

ポアソン画像類推によるテクスチャーを考慮した画像合成

吉澤 信^{1,a)} 横田 秀夫^{1,b)}

概要: 二つの画像を境界が分からなく自然に合成するシームレスクローニングやコラージュと呼ばれる画像合成問題は、CG 分野にて様々な応用があり、その計算方法は重要な研究テーマである。もしも合成する画像間でテクスチャー細部のパターンが異なる場合に、既存のアプローチでは色合は自然でも、テクスチャー細部の不連続性により、自然な合成結果を生成しない事が知られている。本稿では、画像のテクスチャー細部とベースとなる低周波カラー情報を別々に処理する事により、テクスチャーを考慮した画像合成を生成する新しい計算フレームワークを提案する。提案フレームワークでは、まず画像の細部とベースを新たに開発したエッジ保存フィルタにより分離する。次に、ベースの色合はポアソン方程式を解く事により補間し、細部は画像類推法と呼ばれる例題に基づくテクスチャー合成法を用いて復元する。提案フレームワークにより、細部のパターンが異なる画像間でも写実的な画像合成に成功した。

キーワード: ポアソン画像合成, 画像類推, シームレスクローニング, エッジ保存フィルタ

Seamless Cloning with Texture Restoration via Poisson Image Analogy

YOSHIZAWA SHIN^{1,a)} YOKOTA HIDEO^{1,b)}

Abstract:

Seamless image cloning, i.e. synthesizing two images with seamless boundaries, has many interesting applications in CG and therefore important to investigate computational methods of it. For a given pair of images, if texture patterns of them are different then conventional approaches do not generate natural-looking seamless cloning results due to their detail differences. In this paper, we propose a new computational framework for texture-aware seamless image cloning. The framework separately processes the image details and its base color information. The separation is carried out by using a novel edge-preserving filter and then the base color is interpolated via solving the Poisson equation. The detail is restored by applying an example-based texture synthesis technique called image analogy. The framework provides realistic cloning results with seamless texture details.

Keywords: Poisson Image Editing, Image Analogy, Seamless Cloning, Edge-Aware Filtering

1. はじめに

画像合成は CG 分野にて多数の応用があり非常に重要である。画像合成には、同じ画像を繋ぎ目が判らない様に敷き詰めるテクスチャー合成などがあるが、本稿^{*1}では二つの異なる画像を境界が自然に見える様に合成するシームレスクローニング (Seamless Cloning) やコラージュと呼ばれる

画像合成問題を取り扱う。合成する画像間でテクスチャー細部のパターンが異なる場合に、よく用いられている既存のアプローチ [23] では、色合は自然でもテクスチャー細部の不連続性により、自然な合成結果を生成しない事が知られている [5], [26](図 1(b))。

本稿では、画像のテクスチャー細部とベースとなる低周波カラー情報を別々に処理する事により、テクスチャーを考慮した画像合成を生成する新しい計算フレームワークを提案する。提案フレームワークでは、まず画像の細部とベースを新たに開発したエッジ保存フィルタにより分離する。次に、ベースの色合はポアソン方程式を解く事により

¹ 画像情報処理研究チーム, 理化学研究所, 埼玉県和光市広沢 2-1 RIKEN, Hirosawa 2-1, Wako, Saitama, Japan

^{a)} shin@riken.jp

^{b)} hyokota@riken.jp

^{*1} 本稿は文献 [33] を拡張し、より詳細を記述したものである。

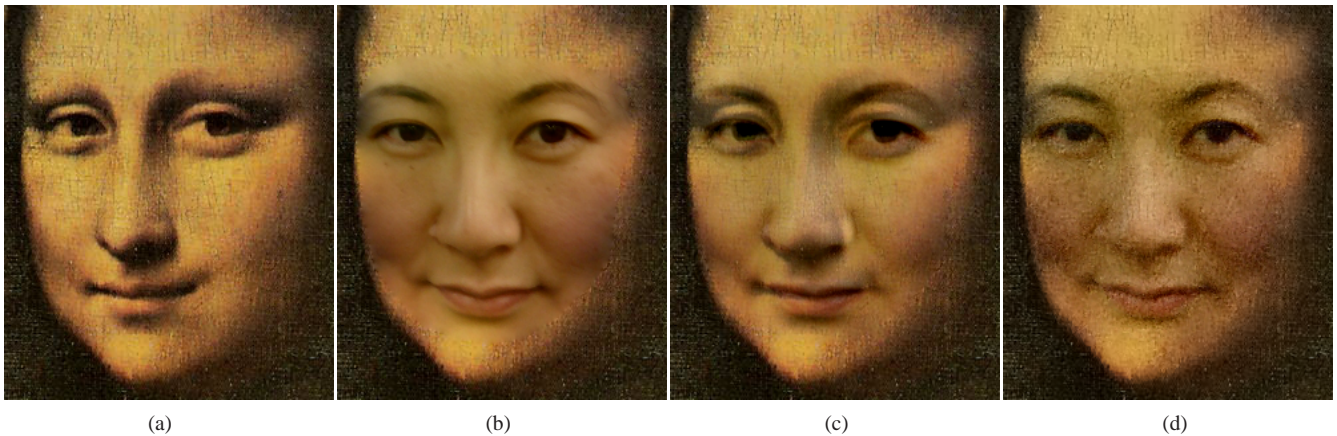


図1 テクスチャーを考慮した合成と比較. 図4中のソース画像とターゲット画像(図2(a))を合成した結果(b,c,d). (b,c):既存法(ポアソン画像合成)[23]により, (b)はソースの勾配のみ, (c)は混合勾配を用いた結果. (d):提案フレームワークを用いた結果.

補間し, 細部は画像類推法と呼ばれる例題に基づくテクスチャー合成法を用いて復元する. 本稿の学術的貢献は以下にまとめられる.

- テクスチャーを考慮した新しい画像合成フレームワーク(ポアソン画像類推法)の提案.
- エッジ保存平滑化法[11]への新たな L^2 距離計量の導入と高速ガウス畳み込み核の実装.

また, 数値実験により提案フレームワークを用いる事で細部のパターンが異なる画像間でも写実的な画像合成を達成した(図1参照).

2. 関連研究と問題点

二つの与えられたソース(Source)画像 S とターゲット(Target)画像 T , 及び S と T 上のそれぞれ対応する領域 Ω_S と領域 Ω_T を考える. シームレスクローニングとは, Ω_S 内のソース画像をターゲット画像中の Ω_T へ境界が自然に見える様に合成する事である. ソース画像を Ω_T にコピーしただけでは, S と T の色合いや幾何的構造の差から境界が特定できる不自然な合成結果となる(図2).

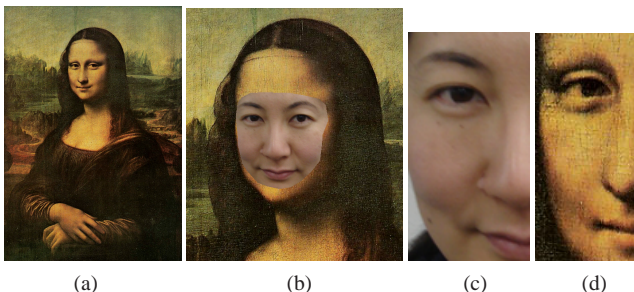


図2 ターゲット画像(a,d)とソース画像(c, 図4参照)を単純なコピー&ペーストによって合成した結果の拡大画像(b).

合成する画像の色合いや幾何構造が近い場合は, Ω_S 内のソース画像を境界からの距離に応じて S の透明度を変えて合成する Feathering や Alpha-matting と呼ばれる方法で

合成(アルファブレンド)する事で境界付近での違いが量された良い結果を得られる. アルファブレンドの距離や強度は S と T の境界付近に含まれる画像周波数によって変える事が望ましく, 多重解像度解析を用いて各周波数帯で異なる距離や強度で合成する方法が知られている[2]. 透明度の分布をポアソン方程式を解く事で半自動的に求める方法[29]なども提案されており, S と T の色合いが近い場合に有用である. 残念ながら, 図2に示す様な S と T の色合いや幾何構造が異なる場合に, アルファブレンドで写実的な合成結果を得る事は困難である.

