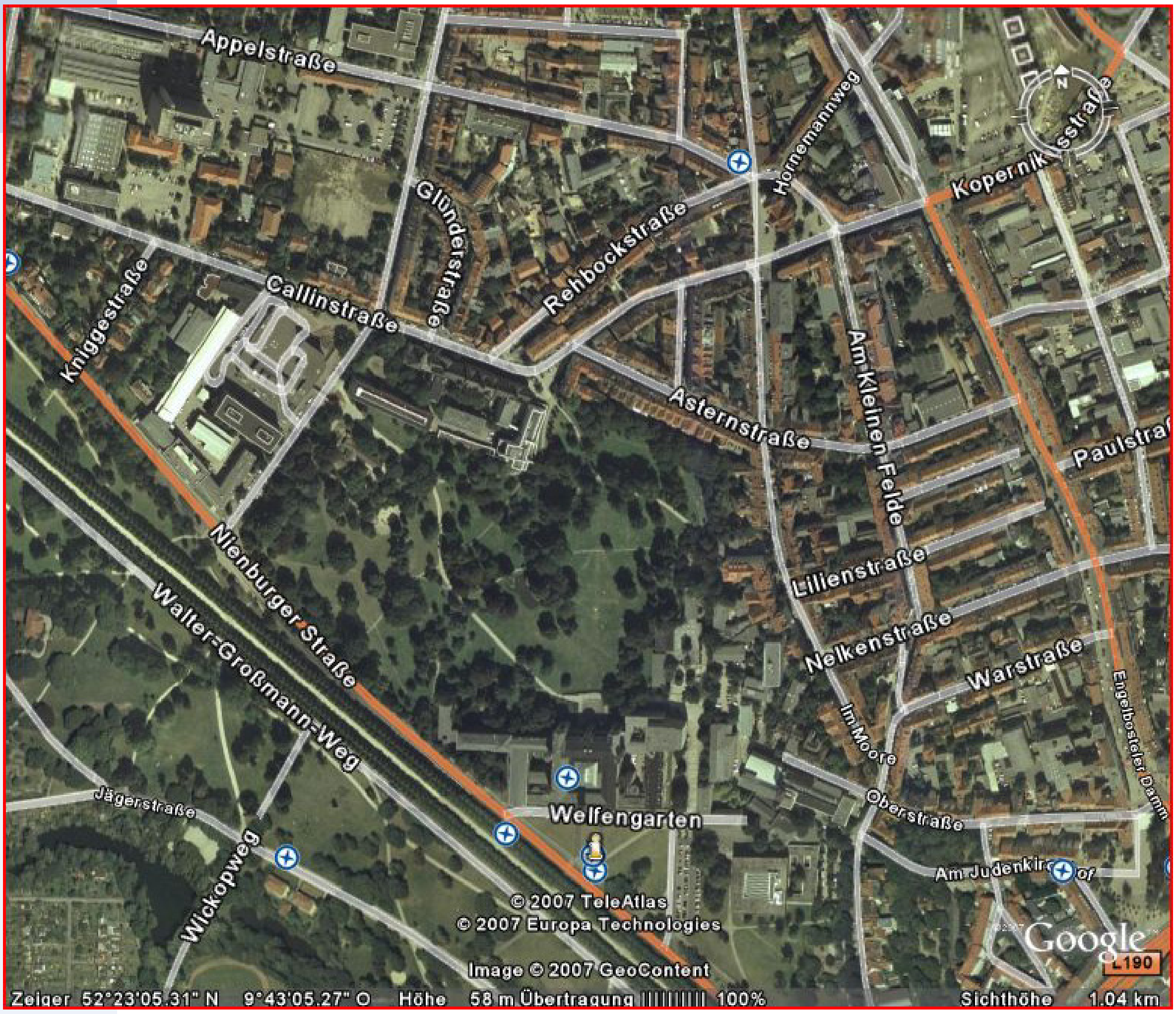
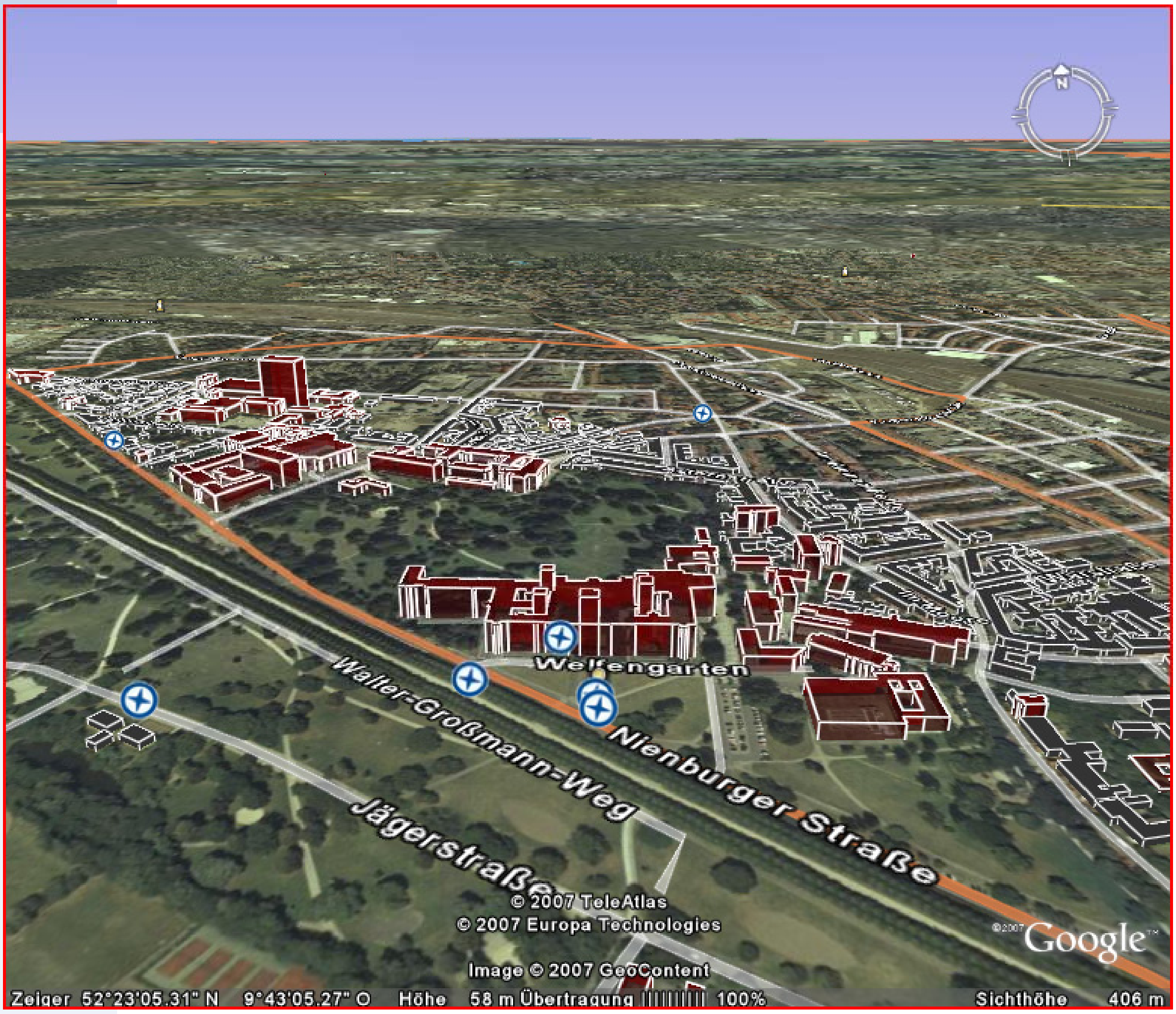
3D Visualization and Generalization

Monika Sester
Institute of Cartography and Geoinformatics
Leibniz University of Hannover
Germany




Appelstraße
Callinstraße
Kniggestraße
Glünderstraße
Rehbockstraße
Hornemannweg
Kopernikusstraße
Asternstraße
Am Kleinen Felde
Paulstraße
Lilienstraße
Nelkenstraße
Warstraße
Engelbosteler Damm
Im Moore
Oberstraße
Am Judenkirchhof
Nienburger Straße
Welfengarten
Walter-Großmann-Weg
Jägerstraße
Wickopweg
© 2007 TeleAtlas
© 2007 Europa Technologies
Image © 2007 GeoContent
Google
L190
Zeiger 52°23'05.31" N 9°43'05.27" O Höhe 58 m Übertragung 100%
Sichthöhe 1.04 km


Welfengarten
Walter-Großmann-Weg
Nienburger Straße
Jägerstraße
© 2007 TeleAtlas
© 2007 Europa Technologies
Image © 2007 GeoContent
Google
Zeiger 52°23'05.31" N 9°43'05.27" O Höhe 58 m Übertragung 100%
Sichthöhe 406 m

3D-city models – photo realistic with textures

Generated with **ATOP**: Free demo download:
http://www.ikg.uni-hannover.de/forschung/vw_stiftung/projekte/download/login.en.php

3D Visualization and Generalization

- 3D data sets are available
  - Acquired by automatic measuring techniques (high resolution imagery, Lidar, ...)
  - Accessible via Spatial Data Infrastructures
- Huge amounts of data
  - Too big to store on individual computer
  - Not adapted to specific applications
- Challenges:
  - Adapt amount of data
  - Adapt information content

Requirements

- Data reduction
  - Reduce amount of data to ease
    - Transmission
    - Visualization
    - Analysis
    - Interaction
- Support immediate cognition of spatial situation
  - Enhance important information
  - Importance is application dependent, e.g.
    - Navigation -> landmark objects
    - Localization -> highlight destination, gradually reduce information in environment
- Semantic and geometric generalization

Overview

- Levels of Detail (LODs) for buildings
- Building generalization in 2D
- Approaches for building generalization in 3D
- Summary and outlook

Level of Detail

- Problems:
  - Different users need / want different models
  - Detailed models cannot be visualized in real-time
    - Simplification of objects in large distance is needed
  - Data reduction, simplification, generalization:
    - complex operations – not (yet) possible in real-time
- Solution: LOD-Concept
  - Pre-compute different models for
    - Different distances, resolutions, scales
    - Different users / applications
  - Access „best“ model in each situation

3D-city models and Level of Details

Level of Detail according to CityGML

(Abb.: Stadtverwaltung Chemnitz)

LOD 0
Regional model
DTM + Photo

LOD 1
Box model
6m x 6m

LOD 2
With roof types
4m x 4m

LOD 3
Architecture model
2m x 2m

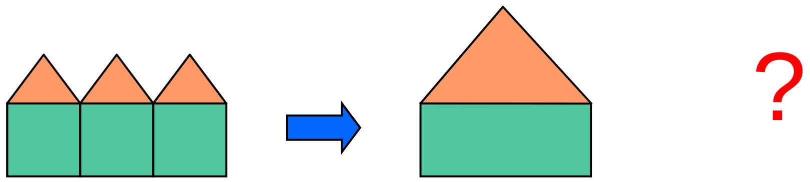
LOD 4
Indoor model
0.2m x 0.2m

Generalization

Progressive Transmission

LODs – Aggregation levels

- Aggregation levels
  - Defined in CityGML
  - However: only based on individual buildings
    - Not aggregates of buildings
    - Or aggregates of building blocks
  - Generation operations are not defined !
- Problem:
  - No “rules” like in 2D cartography
  - Attributes of aggregated objects ? E.g. adjacent objects – which height, roof type, façade structure ?



Generalization operations

Generalization Operations in cartography

- **Aim**: Reduce data volume while preserving important structures

- Simplification
  - Simplify form and geometry
- Aggregation
- Emphasis
  - Exaggerate size in order to be visible
- Displacement
  - Due to simplification and emphasis
  - Displace in order to be visible, if otherwise hidden by other objects

Generalization of buildings – in 2D

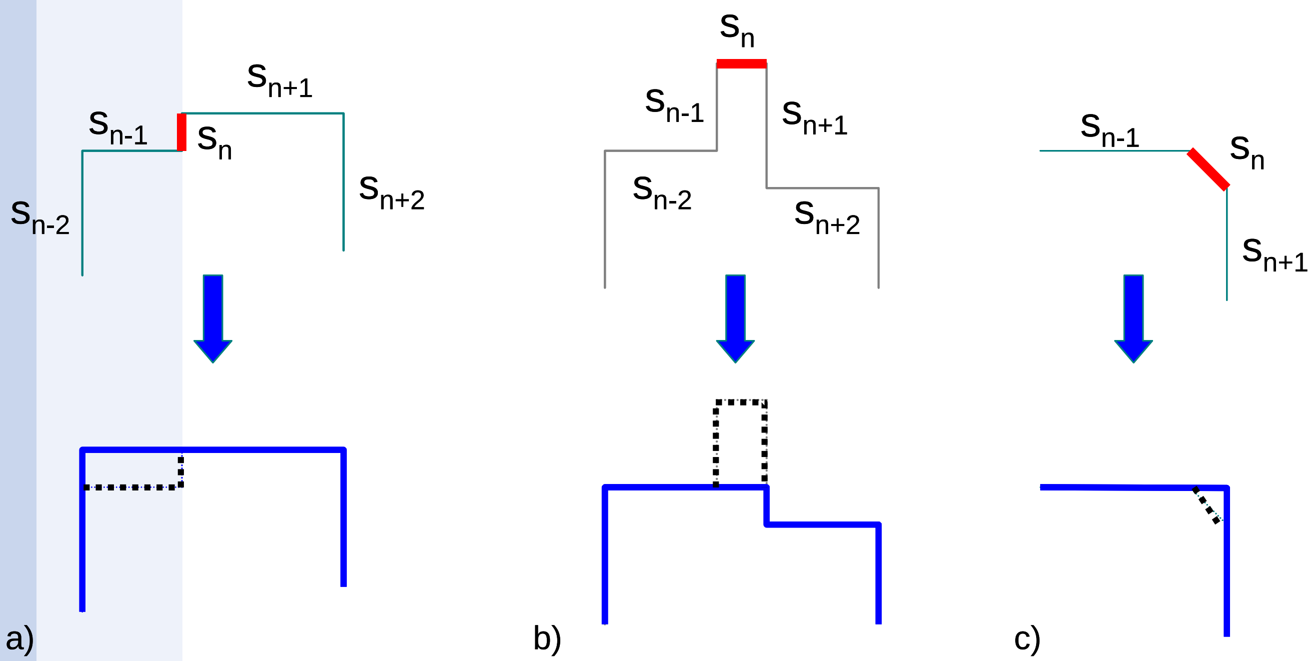
- Individual buidings
  - Simplification of building outline
  - Aggregation of adjacent buildings

- Groups of buildings
  - Aggregation of buildings within local neighborhood
  - Typification of buildings
  - Symbolization of buildings

