

イメージシンセシス

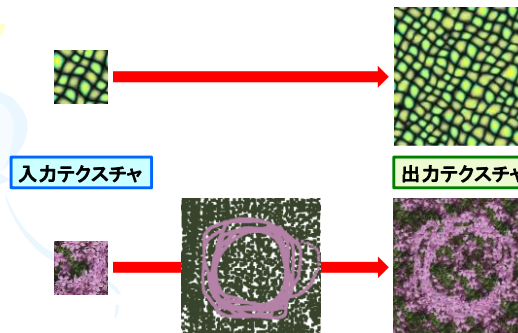
テクスチャ合成 (3D曲面上の合成)

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

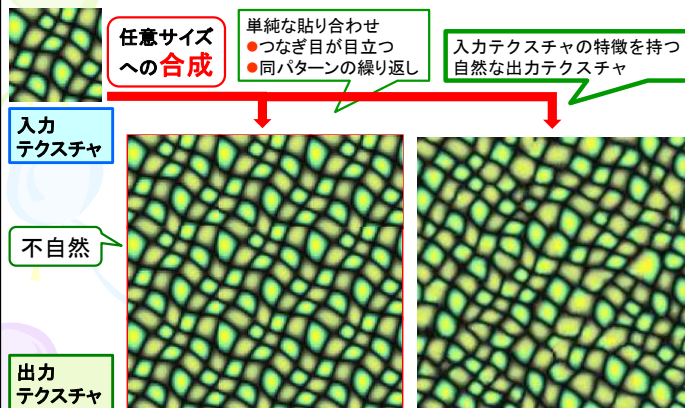
- 入力テクスチャからサイズの異なる自然な出力テクスチャを合成する。



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

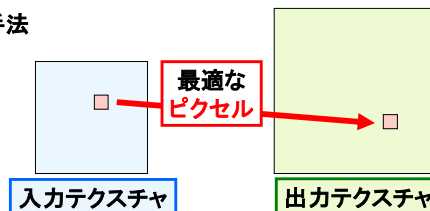


L. Wei, M. Levoy, "Fast Texture Synthesis using Tree-structured Vector Quantization", 2000

テクスチャ合成の分類

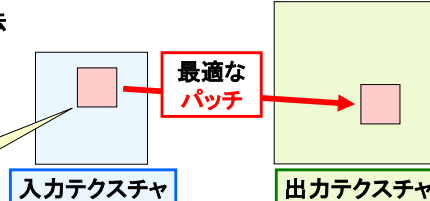
■ ピクセルベース手法

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

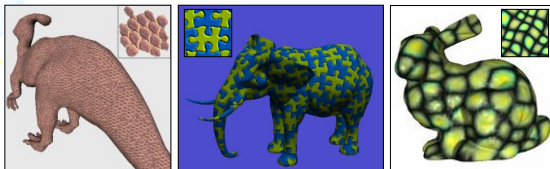
入力テクスチャから部分テクスチャ(パッチ)を切り出し、つなが目が目立たないように貼り合わせる。



一つのパッチは幾つかのピクセルからなる。

3D曲面上のテクスチャ合成

- 2次元テクスチャを用いて3次元物体の表面(surface, サーフェイス)上でテクスチャ合成を行う。
- 3次元物体の表面を“3D曲面”と呼ぶことにする。
- 3次元への拡張に伴い、2次元には無かった問題の解決が必要。
 - つなぎ目や同パターンの繰り返しだけでなく、3次元化に伴う歪みの軽減。
 - 3D曲面に沿ったテクスチャの方向とスケールの決定が必要。
 - どのような順序で曲面上をテクスチャで覆っていくか。



E. Praun, A. Finkelstein, H. Hoppe, “Lapped Textures”, 2000

G. Turk, “Texture Synthesis on Surfaces”, 2001

L. Wei, M. Levoy, “Texture Synthesis over Arbitrary Manifold Surfaces”, 2001

Lapped Textures

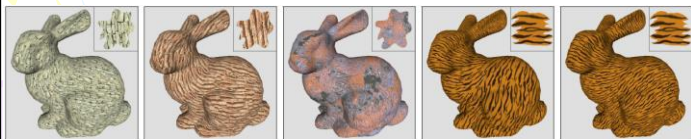
- E. Praun ら (2000) による Lapped Textures
 - パッチベース手法のアプローチにより、2D入力テクスチャを3D曲面上に合成する。
 - 3D曲面をあらゆるメッシュ上の各部分でローカル座標系に基づく表面パッチを生成し、テクスチャパッチを貼り付ける。
- E. Praun, A. Finkelstein, H. Hoppe, “Lapped Textures”, SIGGRAPH 2000



E. Praun, A. Finkelstein, H. Hoppe, “Lapped Textures”, 2000

Lapped Textures

- テクスチャパッチの生成
- メッシュ上における方向とスケールの設定
- 表面パッチの生成
- マッピング関数の補正



E. Praun, A. Finkelstein, H. Hoppe, “Lapped Textures”, 2000

Lapped Textures: テクスチャパッチの生成

- 元のテクスチャから貼り付ける単位としてテクスチャパッチを切り出す。
- ユーザによる手動生成
 - エッジが明瞭なパターン等の場合、パッチの形状をエッジに合わせて、不自然な切れ目ができないようにする。
- 自動生成
 - 一様なランダムなパターン等の場合、パッチの形状は結果に大きくは影響しないので、円形などの適当な形状で切り抜く。



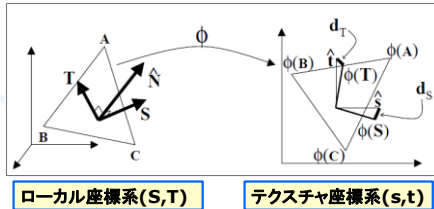
手動生成

自動生成

E. Praun, A. Finkelstein, H. Hoppe, “Lapped Textures”, 2000

Lapped Textures: メッシュ上における方向とスケールの設定

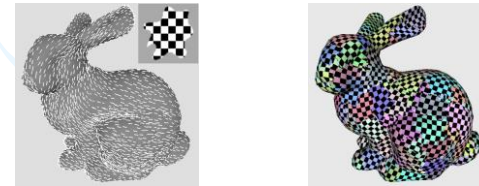
- 3角形メッシュ上の各3角形面に対するローカル座標系(S,T)の設定
 - 3DベクトルSを右向きベクトル, Tを上向きベクトルとする.
 - テクスチャパッチの貼り付けの際, 3D→2Dのマッピング関数 Φ による $(\Phi(S), \Phi(T))$ をテクスチャパッチの2Dテクスチャ座標系(s,t)の方向に合わせる.
 - 3角形面の法線ベクトルNに対して, ベクトルS,T,Nは互いに直交.



