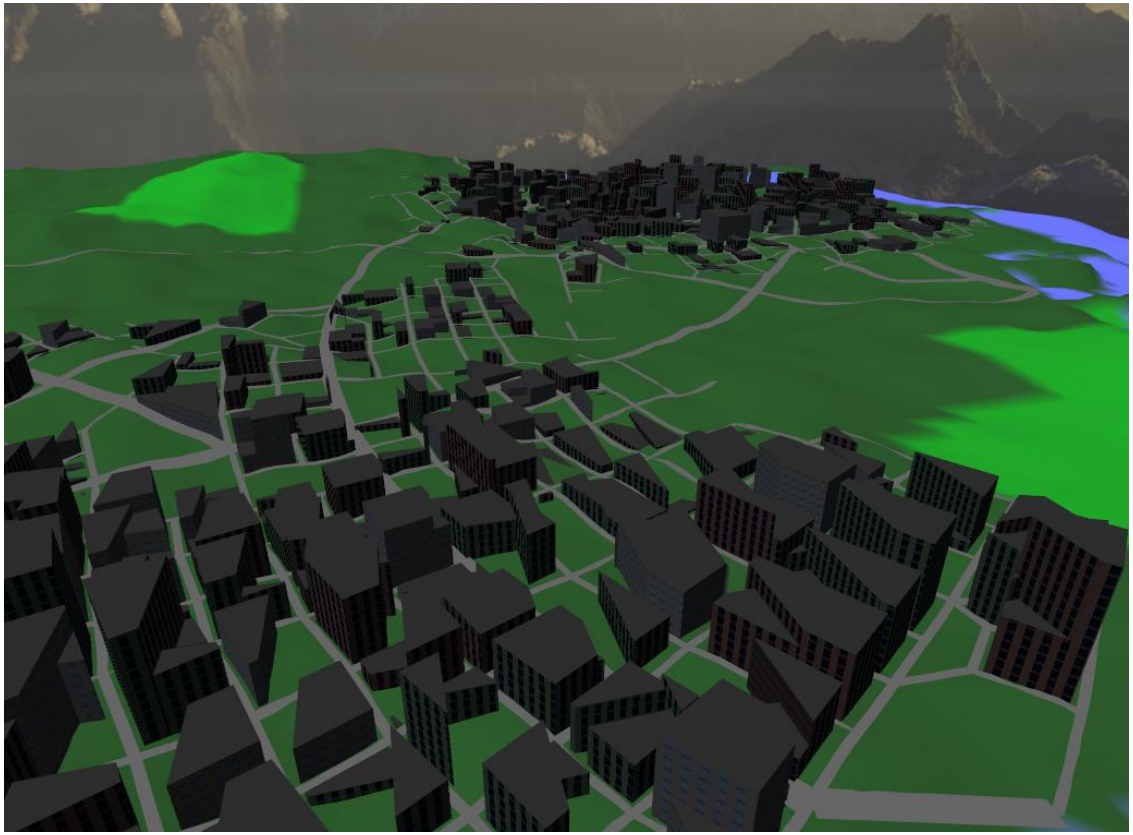
ALGORITHMIC ARCHITECTURE

Procedurally Modelling Cities

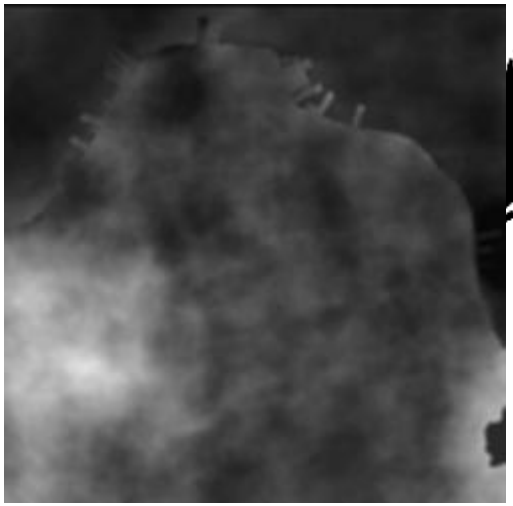
Making a City

- Procedural generation
- Manmade structures
- Natural constraints
- Rendering



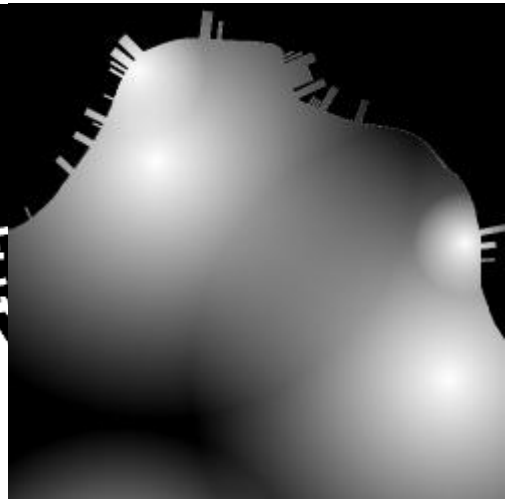
Pipeline (Input)

