Vol.2015-CG-158 No.14
2015/2/27

実写画像に基づく特定画風を反映したアニメ背景画像への自動変換

山口 周悟[1,a)]　古澤 知英[2]　福里 司[2]　森島 繁生[3,b)]

概要：手描きアニメーション制作において、背景画像の作成は多大な労力を必要とする．そこで本研究では，実写の風景画像をアニメ背景画像へ自動変換する手法を提案する．提案手法では，各アニメータの個性的な”色使い”，”輪郭”，”ブラシの塗り方”を再現するために，アニメ作品中の風景画像の特徴を任意の実写風景画像に転写する．各領域の色調，塗り方の違いを考慮することで，従来研究で不十分であった輪郭の保持と，各領域の特徴を考慮した転写の両立を可能とした．

1. はじめに

2D 手描きアニメーション作品は，3DCG アニメーションが主流になりつつある昨今においても，日本を中心に世界中で幅広く人気がある重要コンテンツである．日本国内において放送されるアニメ作品数は年々増加しており，その需要は高まり続けていると言える．手描きアニメーション作品は，動きのあるキャラクタと静止した背景の 2 つから構成される．動きを伴うために比較的シンプルにデザインされるキャラクタに対して，背景は作品の世界観を表現するため非常に詳細に描きこまれる傾向にある．熊谷 [1] によれば，セル・アニメ―ションにおける背景画の要件と特徴の一部として以下が挙げられる．

- キャラクタとの視覚的整合性が重要であり，それが確保できれば背景画の描画技法は何れの方法でもよい．
- 近景と遠景とでは筆致は一様でなく，形状の輪郭は線ではなく，筆致によって表現される．
- 空間や物単位での単純化，意匠化が適宜行われる．

すなわち，作品ごとにアーティスト独自の色使いや塗りを統一して描く必要があり，一つの背景画内では領域によって異なる描き方がなされる．また背景は，作品の場面ごとに異なる画を描かなければならないが，同じ場面においてもカメラの方向や構図などによって異なる背景画を描く必要があるため，大量の枚数を描くことが要求される．大量の枚数に対して作品を通して画風を統一して表現する必要があるため，多くの技術と手間を要する．そのため，手描きアニメーション作品において背景の制作は大きな負担となっている．

手描きアニメーション作品の背景は，実在する場所でロケーションハンティングをおこなうことにより撮影した実写画像を元に，作品の画風を反映して手描きで描かれることがある．この実写画像に基づいたアニメ背景画像への変換を，自動でおこなうことが可能になれば，アニメ背景制作のコストを大きく削減させることができる．また，先述した通り，アニメ背景画像は作品を通して画風を統一させる必要がある．そこで本稿では，既存のアニメ背景画像を入力画像として用意し，入力アニメ背景画像の画風を実写画像に転写することによって，新たなアニメ背景画像を自動生成することを研究目的とする．

実写画像から手描き風の画像を生成する手法の多くは，油彩画風，水彩画風，版画風，カートゥーン風といったような特定の芸術形式の表現を目指したものであり，各ジャンルの中での作家性の表現に着目したものは少ない．アーティストの画風を転写する手法も存在するが，入力画像のテクスチャが保存されないために，画風や生成結果の品質が損なわれたものになっている．

そこで本研究では，以下の 3 工程から構成される手法を提案する．また，フローチャートを図 1 に示す．

(1) 入力画像 (実写，アニメ) の領域分割を行う．
(2) 実写画像の各領域と対応するアニメ背景画像の領域を探索する．
(3) 実写画像の各領域に，対応するアニメ背景画像の領域のテクスチャを転写する．

[1] 早稲田大学
Waseda University, Shinjuku, Tokyo 169-8555, Japan
[2] 早稲田大学/JST
Waseda University/JST
[3] 早稲田大学理工学術院総合研究所/JST
Waseda Research Institute for Science and Engineering/JST
[a)] wasedayshugo@suou.waseda.jp
[b)] shigeo@waseda.jp

