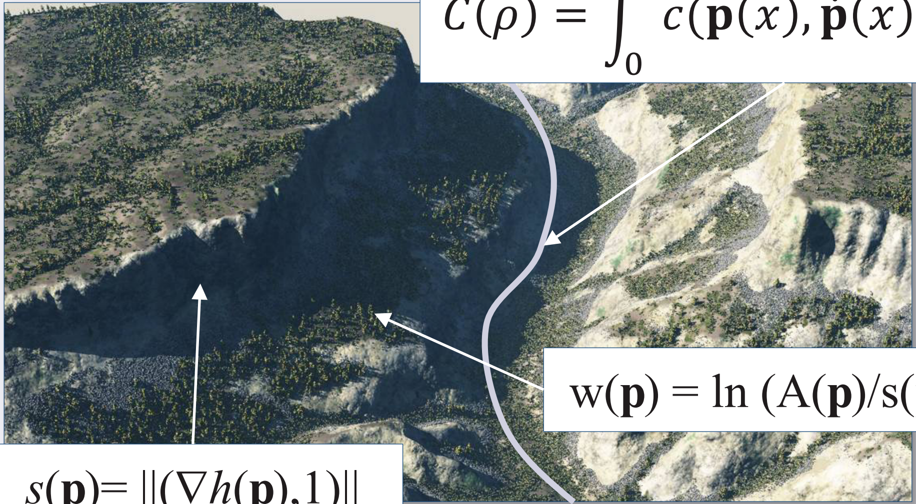
Digital World Modeling

From mathematics …

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



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

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

… 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

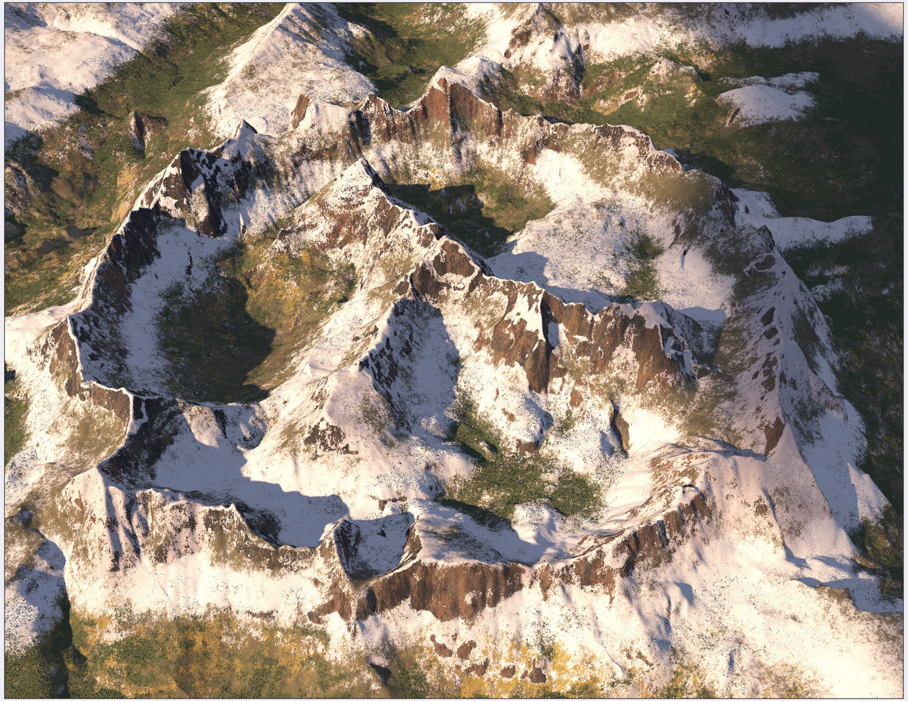
Aging and weathering

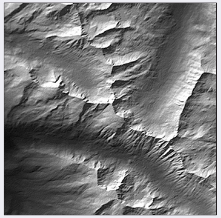
Synthesis from example

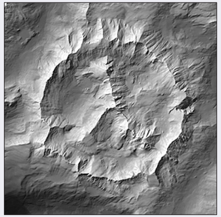
Patch matching

Création

- Recherche et combinaison de motifs [Zhou2007]
- Similaire aux méthodes de synthèse de texture







H. Zhou, J. Sun, G. Turk, J. Rehg. Terrain Synthesis from Digital Elevation Models, *IEEE Transactions on Visualization and Computer Graphics*, **13** (4), 834-848, 2007

Université Claude Bernard Lyon 1

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

Procedural Modeling of Height Fields

Faulting

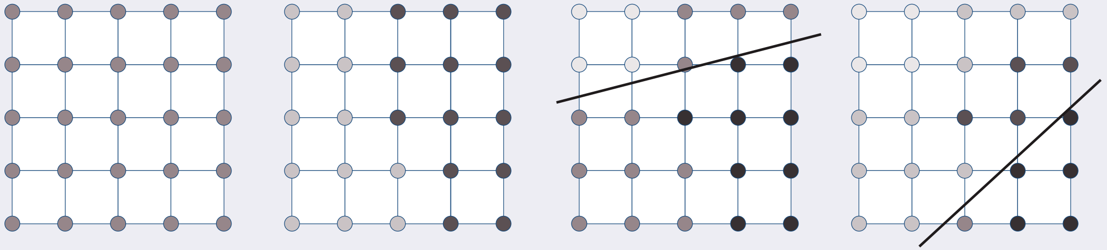
Algorithm

Generate set of faults F_i
Depth is a function of the distance $d\,(\mathbf{p}, F_i)$

Distance to lines $\mathrm{d}(\mathbf{p}, \Delta_i)$
or circles $d\,(\mathbf{p}, \mathbf{C})$

$$h(\mathrm{p}) = \sum_{k=0}^{n} \delta_i(\mathbf{p})$$

$$\delta_i(\mathbf{p}) = 1 - 2\left(1 - \mathrm{d}(\mathbf{p}, F_i)^2/r^2\right)^2$$



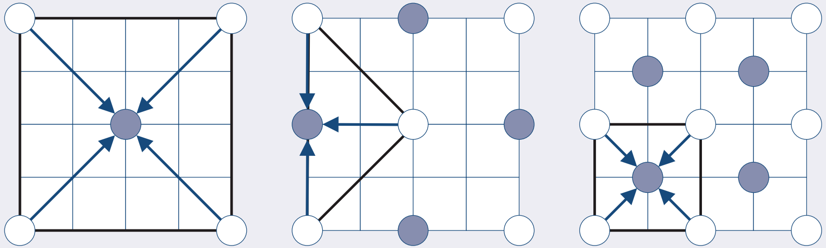
B. Mandelbrot. The Fractal Geometry of Nature. 1982.
R. Voss. Random fractal forgeries. *Fundamental Algorithms for Computer Graphics*, **17**, 1991.

