PA199
Advanced Game Design

Lecture 9

Procedural Forest and Cities

Dr. Fotis Liarokapis

20th April 2017

Forest Modelling

A Lot of Variety

Many different tree species present

Species Variation

Variation within each species

Plants and Vegetation Variation

Plants and vegetation variation

Modeling Forests

- Model a forest scene
  - Procedural modeling
  - Grammars and L-Systems

Procedural Modeling

- Use procedure to generate all needed geometric primitives
- Store complex tasks in procedure
  - Detail level
  - Shaders
  - Internal animation

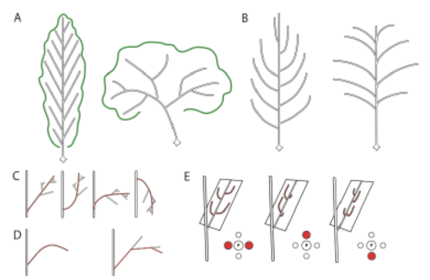
Procedural Trees

- Trees are a classic example of complex natural objects that can be procedurally modeled
- There have been numerous research papers published on various aspects of botanical modeling
  - One recent paper focused on creating the detailed sub-millimeter scale vein patterns seen on leaf surfaces

Procedural Trees .

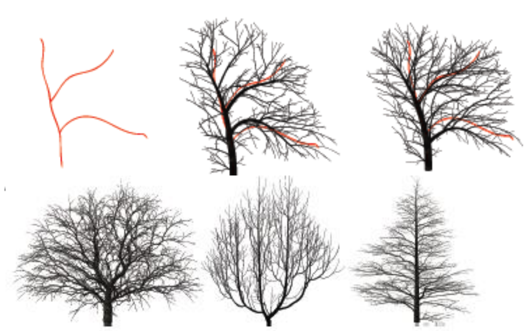
- By varying a number of key parameters, one can model a wide variety of plants and trees to any level of detail desired
- Even with about 10 parameters, can model a wide variety of overall plant shapes:
  - However real plant modeling systems may allow hundreds of parameters as well as the inclusion of custom geometric data to define leaf shapes or branch cross sections

Selected aspects of tree form



A) Tree silhouette and growth habits B) Branch curving: hyponasty and epinasty
C) Tropisms: no tropism, negative gravitropism, plagiotropism, positive gravitropism
D) Branch flow: curved, angular E) Gravimorphism: epitony, amphitony, basitony

Similar Tree Forms



Plant Morphology

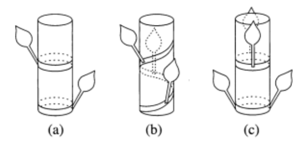
- An important part of a plants morphology is the **bud**
  - A bud is where every part of a plants body starts growing
- There are two different types of buds
  - The apical bud at the end of a branch
  - The lateral buds
    - Can grow new branches, leafs or flowers

Plant Morphology .

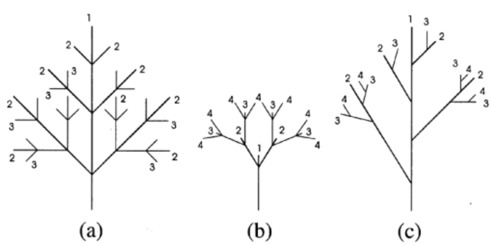
- The starting point for leaves (or sub-branches) at a branch is called **Nodus**
  - The part between two nodes is the Internodium
- A node can grow one or more leaves
  - The angular distance between all leaves grown at one Node will be constant

Plant Morphology ..

- Three basic types:
  - (a) One leaf per node, leafs of consecutive nodes are rotated by 180 degrees
  - (b) One leaf per node, but consecutive leafs are rotated by the mean building up a spiral shape
  - (c) Several leaves per node, leaves alternate



Branching Structure

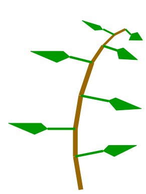


Primary Shoot Growth

- All plants grow according to the same basic pattern
  - Although there is a huge amount of variation within that pattern
- Growth takes place at tips of stems
  - Where new leaves are formed

Primary Shoot Growth .

- Primary growth includes:
  - Development of these new leaves at relatively regular intervals
  - Elongation of the stem between the nodes

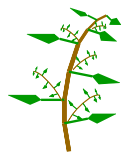


Axillary Growth

- At each leaf node an **axillary** bud is formed
  - Which may remain dormant
  - Or may develop into a whole new stem

Flowers

- Flowers will form at the tips of younger stems at certain times of the year, triggered by seasonal properties
  - i.e. day length or temperature
- Flower petals and other floral components are modified leaves themselves



Plant Shapes

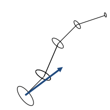
- The variety of plant shapes is mainly due to variations in stem shapes, branching properties, and leaf shapes, and these can be broken down into specific properties that can be used to control a procedural plant model
- A simple way to model a plant is to start by thinking of it as a bunch of branches (stems) and leaves

Branches

- Think of a branch as being a **circular extrusion** along some path
  - But it would certainly be possible to use non-circular cross sections as well
- The radius of the cross section will either remain **constant** or may **taper** from being **thicker** at the base and **thinner** at the tip
  - Simplify by specifying a radius at each end and assume a linear interpolation along the branch

Branches .

- The path of the branch can just be a set of points
- These points could be a straight line
  - Not be hard to add some curvature and randomness
- The path could be randomly created just from
  - A starting point, an initial direction, and a desired length

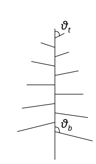


Branching

- Each branch can spawn off new branches
- The new branches would be placed at points along the original branch with some rule to define their initial direction
  - For example, it is nice to allow new branches to start at some percentage along the original branch

Branching .

- The length of the new branches can be defined by two percentages:
  - One describes the **length** of new branches at the **base** of the original branch
  - One describes the **length** at the **tip**
- The branching angle can be described similarly by values at the base and tip

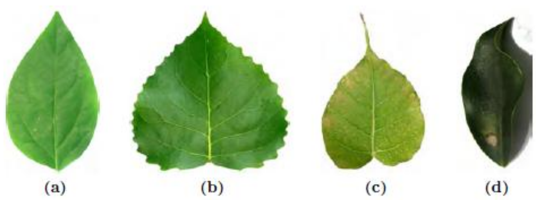


Branching and Leaves

- The branching is usually repeated **two or three times** so that we get sub-branches on sub-branches on branches
- At that point, we branch one final time
  - But create leaves instead of new branches
- The leaves can be placed along the branch according to similar rules as sub-branches

