



応
般

テクスチャ合成技術の 新たな応用と展開

高山 健志
ETH Zurich / JSPS

五十嵐 健夫
東京大学 / JST ERATO

テクスチャ合成とは

テクスチャ合成とは、あるパターンや材質（例：岩肌、木目など）を表す画像をサンプルとして受け取り、それと視覚的に類似した自由な形・大きさの新しい画像を自動的に合成する技術である。テクスチャ合成の典型的な用途は2つあり、1つは3次元CGにおいて物体表面上の模様を表現するために、物体表面をテクスチャで覆う場合である。テクスチャを作成するには、基本的には実物の撮影、または手作業によるペイントのいずれかの方法をとるが^{☆1}、床や壁などの広い面積に貼り付けるための大きなテクスチャをこれらの方法で作成するのは困難であることが多い。一方で小さなテクスチャをタイル状に並べて貼り付けると、規則的な繰り返しのため見映えが非常に不自然になってしまう。そこでテクスチャ合成によって、小さなテクスチャをサン

プルとして必要なサイズのテクスチャを合成することで、この問題を解決できる（図-1）。

2つ目の用途は、画像（または動画）に対して高度な編集を行う場合である。たとえば画像中の物体を削除する場合、それに伴ってできる画像中の穴を何らかの方法で埋め、編集結果の見映えが不自然にならないようにする必要がある。そこで画像中の他の領域をテクスチャのサンプルとして、それに類似した画像を穴の中に合成することで、自然な編集結果を得ることができる（図-2）。

テクスチャ合成の基礎的な技術は1990年代後半から2000年代前半にかけて急速に進展し、現在では上記のような初期の目標はかなりの部分まで達成されたと言える状況にある。近年はこれらの基礎技術を応用してより高度な問題を解いたり、上記とはまったく異なる用途に応用したりすることに分野全

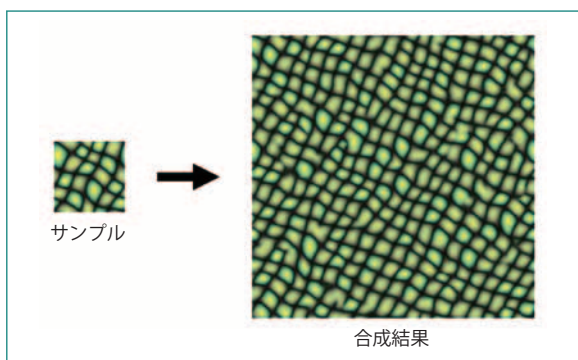


図-1 小さなテクスチャサンプルから、大きなサイズのテクスチャを合成している様子



出典) Criminisi, et al.: Object removal by exemplar-based inpainting, CVPR 2003.
© 2003 IEEE Inc. Included here by permission.

図-2 写真中の物体を削除し、それに伴う画像の穴をテクスチャ合成で埋めている様子

^{☆1} シェーダプログラムとして手続き的に定義する方法もあるが、望みのテクスチャを得るためにはプログラミングの試行錯誤が必要であり、ここでは詳しく触れない。

体の主眼が移ってきている。本稿では、テクスチャ合成技術の最近の動向を俯瞰することを目的として、最初にテクスチャ合成の基礎技術についてその基本的考え方を手短かに紹介した後で、より高度な画像合成への応用、ソリッドテクスチャへの拡張、画像合成以外への応用、および新たな方式による画像合成、という4つのトピックについて、比較的最近提案さ