Université Claude Bernard Lyon 1
eric.galin@liris.cnrs.fr
http://liris.cnrs.fr/~egalin

Recursive subdivision

Diamond Square

Height map of size $2^n + 1$
Requires explicit **storage**



```
// Initialise les valeurs aux coins
z[0][0] = z[0][n-1] = z[n-1][0] = z[n-1][n-1] = SEED;

double h = 200.0;
// Itération sur les niveaux, k = taille d'un carré
for (int k = n-1; k >= 2; k /= 2, h /= 2.0)
{
  int l = k/2; // Demi coté

  // Génération pour les carrés
  for (int x=0; x<n-1;x+=k)
  {
    for (int y=0; y<n-1; y+=k)
    {
      double a = (z[x][y] + z[x+k][y] +
        z[x][y+k] + z[x+k][y+k])/4.0;

      z[x+l][y+l] = a + random(-h,h);
    }
  }
```

```
  // Génération pour les losanges
  for (int x=0; x<n-1; x+=l)
  {
    for  (int y=(x+l)%k; y<n-1; y+=k)
    {
      double a = (z[(x-l+n)%n][y] +
        z[(x+l)%n][y] + z[x][(y+l)%n] +
        z[x][(y-l+n)%n])/4.0;

      z[x][y] = a+random(-h,h);

      // Cas spécial pour les arêtes
      if (x == 0) z[n-1][y] = a;
      if (y == 0) z[x][n-1] = a;
    }
  }
}
```


J. Lewis. Generalized stochastic subdivision. *ACM Transactions on Graphics*. **6**(3), 167–190, 1987.

LIRIS CNRS
Université Claude Bernard Lyon 1
eric.galin@liris.cnrs.fr
http://liris.cnrs.fr/~egalin

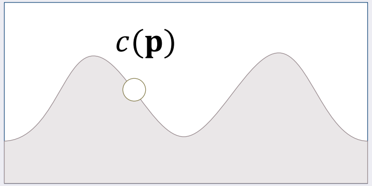
Trigonometric functions

Base **sine** or **cosine** $\mathbf{R}^2 \to [-1,1]$

Scaled cosine c characterized by amplitude and wavelength

$$c(\mathbf{p}) = a\cos(\mathbf{p}/\lambda)$$

Amplitude Wavelength



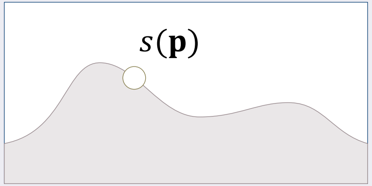
Noise

Base **noise functions** $n: \mathbf{R}^2 \to [-1,1]$

Scaled noise characterized by amplitude and wavelength

$$s(\mathbf{p}) = a\, n(\mathbf{p}/\lambda)$$

Amplitude Wavelength



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

Université Claude Bernard Lyon 1
eric.galin@liris.cnrs.fr
http://liris.cnrs.fr/~egalin

Modified noise functions

Basis function enhanced to generate ridges

$$r(\mathbf{p}) = 2(1 - |n(\mathbf{p})|) - 1 = 1 - 2|n(\mathbf{p})|$$

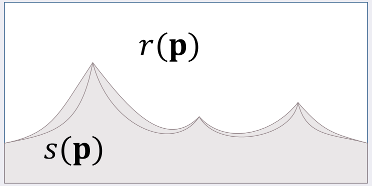
Ridge noise   Noise

$$s(\mathbf{p}) = 2(1 - |n(\mathbf{p})|)^2 - 1$$

Sharpened ridge noise

$$m(\mathbf{p}) = \min_i (n, n \circ t_i)$$

Intersection ridge noise t_i is a translation (offset)



S. Worley. A cellular texture basis function. In Proceedings of SIGGRAPH ’96, 291–294, 1996

Université Claude Bernard Lyon 1
eric.galin@liris.cnrs.fr
http://liris.cnrs.fr/~egalin

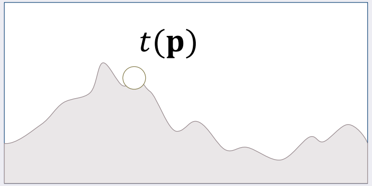
More complex functions

Fractional Brownian motion

Combination of **noise functions** $n: \mathbf{R}^2 \rightarrow [-1,1]$ to obtain fractal Brownian motion ***fBm***

$$h(\mathbf{p}) = t(\mathbf{p}) = \sum_{k=0}^{n} a_k n(\mathbf{p}/\lambda_k)$$

Turbulence



In general, $a_k = a_0 2^{-k}$ and $\lambda_k = \lambda_0 2^{-k}$

Lacunarity [Ebert1998]

Implementation details

Avoid grid artefacts

Transformed point

$$h(\boldsymbol{p}) = \sum_{k=0}^{n} a_k n(\mathbf{T}_k(\mathbf{p})\ /\lambda_k)$$

$$\mathbf{T}_k(\mathbf{p}) = \mathbf{R}_k \mathbf{p} + \mathbf{o}_k$$

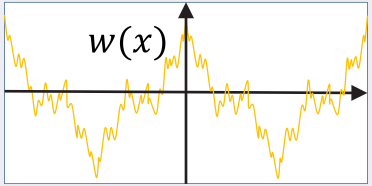
Random offset

Random rotation matrix $\mathbf{R}_k = \mathbf{R}_0^k$

Note

Le value noise n ressemble à la fonction de Weierstrass (1872)
Avec des nombres aléatoires dans $\{-1,1\}$ et une interpolation en cosinus :

$$n \equiv \sum_{i=1}^{\infty} \frac{1}{a^i} \cos(b^i \pi x)$$



Université Claude Bernard Lyon 1
eric.galin@liris.cnrs.fr
http://liris.cnrs.fr/~egalin

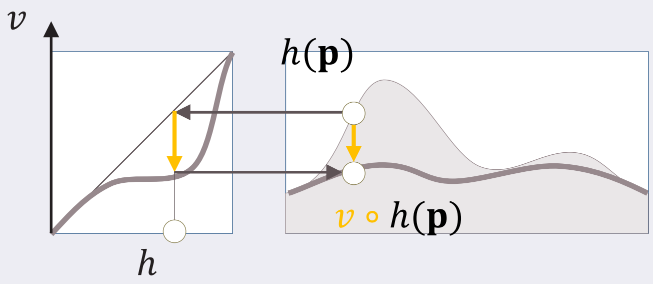
Deformations

Vertical deformations

Any function $v: \mathbf{R} \to \mathbf{R}$ that modifies the elevation

$$\tilde{h} = v \circ h$$

Deformation applied to the elevation



Horizontal deformations

Any warping $\omega^{-1}: \mathbf{R}^2 \to \mathbf{R}^2$

$$\tilde{h} = h \circ \omega^{-1}$$

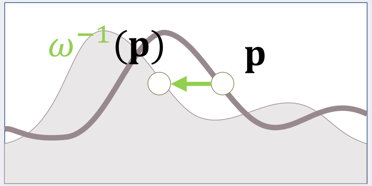
Space deformation applied to the point

$$\omega^{-1}(\mathbf{p}) = \mathbf{p} - \mathbf{t}$$

Translation

$$\omega^{-1}(\mathbf{p}) = \mathbf{p} + n(\mathbf{p})$$

Noise displacement



Université Claude Bernard Lyon 1
eric.galin@liris.cnrs.fr
http://liris.cnrs.fr/~egalin

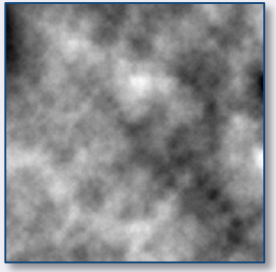
Fractional Brownian motion

Global function representation

Combination of **noise functions** $\mathrm{n}: \mathbf{R}^2 \rightarrow [-1,1]$ to obtain fractal Brownian motion *fBm*

$$h(\mathrm{p}) = t(p) = \sum_{k=0}^{n} a_k n(\mathbf{p}/\lambda_k)$$

Turbulence ⇔ fBm



In general, $a_k = a_0 2^{-k}$ and $\lambda_k = \lambda_0 2^{-k}$

Implementation details

Avoid grid artefacts in the noise n

$$h(\mathrm{p}) = \sum_{k=0}^{n} a_k n(\mathbf{T}_k(\mathbf{p})/\lambda_k)$$

Transformed point

$$\mathbf{R}_k \mathbf{p} + \mathbf{o}_k$$

Random offset

Random rotation matrix $\mathbf{R}_k = \mathbf{R}_0^k$

Challenge: assume the n is Lipschitz with constant λ compute the Lipschitz constant of h

