3D BUILDING GENERALIZATION
BASED ON HALF-SPACE MODELING

Martin Kada

Institute for Photogrammetry (ifp), University of Stuttgart, Germany
Geschwister-Scholl-Strasse 24D, D-70174 Stuttgart
martin.kada@ifp.uni-stuttgart.de

Commission II, WG 3

KEY WORDS: Generalization, Algorithms, Building, Three-dimensional, GIS, Visualization

ABSTRACT:

This article presents a novel approach for the automatic generalization of 3D building models with regard to a cartographic 3D visualization. Here, the goal is not to render realistic, but rather visually appealing images of an urban scene. The geometric simplification is realized by remodeling the object by means of a process similar to half space modeling. Approximating planes are determined from the polygonal faces of the original model, which are then used as space dividing primitives to create façade and roof structures that are of simpler shape. In order to improve the final shape of the simplified building models, form optimization steps like hole filling, fragment removal and vertex contraction are further used.

1. INTRODUCTION

The geometric simplification of complex 3D objects has been a topic in various fields of research. In computer graphics, e.g., discrete and continuous levels of detail representations of polygonal and triangular meshes are automatically generated in order to reduce storage space, speed up network transmission, shorten geometric computations and improve rendering performance. During the surface simplification processes, the geometric accuracy of the input model is preserved as good as possible. Due to the geometric error metric, the algorithms show best results on highly complex, smooth objects made from hundreds of thousands to millions of primitives. Some datasets are even too complex to fit in main memory and must be processed out-of-core (Lindstrom, 2000). A good overview of surface simplification algorithms is given in (Heckbert and Garland, 1997) and (Puppo and Scopigno, 1997).

In contrast, single 3D building models are of comparatively low complexity. Collected from data like aerial images or airborne laser scanning, they typically consist of at most a few hundred polygons. Each building model, however, exhibits special characteristics that need to be preserved during simplification. These are e.g. right angles that can often be found in the building architecture. The application of surface simplification algorithms on this type of data often results in tilted objects.

Similar problems occur for the automatic simplification of 2D building ground plans. Line simplification, e.g. the algorithm of (Douglas and Peucker, 1973) does not yield very good results for such objects. Consequently, a number of specialized generalization solutions have been proposed in the past since the early work of (Staufenbiel, 1973). (Sester, 2000)) e.g. simplifies the shape of 2D ground plans by applying a set of simple rules followed by a least squares adjustment.

In (Kada, 2002), an analogous approach for the generalization of 3D building models is presented. The polygonal building models are iteratively simplified by combining a number of edge collapse operations into a single step. Building symmetries, that need to be detected prior to the simplification, are maintained in the process (see e.g. Figure 1).

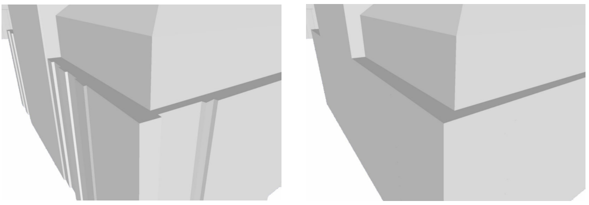


Figure 1. The symmetries of the original building (left) are maintained even after simplification (right).

Another work on 3D generalization is based on scale-space theory (Forberg, 2004)). Orthogonal building structures are simplified by shifting parallel facets that have been found to be under a certain distance until they merge. A squaring operation is used for non-orthogonal structures such as roofs. (Thiemann and Sester, 2004) presents a segmentation algorithm that is adapted from the work of (Ribelles et al., 2001). The resulting partitioning of the 3D building model is transformed into a Constructive Solid Geometry (CSG) tree. Fragments of the building can then be emphasized, aggregated or eliminated depending on their importance.

All the abovementioned methods heavily dependent on the quality of the input models or rely on specific building characteristics. If e.g. angles are close but not exactly orthogonal, building symmetries may not be found (Kada, 2002), shifted facets never merge (Forberg, 2004)) or the segmentation produces too many fragments (Thiemann and Sester, 2004). Furthermore, the proposed simplification operators are not general enough and only work on a subset of structural elements that can be found in 3D building models.

The presented algorithm in this paper is an extension of the work first published in (Kada, 2005). It extends the approach by form optimization methods that are described in subsections 3.5 to 3.7. For better understanding of the algorithm, a more thorough algorithm outline is given in section 2.

