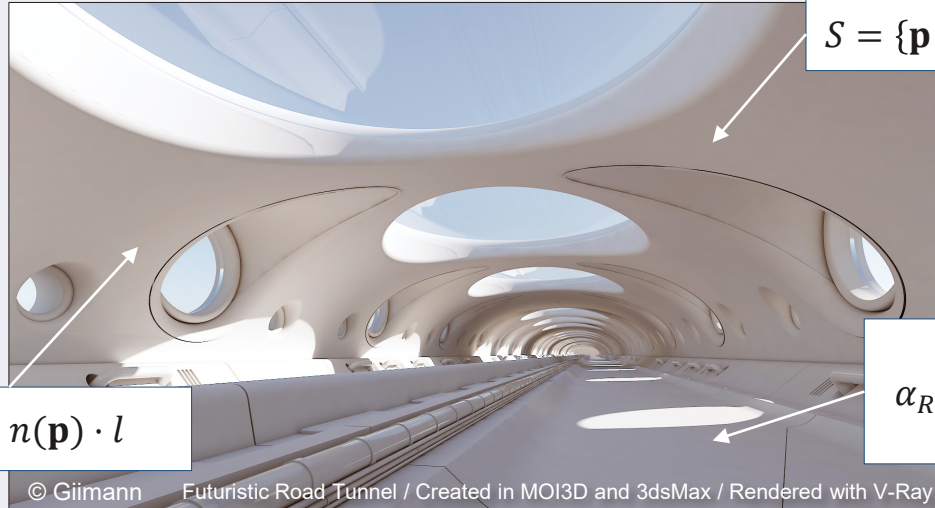


Computer Graphics

From mathematics ...



$$S = \{\mathbf{p} \in \mathbf{R}^3, f(\mathbf{p}) = 0\}$$

$$d(\mathbf{p}) = n(\mathbf{p}) \cdot l$$

$$\alpha_R(\mathbf{p}) \approx \frac{1}{n} \sum_{i=0}^n \delta_i$$

© Giimann Futuristic Road Tunnel / Created in MOI3D and 3dsMax / Rendered with V-Ray

... to the screen

E. Galin
Université Lyon 1

Computer Graphics

Mathematics

Modeling

Color and Texturing

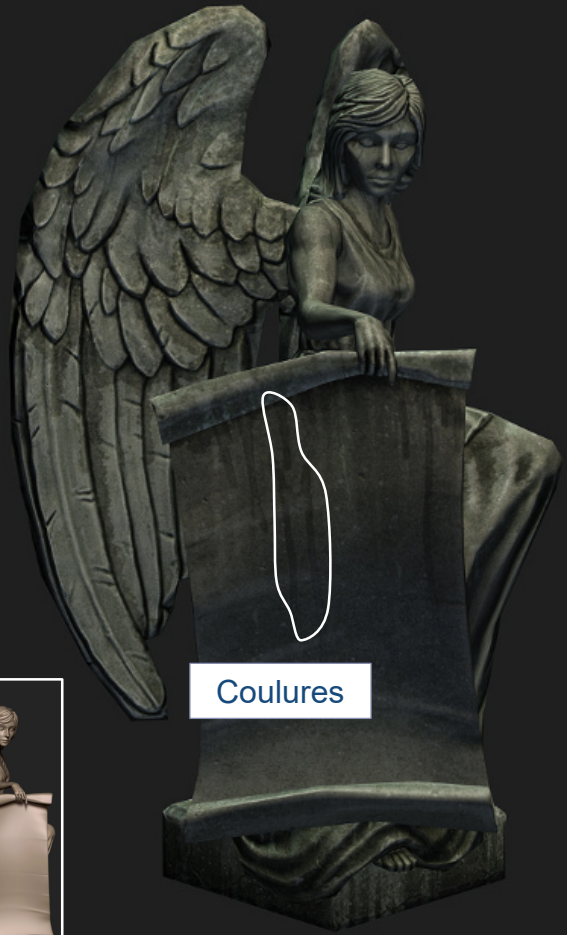
Shading

Realistic Rendering

Acceleration



Vieillessement



Coulures



A photograph of a riverbed composed of numerous smooth, rounded stones. In the center-right, a large, cylindrical structure is built from these stones, protruding above the water level. A white oval highlights the top edge of this structure. A white rectangular box with the text 'Relief de déplacement' is positioned to the right of the oval. The background shows a wide expanse of the stone riverbed under a clear blue sky.

Relief de déplacement

A close-up photograph of a single stone from the riverbed. The stone has a rough, textured surface. A white oval highlights a specific area on the stone's surface. A white rectangular box with the text 'Rugosité de surface' is positioned to the right of the oval.

Rugosité de surface



Introduction

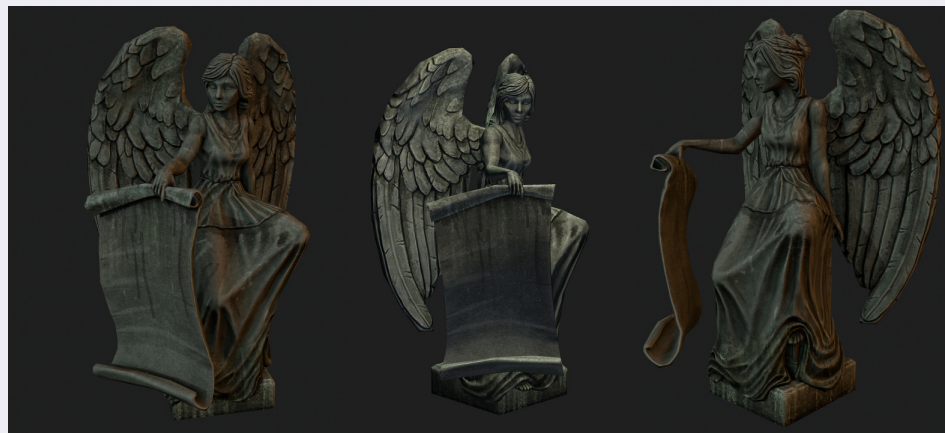
Overview

Mapping

Projection

Procedural

Synthesis



Textures

Définition des matériaux avec propriétés et détails

Placage de texture

Approche **surfactive** : image $I: [0,1]^2 \rightarrow \mathbf{E}$,
correspondance I et S, coordonnées **inverses**

Textures procédurales

Approche **volumique** : **fonctions** $t: \mathbf{R}^3 \rightarrow \mathbf{E}$ et
plongement de l'objet

