Techniques for Generalizing Building Geometry of Complex Virtual 3D City Models

Tassilo Glander, Jürgen Döllner

Hasso-Plattner-Institute at the University of Potsdam
{tassilo.glander, doellner}@hpi.uni-potsdam.de

Abstract

Comprehensible and effective visualization of complex virtual 3D city models requires an abstraction of city model components to provide different degrees of generalization. This paper discusses generalization techniques that achieve clustering, simplification, aggregation and accentuation of 3D building ensembles. In a preprocessing step, individual building models are clustered into cells defined by and derived from its surrounding infrastructure network such as streets and rivers. If the infrastructure network is organized hierarchically, the granularity of the cells can be varied correspondingly. Three fundamental approaches have been identified, implemented, and analyzed: The first technique uses cell generalization; from a given cell it extrudes a 3D block, whose height is calculated as the weighted average of the contained buildings; as optimization, outliers can be managed separately. The second technique is based on convex-hull generalization, which approximates the contained buildings by creating the convex hull for the building ensemble. The third technique relies on voxelization, which converts the buildings' geometry into a regular 3D raster data representation. Through morphological operations and Gaussian blurring, aggregation and simplification is yielded; polygonal geometry is created through a marching cubes algorithm. The paper closes with conclusions drawn with respect to the characteristics and applicability of the

presented generalization techniques for interactive 3D systems based on complex virtual 3D city models.

1 Introduction

A virtual 3D city model represents both a technical and conceptual framework to assemble, integrate, present, and use 3D geoinformation as well as for 3D geovisualization [8]. Besides their application in GIS, they provide an effective user interface to complex spatial 3D information in a growing number of IT applications and system such as enterprise systems, navigation, telecommunication, disaster management, simulators, and e-government. In particular, virtual 3D city models facilitate the integration of heterogeneous 2D and 3D geo data, and their interactive visualization offers comprehensible and efficient communication, exploration, and analysis of complex geoinformation. In addition, 3D city models form part of administrative geoinformation databases, services, and infrastructures [7].

A common problem for implementation and usability of virtual 3D city model systems arises from their complexity with respect to the number of individual components, their computer graphics representations, and the rendering resources [2][35]. A typical 3D city model consists of several hundreds of thousand objects, including models of buildings, vegetation, and infrastructure elements. To achieve efficient rendering and interactive frame rates, level-of-detail geometry representations are required that control the polygon count and texture resources [3]. Furthermore, the 3D city model needs to be represented at different generalization levels, for example, to enable context-&-detail views, to enhance comprehensibility of depictions, and to support hierarchical navigation and browsing.

Our investigation addresses a fundamental problem of today's visualization of fine granular, complex 3D city models: Their visualization frequently shows "noise" and "flickering" that appears in areas that are far away from the view point because 3D objects are mapped to few pixels or even only to a fraction of a pixel. Similarly, pedestrian or car driver perspectives tend to produce a "sea of houses" beyond a certain distance, making it impossible for the user to identify these areas. Furthermore, abstract information, such as hierarchy information such as whole districts and quarters of a city, is not explicitly visible if the full model resolution is used for depiction. All these phenomena result because *information density* of the 3D city model is not adjusted.

As a key technique to control information density, generalization both helps to reduce the graphics complexity as well as the cognitive complex-

ity of 3D city models. Similar to maps, features of city models should be visualized at different scales to accomplish different spatial tasks. In addition, "[…] this abstraction and concentration also helps to discern between relevant and irrelevant information: only those objects have to be presented, that are important for the current task – irrelevant information can be suppressed." [29] Traditionally, generalization of map objects requires experienced cartographers, who are using the human ability to abstract. For 2D vector data as well as for 3D objects, downscaling does not suffice because readability and comprehensibility have to be preserved [13]. These requirements are more important than exactly scaled geometry or its exactly preserved appearance. For this reason, generalization does not only simplify objects but also deforms, drops, aggregate, classifies, or unifies objects and their features. That is, generalization techniques apply operations such as simplification, aggregation, classification, and displacement.
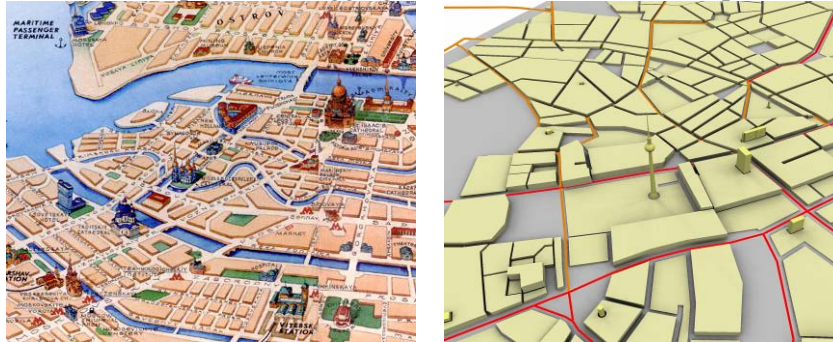

Fig. 1: Example of artistic 3D city map of St. Petersburg[1] (left) and an automatically generalized 3D city model of Berlin, generated by the presented cell-based method (right).