© 2015 Information Processing Society of Japan

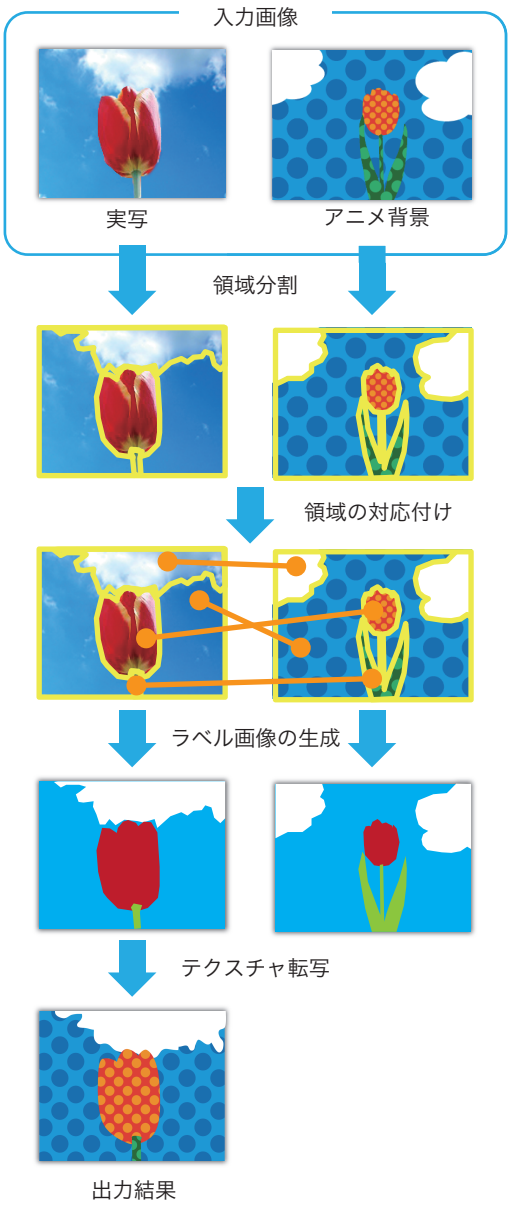


図 1 提案手法のフローチャート

領域の分割と対応付けを行うことにより，テクスチャ転写後も入力アニメ背景画像のテクスチャが保存された新規アニメ背景画像を自動生成することが可能となる．(3) の工程に関連して，指定領域から別の指定領域へとテクスチャを流し込む Texture-by-Numbers[2] と呼ばれる手法が存在するが，Texture-by-Numbers は領域指定が手入力であり，手間を要するという問題があった．提案手法においては，(1), (2) の工程により領域指定を自動でおこなうことを可能とし，従来手法にかかる手間を大幅に削減する．

2. 関連研究

本研究において目的とするのは，実写画像に基づく手描きアニメ背景画像の自動生成だが，実写画像からアニメ背景画像以外の手描き風画像を生成する研究には様々なアプローチがある．Haeberli ら [3] は，入力実写画像の輝度勾配を取得し，あらかじめ用意したブラシテクスチャ画像を起動勾配に垂直に合成することによって，手描き風の画像を生成した．ブラシテクスチャ画像の種類を変更したり，ブラシサイズを輝度勾配の大きさと対応させたりすることによって，様々なパターンの生成結果を得ることが可能である．しかしこの手法には，出力結果に実写画像らしさが残ってしまうという問題点が存在する．また，出力結果の画風は用意したブラシテクスチャ画像に依存することになるため，既存の画像の画風を反映させることは困難である．Zeng ら [4] は，Tu ら [5][6] の領域分割に基づく画像分析技術を応用し，領域分割した入力実写画像の各領域ごとに，あらかじめ用意したブラシテクスチャのデータベースから最適なブラシテクスチャを選択し，Haeberli らと同様に輝度勾配に基づいてブラシテクスチャを合成する手法を提案した．この手法では Haeberli らの手法と比較して，より手描きらしさを表現することを可能とするが，生成結果の画風はブラシテクスチャのデータベースに依存するという Haeberli らと同様の問題がある．Winnemoller ら [7] は，実写画像に対し，輪郭を保存した平滑化フィルタであるバイラテラルフィルタをかけた後，ポスタリゼーションや DoG フィルタによる輪郭線の付加を行うことにより，カートゥーン風の生成結果を得る手法を提案した．この手法では，バイラテラルフィルタの分散値や適用回数等といったパラメータを調整することにより様々な表現の生成結果を得ることができるが，特定の画風に対応するパラメータを取得することが困難であるため，Haeberli らの手法と同様に既存の画像の画風を反映させることは困難である．また，Haeberli らと同様に実写らしさを残した出力結果となっている．

