

NAIST-IS-MT1251066

修士論文

パターン類似度を用いたエネルギー最小化による
事例参照型画風変換

常松 佑太

2015年3月12日

奈良先端科学技術大学院大学
情報科学研究科 情報科学専攻

本論文は奈良先端科学技術大学院大学情報科学研究科に
修士(工学) 授与の要件として提出した修士論文である。

常松 佑太

審査委員：

横矢 直和 教授	(主指導教員)
加藤 博一 教授	(副指導教員)
佐藤 智和 准教授	(副指導教員)
河合 紀彦 助教	(副指導教員)

パターン類似度を用いたエネルギー最小化による 事例参照型画風変換*

常松 佑太

内容梗概

近年、油絵や鉛筆画など芸術的な表現を持つ画像を生成する非写実的画像生成 (NPR: Non-Photorealistic Rendering) が盛んに研究されている。特に、実写画像を非写実的な画像へと変換する手法は NPR の一分野として研究されており、これらはフィルタベース型アプローチと事例参照型アプローチに大別される。フィルタベース型アプローチは、平滑化フィルタやエッジ強調フィルタなどを利用し、デッサン調やパステル調などの特徴を再現する。このアプローチでは、画風変換を行う原画像を対象に、ユーザがフィルタのパラメータを画像を見ながら試行錯誤的に調整し、理想とする生成結果に近づけていくプロセスが必要である。事例参照型アプローチでは、参照画像を用いてその特徴を反映させるように原画像を変換する。入力として、画風変換を行う原画像とは別に、ユーザが希望する特徴を持つ参照画像も必要となるが、フィルタベース型と比べて調整すべきパラメータが少なく、ユーザの負担は小さい。本論文では、事例参照型アプローチに着目し、実写風景画像を原画像、絵画を参照画像とした新たな事例参照型画風変換システムを提案する。具体的には、まず、原画像と参照画像を入力し、ユーザが必要と判断した場合は原画像を参照画像の輝度値に合わせるように輝度調整する。次に、生成画像と原画像間の輝度差、および生成画像と参照画像間のパターン相違度の双方を利用したエネルギーを新たに定義し、これを最小化することで参照画像風の画像生成を行う。提案手法は、局所的なテクスチャを逐次的に合成する

*奈良先端科学技術大学院大学 情報科学研究科 情報科学専攻 修士論文, NAIST-IS-MT1251066, 2015年3月12日.

従来手法と異なり，エネルギー最小化によりテクスチャのずれが生じにくい．また，原画像と参照画像の輝度が大きく異なる場合でも，事前に原画像の輝度調整を行うことで，生成画像と参照画像間の不適切なパターンの対応付けを防ぎアーチファクトの生成を抑制する．最後に，提案手法において輝度調整を行わない場合および行う場合の画像生成実験を行い，提案手法の有効性を示す．

キーワード

Non-Photorealistic Rendering (NPR), Painterly Rendering, 画風変換, 事例参照, エネルギー最小化, 粗密法

Example-based Conversion of Painting Style by Minimizing Energy Based on Pattern Similarities *

Yuta Tsunematsu

Abstract

Recently, Non-Photorealistic rendering (NPR), which creates images with artistic styles of paintings such as oil paintings, pencil drawing, and so on, has been investigated. Especially, methods of converting photographed images into non-photorealistic ones have been intensively investigated in the field of NPR. These methods can be categorized into filter-based and example-based approaches. The filter-based approach reproduces painting features by applying filters such as smoothing filters and unsharp masking to a source image. Although this approach requires only a source image, users have to adjust many parameters in order to obtain their desired output. The example-based approach uses reference images and converts a source image to an output image with features of the reference images. Although this approach requires reference images with user's desirable features in addition to a source image, the number of parameters which users have to adjust is much smaller than that for the filter-based approach. In this study, we focus on the example-based approach and propose a novel example-based system which converts a photographed image into a pictorial style using a reference pictorial image. Specifically, we first input a source image and a reference image, and adjust the intensities in the source image according

*Master's Thesis, Department of Information Science, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT1251066, March 12, 2015.

to those in the reference image if a user desires this process. Next, we define a new energy function which reflects the luminance difference between an output image and a source image, and the pattern dissimilarity between an output image and a source image. The conversion is then achieved by minimizing the energy function. The proposed method generates continuous textures by minimizing the energy function unlike conventional methods that successively copy local textures to an output image. We also prevent non-appropriate correspondences between the output and reference images, which causes artifacts in the output image, by adjusting the intensities in the source image. In order to investigate the validity of the proposed method, we have carried out an experiment of generating output images with and without intensity control.

Keywords:

Non-Photorealistic Rendering (NPR), conversion of painting style, example-based, energy optimization, coarse to fine

目次

1. はじめに	1
2. Painterly Rendering の関連研究	3
2.1 フィルタベース型画風変換手法	3
2.2 事例参照型画風変換手法	8
2.3 本研究の位置づけ	17
3. エネルギー最小化による事例参照型画風変換	18
3.1 提案手法の概要	18
3.2 画像の輝度調整	18
3.3 パターン類似度を用いたエネルギー関数の定義	20
3.4 エネルギー最小化による参照事例風画像生成	21
3.4.1 パッチマッチによる類似パターンの探索	21
3.4.2 生成画像の画素値の更新	21
3.4.3 粗密法によるテクスチャ反映処理の効率化	23
4. 実験	25
4.1 輝度調整を行わない場合の画風変換結果と考察	25
4.1.1 画風変換の結果について	25
4.1.2 処理時間およびエネルギーの収束について	30
4.2 輝度調整を行った場合の画風変換結果と考察	68
4.3 提案手法と従来手法との生成結果の比較	82
5. まとめ	92
謝辞	93
参考文献	94

目 次

1	Haeberli ら [1] の生成結果	4
2	Hertzmann [2] の生成結果例	5
3	Kovács ら [4] の生成結果例	7
4	Zeng ら [6] の生成結果例	7
5	Hertzmann ら [10] の参照画像組と生成結果例	8
6	Efros ら [11] の画風変換例	10
7	Ashikhmin らの手法 [13] の生成結果例	11
8	Hara ら [15] の生成結果	12
9	Lee ら [16] の生成結果	13
10	Chang ら [18] の生成結果と Hertzmann ら [10] の応用	15
11	市野らの手法 [20] の生成結果	16
12	提案手法の流れ	19
13	エネルギー算出における画素の関係	22
14	多重スケールによるテクスチャパターンの違い	24
15	原画像群	26
16	参照画像群	27
17	原画像 (A1) と参照画像 (B1) の生成結果	32
18	原画像 (A1) と参照画像 (B2) の生成結果	33
19	原画像 (A1) と参照画像 (B3) の生成結果	34
20	原画像 (A1) と参照画像 (B4) の生成結果	35
21	原画像 (A1) と参照画像 (B5) の生成結果	36
22	原画像 (A1) と参照画像 (B6) の生成結果	37
23	原画像 (A2) と参照画像 (B1) の生成結果	38
24	原画像 (A2) と参照画像 (B2) の生成結果	39
25	原画像 (A2) と参照画像 (B3) の生成結果	40
26	原画像 (A2) と参照画像 (B4) の生成結果	41
27	原画像 (A2) と参照画像 (B5) の生成結果	42
28	原画像 (A2) と参照画像 (B6) の生成結果	43

29	原画像 (A3) と参照画像 (B1) の生成結果	44
30	原画像 (A3) と参照画像 (B2) の生成結果	45
31	原画像 (A3) と参照画像 (B3) の生成結果	46
32	原画像 (A3) と参照画像 (B4) の生成結果	47
33	原画像 (A3) と参照画像 (B5) の生成結果	48
34	原画像 (A3) と参照画像 (B6) の生成結果	49
35	原画像 (A4) と参照画像 (B1) の生成結果	50
36	原画像 (A4) と参照画像 (B2) の生成結果	51
37	原画像 (A4) と参照画像 (B3) の生成結果	52
38	原画像 (A4) と参照画像 (B4) の生成結果	53
39	原画像 (A4) と参照画像 (B5) の生成結果	54
40	原画像 (A4) と参照画像 (B6) の生成結果	55
41	原画像 (A1) に対する画風変換処理のエネルギー変化 (a)	56
42	原画像 (A1) に対する画風変換処理のエネルギー変化 (b)	57
43	原画像 (A1) に対する画風変換処理のエネルギー変化 (c)	58
44	原画像 (A2) に対する画風変換処理のエネルギー変化 (a)	59
45	原画像 (A2) に対する画風変換処理のエネルギー変化 (b)	60
46	原画像 (A2) に対する画風変換処理のエネルギー変化 (c)	61
47	原画像 (A3) に対する画風変換処理のエネルギー変化 (a)	62
48	原画像 (A3) に対する画風変換処理のエネルギー変化 (b)	63
49	原画像 (A3) に対する画風変換処理のエネルギー変化 (c)	64
50	原画像 (A4) に対する画風変換処理のエネルギー変化 (a)	65
51	原画像 (A4) に対する画風変換処理のエネルギー変化 (b)	66
52	原画像 (A4) に対する画風変換処理のエネルギー変化 (c)	67
53	原画像 (A1) と参照画像 (B1) の生成結果	70
54	原画像 (A1) と参照画像 (B2) の生成結果	70
55	原画像 (A1) と参照画像 (B3) の生成結果	71
56	原画像 (A1) と参照画像 (B4) の生成結果	71
57	原画像 (A1) と参照画像 (B5) の生成結果	72

58	原画像 (A1) と参照画像 (B6) の生成結果	72
59	原画像 (A2) と参照画像 (B1) の生成結果	73
60	原画像 (A2) と参照画像 (B2) の生成結果	73
61	原画像 (A2) と参照画像 (B3) の生成結果	74
62	原画像 (A2) と参照画像 (B4) の生成結果	74
63	原画像 (A2) と参照画像 (B5) の生成結果	75
64	原画像 (A2) と参照画像 (B6) の生成結果	75
65	原画像 (A3) と参照画像 (B1) の生成結果	76
66	原画像 (A3) と参照画像 (B2) の生成結果	76
67	原画像 (A3) と参照画像 (B3) の生成結果	77
68	原画像 (A3) と参照画像 (B4) の生成結果	77
69	原画像 (A3) と参照画像 (B5) の生成結果	78
70	原画像 (A3) と参照画像 (B6) の生成結果	78
71	原画像 (A4) と参照画像 (B1) の生成結果	79
72	原画像 (A4) と参照画像 (B2) の生成結果	79
73	原画像 (A4) と参照画像 (B3) の生成結果	80
74	原画像 (A4) と参照画像 (B4) の生成結果	80
75	原画像 (A4) と参照画像 (B5) の生成結果	81
76	原画像 (A4) と参照画像 (B6) の生成結果	81
77	Hertzmann ら [10] との比較 (A)	85
78	Hertzmann ら [10] との比較 (B)	86
79	Efros ら [11] との比較	87
80	Lee ら [16] との比較 (A)	88
81	Lee ら [16] との比較 (B)	89
82	Lee ら [16] との比較 (C)	90
83	市野ら [20] との比較	91

表 目 次

1	物体の区分け [6]	6
2	入力した画像の解像度による処理時間の比較	31

1. はじめに

近年のコンピュータグラフィックス (CG) の表現は目覚ましい発展を遂げている。特に現実世界の表現に近い写実性の高いグラフィックスを生成する Photorealistic Rendering (PR) は、重点的に研究されており、グラフィックスに関するハードウェアの高性能化、ツールが充実したこともあり、今では映画やビデオゲームの特殊効果・視覚的演出などに多く利用されている。このように、映像や画像を生成する際に PR 技術はなくてはならないものとなっているが、一方でハードウェアやツールは以前よりも比較的安価で入手することが可能となり、誰もが一定レベルの写実的表現ができるようになったことで、映像表現の差別化が難しくなったという側面もある。

このため、PR と対する形で、Non-Photorealistic Rendering (NPR) と呼ばれる、油彩画や水彩画などの絵画風表現や、イラスト、漫画、墨絵等の人間の手による創作美術の表現を CG によって実現する技術が研究されるようになった。これにより、同じ題材を扱った作品でも、画風や製作者によって大きく差が出るため表現のバリエーションが豊富になり、表現の差別化が容易となる。

このような NPR の研究の一分野として、Painterly Rendering と呼ばれる分野が存在する。この分野では、実写画像を入力して、これを非写実的な画像へと変換する手法が研究されており、これらはフィルタベース型アプローチと事例参照型アプローチに大別される。フィルタベース型アプローチでは、平滑化フィルタやエッジ強調フィルタなどの画像処理フィルタを応用し、水彩画やパステル画など目的の表現の特徴を再現する。これは Adobe 社の Photoshop を代表とするフォトタッチツールなどが採用する手法であり、変換するための原画像のみを入力とすればよいが、それぞれの画風の一般的な特徴を再現するため、ユーザが理想とする表現を再現する場合には、ユーザは画像を見ながら多くのパラメータを試行錯誤的に調整し、目的の表現に近づけていくプロセスが必要である。一方、事例参照型アプローチでは、参照画像を用いてその特徴を反映するように原画像を変換する。入力には原画像のほかにユーザが理想とする特徴を持つ参照画像が必要になるが、参照に応じて画像が生成されるため、フィルタベース型と比べて調整すべきパラメータは少なく、ユーザの負担も小さくなり容易に多様な画像変換

を行うことができる。しかし、手法によっては参照画像の画風をが限定される。また、限定されない場合も画風ごとに事前に試行錯誤的な調整が必要になる場合がある。また、参照画像の類似テクスチャを逐次的にコピーする手法では、パッチの貼り合わせの境界でテクスチャのずれが生じる問題もある。

上記のことを踏まえて、本研究では事例参照型アプローチの画像生成の容易さと表現力に着目し、実写風景画像を原画像、絵画を参照画像とした新たな事例参照型画風変換システムを提案する。具体的には、まず、原画像と参照画像を入力し、ユーザが必要と判断した場合は原画像を参照画像の輝度値に合わせるよう輝度調整する。次に、生成画像と原画像間の輝度差を用いた画像全体の相違度、および生成画像と参照画像間の局所的なパターン相違度の双方を利用したエネルギーを新たに定義し、これを最小化することで画像生成を行う。これにより、参照画像の画風を限定せずに反映することができ、また全体最適化によりパッチの貼り合わせによる不自然な境界は発生しない。また、原画像と参照画像間の輝度の差が大きく異なる場合には、事前に参照画像と原画像の輝度ヒストグラムを用いた輝度調整を行う。

本論文では、2章で従来の Painterly Rendering の2つのアプローチであるフィルタベース型アプローチと事例参照型アプローチについて紹介し、本研究の位置づけと方針を述べる。3章では、原画像と参照画像間での輝度調整と提案手法が用いるエネルギー関数の定義とその効果、およびどのようにテクスチャの生成を行うかを述べる。4章では、提案手法による画像生成を行い、参照画像の違いやパラメータ、輝度調整の有無による生成結果をまとめ、その考察を行う。そして5章でまとめおよび今後の課題について述べる。

2. Painterly Renderingの関連研究

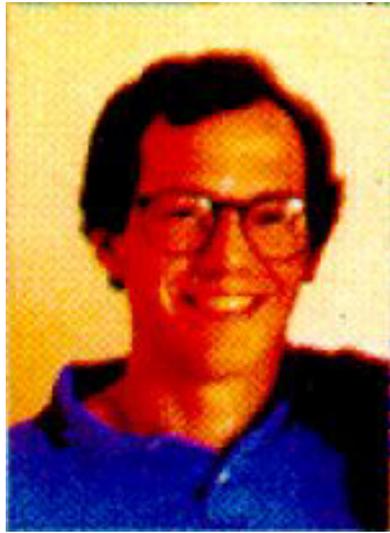
本章では Painterly Rendering の2つのアプローチであるフィルタベース型の画風変換手法と事例参照型の画風変換手法について述べる。まず、フィルタベース型アプローチの従来手法について述べ、続いて本研究で用いる事例参照型アプローチの従来手法について述べる。最後に本研究の位置づけを述べる。

2.1 フィルタベース型画風変換手法

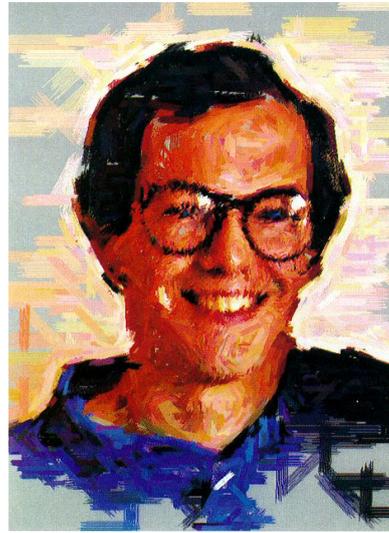
フィルタベースの手法は油彩風、水彩風から鉛筆デッサン、など多岐に渡って研究されている。特にフィルタベース型のアプローチとして、Stroke-Based Rendering (SBR) がある。これは、実際の絵画やデッサンは複数のブラシのストロークの集合であるとみなせることから、様々な形状や色の複数のブラシストロークを生成していくことで画像生成を行うアプローチである。Haerberli [1]の手法は、キャンバス上でユーザが逐一ブラシストロークを指示して画像を変換する手法であり、フォトタッチツールが採用する代表的な手法である。原画像に対してストロークの位置や方向、ブラシ形状、ブラシサイズをマウスなどで指定することで、それに応じたストロークを生成する。この際、ストロークの色は原画像で指定された位置の色情報を用いる。これを繰り返すことによって原画像を変換する。

Haerberli ら [1] はストロークの生成において、その位置と方向を原画像の輝度勾配を用いて自動的に決定する手法も提案した。これにより、同一サイズでかつ同一形状のストロークに限るが、原画像のエッジに合わせて自動生成することが可能となる。この手法によって生成された画像を図1に示す。

Hertzmann [2] は、初めにテクスチャの粗い部分から大きなストロークを用いて描き、徐々に細かい部分を小さなストロークで描くという実際の絵画制作の手順に基づき、原画像のテクスチャの細かさに応じてストロークの形状や長さを変化させる手法を提案した。ストロークのブラシサイズに基づき原画像に対してガウシアンフィルタをかけることで画像レイヤーを作成し、各レイヤーを対応するブラシサイズで塗りつぶしていくことでブラシサイズの変化を自動化している。また、ストロークを生成する際、キーポイントを追加していくことでその長さ



(a) 原画像



(b) 生成画像

図 1 Haeberli ら [1] の生成結果

形状を変化させる。キーポイントの候補はグリッド状に配置されており、ひとつのキーポイントの輝度勾配とブラシサイズを利用して次のキーポイントの位置を決定する。次のキーポイントの色と現在生成しているストロークの色の差が閾値以上になるまでこの処理を繰り返すことで長さや形状の変化を自動化した。この手法では、ブラシサイズやストロークの最大長などのパラメータを調整することで、様々な画風を再現することができる。その例を図2に示す。また、Hertzmann [3] は上記の手法を改良し、未処理の画素数に基づく項と画像中のストローク数に基づく項からなるエネルギー関数を最小化することでストロークの配置を最適化する手法を提案した。

Kovács ら [4] の手法では、原画像中のオブジェクトに着目し、オブジェクトのエッジや稜線からストロークの方向と位置、ブラシサイズを自動的に調整する。これにより、テクスチャの弱い領域でストローク方向がノイズの影響を受けやすく歪んでしまうという従来の問題を解決している。あらかじめ原画像に複数のスケールのガウシアンフィルターをかけた画像を用意し、Lindeberg [5] の手法を用いて原画像の重み付きエッジマップおよび重み付き稜線マップを作成する。稜線



(a) 原画像



(b) 印象派風



(c) 表現主義風



(d) 水彩風



(e) 点描画風

図 2 Hertzmann [2] の生成結果例

マップを構成する点がストローク位置，稜線マップの重みがブラシサイズとなり，各ストローク位置からもっとも近いエッジの近傍点との勾配がストローク方向となる．この手法の生成結果を図3で示す．