This paper concentrates on generalizing 3D building ensembles as one main category of city model objects. The presented techniques first cluster 3D building models into ensembles according to a given hierarchical street and infrastructure network (e.g., car navigation data) and then automatically generalize the ensembles in the underlying cells defined by the network (Fig. 1). Three different approaches, cell-based, convex-hull-based, and voxel-based generalization algorithms, are outlined and evaluated.

Our work emphasizes the generalization of clustered 3D building models in contrast to various 2D footprint generalization algorithms as well as various 3D building generalization techniques that simplify and abstract single 3D building models. Furthermore, our core goal is to automatically derive generalized building ensembles at various levels of granularity, op-

---

[1] From www.escapetravel.spb.ru

timized for using these generalized models in interactive 3D city model visualization in contrast to generalized 2D map production.

2 Related Work

MENG and FORBERG [24] give an overview of state-of-the-art and challenges of 3D building generalization, describing the current scale space of 3D buildings as "a linear continuum, along which an arbitrary number of milestones can be said to exist referred to as Levels of Detail (LoD)". The LoD provide different representations of the buildings with different degrees of generalizations. However, there exist different classifications of the LoD, so no standards compared to the scales in cartography have been established yet. Elementary 2D map generalization approaches are described, e.g., by [13][14][30].

Among the first techniques for 3D building generalization, the application of morphological operations on 3D geometry was suggested by MAYER [22][23]: a curvature space simplification has been developed, which detects local curvature and shifts the adjacent polygons accordingly. Both methods apply to specific geometry structures but are costly in terms of processing time. In [11] generalization is based on moving near parallel faces of building geometry to a common plane and merging them if possible. Unfortunately, the algorithm requires orthogonal buildings to work. An automated algorithm for generalizing 3D building geometry is described in [26].

The feature removal algorithm of RIBELLES et al. [27] was applied on buildings by THIEMANN [33] to create a constructive solid geometry (CSG) representation of the given building geometry. It uses the planes of the building's faces to subdivide the geometry into a body and features, which can be integrated or left out of the generalized representation, depending on the degree of generalization. Similarly, KADA [17] uses a few approximating planes to remodel the building with simpler geometry. Another technique for LoD creation [26] flattens roofs and merges adjacent polyhedra, followed by collapsing facades while respecting visually important walls.

While these 3D generalization approaches focus on single buildings, aggregation of multiple buildings currently is usually referred to as the next important step. STÜBER [31] presents a framework for generalization of building models while preserving visual correctness. This approach simplifies buildings based on feature removal and aggregates buildings depending on their visibility. However, automatic aggregation appears to be limited to simple configurations.

Motivated by classical cartography, ANDERS [1] applies generalization techniques on 2D projections of linear building groups. For each of the main axes' projections, the shapes are aggregated and simplified using a specific generalization technique [28]. The simplified shapes are extruded and intersected to form the generalized building group. The approach achieves aggregation and simplification, however it is limited to linear building groups. SESTER [29] suggests a 3D visualization providing simplification, aggregation, displacement and enlargement by extruding the processed ground plans to a certain height. In addition, the height can be used to further emphasize special buildings in for pedestrian navigation.

CityGML [5], a proposed interchange format for virtual 3D city models currently discussed by the Open Geospatial Consortium (OGC), differentiates five consecutive levels of detail (LOD-0 to LOD-4) [18], where objects become more detailed with increasing LOD regarding both geometry and thematic differentiation. CityGML models can contain multiple representations for each object in different LOD simultaneously but does not address the way these LODs are created or transferred. In [6] a continuous level-of-detail concept for individual building models has been introduced, but no automated derivation of generalized building ensembles is considered.

Real-time 3D rendering relies on efficient treatment of polygonal 3D data sets, and it provides a variety of LoD techniques, which can be classified into static and dynamic techniques in general. Static techniques provide discrete LoD representations (e.g., [9][19][12]), whereas dynamic techniques transform polygonal surfaces partially according to the current viewing situation (e.g., [16]). Common to all techniques, the original high-resolution 3D object is simplified such that its appearance is preserved, but it is not generalized nor do the techniques consider specific semantics or characteristics of the type of the 3D object to be simplified.

3   Generalization Techniques for 3D Buildings

In this section, three generalization techniques are presented that are primarily based on simplifying and aggregating 3D building geometry. Inspired by classical city plans and bird's eye views (Fig. 1), these techniques achieve generalized 3D building ensembles that facilitate context views of geovirtual 3D environments.

As a control parameter, we introduce the term *degree of generalization* (DoG) in contrast to level of detail (LOD). While LOD usually refers to simplified representations of a single object, DoG describes the level of abstraction, which allow us, for example, to represent a group of

neighboring buildings by a single block. In addition, to achieve a simplified and comprehensible visualization, we do not use façade textures that would amplify the visual complexity.

The input data include 3D building models and 2D vector-based, hierarchical street and infrastructure networks (e.g., streets, rivers). In the following examples, part of the Berlin 3D city model (Fig. 2) is used for illustration together with a navigation street data set. The streets are attributed by weights, which differentiate four street types. The weights are used by the techniques to define the streets' width.

Fig. 2. Snapshot from the visualization of part of the 3D city model of Berlin containing approx. 60,000 3D building models and approx. 6,500 streets without generalized building models.