2. ALGORITHM OUTLINE

The paper presents a new generalization algorithm for 3D building models that overcomes the aforementioned shortcomings. It generates simplified versions of an object by recreating the building using a half-space modeling process. In half-space modeling, a cuboid block building is e.g. generated by trimming the infinite space using six planar half-space primitives. Assuming the plane normals point out of the object, the building is on the negative side of all six planes.

In order to get a simplified model, the planes that are used here as half-space primitives approximate the facades of the original building model. They are like buffers that include protrusions and other small structural elements that are to be eliminated. As an example, the original building model depicted as a 2D ground plan in Figure 2 (top) is approximated by the six half-space primitives that are illustrated as red lines (bottom).

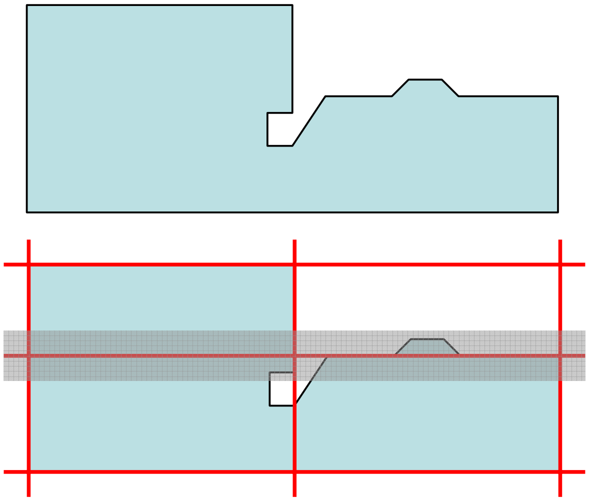


Figure 2. The 2D building ground plan (top) is approximated by six half space primitives (bottom). The grey area shows the buffered area of the horizontal primitive.

These half-space primitives are then used to subdivide an initially infinite space into smaller subspaces. Because our approximating primitives have no preferred orientation (see subsection 3.1), a recursive subdivision is, however, not possible. Consequently, the algorithm divides all subspaces that intersect a half-space primitive by brute force, creating a large number of fragments in the process. These Fragments must be further differentiated in building and non-building objects (see subsection 3.3). Here, a percentage value is calculated that denotes the overlap of the fragment compared to the original building model. Fragments with a low overlap are non-building fragments and are discarded subsequently (see Figure 3).

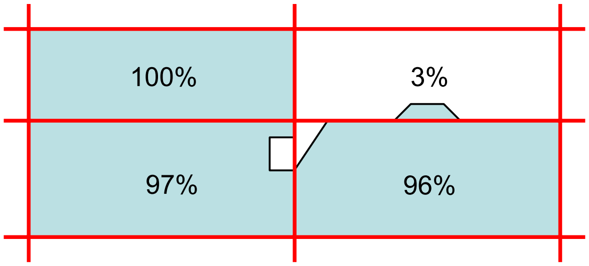


Figure 3. The percentage value denotes the overlap of the fragment compared to the original ground plan.

The generalized ground plan is then gained by gluing the identified building fragments together (see Figure 4).

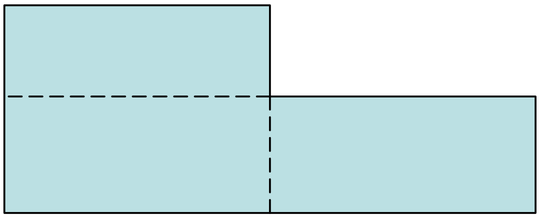


Figure 4. Generalized building ground plan. Fragments are glued together at the dashed lines.

The algorithm works basically the same for 3D building models. But now the half-space primitives are planes that are determined from the polygonal faces. First, only the façade polygons are considered which leads to flat-top building block fragments of a generalized ground plan. Because the structure of the roof can be very complex, we remodel it individually per fragment using approximating planes averaged from the roof polygons (see subsection 3.4). The resulting objects are referred to as building primitives. The generalized building model is then generated by gluing these building primitives.

3. 3D BUILDING GENERALISATION WITH APPROXIMATING PLANES

The presented generalization algorithm is designed for polyhedral 3D building models. It is assumed that most polygonal wall elements are oriented in parallel, coplanar or rectangular and that these symmetries must be preserved during simplification.

