャンバス画像，その画素をキャンバス画素，その近隣をキャンバス近隣と呼ぶ．サンプル画像の物体領域内の完全に不透明な部分（$\alpha = 1$）を「内部物体領域」，その境界部分（$0 < \alpha < 1$）を「境界物体領域」と呼び，ペイント画像のペイント領域内の完全に不透明な部分を「内部ペイント領域」，その境界部分を「境界ペイント領域」と呼ぶ．キャンバス画素 P_o の合成処理のため，そのキャンバス近隣 N_o に形状と色パターンが類似するサンプル近隣 N_i を探索して選択し，合成されるサンプル画素 P_i を決定するものとする．

(1) 係数 $w_{\alpha c}$ が大きい場合，相違度 F_α の影響が大きく，相違度 F_c の影響は小さい．よって，近隣 N_o と近隣 N_i の形状の類似性が重視され，色の類似性は重視されない．

(1-1) 近隣 N_o が内部ペイント領域内に収まる場合，内部物体領域内に収まる全ての近隣 N_i が近隣 N_o と形状が一致する（近隣全体で $\alpha = 1$ のため）．この場合，相違度 F_α は意味を持たず，結果的に相違度 F_c が重視されることになり，近隣 N_o に類似した色パターンを持つ近隣 N_i が選択される．

(1-2) 近隣 N_o が境界ペイント領域にある場合，近隣 N_o と形状（α値の分布）が類似して選択される近隣 N_i は境界物体領域にある可能性が高い．この場合，画素 P_o に合成される画素 P_i，その近隣の画素 P_o' に合成される画素 P_i' には背景画素が多く含まれ，それらが境界画素として合成されることで境界拡張が止まる．合成キューによる画素の処理順序により，ペイント領域内の画素から処理が行われるので，合成処理が境界ペイント領域に達した時点ではペイント領域外への境界拡張は行われていないため，近隣 N_o は元のペイント領域の境界形状を持つ．よって，それに類似した境界形状を持つ近隣 N_i が選択され，結果的に，元のペイント領域の境界形状に類似する輪郭形状が出力画像上に再現される．

(2) 係数 $w_{\alpha c}$ が小さい場合，相違度 F_c の影響が大きく，相違度 F_α の影響は小さい．よって，近隣 N_o と近隣 N_i の色の類似性が重視され，形状の類似性は重視されない．この場合には，4.6 節の表 1 による相違度 F_m に関する議論から，近隣 N_o が内部ペイント領域内に収まる場合と境界ペイント領域にある場合のいずれでも，近隣 N_o と近隣 N_i の類似性の評価において前景画素どうしの色値の比較が支配的となり，背景画素の存在はほとんど影響しない．その結果，近隣 N_o

に色パターンだけが類似する近隣 N_i が選択される．よって，近隣 N_o が境界ペイント領域にある場合であっても，選択される近隣 N_i が境界物体領域にはないことも多く，画素 P_o に合成される画素 P_i が背景画素にならず，画素 P_o が境界画素にならないため，境界拡張が進行する．この場合，進行中の境界拡張が止まるのは，境界に沿った連続する画素群 $\{P_o\}$ のそれぞれの近隣 N_o に対して，境界物体領域にある近隣 N_i が選択され，背景画素である画素 P_i が画素 P_o に合成される場合である．このとき，その画素群 $\{P_o\}$ に合成される背景画素群 $\{P_i\}$ はサンプル画像上で物体輪郭に沿って連続しているものである可能性が高い．この場合，サンプル画像上の輪郭形状と同じものが出力画像上に再現されることになり，結果として，輪郭形状の再現性が高くなる．

以上から，係数 $w_{\alpha c}$ は境界拡張の度合いを制御し，輪郭形状の再現性を決定する役割を果たすことが分かる．これは本手法の有用な機能である．図 7，図 8 の比較から，サンプル画像の物体輪郭の特徴と半透明領域の分布により適切な $w_{\alpha c}$ の値が異なることが分かる．

