

イメージシンセシス

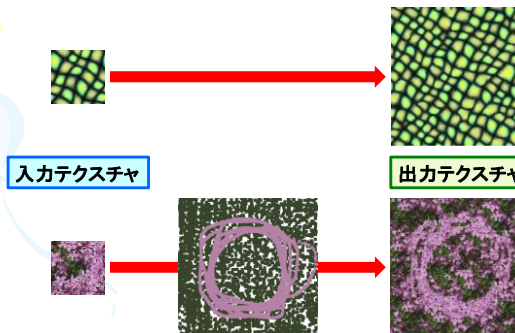
テクスチャ合成 (パッチベース手法)

岩手大学大学院
総合科学研究科 理工学専攻
デザイン・メディア工学コース

藤本 忠博

テクスチャ合成

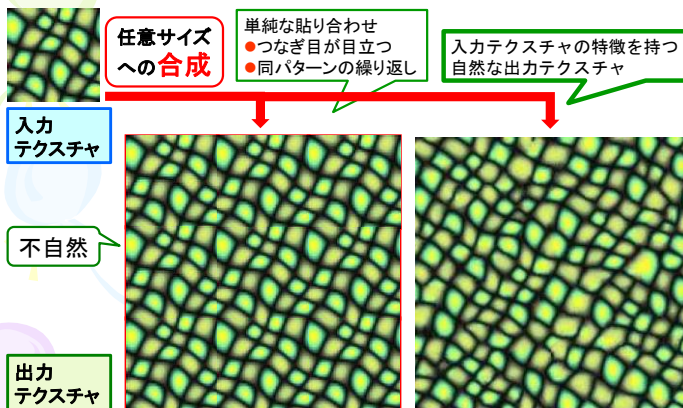
- 入力テクスチャからサイズの異なる自然な出力テクスチャを合成する。



L. Wei, M. Levoy, "Fast Texture Synthesis using Tree-structured Vector Quantization", 2000

M. Ashikhmin, "Synthesizing Natural Textures", 2001

テクスチャ合成

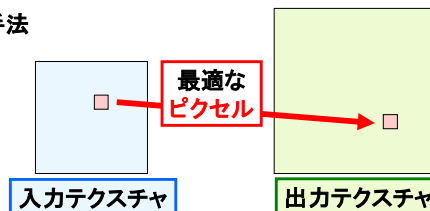


L. Wei, M. Levoy, "Fast Texture Synthesis using Tree-structured Vector Quantization", 2000

テクスチャ合成の分類

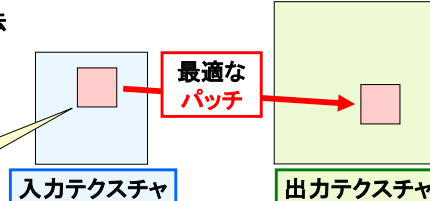
■ ピクセルベース手法

入力テクスチャから適切なピクセルを選択し、そのピクセルの色を出力テクスチャに割り当てる。



■ パッチベース手法

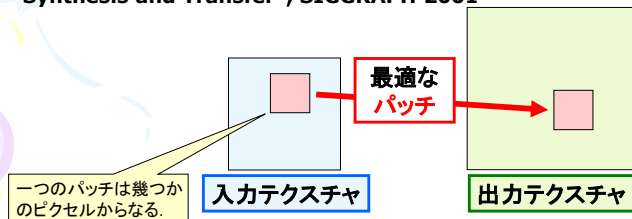
入力テクスチャから部分テクスチャ(パッチ)を切り出し、つながり目が目立たないように貼り合わせる。



一つのパッチは幾つかのピクセルからなる。

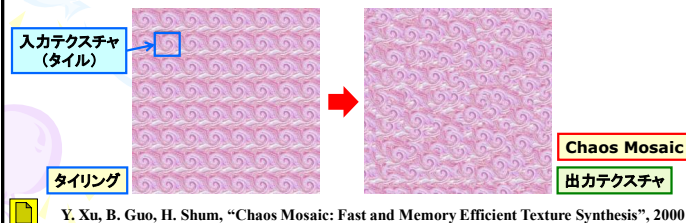
パッチベース手法

- 入力テクスチャから部分テクスチャ(パッチ)を切り出し、つなぎ目が目立たないように貼り合わせる。
- 初期の代表的な手法に Y. Xu らによる Chaos Mosaic (2000), A. Efros らによる Image Quilting (2001) がある。
- Y. Xu, B. Guo, H. Shum, "Chaos Mosaic: Fast and Memory Efficient Texture Synthesis", 2000
- A. Efros, W. Freeman, "Image Quilting for Texture Synthesis and Transfer", SIGGRAPH 2001



Chaos Mosaic

- Y. Xu ら (2000) による Chaos Mosaic
- 入力テクスチャの局所の特徴を一様で確率統計的に不規則な分布で持つような出力テクスチャを生成する。
- 入力テクスチャ(タイル)を規則的に並べて出力テクスチャの下地(タイリング)を作り、カオス挙動を生み出す画像を用いて、各タイルから切り出した複数のパッチ(ブロック)をタイリング上にランダムに配置する。
- Y. Xu, B. Guo, H. Shum, "Chaos Mosaic: Fast and Memory Efficient Texture Synthesis", 2000



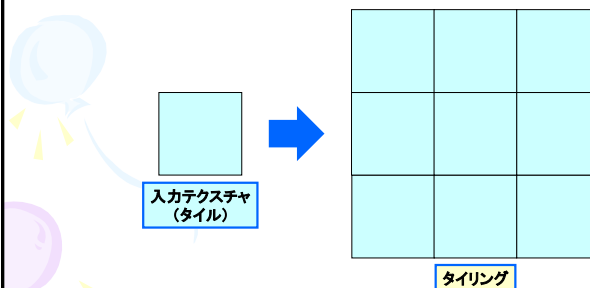
Chaos Mosaic : アルゴリズム

- Chaos Mosaic のアルゴリズム
- 1. 入力テクスチャ(タイル)を規則的に並べて出力テクスチャの下地(タイリング)を作る。
- 2. タイリング中の各タイルからパッチ(ブロック)を切り出す。通常、1つのタイルからランダムな位置の1つのパッチを切り出す。
- 3. カオス挙動を生み出す画像を繰り返し適用することで、各パッチの配置をタイリング上で反復的に移動させる。反復回数は任意。
 - フィルター処理
 - 制約付きテクスチャ合成

Y. Xu, B. Guo, H. Shum, "Chaos Mosaic: Fast and Memory Efficient Texture Synthesis", 2000

Chaos Mosaic : アルゴリズム

- Chaos Mosaic のアルゴリズム
- 1. 入力テクスチャ(タイル)を規則的に並べて出力テクスチャの下地(タイリング)を作る。

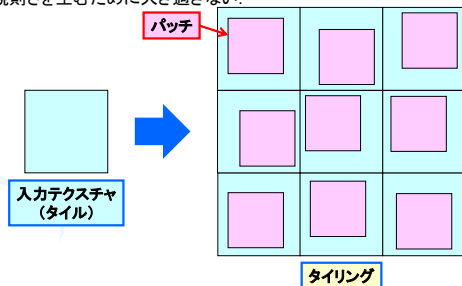


Y. Xu, B. Guo, H. Shum, "Chaos Mosaic: Fast and Memory Efficient Texture Synthesis", 2000

Chaos Mosaic : アルゴリズム

■ Chaos Mosaic のアルゴリズム

2. タイリング中の各タイルから**パッチ(ブロック)**を切り出す。通常、1つのタイルからランダムな位置の1つのパッチを切り出す。
→ 入力テクスチャの局所の特徴を保持する大きさと、かつ、ランダムな配置で不規則さを生むために大き過ぎない。



