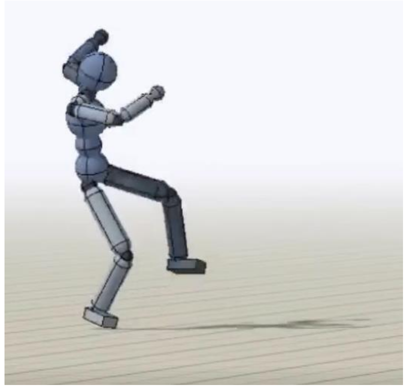
   

# *Advanced 3D Graphics*

## *Part 1: Creative AI*
## *Week 2: Extension to Virtual Worlds*

**Marie-Paule Cani**

# *« Advanced 3D Graphics »*

## *Focus: Graphical techniques with strong links to AI*

***Part 1. From 3D modeling & simulation to creative AI***

1. Expressive 3D modeling : smart geometry controlled by gestures
   – Constructive modeling & shape representations
   – Sculpting, sketching, transfer metaphors
2. **Extension to virtual worlds**
   – **Background: Procedural and layered models for natural scenes**
   – **Expressive creation and control of animated virtual worlds**

***Part 2. Autonomous characters – Animation & control***

3. Motion planning for characters and crowds (Julien Pettré)

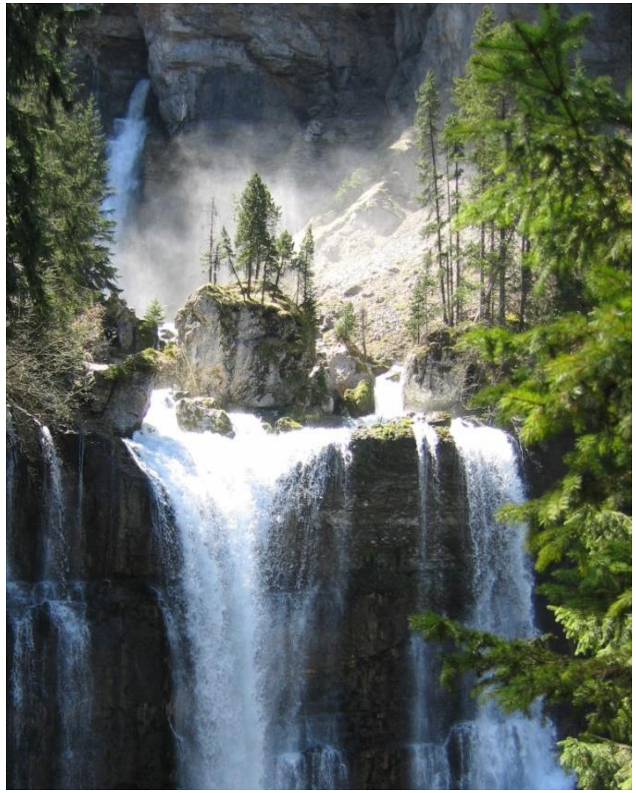
4. Individual motion generation and control

# *Computer Graphics*
# *Impressive virtual worlds… How are they created?*

**@Crytek**]

# *Modeling and Animating Virtual Worlds*
# *Main challenges*

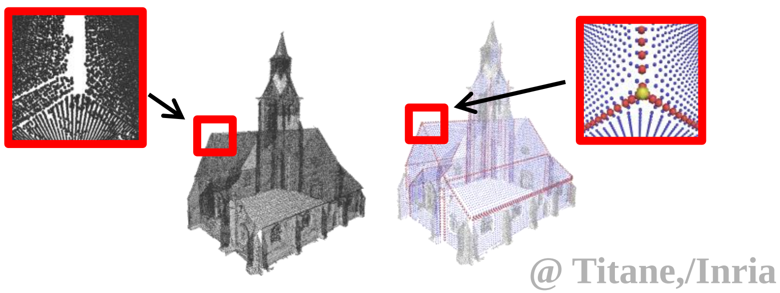
- Huge numbers of elements, all different
- Multiple rules to maintain
  - Shapes: geology, biology, statistics
  - Motion: dynamics, mass preservation
- Complex interactions & time-evolution
  - Water with terrain
  - Ecosystems with resources

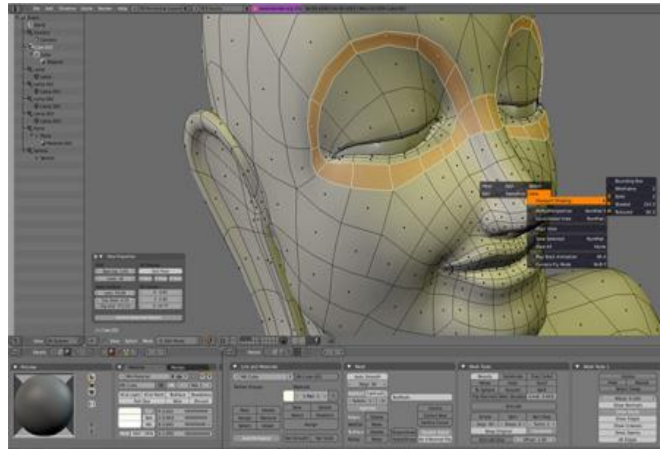


***Can we combine consistency and user control?***

# *Content creation ?*
# *Data reconstruction / Interactive modeling*


- We cannot capture everything!
- How can we create new content?



- Model everything by hand?
- Very tedious
- No help for consistency!

# *In this course : Virtual worlds*
# *Introduction to creation, animation & control*

***Methodology***

A. **Procedural modeling**

B. Layered models for animation

C. Extension of expressive modeling :

Combining knowledge, learning & control

***Case studies***

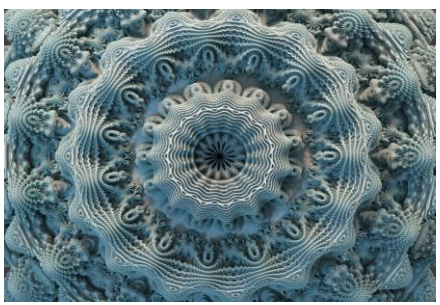
- Modeling terrains
- Liquids & flows: lava, ocean, streams
- Plants and ecosystems

# *A. Procedural modeling*
## *"Modeling with program procedures"*

**Goal:** Modeling **hudge, detailed geometry** (too tedious for artists)

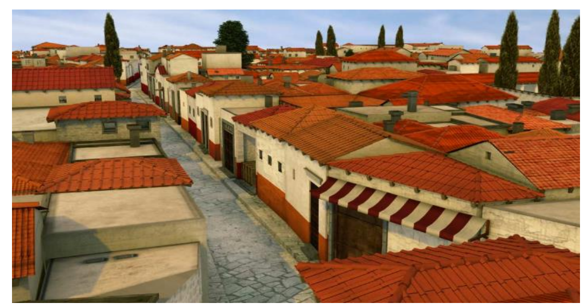
- Applied to complex shapes or scenes
- Examples: Natural scenes, cities, planets, etc

**Principle:** A few rules applied recursively or iteratively



**1. Fractals**

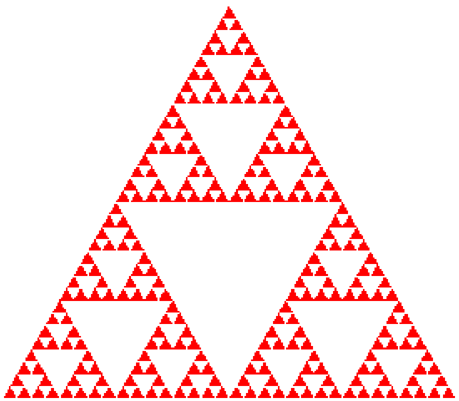
**2. Simulation**

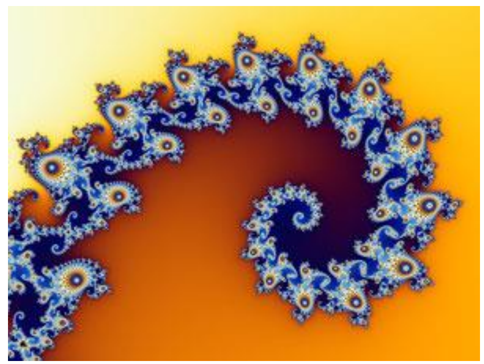


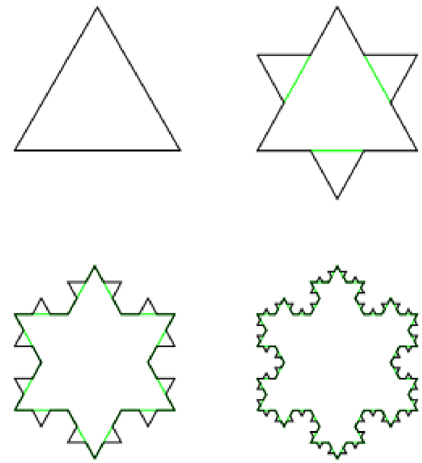
**3. Grammars**

# *Procedural modeling*
# *1. Fractals*

**Fractals:** Recursively add self-similar details





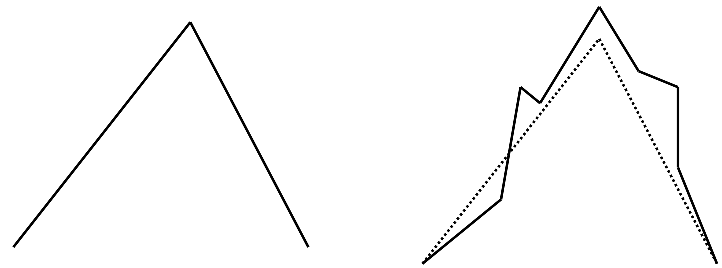


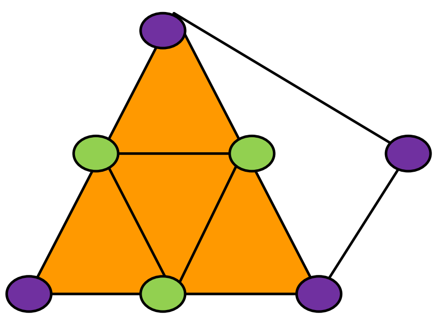
# *Procedural modeling*
# *1. Fractals*

Fractal modeling of terrains

- Needs to be more pseudo-random!
  - Start with a pyramid.
  - Recursively subdivide each face
    - Add random vertical displacements at each iteration

D = random (0, 1/2) . Edge's height



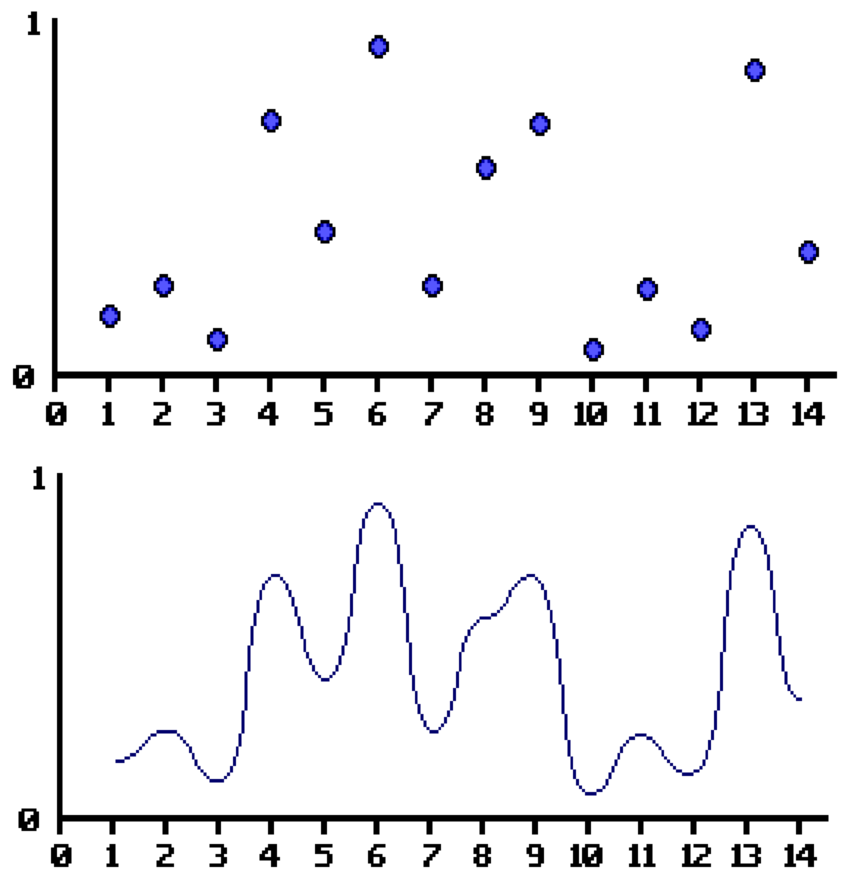


A. Procedural modeling
B. Layered animation
C. Expressive design

# *Terrain modeled as an elevation map*
# *Perlin's pseudo-random fractal noise*

## 1D basis function

- Pre-computation of values (1D table)
- *B(x)* = interpolation of evenly spaced random values
  - – Pseudo-period!
- To reduce smoothness
  *B'(x) = | 2B(x) -1 |*