LIRIS cnrs
Université Claude Bernard Lyon 1
eric.galin@liris.cnrs.fr
http://liris.cnrs.fr/~egalin

Other basis functions

From example
Procedural
Local primitives
Implicit modeling
Appendix

Modified noise functions

Basis function enhanced to generate ridges

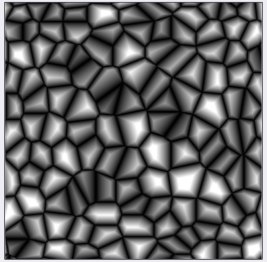
Ridge noise : $r = 2(1 - |n|) - 1 = 1 - 2|n|$

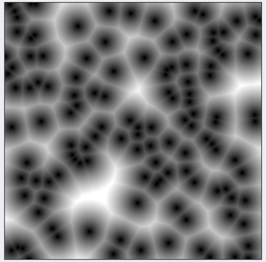
Sharpening operators : $s = 2(1 - |n|)^2 - 1$

Intersection : $m = \min_i (n, n \circ t_i)$ where t_i is a translation (offset)

Other basis functions

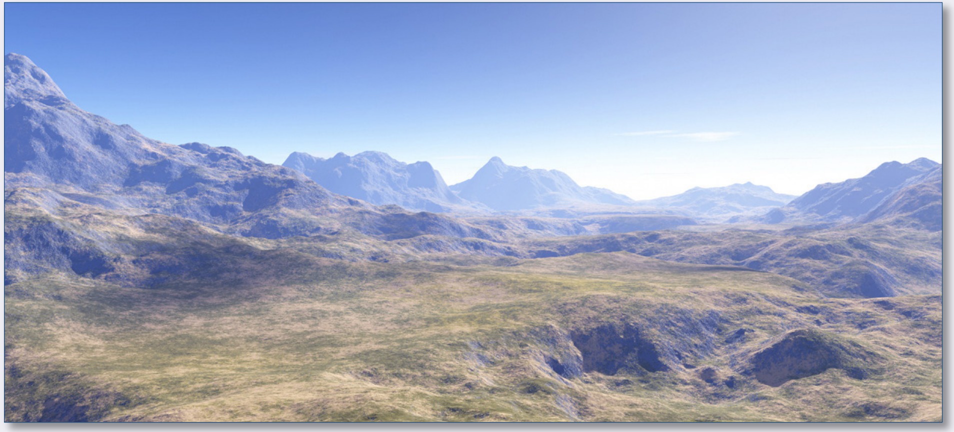
Cellular noise [Worley1996] $c: \mathbf{R}^2 \rightarrow [-1,1]$





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

Université Claude Bernard Lyon 1
eric.galin@liris.cnrs.fr
http://liris.cnrs.fr/~egalin



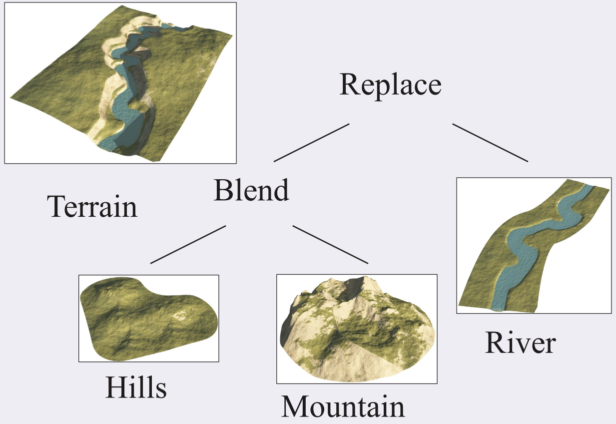
LIRIS CNRS
Université Claude Bernard Lyon 1
eric.galin@liris.cnrs.fr
http://liris.cnrs.fr/~egalin

Procedural Authoring of Height Fields

Construction tree

Local function representation

- **Hierarchical** representations using primitives organized in a tree
- Sparse combination of **landforms**
- Combination of implicit surface properties and elevation functions



Génevaux *et al*. Terrain modeling from feature primitives. *Computer Graphics Forum*, 2015.
Guérin *et al*. Sparse representation of terrains for procedural modeling. *Computer Graphics Forum*, **35**, 2, 2016

LIRIS CNRS Université Claude Bernard Lyon 1
eric.galin@liris.cnrs.fr
http://liris.cnrs.fr/~egalin

Fundamentals

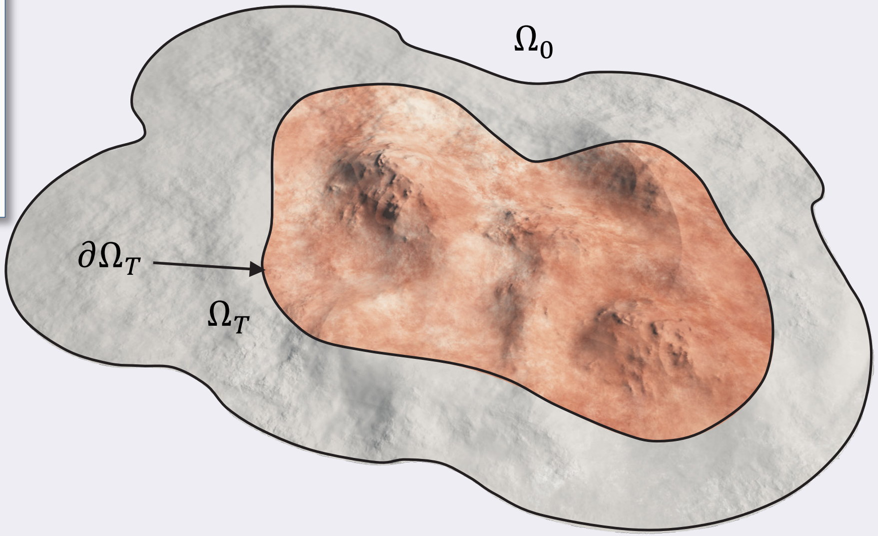
Nodes define two functions

- Elevation $h(\mathbf{p}): \mathbf{R}^2 \rightarrow \mathbf{R}$
- Weight $\alpha(\mathbf{p}): \mathbf{R}^2 \rightarrow \mathbf{R}^+$

