Representing 3D data structures

EPFL, STI, IMT, LOA
Stéphane Gobron
Institute of Movement Sciences
CNRS, Aix-Marseille University, France
stephane.gobron@univmed.fr

May 23, 2008
Lausanne, Switzerland
Content

Context, topic, and approach

- Polygonal cellular structure
- Cellular networks
- GPU-accelerated 3D CA

Including an introduction to CA and GPU programming as options
Introduction

- Line of thought
- Today’s talk
Line of thought

1990~98
- Geometric modeling

1998~2000
- Polygonal cellular structure for natural phenomena sim.
  - Hybrid-CA [h-CA]

2000~01
- Cellular networks [CN]

2003~05
- GPU-accelerated multi-dim. CA

2006~
- GPU prog. – e.g. retina sim.

Future
- Real-time generalized model
- Emerged 3D visualization
- VR visu.
Today’s talk

- Polygonal cellular structure for natural phenomena sim.
- Hybrid-CA [h-CA]
- Cellular networks [CN]
- GPU-accelerated multi-dim. CA

2006~
2003~05
1998~2000

Need for 3D visu.
GPU prog. – e.g. retina sim.
1. Polygonal cellular structure

- Context
- Introduction to CA – optional
- First approaches
- Results
Context

- General idea
  - The world can be seen as a set of cells
  - Cells interacting with their neighborhood
  - Cells follow rules

- General concepts
  - Scales
  - Dimensions
  - States $\leftrightarrow$ identity
Introduction to CA (1/4)

- Abstracting life, *The chess-board is the world; the pieces are the phenomena of the universe; the rules of the game are what we call the laws of Nature* (T. H. Huxley): “a rough definition of CA”
- Set of cells
- Table of rules: applied to all cells, time discretization
- Examples in 1D and 2D
Introduction to CA (2/4)

- e.g. a symmetric rule
Introduction to CA (3/4)

- *e.g.* an asymmetric rule

- Unpredictable emerging behavior
Introduction to CA (4/4)

- In terms of # of possible key combinations

\[ \rho = n^{(N+1)} \]

with

- \( N \) the # of neighbors
- \( n \) the # of thresholds – e.g. Boolean => 1 threshold

- Unpredictable behavior
- Efficiency of convergence
- Information visualization beyond imagination
- Very large # of information and computation
A first polygonal cellular model

- A hybrid model between using polygons and cells

For every polygon $P$ an independent local coordinate system with cells $C_{I(x,y)}$

Regular grid $\Gamma$

Input polygons $P$

Cellular sampling

$n$ layers –hypertexture– « $\lambda$ »

Considering the set $N$ of the neighbors of a cell $C$ with state $S$, each cell’s type (i.e. layer $\lambda_i$) a function $\Phi$ is associated so that

$$S_{t+1}[C_{I(x,y)}] \leftarrow \Phi_{\lambda}(S_t[C_{I(x,y)}])$$
A second polygonal cellular model

- Macro and micro cells
- Only useful cells are subdivided
- Gain in memory and precision

Non used micro cells
Non used macro cells
Results

- Patina and corrosion
- Fracture propagation
- Peeling simulation

[GC01a]  [GC01b]  [GC99]
2. Cellular networks

- Improvements
- Model
- Applications
  - Computer Graphics [CG]
  - Cellular Automata [CA]
  - Image processing [IP]
Improvements

- Allowing local behavior to be parallelized
- Getting close to regular grids
- Having multiple grid types
- Reaching real-time computation and visualization

=> use of cellular networks
Model

- Storing only useful voxel space cells

From any set of polygons...
...to a harmonious set of cells

Dynamics double linked list
multi-dimensional structure:
only useful voxels are stored
CG application

- From simple topology...
- ...to complex behavior

$$p_{c,t+1} = \frac{p_{c,t} \sum_i p_{i,t} \left(1 + \frac{(h_i h_c)}{d(C_i,C_c)}\right)}{1 + k} - \Delta p$$

e.g. watercolor local pigment concentration change of state

[GF06]
CG application for reconstruction

- E.g. architecture
  - Stone erosion
  - Water concentration
  - Vegetal propagation

