# Modeling virtual urban environments

Julien Perret

# The goals

- Modeling complex environments
- Using real data
  - Height maps
  - Water maps
- Modeling the evolution of the environment
- Simulating emergent phenomena
- Visualizing the generated environments

# Previous work

- Natural environments
- Street networks
- Urban growth and suburban sprawl
- Buildings

# Natural environments L-systems, a quick overview

- Parallel string rewriting systems
  - Parallel application of rules
  - An application represents a lapse of time
- Rules of type
  - predecessor: condition → successor

# L-systems
# parallel application

**Modules:** A B

**Axiom ω:** B

**productions :**

A → AB

B → A

produces:

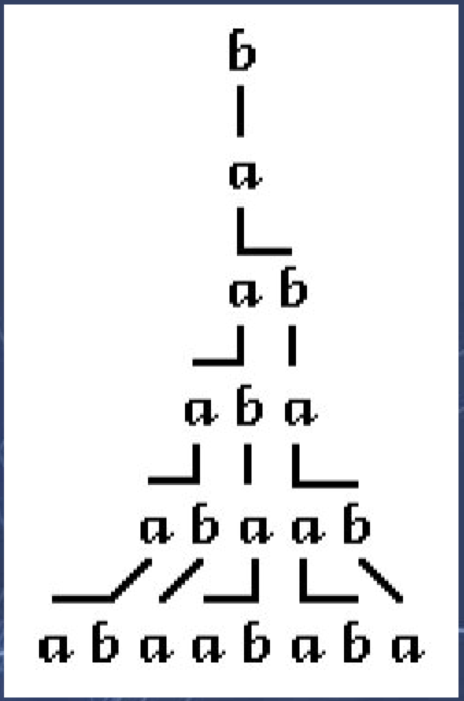
**n=0 B → A**

**n=1 A → AB**

**n=2 AB → ABA**

**n=3 ABA → ABAAB**

**n=4 ABAAB → ABAABABA**



# L-systems
# Turtle Interpretation

State of the turtle: (x, y, $\alpha$)

(x, y): ***Cartesian position* of the turtle**

$\alpha$: ***heading* of the turtle, i.e. the direction in which it is heading**

**Also**

d: ***step size***

$\delta$ **:** ***angle increment***

**Commands:**

$F$ **move forward a step of length** d **drawing a line segment.**

f **the same without drawing.**

\+ **turn left by angle** $\delta$.

