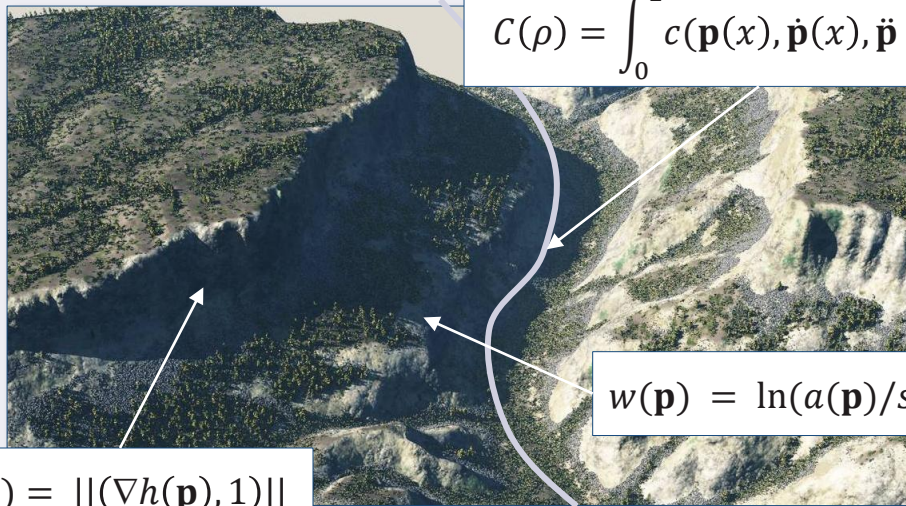


Digital World Modeling

From mathematics ...

$$C(\rho) = \int_0^1 c(\mathbf{p}(x), \dot{\mathbf{p}}(x), \ddot{\mathbf{p}}(x)) dx$$



$$s(\mathbf{p}) = ||(\nabla h(\mathbf{p}), 1)||$$

$$w(\mathbf{p}) = \ln(a(\mathbf{p})/s(\mathbf{p}))$$

... to the screen

E. Galin
Université Lyon 1

Digital World Modeling

Data Structures

Procedural Modeling

Erosion Simulation

Procedural Road Generation

Vegetation and Ecosystems

Growth models

Aging and weathering

Classification

Surfaces

Volumes

Analysis

Verrous scientifiques et techniques

Terrains de très grande dimensions à différentes résolutions

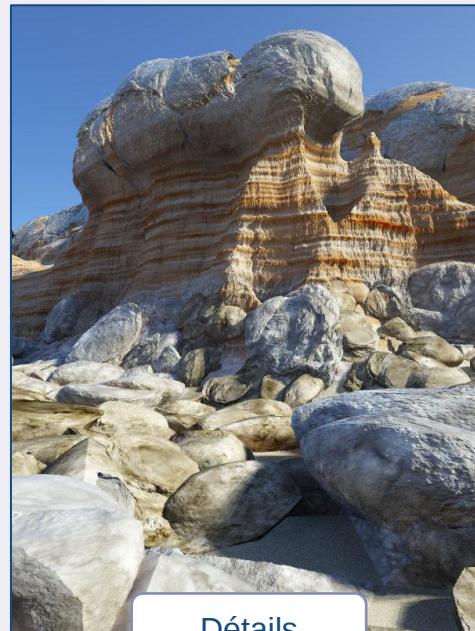
Terrain géologiquement corrects

Prise en compte des différents matériaux

Détails géométriques (rochers, cailloux)