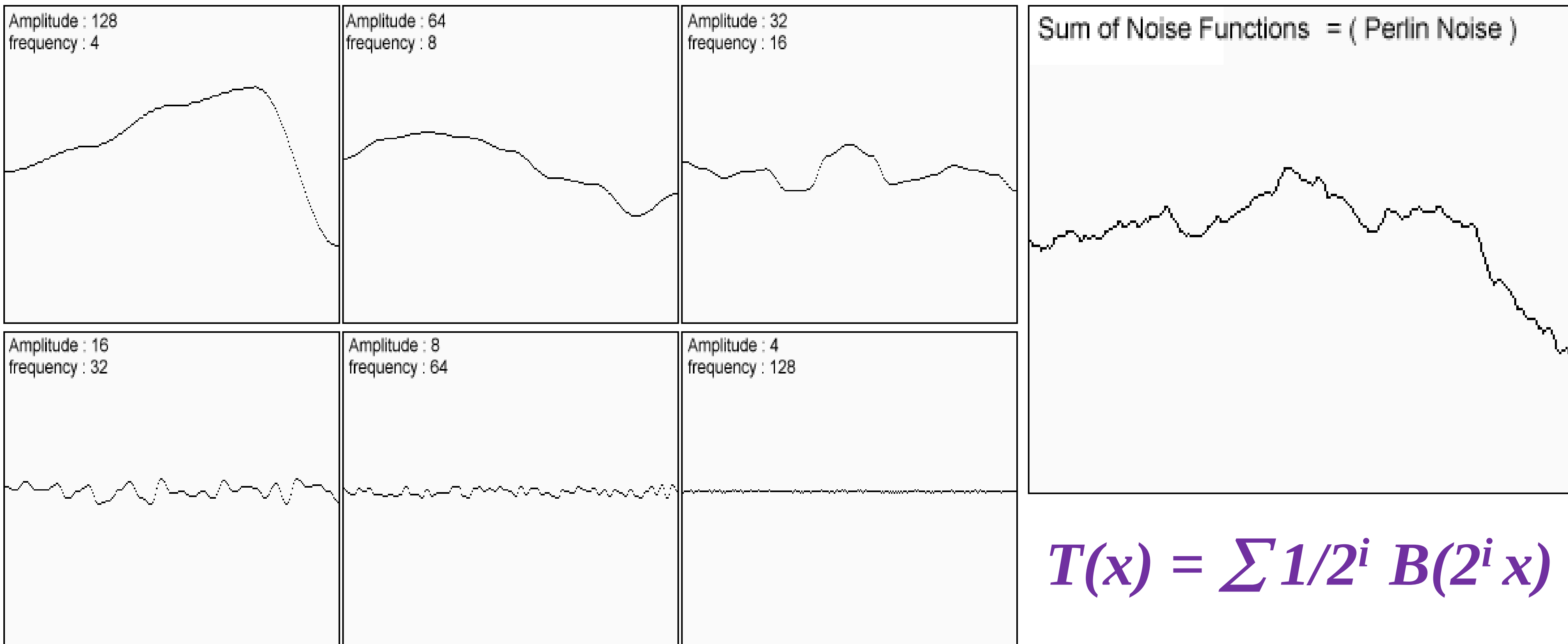


A. Procedural modeling
B. Layered animation
C. Expressive design

# *Terrain modeled as an elevation map*

# *Perlin's pseudo-random fractal noise*

Turbulent noise : sum copies of *B* at different scales



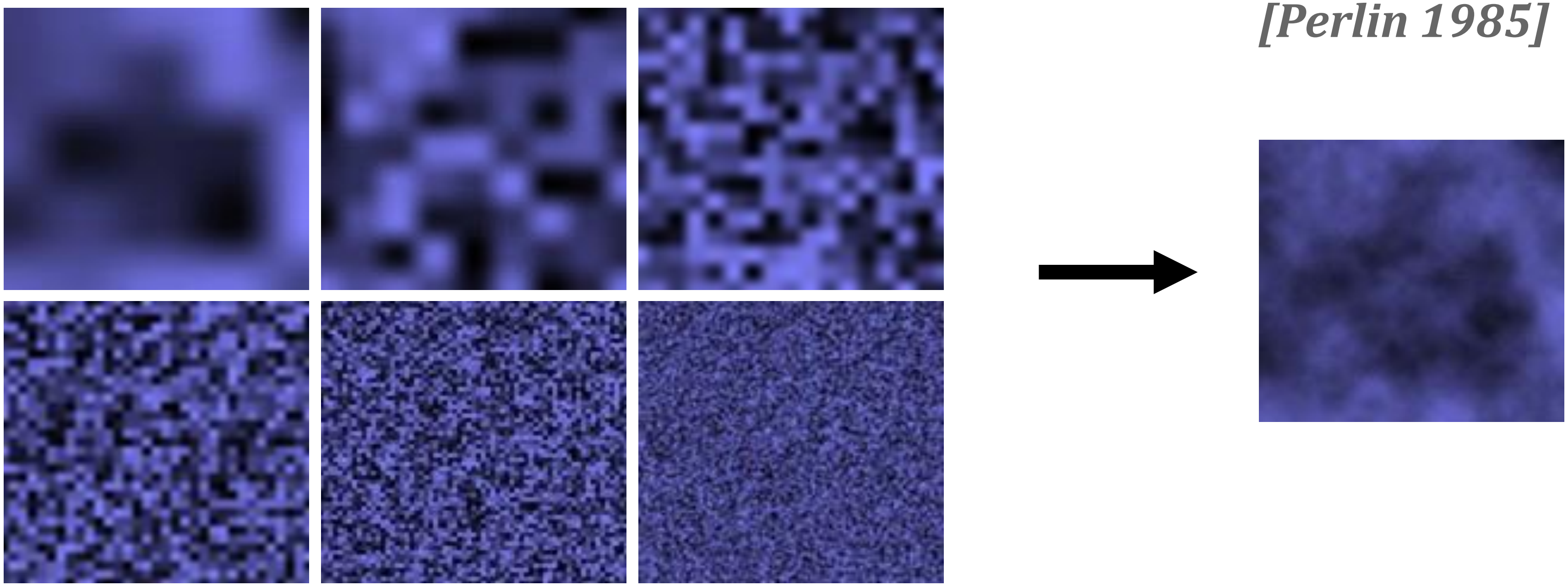

$$T(x) = \sum 1/2^i \; B(2^i x)$$

A. Procedural modeling
B. Layered animation
C. Expressive design

# *Terrain modeled as an elevation map*
# *Perlin's pseudo-random fractal noise*

Works in any dimension : Example in 2D "Perlin textures"