Y. Xu, B. Guo, H. Shum, "Chaos Mosaic: Fast and Memory Efficient Texture Synthesis", 2000

Chaos Mosaic : アルゴリズム

■ Chaos Mosaic のアルゴリズム

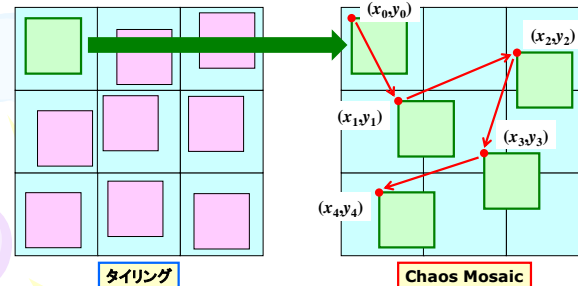
3. **カオス挙動を生み出す画像**を繰り返し適用することで、各パッチの配置をタイリング上で反復的に移動させる。反復回数は任意。

Cat map

$$x_{i+1} = (x_i + y_i) \bmod m$$

$$y_{i+1} = (x_i + 2y_i) \bmod m$$

タイリングの大きさ $m \times m$



Y. Xu, B. Guo, H. Shum, "Chaos Mosaic: Fast and Memory Efficient Texture Synthesis", 2000

Chaos Mosaic : アルゴリズム

■ Chaos Mosaic のアルゴリズム

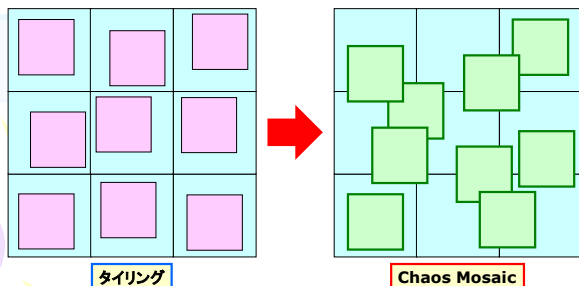
3. **カオス挙動を生み出す画像**を繰り返し適用することで、各パッチの配置をタイリング上で反復的に移動させる。反復回数は任意。

Cat map

$$x_{i+1} = (x_i + y_i) \bmod m$$

$$y_{i+1} = (x_i + 2y_i) \bmod m$$

タイリングの大きさ $m \times m$



Y. Xu, B. Guo, H. Shum, "Chaos Mosaic: Fast and Memory Efficient Texture Synthesis", 2000

Chaos Mosaic : アルゴリズム

■ Chaos Mosaic のアルゴリズム

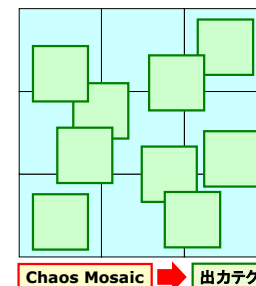
4. パッチの境界が目立たないように処理する。

■ フィルター処理

境界の数ピクセル幅のピクセル色をフィルターでスムージングする。
→ 計算時間小だが、エッジ等が多いパターンには不自然な結果

■ 制約付きテクスチャ合成

境界の数ピクセル幅を欠落領域としてピクセルベース手法でテクスチャ合成し、周囲のピクセル色と自然につながる色を割り当てる。
→ 自然な結果となるケースが増えるが、計算時間大

