AUTOMATIC GENERALISATION OF 3D BUILDING MODELS

Martin Kada

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

Commission IV, WG IV/3

KEY WORDS: Generalisation, Buildings, Algorithms, Visualisation

ABSTRACT:

The paper presents an automatic approach for the generalisation of 3D building models with regard to the visualisation of urban landscapes. Simplified versions of such models are not only needed for level of detail structures in real-time rendering, but also for web-based 3D GIS and for the presentation on mobile computing devices. To yield more sophisticated building models compared to already known surface simplification algorithms from the field of computer vision, the presented solution is based on least squares adjustment theory combined with an elaborate set of surface classification and simplification operations. This concept allows for the integration of surface regularities into the building models which are important for visual impression. These regularities are stringently preserved over the course of the generalisation process.

1. INTRODUCTION

The development of tools for the efficient collection of 3D city models has been a topic of intense research for the past years. In addition to Digital Height Models and 3D data representing streets and urban vegetation, building models are the most important part thereof. Meanwhile, a number of algorithms based on 3D measurement from aerial stereo imagery or airborne laser scanner data are available for automatic and semi-automatic collection of 3D building models. As an example, Figure 1 shows a 3D model of Stuttgart collected by the approach of (Haala and Brenner, 1999). A good overview on the current state-of-the-art of experimental systems and commercial software packages is for example given in (Baltsavias, Grün and van Gool, 2001). Almost all of these systems describe the reconstructed buildings by general polyhedrons, since a building representation by planar faces and straight edges is feasible for most cases. The resulting 3D boundary representation is either provided directly or constructed from a number of building primitives which are used during the measurement process.

Originally, simulations for the propagation of electromagnetic waves used for the planning of antenna locations were the major application areas for 3D building models. Meanwhile visualisation in the context of three-dimensional car navigation systems, virtual tourism information systems or city and building planning has become the key market for that type of data. In our opinion one of the most important development-driving forces for the application of 3D city models is the widespread use of mobile devices for the provision of location based services. Features like personal navigation or telepointing, i.e. the provision of spatial information by pointing to regions of interest directly on the display, presume a realistic visualization of the 3D urban environment on these mobile devices. Due to the limited amount of computational power and small size of the displays on the one hand and the huge amount of data contained within a 3D city model on the other hand, the amount of information to be handled, stored and presented has to be reduced efficiently. Thus, the generalisation of the 3D building models as it is described within this article becomes a topic of major interest.

Figure 1. A 3D city model of Stuttgart.

In general, this process presumes the elimination of unnecessary details, whereas features, which are important for the visual impression, have to be kept. Especially for man-made-objects like buildings, symmetries are of major importance. For this reason, during the process of generalisation the preservation of regular structures and symmetries like parallel edges, perpendicular intersections or planar roof faces has to be guaranteed. In principle this geometrical regularisation is also required during data collection, since otherwise geometric errors introduced during measurement could result in erroneous structures of the building. Compared to the geometric regularisation during measurement, as e.g. described in (Grün and Wang, 2001), this problem is even aggravated during the simplification process due to the increasing deviations from the true building geometry.

2. RELATED WORK

The focus of this paper is on automatic simplification of polyhedral building models. A vast amount of research efforts have already been put into the generalisation of building ground plans and on general surface simplification algorithms. Whereas the former is a typical domain of cartography, surface simplification is a widely used technique in the field of computer vision to speed up the visualisation of highly complex models. As of today, very little research has been done to extend existing model generalisation techniques to work with three-dimensional building models or to adapt surface simplification algorithms to the specific needs of buildings.
Both techniques – model generalisation and surface simplification – which are combined in our work will be discussed briefly.