E. Praun, A. Finkelstein, H. Hoppe, "Lapped Textures", 2000

Lapped Textures: メッシュ上における方向とスケールの設定

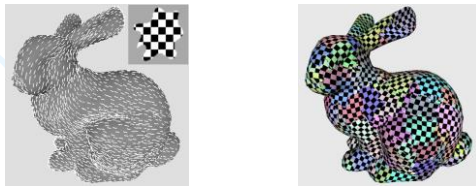
- 3角形面のローカル座標系(S,T)の設定方法
 - メッシュ上の幾つかの3角形面にユーザが上向きベクトルTを設定.
 - 上向きベクトルTの大きさによりテクスチャパッチのスケールを決定.
 - 他の3角形面のベクトルTを補間により求める.
 - 全ての3角形面について, 上向きベクトルTと法線ベクトルNから右向きベクトルSを求める.



E. Praun, A. Finkelstein, H. Hoppe, "Lapped Textures", 2000

Lapped Textures: 表面パッチの生成

- メッシュ上の隣接する複数の3角形面から, 一つのテクスチャパッチを貼り付けるための表面パッチを生成する.
- まだテクスチャパッチで覆われていない3角形面をランダムに選択し, 一つの表面パッチを構成する初期3角形面とする.
- 初期3角形面の内部でランダムな一点を選び, テクスチャパッチのテクスチャ座標系の中心に合わせる.
- 初期3角形面の周囲の3角形面を表面パッチに併合していき, 一つのテクスチャパッチの大きさになるまで広げる.



E. Praun, A. Finkelstein, H. Hoppe, "Lapped Textures", 2000

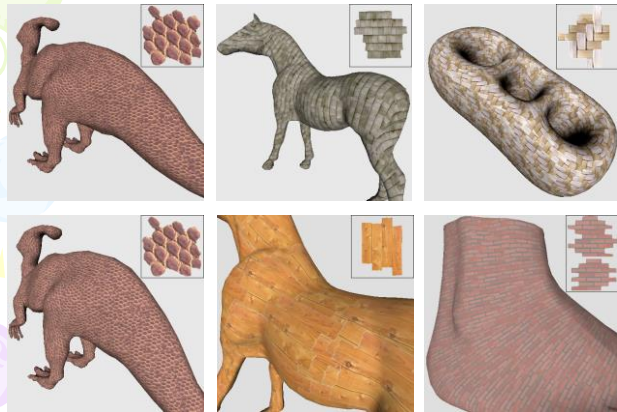
Lapped Textures: マッピング関数の補正

- 一つの表面パッチにおいて, マッピング関数 Φ を初期3角形面のローカル座標系(S,T)からテクスチャ座標系(s,t)への投影として定義する.
- 初期3角形面内のランダムな一点(S,T)に関する $(\Phi(S), \Phi(T))$ をテクスチャ座標系(s,t)の中心に合わせる.
- ➡ 表面パッチ内の他の3角形面について, ローカル座標系(S,T)とテクスチャ座標系(s,t)の対応が不十分.
- ➡ 隣り合う表面パッチの間でテクスチャパッチの方向が不連続.



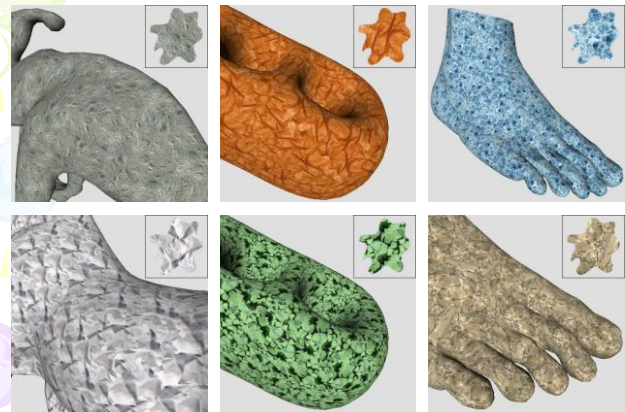
E. Praun, A. Finkelstein, H. Hoppe, "Lapped Textures", 2000

Lapped Textures: 実験結果



E. Praun, A. Finkelstein, H. Hoppe, "Lapped Textures", 2000

Lapped Textures: 実験結果



E. Praun, A. Finkelstein, H. Hoppe, "Lapped Textures", 2000

Lapped Textures: 実験結果

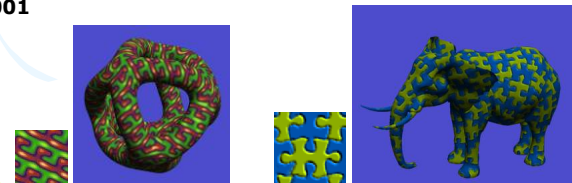
- 不自然な合成結果



E. Praun, A. Finkelstein, H. Hoppe, "Lapped Textures", 2000

G.Turkの方法

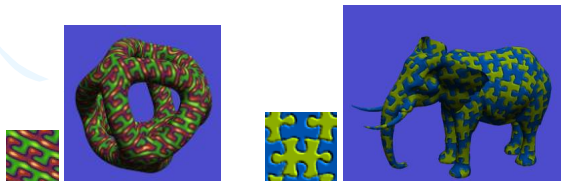
- G. Turk (2001) による3D曲面上のテクスチャ合成
- ピクセルベース手法の応用により, 2D入力テクスチャを3D曲面上に合成する.
- 3D曲面をあらゆるメッシュ上にベクトル場を生成し, メッシュ頂点の処理順序を決め, ベクトル場の方向に合わせてテクスチャを合成する.
- 入力テクスチャとメッシュを多重解像度化(階層化)して合成処理に利用することで, 品質の高いテクスチャ合成を実現する.
- G. Turk, "Texture Synthesis on Surfaces", SIGGRAPH 2001