3.1 Simple Cell Generalization

This technique aggregates all buildings within one cell defined through the enclosing network system by extruding the cell's boundary to a certain height creating a prismatic block. To leave enough space for the streets, they are buffered with a characteristic width and cut out of the block ground plan before the extrusion stage using Boolean operations. The degree of generalization (DoG) can be controlled by the hierarchy level of the streets considered for parceling.

3.1.1 Parceling

Each street is defined by a consecutive points and a weight. The Computational Geometry Algorithms Library (CGAL [4]) is used to intersect the

curves defined by the streets and to calculate the cell geometry [10][34]. After the parceling, *CGAL* supports aggregated point location queries and returns the hit cell for each point. This is done to cluster the buildings represented by their individual centroids according to shared cells. The results of this stage are cells (Fig. 3) and a mapping from each cell to a set of contained buildings, the *building ensemble*.
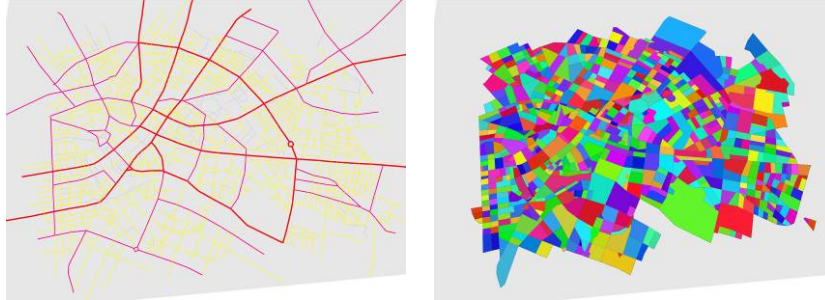

Fig. 3. Street network (left) and calculated cells (right).

3.1.2 Calculating the Cell Height

The cell height is calculated by the weighted average height of the buildings of a cell. The weight of a building can be defined by its footprint area.

Let $height(b_i)$ be the height of building b_i and $area(b_i)$ its area, then the weighted average height $\bar{h}$ of a set of n buildings is

$$\bar{h} = \frac{\sum_{i=1}^{n} height(b_i) \cdot area(b_i)}{\sum_{i=1}^{n} area(b_i)} \tag{1}$$

For low-density cells the calculated value obviously does not reflect the real situation. Instead of dividing by the sum of the buildings' area, the cell area should be used. However, in medium to high-density cells this leads to very small blocks. Therefore, if the ratio of the building area sum to the cell area gets too small, either no block should be created, or the original buildings have to be preserved.

3.1.3 Subtracting the Network Elements

The cells have to be shrunken to leave enough space for the network elements such as the streets. To accomplish this while supporting different street weights, the streets are buffered before to yield polygons (Fig. 4).

Then the adjacent street polygons are subtracted from the cell polygon using 2D Boolean operations.

Fig. 4. Vector-based network elements (left) and buffered variant (right).

3.1.4 Handling of Outlier Buildings

To improve the appearance of generalized building ensembles, outliers can be excluded from the aggregation and placed separately into the generalized version. By outliers we refer to

- **landmark buildings**, i.e., buildings that explicitly have been assigned a higher weight (e.g., landmark buildings, public buildings);
- **outlier buildings,** i.e., buildings that stand out locally as they are considerably higher than their neighborhood.

Outlier building can be determined by comparing the building height with the calculated weighted average height of the cell. To respect the characteristics of the local height distribution, the variance, respectively the standard deviation, is calculated:

$$\mathrm{var}(h_1,...,h_n) = \sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(h_i - \bar{h})^2 \ \text{ with } \ h_i = height(b_i) \tag{2}$$

The standard deviation σ can be used more intuitively as it has the same scale and units as the height. Thus, a building is considered as an outlier, if its height is larger than the average height $\bar{h}$ plus k-times the standard deviation:

$$is\_outlier(b) = \begin{cases} true & height(b) > \bar{h} + k \cdot \sigma \\ false & else \end{cases} \tag{3}$$

With $k = 2$, a reasonable identification of outliers can be done for large scales. A smaller selection in smaller scales can be achieved with bigger values for k.

3.1.5 Results

The final generalized geometry of a cell is given by extruding the cell polygon to its calculated average height. Fig. 5 shows the generalized geometry for different network hierarchy levels.

Fig. 5. Generalized cells for three degrees of generalization (no outlier handling).

The results of the simple cell generalization come close to depictions found in many 2D topographic maps. The cutout of the streets contributes most to the familiar map-look. The height of the cells observed from appropriate viewing angles gives a hint toward the real-world situation and enables relative assessment. As main advantages, this technique requires little processing time and the geometric complexity of the generalized models is low as well. The abstraction of the complex models might pose a way for content providers to offer an overview version of a 3D city model. As a disadvantage, the bare cell blocks do not preserve the appearance. In the case of top views, different heights are barely noticeable even with appropriate shading. With the additional integration of outliers and landmark buildings (Fig. 6), the effectiveness can be improved, as orientation is by far easier especially from low perspectives.
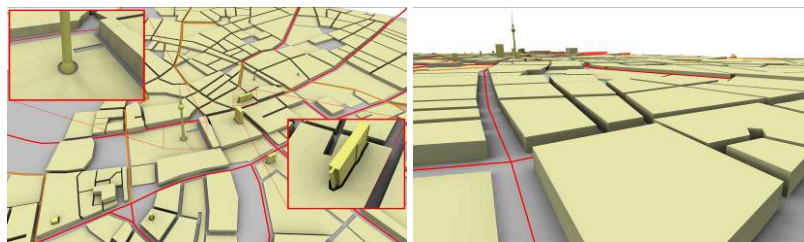

Fig. 6. Generalized cells with outlier handling (left), which are particularly important for the recognition of a city's panorama (right).