Poisson 画像合成:

色合いの異なる場合に, ポアソン方程式を用いてカラー情報の補間を行うポアソン画像合成法[23]が有用な方法として知られている. ポアソン画像合成法は, ピクセル座標 $\mathbf{x} \in \Omega_T$ における合成後のカラー値を $I = I(\mathbf{x}) \in \mathbb{R}$ とすると, その画像勾配 ∇I とソース画像の勾配ベクトル $\mathbf{g} = \mathbf{g}(\mathbf{x}) \in \mathbb{R}^2$ の差を最小化する. 対応する Euler-Lagrange 方程式はポアソン方程式となる:

$$\min_I \int_{\Omega_T} |\nabla I - \mathbf{g}|^2 dx \rightarrow \Delta I = \text{div} \mathbf{g} \quad (1)$$

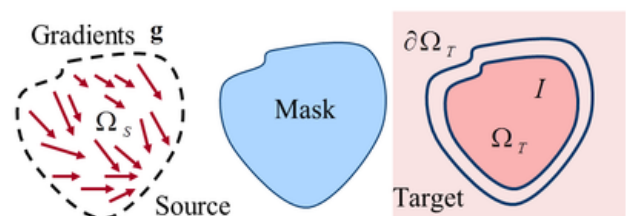


図3 合成に用いる領域 Ω_S と領域 Ω_T 及びその境界 $\partial\Omega_T$.

ここで ∇ , Δ , 及び div はそれぞれ勾配作用素, ラプラス作用素, 及び発散作用素(divergence), ベクトル \mathbf{g} は Ω_S 内

のソース画像勾配を Ω_T にコピーしたベクトルであり (図3参照), ターゲット画像のカラー情報を境界条件として用いる事で色合いの差が最小となる合成を与える.

ポアソン方程式による補間技術は, シームレスクローニングだけでなくパノラマ画像生成や Inpainting/completion など様々な合成問題に適応されている強力な道具であり, インタラクティブ応用に向けての高速計算法 [1], [9], [10], [19], [25] など, 多数の拡張や応用が提案されている. また, 画像だけではなく, 曲面メッシュの合成や変形など3次元形状への拡張 [24], [34] も提案されており, 2000年代以降のCG分野で最も華やかに研究されている技術の一つである.

ポアソン画像合成法がよく知られた問題点の一つは, Bleeding アーティファクトと呼ばれる合成境界での色混合である. これは, 合成境界にて色合いが滑らかに接続する大抵の場合には起こらないが, 合成境界に沿って不連続なエッジを構成したい場合に, そのエッジを跨いで色が混ざってしまう問題である. この色混合問題に対しては, ユーザーインターフェースを用いて反射境界条件など不連続性を持たせる事で色混合を防ぐ方法 [8], [20] や勾配の回転演算子強度も最小化に用いる方法 [27] など多数の有用な方法が提案されているため, 本稿では取り扱わない.

もう一つよく知られたポアソン画像合成法の問題点は, テクスチャー細部の不連続性により, 自然な合成結果を生成しない問題である [5], [26]. もしもソース画像 S とターゲット画像 T のテクスチャーの特性が異なる場合 (例えば滑らかな画像と細かい模様のある画像), ポアソン画像合成法による結果は不自然である. これは Ω_S と Ω_T の領域間の境界が, 色合いは滑らかに接続されていても, テクスチャー細部が異なるために際立ってしまうからである (図1(b)参照). 文献 [23] では, ソース画像とターゲット画像の両方のエッジ情報を合成後の画像に反映するために, S と T 両方の勾配を混ぜた混合勾配を g の代わりに式 (1) にて用いる方法も提案している. 残念ながら, 画像中に顕著な幾何特徴やパターンがある場合 (顔など) は, 図2(c)に示す様に, 期待される合成結果を得る事が困難である.

State of the Art 画像合成法:

上記テクスチャーの不連続性に起因する問題を克服する試みもいくつか提案されている [5], [26]. Sunkavalli ら [26] はランダムなノイズを多重解像度解析を用いて合成過程に導入する方法を提案した. この方法では, ポアソン方程式の代わりにヒストグラムマッチングと呼ばれる方法を色合いの補間に用いている. ターゲット画像 T のエッジ情報は用いられていないため, この方法では T の幾何特徴を合成結果に反映できない. Darabi ら [5] は画像の小領域 (パッチ) に基づいて色合いと勾配の変換も含めた差を最適化する方法を提案した. このパッチに基づく方法ではターゲット画像 T の勾配情報を用いているため, Sunkavalli らの方法 [26] と比べて, テクスチャーを考慮したより良い合成結

果を得ている. 本稿で提案するフレームワークと比べて非常に複雑な最適化プロセスを実装する必要がある事が難点である. また, 色合いとテクスチャー情報を変更する画像合成としてカムフラージュ画像生成 [3], [7] がある. 目的が異なるため, 既存のカムフラージュ画像生成法をシームレスクローニングに適応する事は容易ではない.

パッチを用いた方法 [5] と同様に, 曲線などの幾何構造に基づいた合成法 [17] も提案されている. この方法では画像のギャップを埋める補外が目的であり, 合成結果のテクスチャー細部を変更する目的には適さない. 色合いの補間に関しては, 統計的な情報を用いた Data-driven な方法 [30] も提案されているが, テクスチャーの不連続性は考慮されていない. 人間の顔に関する画像やビデオの合成問題に対しては, 3次元曲面のテンプレートや2次元の顔特徴線群などのモデルに基づく方法 [4], [31] も提案されている. これらは一般のシームレスクローニングへの適用は可能ではなくテクスチャーの不連続性も考慮されていない.

本稿では, 上記既存法群とは異なる新しい計算フレームワークを提案し, テクスチャーの不連続性に起因する問題の克服を試みている.

3. ポアソン画像類推法

本稿で提案する計算フレームワーク (ポアソン画像類推法) は, 画像のテクスチャー細部とベースとなる低周波カラー情報を別々に処理するという非常に簡単なアイデアに基づいている. 本章では, フレームワークの概要を記述し, 次章にてフレームワークの重要な要素技術である画像のテクスチャー細部とベースカラーの分離法を述べる. 図4は提案フレームワークの処理工程を表し, 大きく分けて下記の三つのステップで構成される.

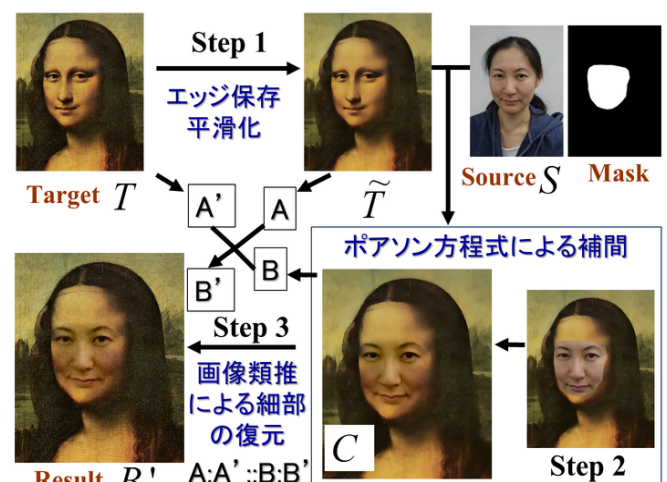


図4 ポアソン画像類推法の概要.

