Interactive Design of Urban Spaces using Geometrical and Behavioral Modeling

Carlos Vanegas¹  Daniel Aliaga¹
Bedřich Beneš¹  Paul Waddell²

¹Purdue University, USA
²University of California, Berkeley, USA
Objective and Applications

Objective
• Fast generation of 3D urban models that reflect the behavior of real-world cities

Applications
• Urban planning
• Urban visualization
  • Policy evaluation and education
• Content generation for games and movies
Examples (1)

- Designer increases height of buildings in downtown
- Number of jobs increases
- Population increases
- New housing appears in accessible areas
Examples (2)

Designer inserts a highway into the model → Accessibility increases in formerly remote areas → Population redistributes → Location of jobs, houses, and buildings changes
Observation

Behavioral Modeling + Geometrical Modeling = Reduce design time
Produce plausible urban models
System Overview

Geometrical Modeling
- Roads
- Blocks
- Parcels
- Buildings

3D Model

User

Behavioral Modeling
- Population
- Jobs
- Accessibility
- Land Value
**System Overview**

- **Input**: Interactive design interface to change and constrain values of simulation variables

**User**

**Geometrical Modeling**
- Roads
- Blocks
- Parcels
- Buildings

**Behavioral Modeling**
- Population
- Jobs
- Accessibility
- Land Value

**3D Model**
System Overview

**Process**: Interactions between the variables are continually simulated to bring the system back into equilibrium.

- **Geometrical Modeling**
  - Roads
  - Blocks
  - Parcels
  - Buildings

- **Behavioral Modeling**
  - Population
  - Jobs
  - Accessibility
  - Land Value

3D Model
**Output**: Procedurally generated 3D urban model (based on the state of the system)
Related Work

**Geometrical Modeling**
- Procedural Modeling of Cities, Buildings, Facades, Streets
  - Müller et al., 2001, 2006
  - Wonka et al., 2003
  - Chen et al., 2008
  - Aliaga et al., 2008

**Behavioral Modeling**
Related Work

in Graphics

- Large crowd modeling
  - Sung et al., 2004
  - Treuille et al., 2006
- Flocking and animal behavior modeling
  - Reynolds, 1987
  - Tu and Terzopoulos, 1994
Related Work

**Geometrical Modeling**

**Behavioral Modeling**

**in Urban Simulation**

- Cellular automata, Agent-based
- Microsimulation models
  - Discrete-choice models
  - Agents make decisions to locate and move within urban space
  - UrbanSim (Waddell et al., 2002)
Related Work

Integration

- Interactive geometric simulation of 4D cities
  - Weber et al., 2009
- Visualization of Simulated Cities
  - Vanegas et al., 2009

Do not provide a simulation-assisted urban design system
Contents

- Introduction
- Urban Design as a Dynamical System
- Behavioral Modeling
- Geometrical Modeling
- Results
Urban Design as a Dynamical System

**System**
- Consists of $N$ variables defined over a spatial domain
- Each variable sampled over a 2D spatial grid $G$ of size $W \times H$
- $v_k(i,j)$ denotes the value of $k$-th variable at grid cell $(i,j)$

$\begin{align*}
  v_0(i,j), v_1(i,j), \ldots, v_N(i,j)
\end{align*}$
Urban Design as a Dynamical System

- Minimal set of design variables used in our system
  - Population count
  - Job count
  - Accessibility
  - Land value
  - Road length
  - Building volume
  - Average tortuosity
  - Terrain elevation
Urban Design as a Dynamical System

- **Variable modeling**
  - The change in each variable \( v_k(i,j) \) is represented as a differential equation
  
  \[
  \dot{v}_k(i,j) = f_k(v_1, v_2, \ldots v_N)
  \]

  - If the user changes a variable, the system iteratively updates all other variables in order to return to a state of **equilibrium**
  
  \[
  |\dot{v}_k(i,j)| \leq \varepsilon
  \]
Urban Design as a Dynamical System

- **Iterative System** (a classical formulation)

\[ v_{k+1}^{n+1}(i, j) = v_k^n(i, j) + \dot{v}_k(i, j) \]

In urban design, the term \( \dot{v}_k(i, j) \) is difficult to express symbolically due to widespread dependencies.

