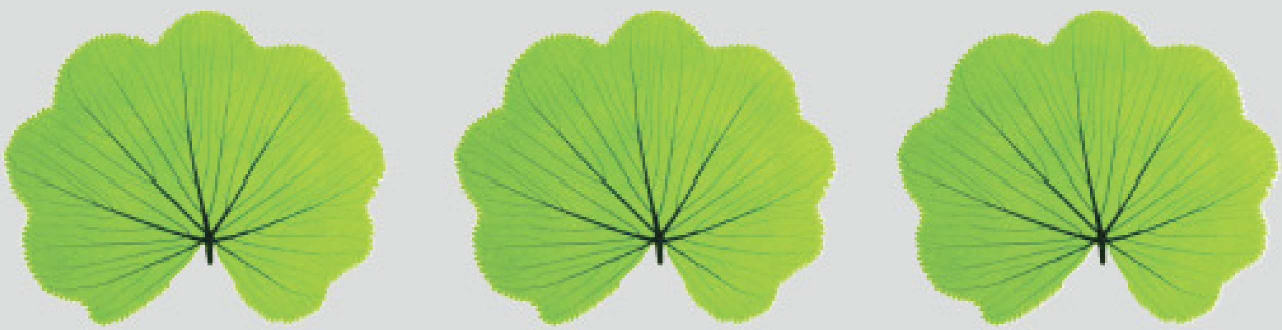
Fenômenos Naturais

Soraia Raupp Musse

Padrões da natureza
Plantas - Paper Dr. P

Modeling and Visualization of Leaf Venation Patterns

Adam Runions, Martin Fuhrer, Brendan Lane, Pavol Federl, Anne-Gaëlle Rolland-Lagan, Przemyslaw Pruzinkiewicz



Sumário

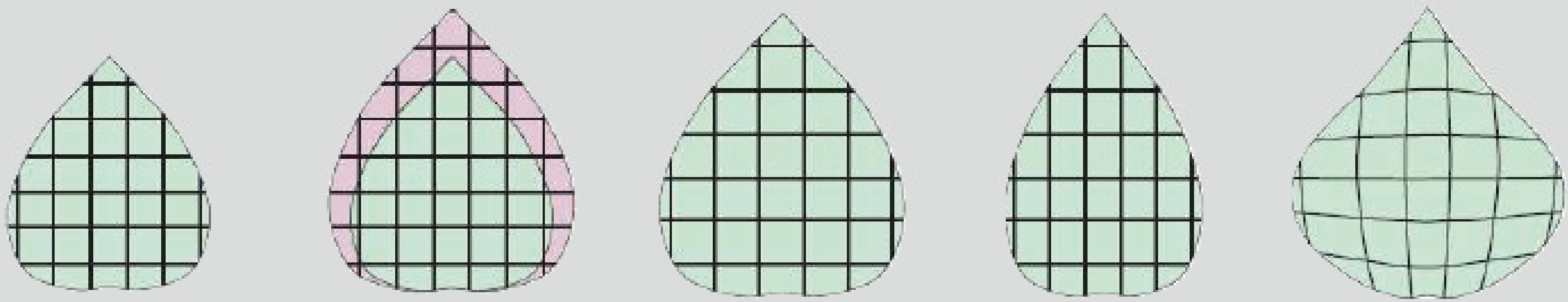
- Motivação
- Folhas
- Modelo
- Modelo Alterado
- Implementação
- Resultados
- Conclusões

Motivação

- Modelagem baseada em simulação de padrões encontrados em organismos vivos
- Padrões de nervuras em folhas: admiráveis, porém relativamente pouco compreendidos
- Síntese dos padrões de nervuras em folhas
- Digitalização nem sempre é melhor saída

Folhas

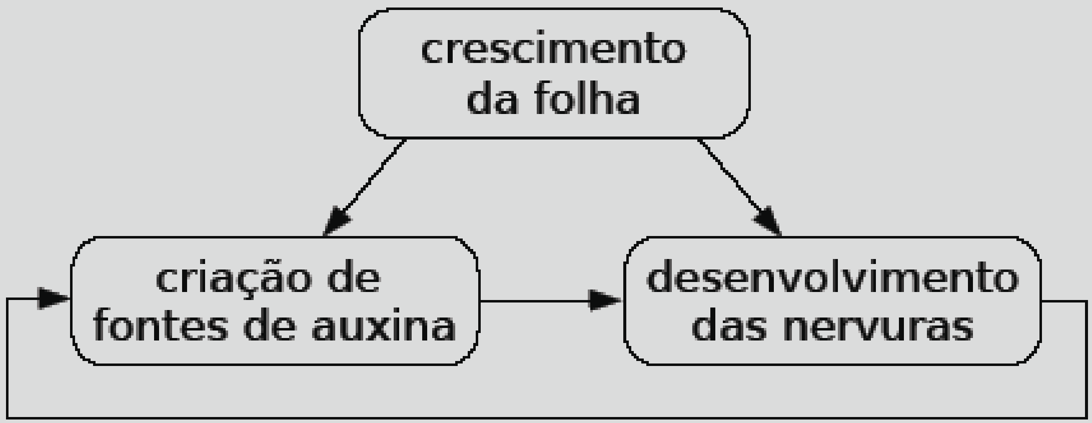
- Várias formas de crescimento da folha



- Hipótese da Canalização
  - Fontes de auxina (um hormônio), surgem em certos pontos da folha
  - Auxina flui pela folha em direção a nervuras existentes, originando novas nervuras

Modelo

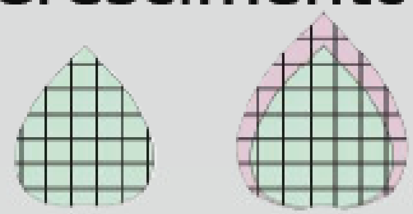
- Assumem que a formação dos padrões das nervuras decorrem da interação entre três processos



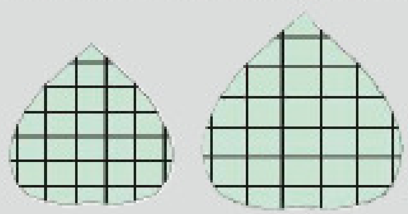
- Padrão é representado por um grafo
  - Nodos de nervura
  - Fontes de auxina

Modelo

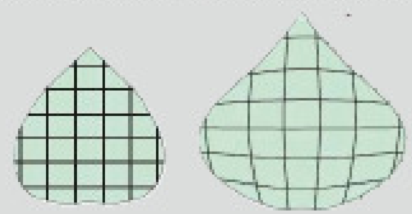
- Crescimento da folha
  - Crescimento marginal



  - Crescimento uniforme



  - Crescimento anisotrópico não-uniforme



Modelo

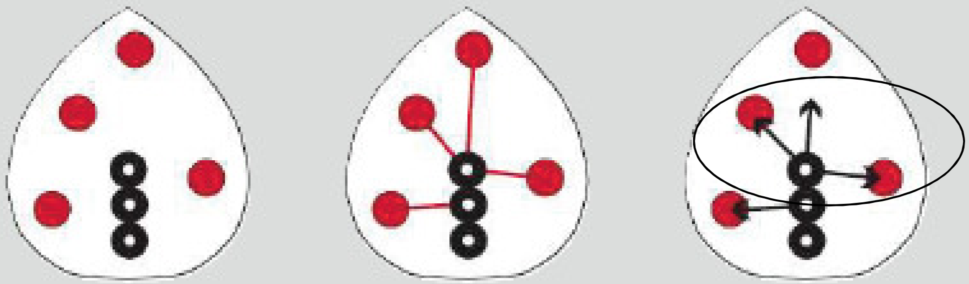
- Criação de fontes de auxina
  - Devem ser criadas a uma distância maior que b_s de todas as fontes anteriores
  - Devem ser criadas a uma distância maior que b_v de todas as nervuras existentes
  - A cada iteração são gerados pontos aleatoriamente na superfície da folha; os que atenderem estas condições são usados
  - Parâmetro ρ: número de pontos por iteração por unidade de área

Modelo

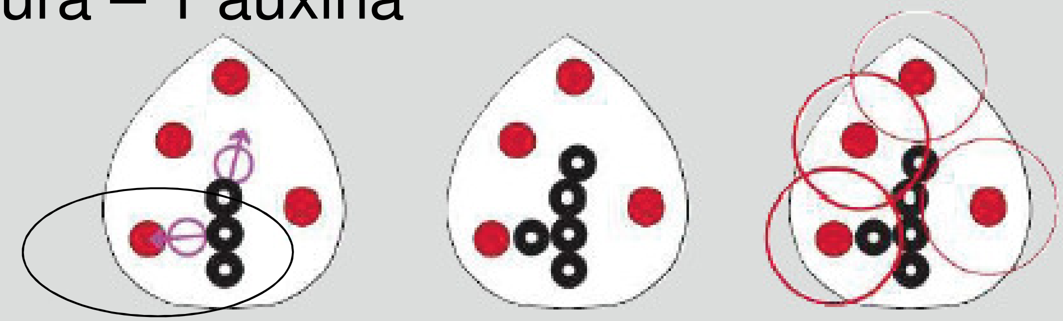
- Desenvolvimento de nervuras
  - Cada fonte de auxina influencia o nodo de nervura mais próximo de si
  - Cada nodo de nervura influenciado por ao menos uma fonte de auxina origina um novo nodo de nervura
  - O novo nodo é colocado a uma distância *D* do anterior, numa direção definida pela média dos vetores normalizados do nodo original a cada fonte que influencia

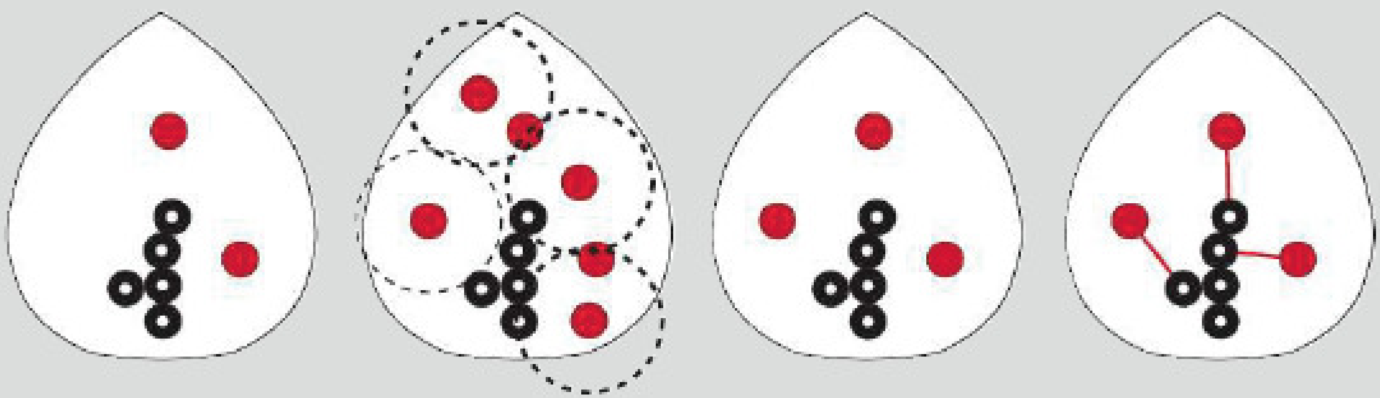
Modelo

1 nervura – 3 auxinas



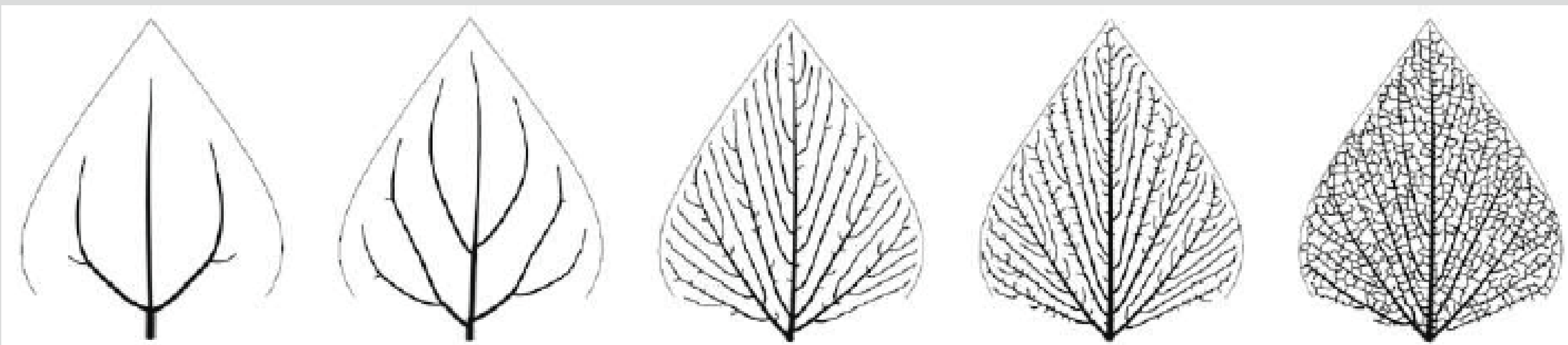
1 nervura – 1 auxina





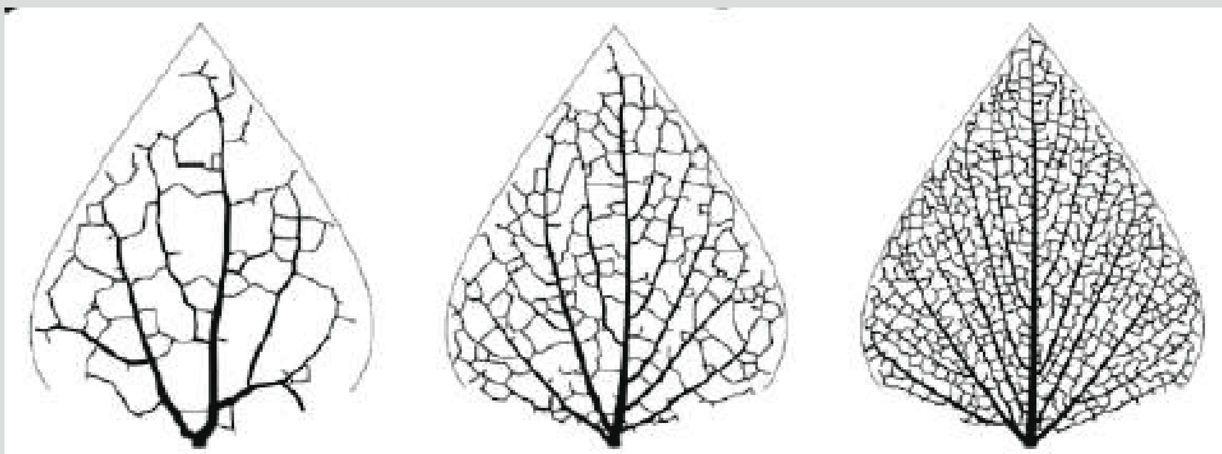
Resultados

- Exploração dos parâmetros: k_d (*kill distance*)