Large -> small Scale

Rules

- Rules to eliminate short edges s_n

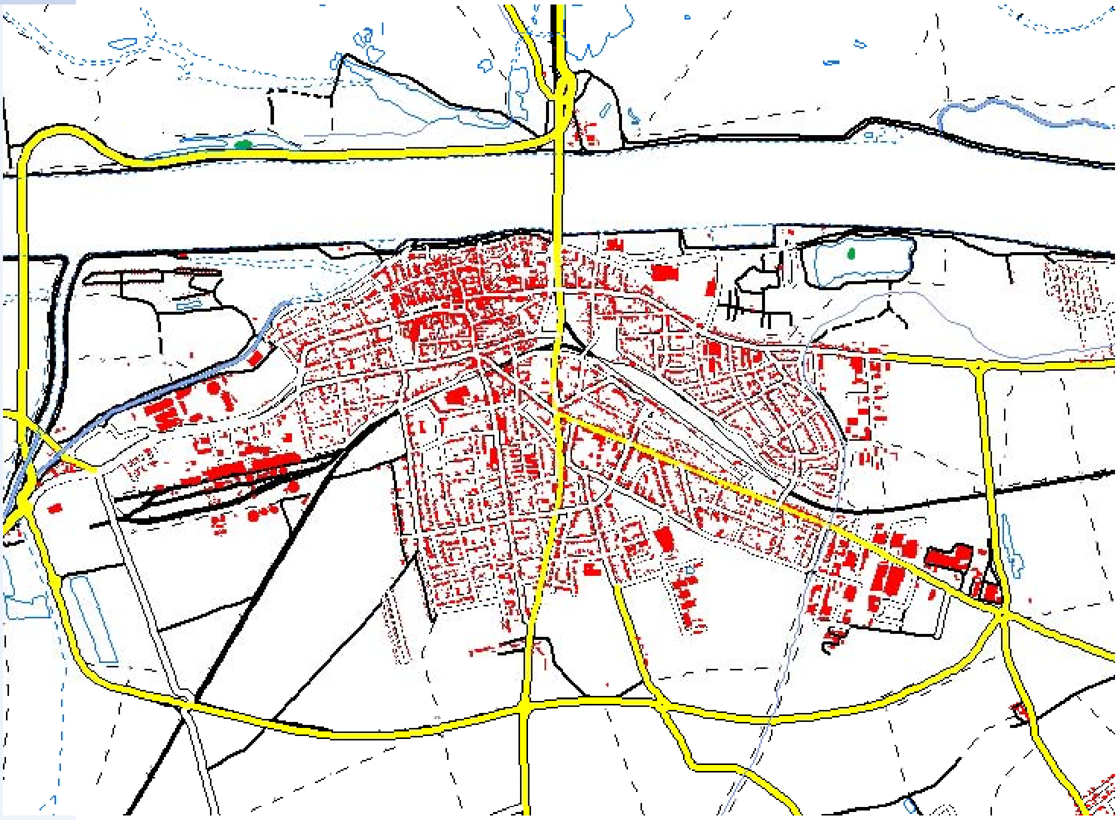


Simplification and aggregation with CHANGE

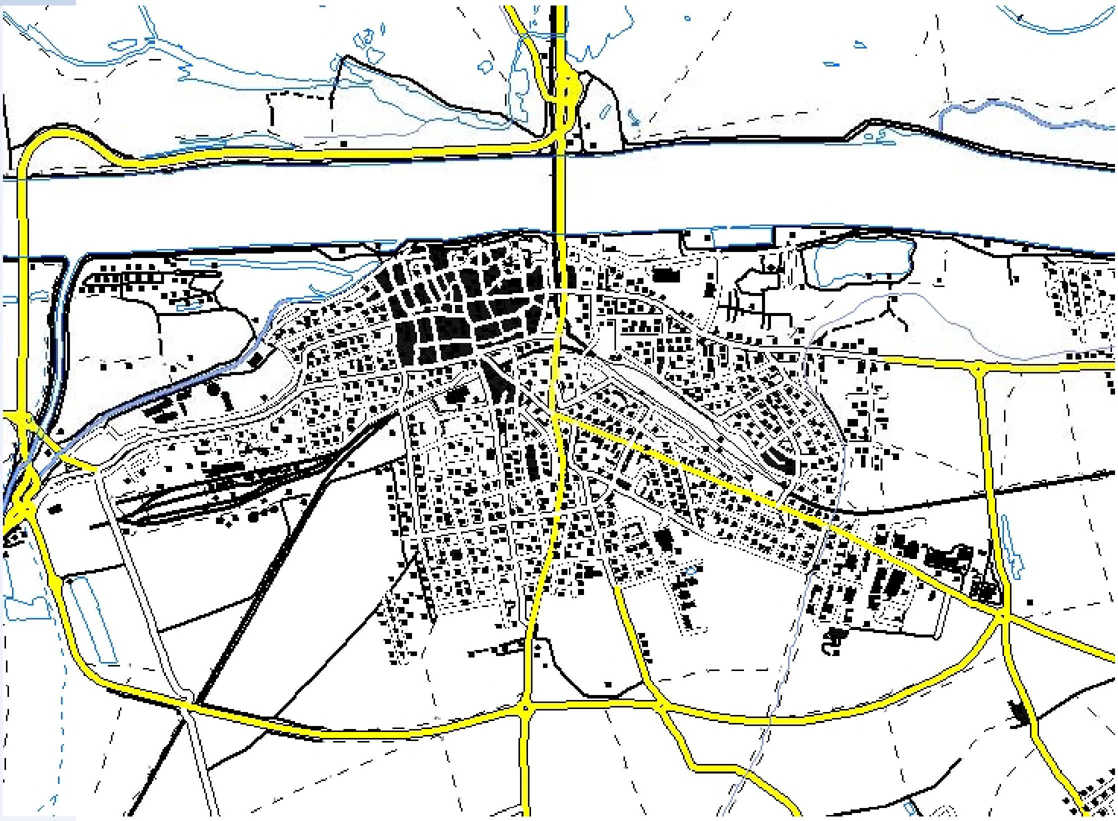
Smaller scales -> typification

After typification – density based reduction

Buildings before generalization

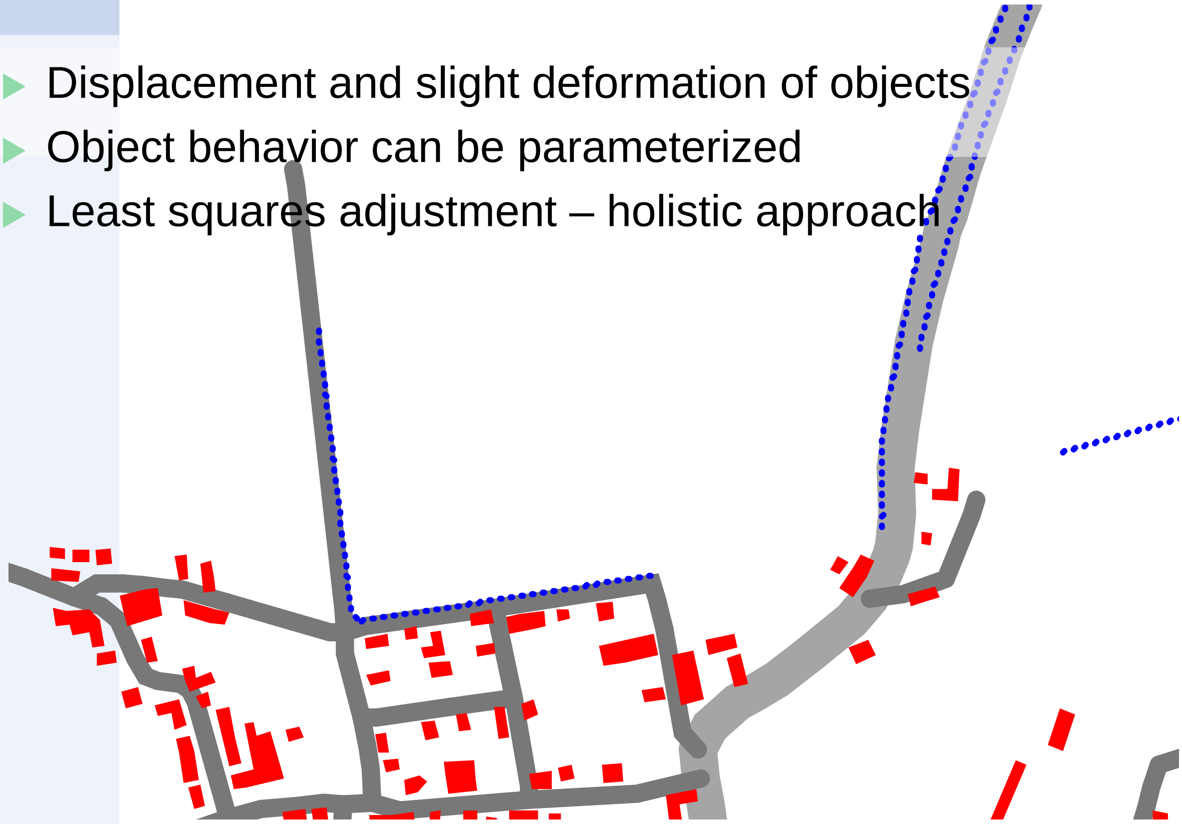


Buildings after typification and aggregation



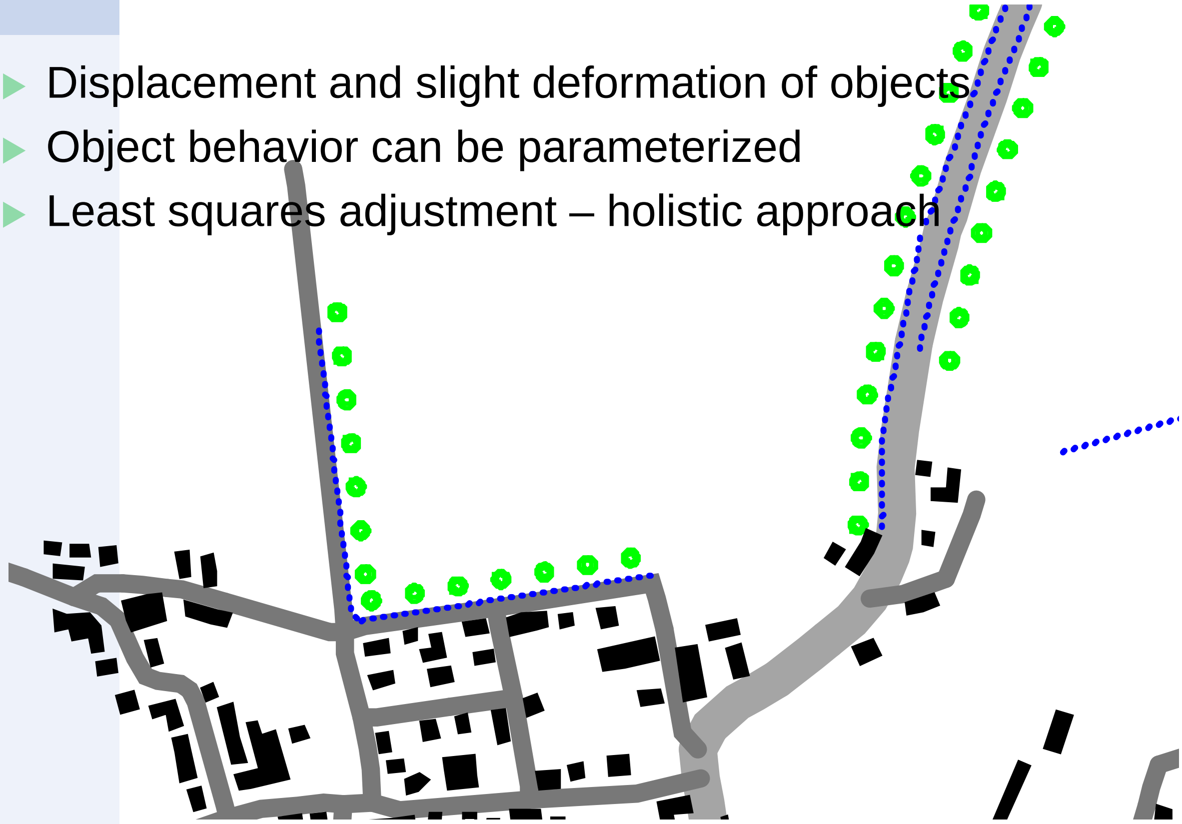
Automatic displacement – PUSH

- Displacement and slight deformation of objects
- Object behavior can be parameterized
- Least squares adjustment – holistic approach



After automatic displacement with PUSH

- Displacement and slight deformation of objects
- Object behavior can be parameterized
- Least squares adjustment – holistic approach



3D-generalization of buildings

Generalization: generation of LOD's

- Given:
  - Detailed representation
  - Different aggregation levels

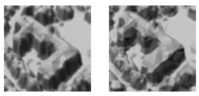
- generalization algorithms
  - **Computer Graphics**: simplification (e.g. mesh simplification) – depending on geometry and topology
  - **Cartographic generalization**: include also object specific properties – object dependent generalization
  - prerequisite:
    - Meaning of objects (and object parts) has to be known
    - … or derived by interpretation techniques

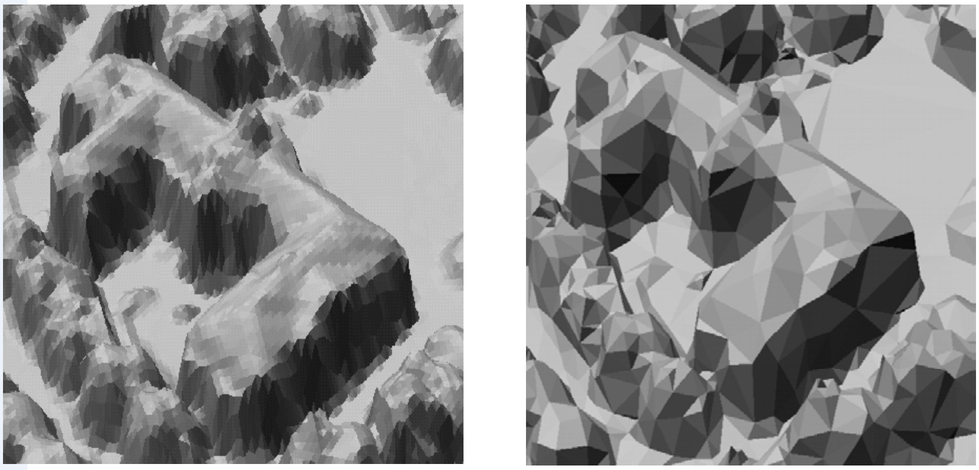
Generalization: generation of LOD's

- Given: detailed representation
- Definition of aggregation levels
- generalization algorithms
  - Computer Graphics: smoothing (e.g. mesh simplification) – depending on geometry and topology
  - Include also object specific properties: object dependent generalization
  - assumption: Meaning of objects (and object parts) is known

Mesh simplification – Computer Graphics

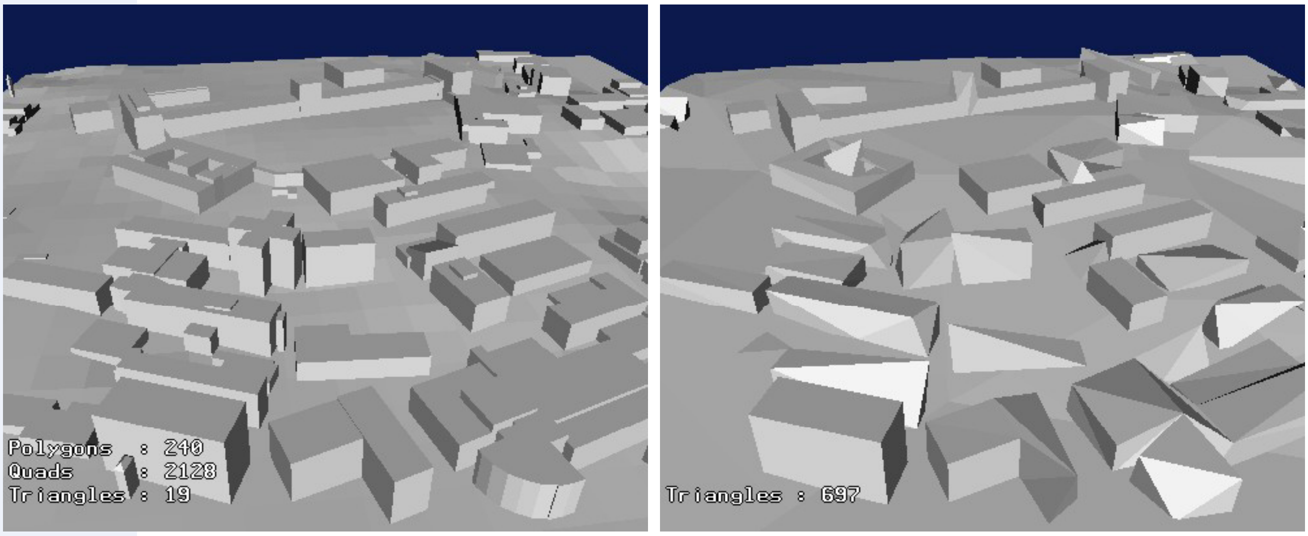
- Models with many redundant faces
  - Mesh simplification approaches
  - Decision mainly based on geometry





Mesh simplification with buildings

- Models with only a few faces

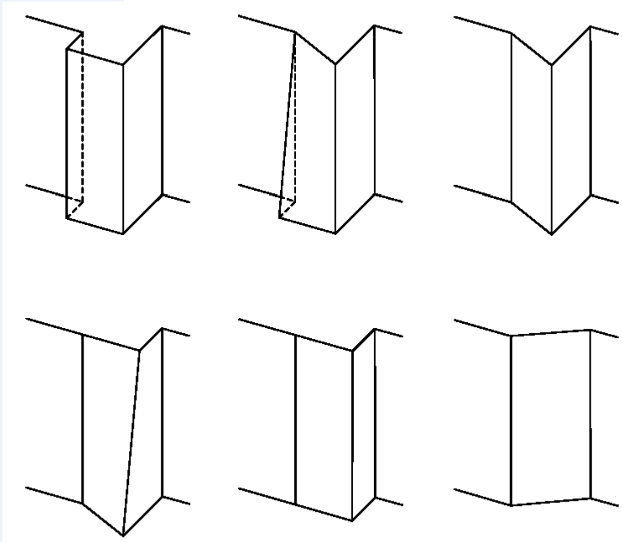


- Simple removal of edges is not possible
- Additional constraints have to be satisfied:
  - Rectangular, parallel, horizontal, vertical, ...