G. Turk, "Texture Synthesis on Surfaces", 2001

G.Turkの方法

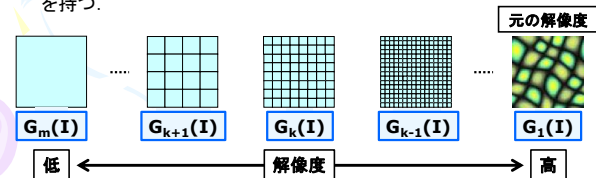
- 入力テクスチャとメッシュの多重解像度化
- メッシュ上のベクトル場の生成
- メッシュ上の頂点の処理順序
- メッシュ上のテクスチャ合成



G. Turk, "Texture Synthesis on Surfaces", 2001

G.Turkの方法: 入力テクスチャとメッシュの多重解像度化

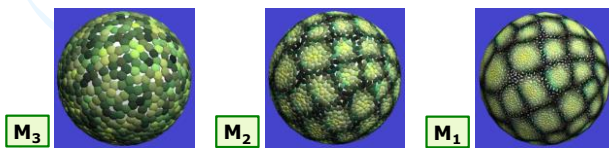
- 入力テクスチャ I の多重解像度化
 - 最高解像度(元の解像度)のテクスチャ $G_1(I)$ を, 順次, 低解像度化して, $G_2(I), G_3(I), \dots, G_m(I)$ を生成する.
 - 解像度レベル k のテクスチャ $G_k(I)$ はレベル $k+1$ のテクスチャ $G_{k+1}(I)$ の **4倍の個数のピクセル** を持つ.
 - レベル $k+1$ の親ピクセル $G_{k+1}(i,j)$ はレベル k の **4個の子ピクセル** $G_k(2i,2j), G_k(2i+1,2j), G_k(2i,2j+1), G_k(2i+1,2j+1)$ を持つ.



G. Turk, "Texture Synthesis on Surfaces", 2001

G.Turkの方法: 入力テクスチャとメッシュの多重解像度化

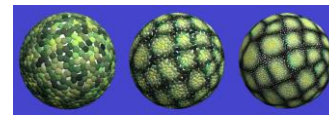
- メッシュ M の多重解像度化
 - 3D曲面は**3角形メッシュ**で表現する.
 - 元のメッシュは頂点の粗密が一律な保証がない等の問題があり得る.
→ 3D曲面上にテクスチャ合成のための頂点群を生成する.
 - 高解像度から低解像度にメッシュ $M_1, M_2, M_3, \dots, M_m$ を生成する.
 - 解像度レベル k のメッシュ $M_k = (V_k, T_k)$ は頂点群 V_k と3角形群 T_k からなる.
 - 頂点群 V_k は頂点群 V_{k+1} の **4倍の個数の頂点** を持つ.



G. Turk, "Texture Synthesis on Surfaces", 2001

G.Turkの方法: 入力テクスチャとメッシュの多重解像度化

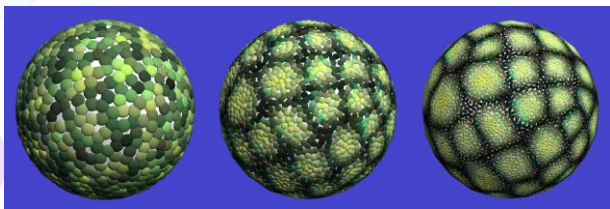
- メッシュ M の多重解像度化
 - 多重解像度化の方法
 - 3D曲面上に n 個の頂点をランダムに置き, 互いの斥力により一様な分布にする. これらを最低解像度のメッシュ M_m の頂点群 V_m とする.
 - **3n** 個の頂点を曲面上に追加し, 計 **4n** 個の頂点を斥力により一様な分布にする. ただし, 頂点群 V_m の位置は固定しておく. この **4n** 個の頂点をメッシュ M_{m-1} の頂点群 V_{m-1} とする.
 - 以降, 同様に, 頂点群 $V_{m-2}, V_{m-3}, \dots, V_1$ を生成する.
 - 各解像度レベル k の頂点群 V_k に **Delaunay triangulation** を適用して3角形群 T_k を生成する.



G. Turk, "Texture Synthesis on Surfaces", 2001

G.Turkの方法: 入力テクスチャとメッシュの多重解像度化

- 多重解像度メッシュ上のオペレーション
 - 補間 (Interpolation)
 - ローパスフィルタ (Low-pass filter, blur)
 - ダウンサンプリング (Downsampling)
 - アップサンプリング (Upsampling)



G. Turk, "Texture Synthesis on Surfaces", 2001

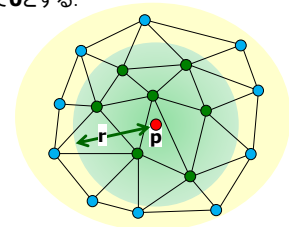
G.Turkの方法: 入力テクスチャとメッシュの多重解像度化

- 多重解像度メッシュ上のオペレーション
 - 補間 (Interpolation)
 - 下式により, 曲面上の任意の点 p について, 点 p から半径 r 以内の N_r 個のメッシュ頂点 v_i の色 $C(v_i)$ から点 p の色 $C(p)$ を求める.
 - 各頂点 v_i に与える重み w は, 点 p から距離が遠いほど小さくし, 点 p の位置で 1 , 半径 r だけ離れた位置で 0 とする.

