Computer-Generated Residential Building Layouts

Paul Merrell
Eric Schkufza
Vladlen Koltun

Stanford University
Modeling Buildings with Interiors

- Goal: Model the internal structure of buildings
- Crucial in many interactive applications
  - Buildings that can be entered and explored
- Commonly created by hand
Residential Buildings

- Focus on residential buildings
  - Common in games, virtual worlds
  - Have complex structure

- Office buildings and schools
  - Highly regular layouts
Related Work

- Automated Spatial Allocation
  - March and Steadman, 1971
  - Shaviv, 1987

- Physically Based Modeling
  - Arvin and House, 2002
  - Mass-spring system
  - Sensitive to initial conditions

- VLSI Layout
  - Sarrafzadeh and Lee, 1993
Computer Graphics Research

Müller et al., 2006

Whiting et al., 2009

Legakis et al., 2006

Pottmann et al., 2007
Architectural Design in the Real World

Client’s high-level specifications
- Number of bedrooms
- Bathrooms
- Total square footage, etc.

Architectural program
Rooms & adjacencies

Set of floor plans

Exterior style
Overview

First end-to-end approach to automated generation of building layouts from high-level requirements
Possible Approaches to Building Layout Design

- Use a grammar
  - Shape grammar [Stiny, 2006]
  - Hard to capture the functional relationships
- Use guidelines from architects
  - Too many rules of thumb, ill-specified
- Use a data-driven approach
  - Infer design principles using machine learning techniques
Data-Driven Architectural Programming

- Sample from a distribution of architectural programs
- Conditioned on the high-level contraints
Bayesian Network

- Represent the distribution in a Bayesian network
  - Compact representation
- Nodes – probabilities
- Edges – conditional dependencies
- Sample from conditional distributions
  - Use high level specifications
Structure Learning Results

Architectural programs

Output one sample
Overview

Client’s high-level specifications

Architectural program
Rooms & adjacencies

Set of floor plans

3D model
Floor Plan Optimization

- Metropolis algorithm
  - Propose a new floor plan
  - Evaluate it, then accept or reject it
  - Not a greedy algorithm
Metropolis Algorithm

- Objective function
  \[ f(x) = \exp(-\beta C(x)) \]
  \begin{align*}
  x & \quad \text{Building layout} \\
  \beta & \quad \text{Constant} \\
  C(x) & \quad \text{Cost function}
  \end{align*}

- In each iteration, propose a new building layout \( x^* \)
- Accept with probability
  \[ \alpha(x^*|x) = \min \left( 1, \frac{f(x^*)}{f(x)} \right) \]
Proposal Moves

- Slide a wall

\[ d \sim \mathcal{N}(0, \sigma^2) \]

- Snap walls together

- Slide the entire wall

- Split into two collinear walls
Proposal Moves

- Swap two rooms

- Helps to explore the space more rapidly
The Cost Function

- Evaluates the quality of the layout

\[ C(x) = k_a C_a(x) + k_d C_d(x) + k_f C_f(x) + k_s C_s(x) \]

- Accessibility term
- Dimension term
- Floor compatibility term
- Shape term
Accessibility Term

- Architectural program specifies adjacencies
- Outdoor access for entrances, patios, and garage.
Dimension Term

- Likelihood of a room’s area and aspect ratio
  - Uses Bayesian network

\[ C_d(x) = -\sum_{i=1}^{n} \left( l^i_a(x) + l^i_{as}(x) \right) \]
Shape Term

- Measure concavity of a shape, $S$
Shape Term

Shape term excluded
Cost Function

- All terms included
Floor Compatibility Term

- Each floor should be supported by the floor below it
Floor Plan Optimization

- 200 iterations
- 2,000 iterations
- 20,000 iterations
- 100,000 iterations
Overview

Client’s high-level specifications

Architectural program
Rooms & adjacencies

Set of floor plans

3D model
Different Exterior Styles

Cottage

Italianate

Tudor

Craftsman
Results
Results
Results
Results
Results
Future Directions

- Non-rectilinear / curved wall segments
- Site-specific and client-specific factors
- Integrate structural stability
- Interactive exploration of layout designs
- Other building types
Conclusion

- First end-to-end approach to automated generation of building layouts from high-level requirements
- Data-driven approach to procedural modeling
Questions?