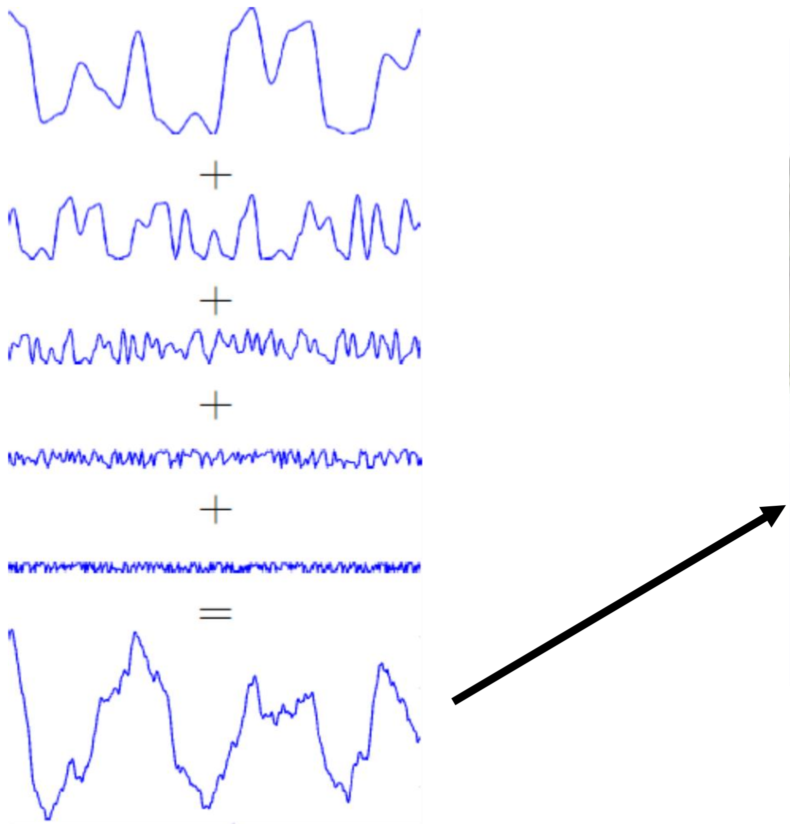
*[Perlin 1985]*

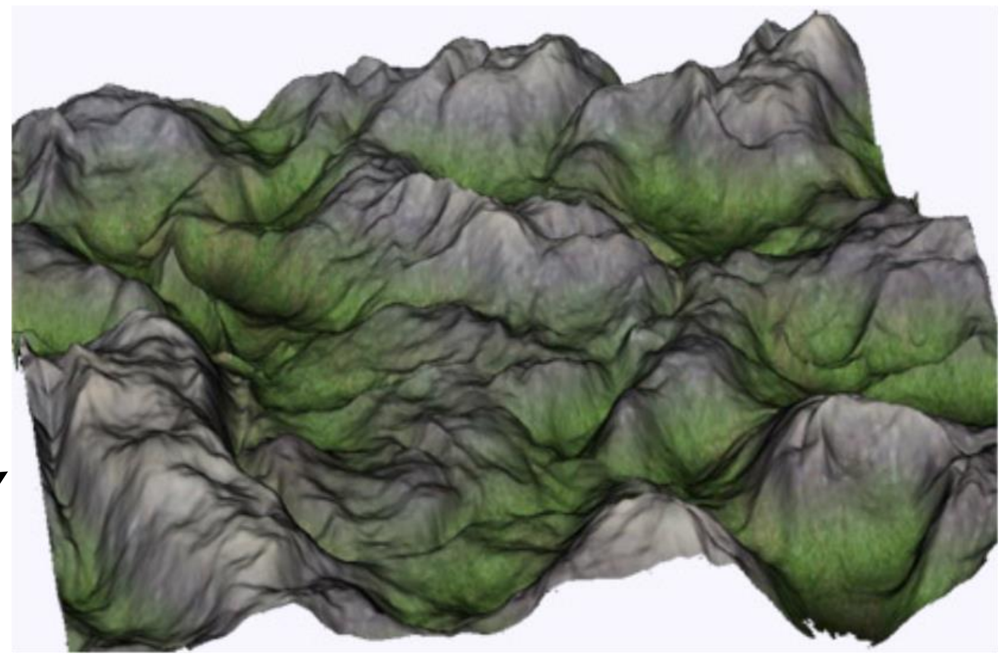


**Random values (x,y) = 1D table [(x + permut(y)) mod n ]**

# *Procedural modeling*
# *1. Fractals*

- Application to terrains : Noise value = elevation

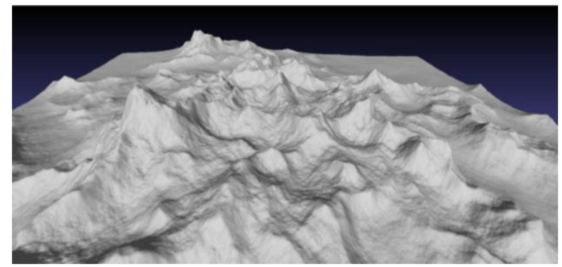




*Naïve terrain*

A. Procedural modeling
B. Layered animation
C. Expressive design

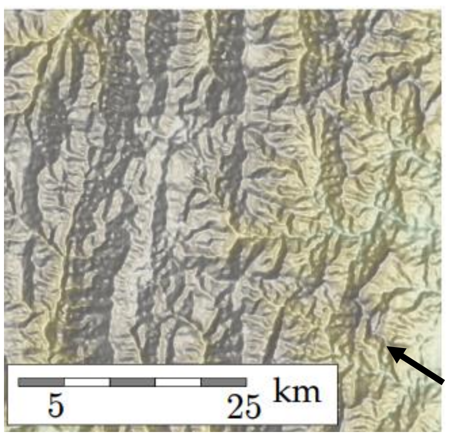
# *Procedural modeling*
# *1. Fractals*

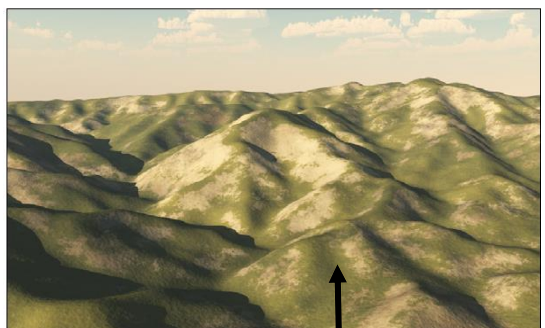


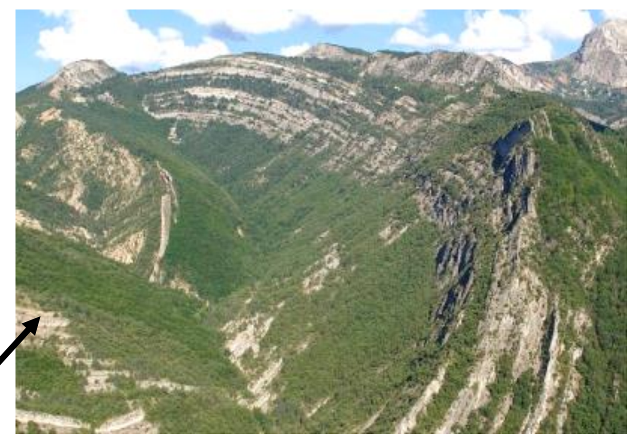
## **Drawbacks of fractal terrains**

- No mountain ranges
- No stream beds and V valleys…
- No eroded cliffs (with consistent vegetation)

## Real terrains





Google earth DEM Photo

# *Procedural modeling*
# *2. Simulation*

**Idea :** Add knowledge on physical rules

- Input initial conditions & rule parameters
- Output consistent result

**Example :** Terrains product of uplift + Erosion

$$\frac{\mathrm{d}h(\mathbf{p})}{\mathrm{d}t} = u(\mathbf{p}) - kA(\mathbf{p})^m s(\mathbf{p})^n$$

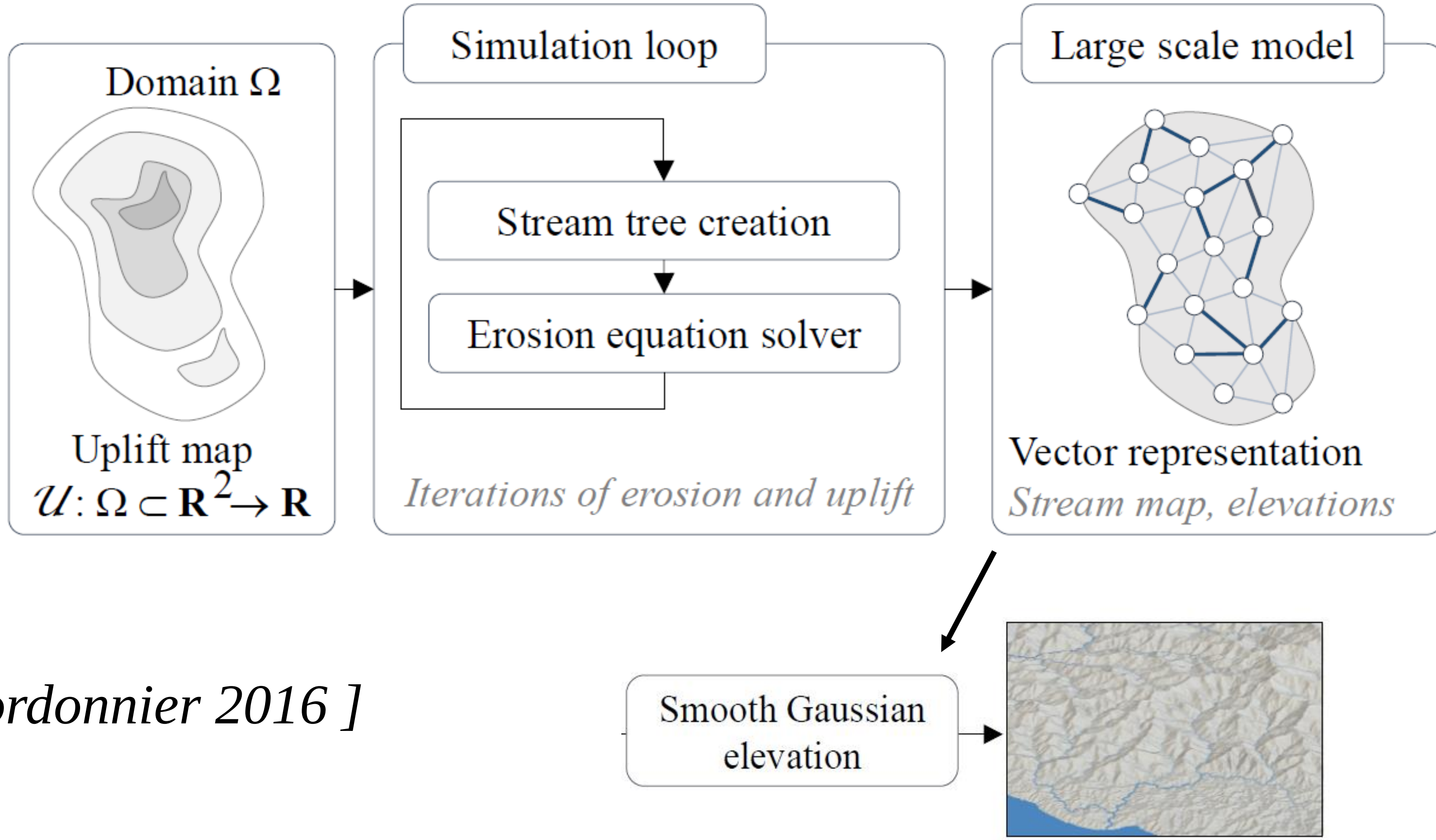
High change speed

Uplift
(mountain growth speed)

Fluvial erosion fct of:
- drainage area A
- slope s

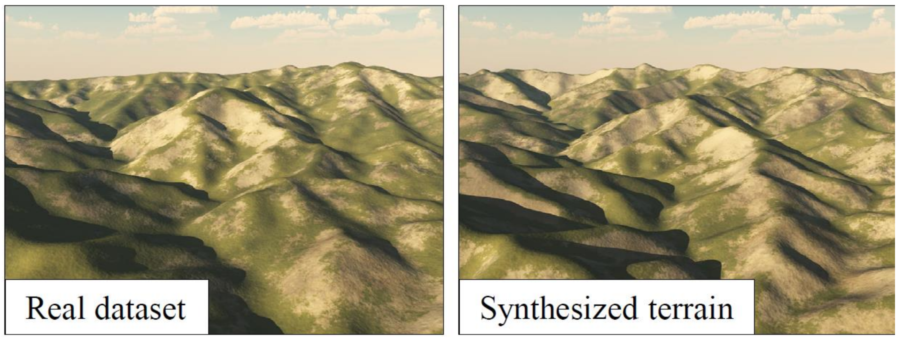
A. Procedural modeling
B. Layered animation
C. Expressive design

# 2. Simulation : Example of terrains Uplift + Fluvial Erosion

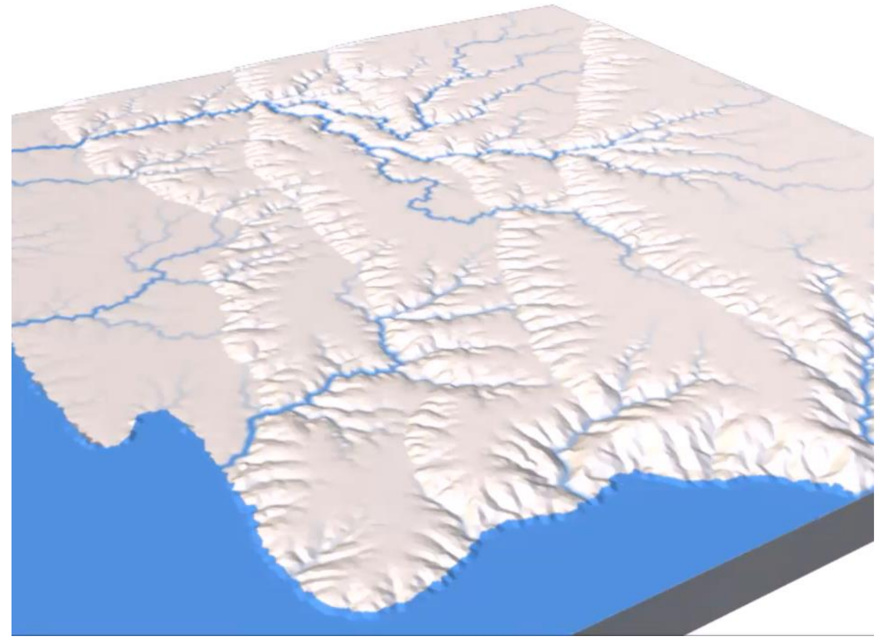



*[Cordonnier 2016 ]*

# 2. Simulation : Example of terrains Uplift + Fluvial Erosion




*[Cordonnier 2016 ]*



***Simulation methods***

+ Maintain consistency (ex: dendritic patterns)
+ Able to capture time-evolving phenomena (model causes)
- Indirect control through simulation parameters… Trials & errors
- Only captures what is in the model (ex: no mountain range)

# *Procedural modeling*
# *2. Simulation : Applications to plants*

## First natural scenes in Computer Graphics

### Method

- Particle systems
  - Point-masses under gravity

### Fake simulation

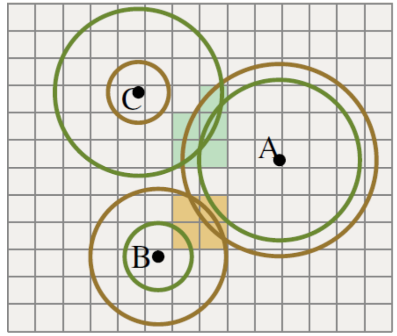
- Grass: trajectories of particles
- Wind particles interact with grass
- Trees: recursively throw particles