オブジェクトに注目するアプローチとしては，Zengら [6]の手法がある．これは原画像中の物体ごとにストロークの要素を決定し生成する手法である．この手法では，まず Tu [7] [8] や Guo [9]の手法を用いて原画像中の物体を抽出し，それぞれにユーザがラベル付けを行う．このラベル付けは表1に従って行う．次に，物体ごとにストロークの場所と方向を決定する．ストロークの流れは物体ごとに anisotropic diffusion を用いて流れ場を作成し，その方向にストロークの方向を合わせる．また，印象派の絵画のストロークを物体別にサンプリングした brush dictionary を利用することで，それぞれの物体に適したブラシ形状を選択することができる．この手法の生成結果を図4で示す．

表1 物体の区分け [6]

顔・肌	髪	服	空・雲
水面	波しぶき	山	道・建物
岩	地面	木・プラスチック	金属
花・果物	草	葉	木の幹・枝
背景	その他		



図 3 Kovács ら [4] の生成結果例



図 4 Zeng ら [6] の生成結果例

2.2 事例参照型画風変換手法

Hertzmann ら [10] は、参照画像と参照画像を任意の画像処理によって変換した画像のペアから学習したデータを用いて、任意の画像を変換する手法を提案した。この手法ではユーザは参照画像と参照画像に対して好みの画風変換を行った画像のペアが必要となる。参照画像のペアからストロークの形状の変化を学習し、学習したストローク形状を原画像へと反映する。この手法の生成結果を図5に示す。



(a) 参照画像



(b) 変換済み参照画像



(c) 原画像



(d) 生成結果

図5 Hertzmann ら [10] の参照画像組と生成結果例

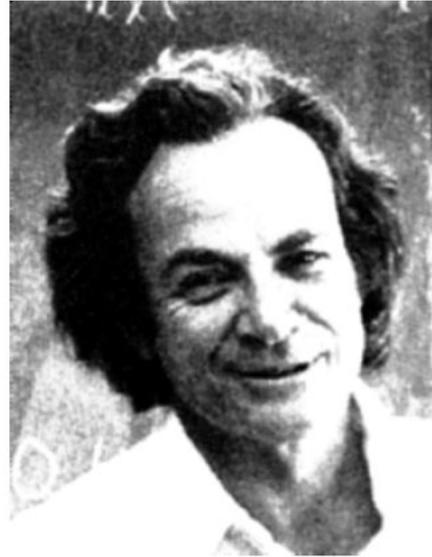
Efros ら [11] は、一枚のテクスチャ画像を任意のサイズに違和感なく変換する texture synthesis の手法を提案し、それを画風変換に応用した。texture synthesis の方法は、まず与えられたテクスチャパッチからランダムにブロックを切り抜き

ラスタスキャン方式でパッチをオーバーラップさせながらつなげていく。この時、選ばれるブロックは隣り合うブロックとある程度連続的になるまで選ばれる。続いて、Davis [12] の手法で用いている方法と同様の方法でブロック間の最小誤差境界を探す。全てのブロック間の誤差の和をエネルギーとして、これを最小化することでテクスチャを生成する。画風変換は、上記のエネルギーの項に correspondence map を用いた原画像と参照画像間の類似度の項を加え、そのエネルギーを最小化することで行う (図 6)。

Ashikhmin [13] の手法は、自身が提案した texture synthesis の手法 [14] を応用した画風変換手法である。Ashikhmin が提案した texture synthesis は、ラスタスキャン方式で生成画像の画素を生成する。生成する画素に最も類似すると考えられる画素値を持つ点を、L-Shaped neighborhood の範囲で SSD を計算して原画像中から探索する。この時、既に見つけた対応点についてはリストとして保持することで、高速に新規の対応点を探索し、原画像の輝度値を生成画像に反映する。この手法を事例参照型画風変換にそのまま応用すると、生成画像の質を上げるために何十回とプロセスを繰り返す必要があること、対応点リストの偏りによって水平方向の不自然なエッジが出やすいという問題がある。そこで Ashikhmin の画風変換手法では、新規の対応点の候補を対応点リストからのみ探すだけでなく、ランダムに点を 1 つ選び、それを対応点の候補として対応点を探すようにすることでこの問題を解決した。この手法の生成結果を図 7 に示す。



source texture



target image



correspondence maps



texture transfer result

図 6 Efros ら [11] の画風変換例



図 7 Ashikhmin らの手法 [13] の生成結果例

Hara ら [15] は、テクスチャの細かさに応じてパッチのサイズを変えて分割し、原画像のエッジや細かい構造を維持したまま参照画像のテクスチャを反映させる手法を提案した。画像の分割は原画像のエッジを抽出し、それをもとに作成した距離画像を用いる。まず画像全体をそれぞれのパッチが一部重複するように一定のサイズのパッチで分割する。あるパッチにエッジが集中する場合、距離画像のパッチ内の平均輝度は高くなる。その値がユーザが指定した閾値より高ければさらに分割を繰り返す。こうして分割したパッチでサイズが大きいものから順に処理をしていく。画風変換処理はエネルギーを最小化することで行う。このエネルギーは隣り合うパッチが同じパターンになることを防ぐ項、隣合うパッチの局所的な連続性を保つ項、再構成した画像と追加するパッチの連続性を保つ項の3つからなる。エネルギーを直接最小化するのではなく、確率伝播法によって逐次的に解く。この手法の生成結果を図8に示す。

Lee ら [16] は、Ashikhmin ら [13] の手法に原画像の流れ場を考慮したストローク生成を加え改良した手法である。この手法ではエネルギーを最小化することで画風変換を行う。エネルギー関数は3つの項からなっており、それぞれ原画像と参照画像の画素周囲の輝度の近さを出す項、高周波成分を避けノイズが少ない部分を選ぶ項、原画像の流れ場とストローク方向を一致させるための項からなっている。原画像の流れ場はノイズなどに頑健な Kang ら [17] の手法を用いて求める。また、流れ場とストローク方向を一致させる項の重みを調整することで様々な画風の参照画像に対応できる (図9)。



(a) 原画像



(b) 参照画像



(c) 生成画像

図 8 Hara ら [15] の生成結果



(a) 原画像



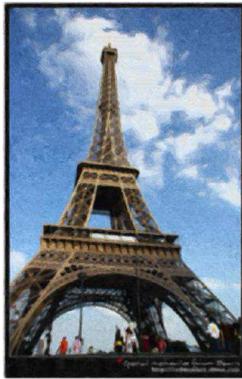
(b) 参照画像 1



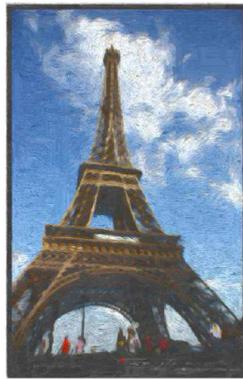
(c) 参照画像 2



(d) 参照画像 3



(e)(b) による変換結果



(f)(c) による変換結果



(g)(d) による変換結果

図 9 Lee ら [16] の生成結果

Chang ら [18] の手法は、画風変換ではないが、参照画像を用いた色彩変換を行う。この手法では、人間が変換後の画像を見て違和感を感じないように色彩を変換する。そのために、輝度や色相に応じて色を分類する。色の分類は Berlin ら [19] が提唱する基本色彩語 (Basic Color Terms: BCTs) を基にして、主観実験を行い 11 色に分類する。この分類に合わせて原画像と参照画像の両方で各画素ごとにラベリングを行い、最も類似するものと置き換えていく。基本的にラベリングは離散的になるが、原画像中で滑らかに色に変化する部分に関しては、anisolated diffusion を応用してその変化を維持するように工夫されている。また、他の画風変換手法と組み合わせることでより魅力的な画風変換が行えることが示唆されている。例として Hertzmann ら [10] の手法を利用した結果が挙げられている。この手法の生成結果を図 10 で示す。

市野らの手法 [20] では、上述の手法と異なり、参照画像のテクスチャパッチを用いて原画像の画風変換を行う。市野らは画像修復の手法と画風変換との共通点に注目し、特に河合らの手法 [21] を応用してエネルギー関数を設計し、画像生成に用いる最適なテクスチャパッチを参照画像内から探索し、逐次的に貼り合わせることで参照画像風の画像を生成する。エネルギー関数は具体的には 3 つの項からなり、原画像と参照画像の色彩の類似度を考慮する項、既に隣接したテクスチャパッチの連続性を考慮する項、参照画像中の類似したテクスチャ同士は参照画像内の近傍に存在すると仮定したテクスチャの局所性を考慮する項である。また、原画像の特徴を保ちつつ参照画像のテクスチャを反映させるために、Hara らの手法 [15] で用いられているパッチの再帰的な分割を行っている。この手法の生成結果を図 11 で示す。



(a) 原画像



(b) 参照画像



(c) 生成画像



(d) 原画像



(e) 参照画像

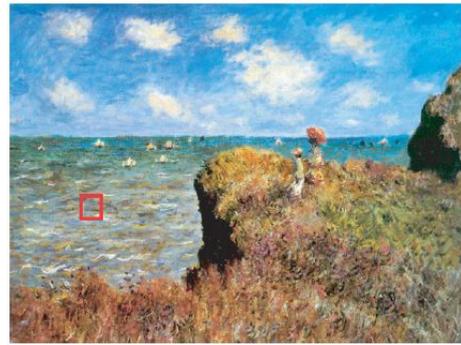


(f) Chang ら [18] + Hertzmann ら [10]

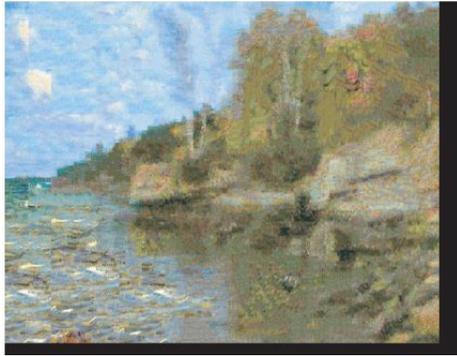
図 10 Chang ら [18] の生成結果と Hertzmann ら [10] の応用
15



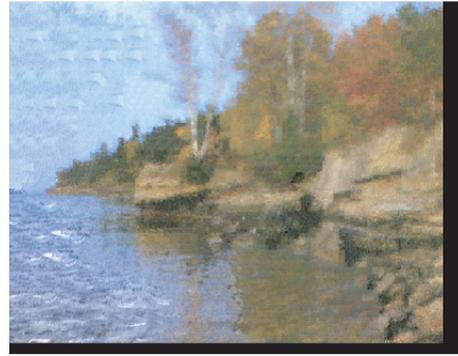
(a) 原画像



(b) 参照画像



(c) 生成結果



(d) 原画像の色彩情報を利用した結果

図 11 市野らの手法 [20] の生成結果

2.3 本研究の位置づけ

これまで概観した通り，画風変換手法としてフィルターベース型と事例参照型のアプローチが提案されている．フィルターベース型アプローチは主に特定の画風への変換に着目するため，単一の手法を用いるだけでは，変換できる画風の種類が限られているが，複数のフィルターベース型手法を組み合わせかつパラメータを試行錯誤的に調整することで，多様な画風に変換することができる．しかし，このためにはユーザに多様な知識および高度な技術が要求される．一方，事例参照型アプローチでは，対象画像に加えて参照画像が必要になるが，参照に応じて対象画像が変換されるため，誰でも容易に用いることができる．しかし，手法によっては扱える画風が限定される場合もある．また，類似パッチを探索して合成するアプローチでは，様々な画風を扱えるが，逐次的に貼り合わせるアプローチを用いているため，貼り合わせの境界でテクスチャのずれが生じる問題もある．本研究では，一般のユーザが利用することを想定して，ユーザに負担が少ない事例参照型画風変換手法に着目し，全体最適化の枠組みにより，参照画像の画風を限定せずかつテクスチャの不連続が生じない画風変換手法を提案する．なお，原画像と参照画像間で極端に輝度差がある場合に，参照画像内の不適切なテクスチャを参照してしまうことで良好な画風変換が行えない場合がある．この解決として，原画像の輝度調整を事前に行うかどうかをユーザが選択する．

3. エネルギー最小化による事例参照型画風変換

3.1 提案手法の概要

本論文では，原画像 S の変換後の画像を生成画像 O としたとき，生成画像 O と原画像 S 間の輝度の相違度および生成画像 O と参照画像間 R のパターン相違度に基づくエネルギー関数を生成画像の輝度値について最小化することで画風変換を行う。ただし，生成画像 O の解像度は原画像 S と同じであるが，参照画像 R は O と同一である必要はない。提案手法の画像生成の流れを図 12 に示す。まず，原画像と参照画像間で輝度調整を行うかどうかをユーザが選択し，輝度調整を行う場合はエネルギーの最小化の前処理として実行する。続いて生成画像に初期値を与え，参照画像内での類似パターンの探索および，生成画像の画素値の更新を繰り返すことでエネルギーを最小化し，原画像 S に対する変換後の生成画像 O を得る。また，提案手法ではスケールの異なる様々なテクスチャを反映するために，一定サイズの正方ウインドウを用いた粗密法による多重スケール処理を導入する。具体的には，上記の画像生成処理を，最も解像度の低い最上位スケールから原画像の解像度に一致する最下位スケールまで実行する。ここで，初期値を，最上位スケールでは原画像の画素値をコピーすることで与える。それ以降は，上位スケールでの生成結果を初期値として与える。以下，3.2 節で画像の輝度調整について，3.3 節で本研究で定義するエネルギー関数について，3.4 節でエネルギー最小化による生成画像の画素更新および粗密法について説明する。

3.2 画像の輝度調整

原画像と参照画像間で，輝度が大きく異なる場合は，ユーザが想定する生成画像と参照画像間の対応付けが行われず，違和感のある画像が生成される場合がある。この問題を解決するために，ユーザが必要であると判断した場合には原画像を参照画像の輝度に合わせる輝度調整処理を実行する。本処理では，まず原画像 S と参照画像 O それぞれの画素を輝度について昇順にソートし，原画像 S と参照画像 R それぞれの輝度昇順ヒストグラムを作成する。次に， R の i 番目の画素 x_i

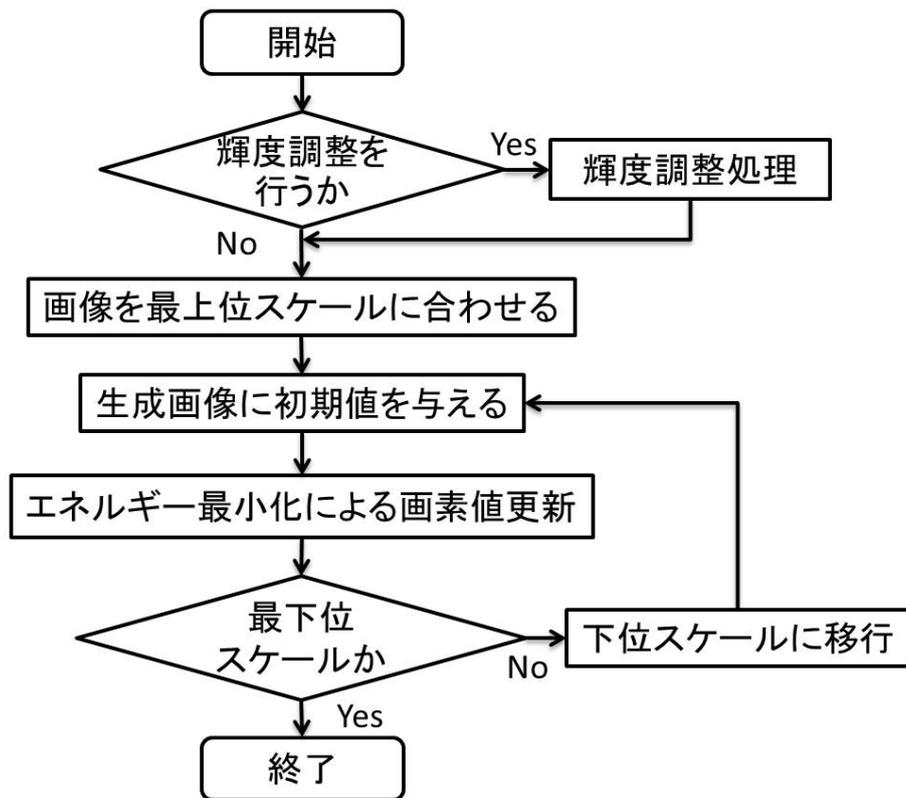


図 12 提案手法の流れ

と対応する S の画素 \mathbf{x}_j の輝度値 $I_S(\mathbf{x}_j)$ を, \mathbf{x}_i の新たな輝度値 $I_R(\mathbf{x}_i)$ とすることで, ヒストグラムの形状を一致させる. i に対応する j は, i と S の画素数 N_S と j と R の画素数 N_R の比を用いて次のように表わせる.

$$j = \left[\frac{N_S}{N_R} i \right] \quad (1)$$

ただし, $[\cdot]$ は, \cdot を超えない整数を表す.

3.3 パターン類似度を用いたエネルギー関数の定義

提案手法ではエネルギー関数を, 生成画像 O と原画像 S 間の輝度の相違度に基づく項 SSD_{OS} , および生成画像 O と参照画像 R 間のパターン相違度に基づく項 $SSSD_{OR}$ の重み付き総和として, 以下のように定義する.

$$E = SSD_{OS} + \alpha SSSD_{OR} \quad (2)$$

ただし, α は重みであり, 参照画像の色彩とテクスチャの反映度を調整する.

SSD_{OS} は生成画像 O と原画像 S 間の輝度の差に基づき以下のように定義する.

$$SSD_{OS} = \sum_{\mathbf{x}_i \in O} \{I_O(\mathbf{x}_i) - I_S(\mathbf{x}_i)\}^2 \quad (3)$$

ただし, $I_O(\mathbf{x}_i)$ および $I_S(\mathbf{x}_i)$ は, それぞれ生成画像, 原画像における画素 \mathbf{x}_i の画素値である. この項により, 生成画像のテクスチャが原画像から大きく変化することを抑制する.

$SSSD_{OR}$ は, 生成画像 O と参照画像 R 間のパターン相違度に基づき以下のように定義する.

$$SSSD_{OR} = \sum_{\mathbf{x}_i \in O} \sum_{\mathbf{p} \in W} \{I_O(\mathbf{x}_i + \mathbf{p}) - I_R(\mathbf{x}_j + \mathbf{p})\}^2 \quad (4)$$

ただし, $I_R(\mathbf{x}_j)$ は参照画像 R における画素 \mathbf{x}_j の画素値である. また, \mathbf{p} は一定サイズの正方ウィンドウ W 内の画素を示すための遷移ベクトルである. この項により, 参照画像 R 内のテクスチャを生成画像 O に反映させる. 正方ウィンドウ W のウィンドウサイズが大きくなると, 参照画像中のよりスケールの大きなテクスチャが生成画像 O に反映される.