\- **turn right.**

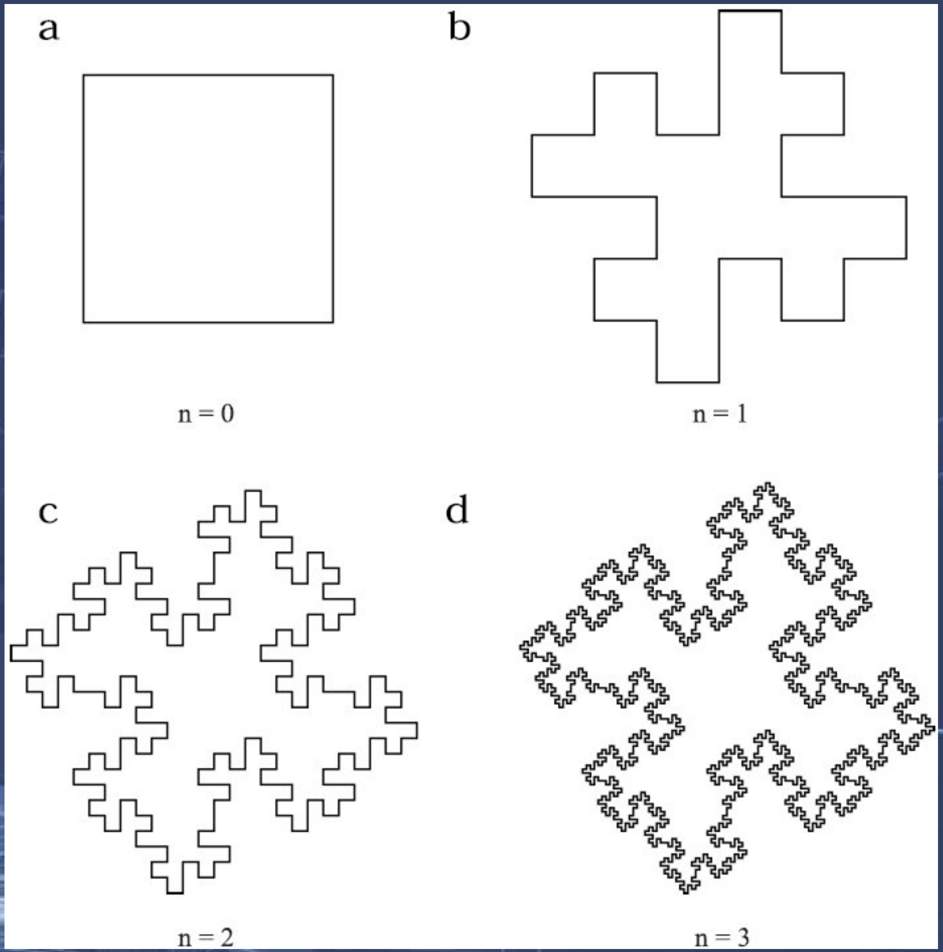
# L-systems
# Koch island

ω **F-F-F-F**

p

F→ F-F+F+FF-F-F+F

- **δ=90°**
- d **is decreased 4 times between each derivation step**



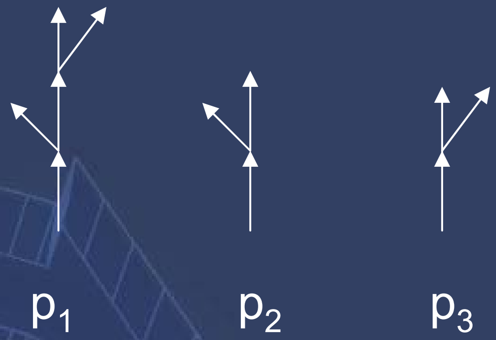
# L-systems
# Branching structures

$\omega$ **F**

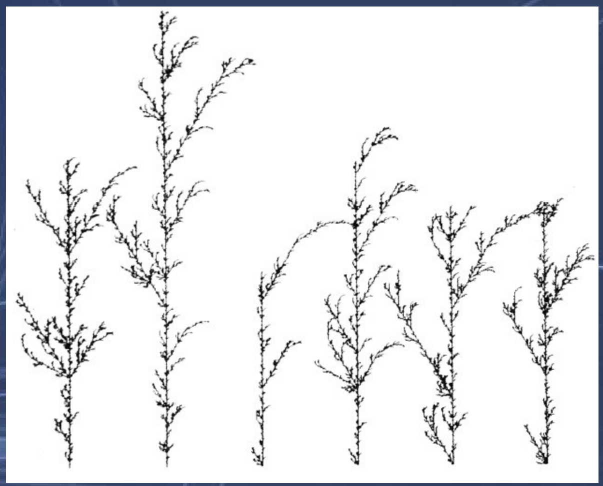
$p_1$ **F: .33 → F[+F]F[−F]F**

$p_2$ **F: .33 → F[+F]F**

$p_3$ **F: .34 → F[−F]F**




- ***[*** **and** ***]*** **create a branching structure**
- **Probabilities of application are**
  Added at the end of the rules
- **A single L-system creates a**
  variety of plants