Step 1: ベース画像 \tilde{T} の抽出: ターゲット画像 T に対してエッジ保存平滑化 (第4章にて記述) を適用し, 平滑化さ

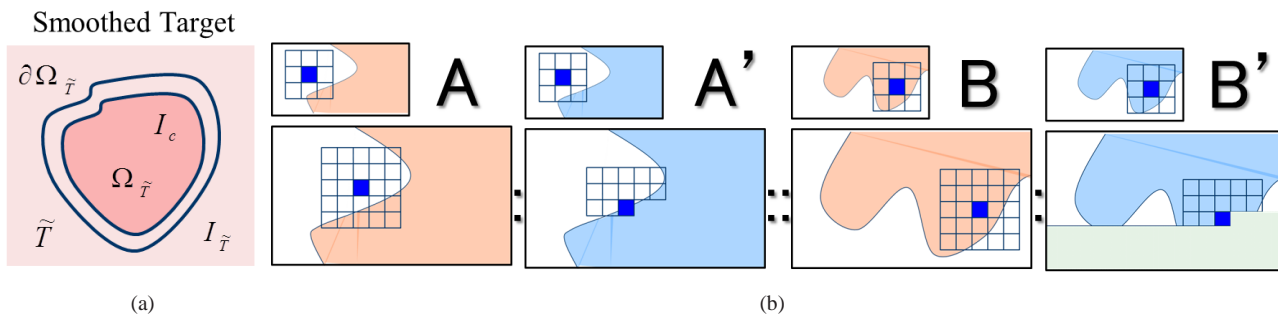


図5 (a):合成に用いる領域 $\Omega_{\tilde{T}}$ 及びその境界 $\partial\Omega_{\tilde{T}}$. 対応するソース画像領域 Ω_S は図3を参照. (b):画像類推法 [16]におけるパターンマッチング構造.

れたベース画像 \tilde{T} を計算する. この \tilde{T} をベースとなる低周波カラー情報として用いる. つまり T との差分 $T - \tilde{T}$ がエッジ情報を含むテクスチャー細部を表す (図6参照).

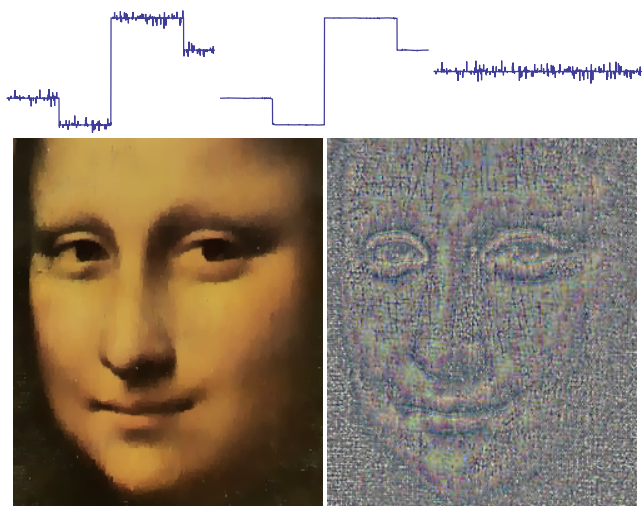


図6 エッジ保存平滑化によるベースカラー画像 \tilde{T} (左) とテクスチャー細部 $T - \tilde{T}$ (右). ここで対応する T は図1の画像 (a).

Step 2: 中間合成画像 C の生成: ベース画像 \tilde{T} 上の Ω_T に対応する領域 $\Omega_{\tilde{T}}$ とその境界 $\partial\Omega_{\tilde{T}}$ を考える (図5(a)参照). 中間合成画像 C のピクセル座標 $\mathbf{x} \in \Omega_{\tilde{T}}$ におけるカラー値を $I_c = I_c(\mathbf{x}) \in \mathbb{R}$ とすると, I_c はポアソン方程式 $\Delta I_c = \text{div } \mathbf{g}$ を境界条件 $I_c|_{\partial\Omega_{\tilde{T}}} = I_{\tilde{T}}$ を用いてカラーチャンネル毎に解く事で求める. ここで \mathbf{g} はソース画像 S の領域 Ω_S 内の勾配ベクトルをベース画像 \tilde{T} の対応する領域 $\Omega_{\tilde{T}}$ にコピーしたベクトル, $I_{\tilde{T}} \in \mathbb{R}$ は \tilde{T} のカラー値である. 中間画像 C はソース画像 S とベース画像 \tilde{T} の各色に対して勾配差 $\int_{\Omega_{\tilde{T}}} |\nabla I_c - \mathbf{g}|^2 dx$ を最小化し, 色合いが補間された画像である.

Step 3: 細部を復元した合成画像 B' の計算: 最終的な合成結果画像 B' は画像類推法 [16](Image Analogy) と呼ばれる例題に基づくテクスチャー合成法を用いて計算する. 画像類推法は, 図5(b)に示す構造を用いてパターンマッチングを多重解像度で適用する事により, 与えられた二つの例題画像 A 及び A' の A から A' への変化のエフェクトを

入力画像 B に与え, 合成画像 B' を生成する. この結果, B から B' への変化は A から A' の変化を模倣し, 「 A が A' となるならば, B は B' になる」という画像の類推関係 $A : A' :: B : B'$ に対応する. 提案フレームワークでは, $A = \tilde{T}$, $A' = T$, 及び $B = C$ とする事で, \tilde{T} から T へ細分が復元する効果をベース画像 C に付加し, 最終合成画像 B' を得る. つまり $\tilde{T} : T :: C : B'$ とする事で, B' はターゲット画像 T のテクスチャー細部が考慮された結果となる. 本稿で提案しているフレームワークは, 画像類推法のまったく新しい応用とも考えられる. 以下, 本稿で用いた画像類推法 [16](Artistic フィルタ) のアルゴリズムを簡単に紹介する.

画像類推法:

- (1) 与えられた三つの画像 A , A' , 及び B に対して Gaussian Pyramid[2] を構成する.
- (2) A , A' , 及び B の各画素に対して, 高次元特徴ベクトルを図5(b)に示す構造をユーザーが指定する半径 $r \geq 1$ を用いて生成する. Pyramid の各階層では $(2r+1)^2$ の局所窓及び一つ低い階層の $(2(r-1)+1)^2$ の局所窓を用いる. 特徴ベクトルの各要素は, 図5(b)に示す構造の各画素値を全てのカラーチャンネルでつなぎ合わせて構成する.
- (3) 特徴ベクトルを A と A' で連結し, 高次元検索構造を ANN ライブラリー [22] を用いて初期化する.
- (4) Pyramid の低解像度から高解像度へのループ: 各階層にて, B' の各ピクセル \mathbf{x} に対してのループ:

B と B' の特徴ベクトルを Query とし, BestMatch となる画素の A' の値を B' の対応する画素の値とする. また, その画素座標を保存しておく.

BestMatch:

ApproximationMatch となる画素を \mathbf{x}_a とし Coherence-Match となる画素を \mathbf{x}_c とする. 階層 l での特徴ベクトルを $\mathbf{F}_l(\cdot)$ とし, 画素 \mathbf{x} との L^2 距離をそれぞれ $d_a = |\mathbf{F}_l(\mathbf{x}_a) - \mathbf{F}_l(\mathbf{x})|^2$ と $d_c = |\mathbf{F}_l(\mathbf{x}_c) - \mathbf{F}_l(\mathbf{x})|^2$ とすると, Best-Match は以下の画素を与える.

