Procedural Wood Texture Generation, Solid Texturing and Simulation

Jérémie Dumas
Supervisor: Pierre Poulin

Laboratoire d'Informatique Graphique de l'Université de Montréal

May-July 2011

ENS DE LYON

Université de Montréal

Motivations

Why Wood Textures ?

- Wood $\rightarrow$ one of the most often used material in CG
- Various applications for biologists, artists, graphic designers
- Can use either 2D textures or 3D textures (*solid textures*)
- Faithful simulation process for texture rendering
- Modelling of knots, arbitrary shapes, mechanical forces, etc.

2D Textures versus 3D Textures

Advantages and Drawbacks

2D Textures

+ Fast and straightforward, can create high quality results
+ Multiple use (*color maps*, *bump maps*, *displacement maps*)
– Mapping issues : arbitrary shape parametrization ?

3D Textures

+ Easy of use, no parametrization issues
– Difficult to represent as a simple function $\rho(x, y, z)$
– Can be memory expensive (table of $10^3 \times 10^3 \times 10^3$ elements ?)

Botanical Considerations

Main Phenomena

- Annual ring pattern (*earlywood* : wider, *latewood* : tighter)
- Knots (conical shape, more present around the pith)
- Heartwood and sapwood (reddish color, etc.)

Other Factors

- Wind and gravity forces
- Light and water availability
- Growth environment (fences, diseases, insects, temperature. . .)

Botanical Considerations

Illustration

Figure 1: Section of a Yew branch [1].

1. Source : Wikipedia. Author : MPF.

Previous work

Procedural Wood Textures ([Pea85] and [Nor09])

- Multiple level of details, filtering issues

Voxel Simulation ([Buc98])

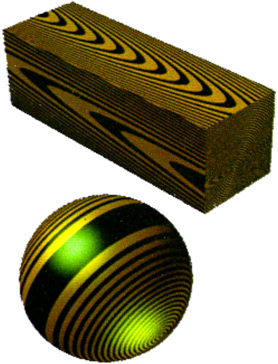
- Memory issues, biased along axis direction

L-systems and 3GMap L-systems ([PL96] and [TGM^{+}09])

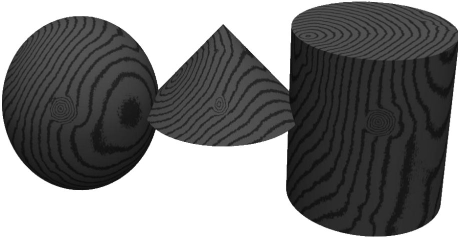
- Formal grammar with parallel application of rules
- Biologicaly faithful, but hard to use

Previous Work

Illustrations



(a) Peachey [Pea85]



(b) Buchanan [Buc98]

Previous Work

Illustrations

(c) Terraz et al. [TGM+09]

(d) Norell [Nor09]

Our Approach

Global Framework

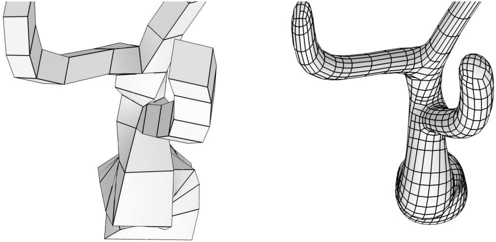


Figure 2: Modeling with blocks (Leblanc, 2011 [LHP11])

Our Approach
Global Framework

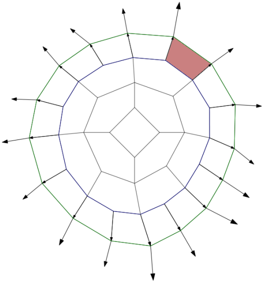
Description

- Refinable polygonal mesh
- Skeleton generation with L-systems
- Surfacic and volumetric parametrization
- Generate cross-section textures
- Interpolation between textures

Cross-section Texturing

Approach Outline

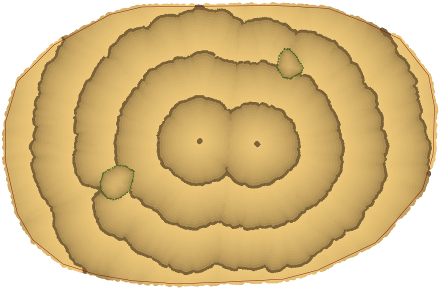
- A first procedural generation ([Nor09])
- A second, particle-based, approach
- Cell (active, dead), groups (generation)
- Parameters : speed, angle, age, color, etc.
- Output : a graph (skeleton) $G = (V, E)$