# L-systems
# generation of trees

$\omega$ $FA(1)$

$p_1$ $A(k) : \min\{1, (2k + 1)/k^2\}$

$\rightarrow$ $/(\varphi)\ [+(\alpha)\ FA(k+1)]\ -(\beta)\ FA(k + 1)$

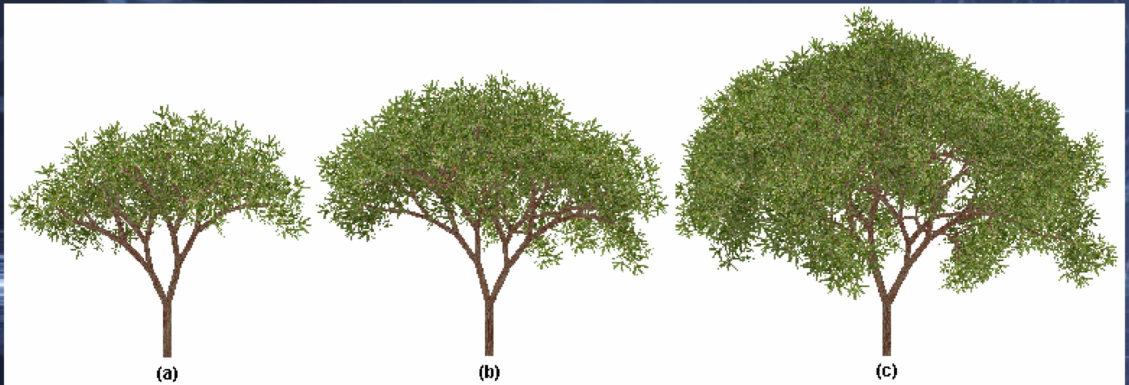
$p_2$ $A(k) : \max\{0, 1 - (2k + 1)/k^2\}$

$\rightarrow$ $/(\varphi)\ -(\beta)\ FA(k + 1)$

Modules +, – denotes rotation around the z-axis (up)

Module / denotes rotation around the y-axis

The angles for the rotations are specified for a given class of trees ($\alpha = 32^\circ$ , $\beta = 20^\circ$ , $\varphi = 90^\circ$ ).

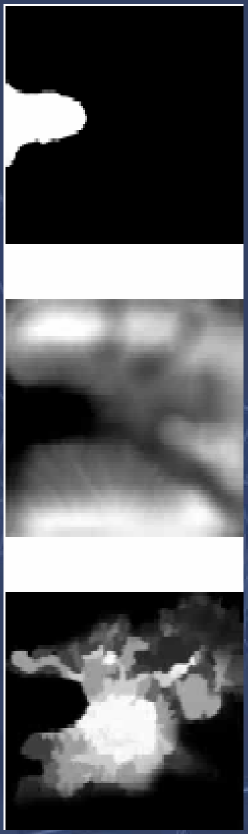


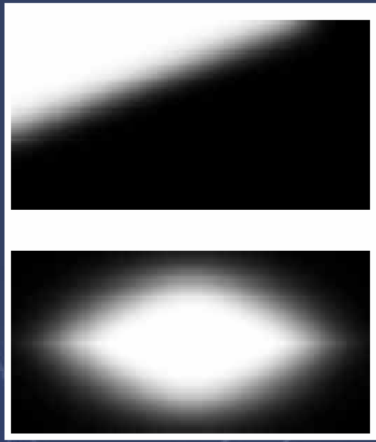
# L-systems applications

- Biological models
  - Cells
  - Plants
  - Trees
- More recently
  - Street networks
  - buildings

# Street network using L-systems

- Data
  - Maps
    - Water
    - Elevation
    - Population
    - Street pattern control





- A set of rules
- Some constraints

# Street network using L-systems the rules

$\omega$: R(0, initialRuleAttr) ?I(initRoadAttr, UNASSIGNED)

*p1*: R(del, ruleAttr) : del<0 $\rightarrow \varepsilon$

*p2*: R(del, ruleAttr) > ?I(roadAttr,state) : state==SUCCEED
{globalGoals(ruleAttr,roadAttr) creates the parameters
for: pDel[0-2], pRuleAttr[0-2], pRoadAttr[0-2]}
$\rightarrow$ +(roadAttr.angle)F(roadAttr.length)
B(pDel[1],pRuleAttr[1],pRoadAttr[1]),
B(pDel[2],pRuleAttr[2],pRoadAttr[2]),
R(pDel[0],pRuleAttr[0]) ?I(pRoadAttr[0],UNASSIGNED)

*p3*: R(del, ruleAttr) > ?I(roadAttr, state) : state==FAILED $\rightarrow \varepsilon$

*p4*: B(del, ruleAttr, roadAttr) : del>0 $\rightarrow$ B(del-1, ruleAttr, roadAttr)

*p5*: B(del, ruleAttr, roadAttr) : del==0
$\rightarrow$ [R(del, ruleAttr)?I(roadAttr, UNASSIGNED)]

