

CEDEC 2008 CEDECラボ (2008.9.11)

プロシージャル技術の動向

北陸先端科学技術大学院大学

宮田一乗

自己紹介

日本アイ・ビー・エムに12年間
東京工芸大・芸術学部で教員4年間
北陸先端科学技術大学院大学で教員7年目

高校2年の頃から、ビデオゲームに熱中
会社員時代にファミコン同好会を作った
結婚してからサッパリ
子供が小学生になってから復活(といってもライトユーザ)



My Best Game



概要

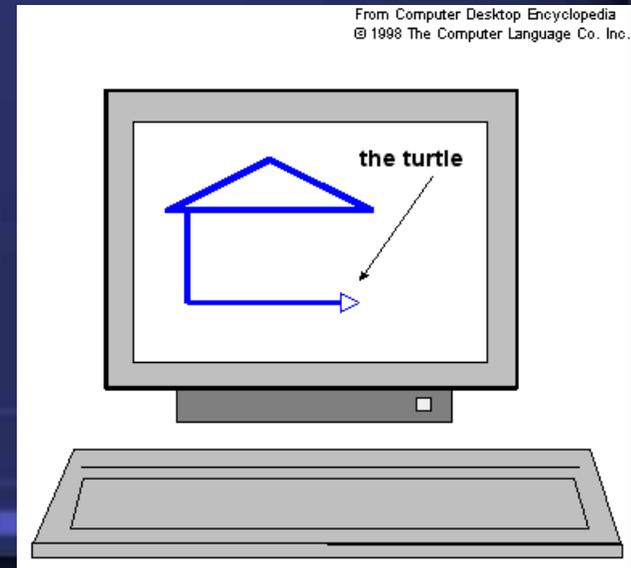
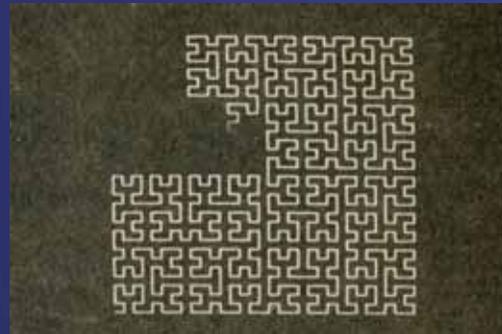
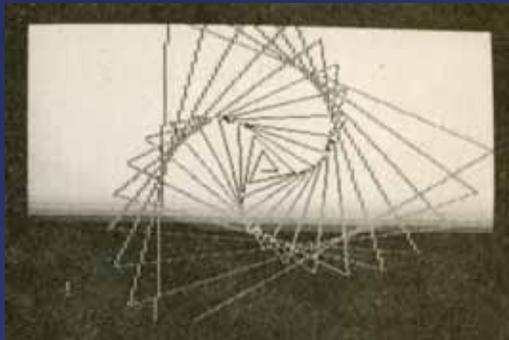
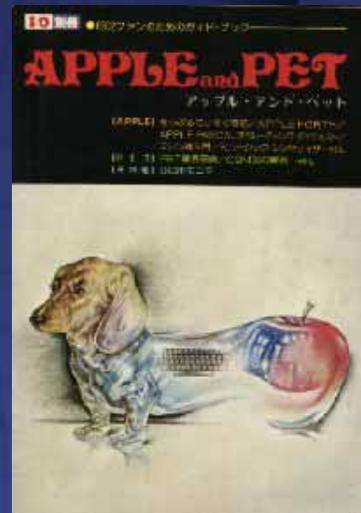
- はじめに 3
- プロシージャル手法について 45
- 最近のプロシージャル手法 10
- 展望: ゲームとの関係 5
- おまけ: 研究の紹介 12

生まれて初めて70MBもあるスライドを作りました！

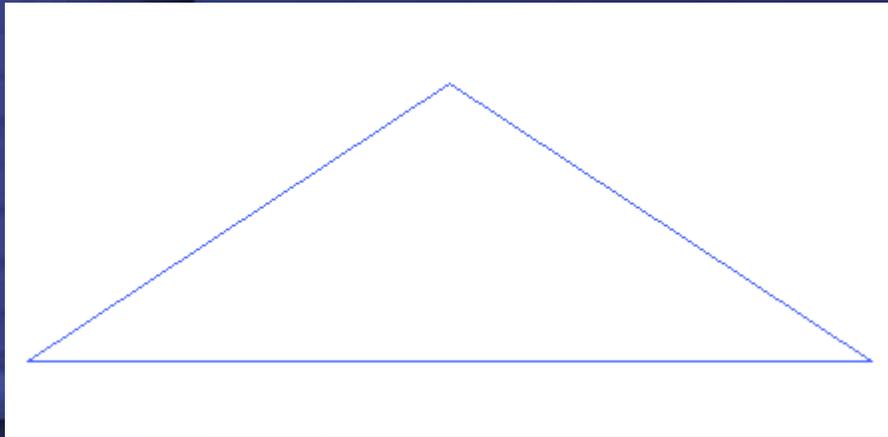
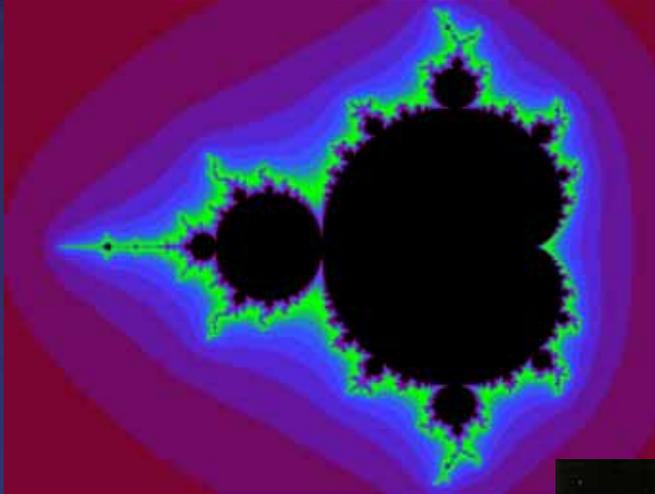
はじめに

My First Encounter...

Apple II [pascal, turtle graphics

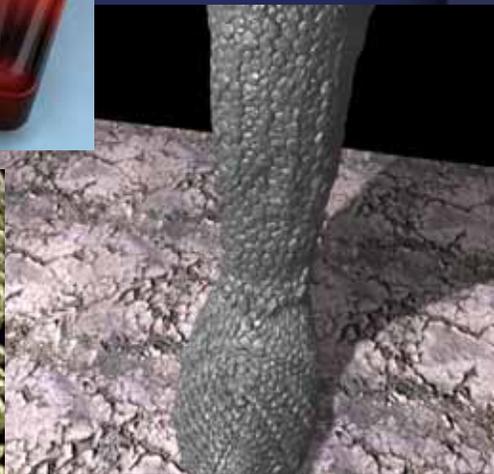
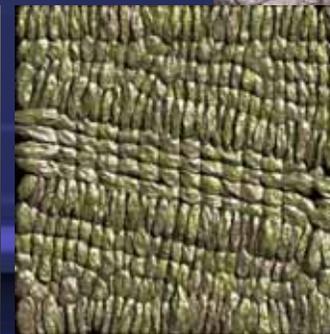
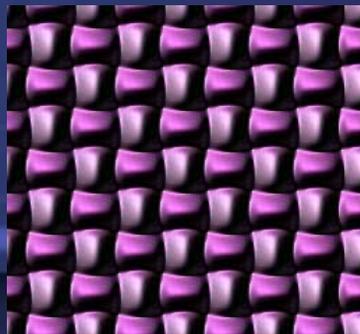
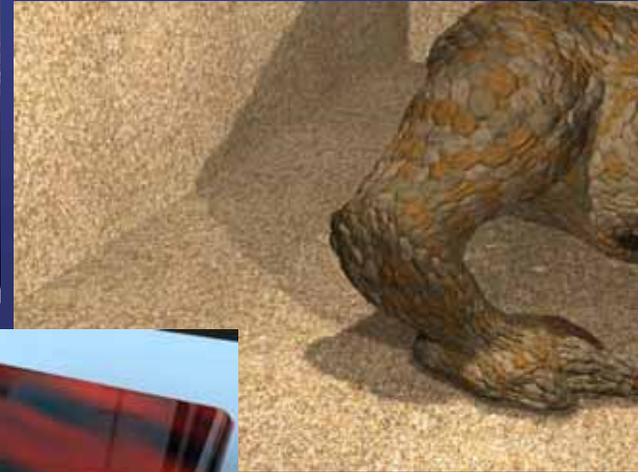
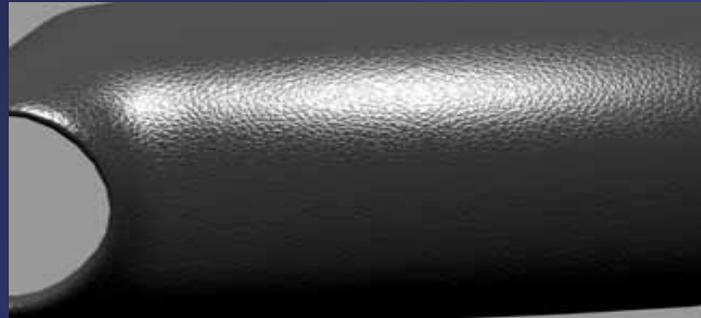


My Second Encounter...



Star Trek II: Wrath of Kahn (1982)

My Procedural Textures



プロシージャル手法について

プロシージャル手法とは

- 形状や模様をアルゴリズムでデザイン
 - 細部を圧縮
 - 必要な時に必要なだけ解凍
- パラメータの変更で、多くのバリエーション
- 形状や模様を手間暇かけずに自由に生成
- 詳細度(LoD)を制御可能

プロシージャル手法の欠点

- 設計が難しい
- 試行が大変
 - その都度, 生成
- 時間がかかる

プロシージャル手法の歩み

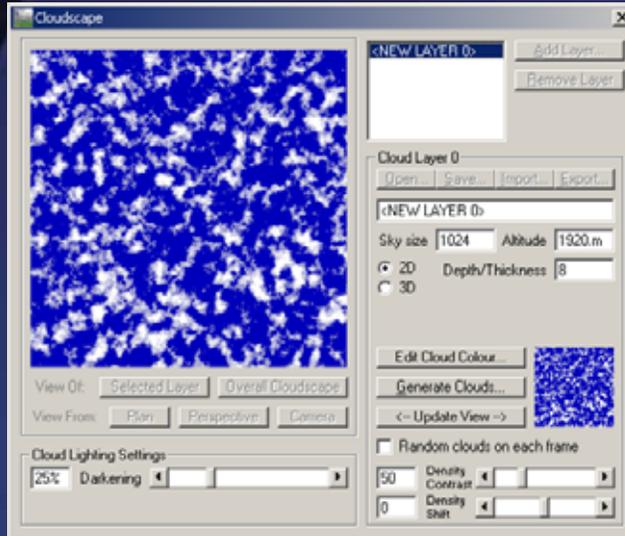
- 初期のモデリング, 表面の模様付加に活用
- 1980年代後半から, 盛んに用いられる
 - 特にテクスチャ生成に威力を発揮
 - 大理石, 木目, 岩などの自然物
 - さらに, 水, 煙, 炎など(こちらは物理モデルへ発展)
- RenderMan Shading Language (Pixar 1989)
により, 一気に拡大

プロシージャル

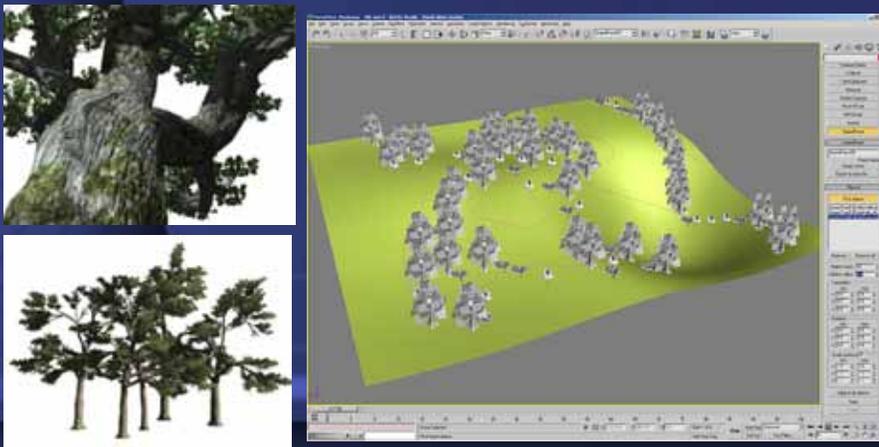
- 画面の高精細化
 - コンテンツ制作の手間
 - どのくらいの粒度で作ればいいのか.
 - すべて貯め置きするのか.
- デザイナ/アーティストの立ち位置
 - 絵が描ける
 - 絵を描く“中の人”を作れる

代表的なソフト

Terragen

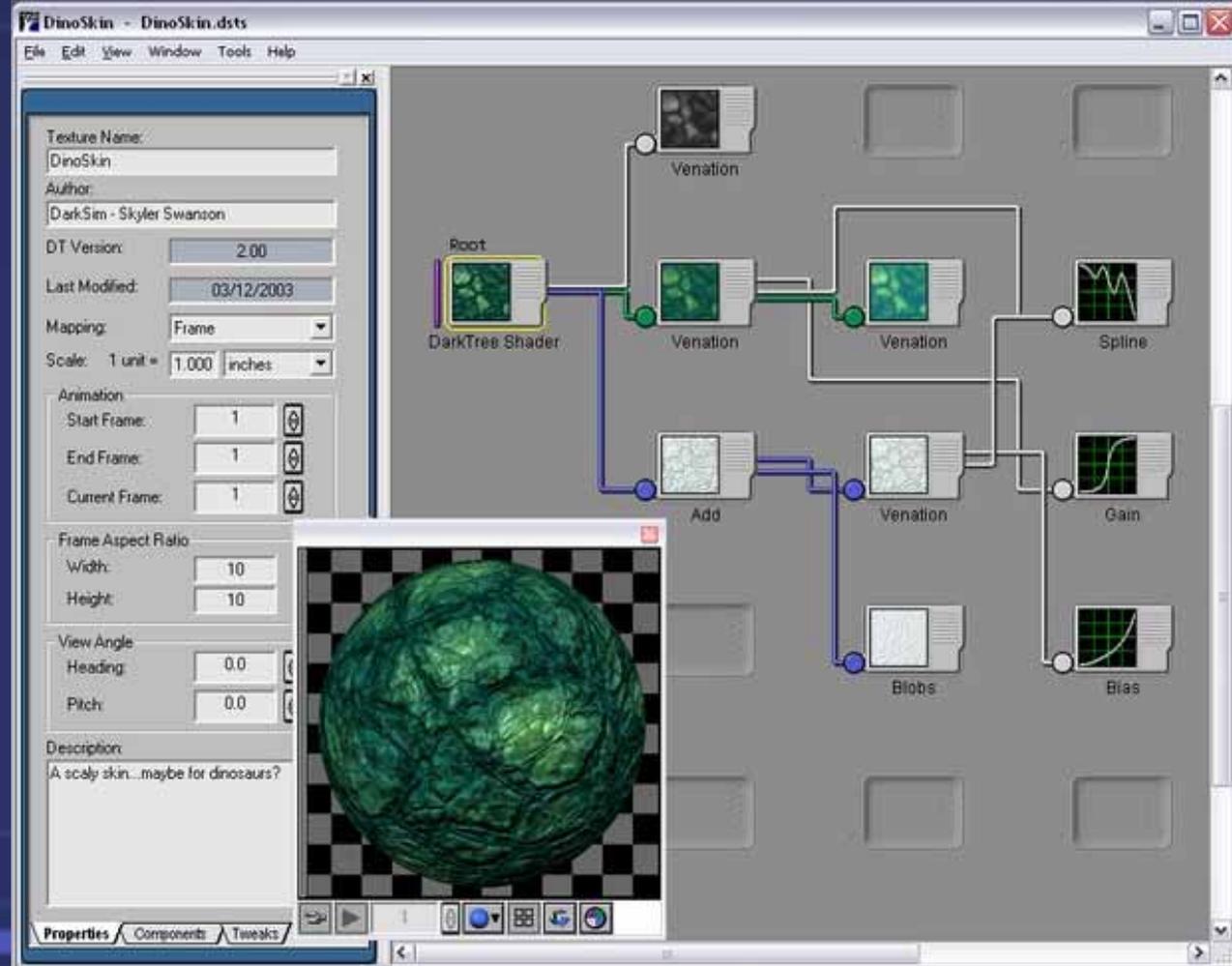


SpeedTree



代表的なソフト

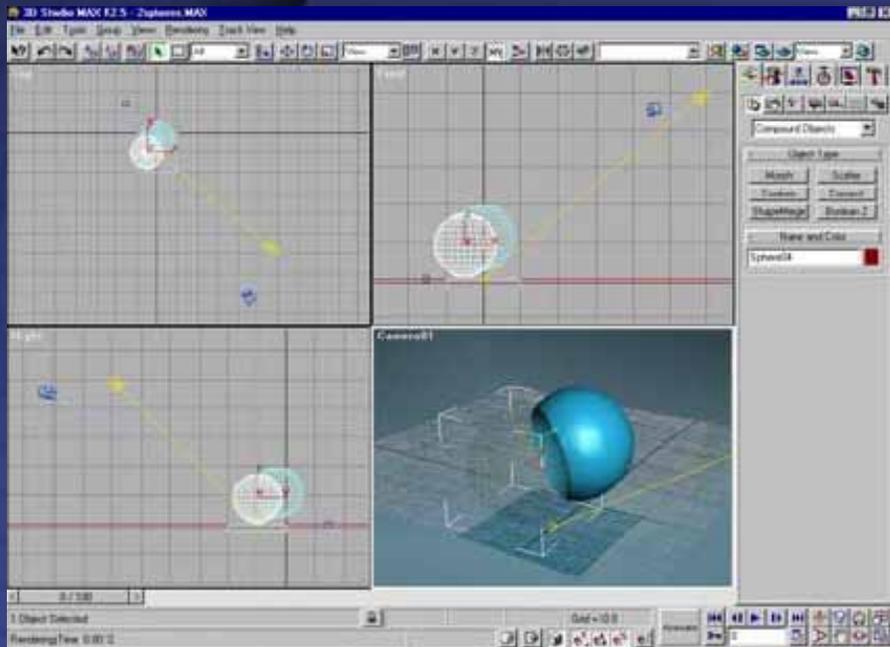
DarkTree



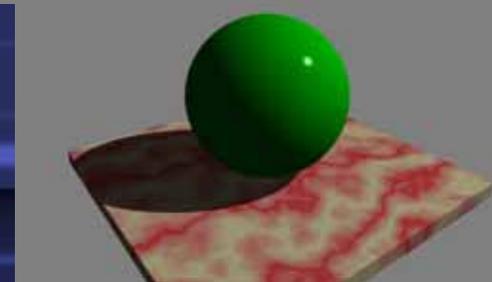
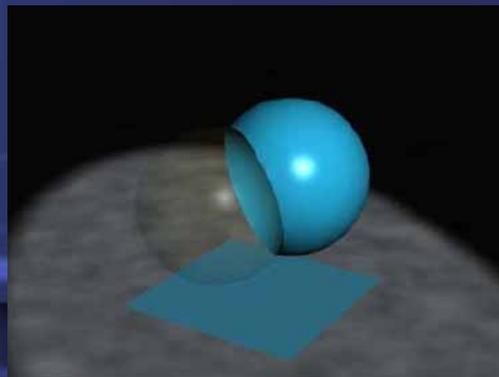
<http://www.darksim.com/>



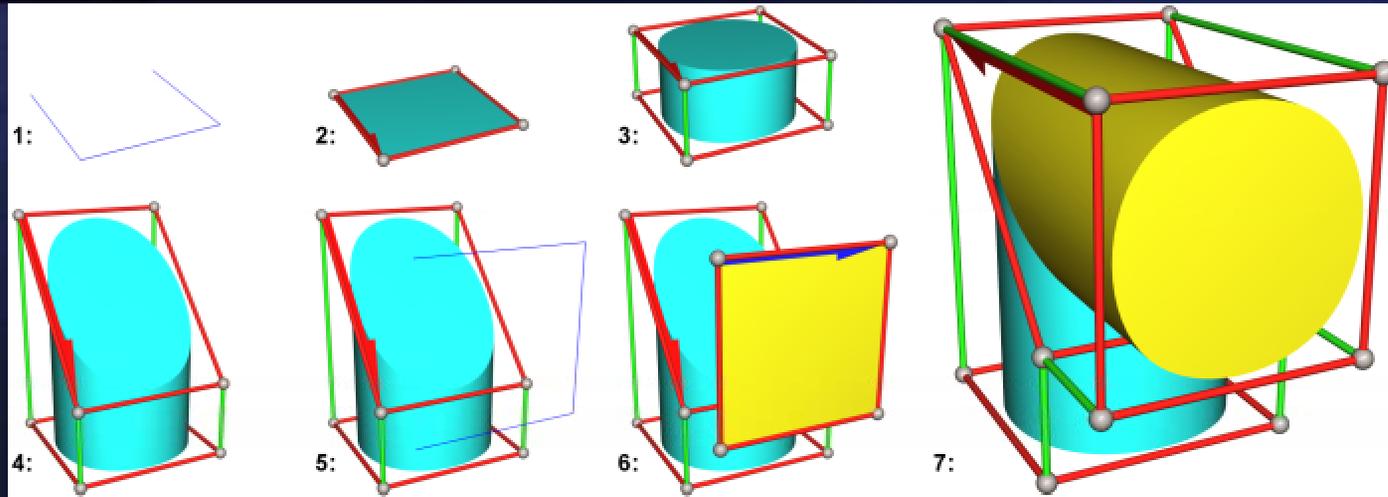
Interactive vs Procedural (script)



```
1: #include "colors.inc"
2: #include "textures.inc"
3:
4: camera { location < -4.0, 2, -3.0 >
5:     look_at < 0.0, 0.0, 0.0 >
6:     angle 60
7: }
8:
9: light_source { < 20, 30, -25 > color White }
10:
11: sphere { < 0, 0, 0 >, 1
12:     pigment { Green }
13:     finish { Shiny }
14: }
15:
16: box { < -2, -1.2, -2 >, < 2, -1, 2 >
17:     texture { Red_Marble }
18:     finish { ambient 0.2
19:         diffuse 0.8
20:         phong 1
21:     }
22: }
23:
24: background { color Gray50 }
```



GML (Generative Modeling Language)



1: (0,0,-2) (1,1,0) 2 quad
 2: /cyan setcurrentmaterial
 5 poly2doubleface
 3: (0,1,1) extrude
 4: (0,0,1) (1,0,1) normalize
 0 project_ringplane

5: (2,0,0) (0,1,-1) 2 quad
 6: /yellow setcurrentmaterial
 5 poly2doubleface
 7: 0 bridgerings

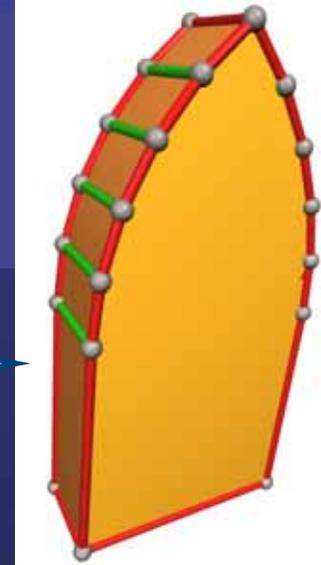
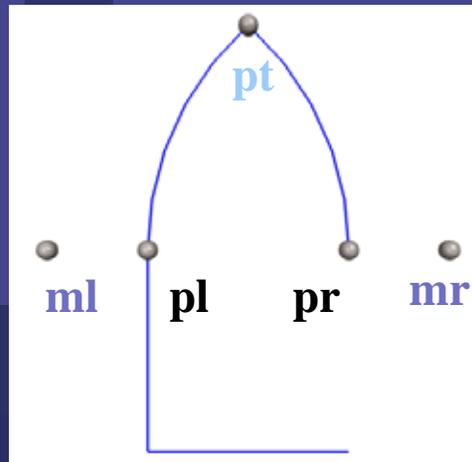
<http://www.generative-modeling.org/>

GML: 生成例

```
[ :pr :ml :pt ] (0,0,1) 5 1 circleseg
[ :pt :mr :pl ] (0,0,1) 5 1 circleseg
arrayappend
```

```
[ (-0.5,-1,0) dup (0.5,-1,0) dup ]
arrayappend
```

```
5 poly2doubleface
(0,0.4,5) extrude
```

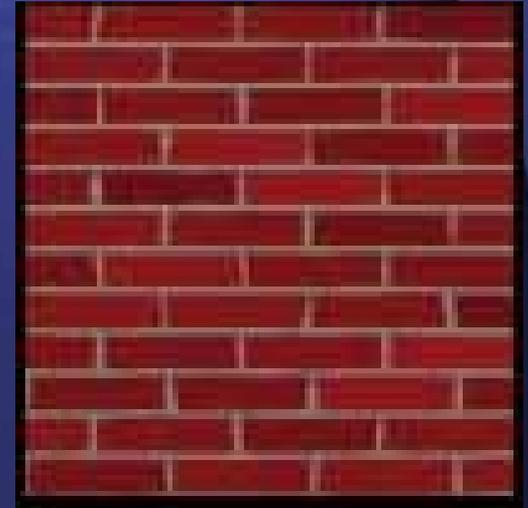


<http://www.generative-modeling.org/>

RenderMan Shading Language

```
surface
brick ( float Ka = 1, Kd = 1;
        color brickcolor = color "rgb" (.6,.1,.1);
        color mortarcolor = color "rgb" (.6,.6,.6);
        float raggedamp = 0.04, raggedfreq = 12;
        float jagged = 0.006, brickvary = 0.3;
        float brickwidth = .28, brickheight = .07;
        float mortarthickness = .014; float rowvary = .5;
        float pitting = 0.01; float pockfrequency = 10,
        groovedepth = 0.01; )

{
#define MINFILTERWIDTH 1.0e-7
#define sqr(x) ((x)*(x))
    color bcolor, Ct;
    normal Nf; float sbrick, tbrick, w, h;
    float ss, tt; float swidth, twidth;
    .....
```

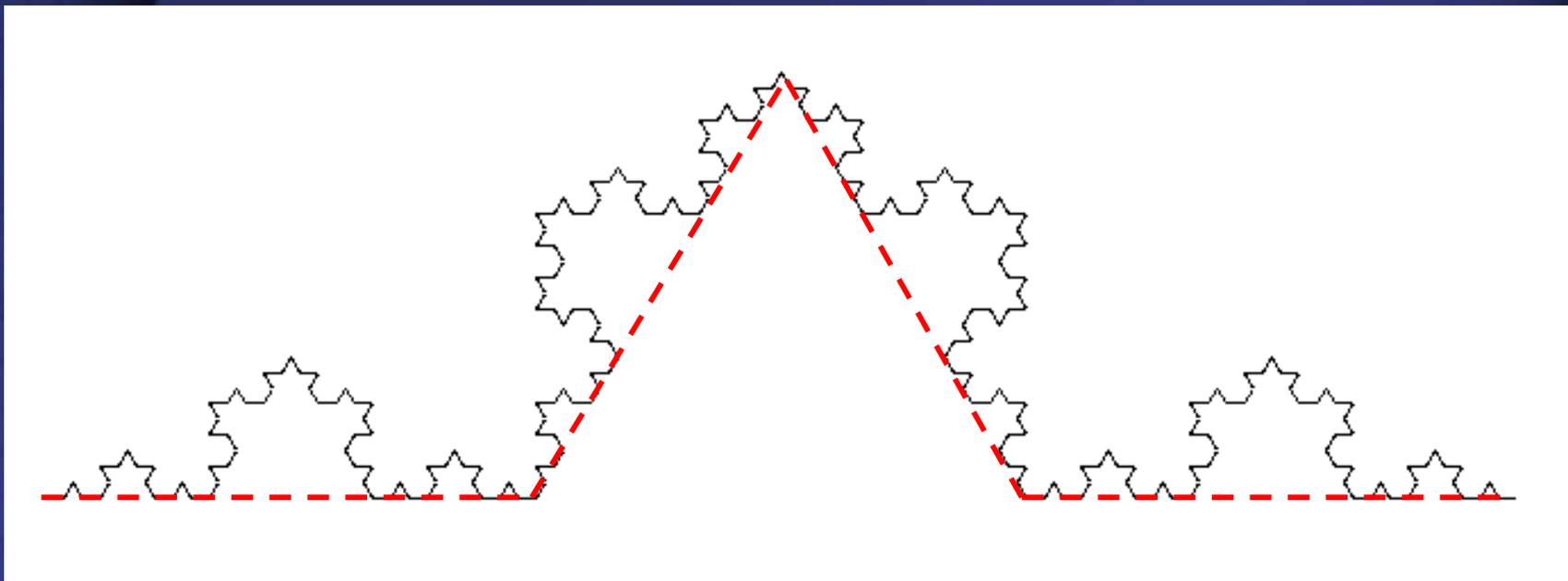


<http://www.renderman.org/>

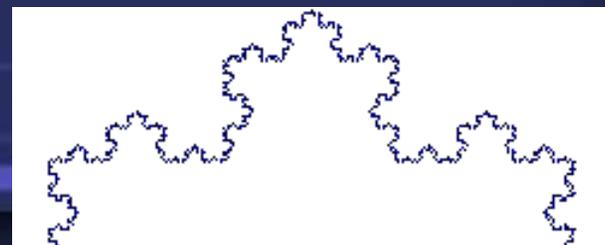
フラクタル (fractal)

フラクタル図形とは？

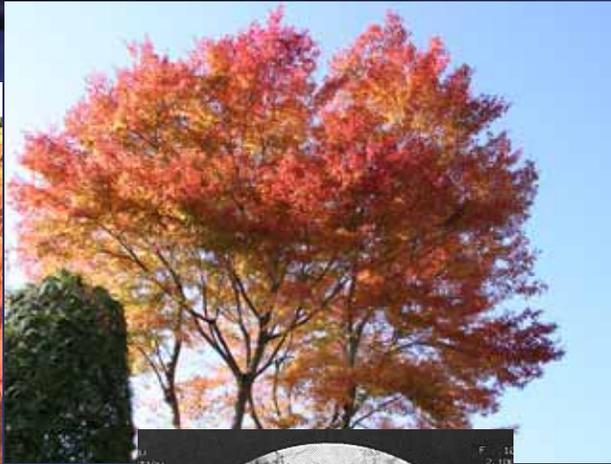
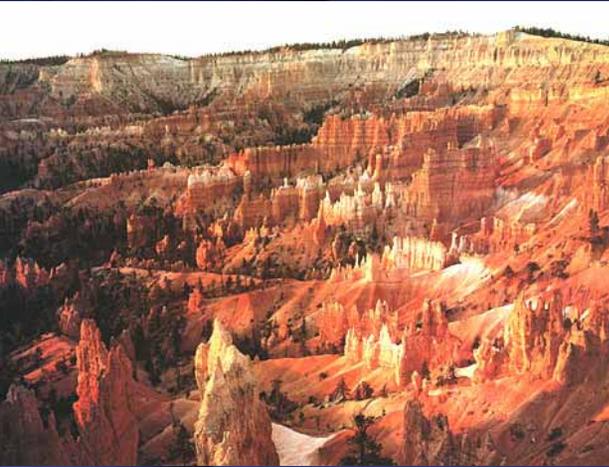
自己相似性