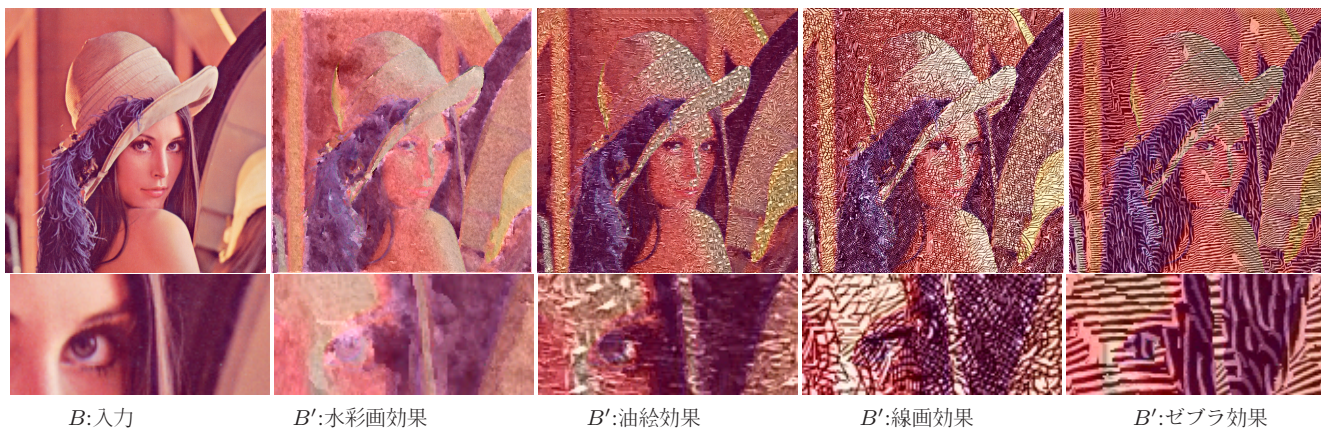


図7 図8をA及びA'の効果画像, 左端を入力画像Bとした場合の, 画像類推法[16]による Artistic フィルタの結果画像B'. テクスチャー細部が与えた効果を再現している.

$$\begin{cases} \mathbf{x}_c & \text{if } d_c \leq d_a(1 + 2^{l-L}k) \\ \mathbf{x}_a & \text{Otherwise.} \end{cases}$$

ここで L は Pyramid 階層の最大レベル, k はユーザーが与える Coherence パラメーターである. この k が高いほど前にマッチした画素の周辺の結果が得られ, 合成結果にテクスチャーパターンの連続性が高まる. また, ApproximationMatch は全ての画素から最も距離が近い画素: $\operatorname{argmin}_{\mathbf{y}} |F_l(\mathbf{x}) - F_l(\mathbf{y})|^2$ を与える. CoherenceMatch は

$$\operatorname{argmin}_{\mathbf{y} \in N(\mathbf{x})} |F_l(s(\mathbf{y}) + (\mathbf{x} - \mathbf{y})) - F_l(\mathbf{x})|^2$$

を与える. ここで, $N(\mathbf{x})$ は \mathbf{x} の近傍で既に合成された画素の集合, $s(\mathbf{y})$ は画素 \mathbf{y} に対して BestMatch した画素である. この式により, B' の既に合成された部分とテクスチャーのつながりがより自然 (Coherent) な合成が行われる.

実装の詳細は文献 [16] を参照のこと. また, 画像類推法のソフトウェアは文献 [16] の著者 HP^{*2}にて配布されている.

図7は図8をエフェクトとして, 本稿で用いた画像類推法による $A : A' :: B : B'$ のフィルタ例である. テクスチャー細部のエッジや幾何パターンを与えたエフェクトに基づいて再現している事がわかる.

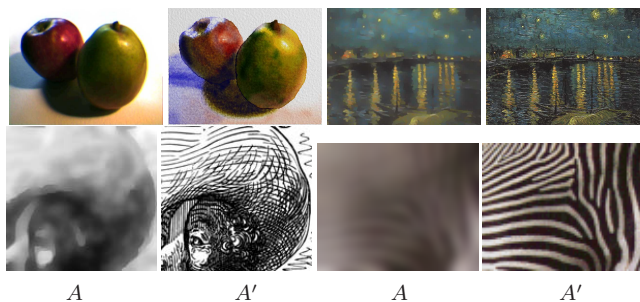


図8 図7の入力画像. 左上:水彩画, 右上:油絵, 左下:線画, 及び右下:ゼブラ模様のエフェクト画像 A : A'.

4. L^2 距離及びガウス核を用いた定義域変換

本章では, 前章で紹介した提案フレームワークの Step 1 にてターゲット画像 T からベース画像 \tilde{T} を生成するために用いたエッジ保存平滑化法を記述する. エッジを保存する平滑化には様々な方法 [12], [32] があるが, 本稿では定義域変換法 [11] と呼ばれる方法を新たに拡張した方法を用いる. 定義域変換法は, 現在提案されている高速エッジ保存平滑化フィルタのなかでも, フィルタ結果のクオリティーと計算速度のバランスが良く, CG 応用に適している [12].

定義域変換法は画素の座標値を 1 次元画像多様体上の測地距離を用いて変換し, その変換された定義域で線形平滑化フィルタをカラー値に対して適用する. 初めに画素座標 \mathbf{x} での正規化畳み込みによる平滑化

$$\frac{1}{\int f(|\mathbf{x} - \mathbf{y}|) d\mathbf{y}} \int f(|\mathbf{x} - \mathbf{y}|) \mathbf{I}(\mathbf{y}) d\mathbf{y}, \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^2$$

を考える. ここで, \mathbf{x}, \mathbf{y} は画素座標, $|\mathbf{x}|$ は \mathbf{x} の L^1 ノルム, $\mathbf{I} = \mathbf{I}(\mathbf{x}) \in \mathbb{R}^3$ は対応する画素のカラーベクトル, 及び $f = f(x) \in \mathbb{R}$ は $x \in \mathbb{R}$ が $|x| \rightarrow \infty$ にて零となる畳み込み核である. 畳み込み核 f は通常, 係数なしガウス関数 $G_\sigma = G_\sigma(x) = \exp(-\frac{x^2}{2\sigma^2})$ 等を用いて, 線形平滑化フィルタを構成する. エッジを保存する平滑化フィルタを構成するためには, f が画像のエッジ部分で小さく, 逆に平坦な部分で大きな値を取る様なデータ依存の非線形な関数 (又は定義域) を採用する事で可能であり, Bilateral フィルタ $f = f_B \equiv G_\sigma(|\mathbf{x} - \mathbf{y}|) G_\rho(|\mathbf{I}(\mathbf{x}) - \mathbf{I}(\mathbf{y})|)$ などの様々な f が提案されている. 画像多様体上の測地線の長さを畳み込み核の変数 (定義域) として用いる事でもエッジを保存するフィルタを構成できる.

画像多様体 S の頂点は, スケーリングパラメーター $\lambda \in \mathbb{R}$ を用いて $\mathbf{p} = (\mathbf{x}, \lambda \mathbf{I}(\mathbf{x})) \in \mathbb{R}^5$ で与えられる. \mathbf{x} の近傍座標 $\mathbf{y} \in \mathbb{R}^2$ に対応する S の頂点を $\mathbf{q} = (\mathbf{y}, \lambda \mathbf{I}(\mathbf{y}))$ とすると, S 上の二点間 \mathbf{p} と \mathbf{q} の測地距離 $\operatorname{Dist}(\mathbf{p}, \mathbf{q})$ は, 画像のエッジ部分で大きく, \mathbb{R}^2 平面に平行な平坦な部分で小さい. つ

*2 <http://mrl.nyu.edu/projects/image/-analogies/lf/>

まり, $f = G_\sigma(\text{Dist}(\mathbf{p}, \mathbf{q}))$ はエッジ保存平滑化フィルタを構成する. 実際, Bilateral フィルタの距離は

$$f_B = \exp\left(-\left(\frac{|\mathbf{x} - \mathbf{y}|^2}{2\sigma^2} + \frac{|\mathbf{I}(\mathbf{x}) - \mathbf{I}(\mathbf{y})|^2}{2\varphi^2}\right)\right) = G_\sigma(|\mathbf{p} - \mathbf{q}|)$$

であるため, $\lambda = \frac{\sigma}{\varphi}$ 及びユークリッド距離を用いた測地距離 $\text{Dist}(\mathbf{p}, \mathbf{q})$ の近似となる (図9 参照).

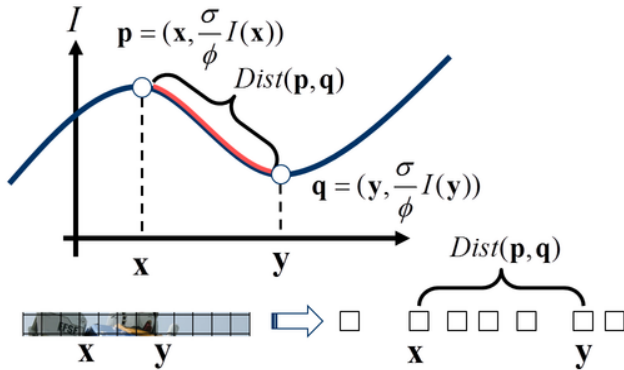


図9 定義域変換の概念図.