5.3 重み β_m の効果

図 9 と図 10 は，本手法において，ペイント・サンプル調整重み β_m に与える固定値 β_P の値を変えて実験した結果である．前述のように，他方の固定値は $\beta_S = 1$ としている．実験結果から，β_S に相対的に β_P の値を小さくするほど，ペイント画像の配色の再現性が低く，サンプル画像の色パターンの再現性が高くなることが分かる．ペイント・サンプル調整重みは，本手法の有用な機能として，目的通りの役割を果たしている．

5.4 輪郭再現性の評価の効果

図 5 から図 10 の実験で生成した各出力画像に対して，4.8 節で提案した輪郭形状の再現率（輪郭再現率） E_c の値を図中に示している．特に，図 7 と図 8 については，5.2 節で述べたように，係数 $w_{\alpha c}$ の値が小さくなるにつれて境界拡張の進行に伴って輪郭形状の再現性が高まっている．そして，実際，輪郭再現率が上昇している．$w_{\alpha c} = 1.00$ の場合は，輪郭再現率が極端に低くなる．これは，5.2 節で述べたように，α 値に関する相違度 F_α による評価のみでは境界拡張がほとんど起こらず，元のペイント領域の境界形状に類似する輪郭形状が出力画像上に再現されてしまうためと考えられる．一方，$w_{\alpha c} = 0.00$ で過度な境界拡張が行われた際には，輪郭再現率が極端に高くなる．これも，5.2 節で述べたように，拡張が止まった境界はサンプル画像上の物体輪郭と同じ形状を持つ可能性が高いことが要因と考えられるが，一方で，出力画像の全体に境界拡張が進行して停止していないことから，評価するべき境界画素が極端に少ないことも関係していると考えられる．しかし，過度な拡張であっても，輪郭形状は適切に再現されているため，輪郭再現率が高くなるのは正しい評価であると言える．以上から，提案した輪郭再現率の評価は妥当なものであると言える．

6. おわりに

本研究では，従来手法を発展させた半透明境界拡張テクスチャ合成法を提案した．不透明度（α 値）の導入により微細な物体輪郭の再現を可能とする本手法の能力が実験により確認された．今後の課題として，ユーザのペイント描画に対してリアルタイムで合成処理を行うインタラクティブシステムの実現，α マッティング技術[18]を利用して容易に高品質のサンプル画像を作成する方法などを検討している．

参考文献

[1] A. A. Efros and T. K. Leung, Texture Synthesis by Non-parametric Sampling, Proc. of IEEE International Conference on Computer Vision 1999, vol. 2, pp. 1033-1038, 1999.

[2] L. Wei and M. Levoy, Fast Texture Synthesis using Tree-Structured Vector Quantization, Proc. of ACM SIGGRAPH 2000, pp. 479-488, 2000.

[3] M. Ashikhmin, Synthesizing Natural Textures, Proc. of ACM Symposium on Interactive 3D Graphics 2001, pp. 217-226, 2001.

[4] X. Tong, J. Zhang, L. Liu, X. Wang, B. Guo and H. Shum, Synthesis of Bidirectional Texture Functions on Arbitrary Surfaces, Proc. of ACM SIGGRAPH 2002, pp. 665-672, 2002.

[5] A. A. Efros and W. T. Freeman, Image Quilting for Texture Synthesis and Transfer, Proc. of ACM SIGGRAPH 2001, pp. 341-346, 2001.

[6] V. Kwatra, A. Schodl, I. Essa, G. Turk and A. Bobick, Graphcut Textures: Image and Video Synthesis Using Graph Cuts, Proc. of ACM SIGGRAPH 2003, pp. 277-286, 2003.

[7] Q. Wu and Y. Yu, Feature Matching and Deformation for Texture Synthesis, Proc. of ACM SIGGRAPH 2004, pp. 364-367, 2004.

[8] V. Kwatra, I. Essa, A. Bobick and N. Kwatra, Texture Optimization for Example-based Synthesis, Proc. of ACM SIGGRAPH 2005, pp. 795-802, 2005.

[9] S. Lefebvre and H. Hoppe, Appearance-space Texture Synthesis, Proc. of ACM SIGGRAPH 2006, pp. 541-548, 2006.