Vieillessement

Simulation de l'évolution de
dans le temps

Synthèse

Génération à partir
d'exemples

Génération

Approches fréquentielles et
phénoménologiques



eric.galin@liris.cnrs.fr
<http://liris.cnrs.fr/~egalin>

Computer Graphics

Texture mapping

Fondamentaux

Overview

Mapping

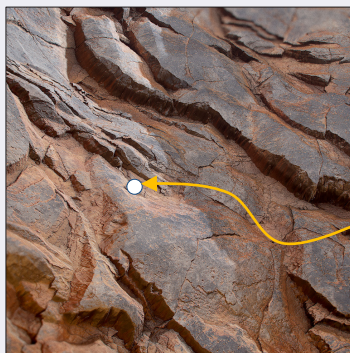
Projection

Procedural

Synthesis

Paramétrisation

Correspondance entre $I: [0,1]^2 \subset \mathbb{R}^2$ et $S \subset \mathbb{R}^3$



Daniel-Thiger – Substance Designer

Texture $I: [0,1]^2 \subset \mathbb{R}^2$
 $\mathbf{q}(u, v)$

Surface $S \subset \mathbb{R}^3$
 $\mathbf{p}(x, y, z)$

Paramétrisation
 $(u, v) = f^{-1}(\mathbf{p})$

Fonction f^{-1} de
paramétrisation inverse

Cas spécifiques simples

Coordonnées (u, v)
de textures

Données explicites
 $(u_k, v_k) = f^{-1}(\mathbf{p}_k)$ pour les
Sommets d'un maillage



eric.galin@liris.cnrs.fr
http://liris.cnrs.fr/~egalin

Paramétrisation analytique

Overview

Mapping

Projection

Procedural

Synthesis

Cylindre et sphère

Coordonnées $(u, v) \in [0, 1]^2$

Inversion des coordonnées cylindriques r, θ, z et sphériques r, θ, φ

Cylindre

$$x = r \cos \theta$$

$$y = r \sin \theta$$

$$z = z$$

Inverse

$$u = \text{atan2}(y, x)/2\pi$$

$$v = z/h$$

Sphère

$$x = r \cos \theta \cos \varphi$$

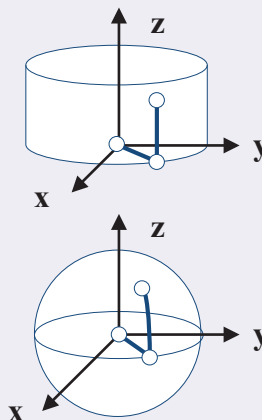
$$y = r \sin \theta \cos \varphi$$

$$z = r \sin \varphi$$

Inverse

$$u = \text{atan2}(y, x)/2\pi$$

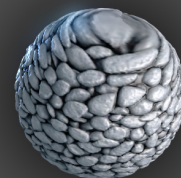
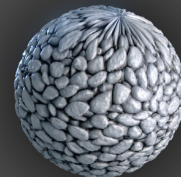
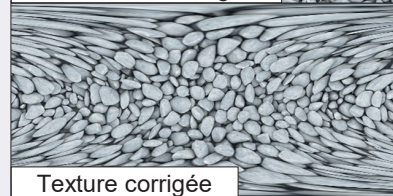
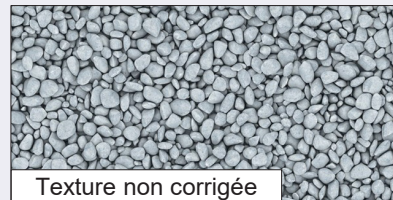
$$v = \sin^{-1} z/r$$



Distorsion inverse

Compensation des distorsions

Déformation **préalable** de l'image I



eric.galin@liris.cnrs.fr

http://liris.cnrs.fr/~egalain

Coordonnées inverses

Overview

Mapping

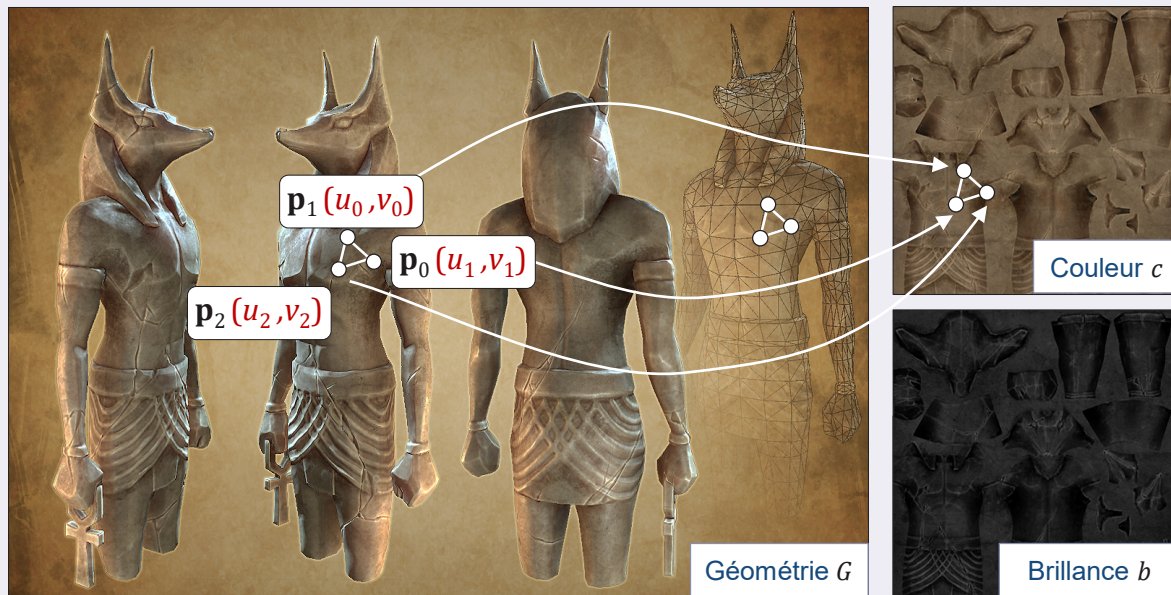
Projection

Procedural

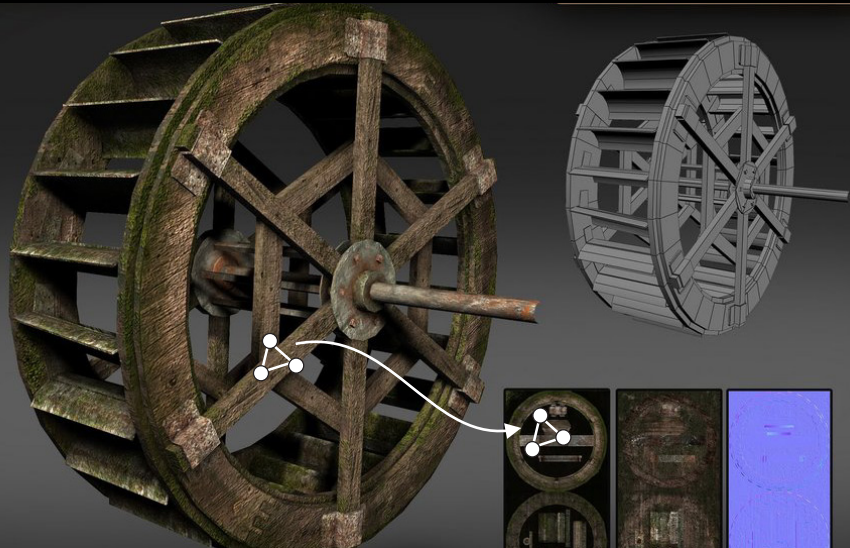
Synthesis

Maillages

Géométrie G augmentée des coordonnées (u, v) pour les sommets \mathbf{p}
I peut stocker la couleur c , la variation de normale $\delta\mathbf{n}$, la brillance b ...