3.2 Convex-Hull-Based Generalization

This generalization technique achieves closer representations of the original buildings of an ensemble using the 3D *convex hull* as an approximation of contained buildings. Since for maps an exact representation is not the primary concern, the convex-hull operation is applied as a mean to simplify and to aggregate.

Compared to the simple cell generalization, buildings are merged to a geometry that reflects the original height distribution in a more sensitive way, as the highest building inside a block creates a peak in the hull.

3.2.1 Implementation

For each cell the geometry of the contained buildings is extracted and the points are fed into the convex hull calculation. For convex hull computation, a number of libraries exist, including CGAL and qHull [25]. The result is returned by a set of polygons representing the hull, which is finally used as the generalized cell geometry.

3.2.2 Results

The results –for typical large city– show an "organic look" (Fig. 7) as the convexity generally induces smoother structures. This is against the principle of visual correctness and especially does not show the typical characteristics of buildings orthogonal, parallel and sharp-edged structures. But, comprehensibility is the main task of maps and clearly the hull hides many details while preserving in a way the height characteristics. A more
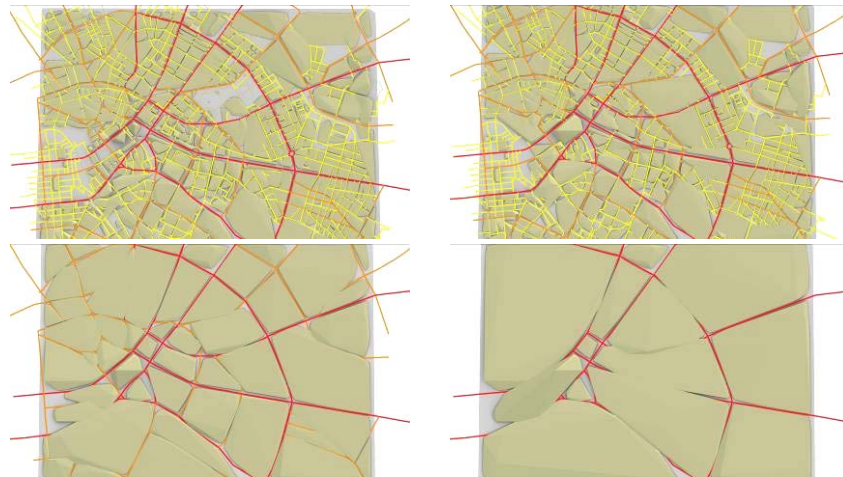

Fig. 7. Examples of the convex hull technique, shown at 4 different levels of generalization.

severe problem is the convexity when a cell is concave. In this case, the generalized block overlaps its cell possibly leading to intersections with other geometry and damages the appearance of the network elements. Additionally, the landmark visualization presented before cannot be integrated in a straightforward way.

As a solution to these problems and to provide for a finer selection of the degree of generalization, clusters based on neighborhood could be created and used as input for the convex hull calculation instead of the mere cells. This would allow us to use the parameter of the minimal cluster distance to define the DoG, which is currently limited to the levels of the network's hierarchy.

3.3 Voxel-Based Generalization

This generalization technique applies raster data filter operations to 3D raster data gained from the buildings' geometry [15]. To do so, the geometry is sampled within a 3D grid. Then, morphological opening and Gaussian blurring is performed. To convert the 3D grid to geometry, we apply the marching cubes algorithm to the grid. In addition, the raster data is processed with morphological opening operations to perform an aggregation.

As stated above, morphological operations have been applied to vector data representations before [22][23]. However, we also wanted to experiment with other raster data filter operations like Gaussian blurring.

3.3.1 Voxelization

The first step transforms the geometry from vector space to 3D grid space. This is done through a regular grid that is laid over the geometry and samples it at equidistant points. The grid is set up as follows: The resolution can be given as the ratio $res = \dfrac{\text{real world units}}{\text{grid units}}$. Thus, the dimensions of the grid **dim** can be calculated by taking the geometry's extent represented through the bounding box's diagonal d_{bbox}, and dividing it by the resolution:

$$\dim = \begin{pmatrix} \dim_x \\ \dim_y \\ \dim_z \end{pmatrix} = \left\lceil \frac{1}{res} \cdot d_{bbox} \right\rceil + 2 \cdot \begin{pmatrix} pad \\ pad \\ pad \end{pmatrix} \text{ with } pad \in N \tag{4}$$

To support raster data operations that expand the extent of the original geometry, a padding *pad* specified in grid units is inserted on each axis' ends. For example a dilation applied to the raster data may enlarge connected structures. The grid's origin is set as the lower left front point of the geometry's bounding box minus the padding. For each grid cell, the object space is sampled to be either inside (1) or outside (0) a solid (building). This leads to a three dimensional binary image.