画風の反映をおこなうためには，手描き画像を入力画像として用意し，その画風を実写画像に転写することが求められる．そのような研究の一つとして，Hertzmann ら [2] は教師学習ペア (実写画像と手描き画像) の間に見られるテクスチャ特徴の対応関係を，任意の実写画像 B に転写することにより，教師手描き画像の画風と類似した新たな手描き風画像を得る手法を提案した．この手法には，入力する手描き画像と対応の付いた実写画像を用意しなければならないという制約がある．解決策として，手描き画像に平滑化フィルタを適用させた画像を教師実写画像の代わりに用いるという手法があるが，ぼかした手描き画像と実写画像が適切に対応づけられているとは言い難い．また，この手法はピクセル単位で合成をおこなっているため高周波成分のノイズが発生しやすく，生成結果は全体的に輪郭が不明瞭なものになっている．油彩画風や水彩画風の一部の絵画ならば問題ないが，アニメ背景画像は比較的輪郭が鮮明であるという傾向があるため，教師学習ペアにアニメ背景画像を用意した場合に対応関係を抽出することが困難であり，

© 2015 Information Processing Society of Japan

生成結果もアニメ背景の画風を反映したものではなくなっている．さらに，この手法も Haeberli らや Winnemoller らの手法と同様に，実写らしさを残した出力結果となっている．Efros ら [8] は 2 枚の異なる入力画像をパッチ単位で輝度値により対応付けることにより，テクスチャ画像や手描き画像を転写する手法を提案した．この手法は，一様なテクスチャやモノクロの手描き画像を転写することは可能である．しかし，アニメ背景画像のような複数の異なるテクスチャや色調を持ったカラー画像に適用した場合，異なる領域のテクスチャが混在し，パッチの対応先が不連続であるために，入力画像のテクスチャが大きく損なわれた生成結果となってしまうという問題がある．また，この手法においても実写らしさの残った出力結果となっている．

これらの関連研究に共通する問題点として，画風転写において手描きテクスチャ情報が損なわれてしまうことと，実写らしさが残ってしまうことの 2 点が挙げられた．テクスチャ情報が損なわれる原因としては，対応パッチの探索は画像全体に対して行われるため，対応先は不連続になりやすく，合成結果において異なるテクスチャが混在しやすいからであると考えられる．すなわち，パッチ探索を画像全体に対しておこなっていることに起因すると考えられる．また，実写らしさが残る原因としては，変換が画像全体に対して一様であり，それゆえ元画像の変化も一様であるために，元画像の実写らしさが残った変換となってしまっているからであると考えられる．そこで，Zeng らの手法と同様に，領域分割をおこなうことを考える．一様なテクスチャを持った領域単位で分割し，それらの領域ごとにテクスチャ転写をおこなうことが可能になれば，テクスチャ合成の混在の問題を解決することができるものと考えられる．さらに，領域によって異なるテクスチャ転写をおこなうため，変換は画像全体に一様ではなくなり，実写らしさが軽減されるものと考えられる．領域ごとのテクスチャ転写の関連研究として，Hertzmann らは，指定した領域から領域へテクスチャ転写をおこなう Texture-By-Numbers と呼ばれる手法を提案した．この手法では，ある程度テクスチャが保存された転写をユーザの任意の形状の領域に対しておこなうことが可能であるが，ユーザの手入力 (領域の塗りつぶし) が不可欠であるため，手間のかかる作業を要求される．Busto ら [9] は高速に Texture-By-Numbers をおこなう手法を提案し，Lukáč ら [10] は境界を保存した Texture-By-Numbers の手法を提案したが，いずれの手法においても同様に手動による領域の指定が必要となる．本稿においては，分割した領域の対応付けを自動でおこなうことにより，全自動でテクスチャの転写がなされた生成結果を得ることを可能とする．