Original building

Reconstruction + animated lichen growth

[EG05]
CA application

- Conway's "Game of Life"

\[ \forall c \left( g_s = \text{active} \right) \alpha 
\]
\[ \text{if} \left( k_p \neq 2 \right) \alpha \quad p_{t+1} \leftarrow \left( k_p = 3 \right) \]

- Maze like

\[ \forall c \left( g_s = \text{active} \right) \alpha 
\]
\[ \text{if} \left( k_p \neq 2or4 \right) \alpha \quad p_{t+1} \leftarrow \left( k_p = 3 \right) \]

- Diffusion / regression

\[ \forall c \left( g_s = \text{active} \right) \alpha \quad p_{t+1} \leftarrow h + J_1 \sum_{rij < R_1} c_n(p_i) + J_2 \sum_{R_1 < rij < R_{21}} c_n(p_i) \]
IP application #1

- Realistic effect: retina’s cellular topology
- Simulating retina: natural cellular topology => fovea

Ocular sphere

Connecting webcam & contour detection

Parallel effects

Cellular connectivity

Spreading cones

[DGGH05]
IP application #2 (1/2)

- Realistic effect: natural layer pipeline on retina
- Simulation accelerated using GPU

(a) Retina’s biological layers

(b) Corresponding architecture

(c) Simplified CS pipeline to implement

[GDH07]
IP application #2 (2/2)

- Simulating natural retina cellular layer pipeline

Webcam input
Bipolar
Saturated bipolar
Amacrin
Ganglion -m
Ganglion -static
CA vectorisation
CA – area reconstruction threshold MIN & MAX

[GDH07]
3. GPU-accelerated approach

- Intro to GPU – optional
- Improvements
- Model
- 3D results
- Perspectives
Introduction to GPU-programming

=> Accelerated computation using *Shaders*

=> Gain can vary from x1 to x13000 depending on:
   - the graphics card, the OS, the motherboard, the RAM
   - the 3D matrix size
Improvements

- Defining generic approaches allowing formal CA, i.e. asymmetric rules
- Real-time
- Real 3D => programmable graphics hardware
- Inside / outside visualization
- Any CA dimension:

*E.g. for hexagonal model (d)*

[GBM08]
Model for voxelspace

- “Testbed” and data pipeline

1. 3D-data
   Matrix of densities of size \([a, b, c]\)

2. Pre-processing
   3D \(\Rightarrow\) Tex2D

3. “Tex2D”
   Single 2D image representing \(n\) slices of \(n \times n\) pixels images with \([n = 16 \parallel 64 \parallel 256] > \{a \parallel b \parallel c\}\)

4. 3D\(_{\text{VH}}\)CA testbed
   User interface options for bottom-up exploration of CA-functions:
   - Real-time CA computation
   - Real-time multiple visualizations
   - Asymmetric and symmetric rules
   - Floating point variation with visual cellular behavior
   - Multiple CA – pipeline process

5.1 3D-data
   Tex2D format

5.2 Rendered Volume
   - Surface simulation
   - X-Ray like simulation

5.3 CA function
   *i.e.* key-code

6. Output process
   Tex2D \(\Rightarrow\) 3D

*Unpublished work: please do not distribute*
Model

- Tomography
- Realistic real-time surface implicit rendering

*Unpublished work: please do not distribute

* [GBC08]
3D results

- Identification of CA families, e.g.: 3D fractals

*Unpublished work: please do not distribute

* [GBC08]
3D results

- Rendering samplings and X-ray simulations
- High resolution implicit surface simulation

*Unpublished work: please do not distribute

*[GBC08]*
3D results

- Benchmark

*Unpublished work: please do not distribute

*[GBC08]*
Perspective

- Virtual reality testbed for real 3D
- Skipping precomputation step
- $\Rightarrow$ Real-time source computation $\Rightarrow$ direct 3D visualisation
- Higher resolution matrices algorithm: especially $1024^3$
Summary

- Geometric modeling
- From polygons to real 3D
  - Polygonal cellular structure
  - Cellular networks
  - GPU-accelerated 3D CA
Demo

- $3D_b^{vNCA}$ testbed
- => Downloadable software
  
  http://www.laps.univ-mrs.fr/~gobron/
Many thanks!

- Question?
References – http://www.laps.univ-mrs.fr/~gobron/