

# OUT-OF-CORE TOPOLOGICALLY CONSTRAINED SIMPLIFICATION FOR CITY MODELING FROM DIGITAL SURFACE MODELS

Sebastian Möser, Roland Wahl, Reinhard Klein

Institute of Computer Science II, Universität Bonn  
53117 Bonn, Germany  
{moeser, wahl, rk}@cs.uni-bonn.de

**KEY WORDS:** DSM, City Model, Geometry Simplification, Abstraction, Visualization

## ABSTRACT:

We present a framework for rapid reconstruction of building models from very large, high-detail digital surface models (DSM) of urban areas. Our method is based on a geometric mesh simplification approach augmented with shape constraints (Wahl et al., 2008). This approach allows to abstract the full-featured DSM in such a way that important structural elements are maintained irrespective of the approximation accuracy.

In this paper we present two major extensions. Firstly, we deal with situations, where the original approach may generate artefacts due to incomplete or inconsistent structural information, mainly caused by vegetation close to the facades. We present refined topological constraints, which handle these problems and a filtering which neglects structural information that contradicts the mesh topology. Secondly, we extend the computational framework to be fully out-of-core capable and present a way to parallelize computations on multiple cores.

We demonstrate the efficiency of our method by showing results for downtown Berlin a dataset containing more than 1 billion height samples processed in less than 30 hours.

## 1. INTRODUCTION



Figure 1: Rendering of the automatically reconstructed city model of downtown Berlin on top of a corresponding DSM.

The amount of digital data available for cities has drastically increased in the recent years. The advances in sensor technology such as airborne LiDAR or stereo cameras combined with post-processing technologies have led to raw city data with point densities of about 200 samples per square meter.

Automatic reconstruction of abstracted city models from such data is a challenging task. On the one hand the detailed signal is demanding, as it allows to model finer structures, on the other hand the enormous size requires efficient, scalable, out-of-core methods (i.e. methods which do not rely on the whole dataset residing in main memory).

Approaches that reconstruct building models from raw data can be roughly categorized into two classes. Model-based approaches

try to fit a model to the data, and are therefore limited to a predefined set of building types, specific characteristics are not maintained. Data-driven approaches characterize a building by a set of primitive shapes (mostly planes) which can build almost arbitrary complex constellations. However, even the generation of the final model based on the detected primitives can be intricate for complex roof structures.

(Wahl et al., 2008) proposed a new data-driven method motivated by DSM simplification which circumvents the problem of model generation. Instead of defining a polyhedron for a given set of planes, they constrain a classical geometric mesh simplification approach to respect the previously detected planes. Although this approach achieves a good abstraction of complex roof structures while maintaining a high positional accuracy of roof edges, problems arise if the shape borders are not well represented in the data. The proposed solution to this problem was the introduction of topological constraints.

In this work we analyse how this approach can be extended to achieve good results also in the presence of severe noise and occlusion artefacts as those caused by vegetation at the borders. To this end we analyse situations that lead to artefacts and extend the concept of topological constraints to deal with such situations. Moreover, we show how this approach can be efficiently extended to handle out-of-core datasets efficiently and demonstrate this with a full-featured dataset of downtown Berlin.

## 2. RELATED WORK

While digital surface models based on laser point clouds usually reveal a higher positional accuracy and due to multiple returns generally allow an easier filtering of vegetation, DSMs based on stereo reconstruction (Hirschmüller, 2008) offer a matching photometric signal and therefore can be used directly in a visualization. Although many publications regarding building reconstruction explicitly mention Lidar data in their title most of the presented techniques can be directly adapted to stereo reconstruction results.

## 2.1 Automatic City modeling

Starting with the availability of airborne lidar data a lot of research was devoted to automatic building reconstruction based on such data. While the model-driven approach (Weidner and Förstner, 1995, Maas and Vosselmann, 1999) is better suited for simple models. As it easily allows to put constraints on the models it is especially suitable for coarsely sampled buildings. Subsequent model-based approaches tried to enlarge the class of possible buildings, recently (Arefi et al., 2008) proposed a projection along dominant axes determined by morphology. Although such enhancements increase the set of possible models, the main restriction remains. The data-driven approach, based on segmentation as in (Rottensteiner and Briese, 2002, Dorninger and Pfeifer, 2008) is generally able to maintain all sorts of features, as it does not need to understand all parts of a model. However, as (Tarsha-Kurdi et al., 2007) points out the main disadvantages of the data-driven approach are sensitivity to detail features and noise as well as high computational costs.