[10] W. Dong, N. Zhou and J. Paul, Perspective-aware Texture Analysis and Synthesis, The Visual Computer, vol. 24, pp. 515-523, 2008.

[11] L. Ritter, W. Li, B. Curless, M. Agrawala and D. Salesin, Painting with Texture, Proc. of the 17th Eurographics Conf. on Rendering Techniques, pp. 371-376, 2006.

[12] M. Lukac, J. Fiser, J. Bazin, O. Jamriska, A. Sorkine-Hornung and D. Sykora, Painting by Feature: Texture Boundaries for Example-based Image Creation, Proc. of ACM SIGGRAPH 2013, pp. 116(1)-(8), 2013.

[13] 越後谷勇介，横山俊希，藤本忠博，物体輪郭を再現する境界拡張テクスチャ合成法，芸術科学会論文誌 Vol. 15, No. 2, pp. 86-97, 2016.

[14] C. Barnes, E. Shechtman, A. Finkelstein and D. B Goldman,

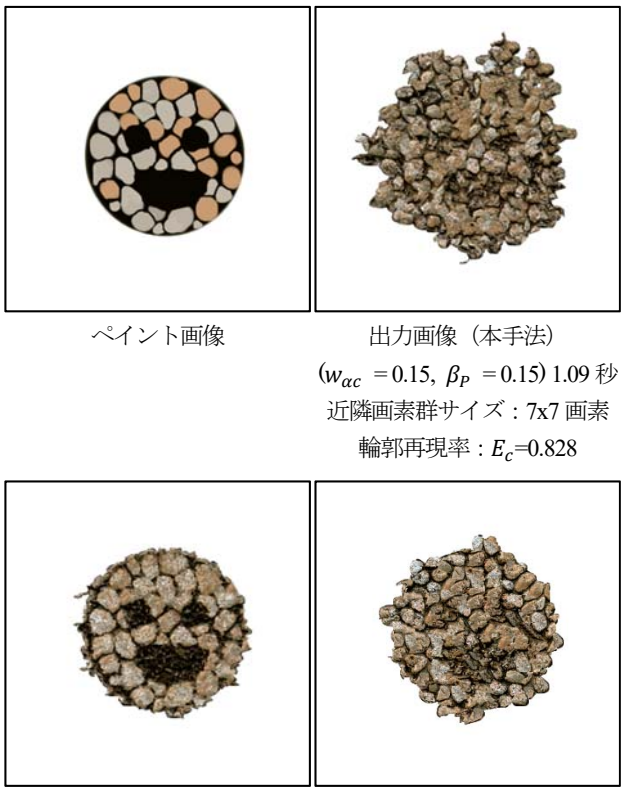
ペイント画像　　αマット画像

出力画像（本手法）
($w_{\alpha c}$ =0.90, β_P =0.10) 7.26 秒
輪郭再現率：E_c=0.700

出力画像（従来手法）4.38 秒

図 5. 本手法と従来手法の比較
サンプル画像：200x200 画素
ペイント画像：400x400 画素
近隣画素群サイズ：9x9 画素
k-候補数：k = 3

サンプル画像
（毛）



ペイント画像　　出力画像（本手法）
($w_{\alpha c}$ =0.15, β_P =0.15) 1.09 秒
近隣画素群サイズ：7x7 画素
輪郭再現率：E_c=0.828

出力画像（従来手法）0.70 秒
近隣画素群サイズ：7x7 画素

出力画像（従来手法）16.51 秒
近隣画素群サイズ：15x15 画素

図 6. 本手法と従来手法の比較
（重み β_m の効果）
サンプル画像：200x200 画素
ペイント画像：400x400 画素
k-候補数：k = 3

サンプル画像
（石）

PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing, Proc. of ACM SIGGRAPH 2009, pp. 24(1)-(12), 2009.

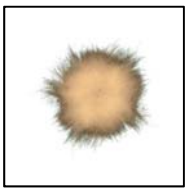
[15] P. P. Busto, C. Eisenacher, S. Lefebvre and M. Stamminger, Instant Texture Synthesis by Numbers, Vision, Modeling and Visualization, pp. 81-85, 2010.