2.1 Model Generalisation

Model generalisation is the transformation of objects into representations of simplified geometry, topology and semantics. An early approach for the simplification of building data is described in (Staufenbiel, 1973), where the outline of the building is based on the intersection of straight lines and a set of rules is proposed for which are too small for presentation. Recent approaches incorporate object-oriented structures and rules to strive for a more holistic solution. (Barrault et al., 2001) e.g. present a hierarchical multi-agent system where a set of agents are delegated to a building, each aiming to improve the overall situation with respect to some attached constraint. Such rules often require a minimal building size, rectification of angles and enlargement of narrow objects inside the building.
A general concept using least squares adjustment theory for the simplification of building ground plans was first introduced by (Sester, 2000). It allows for the introduction of observations in terms of constraints in order to determine unknown parameters in an optimisation process
Methods for the generalisation of 3d building data have also been proposed in recent years. (Meyer, 2000) suggests using a sequence of opening, closing and rectification operations to gain simplified data which can be used to build up a level of detail structure. A different approach is presented in (Coors, 2000): a well known surface simplification algorithm (as described in (Garland and Heckbert, 1997)) is extended to enhance significant features of the model and to aggregate only the less important ones.

2.2 Surface Simplification

In the field of computer graphics, computer vision and computational geometry, a wide range of surface simplification algorithms have been developed. A good overview is given by (Heckbert and Garland, 1997). Those methods are usually applied for the simplification of general objects which are either given as polygonal or as triangular surface meshes. The most important algorithms are either based on vertex clustering or edge collapse operations.
The algorithm introduced by (Rossignac and Borrel, 1993) divides the object's bounding volume into a regular grid of boxes and all vertices inside a cell are clustered together into a single vertex. A simplified model is then synthesized from the remaining vertices according to the original topology. The simplification algorithms presented by (Hoppe, 1996) and (Garland and Heckbert, 1997) both iteratively contract edges to simplify models, but differ in the underlying error metric measuring the geometric error introduced into the model by an edge collapse operation.
Another interesting simplification approach for general polygonal models is described in (Ribelles et al., 2001). Small features including bumps, holes, tabs, notches and decorations are isolated, ranked and removed using a splitting and hole filling operation.
Building regularity as requested for our problems have not been used so far.

3. ALGORITHM OUTLINE

The presented generalisation algorithm is designed for polyhedral three-dimensional building models. Without loss of generality, it is assumed that the 3D building model is given as a 2-manifold M, composed of a set of vertices V and a set of polygonal faces F. Each face may additionally contain a number of interior points, determining the parameters of the associated planar surfaces. These interior points are provided during data collection, e.g. from stereo measurements, or result from vertices that are removed during the geometric simplification of the building model. The algorithm uses these interior points in the least squares adjustment to resolve the new coordinates of vertices after each generalisation step. This approach ensures a minimum deviation of the generalised building model to every vertex of the original model and thus to its original shape.
Our algorithm is based on the fact that most walls are oriented in parallel to the principal axes of the building, which are again often rectangular. It can therefore be assumed that the faces of a building model are usually coplanar, parallel or rectangular to other faces in the same model. A generalisation must preserve these properties as correct as possible. For this reason, the presented algorithm considers the aforementioned properties between faces as constraints during the simplification process. As this information is usually not explicitly available for a building model, the first step in the generalisation algorithm is to create the so-called constraint building model, which is basically the polygonal building model enriched by a set of constraints.
These constraints are not stored for pairs of faces, however, as this would lead to a large number of constraints, but as a hierarchy of constraints. The lowest element in this hierarchy is the coplanarity constraint, which simply groups a set of faces together, each being coplanar to any other face in the same set within some given tolerance. Sets of coplanar faces are then again grouped together by a parallelism constraint if their faces are parallel to faces of another face set. Finally, two or three sets of coplanar or parallel faces are grouped by a rectangularity constraint if the faces of each set are rectangular to faces in the other two or three sets.
Following the generation of the constraint building model, the geometry of the model is iteratively simplified as depicted in Figure 2. First, a feature detection algorithm searches for features like extrusions, intrusions, notches, tips etc. and evaluates their significance to the overall appearance of the model. A feature removal step next eliminates the features of least importance, i.e. which only slightly influence the silhouette of the building. The feature removal step not only alters the geometry of the constraint building model, but also the constraints that are affected by it. This is important, as constraints become obsolete through the process of feature removal. For example, sets of coplanar faces may often be merged together after simplification. Vertices that are removed from the geometry, however, are not just discarded from the model, but the algorithm stores their coordinates as additional

interior points as mentioned in the beginning of this section. The last step of the algorithm uses least squares adjustment to find a new position for every vertex in the constraint building model in order to fulfil all its constraints.