$$C(p) = \frac{\sum_{i=1}^{N_r} w(|p - v_i|/r) C(v_i)}{\sum_{i=1}^{N_r} w(|p - v_i|/r)}$$

$$w(x) = 2f^3 - 3f^2 + 1$$

$$\begin{cases} w(0) = 1 \\ w(1) = 0 \end{cases}$$



G. Turk, "Texture Synthesis on Surfaces", 2001

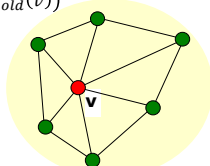
G.Turkの方法: 入力テクスチャとメッシュの多重解像度化

- 多重解像度メッシュ上のオペレーション
 - ローパスフィルタ (Low-pass filter, blur)
 - 下式の反復により, メッシュ上で頂点 v に隣接する N_a 個の頂点 v_i の色 $C(v_i)$ を用いて頂点 v の色 $C(v)$ を **ブラー (スムージング)** する.
 - パラメータ t は小さな値とする ($t=0.1$ など).
 - 重み w_i は頂点 v と v_i の距離の逆数によって決める.

$$C_{new}(v) = C_{old}(v) + t \sum_{i=1}^{N_a} w_i (C(v_i) - C_{old}(v))$$

$$= \alpha C_{old}(v) + \sum_{i=1}^{N_a} \beta_i C(v_i)$$

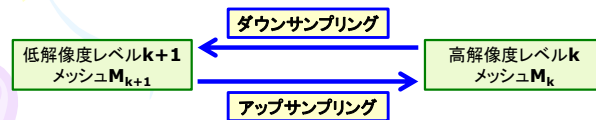
$$\sum_{i=1}^{N_a} w_i = 1$$



G. Turk, "Texture Synthesis on Surfaces", 2001

G.Turkの方法: 入力テクスチャとメッシュの多重解像度化

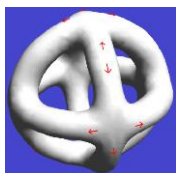
- 多重解像度メッシュ上のオペレーション
 - ダウンサンプリング (Downsampling)
 - 高解像度のメッシュ M_k 上の各頂点の色を **ブラー** する (ぼかす) ことによって, 低解像度のメッシュ M_{k+1} 上の各頂点の色を求める.
 - アップサンプリング (Upsampling)
 - 低解像度のメッシュ M_{k+1} 上の各頂点の色を **補間** することによって, 高解像度のメッシュ M_k 上の各頂点の色を求める.



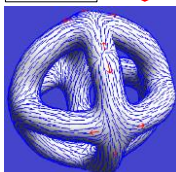
G. Turk, "Texture Synthesis on Surfaces", 2001

G.Turkの方法: メッシュ上のベクトル場の生成

- 3D曲面上の**テクスチャの方向**を決めるため、ユーザの入力に基づき、メッシュ上に**ベクトル場**を生成する。
- 最高解像度メッシュ M_1 の全頂点に**ゼロベクトル**を与える。
- メッシュ M_1 の幾つかの頂点に**ユーザが非ゼロベクトル**を与える。
- (各頂点の色の代わりにベクトルを対象とした)**ダウンサンプリング**をメッシュ M_1 から繰り返し適用して、最低解像度メッシュ M_m の全頂点にベクトルを得る。
- (各頂点の色の代わりにベクトルを対象とした)**ローパスフィルタ**でメッシュ M_m の頂点のベクトルをスムージングし、全頂点のベクトルを**非ゼロ**にする。このとき、はじめから**非ゼロ**のベクトルは固定しておく。
- **アップサンプリング**をメッシュ M_m から繰り返し適用して、最高解像度メッシュ M_1 の全頂点に**非ゼロベクトル**を得る。



ユーザ入力

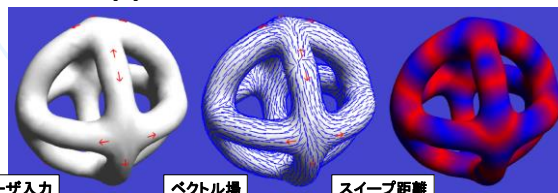


ベクトル場

G. Turk, "Texture Synthesis on Surfaces", 2001

G.Turkの方法: メッシュ上の頂点の処理順序

- メッシュ上の頂点ごとに**ピクセルベースのテクスチャ合成**を行うため、メッシュ上に生成したベクトル場から、頂点の合成処理の順序付けを行う。
- ベクトル場の生成と同時に、低解像度から高解像度に向けて、メッシュ M_m から M_1 のそれぞれについて頂点の順序付けを行う。
- メッシュ上の任意の頂点を**アンカー頂点A**とする。
- ベクトル場の流れに従い、アンカー頂点Aから各頂点vまでの**スイープ距離s(v)**を求める。下流は正值、上流は負値となる。
- スイープ距離s(v)の値をソートすることで**頂点の順序付け**を行う。



ユーザ入力

ベクトル場

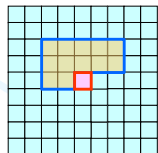
スイープ距離

G. Turk, "Texture Synthesis on Surfaces", 2001

G.Turkの方法: メッシュ上のテクスチャ合成

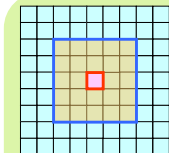
- **単一解像度レベルにおける基本アルゴリズム**
- スイープ距離s(v)の順序でメッシュM上の頂点vごとに入力テクスチャI上の最適なピクセルpの色I(p)を割り当てる。
- 処理中の頂点vの近隣N(v)の色パターンに対して、ピクセルpの近隣N(p)の色パターンが最も類似度するものを探索する。
- 入力テクスチャI上のピクセルpの近隣N(p)には、**正方形(full square)**と**半正方形(half square)**の近隣ピクセル群を用いる。

入力
テクスチャ

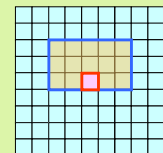


Wei, Levoy

近隣サイズ
5x5



正方形

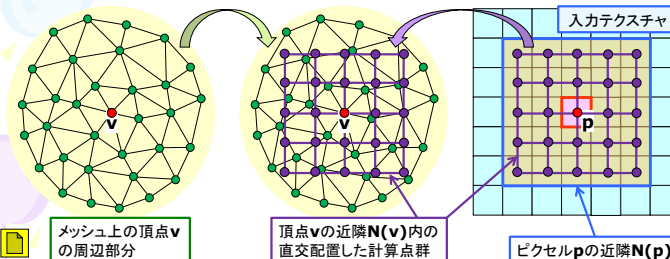


半正方形

G. Turk, "Texture Synthesis on Surfaces", 2001

G.Turkの方法: メッシュ上のテクスチャ合成

- **単一解像度レベルにおける基本アルゴリズム**
- メッシュM上の頂点vの近隣N(v)の色パターンは次のように得る。
 - 頂点vを中心として直交配置した計算点の位置で色を求める。
 - 計算点の配置と数は、比較する入力テクスチャI上のピクセルpの近隣(近隣ピクセル群)N(p)に合わせる。