For a better sampling quality, a box filter can be used. This was done experimentally in this work: For each voxel, not one point is sampled in the original geometry, but 8 points of a box centered at the point. The average of these points is then assigned to the grid cell, which leads to a gray scale image. Currently, the filtering is implemented in a simple way and, therefore, takes 8 times the processing time. The benefits can be seen in Fig 8, where the appearance is smoothed and less artifacts occur if a filtered sampling is performed.
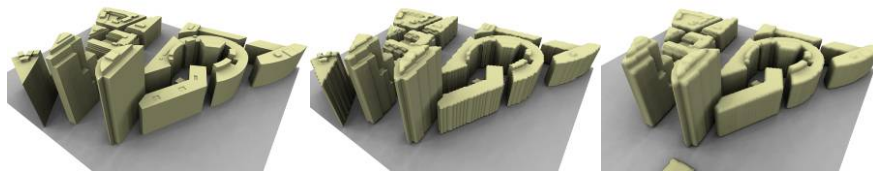

Fig. 8. Voxelization: original 3D building ensemble (left), result of voxelization wit a resolution of 2m (middle), and result of filtered voxelization (right).

As a simplification of the current implementation, only prismatic building shapes are assumed and thus a point considered inside the building, if it lies within the ground plan polygon and within the building's height. After all grid cells have been set, the grid is a voxel representation of the original geometry. However, the rasterization naturally implies a degradation of the original data, which leads to alignment artifacts when transformed back to a polygonal model. For the images, the original building geometry in the example (Fig. 8) has been sampled with a resolution of 2 meters of the real world geometry reflected one grid unit. Even this coarse resolution leads to a grid size of $156 \times 56 \times 170 = 1,485,120$ grid points.

3.3.2 Raster Data Operations

Many filter operations have been developed for use on raster data images. Typically, filters to remove noise and to smooth images rely on morphological operations. In [11][23], morphological operations have been used as vector space operations to aggregate and simplify building

geometry. Since after voxelization, the building geometry has been transformed to 3D raster data representation, these operations can now be applied directly.

The elementary operations can be defined as follows: The raster data is the 3D input image, where for each element (voxel) a structuring element B is applied. The structuring element in this case is a cube of $3 \times 3 \times 3$ units. It is moved over the input image. For each voxel, the structuring element is compared with the input image (source grid) and the output voxel is determined as follows:

- **Erosion**: A grid cell in the target grid is set to 1, if *all* voxels of the structuring element B can also be found in the source grid. Otherwise it is set to 0. This is done for the whole grid.
- **Dilation**: A grid cell in the target grid is set to 1, if *one* voxel of the structuring element B can be found in the source grid. Otherwise it is set to 0. This is done for the whole grid.

Erosion and dilation are elementary operations, which can be used to realize morphological opening and closing. Opening is achieved by applying dilation followed by erosion to the image; it leads to an aggregation of near structures. Closing is achieved by applying erosion followed by dilation of the image; it is useful to eliminate small elements.

However, the effect on the raster data is determined by the grid's resolution, which sets the size of one grid unit. Thus, opening and closing are always parameterized with multiples of one grid unit. In addition, the current implementation only supports binary images, i.e., no filtered voxelization can be applied before. Fig. 9 shows a series of morphological operations applied to our test models.

Apart from morphological operations, there are also other filters such as tent, cubic, or quartic filters as well as the Gaussian blur filter. Its usage on the rasterized building geometry has two benefits: First, a Gaussian blur as a low-pass filter further eliminates small features of the geometry that are still in the raster data. Second, the alignment structures introduced by the sampling are dampened. Generally, sharp edges and the surface are smoothed. Fig. 10 shows how the smoothed geometry looks after performing morphological opening.
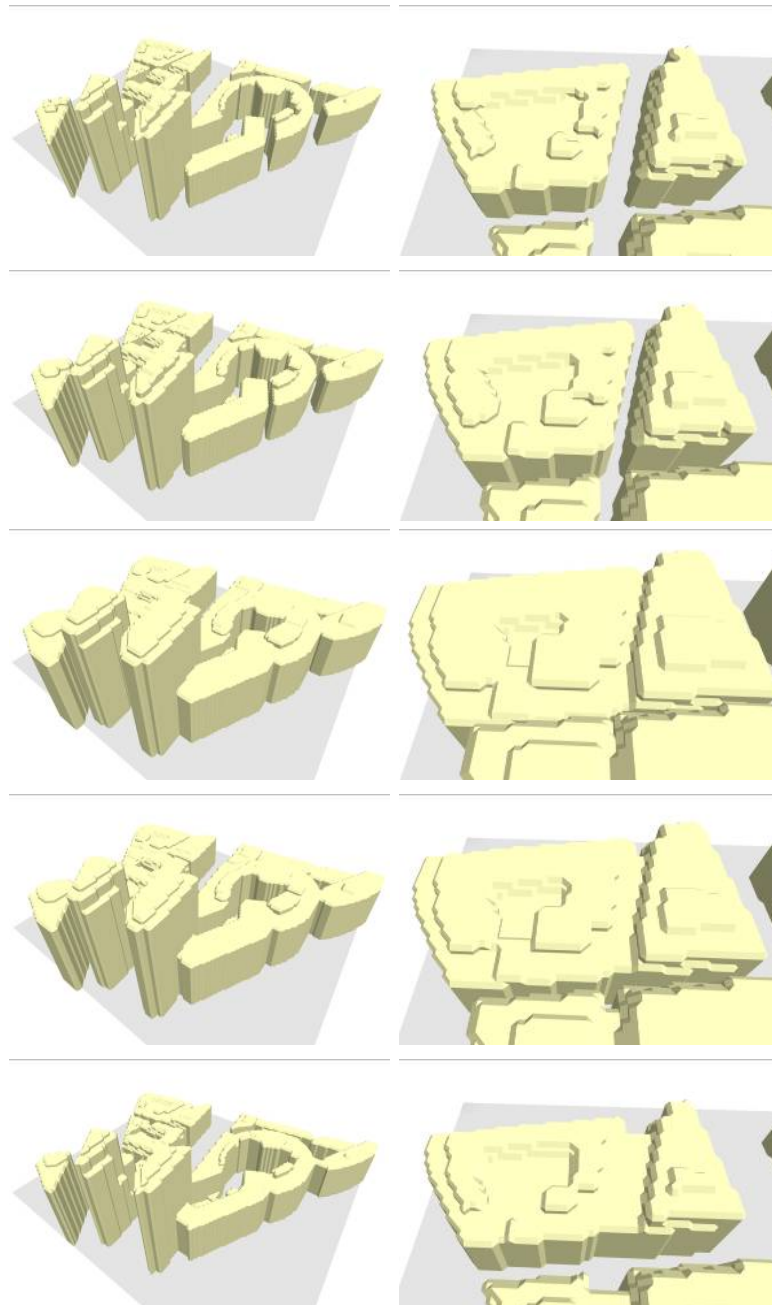
Fig. 9. Application of morphological operations. Starting from the model in row 1, rows 2 and 3 show subsequent dilation operations, and rows 4 and 5 show subsequent erosion operations.