[16] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless and D. H. Salesin, Image Analogies, Proc. of ACM SIGGRAPH 2001, pp. 327-340, 2001.

[17] S. Darabi, E. Shechtman, C. Barnes, D. B Goldman and P. Sen, Image Melding: Combining Inconsistent Images using Patch-based Synthesis, Proc. of ACM SIGGRAPH 2012, pp. 82(1)-(10), 2012.

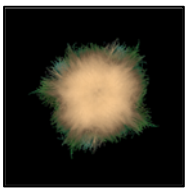
[18] Alpha Matting Evaluation Website, http://www.alphamatting.com/index.html.

[19] 越後谷勇介，藤本忠博，微細な物体輪郭を再現する半透明境界拡張テクスチャ合成法，NICOGRAPH 2016，C-5，pp. 1-8，2016.

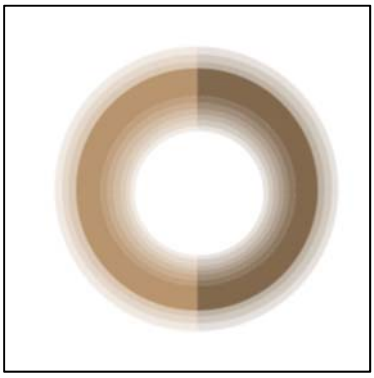


サンプル画像（毛）
400x400 画素

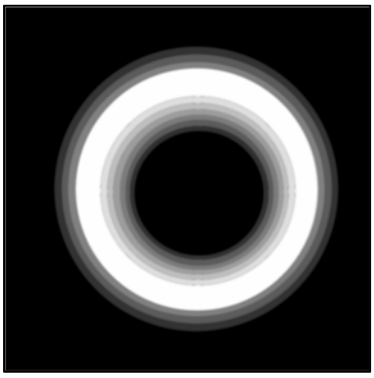
αマット画像



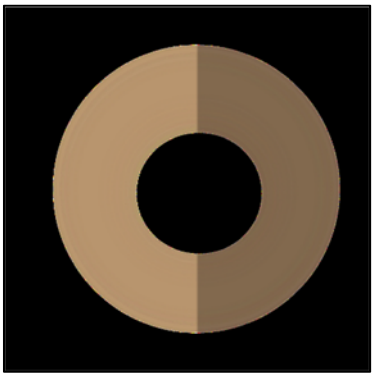
色画像



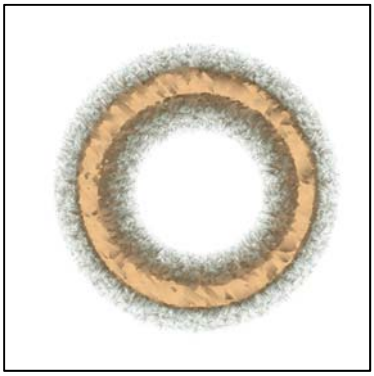
ペイント画像
800x800 画素



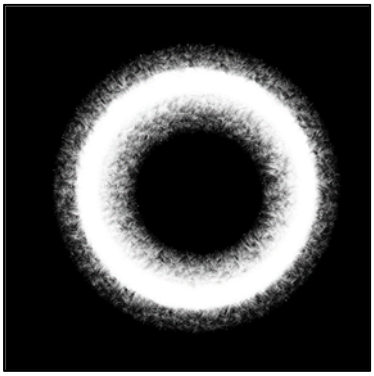
αマット画像



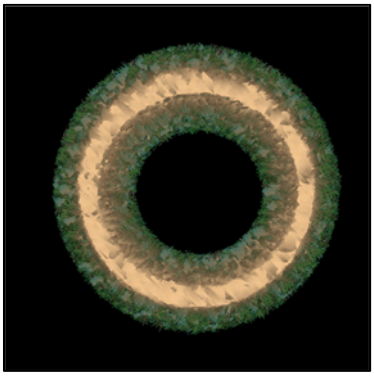
色画像



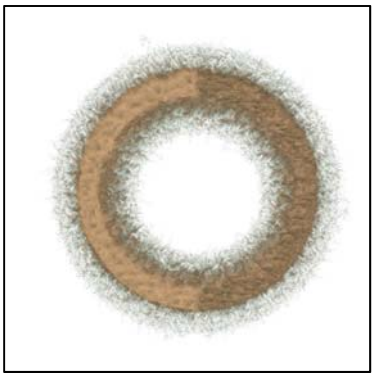
出力画像（$w_{\alpha c}$ ＝1.00）13.37 秒
輪郭再現率：E_c=0.180