メッシュ上の頂点v
の周辺部分

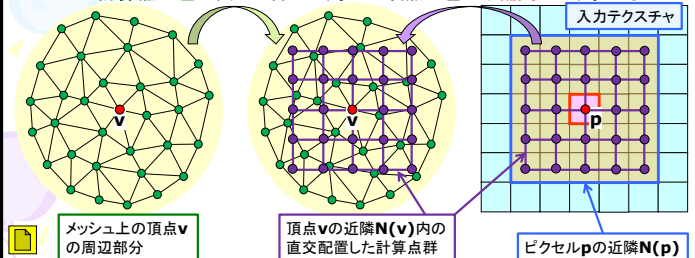
頂点vの近隣N(v)内の
直交配置した計算点群

ピクセルpの近隣N(p)

G.Turkの方法: メッシュ上のテクスチャ合成

■ 単一解像度レベルにおける基本アルゴリズム

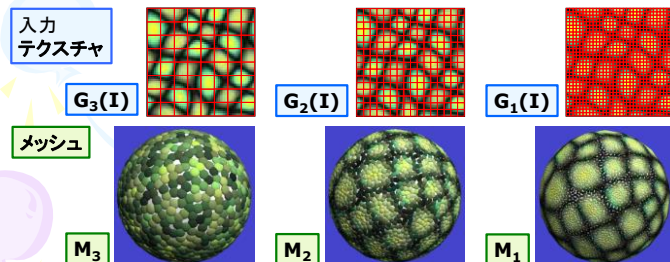
- メッシュM上の頂点 v の近隣 $N(v)$ の色パターンは次のように得る。
 - 計算点の位置は、ベクトル場における頂点 v の持つベクトル $O(v)$ 、ならびに、それに直交するベクトル $P(v)$ (法線ベクトル $Q(v)$)にも直交)を用いて求める。頂点 v からオフセット (i,j) の計算点の位置は、メッシュ頂点間の平均距離 r を用いて、 $irO(v)+jrP(v)$ とする。
 - 計算点の色は、処理済みの周辺の頂点の色から補間により求める。



G.Turkの方法: メッシュ上のテクスチャ合成

■ 多重解像度レベルを利用したアルゴリズム

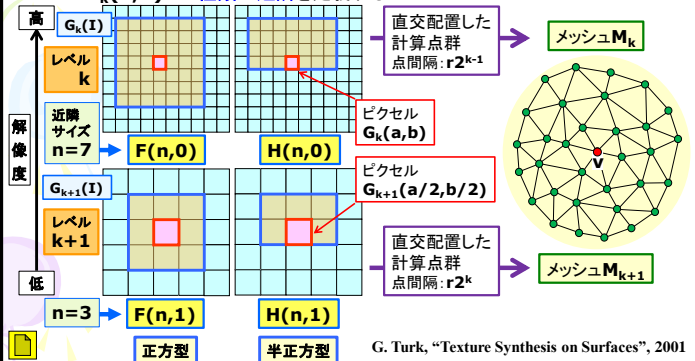
- 多重解像度化した入力テクスチャ $G_k(I)$ とメッシュ M_k を用いてテクスチャ合成を行う。
- 低解像度から高解像度に向けて(最低解像度のテクスチャ $G_m(I)$ とメッシュ M_m から最高解像度の $G_1(I)$ と M_1 に向けて)合成を進める。



G.Turkの方法: メッシュ上のテクスチャ合成

■ 多重解像度レベルを利用したアルゴリズム

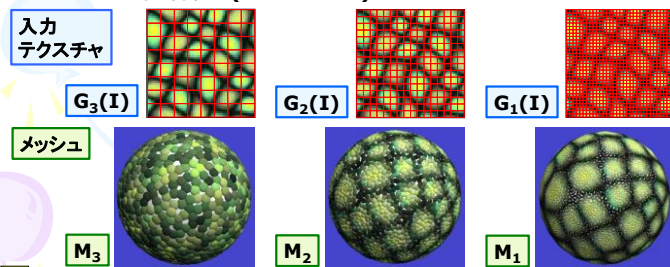
- 2つの解像度レベル $k, k+1$ を利用して、処理中のレベル k のピクセル $G_k(a,b)$ の4種類の近隣を定義する。



G.Turkの方法: メッシュ上のテクスチャ合成

■ 多重解像度レベルを利用したアルゴリズム

- 最低解像度メッシュ M_m の初期化
- 解像度レベル $k = m-1 \sim 1$ のメッシュ M_k 上のテクスチャ合成
 - 1パス目: 外挿(extrapolation)
 - 2パス目: 詳細化(refinement)



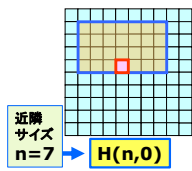
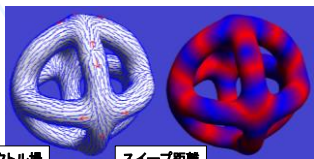
G.Turkの方法: メッシュ上のテクスチャ合成

■ 多重解像度レベルを利用したアルゴリズム

■ 最低解像度メッシュ M_m の初期化

- 最低解像度メッシュ M_m の各頂点 v に対して、最低解像度テクスチャ $G_m(I)$ のピクセル p の色 $I(p)$ をランダムに与える。
- スイープ距離 $s(v)$ の順にメッシュ M_m 上の頂点 v を処理する。最低解像度レベル m の半正方形近隣 $H(n,0)$ を用い、頂点 v と最類似の近隣を持つピクセル p をテクスチャ $G_m(I)$ 上で探索し、その色を頂点 v に与える。