3.4 エネルギー最小化による参照事例風画像生成

本研究では、式 (2) で定義したエネルギー E を、生成画像の輝度 $I_O(\mathbf{x}_i)$ を変数とする Greedy Algorithm の枠組みを用いて最小化する。具体的には生成画像 O と参照画像 R 間の類似パターンの組を固定することでエネルギー E を生成画像 O の各画素で独立に扱えることに着目し、

(処理 1) 類似パターン位置の探索

(処理 2) 生成画像の画素値の更新

をエネルギーが収束するまで繰り返すことで、画像全体のエネルギーを最小化する。以下では処理 1, 処理 2 について順に述べる。

3.4.1 パッチマッチによる類似パターンの探索

生成画像の各画素に対して、以下の式 (5) を満たす画素位置 $f(\mathbf{x}_i)$ を求めることでエネルギーを減少させる。

$$f(\mathbf{x}_i) = \arg \min_{\mathbf{x}_j \in R} \sum_{\mathbf{p} \in W} \{I_O(\mathbf{x}_i + \mathbf{p}) - I_R(\mathbf{x}_j + \mathbf{p})\}^2 \quad (5)$$

ただし、参照画像全体の探索による最小化は計算コストが高いため、ここでは PatchMatch [22] を用いることで、高速に近似解を探索する。

3.4.2 生成画像の画素値の更新

処理 2 では、処理 1 で求めた生成画像 O と参照画像 R 間の類似パターンの組を固定し、エネルギー E を最小化する生成画像の画素値 $I_O(\mathbf{x}_i)$ を並列に更新する。以下では、パターンの組を固定した場合の画素値 $I_O(\mathbf{x}_i)$ の算出手法について詳述する。ここではまず、エネルギー E を、生成画像の各画素での要素エネルギー $E_{\mathbf{x}_i}$ に分解する。図 13 に示すように、更新対象となる画素の位置 \mathbf{x}_i を中心とするウィンドウ W 内の任意の点を $\mathbf{x}_i + \mathbf{p}$ ($\mathbf{p} \in W$) とする。このとき、画素 \mathbf{x}_i を中心とするパターンに対して処理 1 で求まる類似パターンの位置は $f(\mathbf{x}_i + \mathbf{p})$

であり，この類似パターン上において \mathbf{x}_i と対応する画素の位置は $f(\mathbf{x}_i + \mathbf{p}) - \mathbf{p}$ となる．ここで，注目画素 \mathbf{x}_i に関する E の要素エネルギー $E_{\mathbf{x}_i}$ は，以下のよう
に表すことができる．

$$E_{\mathbf{x}_i} = \{I_O(\mathbf{x}_i) - I_S(\mathbf{x}_i)\}^2 + \alpha \sum_{\mathbf{p} \in W} \{I_O(\mathbf{x}_i) - I_R(f(\mathbf{x}_i + \mathbf{p}) - \mathbf{p})\}^2 \quad (6)$$

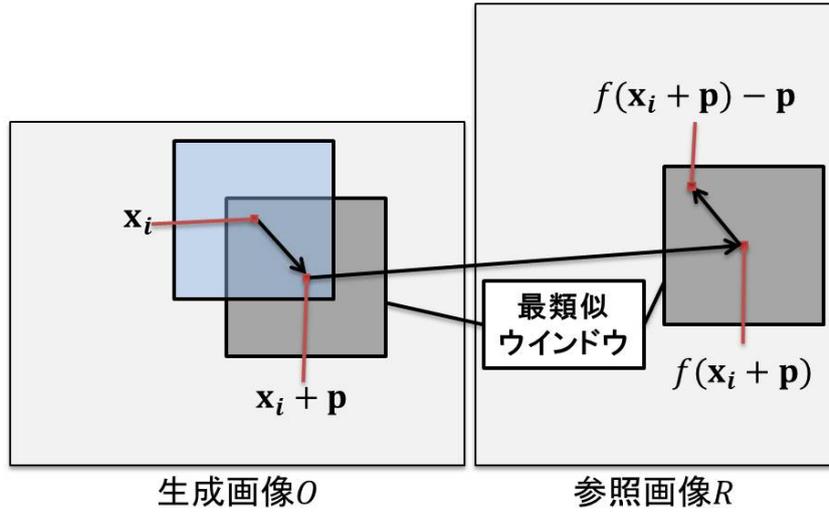


図 13 エネルギー算出における画素の関係

これによりエネルギー E は各画素での要素エネルギー $E_{\mathbf{x}_i}$ を用いて以下のよう
に表せる．

$$E = \sum_{\mathbf{x}_i \in O} E_{\mathbf{x}_i} \quad (7)$$

各要素エネルギー $E_{\mathbf{x}_i}$ は独立であるため，各要素エネルギー $E_{\mathbf{x}_i}$ を最小化するこ
とで，エネルギー E を最小化することができる．最小化する画素値 $I_O(\mathbf{x}_i)$ は，各
要素エネルギー $E_{\mathbf{x}_i}$ を画素値 $I_O(\mathbf{x}_i)$ で微分し極値を求めることで以下のよう
に算出される．

$$I_O(\mathbf{x}_i) = \frac{I_S(\mathbf{x}_i) + \alpha \sum_{\mathbf{p} \in W} I_R(f(\mathbf{x}_i + \mathbf{p}) - \mathbf{p})}{1 + \alpha N_W} \quad (8)$$

ただし， N_W はウインドウ内の画素数である．

3.4.3 粗密法によるテクスチャ反映処理の効率化

本手法では，生成されるテクスチャパターンが正方ウインドウのサイズに大きく依存し，ウインドウサイズが一定であると特定のスケールのテクスチャのみが反映される．2章で挙げた多くの研究で述べられている通り，一般的な絵画ではブラシのサイズを変更しつつ細部を描き込むため，特定のスケールのみのテクスチャパターンの反映では不十分である．そこで，図 14 で示すように画像ピラミッドを作成し，粗密法により画像を生成することで複数のスケールでのテクスチャパターンを反映する．具体的には，原画像と参照画像をそれぞれ一定スケールまで繰り返し画像サイズを2分の1ずつ縮小した画像ピラミッドを作成しておき，まず最も下位のスケールで一定サイズのウインドウを用いてエネルギー最小化処理を行う．得られた結果画像を上位スケールの初期画像として利用する．これを，ウインドウサイズを固定したまま原画像のスケールまで繰り返すことで，複数のスケールのテクスチャを再現する．また，本研究で行うエネルギー最小化は Greedy Algorithm の枠組みを利用するため，局所解に陥ることがあるが，この手法により大局解から遠い局所解に陥ることを防ぐ．さらに，各スケールの初期値として，あらかじめエネルギーの低いテクスチャを用いることができるため，処理の高速化にもつながる．

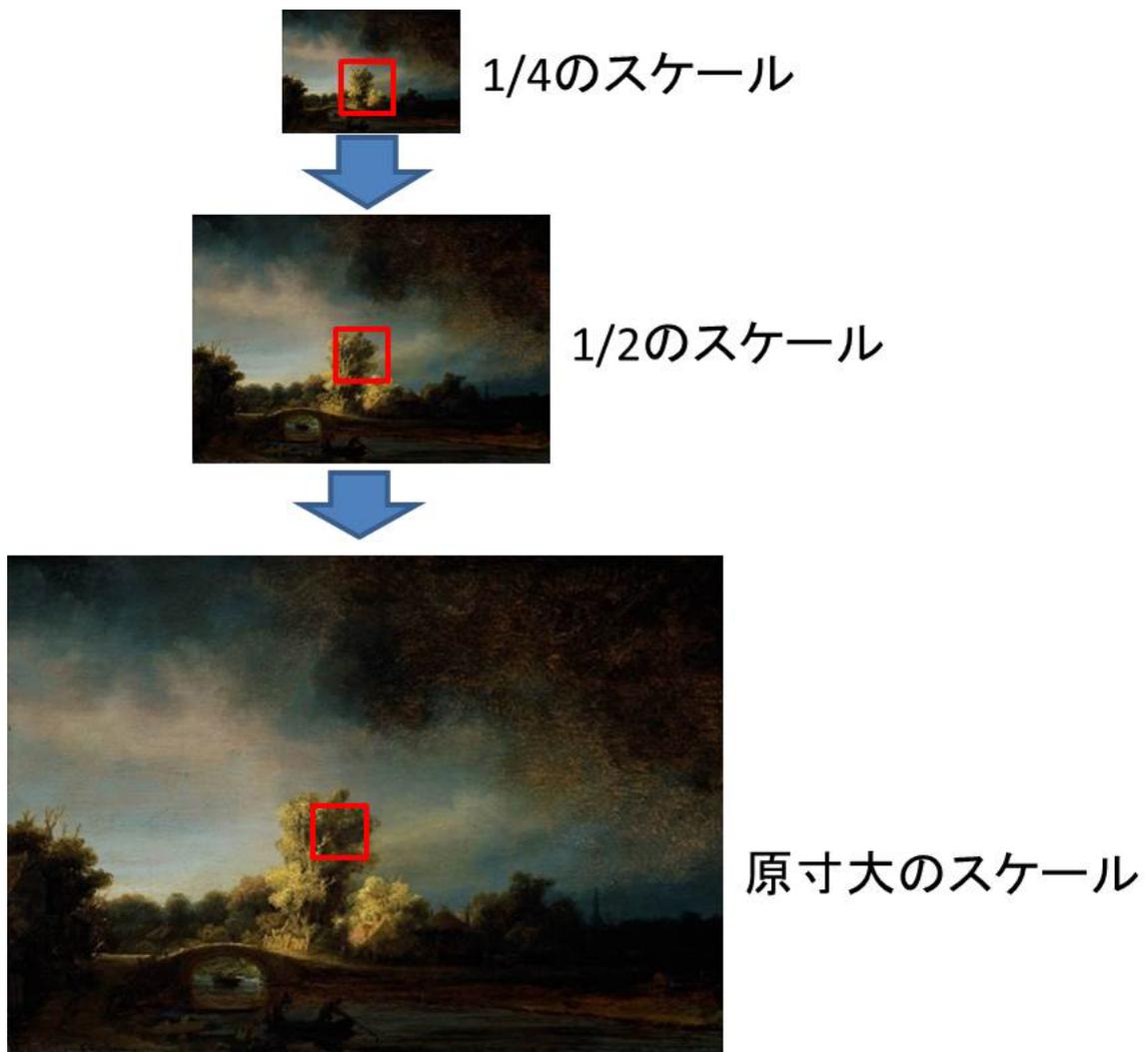


図 14 多重スケールによるテクスチャパターンの違い

4. 実験

本章では，風景写真を原画像，様々な特徴を持つ絵画画像を参照画像として，提案手法による画風変換を行った結果について考察する．本実験では，PC(CPU: Intel Core i7-3930K 3.20GHz, メモリ: 16GB) を用い，図 15 に示す原画像および図 16 に示す参照画像を用いて画風変換を行った．本実験では，エネルギーの減少率が 0.3% を下回った場合にエネルギーが収束したとみなした．また，画像間のテクスチャスケールが大きく異なることを防ぐため，原画像と参照画像の短辺が 300 画素になるように事前にリサイズした．以下，4.1 節では，事前に輝度調整を行わない場合について，パラメータを調整することによる参照画像の色情報やテクスチャの反映度の変化を検証する．また，各画像の生成に要した処理時間やエネルギーの収束について示す．4.2 節では，事前に輝度調整を行った場合に，原画像と参照画像間で極端に輝度差がある場合の参照画像のテクスチャの局所的な参照の問題が解消できるかについて検証する．4.3 節では，提案手法の生成結果と従来手法の生成結果とを比較し，提案手法の有効性を検証する．

4.1 輝度調整を行わない場合の画風変換結果と考察

本節では，輝度調整を行わない場合における原画像 (A1)(A2)(A3)(A4) と参照画像 (B1)(B2)(B3)(B4)(B5)(B6) を用いた場合の画風変換結果について述べる．さらに，画風変換に要した処理時間および，エネルギーの変化を見て，収束条件が妥当であるかを確認する．

4.1.1 画風変換の結果について

ここでは輝度調整を行わない場合の画風変換の結果について述べる．ウィンドウサイズ W を 3×3 , 5×5 の 2 段階に，重み α を 0.2, 0.5, 0.8 の 3 段階に調整し，1 つの原画像と参照画像の組に対して 6 つの生成画像を作成した．

原画像について，それぞれ以下の特徴を持つ．

- (A1) : コントラストが低く，テクスチャが少ない



(A1)



(A2)



(A3)



(A4)

图 15 原画像群



(B1)



(B2)



(B3)



(B4)



(B5)



(B6)

图 16 参照画像群

- (A2) : 風景画像で、極端に細かいテクスチャをを持たない
- (A3) : 人工物を含む風景で一部で細かいテクスチャを持つ
- (A4) : テクスチャが細かく様々な色を持つ

一方、参照画像については、それぞれ以下の特徴を持つ。

- (B1) : 特徴的で長いブラシストロークを持ち色数が少ない
- (B2) : 一般的な風景の水彩画
- (B3) : 現実に即した風景画
- (B4) : はっきりとしたブラシストロークと多くの色彩を重ねた色彩混交テクスチャを持つ
- (B5) : 暗めの色遣い
- (B6) : ブラシストロークがはっきりとしており、自然な色遣いを持つ

図 17 から図 40 に、原画像 4 種と参照画像 6 種の全ての組み合わせからなる計 24 パターンの画風変換結果を示す。まずこれらの結果の全体の傾向について述べ、次に個別の結果について考察する。

まずパラメータに関する全体の傾向として、ウインドウサイズ W が大きくなることで参照画像から反映されるテクスチャのスケールが大きくなり、原画像に存在するエッジが平滑化されやすくなる。また、重みパラメータ α が大きくなることで参照画像の色調がより強く反映される。次に、参照画像のテクスチャの違いについて、参照画像が細かいテクスチャを持つ場合、そのテクスチャが生成画像に強く反映される傾向がある。またそのような参照画像を用いた場合に重みやウインドウサイズが大きくなると、テクスチャが弱い参照画像と比較して、参照画像の影響が強く表れる傾向がある。最後に、原画像と参照画像の関係について、原画像と参照画像間の極端な輝度差がない場合には、重みまたはウインドウサイズを少し調整することで少なくとも 1 枚は原画像の特徴を残しつつ参照画像の特

徴を反映させた画像生成が可能であり、提案手法は事例参照型画風変換に有効であると言える。

次に、各々の結果について考察する。図 17 から図 22 に、原画像 (A1) に対する変換結果を示す。図より、画像内全体でほぼ輝度に変化がない原画像に対して処理を行った場合には、参照画像の一部のテクスチャのみが使用されることで参照画像の全体的な特徴や雰囲気をほとんど反映できていない。また、参照画像 (B1)(B4)(B6) のように、使用された領域のテクスチャが強い参照画像はウィンドウサイズが重みが大きくなるにつれてその特徴を強く反映するが、一方で参照画像 (B2)(B3)(B5) のように、使用された領域のテクスチャが弱い参照画像は原画像が平滑化されたようになっている。これは提案手法が輝度情報のみを用いているため、原画像のコントラストが小さい場合には、原画像中の空と山といった異なる領域においても、参照画像中の同一領域のテクスチャが利用されるなど、不適切なテクスチャ参照が発生しているためだと考えられる。

図 23 から図 28 に、原画像 (A2) に対する変換結果を示す。いずれの参照事例による変換結果も、参照画像の特徴が反映されていることが確認できる。しかし、参照画像 (B1) を用いたときには、ウィンドウサイズと重みがともに最小の時を除いて、不連続なテクスチャが発生している。これは参照画像 (B1) の輝度変化が離散的で、原画像のわずかな輝度変化に対応できていないためだと考えられる。また、参照画像 (B2)(B5)(B6) による生成結果では、水面に反射した山の色などによって不自然なテクスチャが生成されている部分が見受けられる。また、ウィンドウサイズと重みが大きくなるにつれて山の境界などで生成画像の輝度が大きく変わることも確認できる。

図 29 から図 34 に、原画像 (A3) に対する変換結果を示す。参照画像 (B1) は輝度変化が離散的であるために、空の領域で不自然なテクスチャが生成されてしまっている。参照画像 (B2) や (B3) による生成画像で特に顕著であるが、ビルや山がゴーストとして画像に反映されてしまう場合がある。これはビルの先などに対して適切なスケールでの類似度比較ができていないためであると考えられる。

図 35 から図 40 に、原画像 (A4) に対する変換結果を示す。参照画像 (B1) では、色が全く異なる画像間での変換であるため重みやウィンドウサイズを少し増加さ

せただけで原画像の構造はほとんど保てていないが、 $W = 3 \times 3$ かつ重み α が小さいときは原画像の色彩が残っており、大きくなるにつれて参照画像の色彩が強く反映されていることが分かる。また、参照画像 (B2) ではウインドウサイズ W が 5×5 になると、一部のテクスチャのみが強く参照されている。テクスチャスケールが原画像の草のパターンと偶然一致したためそのようなテクスチャが発生していると考えられる。類似したパターンが多くある参照画像 (B4) では、特徴を良くとらえた変換となっている。

4.1.2 処理時間およびエネルギーの収束について

ここでは、提案手法を用いた画像生成に要した時間およびエネルギーの収束について述べる。ここで画像生成に要した時間は、画像ピラミッドを生成してから原寸スケールでのエネルギーが収束し、画像生成が終了するまでに要した時間とする。

まず全体的な傾向として、全ての画像生成に要した時間は1分以内となった。これは画像サイズが小さいことが原因として考えられる。また、生成画像に原画像の構造が残っている場合には画像の生成に要する時間が長くなる傾向が見られた。特に、参照画像にはっきりとしたテクスチャが存在する場合に時間が長くなる傾向がある。これは、画素値の更新処理をした際のエネルギーの変化が大きく、エネルギーの収束に要する反復回数が増加しやすいためだと考えられる。

ウインドウサイズの変化が与える影響について、ウインドウサイズを変更した際、図21のように生成結果に大きく影響を与えない場合には、ウインドウサイズが大きくなることで処理時間が長くなることが分かる。また、重みが与える影響についても、重みを変更した際、生成結果に大きな影響を与えない場合、処理時間に影響を与えない。一方で、図20の $W = 5 \times 5, \alpha = 0.5$ と $W = 5 \times 5, \alpha = 0.8$ とのように、ウインドウサイズや重みによって、画像生成処理の初期段階で均一なテクスチャになりやすい場合、エネルギーが収束するまでに要する反復回数が少なり、処理時間が短くなる傾向がある。

また、入力する画像の解像度を変更した場合について述べる。ここでは原画像 (A2) を用いて、各パラメータを $W = 3 \times 3, \alpha = 0.5$ に固定して画風変換処理を

行った場合の処理時間を表 2 に示す。解像度は短辺が 300 画素の場合と 600 画素の場合の 2 通りとして，原画像の解像度は，それぞれ 453x300，906x600 となるよう調整した。表より，入力した画像の解像度が 4 倍になると，処理時間もほぼ 4 倍となることが分かる。ただし，参照画像 (B1) の場合はこの限りではないが，これは参照画像 (B1) がはっきりとしたテクスチャを持つ画像であることから，エネルギーの収束に要する反復回数が増加したことが原因だと考えられる。

表 2 入力した画像の解像度による処理時間の比較

参照画像	参照画像の解像度	処理時間	参照画像の解像度	処理時間
(B1)	379x300	14(秒)	759x600	81(秒)
(B2)	393x300	15(秒)	786x600	58(秒)
(B3)	455x300	8(秒)	909x600	31(秒)
(B4)	470x300	18(秒)	940x600	64(秒)
(B5)	394x300	5(秒)	788x600	24(秒)
(B6)	471x300	12(秒)	941x600	50(秒)

続いてエネルギーの収束について述べる。図 41 から図 52 に， $W = 3 \times 3, \alpha = 0.5$ で画像生成処理を行った際のエネルギーの変化と反復回数の関係を示す。なお，ここでは各スケールの初期エネルギーを 1 になるように正規化している。式 (2) で定義したエネルギーは，すべての処理において単調減少している。ほとんどの処理では原寸スケールでの処理の終了時にほぼ横ばいのエネルギー変化となっているが，図 41 の参照画像 (B1) を用いた場合の処理や図 42 の参照画像 (B4) 用いた処理の場合などは十分にエネルギーが収束しているとは言えず，収束条件をより厳しくする必要があると考えられる。



原画像 (A1)



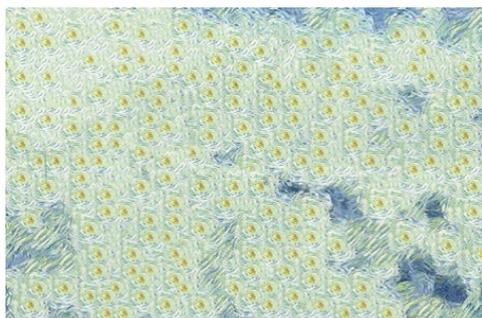
参照画像 (B1)