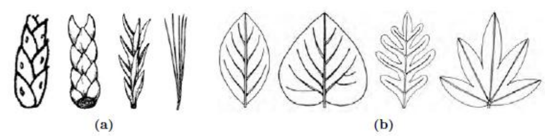
Shapes of Leaves

Details in the Leaf Shapes



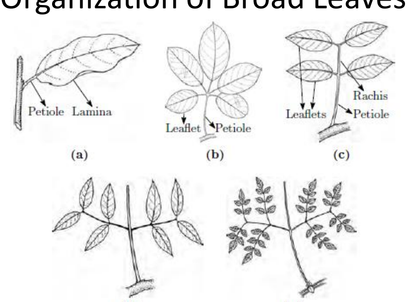
(a)2D leaf without any details, (b) leaf with jagged edges (teeth), (c) leaf with very sharp tip (drip-tip), and (d) leaf warped in 3D space

Types of Leaves



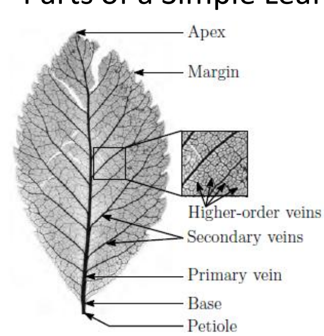
(a) Narrow leaves found in bryophytes, pteridophytes, and gymnosperms are slender
(b) Broad leaves found in angiosperms are wide and their thickness is negligible compared to the surface area

Organization of Broad Leaves

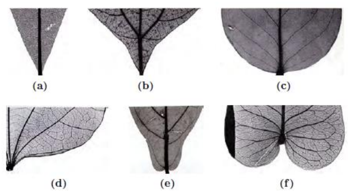


(a) A simple leaf has a single lamina
(b–e) A compound leaf has a lamina that splits into a number of small leaflets

Parts of a Simple Leaf

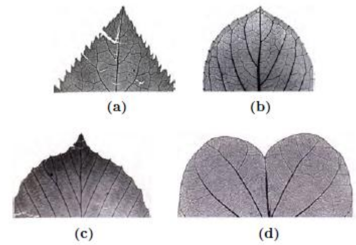


Types of Base Shapes



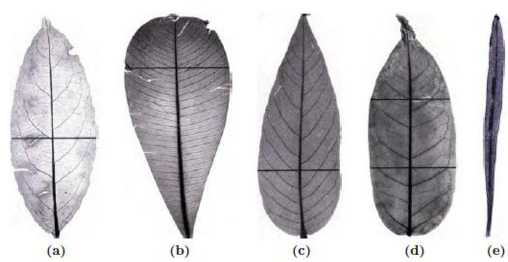
(a) Straight: the margin is straight (b) Concave: the margin curves towards the primary vein. (c) Convex: the margin curves away from the primary vein. (d) Concavo-convex: the margin is concave proximally and convex distally. (e) Complex: the margin has more then one point of inflection. (f) Cordate: the margin extends below the base

Types of Apex Shapes



(a) Straight: the margin is straight (b) Convex: the margin curves away from the primary vein (c) Acuminate: the margin is concave proximally and convex distally or concave only (d) Emarginate: the margin extends above the apex

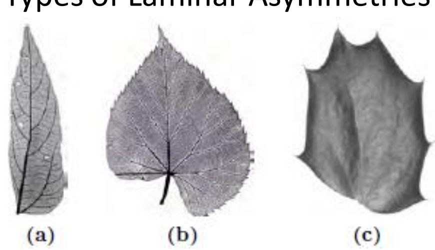
Types of leaves based on the position and Max width of the lamina



(a) Elliptic (b) Obovate (c) Ovate (d) Oblong (e) Linear

Common Leave Shapes

Types of Laminar Asymmetries

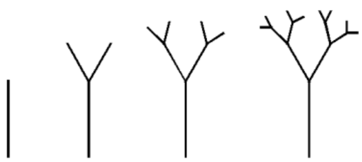


(a) A simple leaf with asymmetric maximum width (b) A simple leaf with asymmetric cordate base (c) A simple leaf with cordate base on one side and no extension on the other

Plant Example

Fractal Trees and Plants

- Fractal trees and plants are among the easiest of fractal objects to understand
  - They are based on the idea of self-similarity



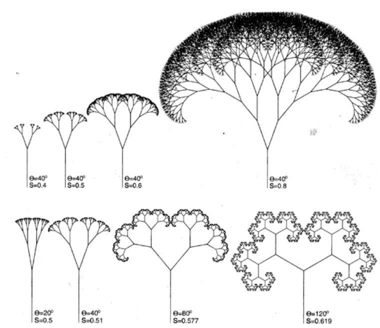
Fractal Trees and Plants .

- The main idea in creating fractal trees or plants is to have a base object and to then create smaller, similar objects protruding from that initial object
  - The angle, length and other features of these *children* can be randomized for a more realistic look
  - This method is a recursive method

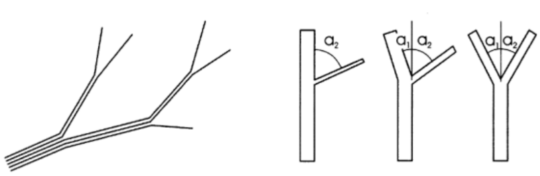
Fractal Tree Pseudo-Code

```
DrawTree(n, direction, length)
  if n > 0 do
    DrawTrunk(direction, length)
    DrawTree(n-1, 3DRandomAngle(direction),
   length*Factor(n))
    DrawTree(n-1, direction + random % 10, length*Factor(n))
    DrawTree(n-1, 3DRandomAngle(direction),
   length*Factor(n))
  else
    DrawLeaf()
  end if
end DrawTree
```

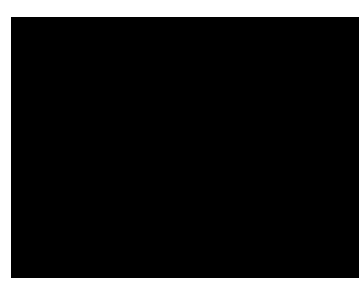
Fractal Branching Structures



Branch Strings



Trees and Plants Example



Grammars and L-Systems

- How can we change the structure of something?
  - With textures can fake it
- To modify the object itself, change its change or add new shapes then need to use L-Systems

Introduction to L-Systems

- Developed by A. Lindenmayer to model the development of plants
  - Originally described cellular growth
- Based on parallel string-rewriting rules
  - Formal set of rules and symbols
  - Rules applied iteratively to start sequence
- Excellent for modeling organic objects and fractals

L-Systems Definition

- The process of replacement can be formalized using a **grammar**, or parallel rewriting system is called an **L-system**

L-Systems Structure

- An L-System is a set of three things:

$$G = (V, w, P)$$

  - V = Alphabet
    - Set of symbols that can be replaced
  - w = Start or axiom
  - P = Set of production rules