Support : $\Omega_0 = \{\mathbf{p} \in R^2, \alpha(\mathbf{p}) > 0\}$

Domain : $\Omega_T = \{\mathbf{p} \in R^2, \alpha(\mathbf{p}) > T\}$

C^0 and Lipschitz property



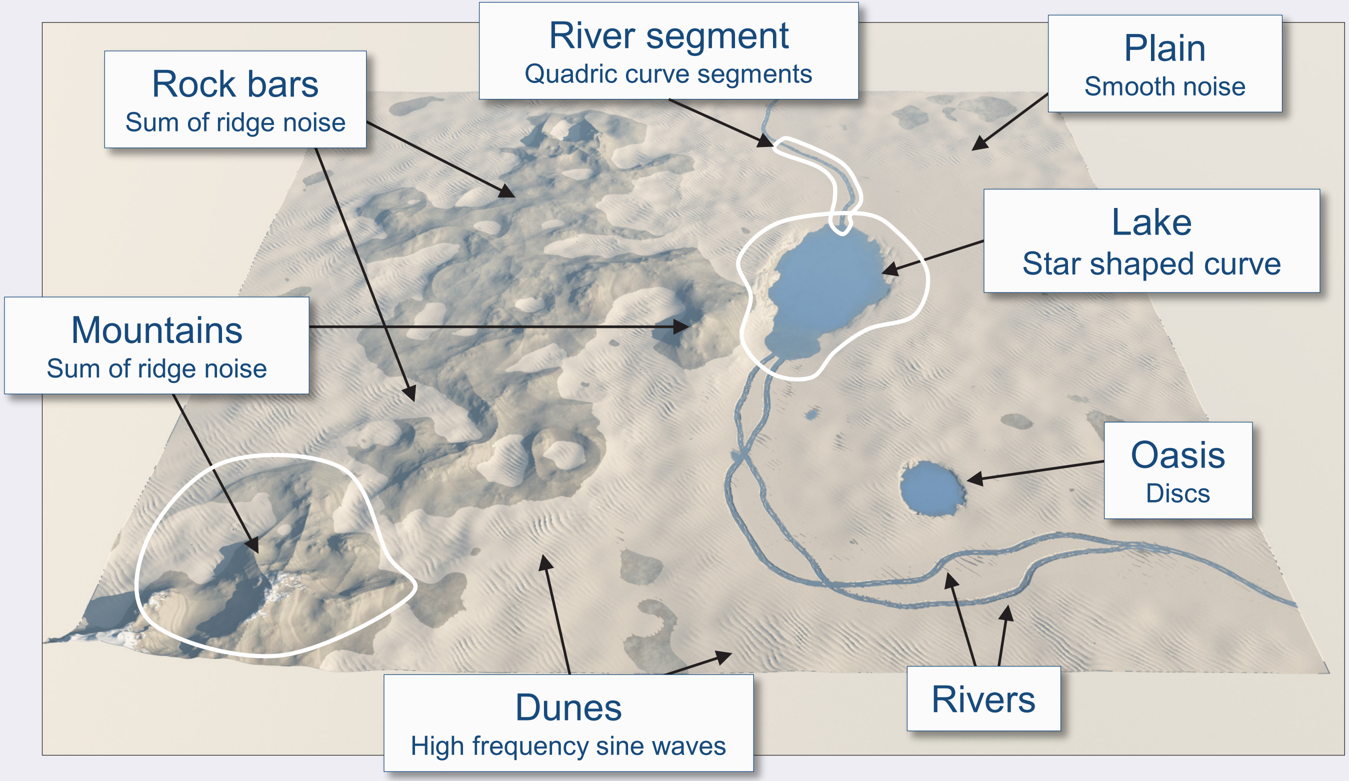
Regions of influence allow for:

- **Continuity** of the terrain
- **Compact** support
- Local **Lipschitz** property

LIRIS cnrs Université Claude Bernard Lyon 1

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

Control



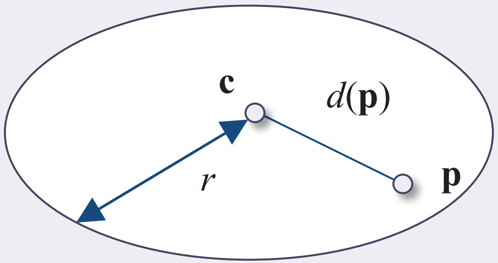
Université Claude Bernard Lyon 1

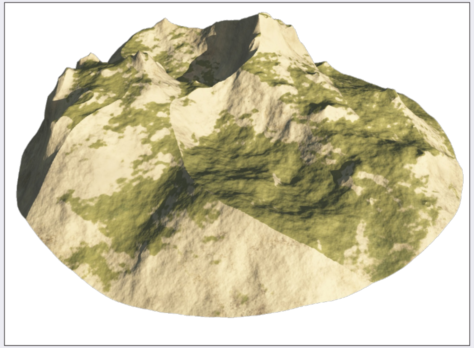
eric.galin@liris.cnrs.fr

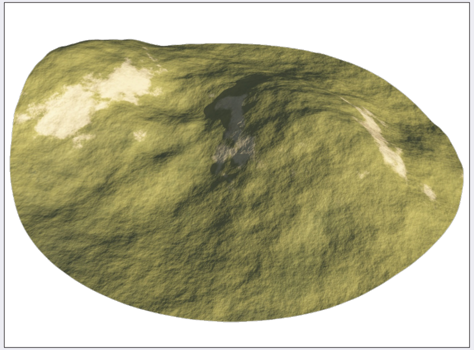
http://liris.cnrs.fr/~egalin

Mountain and hills over a disc

Elevation $h(\mathbf{p})$ as a combination of noises with amplitude and wavelength
Domain of influence over a disc $\alpha(\mathbf{p}) = g \circ |\mathbf{p} - \mathbf{c}|$







$g(x) = \left(1 - \frac{x^2}{r^2}\right)^3$ if $x < r$ and 0 otherwise

Implementation details

Elevation function

$$h(\boldsymbol{p}) = \mathbf{c}_z + t(\mathbf{p}_{xy}) - t(\mathbf{c}_{xy})$$

Controls height at center

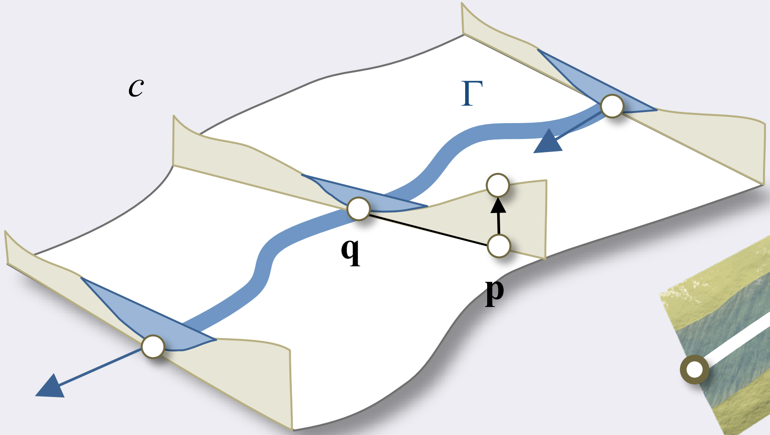
Precomputed

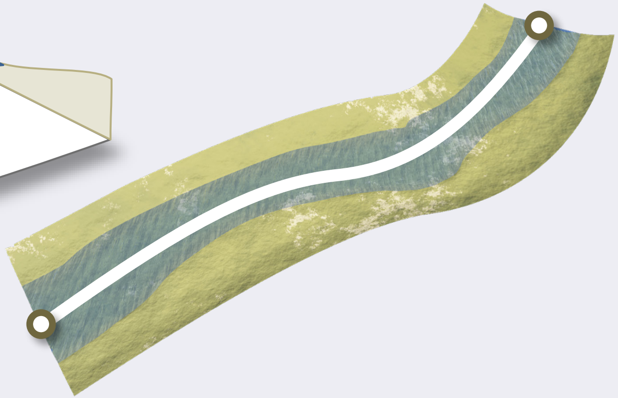
Turbulence ⇔ fBm

LIRIS CNRS
Université Claude Bernard Lyon 1
eric.galin@liris.cnrs.fr
http://liris.cnrs.fr/~egalin

River segments

Elevation $\mathrm{h}(\mathbf{p})$: combination of cross section profile along a curve
Domain of influence around curve $\alpha(\mathbf{p}) = g \circ d(\mathbf{p}, \boldsymbol{\Gamma})$





Compute the projection $\mathbf{q} = \pi_\Gamma(\mathbf{p})$ of $\mathbf{p}$ on the curve
Elevation is defined as $\mathrm{h}(\mathbf{p}) = \mathbf{q}_z + c \circ d(\mathbf{p}, \boldsymbol{\Gamma})$