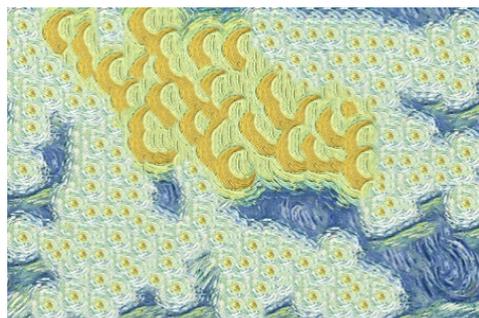
$W = 3 \times 3, \alpha = 0.2$, 処理時間 22(秒)



$W = 5 \times 5, \alpha = 0.2$, 処理時間 9(秒)



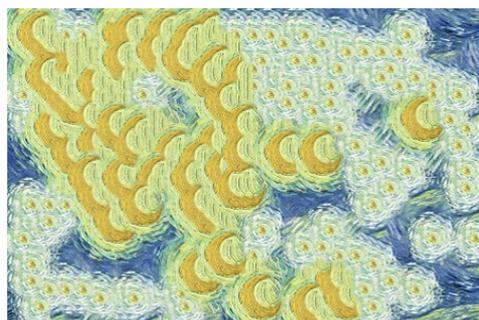
$W = 3 \times 3, \alpha = 0.5$, 処理時間 6(秒)



$W = 5 \times 5, \alpha = 0.5$, 処理時間 8(秒)



$W = 3 \times 3, \alpha = 0.8$, 処理時間 9(秒)



$W = 5 \times 5, \alpha = 0.8$, 処理時間 9(秒)

図 17 原画像 (A1) と参照画像 (B1) の生成結果



原画像 (A1)



参照画像 (B2)



$W = 3 \times 3, \alpha = 0.2$, 処理時間 10(秒)



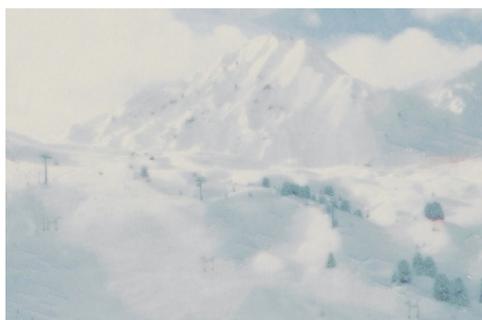
$W = 5 \times 5, \alpha = 0.2$, 処理時間 10(秒)



$W = 3 \times 3, \alpha = 0.5$, 処理時間 8(秒)



$W = 5 \times 5, \alpha = 0.5$, 処理時間 13(秒)



$W = 3 \times 3, \alpha = 0.8$, 処理時間 7(秒)



$W = 5 \times 5, \alpha = 0.8$, 処理時間 11(秒)

図 18 原画像 (A1) と参照画像 (B2) の生成結果



原画像 (A1)



参照画像 (B3)



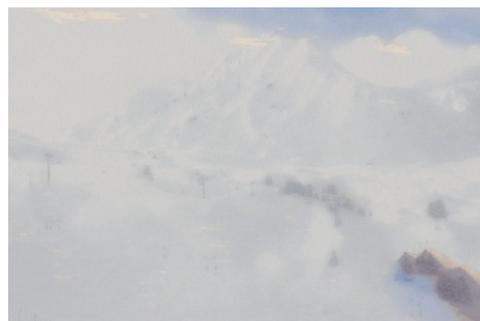
$W = 3 \times 3, \alpha = 0.2$, 処理時間 6(秒)



$W = 5 \times 5, \alpha = 0.2$, 処理時間 16(秒)



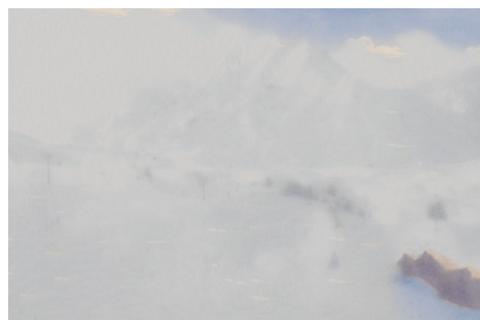
$W = 3 \times 3, \alpha = 0.5$, 処理時間 7(秒)



$W = 5 \times 5, \alpha = 0.5$, 処理時間 8(秒)



$W = 3 \times 3, \alpha = 0.8$, 処理時間 7(秒)



$W = 5 \times 5, \alpha = 0.8$, 処理時間 9(秒)

図 19 原画像 (A1) と参照画像 (B3) の生成結果



原画像 (A1)



参照画像 (B4)



$W = 3 \times 3, \alpha = 0.2$, 処理時間 19(秒)



$W = 5 \times 5, \alpha = 0.2$, 処理時間 50(秒)



$W = 3 \times 3, \alpha = 0.5$, 処理時間 22(秒)



$W = 5 \times 5, \alpha = 0.5$, 処理時間 7(秒)



$W = 3 \times 3, \alpha = 0.8$, 処理時間 16(秒)



$W = 5 \times 5, \alpha = 0.8$, 処理時間 6(秒)

図 20 原画像 (A1) と参照画像 (B4) の生成結果



原画像 (A1)



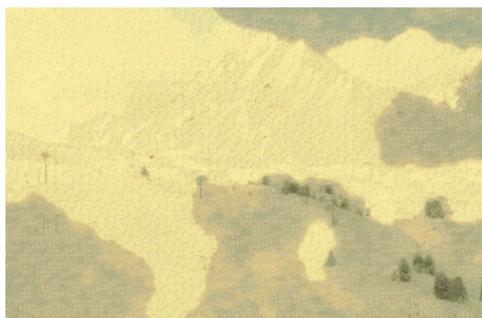
参照画像 (B5)



$W = 3 \times 3, \alpha = 0.2$, 処理時間 5(秒)



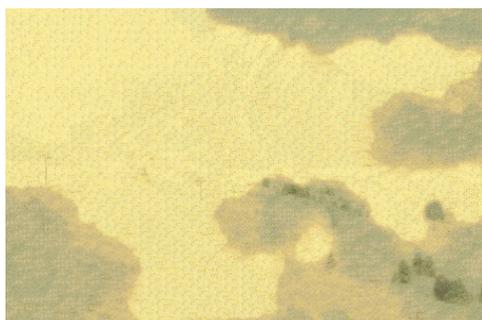
$W = 5 \times 5, \alpha = 0.2$, 処理時間 6(秒)



$W = 3 \times 3, \alpha = 0.5$, 処理時間 4(秒)



$W = 5 \times 5, \alpha = 0.5$, 処理時間 6(秒)



$W = 3 \times 3, \alpha = 0.8$, 処理時間 3(秒)



$W = 5 \times 5, \alpha = 0.8$, 処理時間 6(秒)

図 21 原画像 (A1) と参照画像 (B5) の生成結果



原画像 (A1)



参照画像 (B6)



$W = 3 \times 3, \alpha = 0.2$, 処理時間 13(秒)



$W = 5 \times 5, \alpha = 0.2$, 処理時間 32(秒)



$W = 3 \times 3, \alpha = 0.5$, 処理時間 16(秒)



$W = 5 \times 5, \alpha = 0.5$, 処理時間 7(秒)



$W = 3 \times 3, \alpha = 0.8$, 処理時間 18(秒)



$W = 5 \times 5, \alpha = 0.8$, 処理時間 6(秒)

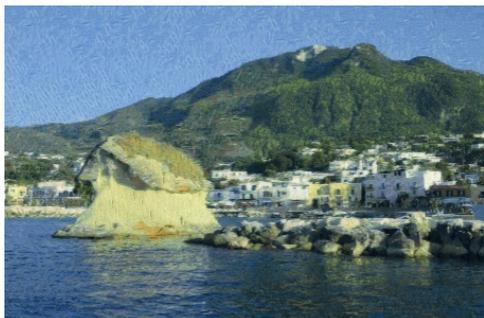
図 22 原画像 (A1) と参照画像 (B6) の生成結果



原画像 (A2)



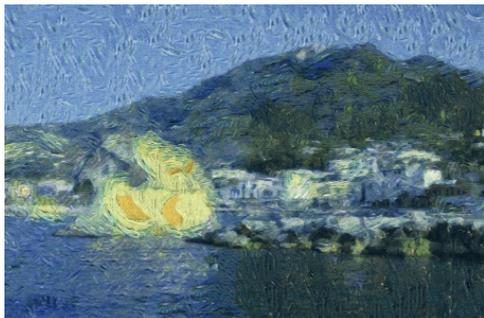
参照画像 (B1)



$W = 3 \times 3, \alpha = 0.2$, 処理時間 18(秒)



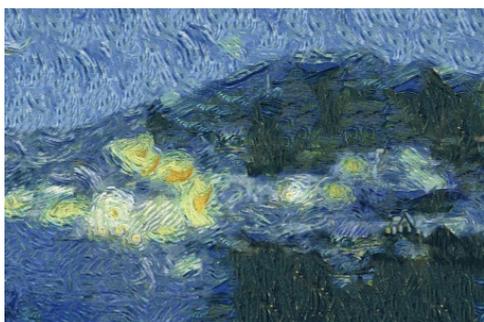
$W = 5 \times 5, \alpha = 0.2$, 処理時間 11(秒)



$W = 3 \times 3, \alpha = 0.5$, 処理時間 14(秒)



$W = 5 \times 5, \alpha = 0.5$, 処理時間 8(秒)



$W = 3 \times 3, \alpha = 0.8$, 処理時間 6(秒)



$W = 5 \times 5, \alpha = 0.8$, 処理時間 6(秒)

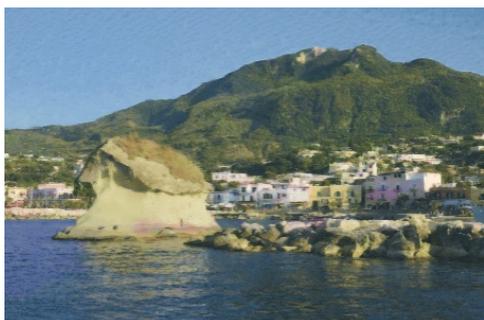
図 23 原画像 (A2) と参照画像 (B1) の生成結果



原画像 (A2)



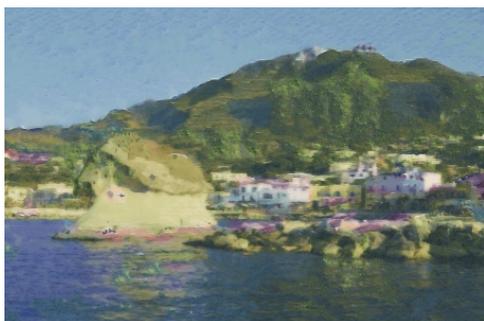
参照画像 (B2)



$W = 3 \times 3, \alpha = 0.2$, 処理時間 13(秒)



$W = 5 \times 5, \alpha = 0.2$, 処理時間 32(秒)



$W = 3 \times 3, \alpha = 0.5$, 処理時間 15(秒)



$W = 5 \times 5, \alpha = 0.5$, 処理時間 9(秒)



$W = 3 \times 3, \alpha = 0.8$, 処理時間 13(秒)



$W = 5 \times 5, \alpha = 0.8$, 処理時間 8(秒)

図 24 原画像 (A2) と参照画像 (B2) の生成結果



原画像 (A2)



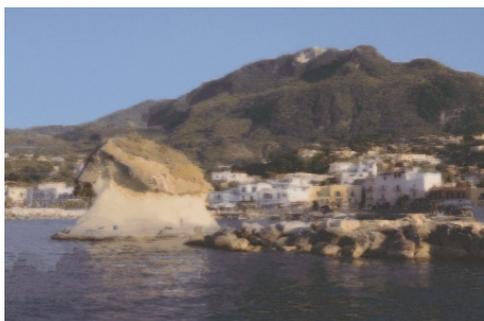
参照画像 (B3)



$W = 3 \times 3, \alpha = 0.2$, 処理時間 7(秒)



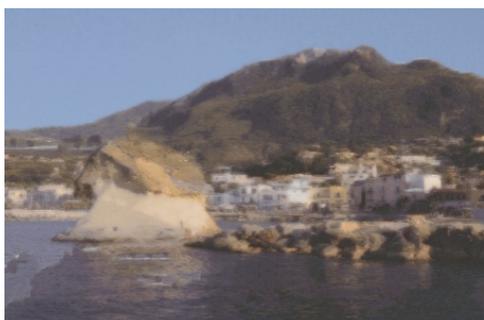
$W = 5 \times 5, \alpha = 0.2$, 処理時間 14(秒)



$W = 3 \times 3, \alpha = 0.5$, 処理時間 8(秒)



$W = 5 \times 5, \alpha = 0.5$, 処理時間 14(秒)



$W = 3 \times 3, \alpha = 0.8$, 処理時間 9(秒)



$W = 5 \times 5, \alpha = 0.8$, 処理時間 9(秒)

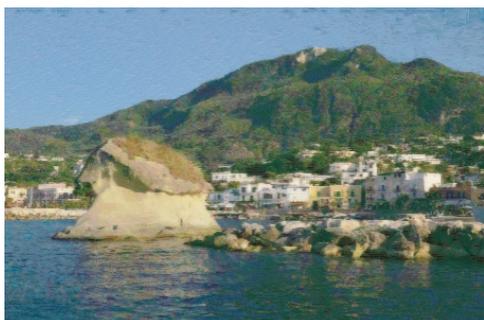
図 25 原画像 (A2) と参照画像 (B3) の生成結果



原画像 (A2)



参照画像 (B4)



$W = 3 \times 3, \alpha = 0.2$, 処理時間 16(秒)



$W = 5 \times 5, \alpha = 0.2$, 処理時間 33(秒)



$W = 3 \times 3, \alpha = 0.5$, 処理時間 18(秒)



$W = 5 \times 5, \alpha = 0.5$, 処理時間 7(秒)



$W = 3 \times 3, \alpha = 0.8$, 処理時間 10(秒)



$W = 5 \times 5, \alpha = 0.8$, 処理時間 7(秒)

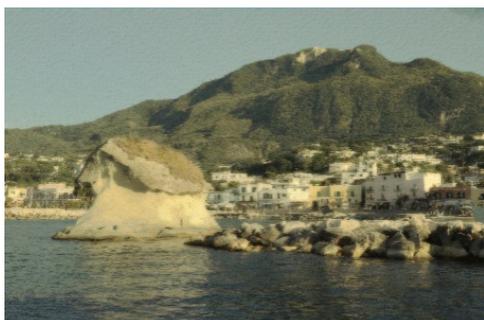
図 26 原画像 (A2) と参照画像 (B4) の生成結果



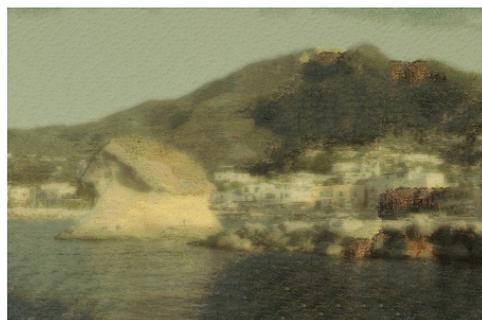
原画像 (A2)



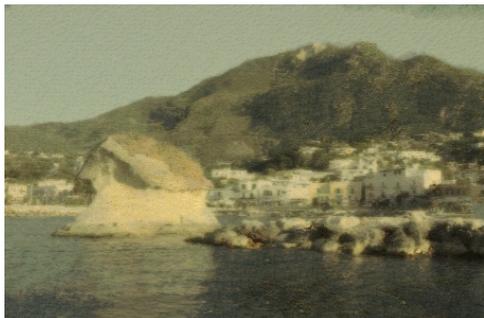
参照画像 (B5)



$W = 3 \times 3, \alpha = 0.2$, 処理時間 5(秒)



$W = 5 \times 5, \alpha = 0.2$, 処理時間 9(秒)



$W = 3 \times 3, \alpha = 0.5$, 処理時間 5(秒)



$W = 5 \times 5, \alpha = 0.5$, 処理時間 6(秒)



$W = 3 \times 3, \alpha = 0.8$, 処理時間 5(秒)



$W = 5 \times 5, \alpha = 0.8$, 処理時間 5(秒)

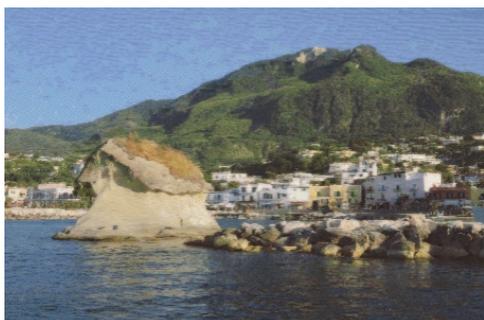
図 27 原画像 (A2) と参照画像 (B5) の生成結果



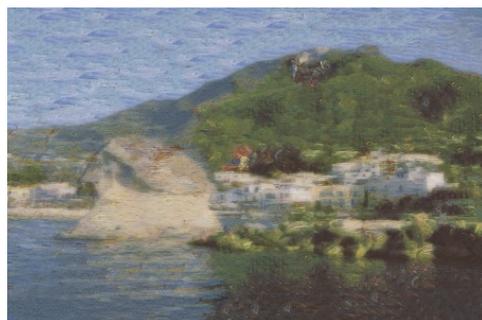
原画像 (A2)



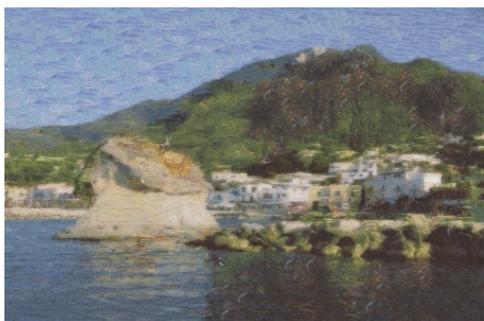
参照画像 (B6)



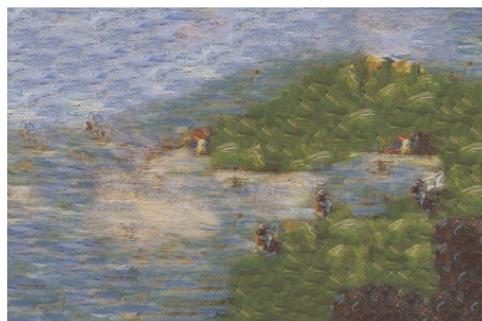
$W = 3 \times 3, \alpha = 0.2$, 処理時間 11(秒)



$W = 5 \times 5, \alpha = 0.2$, 処理時間 21(秒)



$W = 3 \times 3, \alpha = 0.5$, 処理時間 12(秒)



$W = 5 \times 5, \alpha = 0.5$, 処理時間 6(秒)



$W = 3 \times 3, \alpha = 0.8$, 処理時間 10(秒)



$W = 5 \times 5, \alpha = 0.8$, 処理時間 6(秒)

図 28 原画像 (A2) と参照画像 (B6) の生成結果



原画像 (A3)



参照画像 (B1)



$W = 3 \times 3, \alpha = 0.2$, 処理時間 22(秒)



$W = 5 \times 5, \alpha = 0.2$, 処理時間 9(秒)



$W = 3 \times 3, \alpha = 0.5$, 処理時間 10(秒)



$W = 5 \times 5, \alpha = 0.5$, 処理時間 8(秒)



$W = 3 \times 3, \alpha = 0.8$, 処理時間 7(秒)



$W = 5 \times 5, \alpha = 0.8$, 処理時間 8(秒)

図 29 原画像 (A3) と参照画像 (B1) の生成結果



原画像 (A3)



参照画像 (B2)



$W = 3 \times 3, \alpha = 0.2$, 処理時間 15(秒)



$W = 5 \times 5, \alpha = 0.2$, 処理時間 42(秒)



$W = 3 \times 3, \alpha = 0.5$, 処理時間 15(秒)



$W = 5 \times 5, \alpha = 0.5$, 処理時間 9(秒)



$W = 3 \times 3, \alpha = 0.8$, 処理時間 18(秒)



$W = 5 \times 5, \alpha = 0.8$, 処理時間 7(秒)