Surplombs



Détails

Introduction

Classification

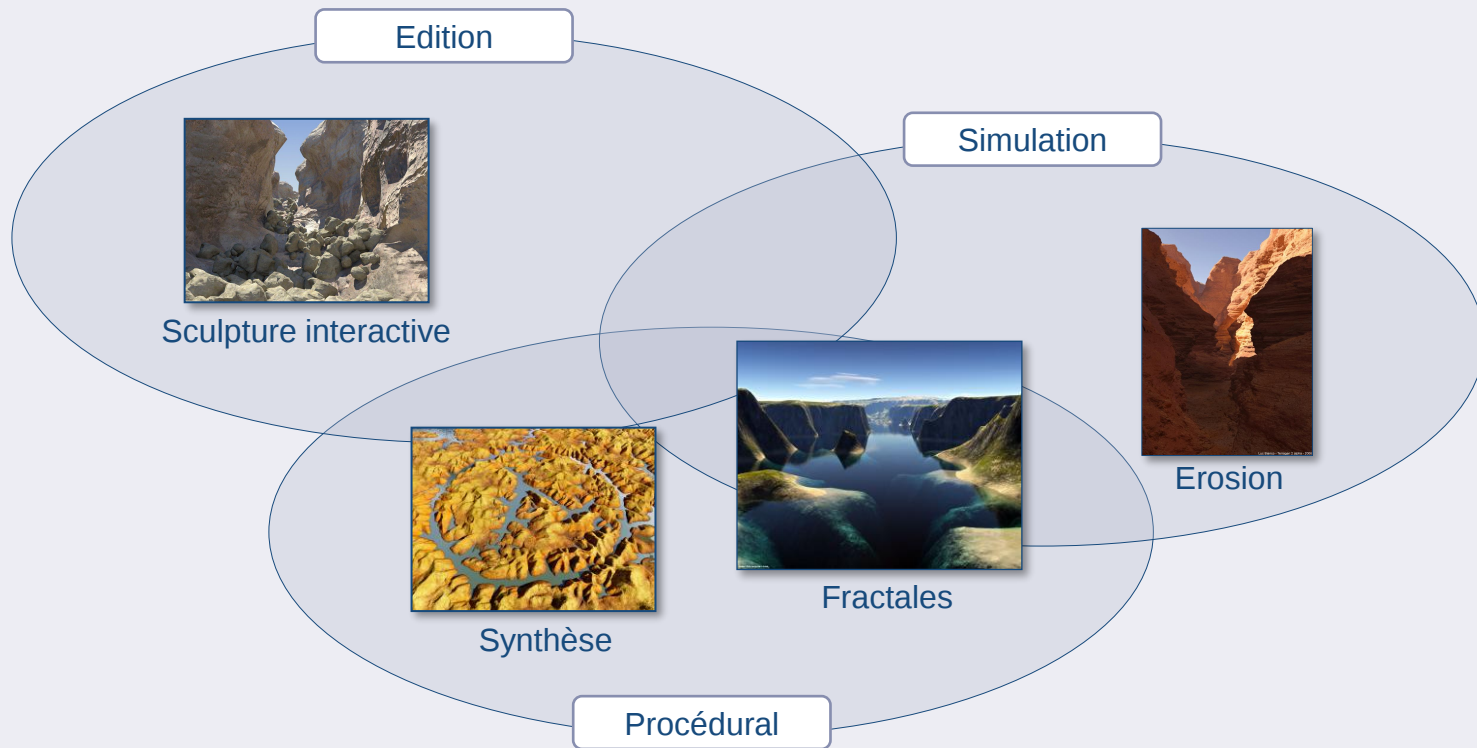
Surfaces

Volumes

Analysis

Techniques de création

Méthodes de génération et de reconstruction





Challenges

- Realism
- Variety of landforms
- Range of scale
- Control and authoring

Constraints

- Memory
- Speed

Classification

Classification

Surfaces

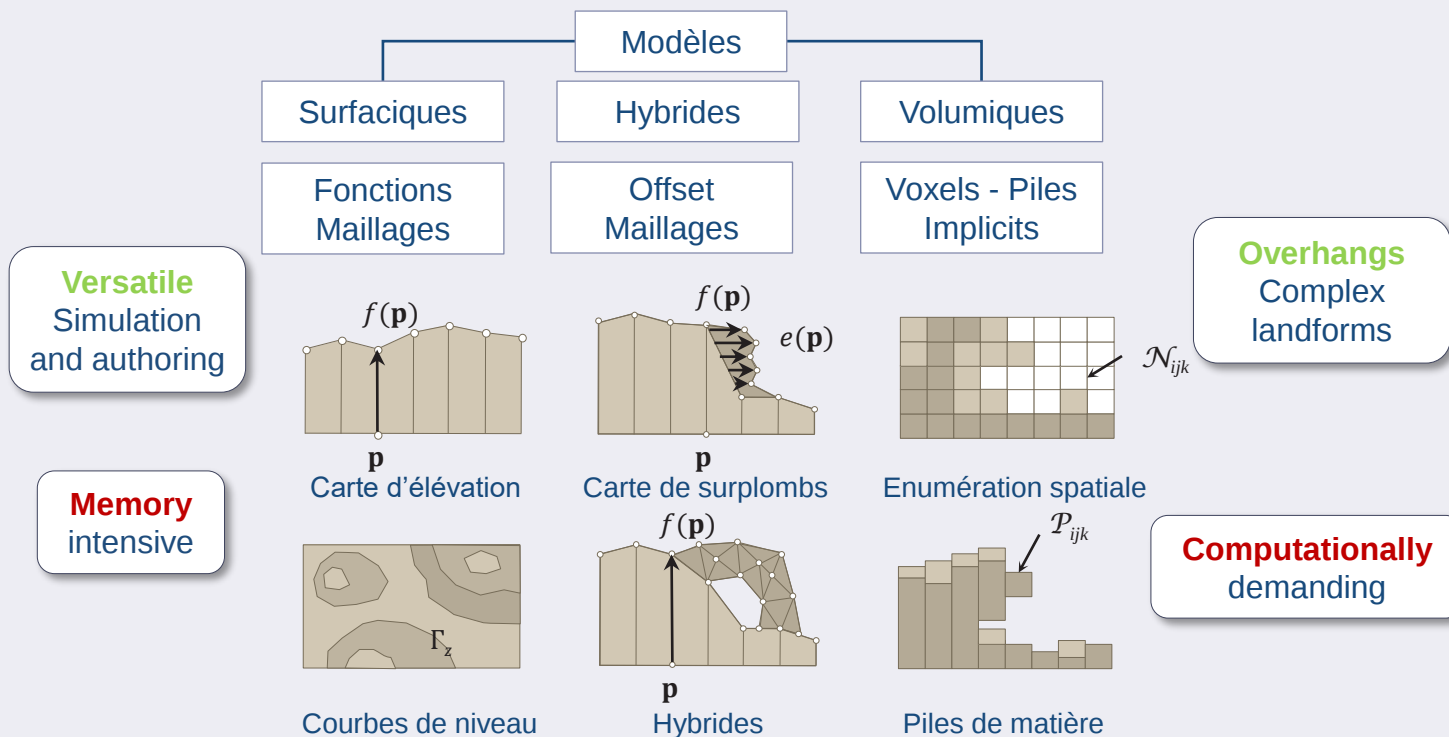
Volumes

Analysis

Structure de données

Représentation surfacique : hauteur du relief

Modèles volumiques : différents matériaux en surface et en profondeur



Elevation Models

Function representation

Classification

Surfaces

Volumes

Analysis

Elevation functions

Explicit function $h : \mathbf{R}^2 \rightarrow \mathbf{R}$, direct elevation $h(\mathbf{p})$

Compact model representation

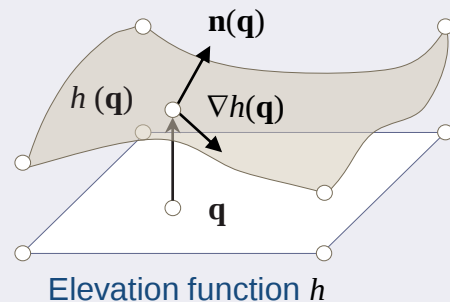
Gradient ∇h defines the steepest slope vector

$$s(\mathbf{p}) = |\nabla h(\mathbf{p})|$$