L-Systems Structure

Variables: A B

Start: A

Rules: A -> B
B -> AB

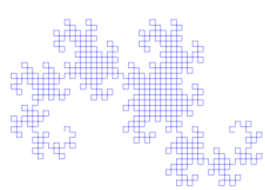
Process:

1. Take the Start (w) as the current string
2. Apply the production rule to *every* symbol in the current string.
3. When finished, call this the new string.
4. Repeat

start:	A	
iteration 1:	B	(rule A->B)
iteration 2:	AB	(rule B->AB)
iteration 3:	BAB	(rules A->B, B->AB)
iteration 4:	ABBAB	(rules B->AB, A->B, B->AB)

L-Systems Physical Interpretation

- L-Systems can have a physical interpretation, if we assign meaning to letters



The dragon curve drawn using an L-system

A variant of the Koch curve which uses only right-angles.

variables : F
constants : + −
start : F
rules : (F → F+F−F−F+F)

Here, *F* means "draw forward", + means "turn left 90°", and - means "turn right 90°" (see turtle graphics).

n = 0:

F

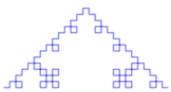
n = 1:

F+F-F-F+F

n = 2:

F+F-F-F+F+F+F-F-F+F-F+F-F-F+F-F+F-F-F+F+F+F-F-F+F

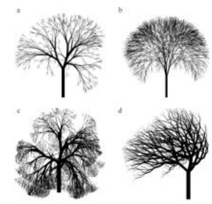
n = 3:



F+F-F-F+F+F+F-F-F+F-F+F-F-F+F-F+F-F-F+F+F+F-F-F+F+ F+F-F-F+F+F+F-F-F+F-F+F-F-F+F-F+F-F-F+F+F+F-F-F+F- F+F-F-F+F+F+F-F-F+F- F+F-F-F+F-F+F-F-F+F+F+F-F-F+F- F+F-F-F+F+F+F-F-F+F-F+F-F-F+F-F+F-F-F+F+F+F-F-F+F+ F+F-F-F+F+F+F-F-F+F-F+F-F-F+F-F+F-F-F+F-F+F-F-F+F+F+F-F-F+F

L-System for Plants

- L-Systems can capture a large array of plant species
- Designing rules for a specific species can be challenging



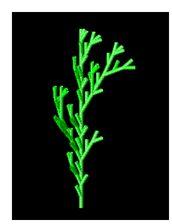
L-System for Plants .

variables : X F
constants : + −
start : X
rules : (X → F-[[X]+X]+F[+FX]-X), (F → FF)
angle : 25°

Here, F means "draw forward", - means "turn left 25°", and + means "turn right 25°". X does not correspond to any drawing action and is used to control the evolution of the curve. **[** corresponds to saving the current values for position and angle, which are restored when the corresponding **]** is executed.

A Simple L-System Example

- F [& + F] F[- > F] [- > F] [& F]



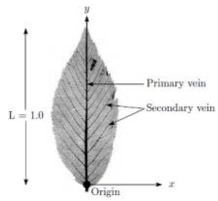
L-Systems Plans & Trees Examples

Parametric Leaf Model

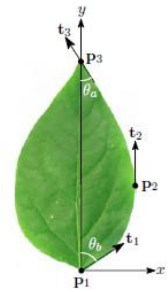
- Coordinate system of the leaf model
- Leaf without basal extension
- Leaf with basal extension

Coordinate system of the leaf model

- The parameters of the leaf model are expressed as:
  - A right-handed coordinate system with origin at the base
  - The y-axis pointing towards the apex of the leaf
  - The primary vein is defined to have a unit length

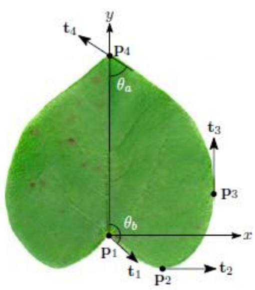


Leaf without Basal Extension



$$\mathbf{p}_1 = (0,0),$$
$$\mathbf{p}_2 = (x_2, y_2),$$
$$\mathbf{p}_3 = (0,1),$$
$$\mathbf{t}_1 = (\sin\theta_b, \cos\theta_b),$$
$$\mathbf{t}_2 = (0,1),$$
$$\mathbf{t}_3 = (-\sin\theta_a, \cos\theta_a).$$

Leaf with Basal Extension



$$\mathbf{p}_1 = (0,0),$$
$$\mathbf{p}_2 = (x_2, y_2),$$
$$\mathbf{p}_3 = (x_3, y_3),$$
$$\mathbf{p}_4 = (0,1),$$
$$\mathbf{t}_1 = (\sin\theta_b, \cos\theta_b),$$
$$\mathbf{t}_2 = (1,0),$$
$$\mathbf{t}_2 = (0,1),$$
$$\mathbf{t}_3 = (-\sin\theta_a, \cos\theta_a).$$

Other L-Systems

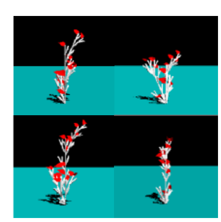
- A number of elaborations on this basic L-system technique have been developed which can be used in conjunction with each other
- Among these are:
  - Stochastic
  - Context sensitive
  - Parametric grammars

Stochastic L-Systems

- Specify more than one production rule for a symbol, giving each a probability of occurring
- When a stochastic grammar is used in an evolutionary context, it is advisable to incorporate a random seed into the genotype
  - So that the stochastic properties of the image remain constant between generations

Stochastic L-Systems .

- Several rules for each symbol
- Rules chosen based on probability
- Create different plants using same L-System
- Simulate environmental effects



http://www.csee.umbc.edu/~ebert/693/TLin/node18.html

Context Sensitive L-Systems

- A context sensitive production rule looks not only at the symbol it is modifying, but the symbols on the string appearing before and after it
- For instance, the production rule:
  - b < a > c → aa transforms "a" to "aa"
    - Only If the "a" occurs between a "b" and a "c" in the input string: …bac…

Context Sensitive L-Systems .

- Multiple productions to handle symbols in different contexts
  - Similar to stochastic productions

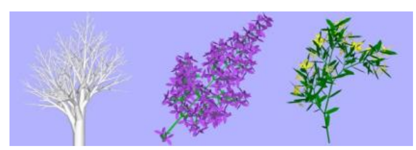
http://www.csee.umbc.edu/~ebert/693/TLin/node17.html

Parametric Grammars L-Systems

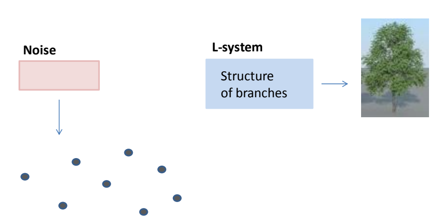
- Each symbol in the alphabet has a parameter list associated with it
- A symbol coupled with its parameter list is called a module, and a string in a parametric grammar is a series of modules
- An example string might be:
  - a(0,1)[b(0,0)]a(1,2)

Parametric Grammars L-Systems .