図 30 原画像 (A3) と参照画像 (B2) の生成結果



原画像 (A3)



参照画像 (B3)



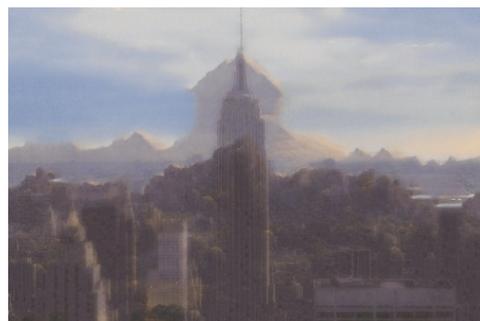
$W = 3 \times 3, \alpha = 0.2$, 処理時間 9(秒)



$W = 5 \times 5, \alpha = 0.2$, 処理時間 16(秒)



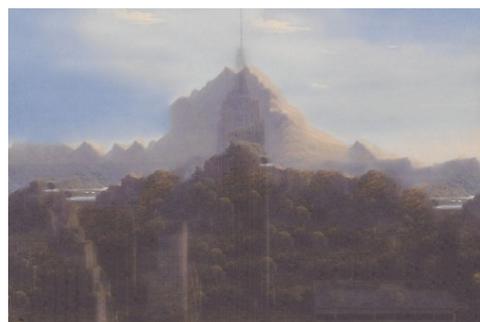
$W = 3 \times 3, \alpha = 0.5$, 処理時間 9(秒)



$W = 5 \times 5, \alpha = 0.5$, 処理時間 15(秒)



$W = 3 \times 3, \alpha = 0.8$, 処理時間 10(秒)



$W = 5 \times 5, \alpha = 0.8$, 処理時間 10(秒)

図 31 原画像 (A3) と参照画像 (B3) の生成結果



原画像 (A3)



参照画像 (B4)



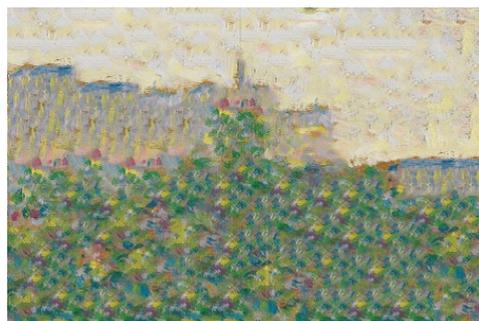
$W = 3 \times 3, \alpha = 0.2$, 処理時間 16(秒)



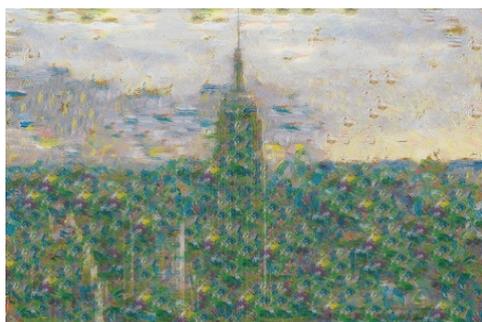
$W = 5 \times 5, \alpha = 0.2$, 処理時間 35(秒)



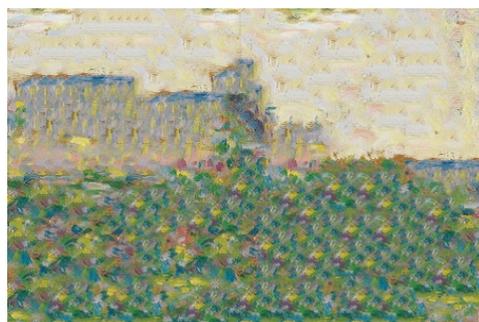
$W = 3 \times 3, \alpha = 0.5$, 処理時間 19(秒)



$W = 5 \times 5, \alpha = 0.5$, 処理時間 7(秒)



$W = 3 \times 3, \alpha = 0.8$, 処理時間 9(秒)



$W = 5 \times 5, \alpha = 0.8$, 処理時間 7(秒)

図 32 原画像 (A3) と参照画像 (B4) の生成結果



原画像 (A3)



参照画像 (B5)



$W = 3 \times 3, \alpha = 0.2$, 処理時間 6(秒)



$W = 5 \times 5, \alpha = 0.2$, 処理時間 9(秒)



$W = 3 \times 3, \alpha = 0.5$, 処理時間 6(秒)



$W = 5 \times 5, \alpha = 0.5$, 処理時間 10(秒)



$W = 3 \times 3, \alpha = 0.8$, 処理時間 4(秒)



$W = 5 \times 5, \alpha = 0.8$, 処理時間 5(秒)

図 33 原画像 (A3) と参照画像 (B5) の生成結果



原画像 (A3)



参照画像 (B6)



$W = 3 \times 3, \alpha = 0.2$, 処理時間 13(秒)



$W = 5 \times 5, \alpha = 0.2$, 処理時間 18(秒)



$W = 3 \times 3, \alpha = 0.5$, 処理時間 13(秒)



$W = 5 \times 5, \alpha = 0.5$, 処理時間 8(秒)



$W = 3 \times 3, \alpha = 0.8$, 処理時間 10(秒)



$W = 5 \times 5, \alpha = 0.8$, 処理時間 7(秒)

図 34 原画像 (A3) と参照画像 (B6) の生成結果



原画像 (A4)



参照画像 (B1)



$W = 3 \times 3, \alpha = 0.2$, 処理時間 16(秒)



$W = 5 \times 5, \alpha = 0.2$, 処理時間 7(秒)



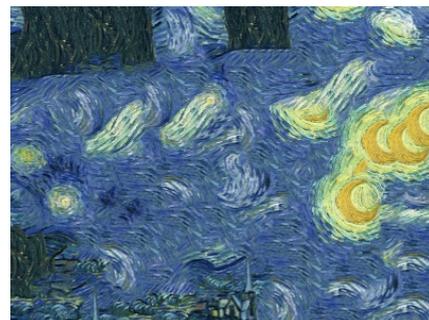
$W = 3 \times 3, \alpha = 0.5$, 処理時間 9(秒)



$W = 5 \times 5, \alpha = 0.5$, 処理時間 6(秒)



$W = 3 \times 3, \alpha = 0.8$, 処理時間 5(秒)



$W = 5 \times 5, \alpha = 0.8$, 処理時間 5(秒)

図 35 原画像 (A4) と参照画像 (B1) の生成結果



原画像 (A4)



参照画像 (B2)



$W = 3 \times 3, \alpha = 0.2$, 処理時間 12(秒)



$W = 5 \times 5, \alpha = 0.2$, 処理時間 19(秒)



$W = 3 \times 3, \alpha = 0.5$, 処理時間 12(秒)



$W = 5 \times 5, \alpha = 0.5$, 処理時間 7(秒)



$W = 3 \times 3, \alpha = 0.8$, 処理時間 10(秒)



$W = 5 \times 5, \alpha = 0.8$, 処理時間 7(秒)

図 36 原画像 (A4) と参照画像 (B2) の生成結果



原画像 (A4)



参照画像 (B3)



$W = 3 \times 3, \alpha = 0.2$, 処理時間 7(秒)



$W = 5 \times 5, \alpha = 0.2$, 処理時間 14(秒)



$W = 3 \times 3, \alpha = 0.5$, 処理時間 8(秒)



$W = 5 \times 5, \alpha = 0.5$, 処理時間 10(秒)



$W = 3 \times 3, \alpha = 0.8$, 処理時間 8(秒)



$W = 5 \times 5, \alpha = 0.8$, 処理時間 6(秒)

図 37 原画像 (A4) と参照画像 (B3) の生成結果



原画像 (A4)



参照画像 (B4)



$W = 3 \times 3, \alpha = 0.2$, 処理時間 14(秒)



$W = 5 \times 5, \alpha = 0.2$, 処理時間 21(秒)



$W = 3 \times 3, \alpha = 0.5$, 処理時間 16(秒)



$W = 5 \times 5, \alpha = 0.5$, 処理時間 5(秒)



$W = 3 \times 3, \alpha = 0.8$, 処理時間 7(秒)



$W = 5 \times 5, \alpha = 0.8$, 処理時間 6(秒)

図 38 原画像 (A4) と参照画像 (B4) の生成結果



原画像 (A4)



参照画像 (B5)



$W = 3 \times 3, \alpha = 0.2$, 処理時間 7(秒)



$W = 5 \times 5, \alpha = 0.2$, 処理時間 11(秒)



$W = 3 \times 3, \alpha = 0.5$, 処理時間 7(秒)



$W = 5 \times 5, \alpha = 0.5$, 処理時間 5(秒)



$W = 3 \times 3, \alpha = 0.8$, 処理時間 6(秒)



$W = 5 \times 5, \alpha = 0.8$, 処理時間 5(秒)

図 39 原画像 (A4) と参照画像 (B5) の生成結果



原画像 (A4)



参照画像 (B6)



$W = 3 \times 3, \alpha = 0.2$, 処理時間 11(秒)



$W = 5 \times 5, \alpha = 0.2$, 処理時間 15(秒)



$W = 3 \times 3, \alpha = 0.5$, 処理時間 10(秒)



$W = 5 \times 5, \alpha = 0.5$, 処理時間 6(秒)

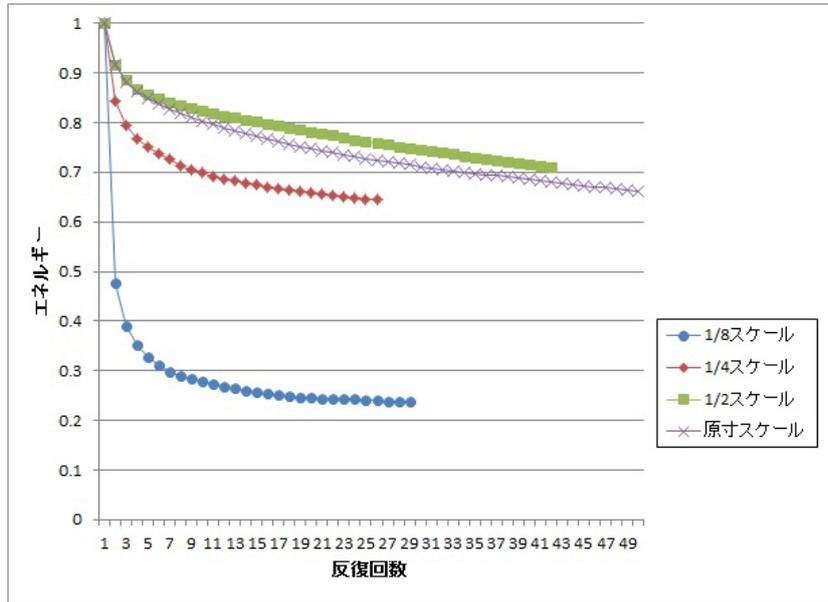


$W = 3 \times 3, \alpha = 0.8$, 処理時間 8(秒)

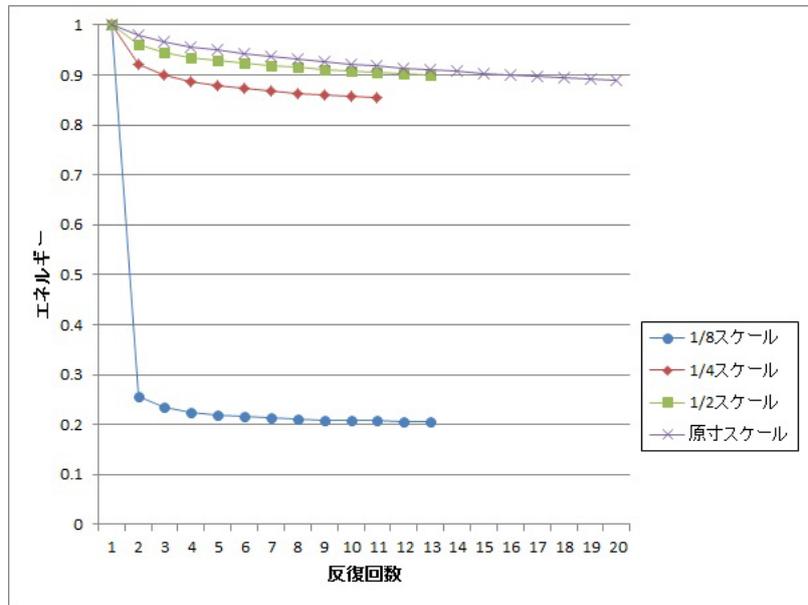


$W = 5 \times 5, \alpha = 0.8$, 処理時間 7(秒)

図 40 原画像 (A4) と参照画像 (B6) の生成結果

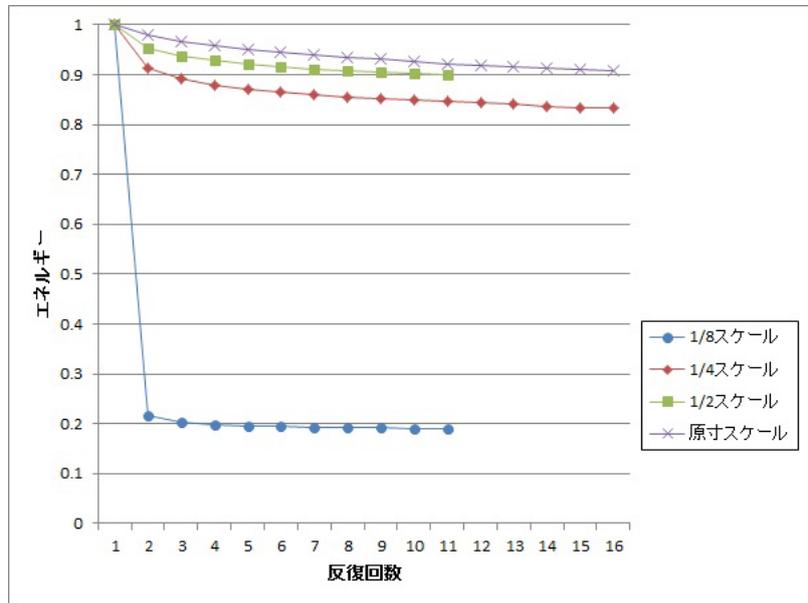


参照画像 (B1) を用いた場合

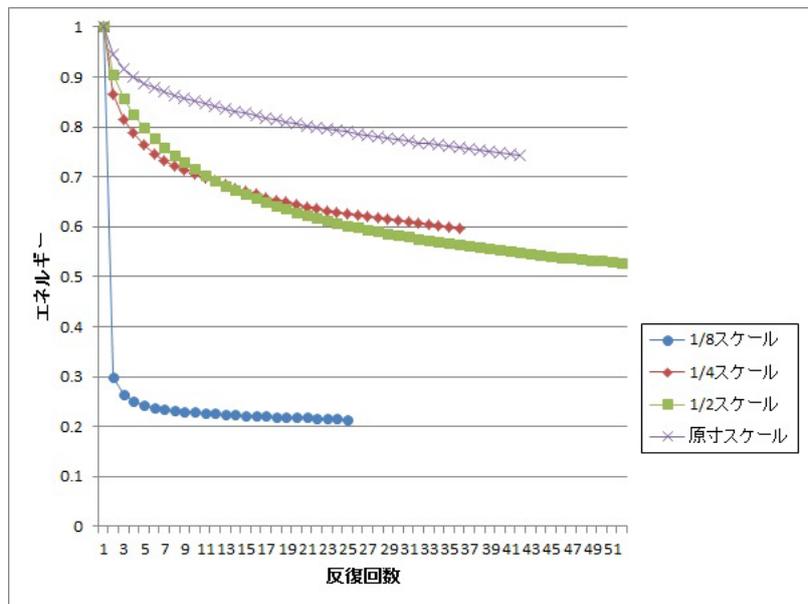


参照画像 (B2) を用いた場合

図 41 原画像 (A1) に対する画風変換処理のエネルギー変化 (a)

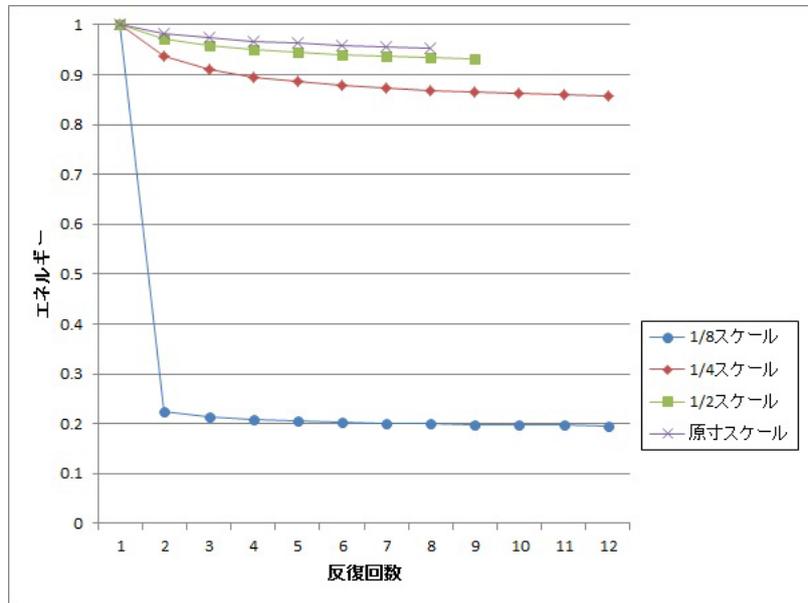


参照画像 (B3) を用いた場合

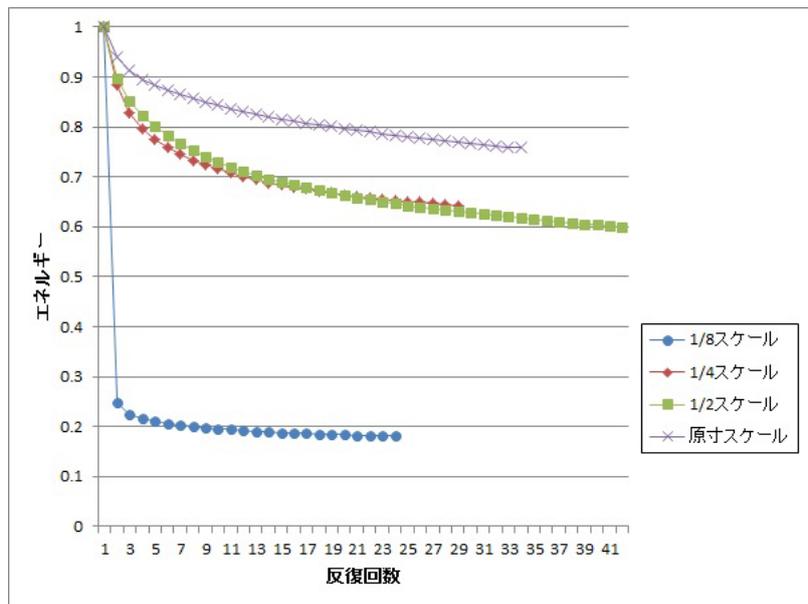


参照画像 (B4) を用いた場合

図 42 原画像 (A1) に対する画風変換処理のエネルギー変化 (b)