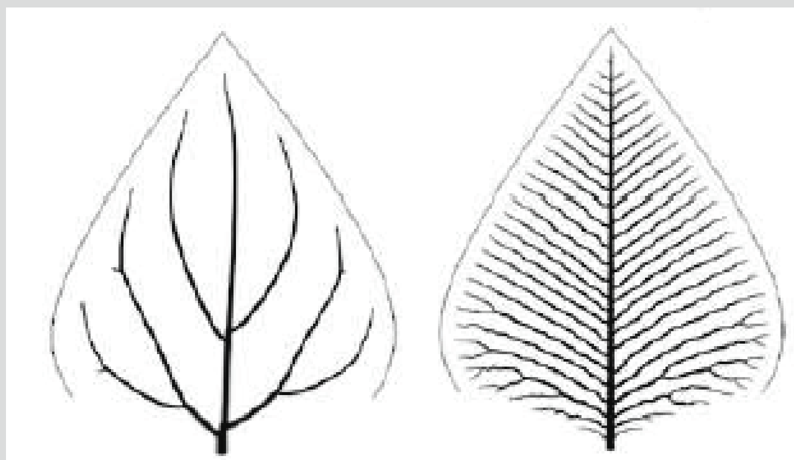
Resultados

- Exploração dos parâmetros: ρ



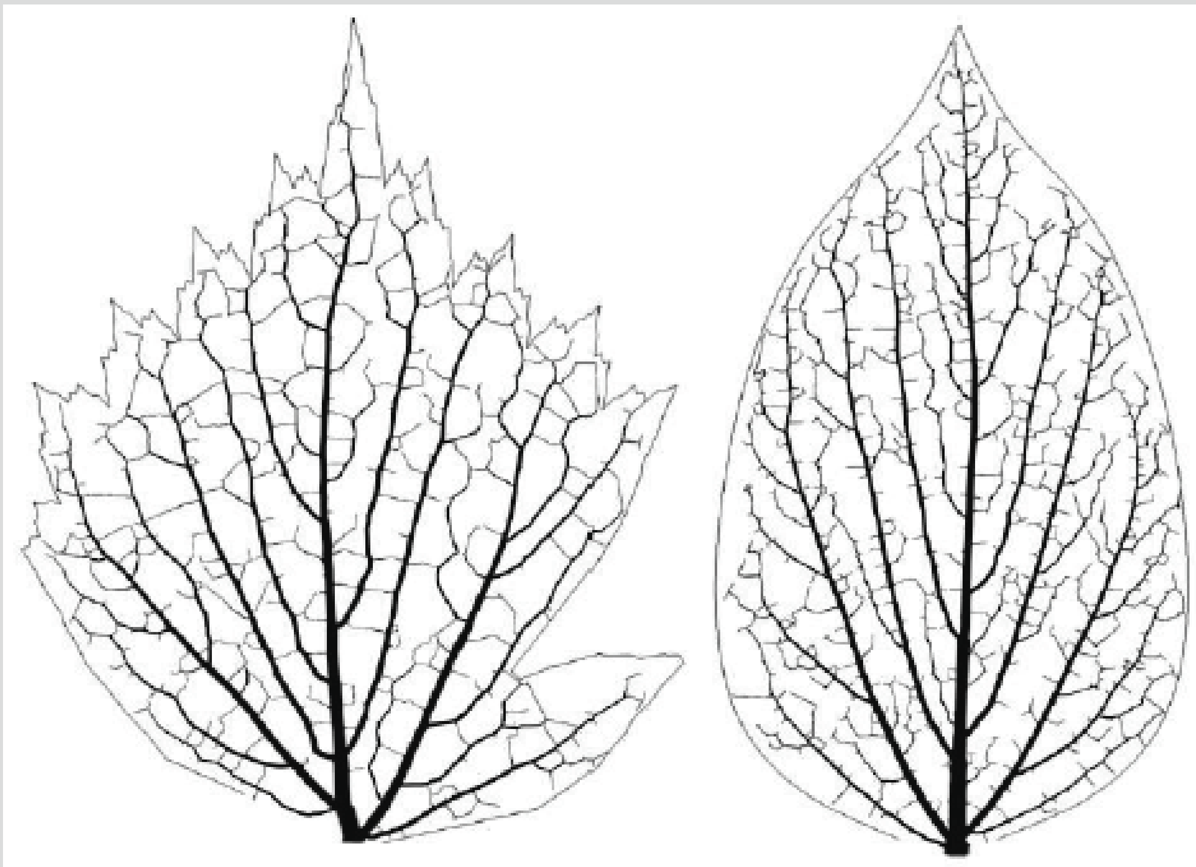
Resultados

- Exploração dos parâmetros: velocidade de cresicmento da folha



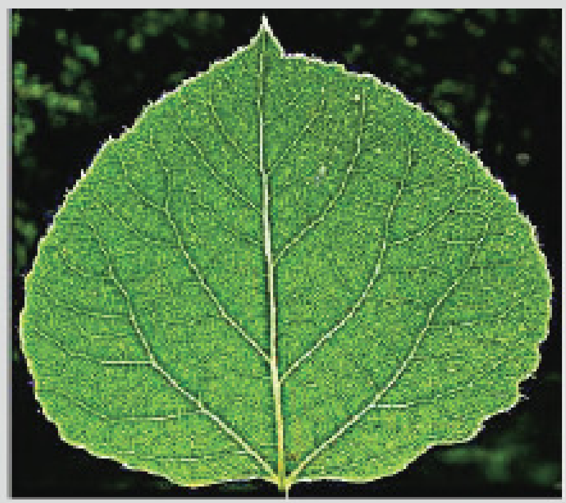
Resultados

- Alguns padrões...



Resultados

- Vizinhanças padrão (2h), fotografia, aproximação de Urquhart (2 min)

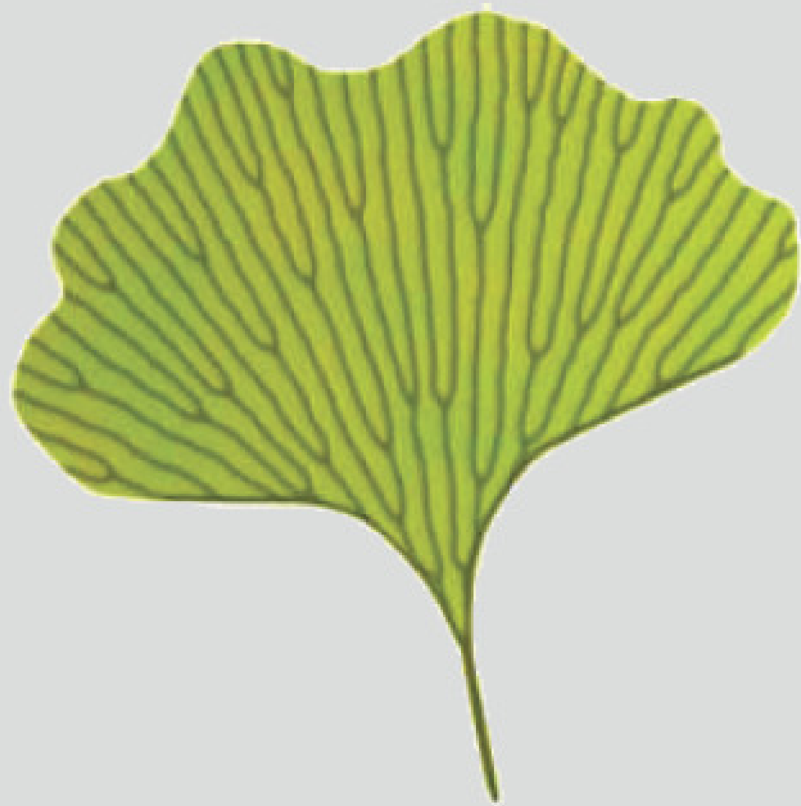




Resultados

- Fotografia x sintetizado

Resultados

- Fotografia x sintetizado

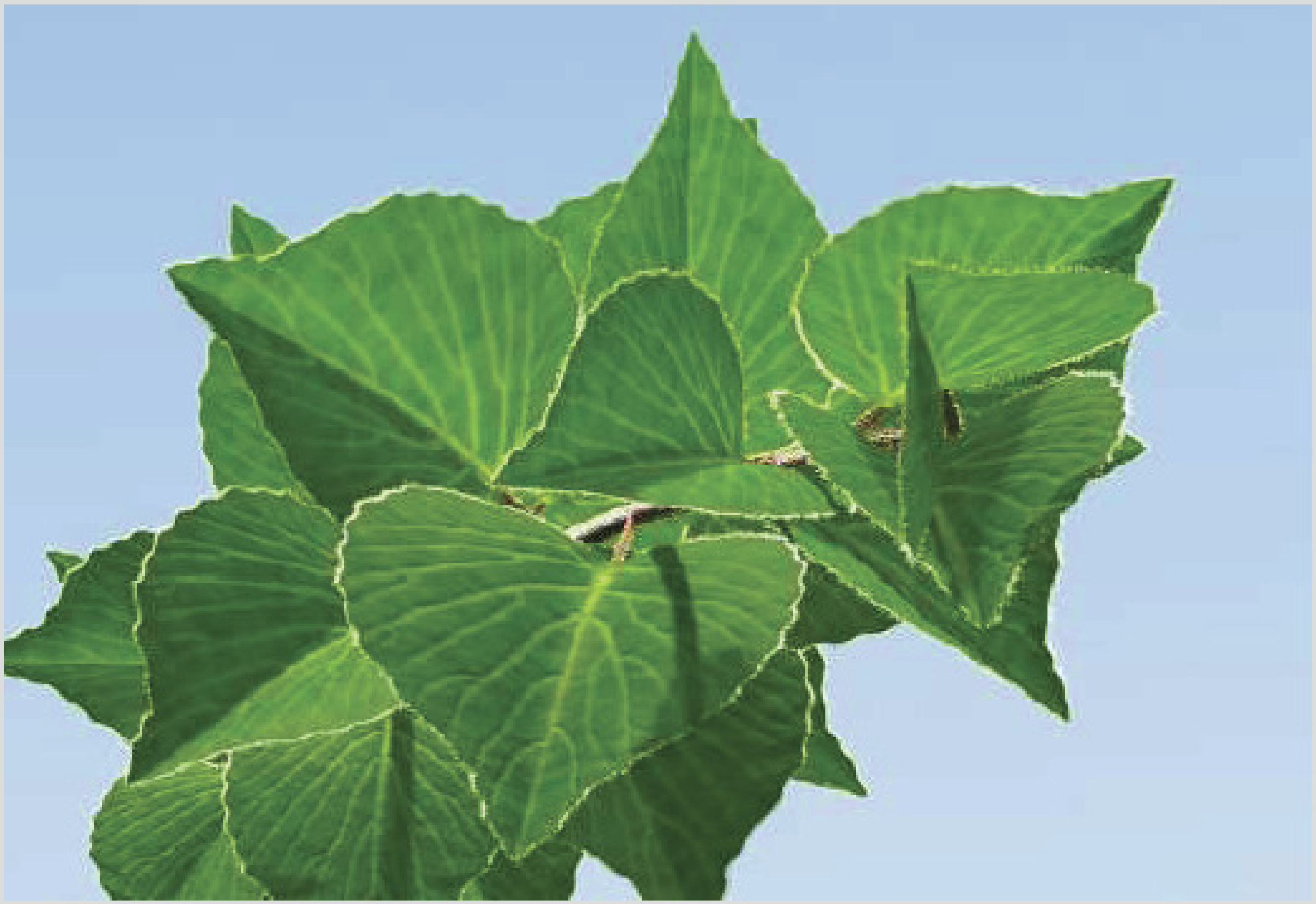


Resultados

- Renderizado

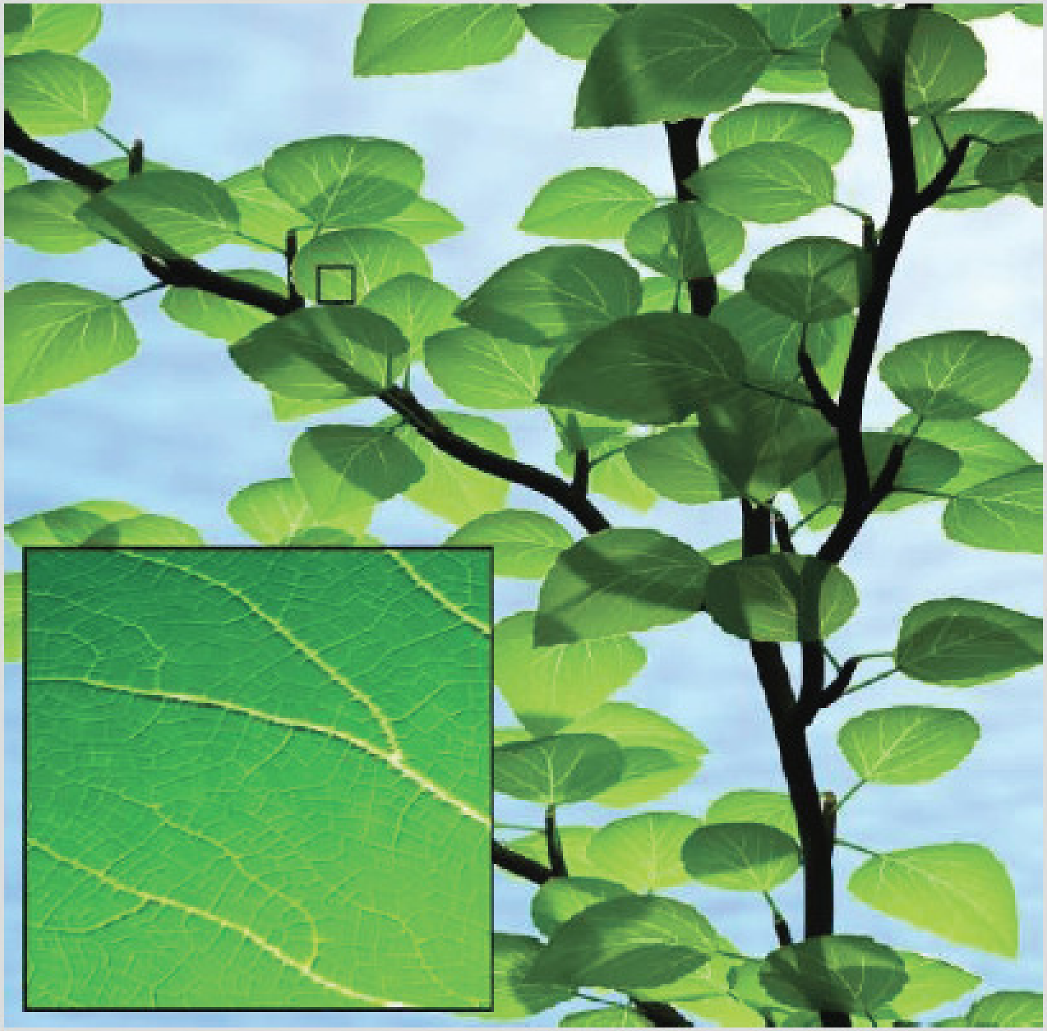
Resultados

- Renderizado



Resultados

- Renderizado



Resultados

- Renderizado

Conclusões

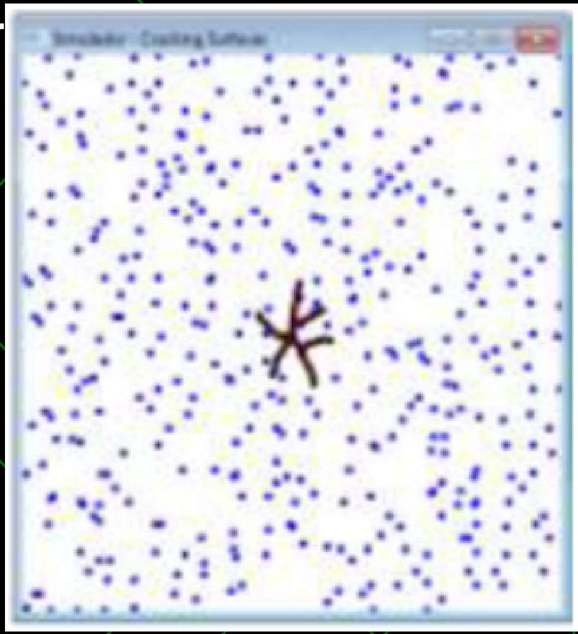
- Modelo para geração de padrões de nervuras de folhas
- Aplicação em síntese de texturas
- Primeiro modelo visualmente convincente baseado em conceitos biologicamente plausíveis

Generation of Cartoon 2D Cracks Based on Leaf Venation Patterns

Julia M. Colleoni Couto, Tiago Marcelino, Henry Braun, Diogo Strube de Lima, Soraia Raupp Musse

Introdução

- Criação de cracks usando Dr.Pi
- Cracks para games
- Customizar o local do tiro para iniciar os cracks



Parâmetros

- kill distance: responsible for defining the threshold distance from crack node to markers that should be removed (default value is equal to 0.01).

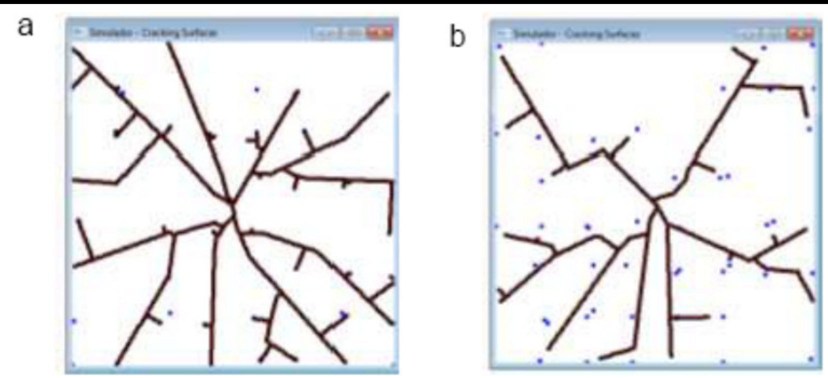


Figure 4: *Two images illustrating different values for kill_distance parameter. In a) kill_distance is equal to 0.01, while in b) it is 0.1. It is possible to see that image a) generates more crack nodes than in b).*

Parâmetros

- D: distance from a crack node to create another node (default value is equal to 0.1). Depending on the used value, crack segment generated between two crack nodes are smaller, for small D, or larger for large D

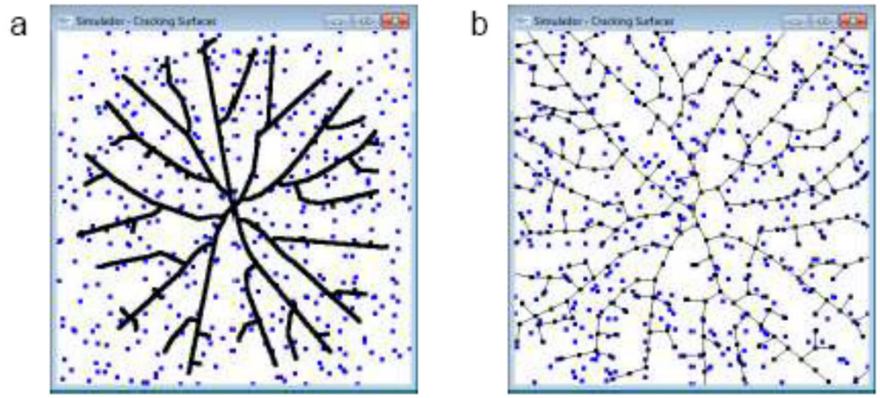


Figure 5: *Two images illustrating different values for D distance parameter. In a) D is equal to* 0.1*, while in b) it is* 0.04*. It is possible to see that image a) generates less crack nodes than in b).*