Slope

$$\mathbf{n}(\mathbf{p}) = (-\nabla h(\mathbf{p}), 1)$$

Normal, normalize $\hat{\mathbf{n}} = \mathbf{n}/|\mathbf{n}|$



Computing h (on the fly) may be **demanding**
Control of landforms, authoring

Height fields

Classification

Surfaces

Volumes

Analysis

Structure

Versatile model (generation, simulation)

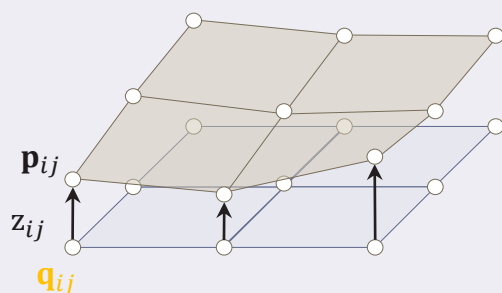
Elevation reconstructed from elevation z_{ij}

Discrétisation régulière ou irrégulière (adaptative) de Ω

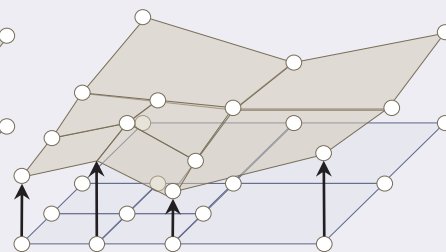
On définit pour $\mathbf{q}_{ij} \in \Omega = B(\mathbf{a}, \mathbf{b})$ les élévations $z_{ij} = h(\mathbf{q}_{ij})$

$$\mathbf{p}_{ij} = (\mathbf{q}_{ij}, z_{ij} = h(\mathbf{q}_{ij}))$$

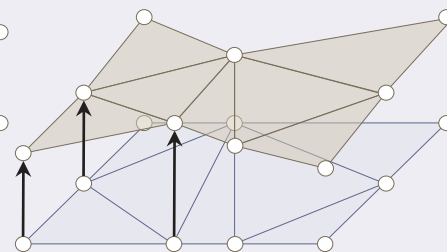
$$\mathbf{q}_{ij} = \mathbf{a} + \left(\frac{(\mathbf{b}_x - \mathbf{a}_x) i}{n_x - 1}, \frac{(\mathbf{b}_y - \mathbf{a}_y) j}{n_y - 1} \right)$$