eric.galin@liris.cnrs.fr
<http://liris.cnrs.fr/~egalin>



Déplacement

Overview

Mapping

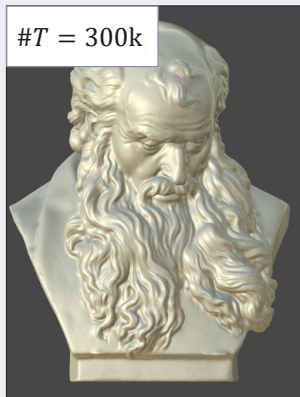
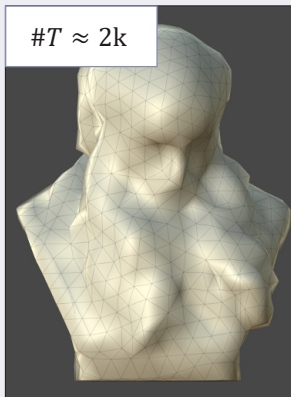
Projection

Procedural

Synthesis

Maillages

Géométrie grossière \tilde{G} raffinée et déplacée avec une carte de hauteur selon les coordonnées (u, v)
 I stocke une carte de hauteur D selon les coordonnées (u, v)



Construction

Géométrie \tilde{G} obtenue par simplification de maillage
Déplacement d par construction

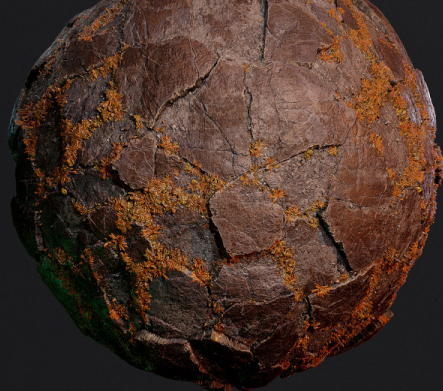
$$d(u, v) = |\tilde{G}(u, v) - H(u, v)|_{\mathbf{n}(u, v)}$$

Différence de hauteur

Direction de la normale $\mathbf{n}(u, v)$



eric.galin@liris.cnrs.fr
<http://liris.cnrs.fr/~egalin>



Transparence

Overview

Mapping

Projection

Procedural

Synthesis

Imposteurs

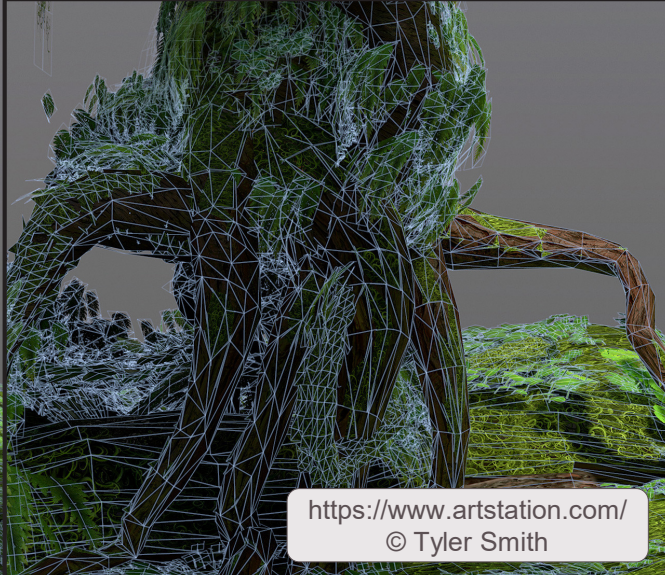
Simplification extrême de la géométrie $G \rightarrow Q$ à un quadrangle ou un maillage simple $G \rightarrow S$

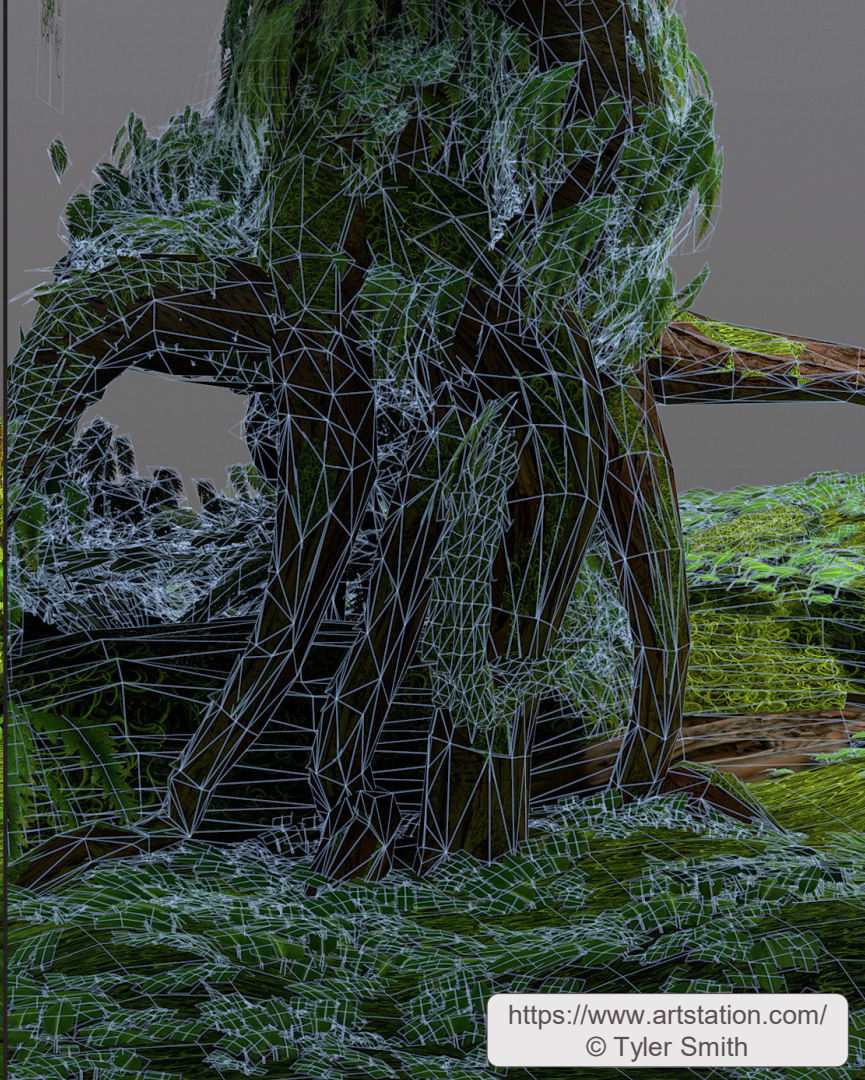
Approximation de la forme géométrique à l'aide de la transparence α dans I