媒介変数 $t \in \mathbb{R}$ を用いて \mathcal{S} 上の測地線を $\mathbf{r}(t) = (\mathbf{x}(t), \lambda \mathbf{I}(\mathbf{x}(t))) \in \mathbb{R}^5$, $\mathbf{x}(t) = (u(t), v(t))$ とし, 二点 \mathbf{p} と \mathbf{q} が $\mathbf{p} = \mathbf{r}(t_p)$ 及び $\mathbf{q} = \mathbf{r}(t_q)$ と媒介変数化されているとする. 測地線の L^2 長さ $s_{L^2}(\mathbf{p}, \mathbf{q}) \in \mathbb{R}$ は $\mathbf{r}(t)$ の接線ベクトル $\frac{d\mathbf{x}}{dt}$ を \mathcal{S} の接平面上で積分する事で与えられる:

$$\begin{aligned} s_{L^2}(\mathbf{p}, \mathbf{q}) &= \int_{t_p}^{t_q} \sqrt{|J \frac{d\mathbf{x}}{dt}|^2} dt = \int_{t_p}^{t_q} \sqrt{\left(\frac{d\mathbf{x}}{dt}\right)^T J^T J \frac{d\mathbf{x}}{dt}} dt \\ &= \int_{t_p}^{t_q} \sqrt{E \left(\frac{du}{dt}\right)^2 + 2F \left(\frac{du}{dt}\right) \left(\frac{dv}{dt}\right) + G \left(\frac{dv}{dt}\right)^2} dt \end{aligned}$$

ここで $E = \left(\frac{\partial \mathbf{r}}{\partial u}\right)^2$, $F = \frac{\partial \mathbf{r}}{\partial u} \cdot \frac{\partial \mathbf{r}}{\partial v}$, 及び $G = \left(\frac{\partial \mathbf{r}}{\partial v}\right)^2$ は \mathcal{S} の第一次規格量の係数であり, \mathbf{r} の Jacobi 行列 J の随伴行列 $J^T J$ で与えられる (リーマンの共変計量). $J \frac{d\mathbf{x}}{dt}$ は \mathcal{S} の接平面上で表された \mathbf{r} の接線ベクトルである. \mathbf{r} を代入すると,

$$E = 1 + \left(\lambda \frac{\partial \mathbf{I}}{\partial u}\right)^2, \quad F = \lambda^2 \frac{\partial \mathbf{I}}{\partial u} \cdot \frac{\partial \mathbf{I}}{\partial v}, \quad G = 1 + \left(\lambda \frac{\partial \mathbf{I}}{\partial v}\right)^2$$

より, \mathbf{I} の勾配 $\nabla \mathbf{I} = \left(\frac{\partial \mathbf{I}}{\partial u}, \frac{\partial \mathbf{I}}{\partial v}\right) \in \mathbb{R}^{2 \times 3}$ を用いて,

$$s_{L^2}(\mathbf{p}, \mathbf{q}) = \int_{t_p}^{t_q} \sqrt{\left(\frac{d\mathbf{x}}{dt}\right)^2 + \lambda^2 |\nabla \mathbf{I} \frac{d\mathbf{x}}{dt}|^2} dt \quad (2)$$

を得る.

測地線を全ての \mathbf{p} と \mathbf{q} の組み合わせで計算し, $\text{Dist}(\mathbf{p}, \mathbf{q}) = s_{L^2}(\mathbf{p}, \mathbf{q})$ とするのは非効率的である. ここで $s_{L^2}(t_p, t_q) = s_{L^2}(T(\mathbf{p}), T(\mathbf{q}))$ となる変換 $T: \mathbb{R}^5 \rightarrow \mathbb{R}$ を求める事が出来れば, 変換後の空間で一次元の線形畳み込みを適用するだけでよい. 残念ながらその様な等長変換は非常に限られた場合にしか存在しない (ガウス曲率が同一の多様体間のみ). しかし, 一次元の画像多様体では, 距離

(長さ) を保存する等長写像を構成可能である. それゆえ, 文献 [11] では画素座標の各次元別に, 一次元の L^1 距離を用いた等長変換 $\int (1 + \frac{\sigma}{\varphi} |\nabla \mathbf{I}(t)|) dt$ を適用し, 次元毎に一次元畳み込みを交互に行う分離実装を提案している.

本稿では, 式 (2) にて $\mathbf{x} \rightarrow t$ とする事で, 新たに導出した L^2 距離による定義域変換

$$T(\mathbf{p}) = \int_0^{t_p} \sqrt{1 + \lambda^2 |\nabla \mathbf{I}(t)|^2} dt \quad (3)$$

を用いる. ここで, $\lambda = \lambda(\sigma, \varphi)$ は平滑化量と保存したいエッジの大きさを調節するパラメータ, $\nabla \mathbf{I}$ はカラーベクトルの一次元方向への勾配である (u 又は v 方向への一階微分). この L^2 距離による定式化は, L^1 距離と比べてより直感的である. 本稿の数値実験では $\lambda = \frac{\sigma}{\varphi}$ を用いている*3

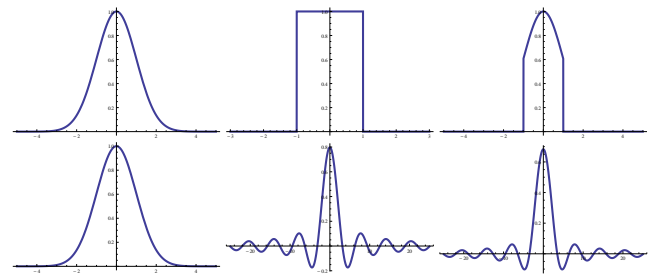


図10 上段は, ガウス関数 (左), Box 関数 (中), 及びガウス関数の打ち切り関数 (右). 下段は対応するフーリエ変換の結果.

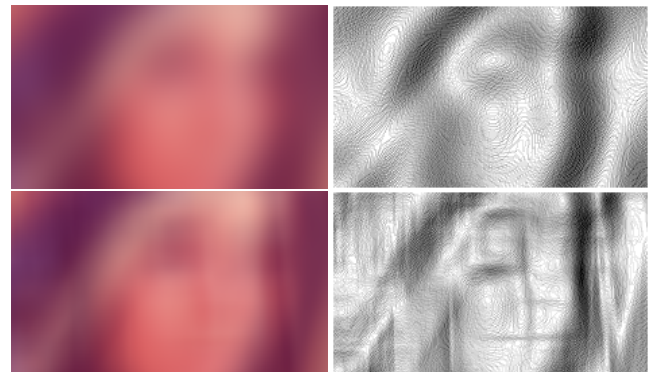


図11 ガウス関数 (上) と Box 関数 (下) による平滑化例. 入力画像は図7の左端画像. 右画像は左画像の勾配強度.

また文献 [11] では, 畳み込み核 f に Box 関数を用いて, 移動平均法 [6] と呼ばれる方法で高速に一次元畳み込みを実装している. Box 関数を用いた平滑化は, そのフーリエ変換が Sinc 関数となるため, 抑制したい高周波帯の制御が困難である (図10 参照). Sinc 関数は振動しながら減衰しているため, 抑制したい周波数帯より高い周波数帯が残り, 図11に示す様なアーティファクトの原因となる. それゆ

*3 定義域変換の場合には, Bilateral フィルタなど L^2 距離に応じた定式化がされている既存のエッジ保存平滑化フィルタのパラメータと比べて平滑化量のパラメータ σ の感度が悪い. それゆえエッジ保存平滑化だけを考えた場合は, $\lambda = \sqrt{\sigma/\varphi}$ 及び G_σ を用いる事で σ の感度を上げる事ができる.