## 3. OVERVIEW

As mentioned above the presented framework extends the approach by (Wahl et al., 2008) who combine geometric simplification with semantic constraints, derived from detected primitive shapes, to reconstruct building models from a full featured DSM. Therefore, we first review the basic approach and afterwards roughly analyse the problems and outline proposed solutions in section 3.3.

Generally the processing pipeline from a full-featured DSM to the final building models consists of two stages. In the first stage, primitive shapes of a desired fidelity are detected in the height-field to identify important features, namely prominent edges and corners, and how they are connected. Subsequently, an abstracted city model is computed by simplifying the triangulated DSM with respect to the detected shapes and geometric error. This combination allows high abstraction in a geometric sense while details like roof structures are accurately preserved. Finally, the models' connected components above the ground are automatically extracted and textured by projecting the image mosaic.

### 3.1 Shape Detection

Although our framework would support any type of shape information, we use primitive shapes to detect important features in the height-field, as they can handle noise and outliers in a robust way and directly provide additional topological information, which is later exploited in the simplification process. We employ the RANSAC based shape detection proposed by (Schnabel et al., 2007) as it allows us to efficiently process huge point-clouds. The height-field is converted into a 3D point-cloud with additional points at discontinuities. For these we use the same sampling density, as in the planar domain, to maintain a constant ratio between number of points and surface area. As facades are badly represented in a DSM and thus often suffer from noise and reconstruction errors, coarse detection parameters are needed to avoid the decomposition whole facades into many small segments. On the other hand, fine parameters are needed to accurately capture fine roof structures. We account for this issue by splitting the point-cloud into roofs and facades, which allows us to adjust the used parameter sets for both domains independently. The shape detection partitions the point-cloud  $P$  into disjoint subsets  $S_i$ , which are associated to a plane  $\Phi_i$ , and a subset  $R$  containing all points not assigned to a shape. Points are compatible to a shape candidate if their Euclidean distance and normal deviation to the shape

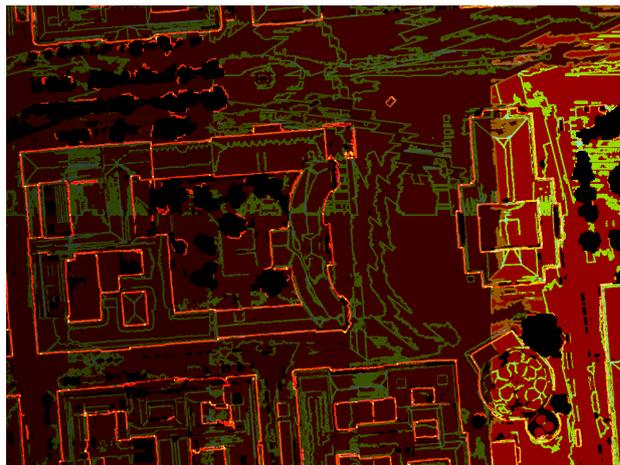


Figure 2: Example of a Shape Map depicted as RGB values: intensities correspond to shape-IDs, reddish color means one associated shape, green and orange regions belong to two shapes, and bright yellow or bluish colors denote regions with three or more shapes.

is smaller than a defined threshold. Additionally, shapes need to have a minimal number of compatible points to be accepted. After removing the compatible points, the algorithm is restarted on the remaining points until no more shapes can be found for the given set of parameters. With the low-level constraints described above, we are able to control the size and accuracy of the detected shapes and thus the maximal possible degree of abstraction. For further details we refer to the original publication.