Simplification of geometry

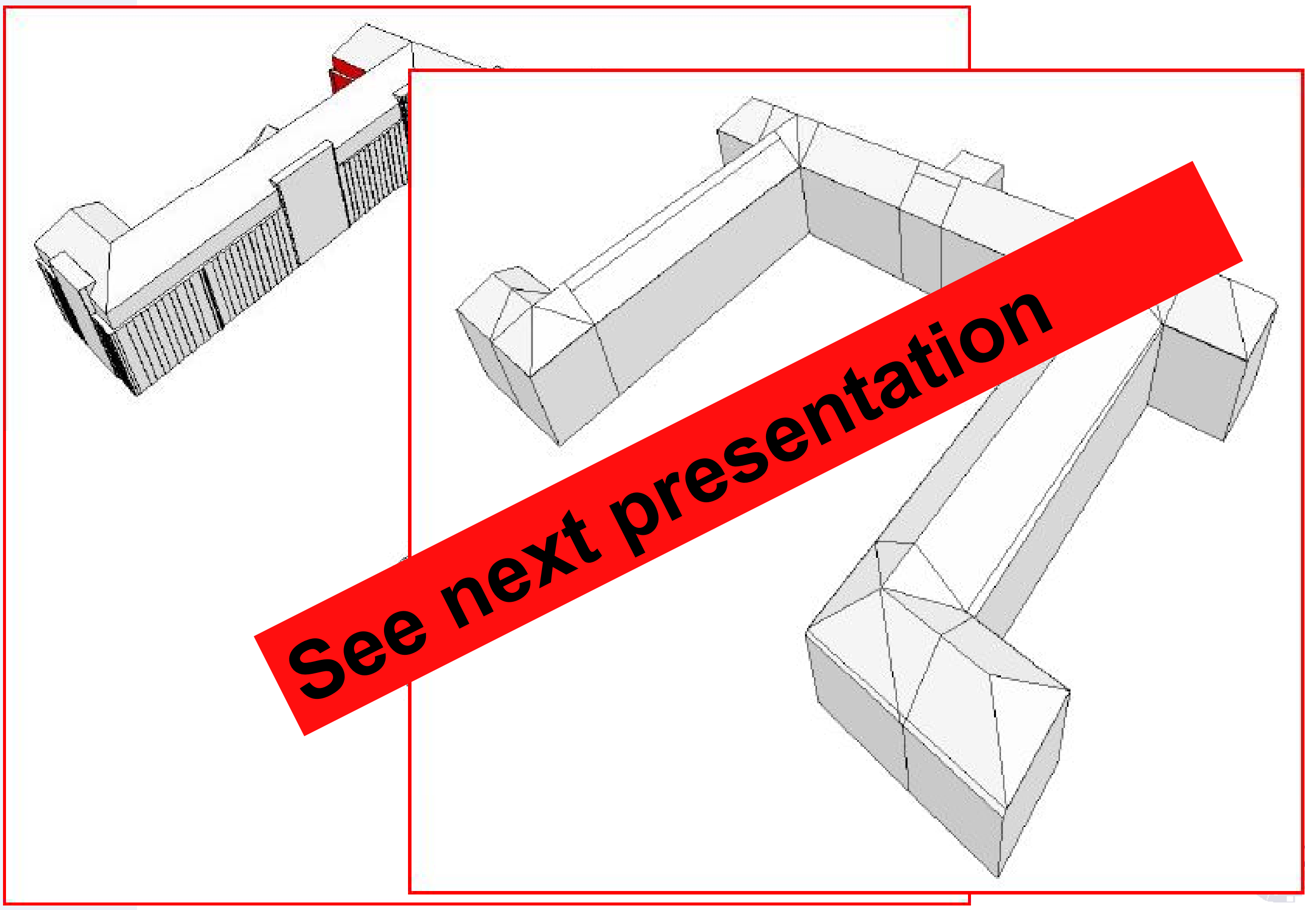
Approach Kada, Stuttgart

- Recognition of protrusions, peaks



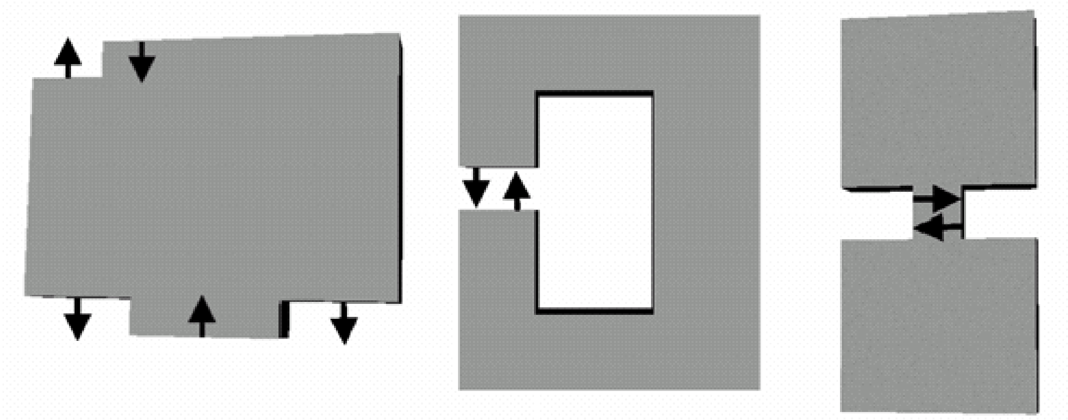
- E.g. protrusion
  - Several edges are eliminated in one step
  - “edge collaps” -> “protrusion collaps”
- Similar to 2D-generalization approaches

Kada (2006): buffered cells

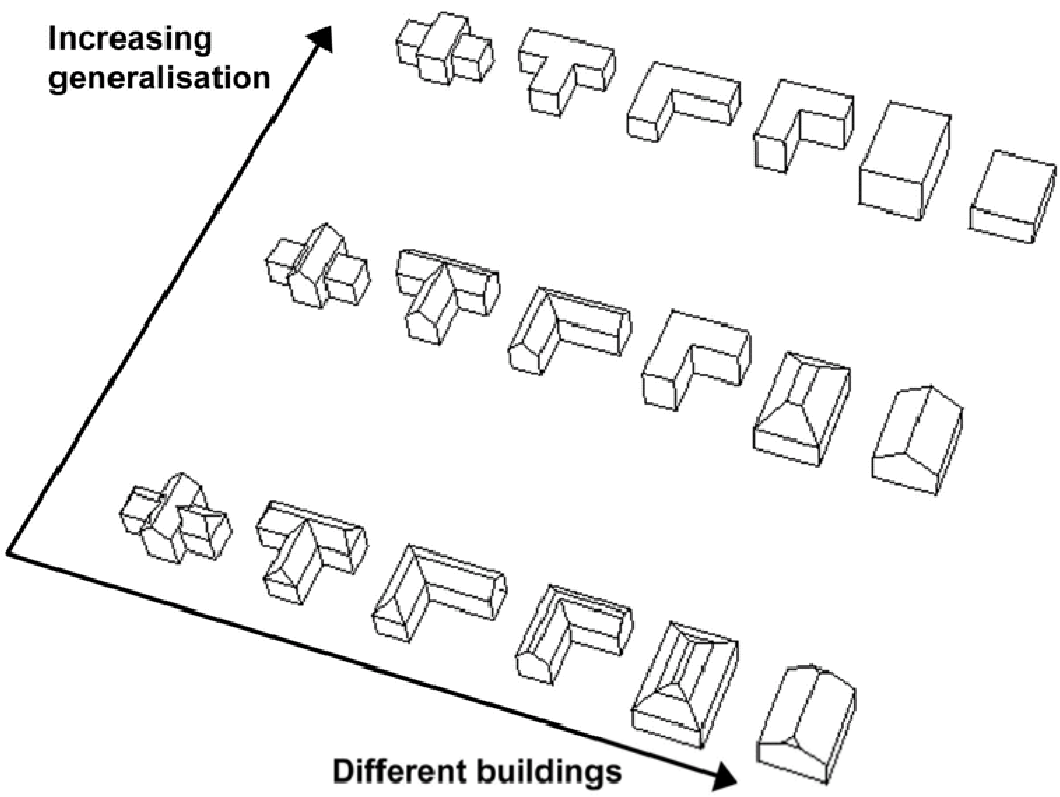


Approach Forberg, Munich

- Adaptation of mathematical morphology operations
- Intelligent succession of parallel shifts of neighboring faces
- Special treatment of roof structures



Generalization, incl. roofs



Approach Thiemann, Hannover

- Separation of generalization process in three steps:

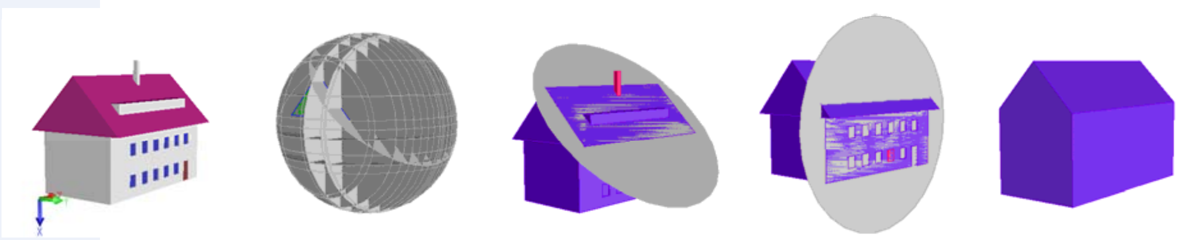
  1) Segmentation into (meaningful) parts

  2) Analysis of parts

  3) Semantic-based generalization

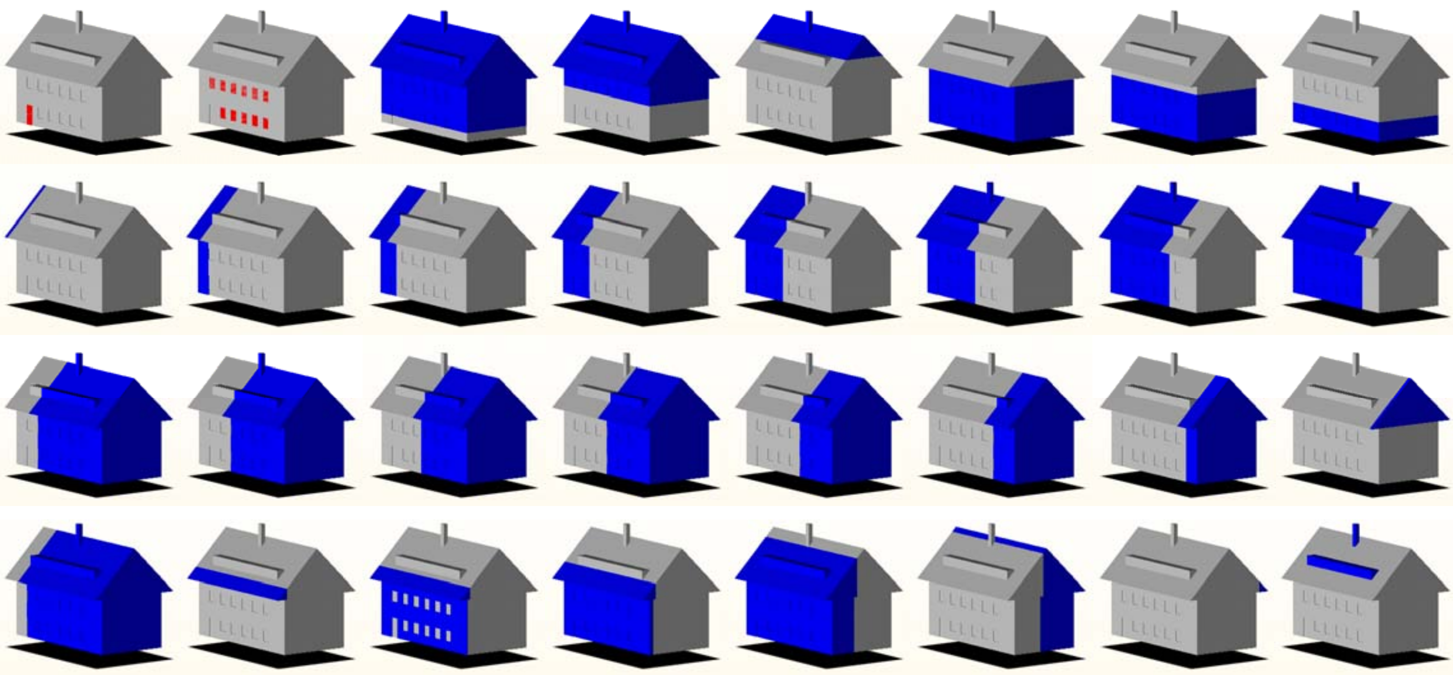
3D-Building Generalisation

- Idea
  - Decomposition of a building into small convex parts
  - Analysis of the parts and derivation of generalisation steps
- Approach
  - Intersection of the building planes [Ribelles et al.]
  - Cut off parts sticking out (e.g., chimney)
  - Holes (recesses, missing corners) will be filled

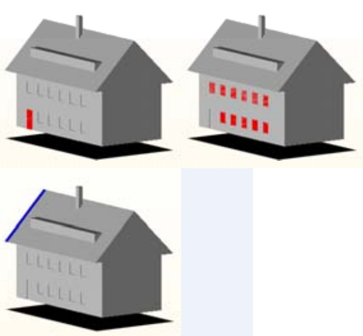


Original | Cutting planes | Cutting | Filling | Main part

Selection of split



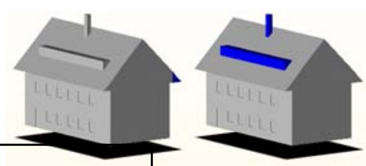
Selection of split



- Test value (after Ribelles et al.)