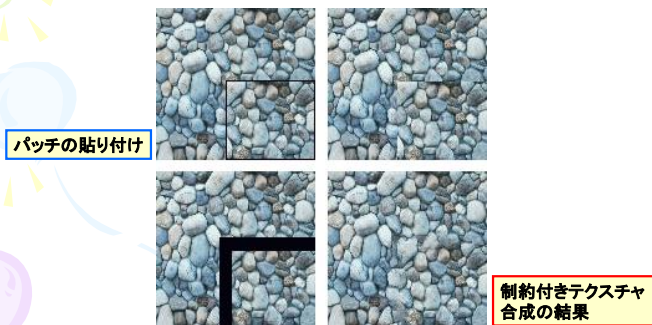


Y. Xu, B. Guo, H. Shum, "Chaos Mosaic: Fast and Memory Efficient Texture Synthesis", 2000

Chaos Mosaic : 実験結果

■ 実験結果

- パッチの境界は制約付きテクスチャ合成で処理

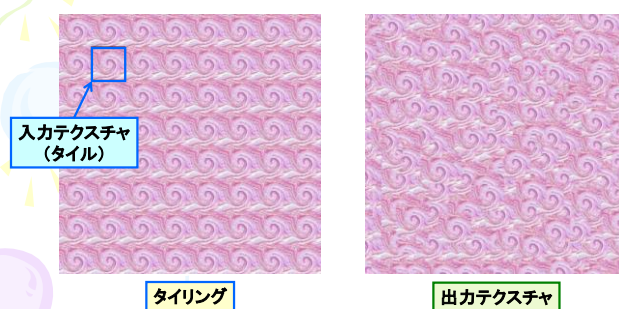


Y. Xu, B. Guo, H. Shum, "Chaos Mosaic: Fast and Memory Efficient Texture Synthesis", 2000

Chaos Mosaic : 実験結果

■ 実験結果

- パッチの境界はフィルター処理

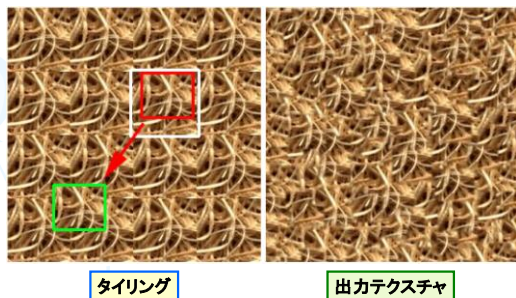


Y. Xu, B. Guo, H. Shum, "Chaos Mosaic: Fast and Memory Efficient Texture Synthesis", 2000

Chaos Mosaic : 実験結果

■ 実験結果

- パッチの境界はフィルター処理

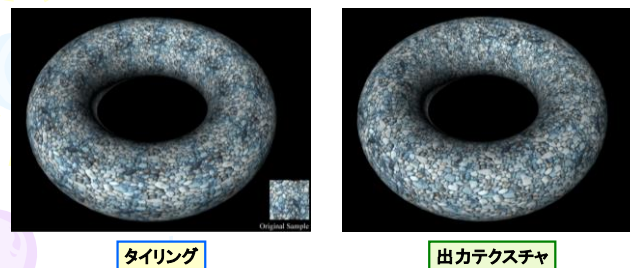


Y. Xu, B. Guo, H. Shum, "Chaos Mosaic: Fast and Memory Efficient Texture Synthesis", 2000

Chaos Mosaic : 実験結果

■ 実験結果

- パッチの境界はフィルター処理

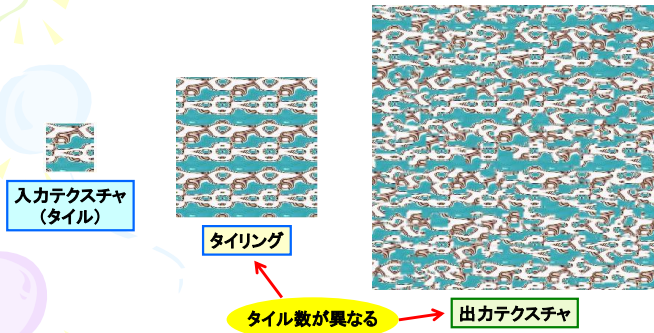


Y. Xu, B. Guo, H. Shum, "Chaos Mosaic: Fast and Memory Efficient Texture Synthesis", 2000

Chaos Mosaic : 実験結果

■ 実験結果

- パッチの境界はフィルター処理

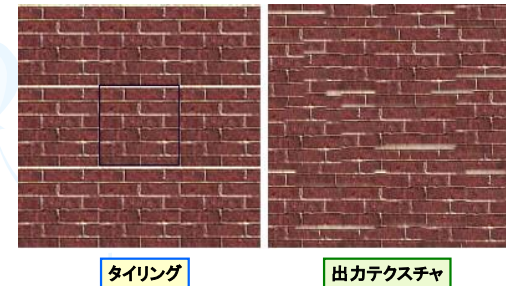


Y. Xu, B. Guo, H. Shum, "Chaos Mosaic: Fast and Memory Efficient Texture Synthesis", 2000

Chaos Mosaic : 実験結果

■ 実験結果

- パッチの境界はフィルター処理

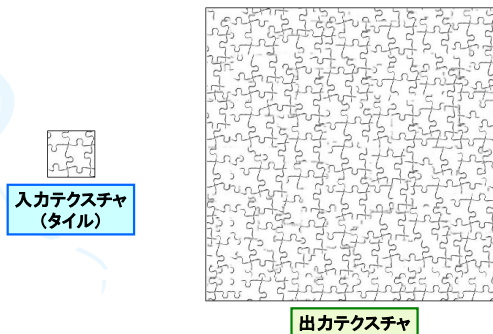


Y. Xu, B. Guo, H. Shum, "Chaos Mosaic: Fast and Memory Efficient Texture Synthesis", 2000

Chaos Mosaic : 実験結果

■ 実験結果

- パッチの境界はフィルター処理

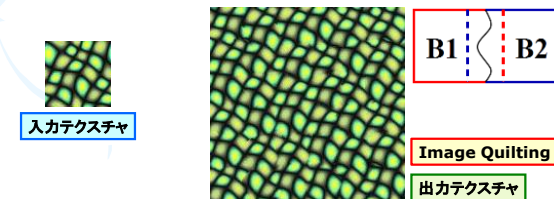


Y. Xu, B. Guo, H. Shum, "Chaos Mosaic: Fast and Memory Efficient Texture Synthesis", 2000

Image Quilting

■ A. Efros ら (2001) による Image Quilting

- 入力テクスチャから切り出した複数のパッチを境界部分が重なるように貼り合わせて出力テクスチャを生成する。
 - 重なり部分が類似するパッチを入力テクスチャ上で探索する。
 - 重なり部分でパッチの最適な境界を求め、つなぎ目を目立たなくする。
- A. Efros, W. Freeman, "Image Quilting for Texture Synthesis and Transfer", SIGGRAPH 2001

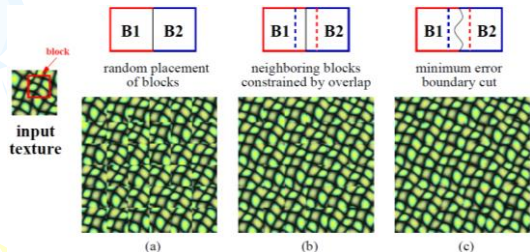


A. Efros, W. Freeman, "Image Quilting for Texture Synthesis and Transfer", 2001

Image Quilting

- 入力テクスチャから切り出した複数のパッチ(ブロック)の配置について、次の3つの方法を比較

- ランダムなパッチを配置する方法
- パッチ間の最適な重なりを考慮する方法
- パッチ間の最適な境界を考慮する方法 (Image Quilting)



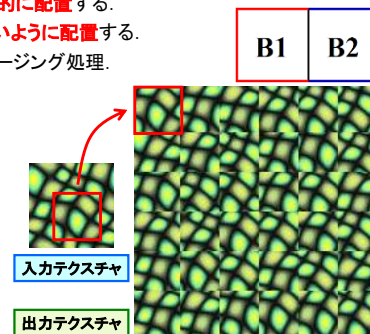
A. Efros, W. Freeman, "Image Quilting for Texture Synthesis and Transfer", 2001

Image Quilting

(a) ランダムなパッチを配置する方法

- 入力テクスチャの複数のランダムな位置からパッチを切り出し、出力テクスチャ上で縦横方向に規則的に配置する。
- パッチ間の境界が重ならないように配置する。
- パッチ間の境界部分でスムージング処理。

- つなぎ目が目立つ。

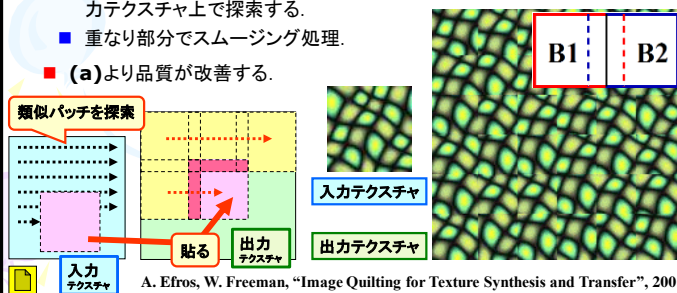


A. Efros, W. Freeman, "Image Quilting for Texture Synthesis and Transfer", 2001

Image Quilting

(b) パッチ間の最適な重なりを考慮する方法

- 入力テクスチャからパッチを切り出し、出力テクスチャ上で走査線順に配置して合成していく。一つのパッチPの合成は次のように行う。
- 出力テクスチャ上の合成済み領域と境界部分を重ねて配置する。
- 重なり部分で合成済み領域と色パターンが最も類似するパッチPを入力テクスチャ上で探索する。
- 重なり部分でスムージング処理。
- (a)より品質が改善する。

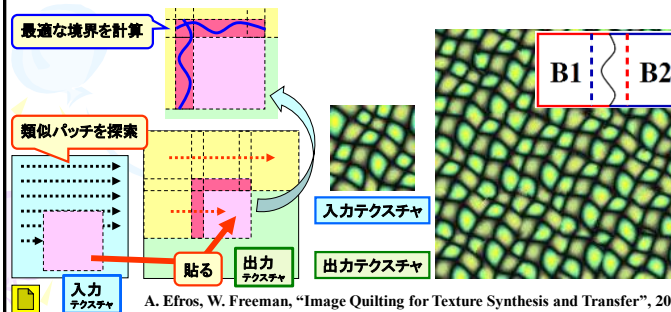


A. Efros, W. Freeman, "Image Quilting for Texture Synthesis and Transfer", 2001

Image Quilting

(c) パッチ間の最適な境界を考慮する方法 (Image Quilting)