*[Reeves 1985]*

# *2. Simulation : Example of plants*
# *Ecosystem simulation*

**Method used by biologists**

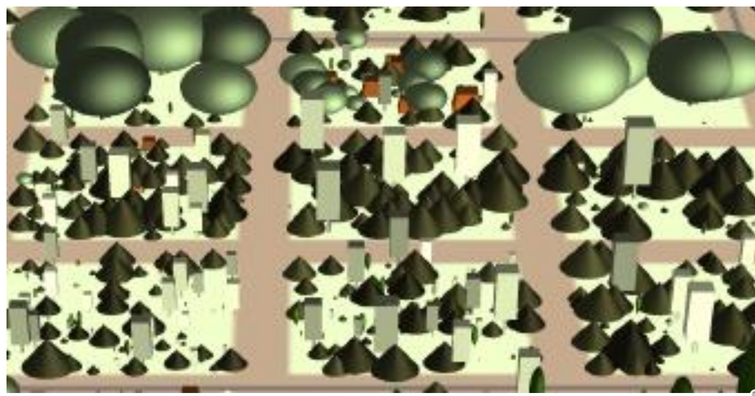
- Plants modeled as pairs of circles
  - canopy and roots
- They compete in cells where they overlap




Each month, update ***Vigor*** from **(*plant type, resources, competition*)**

- If vigor positive, the plant grows
- If negative, probability of depth
- Mature plants seed once a year

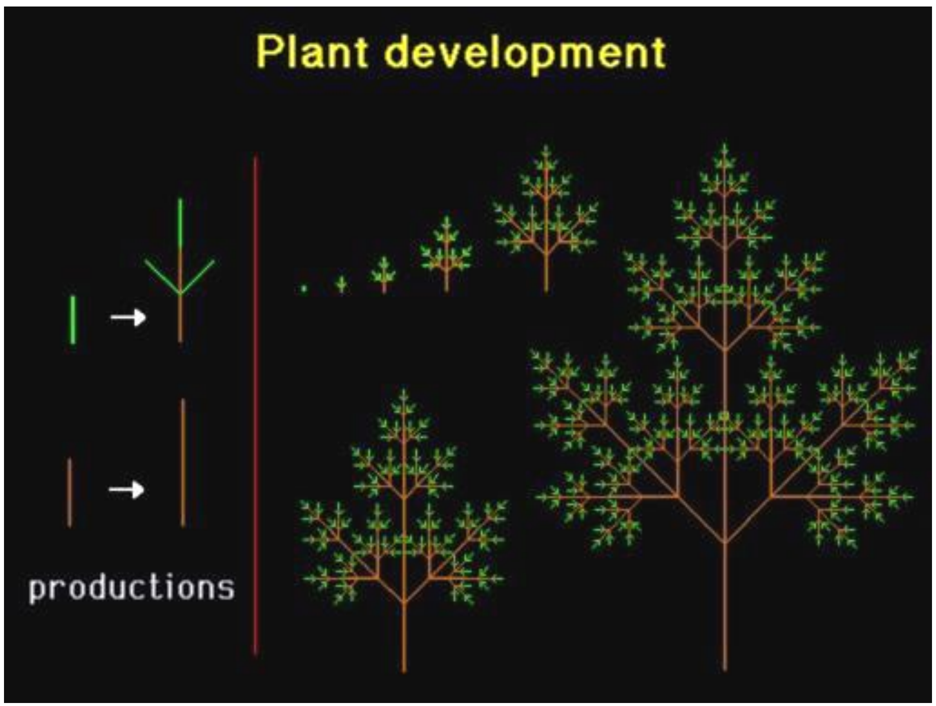
Examples of ecosystems



# *Procedural modeling*
# *3. Grammars*

**Model plant's growth? L-systems** (Lindenmayer 68)

- Simulate progressive plant growth using grammar rules

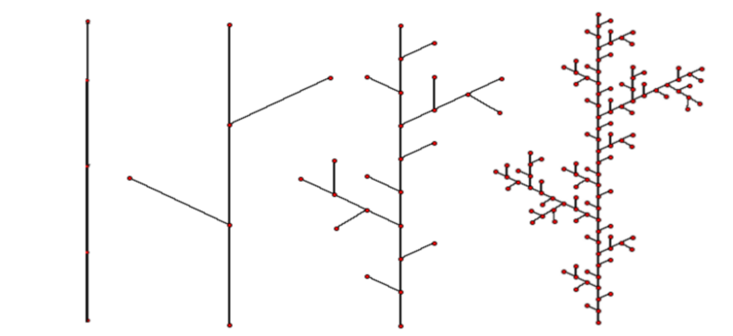



3D embedding + light & damage

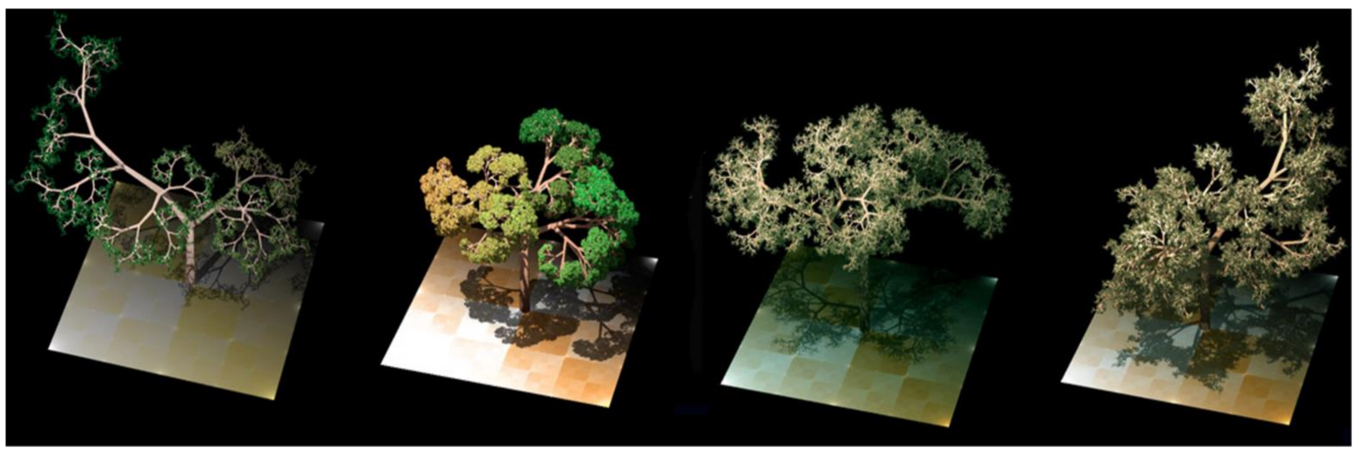
# *Procedural modeling*
# *3. Grammars : Application to plants*

Another substitution rule

$F \rightarrow$ F [+F] F [-F] F , $\theta = 60^\circ$



iteration 1 2 3 4



# *Procedural modeling*
# *3. Grammars : General models*

## Generalization: Shape grammars

1. Set of initial shapes
2. Replacement rules
   - Apply them with a given probability
   - Stochastic variations for embedding created shapes
3. Derivation until « terminal shapes » only
4. Geometrical interpretation of terminal shapes

Many applications!

A. Procedural modeling
B. Layered animation
C. Expressive design

# *Procedural modeling*
# *3. Grammars : City models*

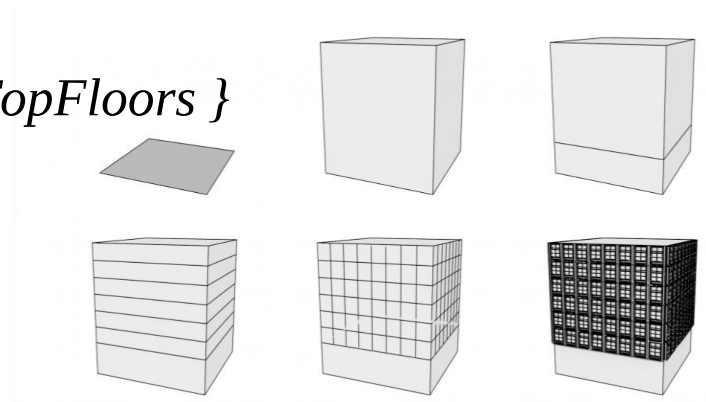
**Example** : Modeling cities

- Intup: streat layout
- Output 3D city

**Rules for a building**

- *Lot -> extrude(10) Mass*
- *Mass -> FaceSplit { sides: Facade }*
- *Facade -> Split("y") { 3: FirstFloor, ~1: TopFloors }*
- *TopFloors -> Repeat("y"){ 1 : Floor }*
- *Floor -> Repeat("x"){ 1 : Window }*
- *Window -> insert("window.obj")*