れた手法を中心に紹介する。

テクスチャ合成の基礎技術

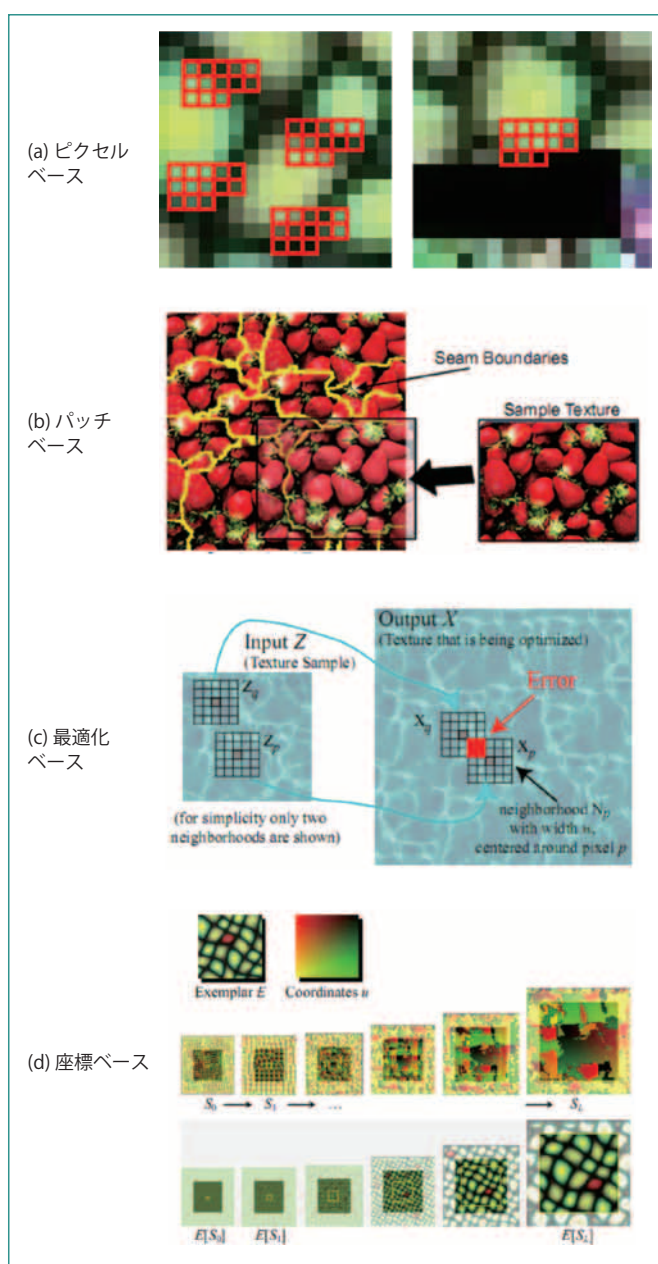
テクスチャ合成の基礎技術として主要なものを時系列順に挙げると、最初にピクセルベースの手法が考案され、次にパッチベースの手法が、その後最適化ベースの手法と座標ベースの手法が同時に考案された。なお後半の2手法が、テクスチャ合成を応用した近年の研究においては最もよく使われる。

Wei と Levoy によるピクセルベースの手法 (図-3 (a)) では、出力画像のピクセルの色を1つずつ左から右、上から下の順番で計算していく。現在のピクセルの色を求めるには、まずその左隣と上隣のピクセルの色を調べ、それと最も類似した近傍を持つピクセルを入力画像の中から探し、その色を現在のピクセルに割り当てる。

Kwatra らによるパッチベースの手法 (図-3 (b)) では、入力画像を切り抜いたパッチを繋ぎ合わせることで、出力画像を合成する。隣接するパッチ間の継ぎ目なるべく目立たなくなるように、パッチの位置と継ぎ目をうまく計算することが技術的ポイントとなる。

同じく Kwatra らによる最適化ベースの手法 (図-3 (c)) では、まず出力画像の各ピクセルをランダムに初期化した後、以下の検索と割り当てという2つのフェーズを交互に繰り返す。検索フェーズでは、出力画像の各ピクセルに対し、その近傍と最も類似した近傍を持つ入力画像のピクセルを検索する。検索フェーズの後では、出力画像の各ピクセルには入力画像のピクセルの近傍が多数重ね合わされている。次の割り当てフェーズでは、出力画像の各ピクセルに対し、これらの多数重ね合わされた入力画像のピクセルの色の平均を割り当てる。これらの処理は、入力画像と出力画像の類似度を表すエネルギーを最小化するものであり、必ず収束することが知られている。