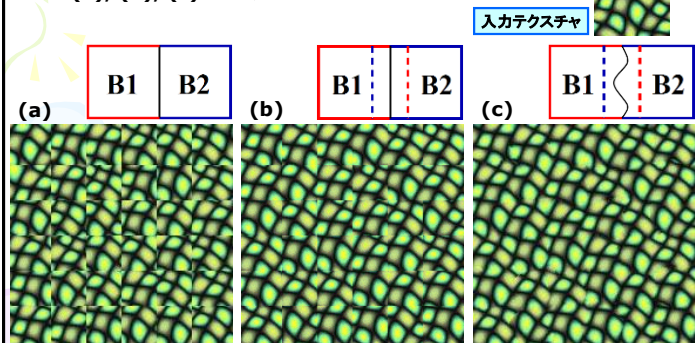
- 基本手順は(b)と同じ。重なり部分の工夫により(b)の品質を改善する。
- 重なり部分で、パッチPの境界形状を変え、境界に沿って合成済み領域とパッチPの色パターンが最も類似するように、最適な境界を求める。



A. Efros, W. Freeman, "Image Quilting for Texture Synthesis and Transfer", 2001

Image Quilting

- (a), (b), (c) の比較



A. Efros, W. Freeman, "Image Quilting for Texture Synthesis and Transfer", 2001

Image Quilting : アルゴリズム

Image Quilting のアルゴリズム

- 出力テクスチャ上で走査線順 (x, y 方向ごとにパッチの幅から重なり部分の幅を除いた間隔ごと) の位置でパッチを合成
- 入力テクスチャ上のパッチの探索: 出力テクスチャの合成済み領域と重なり部分の色パターンが類似する (類似度の評価値が閾値以下の) パッチを候補とし, その中からランダムに一つのパッチ P_{opt} を選択する.
- パッチの境界線の最適化: 重なり部分内でパッチ P_{opt} の左上の境界線の形状を変え, 境界線に沿って合成済み領域とパッチ P_{opt} の色パターンが最も類似するように, 最適な境界線を求める.

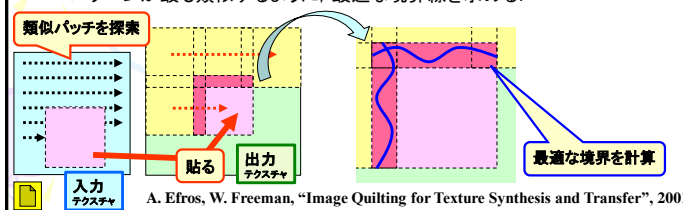


Image Quilting : アルゴリズム

- 入力テクスチャ上のパッチの探索
 - パッチの大きさは, 入力テクスチャの特徴をとらえるほどには大きく, また, 出力テクスチャに多様性を与えられるほどには小さくする.
 - 重なり部分の幅は, パッチの幅の $1/6$ とした.
 - 合成済み領域とパッチの類似度の評価値には, ピクセルごとの色の L_2 ノルム (RGBの要素ごとの差分の2乗の和) を用いた.
 - 候補パッチを選ぶ類似度評価値の閾値は, 最小評価値の 1.1 倍とした.

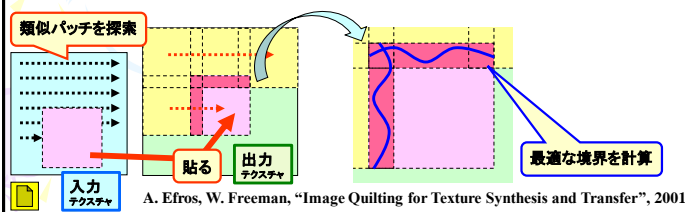


Image Quilting : アルゴリズム

- パッチの境界線の最適化
- 左側の鉛直方向 (j 方向) の境界線の最適化
 - 重なり部分内のピクセル (i, j) の類似度評価値を $e_{i,j}$ とし, 累積評価値 $E_{i,j}$ を鉛直方向に下向きに求めていく ($i=1, \dots, N_{ix}, j=1, \dots, N_{iy}$).
 - $E_{i,1} = e_{i,1}, E_{i,j} = e_{i,j} + \min(E_{i-1,j-1}, E_{i,j-1}, E_{i+1,j-1})$
 - 最下部で累積評価値 $E_{i,N_{iy}}$ が最小のピクセル P_{min} を選ぶ.
 - ピクセル P_{min} に最小の累積評価値を与えたピクセル群を上向きにたどり, 最適な境界線を得る.

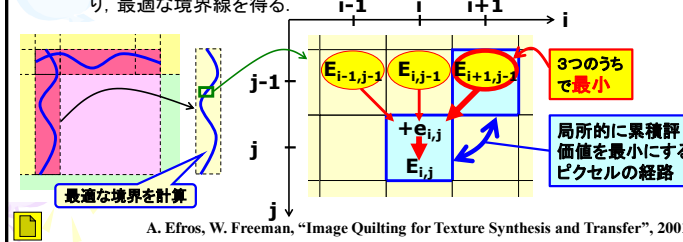


Image Quilting : アルゴリズム

- パッチの境界線の最適化
- 上側の水平方向(i方向)の境界線の最適化
 - 左側の鉛直方向(j方向)の境界線と同様の方法で最適化する。
- 鉛直方向と水平方向の最適な境界線が交わる点をパッチ P_{opt} を合成する左上の角とし、鉛直境界線から右側と水平境界線から下側を出力テクスチャに合成する。

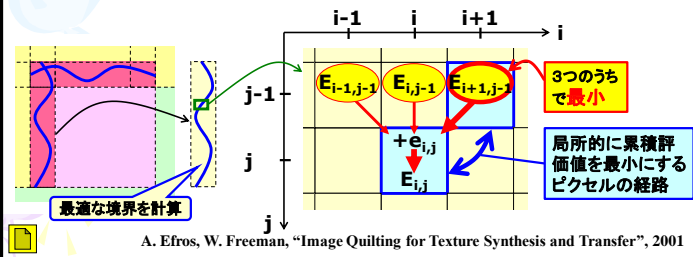


Image Quilting : アルゴリズム

- パッチの境界線の最適化
- 実装には動的計画法(dynamic programming)が利用できる。
- 最短経路問題を効率的に解くダイクストラのアルゴリズム(Dijkstra's algorithm)も利用できる。

