Daylight simulation in historical buildings and urban environments

Adaptive Level Of Detail for urban simulations

Gonzalo Besuievsky
Gustavo Patow
Grupo de Geometría y Gráficos
Universitat de Girona
Benoit Beckers
Universtité de Technologie sw Compiègne
Urban models complexity
Cities are large collections of man-made structures. Different levels of details are represented.
Urban models are used in wide range of areas such as:

- Urban and landscape planning
- Architectural design
- Environmental simulations
- Disaster management
- Vehicle and pedestrian navigation

All requirements are different but shares common structures
Concrete problem

- Problem:
  Huge amount of geometry to be processed
  Excessive for real-time visualization or simulation

- Solution:
  LoD: Create exactly what you need
LoD in 3D city models

- **LoD for buildings**
  - Preserve topology, use textures
  - LoD for BIM

- **Industry Standards for City Level**
  - BLOM3D
  - CityGML
  - Semantic Definition for LoD 0-4
CityGML standard

Footprint → Block model → Coarse exterior → Fine exterior → Interior
Concerning LoD definitions

- Many purposes:
  - Computational reasons
  - Focusing in visualization
  - No more detail than actually required

- Automatic LoD generation

- Difficult to find a single standard
  - Simplification criteria are often driven by applications
Our Works on LoD

- Based on **PROCEDURAL MODELING**
  - Efficient creation of detailed building models at low cost
- Automatic LoD [BP2013]
- Hierarchical urban model
  - Work on natural city level:
    - Districts, Blocks, Buildings, Facades, Elements
- Configurable LoD [BBBP2014]
  - Based on specific simplification criteria
- Adaptive LoD for Solar Simulation
  - Current work
Generic process for urban model generation
Procedural Modeling of Buildings

Notation:

\[ \text{predecessor} \rightarrow \text{CommandA,CommandB : labelB; labelB} \]
\[ \rightarrow \text{CommandC : labelC; } \]
Component Split

Format:

\[ id: A \rightarrow Comp(type, params)\{ B | C | ... | Z \} \]

Example:

1: solid \rightarrow Comp(“sidefaces”){ facade }
Example:

1: facade →

\[ \text{Subdiv(“Y”,3.5,0.3,3,3,3)} \{ \text{floor} | \text{ledge} | \text{floor} | \text{floor} | \text{floor} \} \]

```
+-------+-------+-------+-------+
| B     | A     | A     | B     |
|       |       |       |       |
|       |       |       |       |
| B     | A     | A     | B     |
|       |       |       |       |
| B     | A     | A     | B     |
|       |       |       |       |
| B     | A     | A     | B     |
+-------+-------+-------+-------+
```

} floor 3.0m
} floor 3.0m
} floor 3.0m
} ledge 30cm
} floor 3.5m
The Repeat Split

Example:

1: floor → Repeat(“X”,2) { window }

- Create as many window elements of approximate size 2 as there is space
Example:

1: $A \rightarrow \text{Insert(“window.obj”)}$
2: $B \rightarrow \text{Insert(“door.obj”)}$
Procedural Modeling of Buildings
Building Example

Lot
CreateBase : mass

mass
Comp: facade, sides, roof

facade
Subdiv: bottom, otherFloors

roof
Roof

bottom
Subdiv: floor, door

otherFloors
Repeat: floor

floor
Repeat: window

door
Insert(door)

window
Insert(window)

Insert(frame)
Lot CreateBase : mass

mass
  Comp: facade, sides, roof

facade
  Subdiv: bottom, middle, top

roof
  Roof

bottom
  Subdiv: floor, door

door
  Insert(door)

floor
  Repeat: window

window
  Insert(window)
  Insert(frame)

Rules & Labels

Bottom + Window
LOD by Configurable Rules

- **CityEngine® Software:**
  - Define attributes to apply assets at different model resolution or texture
  - They are manually defined
  - Resolution is pre-fixed

<table>
<thead>
<tr>
<th>Perspective</th>
<th></th>
<th>Perspective</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fps</td>
<td>1000</td>
<td>Fps</td>
</tr>
<tr>
<td>Objects</td>
<td>2 (0)</td>
<td>Objects</td>
</tr>
<tr>
<td>Polygons</td>
<td>475 (0)</td>
<td>Polygons</td>
</tr>
<tr>
<td>Cam Rot</td>
<td>-17.17 -92.01 0.00</td>
<td>Cam Rot</td>
</tr>
<tr>
<td>Cam Pos</td>
<td>-153.11 14.82 -66.96</td>
<td>Cam Pos</td>
</tr>
<tr>
<td>Pol Pos</td>
<td>-102.2 -0.9 -65.2 (Dist 53.3)</td>
<td>Pol Pos</td>
</tr>
</tbody>
</table>
Automatic LoD: [BP2013], [BP2013b]
Automatic LoD

- Work on ruleset level
- Improved by the hierarchy of urban model
  - Districts, Blocks, Buildings, Facades, Elements
• Use bounding box to evaluate affectation
Combining simplification criteria

- Distance_LoD
- ScreenSize_LoD [LE97]
- Semantic_Selection
- Programming combination
Asset-level LoD