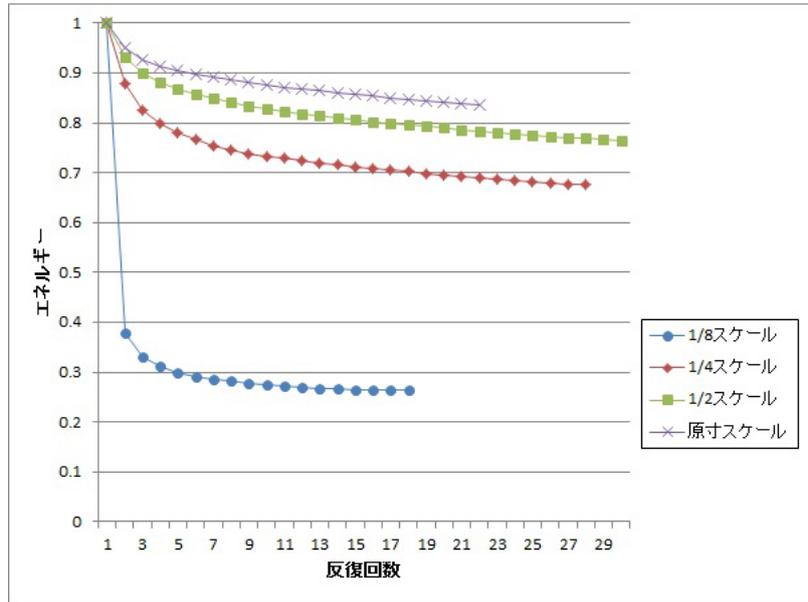


参照画像 (B5) を用いた場合

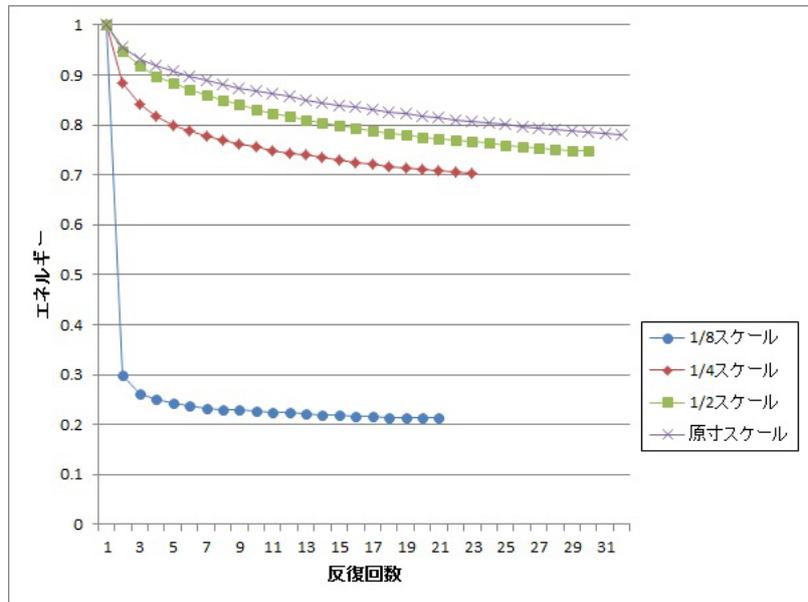


参照画像 (B6) を用いた場合

図 43 原画像 (A1) に対する画風変換処理のエネルギー変化 (c)

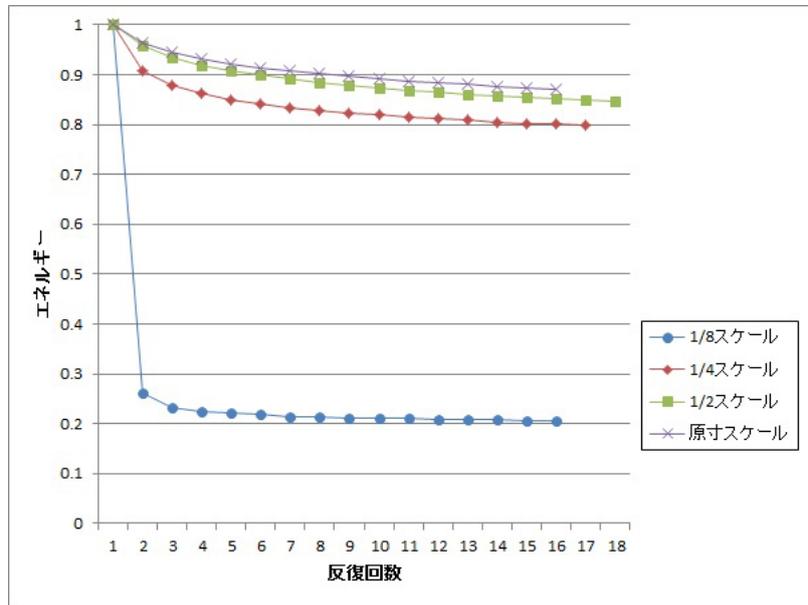


参照画像 (B1) を用いた場合

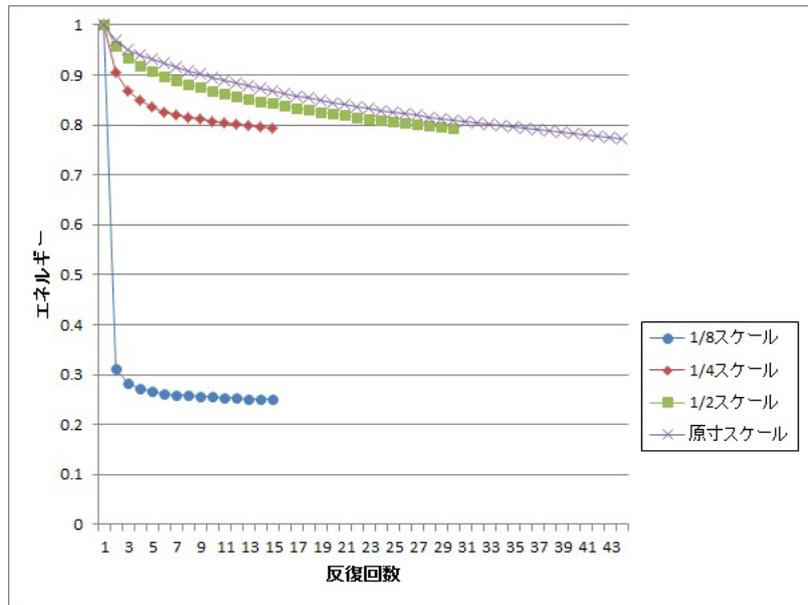


参照画像 (B2) を用いた場合

図 44 原画像 (A2) に対する画風変換処理のエネルギー変化 (a)

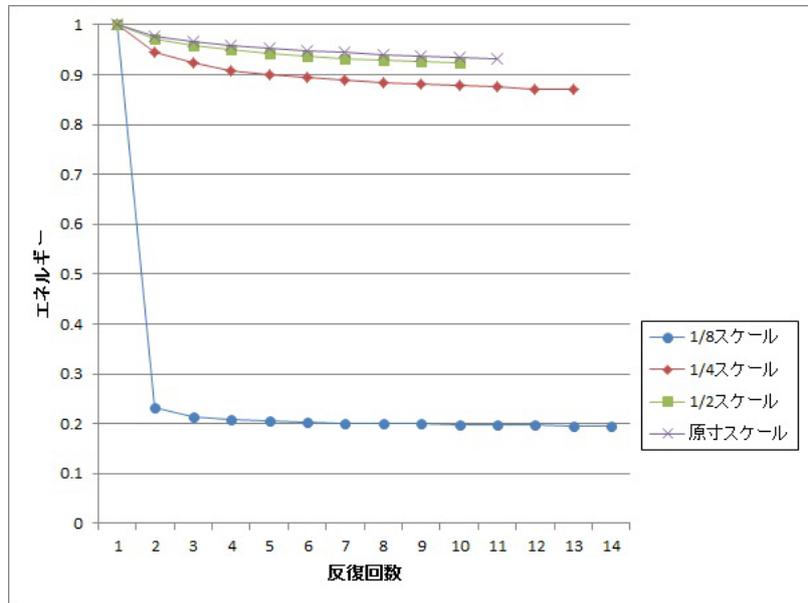


参照画像 (B3) を用いた場合

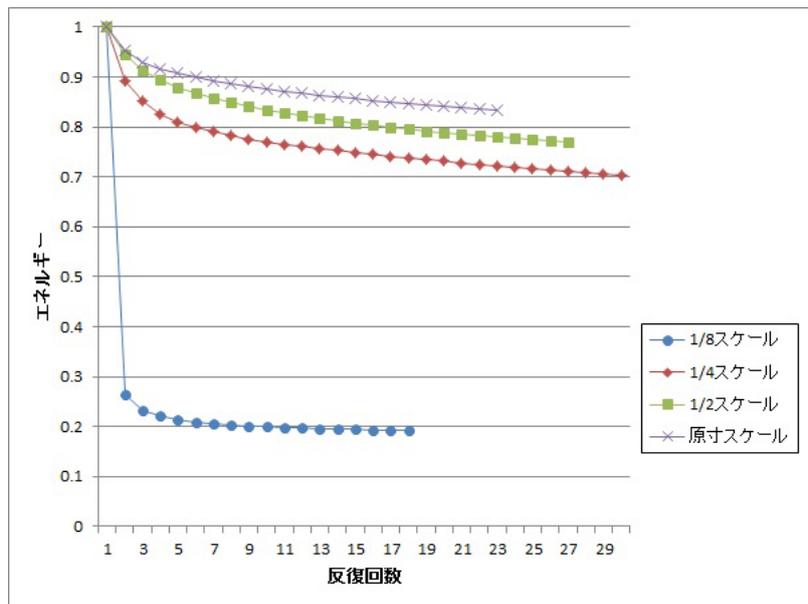


参照画像 (B4) を用いた場合

図 45 原画像 (A2) に対する画風変換処理のエネルギー変化 (b)

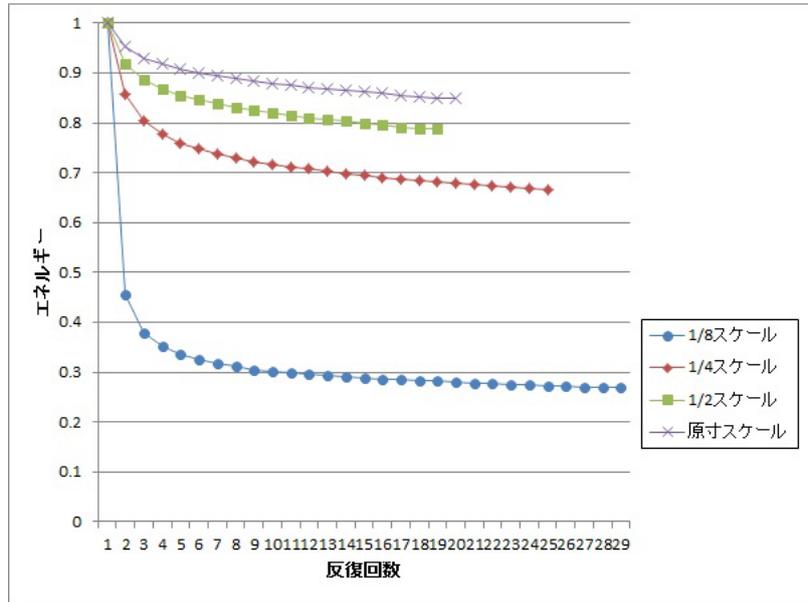


参照画像 (B5) を用いた場合

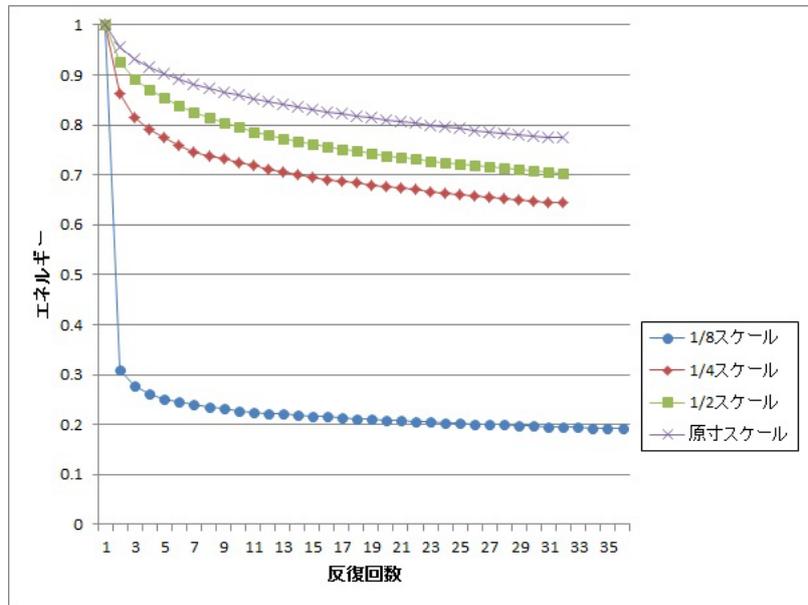


参照画像 (B6) を用いた場合

図 46 原画像 (A2) に対する画風変換処理のエネルギー変化 (c)

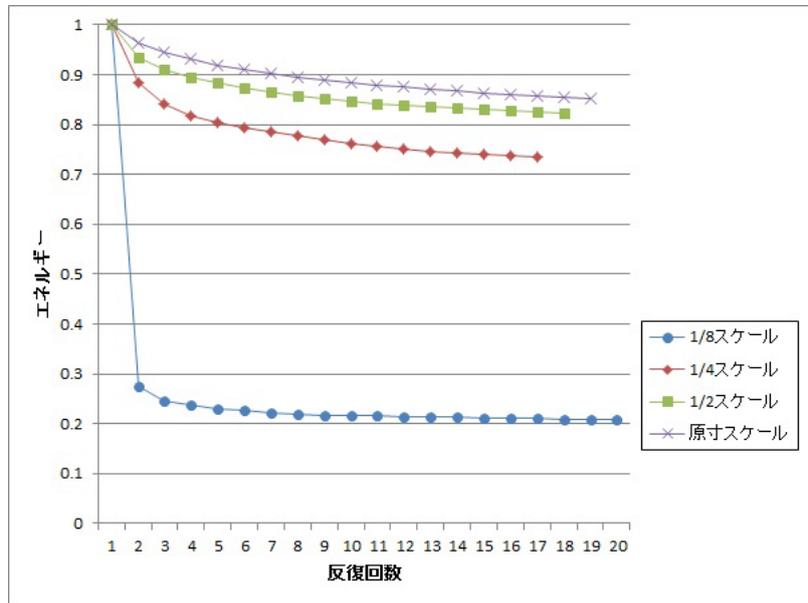


参照画像 (B1) を用いた場合

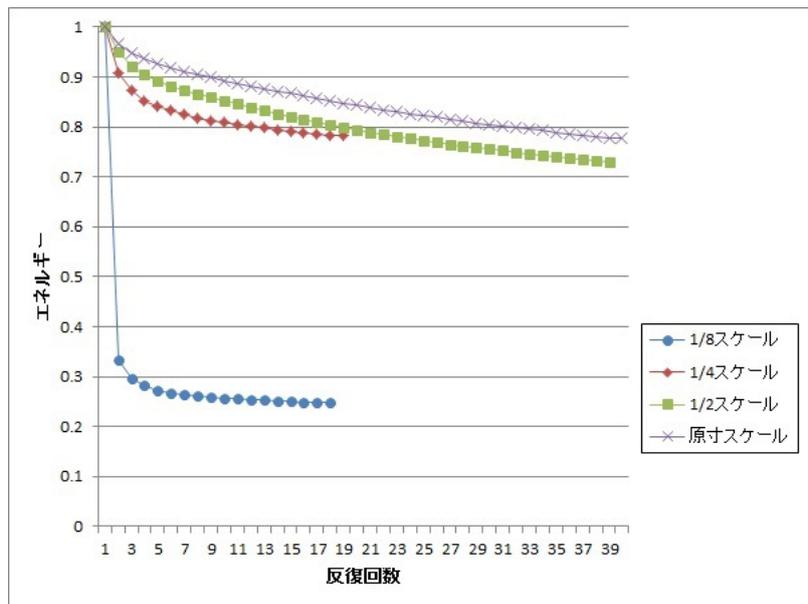


参照画像 (B2) を用いた場合

図 47 原画像 (A3) に対する画風変換処理のエネルギー変化 (a)

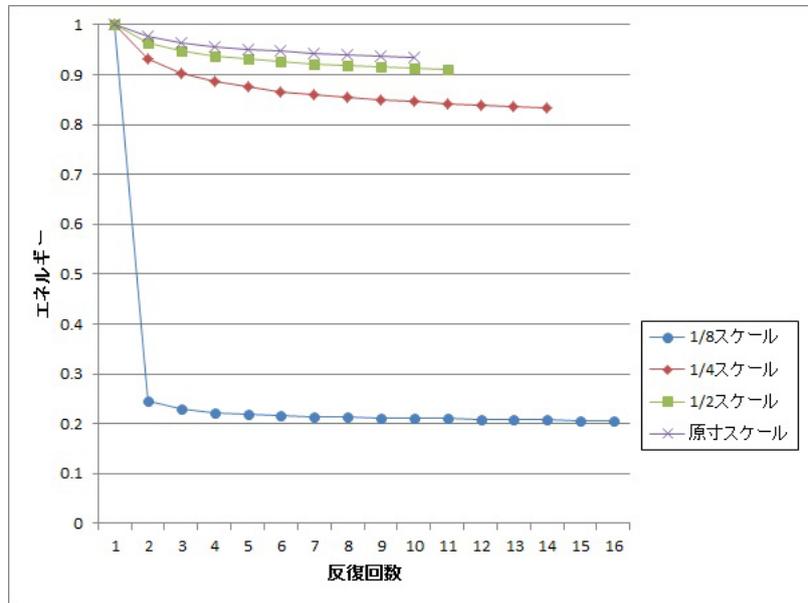


参照画像 (B3) を用いた場合

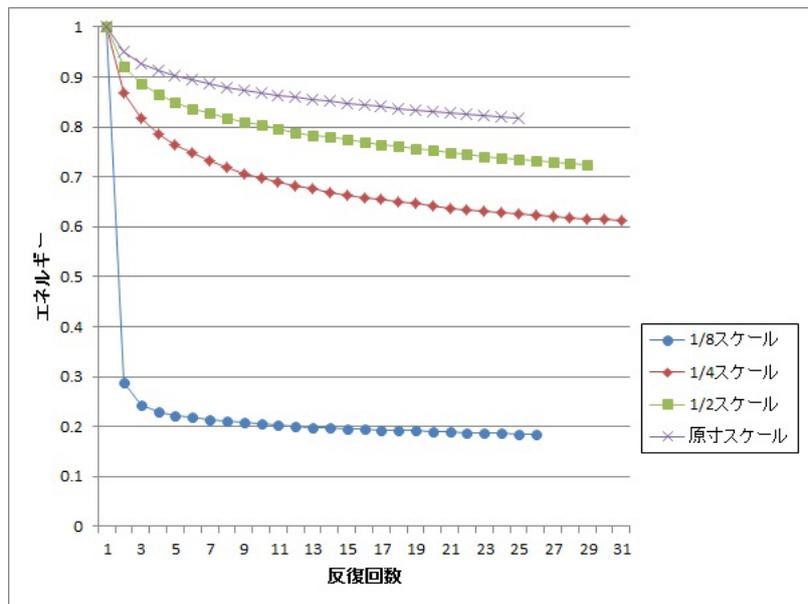


参照画像 (B4) を用いた場合

図 48 原画像 (A3) に対する画風変換処理のエネルギー変化 (b)