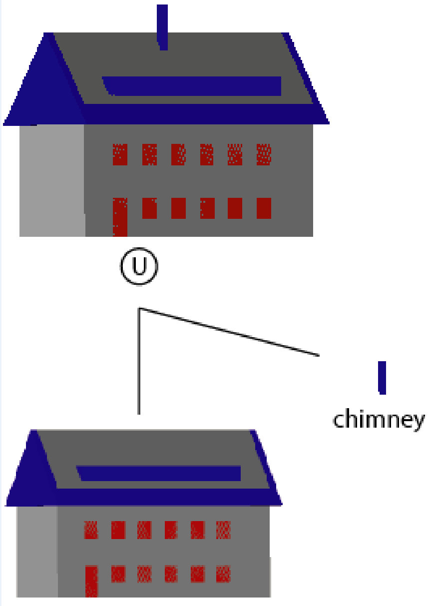
$$q = \frac{\text{area of the new inserted intersection face}}{\text{area of the original split plane}}$$

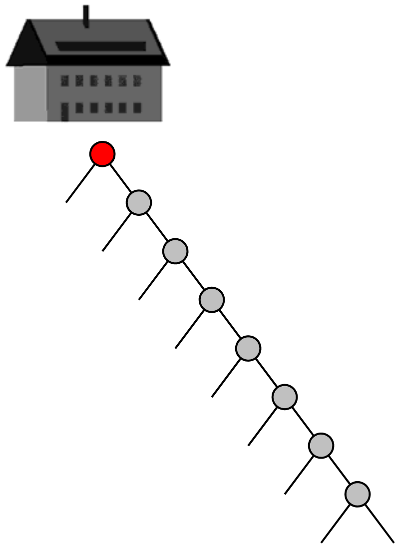
(q smaller → better)



- Use of best split with $q \leq 1$
- Recursive partitioning of both parts
- Store in CSG-tree

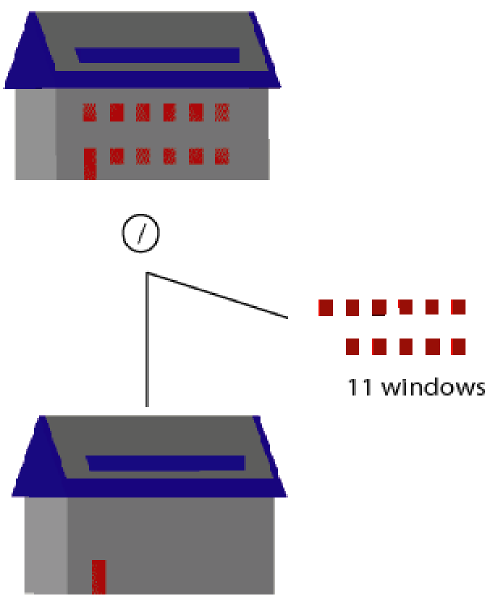
Sample building - Step 1

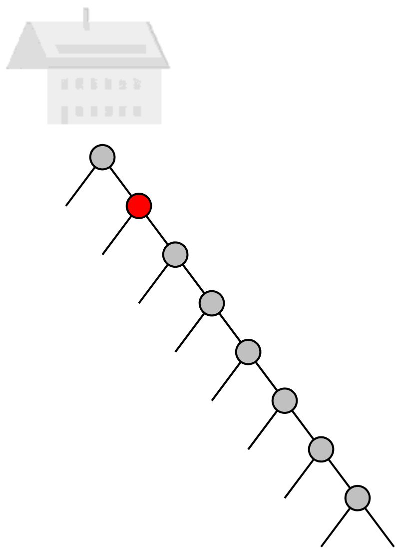




q= 0.003

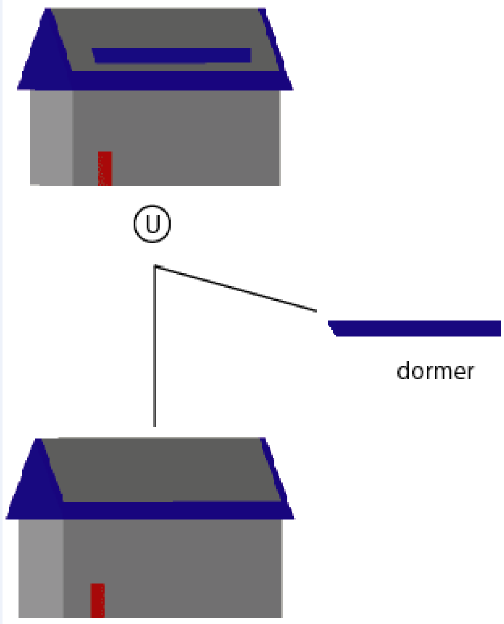
Sample building - Step 2

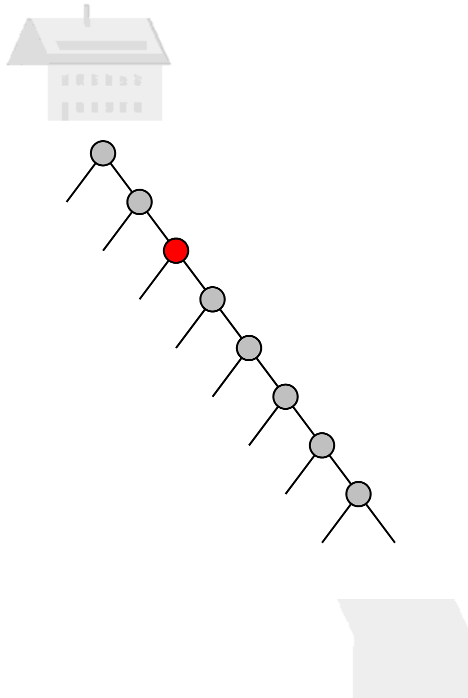




q= 0.014

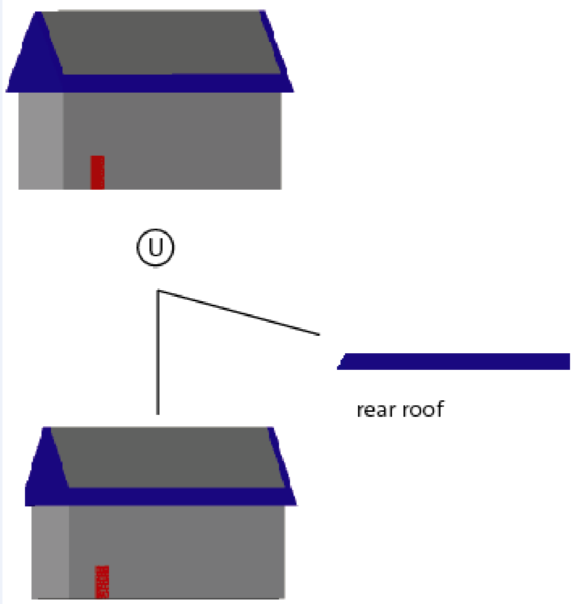
Sample building - Step 3

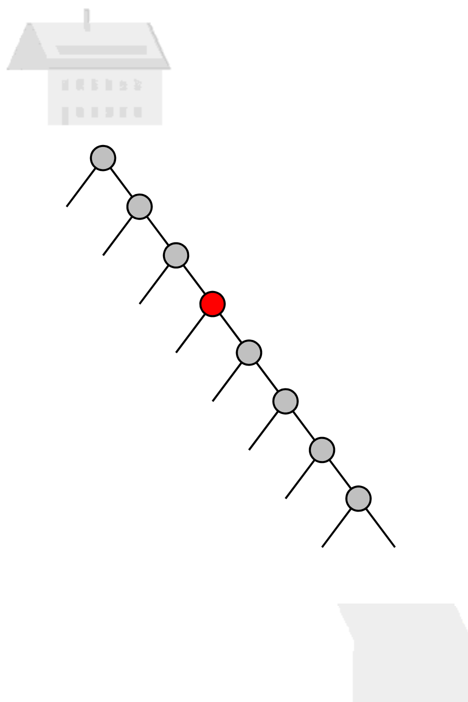




q= 0.162

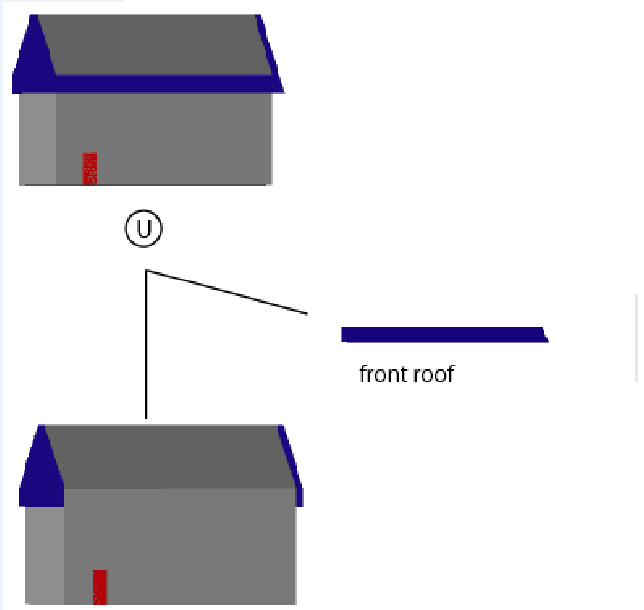
Sample building - Step 4

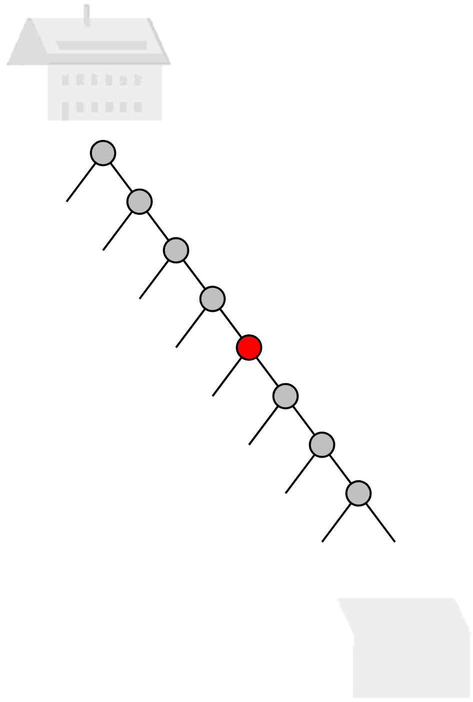




q= 0.205

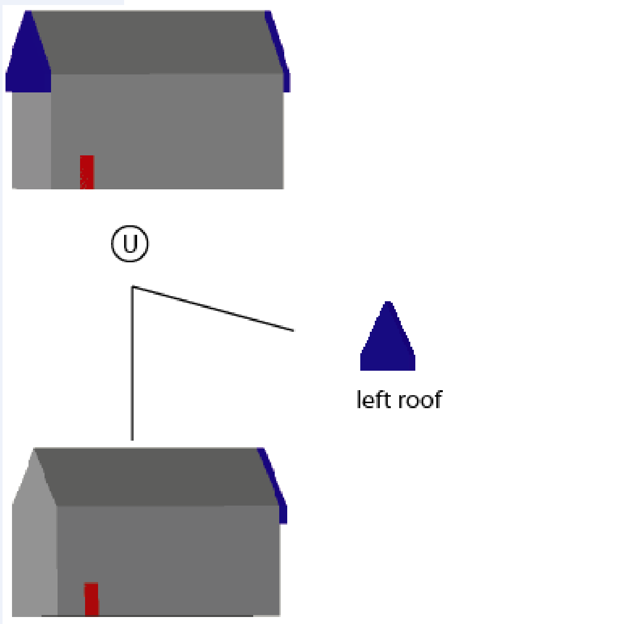
Sample building - Step 5

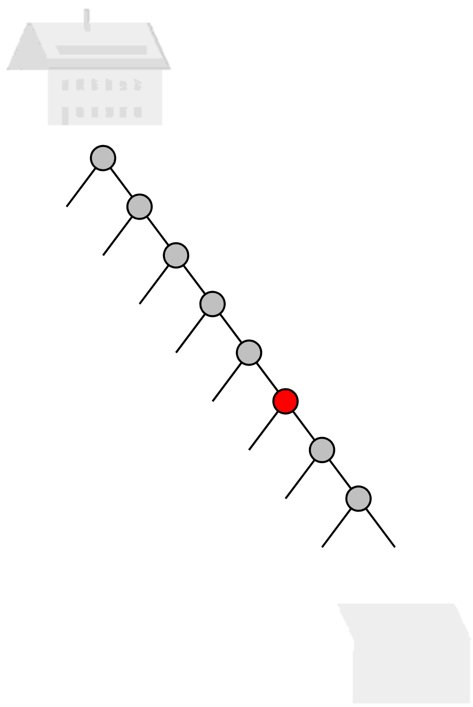




q= 0.210

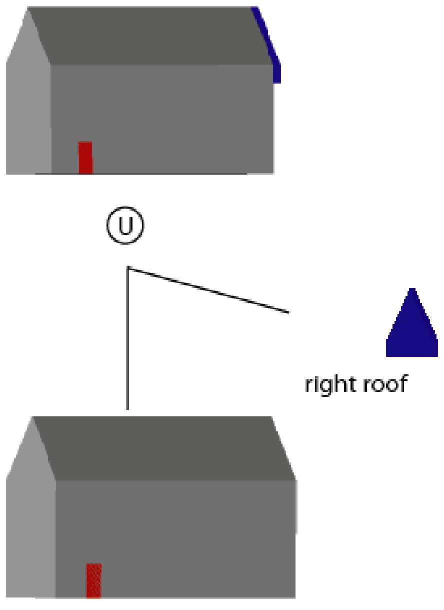
Sample building - Step 6

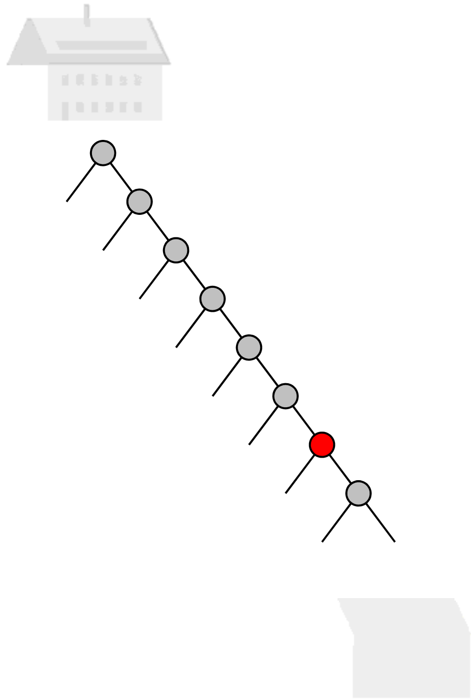




q= 0.517

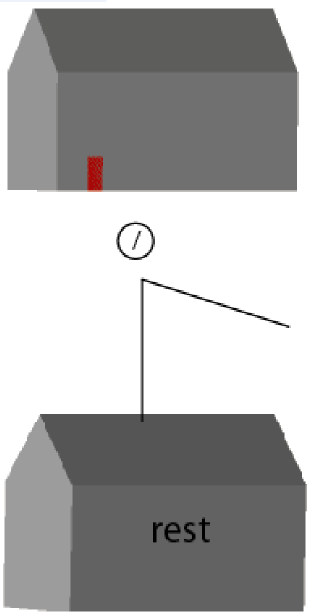
Sample building - Step 7

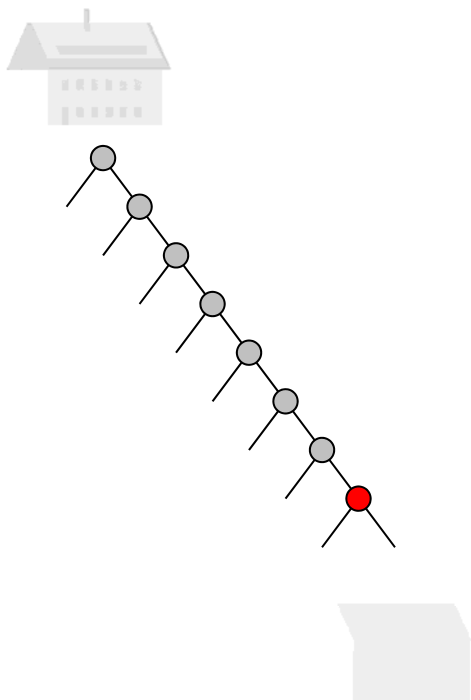




q= 0.517

Sample building - Step 8





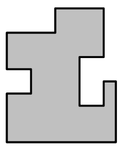
q= 0.009

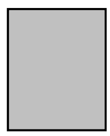
Progressive Transmission of Vector Data

Brenner & Sester

Elementary Generalization Operations

$$P \equiv P^n \equiv P^{i_0} \xrightarrow[g_0]{} P^{i_1} \xrightarrow[g_1]{} \ldots \xrightarrow[g_{k-1}]{} P^{i_k} \equiv P^m$$