近隣には合成済み頂点だけが関与。



G. Turk, "Texture Synthesis on Surfaces", 2001

G.Turkの方法: メッシュ上のテクスチャ合成

■ 多重解像度レベルを利用したアルゴリズム

■ 解像度レベル $k = m-1 \sim 1$ のメッシュ M_k 上のテクスチャ合成

低解像度から高解像度に向けて($k=m-1 \rightarrow 1$), 2パスの合成で、メッシュ M_k の各頂点 v にテクスチャ $G_k(I)$ の最適なピクセル p の色を与える。各パスとも、解像度レベル k のメッシュ M_k 上で、スイープ距離 $s(v)$ の順に頂点 v を処理する。

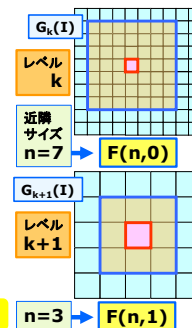
■ 1パス目: 外挿 (extrapolation)

レベル $k+1$ の正方形近隣 $F(n,1)$ を用い、頂点 v と最類似の近隣を持つピクセル p を探索し、その色を頂点 v に与える。

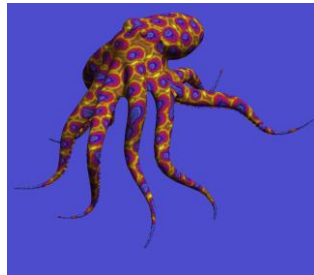
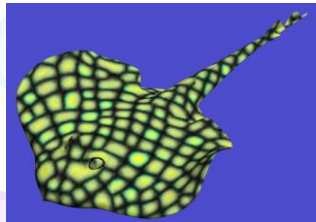
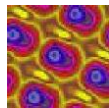
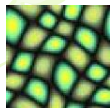
■ 2パス目: 詳細化 (refinement)

レベル k と $k+1$ の正方形近隣 $F(n,0)$ と $F(n,1)$ を用い、頂点 v と最類似の近隣を持つピクセル p を探索し、その色を頂点 v に与える。頂点 v とピクセル p とも、近隣は2レベルで構成される。

両パスとも、近隣には合成済み頂点だけが関与。

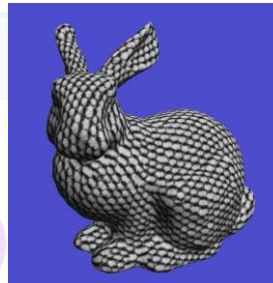
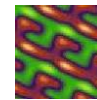
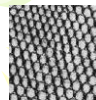


G.Turkの方法: 実験結果



G. Turk, "Texture Synthesis on Surfaces", 2001

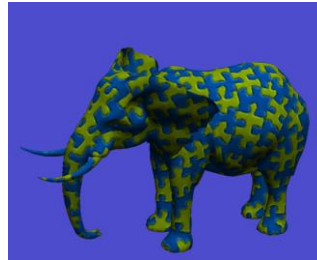
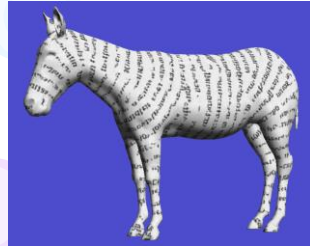
G.Turkの方法: 実験結果



G. Turk, "Texture Synthesis on Surfaces", 2001

G.Turkの方法: 実験結果

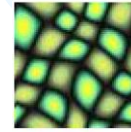
Figure 10: A zebra-like texture synthesized as described in the previous section. The texture is a function of position on the surface. Particular lines of text are visible, such as "each concept will need at least one principle" and "each idea is a function of such a function". The texture is synthesized on the surface of a zebra. Whereas no particular texture (such as "zebra") is visible, the texture is a function of position on the surface. Higher detail features, such as the eyes, are visible in the texture.



G. Turk, "Texture Synthesis on Surfaces", 2001

L.Weil,M.Levoyの方法

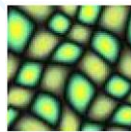
- L. Wei, M. Levoy (2001) による3D曲面上のテクスチャ合成
- ピクセルベース手法の応用により, 2D入力テクスチャを3D曲面上に合成する.
- 3D曲面をあらゆるメッシュの頂点ごとにテクスチャの方向性を考慮したローカル座標系を設定し, 適切な向きにテクスチャを合成する.
- サンプルテクスチャとメッシュを多重解像度化(階層化)して合成処理に利用することで, 品質の高いテクスチャ合成を実現する.
- L. Wei, M. Levoy, "Texture Synthesis over Arbitrary Manifold Surfaces", SIGGRAPH 2001



L. Wei, M. Levoy, "Texture Synthesis over Arbitrary Manifold Surfaces", 2001

L.Weil,M.Levoyの方法

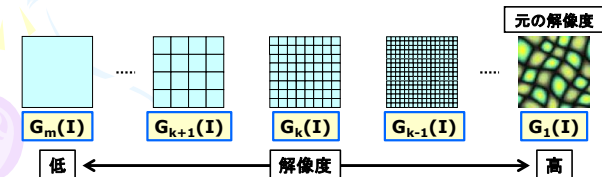
- 入力テクスチャとメッシュの多重解像度化
 - メッシュ頂点のローカル座標系の設定
 - メッシュ頂点の処理順序
 - メッシュ上のテクスチャ合成
- 以降の説明では, 変数名や解像度レベルの昇降順などの表記を G. Turk の論文に合わせている. 実際の L. Wei, M. Levoy の論文では表記が異なる.



L. Wei, M. Levoy, "Texture Synthesis over Arbitrary Manifold Surfaces", 2001

L.Weil,M.Levoyの方法: 入力テクスチャとメッシュの多重解像度化

- 入力テクスチャIの多重解像度化
- G. Turk の方法と同様に, 入力テクスチャIを多重解像度化し, 解像度レベルkの多重解像度テクスチャ $G_k(I)$ を生成する.
- 解像度レベルが1つ異なるごとにピクセルの個数は4倍になる.