コッホ曲線

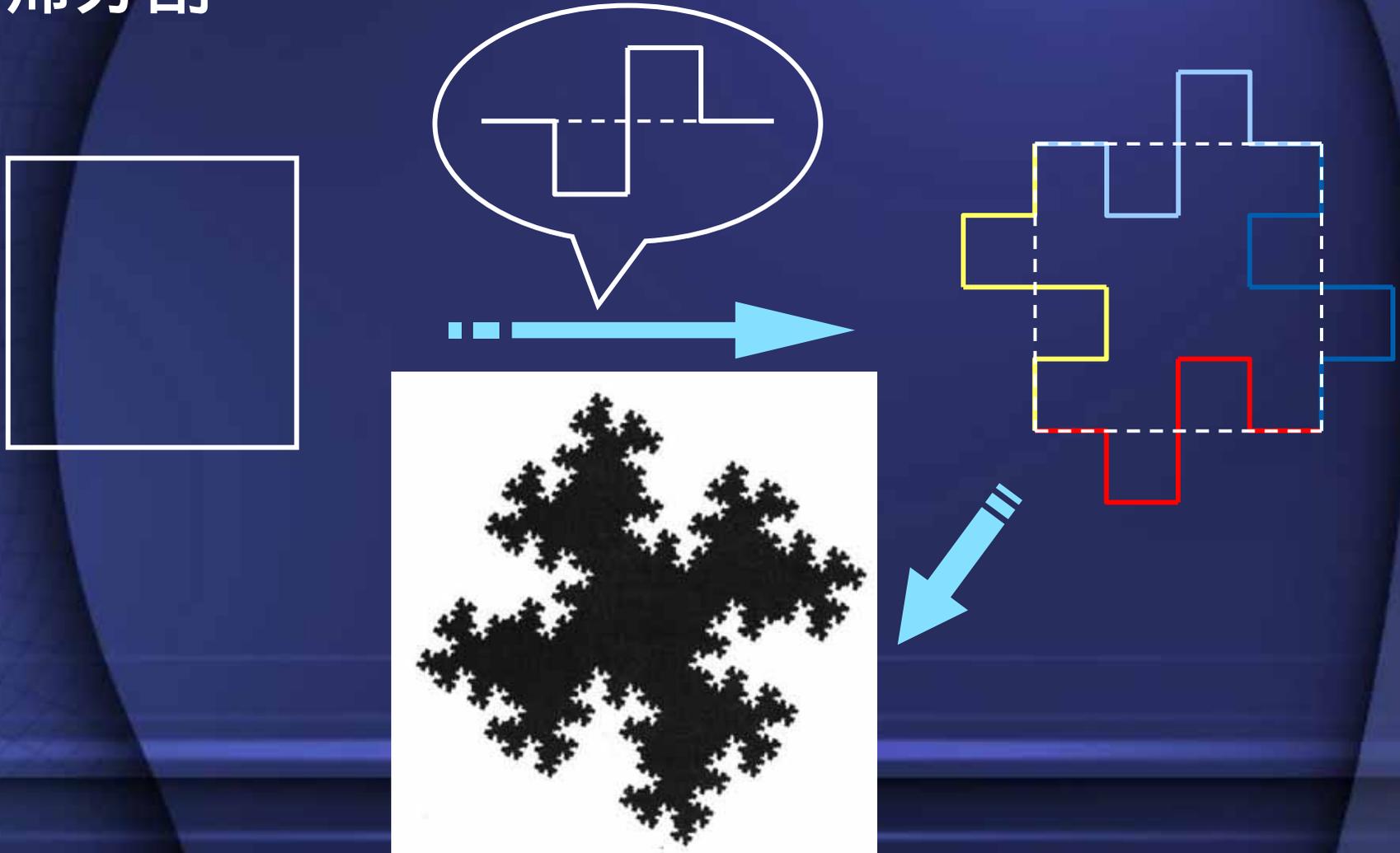


自然の中のフラクタル

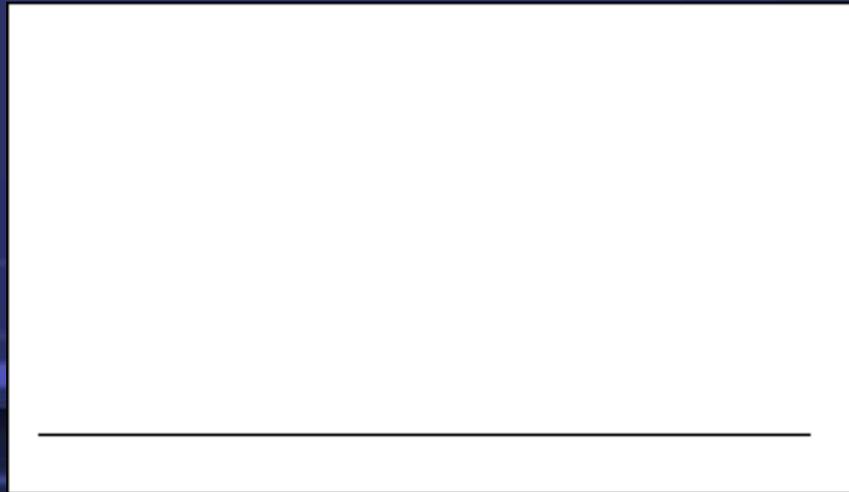
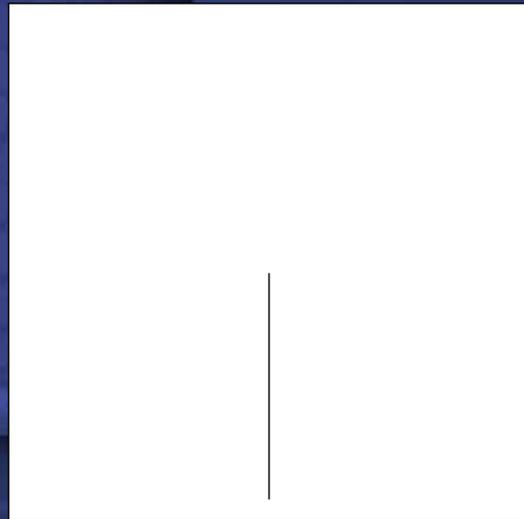
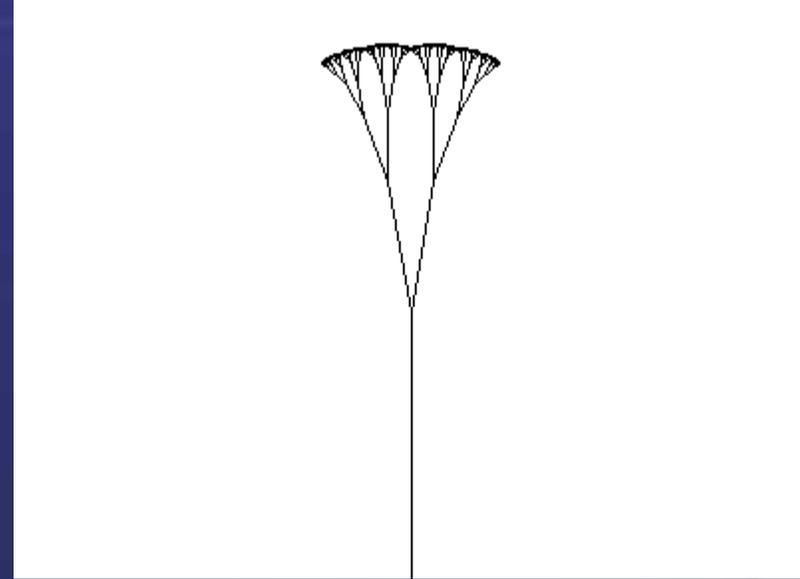
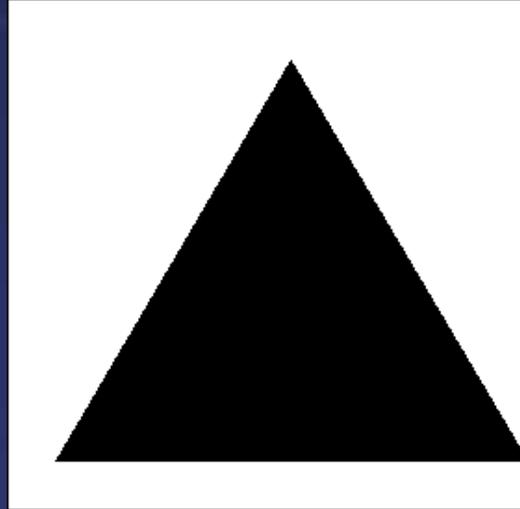
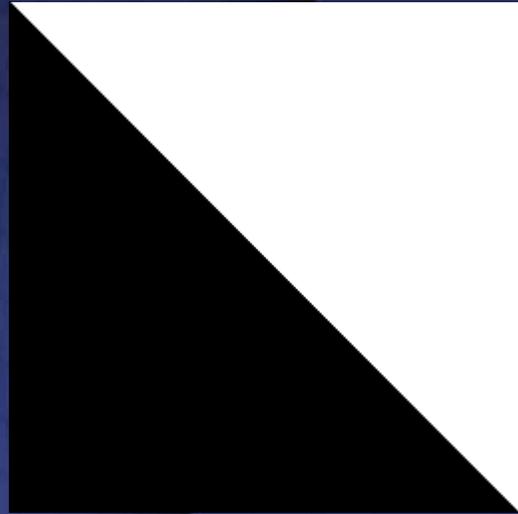


フラクタル図形の生成法

再帰分割



フラクタル図形の生成法

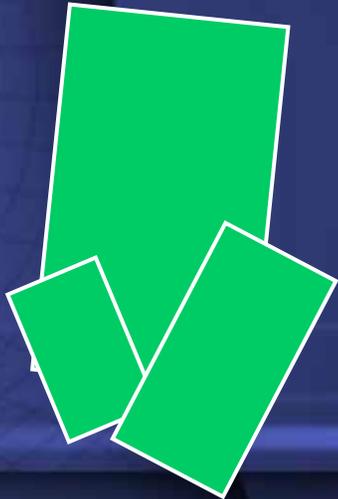
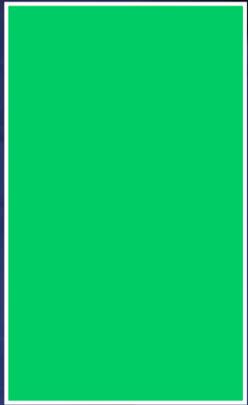


フラクタル図形の生成法

反復コピー



フラクタル図形の生成法

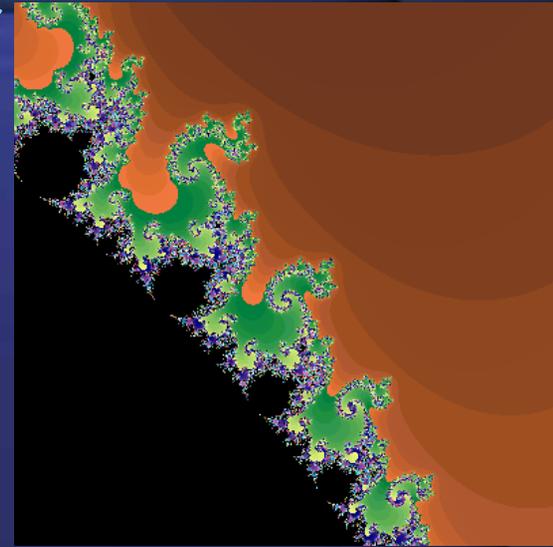
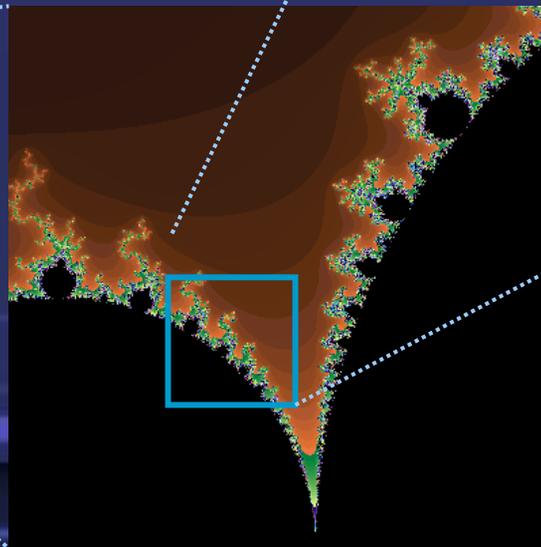
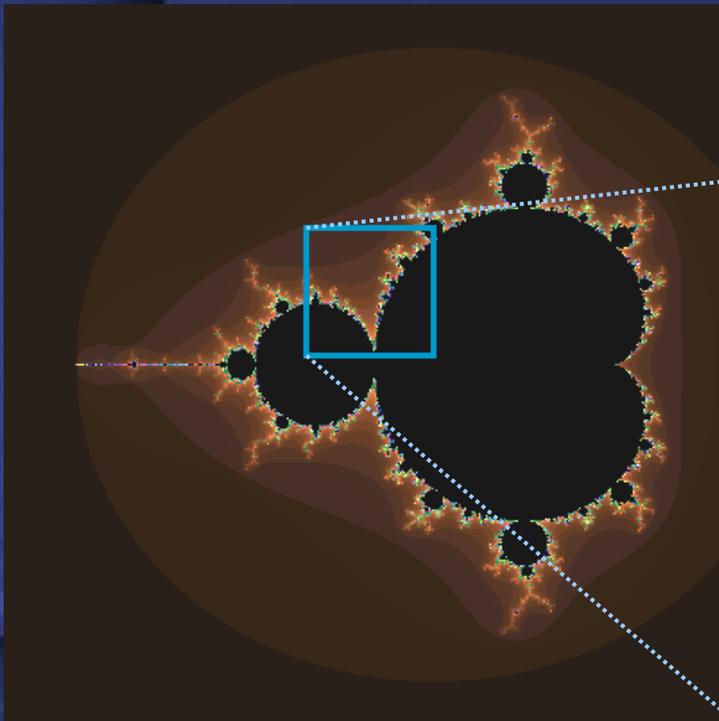


<http://www.home.aone.net.au/~byzantium/ferns/index.html>

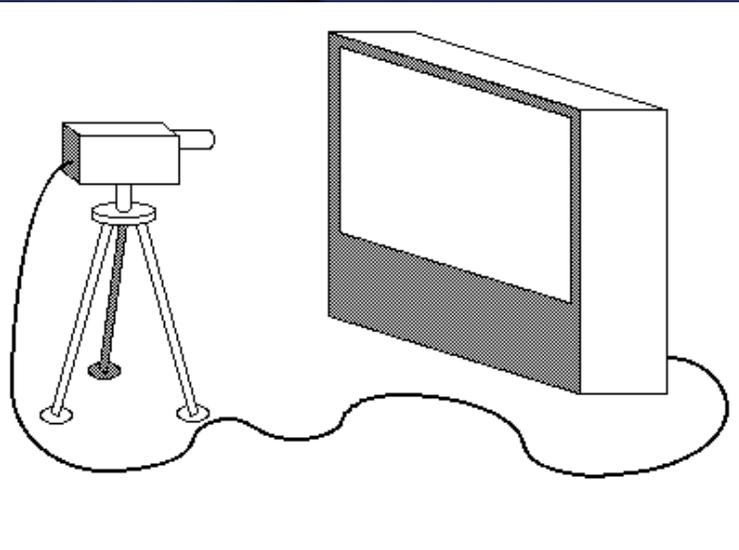
フラクタル図形の生成法

式の反復

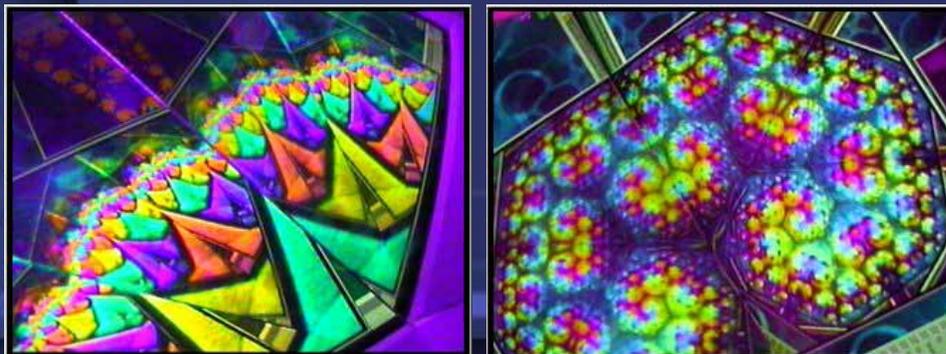
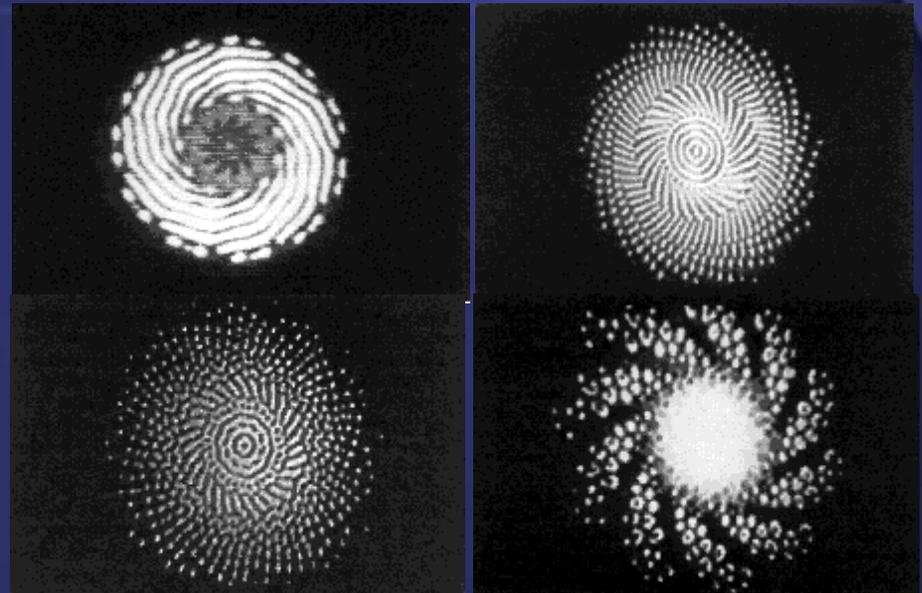
$$z = z^2 + c$$



Video Feedback



<http://classes.yale.edu/fractals/>



David Hagar

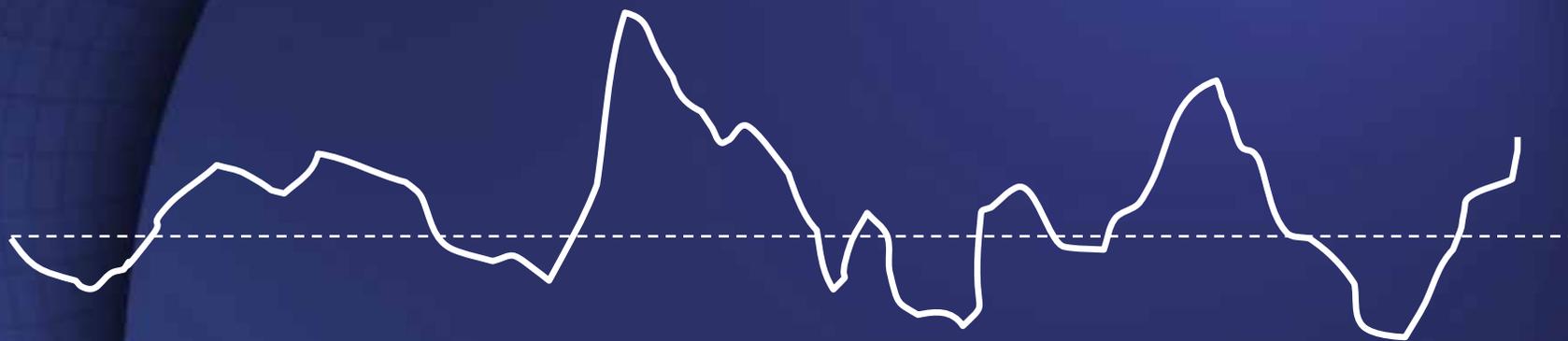
<http://www.geocities.com/CapeCanaveral/Campus/8446/index.html>

山岳の表現

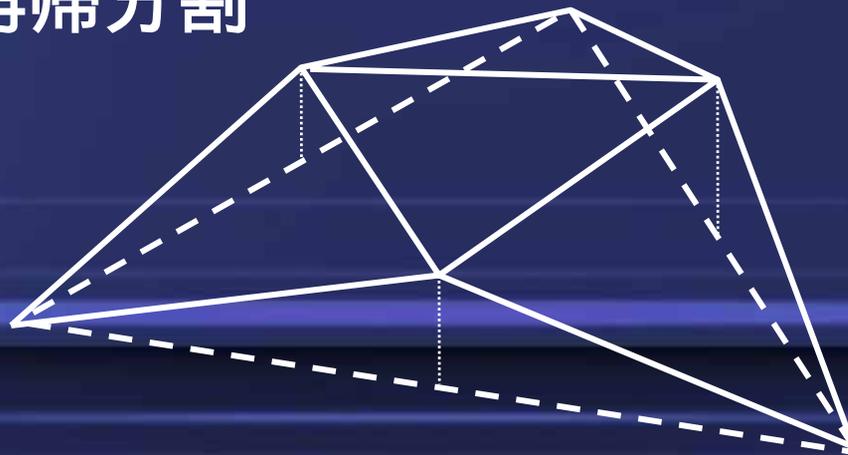


山岳形状の生成

- ・ fBm (fractional Brownian motion) で高さを与える



- ・ 幾何形状を再帰分割



fBm(fractional Brownian Motion)

ブラウン運動

移動: $B(t) - B(t_0) \sim |t - t_0|^{1/2}$

期待値: $E(B(t) - B(t_0)) = 0$

分散: $V(B(t) - B(t_0)) = 2D|t - t_0|$

fBm

移動: $B_f(t) - B_f(t_0) \sim |t - t_0|^H$

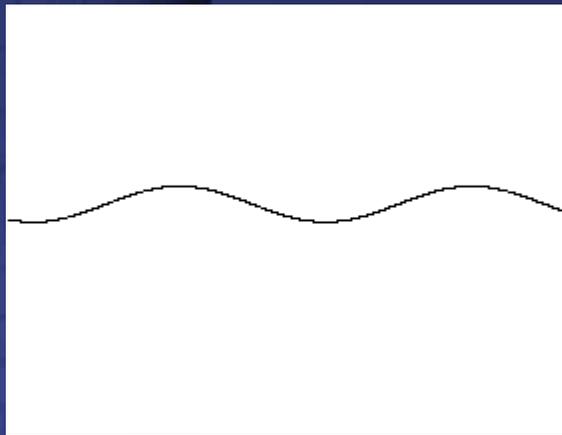
期待値: $E(B_f(t) - B_f(t_0)) = 0$

分散: $V(B_f(t) - B_f(t_0)) = 2D|t - t_0|^{2H}$

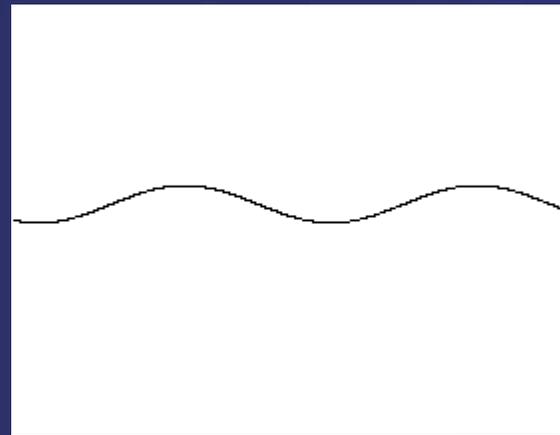
fBm(fractional Brownian Motion)

$$E[B_H(t) B_H(s)] = \frac{1}{2}(|t|^{2H} + |s|^{2H} - |t-s|^{2H})$$

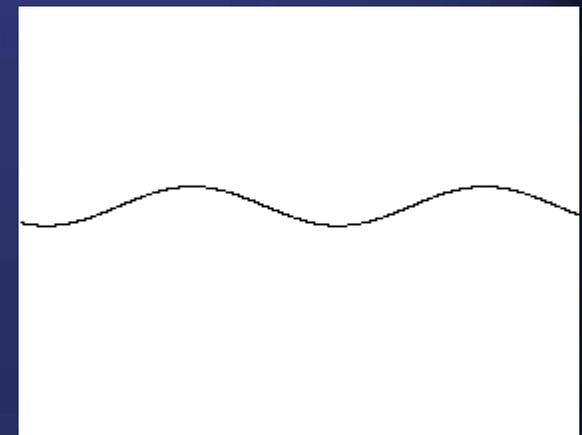
相関関数



$H > \frac{1}{2}$
 (正の相関,
 同方向)

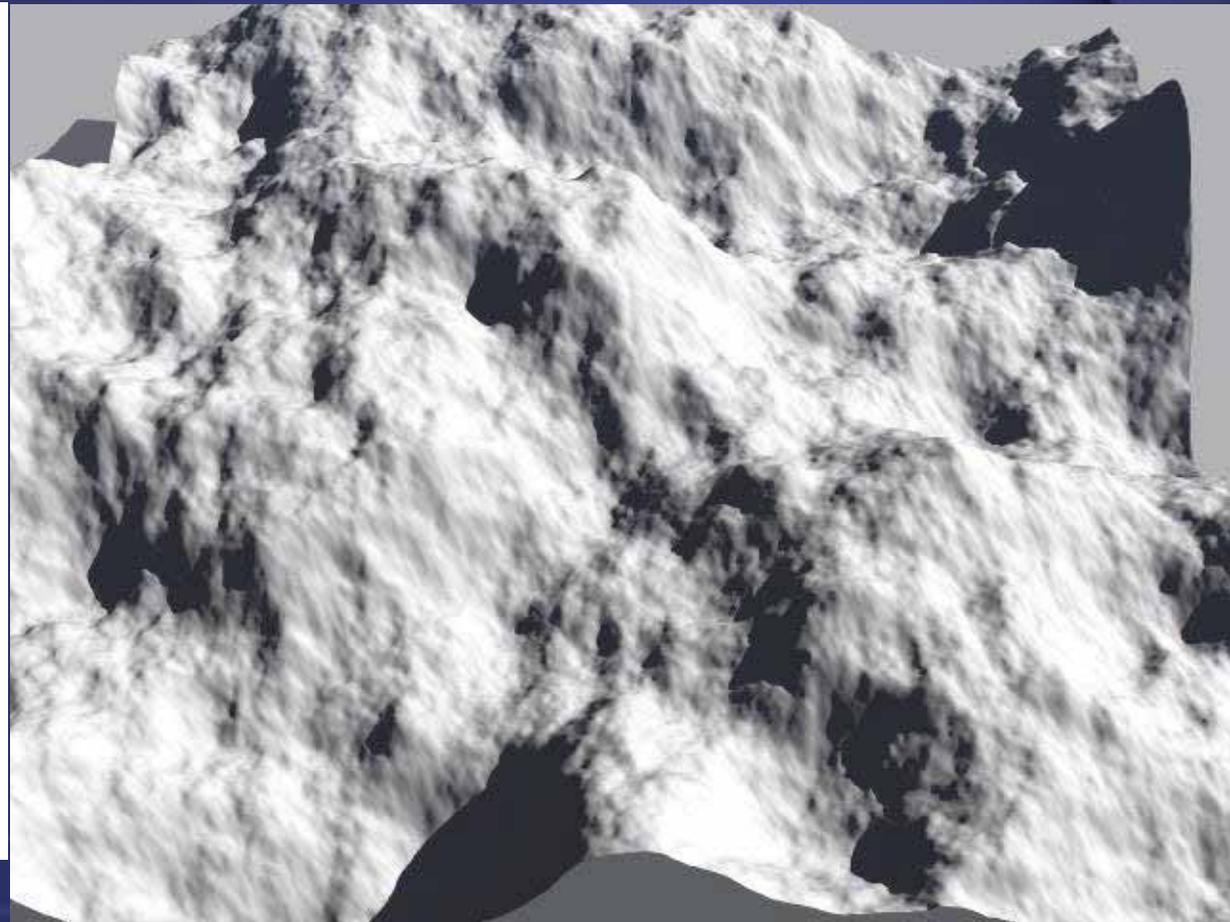
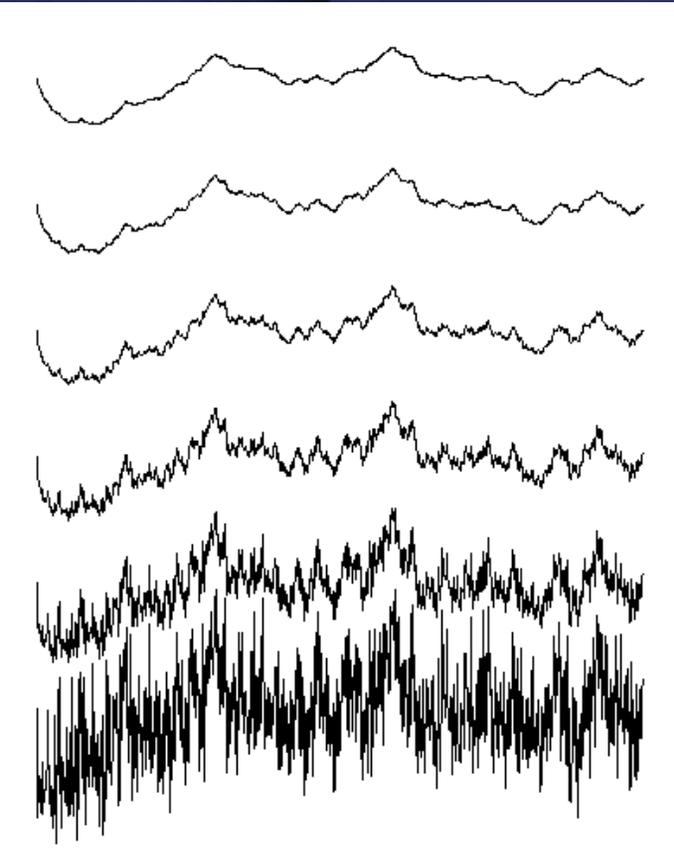


$H = \frac{1}{2}$
 (ブラウン運動)

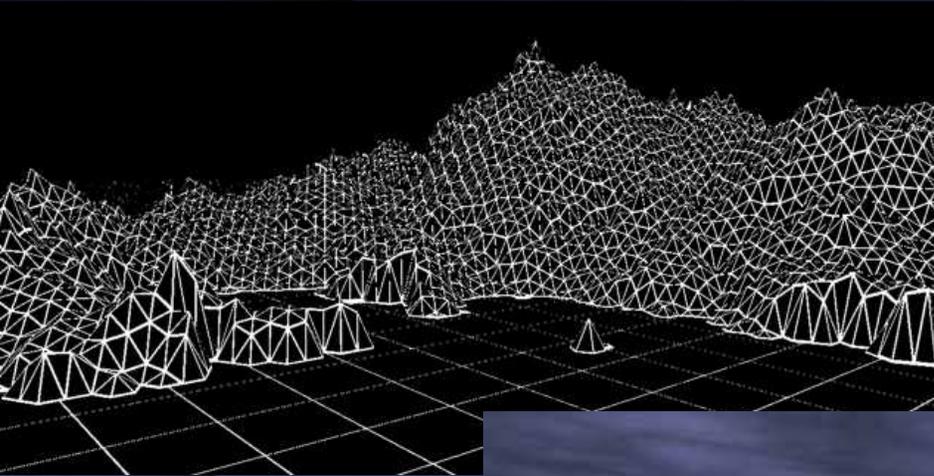


$H < \frac{1}{2}$
 (負の相関,
 逆方向)

fBm(fractional Brownian Motion)



山岳形状の生成例



<http://www.kenmusgrave.com/>



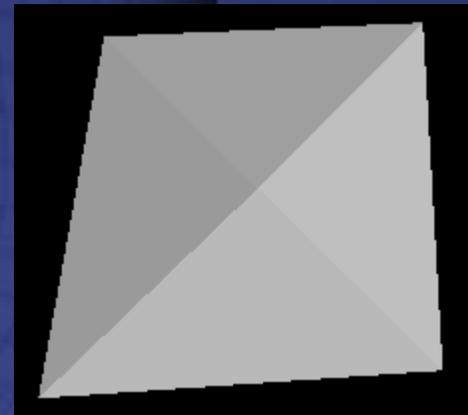
山岳形状の生成例



<http://www.kenmusgrave.com/>

山岳形状の生成

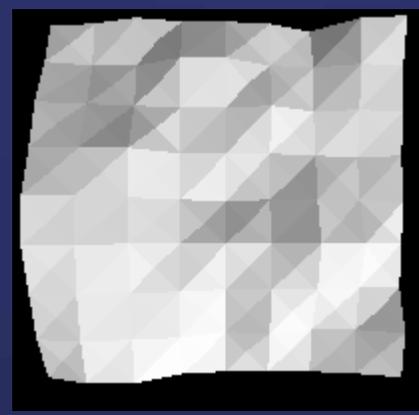
初期形状(三角形や矩形)を再帰的に分割し, 細かな凹凸付けをする



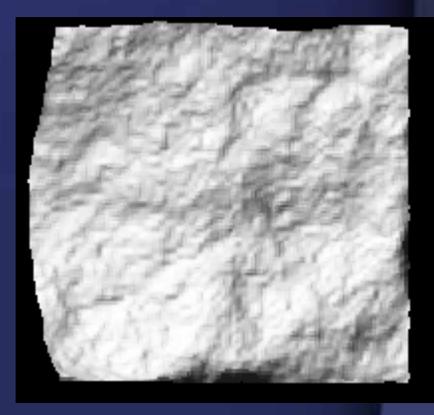
初期形状



1 step



3 steps



7 steps

山岳形状の生成例



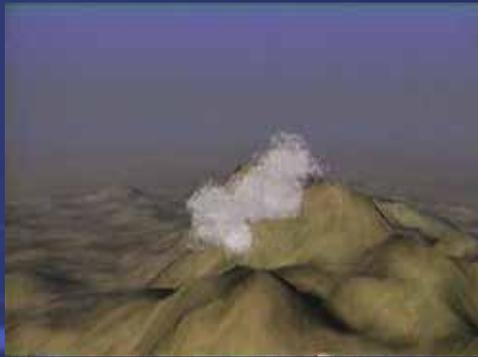
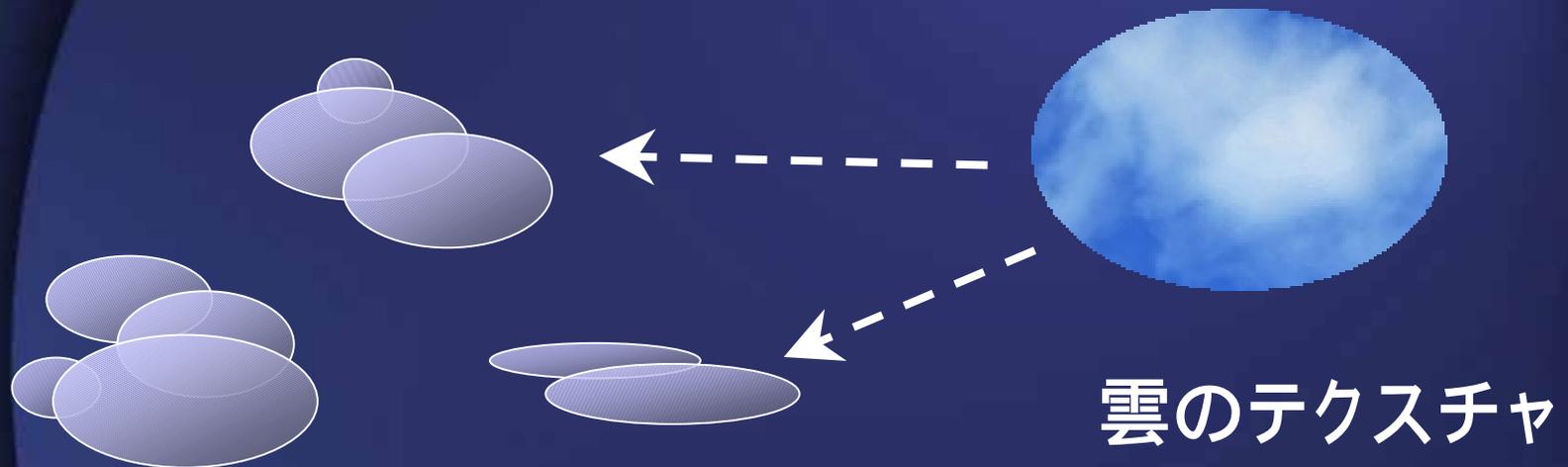
雲の表現

15:05?