Lefebvre と Hoppe による座標ベースの手法 (図-3 (d)) は、GPU (Graphics Processing Unit) の処理能力を活かしてリアルタイムにテクスチャを



出典) (a) Wei and Levoy, Fast texture synthesis using tree-structured vector quantization, SIGGRAPH 2000. (b) Kwatra et al., Graphcut textures: image and video synthesis using graph cuts, SIGGRAPH 2003. (c) Kwatra et al., Texture optimization for example-based synthesis, SIGGRAPH 2005. (d) Lefebvre and Hoppe, Parallel controllable texture synthesis, SIGGRAPH 2005. © 2000, 2003, 2005 ACM, Inc. Included here by permission.

図-3 テクスチャ合成の基礎技術

合成することを念頭に開発されたもので、最適化ベースの手法と同様、出力画像のピクセル全体に対し並列的な処理を反復するが、出力画像の各ピクセルに色ではなく入力画像の1つのピクセルの座標を割り当てる点が大きく異なる。前処理として入力画像中のパターンの現れ方をあらかじめ分析しておき、それを合成時に利用することで高速なテクスチャ合成を実現している。

以下に、これらの基礎技術を応用した最近の研究を紹介する。

より高度な画像合成への応用

古典的なテクスチャ合成では、サンプルのテクスチャ画像はその見映えが全体としては均一であることを仮定していたが、ここではそういった単純な問題設定の枠を出たより高度な画像合成にテクスチャ合成を応用した例を2つ紹介する。

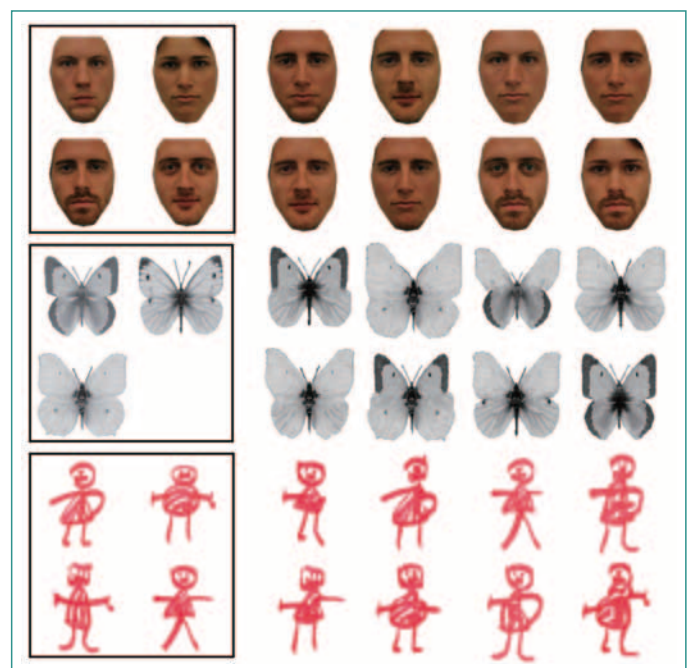
Risserらは、大まかに共通した構造を持つ画像のサンプルがいくつか入力として与えられたときに、どの入力画像とも微妙に異なるような画像のバリエーションを合成する手法を提案した¹⁾。この手法を使うことで、たとえば何枚かの実在する人物の顔写真を元にして実在しない人物の顔写真をたくさん合成したり、1つ1つがすべて異なるような昆虫や花などをキャンバス上にたくさん並べたり、1つ1つがオリジナルな手書きサインをたくさん合成したりすることができる(図-4)。技術的には座標ベースのテクスチャ合成手法を基本としているが、ランダム性を導入する部分で入力画像の構造を壊さないような工夫がなされている。