100% 75% 50% ... 25% 0%
Results: City Demo

1436 buildings

103 blocks

17 Million polygons
Affectation area
Simplification
LoD Criteria
## Complex City

<table>
<thead>
<tr>
<th></th>
<th>Full Urban Model</th>
<th>low LoD</th>
<th>fr1</th>
<th>fr60</th>
</tr>
</thead>
<tbody>
<tr>
<td># Polygons</td>
<td>17M</td>
<td>129K</td>
<td>501K</td>
<td>436K</td>
</tr>
<tr>
<td>% Reduction</td>
<td>0.76%</td>
<td>2.9%</td>
<td>2.5%</td>
<td></td>
</tr>
</tbody>
</table>

![Complex City Model](image)
Our Aim for Solar Simulation

- Configurable LoD for Procedural Modeling
- Develop a tool for flexible generation of buildings at different LoDs
- Propose different criteria for simplification
  - Semantic criteria
  - Point of Interest criteria
- Focus on solar simulation applications
- Which geometry details are important for daylighting analysis?
Configurable LoD

- For daylight simulation:
  - [BBBP2014] Presented at UDMV2014

<table>
<thead>
<tr>
<th>Input model</th>
<th>User Configuration</th>
<th>Output models</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buildings model with details</td>
<td>Elements: • Windows • Doors • Balconies • ...</td>
<td>Level 1</td>
</tr>
<tr>
<td>Criteria: • Point of Interest • Semantic</td>
<td>Level 2</td>
<td>Level 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Multi-scale</td>
</tr>
</tbody>
</table>
- Graph-based transformation
- Create a new command: \textit{LoD}
  - Search for selected elements
  - Inserts automatically for each one
- Parameters configuration
  - Specific implementations for each criteria
- Geometry replacement
  - Implement command \textit{InsertLoD}
Classify geometry according to distance
Point Of Interest

- Reduction of geometry: 12%
- No significant difference in simulation results
Semantic LoD

- Use Semantic combination for selection and further replacement
  - “top floor windows”
  - “middle floor windows”
  - “bottom floor windows”
Adaptive LoD with skyline control

- **Goal:**
  - Work with large urban models providing accuracy for detailed geometry
  - Obtain an automatic model simplification for computing solar radiation from a region of interest

- **Features:**
  - Preserves all roof details by controlling the skyline important impact
  - Intended for Procedural urban models, but also accepts any 3D CAD geometry
Initial observation

- Most of the relevant details for solar simulation belongs to the silhouette.
- For most of the rest, it may be enough to detect just the occluder.
- We also note that silhouettes are more relevant at the skyline.

Sky View Factor test:

- Full model: 85.19%
- Bbox model: 83.32%
- LoD Proposal: 85.19%
Initial observation

**Full Geometry**
Mean SVF:
Hplane = 34.9%
Vplane = 28.2%

**Simplification proposal**
Mean SVF:
Hplane = 34.4%
Vplane = 28.1%
First step: Bounding box Building Generation

- Traverse the procedural model graph detecting the \textit{Insert} command placement
- Compute the Bbox and replace the full geometry to build a new procedural Bbox model
Second Step: Cast rays from a bundle distribution

- Build random samples in the sky vault using equal area cell distribution [B. Beckers & P. Beckers 2012]
- The distribution are useful for SVF computation
Second step: Compute impacts

- Cast rays and compute impacts over the approximated model
LoD Generation Step

- Use the number of impacts to build a model simplification according to a defined criteria

- We define three geometry levels
  - LoD0: The geometry is omitted from the model
  - LoD1: The geometry is maintained at a bounding box level
  - LoD2: The geometry is represented at full level
Criteria algorithm

For procedural models:

\[
\text{if } \text{impacts}(p) > 0 \\
\quad \text{if } p \text{ is roof and } \text{Silhouette}(p) = \text{true} \\
\quad \quad \text{LoD2} \\
\quad \text{else} \\
\quad \quad \text{LoD1} \\
\quad \text{endif} \\
\text{else} \\
\quad \text{LoD0} \\
\text{endif}
\]

- \text{impacts}(p) \text{ returns the number of impact arriving to } p
- \text{Silhouette}(p) \text{ evaluates if the polygon belongs to the skyline or not}

For non-procedural 3D objects:

Get only impacted polygons
Silhouette detection

- Project the candidates polygons onto the POI bundle rays sphere.
- Detect if there are sky points close to the polygon
LoD Generation

- For several POIs:
  - A whole façade
  - A whole street

- Merge all LoD levels resulted
  - Compute the union of all models
LoD Merging
First Results

Full model
297K pol

LoD model
20K pol

Error Rel. ~ 0.01 in %
Perspectives and Discussion

- Allow to compute simulation for any place of a complex urban model
Discussion and Future Work

• A new LoD proposal for performing simulation with detailed geometry in large urban models
  • Automatic for getting relevant geometry
  • Adaptive for region of interest

• Future Work:
  • Extend POI concept for higher level structures: facades, buildings, blocks
  • Explore simplification for other physical simulations: heat, acoustics
Daylight simulation in historical buildings and urban environments

Adaptive Level Of Detail for urban simulations

Gonzalo Besuieevsky
Grupo de Geometría y Gráficos
Universitat de Girona