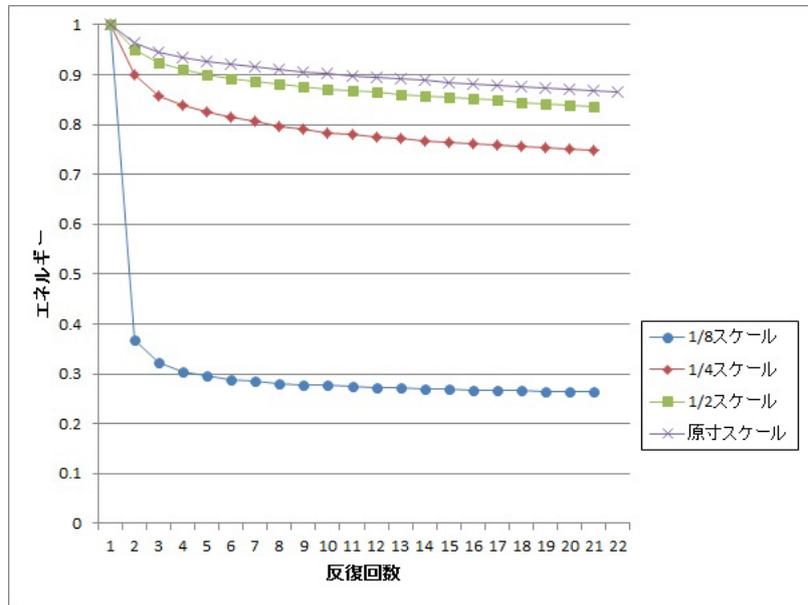


参照画像 (B5) を用いた場合

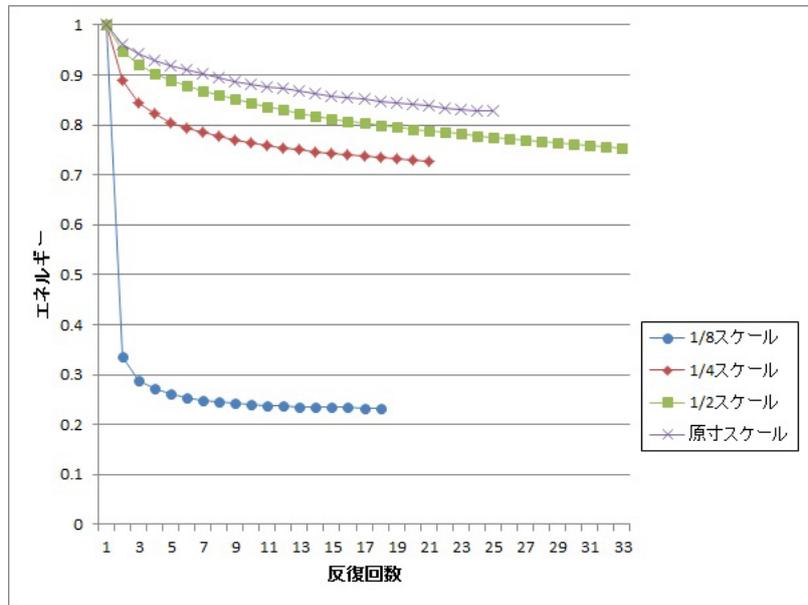


参照画像 (B6) を用いた場合

図 49 原画像 (A3) に対する画風変換処理のエネルギー変化 (c)

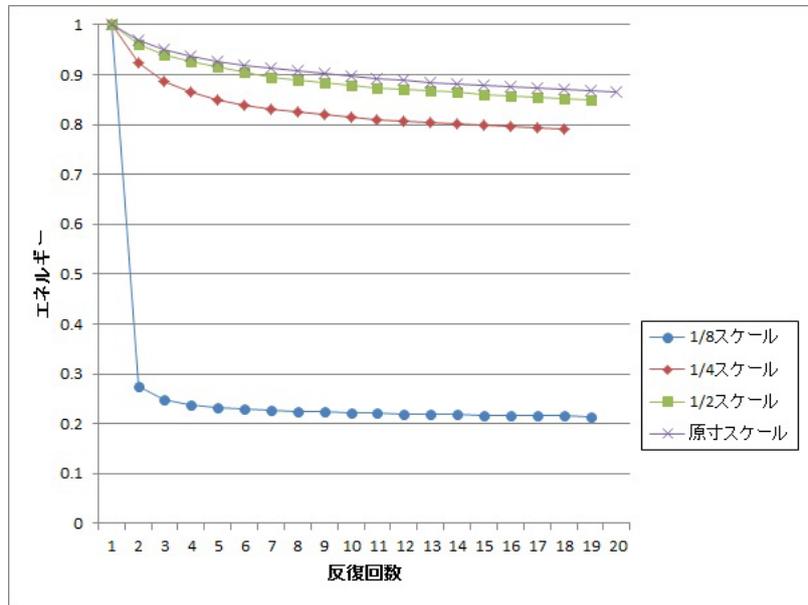


参照画像 (B1) を用いた場合

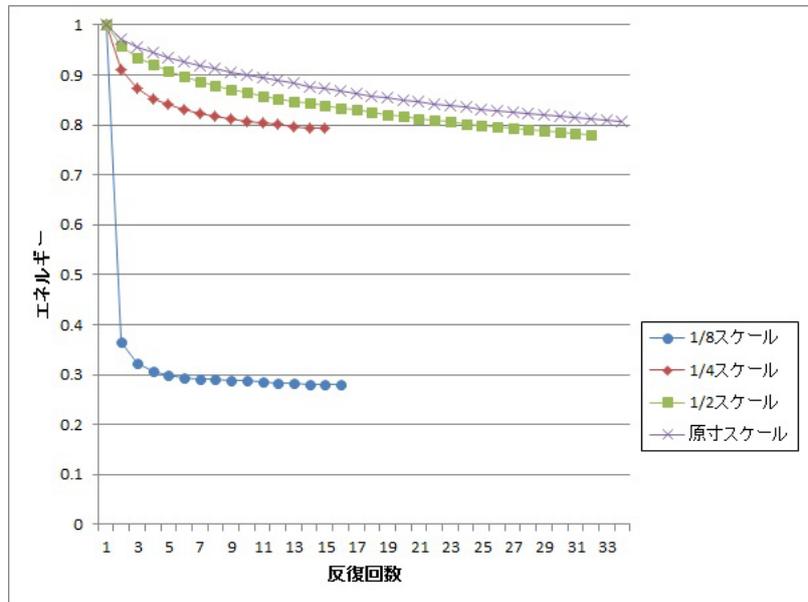


参照画像 (B2) を用いた場合

図 50 原画像 (A4) に対する画風変換処理のエネルギー変化 (a)

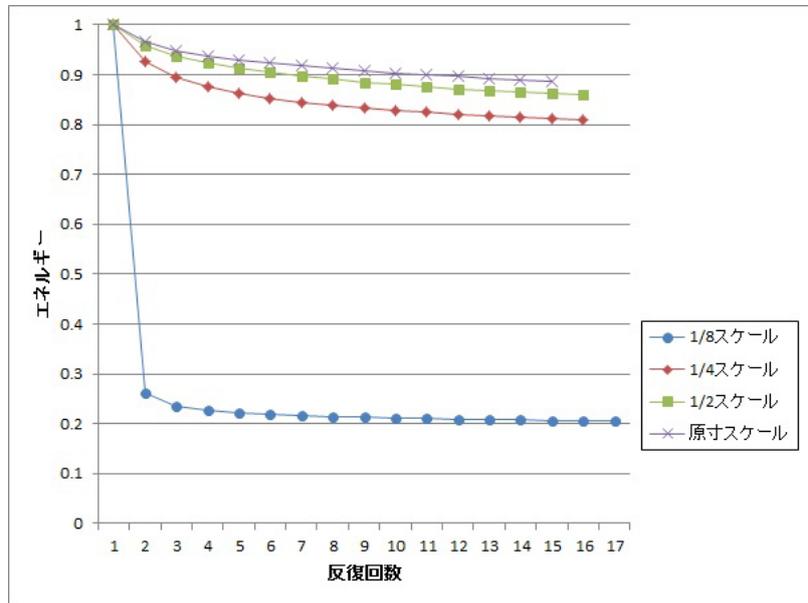


参照画像 (B3) を用いた場合

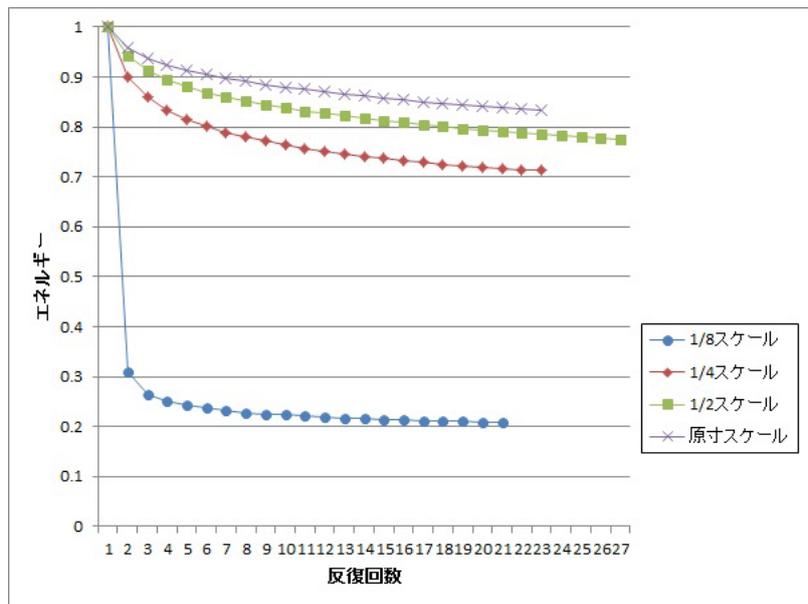


参照画像 (B4) を用いた場合

図 51 原画像 (A4) に対する画風変換処理のエネルギー変化 (b)



参照画像 (B5) を用いた場合



参照画像 (B6) を用いた場合

図 52 原画像 (A4) に対する画風変換処理のエネルギー変化 (c)

4.2 輝度調整を行った場合の画風変換結果と考察

本節では、輝度調整を行った場合における原画像 (A1)(A2)(A3)(A4) と参照画像 (B1)(B2)(B3)(B4)(B5)(B6) を用いた場合の画風変換結果について述べる。ここでは、ウインドウサイズ $W = 3 \times 3$ 、重み $\alpha = 0.5$ に固定し、輝度調整を行わなかった場合と輝度調整を行った場合で画像を生成し、結果の比較を行った。図 53 から図 76 に、すべての組み合わせによる画風変換結果を示す。まず全体の傾向について述べ、次に個別の結果を考察する。

まず、全体として、輝度調整なしで画風変換が成功するような画像の組み合わせでは、輝度調整によってかえって画風変換の結果が悪化する傾向にある。特に輝度調整によってコントラストが低くなる場合はそれが顕著であるが、逆にコントラストが高くなる場合は結果が改善される場合がある。一方、低コントラストの原画像に対して輝度調整を行った場合、参照画像の色調を反映することが可能になり、テクスチャの不適切な参照が抑えられ、画風変換の結果が改善される。次に各画像毎の結果について考察する。

図 53 から図 58 に原画像 (A1) に対する変換結果を示す。原画像 (A1) の画風変換では、全ての画像に対して生成結果のコントラストが上がり、参照されるテクスチャの範囲が拡大している。しかし、原画像の本来の色彩が乏しく、参照画像 (B3)(B4)(B6) による生成結果を見ると、局所的なテクスチャが参照される問題を完全に解消できたとは言えない。また、参照画像 (B2) による生成結果では、黒い部分に山のテクスチャが利用され不自然な画像が生成されている。一方、参照画像 (B5) による変換結果は、参照画像の色彩を反映できていると言える。これは (B5) がツートンカラーに近い色分布をしているためだと考えられる。参照画像 (B6) との画風変換では、参照の局所性は改善されたが、原画像の暗さの情報が失われるだけでなく、海面に草原のテクスチャを用いるといった不適切なテクスチャの参照が発生している。

図 59 から図 64 に原画像 (A2) に対する変換結果を示す。基本的に調整なしで画風変換がうまくいく画像の組が多く、輝度調整は不要といえる。参照画像 (B1) と参照画像 (B5) では輝度調整をすることによって若干の改善がみられるが、輝度調整が必要なほどの改善ではない。

図 65 から図 70 に原画像 (A3) に対する変換結果を示す。参照画像 (B2)(B4)(B6) による変換結果は、明らかに輝度調整しないもの比べて悪化している。参照画像 (B1) については、輝度調整により明るい部分が減ったため、参照画像中の星のテクスチャによる不自然な空は緩和された。一方で水平線の境界が曖昧になり、ビル先端も輝度調整なしの結果と比べ平滑化されている。参照画像 (B3) による変換結果は、全体的にコントラストが落ちている。そのため、建物のテクスチャが弱くなっているが、参照画像中の遠景の山の色彩がよく参照されるようになった。参照画像 (B1)(B3) による結果はユーザの好みによる可能性がある。参照画像 (B5) による生成結果では、輝度調整なしの結果よりもコントラストが強くなり、特に空の領域に関して参照画像の空の色をより多く反映している。

図 71 から図 76 に原画像 (A4) に対する変換結果を示す。全体的に輝度調整の効果は薄く、一部では逆効果になっている。特に参照画像 (B6) を用いた場合は、原画像のテクスチャが潰れている。参照画像 (B5) を用いた場合は、コントラストや基本的な色調は反映されるものの、参照画像自体が持つ特徴が十分に反映されているとは言えない。これは色の変化が参照画像では画像全体で大きく生じているのに対して、生成画像では細かいテクスチャによる小さな変化が目立つためだと考えられる。



原画像 (A1)



参照画像 (B1)



輝度調整画像



調整なし生成画像



調整あり生成画像

図 53 原画像 (A1) と参照画像 (B1) の生成結果



原画像 (A1)



参照画像 (B2)



輝度調整画像



調整なし生成画像



調整あり生成画像

図 54 原画像 (A1) と参照画像 (B2) の生成結果



原画像 (A1)



参照画像 (B3)



輝度調整画像



調整なし生成画像

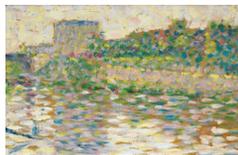


調整あり生成画像

図 55 原画像 (A1) と参照画像 (B3) の生成結果



原画像 (A1)



参照画像 (B4)



輝度調整画像



調整なし生成画像



調整あり生成画像

図 56 原画像 (A1) と参照画像 (B4) の生成結果



原画像 (A1)



参照画像 (B5)



輝度調整画像



調整なし生成画像



調整あり生成画像

図 57 原画像 (A1) と参照画像 (B5) の生成結果



原画像 (A1)



参照画像 (B6)



輝度調整画像



調整なし生成画像



調整あり生成画像

図 58 原画像 (A1) と参照画像 (B6) の生成結果



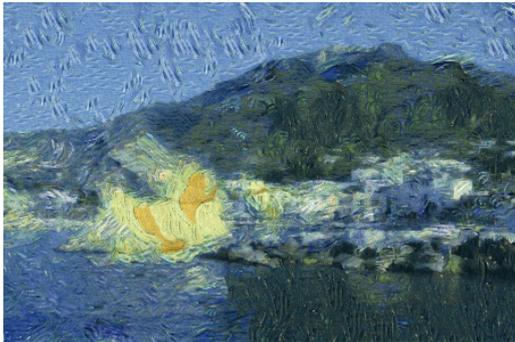
原画像 (A2)



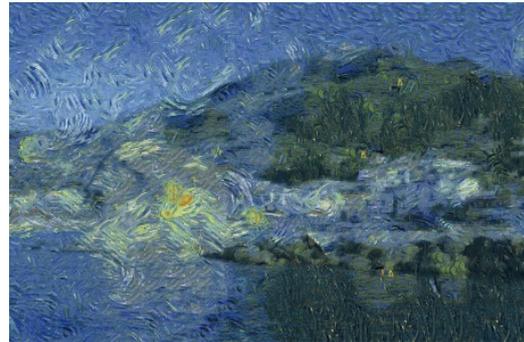
参照画像 (B1)



輝度調整画像



調整なし生成画像



調整あり生成画像

図 59 原画像 (A2) と参照画像 (B1) の生成結果



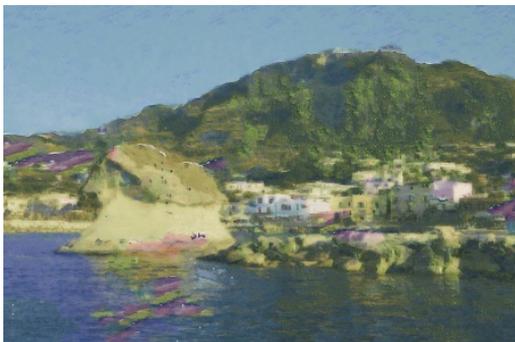
原画像 (A2)



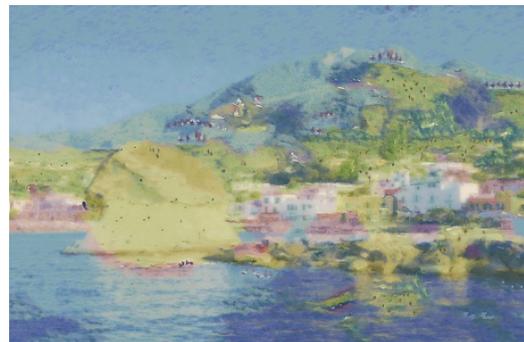
参照画像 (B2)



輝度調整画像



調整なし生成画像



調整あり生成画像

図 60 原画像 (A2) と参照画像 (B2) の生成結果



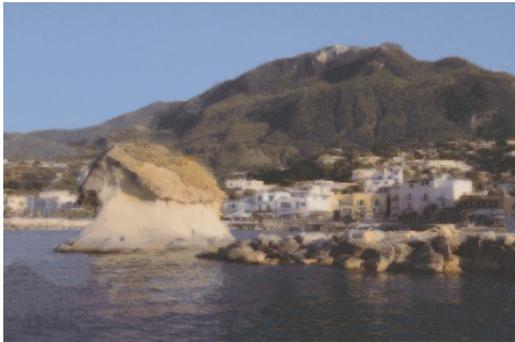
原画像 (A2)



参照画像 (B3)



輝度調整画像



調整なし生成画像



調整あり生成画像

図 61 原画像 (A2) と参照画像 (B3) の生成結果



原画像 (A2)



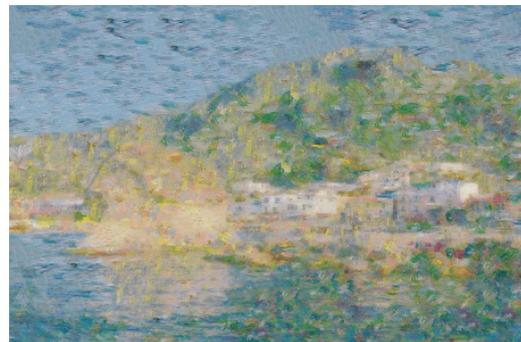
参照画像 (B4)



輝度調整画像



調整なし生成画像



調整あり生成画像

図 62 原画像 (A2) と参照画像 (B4) の生成結果



原画像 (A2)



参照画像 (B5)



輝度調整画像



調整なし生成画像



調整あり生成画像

図 63 原画像 (A2) と参照画像 (B5) の生成結果



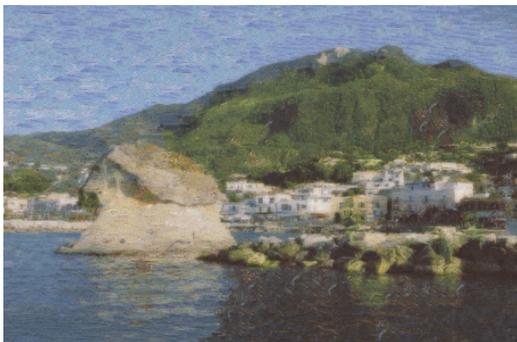
原画像 (A2)



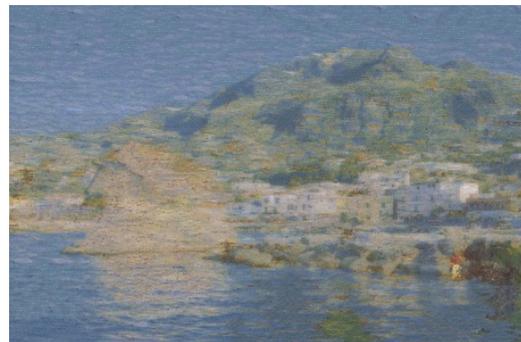
参照画像 (B6)



輝度調整画像



調整なし生成画像



調整あり生成画像

図 64 原画像 (A2) と参照画像 (B6) の生成結果



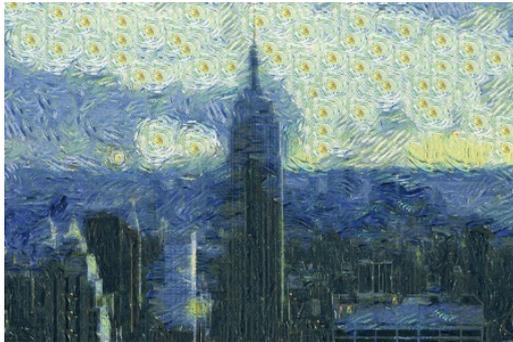
原画像 (A3)