Université Claude Bernard Lyon 1
eric.galin@liris.cnrs.fr
http://liris.cnrs.fr/~egalin

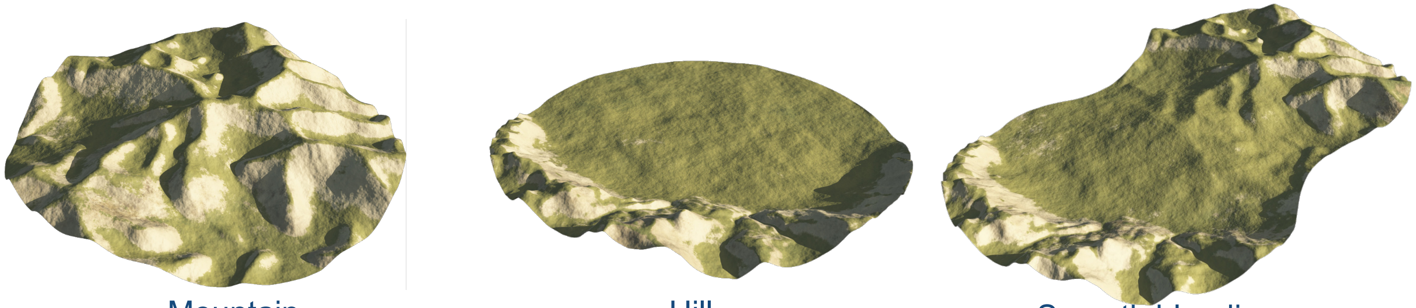
Blending

Blending two primitives with their own domain

Aggregation of landforms, yields new elevation with new domain

Elevation $h = (\alpha_a h_a + \alpha_b h_b)/(\alpha_a + \alpha_b)$

New influence $\alpha = \alpha_a + \alpha_b$



Mountain Hill Smooth blending

The contour $\partial\Omega_T$ of the new domain is the contour of the implicit equation $\alpha - T = 0$

$$\partial\Omega_T = \{\mathbf{p} \in R^2, \alpha(\mathbf{p}) = \mathrm{T}\}$$

Challenge: compute the Lipschitz constant of h inside Ω_T

LIRIS cnrs
Université Claude Bernard Lyon 1
eric.galin@liris.cnrs.fr
http://liris.cnrs.fr/~egalin

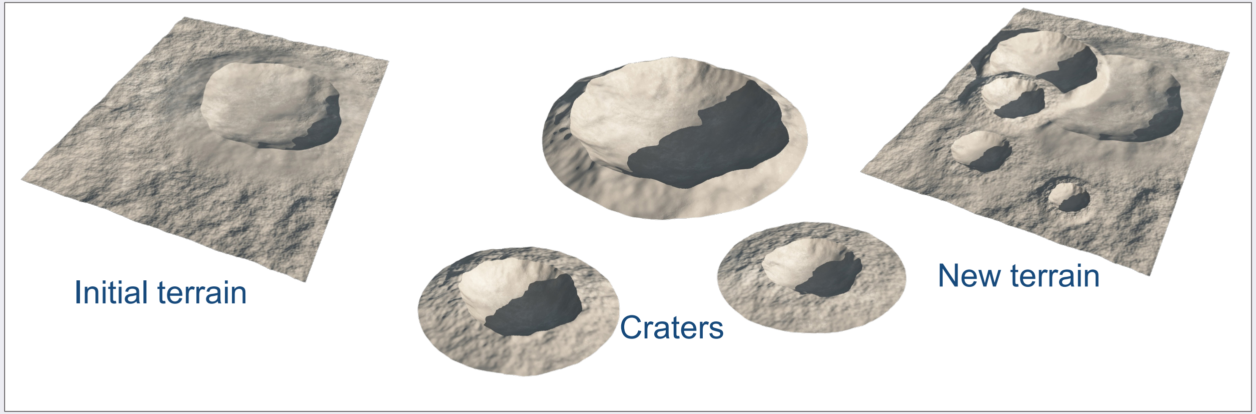
Replacement operator

Replace a part of a terrain with another one

Asymmetric operator

Elevation $h = (1 - \alpha_b)h_a + \alpha_b h_b$

Preserve influence of the left argument $\alpha = \alpha_a$



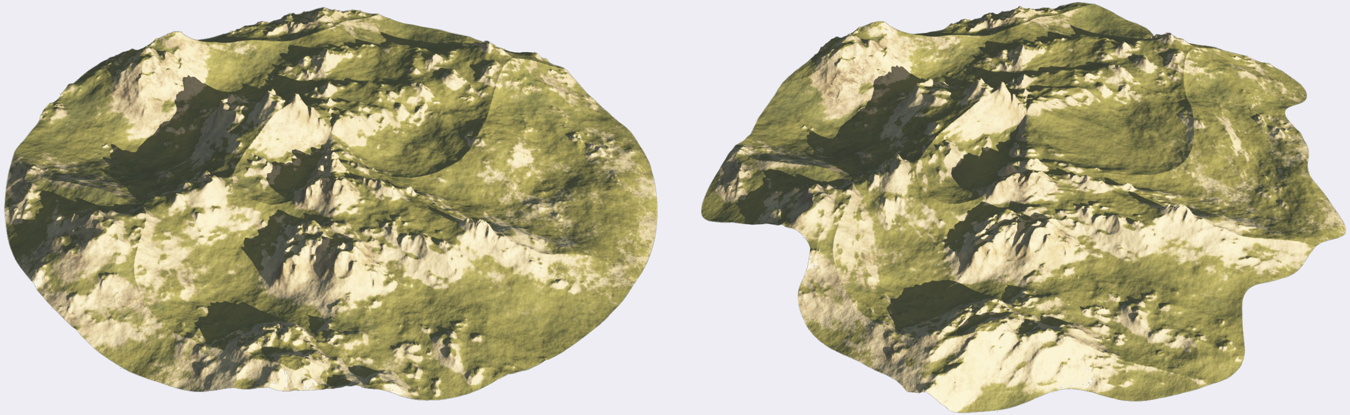
Université Claude Bernard Lyon 1
eric.galin@liris.cnrs.fr
http://liris.cnrs.fr/~egalin