*[Muller et al., SIGGRAPH 2006]*

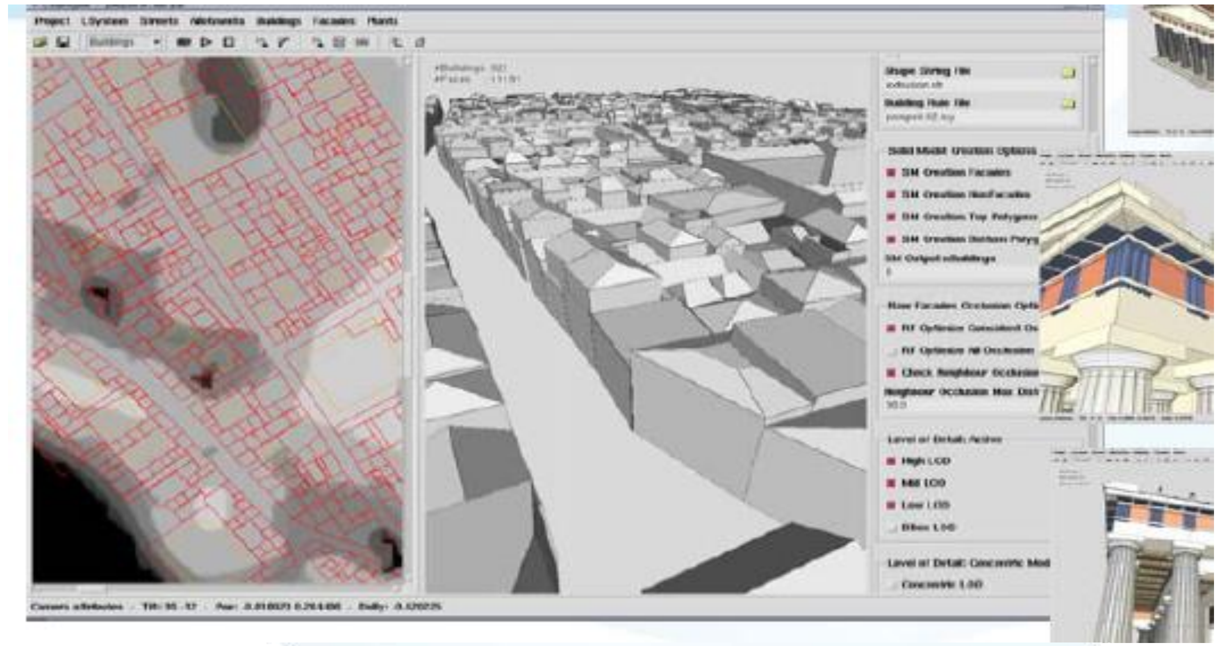


***https://www.gamedev.net***

A. Procedural modeling
B. Layered animation
C. Expressive design

## *Procedural modeling*

# *3. Shape grammars*



+ Can produce LODs

- Not as diverse as real cities

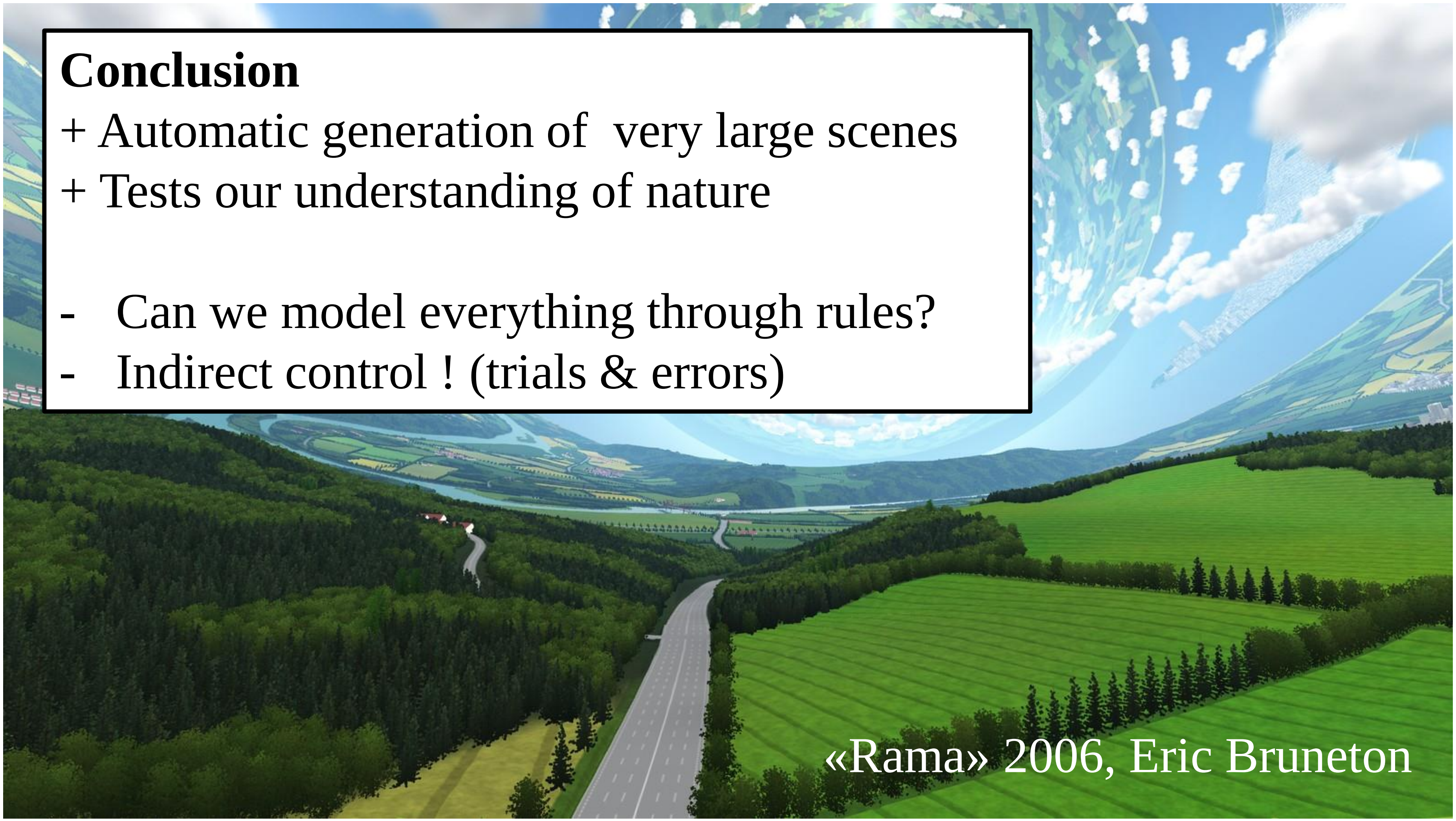
A. Procedural modeling
B. Layered animation
C. Expressive design

# *Procedural modeling*

**Conclusion**

+ Automatic generation of  very large scenes
+ Tests our understanding of nature

- Can we model everything through rules?
- Indirect control ! (trials & errors)

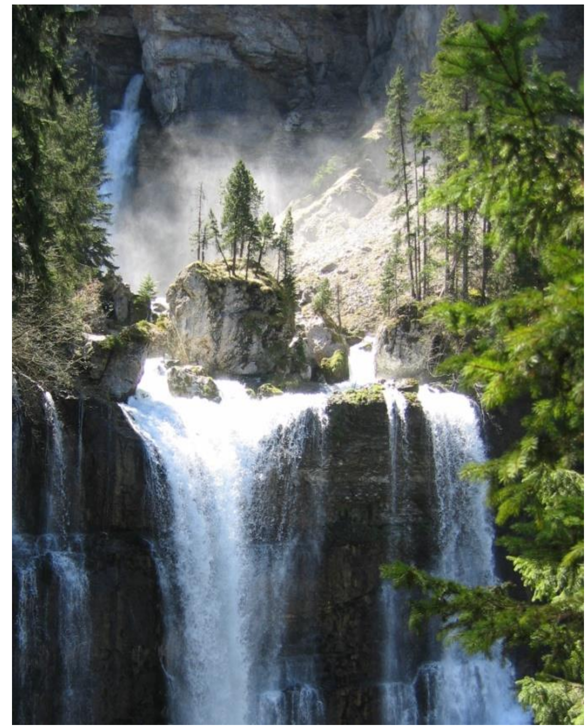


«Rama» 2006, Eric Bruneton

## *Part 1, Week 2*
## *How to create, animate & control virtual worlds?*

### *Extension of expressive design?*

A. Procedural modeling
**B. Layered models for animation**
C. Combining knowledge, learning & control



### *Case studies*

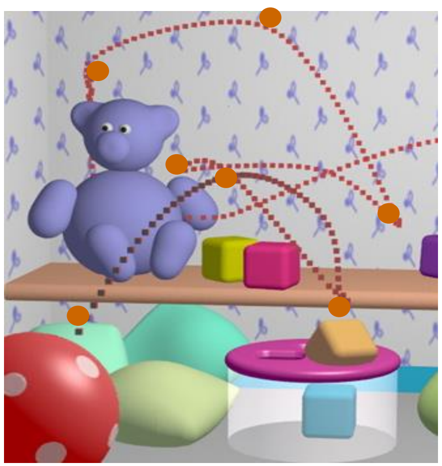
- Modeling terrains
- Liquids & flows: lava, ocean, streams
- Plants and ecosystems

# Background

## *Descriptive vs Physically-based animation*

**Kinematics**

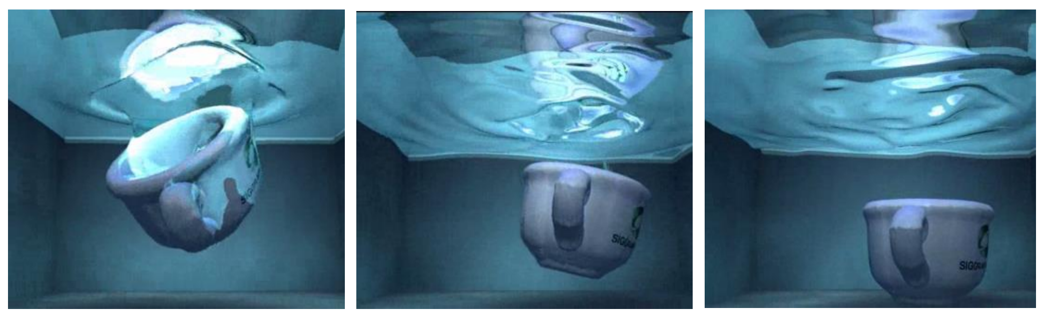
- Kinematic tree
- Interpolating key-frames



Mechanical model
- Parameters
- Initial conditions
- Laws of motion

Integrate differential equations over time
(eg. Navier Stokes for fluids)

**Simulation**



A. Procedural modeling
B. Layered animation
C. Expressive design

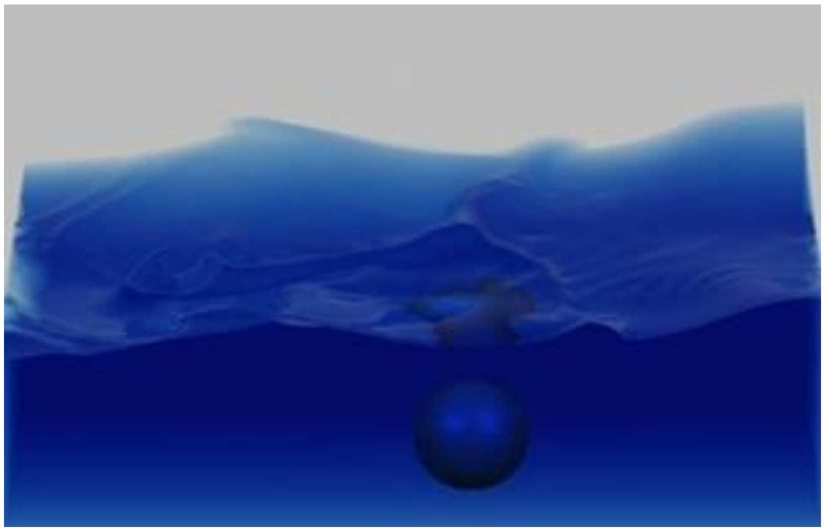
# *Example: Animating liquids*

## *Navier stockes + Eulerian grid + Implicit level-set*

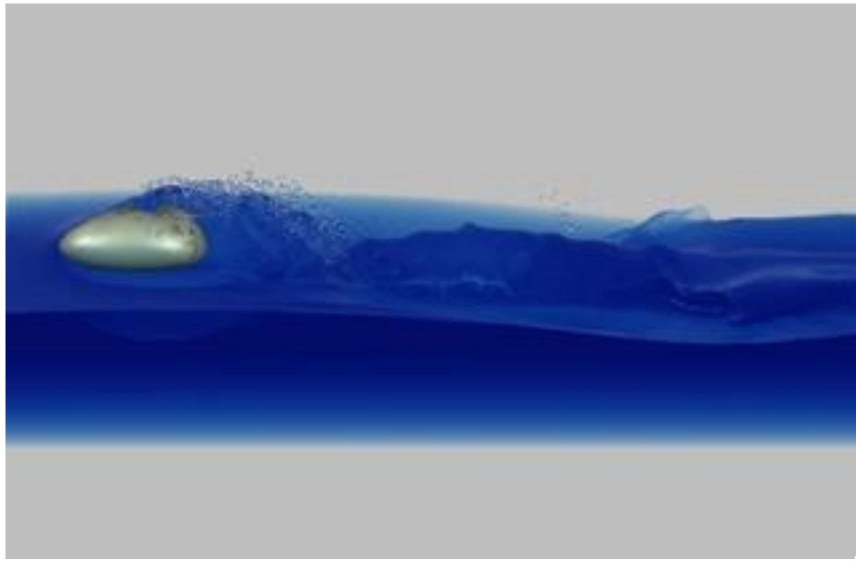
Motivation

- High demand from Film industry

[AntZ 1998]