3.1 Determination of Approximating Planes

The shape of the generalized building model is highly dependent on the geometric properties of the approximating planes that are used in its generation. Because each subdivision adds complexity to the final model, it is favorable to use the smallest number of planes that optimally approximate the original shape. Each approximating plane represents a set of polygons that belong to the same building façade. These are primarily the polygons with the same orientation. However, also other arbitrary polygons are included that represent protrusions and other structural elements in the façade (see Figure 5). These elements will be eliminated in the generalized model.

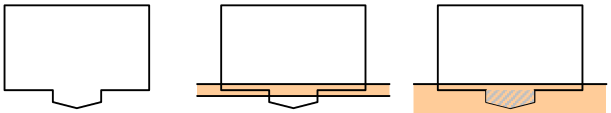


Figure 5. Approximating planes include polygons of the façade and structural elements that are to be eliminated. The highlighted area represents the buffer.

In order to keep the number of approximating planes small, polygons with oppositely directed normal vectors are linked to the same plane. The benefit of this approach is that opposing façades under a certain distance threshold will be made coplanar or even merge together (see Figure 6).

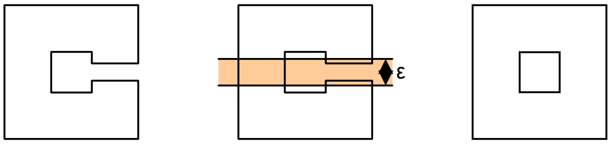


Figure 6. Façades with oppositely directed normal vectors are linked to the same plane and therefore merge together.

Matching façade polygons to corresponding planes is often ambiguous for complex models even if human interaction is involved. We therefore kept the algorithm for finding the approximating planes rather simple, accepting that a small number of polygons are erroneously linked to planes which represent other façades. This effect can usually be neglected, however, as it only affects small polygons that are usually of limited significance to the overall shape of the building.

We use an iterative approach for finding the set of approximating planes. Each iteration results in the plane of greatest importance, which is measured by the total area of included polygons that are parallel to the approximating plane. At the beginning of an iteration step, a buffer defined by two delimiting, parallel planes is created for each polygon (see Figure 5). The initially planar buffers are then merged pair wise to create larger buffers until the delimiting planes reaches a maximum threshold distance. In order to create buffers with many polygons, the orientation of the resulting buffer is averaged before merging. If the polygons from both buffers can not be included in the new averaged buffer, the merge is aborted. The iteration step stops when no more buffers can be merged and the averaged plane equation of the buffer with the largest total area is returned. The polygons inside the buffer are discarded from further processing. By repeating this process, the set of approximating planes is found in descending order of importance. To limit the number of approximating planes, we eliminate all planes with a total area that is lower than the square of the threshold distance. See Figure 7 (a-f) for example planes determined for a 3D building model.