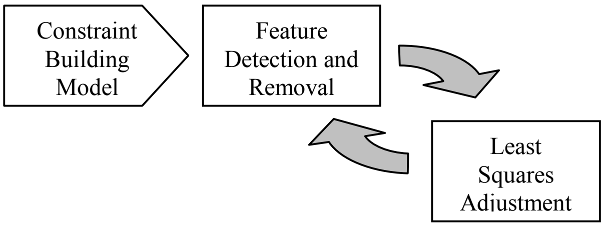

Figure 2. The generalisation algorithm iteratively simplifies the constraint building model using least squares adjustment to preserve the building regularities.

4. CONSTRAINT BUILDING MODEL

The initial step of the generalisation algorithm is to build the constrained building model, which stores constraints between two or more faces of the polyhedral building model. The following adjustment step of the algorithm is based on this information to optimise the coordinates of all vertices after each feature removal step. The simplification step on the other hand is designed to avoid violating constraints until they become obsolete. Simplification operations that are carried out on the building model also aim on preserving the coordinates of affected vertices and rely on the adjustment stage to determine their final position. Thus, the quality of the final, generalised building model directly depends on the quality of the constraints that are stored within the constraint building model.
It is the author's belief that not every constraint can be found by an automatic approach. Dependent on the quality of the input model, a number of constraints will almost always be missed due to errors introduced in the generation of the model. Those absent constraints might reduce the quality of the final model if missed in high quantities. An application should therefore offer the possibility to identify and insert more constraints into the constraint building model in a semi-automatic fashion to work around those errors and to improve the overall quality of the final building model. A semi-automatic tool also helps surveying the effects of certain constraints on the generalisation process by manually adding or removing those constraints.

4.1 Prerequisites

It is assumed that for each face F_i, a non-ambiguous plane can be computed using the coordinates of all its vertices and interior points. Each planar Face F_i is then given by the following plane equation:

$$F_i : A_i x + B_i y + C_i z + D_i = 0 \qquad (1)$$

where A_i, B_i, C_i is the normal vector of its plane and D_i the closest distance of the plane to the origin of the coordinate system. The angle ϕ between two planar faces F_1 and F_2 can then be computed using:

$$\phi = \arccos(A_1 A_2 + B_1 B_2 + C_1 C_2) \qquad (2)$$

The parameters and the angle between two planar faces are not only used to find properties between faces, but also to classify features of the building in later stages.

4.2 Properties between Faces

The algorithm identifies faces being coplanar, parallel and rectangular to other faces as described in the following subsections.

Coplanarity: Two faces are assumed to be coplanar if the angle between the normal vectors is close to 0° or 180° and the difference of the absolute value of the distances lies under a given threshold. It is worth mentioning, that coplanar faces may have normal vectors pointing in opposite directions. This is totally legal as the other constraints are expressed independently of the true direction of the faces.

Parallelism: Two faces are assumed to be parallel if the angle between their normal vectors is close to 0° or 180°.

Rectangularity: Two faces are assumed to be rectangular if the angle between their normal vectors is close to 90°.

4.3 Organisation of Constraints

As mentioned earlier in this paper, the algorithm does not store one constraint for each pair of faces as this would lead to a large number of constraints. Rather, the algorithm generates groups of faces, so that there exists a coplanarity constraint between any two faces placed in the same group. As each one of those face groups defines a unique plane inside the buildings own models space, the real goal of detecting coplanar faces is to find a minimal set of planes and to associate every face with exactly one of those planes.
The algorithm further groups two or more planes into sets of parallel planes if the faces in those planes are parallel to faces in the other planes. Finally, two or more (parallel) planes are grouped by a rectangularity constraint to yield the aforementioned hierarchy of constraints.