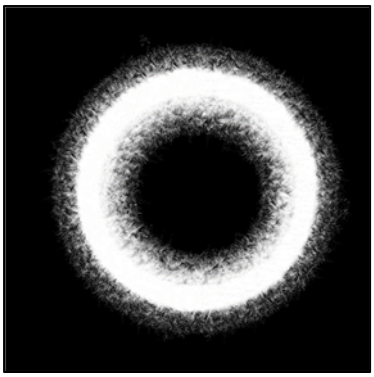
αマット画像



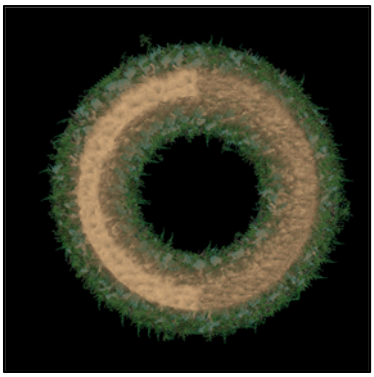
色画像



出力画像（$w_{\alpha c}$ ＝0.90）15.03 秒
輪郭再現率：E_c=0.789

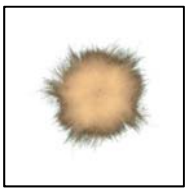


αマット画像



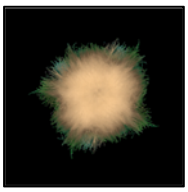
色画像

図 7. 係数 $w_{\alpha c}$ の比較実験 1
近隣画素群サイズ：9x9 画素　k-候補数：k＝3　重み：β_P ＝0.50

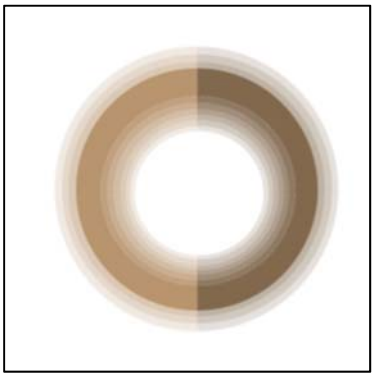


サンプル画像（毛）
400x400 画素

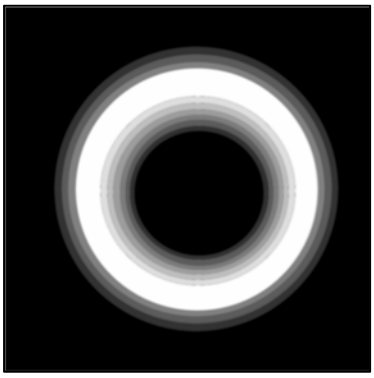
αマット画像



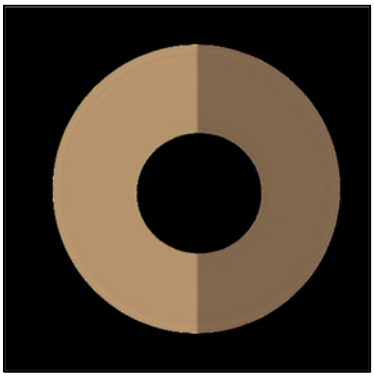
色画像



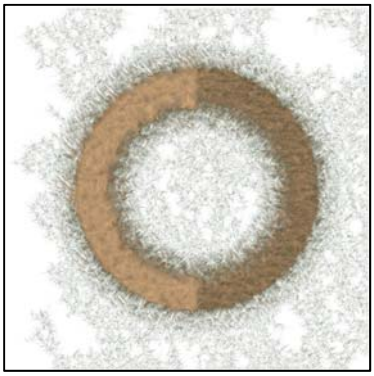
ペイント画像
800x800 画素



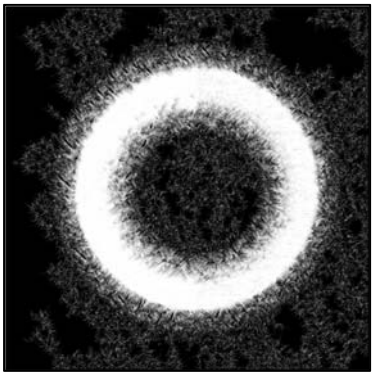