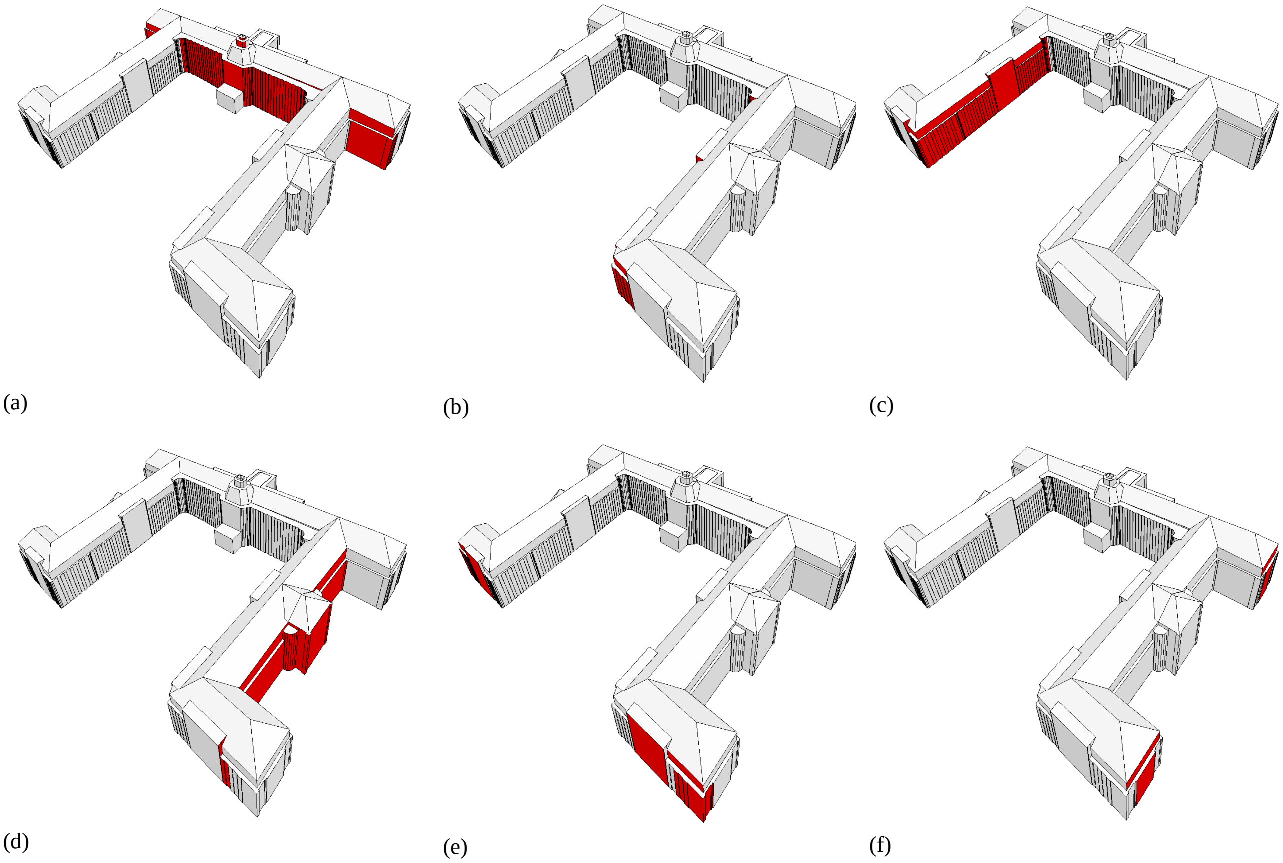


Figure 7. Thirteen approximating planes have been determined for the building model. (a-f) shows six planes and their corresponding polygons (highlighted). (e) Some polygons have been erroneously matched to planes of higher importance (b, c+d).

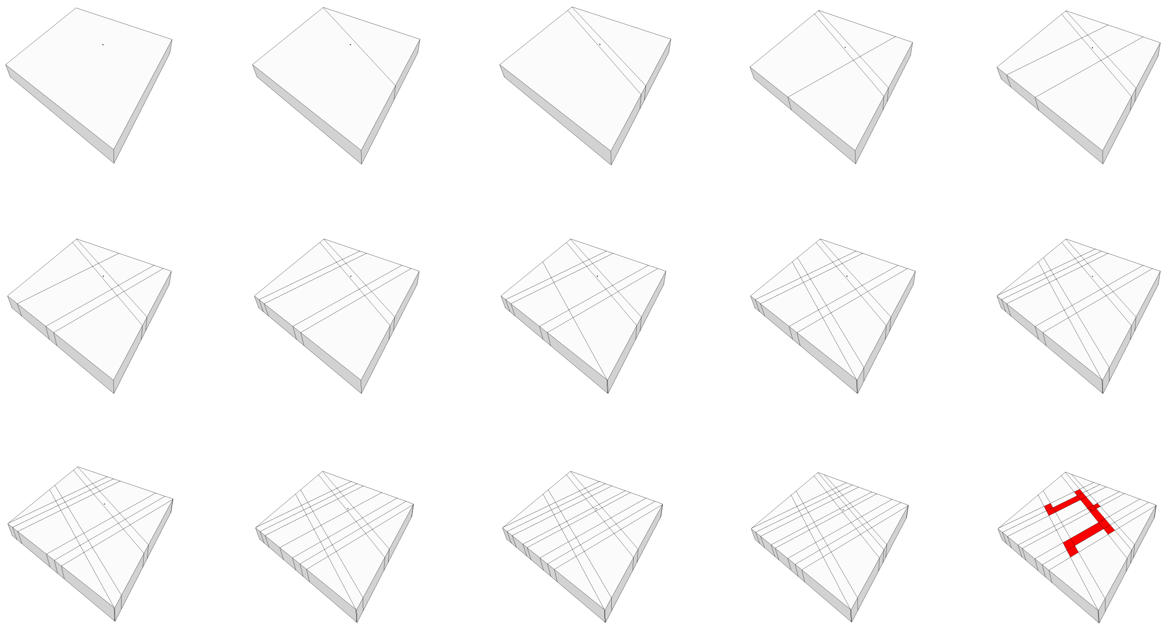


Figure 8. The infinite space is subdivided brute force by the approximating planes. Building fragments are then differentiated in building (highlighted) and non-building fragments.