Instead, we propose algorithms for computing \( v_k(i, j) \) or \( \dot{v}_k(i, j) \).
Actual dependencies between variables are:

The system contains algorithms for modeling variables:
Urban Design as a Dynamical System

- **Variable Modeling:** The system considers these dependencies and for each variable:
  - Simulates its change $\dot{v}_k(i,j)$ as a function of other variables (behavioral variables)
  - Calculates its target value $v_k(i,j)$ as function of other variables, and procedurally generates the geometry that matches the target values (geometrical variables)
Urban Design as a Dynamical System
System Overview

- Geometrical Modeling
  - Roads
  - Blocks
  - Parcels
  - Buildings

- Behavioral Modeling
  - Population
  - Jobs
  - Accessibility
  - Land Value

3D Model
Behavioral Modeling

• Roads
• Blocks
• Parcels
• Buildings

Geometrical Modeling

3D Model

User

• Population
• Jobs
• Accessibility
• Land Value

Behavioral Modeling
Behavioral Modeling

- **Variables**
  - Population count
  - Job count
  - Accessibility
  - Land value
Behavioral Modeling

• Population and Jobs count
  • Operations:
    • Remove a fraction of population/jobs from their current location (**Mobility** algorithm)
    • Locate population/jobs to predicted locations (**Location choice** algorithm)

• Each unit of population/jobs is referred to as an **agent**
Behavioral Modeling

• Overview
  • Variables are updated using major components of an urban simulation
  • Major components(*): algorithms which are applied to a set of variables representing population and jobs:
    • Transition
    • Real-estate developments
    • Mobility
    • Location choice

(*) From Waddell and Ulfarsson (2004)
Behavioral Modeling

- **Major components**
  - Transition
  - Real-estate developments
  - Mobility
  - Location choice
Behavioral Modeling

- **Major components**
  - Transition
    - Adds/removes population and jobs based on exogenous macroeconomic model
  - Real-estate developments
    - Changes land-use due to markets and regulations

- *These are user-controlled components*
Behavioral Modeling

- **Major components**
  - Transition
  - Real-estate developments
  - Mobility
  - Location choice
Behavioral Modeling

- **Major components**
  - Mobility
    - Moves a *fraction* of population/jobs from their current location
  - Location choice
    - Predicts the *locations* where population/jobs will move to

- *Primary way for dynamical system to alter its state in order to reach equilibrium*
**Behavioral Modeling**

- **Operation: Remove agents** (Mobility Algorithm)
  - Agents de-located from grid are moved to queue $Q$
  - Agents added by user are moved to queue $Q$

\[
S = \alpha E
\]
\[
E = \sum_i \sum_j e(i, j)
\]

User adds population/jobs
Behavioral Modeling

- **Operation**: Locate agents (Location Choice Algorithm)
  - Agents from queue are placed back in the grid
Behavioral Modeling

• **Operation: Locate agents** (Location Choice Algorithm)
  - Uses weighted attractiveness measure as a probability
  - The probability $q_{st}$ that an agent $e_s$ will locate at the grid cell $(i_t, j_t)$ is given by

$$q_{st} = \frac{w_a a_t + w_b l_t}{T_s}$$

$a_t$: **accessibility** at the grid cell
$l_t$: **land value** at the grid cell
$T_s$: total count of the agent throughout the entire grid
Behavioral Modeling

- **Accessibility**
  - Measure of access that a grid cell has to jobs and to the rest of the population
  - Intuitively
    - Decreases with increasing terrain slope
    - Increases with higher road connectivity and nearby population/jobs
Behavioral Modeling