αマット画像



色画像

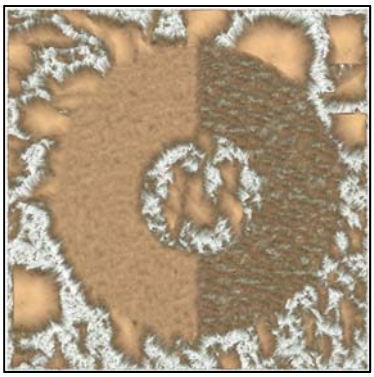


出力画像（$w_{\alpha c}$ ＝0.50）30.54 秒
輪郭再現率：E_c=0.839

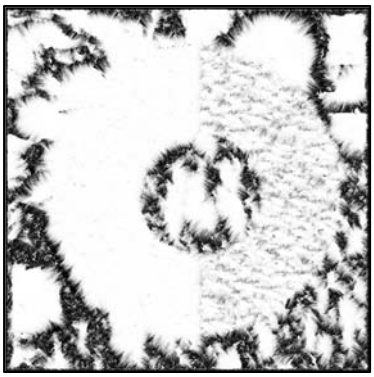


αマット画像

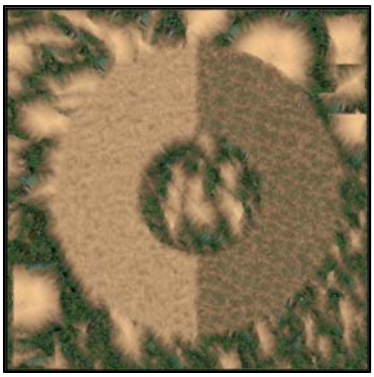
色画像



出力画像（$w_{\alpha c}$ ＝0.00）33.26 秒
輪郭再現率：E_c=0.968



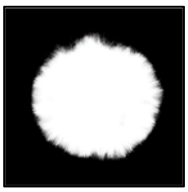
αマット画像



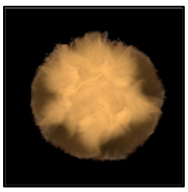
色画像

図 7（続き）．係数 $w_{\alpha c}$ の比較実験 1
近隣画素群サイズ：9x9 画素　k-候補数：k＝3　重み：β_P ＝0.50

サンプル画像（毛）
400x400 画素



αマット画像



色画像

ペイント画像
800x800 画素

αマット画像

色画像

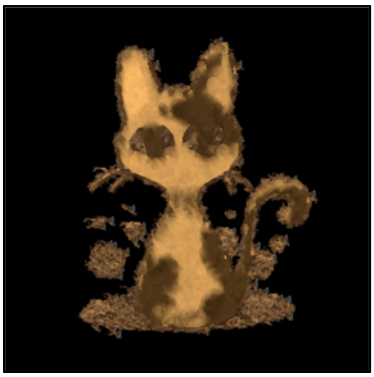
出力画像（$w_{\alpha c}$ ＝1.00）9.21 秒
輪郭再現率：E_c=0.271