To identify edges and corners in the DSM, we create an additional raster layer (shape map, see figure 2) which stores the globally unique indices of associated shapes for each vertex of the height-field. The shape map for a vertex  $v$  is defined as  $\zeta(v) = \{i | \exists w \in S_i : d(v, w) < \epsilon\}$ , where  $d(x, y)$  is the Euclidean distance of both vertices and  $\epsilon$  a predefined distance threshold. Vertices  $v$  are defined as edge vertices if  $|\zeta(v)| = 2$  or as corner vertices if  $|\zeta(v)| > 2$ . Now we can define the edge of two shapes  $\Phi_i$  and  $\Phi_j$  as  $Edge(\Phi_i, \Phi_j) = \{v \in DSM | i \in \zeta(v) \wedge j \in \zeta(v)\}$ . As can be easily observed in figure 2 these shape edges will generally not form simple chains of vertices but elongated sub-meshes. The same holds for corners. This is the main reason for the definition of topological constraints which deal with situations along the borders of such edge or corner meshes.

The shape map is used in the subsequent simplification and allows us not only to identify important features, but to also determine the involved shapes.

### 3.2 Constrained Simplification

Our simplification framework is build around the edge-collapse operation with tight upper bounds on the Hausdorff distance against the original mesh. Each edge of the original mesh generates two halfedge-collapse candidates. These collapse candidate are then checked for validity, that is whether it introduces flipping of orientations or degeneration of neighboring triangles, and scheduled in a priority queue keyed to its approximation error. For the sake of speed we use the quadric error metric (Garland and Heckbert, 1997) for computing priorities as it is fast and easy to compute. After that, iteratively the best collapse candidate is evaluated, this time using the actual distance metric. If the Hausdorff distance does not surpass the current error threshold, the halfedge-collapse will be applied to the mesh. As it changes the appearance of

its 1-ring, all conflicting candidates are rescheduled or deleted from the priority queue. This process comes to an end when each remaining valid collapse operation surpasses the threshold and therefore the bottom-up simplification scheme is in a local optimum. Although this approach is greedy, it is able to collapse a mesh completely, if the distance threshold allows it (i.e. it does not get stuck in a local minimum).

The basic rule in the constrained simplification is, that a vertex  $v$  may only collapse to another vertex  $w$  if the shape-IDs  $\zeta(v)$  are a subset of the shape-IDs  $\zeta(w)$ . Assuming we have a perfect segmentation of our DSM into planes, meaning all planes are bounded by edges which end in corners, this constraint ensures that vertices stay inside their associated planar domains. Edge vertices have two domains and to stay in both, they may only collapse along the edge, until they reach a corner. Since corners have three or more shapes to respect, they are only allowed to collapse to other vertices of the same corner, which may exist as we propagated the shape information in an  $\epsilon$  region around the shape. Problems arise if the shape information is not perfect, e.g. edges are discontinuous or corners are missing. If allowed by the geometric error, edges and whole shapes can shrink to a single vertex if they are not constrained in all directions, as collapses along the edges and inside the shapes are legitimate. Figure 3 depicts two critical cases.

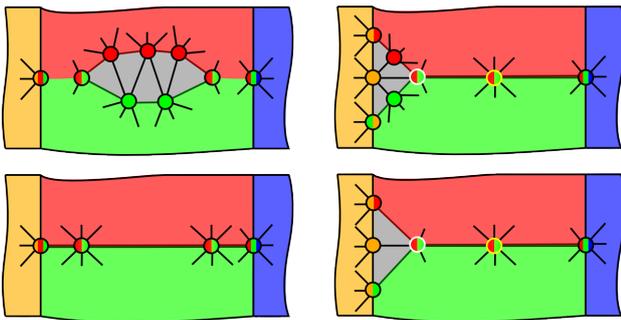


Figure 3: Two cases with incomplete shape constraints. Small circles represent vertices with color coded shape assignments. Left: discontinuous edge, collapsing of red or green vertices could lead to visible artefacts, Right: the implicit corner, marked as the dot with the white outline must not collapse to the incident border vertex (yellow outline). Upper row shows the initial constellation, lower row shows the allowed solution with topological constraints.