*p6*: B(del,ruleAttr,roadAttr) : del<0 $\rightarrow \varepsilon$

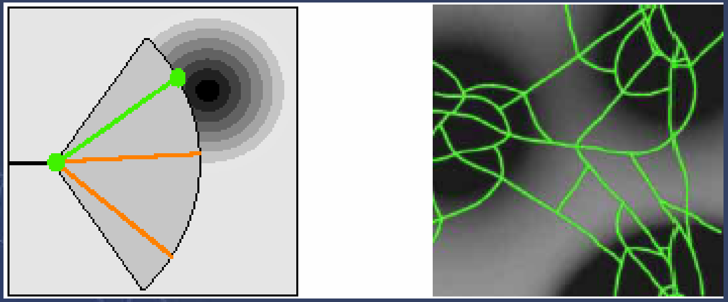
*p7*: R(del,ruleAttr) < ?I(roadAttr,state) : del<0 $\rightarrow \varepsilon$

*p8*: ?I(roadAttr,state) : state==UNASSIGNED
{localConstraints(roadAttr) adjusts the parameters for:
state, roadAttr}
$\rightarrow$ ?I(roadAttr, state)

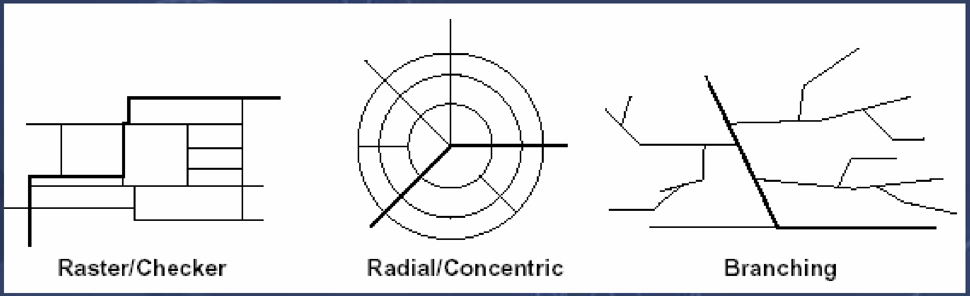
*p9*: ?I(roadAttr,state) : state!=UNASSIGNED $\rightarrow \varepsilon$

# Street network using L-systems

- Global constraints
  - Population density
    - For highways

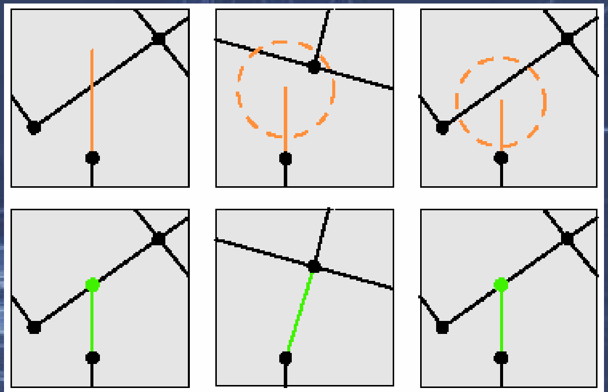


  - Road patterns




  - For streets

- Local constraints



# Street network using L-systems Limitations

- Invalid intermediate states
  - No intermediate model
- We need the actual maps
  - Why not using it?

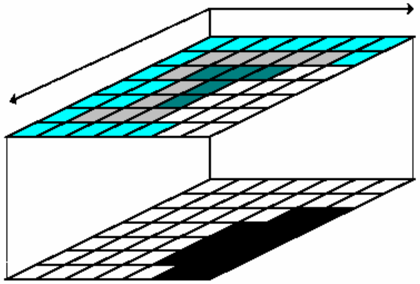
# Street network using L-systems Conclusion

- To model the evolution of the L-system properly, we need a model of the evolution of the population density
- Otherwise, we can use real data
  - To model the evolution of a street network, we need at least two of its states and the corresponding population maps

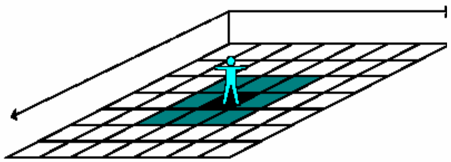
# Urban growth and suburban sprawl

- Simulating the evolution of the population
- Using
  - Cellular automata
  - Multi-agent systems

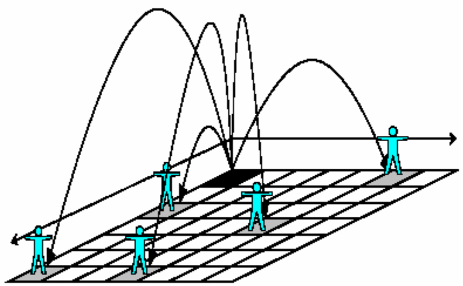
# Urban growth and suburban sprawl: the rules