Warping operators

Deformation of space

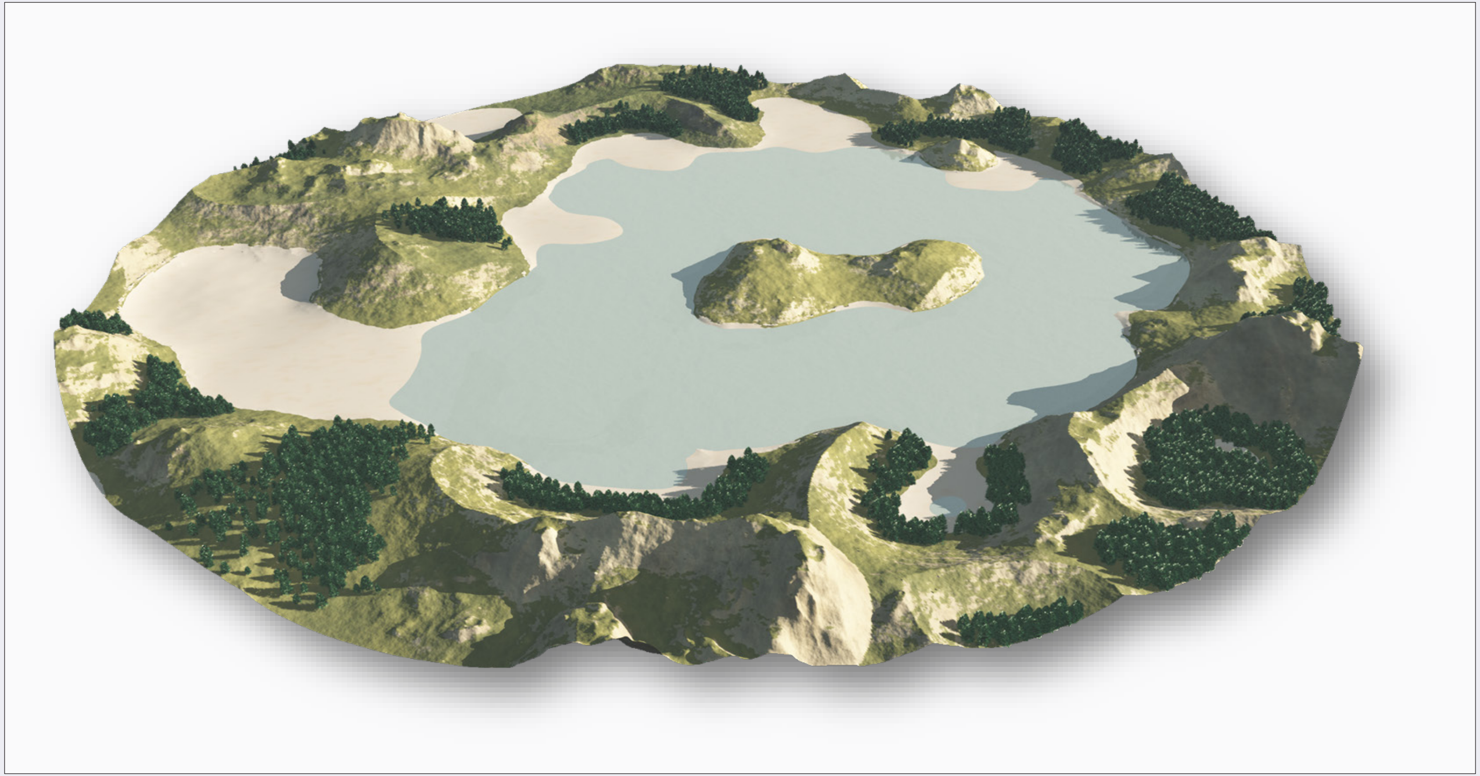
Any deformation $\omega^{-1}: \mathbf{R}^2 \to \mathbf{R}^2$ can be used as warping

Elevation $h = h_a \circ \omega^{-1}$ and coefficient $\alpha = \alpha_a \circ \omega^{-1}$



LIRIS
CNRS
Université Claude Bernard Lyon 1
eric.galin@liris.cnrs.fr
http://liris.cnrs.fr/~egalin

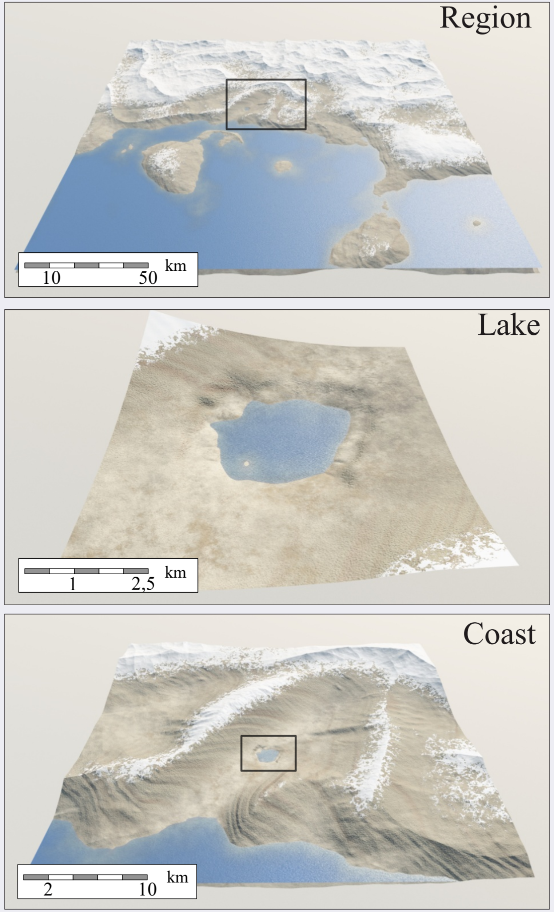
Scenery



LIRIS CNRS
Université Claude Bernard Lyon 1
eric.galin@liris.cnrs.fr
http://liris.cnrs.fr/~egalin

Large scenery

- Function representation
- $200 \times 200\ \mathrm{km}^2$ with specific localized details
- Blended primitives authored manually
- Tree size **81 kB**



LIRIS CNRS
Université Claude Bernard Lyon 1
eric.galin@liris.cnrs.fr
http://liris.cnrs.fr/~egalin

Procedural Modeling of Volumetric Landforms

Implicit modeling

Implicit surface representation

Elevation functions can be easily converted to an implicit form
Scalar field representation

Elevation function $z = h(x, y)$ → **Scalar field** function $f(x, y, z) = z - h(x, y)$

f is **not** a distance bound

Distance bound

If f is μ – Lipschitz, then f/μ is a distance bound to the surface [Hart1995]
Fortunately, h is generally λ – Lipschitz, therefore:

$$\mu = \sqrt{1 + \lambda^2}$$

This global bound can be optimized when evaluating along a ray [Galin2020]

f **is** a distance bound → **Scalar field** function $f(x, y, z) = (z - h(x, y))/\mu$

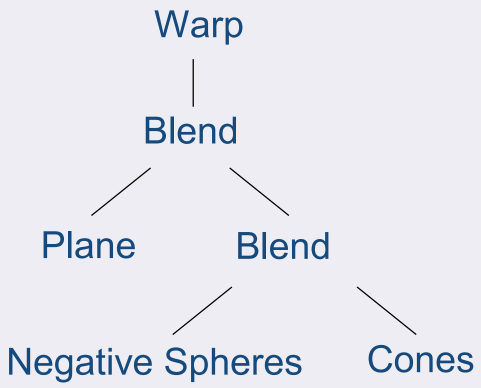
J. Hart. Sphere Tracing. The Visual Computer 12(10), 1995
E. Galin, E. Guérin, A. Paris, A. Peytavie. Segment Tracing Using Local Lipschitz Bounds. *Computer Graphics Forum*, **39**(2), 2020.

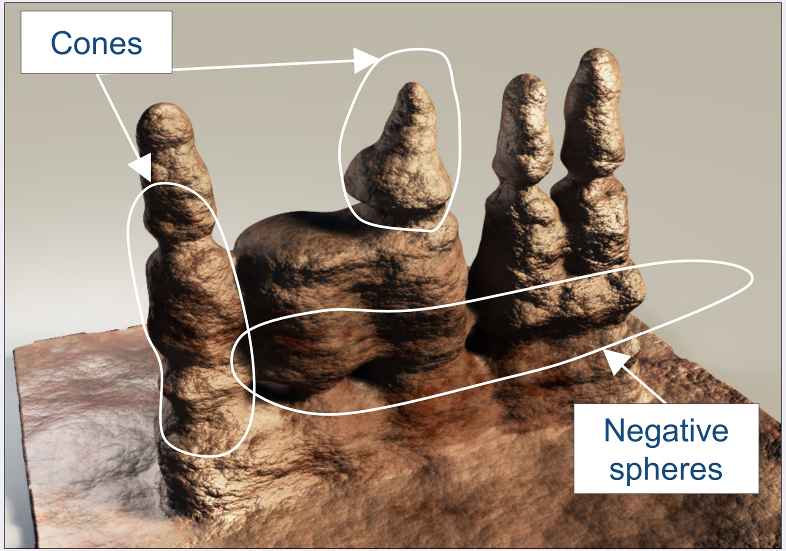
LIRIS CNRS Université Claude Bernard Lyon 1
eric.galin@liris.cnrs.fr
http://liris.cnrs.fr/~egalin