- Parametric grammars allow line lengths and branching angles to be determined by the grammar
  - Rather than the turtle interpretation methods



Complete Forest Generation

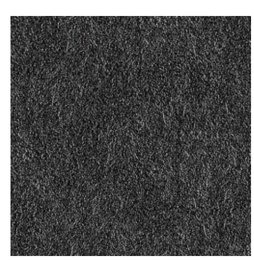


Noise can tell us where to plant trees.
L-system can tell us how to build a three-dimensional tree.

But do we make **one tree** and copy it, or do we **make lots of different trees**?

Noise

- What is randomness?



Randomness Definitions

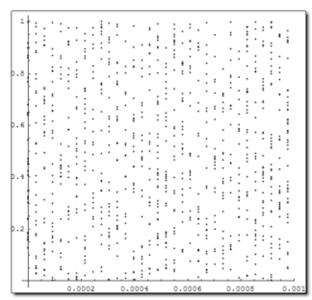
- "Having no definite aim or purpose; not sent or guided in a particular direction.."
  - – Oxford English Dictionary
- "Free will is limited to low-level decision making."
  - – Martin Luther
- "God does not play dice with the universe."
  - – Albert Einstein

What is Random?

Computer Random Numbers

- Numbers are generated as a sequence, starting from some seed point
- Problem is finding the next number so that it has nothing to do with the last

Plot of the rand() Function



int r = rand();

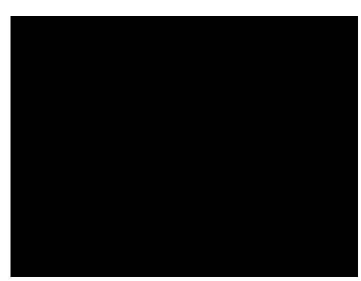
Randomness in CG

- Typical example is Fractals

"Clouds are not spheres,
mountains are not cones,
coastlines are not circles,
and bark is not smooth.."

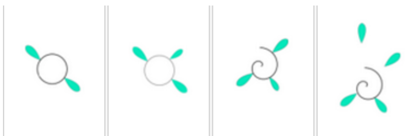
Benoit Mandelbrot, (b. 1924)
The Fractal Geometry of Nature, 1982

Mountain Forest Example



Spiral Phyllotaxis

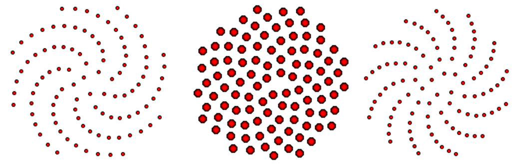
- Many patterns in nature can be understood with very simple iterated functions



Two leaves appear on opposite sides. A third appears between them. The first two push the 3rd away. The result is the *golden angle.*

Iteration, adding new leaves, settles into the minimization of energy, which results in the golden ratio (angle)

Fermat's Spiral



Phyllotaxis Pattern - Sunflowers

Given a sequence of integers *n*:

$r = c\sqrt{n}$
$\Theta = n * 137.50$ (golden angle)

Where

r = distance from center
Θ = angle

Phyllotaxis Examples

A Collision-based Model of Spiral Phyllotaxis, 1992
Deborah Fowler, Przemyslaw Prusinkiewicz, Johannes Battjes

Plant Ecosystems

- Very common is the use of a pipeline system to create ecosystems, consisting of a number of succeeding steps such as:
  - Terrain Specification
  - Distribution Specification/Simulation
  - Plant Specification

Plant Ecosystems .

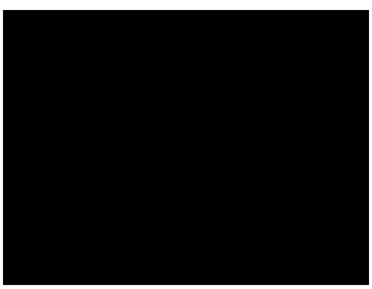
- Depending on the application there are two steps in generating a virtual ecosystem (can be used alone or in conjunction):
  - Generating/specifying plant distributions
  - Simulating plant distributions

Forest Example 1

Some times **instancing** can look pretty good. Pines trees all look similar

Forest Example 2

Terrain and Forest Example

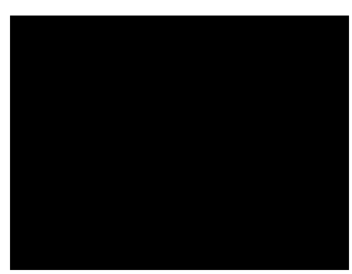


City Modeling

Introduction to City Modeling

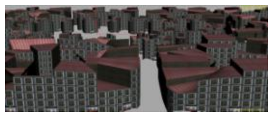
- Modeling 3D environments expensive
- Use CGA shape to create large detailed urban environments

GIS and City Modeling



SkylineEngine

- Research oriented project
- Aims at the development of a procedural urban modeling engine

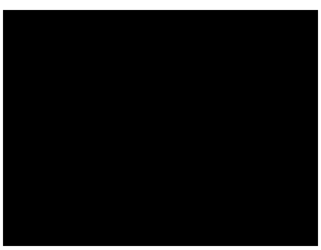
http://ggg.udg.edu/skylineEngine/

Pixel City

- Written in C++ using OpenGL
- No fragment or vertex shaders are used

http://code.google.com/p/pixelcity/

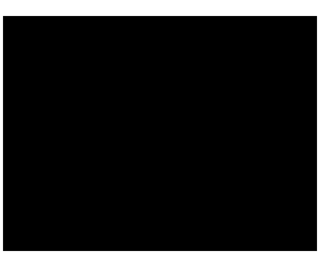
Pixel City Example



CityScape

- By PixelActive
- Features:
  - Enhanced road editing capabilities, improved GIS importing, a flexible meta data system, new import and export formats, a redesigned user interface with icons and hotkeys, and performance improvements
- The road editing tools add a tangent editor, tunnels, and complex intersections

CityScape Example



CityEngine

- Uses maps of population density to create a road network using Open L-systems
  - Spaces between blocks subdivided into lots for individual buildings
- Buildings generated using parametric stochastic L-systems, detail


http://www.esri.com/software/cityengine

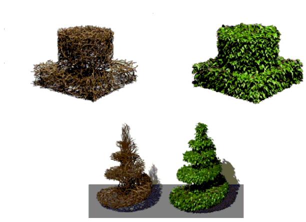
City Engine Example

Environmentally Sensitive L-Systems

- So far, L-systems have been more or less self-contained
- Parameters provide global influences from the environment
  - Provide a way to have the environment affect the development of the system locally

Environmentally Sensitive L-Systems Examples

- Examples from Synthetic Topiary, by P. Prusinkiewicz, M. James & R. Měch



Open L-systems

- Generalisation of environmentally sensitive L-systems
- When interpreted, control is passed to an external function representing some part of the environment
- Environment performs calculations, optionally passes back modified parameter values to be used in the next derivation step
- Two-way communication between system and environment

Procedural Road Modeling

- Roads can be modeled as **cross sections** that get path extruded along some curve
- The cross section can include:
  - Lanes, curbs, sidewalks, center islands
- Intersections require special handling
  - Can still be generated using a set of procedural techniques

Procedural Road Modeling .