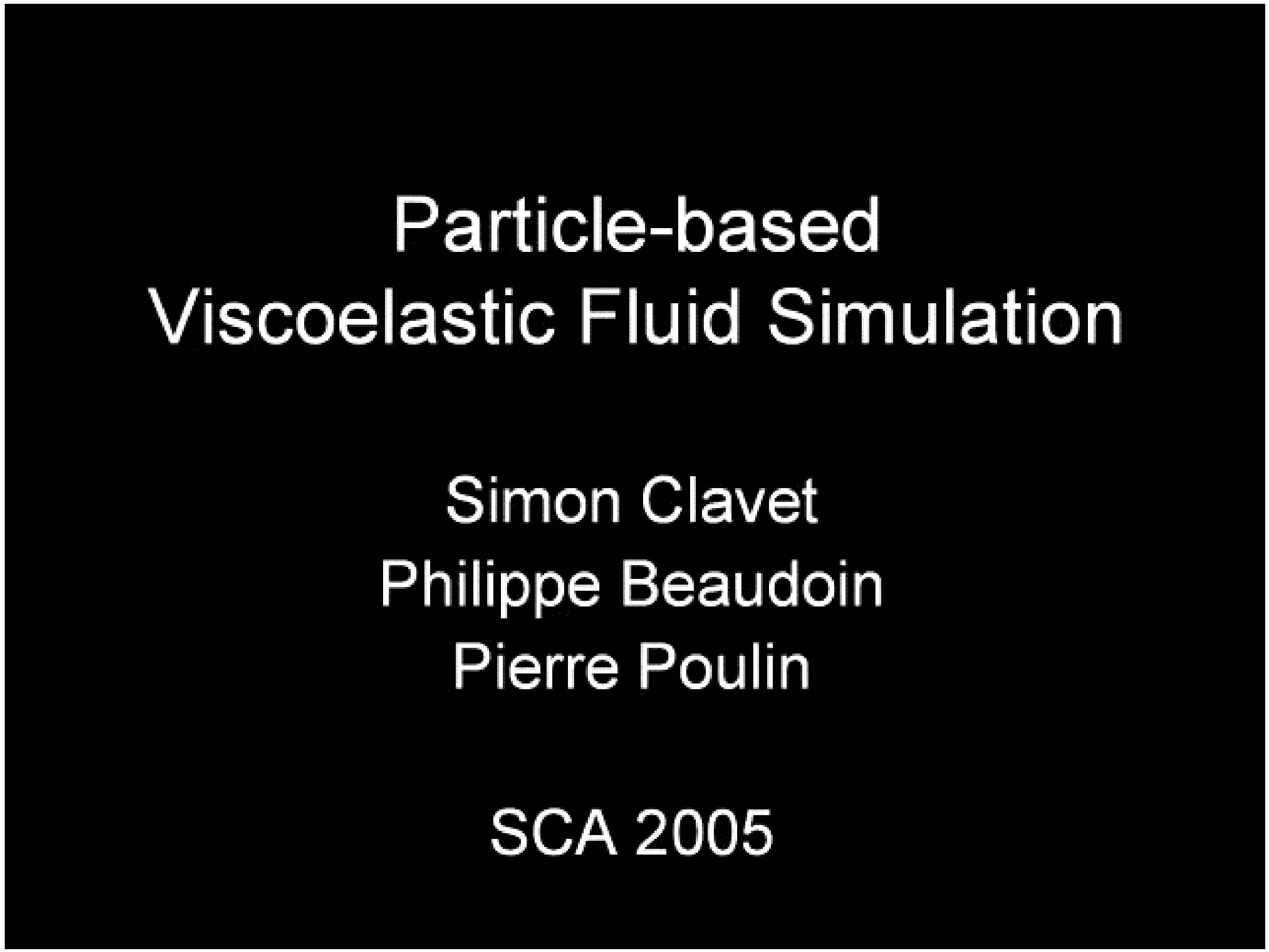
*[Foster & Fedkiw 2001]*



*[Enright et al. 2002]*

# *Example: Animating liquids*

## *Lagrangian particles + Implicit surface*




*[Clavet 2005]*

# *Example: Animating liquids*

## *Vortex particles + Eulerian grid*

Bi-phasic fluids

*[Coquerelle 2006]*

# *Methodology*
# *Layered Models for Animation*

## *Motivation : Complex scenes*

- Grass blowing in the wind, interacting with the feet
- Trees, clouds…
- Characters, clothes, hair..

Different time scales

## *Model choice ?*

- Kinematics / Physics ?
- At which level of details?

(3D geometry, texture…)

# *Layered Models for Animation*

1. Identify sub-phenomena
2. Represent them independently
   – Best model for each feature
   Physics, kinematics, geometry, textures
   – Adapted time & space sampling
3. Couple the sub-models

***Animation loop***

Successive update of each layer + possible retroaction

***Benefits :*** efficiency + easier user control

# *Layered models for animation*
# *Example : Lava flows*

***Aim:*** visual realism

***Challenges***

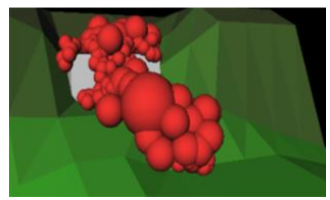
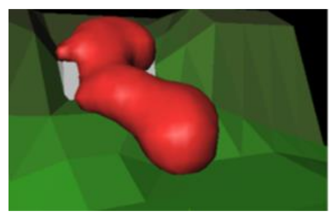
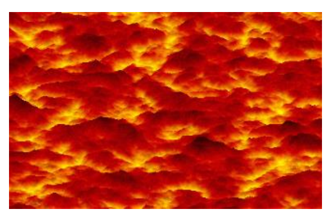
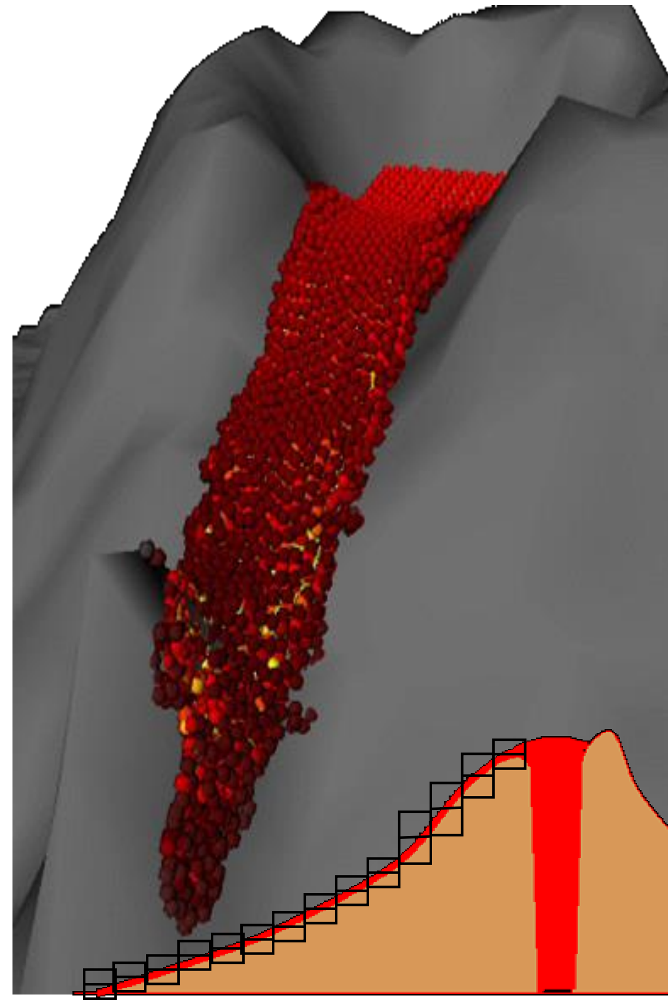
- Viscous liquid
  - Separation, fusion
- Time-varying behavior
  - Viscosity function of temperature
- Two important scales
  - Global trajectories
  - Details of the crust, moving with the flow

*Real lava type « AA »*

# *Layered models for animation*
# *Example 1 : Lava flows*

## *Sub-models* *[Stora 1999]*

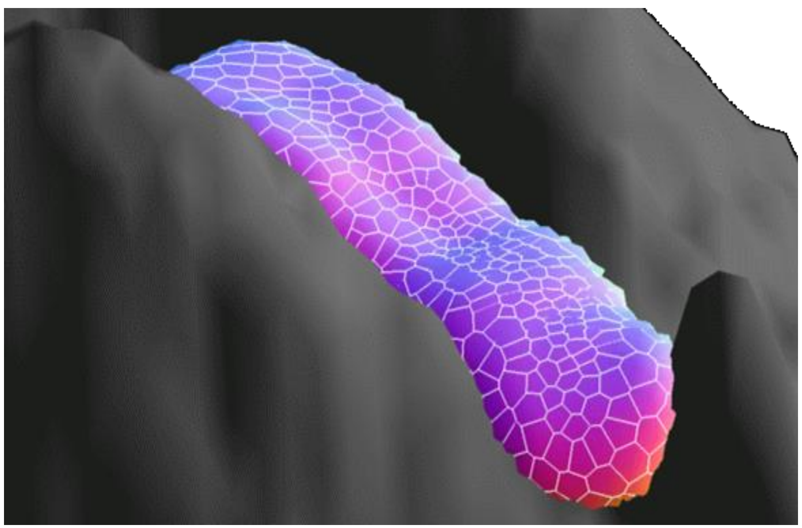
- **Trajectory**
  - Particules + Temperature *[m,X,V,**T**]*
  - Diffusion of T: Heat equation
- **3D surface**
  - Implicit surface
- **Crust details**
  - Procedural texture
  - Time-varying aspect

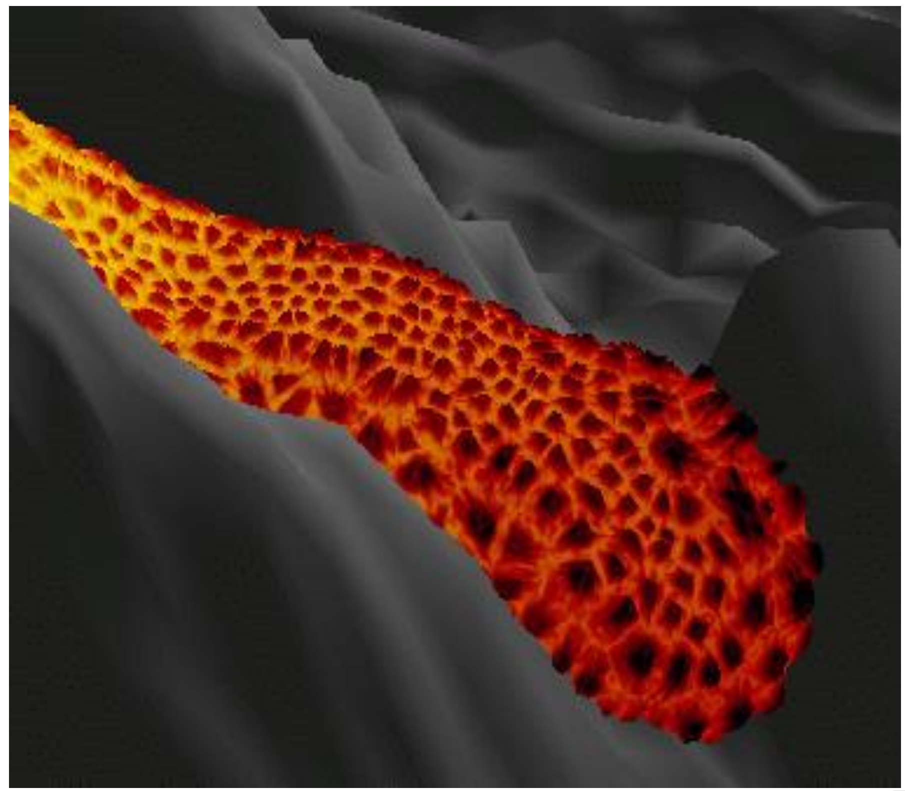





# *Layered models for animation Example 1 : Lava flows*

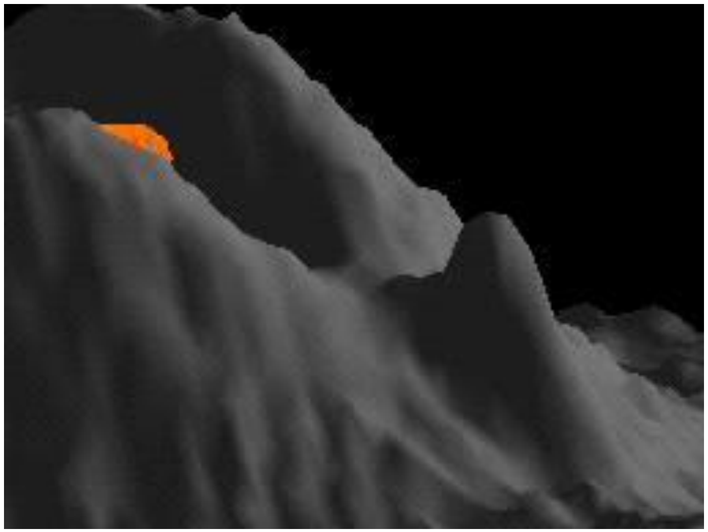
**Particles → Surface → Texture (one way coupling)**

- Particles on top carry “scales” (Voronoi regions)
- Texture function of T(P, t )