Parâmetros

- Num_interaction
- Line_grid/col_grid
- Crack_radius
- Line_growth

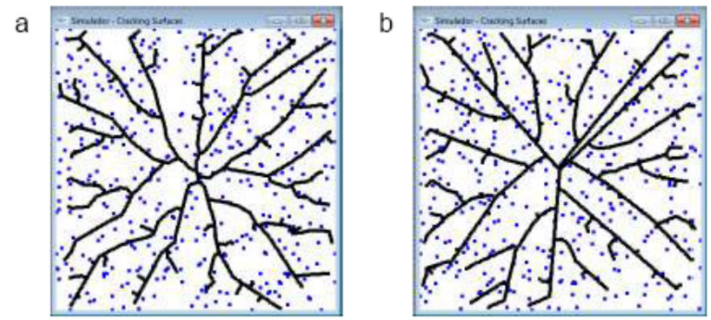


Figure 6: *Two images illustrating different values for crack_radius parameter. In a) crack_radius is equal to 0.5, while in b) it is 20.0. It is possible to see that image b) generates straighter crack segments than in a).*

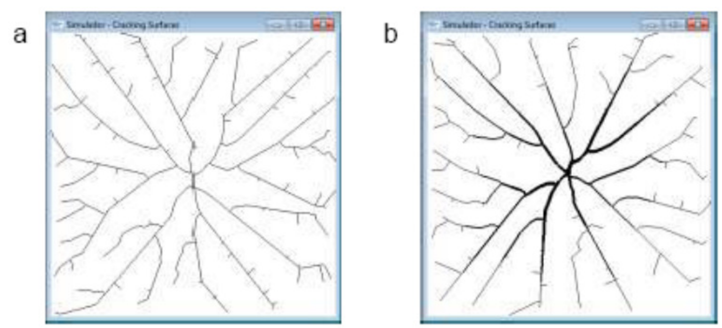


Figure 7: *Two images illustrating different values for line_growth parameter. In a) line_growth is equal to 0.5, while in b) it is 0.0. It is possible to see that image b) generates varied lines width as a function of cracks hierarchy in generation process.*

Parâmetros

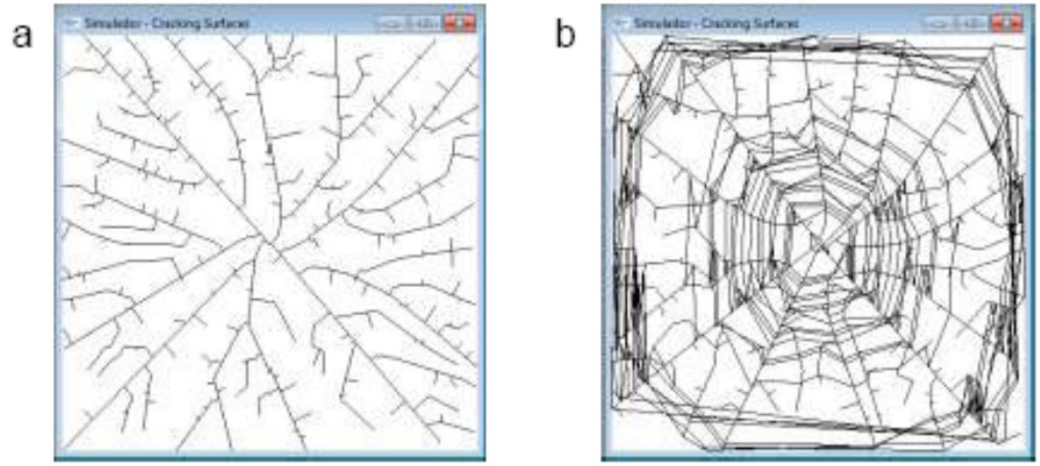
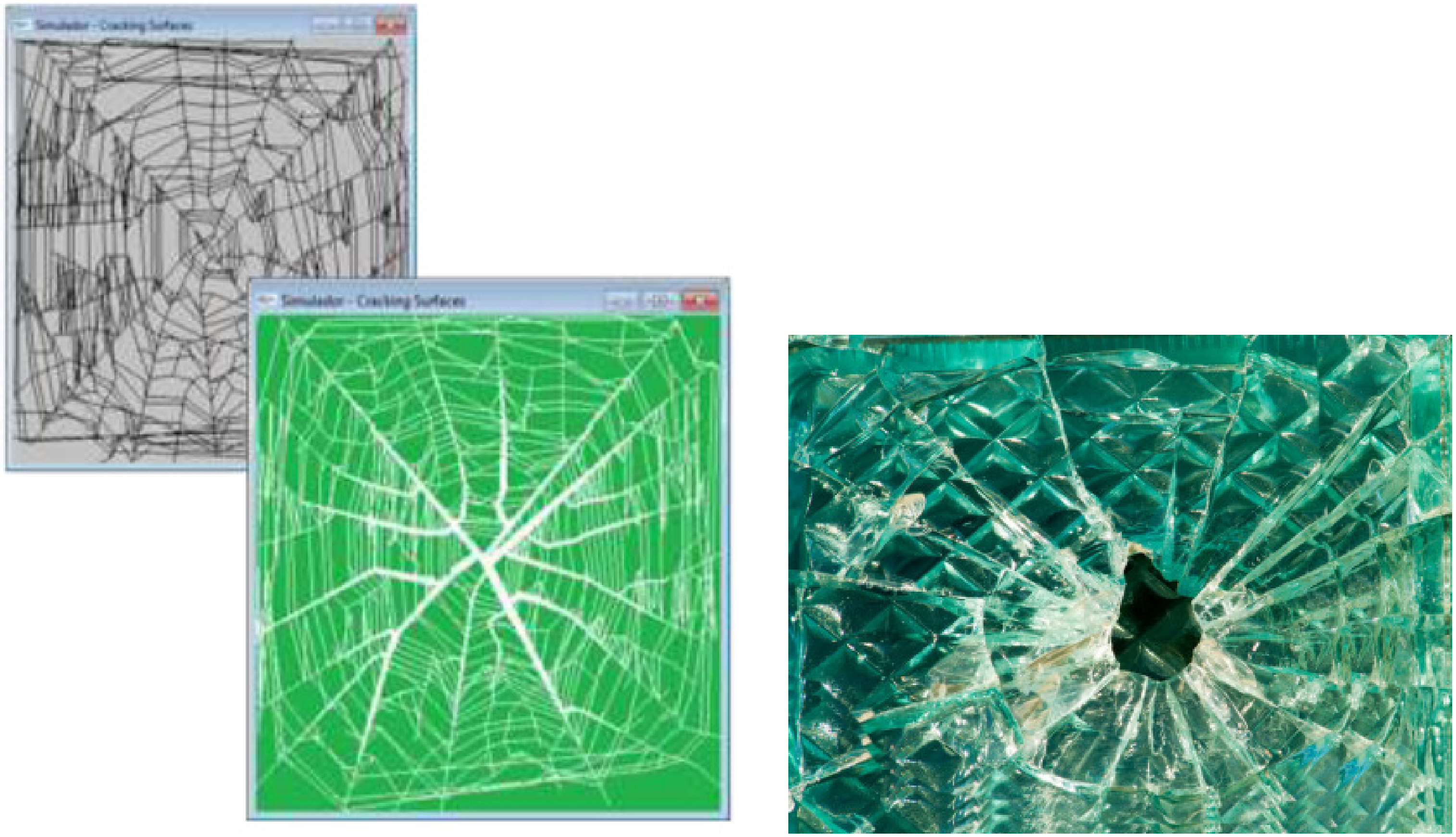


Figure 8: *Two images illustrating the effect of draw_connections parameter. In a) draw_connections is FALSE, while in b) it is TRUE. Other parameters used in these simulations are:* $kill\_distance = 0.02$, $D = 0.2$, $num\_iterations = 80$ *and* $num\_lines\_grid = num\_column\_grid = 20$.

Parâmetros



(a) (b)

Modeling and Rendering of Individual Feathers

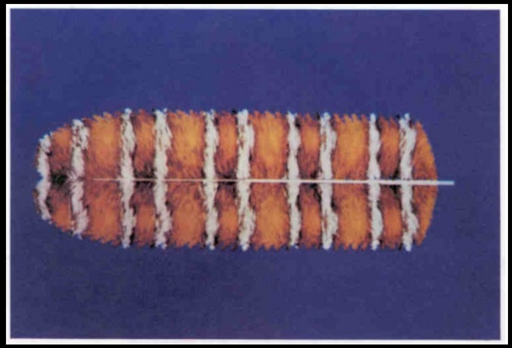
Cristiano Franco
Marcelo Walter

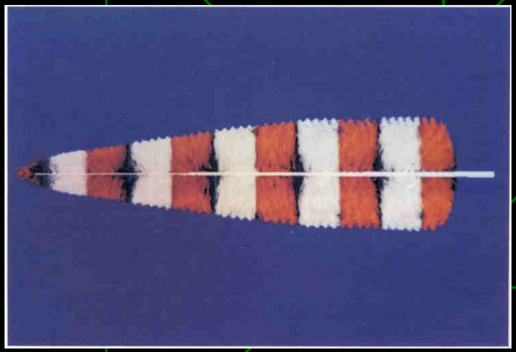
Introduction

- Modeling of Natural phenomena
- Development of this research area follows the complexity of nature
  - Mineral kingdom
  - Vegetable kingdom
  - Animal kingdom
- Challenges: humans and animals
- Modeling of birds and feathers

Previous Work

- Few works in this area
- The first one: Dai *et al*. in 1995
  - They model the structure of the feathers as line segments branching from a main structure
  - Overall structure is controlled by functions acting on the parameters
  - Main objective was generate textures based on dynamical systems



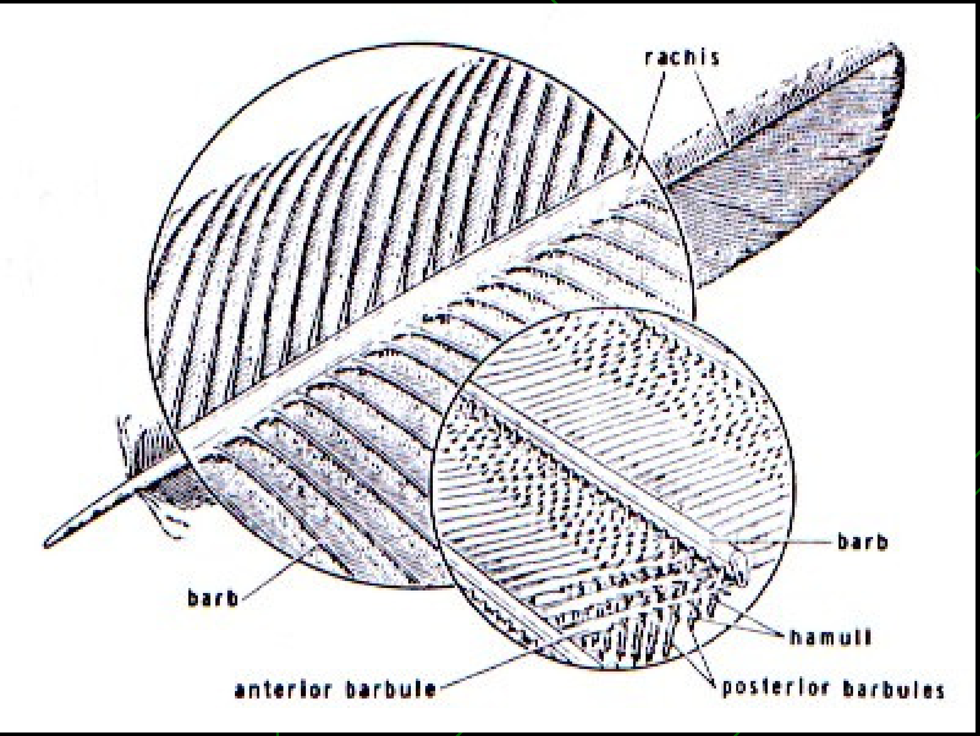


Previous Work

- Recently, two works were developed
- Chen *et al.*
  - Feathers are modeled with L-Systems
  - Approach to cover a bird with the feathers generated by the user
- Streit and Heidrich
  - Feathers are modeled with Bézier curves, as in our case
  - User specify key barbs from which the other barbs are derived by interpolation
  - Definition of key barbs can be a time consuming and boring task to the user

Feather Basics

- Model based on real feathers
- Feathers are a type of branching structure
- Main parts
  - Rachis
  - Barbs
  - Barbules
  - Vanes



The Proposed Model

- The model has a biological motivation
- Besides biological-based parameters, we have others created in order to facilitate the process of building feathers

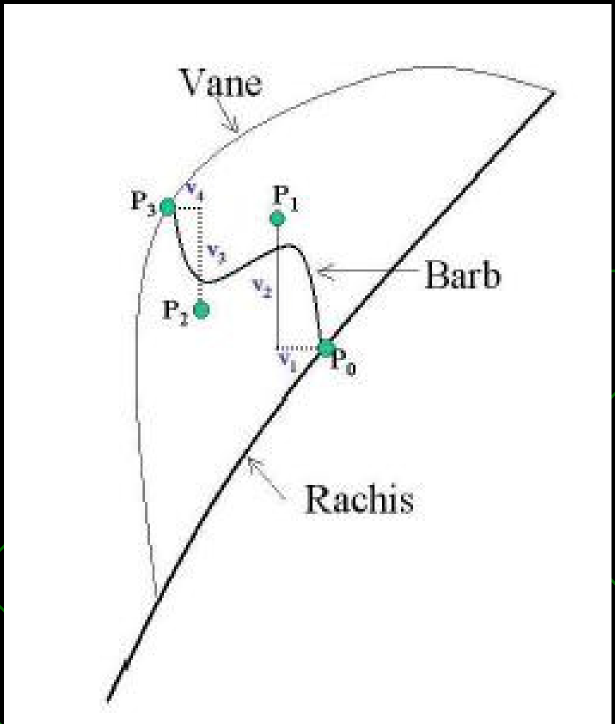
Process of modeling:

1. The user defines a cubic Bézier, which represents the rachis
2. Then, the user defines 2 Bézier curves with 5 control points, that defines the boundaries of the structure
3. After that, the user manipulates the parameters in order to build in realtime a feather like he wishes

Parameter	Range Values	Description
N_b	[0,5000]	Number of Barbs
F_{pv}	[0,1]	Variation on length of barbs
F_b	[0,1]	Variation on form of barbs
S_v	T v F	Symmetry of Vanes
U_f	T v F	Uniformity of barbs (form)
U_c	T v F	Uniformity of barbs (length)
I_s	[0,1]	Start of second segment

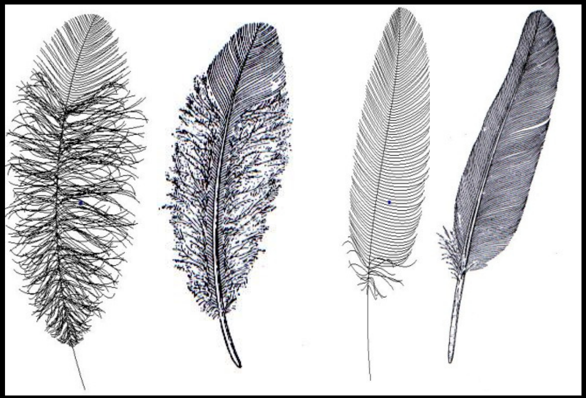
The Proposed Model

- For each barb, four control points
- Initial condition, P_0 and P_1 in the rachis and P_2 and P_3 in the limit of the vane
- The procedure will move points P_1 and P_2 away from their initial position, according to parameters



Rendering

- The rendering phase can be done in three ways:
  - We can render the feather in a non-realistic way (a)
  - We can render the feather with texture, taken from a real feather or generated artificially, using OpenGL
  - We can build a .pov file and render it using the ray-tracer POV-Ray (b) (not in the paper)



(a) Non-realistic rendering

(b) POV-Ray

Results - Hawk Feather

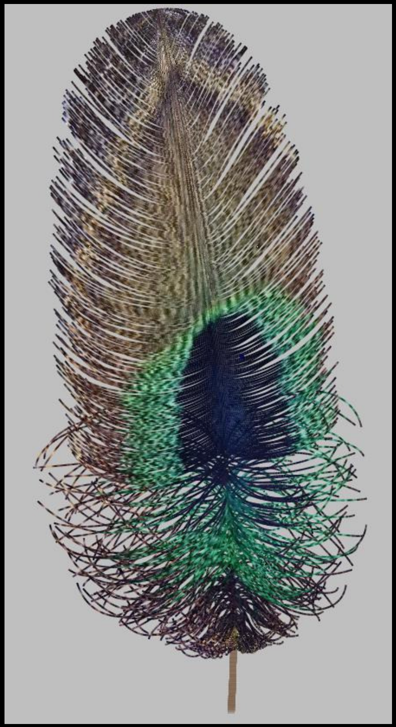
Segment	Nb	Fpv	Fb	Sv	Uf	Uc	Is
Superior	250	0.03	0.3	T	T	F	0.2
Inferior	50	0.2	0.3	F	F	F	