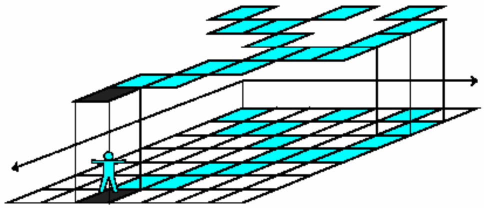
*Rule 1: Developable sites: cells develop which meet pre-existing constraints on land development*



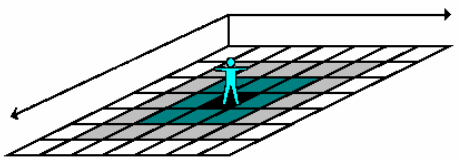
*Rule 2: Compact growth: agents locate on cells as close to the basic seeds or growth poles as is possible*



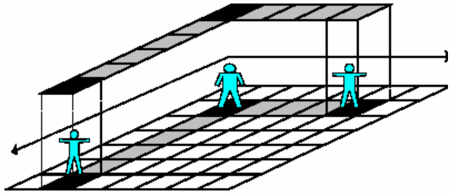
*Rule 5: Scattered growth: agents develop cells by leapfrogging the growing cluster to optimize access*



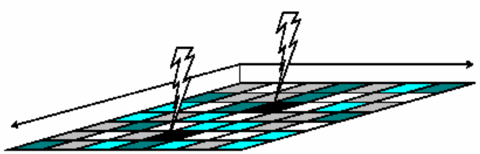
*Rule 6: Bifurcated growth: agents develop new cells which connect and compete with the existing cells*



*Rule 3: 'Hub'-influenced growth: agents develop cells to maximize agglomeration economies*



*Rule 4: Road-influenced growth: linear routes attract agents, thus development*



*Rule 7: Decline: cells lose their attraction to growth as they age or lose their comparative advantage*

# Urban growth and suburban sprawl: the results

*Simulation up to time 1850*
*t=50*

*Simulation up to time 1950*
*t=150*

*Simulation up to time 2000*
*t=200*

# Urban growth and suburban sprawl: limitations

- Works on grids (cellular automata)
- Do not build the street network
- We can add rules to make it converge to real data

# Goals

- Development of an environment in continuous time
- Unify the modeling of roads, buildings and plants
- Model interactions between objects

# Assumptions

- Roads and buildings are built by agents
- The number of agents is a ratio of the population
  - there is at least an agents at the beginning
  - non-builders are useless
- A chicken and egg problem:
  - What comes first?
    - Better transportation or population growth?
  - We assume it is the transportation

# the street network

- Rules of street network evolution
  - Using the terrain
  - Using the population
  - Centrifugal
  - Centripetal
  - Space-filling/closure
- The map of the population
  - updated by the construction of buildings
  - Determines the density of the street network

# the buildings

- Buildings are spread along the roads
- They trigger the evolution of the population
- They can be replaced if the density of population needs it
- Buildings construction is simulated using L-systems

# the agents

- Availability
  - The number of available agents evolves with the population
- Scheduling
  - We have to choose the roads and buildings that will be built first
  - This is done by attributing them an importance
  - The importance of a road construction is determined by
  - The importance of a buildings is defined by
- Communications
  - We may have to decide whether to destroy a building to build a road or not

# Modeling buildings using grammars

- Describing rules of architecture as grammar rules is an old idea
  - Vitrivus, Palladio, Durand
- This idea was proved usable using shape grammars
  - But shape grammars are automatically applied
  - Why not using L-systems instead?

# Buildings using L-systems

# Buildings using L-systems

- If we use L-systems, we can also simulate the buildings construction process

# Coming up next

- Urban features
- Grade-separated highways
- Bridges
- Adding functions
  - Residential
  - Commercial
  - Industrial
  - Historical
  - Community
  - College

# Coming up next

- Using real street networks
  - Acquisition
  - Interpolation between consecutive networks
  - Extracting/testing rules

# the plants

- Interactions with plants

# Urban growth and suburban sprawl

- Sprawl is defined as 'uncoordinated growth'