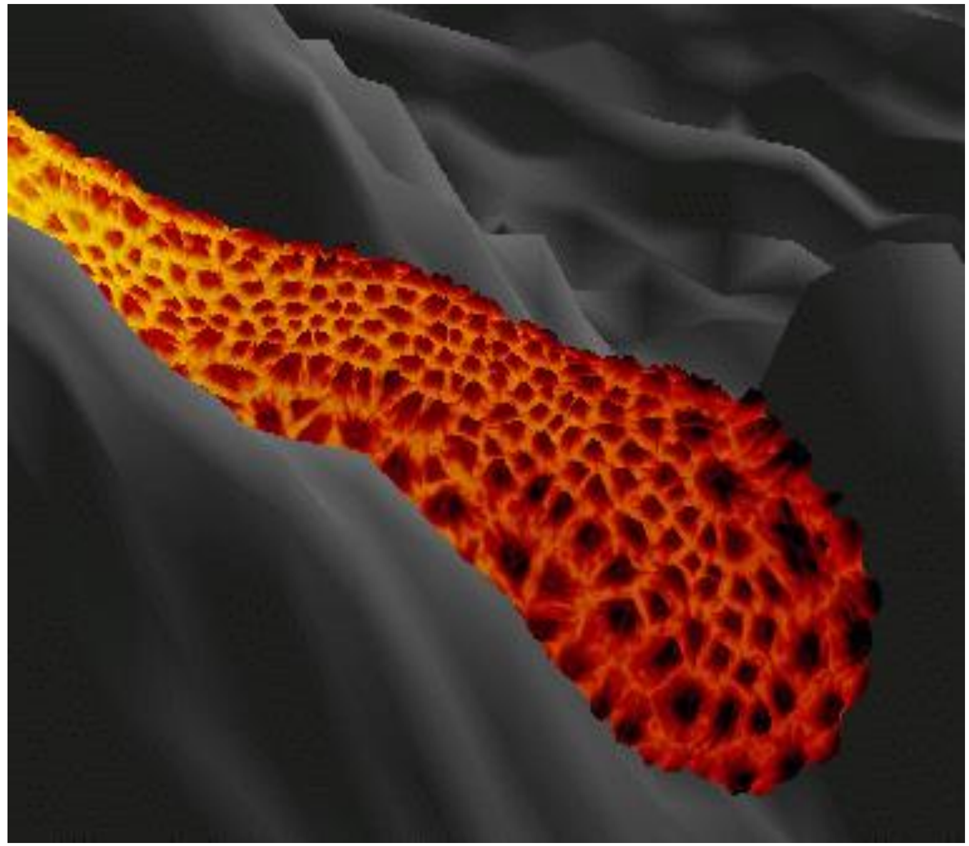




# *Layered models for animation*
# *Example 1 : Lava flows*



*[Stora 1999]*



# *Layered models for animation*
# *Example 2 : Prairies blowing in the wind*

View of a walker in real-time?

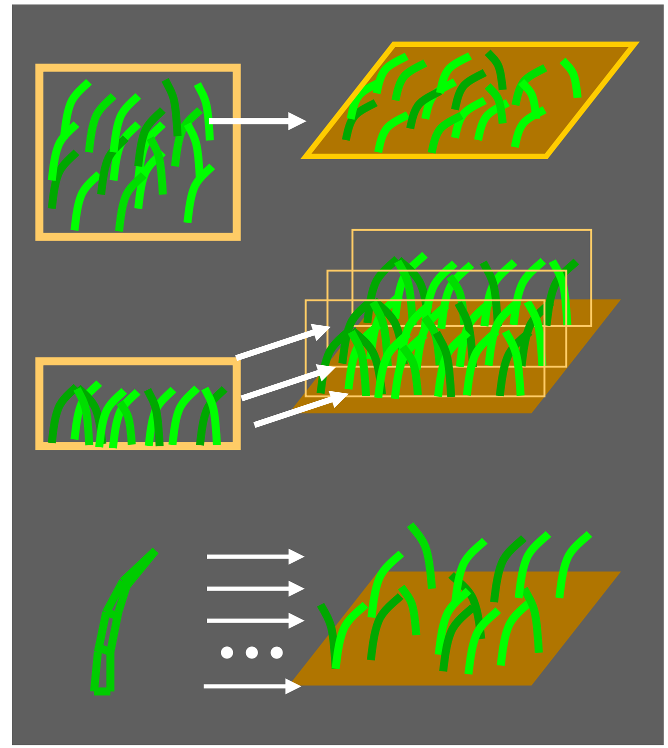
**Challenges**

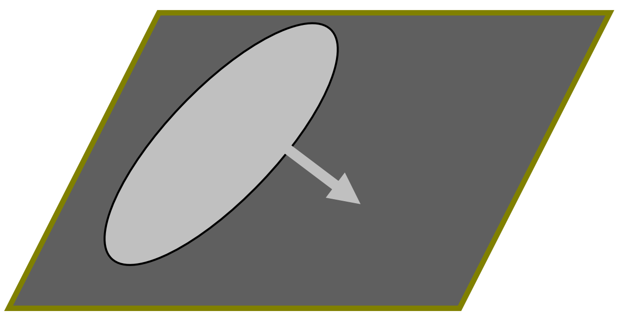
- Number of blades of grass
  - Rendering: aliasing problems
- Control of the wind
  - Breeze, gusps of wind, wind swirls
- Model the **action** of the walker in real-time

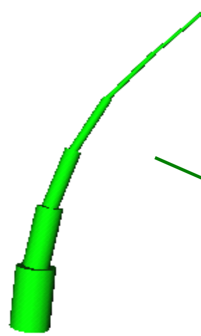
# *Layered models for animation*
# *Example 2 : Prairies blowing in the wind*

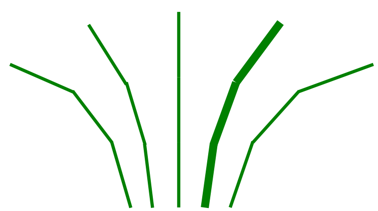
***Sub-models*** *[Perbet 2001]*

- Grass: 3 levels of detail
- Wind model : mask + action
  - Breeze, gusps of wind, wind swirls
- Receiver : blade of grass
  - deformations : pre-simulation







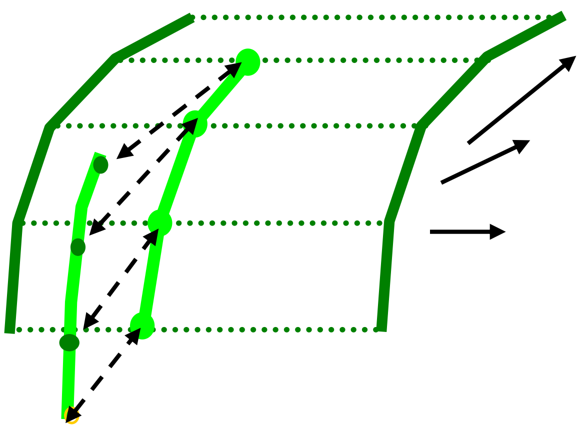


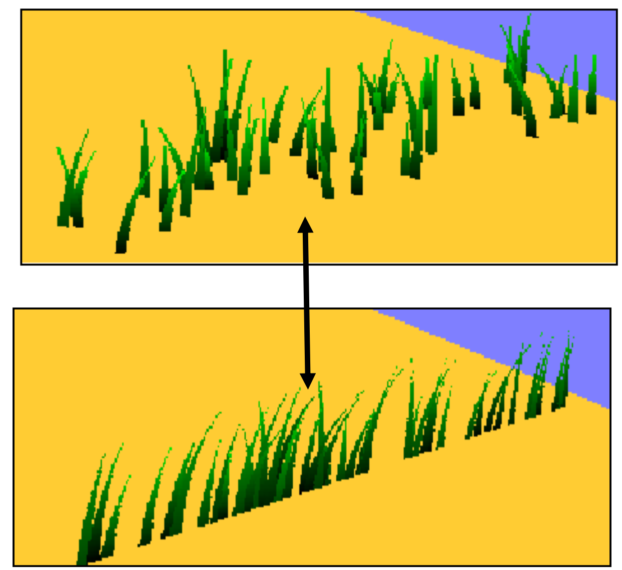
# *Layered models for animation*
# *Example 2 : Prairies blowing in the wind*

## *Transitions between levels of details*

- 3D blades of grass / 2D1/2 texture
- 2D 1/2 Texture / Flat texture

Computed during motion!





A. Procedural modeling

B. Layered animation

C. Expressive design

# *Layered models for animation*

# *Example 2 : Prairies blowing in the wind*

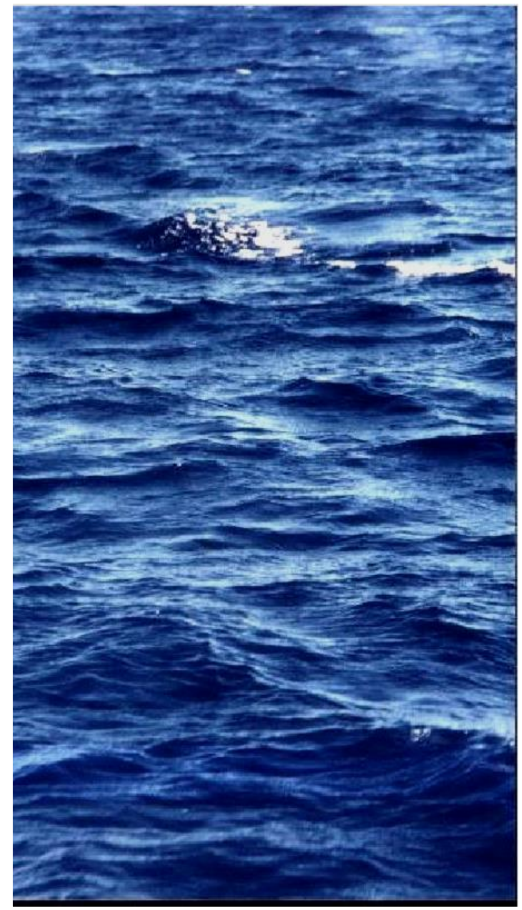
# *Layered models for animation Example 3 : Virtual ocean*

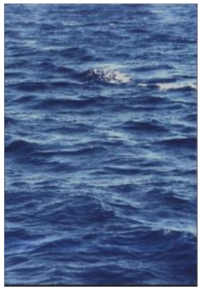
**Goals**

- User slider realism / efficiency
- Moving camera
- Virtually infinite ocean

**Challenges**

- Complex deformations
- Different scales of waves
- Aliasing problems (close/far elements)

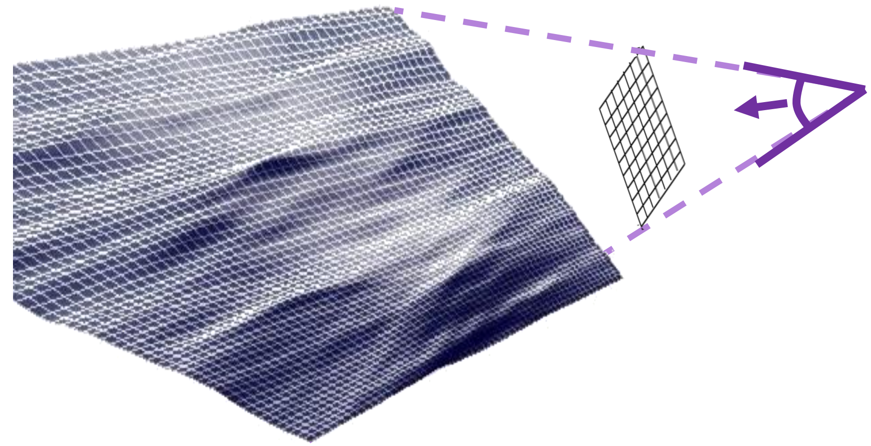


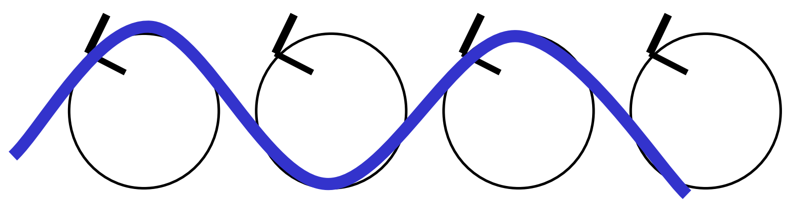


# *Layered models for animation*
# *Example 3 : Virtual ocean*

***Sub-models*** *[Hinsinger 2002]*

- Adapted geometry
  - Non-Uniform mesh
  - Projection of screen pixels!
- Wave trains
  - Mask + action
  - At different scales