Start Location/Direction



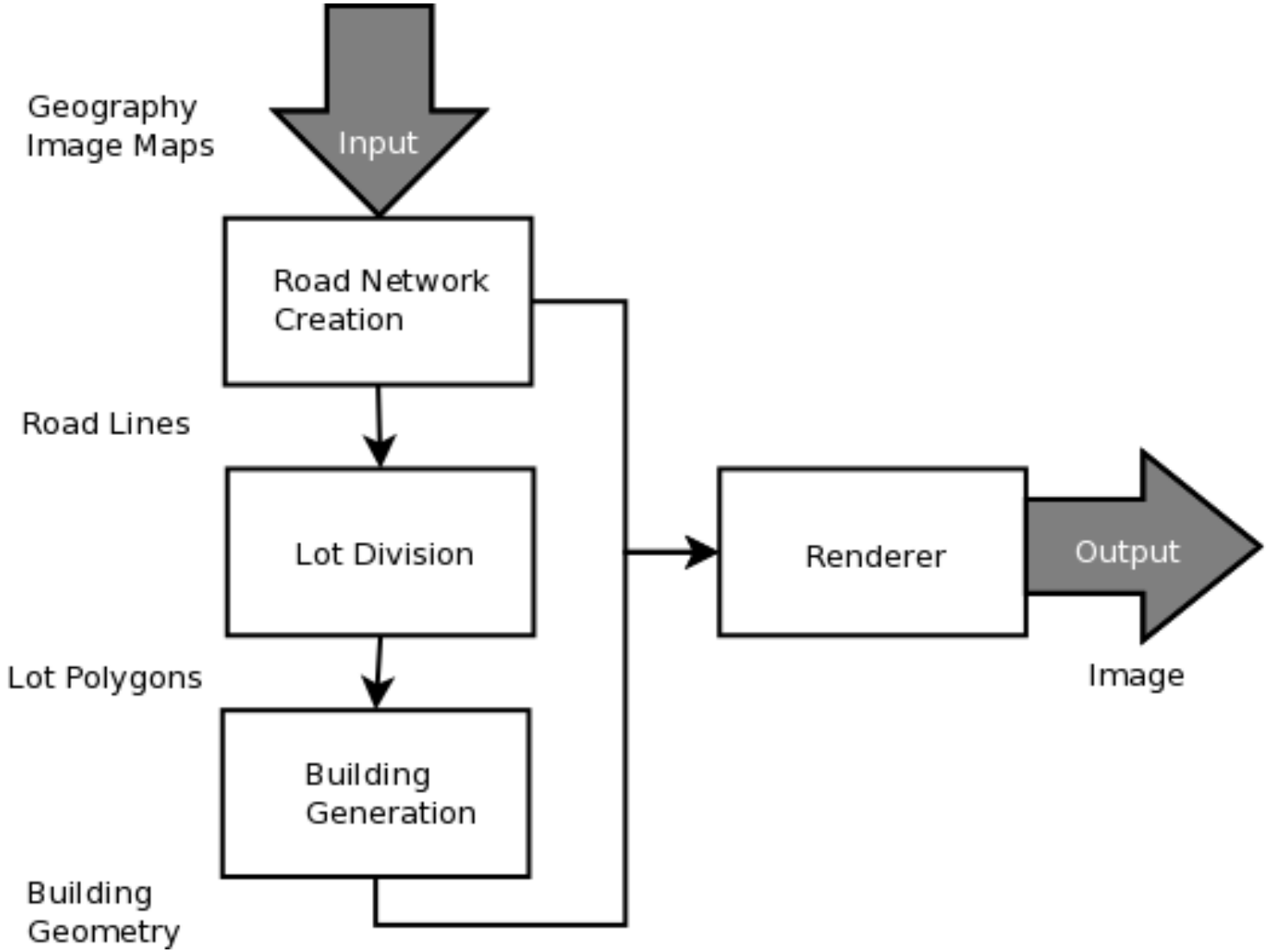
Height Map

Land Map



Population Map

Pipeline (Process)



Road Generation

- Takes input maps and generates roads
- Goals
  - Streets to connect and populate population centers
  - To follow the input's constraints
- Solution: L-Systems

Road Generation

Self-sensitive L-Systems

- Same ideas used for dynamically creating plants

Axiom: A(1)B(3)A(5)

P1: A(x) $\longrightarrow$ A(x+1) : 0.4

P2: A(x) $\longrightarrow$ B(x-1) : 0.6

P3: A(x) < B(y) > A(z) : y < 4 $\longrightarrow$ B(x+z)[A(y)]

A(2)B(6)[A(3)]B(4)

- Extended to allow existing roads to interact with new ones
- Hardcoded into our system through `if` statements

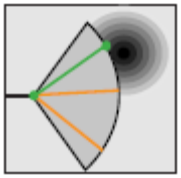
Visual Models of Plants Interacting with Their Environment (1996)

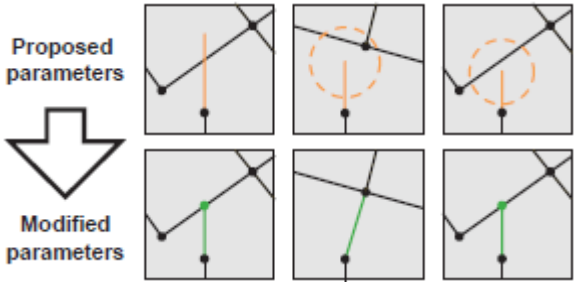
http://algorithmicbotany.org/papers/enviro.sig96.pdf

Road Generation

The paper lacks important details

- Global Goals
  - Population Density
  - *Road Patterns*
- Local Constraints
  - Road Intersections
  - Parks and Water
  - Bridges
  - Height and Slopes





- Approximated through road-end distances

Procedural Road Generation from a game development blog (2010)

http://britonia-game.com/?p=28

Road Generation

Building Generation

- Takes road layout from previous steps
- Goals
  - Buildings which are interesting and varied
  - To stay within the road constraints
- Solution: CGA Shape

Building Generation

CGA Shape

- Set grammar with similar ideas to L-Systems
- Symbol
  - Geometry
  - Numeric Attributes
  - Scope (local coordinate system)

Production Rules

- Predecessor → Successor(s) : Probability
- Take a symbol, perform an operation, then return a result
- Probability

Building Generation

Scope Rules

- Modify the scope of a building
- Translate / Scale / Rotate

Split Rules

- Modify the scope and underlying geometry
- Split geometry along a particular axis to get pieces

Building Generation

Production System

- Axiom: lot (from road system)
- Add axiom to a shape list
- For each shape in shape list
  - Find a matching rule, apply it
  - New shapes replace the old
- Loop until all active shapes are terminals