3.2 Adjustment of Planes

In order to preserve right angles and also to endorse parallelism in the building model, the approximating planes are analyzed in a last step. If the angle of the normal vectors from two or more planes is found to be below a certain threshold, these planes are made rectangular or parallel, respectively. If the deviation is only a small angle, this can be done approximately by changing the normal vector of the plane equation and adjusting the distance value. For larger values, a rotation of the planes centered in their weighed centroid of the polygons should be chosen.

3.3 Construction of Building Fragments

After the approximating planes equations have all been determined, they are then used to generate a fragmentation of the building model. For this purpose, an infinite 3D space is subdivided brute force by the planes which serve the purpose of half-space primitives (see Figure 8). In practice an infinite space is unsuitable, so a solid two times the size of the building's bounding box is used as a substitute. Because the equations of the approximating planes have no horizontal component, the infinite space is at this point only divided in two dimensions. The resulting solids are therefore convex 2D ground polygons swept along the vertical direction.

As previously mentioned, the dividing process makes it impossible to directly identify the building fragments. Therefore, solids must be differentiated in building and non-building fragments in a subsequent step (see Figure 8). For each fragment, a percentage value is calculated that denotes the fraction of building to non-building space. Because the space division was basically done in 2D, we use the area of the original building ground plan inside the fragment divided by the ground area of the fragments itself. All solids with a percentage value under a given threshold value are then denoted as non-building fragments and discarded from further processing. A suitable threshold value is dependent on the buffer size used to determine the approximating planes

At this point, the merging of the building fragments would result in a flat roof building model with a generalized building ground plan. It should be noted, however, that the process so far is not just a 2D ground plan generalization, because the fragmentation is essential for generating the generalized roof structure.

3.4 Simplification of the Roof Structure

The roof structure can be very complex for general 3D building models. Therefore, the generalization process has neglected the roof polygons so far. However, the resulting fragmentation produced from previous steps highly simplifies this task as the roof can now be generated for the individual building fragments. It has to be ensured, though, that the roof polygons of neighboring fragments still fit against each other. So first, a set of approximating roof planes is determined globally from the original model using the same approach as described in subsection 3.1. Then a subset is created for each fragment that includes planes that have polygons intersecting the respective fragment. The geometric properties of the planes are not changed in this process. Fragments are then divided by their individual set of approximating planes as described above (see Figure 9).

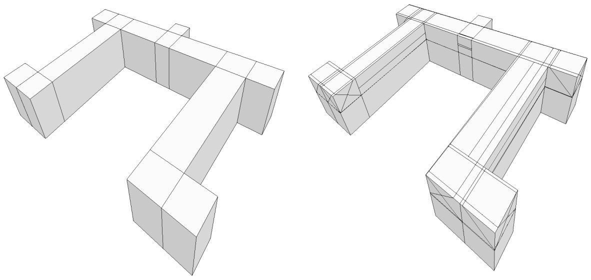


Figure 9. Fragments divided by roof planes.

The resulting fragments are now real 3D solids, so the differentiation in building and non-building fragments is done in 3D space. A percentage value is calculated this time that denotes the volume of the original building model inside the respective fragment. However; the non-building fragments are not entirely discarded yet, but rather saved for further processing. Figure 10 shows the resulting building fragments for the example building in Figure 9.

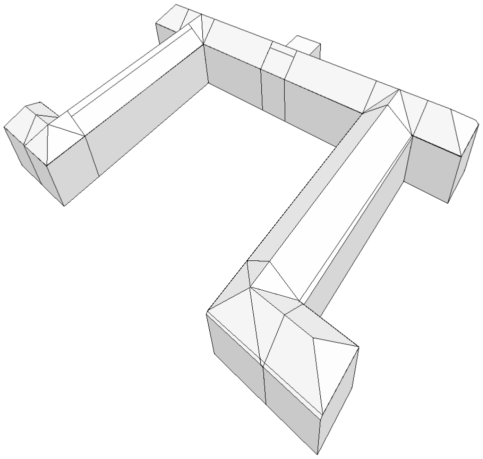


Figure 10. 3D building model with simplified roof.