次に、複数スケールにわたる構造を含む画像を合成するHanらの手法²⁾を紹介する。自然物は多くの場合、複数のスケールで異なる構造を持つ。たとえば人間の皮膚は、普通のスケールで見ればシワやホクロなどが見えるが、拡大していくと産毛や毛穴が見え始め、さらに拡大すると細胞や血管などが見えてくる。別の例として衛星写真を考えると、最もズームアウトしたところからは大陸や島が見えるが、拡大していくと山脈や湖が見え始め、さらに拡大す

ると個々の尾根や川が見えてくる。こういった複数のスケールにわたる構造を画像で表す際に、単純に拡大率最大の画像を領域全体にわたって用意すると、データ量が膨大になってしまうという問題がある。そこで各スケールにおける局所的な構造の類似性に着目し、似たような部分はテクスチャ合成によって動的に生成してしまおうというのが、この手法の基本的な考え方である。システムは複数スケールのテクスチャサンプルの集合と、それらの間の倍率の関係を表すグラフを入力として受け取る。ユーザが閲覧したいズームレベルと表示範囲を指定すると、システムはそれに応じた画像をリアルタイムに合成して出力する(図-5)。技術的にはこれも座標ベースの手法に基づいており、GPU実装により高速に動作するため、高解像度な画像全体を一度に合成して記憶するのではなく、現在表示している範囲のみをリアルタイムに合成することができ、コンパクトな表現形式を実現している。

ソリッドテクスチャへの拡張

ソリッドテクスチャとは3次元ボリュームとして色情報を格納する表現形式であり、3次元CGにお

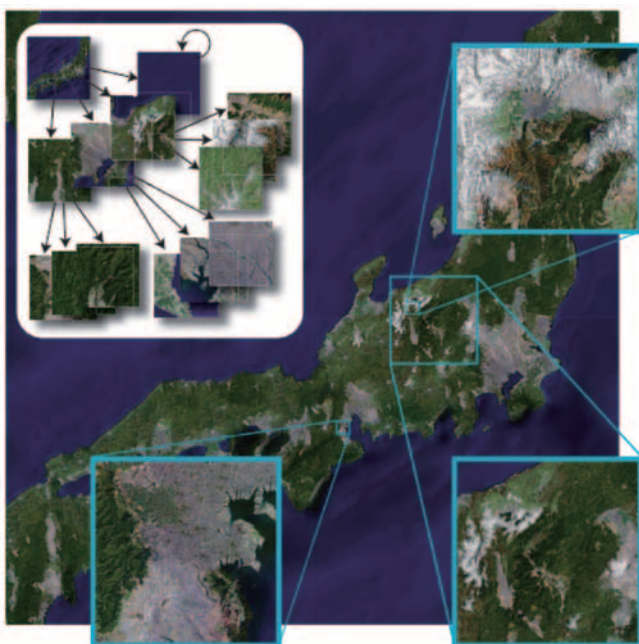


© 2010 ACM, Inc. Included here by permission.

図-4 構造を持つ画像を対象としたテクスチャ合成¹⁾

いてさまざまな点で有用である。たとえば物体表面にテクスチャで模様を加えたい場合に、通常の2次元テクスチャを使う場合は3次元サーフェスと2次元平面の間のマッピング（UVマッピングと呼ばれる）をうまく与える必要があるが、ソリッドテクスチャを使う場合そのようなマッピングは不要になる。またソリッドテクスチャは物体内部の情報も保持しているため、切って断面を表示したりボリュームレンダリングを行ったりすることが可能になる。一方でソリッドテクスチャは、2次元テクスチャと比べて作成が非常に難しいという問題がある。手作業でのペイントは当然ながら難しいが、CTやMRIなどの特殊な計測装置を使ったとしても、CGで必要となる見映えの情報までは計測できない。

Kopfらは、2次元テクスチャサンプルを入力として、それと類似した断面を持つようなソリッドテクスチャを合成できる手法を提案した³⁾。この手法を使えば、テクスチャとして利用したい実物の模様を写真に撮ってシステムに入力するだけで、UVマッピングを経ることなく3次元モデルに簡単にテクスチャを貼ることができる。また、断面の方向ごとに異なるテクスチャサンプルを与えることで異方



© 2008 ACM, Inc. Included here by permission.