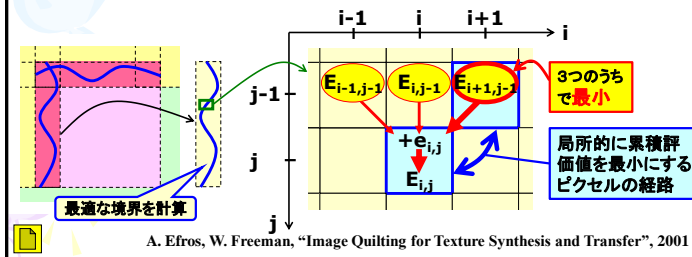


Image Quilting : 実験結果

■ 実験結果

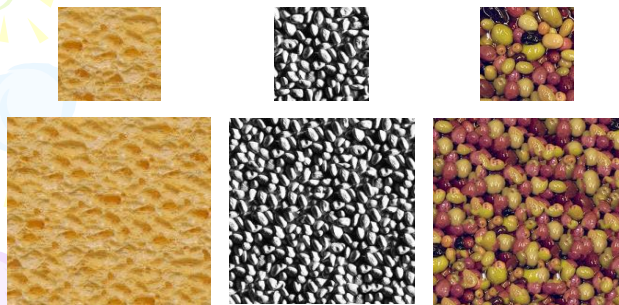


Image Quilting : 実験結果

■ 実験結果

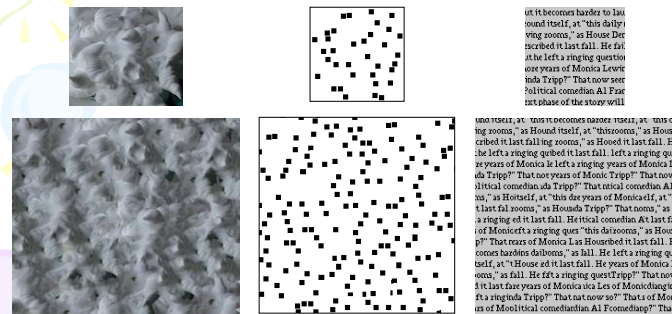
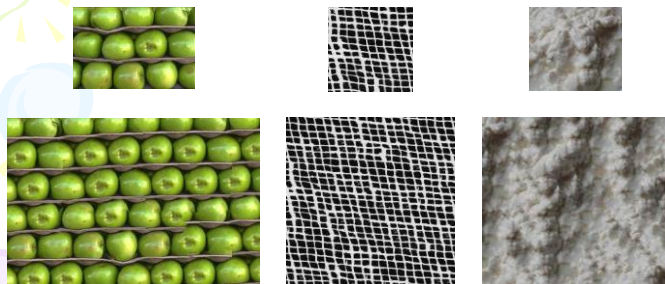


Image Quilting : 実験結果

■ 実験結果



A. Efros, W. Freeman, "Image Quilting for Texture Synthesis and Transfer", 2001

Image Quilting : 実験結果

■ 実験結果



A. Efros, W. Freeman, "Image Quilting for Texture Synthesis and Transfer", 2001

Image Quilting : 実験結果

■ 実験結果



A. Efros, W. Freeman, "Image Quilting for Texture Synthesis and Transfer", 2001

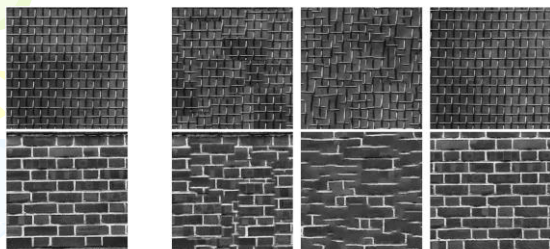
Image Quilting : 実験結果

■ 実験結果



A. Efros, W. Freeman, "Image Quilting for Texture Synthesis and Transfer", 2001

Image Quilting : 実験結果



入力テクスチャ

Chaos mosaic

Wei, Levy
ピクセルベース手法

Image Quilting

A. Efros, W. Freeman, "Image Quilting for Texture Synthesis and Transfer", 2001

Image Quilting : Texture Transfer

- **Image Quilting** の応用として、同じ論文中で **Texture Transfer** という手法が提案された。
- ソーステクスチャの特徴を与えるようにターゲット画像を出力画像に変換する。



ソーステクスチャ



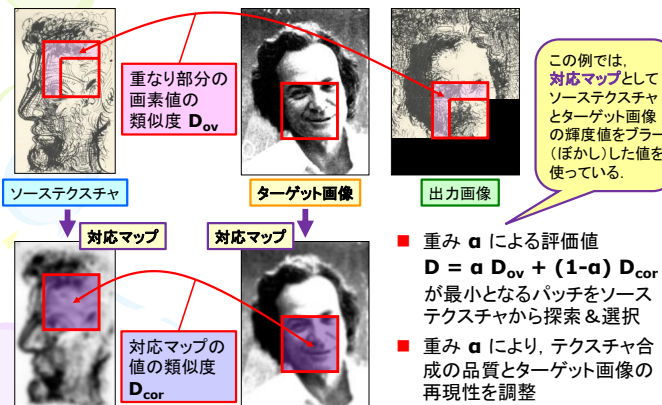
ターゲット画像



出力画像

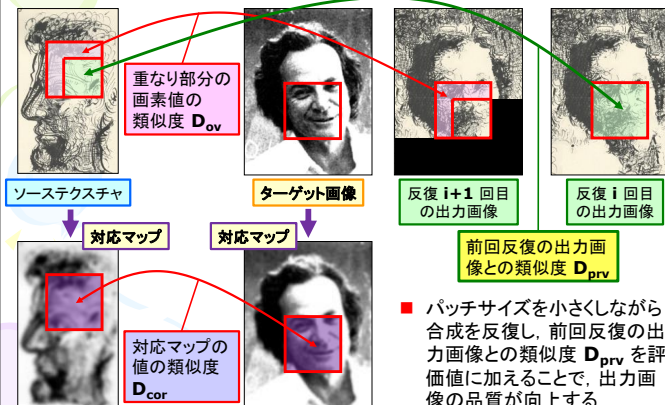
A. Efros, W. Freeman, "Image Quilting for Texture Synthesis and Transfer", 2001

Image Quilting : Texture Transfer



A. Efros, W. Freeman, "Image Quilting for Texture Synthesis and Transfer", 2001

Image Quilting : Texture Transfer



A. Efros, W. Freeman, "Image Quilting for Texture Synthesis and Transfer", 2001

Image Quilting : Texture Transfer

■ 実験結果

ソーステクスチャ

ターゲット画像

出力画像

A. Efros, W. Freeman, "Image Quilting for Texture Synthesis and Transfer", 2001

Image Quilting : Texture Transfer

■ 実験結果

ソーステクスチャ

Image Quilting による出力画像

ターゲット画像

Texture Transfer による出力画像

A. Efros, W. Freeman, "Image Quilting for Texture Synthesis and Transfer", 2001

Patch-Based Sampling

■ L. Liang ら (2001) による Patch-Based Sampling

- 基本的な仕組は Image Quilting と同様. 重なり部分が類似するパッチを入力テクスチャ上で探索し, 出力テクスチャ上で貼り合わせる.
- Image Quilting とは主に以下の点異なる.
 - 高速化によるリアルタイム処理の実現
 - 制約付きテクスチャ合成の実現
 - 重なり部分の feathering によるつなぎ目が目立つことの軽減
- 詳細な実装方法と様々な条件下での実験結果が報告されている.