3.5 Filling Holes

In our experience, the threshold buffer for determining the approximating planes for the roof must be chosen smaller then for the façades. Else the result is often a flat-top roof or a very asymmetrical roof structure. As a rule of thumb, we used a buffer for the roof planes half the size as for the façade planes. However, the small buffer size can lead to very small objects during fragmentation. Even though these small fragments might be correctly identified as non-building fragments, it is sometimes desirable to keep them in order to produce roof structures of simpler shape.

Each non-building fragment is tested against every other fragment it shares a common face with. Thereby, two values are computed: one is the total area of faces that share a common border to an already accepted building fragment and the other is the total area of all other faces. If the fragment borders more already accepted building then non-building fragments, the candidate is not discarded after all. This simple hole-filling algorithm produces roof shapes of overall smoother appearance (see Figure 11).

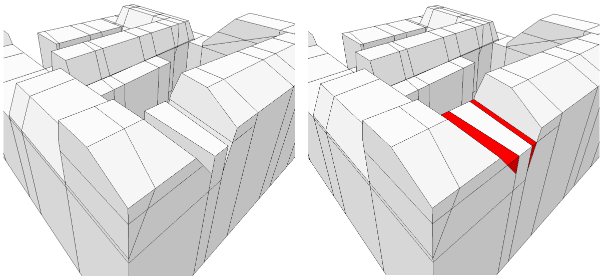
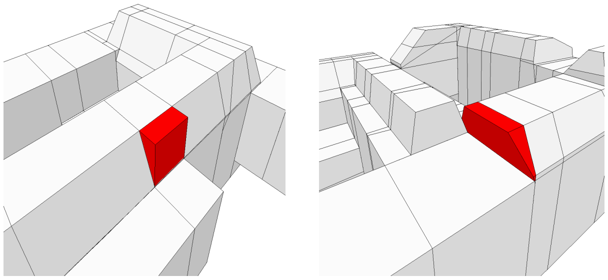


Figure 11. Holes are filled with non-building fragment to improve the overall shape of the roof.

3.6 Fragment Removal

A similar problem as described above are fragments that have been identified as building fragments, but are only adding perturbing details to the roof structure. Many such fragments can easily be identified by the fact that they have only one face that borders a neighbour fragment. We have stated two heuristic rules that determine if building fragments should be discarded:

1. If the neighbouring face is not a flat-top roof, but the fragment itself has a flat top face, then the fragment is removed from the model (see e.g. Figure 12a).
2. And, if the normal vector of the neighbouring face points downward, the fragment is also removed (see e.g. Figure 12b).



(a) (b)

Figure 12. Building fragments that disturb the roof shape (highlighted) are removed by a set of simple rules.

With these two simple rules, a lot of fragments are removed that disturb the shape of the roof.

3.7 Vertex Contraction

Even after applying the heuristic methods that improve the differentiation of building and non-building fragments, the resulting model still yields small inaccuracies. These can not be treated by removing or adding any more fragments. Rather, the building fragments need now be geometrically changed. Here, a vertex contraction approach is used that contracts all vertices that are located closely together into a single point (see e.g. Figure 13).

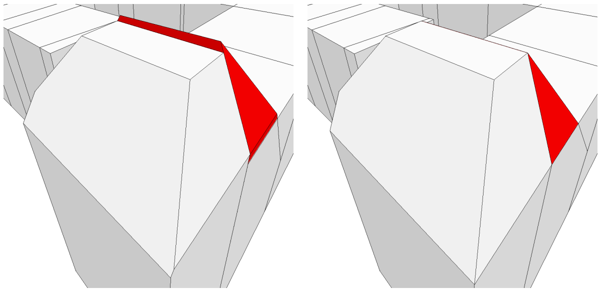


Figure 13. Fragments are geometrically altered by vertex contraction to better fit neighbouring fragments.