雲の表現 - Gardnerの手法

2次曲面へのテクスチャマッピング

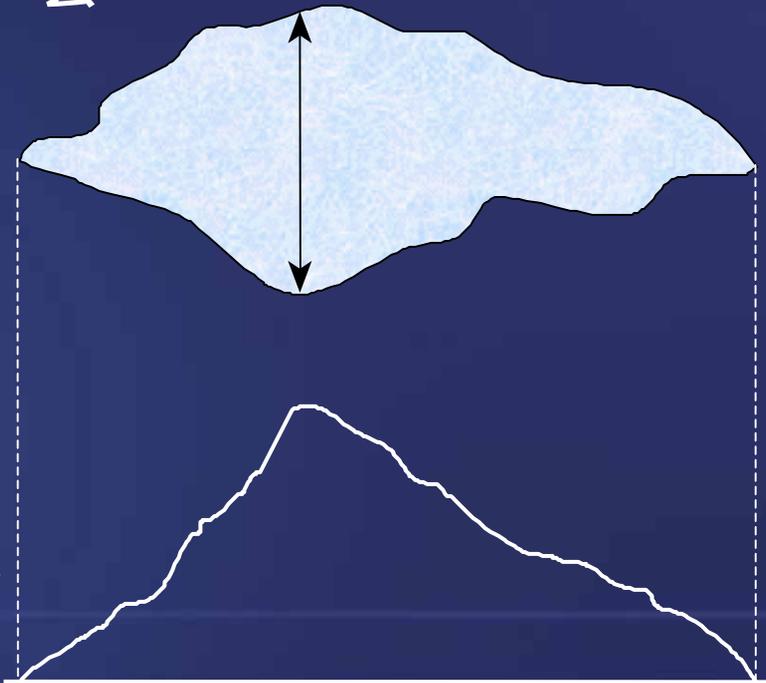


G.Y.Gardner, "Send in the clouds," SIGGRAPH1990

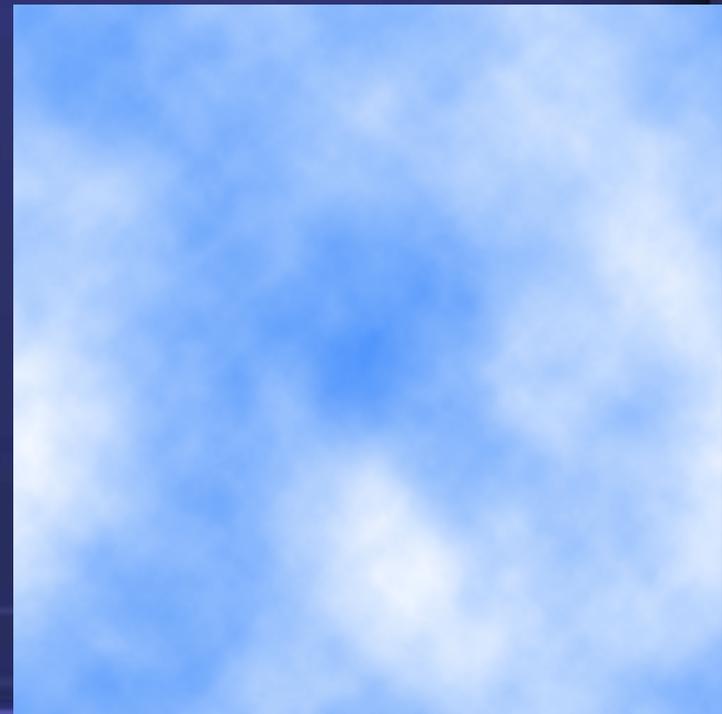
雲の表現：フラクタル

雲の厚みをフラクタルノイズ関数で生成

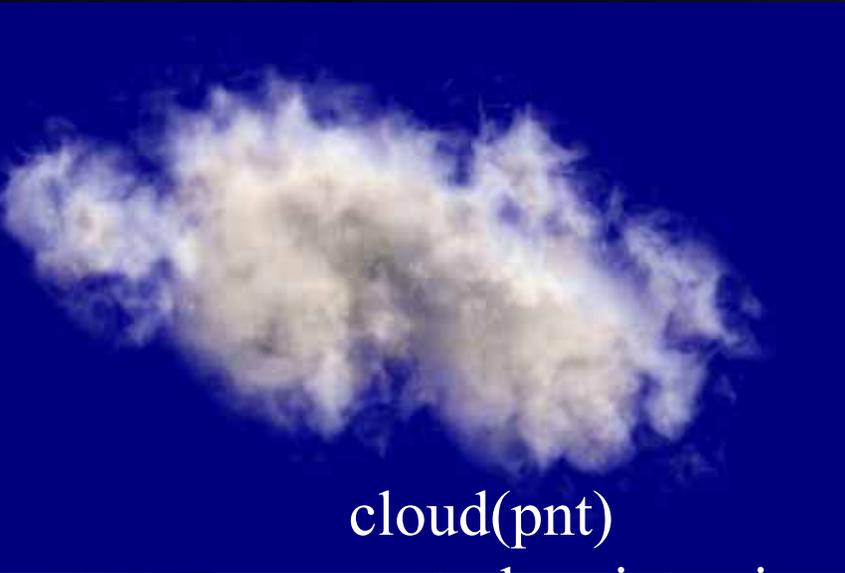
雲



雲の厚み



Volumetric Cloud



陰関数とノイズ関数の 組み合わせによる表現

David S. Ebert (1997)

```
cloud(pnt)
```

```
  perturb point using noise and turbulence
```

```
  density1 = implicit_function(perturbed_point)
```

```
  density2 = turbulence(pnt)
```

```
  blend = blend% * density1 + (1 - blend%) * density2
```

```
  returned density = power(blend*max_density,  
                           exponent)
```

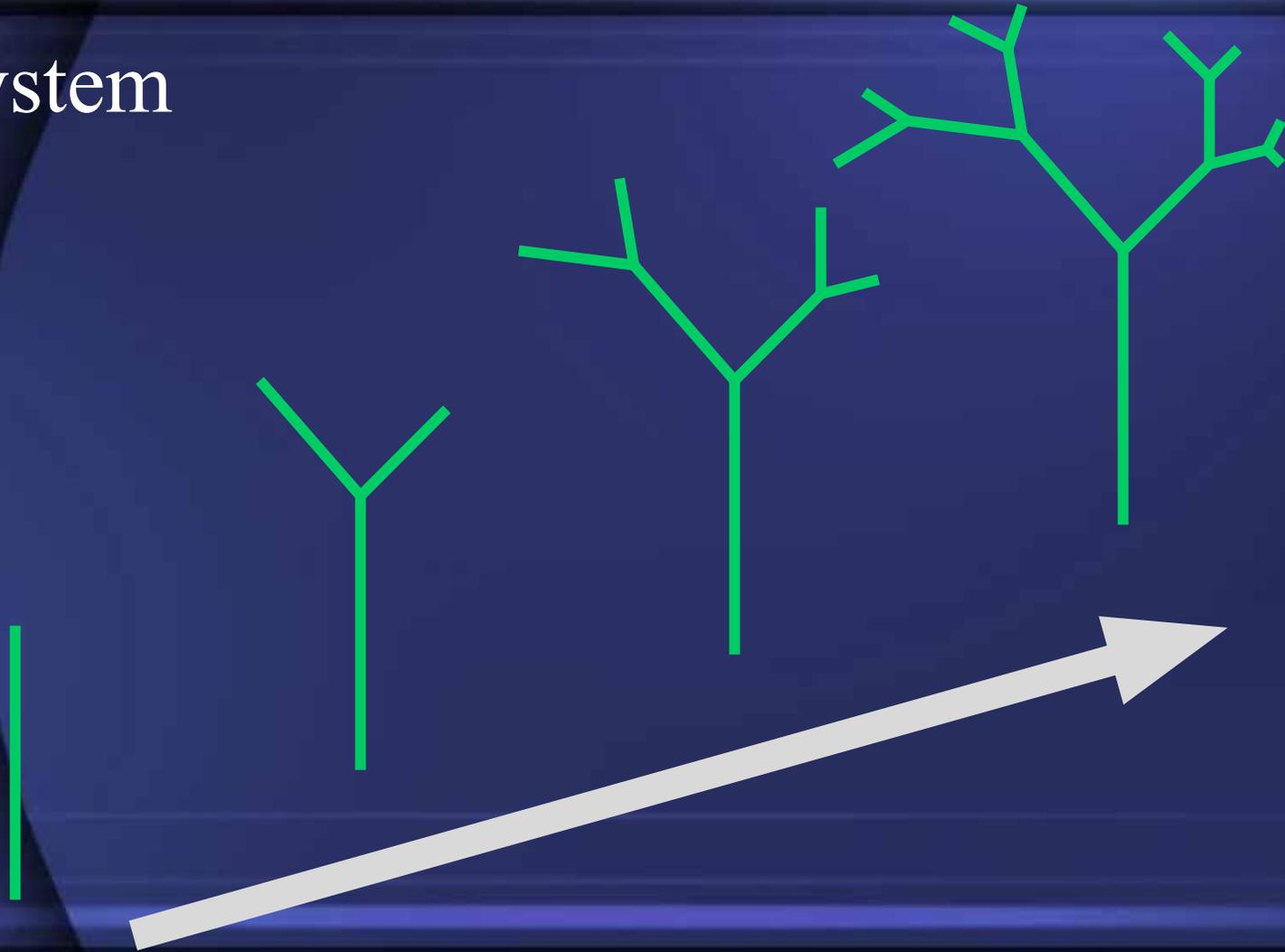
```
end cloud
```

植物の表現



植物の生成法

L-System



Lindenmayer-Systems (L-System)

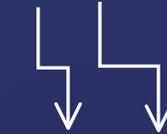
- 植物学者 Aristid Lindenmayerによる
- 文字列 (シンボル) の変換メカニズム
 - 初期値
 - 変換ルール
- 簡単な例
 - 初期値 $\omega : A$
 - 変換ルール $\rho_0 : A \rightarrow xB$
 $\rho_1 : B \rightarrow yA$

L-Systemsの簡単な例

- 初期値 $\omega : AB$
- 変換ルール $\rho_0 : A \rightarrow xB$
 $\rho_1 : B \rightarrow yA$

初期値

AB



1回目の変換

$xByA$



2回目の変換

$xyAyxB$

L-Systemsとコンテキスト

コンテキスト(文脈)

文章の前後関係

変換するシンボルの前後(左右)の関係

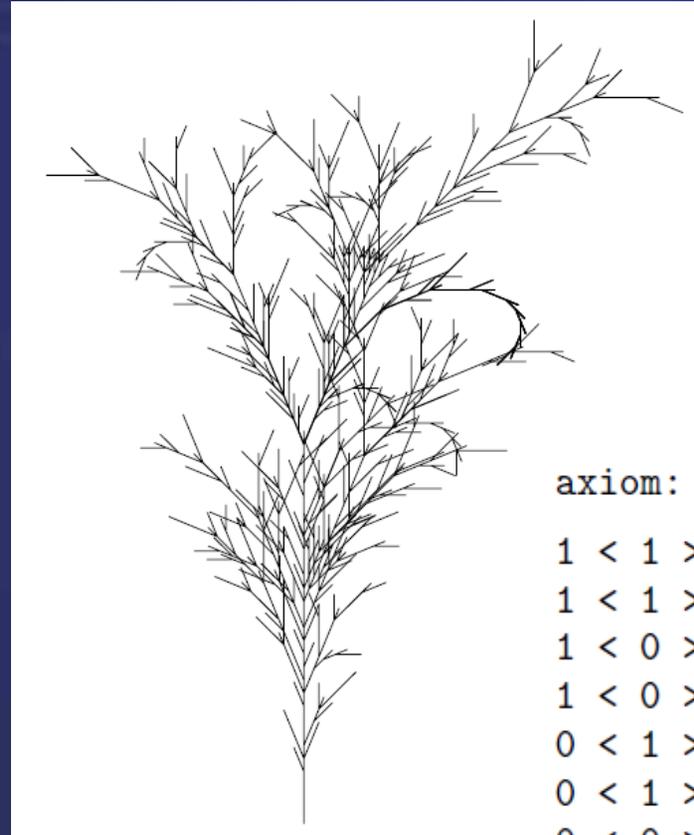
L-systemでの文脈依存(context-sensitive)のルール

$$\rho_x : \textit{left-context} < \textit{pattern} > \textit{right-context} \\ \rightarrow \textit{replacement}$$

pattern の左側に *left-context* 右側に *right-context* がある場合, *pattern* を *replacement* に置き換える

文脈依存 L-Systems

- $\omega : xA$
- $\rho_0 : A > y \rightarrow Bx$
- $\rho_1 : B \rightarrow A$
- $\rho_2 : x < A \rightarrow By$



- 1: xA
- 2: xBy
- 3: xAy
- 4: $xBxy$
- 5: $xAxy$
- 6: $xByxy$

axiom: F1F1F1

```

1 < 1 > 1 --> 0
1 < 1 > 0 --> 1
1 < 0 > 1 --> 1F1
1 < 0 > 0 --> 0
0 < 1 > 1 --> 1
0 < 1 > 0 --> 1
0 < 0 > 1 --> 1[-F1F1]
0 < 0 > 0 --> 0
* < - > * --> +
* < + > * --> -
    
```

...

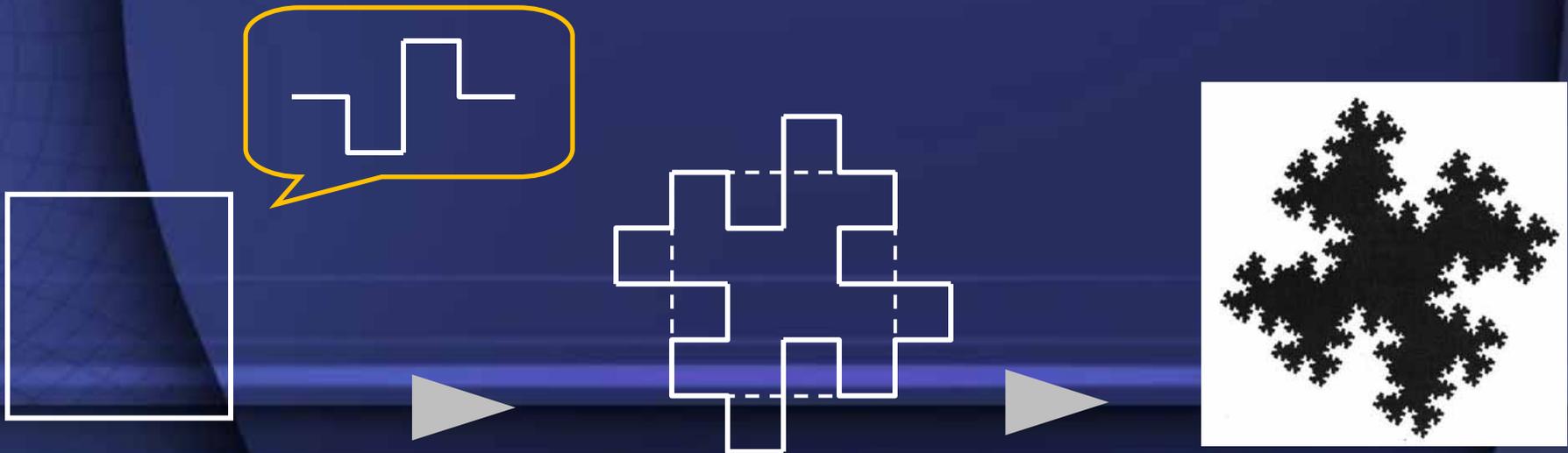
<http://algorithmicbotany.org/>

L-Systemによる形状の描画

タートルグラフィックス

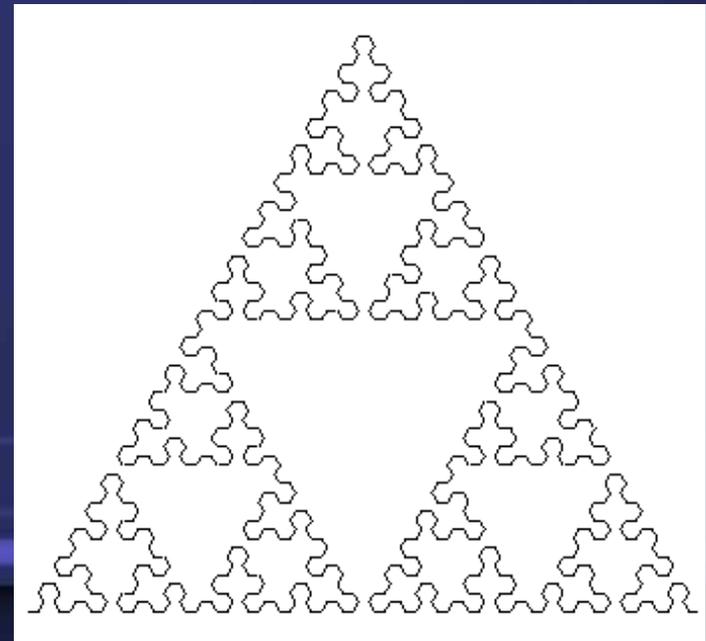
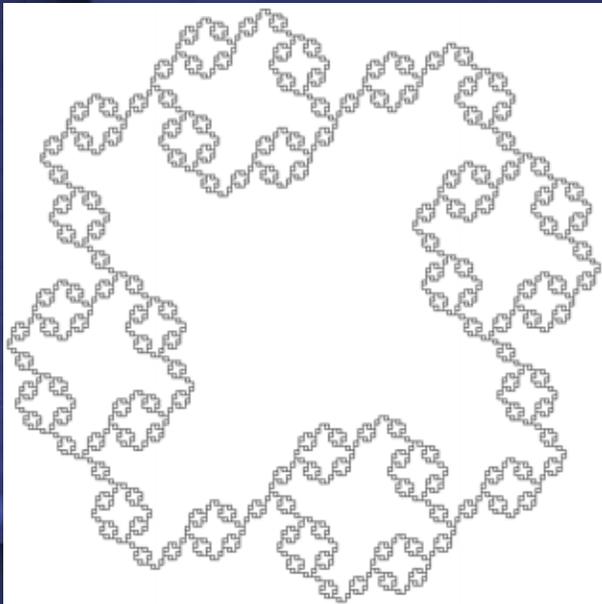
- F : 線を描きながら前進
- + : 時計回りに回転
- : 反時計回りに回転

生成ルール: $F \rightarrow F+F-F-F+FF+F+F-F$



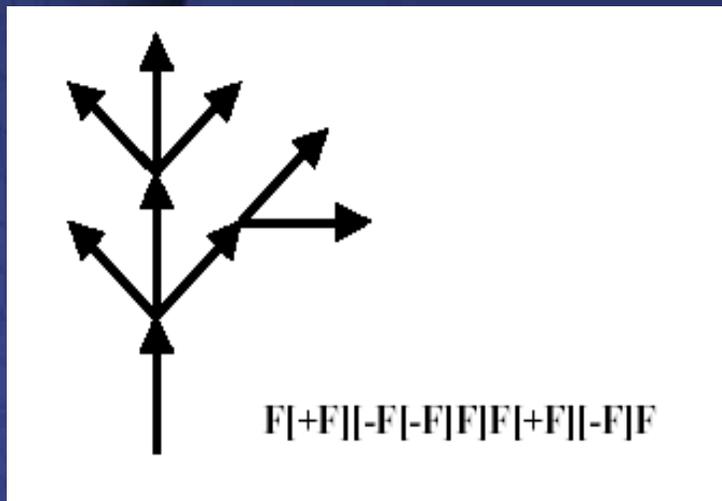
L-Systemによる図形の描画

- $\omega : F-F-F-F$
- $\rho_0 : F \rightarrow FF-F-F-F-F-F+F$
- $\delta = 90^\circ$, 4世代目
- $\omega : RF$
- $\rho_0 : L \rightarrow RF+LF+R$
- $\rho_1 : R \rightarrow LF-RF-L$
- $\delta = 60^\circ$, 6 世代目

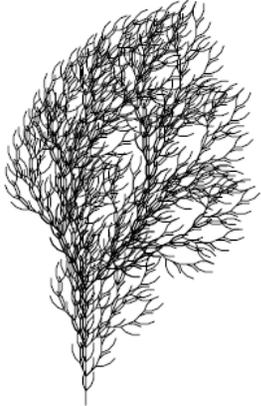
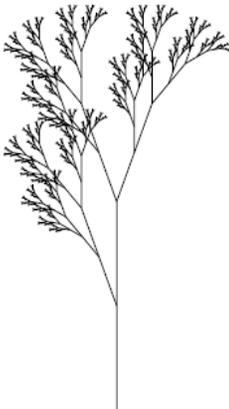
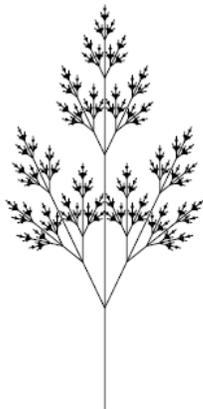


L-Systemによる分岐の表現

[: 状況をスタックにpush
] : 状況をスタックからpop



<http://algorithmicbotany.org/>

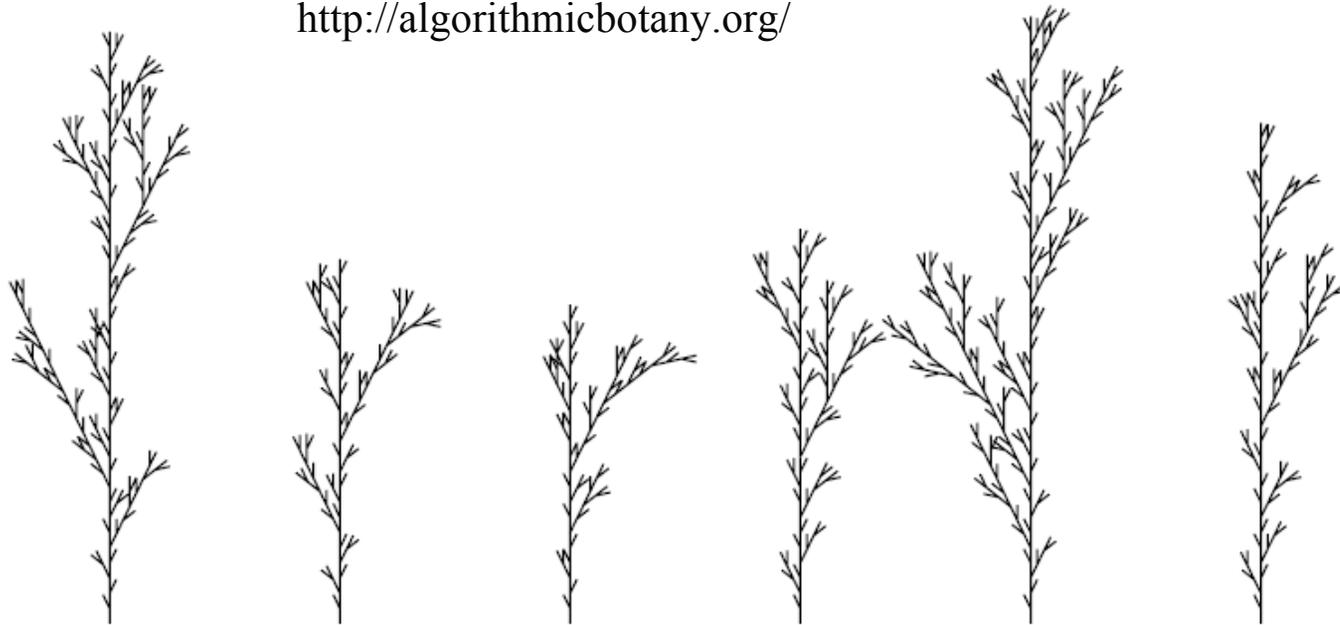
		
$F \rightarrow F[+F]F[-F]F$	$F \rightarrow F[+F]F[-F][F]$	$F \rightarrow FF[-F+F+F][+F-F-F]$
		
$X \rightarrow F[+X]F[-X]+X$ $F \rightarrow FF$	$X \rightarrow F[+X][-X]FX$ $F \rightarrow FF$	$X \rightarrow F-[[X]+X]+F[+FX]-X$ $F \rightarrow FF$

Stochastic L-System

W : F ▲ 選択の確率
P1 : F^{0.33} F[+F]F[-F]F
P2 : F^{0.33} F[+F]F[-F]F
P3 : F^{0.34} F[+F]F[-F]F

} ランダム選択

<http://algorithmicbotany.org/>



L-Systemによる植物形状の描画

Przemyslaw Prusinkiewicz
 “The Algorithmic Beauty
 of Plants”

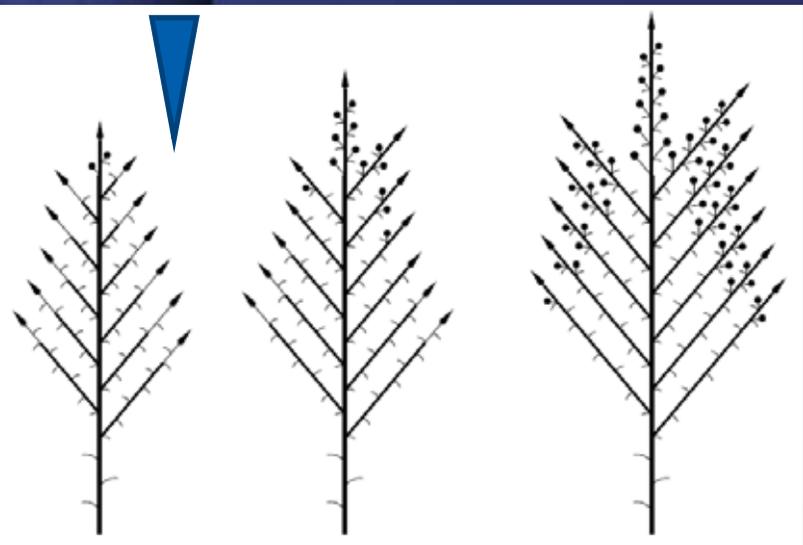
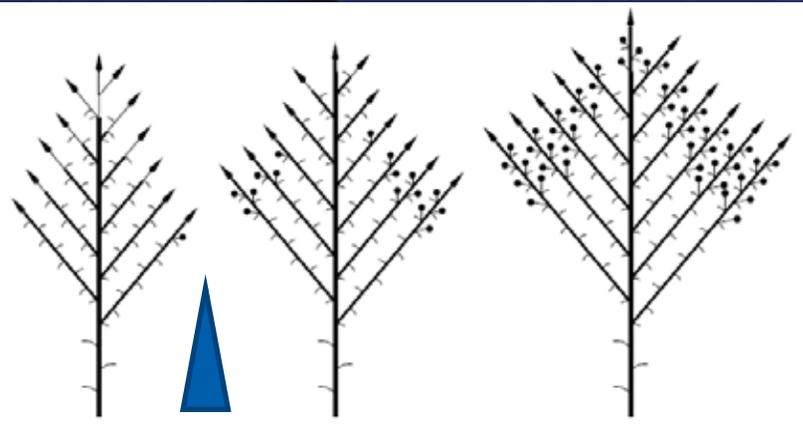
<http://algorithmicbotany.org/>



$n=7, \delta=22.5^\circ$

ω : A
 p_1 : A \rightarrow [&FL!A]/////’ [&FL!A]////////’ [&FL!A]
 p_2 : F \rightarrow S ///// F
 p_3 : S \rightarrow F L
 p_4 : L \rightarrow [’’’^^{-f+f+f-|-f+f+f}]

植物の生長シミュレーション



<http://algorithmicbotany.org/>

植物表現の例

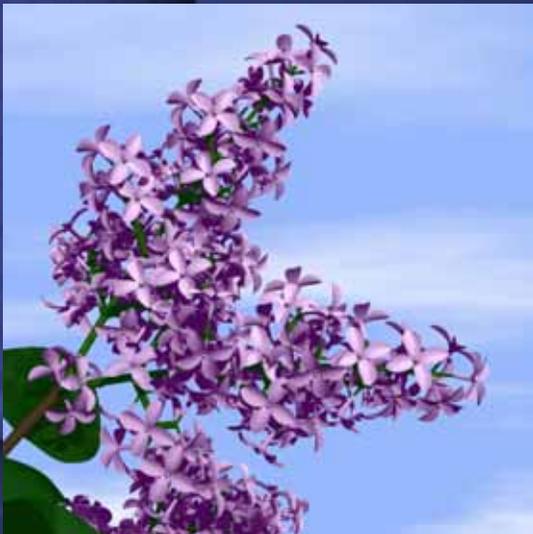
盆栽



千葉@岩手大学



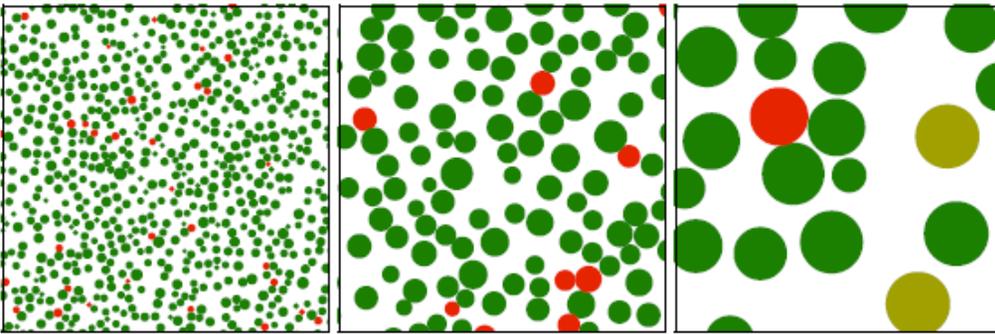
O. Deussen



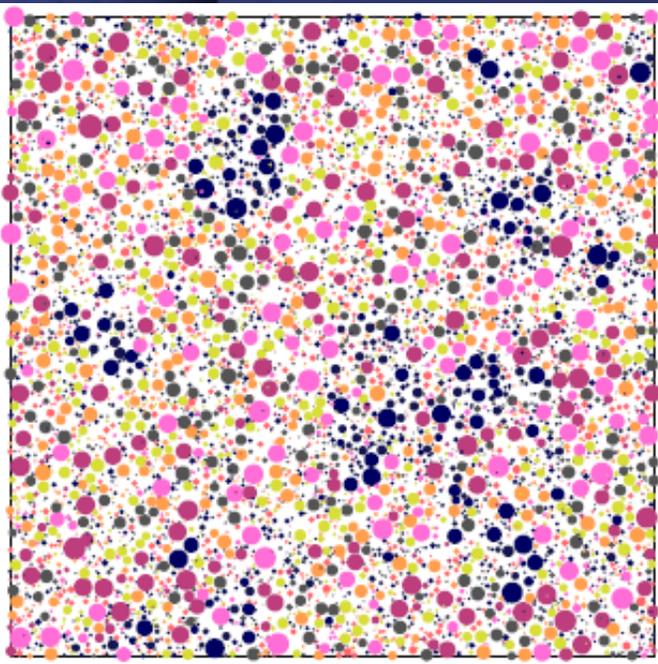
Przemyslaw Prusinkiewicz
 “The Algorithmic Beauty
 of Plants”

<http://algorithmicbotany.org/>

植物の生態系



O.Deussen, et.al., "Realistic modeling and rendering of plant ecosystems," SIGGRAPH1998



植物の生態系



Przemyslaw Prusinkiewicz:
“Simulation Modeling of Plants and Plant
Ecosystems,” *CACM*, 43, 7, pp. 84-93.