To rely on a robust implementation of the raster data operations, the *nrrd* library from the *teem* project [32] was chosen. The *nrrd* library is accessed through the command-line tool *unu*, which works on a simple file format *.NRRD* (for "nearly raw raster data"). It supports a wide range of operations of which resampling is used for this work. With the aim to create smoother surfaces with less alignment artifacts, the Gaussian blur is applied for the reasons mentioned.
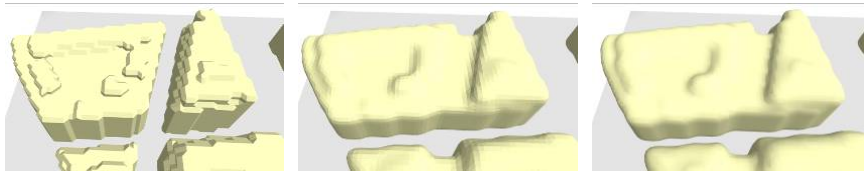

Fig. 10. Application of voxelization, opening and Gaussian blurring (left to right).

3.3.3 Marching Cubes

In the final step, the processed raster data is transferred to polygonal representation. For this, we extract isosurfaces, i.e., surfaces with a common isovalue everywhere on the surface, producing a surface from the samples of a scalar field defined as a mapping and a given isovalue, setting the threshold to separate between inside and outside [21]. To accomplish this, a freely available marching cubes implementation from [20] is used to create the geometric model. Note that all images presented in this section including the intermediate steps (Fig. 8, 9, 10) are created after applying the Marching Cubes to get a renderable polygonal representation.

3.3.4 Results

The workflow of this technique currently prohibits a completely automatic handling since the parameters (e.g., grid resolution, morphological operations step-width, Gaussian blur factor) have been set manually in our test model.

The result is characterized by its "bubble look", lacking a typical city model look. Through the blurring, soft shapes are introduced, while without filtering or blurring the alignment steps are clearly visible. To control the visualization, the resolution of the grid can be adapted, and the offset used in the opening operation can be varied. Finally, also the Gaussian blurring could be used to tune the DoG, however it might be only used to smooth the alignment structures introduced by the sampling.

4 Comparison

It is difficult to define appropriate and objective quality measures, as abstraction and generalization are ambiguous in their result, being understood differently by different individuals. In addition, yet few conventions exist for digital 3D city maps. While it is possible to define quantitative measures for different generalization techniques, the current results are rather preliminary and serve as the basis for further research. Usability tests should help to provide these more quantitative measures in the future.

4.1 Qualitative Measures

In the following, we concentrate on qualitative measures of the presented generalization techniques to evaluate their potential:
- **Similarity to Maps:** How do the results show similarities to 2D map presentations?
- **Similarity to Reality:** How do the results show similarities to (photorealistic) depictions of non-generalized 3D city model depictions?
- **Aggregation Capabilities:** How do the techniques support building aggregation?
- **Landmark Handling:** How do the techniques handle landmark and outlier buildings?
- **Controllability:** How can the application control the degree of generalization?
- **Dynamic Adaptation:** How can the application change the technique's parameters in response to dynamically changing viewing parameters?

4.2 Simple-Cell Generalization

This technique creates 3D city model depictions similar to traditional generalized bird's eye view maps if looked from above. The block structure made from cells of intersected streets is typical and can be understood easily. In terms of realism, the model still permits –while being an abstraction– to recognize the extents of a block. The added height gives a further hint about the original building ensemble. The aggregation is done rather naively, depends very loosely on the concrete building geometry and thus is insensitive against potential complexity. Landmarks can be effectively emphasized because their contours can be simply cut out of the cell and the landmark placed into the gap.