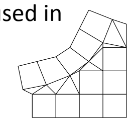
- Roads can be placed on **height fields** by placing the control points of the road curves on the height field
  - Note that this will require some local flattening for the road and the area to the side of the road
- Render road triangles onto the height field
  - Where a low detail road is extruded and ‘rendered’ into the cells of the height field to set their heights

Procedural Road Modeling ..

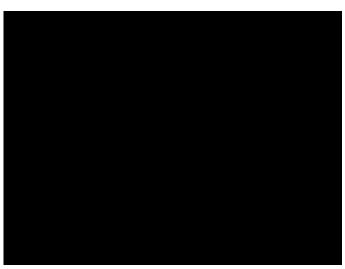
- Sides of roads can be blended
  - Using techniques similar to alpha blending
- Note that the above operations will modify the existing shape the height field
  - Similarly to how real roads are constructed

Procedural Road Modeling ...

- Ideally, the road surface would be an extrusion and the open terrain would be a height field
- They can be sewn together into a single triangle mesh
  - Similarly to how trim curves are used in patch tessellation



Road Modeling Example



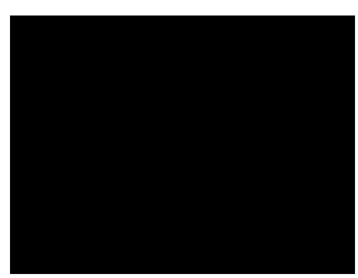
CityEngine - Road Generation Algorithm

- Uses Open L-systems for road generation
- Environment functions are the constraints imposed on the roads
- Two types of roads: highways and streets
  - Highways connect areas of high population density
  - Streets connect the rest of the populace to the nearest highway

Example of Production Rules for Roads

```
– ω: R(0, initRuleAttr)?I(initRoadAttr, UNASSIGNED)
– p1: R(del, ruleAttr): del < 0 → ε
– p2: R(del, ruleAttr) > ?I(roadAttr, state): state ==
        SUCCEED
        → +(roadAttr.angle)F(roadAttr.length)
              B(del1, ruleAttr1, roadAttr1)
              B(del2, ruleAttr2, roadAttr2)
              R(del0, ruleAttr0)?I(roadAttr0,UNASSIGNED)
– p3: R(del, ruleAttr) > ?I(roadAttr, state): state ==
        FAILED
        → ε
– p4: B(del, ruleAttr, roadAttr): del > 0
        → B(del – 1, ruleAttr, roadAttr)
– p5: B(del, ruleAttr, roadAttr): del == 0
        → [R(del, ruleAttr)?I(roadAttr, UNASSIGNED)]
– p6: B(del, ruleAttr, roadAttr): del < 0 → ε
– p7: R(del, ruleAttr) < ?I(roadAttr, state): del < 0
        → ε
– p8: ?I(roadAttr, state): state == UNASSIGNED
        → ?I(roadAttr, state)
– p9: ?I(roadAttr, state): state != UNASSIGNED
        → ε
```

CityEngine Parcel Generation



Population Density

- Highways connect peaks on **population density map**
- End of segment shoots rays across map
  - Each ray is sampled and sample points look up population value
- Highway generation continues in the direction of the fittest ray

Street Patterns

- New York style
  - Streets form blocks by restricting the length of a block and the angles that streets follow
- Paris style
  - Concentric rings around a central point, connected by short radial streets
- San Francisco style
  - Tries to reduce the length of non-contour streets and uses gradient of elevation map

Other Constraints

- Modify attributes (length, angle) of road segments in response to surroundings
- Make sure roads do not cross water and any other places you'd expect not to find them
- Creates intersections with other roads

Road Intersections

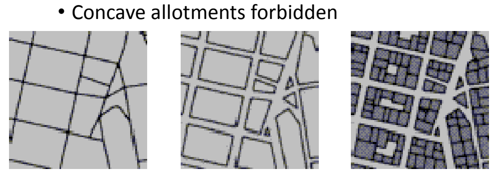
- No more dead ends than intersections
- No intersecting segments without creating an intersection (node in the road graph)
- Three rules:
  - 2 streets cross: generate an intersection
  - Segment ends close to intersection: join it to the intersection
  - Segment ends close to another segment: extend it and create an intersection

Buildings Modeling

- Once the roadmap has been generated
  - Allotments can be generated
  - Geometry can be generated
  - Buildings can be assigned textures
    - Preferably procedural

City Block Divisions

- City divided into blocks
- Blocks subdivided into allotments
  - Blocks assumed to be convex and rectangular
    - Concave allotments forbidden

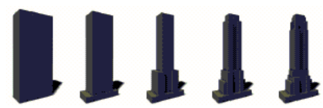


City Block Divisions .

- Simple recursive algorithm
  - Divide longest edges that are approximately parallel
  - Stop when size less than threshold value
- Allotments without street access or too small discarded
- Maximum building height from image map

Building Geometry

- Parametric, stochastic L-System
- One building per allotment
- Each building style has own set of production rules
- Manipulate arbitrary ground plan

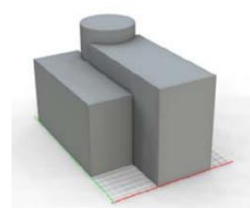


Building Geometry .