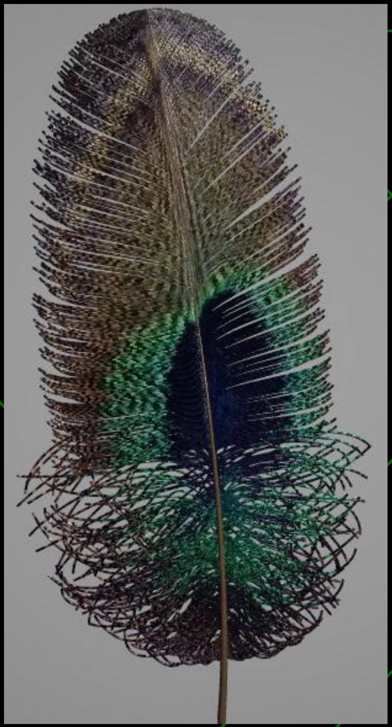
Results - Blue Feather

Segment	*Nb*	*Fpv*	*Fb*	*Sv*	*Uf*	*Uc*	*Is*
Superior	250	0.1	0.4	T	T	F	0.2
Inferior	50	0.01	0.2	F	T	F	

Results – Artistic Peacock

Segment	*Nb*	*Fpv*	*Fb*	*Sv*	*Uf*	*Uc*	*Is*
Superior	400	0.2	0.4	T	T	F	0.4
Inferior	200	0.4	0.6	F	F	F	





Conclusions

- Our system allows a great deal of user control
- The set o parameters allows to create several types of feathers
- Future works include an investigation of an illumination model targeted to feathers

Integration of Complex Shapes and Natural Patterns

Marcelo Walter
Dept. of Computer Science
The University of British Columbia

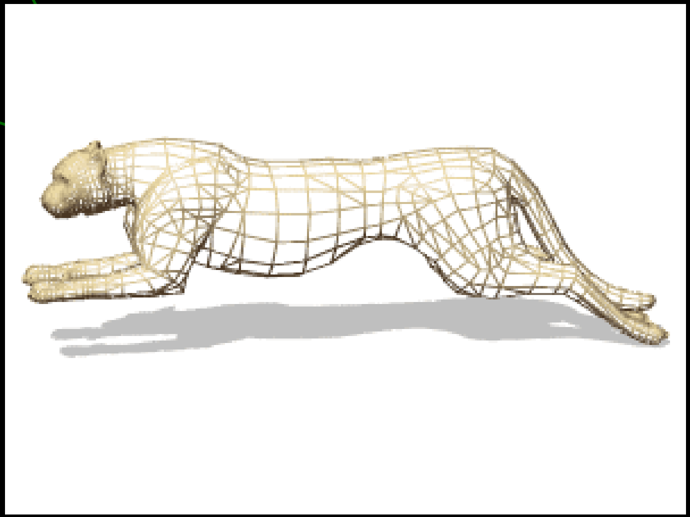
Overall Goal

- Development of an integrated scheme for generating rich visual details for natural objects
- Domain: Patterned Animals

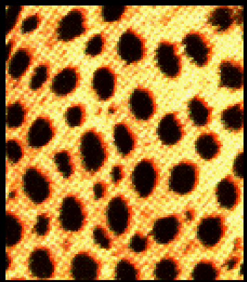
The Problem

- Integration of the visual and shape elements of an object
  - Usual method is to model shape, then add pattern
  - Usually done with texture mapping

Example of Texture Mapping



+



=

Strategy

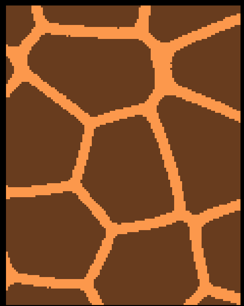
1 - Develop a method to generate patterns found in mammalian coats, especially giraffes and big cats

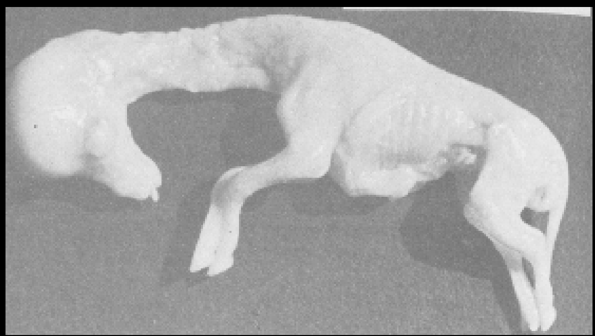
2 - Controlled transformation of a shape

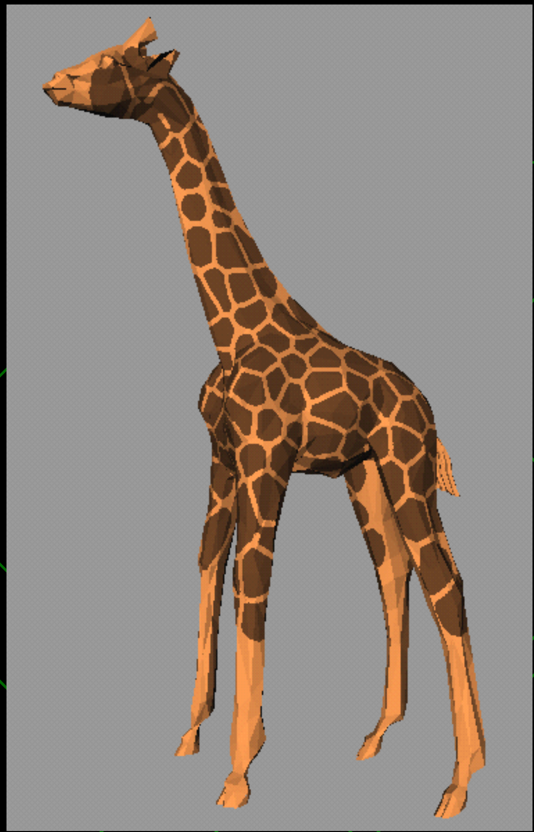
3 - Integration of the two

- **"Inspired" by Nature itself**

Pattern Formation "in place"







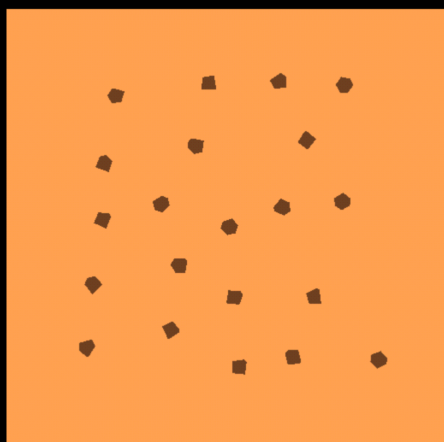
Overview of the Model

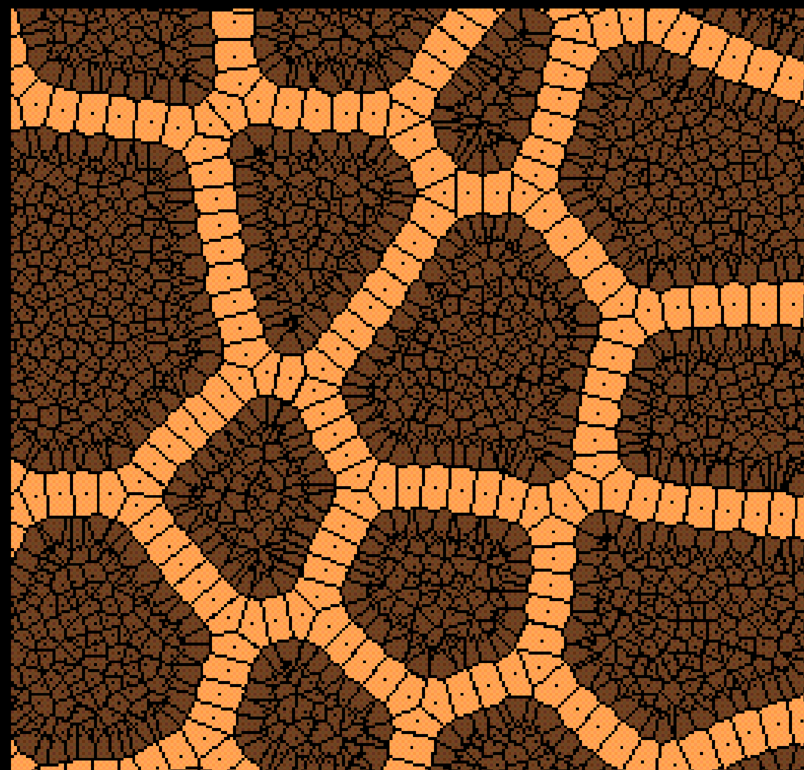
- Cells are modelled as points
- Types of cells characterized by:
  - Division rates, adhesion, colour, initial probability, mutation probability
  - Type: randomly or manually determined
- Cell size: relaxation procedure to keep min distance between points

Parameters and their roles

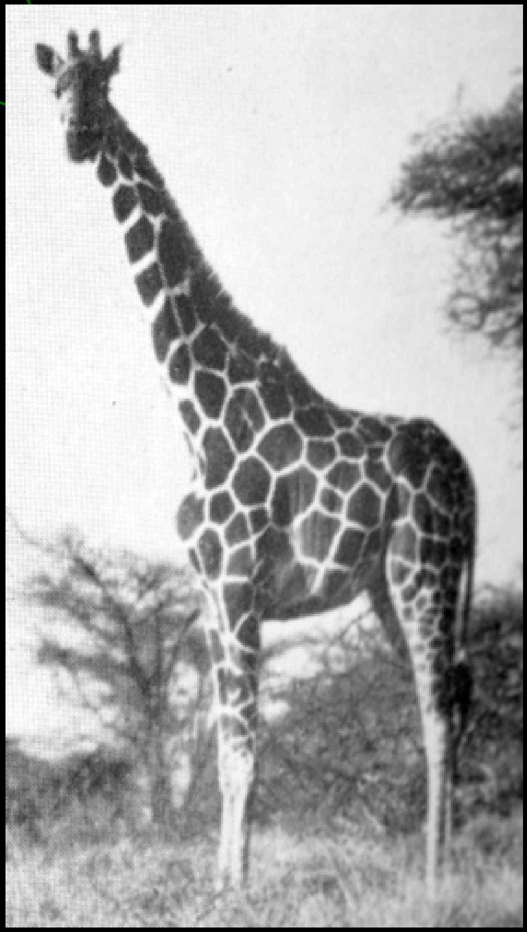
- Division rate: absolute and relative numbers of cells of each type
- Colour: colour
- Adhesion: tendency of cells to stay together
- Anisotropy: tendency of cells to move in a preferred direction
- Probabilities: distribution of types of cells

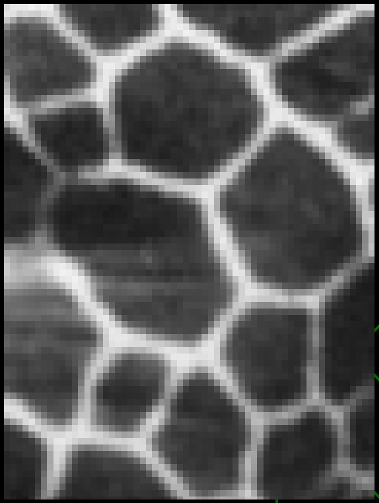
Illustration

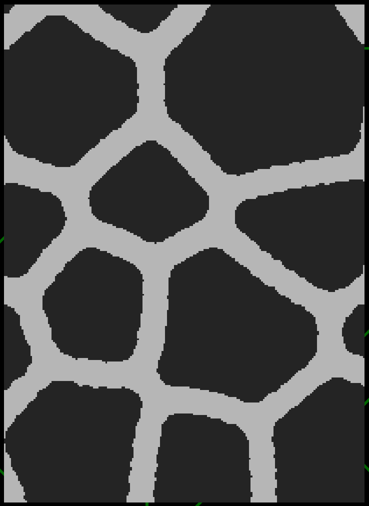




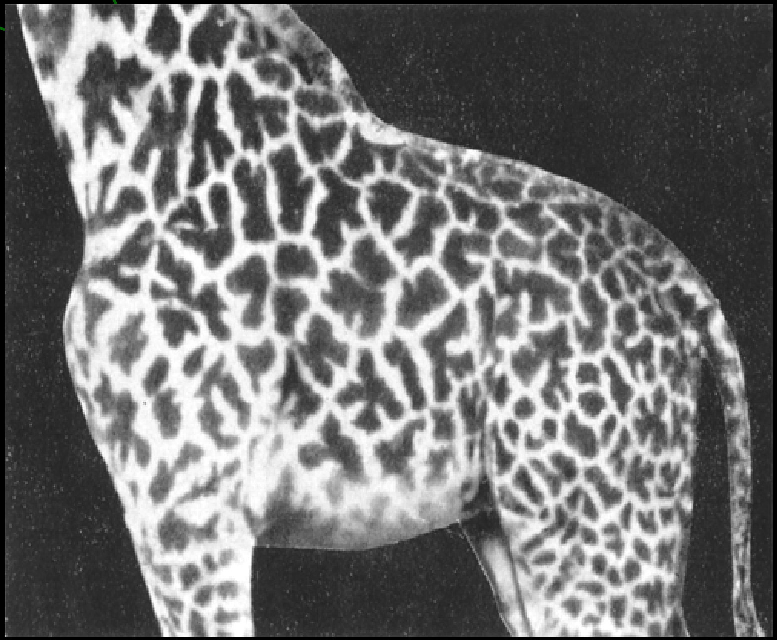
Giraffe Patterns (Reticulata)

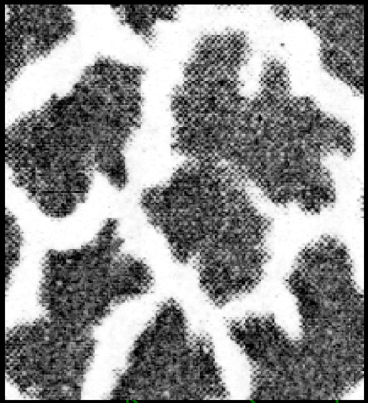


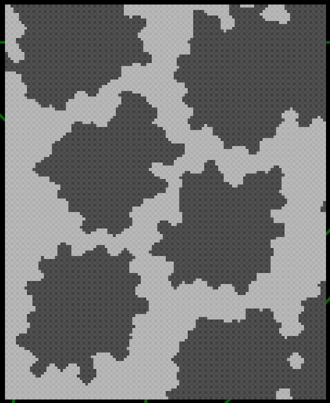




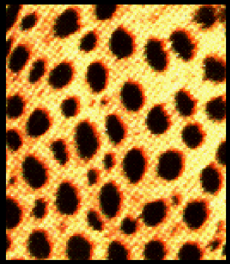
Giraffe Patterns (Tippelskirschi)