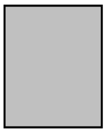
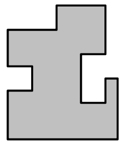


or $\varnothing$

- Generalization chain
  - Maximum representation: $P^n \equiv P$
  - Minimum representation: $P^m \quad m \leq n$
  - Number of polygon edges: $i_0 = n \quad \ldots \quad i_k = m$
  - Elementary generalization operations: g_j

Elementary Generalization Operations

$$P^m \equiv P^{i_k} \xrightarrow[g_{k-1}^{-1}]{} P^{i_{k-1}} \xrightarrow[g_{k-2}^{-1}]{} \ldots \xrightarrow[g_0^{-1}]{} P^{i_0} \equiv P^n$$

 or $\varnothing$ 

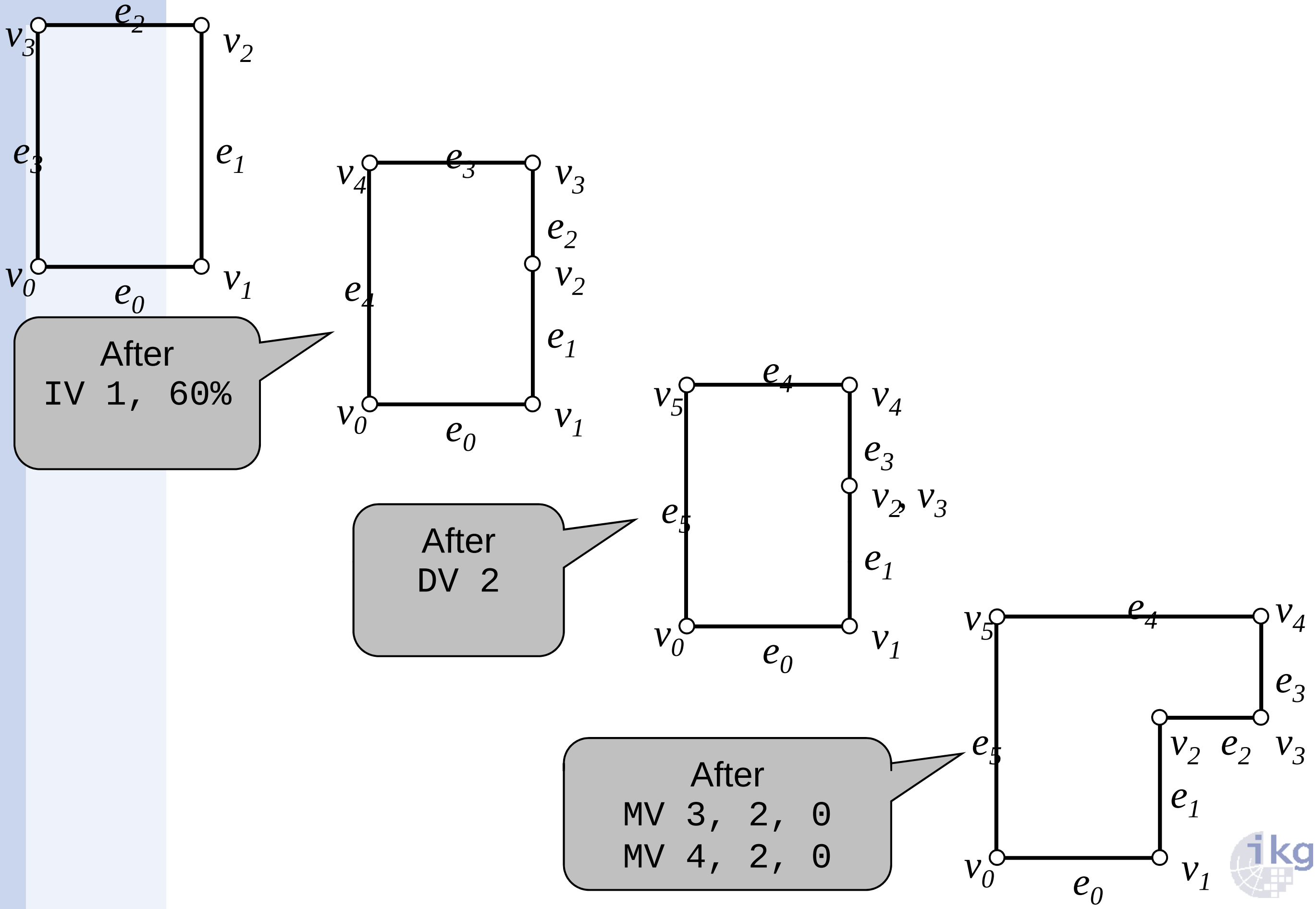
- Inverse generalization chain
  - Pre-computed
  - Can be used for progressive transmission
  - Each g_j^{-1} is associated with a parameter ε_j
  - Parameters ε_j are decreasing (inverse chain)

Simple Operations

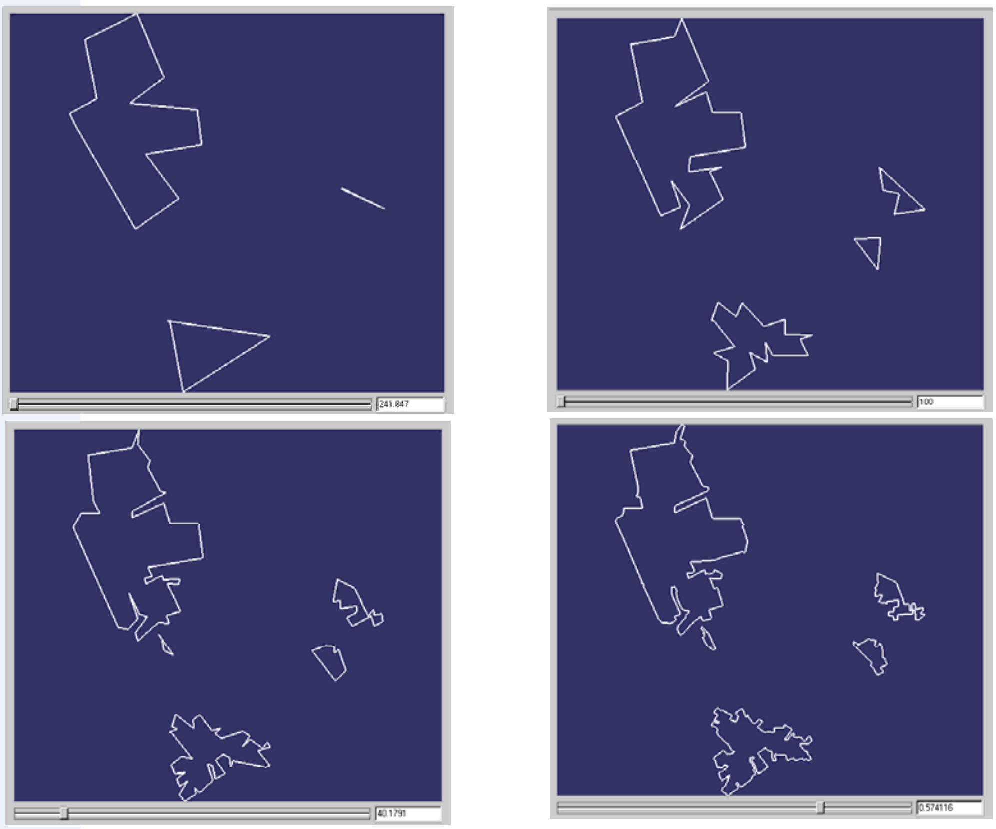
- EGO's can be subdivided into simple operations (SO's)
- SO's may modify geometry or topology

Opcode	Description	Parameters	Inverse Operation
IV	Insert Vertex	IV <edge id> <rel. position>	RV <edge id + 1>
DV	Duplicate Vertex	DV <vertex id>	RV <vertex id + 1>
MV	Move Vertex	MV <vertex id> <dx> <dy>	MV <vertex id> <-dx> <-dy>
RV	Remove Vertex	RV <vertex id>	–

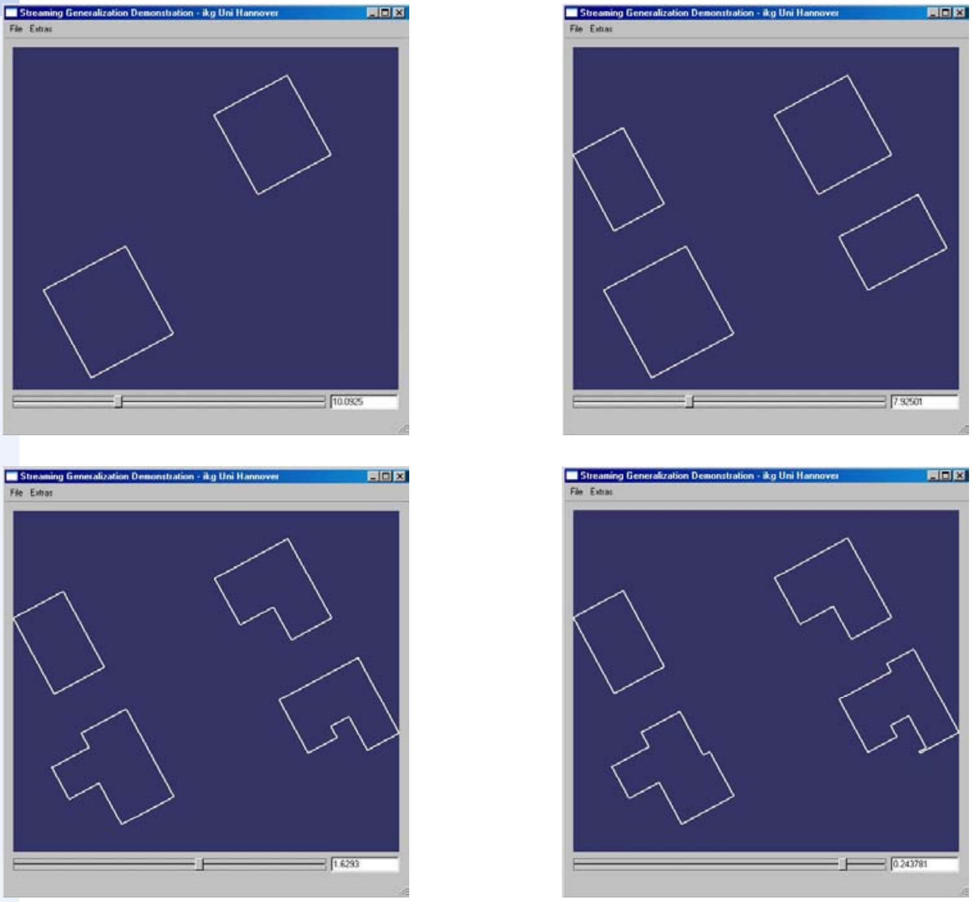
Simple Operations: Example



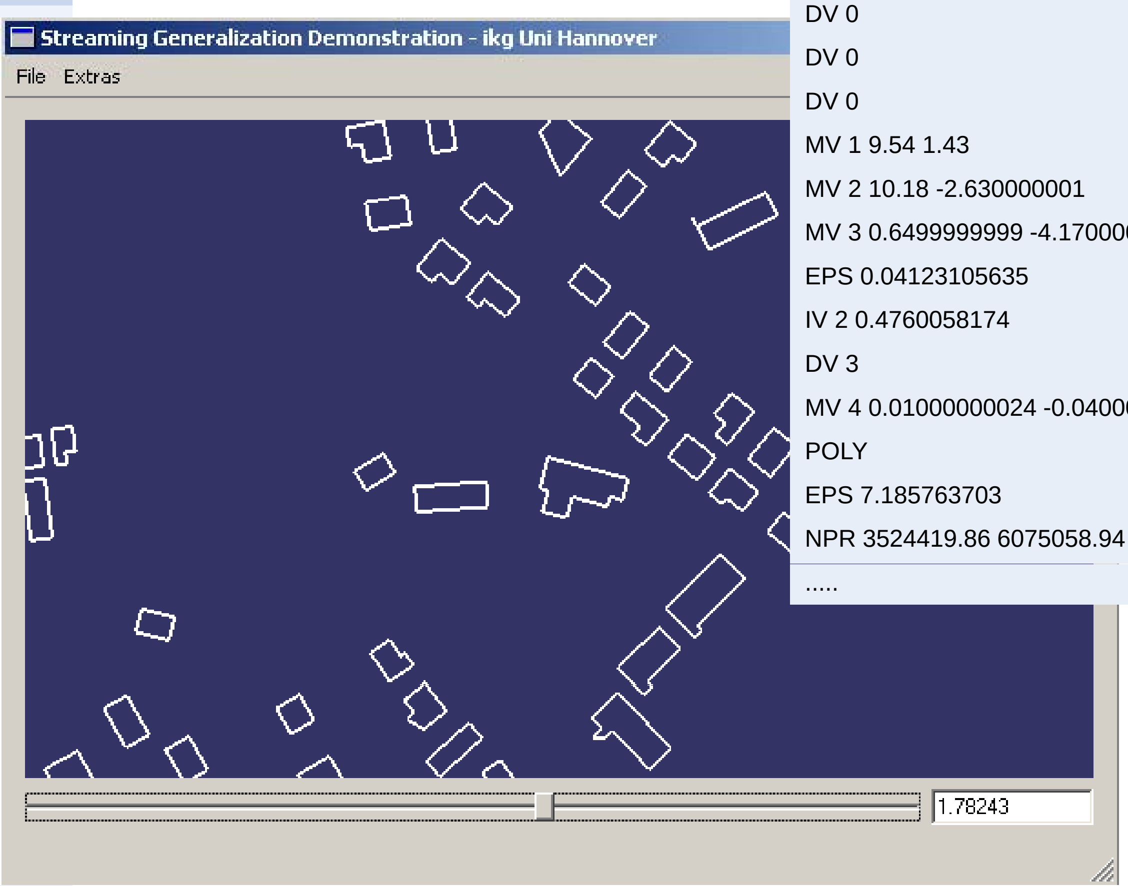
Generalization of polygons using Douglas-Peucker



Generalization of buildings



Continuous generalization



POLY
EPS 4.110133818
NPR 3524488.89 6074992.44
DV 0
DV 0
DV 0
MV 1 9.54 1.43
MV 2 10.18 -2.630000001
MV 3 0.6499999999 -4.170000001
EPS 0.04123105635
IV 2 0.4760058174
DV 3
MV 4 0.01000000024 -0.04000000004
POLY
EPS 7.185763703
NPR 3524419.86 6075058.94
.....