Regular grid



Adaptive grid



Triangle network

L'élévation $h(\mathbf{p})$ à l'intérieur des cellules de Ω est définie par interpolation des valeurs aux sommets

Note

Memory demanding model : storage increases in $O(n^2)$

Algorithms **do not scale** for large terrains with a high resolution

1k × 1k
4Mb, ≤ 1s

16k × 16k
1Gb, ≥ 2h



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

Height fields

Classification

Surfaces

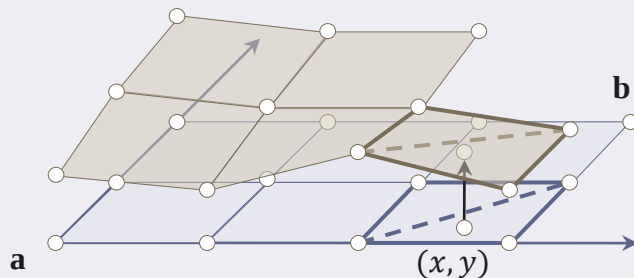
Volumes

Analysis

Elevation computation

Elevation z_{ij} is stored for $\mathbf{q}_{ij} \in \Omega$

Compute $h(\mathbf{p})$ by bilinear interpolation



```
class HeightField
{
protected:
    Vector a, b;      // Bounding box
    int nx, ny;       // Discretization
    double z[];       // Array of heights
public:
    double Height(const double&, const double&);
    double HeightGrid(int, int);
    // ...
};
```

```
double HeightField::Height(const double& x,
                           const double& y)
{
    // Local coordinates
    double u=(x-a[0])/(b[0]-a[0]);
    double v=(y-a[1])/(b[1]-a[1]);

    // Cell location within grid
    int nu=int(u*nx);
    int nv=int(v*ny);

    // Local coordinates within cell
    u=u-nu*(b[0]-a[0])/nx;
    v=v-nv*(b[1]-a[1])/ny;

    if (u+v<1)
    {
        return (1-u-v)*HeightGrid(i,j)
            +u*HeightGrid(i+1,j)
            +v*HeightGrid(i,j+1);
    }
    else
    {
        return (u+v-1)*HeightGrid(i+1,j+1)
            +(1-v)*HeightGrid(i+1,j)
            +(1-u)*HeightGrid(i,j+1);
    }
}
```



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

Limitations

Classification

Surfaces

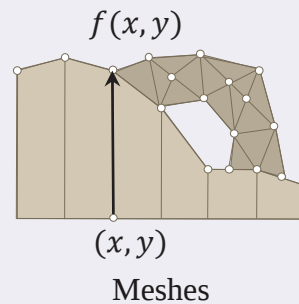
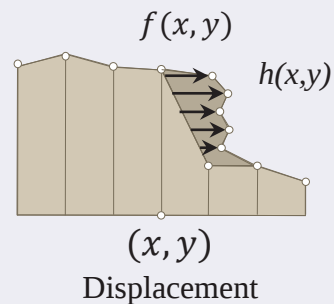
Volumes

Analysis

Limitations of elevation models

Overhangs and cliffs

Specific displacement or meshes for vertical parts





Ghost Recon Wildlands

Layered Models

Layered representations

Classification

Surfaces

Volumes

Analysis

Concept

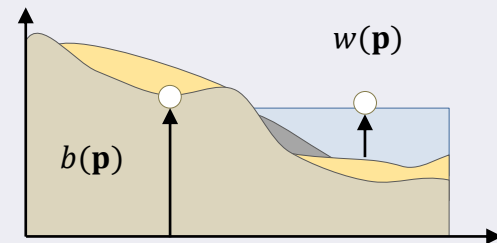
Bedrock elevation b is augmented with the thickness t_i of other material layers