The position of the contraction point is determined in two steps using a distance threshold. First, all vertices that are in the horizontal vicinity of a ground plan vertex are given the x- and y-coordinates of this point. Vertices that are not close to such a ground plan vertex are contracted to the closest vertex with the highest total area of all adjacent faces. Second, the smallest set of height levels is determined from all vertices of the building fragments. Then the vertices are given the z-coordinate of the closest height level. During this process, however, some of the building fragments will become degenerated and need to be removed afterwards.

3.8 Fragment Merging

Now that a set of fragments have been determined which define the final model, the last step is to merge all fragments into a single, generalized building object. This can easily be realized in a union operator as fragments do not intersect each other. The operator removes both faces from fragments that border one another and merges the remaining faces. The result is a topologically correct 3D solid approximating the original building model.

4. IMPLEMENTATION AND RESULTS

The generalization algorithm has been implemented and tested on a number of 3D building models. It shows good results for models ranging from rather simple to very complex shapes. For a selection of examples see Figure 14. In all cases, the complexity of the objects could be highly reduced without destroying the overall appearance of the building. However, there are circumstances where wrong symmetries are generated during the determination of the approximating planes. Because the erroneous symmetries do usually not affect the topology of the resulting building model, this is corrigible during a geometric post processing.

As for the threshold values used in our testing. We basically only have two thresholds of importance: the buffer threshold for façade planes and the buffer threshold for roof planes. Other thresholds, e.g. for determining parallel or rectangular angles, are set to a fixed value of ten degrees. Best results for determining the approximating planes for the facades are between seven to ten meters. Lower thresholds yield too many fragments and therefore keep too many details. For the approximating roof planes, a smaller threshold of three to five meters was used. In the optimization by vertex contraction, a fixed threshold value of one meter was used. This was enough for all models we used the algorithm on to filter the usually small glitches out of the model.

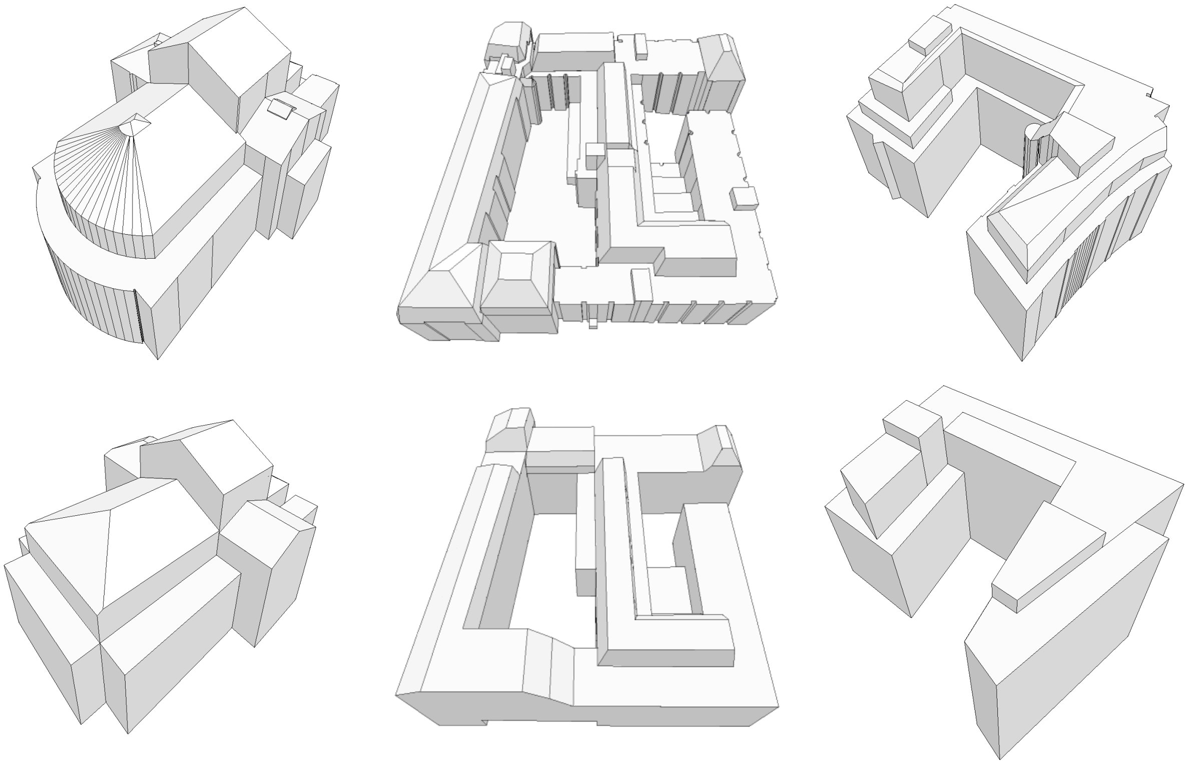


Figure 14. Generalized 3D building models in their original (top) and simplified (bottom) shape.