Construction tree

Scalar field

Volumetric primitives with roughness created from spheres, curves [Paris2019]
Operators : blending, intersection, difference, warping [Wyvil1999]





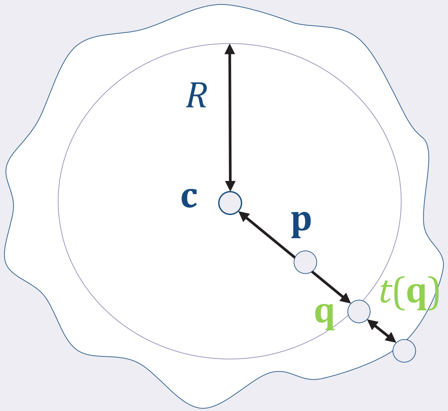
B. Wyvill, A. Guy, E. Galin, The Blob Tree. *Computer Graphics Forum* ***18***(2) 1999
A. Paris, E. Galin, A. Peytavie, E. Guérin, J. Gain. Terrain Amplification with Implicit 3D Features. *ACM Transactions on Graphics*, **38**(5), 2019

Université Claude Bernard Lyon 1
eric.galin@liris.cnrs.fr
http://liris.cnrs.fr/~egalin

Primitives with details

Roughness

- Enhance smooth primitives with noise
- Adding noise may cause holes or floating elements



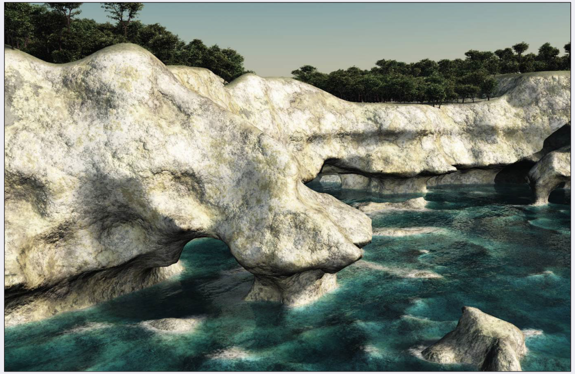
Point skeleton

Star shaped primitive

$$\mathbf{q} = c + \frac{\mathbf{p} - c}{\| \mathbf{p} - c \|} + R$$

$$d(\mathbf{p}) = \frac{\| c - \mathbf{p} \|}{R + t(\mathbf{q})}$$

Turbulence



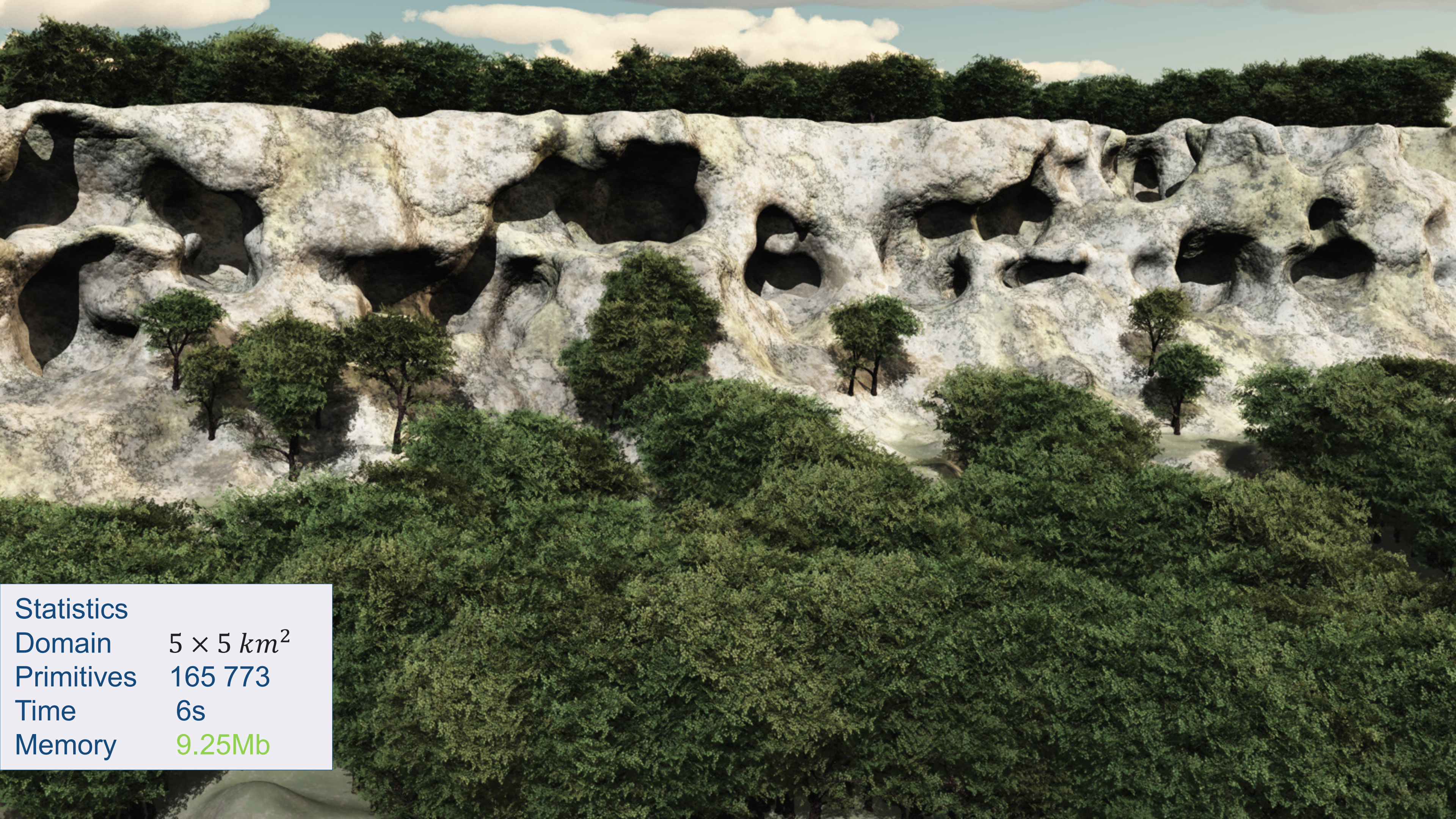
LIRIS CNRS Université Claude Bernard Lyon 1
eric.galin@liris.cnrs.fr
http://liris.cnrs.fr/~egalin