5. FEATURE DETECTION AND REMOVAL

In order to simplify the geometry of a building model, it is not sufficient to just remove arbitrary vertices or edges. Even if the introduced geometric error is small, the symmetry of the building model will irretrievably get disturbed. It is thus necessary to take notice to the regularity of the model during its simplification. Our feature detection and removal algorithm for generalisation allows the use of a manifold set of surface simplification operators, each designed to remove one specific class of feature types. In contrast to the rather simple operators used in traditional surface simplification algorithms, our operators remove entire features in one continuous process, while preserving the integrity of the remaining parts of the building model.
Prior to removing some feature, it must first be detected and identified, because each feature type requires its particular feature removal operator. Three classes of features types can be distinguished, each based on one of the three primitive types: the extrusion, the notch and the tip (Figure 3).
The presented algorithm detects features by looping through all primitives of the building model and by testing them against the

feature types of the particular feature class. Once the algorithm detects a feature, its impact to the appearance of the silhouette of the building model is evaluated. This can be a very complicated process as small features may be important due to their semantic meaning. At this point, we use a simple metric to measure a value which corresponds to the maximum distance moved by a vertex during the removal of the feature.

After feature detection is completed, the algorithm removes the features of lowest importance to simplify the geometry of the building model. In our example, an extrusion is removed by using a combination of edge collapse and edge foreshortening operations (see section 5.2). Then, the algorithm checks the validity of constraints between affected faces and updates them according to their new condition.

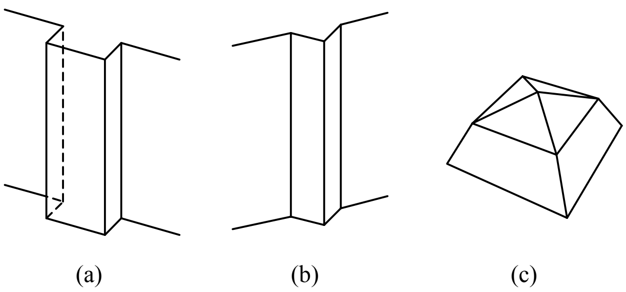

Figure 3. Feature detection distinguishes between face, edge and vertex based features: e.g. (a) extrusion, (b) notch and (c) tip.

5.1 Detecting Extrusions

Currently, our algorithm reliably detects and removes extrusions, which belong to the class of face based features types. Two examples for features that are based on edges and vertices are the notch and respectively the tip. The feature detection algorithm identifies an extrusion if the following two requirements are met for a face. First, the angle between the normal vector of the face and the normal vector of every neighbour face is within some given tolerance of 90°. This tolerance angle should not be chosen too high as faces that are used to approximate curved elements of the building model might erroneously be identified as extrusions. Second, all edges of neighbour faces that start or end at the front face must lie behind that face.

5.2 Removing Extrusions

For the removal of an extrusion, we use a combination of two operators, namely the edge collapse operator and the edge foreshortening operator. The edge collapse, or edge contraction, operator deletes an edge and merges its two endpoints into a single vertex (Figure 4). After this operation, the number of edges of the adjacent faces are reduced by one. If the adjacent faces happen to be triangles, they are completely removed from the model. The edge foreshortening operator on the other hand preserves all edges and faces, but moves one of its endpoints along the edge towards the other endpoint in order to shorten the length of the edge.

The removal operator for extrusions performs either an edge collapse or edge foreshortening operation to all edges that emanate from the front face of the feature (as depicted in sketches of Figure 5). Edges that have approximately the same length as the shortest candidate are collapsed into the base vertices (Figure 5b+c). The foreshortening operator must be used for edges that do not completely belong to the extrusion themselves (Figure 5d+e). For this it follows that edges longer than the shortest edge are shortened by this length. If no foreshortening operations are used for long edges, too much of the model geometry gets removed (Figure 5f). In a final step, the original front face of the extrusion is tested for coplanarity to its new neighbour faces and invalidated constraints are removed from the constraint building model.