We also have tested and analyzed the performance of our implementation on a 3GHz Intel CPU. More then 50% of the computing time is currently used determining the non-building fragments during the simplification of the roof. Here, a brute force point in polyhedron test is applied for a large number of points in a three dimensional regular grid. Although there is room for improvement at this particular component of our implementation, the processing time for the example buildings still remained under 30 seconds per model.

5. CONCLUSION

In this article, a new approach for the generalization of 3D building models has been described. Unlike other algorithms that have been proposed in the past, the objects are not simplified in an iterative process. Instead, the generalized object is remodeled with approximating planes in a process that is similar to half-space modeling. Building symmetries like coplanarity, parallelism and rectangularity are preserved during the process or can even be enforced if needed. The generalization is solely controlled by an intuitive distance threshold that specifies the minimum size of building elements that are created.

So far our focus has been on single 3D buildings that are individually generalized in disregard of their surrounding area. This can cause problems as the resulting models might overlap with neighboring buildings. Also, symmetries of neighboring models may be violated if no special care is taken. Future work will therefore concentrate on the generalization of building blocks. The result may either be individual building objects, if e.g. object identification is required, or an aggregated building block.

Another topic will be the geometric adjustment of points, lines and faces to enforce both local and global symmetries.

6. ACKNOWLEDGEMENTS

The research described in this paper is founded by "Deutsche Forschungsgemeinschaft" (DFG – German Research Foundation). It takes place within the Centre of Excellence No. 627 "NEXUS – SPATIAL WORLD MODELS FOR MOBILE CONTEXT AWARE APPLICATIONS" at the University of Stuttgart. The geometry of the 3D building models is provided by Stadtmessungsamt Stuttgart.

7. REFERENCES

Douglas, D. and Peucker, T., 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. The Canadian Cartographer 10 (2), pp. 112-122.

Forberg, A., 2004. *Generalization of 3D Building Data Based on Scale-Space Approach*. In: International Archives of Photogrammetry and Remote Sensing, Vol. XXXV, Part B, Istanbul, Turkey.

Heckbert, P. S. and Garland, M., 1997. *Survey of Polygonal Simplification Algorithms*. In: Multiresolution Surface Modeling Course, SIGGRAPH '97, Los Angeles, USA.

Kada, M., 2002. *Automatic Generalisation of 3D Building Models*. In: Proceedings of the Joint International Symposium on Geospatial Theory, Processing and Applications '02, Ottawa, Canada.

Kada, M., 2005. *3D Building Generalisation*. In: Proceedings of the 22th International Cartographic Conference, La Coruna, Spain, 2005.

Lindstrom, P., 2000. *Out-of-Core Simplification of Large Polygonal Models*. In: Proceedings of ACM SIGGRAPH 2000, New Orleans, USA.

Puppo, E. and Scopigno, R., 1997. *Simplification, LOD and Multiresolution – Principles and Applications*. In: Eurographics '97 Tutorial Notes, Budapest, Hungary.

Ribelles, J., Heckbert, P. S., Garland, M., Stahovich, T. F. and Srivastava, V., 2001. *Finding and Removing Features from Polyhedra*. In: Proceedings of DETC'01, Pittsburgh, USA.

Sester, M., 2000. *Generalization Based on Least Squares Adjustment*. In: International Archives of Photogrammetry and Remote Sensing, Vol. XXXIII, Part B4, Amsterdam, Netherlands.

Staufenbiel, W., 1973. *Zur Automation der Generalisierung topographischer Karten mit besonderer Berücksichtigung großmaßstäbiger Gebäudedarstellungen*, Ph.D. thesis, Fachrichtung Vermessungswesen, Universität Hannover.

Thiemann, F. and Sester, M., 2004. *Segmentation of Buildings for 3D-Generalisation*. In: Proceedings of the ICA Workshop on Generalisation and Multiple Representation, Leicester, UK.