eric.galin@liris.cnrs.fr

<http://liris.cnrs.fr/~egalin>





Computer Graphics

Texture projection

Projections

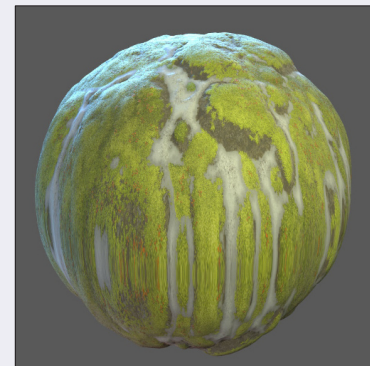
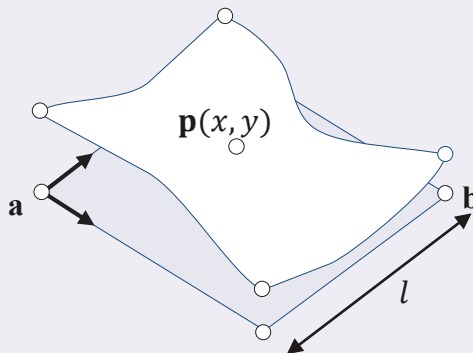
- Overview
- Mapping
- Projection
- Procedural
- Synthesis

Projection orthogonale

Calcul direct
Distorsions sur fortes pentes

Planaire
 $u = x/l$
 $v = y/l$

Couleur
 $c = t(u, v)$



Tri planar mapping

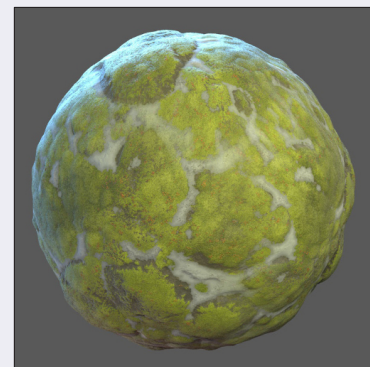
Pas de paramétrisation de S
Projections orthogonales combinées
Pondération selon les composantes de \mathbf{n}

Tri – Planaire
 $u = x/l$
 $v = y/l$
 $w = z/l$

$$c = \frac{t(u, v)(\mathbf{n}_z)^\mu + t(u, w)(\mathbf{n}_y)^\mu + t(v, w)(\mathbf{n}_x)^\mu}{(\mathbf{n}_z)^\mu + (\mathbf{n}_y)^\mu + (\mathbf{n}_x)^\mu}$$

Degré de mélange μ

Moyenne pondérée



eric.galin@liris.cnrs.fr
<http://liris.cnrs.fr/~egalain>

Projections

Overview

Mapping

Projection

Procedural

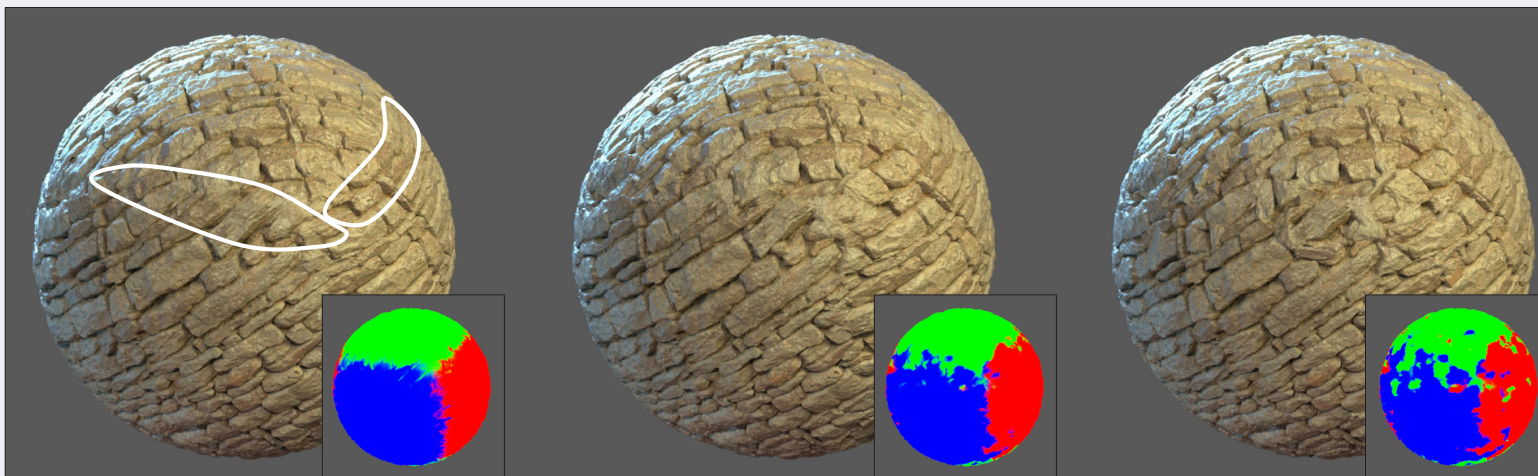
Synthesis

Réduction des artefacts

La région de mélange frontière est parfois trop floue (γ faible) ou trop visible (γ élevé)
Réduction de l'effet de seuil avec un bruit n ou une turbulence t sur les coefficients

$$c = \frac{t(u, v)\alpha + t(u, w)\beta + t(v, w)\gamma}{\alpha + \beta + \gamma}$$