$$h(\mathbf{p}) = b(\mathbf{p}) + \sum_{i \in L} t_i(\mathbf{p})$$

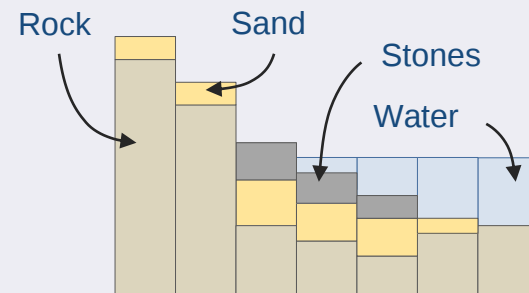
Applications

Erosion simulation produce fallen rocks or sediments

Ecosystems may use material layers



Elevation functions



Discrete layer stacks



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

Benes *et al.* Layered data representation for visual simulation of terrain erosion. *Spring Conference on Computer Graphics*, 2001.
Musgrave *et al.* The synthesis and rendering of eroded fractal terrains. *Computer Graphics*, **23**, 3, 1989.

Layered representations

Classification

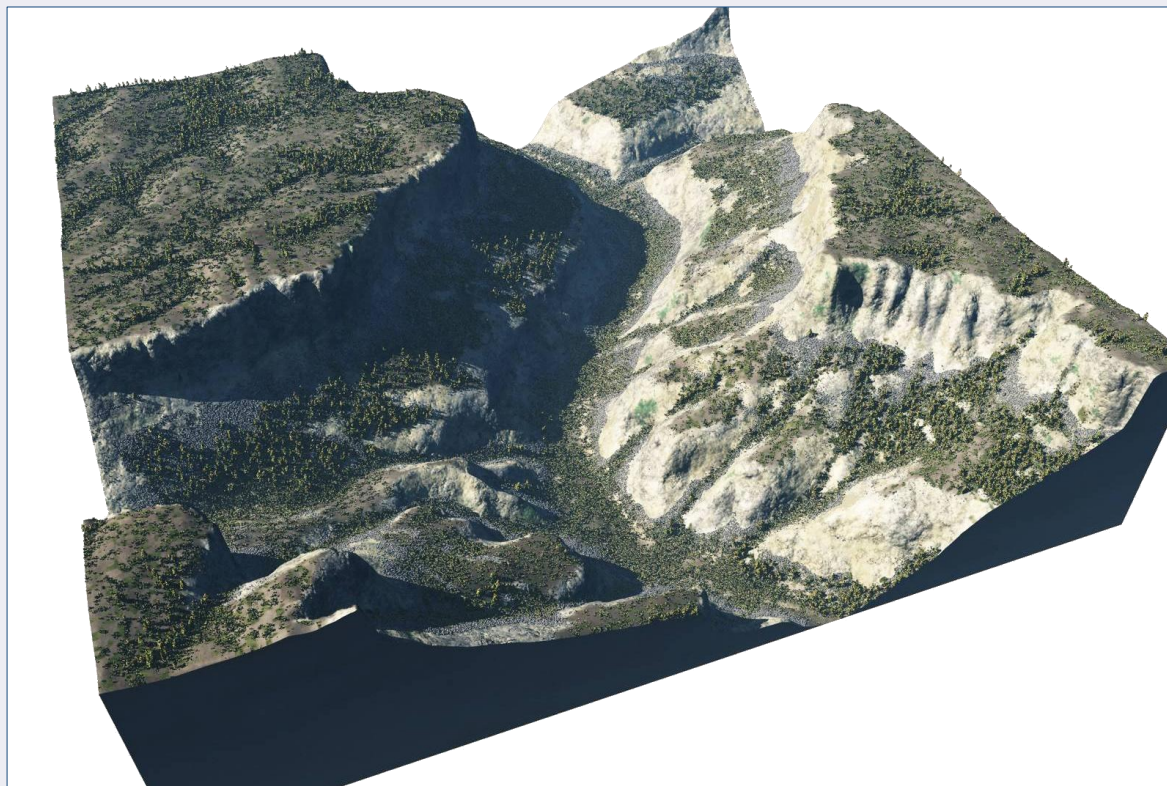
Surfaces

Volumes

Analysis

Combined terrain ecosystem simulation

Interacting layers



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

Volumetric models

Voxels

Classification

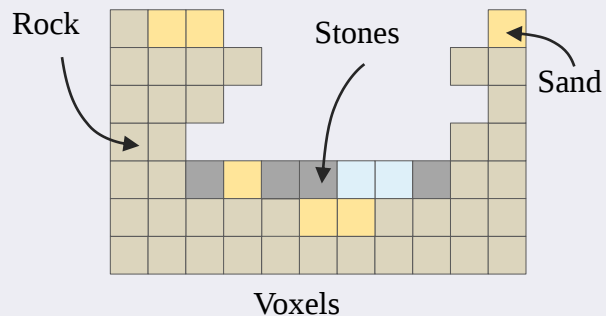
Surfaces

Volumes

Analysis

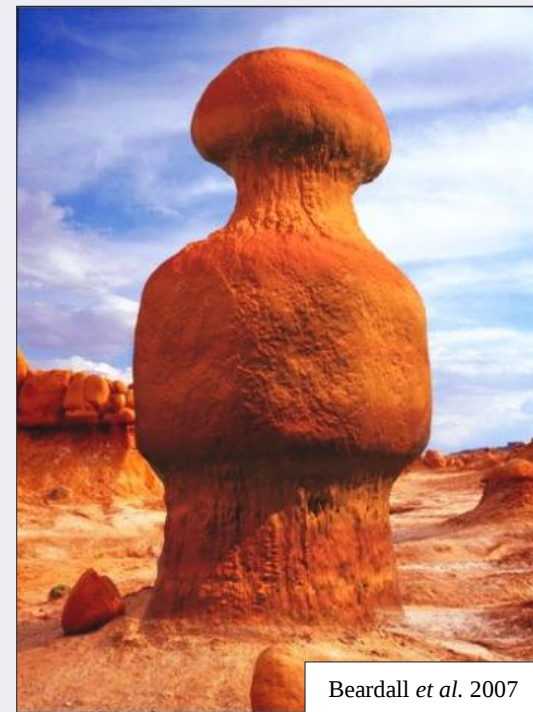
Model

Material function $\mu(\mathbf{p})$ defined in a voxel grid [Jones2010]