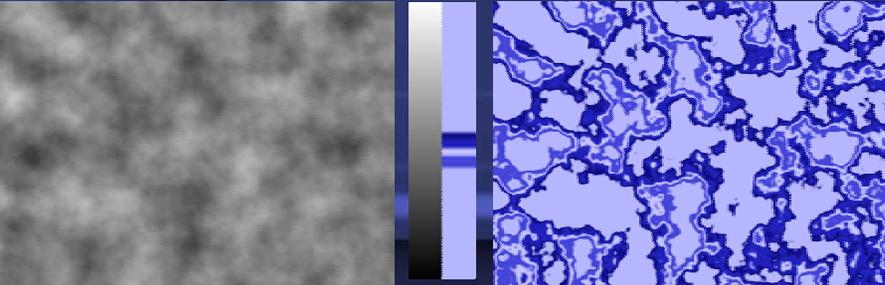
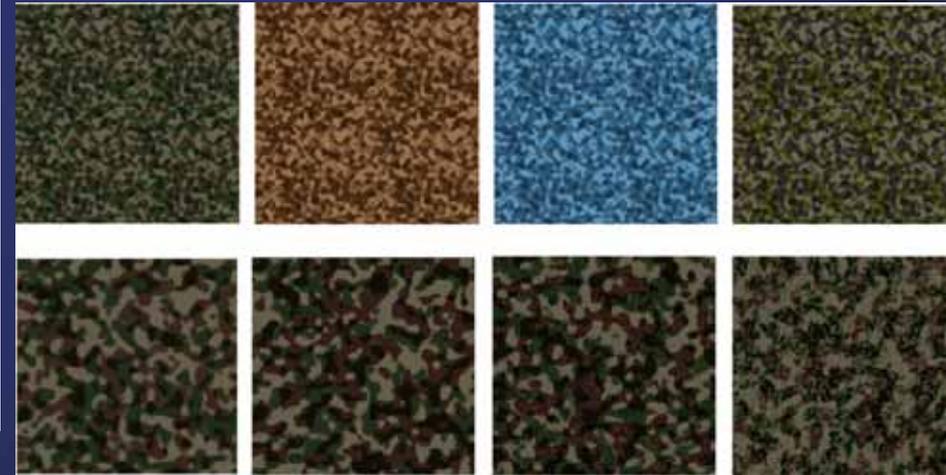
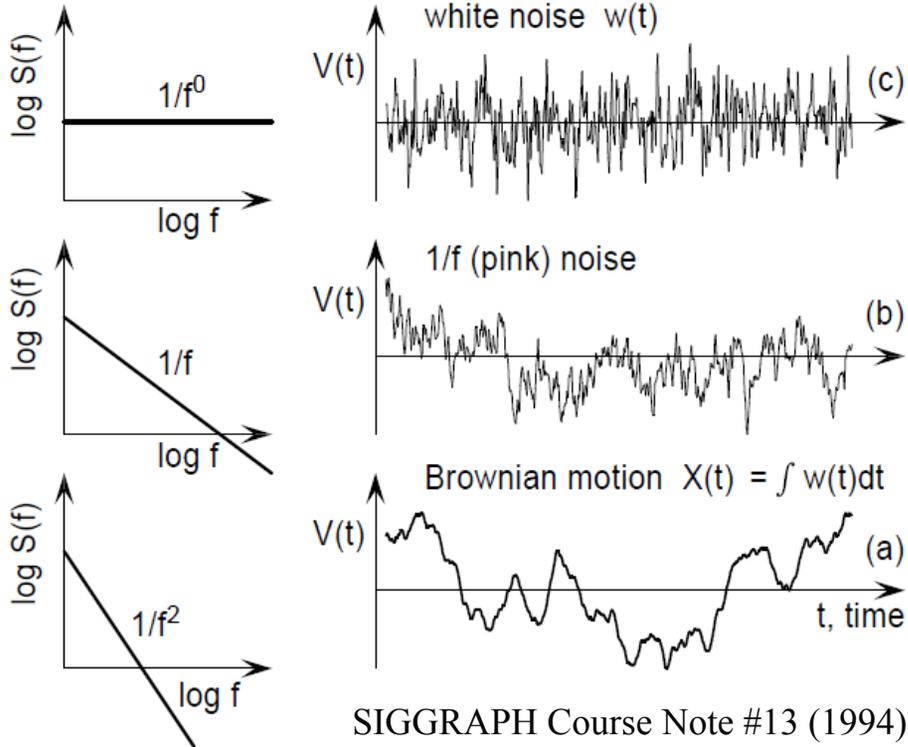
テクスチャの表現

- Noise Based (fractal, Perlin)
- Reaction-Diffusion Texture
- Cellular Texture

15:18?

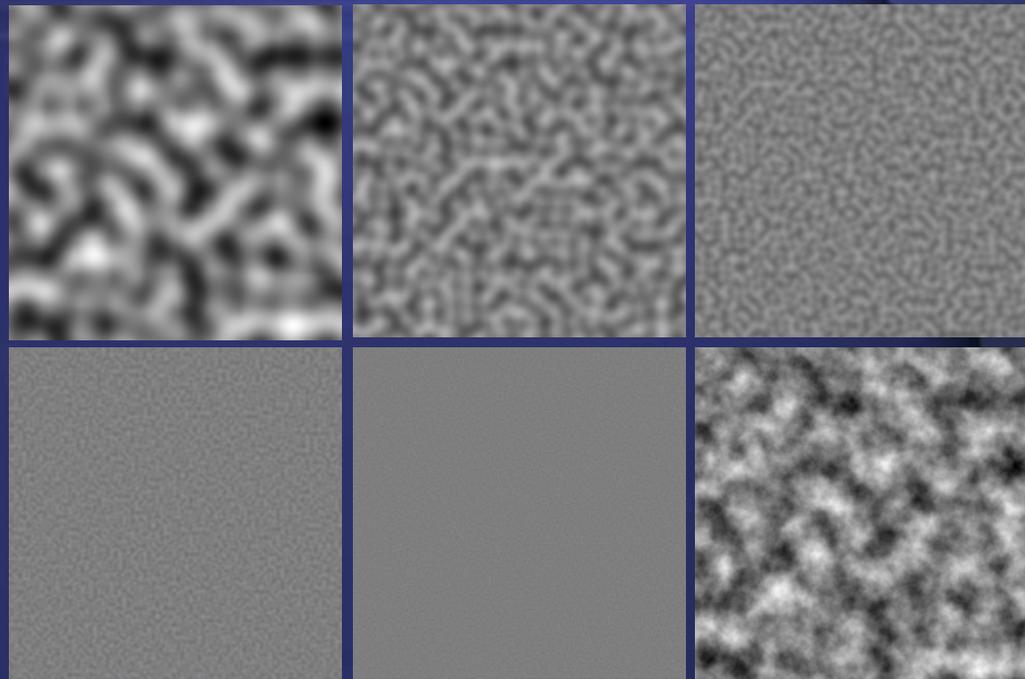
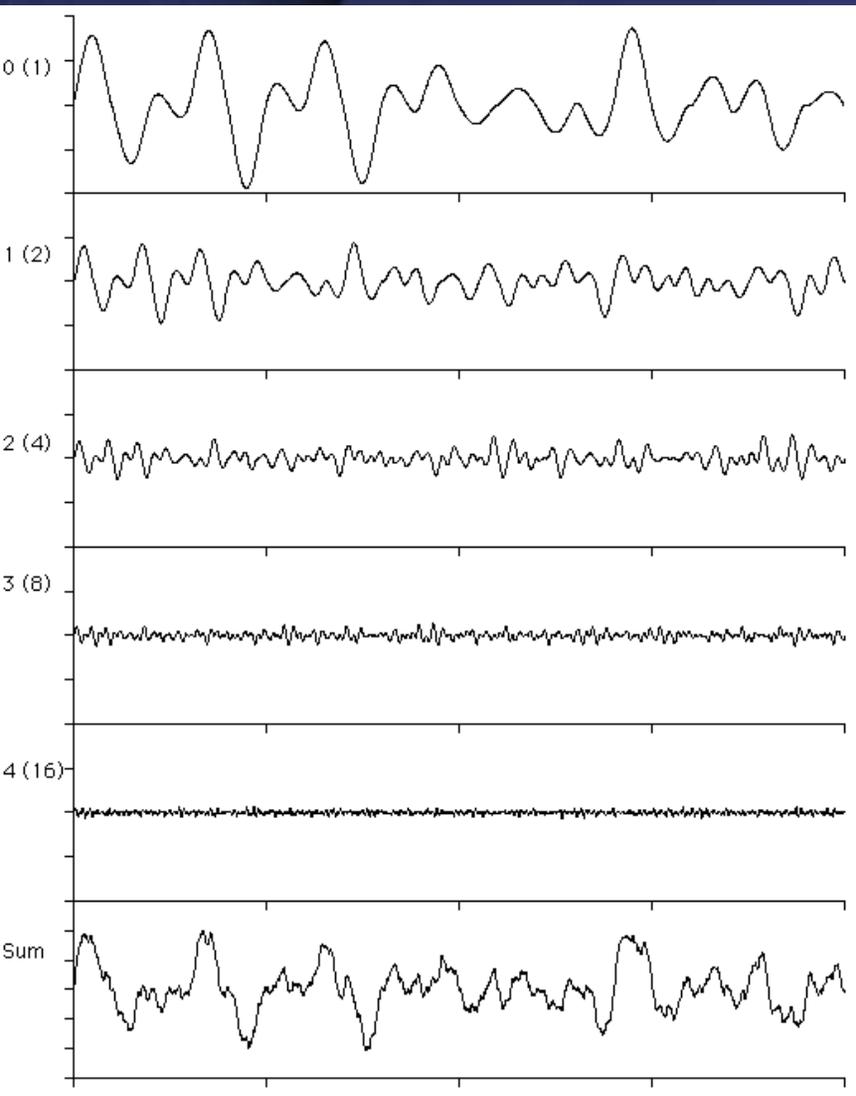


ノイズ

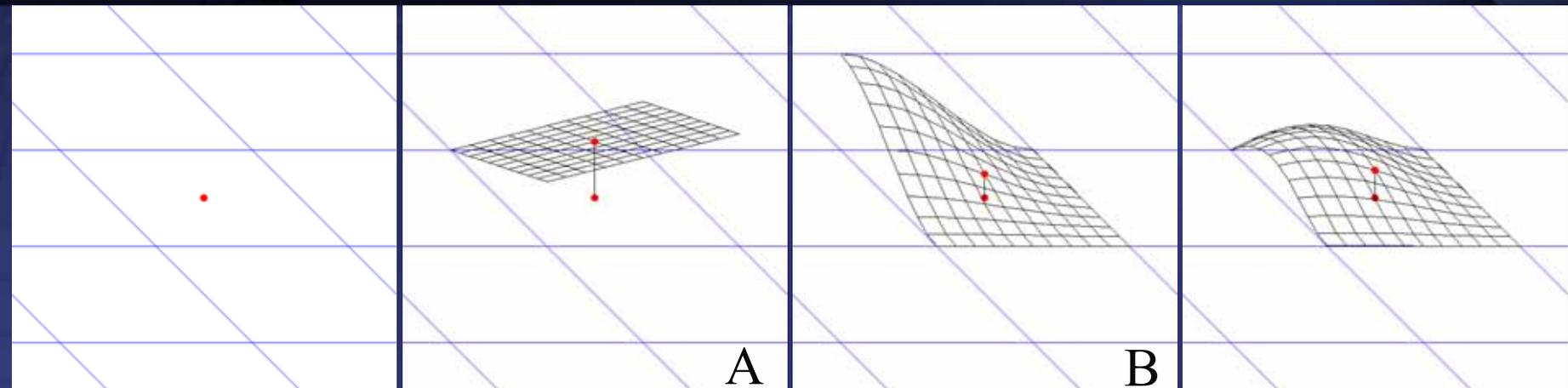


<http://www.davidcornette.com/gsl/noise.html>

Perlinノイズ



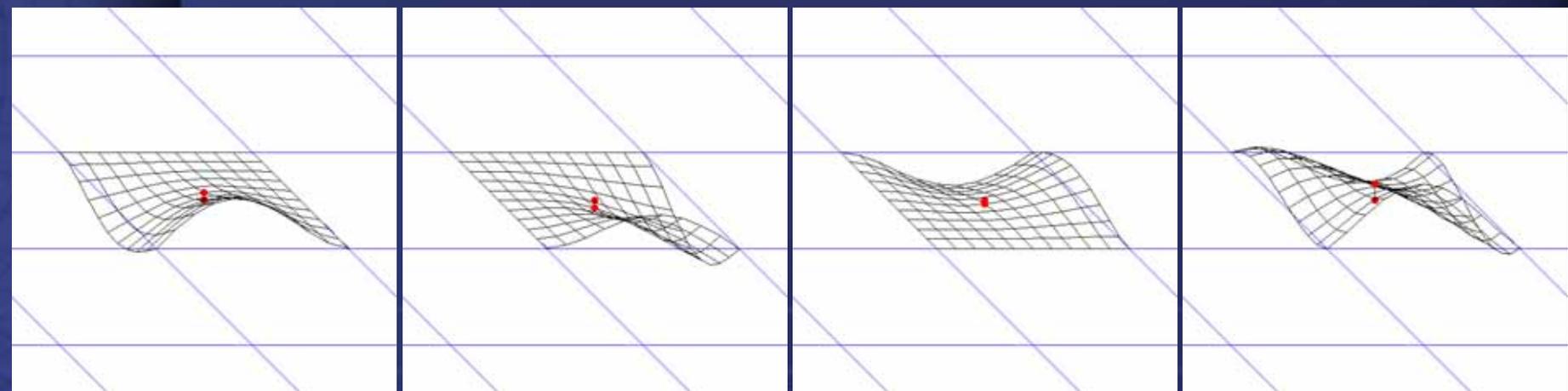
Perlinノイズ



ランダムな傾きを求める

減衰関数を与える

$A \times B$



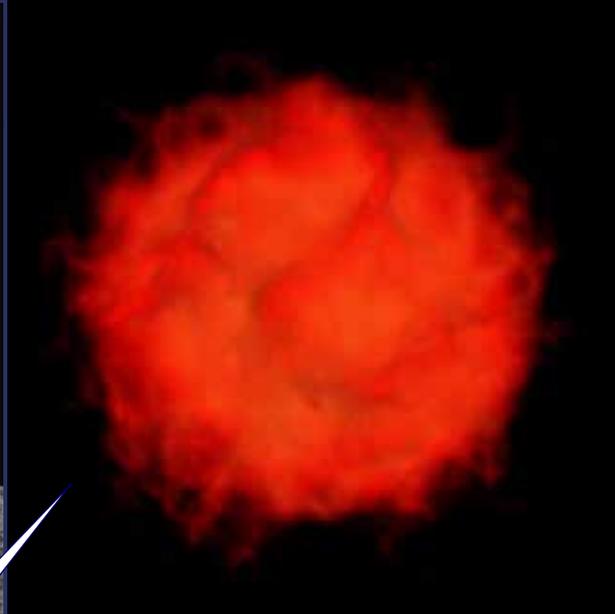
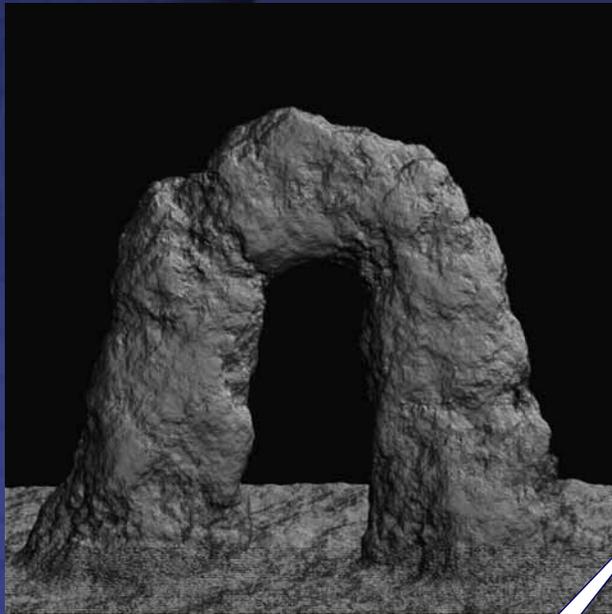
他の3点も同様にランダム傾き × 減衰関数

最終結果

K.Parlin, "Making Noise", <http://www.noisemachine.com/talk1/>

Hyper Texture

形状 + テクスチャ: 空間充填関数



$$D(x) = \text{sphere}(x(1 + \text{turbulence}(x)))$$

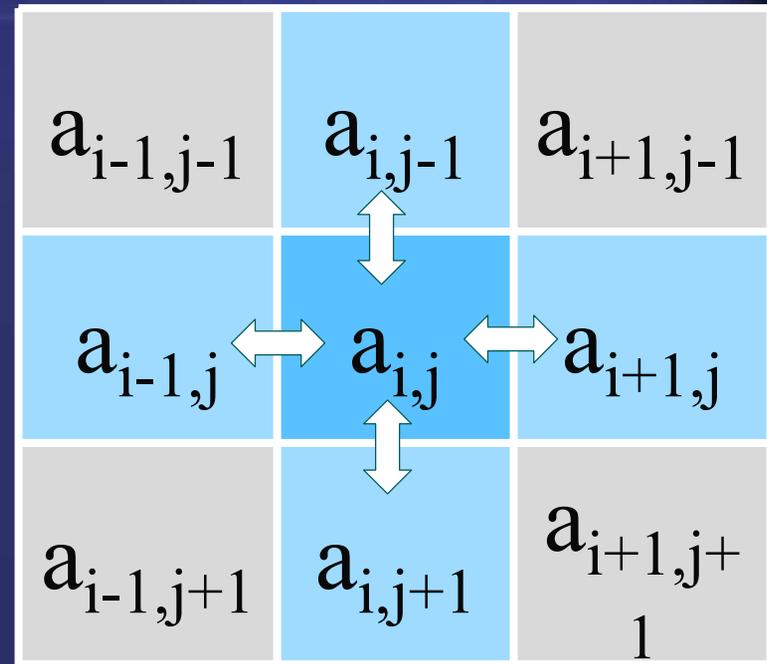
$$\text{turbulence} := \text{abs}(1/2^i \text{noise}(2^i x))$$

Reaction-Diffusion テクスチャ

Turingモデル

$$\frac{\partial a}{\partial t} = F(a, b) + D_a \nabla^2 a$$

$$\frac{\partial b}{\partial t} = G(a, b) + D_b \nabla^2 b$$

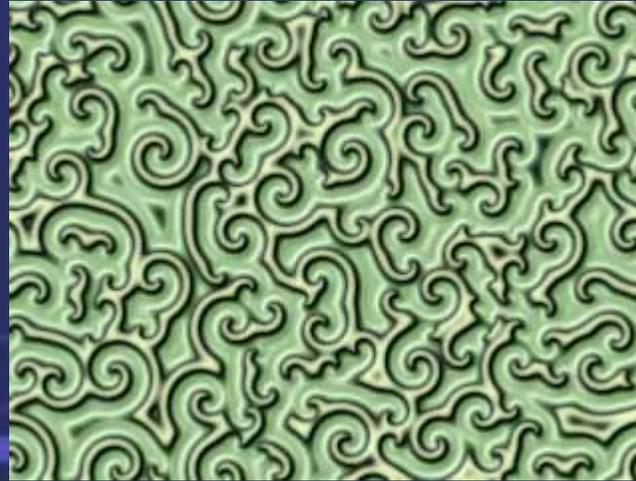
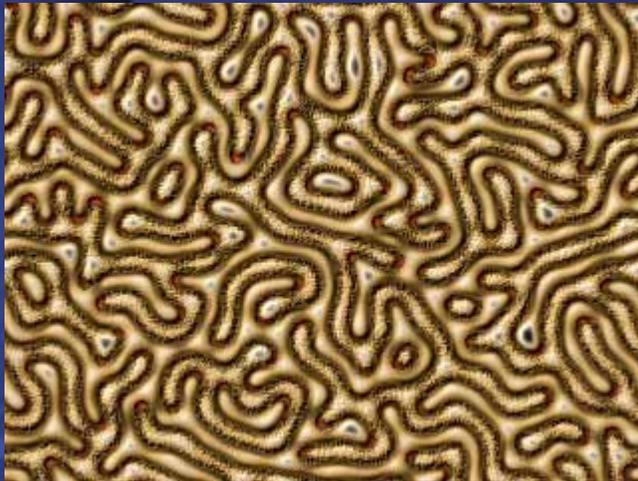


2種類の因子(a,b)の化学反応と隣接点間の
拡散反応によりパターンを形成

Reaction-Diffusion テクスチャ

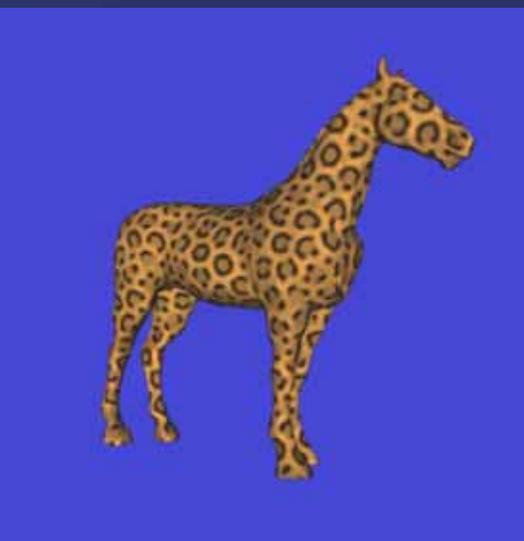
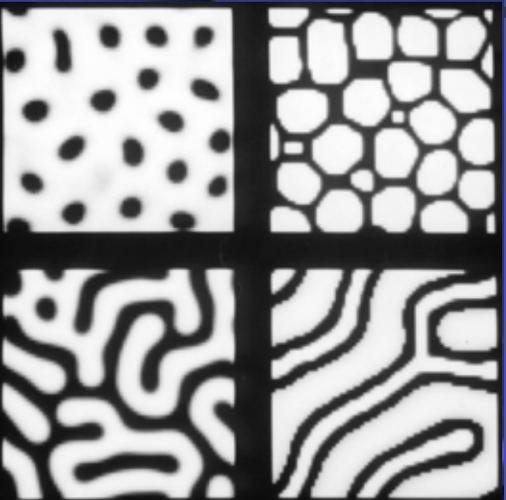
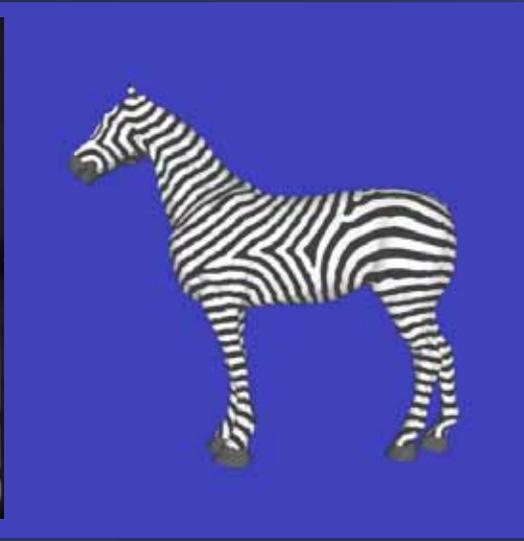
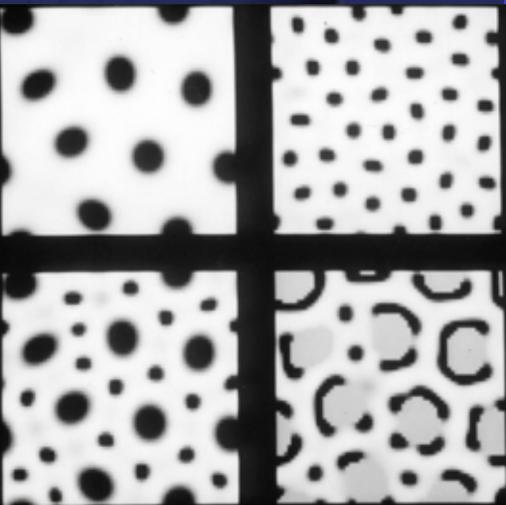


<http://cnls.lanl.gov/~aric/Simulations/Simulations.html>

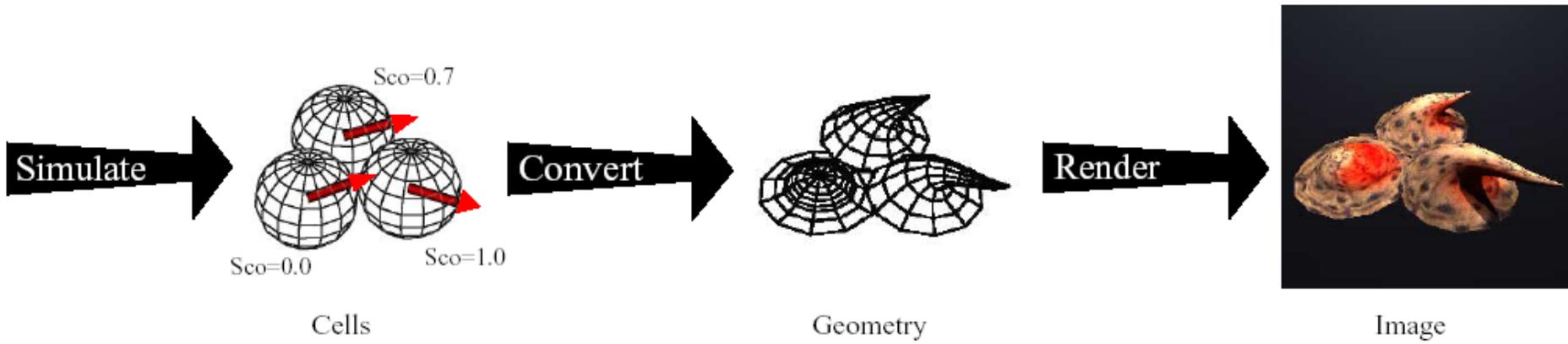


<http://softology.com.au/gallery/galleryrd.htm>

Reaction-Diffusionテクスチャ



Cellular Texture

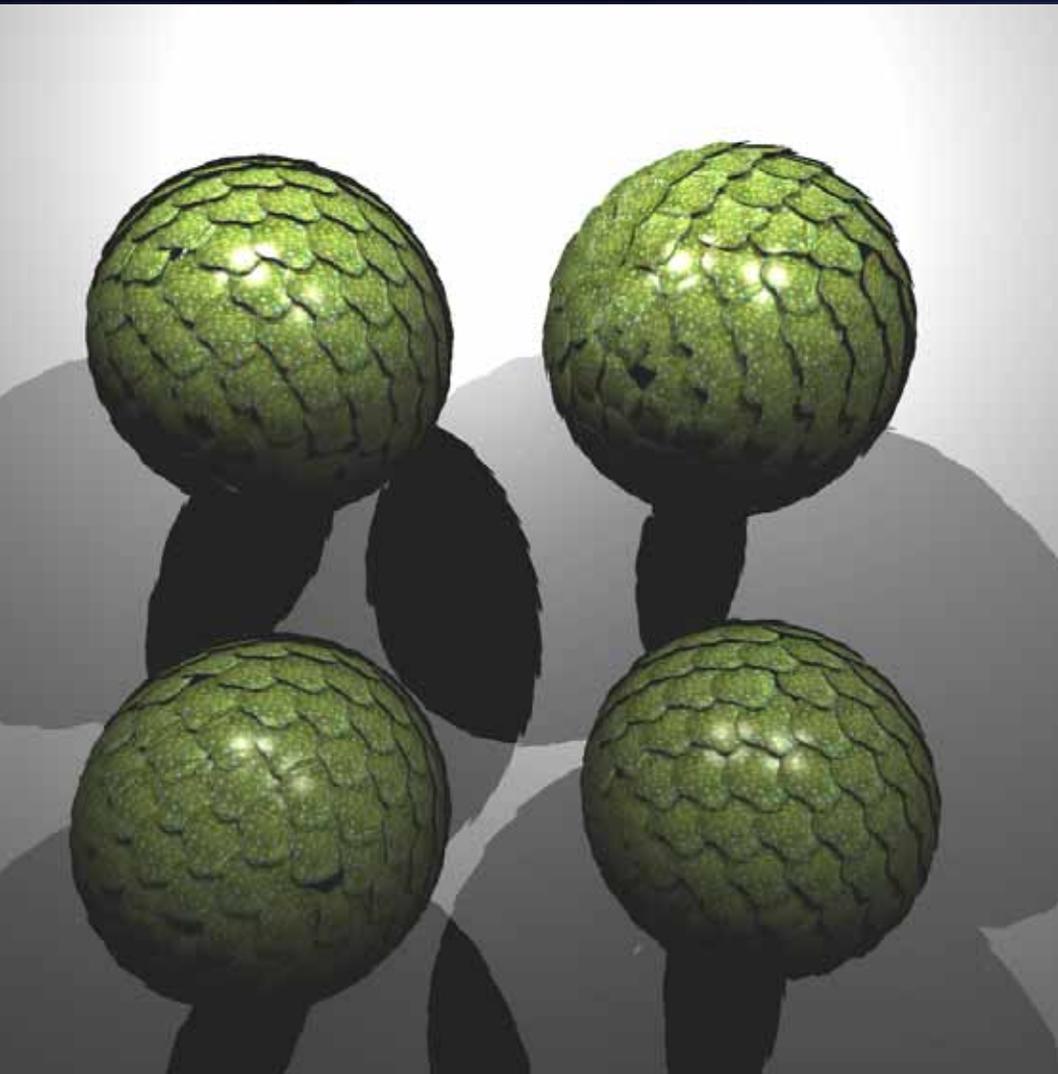


“Cellular Texture Generation”
 Kurt W. Fleischer, et. al., SIGGRAPH'95

Cell Programs

Behavior	Environment Requirements	Cell Program
Go to a surface.	An implicit surface $f(x) = 0$.	$\mathbf{p}' += -k f(\mathbf{p}) \nabla f(\mathbf{p})$
Die if too far from surface.	An implicit surface $f(x) = 0$.	$S'_{die} += \frac{1}{d} f(\mathbf{p}) \theta_{die} - S_{die}$
Align an axis with a vector field.	A vector field, $\mathbf{v}(x)$. \mathbf{e}_y is the cell's y -axis.	$\omega_v \equiv k \frac{\mathbf{e}_y \times \mathbf{v}(\mathbf{p})}{\ \mathbf{e}_y \times \mathbf{v}(\mathbf{p})\ } \cos^{-1}(\mathbf{e}_y \cdot \frac{\mathbf{v}(\mathbf{p})}{\ \mathbf{v}(\mathbf{p})\ })$ $\mathbf{q}' += (1/2) \omega_v \mathbf{q}$
Align x -axis with neighbors.	$\Lambda_{\omega x}$, x -axis orientation relative to neighbors.	$\mathbf{q}' += (1/2) \Lambda_{\omega x} \mathbf{q}$
Align z -axis with neighbors.	$\Lambda_{\omega z}$, z -axis orientation relative to neighbors.	$\mathbf{q}' += (1/2) \Lambda_{\omega z} \mathbf{q}$
Maintain unit quaternion.		$\mathbf{q}' += 4k(1 - \mathbf{q} \cdot \mathbf{q})\mathbf{q}$
Adhere to other cells	Membrane chemical $a2$ which binds to itself.	$S'_{a2} += 1.0 - S_{a2}$
Divide until surface is covered.	Λ_{a2} , amount of $a2$ which is bound.	$S'_{split} += \phi(\gamma, \Lambda_{a2}) \theta_{split} - S_{split}$
Set size relative to surface feature size	Λ_{u0} , a value which reflects the size of the nearest feature on the surface.	$r' += \Lambda_{u0} - r$
Example of reaction-diffusion in discrete cells.	$\Lambda_{a0}, \Lambda_{a2}$ amounts of bound membrane chemicals. The user has specified that the membrane chemical $a0$ binds to $a1$, and that $a2$ binds to itself.	$S'_{c0} += -20 \Lambda_{a0} / \Lambda_{a2} +$ $10 S_{c0}^2 / (1 + S_{c0}^2) - 0.5 S_{c0} + 13$ $S'_{a1} += 0.95 \Lambda_{a0} / \Lambda_{a2}$ $S'_{a1} += \phi(3, S_{c0}) S_{c0}$ $S'_{a1} += -S_{a1}$ $S'_{a0} += 5 - S_{a0}$ $S'_{a2} += 1 - S_{a2}$

生成例：鱗



表面をカバーするように分割

パーティクルは表面にとどめる

表面から離れたら死滅

各パーティクルは4枚のポリゴン
で置き換え

鱗状になるように、若干傾ける

生成例：棘



Reaction-Diffusionモデルを使って、棘の配置を制御

生成例：熊

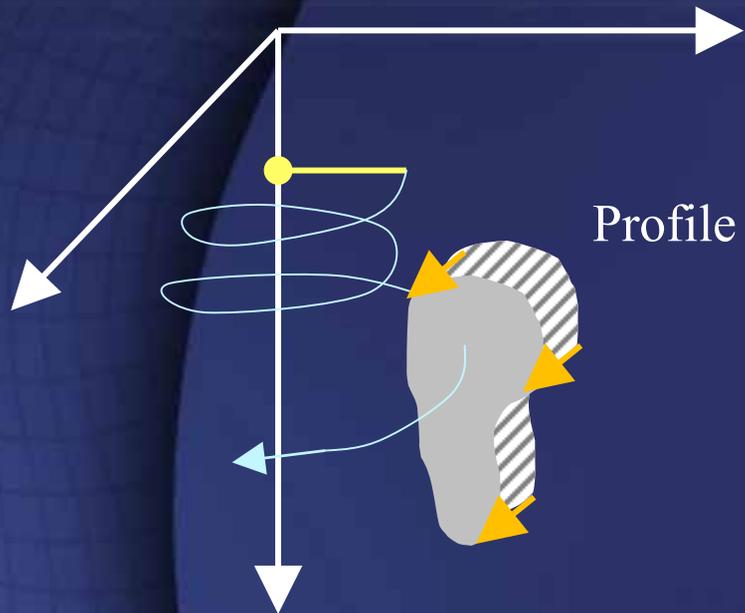


その他のモデルによる表現

- Growth Model
- Particle System
- Aging



貝の成長モデル

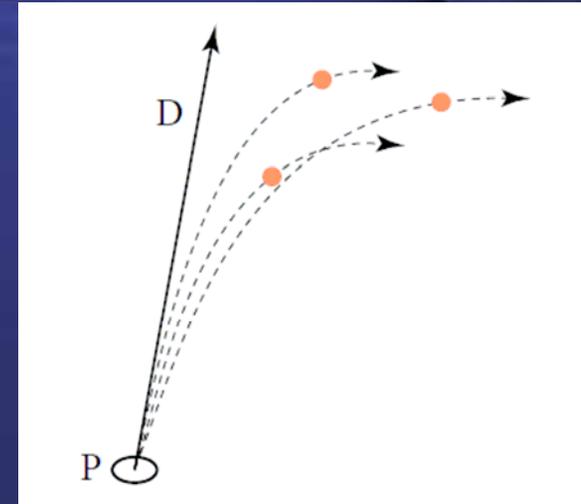


Reaction-Diffusionモデルによる模様づけ

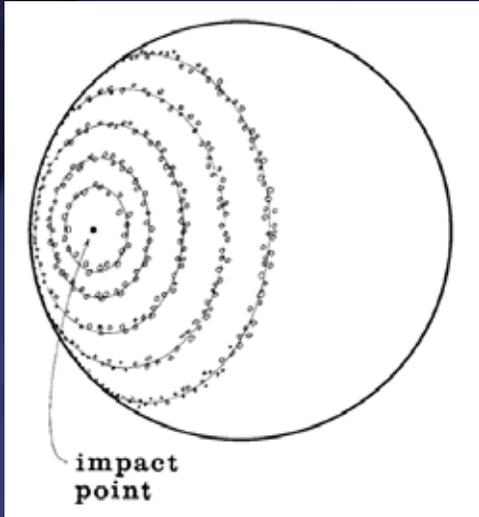
D.R.Fowler et.al, "Modeling seashells," SIGGRAPH1992