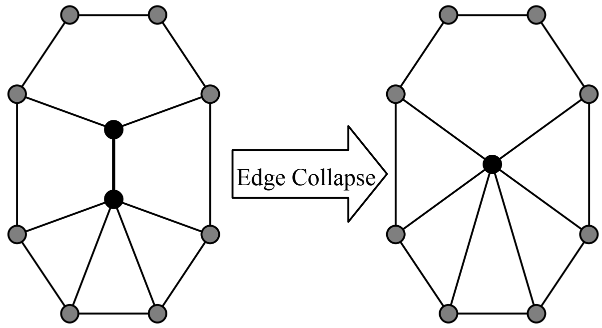

Figure 4. The edge collapse operation contracts the endpoints of the highlighted edge into a single vertex.

The original position of the front face vertices are not just discarded by the removal operator, but they are stored as additional interior points of the face. The next step of the generalisation algorithm, the least squares adjustment, uses these points to determine the new parameters of the planar faces of the building model.

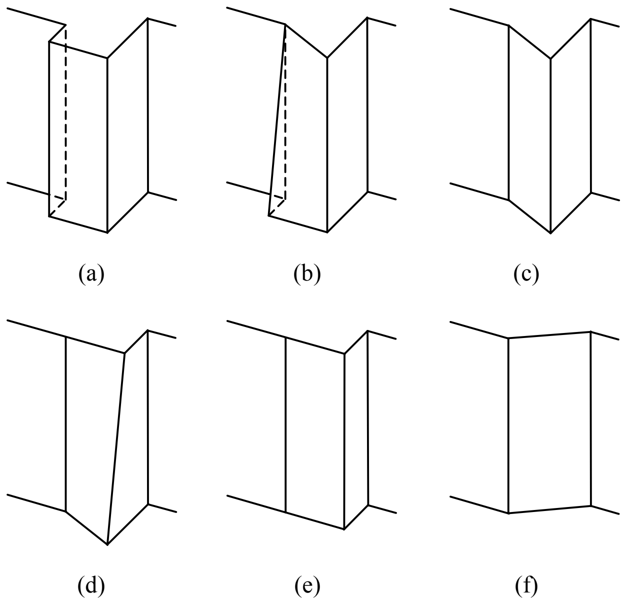

Figure 5. (a-e) Removal of an extrusion: The endpoints of the short edges are collapsed into their base vertices (b+c), whereas the longer edges are foreshortened by the same length (d+e). Collapsing all edges results in the removal of too much geometry (f).

6. LEAST SQUARES ADJUSTMENT

By feature removal, parts of the object are detected and completely eliminated from the dataset. The optimal shape of the reduced model, however, should still be determined by all original points, even though the number of planar faces is reduced by the preceding step. In order to resolve the final shape of the simplified model, a least squares adjustment is applied using the available constraints between the remaining faces as well as the points of the original model. Here the Gauss-Helmert model

$$\mathbf{Bv} + \mathbf{A\hat{x}} + \mathbf{w} = 0 \qquad (3)$$

is used. During adjustment, the parameters of the buildings' faces as well as their points of intersection are determined. For parameter determination of the faces, the following constraints are applied.

6.1 Coplanarity

The parameters of coplanar faces F_i are determined using the x_k, y_k and z_k coordinates of the original and interior points. Just as 6.2 and 6.3, the coplanarity constraint follows from the constraint building model.

$$F_i : A_i x_k + B_i y_k + C_i z_k + D_i = 0 \qquad (4)$$

Since the definition of the plane parameters for face F_i in (4) is over-parameterised, the norm of the vector A_i, B_i, C_i is integrated as an additional constraint during least squares adjustment.

$$A_i^2 + B_i^2 + C_i^2 = 1 \qquad (5)$$

6.2 Parallelism

For parallel groups of coplanar faces, the parameters A_i, B_i, C_i of the coplanar face group F_i are shared with the other n coplanar face groups parallel to F_i.