Generalization

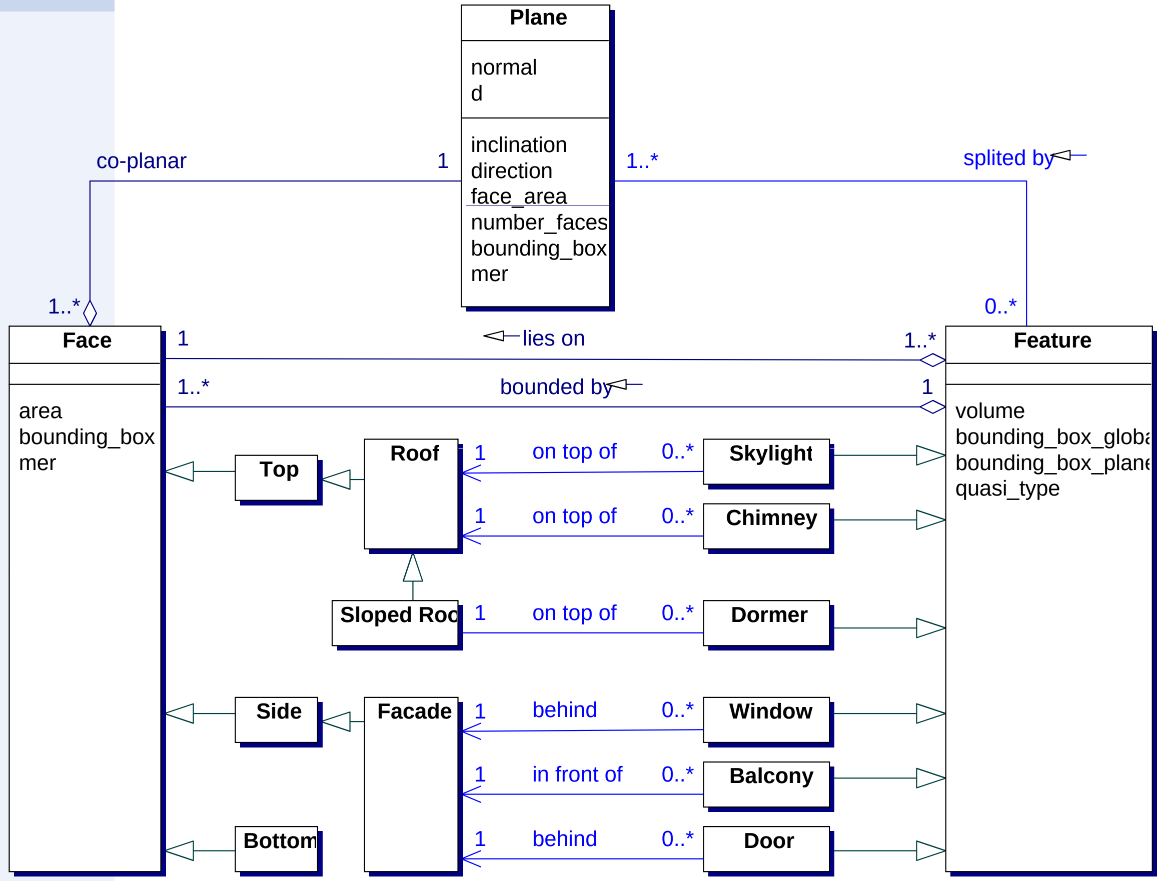
Two different goals:

1. Simplification by removal of small (unimportant) details
   → reduction of data
   → Needed: decision of what is – visually – unimportant

2. (cartographic) generalization using emphasis, aggregation, symbolization, typification:
   → Important aspects / parts are still visible (even if they are geometrically too small)
   → Prerequisite: interpretation of (important) parts

Semantic based generalization

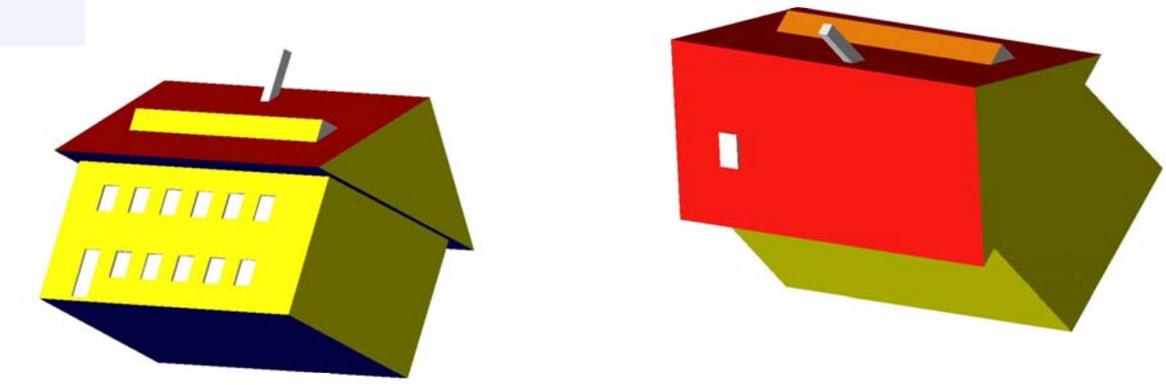
Interpretation - Model of a Building



Interpretation Strategy

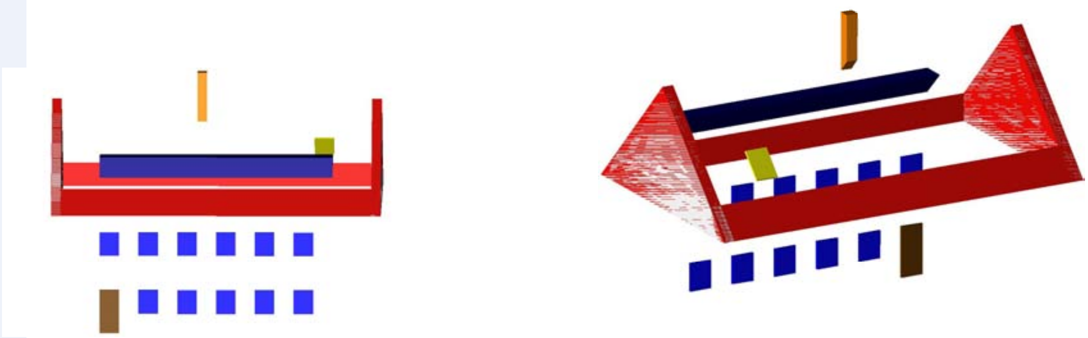
- First: interpretation of faces
  - Top: roof
  - Side: wall / façade
  - Bottom
- Second: interpretation of features related to faces
  - Roof: chimney, skylight
  - Façade: window, balcony, door

Results – Classification of faces I



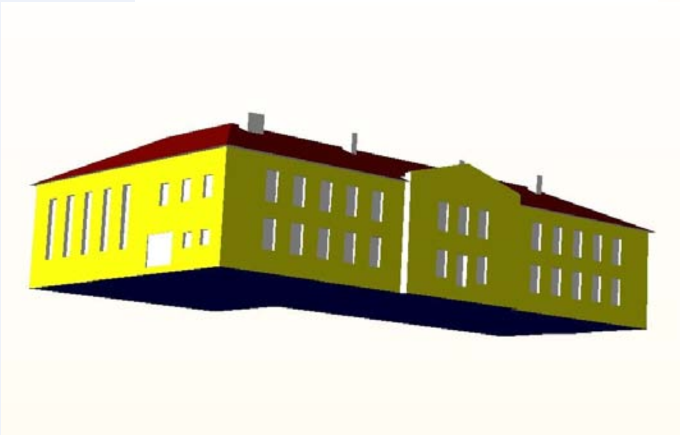
facade | flat roof | sl. roof | bottom

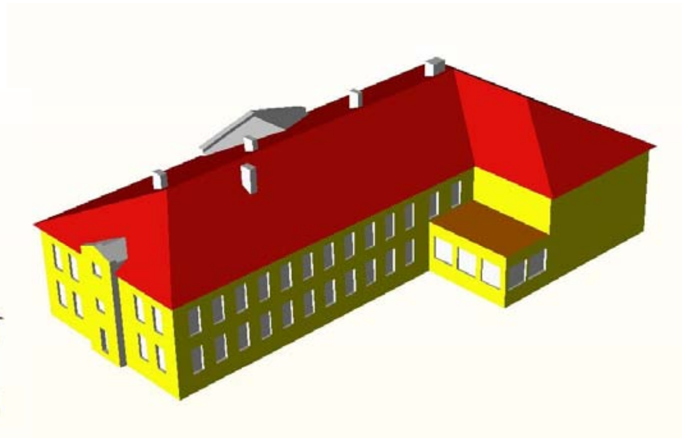
Results – Classification of features I



window	roof projection	chimney
door	protrusion	skylight
set-off	gap	dormer

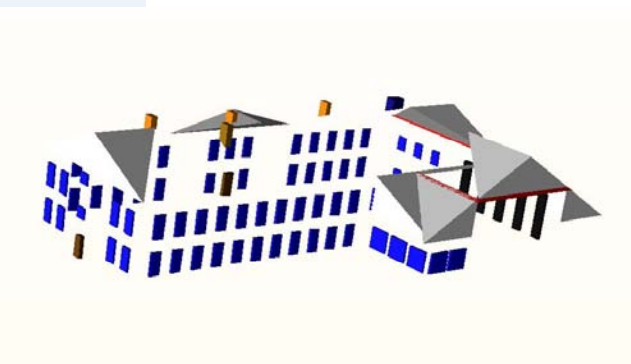
Results – Classification of faces II

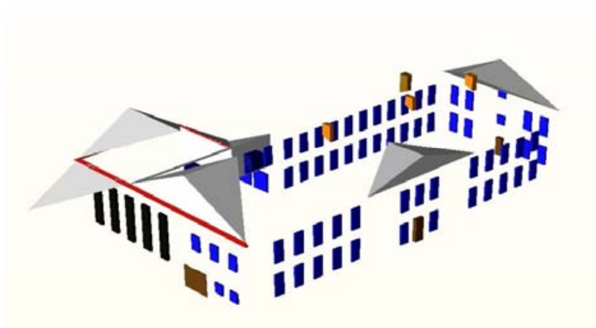




facade	flat roof	sl. roof	bottom

Results – Classification of features II

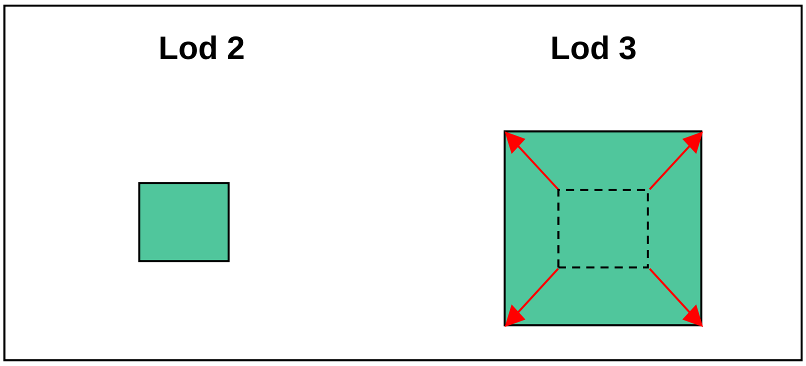




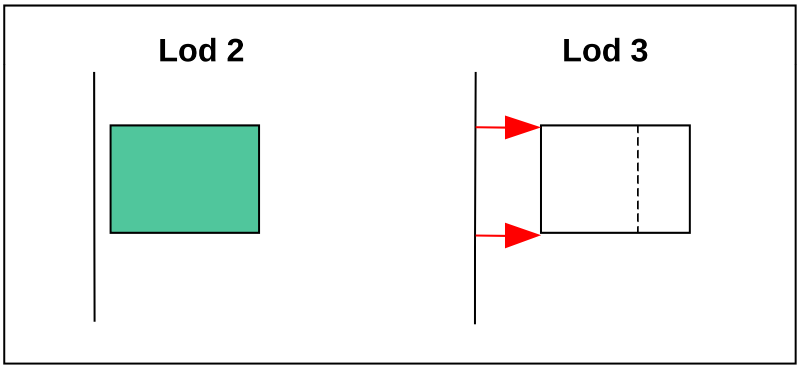
window	roof projection	chimney
door	protrusion	skylight
set-off	gap	dormer