The DoG can be controlled by the hierarchy level selected for the network and the width used for network elements. A continuous transfer between different DoG representations has not been investigated so far; a

continuous blending with a hysteresis during interactive zooming might be one solution.

4.3   Convex-Hull-Based Generalization

The convex-hull approach has not been used so far in maps or map-like visualizations. The "organic" shapes are in contrast to photorealistic depictions. Constrained by the network, the convex hull still reveals and emphasizes the original block boundaries. The height of the largest building contributes much to the occurrence. However, concave cells are handled wrong by the convex hull; in a future version, we plan to replace the convex hull by alpha shapes [9], which can adapt the resulting hull more closely to the building geometry. In addition, network segments running into blocks are covered by the hulls. Handling landmarks and outliers is difficult because they would have to be cut out of the block using 3D Boolean operations. Very tall landmark buildings could just be placed within the hull, but the result would create intersecting geometry. Nevertheless, 3D convex hull creation relies on a mature algorithm working on simple points. For this reason, the technique is robust and insensitive against geometric problems.

The parameter to control the convex hulls is the size of the cells defined by the network. To provide more flexibility as well as to solve the aggregation problems, one could run a clustering algorithm initiated with a given minimal distance. The current solution does not suggest an easy mechanism for the continuous change between different DoG representations.

4.4   Voxel-Based Generalization

The voxel-based technique is limited to small areas due to the huge amount of required grid data. The result is unlike every map or map-like visualization. The bubble shapes remind the original buildings but do not feature the typical sharp edges and orthogonal structures. Still, aggregation and simplification can be achieved and the unusual look underlines the abstraction. The viewers know from the first view that they look at an abstraction of reality, not a photorealistic visualization.

Landmarks can be excluded from the common voxelization at all and inserted later. As an alternative, they could be voxelized separately using a higher resolution and then inserted without further simplification. This would have the advantage that no visual break occurs in the image, but still the building would be emphasized.

To control this technique, there is first the grid resolution. While a higher resolution yields better quality with less alignment artifacts, it also leads to an explosion of the data to be processed. The second parameter is the buffer size, which controls the morphological opening. Finally the Gaussian blur (or also other filters) can be executed with a given variance and cutoff. Though, the blurring can be seen as a post-processing step independent from the generalization but just to smooth the alignment stair-structures.

Voxel-based generalization cannot be used in conjunction with a dynamic DoG in the current implementation, as the processing is not done all natively by the prototype system. The filter operations are done using an external utility and are re-read later from a file. Additionally, the computation time and the memory needed prohibit a change on-the-fly.

The marching cubes algorithm used introduces a number of redundant polygons, which is no problem for small scenes but might be a problem for bigger city plans. Here, another post-processing step and / or a dynamic level of detail adaptation might be necessary.

5   Conclusions

This paper presented and discussed three different techniques to automatically derive generalized 3D building ensembles for a cell structure defined by hierarchical networks that divide the reference plane of a virtual 3D city model. The generalized models can be applied to improve the comprehensibility and effectiveness for complex, large-area 3D city models.

The comparison revealed that while the cell-based generalization technique leads to convincing results, the voxel- and convex hull-based techniques currently are less feasible.

Therefore, in the future we will investigate how to sharpen the presented methods towards characteristic architectural elements of 3D building ensembles. In addition, we want to expand the methods towards further city model elements such as vegetation and site objects. An important remaining challenge concerns the handling of multiple scales: A continuous mapping of the DoG to a geometric representation would allow us to combine continous scales in one view of the scene. Also, an optical zoom could be accompanied with a smooth semantic zoom. We will work on this problem when moving forward with our research.

Acknowledgement

References

[1] K.-H. Anders. Level of Detail Generation of 3D Building Groups by Aggregation and Typification. *Proc. 22nd International Cartographic Conference, La Coruña, Spain*, 2005.

[2] M. Beck. Real-Time Visualization of Big 3D City Models. *International Archives of the Photogrammetry Sensing and Spatial Information Sciences*, Vol. XXXIV-5/W10, 2003.

[3] H. Buchholz, J. Döllner. View-Dependent Rendering of Multiresolution Texture-Atlases. *Proc. IEEE Visualization 2005*, Minneapolis, 2005.

[4] CGAL – Computer Geometry Algorithm Library, www.cgal.org

[5] CityGML, www.citygml.org

[6] J. Döllner, H. Buchholz. Continuous Level-of-Detail Modeling of Buildings in Virtual 3D City Models. *Proc. 13th ACM International Symposium of Geographical Information Systems* (ACMGIS 2005), 173-181, 2005.

[7] J. Döllner, T. H. Kolbe, F. Liecke, T. Sgouros, K. Teichmann. The Virtual 3D City Model of Berlin - Managing, Integrating and Communicating Complex Urban Information. *Proc. 25th International Symposium on Urban Data Management UDMS 2006*, Aalborg, Denmark, 2006.

[8] J. Dykes, A. MacEachren, M.-J. Kraak. *Exploring Geovisualization*. Elsevier Amsterdam, Chapter 14, 295-312, 2005.