$$\begin{aligned} F_i &: A_i x + B_i y + C_i z + D_i = 0 \\ F_n &: A_i x + B_i y + C_i z + D_n = 0 \end{aligned} \qquad (6)$$

6.3 Rectangularity

For two rectangular groups of coplanar or parallel faces, we use the constraint

$$\cos\phi = \frac{A_1 A_2 + B_1 B_2 + C_1 C_2}{\sqrt{A_1^2 + B_1^2 + C_1^2}\sqrt{A_2^2 + B_2^2 + C_2^2}} \qquad (7)$$

As the cosine of 90° is 0, constraint (7) can be simplified to

$$0 = A_1 A_2 + B_1 B_2 + C_1 C_2 \qquad (8)$$

6.4 Point of Intersection

The new position of the remaining vertices of the model are determined by the intersection of three or more faces. In order to provide a complete solution, not only the planar surfaces, but also their points of intersection are integrated into the adjustment. This approach is additionally motivated by the fact, that the topological information about the intersection of the planar surfaces is not yet used. If this information is ignored, four or more planar surfaces are not guaranteed to intersect in one unique point after generalisation. Using a different weight value for each type of constraint, helps to exert influence on the adjustment, e.g. to favour unique intersection points over parallel planar surfaces.

The X_i, Y_i, and Z_i coordinates of the intersection point P_i are determined by using the following constraint for every plane F_k:

$$P_i : A_k X_i + B_k Y_i + C_k Z_i + D_k = 0 \qquad (9)$$

7. RESULTS AND FUTURE WORK

The algorithm above has been implemented and tested on polygonal building models of a 3D city dataset. In order to measure the complexity of each model, we used the number of triangles gained by triangulating the planar surfaces. The algorithm showed promising results on both complex and simple models. The complexity of the building models could in many cases be reduced by over 30%, in some cases, where the model exhibited a lot of extrusions, even by 50%. The model of the New Palace of Stuttgart (Figure 6), e.g., comprises of 721 planar surfaces, that make up a total number of 2730 triangles.

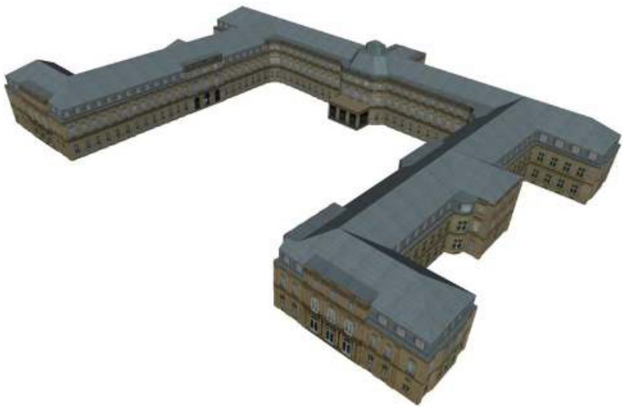

Figure 6. The New Palace of Stuttgart is used to show the results of our generalisation algorithm.

Our generalisation approach was able to detect 110 extrusions using three iterations. After removal of the extrusions, the model only comprised of 1837 triangles. The results are demonstrated in Figure 7 to Figure 10. Figure 7 shows part of the original model as it was captured from stereo imagery and an existing outline from the public Automated Real Estate Map, respectively. Figure 8 shows the result of the generalisation process. As it is visible, parallelism and rectangularity have been preserved for the remaining faces. Using textured models, as it is depicted in Figure 9 and Figure 10, this amount of detail is sufficient for visualisation in most cases.

As the general algorithm design proved to be correct, our future work mainly consists of defining more features types that can be detected and removed. Especially features that are based on edges and vertices have not yet been evaluated. More research has to be put into how complex features need to be dealt with and how curved elements of building models can be simplified.