3. 提案手法

3.1 領域分割

入力された実写画像とアニメ背景画像それぞれに対して領域分割を行う．テクスチャ転写の工程において，領域から領域へとテクスチャの流し込みをおこなうため，分割される領域はひとまとまりのテクスチャを持ったものである必要がある．また，ひとつひとつの領域の面積がそれぞれ小さくなりすぎないようになることが望ましい．そのため，例えば色空間における K-means クラスタリングを用いた領域分割の手法は，ひとつのテクスチャ内の異なる色の模様を分割してしまう可能性があることや，ノイズが発生しやすいため，不適当である．そこで，Graph-Cut[11][12] ベースの領域分割をおこなうことを考える．テクスチャの変化する部分を切断するため，

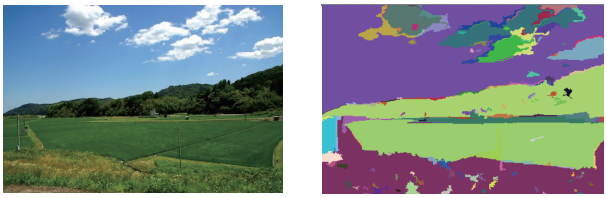
Felzenszwalb ら [13] は，変動の小さな領域の細部を維持し，高速な Graph ベースの領域分割手法を提案した．本稿においては，Felzenszwalb らの手法を用いて領域分割をおこなう．また，ラベリング処理を施すことにより，各領域に番号を振る．すなわち，画像の全画素に対して属する領域の番号が与えられる．

3.2 領域の対応付け

分割された実写画像とアニメ背景画像の各領域の対応付けをおこなう．その方法として，実写実写画像とアニメ背景画像の各領域の平均値をとり，知覚的にほぼ均等な歩度をもつ色空間である L*a*b*色空間におけるユークリッド距離が最小となる実写領域とアニメ領域を対応づける，という手法が考えられる．しかしこの手法には問題がある，例えば，全体的に明度の低い実写画像の各領域の対応先が全て，全体的に明度の高いアニメ背景画像の最も暗い領域に集中してしまう，というような現象が発生する．つまり，実写とアニメ背景の画像全体の色調が異なるために領域の対応付けが困難となる．

そこで，対応付けを行う前に，2 枚の入力画像の色味を全体的にフィッティングさせることを考える．Reinhard ら [14][15] は，2 枚の画像のヒストグラムをフィッティングさせることにより，色味を転写する手法を提案した．入力画像の L*a*b*値を L^{src}, a^{src}, b^{src}, 標準偏差を σ_L^{src}, σ_a^{src}, σ_b^{src}, 参照画像の L*a*b*値を L_{ref}, a_{ref}, b_{ref}, 標準偏差を σ_L^{ref}, σ_a^{ref}, σ_b^{ref}, 変換後の L*a*b*値を $L^{dst}, a^{dst}, b^{dst}$ とおくと，変換は以下の式に従う．

© 2015 Information Processing Society of Japan



(a) 入力画像　(b) 領域分割結果

図 2　領域分割結果



(a) 入力実写画像　(b) 入力アニメ背景画像　(c) 提案手法による色変換結果　(d) Reinhard et al. []