え、本稿ではガウス関数を畳み込み核に用いる。ガウス関数のフーリエ変換はガウス関数となり、低周波から高周波へ滑らかに減衰しているため、周波数帯の抑制が直感的である。ガウス関数による畳み込みは、大域的な積分が必要であり、少なくとも 3σ 以上の幅で打ち切らないと、図 10 右画像の様に Box 関数と同様の結果となる。

残念ながら移動平均法は、ガウス関数には適用できない。本稿では次節 4.1 で紹介する高速ガウス変換と呼ばれる数値解析法を定義域変換法に適用する。最終的なエッジ保存平滑化フィルタは下記の非線形正規化畳み込み式 (4) で与えられる。

$$\frac{\int G_\sigma(|T(\mathbf{p}) - T(\mathbf{q})|)\mathbf{I}(\mathbf{x})d\mathbf{y}}{\int G_\sigma(|T(\mathbf{p}) - T(\mathbf{q})|)d\mathbf{y}} \quad (4)$$

実際の実装では、式 (3) を用いて u 及び v 方向 (画像の縦横) への定義域変換を行っておき、 u 及び v 方向の一次元畳み込みを分離実装により交互に実行する。文献 [11] と同様に、分離実装によるアーティファクトを軽減するため、式 (4) の σ を (u, v の組み合わせ一回に対して) $\sigma_i = \sigma\sqrt{3}(2^{V-i})/(\sqrt{4^V-1})$, $i = 1, 2, \dots, V$ と変化させて V 回繰り返し適用する。 L^1 距離と Box 畳み込み核による定義域変換法の実装は文献 [11] の著者 HP^{*4}にて配布されている。

4.1 高速ガウス変換

本節では式 (4) のガウス関数 G_σ による畳み込みを高速近似計算する方法を記述する。本稿で用いる高速ガウス変換 [14], [15] は、高速多重極展開 [13](Fast Multipole Method) をガウス関数に特化させたものである。近似精度を解析的に保証するため、計算物理学やビジョンで高速かつ高精度にガウス関数による畳み込みを実行するために幅広く用いられており、高速 Bilateral フィルタ [32] などエッジ保存平滑化にも適用されている。通常は要素数の二乗に比例する計算量が必要な畳み込み操作を線形の計算量で近似する事が可能である。均一なサンプル点に対してしか適用できない通常の高速フーリエ変換などと比べて、高速ガウス変換は非均一なサンプルに対しても適用できるため、式 (3) によって定義域変換された (非均一な) 座標に対して高速処理が可能である。

高速ガウス変換は、初めに与えられたソースとターゲットの二つの点群を領域分割する。本稿では、ソースとターゲット共に式 (3) で生成された一次元点群 $T(\mathbf{p})$ を用いる。次に、区画の中心点を代表点として近傍と遠方で以下のように関数展開を行う事で、ガウス関数による畳み込みを近似する。

一次元数直線上にてソースとターゲットの位置を $s, t \in \mathbb{R}$ とし、対応する区画の中心点をそれぞれ s_0 及び t_0 とすると (図 12 参照), ガウス関数 $G_\sigma(t-s)$ の s_0 を中心とした遠方での関数展開はエルミート展開

$$\sum_{i=0}^{\infty} \left[\frac{1}{i!} \left(\frac{s-s_0}{\sigma} \right)^i \right] h_i \left(\frac{t-s_0}{\sigma} \right) = \sum_{i=0}^{\infty} A_i h_i \left(\frac{t-s_0}{\sigma} \right).$$

で与えられる。ここで $h_i(x) = (-1)^i \frac{d^i}{dx^i} \exp(-x^2)$ はエルミート関数である。次に t と s を入れ替える事で、近傍のターゲット t_0 を中心とした近傍での関数展開はテイラー展開

$$G_\sigma(t-s) = \sum_{j=0}^{\infty} \left[\frac{1}{j!} h_j \left(\frac{s-t_0}{\sigma} \right) \right] \left(\frac{t-t_0}{\sigma} \right)^j = \sum_{j=0}^{\infty} B_j \left(\frac{t-t_0}{\sigma} \right)^j,$$

$$B_j = \frac{(-1)^j}{j!} \sum_{i=0}^{\infty} A_i h_{i+j} \left(\frac{s_0-t_0}{\sigma} \right).$$

にて表せる。区画内にて代表点との畳み込み結果を事前に計算しておく事で高速な近似が可能である。ガウス関数は零に指数レートで収束する。それゆえ、与えられた誤差 (精度) 制御を満たすのに必要な上記展開式の項数は、ある限られた数で打ち切る事が可能である。高速ガウス変換の大きな特徴は、畳み込みを実行する前に、この打ち切り項数を決定できる点である。本稿では、文献 [28] の誤差推定を用いる事で打ち切り項数を自動的に決定した。高速ガウス変換のソースコードは文献 [18] の HP^{*5}にて配布されている。

図 13 及び図 14 は本稿で提案した L^2 距離及びガウス核を用いた定義域変換によるエッジ保存平滑化例である。

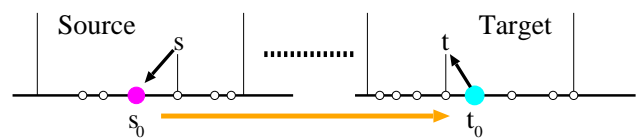


図 12 高速ガウス変換の概念図。



図 13 エッジ保存平滑化例。左上は入力画像。 $\sigma = 100$ で右上 $\varphi = 1.0$, 左下 $\varphi = 1.5$, 及び右下 $\varphi = 2.0$ を用いた結果。

^{*4} <http://inf.ufrgs.br/~eslgastal/DomainTransform/>
他にも、吉澤 信著、「高速エッジ保存平滑化フィルタ」, Computer Graphics Gems JP 2012 - コンピュータグラフィックス技術の最前線-, Chapter 5, pp. 89-120, 2012 には、日本語の解説とソースコードがある。

^{*5} www.ics.uci.edu/~ihler/code



図 14 エッジ保存平滑化例。左は入力画像。右は $\sigma = 100$ 及び $\varphi = 1.0$ を用いた結果。

5. 数値実験と考察

本章では提案したフレームワークを用いた合成結果、要素技術の性能評価、及び制約と今後の課題を述べる。

実験条件と数値解法:

本稿の数値実験は全て GNU C++ (g++4.4.3) を用いて実装し、Core i7 X990 CPU 及び 24G Byte RAM の PC で実行した。並列化や GPU は用いていない。ポアソン方程式は文献 [23] と同様に差分法と Gauss-Seidel 法を用いて解いた。式 (3) の勾配も差分近似を行った (前進一次)。ANN ライブラリーの誤差パラメーターは零、定義域変換の繰り返し回数 $V = 3$ 、高速ガウス変換の誤差パラメーターは 1.0、及び近傍と遠方の重なりを制御する IRC パラメーターは文献 [14] の推奨どおり $IRC = 6$ とした。エッジの大きさと平滑化量を同時に調節するために、定義域変換にて $\sigma = \varphi$ (つまり $\lambda = 1$ 及び G_φ) を合成に用いた。また、RGB カラー値の標準偏差の平均を α とし、 $\varphi = \alpha\phi$ の ϕ を調節した。合成毎に調節が必要なパラメーターは定義域変換の ϕ 、画像類推の Coherence パラメーター k 、及びマッチング半径 r であるが、本稿では $\phi = k = 1$ 及び $r = 2$ で良い結果を得られた。

計算速度:

図 15 は本稿で提案した L^2 距離及び高速ガウス変換を用いた定義域変換の計算速度のプロットである。パラメーターは $\phi = \{0.2, 0.4, 0.6, 0.8, 1.0\}$ 及び $\sigma = \{5, 15, 25, 35, 45, 55\}$ の 30 種類を用いた。エッジを保存する平滑化処理が、画像サイズに線形比例の計算時間で高速に適用出来ている。また、高速ガウス変換を用いているため、パラメータに適應した計算速度が達成されている (平均 30K から 50K 画素/秒)。