Allows **sculpting** and **simulations**

Memory demanding



Beardall *et al.* 2007

Jones *et al.* Directable weathering of concave rock using curvature estimation. *Transactions on Visualization and Computer Graphics*, **16** (1), 2010
Beardall *et al.* Goblins by spheroidal weathering. *Eurographics Workshop on Natural Phenomena*, 2007

Hybrid representations

Classification

Surfaces

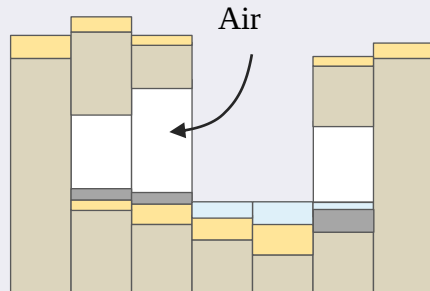
Volumes

Analysis

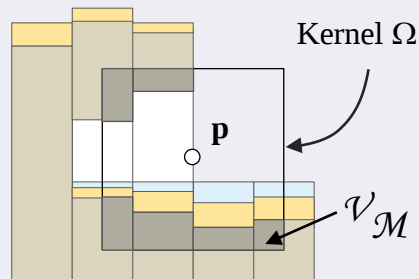
Smoothed layered stack

Layered materials with air allow the creation of overhangs and cavities [Peytavie2009]

Implicit model : convolution $\mu(\mathbf{p}) = 0$ for air, $\mu(\mathbf{p}) = 1$ for material



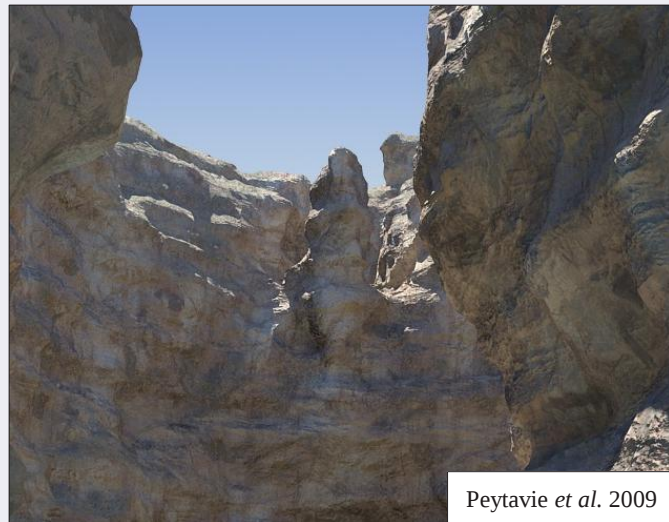
Layer stacks with overhangs



Layer stacks

$$\text{Convolution } f = 2\mu * k - 1$$

$$\text{Fast computation } \mu * k = \frac{V_M}{V_\Omega}$$



Peytavie et al. 2009

Peytavie et al. Arches: a framework for modeling complex terrains. *Computer Graphics Forum*, **28**, 2, 2009