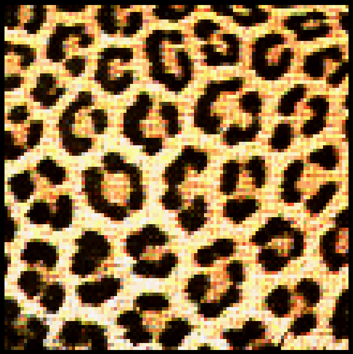


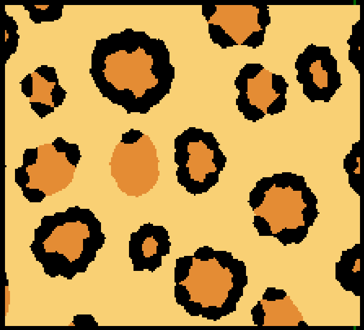


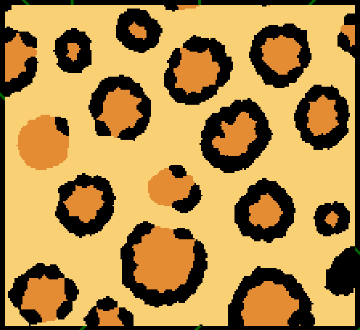
Spotted Patterns



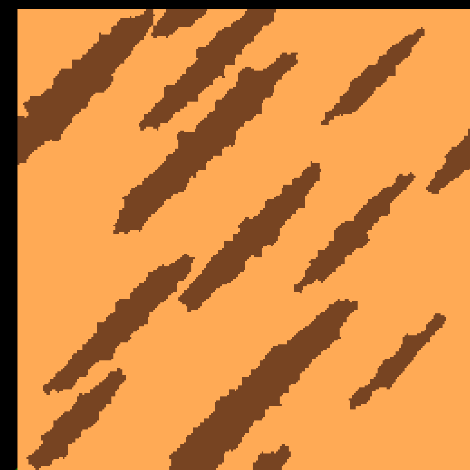
Rosette

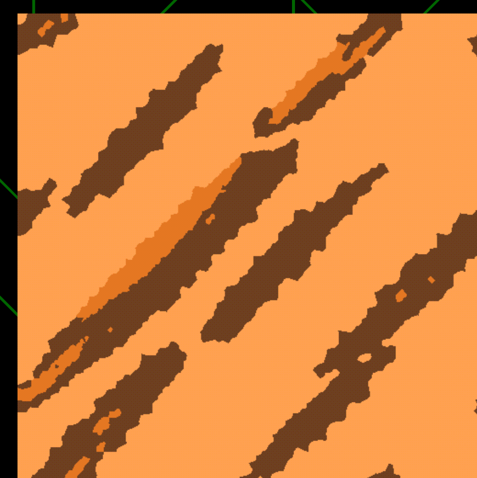




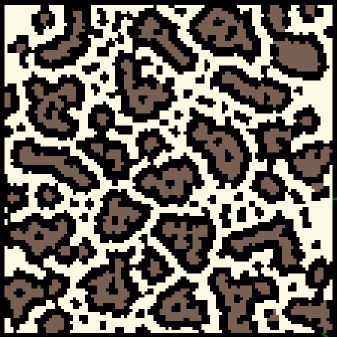


Anisotropic Patterns





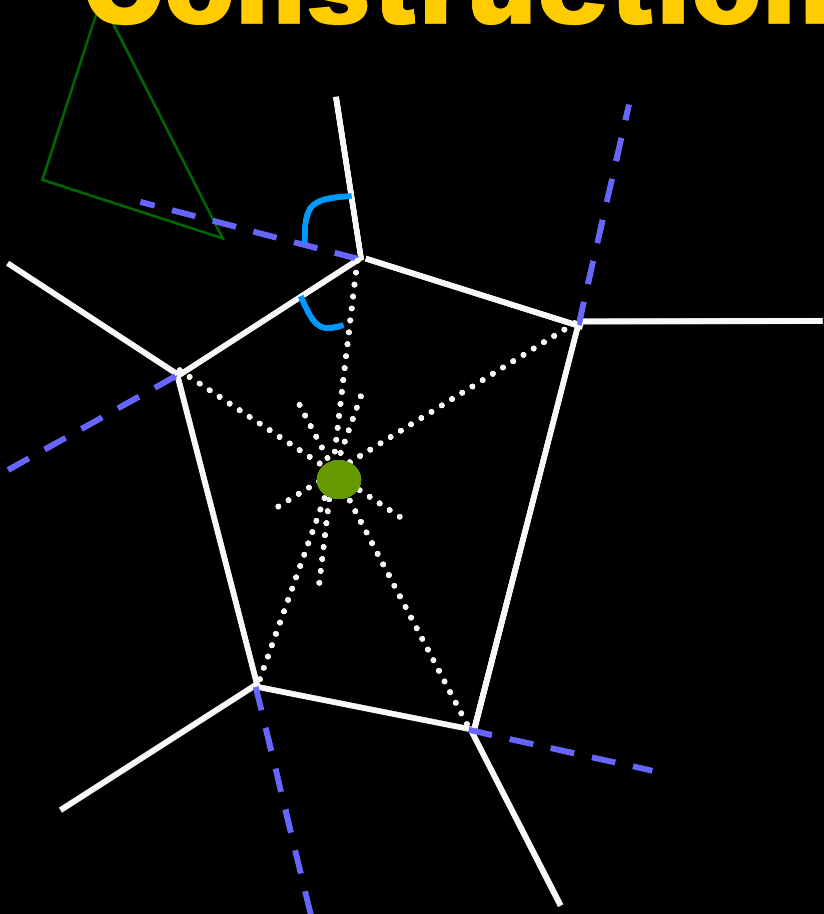
Other Big Cats (Ocelot)



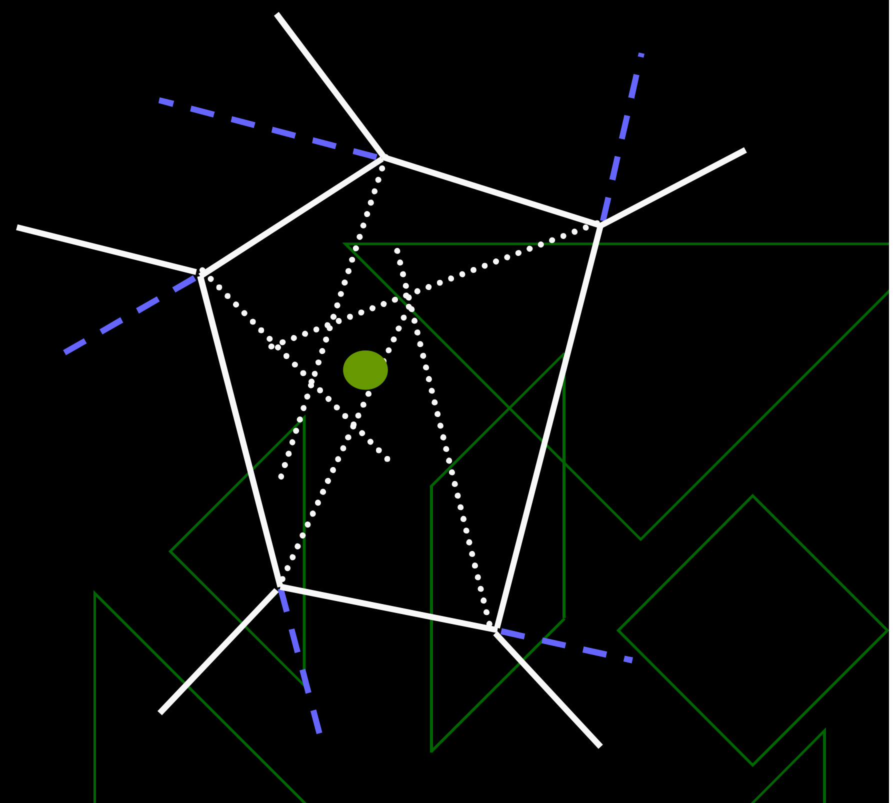
Assessing the results

- Qualitative (looks good)
- Quantitative
  - Many patterns have Voronoi diagrams as basis

Metric: Geometric Construction



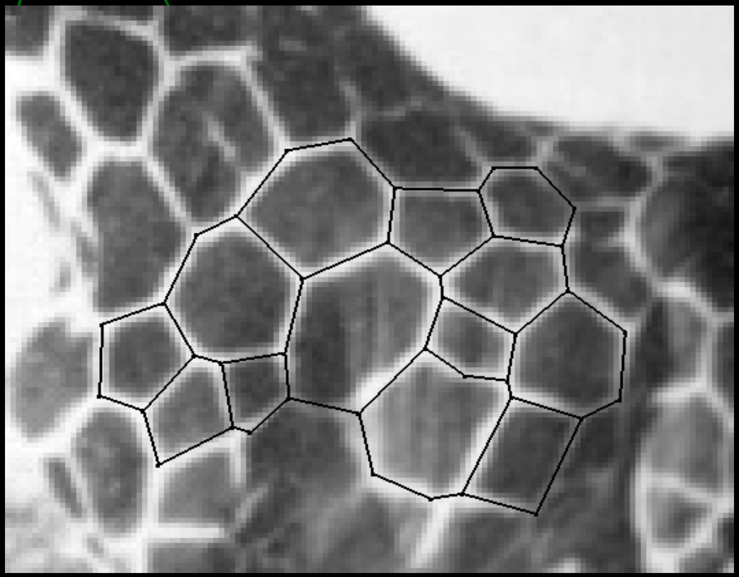
True Voronoi cell



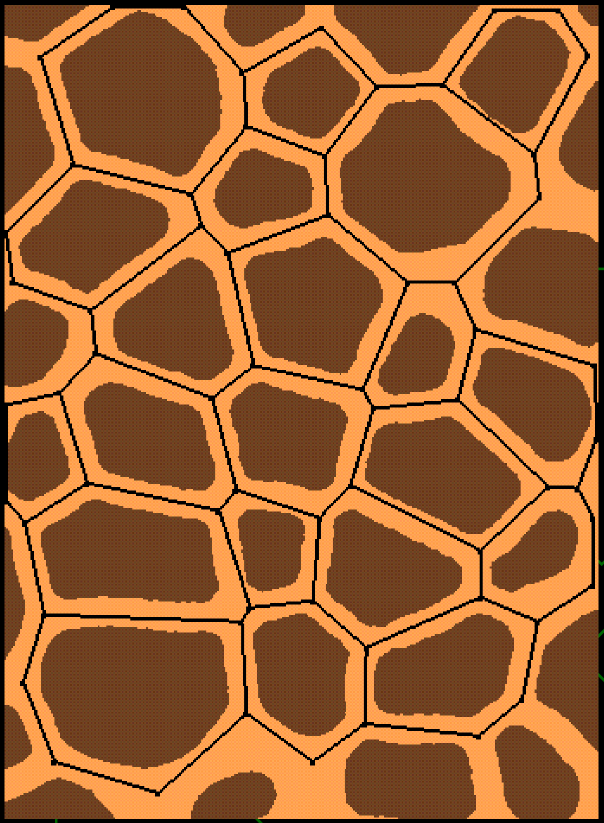
Approximate Voronoi cell

Measure is sum of squared distances

Assessing the patterns



Real Giraffe M =1.43%



Computed M=0.896%

Simulação em Tempo-Real de Nuvens utilizando Billboards

Rafael Piccin Torchelsen
Soraia Raupp Musse

cromoslab
Virtual Humans Simulation

Sumário

- Introdução
- Estado da Arte
- Modelo Proposto
- Resultados
- Demonstração
- Conclusão

cromoslab
Virtual Humans Simulation

Motivação

Realismo visual de nuvens e variações ambientais de céus para aplicações em tempo-real, em especial jogos 3D

cromoslab
Virtual Humans Simulation

Problema

- Complexidade Computacional
- Volume
- Variação nos modelos tridimensionais
- Variação ambiental
- Iluminação

cromoslab
Virtual Humans Simulation

Objetivos

Sistema de simulação em tempo-real de nuvens volumétricas, utilizando um sistema semi-automático de geração de modelos tridimensionais de nuvens

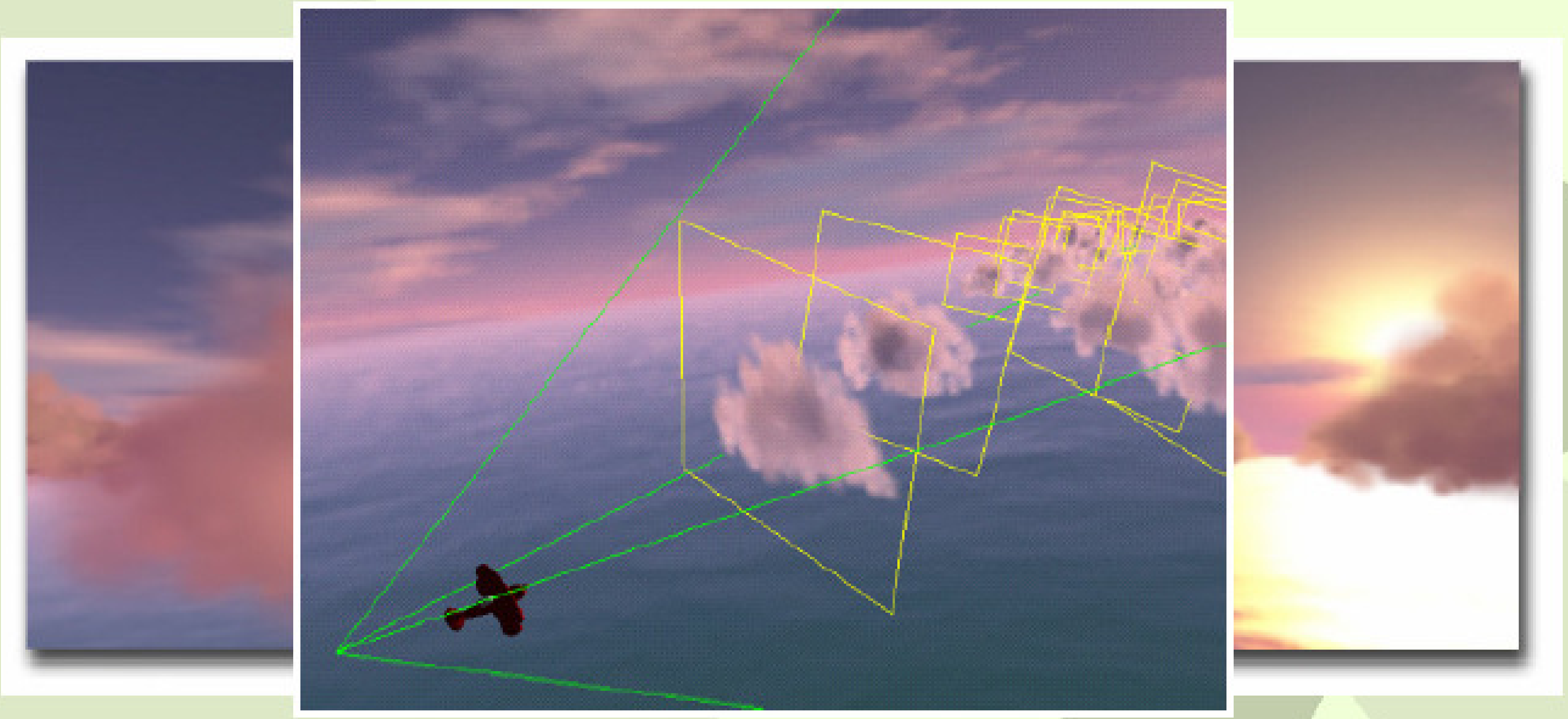
cromoslab
Virtual Humans Simulation

Skybox

cromoslab
Virtual Humans Simulation

Estado da Arte

- Mark J. Harris, Anselmo Lastra Real-Time Cloud Rendering, 2001, EUROGRAPHICS, Volume 20, Number 3.



cromoslab
Virtual Humans Simulation

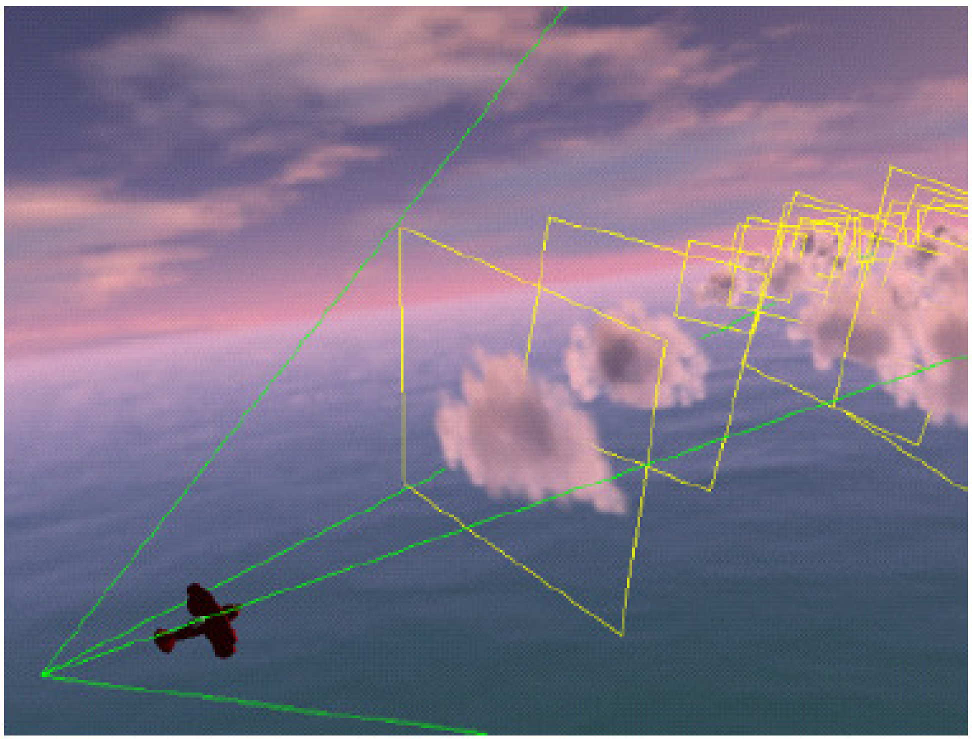
Estado da Arte (cont.)