■ L. Liang, C. Liu, Y. Xu, B. Guo, H. Shum, "Real-Time Texture Synthesis by Patch-Based Sampling", 2001

L. Liang, C. Liu, Y. Xu, B. Guo, H. Shum, "Real-Time Texture Synthesis by Patch-Based Sampling", 2001

Patch-Based Sampling

■ 高速化によるリアルタイム処理

- 出力テクスチャの合成済み領域と重なり部分の色パターンが類似する(類似度の評価値が閾値以下の)候補パッチを入力テクスチャ上で高速に探索することが重要.
- 入力テクスチャ上の全てのパッチから重なり部分が類似するパッチを正確に探索する代わりに, 近似的に高速な探索を行う.
 - approximate nearest neighbor (ANN) search

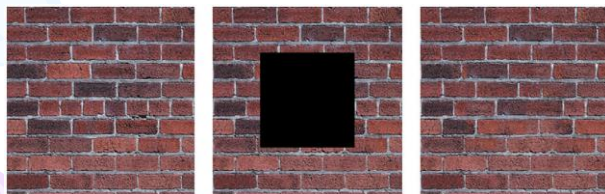
■ 次の3つにより高速化をはかる.

- optimized kd-tree
- quadtree pyramid
- 主成分分析 (principal component analysis : PCA)

L. Liang, C. Liu, Y. Xu, B. Guo, H. Shum, "Real-Time Texture Synthesis by Patch-Based Sampling", 2001

Patch-Based Sampling

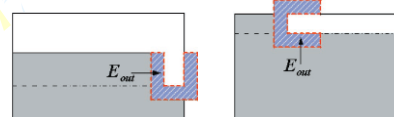
- 制約付きテクスチャ合成
 - 欠落領域の合成(hole filling)
 - 欠落領域の境界で自然な合成を行うため、出力テクスチャ上で**螺旋順序(spiral order)**でパッチを合成する。



L. Liang, C. Liu, Y. Xu, B. Guo, H. Shum, "Real-Time Texture Synthesis by Patch-Based Sampling", 2001

Patch-Based Sampling

- 制約付きテクスチャ合成
 - 出力テクスチャの環状化(tileable texture synthesis)
 - 出力テクスチャの左端と右端、上端と下端を連続化する。
 - 出力テクスチャ上の走査線順によるパッチの合成において、最終行、最終列の位置で、貼り付けるパッチの(出力テクスチャからはみ出た部分を環状化させて反対側の合成済み領域に重さね、重なり部分に含めることで類似度評価を行う。

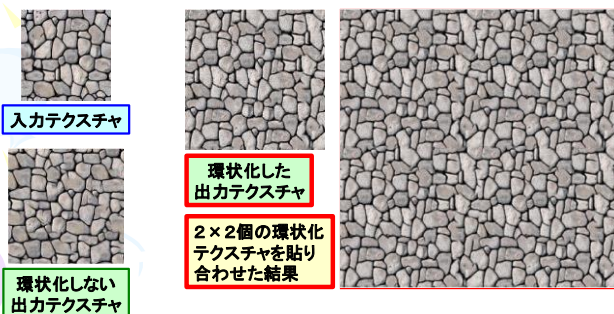


この論文では、走査線順を左→右, 下→上としている。

L. Liang, C. Liu, Y. Xu, B. Guo, H. Shum, "Real-Time Texture Synthesis by Patch-Based Sampling", 2001

Patch-Based Sampling

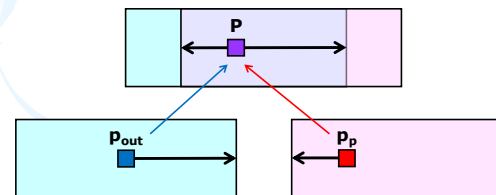
- 制約付きテクスチャ合成
 - 出力テクスチャの環状化(tileable texture synthesis)



L. Liang, C. Liu, Y. Xu, B. Guo, H. Shum, "Real-Time Texture Synthesis by Patch-Based Sampling", 2001

Patch-Based Sampling

- つなぎ目が目立つことの軽減
 - 重なり部分に**feathering**を適用する。
 - 出力テクスチャの重なり部分のピクセルPについて、合成済み領域のピクセル色 P_{out} と入力テクスチャから切り出したパッチのピクセル色 P_p を用いて、それらを重なり部分の境界までの距離に比例した重みによって重み付け平均化した色をピクセルPの合成色とする。

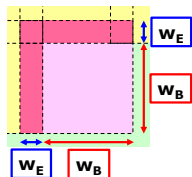


L. Liang, C. Liu, Y. Xu, B. Guo, H. Shum, "Real-Time Texture Synthesis by Patch-Based Sampling", 2001

Patch-Based Sampling : 実験結果

■ パッチサイズの変更

- パッチの幅 w_B が大きいほど良好な合成結果。
- 重なり部分の幅は $w_E = 4$ ピクセル。



入力テクスチャ
64 x 64



$w_B = 16$



$w_B = 24$



$w_B = 32$

出力テクスチャ

L. Liang, C. Liu, Y. Xu, B. Guo, H. Shum, "Real-Time Texture Synthesis by Patch-Based Sampling", 2001

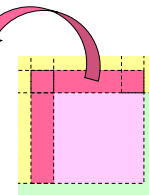
Patch-Based Sampling : 実験結果

■ 類似度評価

- 重なり部分での出力テクスチャの合成済み領域 E_{out} に対して、入力テクスチャから切り出したパッチの領域 E_p の類似度評価値 d が閾値 d_{max} 以下の場合、貼り付ける候補パッチとする。
- パラメータ ϵ によって閾値を調整する。

$$d(E_p, E_{out}) = \left[\frac{1}{A} \sum_{i=1}^A (p_p^i - p_{out}^i)^2 \right]^{\frac{1}{2}}$$

$$d_{max} = \epsilon \left[\frac{1}{A} \sum_{i=1}^A (p_{out}^i)^2 \right]^{\frac{1}{2}}$$

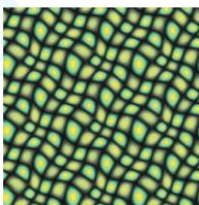


L. Liang, C. Liu, Y. Xu, B. Guo, H. Shum, "Real-Time Texture Synthesis by Patch-Based Sampling", 2001

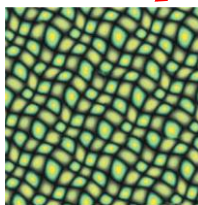
Patch-Based Sampling : 実験結果

■ 類似度評価

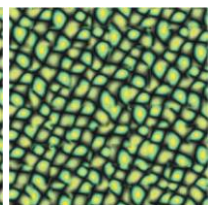
$$d(E_p, E_{out}) = \left[\frac{1}{A} \sum_{i=1}^A (p_p^i - p_{out}^i)^2 \right]^{\frac{1}{2}}, d_{max} = \epsilon \left[\frac{1}{A} \sum_{i=1}^A (p_{out}^i)^2 \right]^{\frac{1}{2}}$$