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

Hybrid representations

Classification

Surfaces

Volumes

Analysis



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

Hybrid representations

Classification

Surfaces

Volumes

Analysis

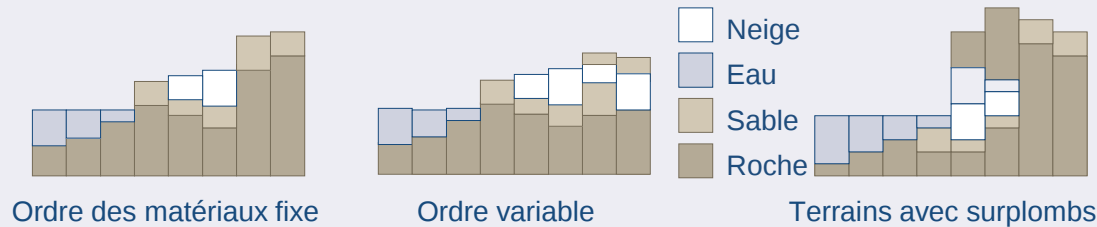
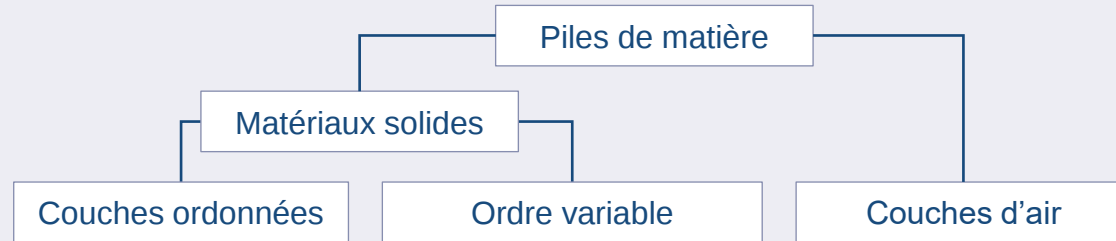
Principe

Un terrain est une grille de piles de matériaux

L'élévation finale correspond à la hauteur totale de la pile

Lorsque l'ordre est fixe, on simplifie la structure en une superposition de cartes de matière

$$h(\mathbf{p}) = b(\mathbf{p}) + \sum_{i \in L} t_i(\mathbf{p})$$



Hybrid representations

Classification

Surfaces

Volumes

Analysis

Structures de données

Comparaison entre plusieurs cartes
d'élévations et une structure dynamique

```
class LayerField
{
protected:
    Vector a, b;      // Bounding box
    int nx, ny;       // Discretization
    double rock[];    // Array of heights for
rock layer
    double sand[];    // Array of heights for
sand
    double water[];   // Array of heights for
water
public:
    double Height(const double&,const
double&);
    double HeightGrid(int,int);
    // ...
};
```

```
class Material      // Definition of a single cell
{
protected:
    double h;        // Height of material in cell
    int type;        // Material type
public:
    // ...
};
```

```
class MaterialStack // Stack of cells
{
protected:
    Material stack[]; // Array of different materials
    int n;            // Size of array
public:
    // ...
};
```

```
class MaterialStackField // Array of stacks of cells
{
protected:
    Vector a, b;      // Bounding box
    int nx, ny;       // Discretization
    MaterialStack array[]; // Array of material stacks
public:
    // ...
};
```



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

Hybrid representations

Classification

Surfaces

Volumes

Analysis

Structure

Modèle hybride combinant surface implicite – modèle discret
Combinaison pour l'édition et la simulation

	Implicite	Discret
Roche	Sculpture Fissures	
Matériaux granuleux		Dépôt Erosion Stabilisation
Liquides		Remplissage Ecoulement Phases



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

Hybrid representations

Classification

Surfaces

Volumes

Analysis

Structure de données

Caractérisation volumique en piles de matière g

Lissage de la surface par convolution [Peytavie2009] par un noyau h à support compact

$$S = \{\mathbf{p} \in \mathbb{R}^2, f(\mathbf{p}) = 0\}$$

$$f(\mathbf{p}) = \frac{i(\mathbf{p})}{4\sigma} - 1$$

$$i(\mathbf{p}) = g * h(\mathbf{p}) = \int_{\mathbb{R}^3} g(\mathbf{p}) h(\mathbf{p} - \mathbf{q}) d\mathbf{q}$$

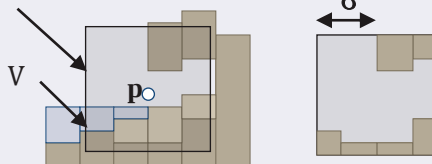
$$g(\mathbf{p}) = \begin{cases} 1 & \text{si } \mathbf{p} \in M \\ 0 & \text{sinon} \end{cases}$$