図 3　色変換

$$L^{dst} = \frac{\sigma_L^{ref}}{\sigma_L^{src}}(L^{src} - \langle L^{src}\rangle) + \langle L^{ref}\rangle \quad (1)$$

$$a^{dst} = \frac{\sigma_a^{ref}}{\sigma_a^{src}}(a^{src} - \langle a^{src}\rangle) + \langle a^{ref}\rangle \quad (2)$$

$$b^{dst} = \frac{\sigma_b^{ref}}{\sigma_b^{src}}(b^{src} - \langle b^{src}\rangle) + \langle b^{ref}\rangle \quad (3)$$

しかし，Reinhard らの手法は，入力画像内に複数の異なる色合いの領域が存在する場合に自然な転写をすることができないという問題がある (図 3(d))．図 3 の場合，空の青系統の色調と，田んぼや草むらなどの緑系統の色調がお互いに影響を及ぼし合うために理想的な色味転写をすることができていない．そこで我々は，2 枚の画像の全領域を色相，彩度，明度に基づいて 9 色 (黒，白，灰，赤，橙，黄，緑，青，紫) のグループに分け，同色のグループに属する領域の画素群に対して Reinhard らの手法を適用する (図 3(c))．これにより，同系統の色調の領域において実写の色味がアニメ背景のものにフィッティングされる．色味のフィッティングをおこなった後，実写画像の各領域において L*a*b*色空間におけるユークリッド距離が最小であるアニメ領域を対応づける．図 4 に示すように，実写画像の全ての領域は，対応するアニメ背景画像の領域を持つが，アニメ背景画像の領域は対応する実写画像の領域を持たないものも存在し得る．また，実写領域の対応先が，同一のアニメ背景領域となることもあり得る．すなわち，実写領域とアニメ背景領域の対応関係は，基本的に全射ではなくかつ単射ではない．ただし，実写領域とアニメ背景領域の個数が等しいときに限り，全単射になる可能性がある．

3.3　テクスチャ転写

Texture-by-Numbers によるテクスチャ転写では，対応する領域を指定するラベル画像が必要となる．どこで，テクスチャ転写に先立って最適なラベル画像の自動生成をおこなう．ラベル画像の生成にあたっては，以下の項目を満たしていることが要求される．

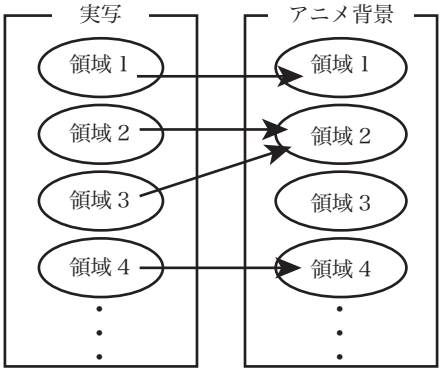


図 4　領域の対応関係

- 対応する領域が等しい範囲の色で塗りつぶされていること．
- 対応しない領域が異なる色で塗りつぶされており，色の範囲が重複しないこと．
- 実写領域はすべて塗りつぶされていること (アニメ領域は塗り残しがあってもよい)．

対応する領域が固有の色で塗りつぶされ，対応しない領域同士が出来るだけ色差の大きな色であれば，異なる領域のテクスチャが混在することなく転写をすることができる．さらに，当然ではあるが，実写領域のラベル画像は隙間なく塗りつぶされていなければならない．アニメ領域は実写領域と対応の付いていないものも存在しているので，隙間が生じていても問題ない．また，図 4 の実写領域 2,3 のように，同一のアニメ背景領域と対応づけられている実写領域は，同じ領域であると見なし，等しいもの色を用いる．

生成したラベル画像をもとに，Texture-by-Numbers の手法を用いてテクスチャ転写をおこなう．テクスチャの低周波成分の構造を合成するため，入力画像をダウンサンプ