$\epsilon = 0$



$\epsilon = 0.2$



$\epsilon = 1.0$

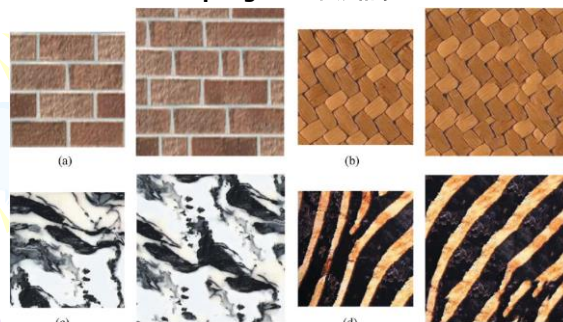
入力テクスチャ

出力テクスチャ

L. Liang, C. Liu, Y. Xu, B. Guo, H. Shum, "Real-Time Texture Synthesis by Patch-Based Sampling", 2001

Patch-Based Sampling : 実験結果

■ Patch-Based Sampling による合成結果



入力テクスチャ: 200 x 200

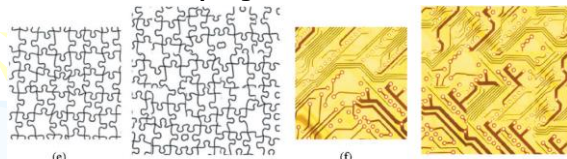
出力テクスチャ: 256 x 256

L. Liang, C. Liu, Y. Xu, B. Guo, H. Shum, "Real-Time Texture Synthesis by Patch-Based Sampling", 2001

合成時間: 約0.02秒
(667 MHz PC)

Patch-Based Sampling : 実験結果

■ Patch-Based Sampling による合成結果



入力テクスチャ: 200 x 200

出力テクスチャ: 256 x 256

L. Liang, C. Liu, Y. Xu, B. Guo, H. Shum, "Real-Time Texture Synthesis by Patch-Based Sampling", 2001

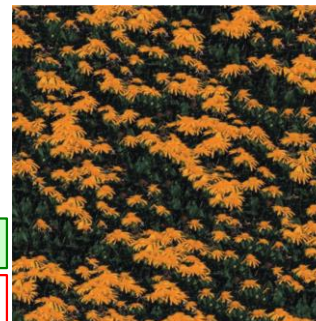
合成時間: 約0.02秒
(667 MHz PC)

Patch-Based Sampling : 実験結果

■ Patch-Based Sampling による合成結果



入力テクスチャ
200 x 200



出力テクスチャ
800 x 800

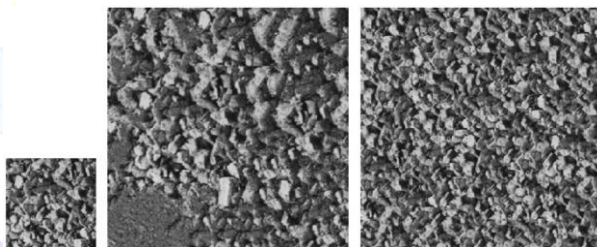
合成時間: 約0.65秒
(667 MHz PC)

L. Liang, C. Liu, Y. Xu, B. Guo, H. Shum, "Real-Time Texture Synthesis by Patch-Based Sampling", 2001

Patch-Based Sampling : 実験結果

■ 他手法との比較

- ピクセルベース手法 (Efros, Leung) では、入力テクスチャ上で選択されるピクセルが不適切な領域に留まってしまう可能性がある。



入力テクスチャ

Efros, Leung
ピクセルベース手法

Patch-Based
Sampling

L. Liang, C. Liu, Y. Xu, B. Guo, H. Shum, "Real-Time Texture Synthesis by Patch-Based Sampling", 2001

Patch-Based Sampling : 実験結果

■ 他手法との比較

- ピクセルベース手法 (Efros, Leung) では、入力テクスチャのコピーが繰り返される可能性がある。



入力テクスチャ
132 x 110

Efros, Leung
ピクセルベース手法

Patch-Based
Sampling
 $w_b=32$

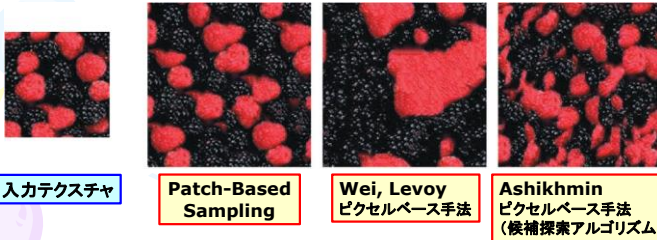
Patch-Based
Sampling
 $w_b=40$

出力テクスチャ
216 x 216

L. Liang, C. Liu, Y. Xu, B. Guo, H. Shum, "Real-Time Texture Synthesis by Patch-Based Sampling", 2001

Patch-Based Sampling : 実験結果

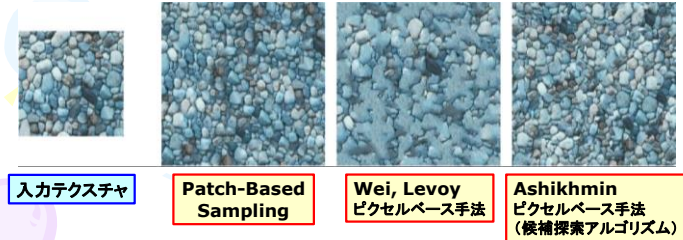
- 他手法との比較
- ピクセルベース手法では, **Ashikhmin** の手法はエッジの多い特徴パターンに対する合成結果が良好. **Wei, Levoy** の手法は不良.
- **Patch-Based Sampling** は良好.



L. Liang, C. Liu, Y. Xu, B. Guo, H. Shum, "Real-Time Texture Synthesis by Patch-Based Sampling", 2001

Patch-Based Sampling : 実験結果

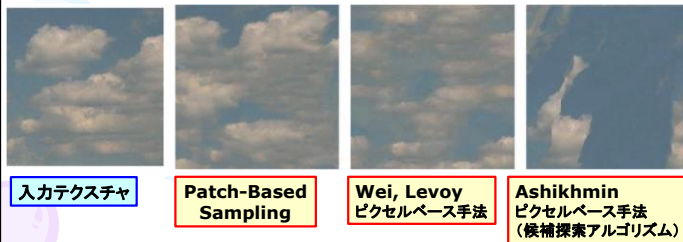
- 他手法との比較
- ピクセルベース手法では, **Ashikhmin** の手法はエッジの多い特徴パターンに対する合成結果が良好. **Wei, Levoy** の手法は不良.
- **Patch-Based Sampling** は良好.



L. Liang, C. Liu, Y. Xu, B. Guo, H. Shum, "Real-Time Texture Synthesis by Patch-Based Sampling", 2001

Patch-Based Sampling : 実験結果

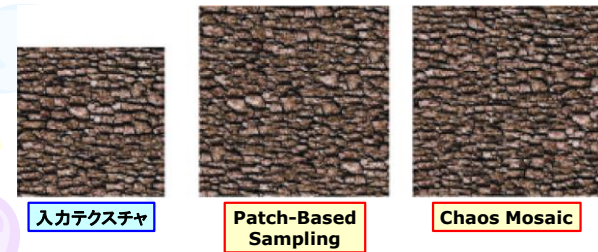
- 他手法との比較
- ピクセルベース手法では, **Wei, Levoy** の手法はスムーズな特徴パターンに対する合成結果が良好. **Ashikhmin** の手法は不良.
- **Patch-Based Sampling** は良好.