Generalization operations

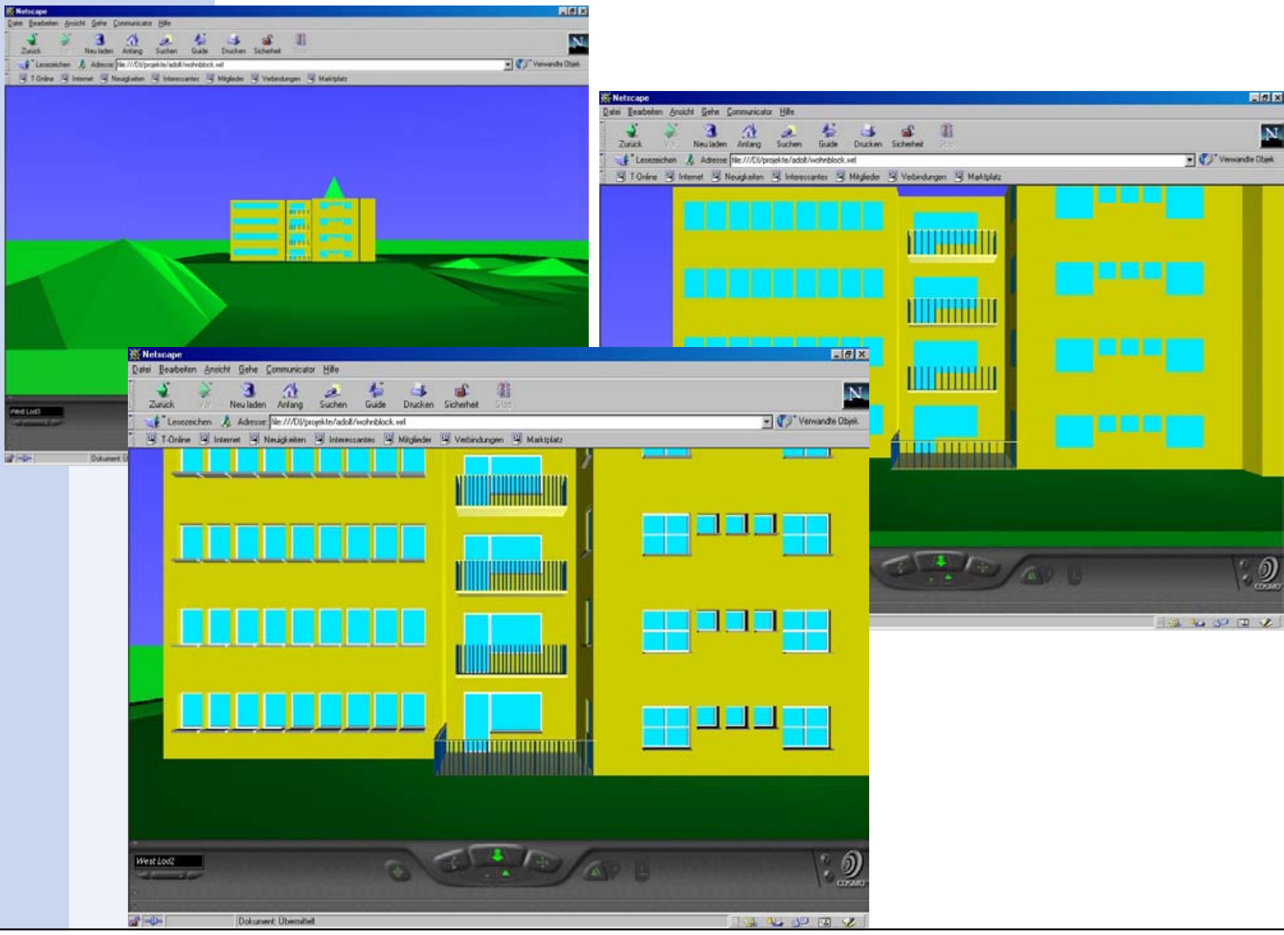
- exaggeration



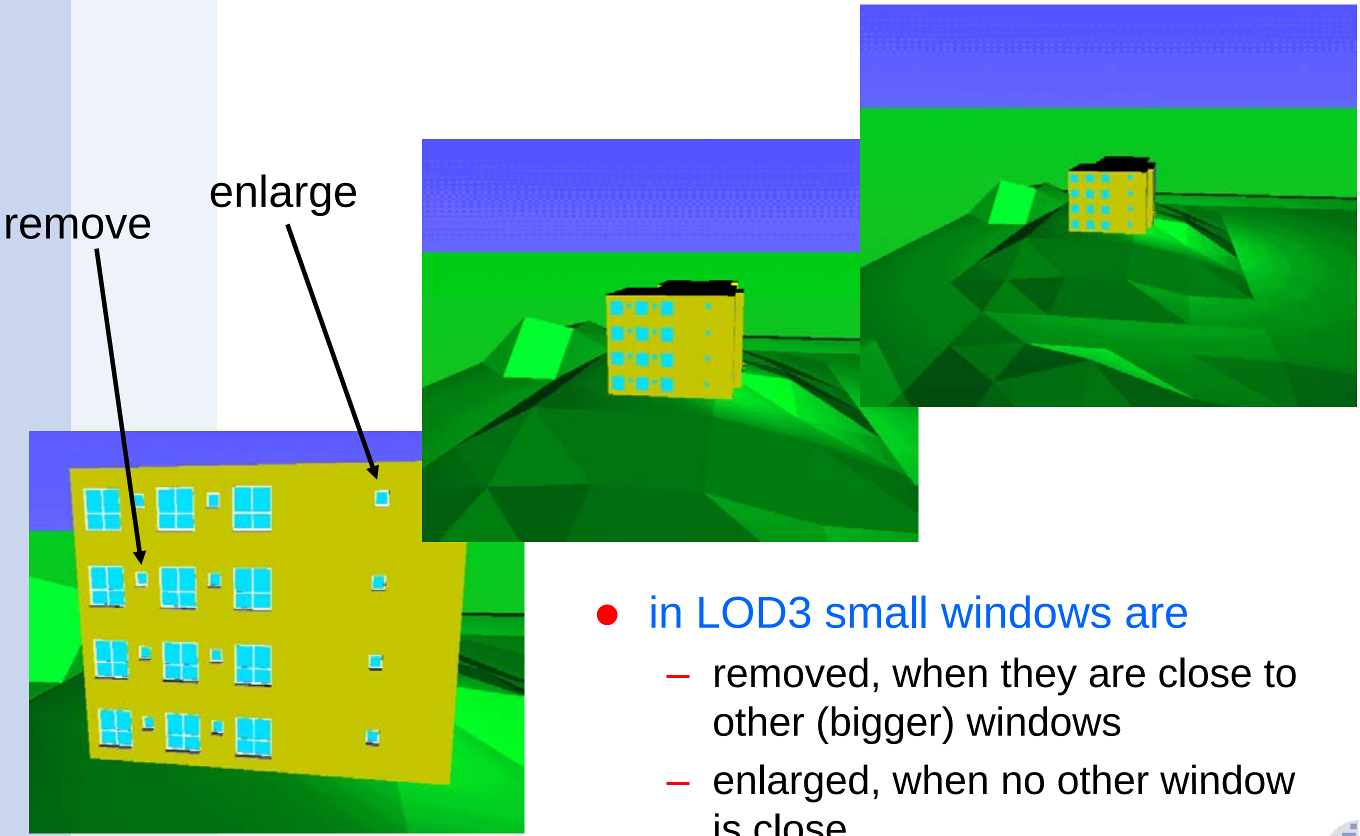
- displacement



Scale dependent representation of buildings



Context dependent removing and enlarging objects



- in LOD3 small windows are
  - removed, when they are close to other (bigger) windows
  - enlarged, when no other window is close

Generalization using hierarchical structural description

Example: façade reconstruction

(Nora Ripperda, ikg)

Typical facades (in Germany, in Hannover, in the northern part of the city…)

Example: Partitioning of a facade

- Recursive subdivision

Façade

Upper part

Ground floor

Example: Partitioning of a facade

Symmetric façade

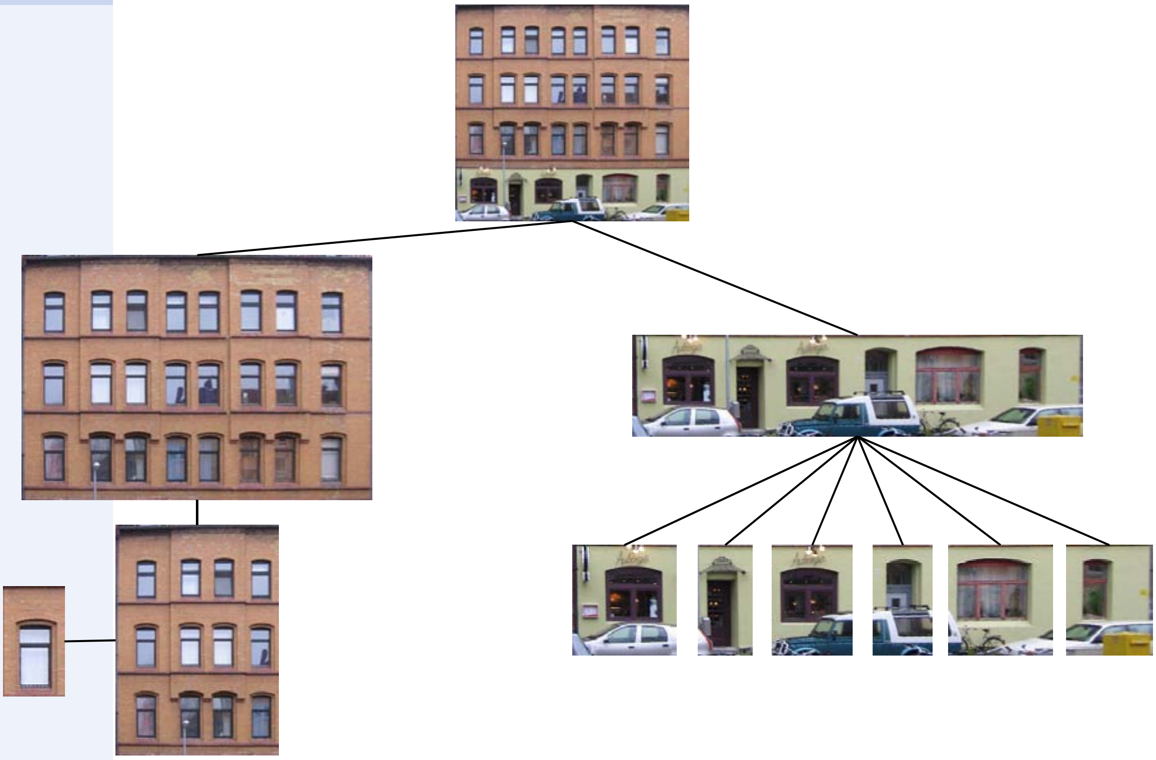
Windows in regular grid

Window

Ground floor

Façade elements

Example: Partitioning of a facade



Grammar for façade modelling

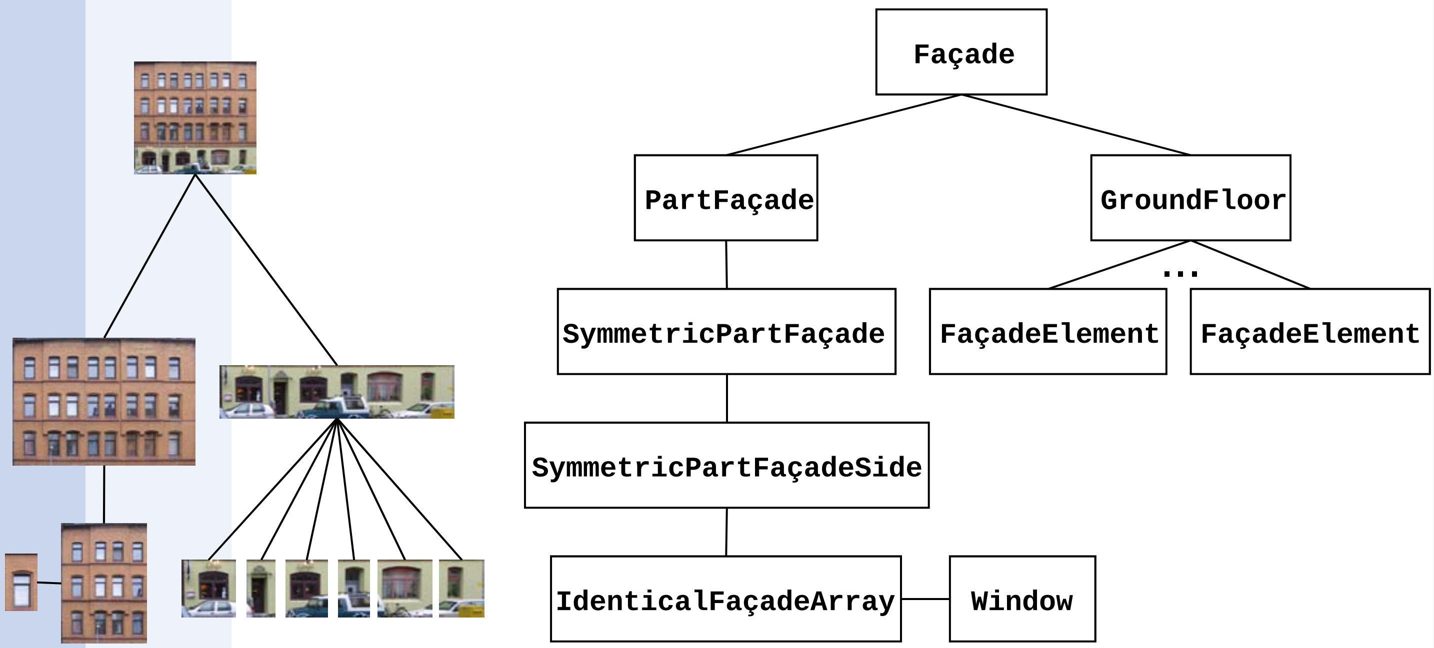
- Grammar describes façade
- Symbols
  - Nonterminals (here: Container)

`AboveDoor`	`FaçadeRow`	`SymmetricPartFaçade`
`AboveWindow`	`Gable`	`SymmetricPartFaçadeMiddle`
`Façade`	`GroundFloor`	`SymmetricPartFaçadeSide`
`FaçadeArray`	`IdenticalFaçadeArray`	`SymmetricPartFaçade`
`FaçadeColumn`	`PartFaçade`	`SymmetricPartFaçadeMiddle`
`FaçadeElement`	`StaircaseColumn`	`SymmetricPartFaçadeSide`