© 2015 Information Processing Society of Japan

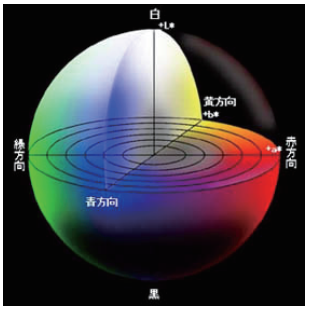
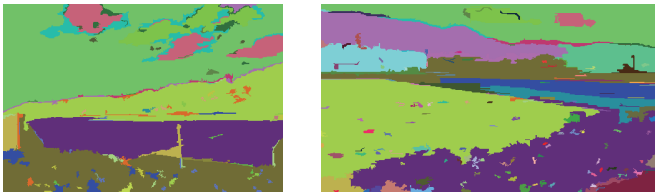


図 5 L*a*b 色空間 [16]



(a) (b)

図 6 ラベル画像

リングしたものを用いて合成したのちアップサンプリングする多重解像度合成をおこなう．

4. 結果

アニメ画像の画風を実写画像に転写した結果を図 7(d)(g), に示す．本手法の有効性を確かめるため，Hertzmann らの手法で転写した結果を図 7(e)(h) に，Efros らの手法で転写した結果を図 7(f)(i) に示す．ただし，Hertzmann らの手法においてはアニメ背景画像にバイラテラルフィルタを適用したものを教師画像セットに用いた．Hertzmann らの手法では，アニメ背景画像のテクスチャを転写することができず，Efros らの手法では，テクスチャ合成が混在していてアニメのテクスチャが保存されていないが，提案手法ではそれらの問題を解決している．従来手法と比較して，より入力アニメ背景の画風を反映した結果が得られたと言える．

また，異なる画風のアニメ背景画像を用意した場合，用意した画像によって異なる画風の結果が出力される．しかし，図 7(g) における山の部分のように，入力アニメ背景画像にない部分のテクスチャを転写することができないという制約が存在する．

5. まとめと今後の課題

本研究では，実写画像とアニメ背景画像の領域ごとの対応関係を基に，アニメ背景の画風 (テクスチャ) を自動で転写する手法を提案した．従来手法と比較して，より入力アニメ背景の画風を反映した生成結果を得ることができた．手入力を必要としないため，簡易的に生成することが可能となる．今後の課題としては，以下が挙げられる．領域の対応付けにおいて，本手法では色の平均値のみを考慮しているが，対応付けの精度向上のために，各領域の色の分散や，テクスチャスケール，テクスチャ方向などの情報も複合した分析・評価が必要となる．また，Zeng ら [4] のように，意味的な領域の分析を踏まえた対応付けにも取り組みたい．テクスチャ転写においては，本手法では色とテクスチャをまとめて合成しているが，Bousseau ら [18] が提案する色とテクスチャと陰影の分離や，Fang ら [19] が提案する陰影を考慮したテクスチャ貼付け等を応用することによって，入力した実写画像とアニメ背景画像双方の特徴をより最適に反映した合成をおこなえるようにしたい．また，本手法では入力できるアニメ背景は一枚のみだが，複数のアニメ背景画像の入力に対応することにより，生成結果の質を向上させたい．動画への適用や，生成結果に対して簡易的に編集をおこなえるようにすることも挙げられるが，そのためには Barns ら [17] の PatchMatch 等を用いたテクスチャ転写の高速化が不可欠である．建物のような複雑な形状なオブジェクトへの転写も課題である．