図-5 複数スケールにわたる構造を含む画像の合成²⁾

性のあるソリッドテクスチャを合成したり、色以外にも反射率などの情報も同時に合成することで複雑な見映えを作り出したりすることもできる（図-6）。技術的には、最適化ベースのテクスチャ合成手法を3次元に拡張したものとなっている。

筆者らはこの手法で合成した、あるいは手動で作成したソリッドテクスチャを入力として、より複雑なソリッドテクスチャを合成する手法を提案した。基本的なアイデアは、一定の大ききで切り抜かれたソリッドテクスチャのパッチを、ユーザがスケッチUIで与えた3次元的な方向場に沿って、物体内部に繰り返し貼りつけていくというものである。同じソリッドテクスチャを使い回すため、中身の詰まった物体を非常にコンパクトに表現できる所が利点となっている。また層構造を持つソリッドテクスチャをうまく扱うことで、キウイ、ニンジン、樹木などさまざまな自然物を表現することができる（図-7）。

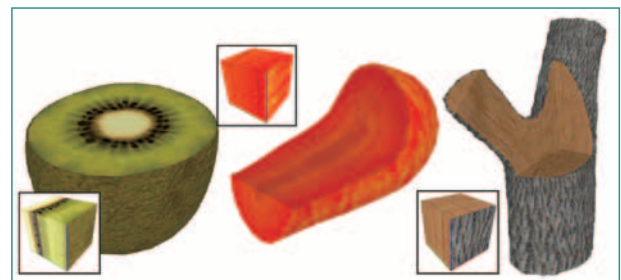
画像合成以外への応用

テクスチャ合成の技術は画像合成以外にも、形や動きなど一般的なデータをサンプルから自動合成す



© 2007 ACM, Inc. Included here by permission.

図-6 2次元テクスチャサンプルを入力とした、ソリッドテクスチャの合成³⁾



出典) Takayama et al., Lapped solid textures: filling a model with anisotropic textures, SIGGRAPH 2008.

© 2008 ACM, Inc. Included here by permission.

図-7 入力ソリッドテクスチャを方向場に沿って繰り返し貼り付けることで、より複雑なソリッドテクスチャを合成する手法

るのに応用できる可能性を持つ。ここではそういった画像合成以外の用途にテクスチャ合成を応用した例を3つ紹介する。

Wangらは頭髪形状の生成にテクスチャ合成を応用した⁴⁾。3次元CGにおいて頭髪の自然な表現はきわめて重要であり、さまざまなモデリング手法が開発されてきた。しかしモデリング対象ごとに毎回ゼロから髪型を作り直すのは手間がかかる。そこですでに作成済みの髪型データから、それと類似した髪型のバリエーションを自動合成することができれば、モデリングの効率を大幅に改善できる、というのがWangらの着眼点である。システムは頭髪形状のサンプルを入力として受け取り、それと類似する新しい頭髪形状を出力する(図-8)。基本的なアイデアは、髪の毛1本を1つの特徴ベクトルで表し、頭髪全体をそれらの特徴ベクトルが2次元的に分布した画像と見なすことで、テクスチャ合成によって新しい頭髪形状を合成するというものである。

次に、流体の速度場の生成にテクスチャ合成を応用したMaらの手法を紹介する⁵⁾。CG分野において流体のシミュレーション方法は多数開発されてきたが、細かい局所的な流体の振舞い(渦のでき方など)を直感的に指定することは、シミュレーションだけでは難しい。Maらはこういった細かい流体の動きを直感的に与える目的にテクスチャ合成を応用した。この手法を使うことで、たとえばハート型の渦を作るような流体を表現することができる