• **Accessibility**
  - Represented by logistic function
    \[ a(i, j) = \frac{1}{1 + e^{-z(i,j)}} \]
    \[ z(i, j) = \beta_0 + \beta_1 u_1(i, j) + \beta_2 u_2(i, j) + \beta_3 u_3(i, j) \]
  - \( u_1 \): proximity to highways, arterials and streets
  - \( u_2 \): local slope of the terrain
  - \( u_3 \): Distance-normalized measure of activity level at \((i,j)\)
    \[ u_3(i, j) = \sum_r \frac{D_{ijr}}{d_{ij,i_rj_r}} \]
Behavioral Modeling

- Example accessibility distribution
Geometrical Modeling

- Roads
- Blocks
- Parcels
- Buildings

Geometrical Modeling

User

Behavioral Modeling
- Population
- Jobs
- Accessibility
- Land Value

3D Model
Geometrical Modeling

• **Overview**
  • We define a set of geometric variables for which relations with behavioral variables can be established
  • Variables are stored in grid-cells, but actual geometry can cross cell boundaries
Geometrical Modeling

- **Variables**
  - Road length
  - Average tortuosity
  - Building volume
  - Terrain elevation at grid cell (user-controlled)
Geometrical Modeling

- **Road length**
  - Total length of roads in grid cell
  - Two types: Arterials and Streets
  \[
  r^{n+1}(i, j) = \min \left( w_{pr} p^n(i, j) + w_{br} b^n(i, j), r_{max} \right)
  \]

- **Tortuosity**
  - Ratio between road segment length and distance between segment endpoints
  \[
  \tau^{n+1}(i, j) = 1 + k \left( 1 - \frac{p^n(t, j) + b^n(t, j)}{p_{max} + b_{max}} \right)
  \]

- **Building volume**
  - Total volume of all the buildings in grid cell
  - Computed as a function of population and jobs
  \[
  m^{n+1}(i, j) = w_{pm} p^n(i, j) + w_{bm} b^n(i, j)
  \]
Geometrical Modeling

- Arterials and Streets: Seeds
  - To connect the main population clusters a set of seeds is generated considering the population/jobs distribution and the location of highways
Geometrical Modeling

- **Arterials and Streets: Expansion of Arterials**
  - Each seed is used as an intersection of the arterial roads network and used to generate arterial segments.
Geometrical Modeling

- **Arterials and Streets: Expansion of Streets**
  - Street seeds are generated along arterial road segments and used to create streets
**Arterials and Streets: Observations and Assumptions**
- Road networks exhibit a variety of styles which cannot be inferred from socioeconomic parameters.
- The predominant pattern of arterials and streets are grid style and radial style.
Geometrical Modeling

- Arterials and Streets: Seed Generation Algorithm
- Text
Geometrical Modeling

- **Arterials and Streets: Expansion Algorithm**
  - Using the seeds positioned according to population/jobs and/or along arterials, we generate road segments using a breadth-first expansion method.
  - These seeds are placed into a queue and sequentially processed.
Geometrical Modeling

Grid

Radial

Tortuosity
Geometrical Modeling

• **Parcels**
  - Blocks are extracted from the road network and partitioned into parcels
  - The number of parcels in the block is proportional to the product of the area of the block and the count of population/jobs in the grid cells inside the block
Geometrical Modeling

- Parcels
Geometrical Modeling

- **Buildings**
  - Procedurally generated inside each parcel based on the socioeconomic information of the area
  - Process:
    - Calculate geometry of the building footprint
    - Calculate building height
    - Use procedural rules to generate 3D geometry that matches these attributes
Results