cromoslab Virtual Humans Simulation

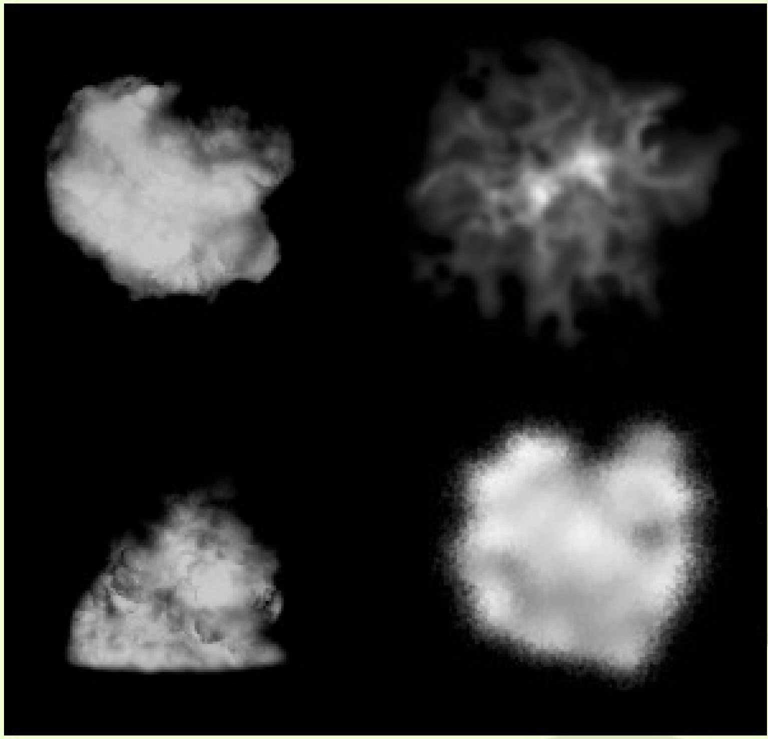
Visão Geral

Semente – Modelo Base – Disposição dos Billboards - Nuvem

Billboards



Utilização de impostores por Mark J. Harris

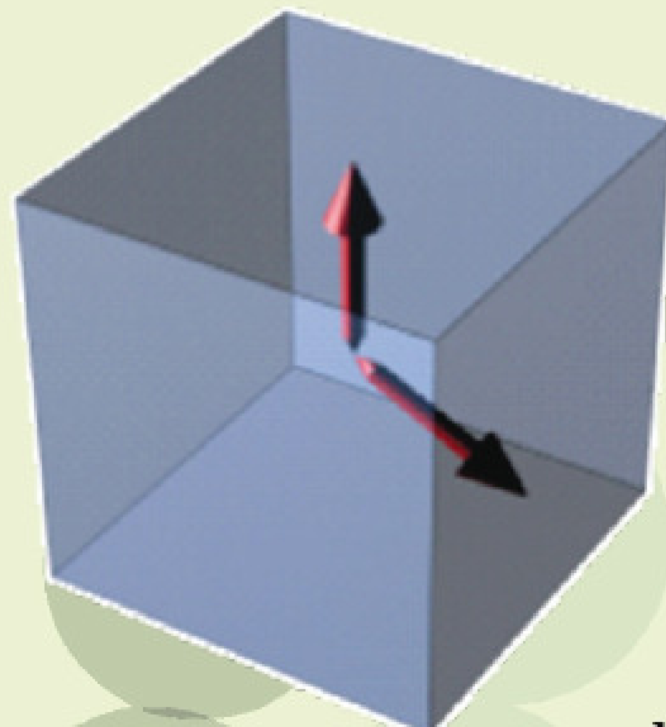


Exemplo de texturas utilizadas

cromoslab Virtual Humans Simulation

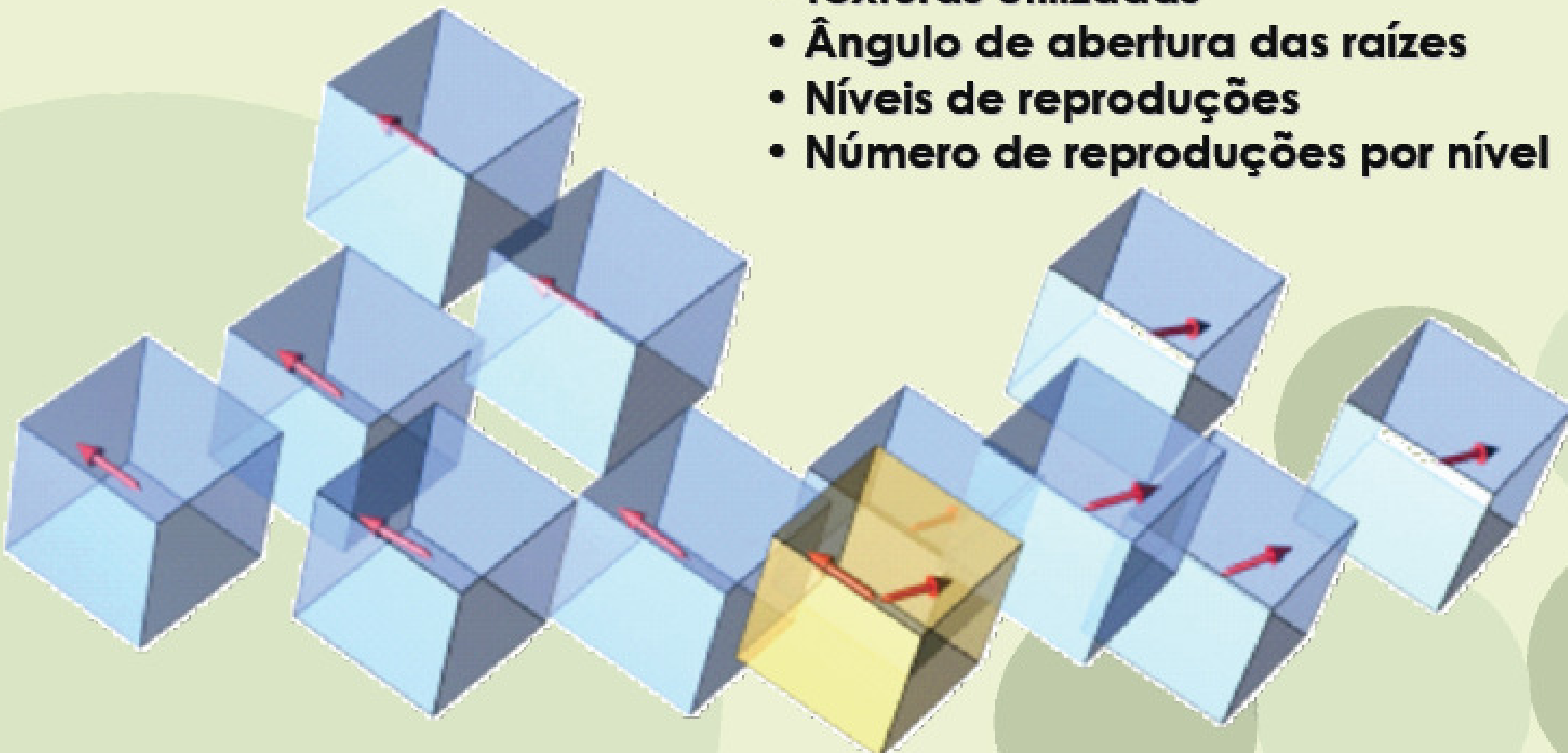
Sementes

- Billboards
- Raízes
- Absorção de luz
- Artista



Raízes

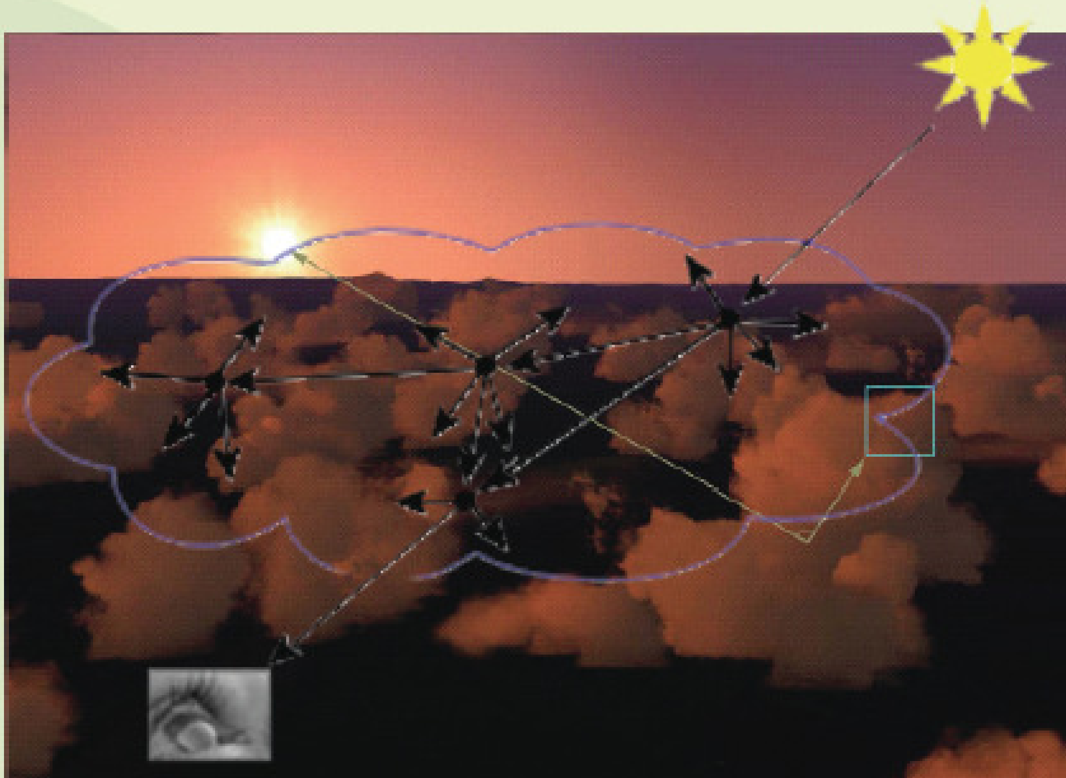
- Tamanho dos Billboards
- Distribuição dos Billboards
- Texturas utilizadas
- Ângulo de abertura das raízes
- Níveis de reproduções
- Número de reproduções por nível