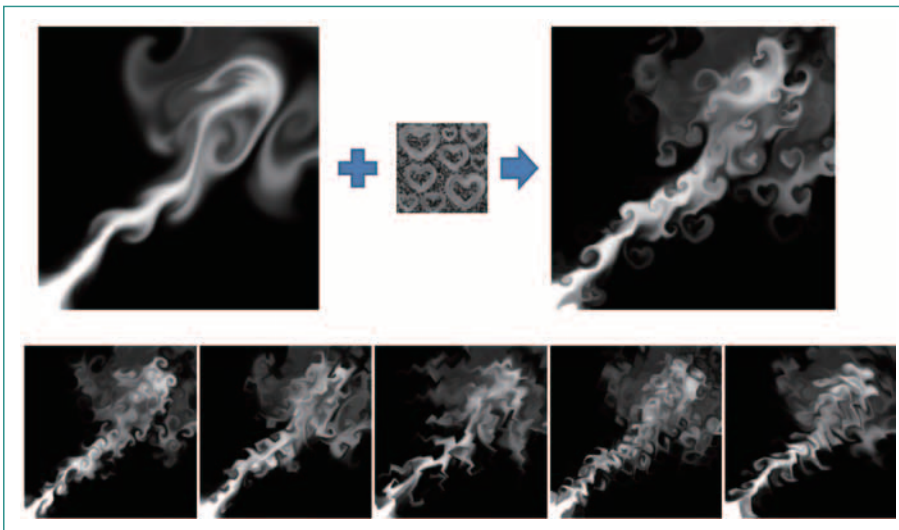
(図-9)。また2次元の速度場サンプルから3次元の速度場を合成することもできる。技術的には、速度ベクトルをピクセルの色と見なしてテクスチャ合成を適用する、という基本的な考え方に基づいている。

同じくMaらは、多数の物体が凝集してできる物体配置パターンの生成にテクスチャ合成を応用した⁶⁾。多数の物体が1カ所に集められて塊となる現象は、自然物と人工物の両方において頻繁に見られる。例としては積み上げられた果物の山、河原の小石、皿に盛られたスパゲッティなど、身の周りでも多く見つけられる。こういった多数の物体の配置をCGでモデリングしたい場合、単に手作業で配置するのは手間がかかりすぎるし、一方で物理シミュレーションを使って自動的に生成しようとしても、思い通りの結果を得るためには何度もシミュレーションを走らせねばならず、やはり大変である。そこでテクスチャ合成を応用して、物体配置の小さなサンプルから大規模な物体配置を自動的に合成しよう、というのがMaらの着眼点である。システムは入力として、物理シミュレーションや手作業などで作った配置パターンのサンプルおよび物体を配置したい領域を受け取ると、サンプルと類似した配置パターンで領域内を満たす(図-10)。技術的には、最適化ベースのテクスチャ合成手法を基に、離散的な要素配置の合成に適用できるようにさまざまな修正を加えている。



© 2009 ACM, Inc. Included here by permission.

◀ 図-8 テクスチャ合成を応用した頭髪形状の合成⁴⁾ 左: サンプル, 右: 合成結果



© 2009 ACM, Inc. Included here by permission.