参照画像 (B1)



輝度調整画像



調整なし生成画像



調整あり生成画像

図 65 原画像 (A3) と参照画像 (B1) の生成結果



原画像 (A3)



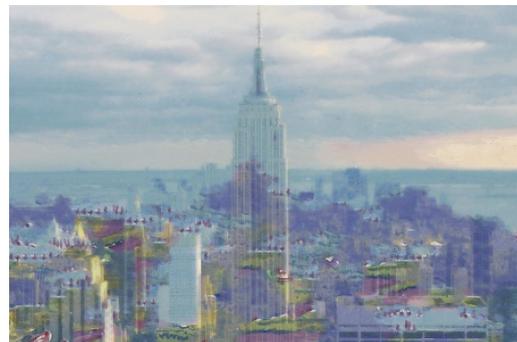
参照画像 (B2)



輝度調整画像



調整なし生成画像



調整あり生成画像

図 66 原画像 (A3) と参照画像 (B2) の生成結果



原画像 (A3)



参照画像 (B3)



輝度調整画像



調整なし生成画像

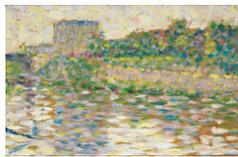


調整あり生成画像

図 67 原画像 (A3) と参照画像 (B3) の生成結果



原画像 (A3)



参照画像 (B4)



輝度調整画像



調整なし生成画像



調整あり生成画像

図 68 原画像 (A3) と参照画像 (B4) の生成結果



原画像 (A3)



参照画像 (B5)



輝度調整画像



調整なし生成画像



調整あり生成画像

図 69 原画像 (A3) と参照画像 (B5) の生成結果



原画像 (A3)



参照画像 (B6)



輝度調整画像



調整なし生成画像



調整あり生成画像

図 70 原画像 (A3) と参照画像 (B6) の生成結果



原画像 (A4)



参照画像 (B1)



輝度調整画像



調整なし生成画像



調整あり生成画像

図 71 原画像 (A4) と参照画像 (B1) の生成結果



原画像 (A4)



参照画像 (B2)



輝度調整画像



調整なし生成画像



調整あり生成画像

図 72 原画像 (A4) と参照画像 (B2) の生成結果



原画像 (A4)



参照画像 (B3)



輝度調整画像



調整なし生成画像

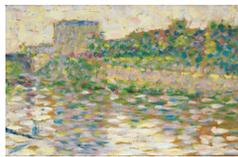


調整あり生成画像

図 73 原画像 (A4) と参照画像 (B3) の生成結果



原画像 (A4)



参照画像 (B4)



輝度調整画像



調整なし生成画像



調整あり生成画像

図 74 原画像 (A4) と参照画像 (B4) の生成結果



原画像 (A4)



参照画像 (B5)



輝度調整画像



調整なし生成画像



調整あり生成画像

図 75 原画像 (A4) と参照画像 (B5) の生成結果



原画像 (A4)



参照画像 (B6)



輝度調整画像



調整なし生成画像



調整あり生成画像

図 76 原画像 (A4) と参照画像 (B6) の生成結果

4.3 提案手法と従来手法との生成結果の比較

本節では、提案手法と従来手法 [10] [11] [16] [20] との画風変換結果の比較を行う。提案手法のパラメータは各画像ごとに調整し、必要と判断した場合には輝度調整を行った。

図 77 から図 83 に、本実験の結果を示す。まず全体の傾向について述べ、次に個別の結果を考察する。

まず、全体として、提案手法による画風変換は、パラメータの調整や輝度調整を行うことで概ね良好な結果を得た。一方で、輝度の類似度を用いることによるテクスチャの不適切な参照が行われ、同一オブジェクト内での不自然な色の変化が発生した。従来手法の生成結果と比較した場合に、提案手法でのみ再現した特徴と従来手法でのみ再現した特徴が存在している。そのため、生成結果についてはユーザの好みによる点が大きいと考えられる。

図 77 および図 78 は、Hertzmann ら [10] との比較結果である。比較結果 (A) は、水彩画風の果物の画像を参照画像とした変換結果である。提案手法では、画用紙風のテクスチャが良く再現されているが、空の部分で原画像の暗さに影響され、一部不自然な黒い領域が生成されている。従来手法では、参照画像に見られる画用紙風のテクスチャがやや弱まっているが、一方でオブジェクトのエッジ部分が強調されるという特徴を再現している。比較結果 (B) は、ブラシストロークがはっきりとしている夜景の画像を参照とした変換結果である。提案手法では、生成画像中にブラシストロークの一部は再現したが、原画像の暗い空領域にはブラシストロークが反映できておらず、また地面領域でもテクスチャの不連続が発生している。一方、従来手法では、生成画像全体に強い横方向のブラシストロークが再現されている。比較結果 (B) では、従来手法のほうがより良く参照画像の特徴を反映していると言える。これは、参照画像の色数が少なく、原画像の雲や地面に相当する部分が参照画像内にほぼ存在しなかったため生成画像全体のコントラストが低くなったことに加え、参照画像の特徴的なブラシストロークが特に空の明るい領域に存在し、暗い領域と対応が取れなかったためと考えられる。

図 79 は、Efros ら [11] との比較結果である。提案手法と従来手法双方でほぼ同等の生成結果が得られている。提案手法は、画像生成の際に原画像の輝度情報と

参照画像の輝度情報の双方を用いるため、原画像のテクスチャがそのまま現れている部分が一部見られる。また、原画像の輝度変化が反映されたことで、参照画像の持つのっぺりとした雰囲気をも十分に再現しているとは言えない。一方で、従来手法は、参照画像のパッチを利用するため原画像のテクスチャがそのまま表れる部分はないが、一方で目などの細かいテクスチャが潰れている。

図 80 から図 82 は、Lee ら [16] との比較結果である。比較結果 (A) は、ブラシストロークがはっきりとしている油絵の画像を参照画像とした画風変換である。提案手法では、塔の部分や黒枠の文字などが平滑化されている。また、背景の空の部分に対して参照画像の特徴的なブラシストロークの反映が不十分であるが、ストロークの合間に見られる異なる色の混合を反映している。また、原画像の雲の領域に対して特徴的なブラシストロークを反映している。一方で従来手法では、オブジェクトがはっきりとしており、特に空の部分で参照画像のブラシストロークを反映している。一方で、画像生成の際に参照画像の輝度を用いないため、生成画像は、同一領域で大きな色の変化を含んでいない。比較結果 (B) は、水彩画風の果物の画像を参照画像とした変換結果である。提案手法では、原画像中の黒縁の影響で画像の端付近で不自然なにじみが発生しており、また同一オブジェクト中の不自然な色変化も発生している。一方で、背景の空領域では参照画像に含まれる画用紙風のテクスチャを反映している。従来手法では、全体的に水彩画風の特徴が現れているが、画用紙風のテクスチャがはっきりと現れておらず、参照画像の特徴を十分に反映しているとは言えない。比較結果 (C) は、パステル調の画像を参照画像とした画風変換である。提案手法では、空の領域で参照画像のかすれたようなブラシストロークが反映されており、また雲の領域では参照画像の鳥領域の網掛けのようなテクスチャが反映されている。このことから、参照画像の広い部分のテクスチャが利用されていると言える。しかし、塔の領域に対しては参照画像のテクスチャが反映できていない。一方で、従来手法では、生成画像全体に参照画像のかすれたようなブラシストロークを反映しており、塔の領域に対してもこのブラシストロークを反映している。

図 83 は、市野ら [20] との比較結果である。提案手法では、植物の領域に対して細かいテクスチャを再現しており、また原画像の植物の色鮮やかさを保持してい

る。一方で、本来の物体とは異なる物体のテクスチャの参照が目立っており、特に、海領域に対して参照画像の空の領域のテクスチャが利用されている領域が目立つ。一方で従来手法では、海の領域に対してうまく海のテクスチャを再現することができており、領域ごとに参照するテクスチャを適切に参照できている。一方で、木など細かいテクスチャが平滑化されたようになっており、また適用するパッチ間の不連続が目立っている。



原画像



参照画像



提案手法の結果

$W = 3 \times 3$, $\alpha = 0.7$, 輝度調整あり



Hertzmann ら [10] の結果

図 77 Hertzmann ら [10] との比較 (A)



原画像



参照画像



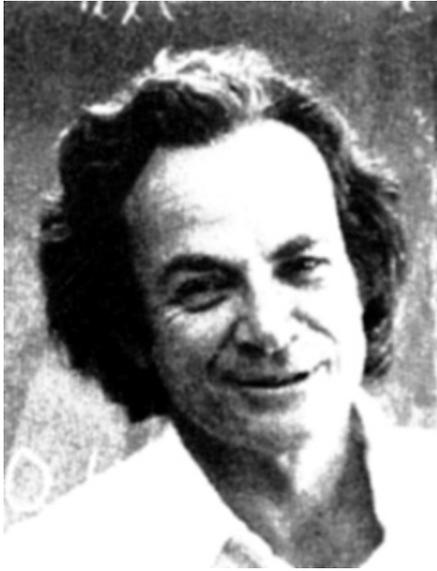
提案手法の結果

$W = 3 \times 3$, $\alpha = 0.9$, 輝度調整なし



Hertzmann ら [10] の結果

図 78 Hertzmann ら [10] との比較 (B)



原画像



参照画像



提案手法の結果



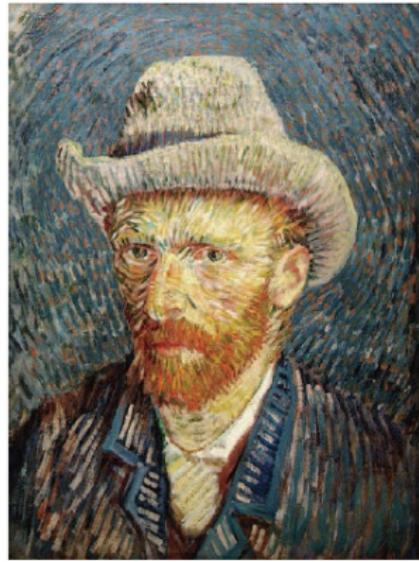
Efros ら [11] の結果

$W = 5 \times 5$, $\alpha = 0.5$, 輝度調整なし

図 79 Efros ら [11] との比較



原画像

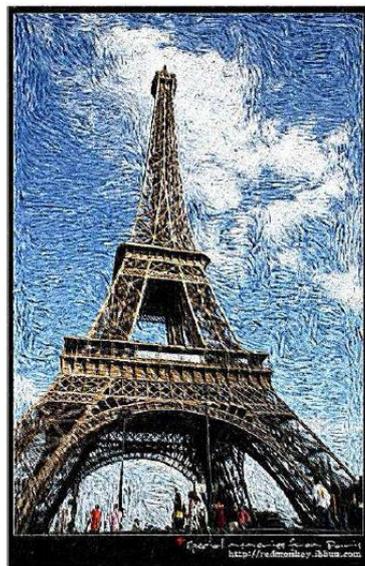


参照画像



提案手法の結果

$W = 3 \times 3$, $\alpha = 0.8$, 輝度調整なし



Lee ら [16] の結果

図 80 Lee ら [16] との比較 (A)



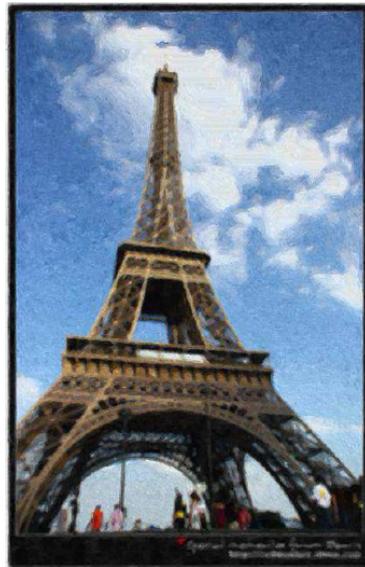
原画像



参照画像



提案手法の結果



Lee ら [16] の結果

$W = 3 \times 3$, $\alpha = 0.6$, 輝度調整なし

図 81 Lee ら [16] との比較 (B)



原画像



参照画像



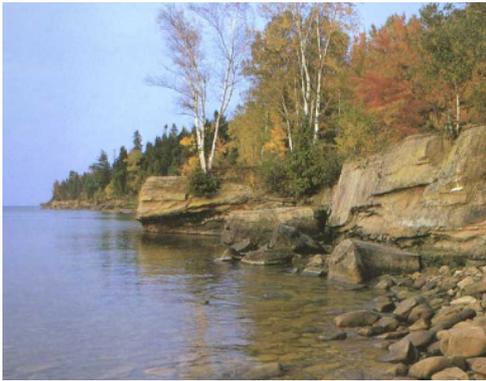
提案手法の結果

$W = 5 \times 5$, $\alpha = 0.2$, 輝度調整なし



Lee ら [16] の結果

図 82 Lee ら [16] との比較 (C)



原画像

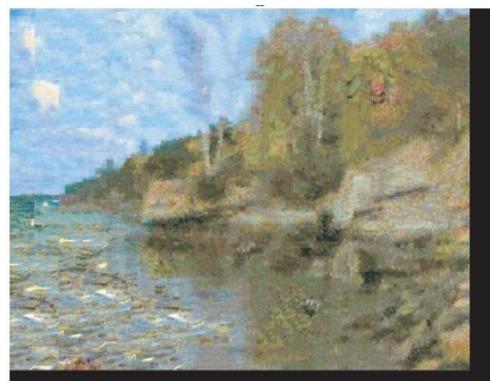


参照画像



提案手法の結果

$W = 3 \times 3$, $\alpha = 0.5$, 輝度調整なし



市野ら [20] の結果

図 83 市野ら [20] との比較

5. まとめ

本論文では、Painterly Rendering の主要なアプローチの中でも、事例参照型アプローチに着目し、生成画像と原画像間の輝度の類似度および生成画像と参照画像間のパターン類似度に基づくエネルギーの最小化によって、参照画像の画風を限定せずにテクスチャの不連続の発生を抑止する画風変換手法を提案した。実験では、重みやウインドウサイズを変更することで、反映するテクスチャのスケール調整や色調反映度の調整が可能であることを確認した。また、比較実験により、同一の手法で参照画像の画風を限定せず、良好な画風変換が可能であることを確認した。

一方で、提案手法はエネルギーを画像間の輝度情報を用いた類似度によって定義しているため、原画像や参照画像間に極端な輝度の差がある場合などにそのまま変換を行うと、参照画像の不適切なテクスチャを参照してしまい、良好な画風変換が行えない問題がある。これに対して、本論文では原画像の輝度を参照画像の輝度に合わせる輝度調整を事前に行うことでこの問題の解消を試み、実験によって一定の効果があることを示した。ただし、原画像中に、ハイライトなどがあった場合に、その色の変化によって同じオブジェクトの領域に不自然なテクスチャが生成される場合があり、輝度調整を行った場合にもこの問題は残されていることを確認した。このような不自然なテクスチャの生成を抑止するためには、Zeng ら [6] や Chang ら [18] が行ったように、原画像と参照画像双方のオブジェクト、あるいは色領域に対して事前にラベル付けを行うことでパターンを参照する部分を制約することが考えられる。

謝辞

本研究を進めるに当たり，終始暖かくご指導，ご鞭撻頂いた視覚情報メディア研究室横矢直和教授に心より感謝申し上げます。そして，本研究の遂行にあたり，有益な御助言，御鞭撻を頂いたインタラクティブメディア設計学研究室加藤博一教授に厚く御礼申し上げます。さらに，本研究を進めるに当たり，終始細やかなご指導，ご助言頂いた視覚情報メディア研究室佐藤智和准教授に厚く御礼申し上げます。終始暖かくご指導，ご鞭撻頂いた視覚情報メディア研究室河合紀彦助教，中島悠太助教に深く感謝致します。特に佐藤智和准教授，河合紀彦助教のお二方には，本学での研究におけるあらゆる面で，多大なご助言，ご協力を賜りましたことを深く感謝致します。また，研究室での生活を支えて頂いた視覚情報メディア研究室石谷由美女史に心より感謝いたします。

参考文献

- [1] Paul Haeberli. Paint by numbers: Abstract image representations. *ACM SIGGRAPH Computer Graphics.*, Vol. 24, No. 4, pp. 207–214, 1990.
- [2] Aaron Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pp. 453–460, 1998.
- [3] Aaron Hertzmann. Paint by relaxation. In *Proceedings of Computer Graphics International 2001*, CGI '01, pp. 47–54, 2001.
- [4] Levente Kovács and Tamás Szirányi. Painterly rendering controlled by multiscale image features. In *Proceedings of the 20th Spring Conference on Computer Graphics*, SCCG '04, pp. 177–184, 2004.
- [5] Tony Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, Vol. 30, No. 2, pp. 79–116, 1998.
- [6] Kun Zeng, Mingtian Zhao, Caiming Xiong, and Song-Chun Zhu. From image parsing to painterly rendering. *ACM Transaction on Graphics*, Vol. 29, No. 1, pp. 2:1–2:11, 2009.
- [7] Zhuowen Tu, Xiangrong Chen, Alan L. Yuille, and Song-Chun Zhu. Image parsing: Unifying segmentation, detection, and recognition. *International Journal of Computer Vision*, Vol. 63, No. 2, pp. 113–140, 2005.
- [8] Zhuowen Tu and Song-Chun Zhu. Parsing images into regions, curves, and curve groups. *International Journal of Computer Vision*, Vol. 69, No. 2, pp. 223–249, 2006.
- [9] Cheng-en Guo, Song-Chun Zhu, and Ying Nian Wu. Primal sketch: Integrating structure and texture. *Computer Vision and Image Understanding*, Vol. 106, No. 1, pp. 5–19, 2007.

- [10] 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, pp. 327–340, 2001.
- [11] 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, pp. 341–346, 2001.
- [12] James. Davis. Mosaics of scenes with moving objects. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR '98, pp. 354–360, 1998.
- [13] Michael Ashikhmin. Fast texture transfer. *IEEE Computer Graphics and Applications*, Vol. 23, No. 4, pp. 38–43, 2003.
- [14] Michael Ashikhmin. Synthesizing natural textures. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, I3D '01, pp. 217–226, 2001.
- [15] Kenji Hara, Kohei Inoue, and Kiichi Urahama. Adaptive image translation for painterly rendering. In *Proceedings of IAPR Conference on Machine Vision Applications*, pp. 566–579, 2005.
- [16] Hochang Lee, Sanghyun Seo, Seungtaek Ryoo, and Kyunghyun Yoon. Directional texture transfer. In *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering*, NPAR '10, pp. 43–48, 2010.
- [17] Henry Kang, Seungyong Lee, and Charles K. Chui. Coherent line drawing. In *Proceedings of the 5th International Symposium on Non-photorealistic Animation and Rendering*, NPAR '07, pp. 43–50, 2007.
- [18] Youngha Chang, Suguru Saito, Keiji Uchikawa, and Masayuki Nakajima. Example-based color stylization of images. *ACM Transactions on Applied Perception*, Vol. 2, No. 3, pp. 322–345, 2005.

- [19] Brent Berlin and Paul Kay. *Basic Color Terms: Their Universality and Evolution*. University of California Press, 1969.
- [20] 市野早徒, 西山昂志, 中村剛士, 加納政芳. テクスチャの局所性を考慮したエネルギー最小化による絵画風画像生成. 日本知能情報ファジィ学会 ファジィシステム シンポジウム 講演論文集, Vol. 27, pp. 281–281, 2011.
- [21] 河合紀彦, 佐藤智和, 横矢直和. テクスチャの明度変化と局所性を考慮したパターン類似度を用いたエネルギー最小化による画像修復. 電子情報通信学会論文誌. D, Vol. J91-D, No. 9, pp. 2293–2304, 2008.
- [22] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics*, Vol. 28, No. 3, pp. 24:1–24:11, 2009.