ポアソン方程式と画像類推の計算時間は合成する画像に依存する。提案フレームワーク全体では、 1024^2 程度の大きさの画像であれば 2 分程度の計算時間である。例え

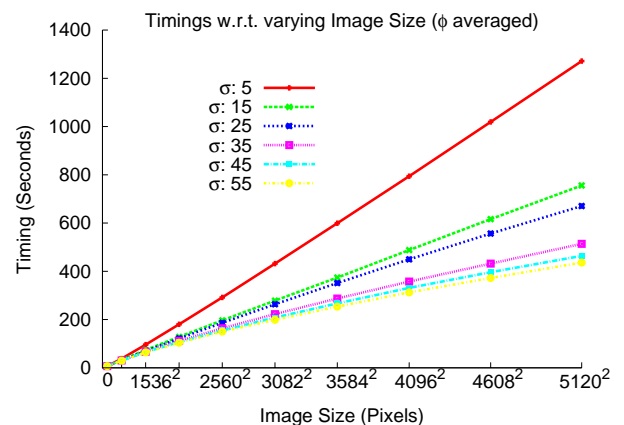
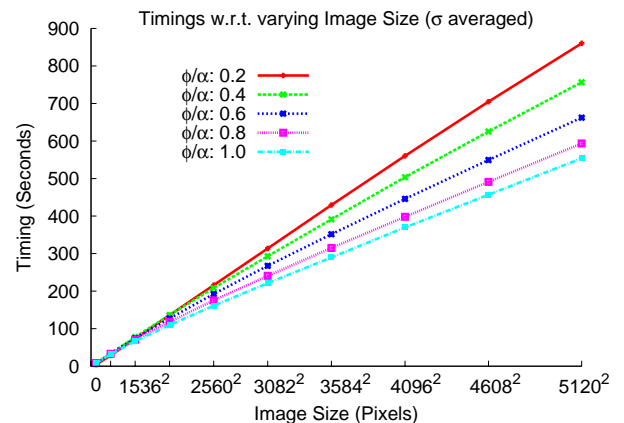


図 15 定義域変換の計算速度。縦横の軸はそれぞれ計算時間 (秒) 及び画像サイズ。上下の図はそれぞれ σ と φ の計算時間を平均したプロットである。

ば、図 1 の結果 (1024×1450 pixels) を得るのに定義域変換 58 秒、ポアソン方程式の解法 6 秒、及び画像類推 29 秒である。画像類推の部分は、文献 [21] などの導入により今後高速化を行いたい。

合成結果と比較:

図 1 及び図 16 は提案フレームワークと既存のポアソン画像合成法 [23] で合成した結果の比較である。ソース画像とターゲット画像のテクスチャー細部の差がある入力を用いているため、既存法では合成の境界が目視できる。提案フレームワークによる結果は、定義域変換法で分離したテクスチャー細部が画像類推により復元されており境界が自然に接続された写実的な結果である。テクスチャーを考慮したシームレスクローニングを試みている最新の方法 [5], [26] との詳細な数値実験比較は今後の課題である。

絵画などの非写実的画像へ実写など写実的画像を合成する場合に提案フレームワークは他の方法と比べ優位である (図 18 参照)。

Limitations:

提案フレームワークでは、テクスチャー細部とベース画像の二つの周波数帯しか用いていない。それゆえ、復元したいテクスチャーが複数の周波数帯から構成される場合や画像の幾何構造が大域的な場合には分離するテクスチャー

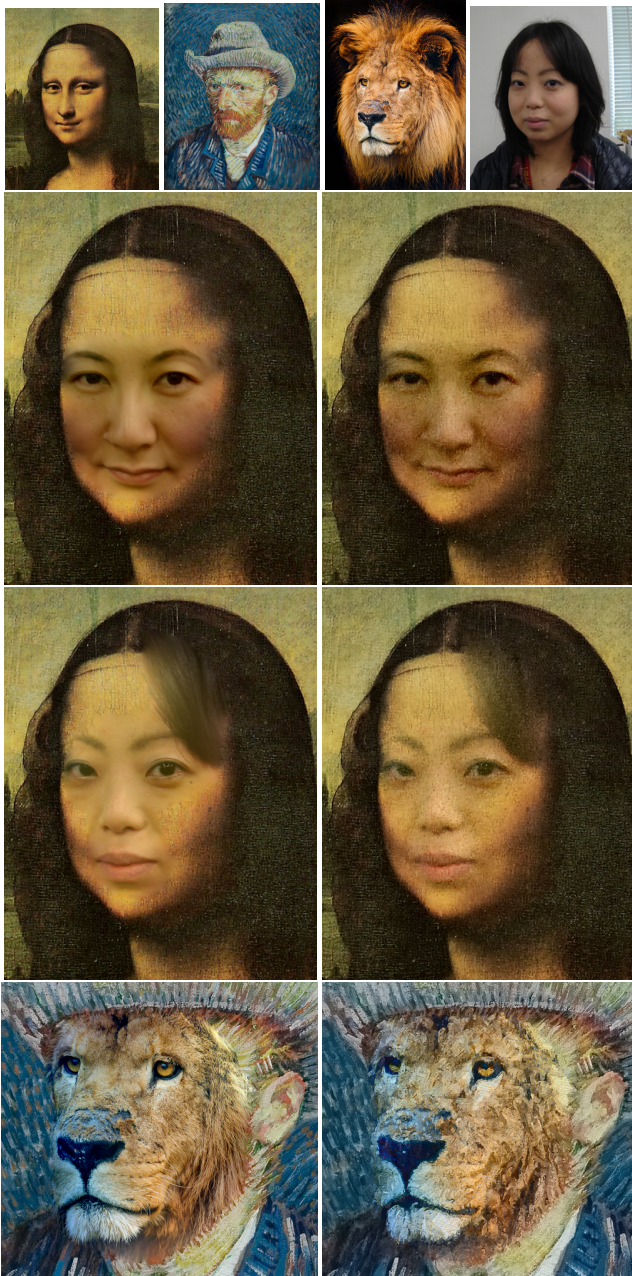


図 16 既存法 [23](左)と提案フレームワーク (右)を用いた合成結果. 上段は入力画像. 提案フレームワーク ($\phi = 1$, $r = 2$, 及び $k = 1$)により, テクスチャ細部が写実的に復元されている.



図 17 既存法 [23](左)と提案フレームワーク (右)を用いた合成結果. 提案フレームワーク ($\phi = 1$, $r = 2$, 及び $k = 1$)により, 細かいテクスチャは復元されているが, ライオンの毛の様な大域的な幾何構造は再現されていない.

細部を選ぶのが困難である (図 17 参照). この制約に対処するため, 定義域変換に用いる ϕ を入力画像から自動的に決定する方法, テクスチャの分離をマルチスケールで行う拡張やパッチに基づく方法 [5] との融合なども今後検討していきたい. また, 現在の実装ではポアソン画像合成にてカラー Bleeding を対処していない. 境界条件指定, 合成領域抽出やソース画像に対する幾何変換などを含むユーザーインターフェースの整備が実用に向けて必要である.

6. まとめ

シームレスクロニングと呼ばれる画像合成問題において, もしも合成する画像間でテクスチャ細部のパターンが異なる場合に, 既存のアプローチでは色合は自然でも, テクスチャ細部の不連続性により, 自然な合成結果を生成しない事が知られている. この問題に対処するため, 本稿ではテクスチャを考慮した新しい画像合成フレームワークを提案した. 提案フレームワークは画像のテクスチャ細部とベースとなる低周波カラー情報を別々に処理するアイデアに基づいている. 提案フレームワークでは, まず画像の細部とベースをエッジ保存平滑化フィルタにより分離する. 次に, ベースの色合はポアソン方程式を解く事により補間し, 細部は画像類推法と呼ばれる例題に基づくテクスチャ合成法を用いて復元する. 本稿の学術的貢献は L^2 距離とガウス関数を用いた新たな定義域変換法を含む. 既存のアプローチでは, 自然な合成結果を生成しない対象について数値実験を行い, 提案フレームワークの有用性を確認した. 本稿で取り扱った画像合成問題は CG 分野にて様々な応用があり, 提案フレームワークの発展が期待される. 実用に向けた高速化やインターフェースの整備, 及び最新の方法との詳細な数値比較が今後の課題である.