◀ 図-9 テクスチャ合成を応用した速度場の合成⁵⁾: (左) 入力の大局的速度場 (中央) 入力の局所的速度場 (右) 合成結果

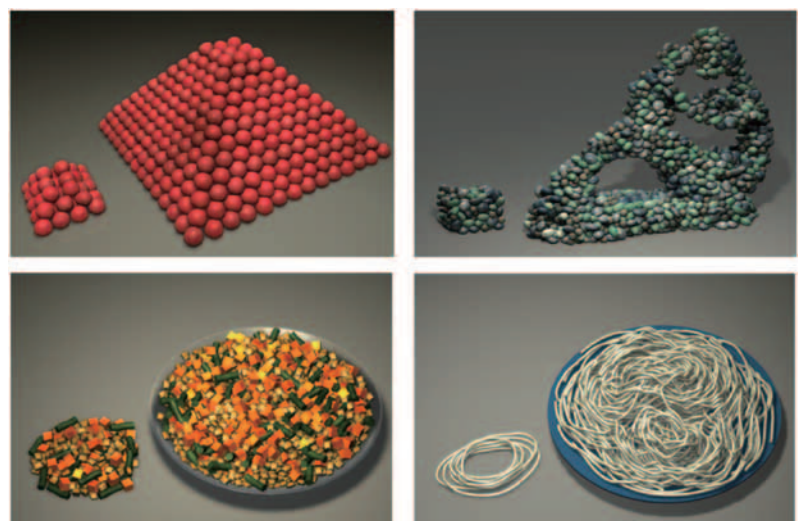
新たな方式による画像合成

最後に、一見すると従来法と同様な画像合成のように見えるが、実はまったく新しい方式に基づいており固有の利点を持つような手法について、最近提案されたものを2つ紹介する。

最初に紹介したテクスチャ合成の基礎技術では、合成画像の各ピクセルについて、その局所近傍と最も類似するテクスチャサンプル中の局所近傍を検索する部分が、処理全体の中でボトルネックとなっていた。これを高速化する試みは従来からあったが、速度が不十分であったり前計算が必要であったりしたため、高解像度画像のインタラクティブな編集にテクスチャ合成技術を利用することは難しかった。Barnesらはこれを劇的に高速化する画期的なアルゴリズム PatchMatch を提案した⁷⁾。詳細な説明は割愛するが、提案アルゴリズムは約 20 ~ 100 倍の高速化を達成している。このアルゴリズムをベースに、画像中の物体を削除してその穴を埋めたり、内容をなるべく歪めずに画像の縦横比を変えたり、画像中の物体を好きな位置に動かしたり、といった高度な画像編集のためのさまざまなツールが提案されている (図-11)。

なお、この手法に基づく画像編集機能は Adobe Photoshop CS5 に搭載され、すでに広く利用されている。

3次元CGで多数の建物からなるシーンを構築する場合、さまざまな縦横幅の壁面に適切なテクスチャを貼り付ける必要がある。既存のテクスチャ合成手法や画像リサイズ手法を用いれば、このようなテクスチャを1枚のサンプルから自動合成することは可能だが、壁面ごとに合成したテクスチャを画像として記憶すると、建物の数が増えた場合にメモリ消費が膨大になってしまうという問題がある。Lefebvreらは合成したテクスチャを、テクス



© 2011 ACM, Inc. Included here by permission.

図-10 テクスチャ合成を応用した物体配置パターンの合成⁶⁾。左の小さな配置パターンをサンプルとして、右の大きな配置パターンを合成している



© 2009 ACM, Inc. Included here by permission.