$$\alpha = (\mathbf{n}_x)^\mu + n(\mathbf{p})$$



eric.galin@liris.cnrs.fr
<http://liris.cnrs.fr/~egalain>

Computer Graphics

Procedural textures

Classification

Overview

Mapping

Projection

Procedural

Synthesis

Définition

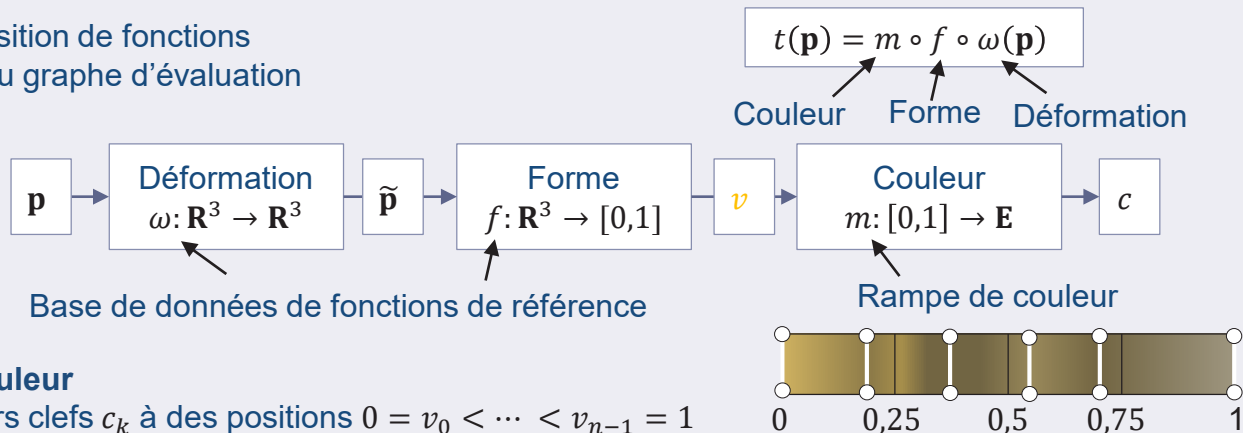
Texture calculée par une procédure $t: \mathbf{R}^3 \rightarrow \mathbf{E}$

Couleur : $\mathbf{E} = [0,1]^3$, autres paramètres : transparence α , brillance, réflectance...

Construction

Composition de fonctions

Arbre ou graphe d'évaluation



Rampe de couleur

Couleurs clefs c_k à des positions $0 = v_0 < \dots < v_{n-1} = 1$

Calcul de $m(v)$ par interpolation

For $k \in [0, n-2]$
If $v_k \leq v \leq v_{k+1}$ break loop
Calculate interpolant \tilde{v}
Compute $c = (1 - \tilde{v})c_k + \tilde{v}c_{k+1}$

$$\tilde{v} = \frac{v - v_k}{v_{k+1} - v_k} \in [0,1]$$

Couleurs clefs c_k
Interpolation



eric.galin@liris.cnrs.fr
http://liris.cnrs.fr/~egalin

Formes

Overview

Mapping

Projection

Procedural

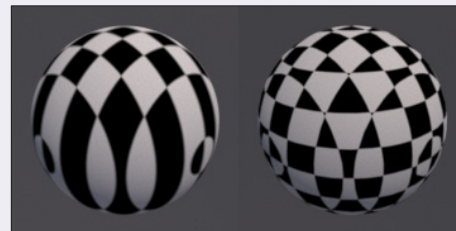
Synthesis

Damier

Dans l'espace, attribuer deux couleurs a ou b selon \mathbf{p}

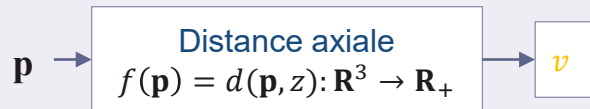


```
int Checker ( vec3 p ) {  
    return mod( int ( p.x ) + int ( p.y ) + int ( p.z ), 2 );  
}
```

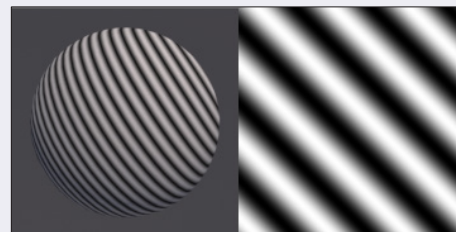


Cylindres concentriques

Feuillets concentriques selon un axe z



```
float Axial ( vec3 p ) {  
    float r = length ( p.xy );  
    return 0.5 + 0.5 * cos ( 3.1416 * r );  
}
```



eric.galin@liris.cnrs.fr

<http://liris.cnrs.fr/~egalin>

Fonctions de bruit

Overview

Mapping

Projection

Procedural

Synthesis

Caractérisation

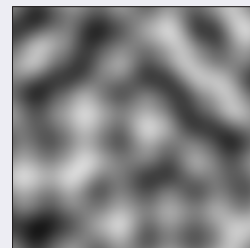
Distribution aléatoire de points \mathbf{c}_k et convolution [Lewis1989]

$$f(\mathbf{p}) = \sum_k h * \delta_k(\mathbf{p})$$

Convolution Noise

$$\delta_k(\mathbf{p}) = \delta(\mathbf{p} - \mathbf{c}_k)$$

Somme sur les points \mathbf{c}_k Noyau de convolution Impulsion en \mathbf{c}_k



Vidéo

Vidéo

Vidéo

J. P. Lewis. Algorithms for solid noise synthesis. *ACM SIGGRAPH Computer Graphics*, **23**(3), 263–270, 1989.



eric.galin@liris.cnrs.fr
http://liris.cnrs.fr/~egalin

Gradient Noise

- Overview
- Mapping
- Projection
- Procedural
- Synthesis

Génération

Interpolation d'un gradient sur une grille

1

Compute grid cell $C(i,j)$
 $i = \lfloor x \rfloor$ and $j = \lfloor y \rfloor$

2

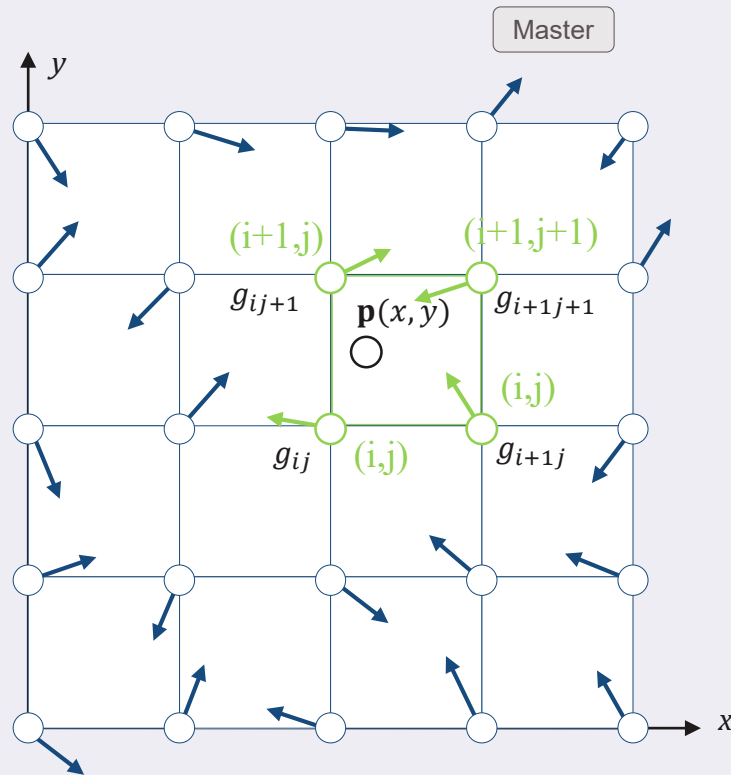
Compute gradients at the corners
Procedural or **precomputed table**

$$\begin{aligned}n_{00} &= g_{ij} \cdot (u, v) \\n_{10} &= g_{i+1j} \cdot (u - 1, v) \\n_{01} &= g_{ij+1} \cdot (u, v - 1) \\n_{11} &= g_{i+1j+1} \cdot (u - 1, v - 1)\end{aligned}$$