Geometrical Modeling
- Roads
- Blocks
- Parcels
- Buildings

Behavioral Modeling
- Population
- Jobs
- Accessibility
- Land Value

3D Model

User
Results: System Specs

- **Dimensions:**
  - Grid-cell size 0.1x0.1 km
  - Grid size up to 50x50 km in our experiments
- **Time:**
  - Update time step during editing:
    - <0.3 seconds for small grid (~5x5 km)
    - <4 seconds for large grids (~15x15 km)
  - **Total Design Time:**
    - <5 minutes for any of our examples
Results: Terrain Editing

Overhead view

Initial  Lake  Mountains

Flyover view
Results: Completion and Validation (1)

- Town center
- Population
- Parcels
- Parks
- Jobs
- Buildings

Terrain

Input
Results: Completion and Validation (2)

Real City

Synthetic City
Conclusions

• We have presented an interactive system to design and edit 3D urban models
• Our key inspiration is to close the loop between behavioral modeling and geometrical modeling producing a single dynamical system that assists a designer in creating urban models
Conclusions

• Limitations
  • Stochastic component that does not allow behaviors and geometries to be exactly repeatable
  • Over-constraining the system can lead to unfeasible urban models
Future Work

• Including additional elements of behavioral modeling, such as a more sophisticated accessibility model
• We are seeking methods to generate more complex geometric structures using socioeconomic data such as generating additional building details
Acknowledgements

- Purdue Research Foundation for partially funding this work
- Aaron Link for his help with exporting models
- Students from the Computer Graphics and Visualization Lab at Purdue for their feedback
- Reviewers for their valuable comments and suggestions
Questions?

area 225 km², created in less than four minutes, real-time render
Genuinity Signatures:
Designing Signatures for Verifying 3D Object Genuinity

Daniel G. Aliaga       Mikhail Atallah
Department of Computer Science

http://www.cs.purdue.edu/cgvlab

Purdue University, USA
Project Goal

- To provide algorithms for placing into a digital 3D object information that enables determining genuinity of the object after its automated manufacturing
Observation #1

• Continually more objects and parts are being digitally designed and digitally manufactured

  – Digital designing: enables simulation and building more complex parts

  – Digital manufacturing: facilitates easier/faster/better manufacturing of complex parts

  – Digital design+manufacturing: can be a significant investment of time
Observation #2

- Many imitations and “knock-offs” are floating around the market
  - Using parts/replicas of unknown specifications may lead to catastrophic failures
    - e.g., in an engine, medical instruments
  - Impact of the counterfeiting industry is negative and large
    - e.g., in 1997, was already US $12B to global automotive industry
Observation #3

• Simple visual inspection is not sufficiently robust
  
  – Cannot easily verify material composition, density, and product quality

  – Since design and manufacturing is digital, it could be that adversaries have equal or better digital technology, thus it is unknown if the counterfeits are “inferior” in quality
Challenge

• To encode information into a physical object that cannot be reproduced by an adversary but do so without

1. depending on the legitimate manufacturer having technology superior to the adversary
   ... else an adversary with equal or better technology can make counterfeit copies unbeknownst to the genuine manufacturer

2. depending on security-through-obscurity
   ...else once the method is compromised, security is lost
Our Approach

- Designed 3D Model
- Manufactured Object
- Designed Signature
- Acquired Fragment
- Manufactured Signature

Does acquired = designed?
Example: Motivation
Presentation Overview

• Introduction
• Previous Works
• Genuinity Testing
• Signature Creation and Encoding
• Signature Acquisition and Verification
• Results
• Conclusions and Future Work
Presentation Overview

- Introduction
- Previous Works
- Genuinity Testing
- Signature Creation and Encoding
- Signature Acquisition and Verification
- Results
- Conclusions and Future Work
Previous Works

• Physical watermarking: uses serial numbers, etching, and security-through-obscurity
• Digital Watermarking
• Specialized solutions for paper documents
Presentation Overview

- Introduction
- Previous Works
- **Genuinity Testing**
  - Signature Creation and Encoding
  - Signature Acquisition and Verification
  - Results
- Conclusions and Future Work
Genuinity Testing