αマット画像

色画像

出力画像（$w_{\alpha c}$ ＝0.90）10.00 秒
輪郭再現率：E_c=0.715

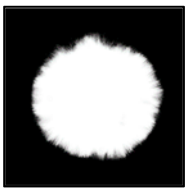
αマット画像



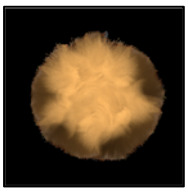
色画像

図 8. 係数 $w_{\alpha c}$ の比較実験 2
近隣画素群サイズ：9x9 画素　k-候補数：k＝3　重み：β_P ＝0.50

サンプル画像（毛）
400x400 画素



αマット画像

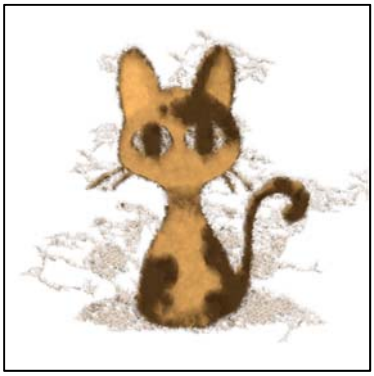


色画像

ペイント画像
800x800 画素

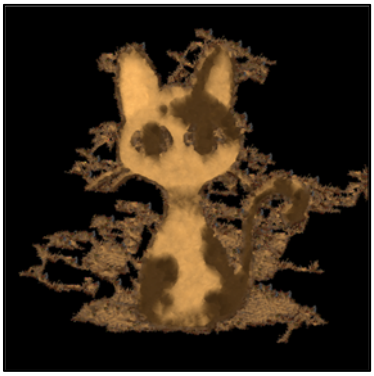
αマット画像

色画像

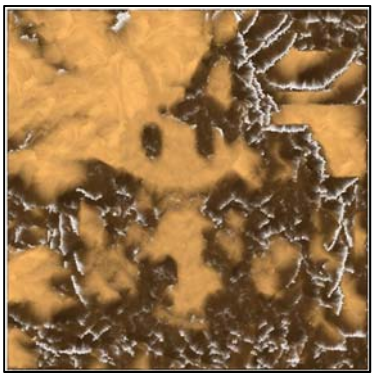


出力画像（$w_{\alpha c}$ ＝0.50）14.16 秒
輪郭再現率：E_c=0.837

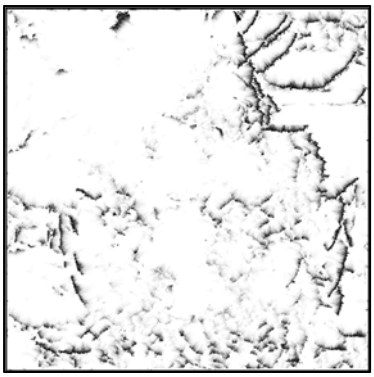
αマット画像



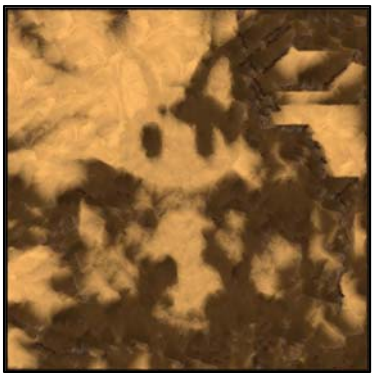
色画像



出力画像（$w_{\alpha c}$ ＝0.00）33.12 秒
輪郭再現率：E_c=1.00

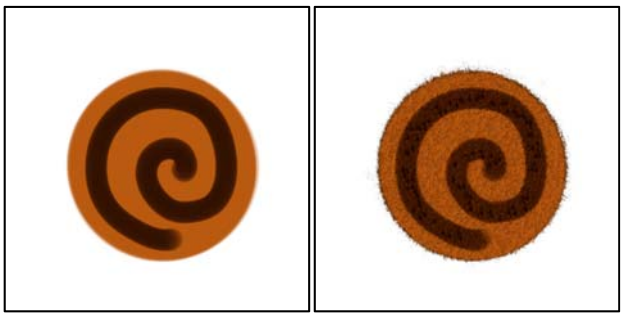


αマット画像

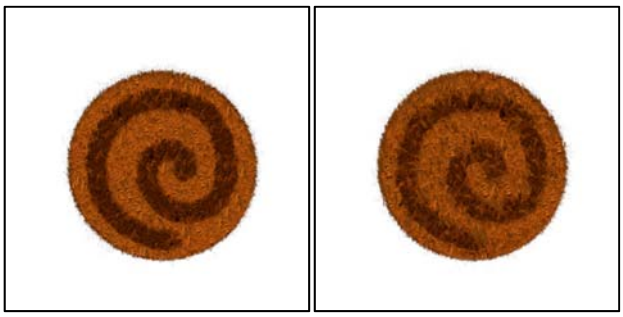


色画像

図 8（続き）．係数 $w_{\alpha c}$ の比較実験 2
近隣画素群サイズ：9x9 画素　k-候補数：k＝3　重み：β_P ＝0.50



ペイント画像　　出力画像（β_P ＝1.00）16.80 秒
輪郭再現率：E_c=0.702



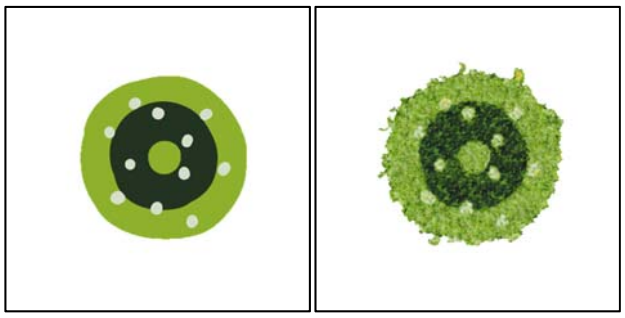