- L-System modules:
  - Transformation modules
  - Extrusion module
  - Branching and Terminating modules
  - Geometric templates
- Final shape is ground plan transformed by L-System
- Building functionality not represented

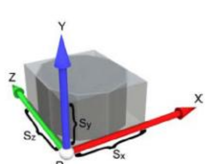
Mass Models Generation

- Crude volumetric building model
- Composed of shape primitives



Shapes Generation

- Used by the grammar
- Consists of:
  - Symbol (string)
  - Geometric attributes
    - Position
    - Coordinate System
    - Size
  - Numeric attributes

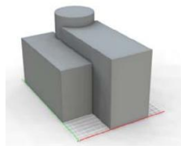


Production Process

- Configuration: Finite set of basic shapes
- Start configuration of shapes
  - Process:
    - Select active shape with symbol B
    - Choose a rule to compute a successor for B
    - Mark B as inactive, add the shapes BNEW to configuration and repeat
- Stop when there are no more non-terminals
- Assign a picking priority based on detail

Scope Rules

- General shape modifying rules
  - Translate, Rotate, Scale

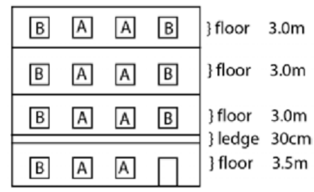


1: A ⇝ [T(0,0,6) S(8,10,18) I("cube")]
T(6,0,0) S(7,13,18) I("cube") T(0,0,16) S(8,15,8) I("cylinder")

Basic Split Rule

- Split current scope along one axis
  - For example:

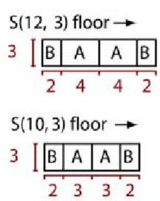
1: fac ⇝ Subdiv("Y",3.5,0.3,3,3,3){ floor | ledge | floor | floor | floor }



Scaling

- Relative values can be used

1: floor ⇝ Subdiv("X",2,1r,1r,2){ B | A | A | B }



Repeat

- Tile a specified element
  - Example:
    - Tile as many B elements as possible along X axis
    - Number of elements = $\lceil x - length/2 \rceil$

1: floor ⇝ Repeat("X",2){ B }

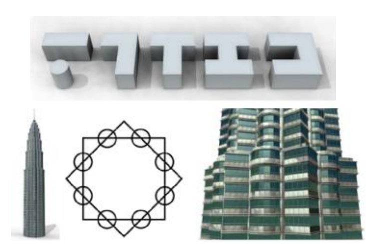
Component Split

- Split into shapes of lesser dimensions
- type = type to split
- param = associated parameters (if any)
- {A| …} = symbol of shape after split

1: a ⇝ Comp(type, param){ A | B | ... | Z }

Mass Modeling

- Union of volumetric shapes
  - Box, Cylinder, etc



Roofs

- Add roof shapes
- Categorize different types of roofs
  - i.e. according to building style or height

Façades

- Extract 2D façades from 3D shapes
- Result
  - Aligned 2D scope
- Grammar refines resulting quads and triangles

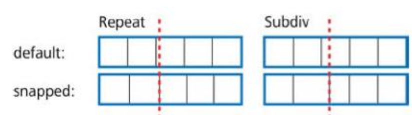
Occlusion

- Test for intersections between shapes
- None, partial, or full
- Avoid placing façade elements on intersections

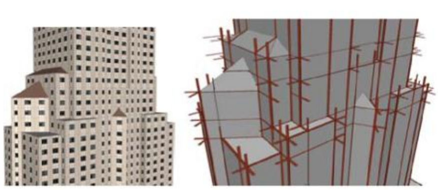
```
1:  tile : Shape.occ("noparent") == "none" ⇝ window
2:  tile : Shape.occ("noparent") == "part" ⇝ wall
3:  tile : Shape.occ("noparent") == "full" ⇝ ε
```

Snapping

- Additional control of façade layout
- Used by repeat and subdivide



Snap Lines Example



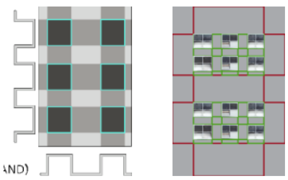
Note floor levels are aligned

Façade Textures

- Division into simple grid-like structures
- Structures can be layered

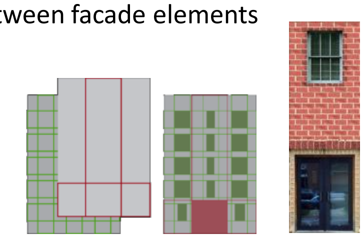
Layered Textures

- Two base functions form a layer
- Every layer defines a facade element



Layering of Planes

- Stacked layers for facade texture
- Functions between layers model relation between facade elements



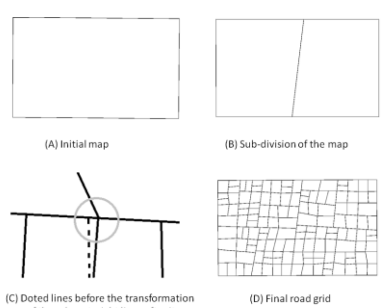
Case Studies for City Modeling

- New Classic Buildings
- Roman Settlements
- Ionic Temples

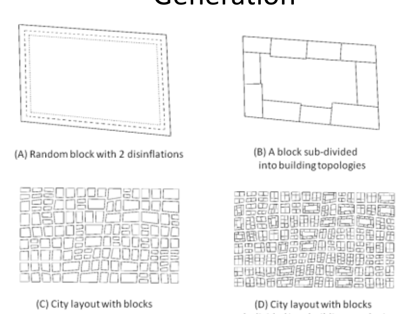
New Classic Buildings of Athens

- Demonstrate how some of the neoclassical buildings in Athens used to be around 100 years ago
  - Based on procedural modeling techniques used for the generation of these characteristic buildings

L-System Road Structures

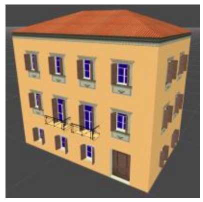


Building Topologies and Pavement Generation

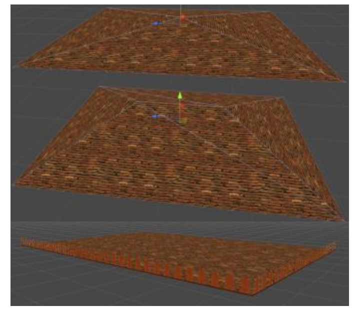


Building Generation

- Buildings were done using boxes
  - Extruding based on the topologies
- Fixed height
  - Multiple floors
  - Variable number of windows



Generation of Building Roofs



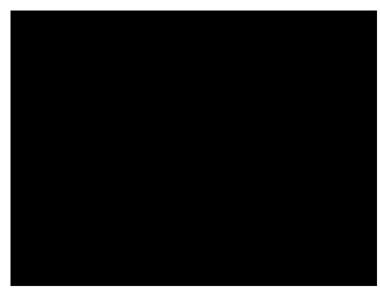
Balconies, Doors and Windows


Balcony


Door


Window

New Classic Buildings Video



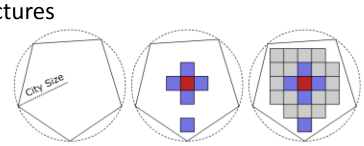
Modelling for Roman Settlements

- Automatic generation of Roman cities
- Grammar for describing Roman settlements derived from the writings of Vitruvius including
  - Location
  - Road structures
  - Important building structures
    - Temples, theatres, baths and civic buildings

Noghani, J., Anderson, E., Liarokapis, F. Towards a Vitruvian Shape Grammar for Procedurally Generating Classical Roman Architecture, Proc. of the 13th International Symposium on Virtual Reality, Archaeology and Cultural Heritage VAST 2012, Short and Project Papers, Eurographics, Brighton, UK, 19-21 November, 41-44, 2012.