1. Design 3D Object Model
2. Design Signature
3. Manufacture Automatically
4. Acquire 3D Fragment
5. Verify Signature
6. Master or Partial Key

Master Key
Genuinity Testing

Design 3D Object Model → Design Signature

Manufacture Automatically

Verify Signature

Acquire 3D Fragment

Master or Partial Key

Master Key
Genuinity Testing

- Design 3D Object Model
- Design Signature
- Manufacture Automatically
- Verify Signature
- Acquire 3D Fragment
- Master Key
- Master or Partial Key
Genuinity Testing

1. Design 3D Object Model
2. Design Signature
3. Manufacture Automatically
4. Verify Signature
5. Acquire 3D Fragment

- Master Key
- Master or Partial Key
Counterfeiting Attacks

• **Re-instancing**
  
  = What happens if the digital model leaks out?
  
  (adversary makes new objects using the digital model)

• **Replication**
  
  = What happens if the adversary just makes the best copies possible of the object, signature, and all?
  
  (making copies by digitally acquiring and then digitally re-manufacturing the object)
Signature Embedding

• Genuinity signature must be encoded somewhere
• Approach is to alter some feature of the object
  – Global features
    • Pro: robust
    • Con: requires acquiring the entire object
  – Local features
    • Pro: only a partial acquisition is needed
    • Con: less robust (e.g., can remove signature)

• We choose local features
  – Only need a partial acquisition for verification
  – If signature is removed, then just reject object
The method used is assumed to be imperfect and modeled by an unbiased normal error distribution.

- Genuine Creator: error is $m_g$ and $v_g$
- Adversary: error is $m_a = (1/\beta)m_g$ and $v_a = (1/\beta)v_g$
  - e.g., adversary has technology that is “$\beta$ times better”

- Note: our method does not depend on the absolute values
  - i.e., $\beta$ matters
Presentation Overview

- Introduction
- Previous Works
- Genuinity Testing
- Signature Creation and Encoding
  - Signature Acquisition and Verification
- Results
- Conclusions and Future Work
Signature Creation and Encoding

Algorithm:

1. Pick subset of object’s surface for signature footprint
2. Create a “signature image”
3. Map the signature image onto the object (= do the displacement map operation)
4. Manufacture the resulting 3D object model
Signature Footprint

- User interactively chooses the subset of the model where the signature should be placed
Signature Image

- Signature image is composed of a 2D array of samples
- Samples are organized in a two-level stratification
  - Spatial Stratum
    - $A_j = \{s_{i1j}, s_{i2j}, s_{i3j}, ... \}$ (spatially near)
    - Reduces need to precisely align synthetic and real samples
    - Makes it easier to detect re-instancing (and replication)
  - Variance Stratum
    - $B_k = \{A_{j1k}, A_{j2k}, A_{j3k}, ... \}$ (spatially disjoint)
    - Makes verification more sensitive to change in variance
    - Improves detecting replication attacks
Signature Image

Footprint  Displacements-only  Full Signature
Signature Image

Footprint

Displacements-only

Full Signature
Signature Image

Signature with a balance of $m$ and $r$

magnitude $m$

variance $r$
Presentation Overview

• Introduction
• Previous Works
• Genuinity Testing
• Signature Creation and Encoding
  ➢ Signature Acquisition and Verification
• Results
• Conclusions and Future Work
Signature Acquisition and Verification

Algorithm:

1. Acquire surface fragment containing signature footprint on real object
2. Align captured surface fragment with digital model of signature
3. Verify captured signature is from genuine object
Signature Acquisition

• Requirements
  – To capture subtle displacements, need high-resolution
  – To facilitate easy deployment, need self-calibration (and relatively fast capture speed)

• Observation
  – Macro structure of the area to capture is usually simple (e.g., planar or low-curvature surface with encoded signature)

• Approach
  – Capture the low-frequency geometric structure and augment high-frequency detail from photometric data
    = Photogeometric Acquisition
Signature Acquisition