cromoslab
Virtual Humans Simulation

Iluminação

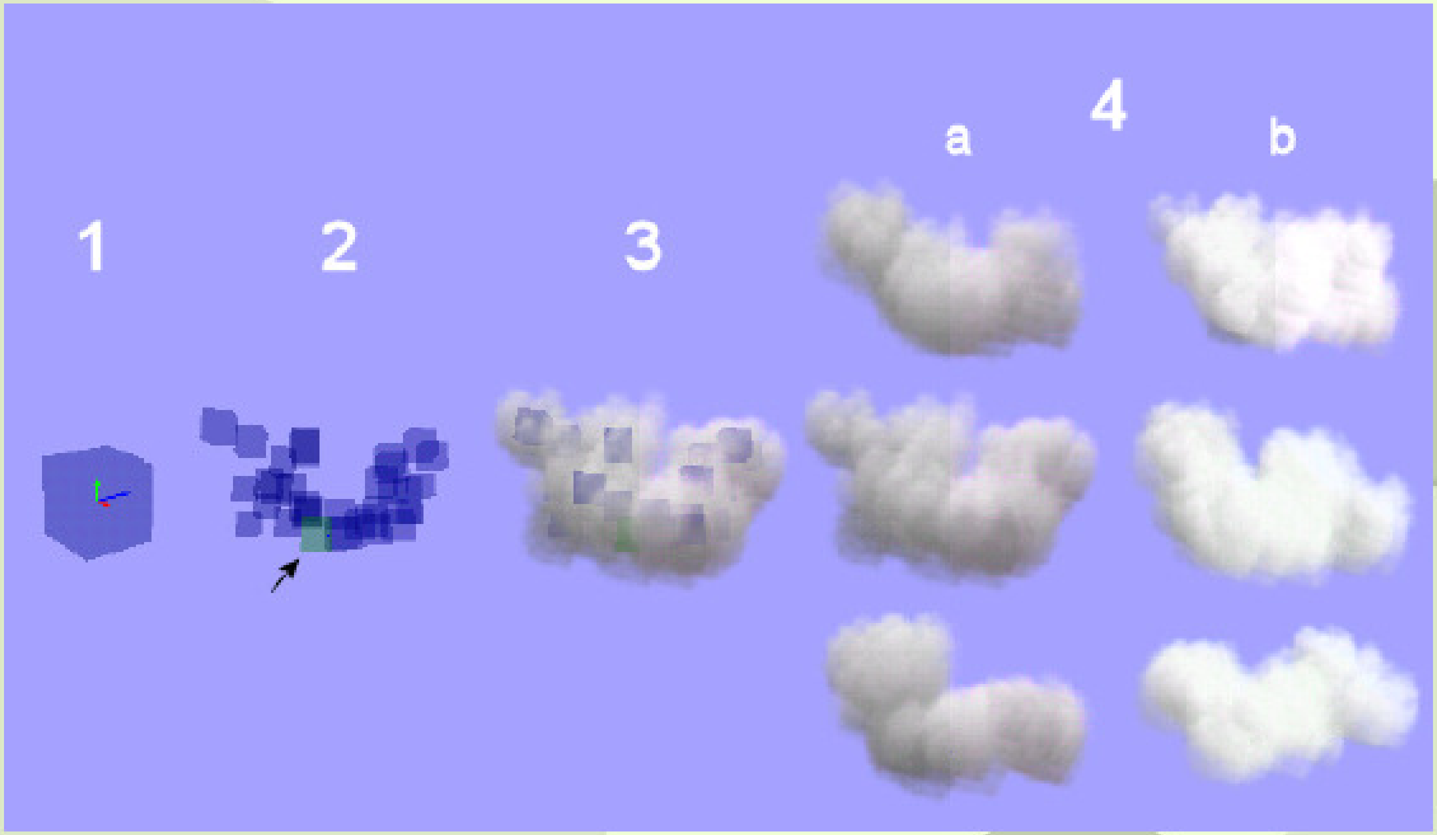
- Luz Direcional
- Vertex-Shader



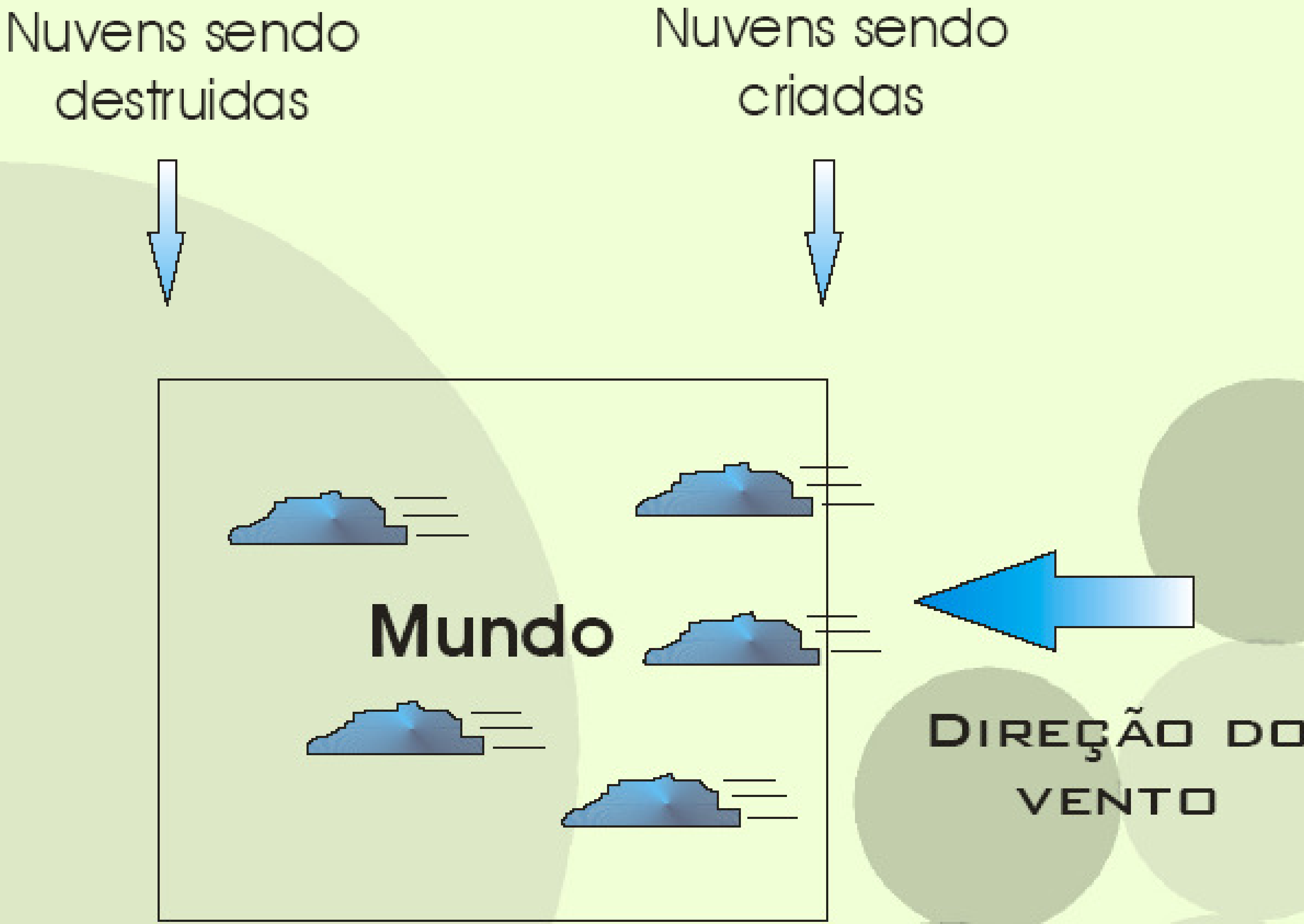
Modelo proposto por Niniane Wang

Céu

- Conjunto de Nuvens
- Representa um Estado Climático

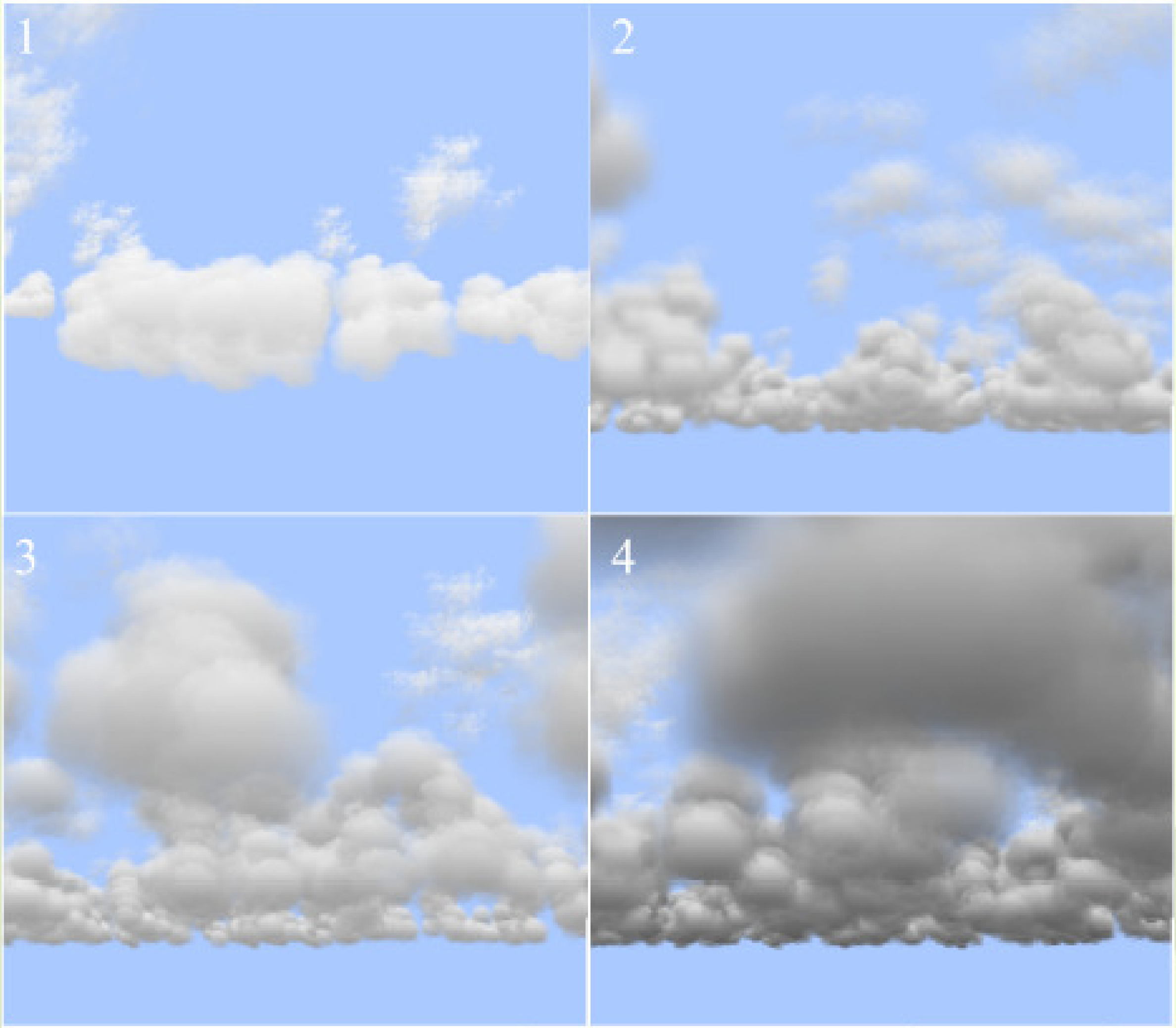


Nuvens


Nuvens sendo destruidas
Nuvens sendo criadas
Mundo
Direção do vento

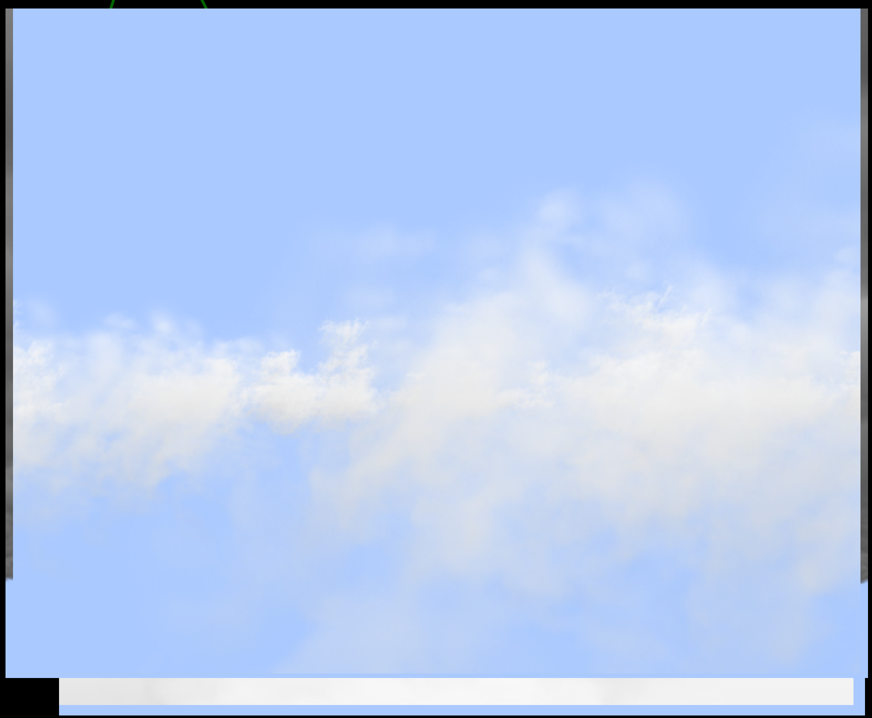
cromoslab
Virtual Humans Simulation

Variação Climática (cont.)



Resultados - Nuvens

Resultados - Céus

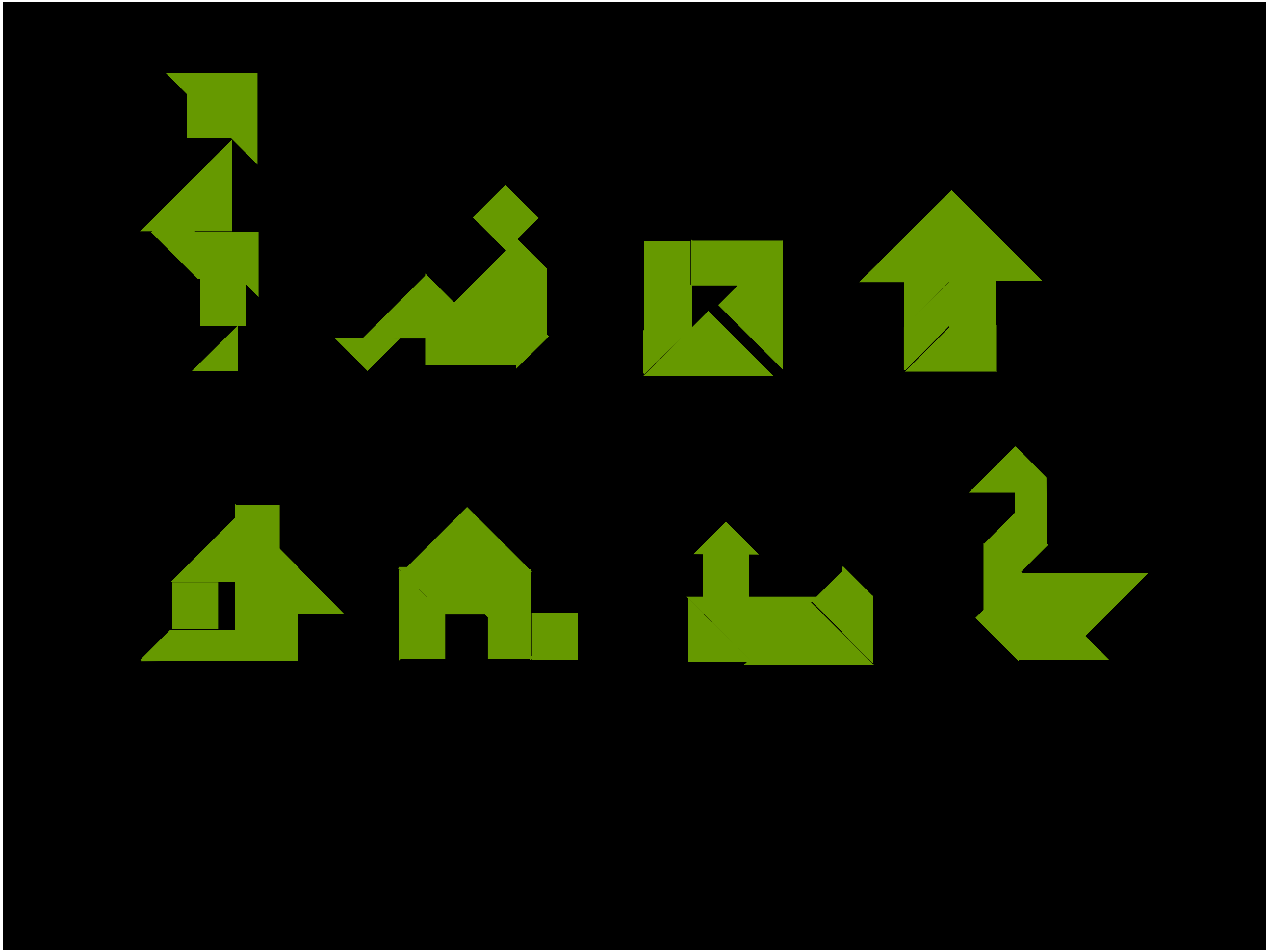


Fractostratus – Render

Fractostratus - Natureza

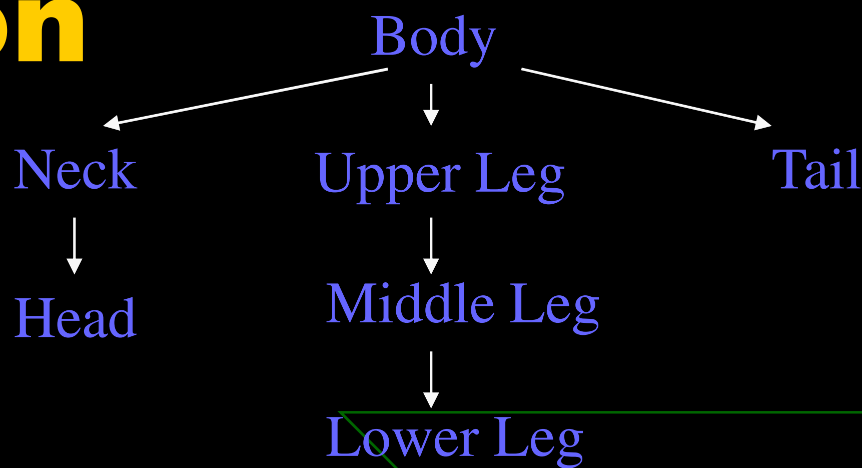
100 nuvens - 30 FPS

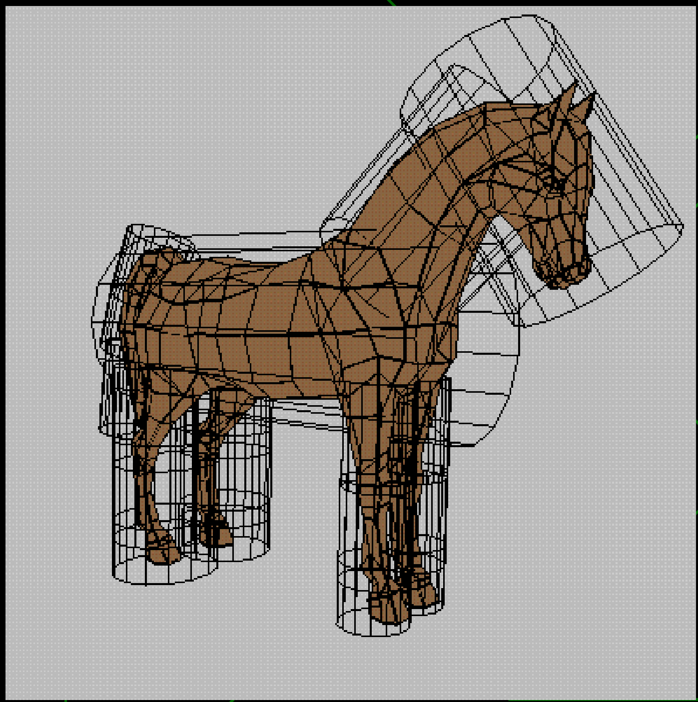
Vídeo



2 - Controlled Shape Transformation

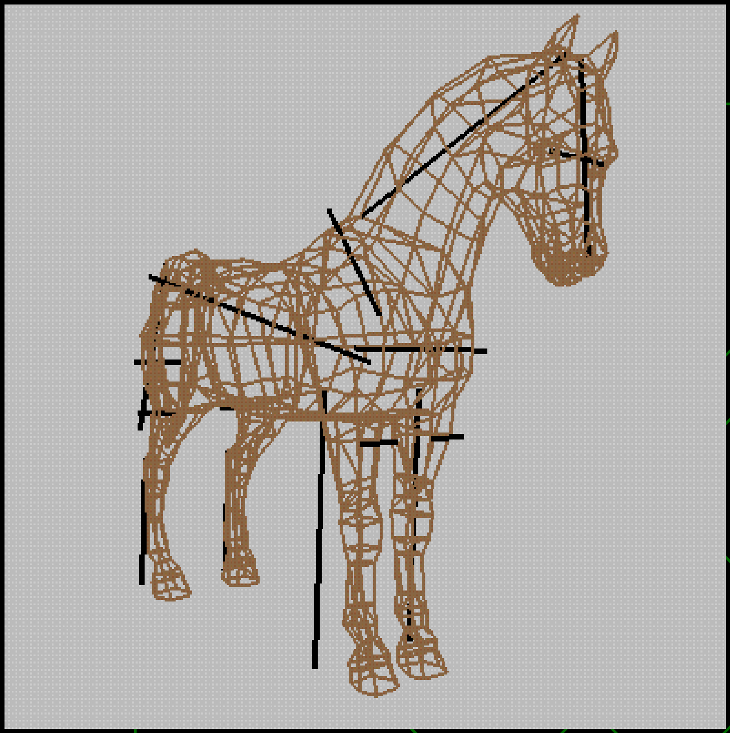
- Set of primitives defines a hierarchical structure. Ancestry is defined by the user
- Overlap of primitives guarantees continuity and smoothness
- Primitives are cylinders





Controlled Shape Transformation

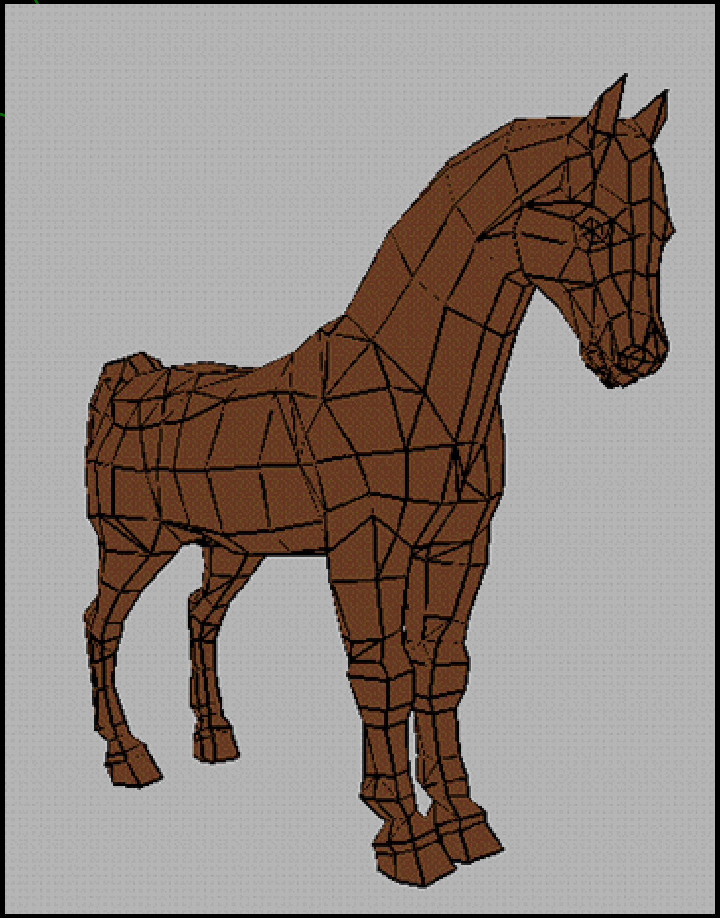
- Set of features drives the transformation
- Position and size of features match real measurements