Cross-section Texturing

Representations

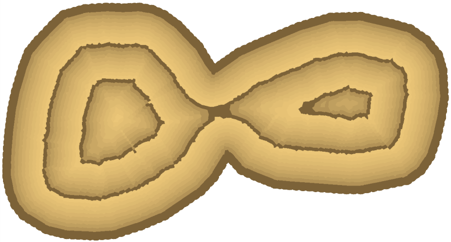
- Random perturbations
- Predefined shape
- Knot simulation
- Inward / Outward
- Multiple piths



Cross-section Texturing

Heuristics

- Intermediary cell generation
- Speed readjustment
- Group splitting
- Cell merging
- Self-intersections
- Orientation check
- Group collisions



Rendering process

2D Version

- *kd*-tree with unstructured points in the plan
- Interpolation methods : weighted (blurred), nearest neighbour

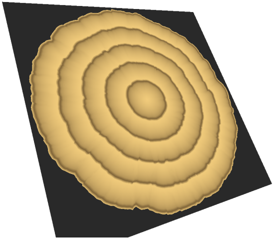
3D Version

- Graph $G = (V, E)$ with polygons
- Triangulation : naive $\mathcal{O}(n^2)$, sophisticated $\mathcal{O}(n)$
- Bilinear color interpolation (direct with OpenGL)

Rendering process

(a) Nearest neighbour

(b) Weighted sum



(c) OpenGL

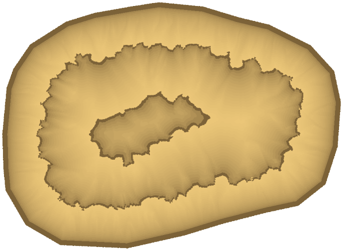
Figure 3: Result comparison of 3 different rendering processes.

Remaining Issues

- Jagged patterns with large inward-growing groups
- Contour-limited growth behave as made inside a mould

Possible Corrections ?

- Spring-mass system
- Element remapping
- Biased regrowth



Biased Regrowth

Algorithm Outline

Algorithm 1 Contour-driven Growth Simulation

1: Contour + pith $\Rightarrow$ initial growth $G = (V, E)$
2: Distance δ from pith to points on the border (eg. : Dijsktra)
3: Map M : angular parameter $\theta \rightarrow$ resulting distance δ
4: Re-run the simulation with speed biased according to M
5: Repeat step 2 to 4 until a visually satisfying result is obtained

3D Interpolation Problem

Introduction

- Cross-section textures at regular intervals
- Pixel-based interpolation of raster textures (blurry)
- Morphing-based modern techniques : automatized $\oplus$ efficient

Possible Heuristic

- Match vertices of G_1 with the second cross-section G_2
- # of generation, # of vertices (dummy cells)

3D Interpolation Problem

Greedy Methods ?

- Generations $S_1, S_2 \rightarrow$ Matching that minimize $\sum c(\alpha_{i,j})$?
- Greedy approximation : fix abitrary match of p_i and q_j with minimum $c(i,j)$, then local algorithm in $\mathcal{O}(n)$
- Quadratic version : try any two starting points p_i and q_j
- Global optimization : assignment problem in $\mathcal{O}(n^3)$

3D Interpolation Problem

Assignment Problem

- Find bijection $f : A \to B$ which minimizes $\sum c(a, f(a))$
- *Hungarian method*, or *Kuhn–Munkres* algorithm, in $\mathcal{O}(n^3)$
- Idea : find maximum potential $y : A \cup B \to \mathbb{R}$ such as $y(a) + y(b) \leqslant c(a, b)$ for all $(a, b) \in A \times B$
- When done, *tight edges* induce a perfect matching

Applications and Limitations

- Matching between two cross-sections X and Y
- Branch creation and trunk splitting : match X with Y and Z ?
- Reverse problem : given a point in the 3D space, find its color
- More complex if the pith follow a curve, and not a straight line
- Can use 1D or 2D textures to add a level of details

Conclusion

- 2D generative method, fast and customizable
- Knots, multiple sources, contour-limited growth
- Possible improvements (biased growth, etc.)
- 3D interpolation models were proposed

Thank you for your attention.
Feel free to ask your questions.

John W. Buchanan.
Simulating wood using a voxel approach.
Computer Graphics Forum, 17(3) :105–112, 1998.

Luc Leblanc, Jocelyn Houle, and Pierre Poulin.
Modeling with blocks.
The Visual Computer (Proc. Computer Graphics International 2011), 27(6-8) :555–563, June 2011.

Kristin Norell.
Creating synthetic log end face images.
In *Image and Signal Processing and Analysis, 2009. ISPA 2009. Proceedings of 6th International Symposium on*, pages 353–358, Sept. 2009.

Darwyn R. Peachey.
Solid texturing of complex surfaces.
SIGGRAPH Computer Graphics, 19(3) :279–286, July 1985.

Przemyslaw Prusinkiewicz and Aristid Lindenmayer.
The algorithmic beauty of plants.