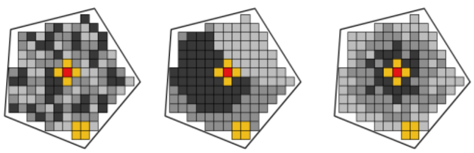
Roman Settlement Planning

- The corners of the city are recorded and the buildable area is marked off
- Next the buildings of significance are added
  - i.e. forum, theatre, and basilica
- The remaining space is filled with generic structures

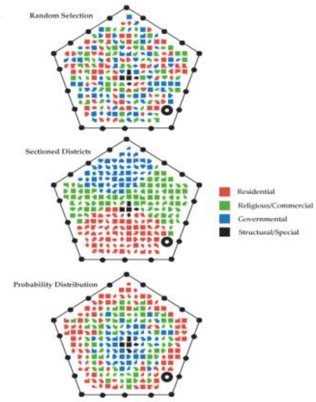


Building Placement Methods

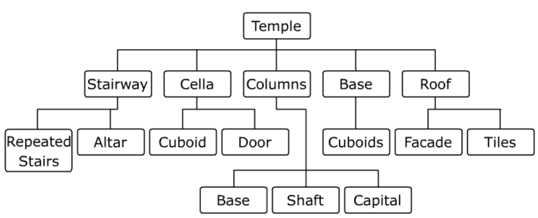
- Method 1: Random Selection
- Method 2: Sectioned Districts
- Method 3: Probability Distribution

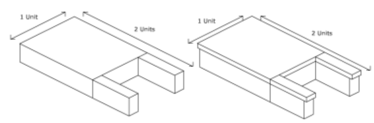


The red square marks the centermost point of the city (i.e. the forum), and the yellow squares indicate a special buildings, such as the coliseum or temple. The light, medium, and dark grey squares represent residential structures, religious structures, and governmental structures respectively

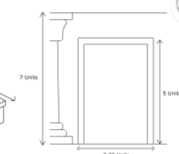


Temple Implementation Example

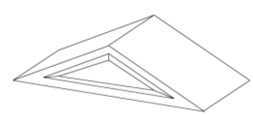




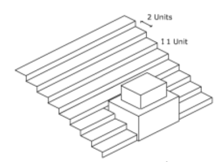
Vitruvius Temple Base Definition



Vitruvius Temple Cella Definition

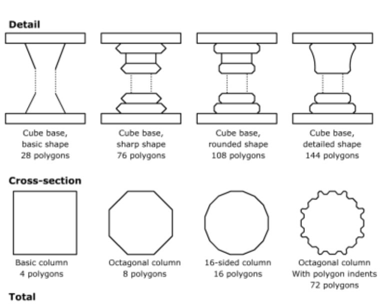


Basic Vitruvius roof shape

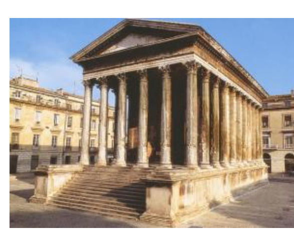


Vitruvius Temple Stairway Definition

Vitruvius Temple Column Definition



Results

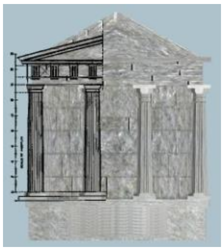


Image

Computer Graphics

Accuracy

- Overlay comparison of Warren's drawings with the rendered output



Amphitheatre Grammar

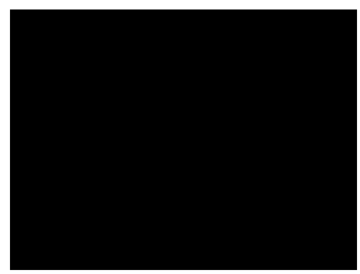
$$\text{Non-Terminals} = \left\{ \begin{matrix} St - Stairway, S - Side, \\ A - Amphitheatre \end{matrix} \right\}$$
$$\text{Terminals} = \{C - Column, Se - Step, Ar - Archway\}$$
$$\text{Start Symbol} = A - Amphitheatre$$
$$\mathbf{A}^{w,h,d}_{x,y,z} \rightarrow \mathbf{S}^{w,h,angle}_{x,y,z}$$
$$angle < 360 \;\; \mathbf{S}^{w,h,angle}_{x,y,z} \rightarrow \mathbf{S}^{w,h,angle+30}_{x,y,z}, \mathbf{Ar}^{w,h,d}_{x,y,z}, \mathbf{C}^{w,h,d}_{x,y,z}, \mathbf{St}^{w,h,d}_{x,y,z}$$
$$\rightarrow \mathbf{Ar}^{w,h,d}_{x,y+h,z}, \mathbf{C}^{w,h,d}_{x,y+h,z}$$
$$\mathbf{St}^{w,h,d}_{x,y,z} \rightarrow \mathbf{Se}^{w,0.2,0.4}_{x,y,z+d} \ldots \mathbf{Se}^{w,0.2,0.4}_{x,y+h,z}$$

Measuring Efficiency

- Test was performed on a Toshiba laptop with a 2.3GHz processor, 6 GB of RAM, and a GeForce graphics card

CitySize	City Radius (Meters)	Number of Buildings	Number of Polygons (triangles)	FPS (Frames Per Second)
100	300	99	87970	208
125	375	104	93610	201
150	450	110	101952	192
175	525	144	138146	185
200	600	197	173660	176
225	675	248	217226	164
250	750	293	257074	150

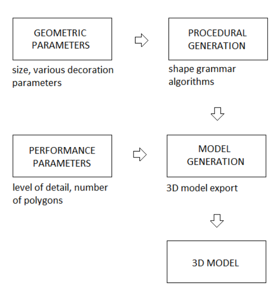
Roman Settlements Video



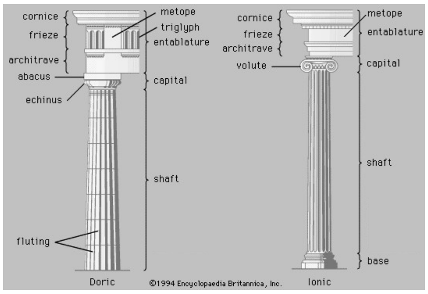
Accurate Modelling of Temples

- Accurate modelling of Roman and Greek temples
  - Focused on Ionic temples
    - Ornaments that usually decorate the column capital have not been taken into account
- Creation of a generic tool
  - For game developers and archaeologists