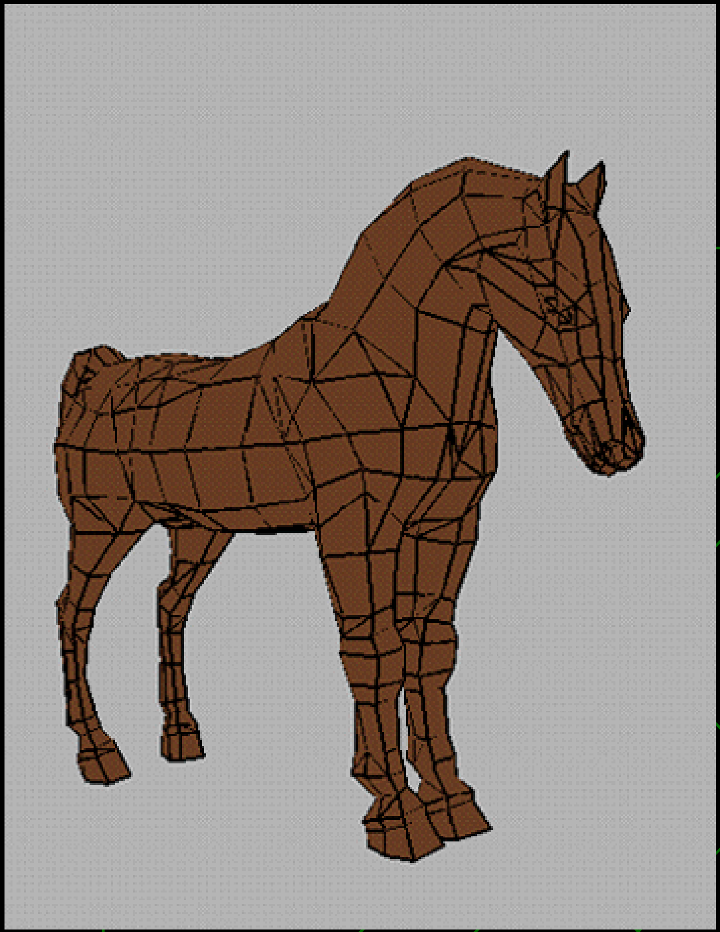


Example - Quarter Horse

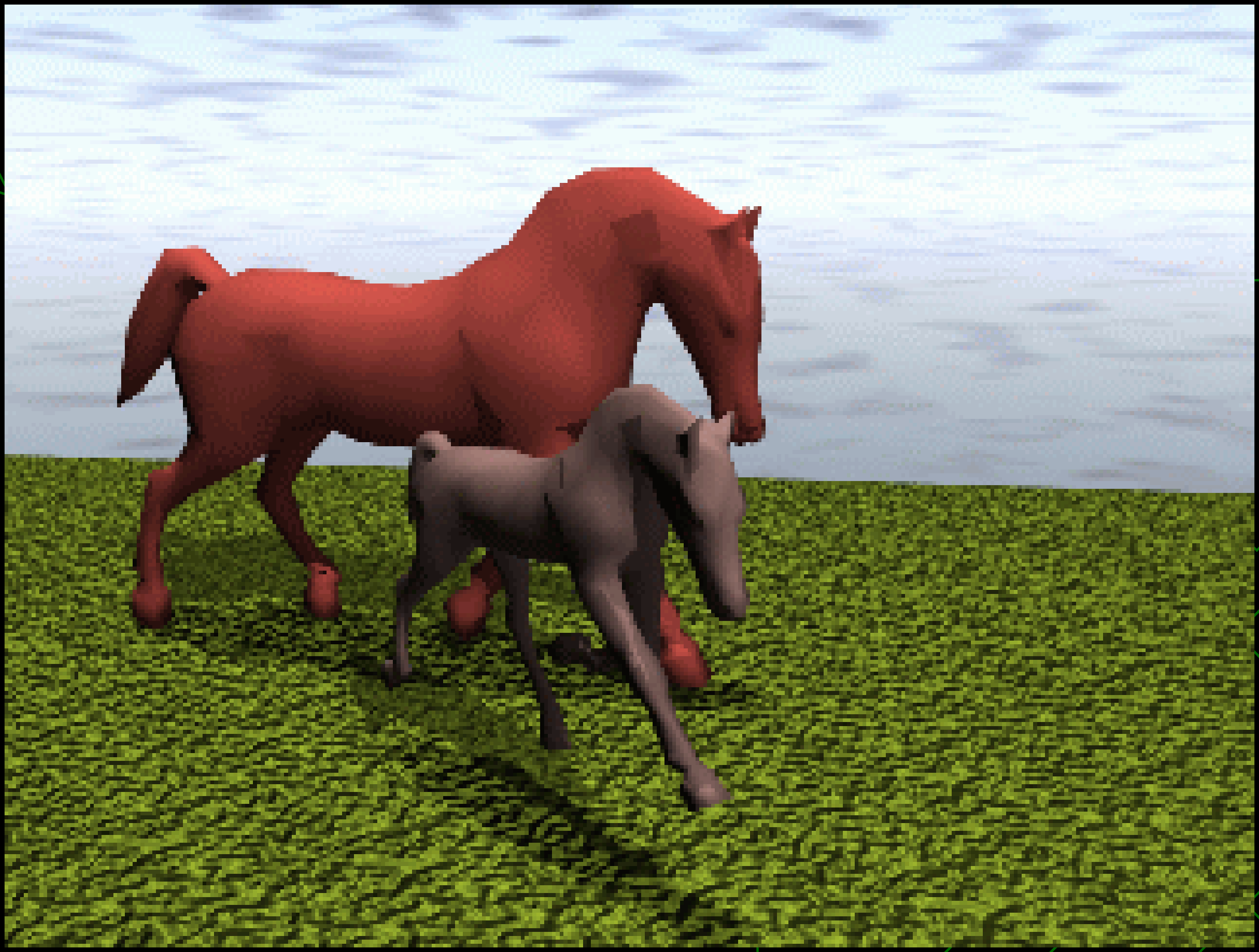
- 9 real measurements (3 legs, 2 body, 2 neck, 2 head) and 2 fake ones (tail)
- 9 ages: 0, 3, 6, 12, 18, 24, 36, 48, and 60 months
- Polygonal model with 674 vertices and 863 faces

Original and Quarter Horse





Applying Growth



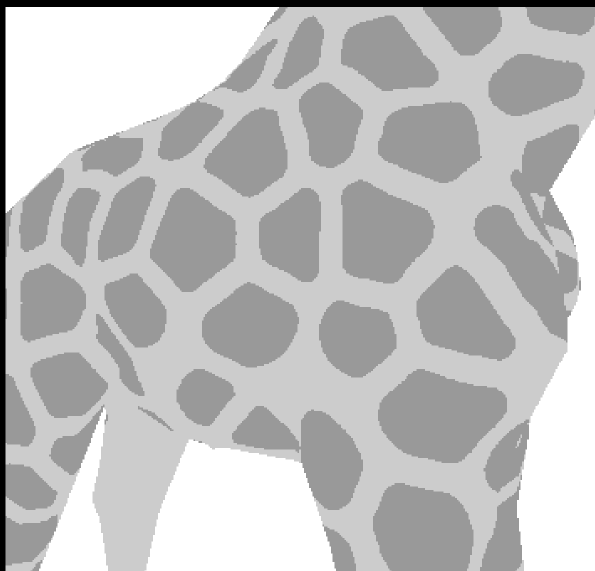
3 - Integration

- Simulate the CM model directly on the surface of geometrical models
  - Generate pattern on a fixed geometry
  - Change geometry and keep pattern
  - Develop pattern on a changing geometry
- Compute splitting rates from growth information

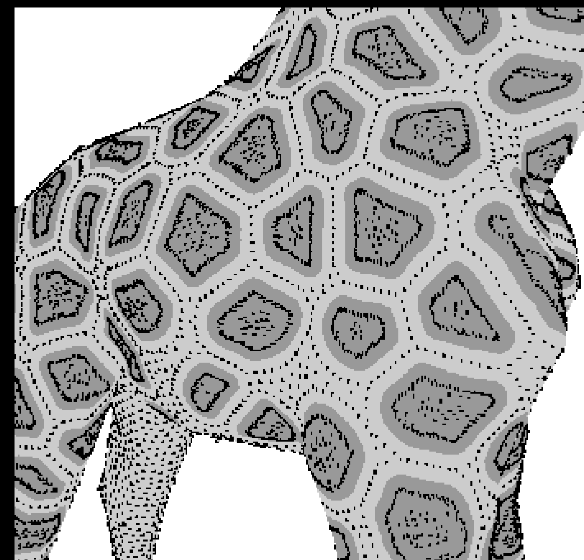
From 2D to 3D

- Distribution of random points on the surface
- Relaxation and computation of the Voronoi on the surface

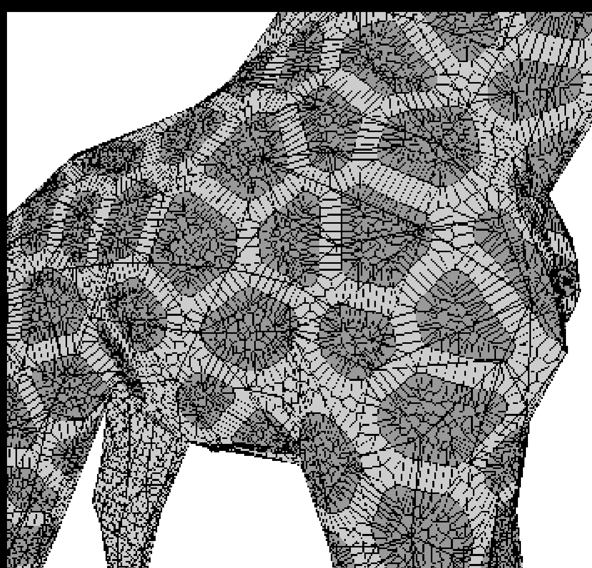
Pattern on the Surface



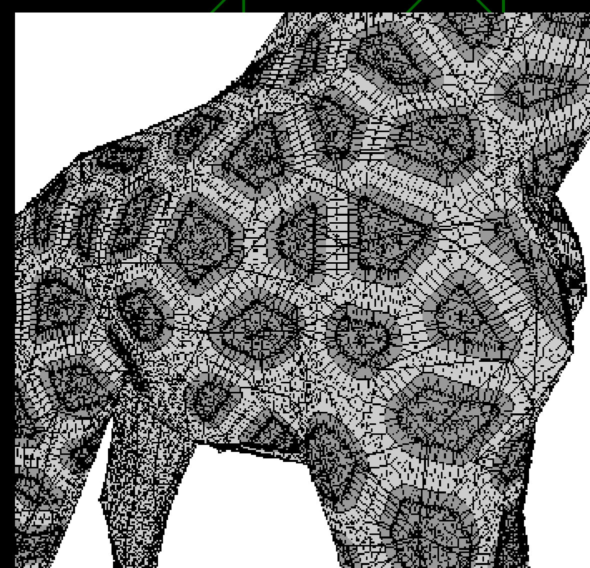
Pattern on the surface



Cell's centers



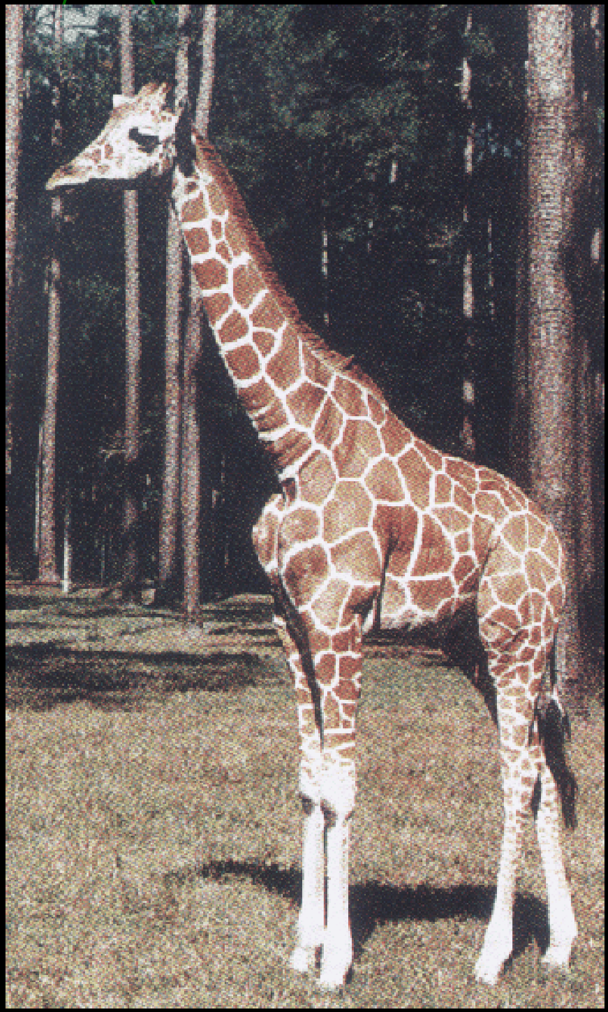
Voronoi borders

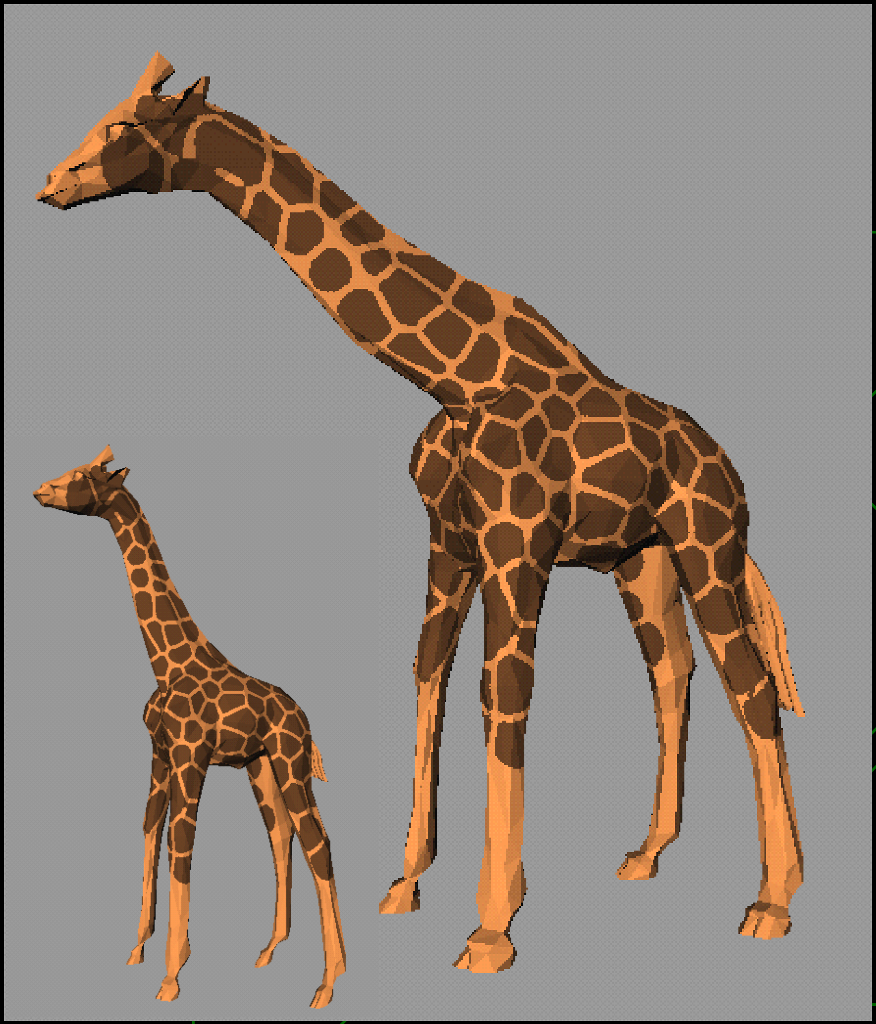


Cell's centers and borders

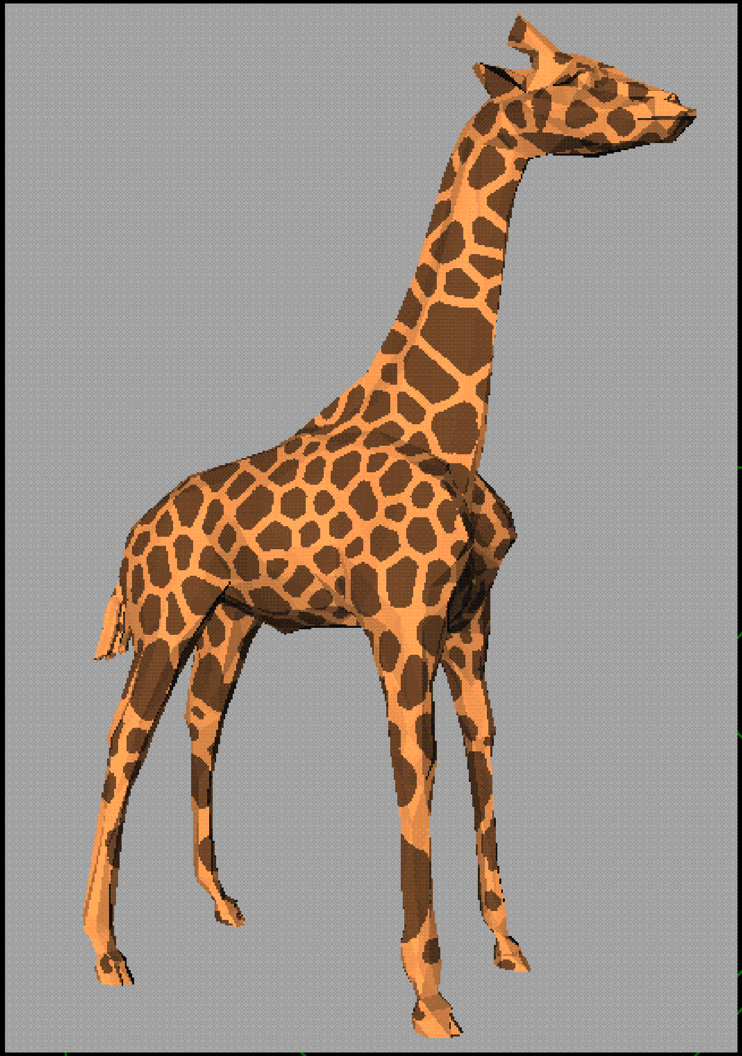
Pattern on a fixed geometry

Change geometry and keep pattern





Develop pattern on a changing geometry



85 days into gestation