Squelette

$$h(\mathbf{p}) = \begin{cases} 1 & \text{si } |\mathbf{q}|_{\infty} \leq \sigma \\ 0 & \text{sinon} \end{cases}$$

Noyau

Support de convolution Ω de volume V_{Ω}



$$f(\mathbf{p}) = \frac{2V}{V_{\Omega}} - 1$$



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

A. Peytavie, E. Galin, J. Grosjean, S. Mérillou. Arches: a Framework for Modeling Complex Terrains. *Computer Graphics Forum (Proceedings of Eurographics)*, **28**(2), 457-467, 2009.

Hybrid representations

Classification

Surfaces

Volumes

Analysis

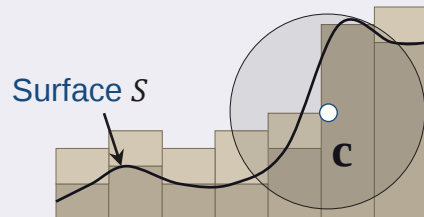
Combinaison des modèles implicites et discrets

Créer une primitive avec un centre $\mathbf{c}$

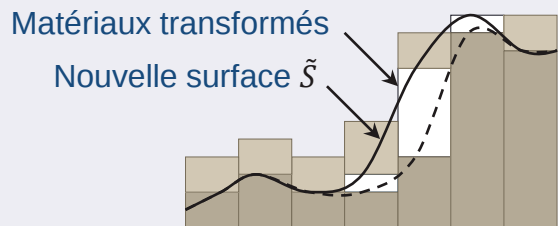
Mélange avec la primitive implicite

Discrétisation de la surface implicite en piles

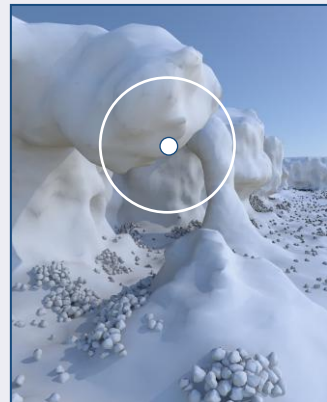
Stabilisation



Couches initiales



Couches modifiées



Hybrid representations

Classification

Surfaces

Volumes

Analysis

Smoothed layered stack

Layered materials with air allow the creation of overhangs and cavities

Implicit model created by a convolution



Peytavie et al. Procedural generation of rock piles using aperiodic tiling. *Computer Graphics Forum* **28**, 7, 2009

Implicit Surfaces

Classification

Surfaces

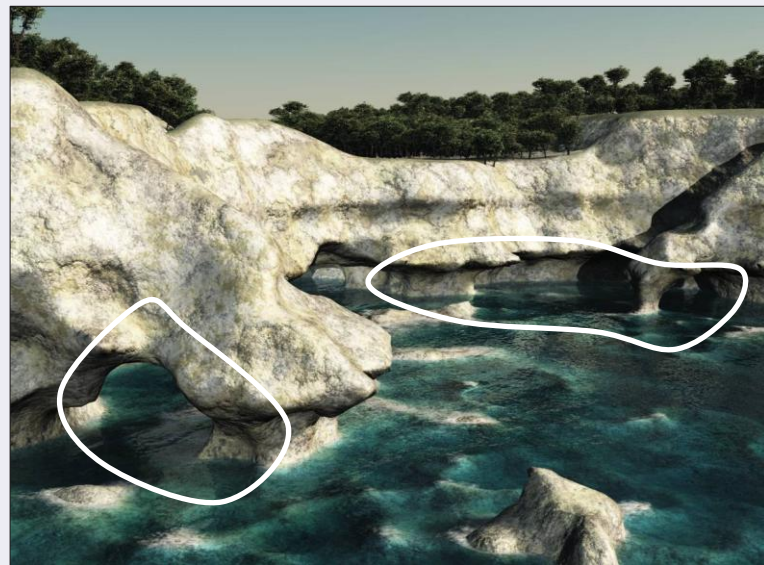
Volumes

Analysis

Field function

Find good primitives and operators for carving landforms

Cliffs, overhangs, caves



Need for effective **operators** and **primitives**

Conclusion

Classification

Surfaces

Volumes

Analysis

Height fields and layered height fields

Conspicuous in terrain modeling

Versatile for a variety of generation methods

Function-based models

Useful for modeling some **specific landforms**

Modeling large landscapes with a high resolution



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