Architecture

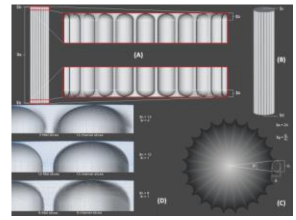


Doric and Ionic Columns



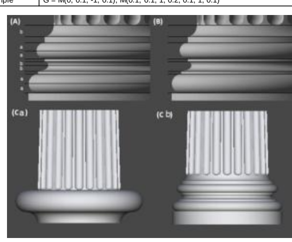
Column Shaft Generation

ID	Name	Description
\$a	Shaft height	Height of the column shaft without the height of channel arcs (Fig. 3 - A)
\$b	Channel arc height	Height of channel arcs at the top and bottom of the columns shaft (both are equal) expressed as a ratio of the height to the radius of a channel (Fig. 3 - A)
\$c	Shaft top radius	Radius of the circle at the top of the column shaft (Fig. 3 - B)
\$d	Shaft bottom radius	Radius of the circle at the bottom of the column shaft (Fig. 3 - B)
\$e	Channels count	Number of channels in the column shaft (Fig. 3 - C)
\$f	Channel angle	Central angle that delimits the diameter of a channel ellipse parallel to a tangent line of the circular cross-section (the ellipse is defined by the channel angle and the channel deep) (Fig. 3 - C)
\$g	Channel deep	Diameter of a channel ellipse perpendicular to a tangent line of the circular cross-section defined by the ratio of the diameter d_1 (perpendicular to the tangent line) to the diameter d_2 (parallel to the tangent line) (Fig. 3 - C)
\$h	Channel slices count	Number of vertical slices (quadrangles) which constitute one channel. This value also defines the number of horizontal rows used in every channel arc. This value controls the number of polygons (although the influence of this parameter on the other parts of the column can be adjusted by some other parameters) (Fig. 3 - D)
\$i	Channel slices per one fillet slice	Number of channel slices needed for one fillet slice. For example, if \$h = 12 and \$i = 4, the number of fillet slices is 3 (12/4 = 3), if \$h=10 and \$i =4, the number of fillet slices is 3 (floor (10/4) = 3). However, the minimum number of fillet slices is 1 (Fig. 3 - D)



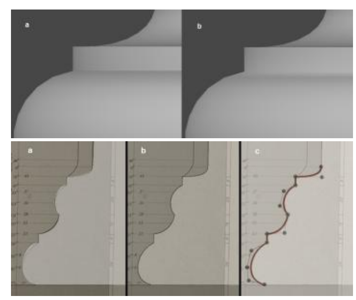
Column Base Generation

	Representation
Parameters	$r \in \mathbb{R}$, $h \in \mathbb{R}$, $c \in \{1,-1\}$
Module	$M(r_1, h_1, c_1, r_2, h_2, c_2, \ldots, r_n, h_n, c_n, r_{n+1})$
Example	G = M(0, 0.1, -1, 0.1), M(0.1, 0.1, 1, 0.2, 0.1, 1, 0.1)



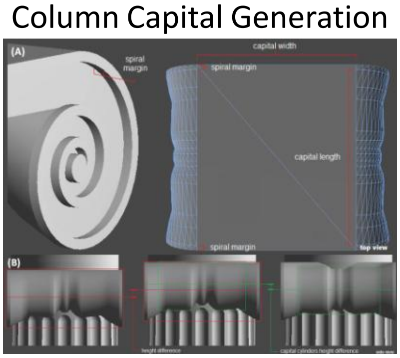
(A) column base subdivided into floors, a is convex and b is concave curvatures (B) column base subdivided into modules (Ca) output using G1 (Cb) output using G2

Using Bézier Curves



A column base constructed using: a) the first b) the second version, figure c) shows the positions of the control points of a Bézier curve

Column Capital Generation



A) Capital general parameters B) Capital height parameters

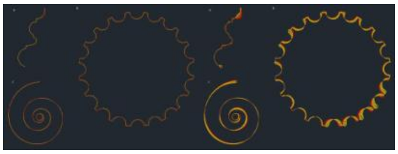
Results

Image

Computer Graphics

Accuracy

	Base	Shaft	Spiral
Number of nodes	97 (original) + 97 (generated)	564 (original) + 564 (generated)	352 (original) + 355 (generated)
Curve length	96 (original) + 96 (generated)	563 (original) + 563 (generated)	351 (original) + 354 (generated)
Fréchet distance	1.7664 (25.23 mm)	2.3741 (33.92mm)	1.2913 (18.45mm)
Hausdorff distance	1.8257 (26.08 mm)	2.3741 (33.92mm)	1.2913 (18.45mm
Average error	0.2797 (4.00 mm)	0.5649 (8.07mm)	0.3355 (4.79mm)
Fréchet distance (%)	1.84%	0.42%	0.37%
Hausdorff distance (%)	1.90%	0.42%	0.37%
Average error (%)	0.29%	0.10%	0.09%



Left Image: Comparison between generated (red) and original (yellow) curves: a) column base, b) column shaft, c) capital spiral, Right Image: Circles with radius equal to node-curve distance between generated (red) and original (yellow) curves: a) column base, b) column shaft, c) capital spiral

Ionic Temples Video

References

- http://www.td-grafik.de/artic/talk20030122/overview.html
- http://en.wikipedia.org/wiki/L-system
- http://davis.wpi.edu/~matt/courses/fractals/trees.html
- http://en.wikipedia.org/wiki/Procedural_modeling
- Garg, S. Procedural Modeling and Constrained Morphing of Leaves, DPhil, University of Singapore, 2012.
- http://prophet.cs.duke.edu/courses/fall01/cps296.3/resources/p351-prusinkiewicz.pdf
- Parish, Y., Müller, P. Procedural Modeling of Cities, SIGGRAPH 2011
- http://ggg.udg.edu/skylineEngine/
- http://code.google.com/p/pixelcity/
- http://www.esri.com/software/cityengine

Useful Links

- http://algorithmicbotany.org/papers/#abop
- http://fractalfoundation.org/
- http://vladlen.info/publications/metropolis-procedural-modeling/
- http://vterrain.org/Plants/Modelling/
- http://www.dpit2.de/navie/index.php?content=plants
- http://forums.odforce.net/topic/16961-procedural-plants-generation-tool/

Questions