出力画像（β_P ＝0.10）15.96 秒　出力画像（β_P ＝0.05）16.10 秒
輪郭再現率：E_c=0.667　輪郭再現率：E_c=0.750

出力画像（β_P ＝1.00）拡大図

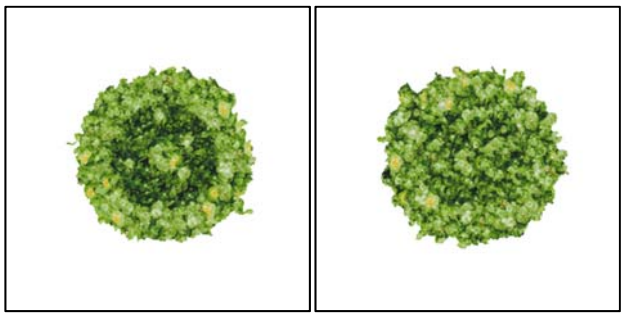
出力画像（β_P ＝0.05）拡大図

図 9.　重み β_P の比較実験 1
サンプル画像：400x400 画素
ペイント画像：800x800 画素
近隣画素群サイズ：9x9 画素
k-候補数：k＝11
係数：$w_{\alpha c}$ ＝0.80

サンプル画像
（毛）



ペイント画像　　出力画像（β_P ＝1.00）0.77 秒
輪郭再現率：E_c=0.712



出力画像（β_P ＝0.20）0.74 秒　出力画像（β_P ＝0.10）0.78 秒
輪郭再現率：E_c=0.745　輪郭再現率：E_c=0.771

出力画像（β_P ＝1.00）拡大図

出力画像（β_P ＝0.10）拡大図

図 10.　重み β_P の比較実験 2
サンプル画像：200x200 画素
ペイント画像：400x400 画素
近隣画素群サイズ：7x7 画素
k-候補数：k＝3
係数：$w_{\alpha c}$ ＝0.20

サンプル画像
（草と花）

越後谷 勇介

2015年岩手大学工学部卒業．2017年同大学大学院工学研究科博士前期課程修了．在学中は知的映像編集に関する研究に従事．現在，株式会社セガ・インタラクティブ勤務．

藤本 忠博

1990年慶應義塾大学理工学部卒業．1992年同大学大学院理工学研究科前期博士課程修了．同年(株)三菱総合研究所入社．1995年同研究所退職．同年慶應義塾大学大学院理工学研究科後期博士課程入学．1999年同大学院単位取得退学．同年岩手大学助手．2000年博士(工学)(慶應義塾大学)．2002年岩手大学講師．2005年助教授．2007年准教授．2016年教授．コンピュータグラフィックス，コンピュータビジョン，画像処理に関する研究に従事．ACM，IEEE，芸術科学会，他会員．