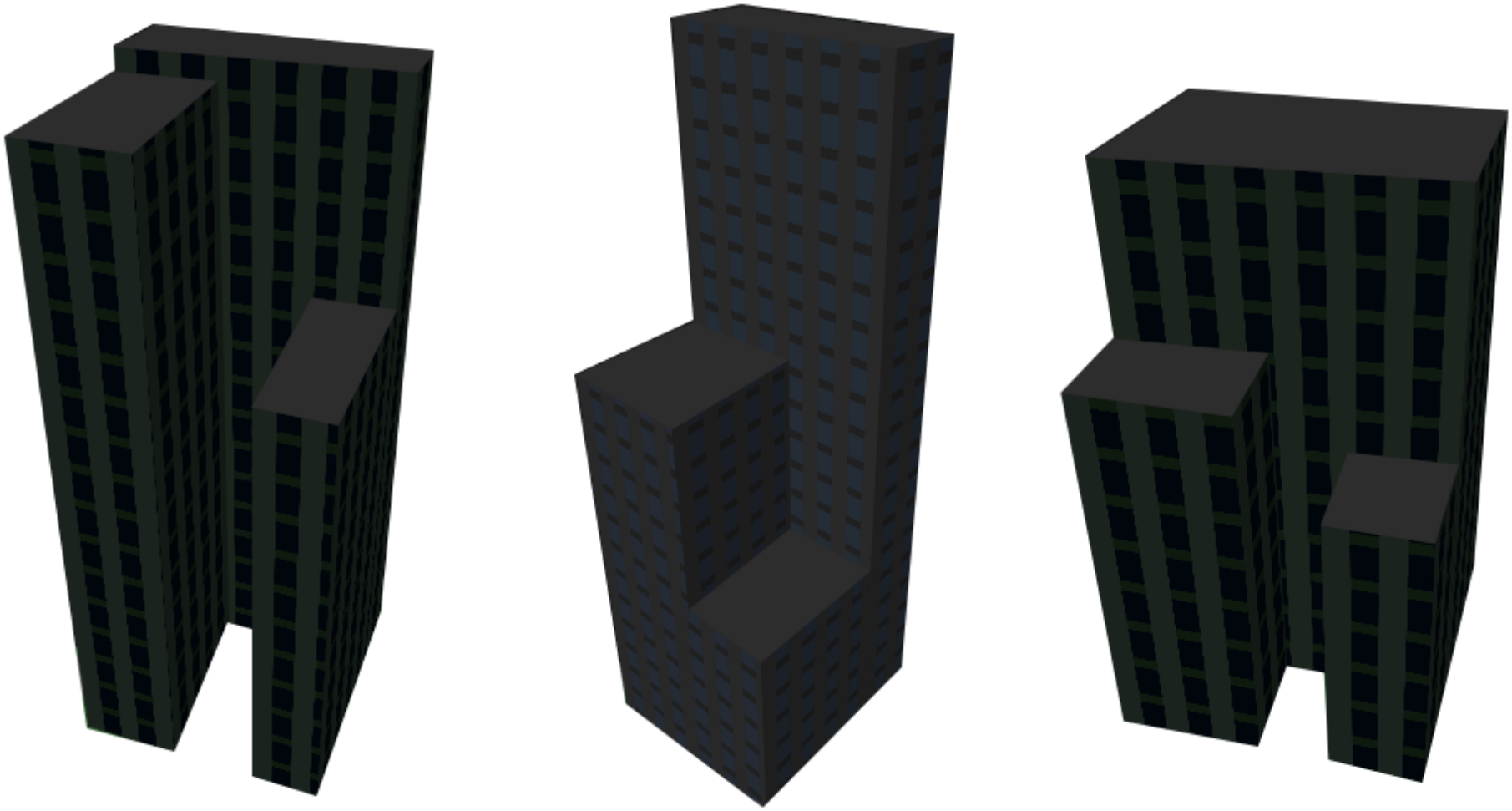
Axiom: Lot

Lot $\rightarrow$ Scale(1r, 70*meters, 1r) { Box }

Box $\rightarrow$ Subdiv(X, 5*meters, 1r) { Wing | main }

Wing $\rightarrow$ Scale(1r, 0.5r, 1r) { main }

Building Generation

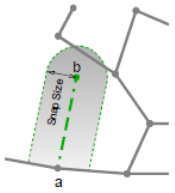


Rendering

- Quads
  - Highways, roads, bridges
- Terrain Projection
- Buildings
  - Textured quads

Problems and Enhancements

- Roads
  - Projection rendering (rearranging vertices around roads)
  - L-Systems (undirected graphs)
  - Intersection approximation (“thick” line intersections)
  - Road Details (stay level, use curves to approximate)



- Buildings
  - Facade detail
  - Geometry instances
- Application
  - User control
  - Saving and loading

Making a City