Exploring the automatic Level of Detail inference for the validation of buildings in 3D city models

MSc thesis presentation
Geomatics for the Built Environment

Balázs Dukai
Supervisors:
  Dr. Filip Biljecki
  Dr. Hugo Ledoux
  Anna Labetski
Co-reader: Dr. Jorge Lopez Gil 30.01.2018
Source: [1]

TU Delft
Philadelphia Redevelopment

Level of Detail (LoD)
Motivation

Source: LoD2 model of Bad Godesberg, NRW, Germany

Source: Google Maps
Motivation

Source: LoD2 model of Amsterdam, virtualcitySystems

Source: Google Maps
Motivation

Source: LoD2 model of Bad Godesberg, NRW, Germany

Source: Google Maps
Motivation

• Knowing the accurate LoD is important for analysis and maintenance

• CityGML 2.0 is not clear on LoD, CityGML 3.0 will probably complicate things

• Roof reconstruction (>LoD2) fails occasionally

• Heterogenous LoD

• CityGML has no explicit LoD attribute per building, non-semantic formats have no tag at all
Research questions (paraphrased)

*How to determine the geometric LoD automatically?*

- How to classify the geometry of 3D building models (in terms of LoD)?
  - How to describe the geometry of a building model for the classification?
Research questions (paraphrased)

• How to validate the LoD automatically?
  − Without comparing to a reference data set?
  − By comparison with a reference data set?
LoD\textsuperscript{3} revisited

CityGML2.0
LoD0.1-0.3, 1.1-2.3
Method

Step 1
- 3D city model (OBJ)
- Extract building surfaces
- Generate features
- Create design set
- Classifier training and evaluation
- Trained classifier
- Classify buildings

Step 2
- Ref. data: point cloud
- Clip point cloud to building extent
- Compute point cloud - mesh distance
- Compute RMSE

Optional: manual LoD validation
Synthetic data – LoD0.1-0.3, 1.1-2.3

1000 buildings
100 per class
Amsterdam data – LoD1.2, LoD2

482 valid buildings (green)
Amsterdam data – LoD2 (and LoD1)

Imbalanced LoD classes – LoD2 (reds), LoD1 (blues)
Extract building surfaces
## Generate features

<table>
<thead>
<tr>
<th>Geometry</th>
<th>Feature</th>
<th>Related LoD requirement</th>
<th>Relevant LoD</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D footprint</td>
<td>Number of Shape Characterising Points (NSCP)</td>
<td>none</td>
<td>all</td>
</tr>
<tr>
<td></td>
<td>Shape Characterising Lengths (SCL)</td>
<td>Size of building parts</td>
<td>≥ 0.1</td>
</tr>
<tr>
<td></td>
<td>Footprint Area</td>
<td>Size of building parts</td>
<td>≥ 0.1</td>
</tr>
<tr>
<td></td>
<td>Building Part Footprint Area</td>
<td>Size of building parts</td>
<td>≥ 0.1</td>
</tr>
<tr>
<td>3D solid</td>
<td>Building Volume</td>
<td>none</td>
<td>all</td>
</tr>
<tr>
<td>3D surface</td>
<td>Roof Type</td>
<td>Roof representation</td>
<td>≥ 1</td>
</tr>
<tr>
<td></td>
<td>Median Roof Gap</td>
<td>Top surface (Single / Multi)</td>
<td>0.2-1.3</td>
</tr>
<tr>
<td></td>
<td>Roof Overhangs</td>
<td>Explicit roof overhangs (if 0.2m)</td>
<td>≥ 2.3</td>
</tr>
<tr>
<td></td>
<td>Footprint-Roof Triangle Ratio</td>
<td>Roof superstructures</td>
<td>≥ 2.2</td>
</tr>
<tr>
<td></td>
<td>Walls</td>
<td>Presence of walls</td>
<td>0</td>
</tr>
<tr>
<td>3D solid, Point Cloud</td>
<td>RMSE of PC-Model distance</td>
<td>(LoD validity)</td>
<td>all</td>
</tr>
</tbody>
</table>
NSCP & SCL

Inner angle < 160°
Building part area

Building part 1

Building part 2
Roof type

- planar
- non-planar
- mixed
Median roof gap

Roof gap 2

Roof gap 1
RMSE

- Signed distance from point cloud to mesh
- With CloudCompare, per building

Figure 2: Signed distance evaluation; distance is positive in $p_1$ and negative in $p_2$ ($S_1$ is the sampled curve).
RMSE
Frequency distribution of Shape Characterising Point per LoD

Amsteram LoD2

Frequency distribution of Shape Characterising Point per LoD

Synthetic data set
Min. SCL

**Frequency distribution of minimal footprint SCL per LoD**

*Amsterdam LoD2*

*Synthetic data set*

Minimum Shape Characterising Lenght (SCL) [m]
Footprint-roof ratio

Footprint-roof triangle ratio per LoD

Amsterdam LoD2

Footprint-roof triangle ratio per LoD

Synthetic data
Classification

- Logistic Regression
- Linear Discriminant Analysis
- K Nearest Neighbours
- Decision Tree
- Gaussian Naive Bayes
- Support Vector Machine
Experiment 1&2

- Not / Standardized features
- Train and test in the same data
- Cross-validation and prediction
Experiment 1 & 2 – Raw and standardized features

Synthetic data
LR prediction: 42.5%

Amsterdam data
DTree prediction: 88.6%
Experiment 4

• Standardized features
• Train and test in Amsterdam data
• Include RMSE
• Binary classes (LoD2 or not)
Experiment 4

Algorithm Comparison

Dtree prediction 92.5% but:

<table>
<thead>
<tr>
<th></th>
<th>Not LoD2</th>
<th>LoD2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not LoD2</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>LoD2</td>
<td>2</td>
<td>83</td>
</tr>
</tbody>
</table>
Experiment 5

- Standardized features
- Train and test in Amsterdam data
- Replace 10, 25, 50 of LoD2 with LoD1
- Include RMSE
- Multi-class and Binary classes (LoD2 or not)
Experiment 5 – mixed LoD1&2

Algorithm Comparison, Multi-label, combined 10%, 25%, 50% LoD1

LR  LDA  kNN  DTree  GaussianNBLinearSVM
Experiment 5 – kNN
Experiment 3&6

- Standardized features
- Train in synthetic and test in Amsterdam
- Replace 10, 25, 50 of LoD2 with LoD1
- Include RMSE
- Multi-class
Experiment 3 & 6

Experiment 3
With LoD0

Experiment 6
No LoD0

<table>
<thead>
<tr>
<th></th>
<th>LR</th>
<th>DTree</th>
<th>NB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>7.4%</td>
<td>3.7%</td>
<td>0.0%</td>
</tr>
</tbody>
</table>
Conclusions

• Synthetic data is not suitable as design set
  – Representative data set
• Features seem to be OK, but are there better?
• 42%, 88%, binary classes 92%
• Class imbalance is an open problem
• Issues with noisy point cloud, distances are not reliable
  – Other reference data?
  – RMSE might be too coarse
• LoD inference and validation
References