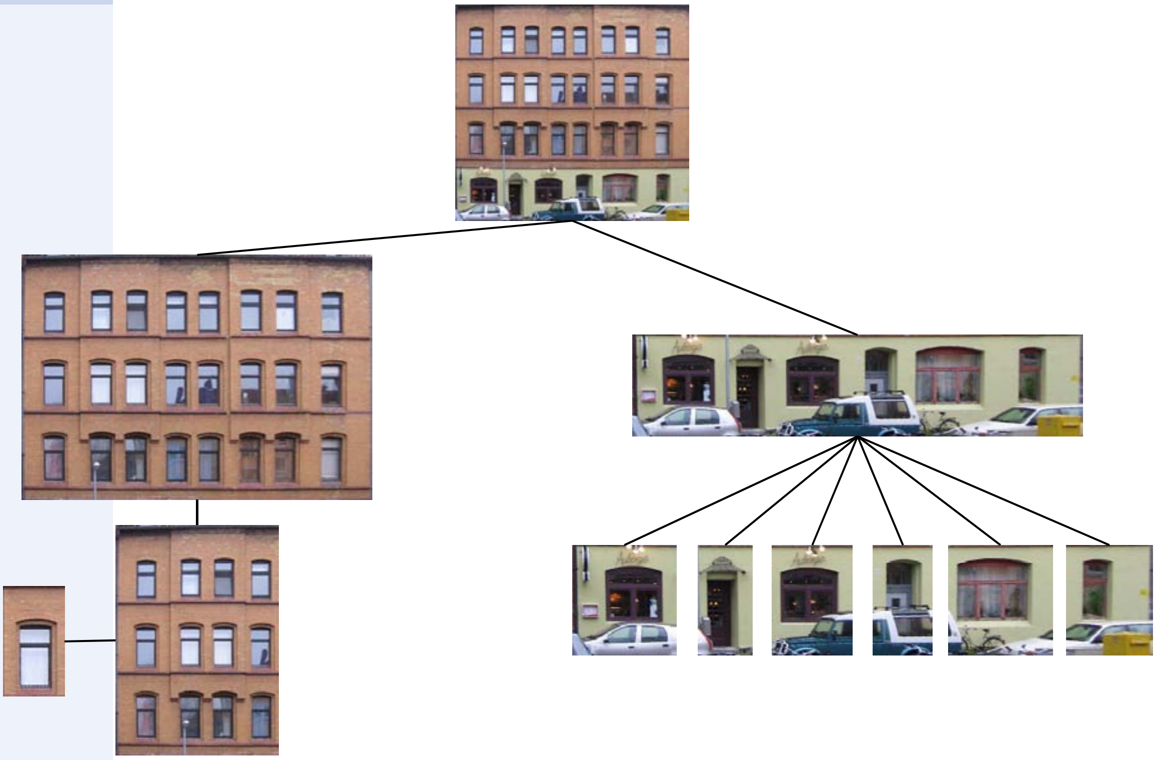
  - Terminals

`Door`	`StaircaseWindow`	`Window`
`DoorArch`	`Wall`	`WindowArch`

Derivation tree for a specific facade

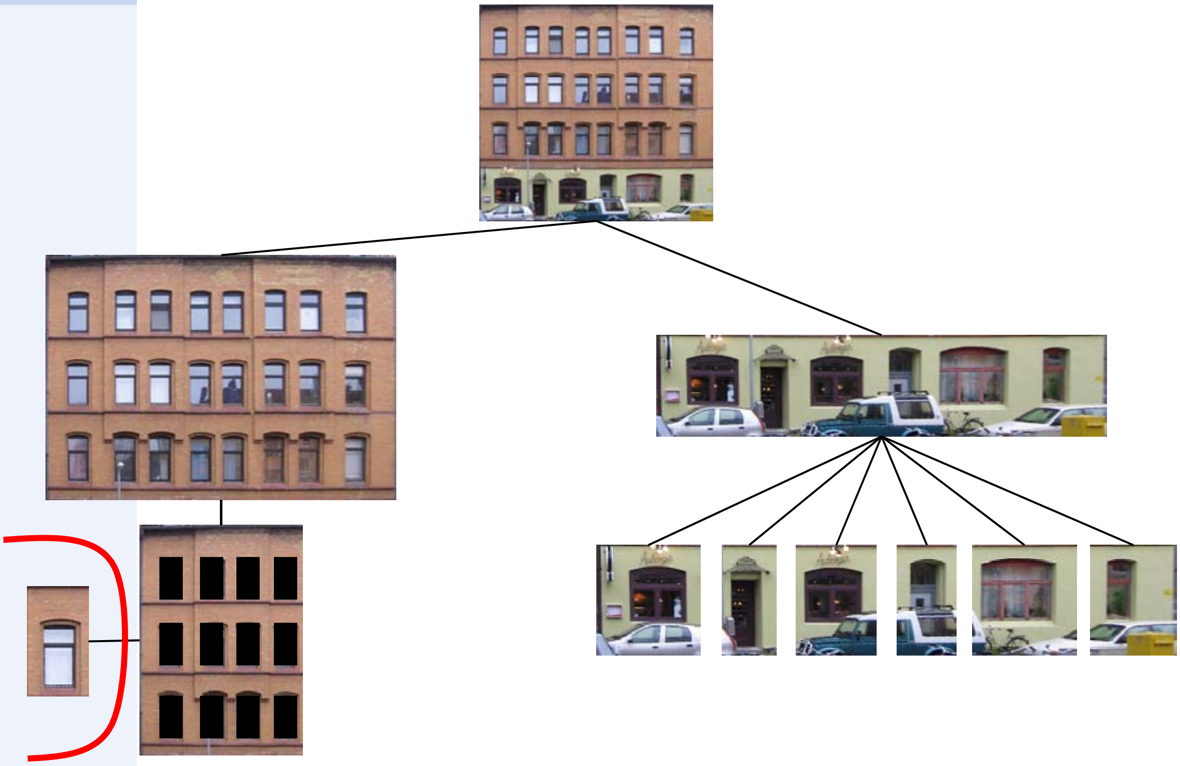


Use derivation tree for generalisation

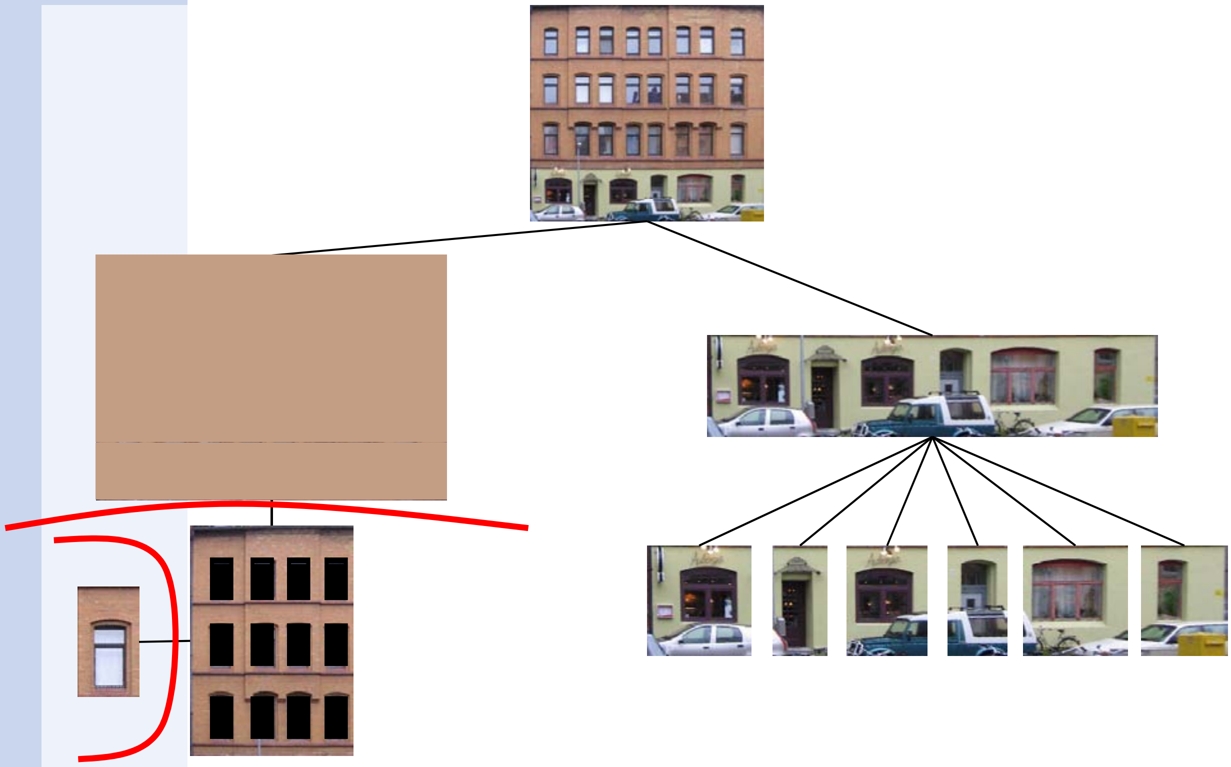


[Claus Brenner]

Use derivation tree for generalisation



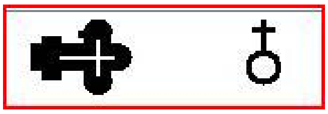
Use derivation tree for generalisation



More elaborate rules possible, e.g. typification

3D-Templates / Prototypes

- Idea:
  - For landscape visualization, **important** objects have to be highlighted in a way that they can be easily recognized
- In cartography:
  - Enhancement / Enlargement
  - Symbolization



- Transfer to 3D:
  - use of 3D symbols / prototypes

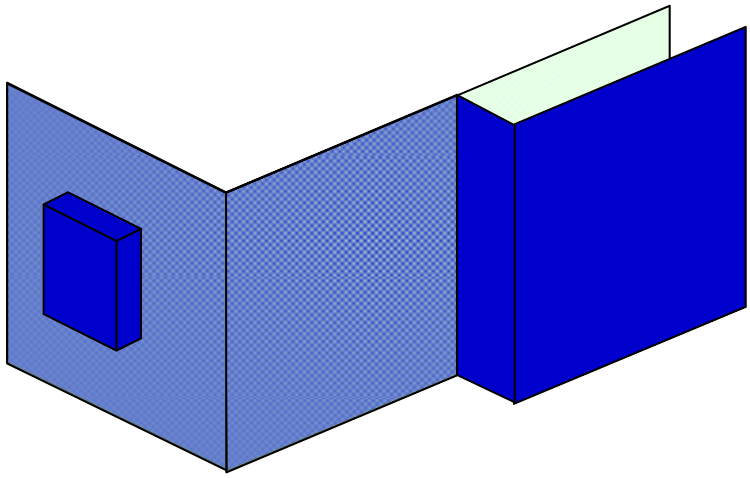
[Thiemann & Sester, 2005]

3D-Templates / Prototypes

- Approach:
  - SIMPLE: Use generic 3D symbol (e.g. for church, town halls,…)
  - MORE CHALLENGING: use symbol which fits to (and resembles) 3D-shape of the original object, e.g.
    - Small church is smaller than others
    - church with two towers is represented as such
- Requirements:
  - Definition of prototype:
    - necessary parts of objects (church: body, tower, bells, …)
    - connectors between these parts (tower on top of roof, next to body, …)
  - Identification of these parts in object
  - Enhancement of important parts and adaptation to original object

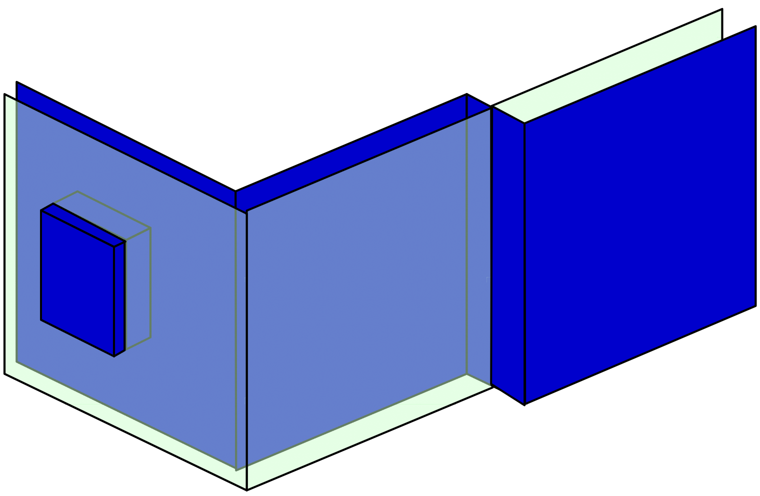
Adaptation of model/template to original object

- Given:
  - Original object (in blue)
  - Template object (in light green)
- Method:
  - Adjustment process: minimizing the distances between template and original object



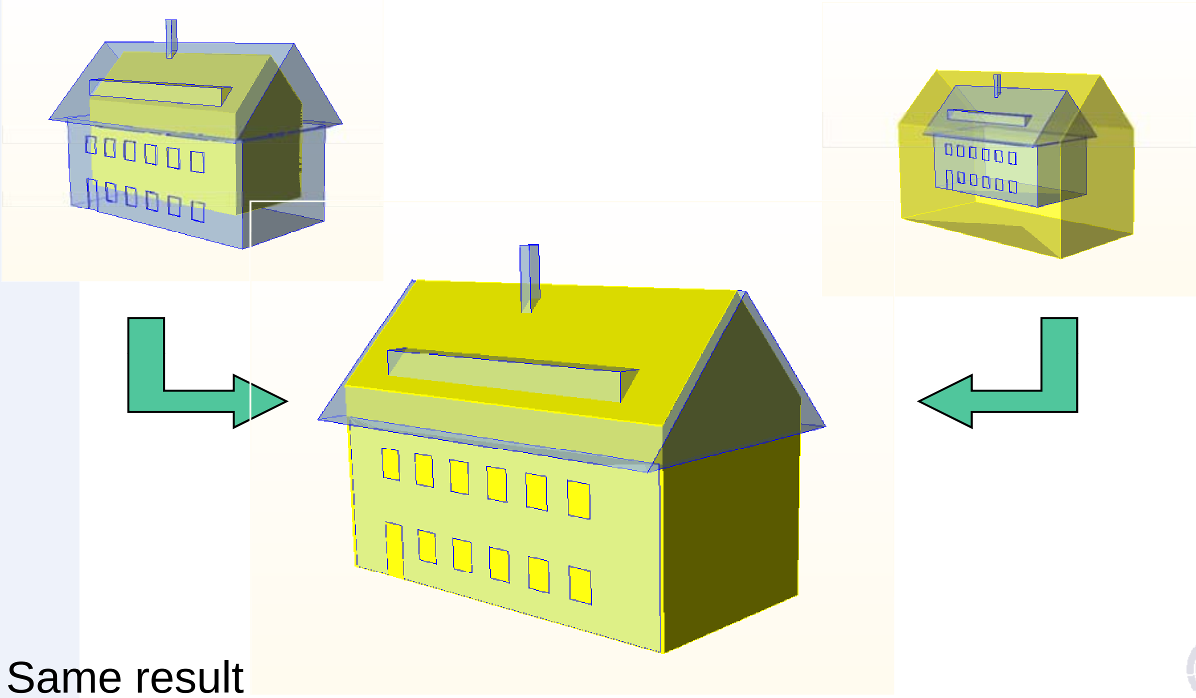
Adaptation of model/template to original object

- Given:
  - Original object (in blue)
  - Template object (in light green)
- Method:
  - Adjustment process: minimizing the distances between template and original object



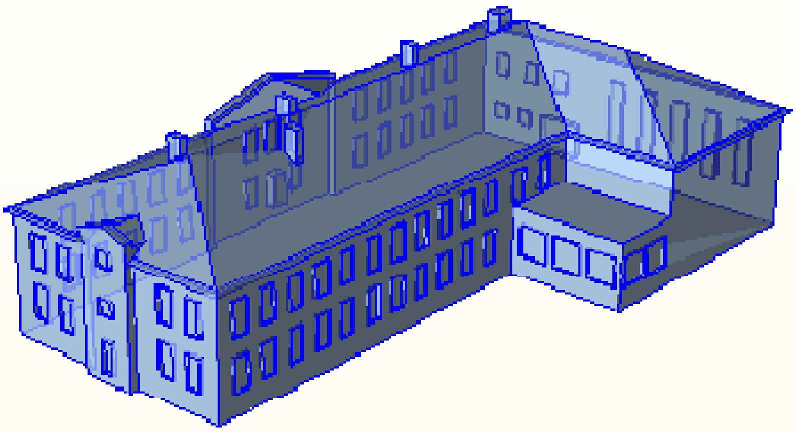
3D-Adaptation

- Adaptation of 3D-template (yellow) to original building (blue)
- Starting with a small or a large template

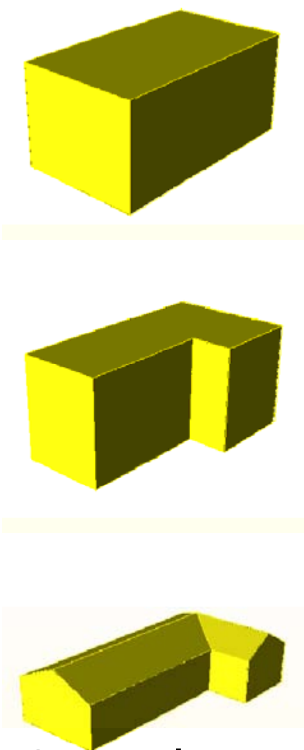


- Same result

Example L-shaped building

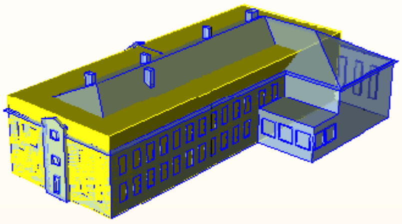


original object



3 templates

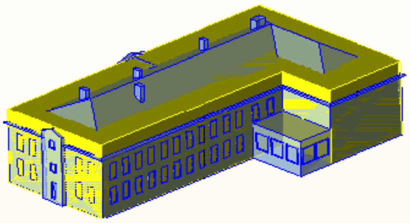
Results (distances from template to original)



I-shape (cuboid)

6 faces

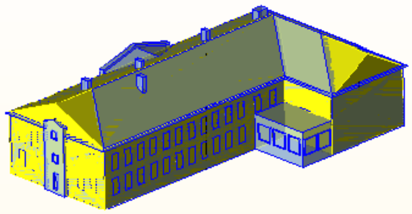
std.deviation: 0.85 m
volume diff.: -1130 m³
(15%)



L-shape flat roof

8 faces

std.deviation: 0.64 m
volume diff.: -260 m³
(3%)



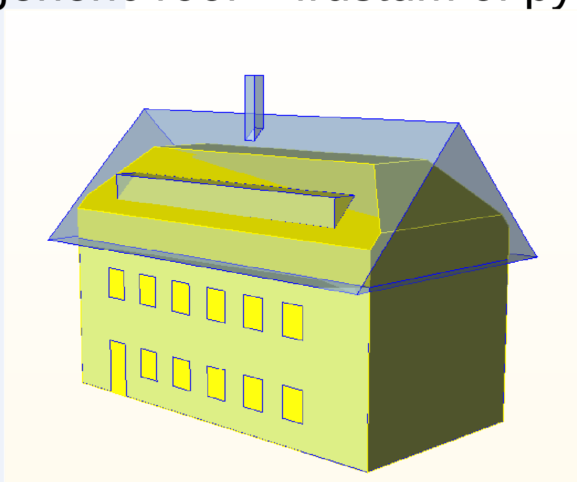
L-shape gable roof

11 faces

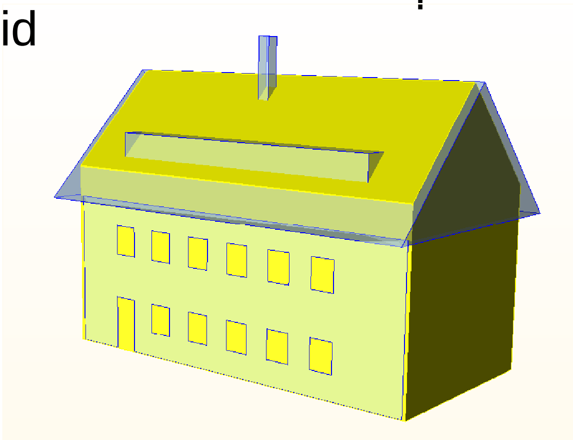
std.deviation: 0.42 m
volume diff: -182 m³
(2%)

Generic roof – “Berliner Dach”

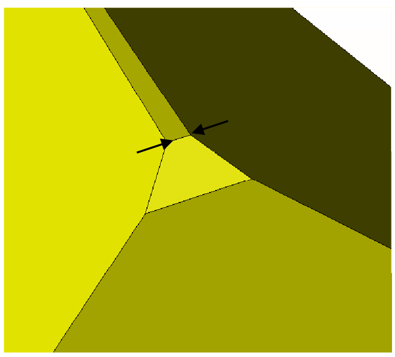
Approximation:
generic roof = frustum of pyramid



-> After adaptation



Detail
(width of ridge: 6cm)



Comparison of approaches

Comparision of approaches for 3D-generalization

Criterion / Characteristics	Kada	Forberg	Thiemann
Handling of roof and façade structures	Yes (Kada 2006)	Yes	Yes
Fixed target scale	Yes	Yes	Yes
Continuous scales	No	No	Yes
Application to highly redundant data (point clouds)	Yes (Kada 2006); problem is the determination of approximate planes	No, as system expects parallel planes	No in Thiemann 2002, system generates all potential cutting planes; yes: possible with Thiemann 2006
Generalization of neighboring buildings	In principle possible, as planes can be extended to neighboring buildings	In principle possible, as parallel shifts can be extended to neighboring buildings	No
Extension to other generalization operations	No	No	Yes, possible, as semantics of object parts is determined

Summary

- Increasing availability of 3D objects
- Increasing demand for 3D-city models (e.g. navigation systems, Google Earth, Microsoft Virtual Earth)

- Generalization is needed to make data handling manageable

- Flexible (semantic) data structure is available: CityGML
- Promising approaches are available for generalization of individual buildings

Outlook

- Need for
  - Other generalization operations (aggregation, displacement, typification, ...)
  - Methods for generalization of groups of buildings are needed

- Integration of other 3D-objects (terrain, bridges, ...)

- Use / exploit intelligence used in generation / extraction process also for generalization

- Concepts for update