L. Liang, C. Liu, Y. Xu, B. Guo, H. Shum, "Real-Time Texture Synthesis by Patch-Based Sampling", 2001

Patch-Based Sampling : 実験結果

- 他手法との比較
- **Chaos Mosaic** は, 確率統計的に不規則な特徴パターンに対する合成結果が良好.
- **Patch-Based Sampling** も良好.



L. Liang, C. Liu, Y. Xu, B. Guo, H. Shum, "Real-Time Texture Synthesis by Patch-Based Sampling", 2001

Patch-Based Sampling : 実験結果

- 他手法との比較
- **Chaos Mosaic** は、規則的な特徴パターンに対する合成結果が不良.
- **Patch-Based Sampling** は良好.



入力テクスチャ

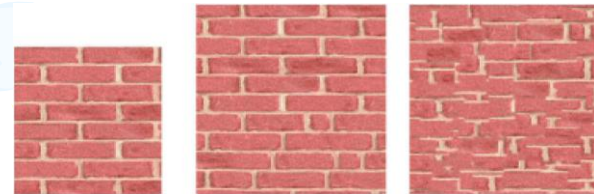
Patch-Based Sampling

Chaos Mosaic

L. Liang, C. Liu, Y. Xu, B. Guo, H. Shum, "Real-Time Texture Synthesis by Patch-Based Sampling", 2001

Patch-Based Sampling : 実験結果

- 他手法との比較
- **Chaos Mosaic** は、規則的な特徴パターンに対する合成結果が不良.
- **Patch-Based Sampling** は良好.



入力テクスチャ

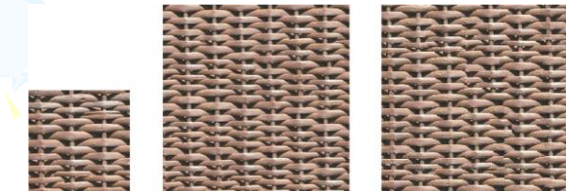
Patch-Based Sampling

Chaos Mosaic

L. Liang, C. Liu, Y. Xu, B. Guo, H. Shum, "Real-Time Texture Synthesis by Patch-Based Sampling", 2001

Patch-Based Sampling : 実験結果

- 他手法との比較
- **Patch-Based Sampling** の **feathering** は重なり部分でスムーズな色の変化を生み出す.
- **Image Quilting** の **境界線最適化** は、境界線に沿って急激な色の変化を生み出す場合がある.



入力テクスチャ

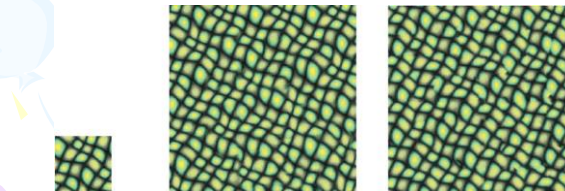
Patch-Based Sampling

Image Quilting

L. Liang, C. Liu, Y. Xu, B. Guo, H. Shum, "Real-Time Texture Synthesis by Patch-Based Sampling", 2001

Patch-Based Sampling : 実験結果

- 他手法との比較
- **Patch-Based Sampling** の **feathering** は重なり部分でスムーズな色の変化を生み出す.
- **Image Quilting** の **境界線最適化** は、境界線に沿って急激な色の変化を生み出す場合がある.



入力テクスチャ

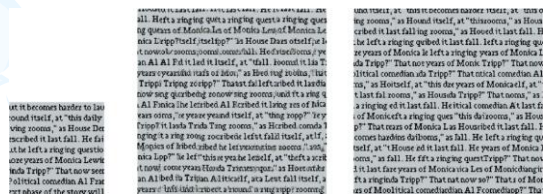
Patch-Based Sampling

Image Quilting

L. Liang, C. Liu, Y. Xu, B. Guo, H. Shum, "Real-Time Texture Synthesis by Patch-Based Sampling", 2001

Patch-Based Sampling : 実験結果

- 他手法との比較
 - 文字パターンのようなテクスチャに対しては、**feathering** では合成結果にぼやける部分が生じ、**境界線最適化**より劣る。



入力テクスチャ

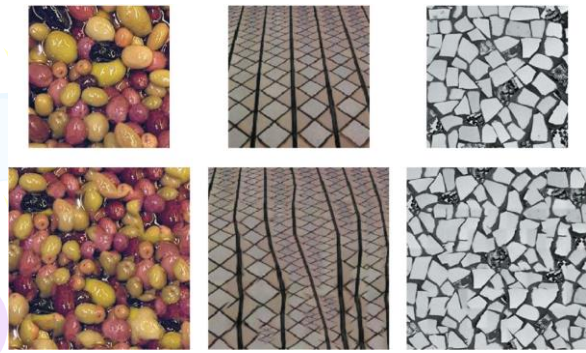
Patch-Based Sampling

Image Quilting

L. Liang, C. Liu, Y. Xu, B. Guo, H. Shum, "Real-Time Texture Synthesis by Patch-Based Sampling", 2001

Patch-Based Sampling : 実験結果

- Patch-Based Sampling の失敗例



L. Liang, C. Liu, Y. Xu, B. Guo, H. Shum, "Real-Time Texture Synthesis by Patch-Based Sampling", 2001

ハイブリッド手法

- C. Wu ら (2004) によるハイブリッド手法
- ピクセルベース手法 (候補探索アルゴリズム) とパッチベース手法を組み合わせて、候補ピクセルが入力テクスチャの下端と右端で外にはみ出す (連続的探索の行き止まり) 問題を解決する。
- C. Wu, Y. Lai, W. Tai, "A Hybrid-based Texture Synthesis Approach", 2004



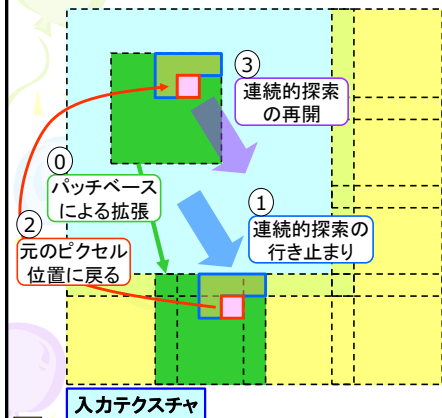
入力テクスチャ

候補探索アルゴリズム

ハイブリッド手法

C. Wu, Y. Lai, W. Tai, "A Hybrid-based Texture Synthesis Approach", 2004

ハイブリッド手法



- 候補探索アルゴリズムの行き止まりの問題を解決する。
- 入力テクスチャの下端と右端をパッチベース手法で拡張する。
- ピクセルベース手法の候補探索アルゴリズムで、候補ピクセルが入力テクスチャの下端と右端で外にはみ出す (探索が行き止まった) 場合 → はみ出した候補ピクセルを元の位置に戻す。

入力テクスチャ

C. Wu, Y. Lai, W. Tai, "A Hybrid-based Texture Synthesis Approach", 2004