L. Wei, M. Levoy, "Texture Synthesis over Arbitrary Manifold Surfaces", 2001

L.Wei,M.Levoyの方法:入力テクスチャとメッシュの多重解像度化

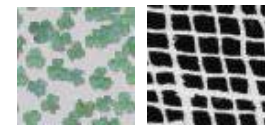
- **メッシュMの多重解像度化**
 - 3D曲面は**3角形メッシュ**で表現する。
 - 元のメッシュに対して **mesh simplification algorithm** を適用し、多重解像度テクスチャ $G_k(I)$ に合わせて、メッシュMを多重解像度化して、解像度レベルkの多重解像度メッシュ M_k を生成する。
 - 各レベルkのメッシュ M_k に**retiling**する。
 - メッシュ上の**頂点の分布を一様**にする。
 - **頂点の密度を調整**する。→ 合成される**テクスチャのスケールの調整**
 - 解像度レベルが1つ異なるごとに**頂点の個数は4倍**になる。



L. Wei, M. Levoy, "Texture Synthesis over Arbitrary Manifold Surfaces", 2001

L.Wei,M.Levoyの方法:メッシュ頂点のローカル座標系の設定

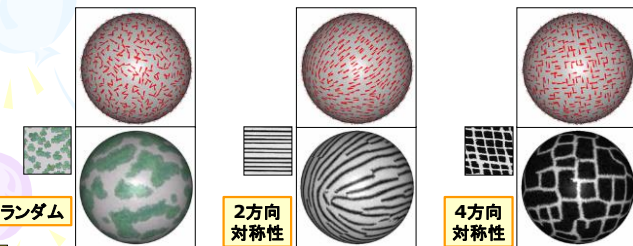
- 3D曲面上の**テクスチャの方向**を決めるため、多重解像度メッシュ M_k の頂点ごとにローカル座標系(直交座標系)を設定する。
 - テクスチャの右方向ベクトル \vec{s}
 - テクスチャの上方向ベクトル \vec{t}
 - 曲面の法線方向ベクトル \vec{n}
- テクスチャの**方向性に関する特徴**により、異なる方法を適用する。
 - **等方性(isotropic)**の場合
 - 頂点ごとにランダムに与える。
 - 合成結果の品質に影響しない。
 - **異方性(anisotropic)**の場合
 - ユーザによる手動入力
 - 緩和法(**relaxation**)による自動最適化



L. Wei, M. Levoy, "Texture Synthesis over Arbitrary Manifold Surfaces", 2001

L.Wei,M.Levoyの方法:メッシュ頂点のローカル座標系の設定

- 異方性テクスチャのための緩和法による頂点ごとのローカル座標系の自動最適配置
 - 異方性テクスチャが**n方向の対称性**を持つとする。
 - 隣接する頂点間でテクスチャの右方向ベクトル \vec{s} が**(360/n)度の整数倍の角度差**に近くなるようにする。



L. Wei, M. Levoy, "Texture Synthesis over Arbitrary Manifold Surfaces", 2001

L.Wei,M.Levoyの方法:メッシュ頂点のローカル座標系の設定

- 異方性テクスチャのための緩和法による頂点ごとのローカル座標系の自動最適配置
 - 最低解像度レベルmのメッシュ M_m 上で各頂点にランダムなローカル座標系を与える。
 - 低解像度から高解像度に向けて、各レベルkのメッシュ M_k 上で、以下の処理により各頂点vに最適なローカル座標系を与える。
 - 一つ低解像度のメッシュ M_{k+1} 上の頂点のローカル座標系を補間してベクトル \vec{s} を初期化する。
 - 頂点vの近隣の頂点 v_i を2つのレベルのメッシュ M_k, M_{k+1} から求め、各頂点 v_i のベクトル \vec{s}_i と頂点vのベクトル \vec{s} の角度の差が**(360/n)度の整数倍**に近くなるように、緩和法により次式を最小化する。

$$E = \sum_{v_i} \left| \phi_i - \text{round} \left(\frac{\phi_i}{360/n} \right) \times (360/n) \right|^2 \quad \phi_i \text{は } \vec{s}_i \text{ と } \vec{s} \text{ の角度差}$$

L. Wei, M. Levoy, "Texture Synthesis over Arbitrary Manifold Surfaces", 2001

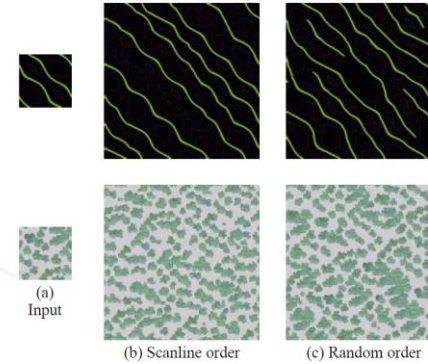
L.Weil,M.Levoyの方法:メッシュ頂点の処理順序

- メッシュ上の頂点ごとに**ピクセルベースのテクスチャ合成**を行うにあたり頂点の処理に特定の順序を設けず、**ランダムな順序で処理**する。
- 著者ら(L.Weil, M.Levoy)の先行研究である2D出力テクスチャの合成では、欠落領域を埋めるための**制約付き合成**において、L型近隣ピクセル群の非対称形状、ならびに、ピクセルの走査線順の処理に起因して、合成結果に不都合が見られた。
 - この解決策の一つとして、**近隣ピクセル群を常に正方形の対称形状とする2パスの多重解像度処理**により、合成結果を改善した。
 - 今回の**ランダム順序による頂点の処理**においても、**2パスの多重解像度処理**を導入し、良好な合成結果を目指す。

L. Wei, M. Levoy, "Texture Synthesis over Arbitrary Manifold Surfaces", 2001

L.Weil,M.Levoyの方法:メッシュ頂点の処理順序

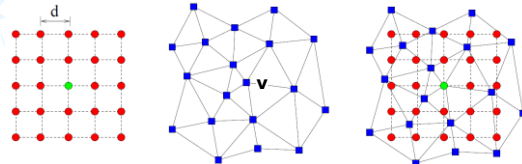
■ 実験結果



L. Wei, M. Levoy, "Texture Synthesis over Arbitrary Manifold Surfaces", 2001

L.Weil,M.Levoyの方法:メッシュ上のテクスチャ合成

- **メッシュ頂点の近隣の色パターンの計算**
G.Turkの方法と同様に、各解像度レベル k のメッシュ M_k 上で、頂点 v の**近隣 $N(v)$** の色パターンを次のように計算する。
- 入力テクスチャの**正方形近隣ピクセル群**に合わせて、頂点 v を中心としてローカル座標系のベクトル \vec{u} と \vec{t} の方向に直交配置した計算点群(**neighborhood template**)の位置で色を求める。