パーティクルシステム

- 3次元空間のある点P(もしくは領域)から、指定された方向D(もしくは全方向)に、ある量のパーティクルを時間経過にしたがって飛散させる
- 初速度や、重力の作用、色、寿命(例:花火の場合は火花が燃え尽きる時間)などの属性を持たせる
- 各パーティクルの生存期間内で物理方程式を解く

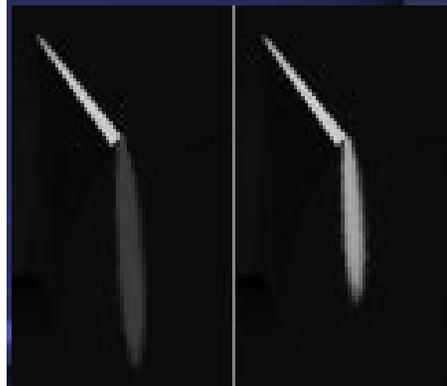
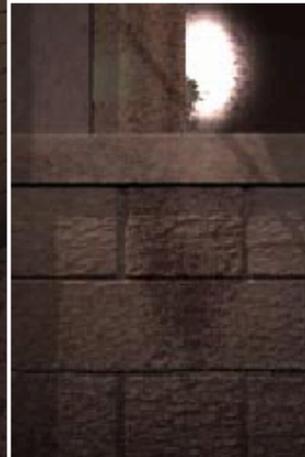
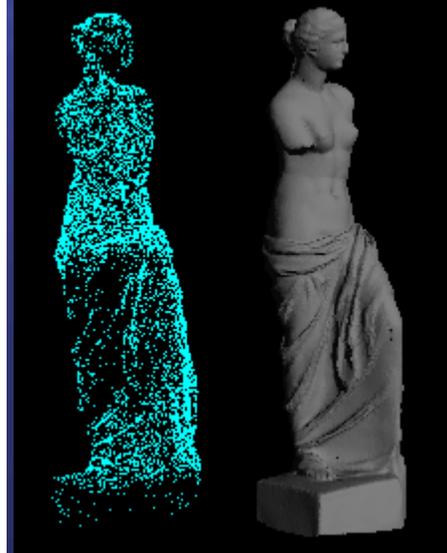
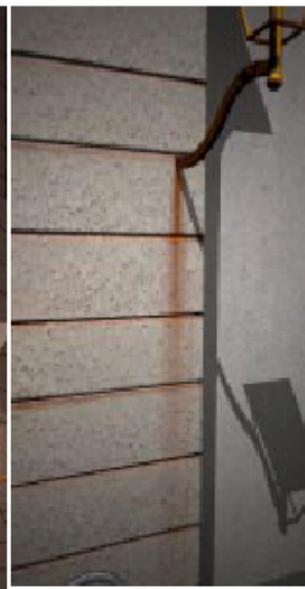


Star Trek II



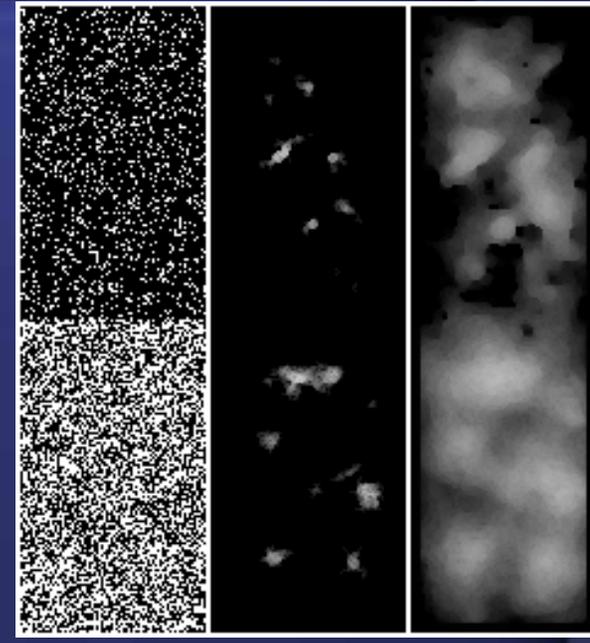
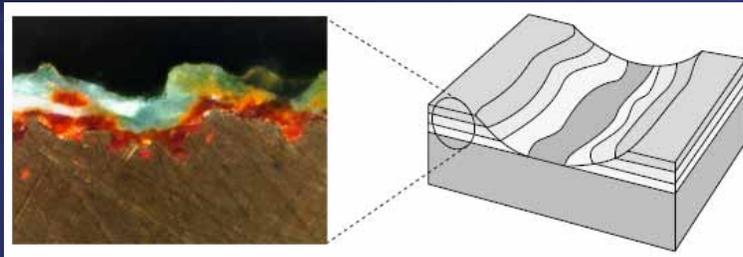
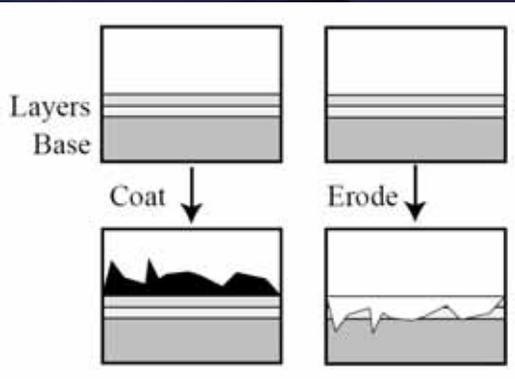
エイジング: 汚れ

パーティクルで汚れ粒子のシミュレーション



J.Dorsey et.al, "Flow and Changes in Appearance," SIGGRAPH1996

エージング: 錆

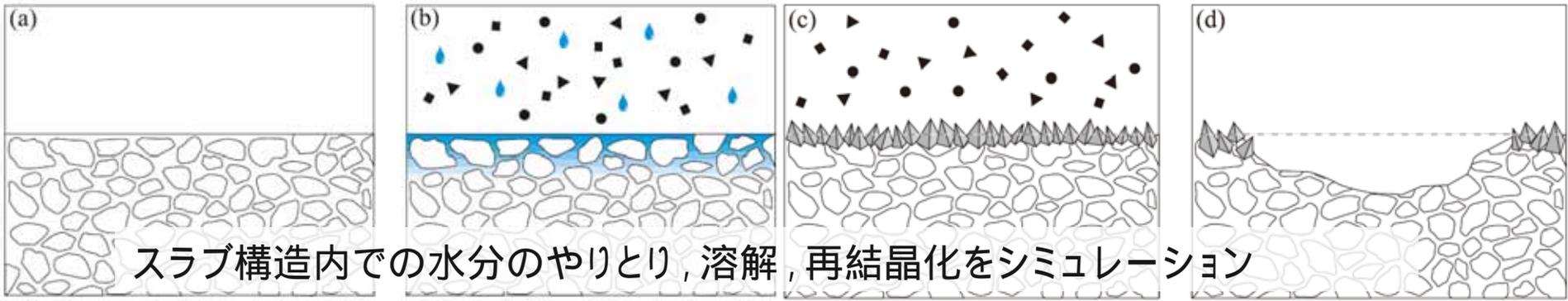


表面をレイヤー構造化し、錆の浸食をシミュレーション



J.Dorsey et.al, "Modeling and Rendering of Metallic Patinas," SIGGRAPH1996

エージング:劣化



J.Dorsey et.al, "Modeling and rendering of weathered stone," SIGGRAPH1999

アートへの展開



Computational Art

Apophysis

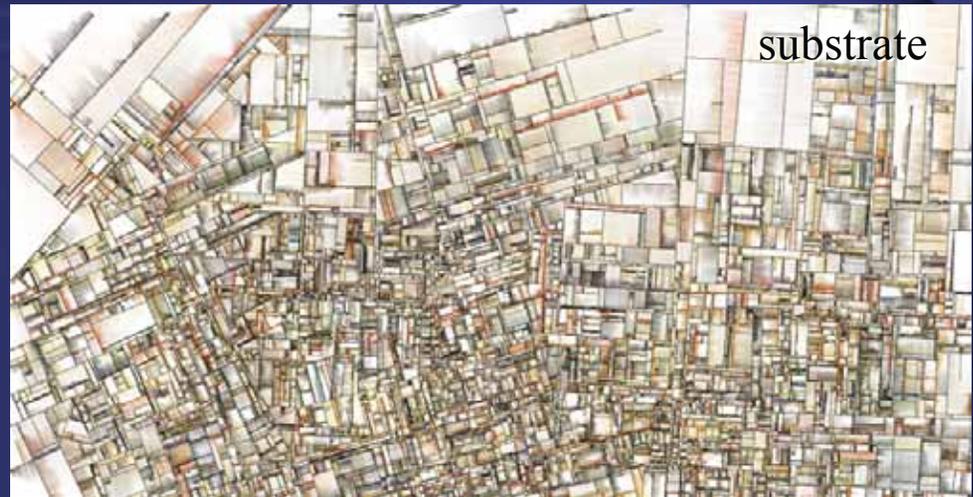


<http://www.apophysis.org/index.html>

Procedural Art

Processingを用いた
 プロシージャルな
 平面アート

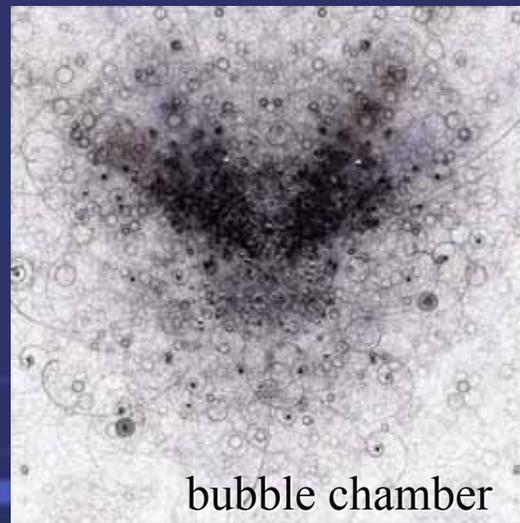
<http://www.complexification.net/>



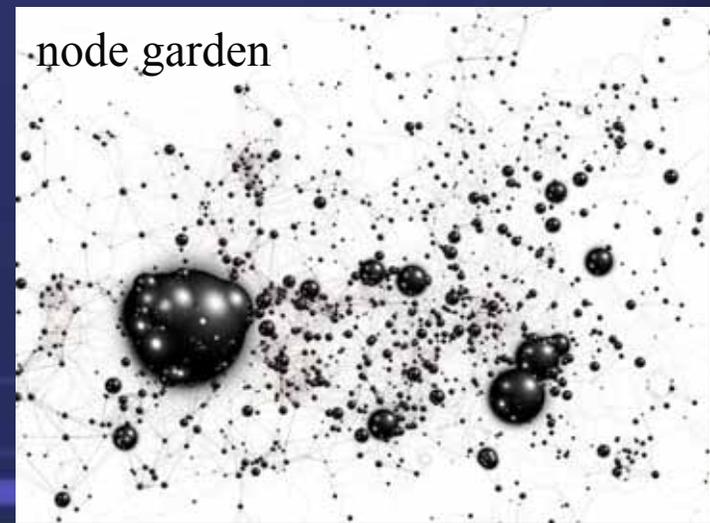
substrate



binary ring



bubble chamber

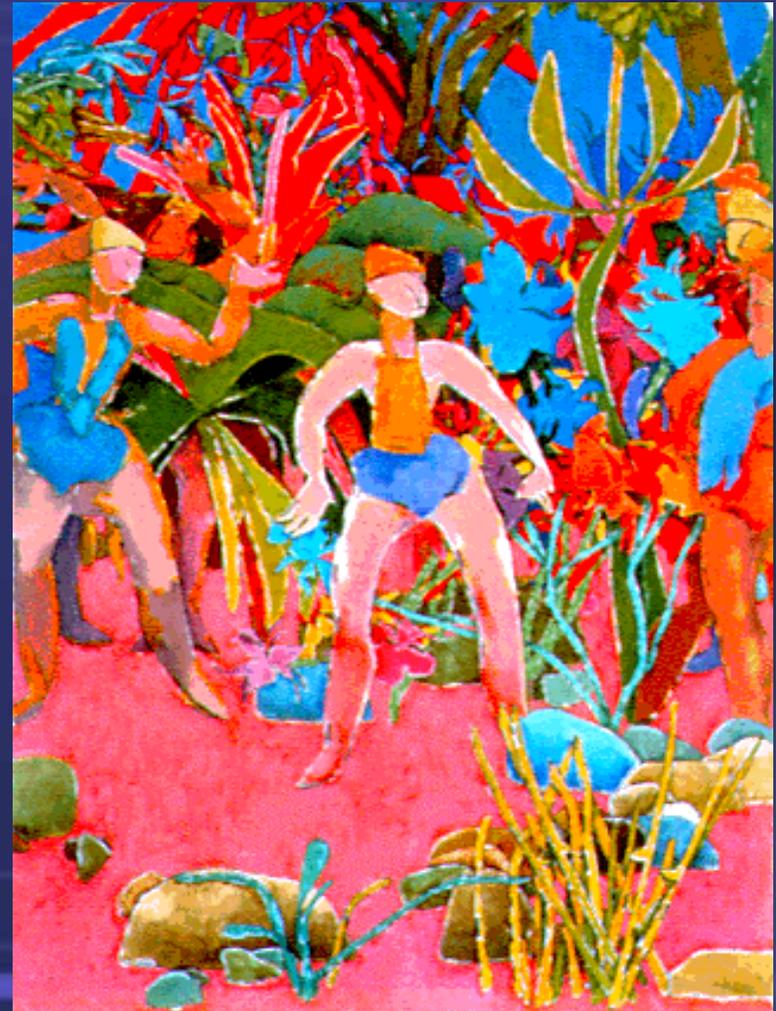


node garden

コンピュータ画家アーロン



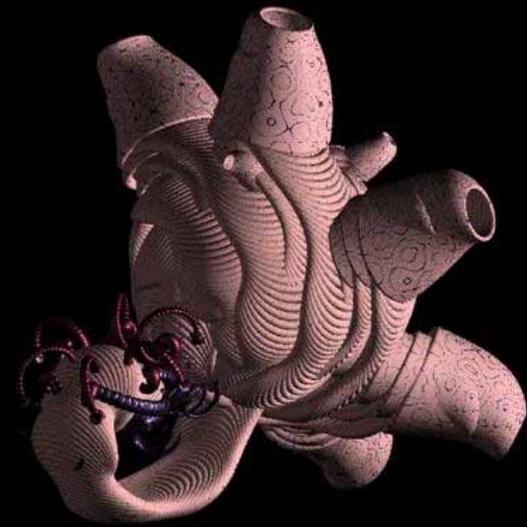
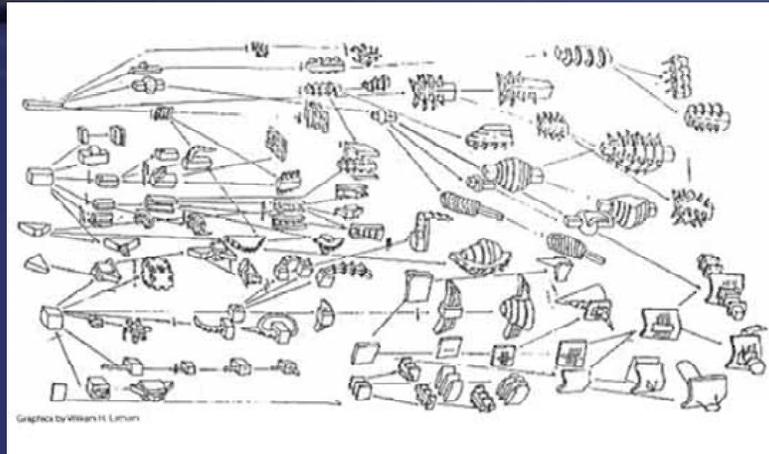
Harold Cohenの知識の外部化
芸術活動の過程と、
付随する事柄を明確化



<http://www.kurzweilcyberart.com/>

GAを用いたCGアート

William Latham



<http://www.doc.gold.ac.uk/~mas01whl/>

最近のプロシージャル技術

15:40?



Procedural Modelling of Cities

Y.Parish, P.Muller, SIGGRAPH 2001

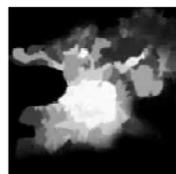
水面と陸地



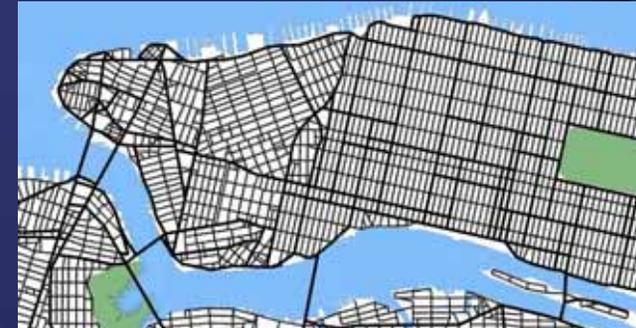
海拔



人口密度



道路網とビル群は、
L-systemで生成する



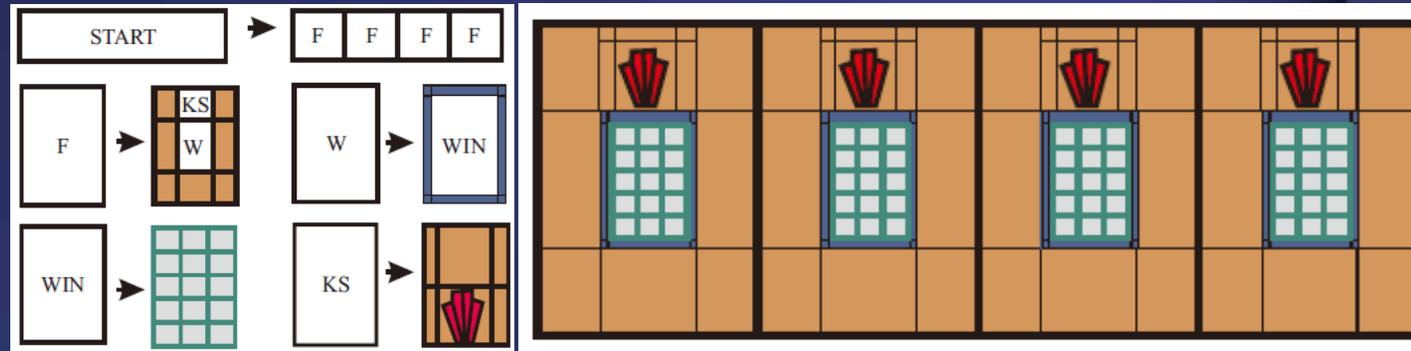
生成された都市景観

Procedural Modeling of Buildings

P.Muller, et.al, SIGGRAPH 2006

“Instant Architecture” P.Wonka, et.al, SIGGRAPH 2003のアイデアの拡張

Split Grammar

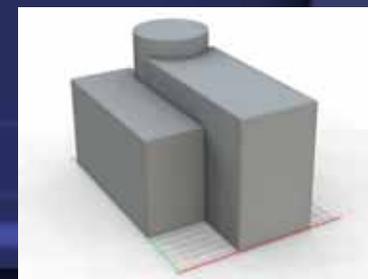
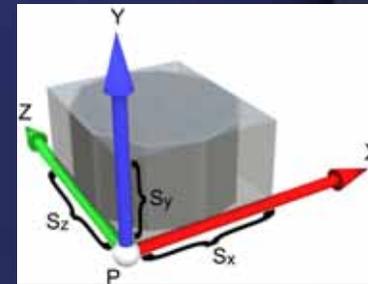


fac(h):h>9 floor(h/3) floor(h/3)floor(h/3)

A [T(0,0,6)S(8,10,18)I(“cube”)]
 T(6,0,0)S(7,13,18)I(“cube”)T(0,0,16)S(8,15,8)I(“cylinder”)

fac Subdiv(“Y”, 3.5, 0.3, 3, 3, 3)
 {floor | ledge | floor | floor | floor}

B	A	A	B	} floor	3.0m
B	A	A	B	} floor	3.0m
B	A	A	B	} floor	3.0m
				} ledge	30cm
B	A	A		} floor	3.5m



Procedural Modeling of Buildings



190のルールで自動生成

City Engine



www.procedural.com



Rome Reborn 2.0

古代ローマをCGで復元
City Engineを使用



展望：ゲームとの関係



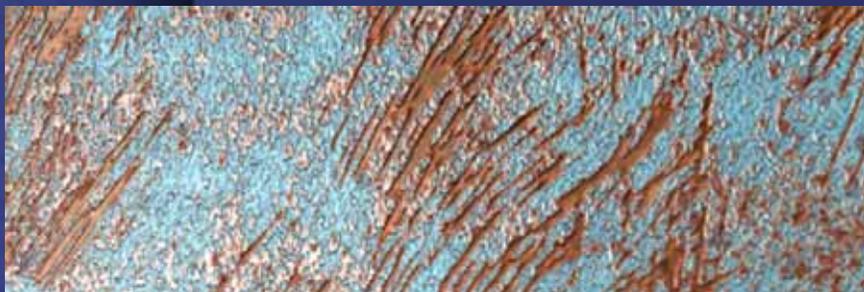
プロシージャルなゲームとツール



Spore

Frostbite@DICE

プロシージャル手法による
ゲーム制作への加速



MaPZone

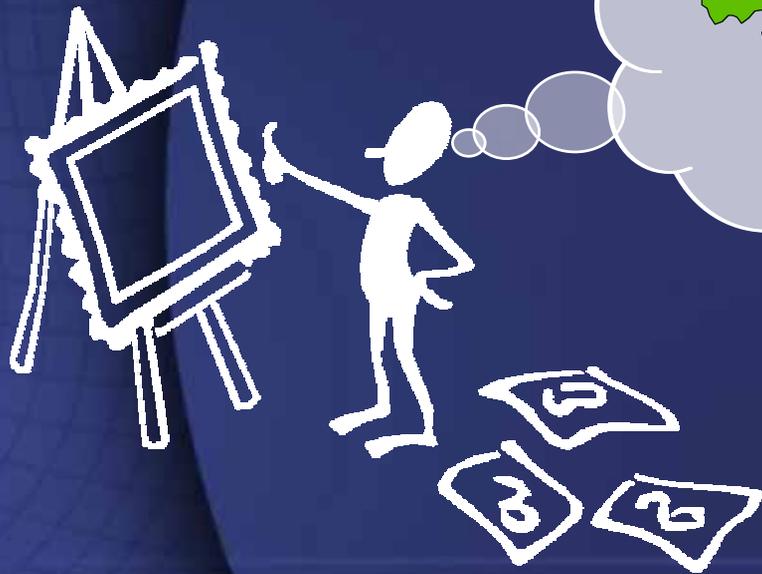


<http://www.allegorithmic.com/>

表現のメンタルモデル

• 直接・直感的

• 間接・論理的



• 暗黙知 身体知

• 暗黙知 形式知

表現のメンタルモデル

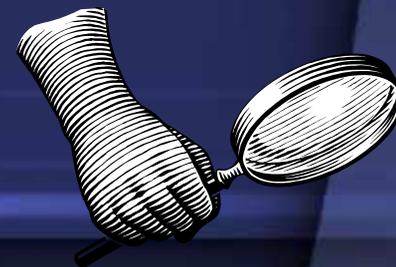


ゲームのテイストにどう合わせるか

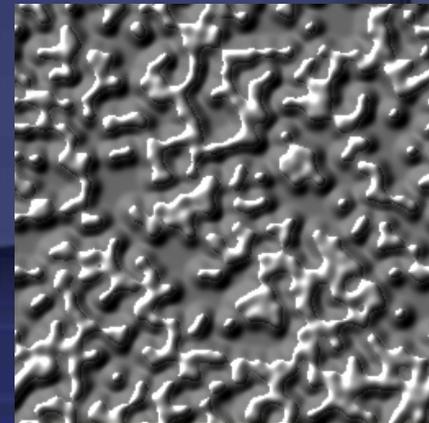
おまけ：研究の紹介

- テクスチャデザイン
- 爬虫類の皮膚
- 革シボ
- 漆工芸
- 風景バーテンダー

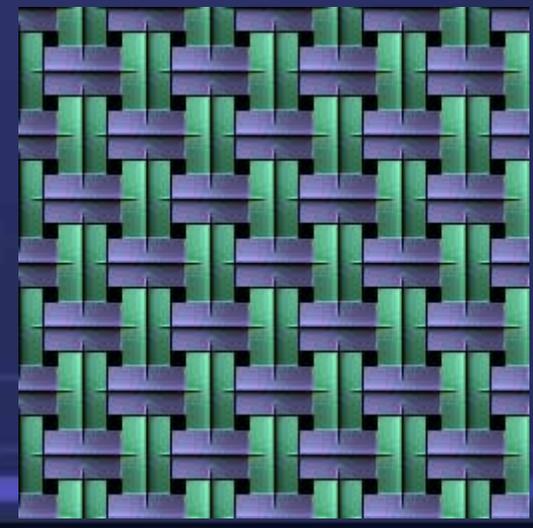
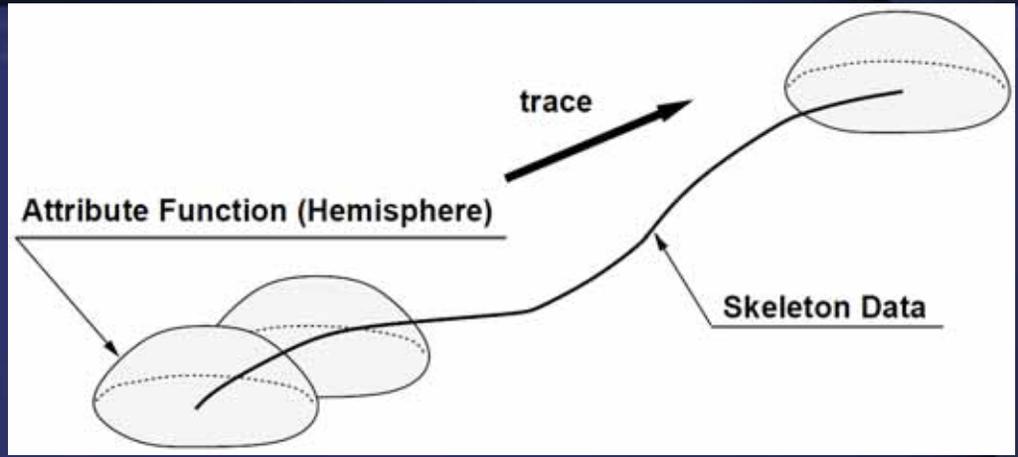
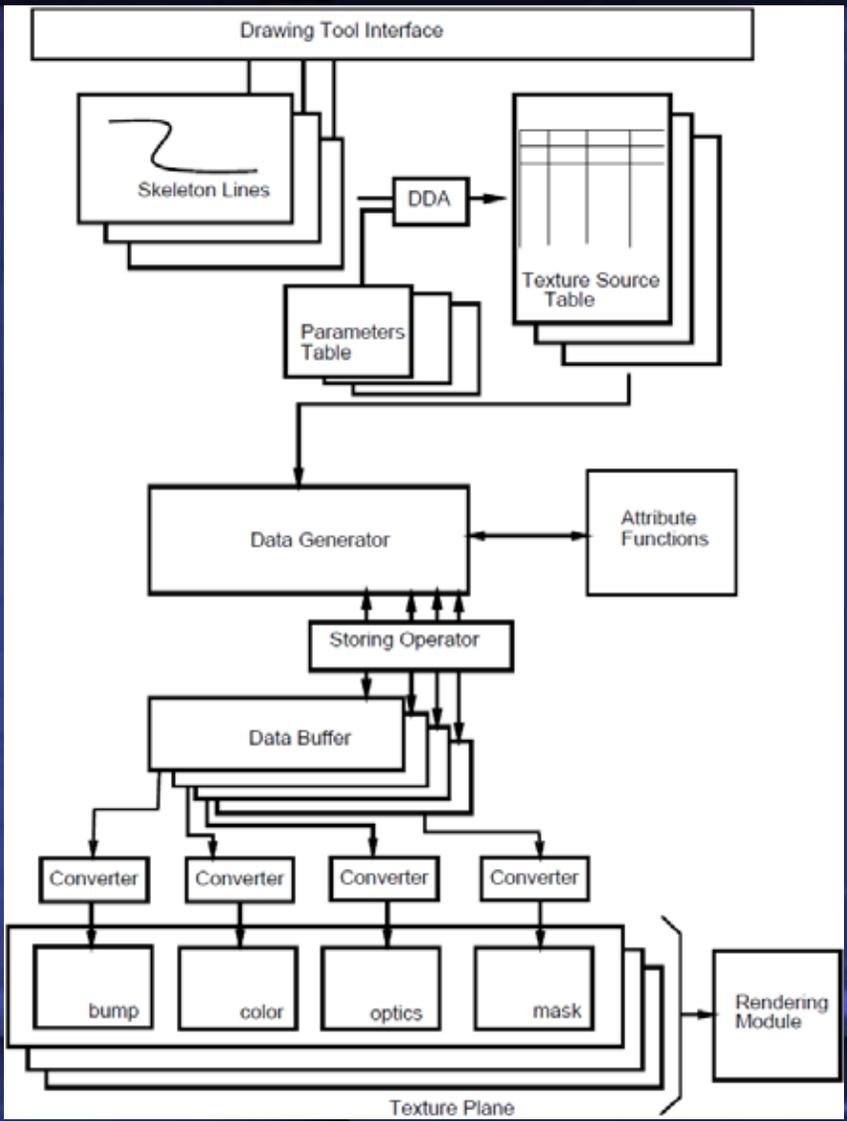
15:55?



