The Room Connectivity Graph: Shape Retrieval in the Architectural Domain

Raoul Wessel, Ina Blümel and Reinhard Klein

University of Bonn
Introduction

• 3D modeling becomes more and more important for architecture
• Large collections of 3D building models in various formats (e.g. 3ds, max, vrml)

Problem:
• Modeling is **expensive** and **time consuming**
• Reusability as templates or inspiration source is limited
  – Lack of shape retrieval methods focusing on architectural needs
Introduction

- Major ingredient for architectural drafting: **2D floor plans**
  - Geometry and structure of buildings
  - Spatial organization (topology and disposition of rooms)
  - Implies scale, style (e.g. gothic, modernism), use and function

- Idea:
  - Extract 2D floor plans to characterize building stories
  - Starting point for retrieval, clustering, classification,...

- Demands:
  - Format independence
  - Robust extraction (non-manifold meshes, modeling errors,...)
Related Work

• Common shape retrieval techniques
  – mostly focus on general 3D objects
  – extraction of rather low-level local or global geometric features

  – View-based methods
    • e.g. Makadia et al., *Light Field Similarity for Model Retrieval*, SMI 2006

  – Matching of local features
    • e.g. Funkhouser et al., *Partial Matching of 3D Shapes with Priority-driven search*, SGP 2006

  – Graph-based approaches
    • e.g. Tal et al., *Mesh Retrieval by Components*, GRAPP 2006

Common approaches not well-suited for high-level semantic features like rooms, doors, windows
Contribution

• Use of 2D floor plans for characterizing architectural data:
  – Introduction of Room Connectivity Graph (RCG) as basic data structure

• Robust extraction method for RCGs
  – Requires only polygon-soup modeled buildings
  – Robustness towards unintended modeling errors

• Efficient retrieval of building models even from large databases based on RCGs
  – Fast graph matching technique using node and edge constraints
Story detection

- Prerequisite for floor plan extraction
  - Determine number of stories
  - Determine location of each single story

- Idea:
  - Stories are bordered by flooring and ceiling, minimum height about 2.40m
  - Determine large planar polygon patches
  - Identify ceilings and floorings according to minimum height criterion
Floor plan generation

- Extract 2D floor plan containing rooms
  - Compute cut between story and horizontal plane slightly below the ceiling
    - Set of line segments $\Rightarrow$ Convert to 2D halfedges
    - Establish connectivity due to threshold $\epsilon$ (1mm)

- Each room will be represented as one single face in the resulting halfedge structure as lintels prevent rooms from being connected
Room extraction

- Determine faces in halfedges structure
  - Distinguish *inside* faces and *outside* faces
    - Inside faces: walls, rooms
    - Outside faces: facade, structures inside another room
    - Drop all outside faces except for facade

- Determine rooms
  - For all inside faces compute area and extent
    - Room area usually larger than 1m²
    - Extent-to-area ratio \( \alpha \) usually larger for walls than for rooms
  - Use \( \alpha \) values of preclassified rooms and walls to predict face types
Gap closing

- Extracted floor plans contain unintended gaps (up to ~10cm)

- Naive approach: Increase $\epsilon$ threshold for floor plan construction

- Drawbacks:
  - floor plan shape changes dramatically
  - walls disappear, windows and doors become unrecognizable

- Instead: Conduct gap closing operations significantly changing the room topology
Gap closing

- Consider two possible gap-closing operations:

- A gap-closing operation is called *valid* if it splits an existing room face into two faces that still satisfy the room conditions (area, extent / area)

```
Gap closing (room face f)
For each node in f
{
    Check for valid gap closing operations
    If valid operation found
    {
        split room face
    }
}
```
Door and window detection

- Compute two additional cuts
  - Breast height (about 1.40m)
  - Marginally above flooring
- Doors and windows create inconsistencies:
  - Door: breast-height cut inconsistent with ceiling cut
  - Window: breast-height cut inconsistent with both other cuts
- Determine inconsistencies
- Add edges to room connectivity graph
Results

Number of polygons: 30629

Extraction time: 10.451s

Number of polygons: 2077

Extraction time: 0.421s
Results

Database extraction time (100 buildings): 436.512s
Retrieval using connectivity graphs

• Create query graph
  – By hand
  – or use existing room connectivity (sub)graph

• Retrieval
  – Compute room connectivity graphs for all building models in database
  – Determine subgraph-isomorphisms between query graph and database graphs by constrained graph matching:
    • Node constraints: area and extent of rooms
    • Edge constraints: structure type (either door or window)
  – Return building models containing query graph
Results and timings

Query graph representing typical apartment

Example for retrieval results from database

Retrieval timings:
Both examples: < 10ms
Whole Database (100 buildings): 0.173s
Conclusion

• Room Connectivity Graph
  – basic structure for characterizing architectural data
  – robust method for extraction
  – allows for efficient retrieval of room configurations even in large databases

• Limitations
  – extraction fails if elements are not modeled uniquely
  – currently restricted to single stories
Future Work

• More detailed room descriptions
  – 2D descriptors (e.g. Zernike moments, centroid distance)
  – amount of sunlight at a certain time
  – (automatic) classification of room type / use

• Extension to 3D
  – interlink graphs of different stories via elevator- or staircase-edges

• Retrieval and Classification
  – graph clustering
  – automatic building classification according to graphs
Thanks for your attention!