Motion triggered by a single wave train

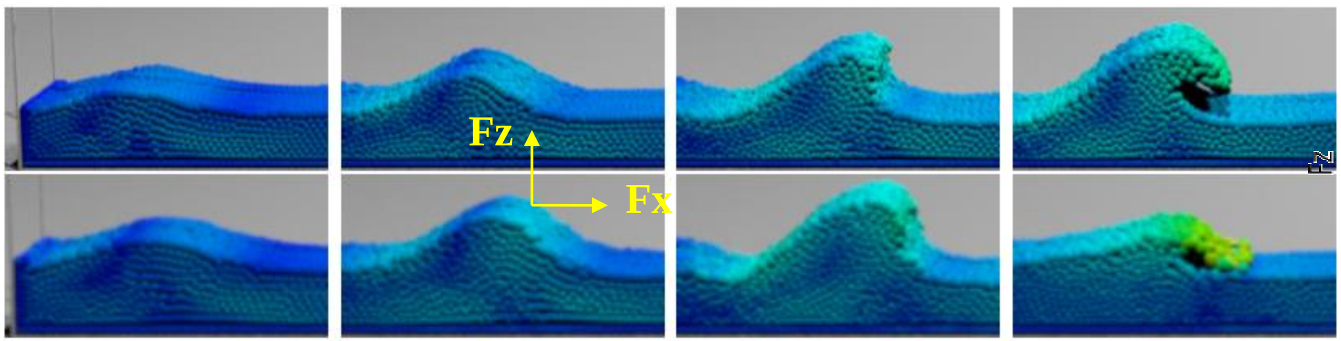
A. Procedural modeling
B. Layered animation
C. Expressive design

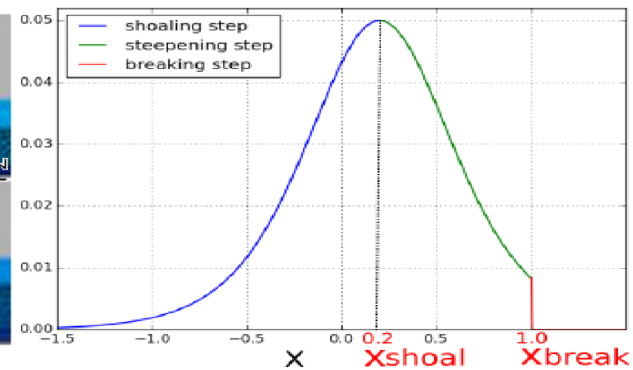
# *Layered models for animation*
# *Example 3 : Virtual ocean*

*[Hinsinger 2002]*

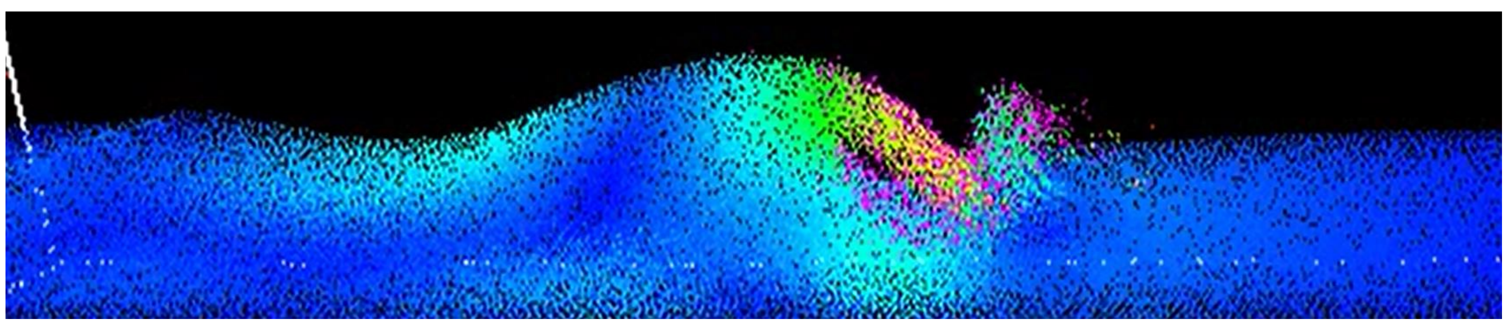
# *Layered models for animation*
# *Example 3 : Ocean with breaking waves*

## ***Lagrangian particles + artificial forces*** *[Brousset 2016]*

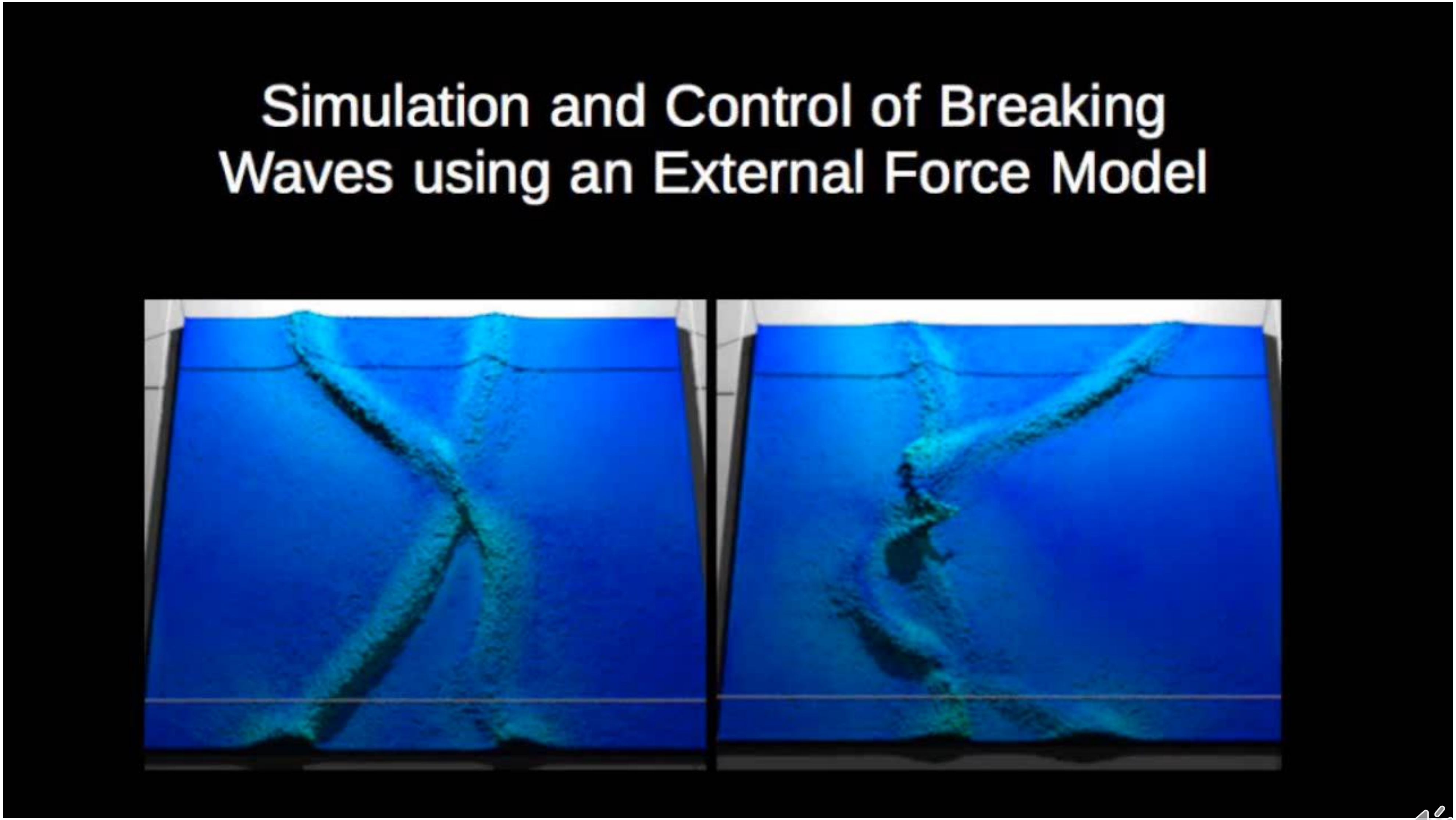



- Spray, foam, bubbles, sand particles added on top (no retroaction)



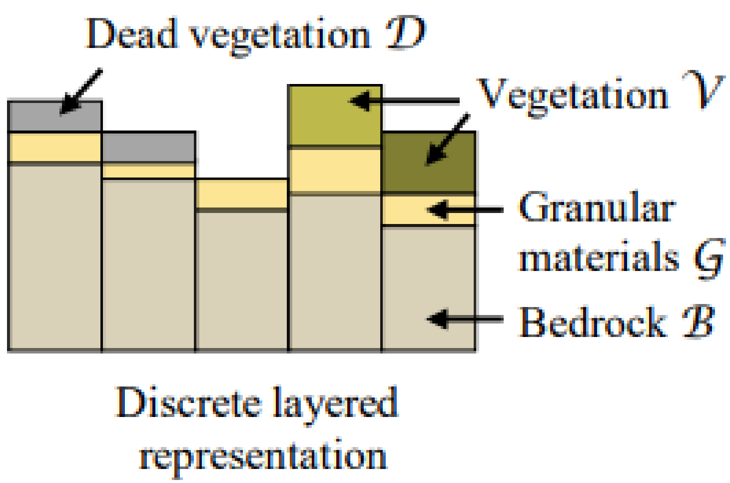
# *Virtual Ocean with waves & foam*



# *Layered models for animation*
# *Example 4 : Erosion + vegetation growth*

*[Cordonnier 2017]*

- Stochastic simulation of a variety of individual events
  - Water flow, stone fall, heath erosion, plant growth
- Consistent thanks to the layers of resources on the terrain




# *Layered models for animation Example 4 : Erosion + vegetation growth*

A. Procedural modeling
B. Layered animation

# *Conclusion*
## *Modeling and Animating virtual worlds*

**Specific methodologies to handle complexity**

- Procedural modeling based on rules
- Efficient animation with layered models
  + Hierarchy of coupled, minimal models
  + Multi-scale in space & time
  + Each phenomena designed and tuned on its own

Indirect control through trials and errors !

- **Authoring virtual worlds? Extend expressive modeling?**

# References


- Brousset, Darles, Meneveaux, Poulin, Crespin. Simulation and Control of Breaking Waves using an External Force Model. Computer & Graphics 2016.
- Clavet, Beaudoin, Poulin, particle- based viscoelastic fluid simulation, SCA'2005
- A Vortex Method for Bi-phasic Fluids Interacting with Rigid Bodies, Coquerelle, Allard, Cottet, Cani  Hal, 2006
- Cordonnier, Galin, Gain, Benes, Guérin, Peytavie, Cani. Authoring landscapes by combining ecosystem and terrain erosion simulation, *ACM TOG*, *Siggraph 2017*.
- [Enright, Marschener and Fedkiw 2002, Animating and rendering of complex water surfaces. SIGGRAPH 2002, ACM TOG.
- Foster and Fedkiw 2001] Practical animation of liquids. SIGGRAPH 2001.
- Hinsinger, Neyret, Cani (2002). Interactive Animation of Ocean Waves. *SCA*.
- Perbet, Cani (2001) Animating Prairies in Real-Time. *ACM I3D*.
- Ken Perlin (1985), An image synthesizer, SIGGRAPH'85.
- Reeves and Blau. Approximate and probabilistic algorithms for shading and rendering structured particle systems, SIGGRAPH 1985
- Stora Agliati Cani Neyret (1999). Animating Lava Flows. *Graphics Interface*.