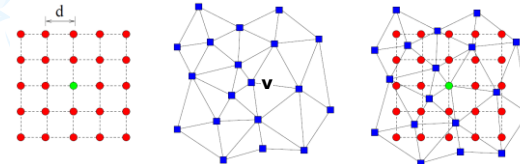
頂点 v の近隣 $N(v)$ 内の直交配置した計算点群

メッシュ上の頂点 v の周辺部分

L. Wei, M. Levoy, "Texture Synthesis over Arbitrary Manifold Surfaces", 2001

L.Weil,M.Levoyの方法:メッシュ上のテクスチャ合成

- **メッシュ頂点の近隣の色パターンの計算**
G.Turkの方法と同様に、各解像度レベル k のメッシュ M_k 上で、頂点 v の**近隣 $N(v)$** の色パターンを次のように計算する。
- 頂点と計算点群の密度を同等とするため、メッシュ M_k 上の3角形の平均面積 A に対して、計算点間の距離を $d = \sqrt{2A}$ とする。
- 計算点の色は、それを含む3角形の頂点の色を補間して求める。



頂点 v の近隣 $N(v)$ 内の直交配置した計算点群

メッシュ上の頂点 v の周辺部分

L. Wei, M. Levoy, "Texture Synthesis over Arbitrary Manifold Surfaces", 2001

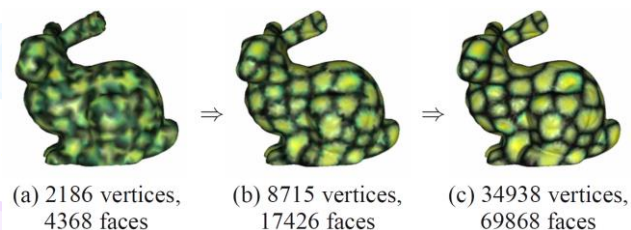
L.We,M.Levoyの方法:メッシュ上のテクスチャ合成

- 2パスの多重解像度処理によるテクスチャ合成
 - 初期化
 - 解像度レベル k ごとに、メッシュ M_k 上の各頂点にテクスチャ $G_k(I)$ のピクセルの色をランダムに与える。
 - 2パスのテクスチャ合成
 - 低解像度から高解像度に向けて、解像度レベル k ごとに、2つのパスで異なる**正方形近隣**を構成して色パターンを類似度評価することで、メッシュ M_k 上の各頂点に対してピクセルベースのテクスチャ合成を行う。
 - 1パス目: 低解像度のレベル $k+1$ 以上のメッシュとテクスチャを用いて正方形近隣を構成する。
 - 2パス目: 1パス目で用いた近隣にレベル k の近隣も含める。
- 各パスで、**頂点の処理順序はランダムとする。**
- 両パスとも、**近隣には合成済み頂点だけが関与。**

L. Wei, M. Levoy, "Texture Synthesis over Arbitrary Manifold Surfaces", 2001

L.We,M.Levoyの方法:メッシュ上のテクスチャ合成

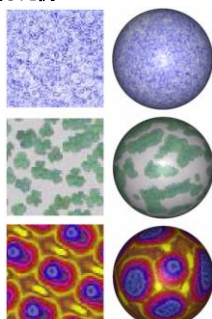
■ 実験結果



L. Wei, M. Levoy, "Texture Synthesis over Arbitrary Manifold Surfaces", 2001

L.We,M.Levoyの方法:実験結果

- 等方性テクスチャに対する頂点ごとのローカル座標系(ベクトル \vec{s} , \vec{t} の方向)をランダムに設定した例

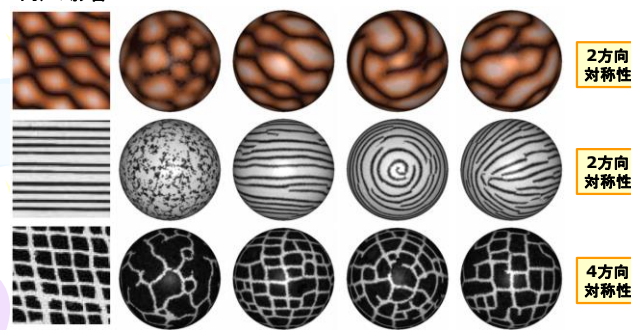


入力テクスチャ ランダム方向

L. Wei, M. Levoy, "Texture Synthesis over Arbitrary Manifold Surfaces", 2001

L.We,M.Levoyの方法:実験結果

- 異方性テクスチャに対する頂点ごとのローカル座標系(ベクトル \vec{s} , \vec{t} の方向)の影響

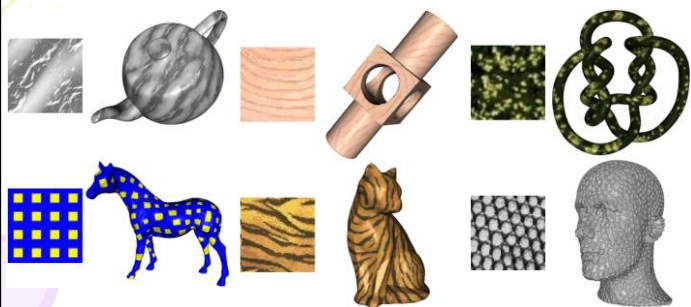


入力テクスチャ ランダム方向 緯度経度方向(側面図, 上面図) 緩和法

L. Wei, M. Levoy, "Texture Synthesis over Arbitrary Manifold Surfaces", 2001

L.Weil,M.Levoyの方法:実験結果

■ 様々な実験例



L. Wei, M. Levoy, "Texture Synthesis over Arbitrary Manifold Surfaces", 2001