参考文献

[1] 熊谷武洋：デジタル技術を応用したセル・アニメーション用背景画作成手法について，研究論叢. 芸術・体育・教育・心理， Vol. 53, No. 3, pp. 253–265 (2003).
[2] Hertzmann, A., Jacobs, C. E., Oliver, N., Curless, B. and Salesin, D. H.: Image analogies, *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM, pp. 327–340 (2001).
[3] Haeberli, P.: Paint by numbers: Abstract image representations, *ACM SIGGRAPH Computer Graphics*, Vol. 24, No. 4, ACM, pp. 207–214 (1990).
[4] Zeng, K., Zhao, M., Xiong, C. and Zhu, S.-C.: From image parsing to painterly rendering, *ACM Trans. Graph*, Vol. 29, No. 1, p. 2 (2009).
[5] Tu, Z., Chen, X., Yuille, A. L. and Zhu, S.-C.: Image parsing: Unifying segmentation, detection, and recognition, *International Journal of Computer Vision*, Vol. 63, No. 2, pp. 113–140 (2005).
[6] Tu, Z. and Zhu, S.-C.: Parsing images into regions, curves, and curve groups, *International Journal of Computer Vision*, Vol. 69, No. 2, pp. 223–249 (2006).
[7] Winnemöller, H., Olsen, S. C. and Gooch, B.: Real-time video abstraction, *ACM Transactions On Graphics (TOG)*, Vol. 25, No. 3, pp. 1221–1226 (2006).
[8] Efros, A. A. and Freeman, W. T.: Image quilting for texture synthesis and transfer, *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM, pp. 341–346 (2001).
[9] Busto, P. P., Eisenacher, C., Lefebvre, S., Stamminger, M. et al.: Instant Texture Synthesis by Numbers., *VMV*, pp. 81–85 (2010).
[10] Lukáč, M., Fišer, J., Bazin, J.-C., Jamriška, O., Sorkine-Hornung, A. and Sỳkora, D.: Painting by feature: texture boundaries for example-based image creation, *ACM Transactions on Graphics (TOG)*, Vol. 32, No. 4, p. 116 (2013).
[11] Boykov, Y., Veksler, O. and Zabih, R.: Fast approximate energy minimization via graph cuts, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 23, No. 11, pp. 1222–1239 (2001).

© 2015 Information Processing Society of Japan



(a) 入力実写画像　(b) 入力アニメ背景画像　(c) 入力アニメ背景画像

(d) 提案手法　(e) Hertzmann et al. [2]　(f) Efros et al. [8]

(g) 提案手法　(h) Hertzmann et al. [2]　(i) Efros et al. [8]

図 7　提案手法の生成結果と既存手法との比較

[12] Kwatra, V., Schödl, A., Essa, I., Turk, G. and Bobick, A.: Graphcut textures: image and video synthesis using graph cuts, *ACM Transactions on Graphics (ToG)*, Vol. 22, No. 3, ACM, pp. 277–286 (2003).
[13] Felzenszwalb, P. F. and Huttenlocher, D. P.: Efficient graph-based image segmentation, *International Journal of Computer Vision*, Vol. 59, No. 2, pp. 167–181 (2004).
[14] Reinhard, E., Ashikhmin, M., Gooch, B. and Shirley, P.: Color transfer between images, *IEEE Computer graphics and applications*, Vol. 21, No. 5, pp. 34–41 (2001).
[15] Reinhard, E. and Pouli, T.: Colour spaces for colour transfer, *Computational Color Imaging*, Springer, pp. 1–15 (2011).
[16] 日本色研事業株式会社：日本色研ホームページ，日本色研事業株式会社（オンライン），入手先 ⟨http://www.sikiken.co.jp/colors/colors11.html⟩（参照 2015-1-31）.
[17] Barnes, C., Shechtman, E., Finkelstein, A. and Goldman, D.: PatchMatch: A randomized correspondence algorithm for structural image editing, *ACM Transactions on Graphics-TOG*, Vol. 28, No. 3, p. 24 (2009).
[18] Bousseau, A., Paris, S. and Durand, F.: User-assisted intrinsic images, *ACM Transactions on Graphics (TOG)*, Vol. 28, No. 5, ACM, p. 130 (2009).
[19] Fang, H. and Hart, J. C.: Textureshop: texture synthesis as a photograph editing tool, *ACM Transactions on Graphics (TOG)*, Vol. 23, No. 3, ACM, pp. 354–359 (2004).

© 2015 Information Processing Society of Japan