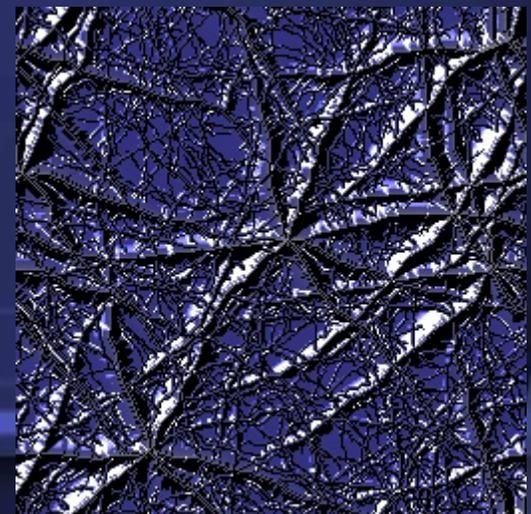
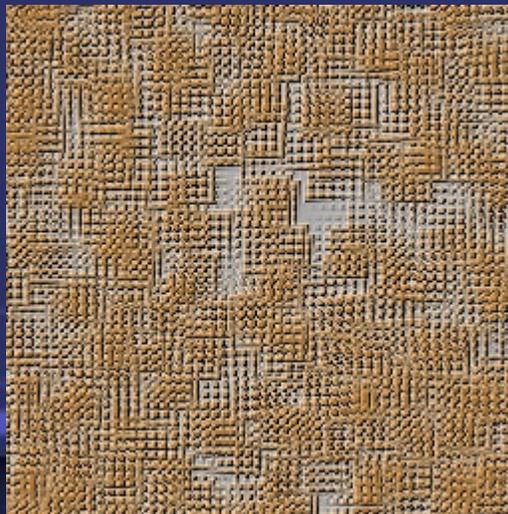
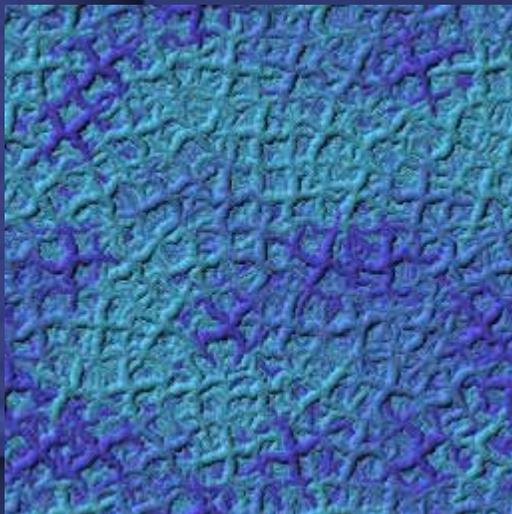
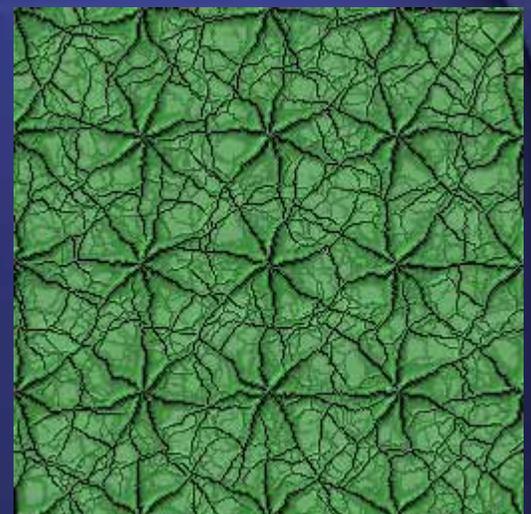
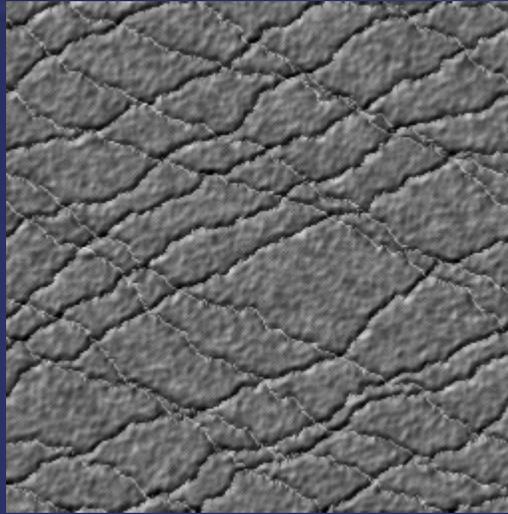
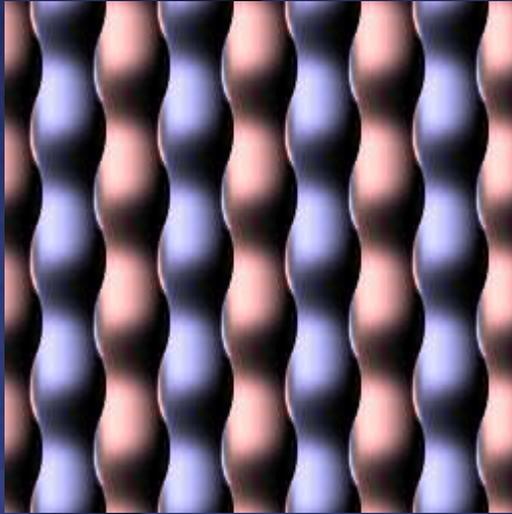
テクスチャデザイン



テクスチャデザイン



生成例



爬虫類の皮膚



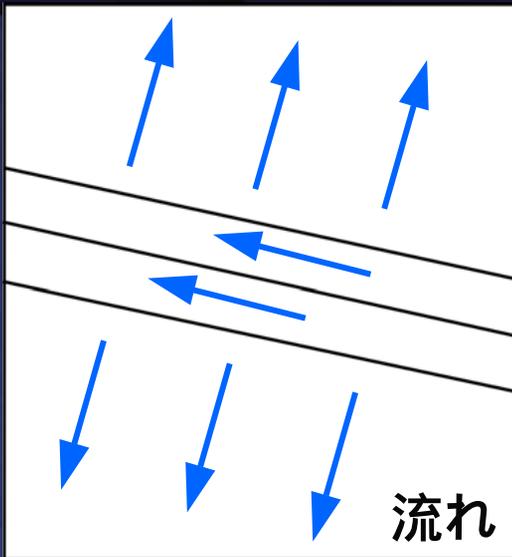
爬虫類の皮膚の例



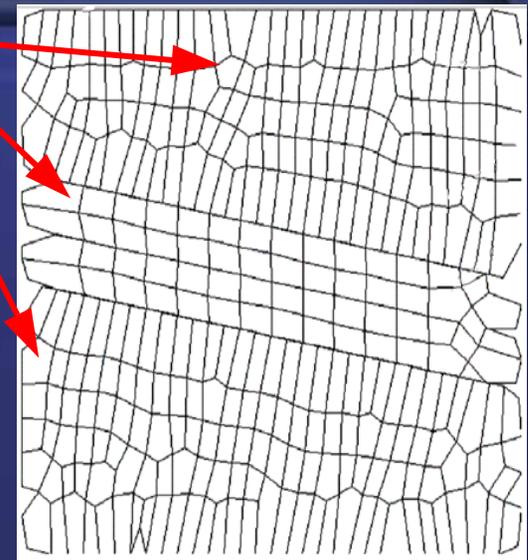
- 四辺形か六角形の組み合わせ
- 流れがある
- 流れの方向に伸びている



処理概要



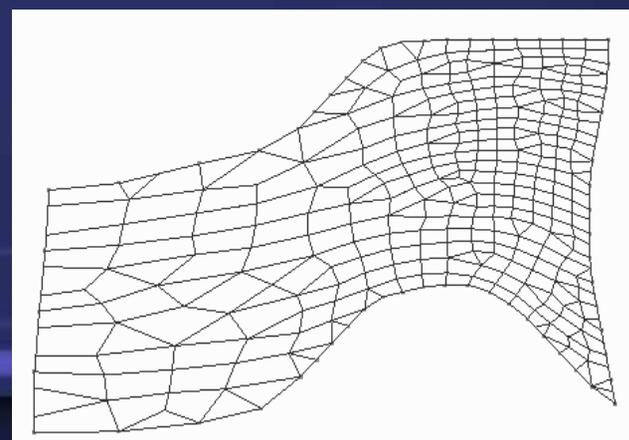
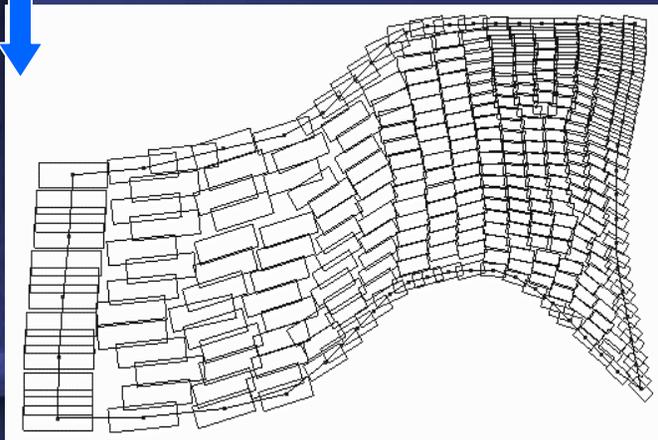
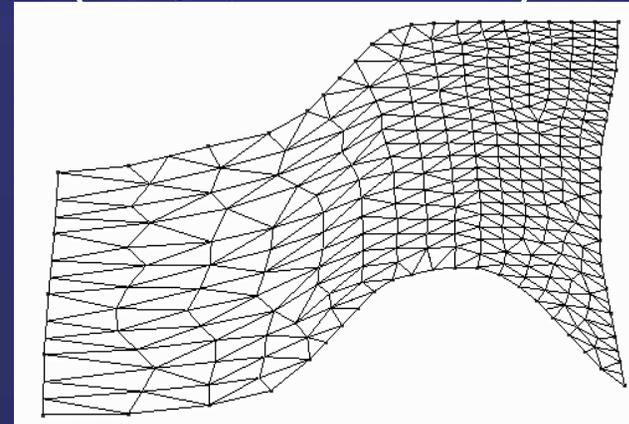
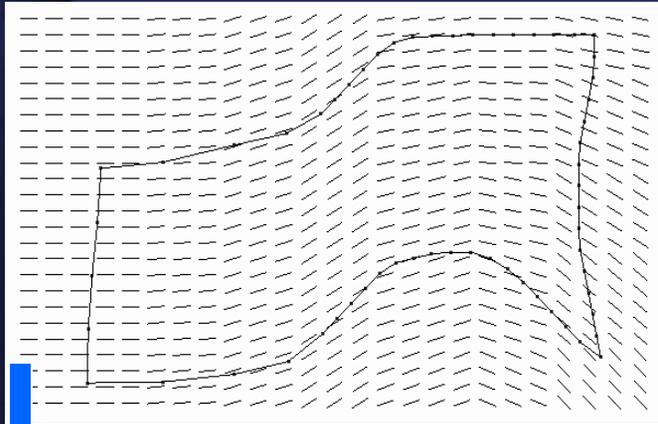
充填パターンの生成
制約条件



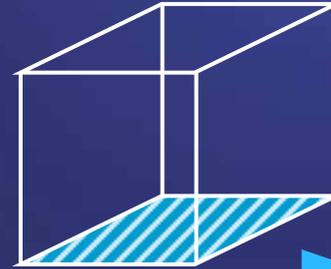
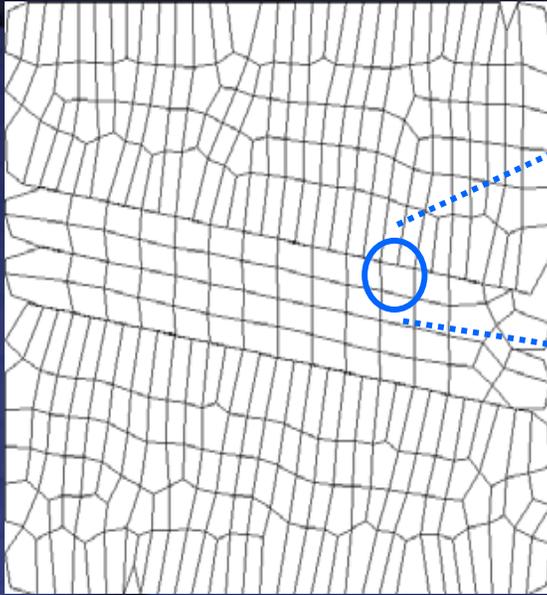
皮膚形状の生成

充填パターンの生成

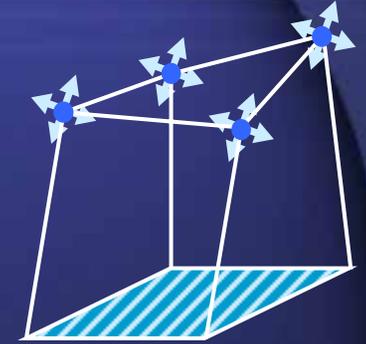
異方性パーティクルパッキング 異方性ドローネ分割
 三角形：四辺形変換 (必要に応じて)



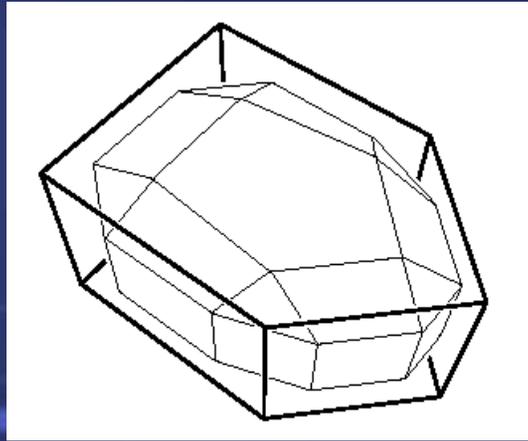
形状の生成



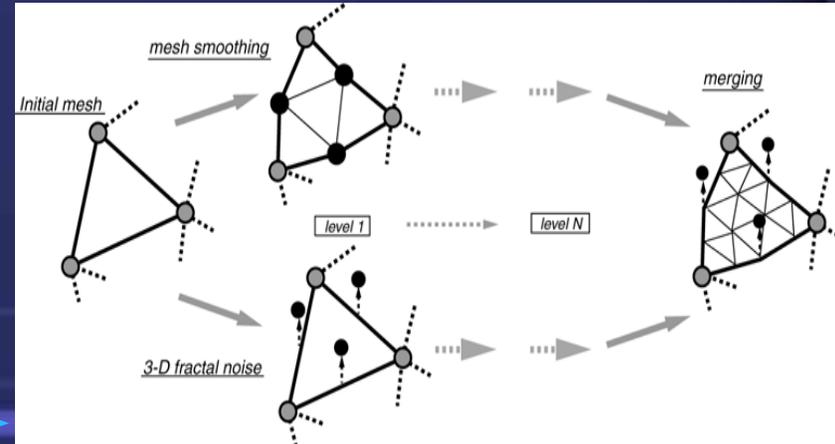
スイープ



変形

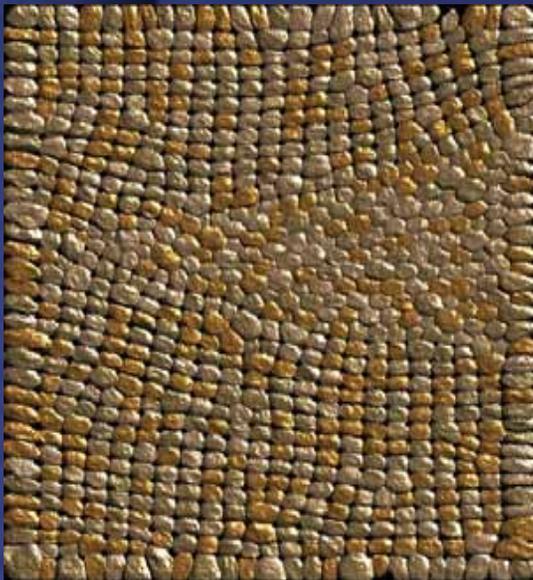
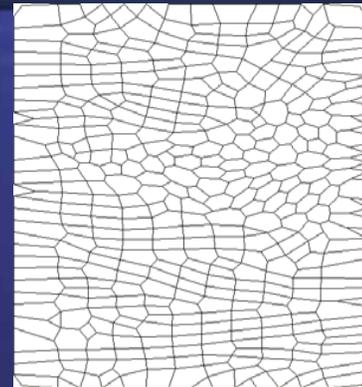
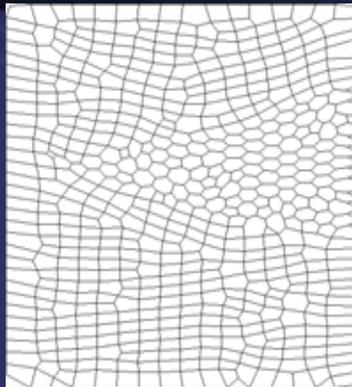
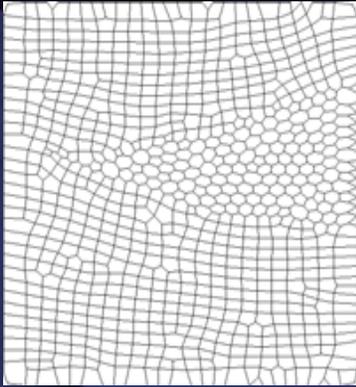


サブディビジョンサーフェイス



ノイズ付加

生成例：異方性の制御

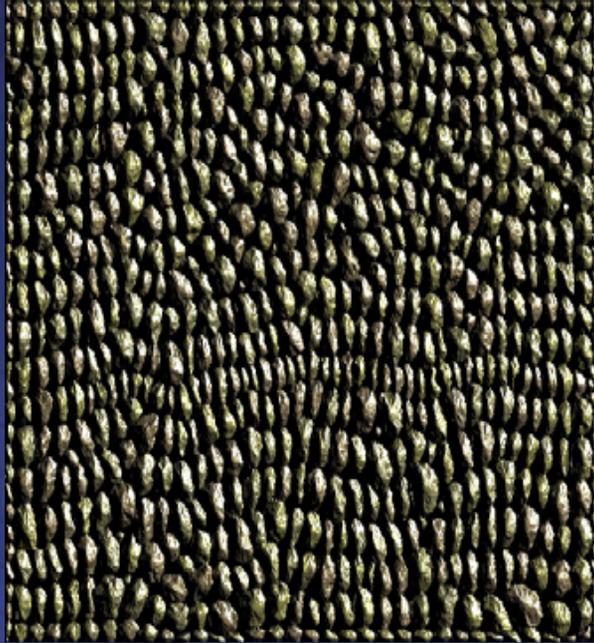


$Ca = 1.5$

$Ca = 2.0$

$Ca = 3.0$

生成例: 棘具合の制御



$$Sc = 0.0$$



$$Sc = 2.0$$



$$Sc = 3.0$$

生成例：盛り上がりの制御



$$T_c = 0.0$$

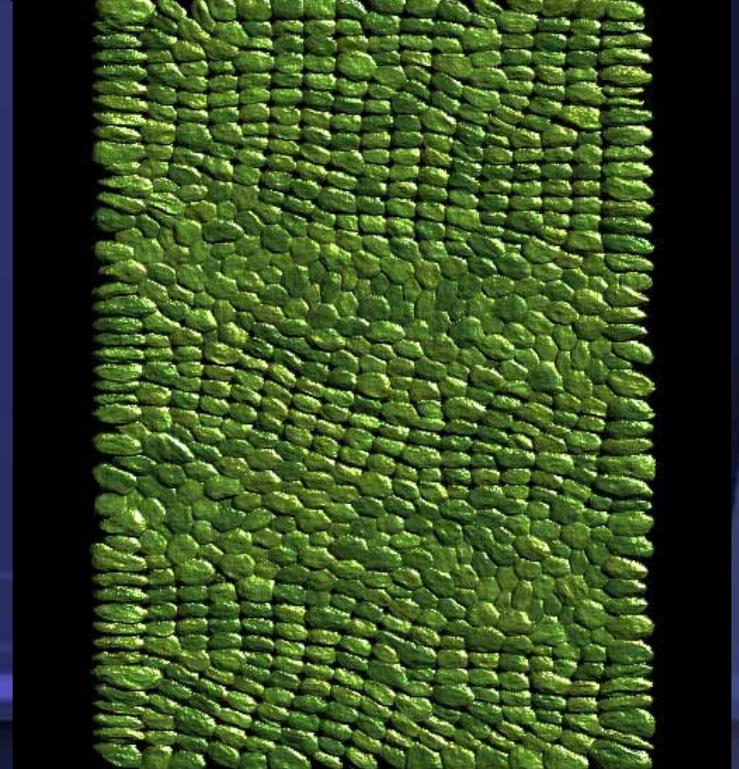
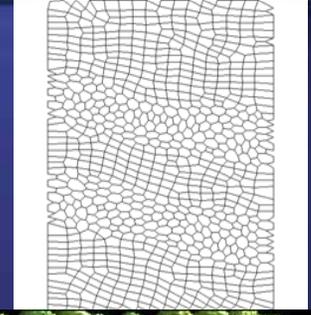
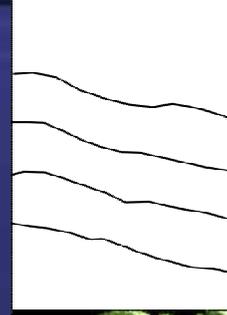
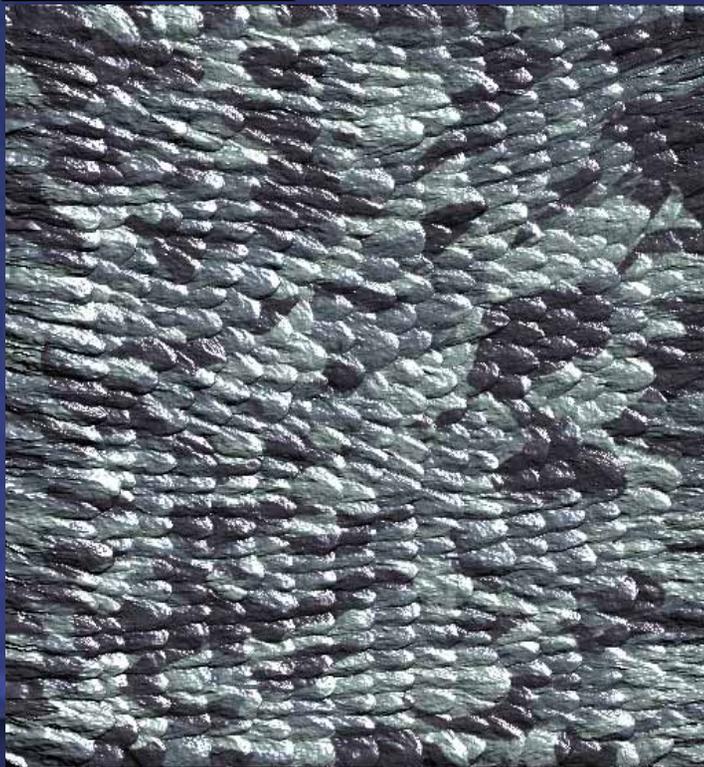
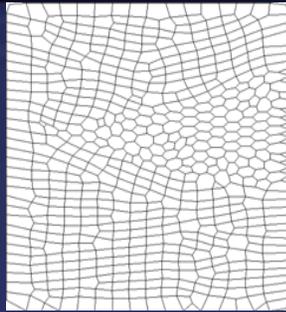
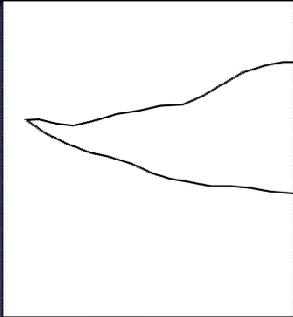


$$T_c = 0.2$$

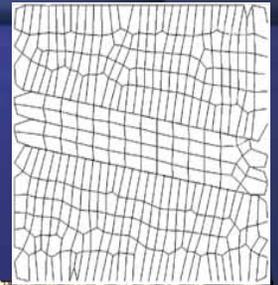
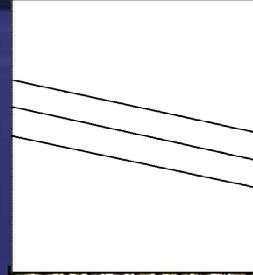
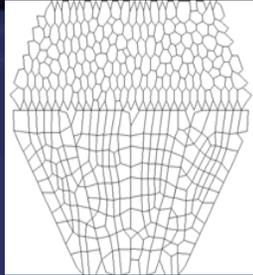
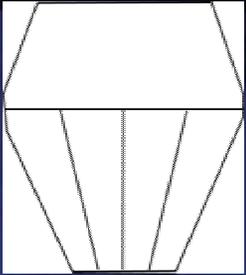


$$T_c = 0.5$$

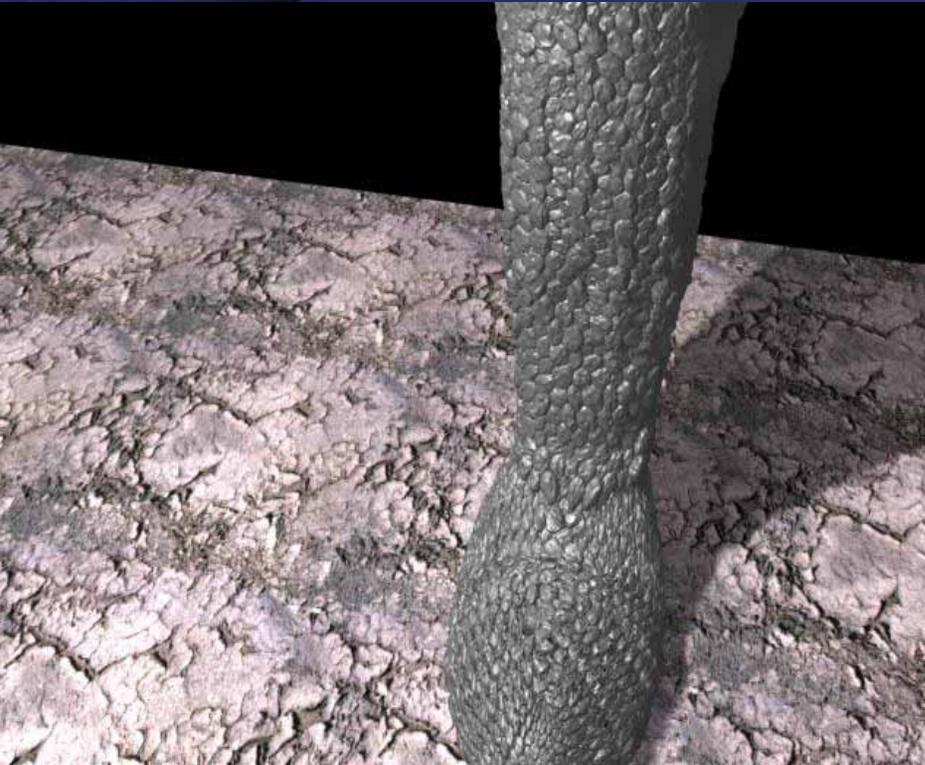
生成例



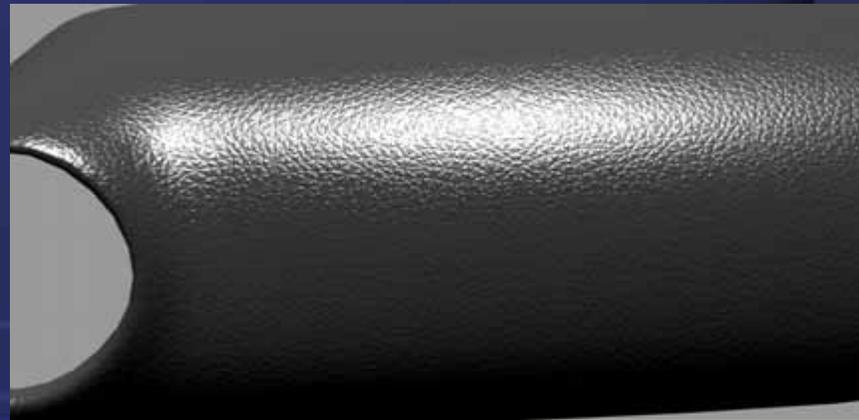
生成例



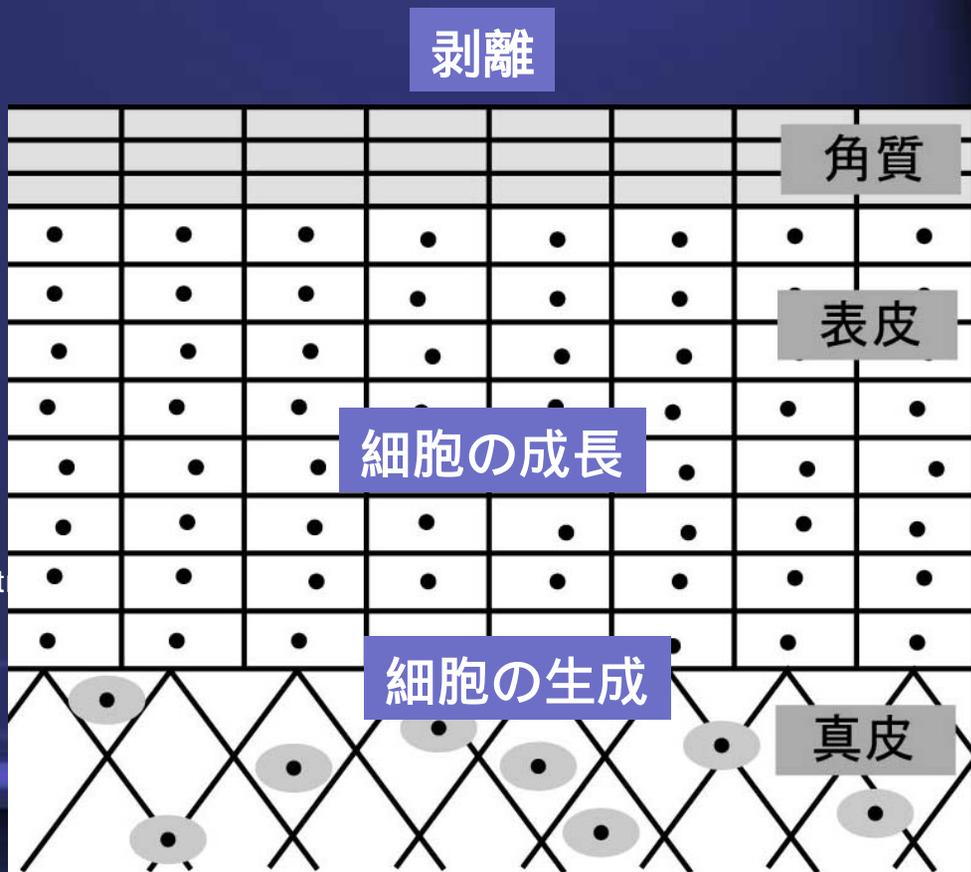
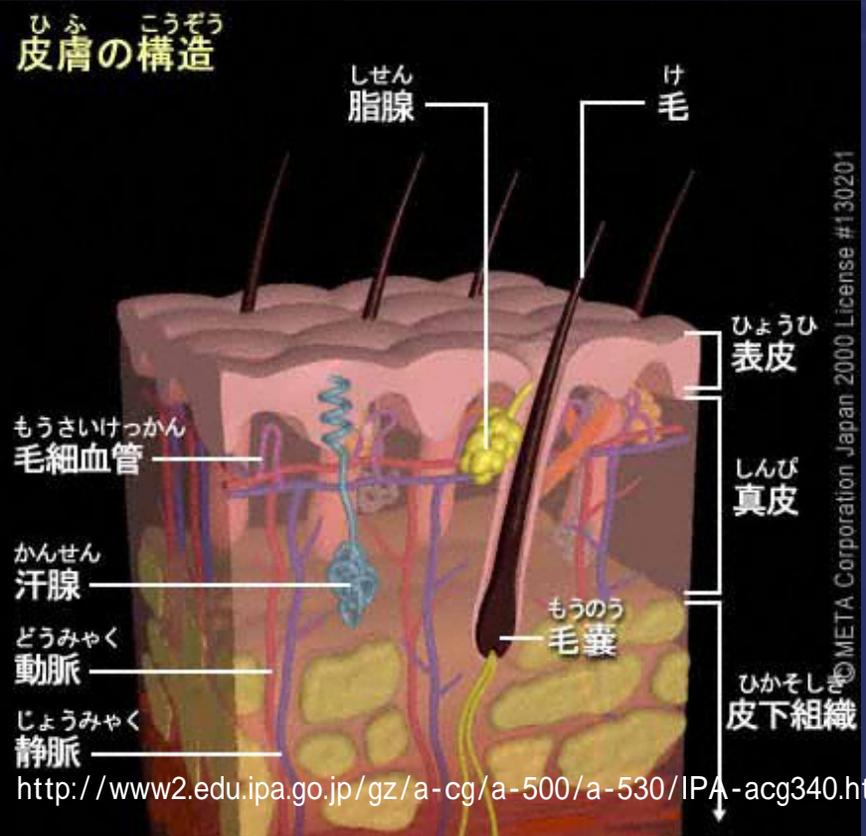
適用例