[9] H. Edelsbrunner, E. Mücke. Three-Dimensional Alpha Shapes. *ACM Transactions on Graphics*, 13, 43-72, 1994.

[10] E. Fogel, R. Wein, B. Zukerman, D. Halperin. 2d Regularized Boolean Set-Operations. In C. E. Board (Ed.) *CGAL-3.2 User and Reference Manual*. 2006.

[11] A. Forberg, H. Mayer. Generalization of 3D Building Data Based on Scale-Spaces. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* (34)4, 225-230, 2002.

[12] E. Gobbetti, F. Marton. Far Voxels - A Multiresolution Framework for Interactive Rendering of Huge Complex 3D Models on Commodity Graphics Platforms. *ACM Transactions on Graphics*, 24(3):878-885, 2005.

[13] G. Hake, D. Grünreich, L. Meng. *Kartographie*. Walter de Gruyter, Berlin, New York, 8. Ed., 2002.

[14] L. Harrie. *An Optimization Approach to Cartographic Generalization*. PhD thesis, Department of Technology and Society, Lund Institute of Technology, Lund University, Sweden, 2001.

[15] T. He, L. Hong, A. Kaufman, A. Varshney, S. Wang. Voxel Based Object Simplification. *Proc. 6th Conference on Visualization*, 296, 1995.

[16] H. Hoppe. Progressive Meshes. *Computer Graphics Proceedings, Annual Conference Series*, 1996 (ACM SIGGRAPH '96 Proceedings), 99-108, 1996.

[17] M. Kada. 3D Building Generalisation. *Proceedings of 22nd International Cartographic Conference,* La Coruña, Spain, 2005.

[18] T.H. Kolbe, G. Gröger, L. Plümer. CityGML – Interoperable Access to 3D City Models. *Proc. 1st International Symposium on Geo-Information for Disaster Management*, Springer Verlag, 2005.

[19] A. Lakhia. Efficient Interactive Rendering of Detailed Models with Hierarchical Levels of Detail. *Proc. 3D Data Processing, Visualization, and Transmission, 2nd International Symposium on (3DPVT'04)*, 275-282, 2004.

[20] T. Lewiner, H. Lopes, A. W. Vieira, G. Tavares. Efficient implementation of marching cubes´ cases with topological guarantees. *Journal of Graphics Tools*, 8(2):1-15, 2003.

[21] W. E. Lorensen, H. E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *SIGGRAPH '87: Proc. 14th Aannual Conference on Computer Graphics and Interactive Techniques*, ACM Press, 163-169, 1987.

[22] H. Mayer. Three Dimensional Generalization of Buildings Based on Scale-Spaces. *Report, Chair for Photogrammetry and Remote Sensing, Technische Universität München*, 1998.

[23] H. Mayer. Scale-Space Events for the Generalization of 3D-Building Data. *International Archives of Photogrammetry and Remote Sensing*, 33:639-646, 1999.

[24] L. Meng and A. Forberg. 3D Building Generalization. In W. Mackaness, A. Ruas, and T. Sarjakoski (Eds.) *Challenges in the Portrayal of Geographic Information: Issues of Generalisation and Multi Scale Representation*, 211-32. 2006.

[25] qHull, www.quhull.org

[26] J.-Y. Rau, L.-C. Chen, F. Tsai, K.-H. Hsiao, W.-C. Hsu. Lod generation for 3d polyhedral building model. In *Advances in Image and Video Technology*, 44-53, Springer Verlag, 2006.

[27] J. Ribelles, P. Heckbert, M. Garland, T. Stahovich, V. Srivastava. Finding and Removing Features from Polyhedra. *American Association of Mechanical Engineers (ASME) Design Automation Conference, Pittsburgh PA, September* 2001.

[28] M. Sester. Generalization Based on Least Squares Adjustment. *International Archives of Photogrammetry and Remote Sensing*, 33:931-938, 2000.

[29] M. Sester. Application Dependent Generalization - the Case of Pedestrian Navigation. *Proc. Joint International Symposium on GeoSpatial Theory, Processing and Applications (ISPRS/Commission IV, SDH2002), Ottawa, Canada, July*, 8-12, 2002.

[30] K. Shea, R. McMaster. Cartographic generalization in a digital environment: when and how to generalize. 9th *International Symposium on Computer-Assisted Cartography.* 56-67, 1989.

[31] R. Stüber. *Generalisierung von Gebäudemodellen unter Wahrung der visuellen Richtigkeit.* PhD thesis, Rheinische Friedrich-Wilhelms-Universität zu Bonn, 2005.

[32] teem, teem.sourceforge.net/unrrdu

[33] F. Thiemann. Generalization of 3D Building Data. *Proc. Joint International Symposium on GeoSpatial Theory, Processing and Applications (ISPRS/Commission IV, SDH2002), Ottawa, Canada, July*, 34(3), 2002.

[34] R. Wein, E. Fogel, B. Zukerman, D. Halperin. 2D Arrangements. In: C. E. Board (Ed.), *CGAL-3.2 User and Reference Manual*. 2006.

[35] J. Willmott, L.I. Wright, D.B. Arnold, A.M. Day. Rendering of Large and Complex Urban Environments for Real-Time Heritage Reconstructions. *Proc. Conference on Virtual Reality, Archaeology, and Cultural Heritage*, 111-120, ACM Press, 2001.