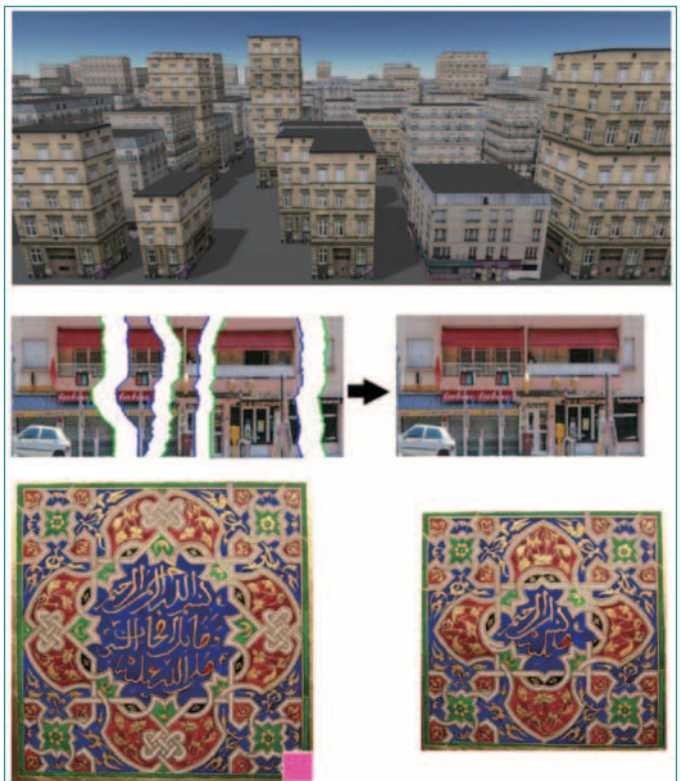
図-11 PatchMatchに基づくインタラクティブな画像編集⁷⁾

チャサンプル1枚と数キロバイト程度の付加的な情報だけで表現し、画像として記憶することを回避できる手法を提案した⁸⁾。基本的なアイディアは、建物は多くの場合縦と横の方向に規則的な繰り返し構造（窓やドアなど）を持つので、繰り返す部分を切り取ってパズルのように繋ぎ合わせることで、任意の縦横幅のテクスチャを簡単に合成できる、というものである。この手法を使うことでさまざまな建物の画像や、建物でなくても縦横に繰り返しのある画像に対して、簡単にサイズの異なる画像を合成することができる（図-12）。

本稿で紹介したように、現在ではテクスチャ合成の基礎技術自体はある程度成熟してしまったが、それを新たな問題設定に応用したり実用化を妨げるボトルネックを解消したりするような革新的な技術が近年も活発に研究されており、テクスチャ合成はこれからも注目すべき分野であると言える。

参考文献

- 1) Risser, E., Han, C., Dahyot, R. and Grinspun, E. :



© 2010 ACM, Inc. Included here by permission.

図-12 建物壁面のパターンに特化したテクスチャ合成⁸⁾

Synthesizing Structured Image Hybrids, ACM SIGGRAPH (2010).

- 2) Han, C., Risser, E., Ramamoorthi, R. and Grinspun, E. : Multiscale Texture Synthesis, ACM SIGGRAPH (2008).
- 3) Kopf, J., Fu, C.-W., Cohen-Or, D., Deussen, O., Lischinski, D. and Wong, T.-T. : Solid Texture Synthesis from 2D Exemplars, ACM SIGGRAPH (2007).
- 4) Wang, L., Yu, Y., Zhou, K. and Guo, B. : Example-Based Hair Geometry Synthesis, ACM SIGGRAPH (2009).
- 5) Ma, C., Wei, L.-Y., Guo, B. and Zhou, K. : Motion Field Texture Synthesis, ACM SIGGRAPH Asia (2009).
- 6) Ma, C., Wei, L.-Y. and Tong, X. : Discrete Element Textures, ACM SIGGRAPH (2011).
- 7) Barnes, C., Shechtman, E., Finkelstein, A. and Goldman, D. B. : PatchMatch : A Randomized Correspondence Algorithm for Structural Image Editing, ACM SIGGRAPH (2009).
- 8) Lefebvre, S., Hornus, S. and Kasram, A. : By-example Synthesis of Architectural Textures, ACM SIGGRAPH (2010).

(2012年2月20日受付)

高山 健志 kenshi84@acm.org

2012年東京大学大学院情報理工学系研究科博士（情報理工学）。同年より日本学術振興会海外特別研究員としてETH Zurichに所属。専門はグラフィックスおよびユーザインタフェース。

五十嵐 健夫（正会員）takeo@acm.org

2000年東京大学大学院工学系研究科博士（工学）。2002年同大学院情報理工学系研究科講師、2005年助教授。2007年よりJST ERATO研究総括。学術振興会賞、SIGGRAPH若手科学者賞等受賞。専門はユーザインタフェースおよびグラフィックス。