3

Bilinear interpolation
Cubic of higher degree polynomials

$$\begin{aligned}n(\mathbf{p}) &= \\(1 - \alpha)(1 - \beta) n_{00} &+ (1 - \alpha)\beta n_{10} \\+ \alpha(1 - \beta) n_{01} &+ \alpha\beta n_{11}\end{aligned}$$



K. Perlin. An image synthesizer. *Siggraph Computer Graphics*, 19(3), 287–296, 1985.

K. Perlin. Improving noise. *ACM Transactions on Graphics*, 21(3), 681–682, 2002.



eric.galin@liris.cnrs.fr
<http://liris.cnrs.fr/~egalin>

Détails d'implémentation

- Overview
- Mapping
- Projection
- Procedural
- Synthesis

Caractérisation

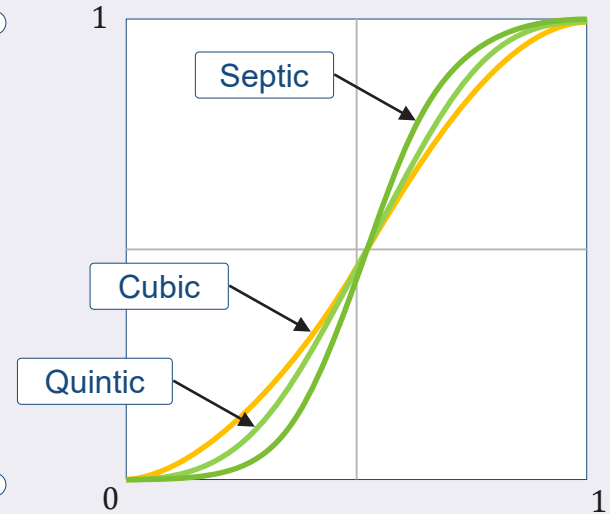
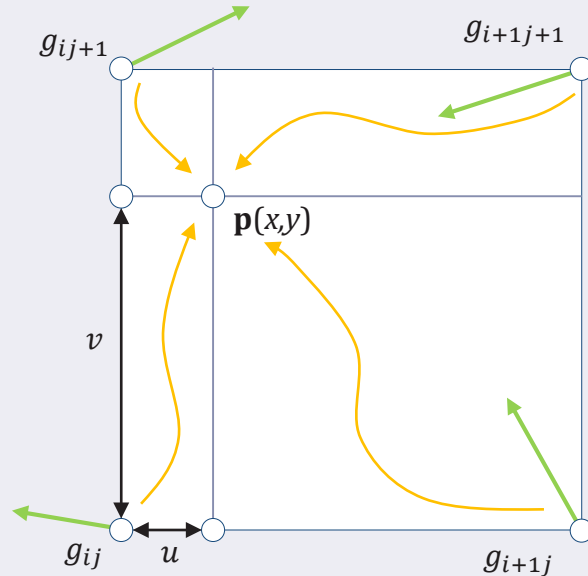
Degré de lissage

Smoothing polynomials

$\alpha = s(u)$ and $\beta = s(v)$ where
 s is a smooth polynomial function

Cubic	C^1	$s(t) = 3t^2 - 2t^3$
Quintic	C^2	$s(t) = 6t^5 - 15t^4 + 10t^3$
Septic	C^3	$s(t) = -20t^7 + 70t^6 - 84t^5 + 35t^4$

Master



eric.galin@liris.cnrs.fr
<http://liris.cnrs.fr/~egalin>

Implémentation

Overview

Mapping

Projection

Procedural

Synthesis

Étapes

Calcul d'un vecteur aléatoire $g(\mathbf{q})$ pour $\mathbf{q} \in \mathbf{N}^2$ sommet sur la grille des entiers

Interpolation bilinéaire pour $\mathbf{p} \in \mathbf{R}^2$ des valeurs aux sommets

```
vec2 g ( ivec2 q ) {  
    int n = q.x + q.y*11111 ;  
    n = (n<<13)^n;  
    n = (n*(n*15731+789221)+1376312589)>>16 ;  
    return vec2 ( cos ( float ( n ) ) , sin ( float ( n ) ) ) ;  
}
```

2

Compute gradients at the corners

```
float noise( vec2 p ) {  
    ivec2 q = ivec2( floor ( p ) ) ;  
    vec2 f = fract ( p ) ;  
    vec2 u = f * f * (3.0 - 2.0 * f);  
    return mix(  
        mix ( dot ( g ( q+ivec2(0,0) ) , f-vec2(0.,0.) ) ,  
            dot ( g ( q+ivec2(1,0) ) , f-vec2(1.,0.) ) , u.x),  
        mix ( dot ( g ( q+ivec2(0,1) ) , f-vec2(0.,1.) ) ,  
            dot ( g ( q+ivec2(1,1) ) , f-vec2(1.,1.) ) , u.x), u.y);  
}
```

1

Compute grid cell

3

Interpolation



eric.galin@liris.cnrs.fr

<http://liris.cnrs.fr/~egalain>

Mouvement Brownien Fractionnaire

- Overview
- Mapping
- Projection
- Procedural
- Synthesis

Construction

Somme de o bruits de fréquences $f_k = 1/\lambda_k$ et d'amplitudes a_k
Réduction des artefacts d'alignement de grille par une rotation R_k

Octaves

Amplitude

Rotation

Longueur d'onde

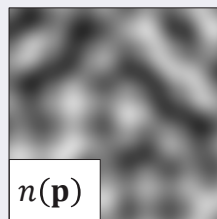
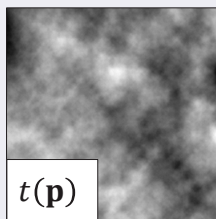
$$t(\mathbf{p}) = \sum_{k=0}^{o-1} a_k n\left(\frac{R_k \cdot \mathbf{p}}{\lambda_k}\right)$$