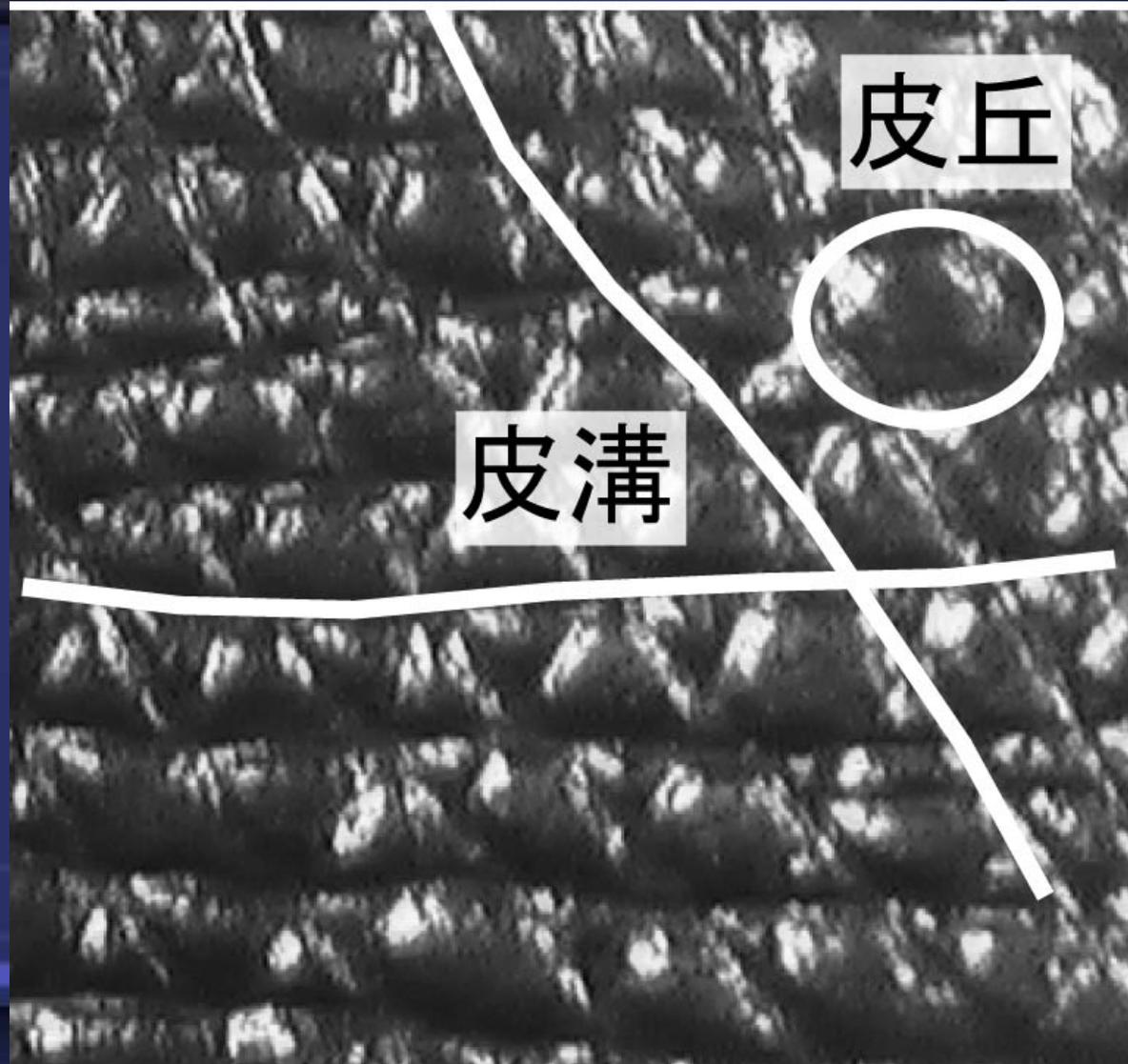
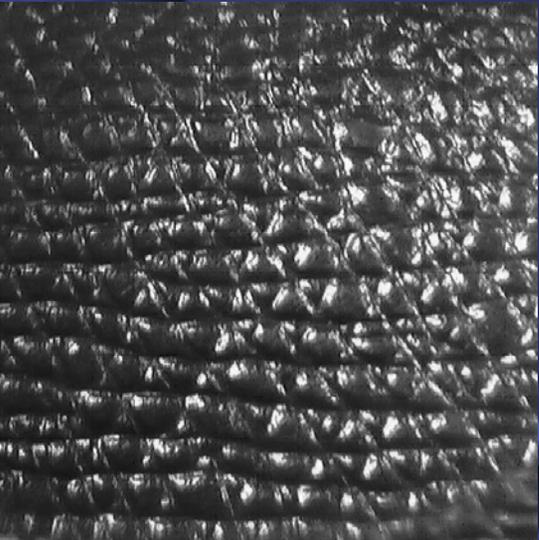
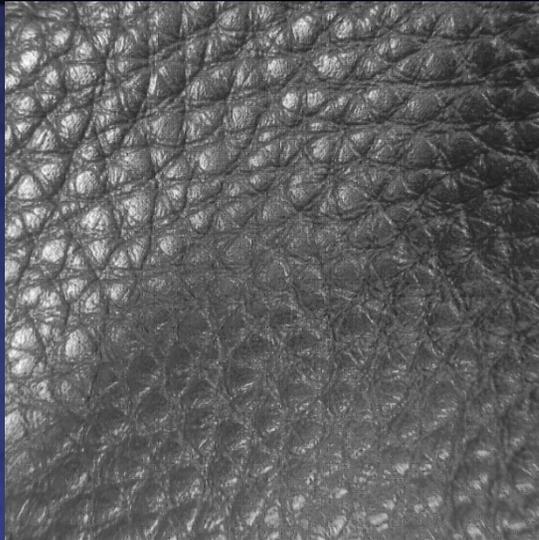
革シボ



皮膚の構造：多層構造

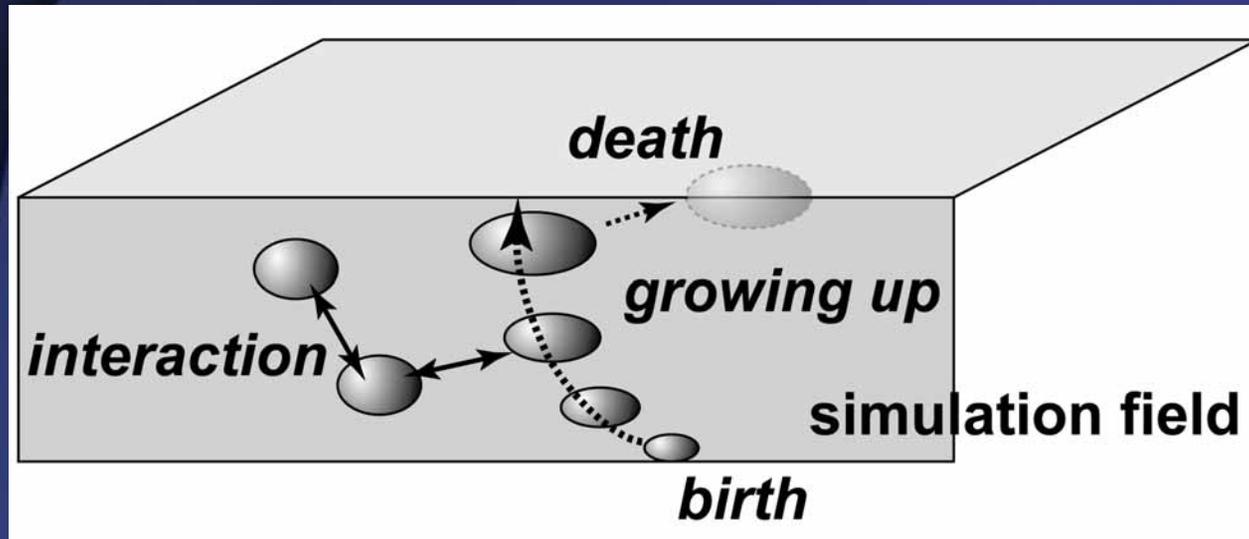


皮膚の構造：皮丘と皮溝



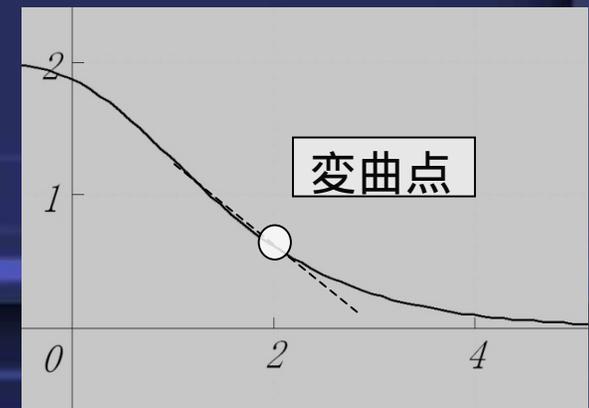
皮革細胞モデル

細胞の成長モデル



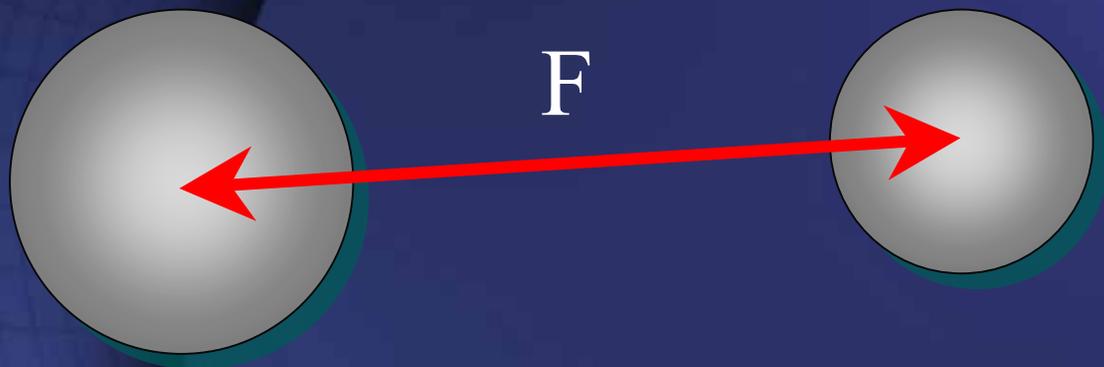
細胞の誕生速度, 大きさ

シグモイド曲線で制御
 生命体の老化を表現

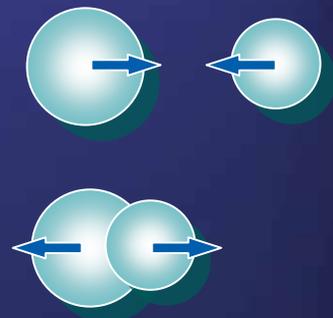


細胞シミュレーション

パーティクルシミュレーション



分子間力が作用



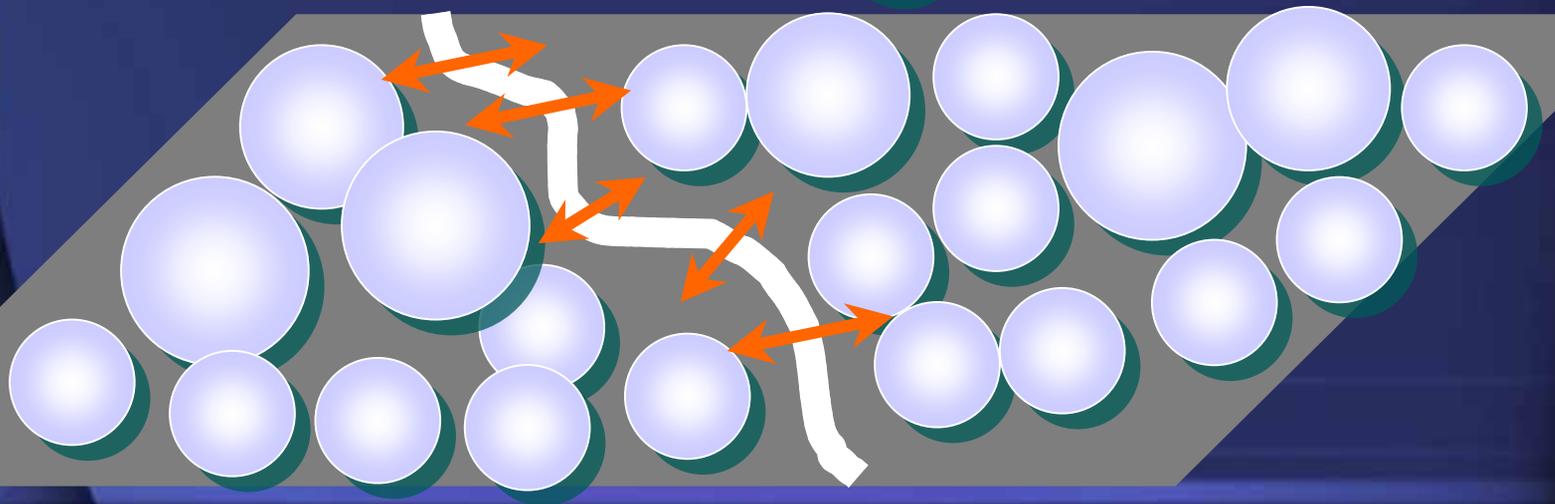
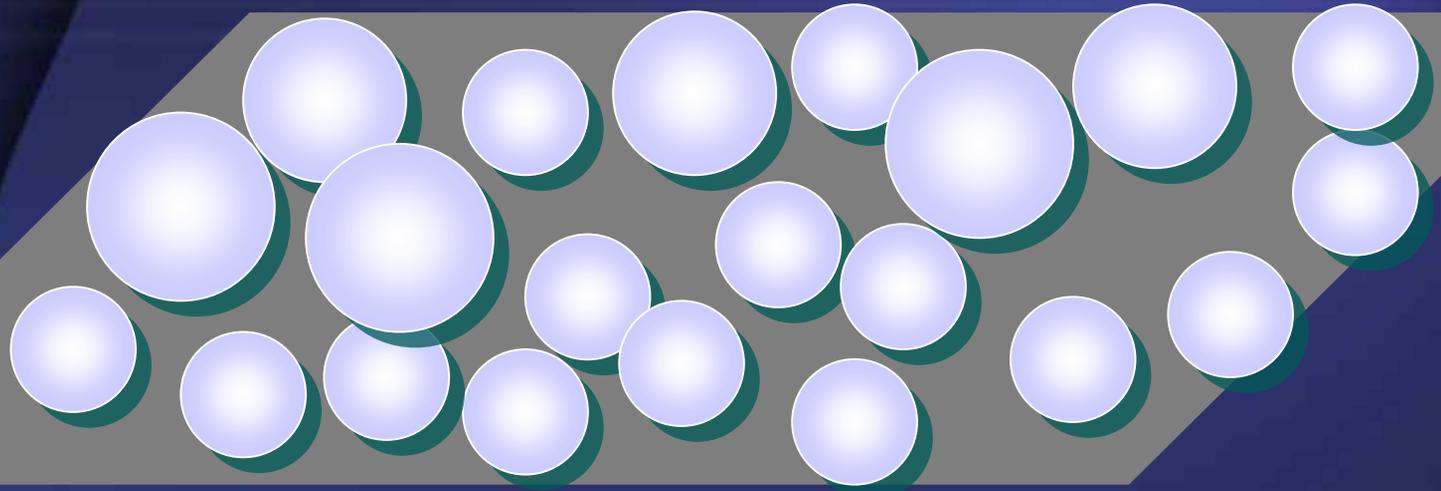
運動方程式

$$m \frac{d^2 x(t)}{dt^2} + c \frac{dx(t)}{dt} = F(t)$$

m: 質量
c: 粘性係数

Runge-Kutta法で数値積分

細胞シミュレーション

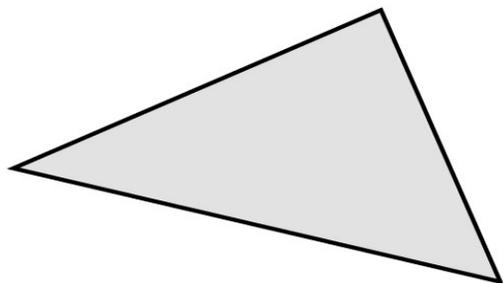


ポテンシャルフィールド

制約条件 = 外力

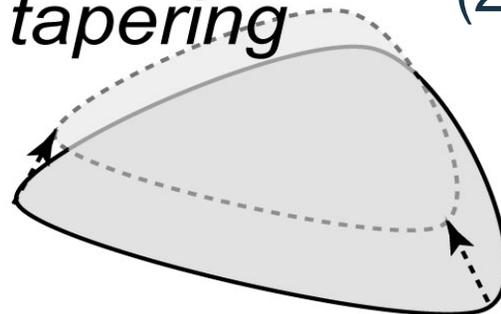
皮革テクスチャの生成：概略

(1) 初期形状

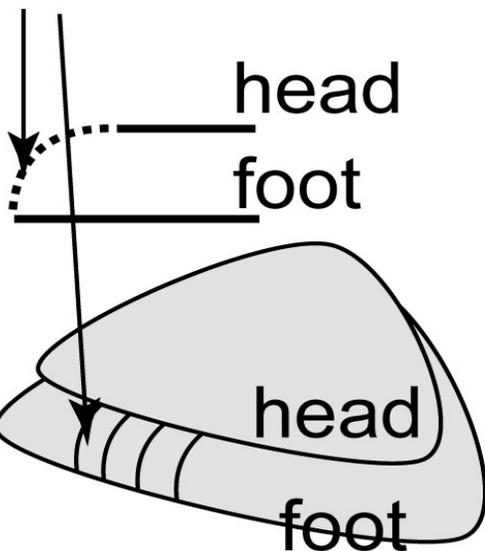


sweep & tapering

(2) 丸め操作

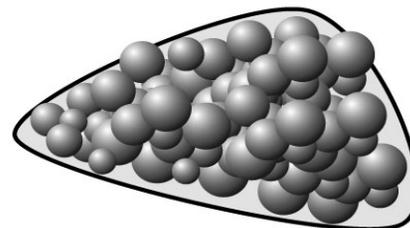


Hermite curve



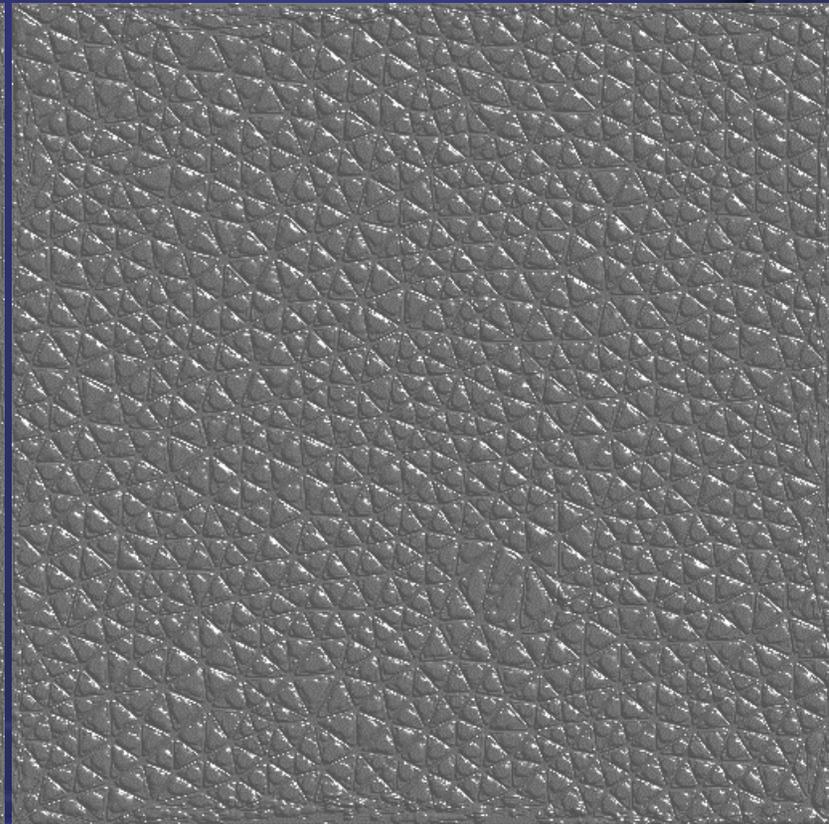
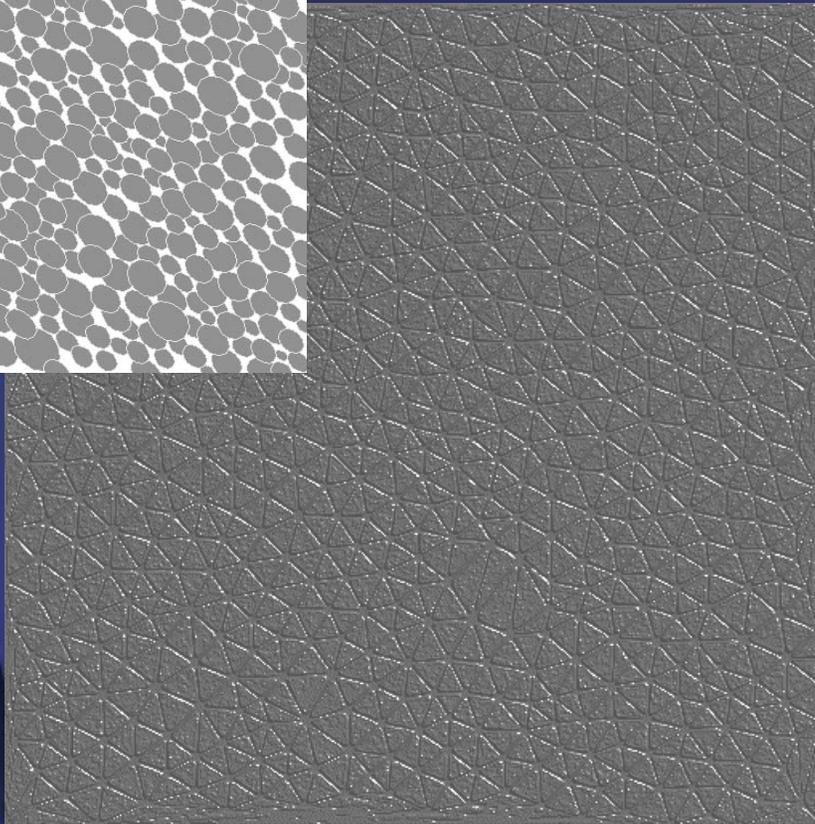
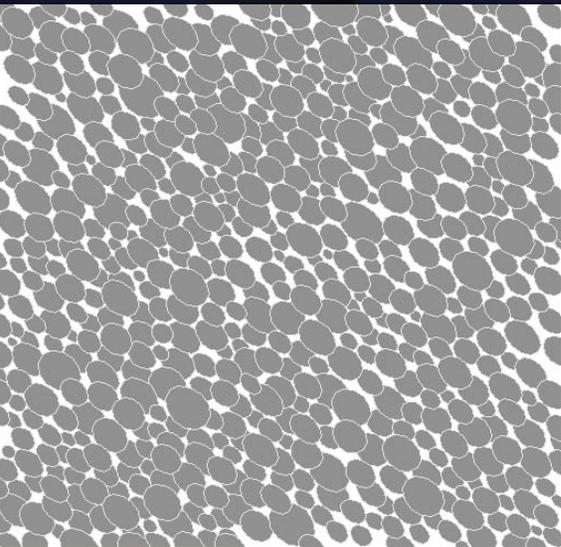
(3) 基底形状

blobs



(4) 表面凹凸

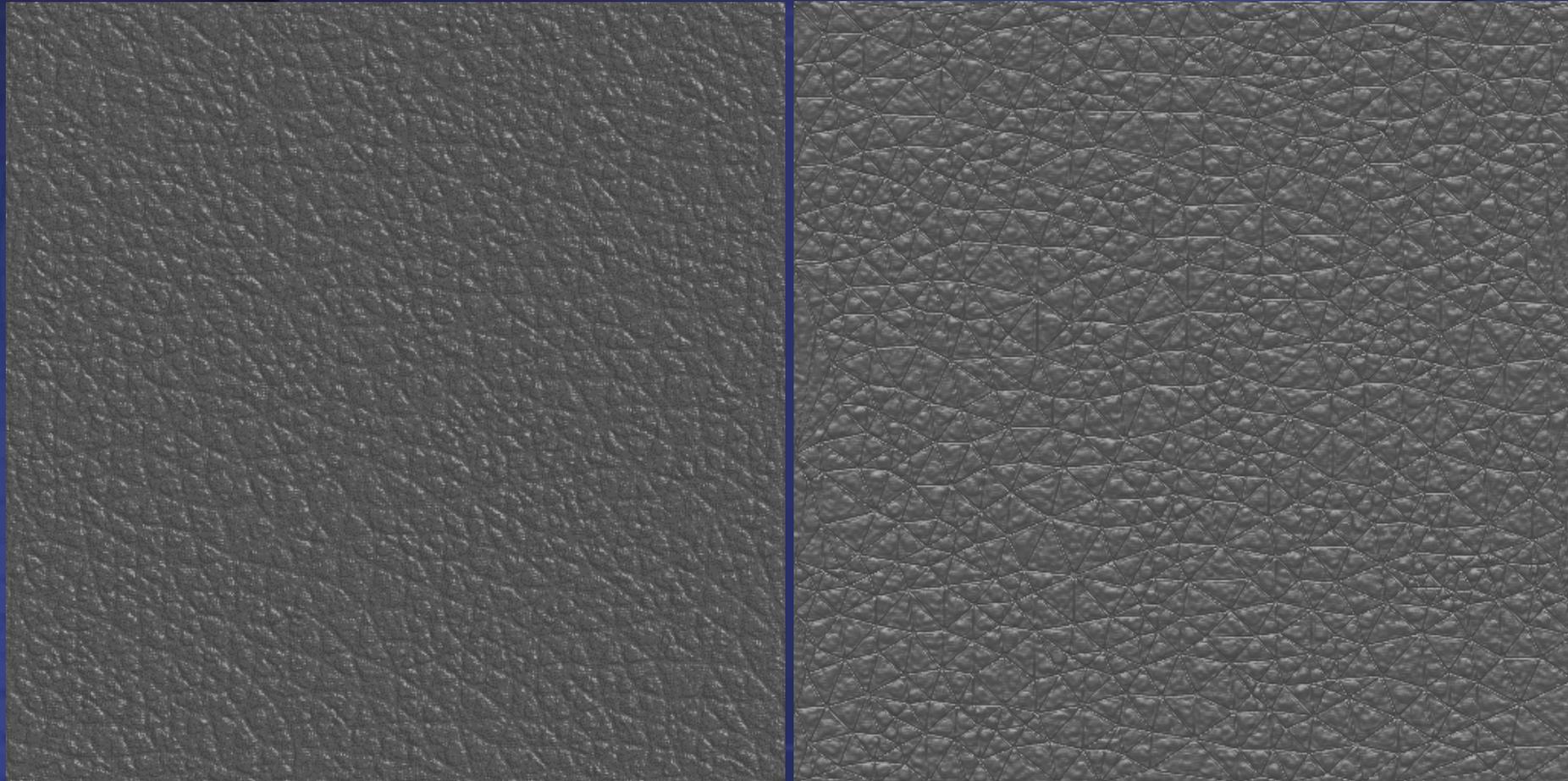
生成結果 # 1



ベース形状のみ

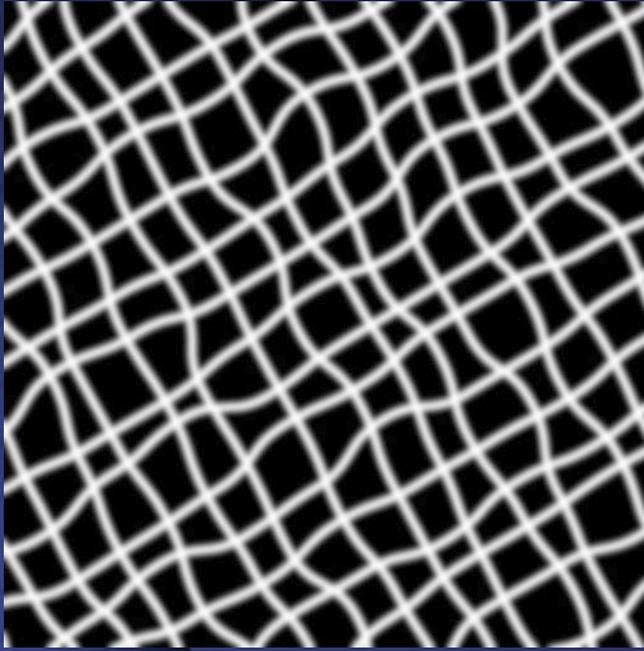
+ メタボール

生成結果 # 2

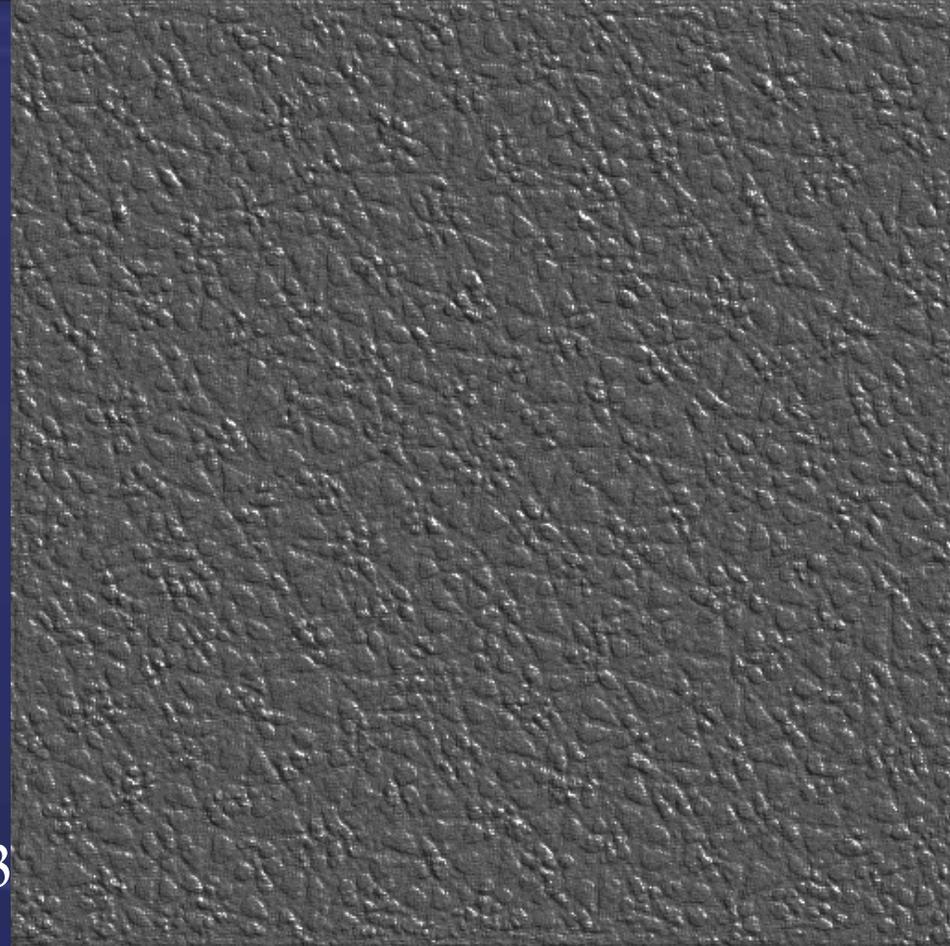


- × 皮溝における複雑なしわの表現
- × マクロな視点で観察できる大きなしわ

生成結果 # 3

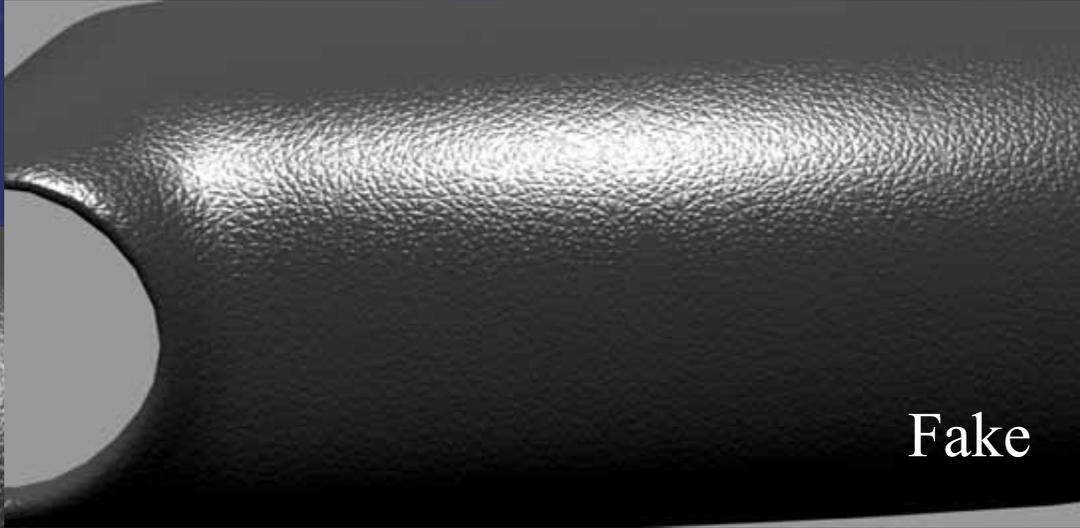
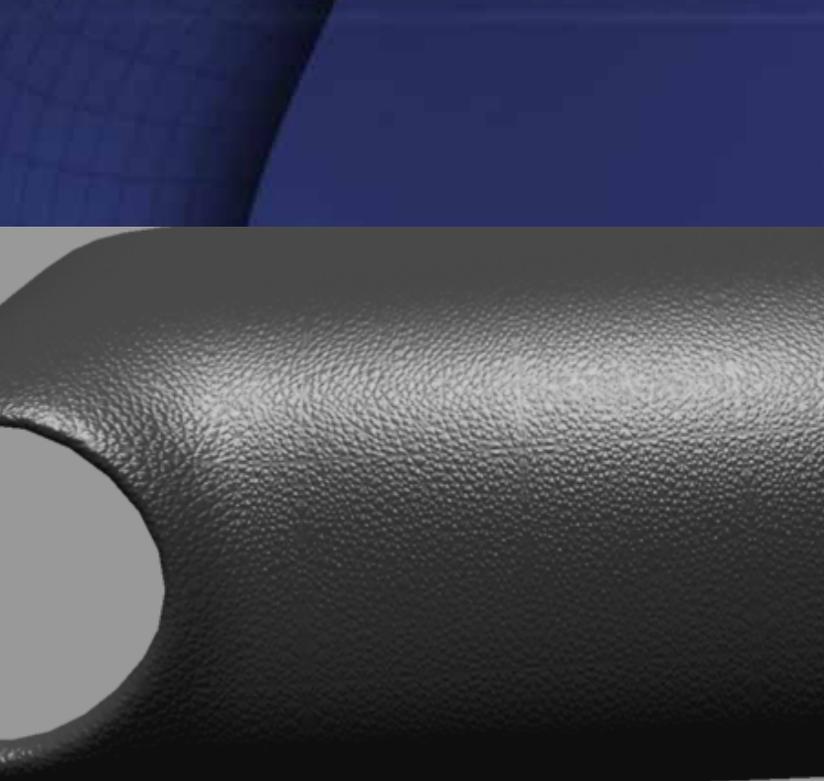