- A self-calibrating system of one or more cameras and three or more projectors to capture high resolution (and multi-viewpoint) models at camera resolution
  - Photogeometric Structured Light, Aliaga and Xu, CVPR 2008

* = first coined by Lu and Little, IJCV 1999
Photogeometric Structured Light

acquired signature
Signature Verification

- Use the aforementioned sample stratification and statistical testing to determine an expected homogeneity of variance
  - similarity of means testing not enough for replication
  - use group and median-based homogeneity of variance testing
Statistical Testing

- We desire
  - 1) variance of all samples in a group to be the same
  - 2) variance of each group to be $v_k + (m_g + v_g)$

- Thus, use a modified version of a Brown-Forsythe test which checks for homogeneity of variance
  - Simple F-test is too fragile
  - Medians, as opposed to means, is more robust to error distributions straying from perfect normalcy
  - Once we account for the changing sample displacement and variance, only the variance $(m_g + v_g)$ should be present in each of the many groups
  - The modification is that we know the desired global variance
Presentation Overview

• Introduction
• Previous Works
• Genuinity Testing
• Signature Creation and Encoding
• Signature Acquisition and Verification

➤ Results
• Conclusions and Future Work
Results

• We used our approach to design, manufacture, verify several CAD-type objects (both in simulation and in real-world)

• Our prototype uses
  – Canon Rebel XTi 10 MP camera
  – Optoma EP910 DLP projectors of 1400x1050 pixels
  – Stereolithography manufacturing
    • max precision = $5/1000^{th}$ of an inch (~0.1mm)
Testing: Genuine Object

\[ \tau = 1; \ m_g = v_g = 0.5; \ m = 0.25; \ r = 0.0025 \] (operational tolerance never exceeded)
Testing: Re-instanced Object

\[ \tau = 1; \, m_g = v_g = 0.5; \, m = 0.25; \, r = 0.0025 \] (operational tolerance never exceeded)
Testing: Replicated $\beta=1,10,100$

- Need more samples to support larger $\beta$ (e.g., bigger signature image)
Real Objects

• Our approach is designed for fragility upon copying (i.e., success is hard), thus our real-world experiments test accurately recovering the signature at the expected levels of error.

Rendering of 3D model  Photograph of real-object  Acquired 3D fragment
Real Objects

- Object measures about 10x10x5 cms
- Manufacturing accuracy is ~0.1mm
- Verification system accuracy is ~0.5mm
- Genuine object passes test
- Re-instanced and replicated object rejected with only a 5% chance of getting the classification wrong
Real Objects

Difference of designed to genuine 
(W=1.16)

Difference of designed to replicated 
(W=2.86)
Real Objects

• **Average test statistics (in simulation) are**
  - Genuine = 1.23
  - Re-instanced = 2.75
  - Replicated ($\beta=10$) = 1.63
Real Objects

Difference of designed to genuine
(W=1.22)

Difference of designed to replicated
(W=2.43)
Works with the “bunny” too...

3D Model without Signature 3D Model with Signature
Presentation Overview

- Introduction
- Previous Works
- Genuinity Testing
- Signature Creation and Encoding
- Signature Acquisition and Verification
- Results

➢ Conclusions and Future Work
Conclusions

• We have presented a novel approach to encode data for detecting object genuinity into the surface of physical objects
  – Re-instancing and replication is handled (replication is harder)
  – Support $\beta > 1$ (e.g., $\beta=100$)

• Our problem is quite different from digital watermarking where perfect copies can be made

• Our work provides a unique blend of computer graphics+vision, information security, and advanced manufacturing
Limitations

• Our approach requires “space” to place the signature

• Our technique requires knowing the variance of the manufacturing and verification processes
Acknowledgements

- Purdue Dept. of Mechanical Technology
- Purdue CGVLAB
- Acquisition methods supported by NSF CCF 0434398
Questions?