A Connection between Partial Symmetry and Inverse Procedural Modeling
Overview

- Paper from Siggraph 2010
- Authors
  - Martin Bokeloh (MPI Informatik)
  - Michael Wand (Saarland University and MPI Informatik)
  - Hans-Peter Seidel (MPI Informatik)
- Given an example model, can we derive rules for producing models similar to the example?
Related work - texture and geometry synthesis

- Approach motivated by non-parametric texture synthesis
- Merrell 2007
  - Building blocks required as input
  - Regular grid structures
- Merrell and Manocha 2008
  - Increased design space, but limited to low complexity input examples
- Cabral et al. 2009
  - User specified building blocks and a connectivity graph
Related work - shape grammars and inv. procedural modeling

- Hart and Flynn 1997
  - Deriving rules from 2D example graphics
  - Limited to L-systems
- Mitra and Pauly 2008
  - Similar to this paper
  - Looks for symmetric parts that can be replicated rather than symmetric areas for attachment
- Modeling by example (Kraevoy et al. 2007 and other papers)
  - Large design space
  - Requires user guidance to assemble building blocks
Approach - Defining similarity

- Given an example model, can we derive rules for producing models similar to the example?
- R-similarity: Every point of $S_2$ locally similar to $S_1$ within radius $r$
Approach - Defining symmetry

- r-symmetry: Like r-similarity, but under a fixed transformation $T$
- Rigid motions

(a) $r$-similarity

(b) $r$-symmetry
the effect of parameter $r$
Approach - Key idea
Approach - Key idea

- *r*-similarity

![Diagram showing small and large *r* similarity](image-url)
Approach - Detecting docking sites

- Red and blue are symmetric
- Cuts are done at the boundaries of symmetric geometry
Approach - Detecting docking sites

- Not all symmetric cuts are valid
- A strategy is needed
Approach - Detecting docking sites

- Identify non-symmetric areas
- Build a graph of their possible connections
- Find elementary docking sites
Approach - Grammar extraction

- Context-free grammar
- Dockers are terminals
- Docking sites are non-terminals
Approach - Regular grids

- Specialized rules to obtain tileable grids
- 1-grids, 2-grids

(a) configuration for a grid patterns

(b) no valid grid
Approach - Regular grids

(a) sofas (1-grid)

(b) spiral stairs (1-grid)

(g) parking structure (2-grid)  
(from the Dosch Design shape collection)
Implementation

- Accepts triangle meshes
- Stores symmetry information in voxel grid using an octree
- Voxel spacing set to r/4
- Optional step for removing outliers
Applications

- Three tools
  - Random shape creation
  - Semi-automatic modeling
  - Grid-based resizing
Results

- Single-threaded C++ implementation
- 2.6 Ghz Core2 Duo, 8GB of RAM
- Town hall facade model
  - 1.2 million points
  - Symmetry detection: 72 sec
  - Docking sites/dockers computation: 18 sec
  - Grammar extraction: 125 sec
- Parking structure model
  - 500,000 triangles
  - Total time: About 10 minutes
- Random generation and interactive Editing is real-time
Limitations and future work

- Limited to almost exact symmetry and similarity
- Implementation limited to rigid similarity
- Restricted to context-free and grid-based rules
- Does not extract all possible rules