CPU: PentiumM 2GHz
RAM: 1GRAM
Microsoft Visual Studio.NET2003

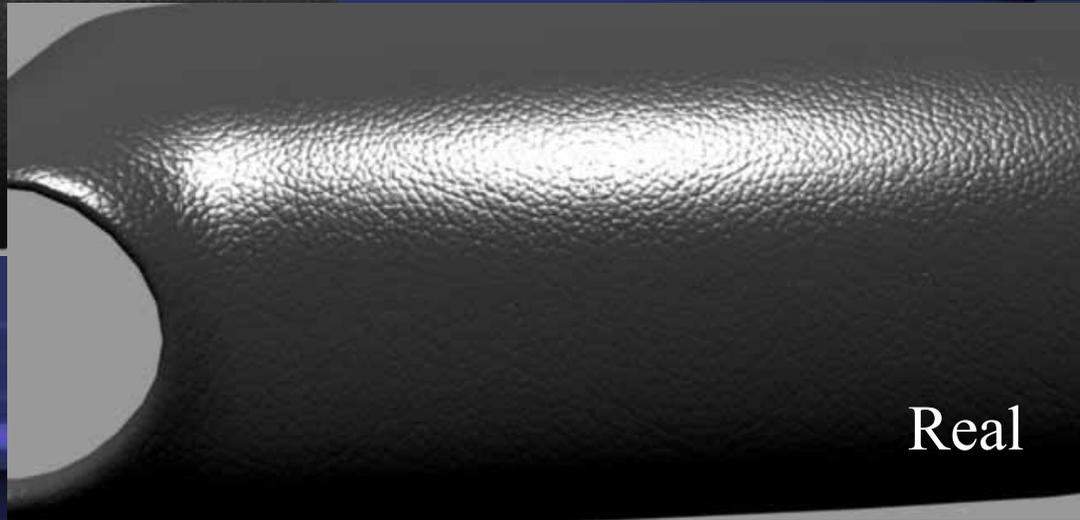


細胞シミュレーション: 約35秒
皮革テクスチャの計算: 約13秒

生成結果 # 4



Fake



Real

Rendered by MAYA

漆工芸



伝統工芸とCG

- 1) 工芸素材のほとんどは**手作業で長時間**かけて制作
試作品を短時間で作成し検証することは困難である。
- 2) 工芸素材の**単価が高い**
試作品を大量に作成することは経済的に馴染まない。
- 3) 新たな商品の開発
色々なバリエーションを短時間で試したい。
- 4) 建築の内装材としての利用
事前に仕上がりを検証したい。

■ CGによるビジュアルシミュレーション

漆工芸の製作過程

素地



木胎, 乾漆など
 木材がほとんど
 布, 紙, 竹など

漆塗

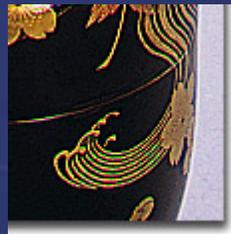


下地 中塗り 上塗り



- ・素地を見せる
- ・塗りと研ぎで覆う

加飾



蒔絵

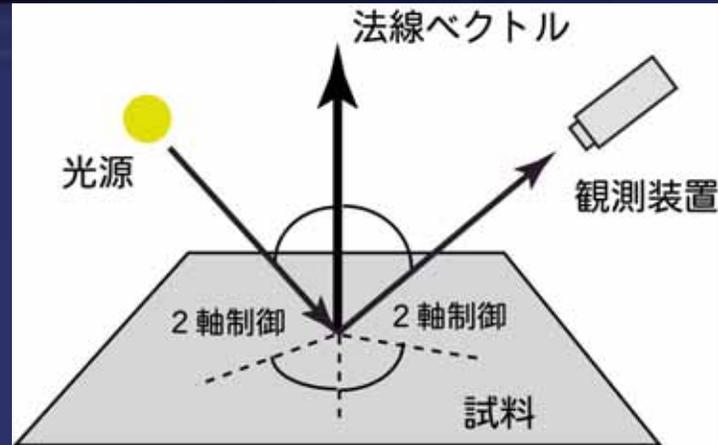
漆が糊
 金などを蒔きつける



沈金

ノミで彫る
 金などを打ち込む

金箔・漆の測定

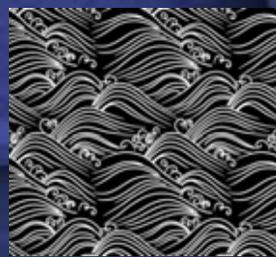
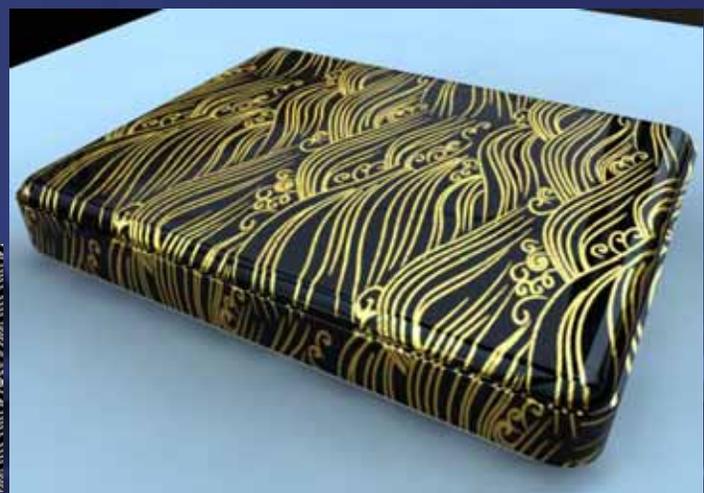
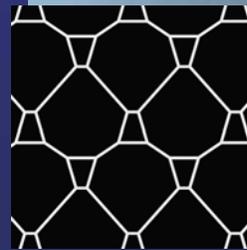
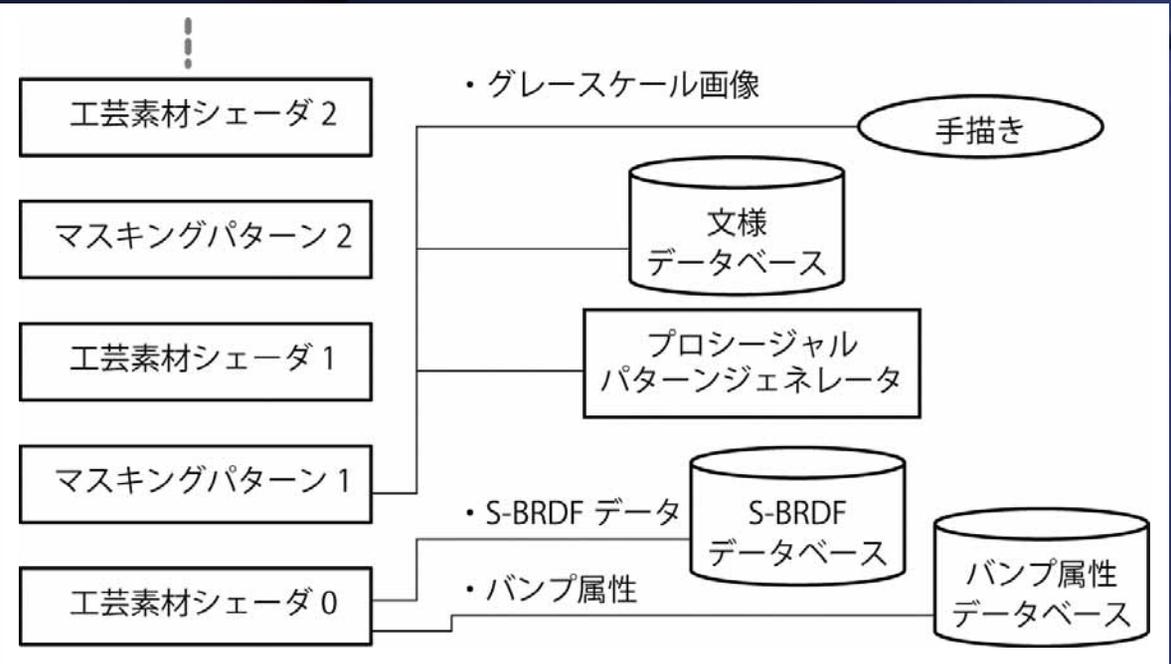


- ディテクタに分光測定器
- NDフィルタを用いた飽和再測定

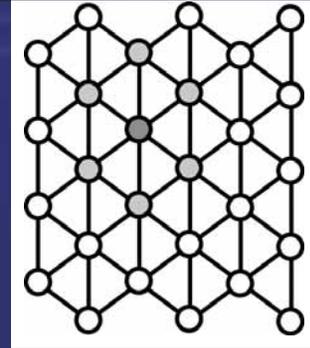
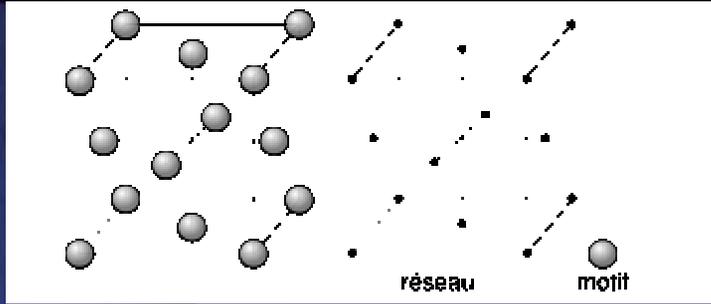


金箔の皺の例

加飾表現

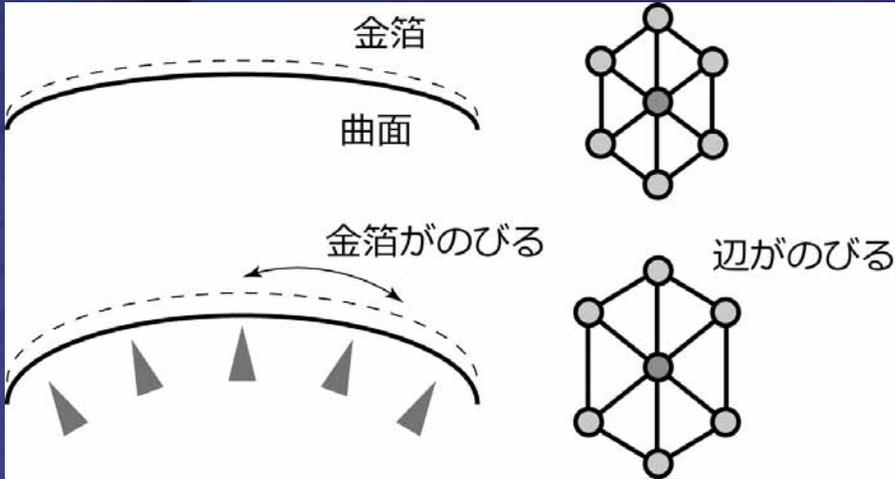


金箔のやぶれ

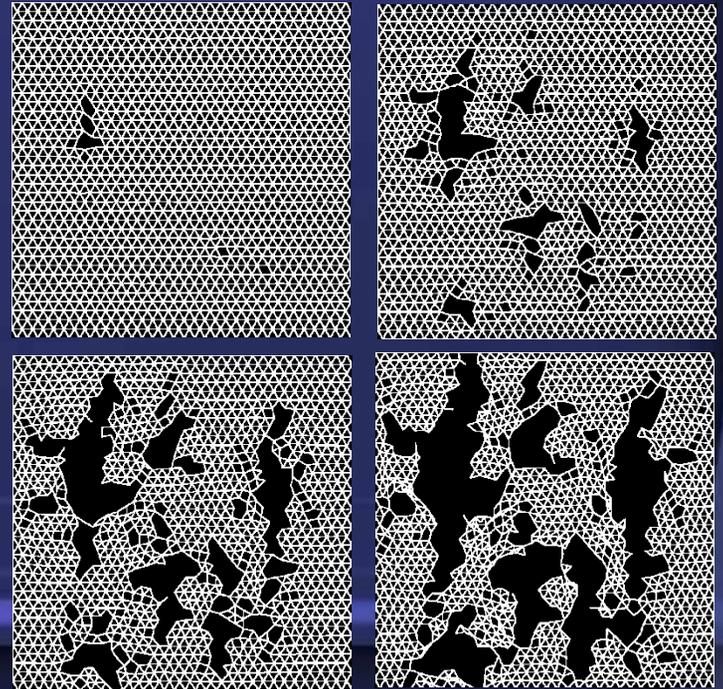


バネメッシュモデル

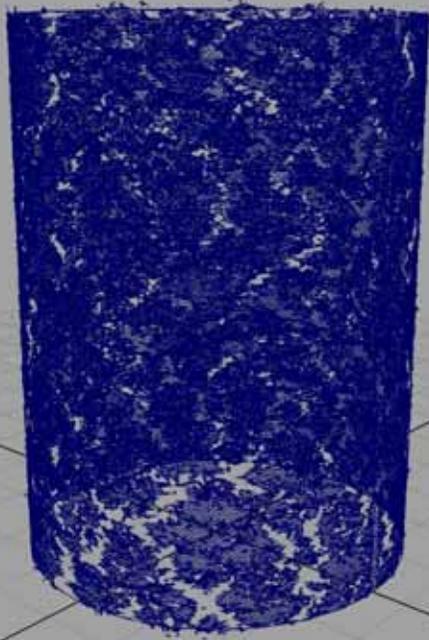
面心立方構造



$$\begin{cases} F = -kx_t \\ \bar{x}_{t+\Delta t} = \bar{x}_t + F/m \cdot \Delta t - C\bar{x}_t \\ x_{t+\Delta t} = x_t + \bar{x}_t \cdot \Delta t \end{cases}$$

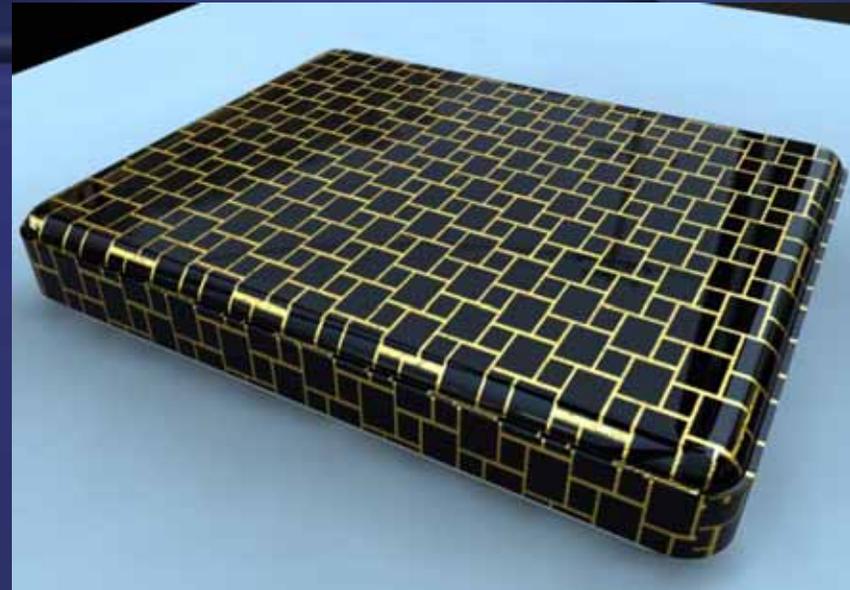


金箔の破れ



230 x 200のメッシュ

加飾の例

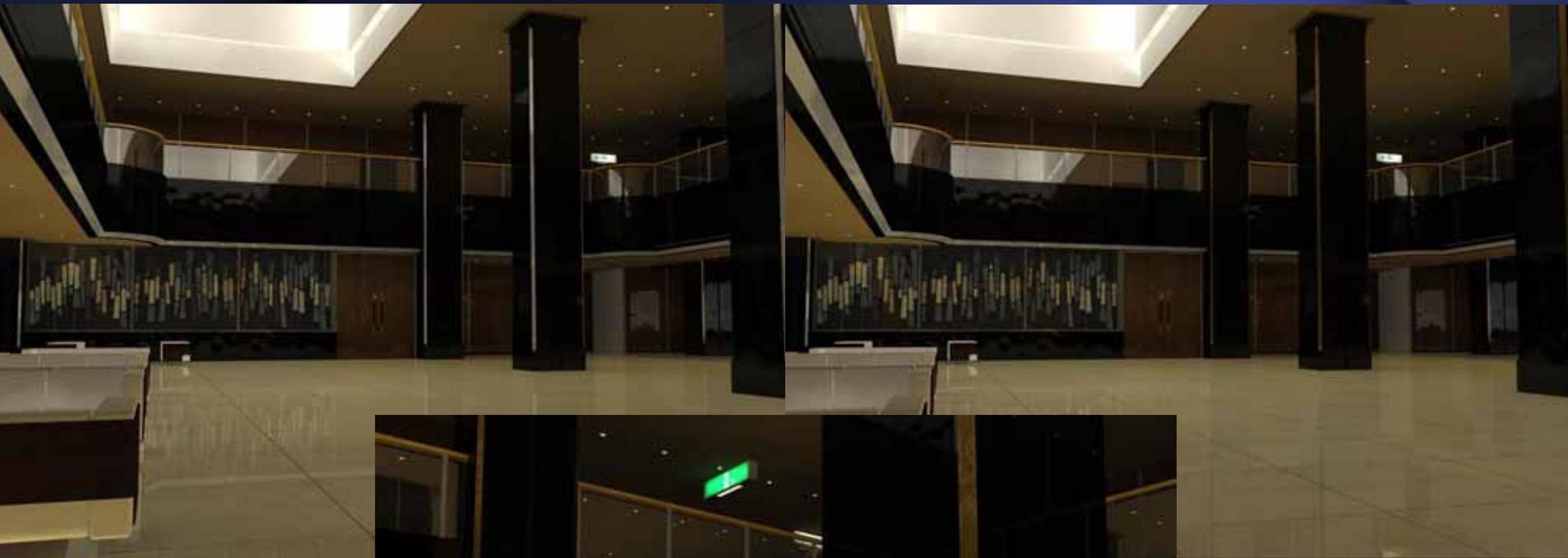


塗り分けの例





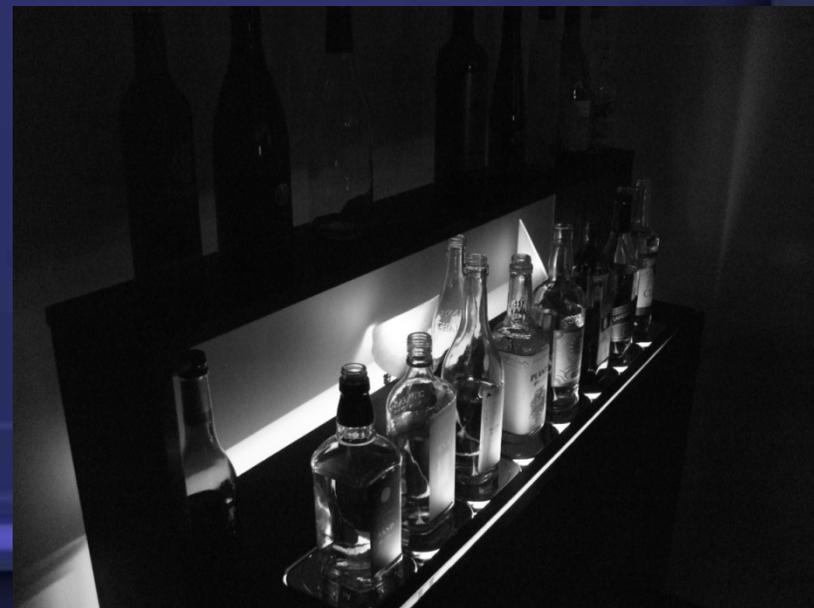
適用例：石川県地場産センター



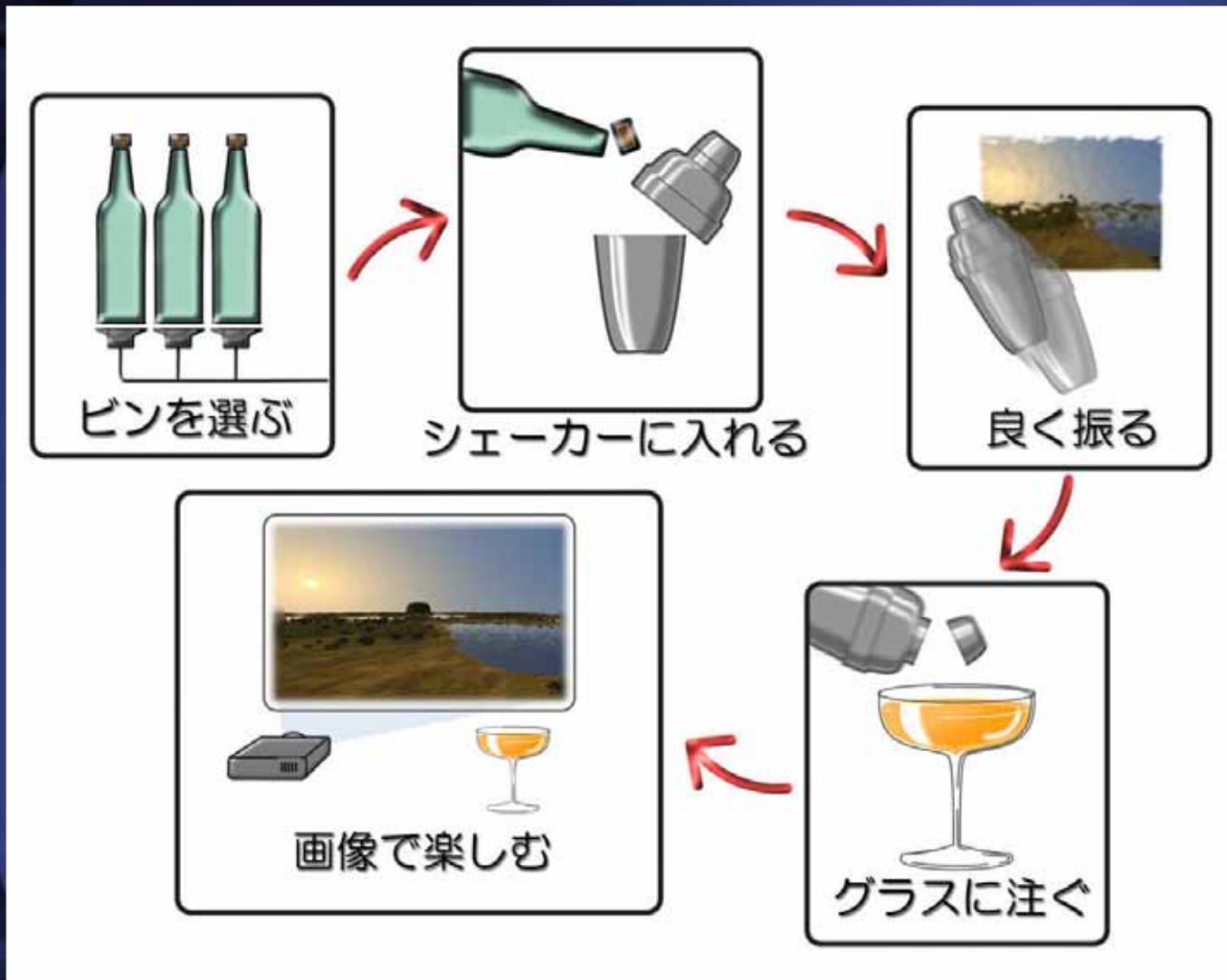
適用例:(株)箔一



風景バーテンダー



システムの概要



風景の要素とボトル

砂

岩

水

植物

太陽

月

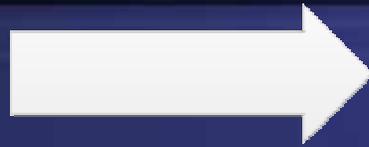
星

雲



オリジナルの風景を作成する

水の配合量



風景要素の割合

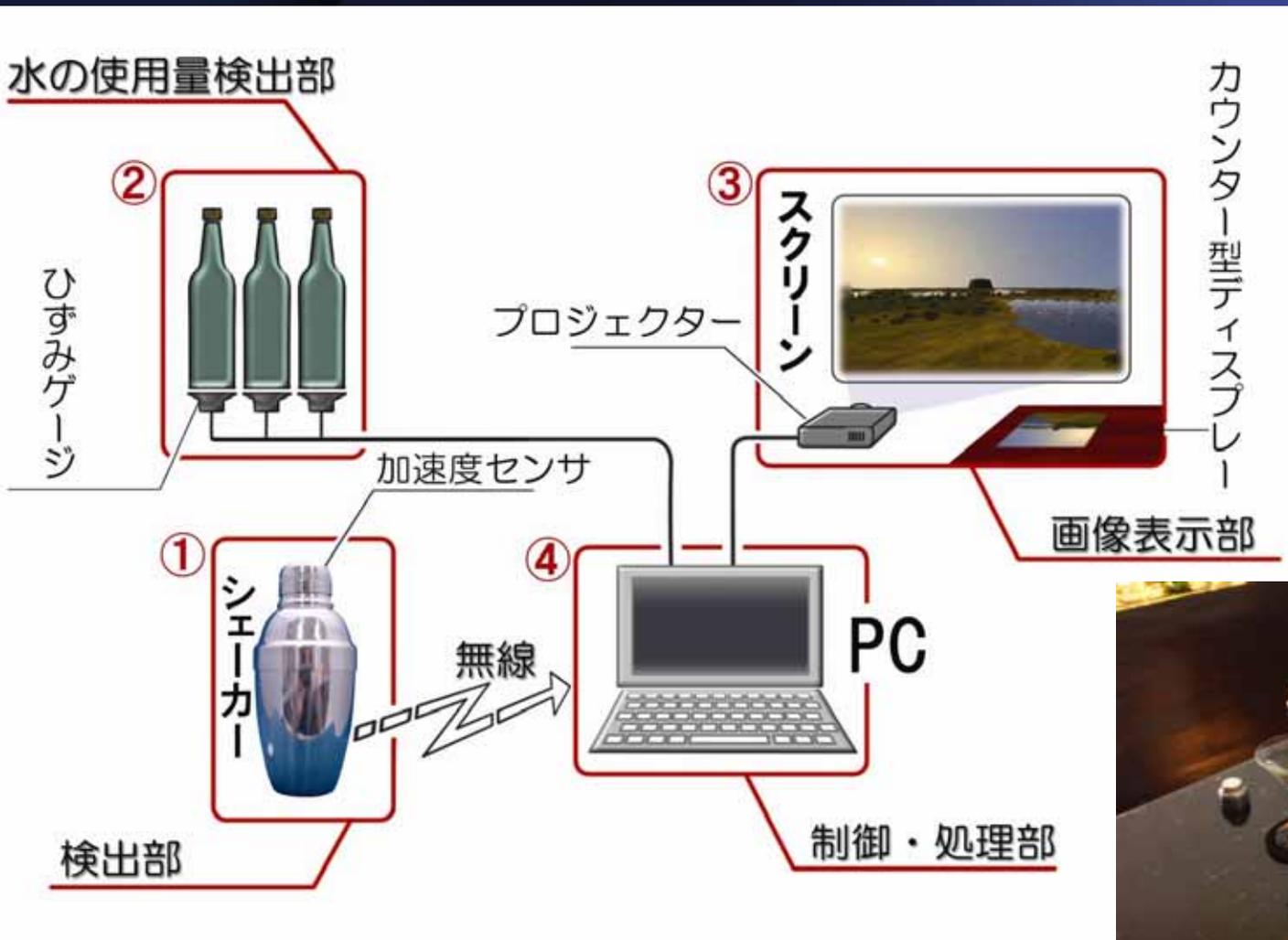
シェーカの振り



地形の起伏
要素の配置



システム構成図



風景生成

入れた材料の比率による変化



太陽多い



月多い



砂多い



岩多い

風景生成

振りによる変化



横振り



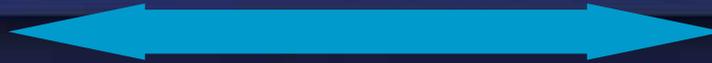
縦振り



放置

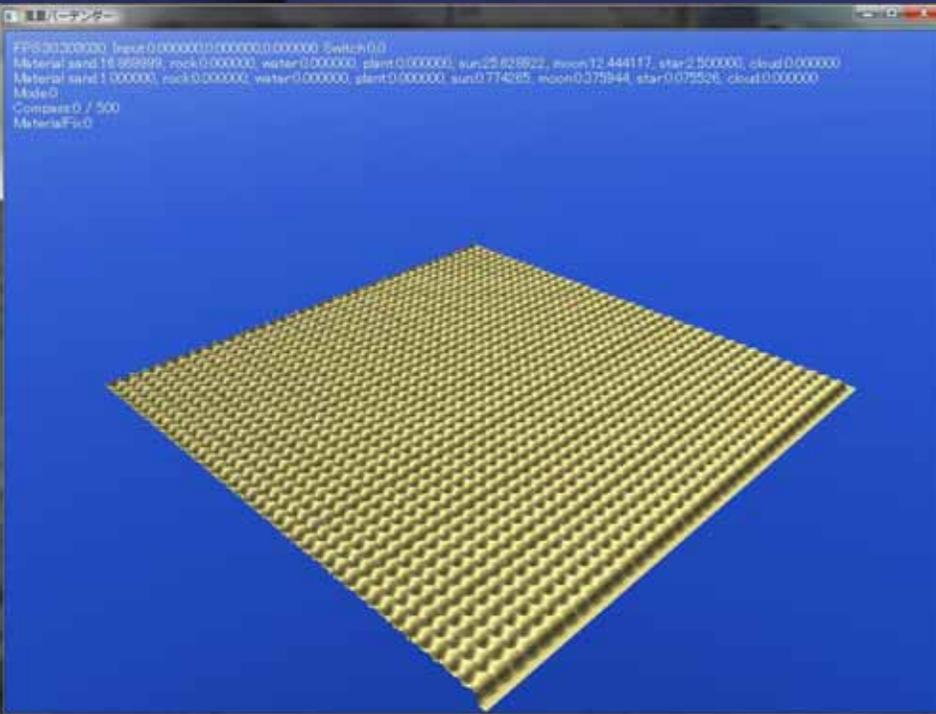


振る

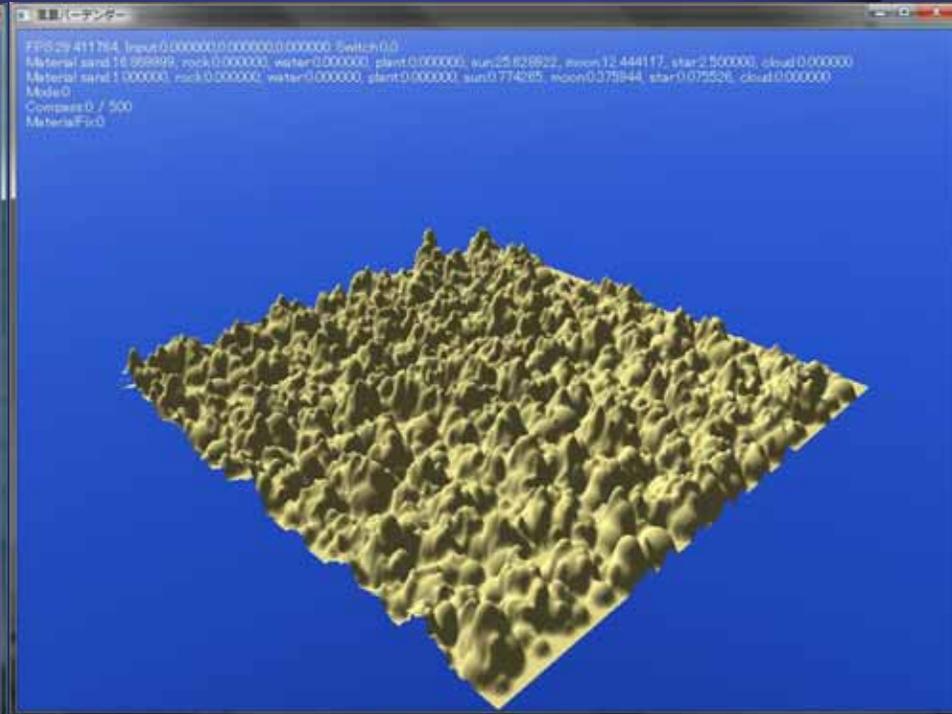


地形の生成

ボールを用いた地形の変形

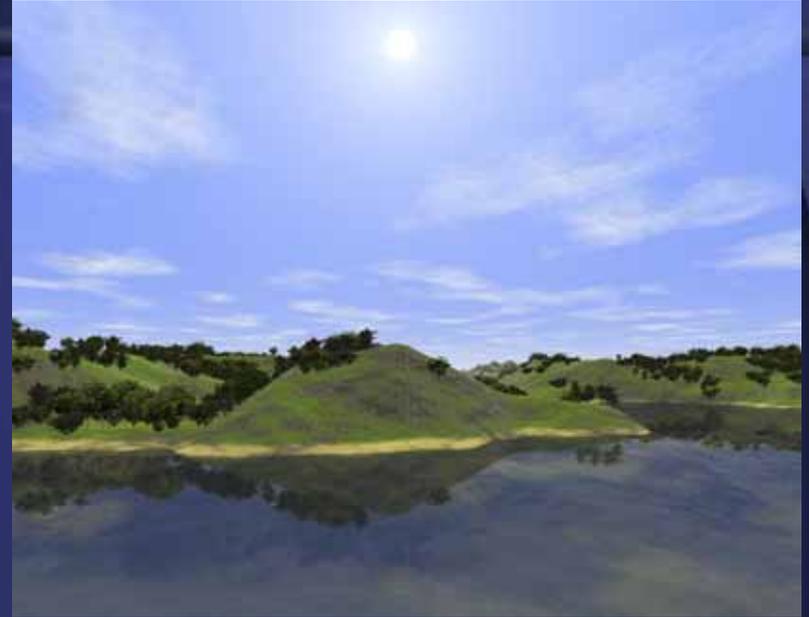


初期配置



縦振りでボールの大きさが分散

生成例



デモムービー

ご静聴ありがとうございました