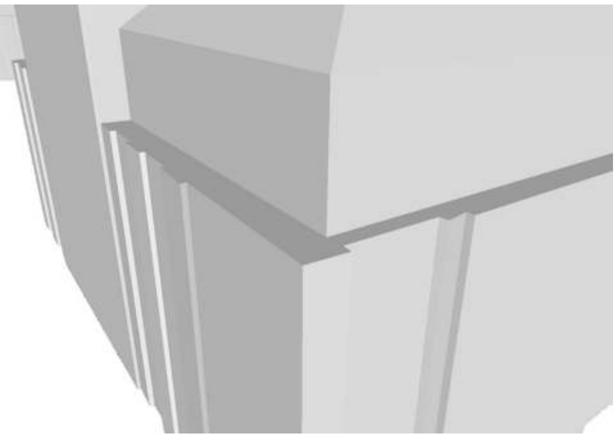

Figure 7. Part of the original building model

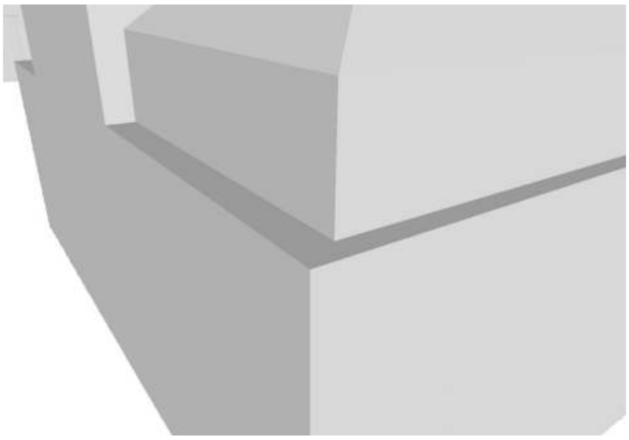

Figure 8. Part of the simplified building model.

Figure 9. Part of the original building model (textured).

Figure 10. Part of the simplified building model (textured).

REFERENCES

Baltsavias, E., Grün, A. and van Gool, L., 2001. *Automatic Extraction of Man-Made Objects From Aerial and Space Images (III)*. Swets & Zeitlinger B.V., Lisse, The Netherlands.

Barrault, M. et al., 2001, Integrating Multi-Agent, Object-Oriented and Algorithmic Techniques for Improved Automatic Map Generalization. In: *Proceedings of the 20th International Cartographic Conference of the ICA*, BeiJing, China.

Coors, V., 2001, Feature-preserving Simplification in Web-based 3D-GIS, 1st International Symposium on Smart Graphics, 2001, Hawthorne, NY, USA.

Garland, M. and Heckbert, P., 1997, Surface Simplification using Quadric Error Metrics. In: *Proceedings of ACM SIGGRAPH 97*, pp. 206-216.

Grün, A. and Wang, X., 2001, *News from CyberCity-Modeler. In: Automatic Extraction of Man-Made Objects from Aerial and Space Images (III)*. Swets & Zeitlinger B.V., Lisse, Netherlands, pp. 93-101.

Haala, N. and Brenner, C., 1999, Virtual City Models from Laser Altimeter and 2D Map Data. In: *Photogrammetric Engineering & Remote Sensing, Vol. 65, no 7*, pp. 787-795.

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

Hoppe, H., 1996, Progressive Meshes, In: *Proceedings of ACM SIGGRAPH 96*, pp. 325-334.

Mayer, H., 2000, Scale-Space Events for the Generalization of 3D-Building Data Adjustment. In: *International Archives of Photogrammetry and Remote Sensing, Amsterdam, Netherlands, Vol. XXXIII, Part B4*, pp. 639-646.

Ribelles, J. et al., 2001, Finding and Removing Features from Polyhedra, In: *Proceedings of DETC ´01*, Pittsburgh, Pennsylvania, USA.

Rossignac, J. and Borrel, P., 1993, Multi-resolution 3D Approximations for Rendering Complex Scenes. In: *Modeling in Computer Graphics: Methods and Applications*, Springer-Verlag, Berling, pp. 455-465.

Sester, M., 2000, Generalization Based on Least Squares Adjustment. In: *International Archives of Photogrammetry and Remote Sensing*, Amsterdam, Netherlands, Vol. XXXIII, Part B4, pp. 931-938.

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