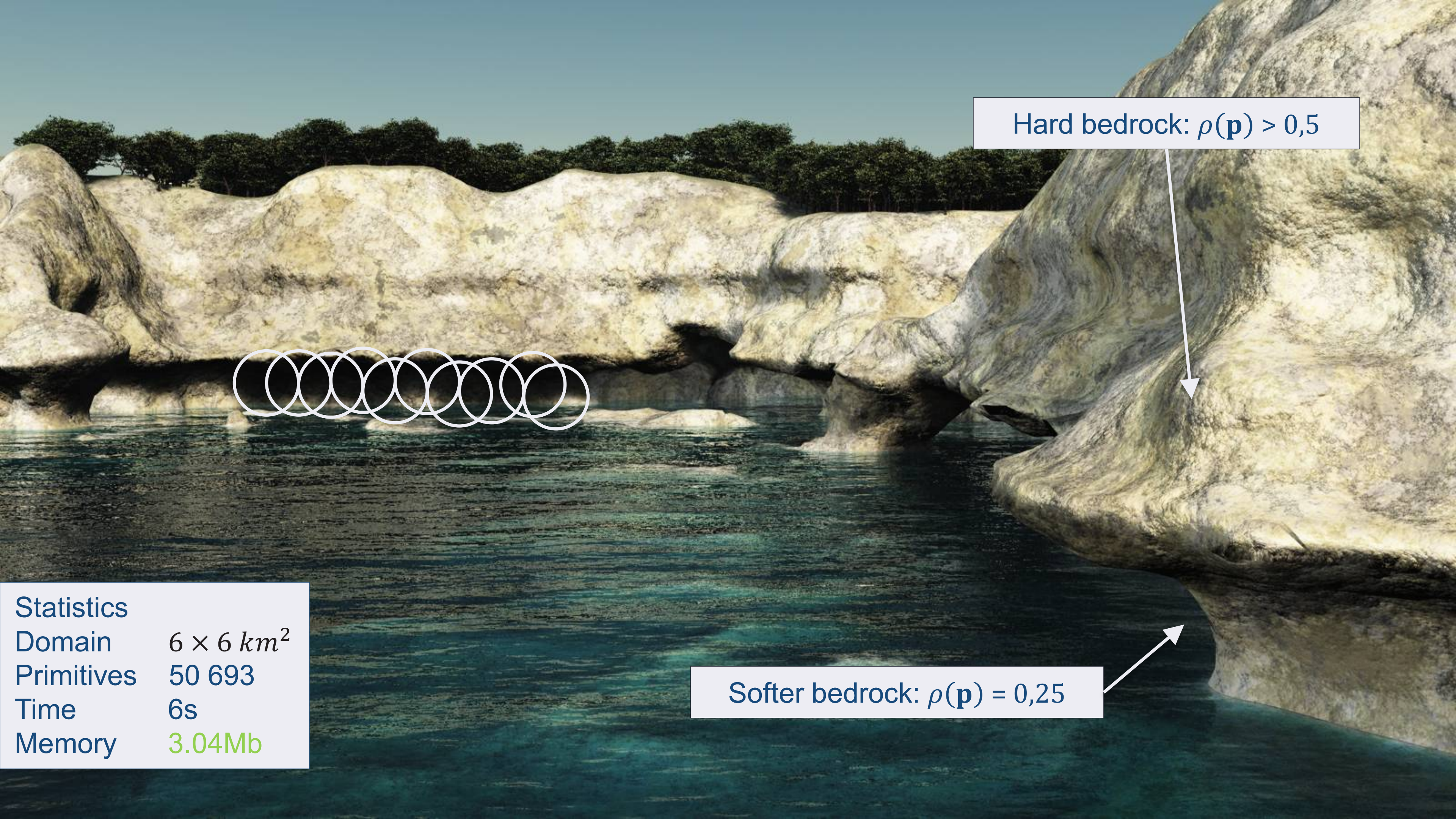
Blocks

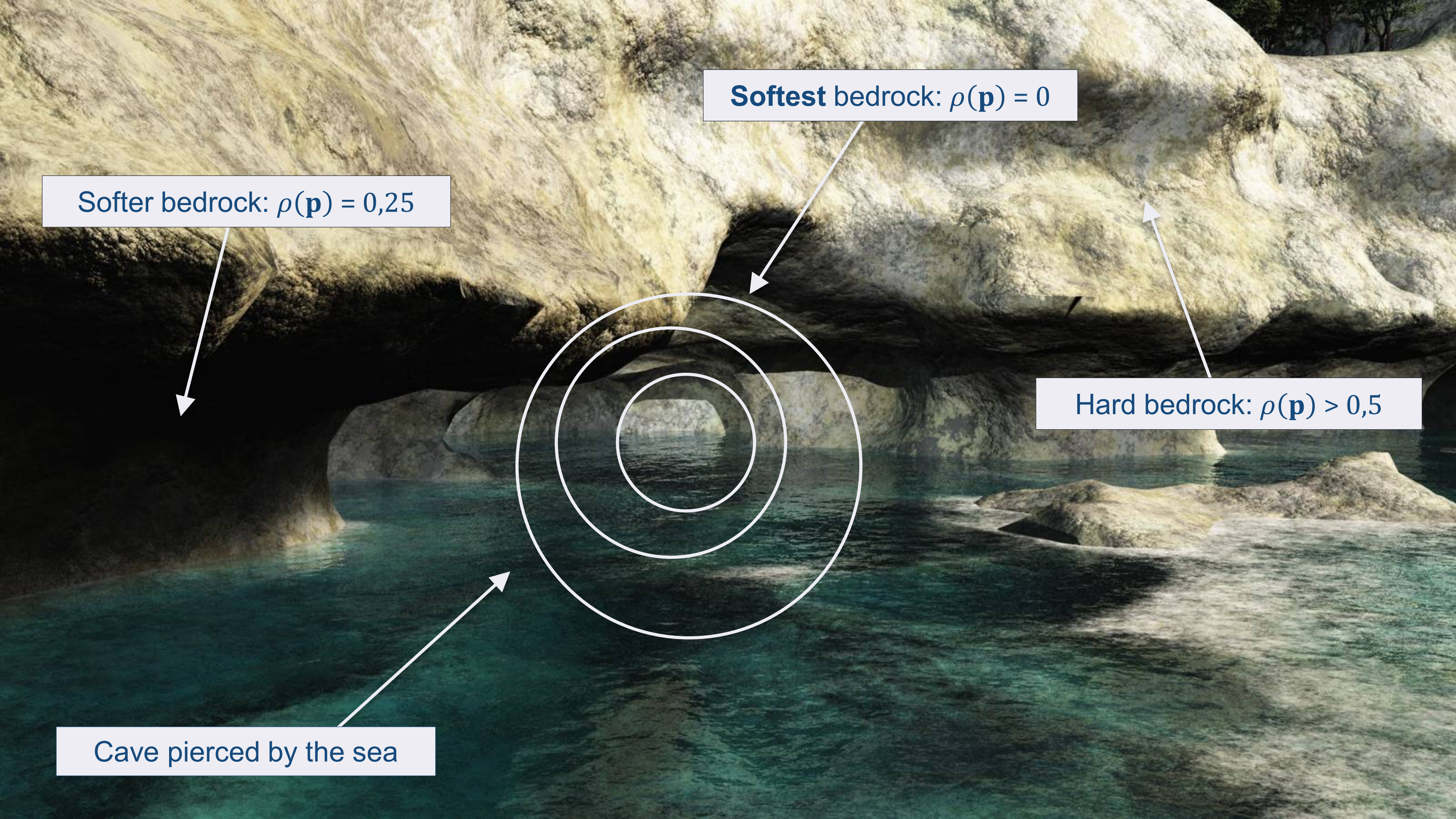
- Aggregate primitives to create an atlas of complex shapes
- Hierarchically reuse blocks

Université Claude Bernard Lyon 1

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


Statistics
Domain $5 \times 5\ km^2$
Primitives 165 773
Time 6s
Memory 9.25Mb


Hard bedrock: $\rho(\mathbf{p}) > 0,5$
Softer bedrock: $\rho(\mathbf{p}) = 0,25$
Statistics
Domain $6 \times 6\ km^2$
Primitives 50 693
Time 6s
Memory 3.04Mb


Softest bedrock: $\rho(\mathbf{p}) = 0$
Softer bedrock: $\rho(\mathbf{p}) = 0{,}25$
Hard bedrock: $\rho(\mathbf{p}) > 0{,}5$
Cave pierced by the sea

Supplementary material

Conclusion

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

Université Claude Bernard Lyon 1

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


© Luck Adwick