Turbulence [Ebert1998]

Bruit

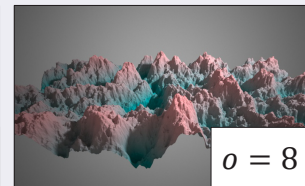
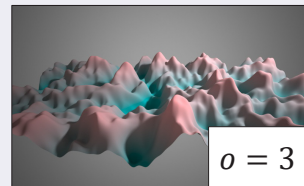
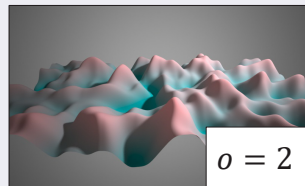
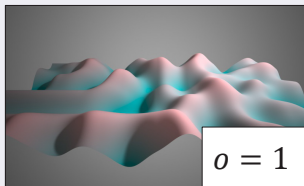
$$a_k = \alpha_0 \beta^{-k}$$
$$\lambda_k = \lambda_0 \beta^{-k}$$

Souvent $\beta = 2$
Calcul incrémental de a_k
et λ_k dans la boucle



$$R_k = (R_\theta)^k = R_{k-1} \cdot R_\theta$$

Souvent $\theta = 0$ donc $R_k = I$
Calcul incrémental de R_k dans la boucle



Ebert et al. Texturing and Modeling: A Procedural Approach. Academic Press Professional, 1998.

Formes

- Overview
- Mapping
- Projection
- Procedural
- Synthesis

Voronoi

Distance minimale entre \mathbf{p} et des points \mathbf{c}_k
Génération aléatoire cohérente de points \mathbf{c}_k

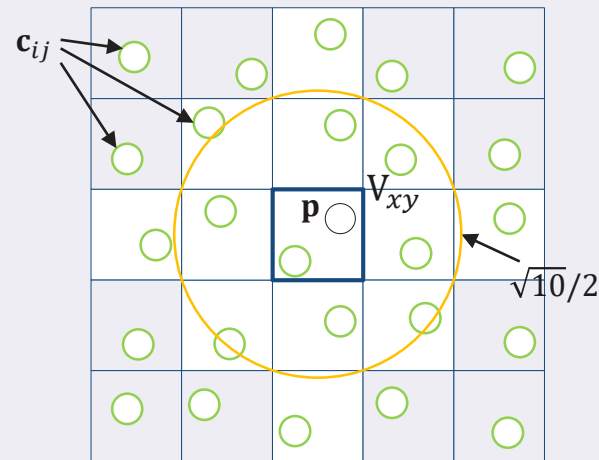
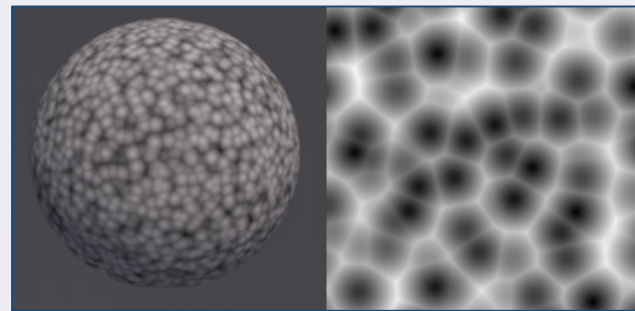
$$\mathbf{p} \longrightarrow \text{Voronoi} \\ f(\mathbf{p}) = \min_k d(\mathbf{p}, \mathbf{c}_k)$$

```
float Voronoi ( const Vector & p ) {  
    float r = Norm ( p - c [ 0 ] );  
    for ( int i = 1 ; i < n ; i++ ) {  
        float ri = Norm ( p - c [ i ] );  
        if ( ri < r ) { r = ri ; }  
    }  
    return r ;  
}
```

Taille du tableau \mathbf{c} ?

```
float Voronoi ( const Vector & p ) {  
    int x = int ( p [ 0 ] ) ; int y = int ( p [ 1 ] ) ;  
    float r = 100000 ; // Infinity  
    for ( int i = x-2 ; i < x+3 ; i++ ) {  
        for ( int j = y-2 ; j < y+3 ; j++ ) {  
            float rij = Norm ( p - Point ( i , j ) ) ;  
            if ( rij < r ) { r = rij ; }  
        }  
    }  
    return r ;  
}
```

Génération des points \mathbf{c}_{ij}



S. Worley. A cellular texture basis function. *Proceedings of SIGGRAPH*, 291–294, 1996



eric.galin@liris.cnrs.fr
<http://liris.cnrs.fr/~egalain>

Déformations

Overview

Mapping

Projection

Procedural

Synthesis

Perturbation

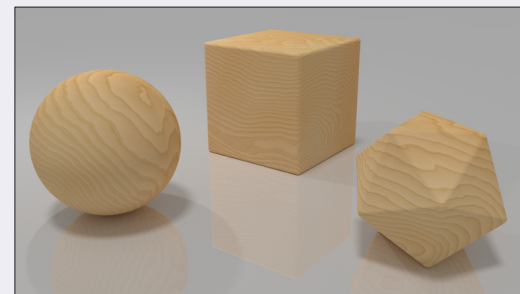
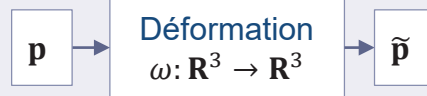
Fonction de déformation $\omega: \mathbf{R}^3 \rightarrow \mathbf{R}^3$

Déplacement stochastique

$$\omega(\mathbf{p}) = \mathbf{p} + t(\mathbf{p})$$

Identité

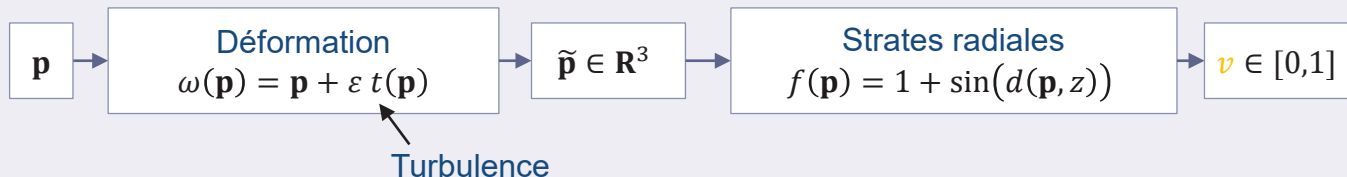
Turbulence [Ebert1998]



Bois

Strates concentriques radiales déformées

Rampe de couleur



```
vec3 Wood ( vec3 p ) {  
    vec3 q = p + Turbulence ( p );  
    float r = 0,5 * ( 1.0 + sin ( length ( p.xy ) ) );  
    return mix ( vec3(0.5,0.5,0,0), vec3(1.0,1.0,0,0), r );  
}
```