謝辞

本研究は科学研究費補助金 (24700182) の助成を一部受けて実施された. 入力で用いた Watercolor, Rhone, Squire, 及び Newreflect3 は NYU Media Research Lab の配布画像を使用した.

参考文献

- [1] Agarwala, A.: Efficient Gradient-domain Compositing Using Quadrees, *ACM Trans. Graph.*, Vol. 26, No. 3, pp. 94:1–94:6 (2007).
- [2] Burt, P. and Adelson, E.: The Laplacian Pyramid as a Compact Image Code, *IEEE Trans. Commun.*, Vol. 31, pp. 532–540 (1983).
- [3] Chu, H.-K., Hsu, W.-H., Mitra, N., Cohen-Or, D., Wong, T.-T. and Lee, T.-Y.: Camouflage Images, *ACM Trans. Graph.*, Vol. 29, pp. 51:1–51:8 (2010).
- [4] Dale, K., Sunkavalli, K., Johnson, M., Vlastic, D., Matusik, W. and Pfister, H.: Video Face Replacement, *ACM Trans. Graph.*, Vol. 30, No. 6, pp. 130:1–130:10 (2011).
- [5] Darabi, S., Shechtman, E., Barnes, C., Goldman, D. and Sen, P.: Image Melding: Combining Inconsistent Images Using

- Patch-based Synthesis, *ACM Trans. Graph.*, Vol. 31, No. 4, pp. 82:1–82:10 (2012).
- [6] Dougherty, E.: *Digital Image Processing Methods*, CRC Press (1994).
- [7] Du, H., Jin, X. and Mao, X.: Digital Camouflage Images Using Two-scale Decomposition, *Comp. Graph. Forum*, Vol. 31, No. 7, pp. 2203–2212 (2012).
- [8] Eisemann, M., Gohlke, D. and Magnor, M.: Edge-constrained Image Compositing, *Proc. of Graphics Interface*, Canadian Human-Computer Communications Society, pp. 191–198 (2011).
- [9] Farbman, Z., Fattal, R. and Lischinski, D.: Convolution Pyramids, *ACM Trans. Graph.*, Vol. 30, No. 6, pp. 175:1–175:8 (2011).
- [10] Farbman, Z., Hoffer, G., Lipman, Y., Cohen-Or, D. and Lischinski, D.: Coordinates for Instant Image Cloning, *ACM Trans. Graph.*, Vol. 28, No. 3, pp. 67:1–67:9 (2009).
- [11] Gastal, E. and Oliveira, M.: Domain transform for edge-aware image and video processing, *ACM Trans. Graph.*, Vol. 30, pp. 69:1–69:12 (2011).
- [12] Gastal, E. and Oliveira, M.: Adaptive Manifolds for Real-Time High-Dimensional Filtering, *ACM Trans. Graph.*, Vol. 31, No. 4, pp. 33:1–33:13 (2012).
- [13] Greengard, L. and Rokhlin, V.: A fast algorithm for particle simulations, *J. Comput. Phys.*, Vol. 73, pp. 325–348 (1987).
- [14] Greengard, L. and Strain, J.: The Fast Gauss Transform, *SIAM J. Sci. Stat. Comput.*, Vol. 12, No. 1, pp. 79–94 (1991).
- [15] Greengard, L. and Sun, X.: A new version of the fast Gauss transform, *Proc. of IEEE Int. Cong. of Mathematicians III*, pp. 575–584 (1998).
- [16] Hertzmann, A., Jacobs, C., Oliver, N., Curless, B. and Salesin, D.: Image Analogies, *Proc. of SIGGRAPH*, ACM, pp. 327–340 (2001).
- [17] Huang, H., Yin, K., Gong, M., Lischinski, D., Cohen-Or, D., Ascher, U. and Chen, B.: "Mind the Gap": Tele-registration for Structure-driven Image Completion, *ACM Trans. Graph.*, Vol. 32, No. 6, pp. 174:1–174:10 (2013).
- [18] Ihler, A. and Mandel, M.: Kernel Density Estimation Toolbox for MATLAB (2003).
- [19] Jeschke, S., Cline, D. and Wonka, P.: A GPU Laplacian Solver for Diffusion Curves and Poisson Image Editing, *ACM Trans. Graph.*, Vol. 28, No. 5, pp. 116:1–116:8 (2009).
- [20] Jia, J., Sun, J., Tang, C.-K. and Shum, H.-Y.: Drag-and-drop Pasting, *ACM Trans. Graph.*, Vol. 25, No. 3, pp. 631–637 (2006).
- [21] Kwok, T.-H., Sheung, H. and Wang, C.: Fast Query for Exemplar-based Image Completion, *IEEE Trans. Img. Proc.*, Vol. 19, No. 12, pp. 3106–3115 (2010).
- [22] Mount, D. M. and Arya, S.: ANN: A Library for Approximate Nearest Neighbor Searching, www.cs.umd.edu/~mount/ANN (2010).
- [23] Pérez, P., Gangnet, M. and Blake, A.: Poisson Image Editing, *ACM Trans. Graph.*, Vol. 22, pp. 313–318 (2003).
- [24] Sorkine, O., Cohen-Or, D., Lipman, Y., Alexa, M., Rössl, C. and Seidel, H.-P.: Laplacian Surface Editing, *Proc. of EG Symp. on Geometry Processing*, pp. 179–188 (2004).
- [25] Summa, B., Scorzelli, G., Jiang, M., Bremer, P.-T. and Pascucci, V.: Interactive Editing of Massive Imagery Made Simple: Turning Atlanta into Atlantis, *ACM Trans. Graph.*, Vol. 30, No. 2, pp. 7:1–7:13 (2011).
- [26] Sunkavalli, K., Johnson, M., Matusik, W. and Pfister, H.: Multi-scale image harmonization, *ACM Trans. Graph.*, Vol. 29, pp. 125:1–125:10 (2010).
- [27] Tao, M., Johnson, M. and Paris, S.: Error-Tolerant Image Compositing, *Int. J. Comput. Vision*, Vol. 103, No. 2, pp. 178–189 (2013).
- [28] Wan, X. and Karniadakis, G. E.: A sharp error estimate for the fast Gauss transform, *J. Comput. Phys.*, Vol. 219, No. 1, pp. 7–12 (2006).
- [29] Wang, J. and Cohen, M.: Image and Video Matting: A Survey, *Found. Trends. Comput. Graph. Vis.*, Vol. 3, No. 2, pp. 97–175 (2007).
- [30] Xue, S., Agarwala, A., Dorsey, J. and Rushmeier, H.: Understanding and Improving the Realism of Image Composites, *ACM Trans. Graph.*, Vol. 31, No. 4, pp. 84:1–84:10 (2012).
- [31] Yang, F., Wang, J., Shechtman, E., Bourdev, L. and Metaxas, D.: Expression Flow for 3D-aware Face Component Transfer, *ACM Trans. Graph.*, Vol. 30, No. 4, pp. 60:1–60:10 (2011).
- [32] Yoshizawa, S., Belyaev, A. and Yokota, H.: Fast Gauss Bilateral Filtering, *Computer Graphics Forum*, Vol. 29, No. 1, p. 2010 (60-74).
- [33] Yoshizawa, S. and Yokota, H.: Poisson Image Analogy: Texture-Aware Seamless Cloning, *EUROGRAPHICS'13 Poster Proc.*, EG Association, pp. 5–6 (2013).
- [34] Yu, Y., Zhou, K., Xu, D., Shi, X., Bao, H., Guo, B. and Shum, H.-Y.: Mesh Editing with Poisson-based Gradient Field Manipulation, *ACM Trans. Graph.*, Vol. 23, No. 3, pp. 644–651 (2004).



図 18 提案フレームワークによる合成結果.