Ebert *et al.* Texturing and Modeling: A Procedural Approach. *Academic Press Professional*, 1998.



eric.galin@liris.cnrs.fr

http://liris.cnrs.fr/~egalin

Computer Graphics

Texture synthesis

Synthèse

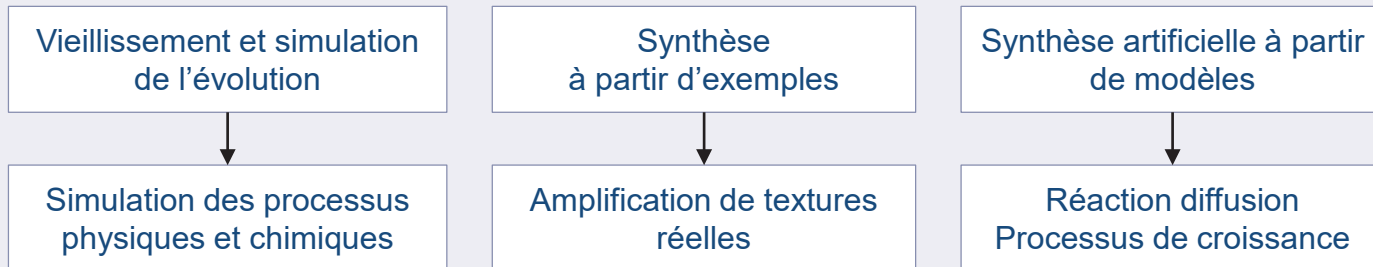
Overview

Mapping

Projection

Procedural

Synthesis



eric.galin@liris.cnrs.fr

<http://liris.cnrs.fr/~egalin>

Réaction diffusion

Overview

Mapping

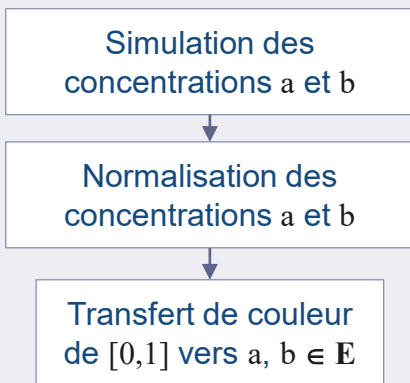
Projection

Procedural

Synthesis

Fondamentaux

Simulation de l'interaction de deux espèces **A** et **B**
Problème mathématique complexe

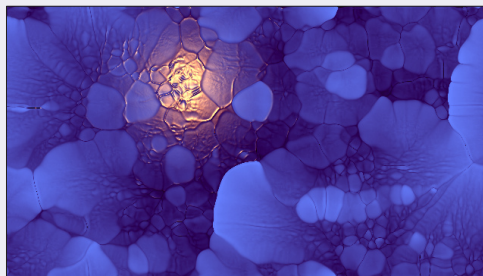


Concentrations de **A** et **B**

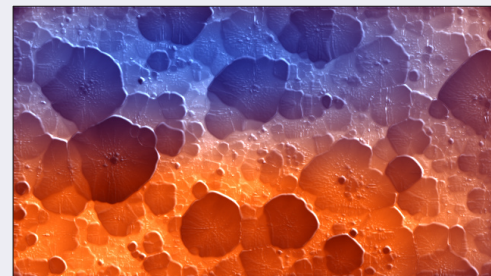
$$\frac{\partial a}{\partial t} = f(a, b) + k_a (\nabla a)^2$$
$$\frac{\partial b}{\partial t} = g(a, b) + k_b (\nabla b)^2$$

Fonctions de la concentration

Coefficients de diffusion



Video



Video



eric.galin@liris.cnrs.fr
<http://liris.cnrs.fr/~egalin>

G. Turk. Generating Textures on Arbitrary Surfaces Using Reaction-Diffusion. *Proceedings of Siggraph*, 289–298, 1991
K. Fleischer, D. Laidlaw, R. Currin, A. Barr. Cellular Texture Generation. *Proceedings of Siggraph*, 239–248, 1995

Génération à partir d'exemples

Overview

Mapping

Projection

Procedural

Synthesis

Objectif

Générer des textures complètes à partir d'exemples de taille réduite

Pixel
Génération par pixel
ou par régions

Surface
Champ de vecteur et
synthèse à la surface

Pavage
Anisotropie à partir
d'un pavage de tuiles

Textures Solides
Matériaux à partir de
coupes



Synthèse d'image à partir d'un exemple



Distribution spatiale à partir d'une image

L. Wei, S. Lefebvre, V. Kwatra, G. Turk. State of the Art in Example-based Texture Synthesis. *Eurographics*, 2009
R. Jagnow, J. Dorsey, H. Rushmeier. Stereological techniques for solid textures. *Proceedings of SIGGRAPH*, 329–335, 2004

Simulation

Overview

Mapping

Projection

Procedural

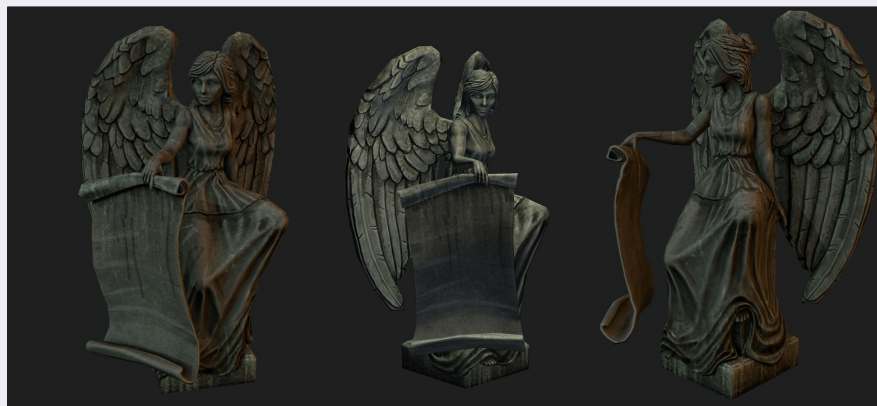
Synthesis

Viellissement et changements d'apparence

Simulations de différents phénomènes naturels sur l'objet

Oxydation [Dorsey1996], poussière, fractures

Moisissures, lichens, mousses



Simulation

Simulation **globale** du phénomène physique, chimique ou biologique

Procédural

Approximation des effets par une méthode phénoménologique **locale**

J. Dorsey, H. Pedersen, P. Hanrahan. Flow and Changes in Appearance. *Proceedings of Siggraph*, 1996



eric.galin@liris.cnrs.fr

<http://liris.cnrs.fr/~egalin>