In case of incomplete edges (Wahl et al., 2008) propose to restrict the topological shape borders. These are defined as those vertices with one incident edge pointing to a vertex not in the same shape. To simplify along the shape border, vertices may only collapse to neighboring border vertices. This effectively allows to reduce the complexity even of defective edges if the both restricting corners are present. But if a corner is missing, the involved edges are only connected via a series of border vertices, the shapes still degenerate. To cope with this issue, additional implicit corners are defined as those edge vertices which only have one neighboring border vertex in the same edge with respect to one of their shapes and treated as regular corners (see figure 3).

### 3.3 Problem Analysis

The basic method provides two orthogonal ways of defining the result. The first is to steer the segmentation result and the second is to set the simplification accuracy. However, if the segmentation results are intended to give a coarse approximation, a lot of

constraints disappear and therefore a high geometric error threshold can easily lead to visible artefacts. This circumstance makes it difficult to filter out large unwanted regions, especially vegetation and still maintain a high-quality building model. Therefore, we analysed the situations in the presence of incomplete shape information and extended the concept of topological constraints to avoid potential sources of artefacts.

When dealing with whole cities instead of small regions, we run into serious problems considering the quality of the shape map. Firstly, huge surfaces, e.g. ground surface, are segmented into several shapes. These slightly different planes introduce additional edges, subsequently called pseudo edges, which can be very broad (see the greenish areas in Fig. 2). Merging both planes to avoid such edges violates the assumption, that the maximal possible error of inner shape collapses is bounded by the error threshold used to detect the planes. While this might not pose a problem for two shapes, the systematic error may accumulate and cause artefacts if many such planes are merged.

Secondly, we often have vegetation close to buildings, which can cause holes in the shape map and inhibit the robust detection of edges between ground and buildings. In this case, but especially in combination with the aforementioned pseudo edges, the proposed constraints are too weak to avoid undetected facade vertices to be pulled away. To deal with these issues, we define the semantic constraints in a more general and stricter form in section 4., which also includes a more general notion of borders and corners. As these only restricts vertices associated to a primitive shape, artefacts caused by collapsing vertices without shape information are still possible. As shown in figure 5, these can be caused by collapse operations, that establish a connection between non neighboring shapes, subsequently called tunnel. The prevention of such tunnels is discussed in section 4.2.

Additionally, the accuracy of the height-field differs between buildings or even within building parts. While the representation of roof-structures in a digital surface model is usually quite accurate, facade edges are often jagged or even appear twice. The first case can be easily handled with appropriate shape detection parameters, but the second is only covered by the original approach if the facade edges are continuous. To remove these double facades, which are usually very short and thus appear as spikes, we propose an additional filtering in section 4.3.

Another drawback of the original approach is its limitation to small datasets that completely fit into main memory, which inhibits the application to whole cities. To remedy this drawback, we modified all stages of the processing pipeline to efficiently run out-of-core with a small memory footprint. Moreover, for practical purposes we introduce a high-level parallelization scheme which gains a speed-up in the order of the employed number of cores.

The contributions of this paper are

- more general topological constraints to avoid artefacts
- a tunnel constraint for better results at unclassified vertices
- a topological filtering, leading to simpler models
- a processing schema, allowing for whole cities to be processed
- a parallelization schema, improving the efficiency

## 4. TOPOLOGICAL CONSTRAINTS

### 4.1 Topological Corners and Edges

Although we obtain satisfying results using the original definition of edges and corners (implicit and explicit) for a moderate geometric error, high error bounds which are needed for a strong abstraction often cause severe artefacts in combination with missing shape information. Especially in regions with glancing intersections or multiple borders of the same shape meeting in a common vertex, vertices of undetected edges can be pulled away from facades. We illustrated these effects in figures 5 and 6. Both are very common situations in a full-featured DSM. The first is caused by intersections of nearly coplanar planes, which often happens in the segmented ground surface, resulting in large areas that are classified as edges. If these edges are not completely bounded, implicit corners are only detected where the borders of both involved shapes diverge, while vertices in between may move inside. The second is usually a result of missing shape information and may also result in the demolishing of not explicitly detected edges.

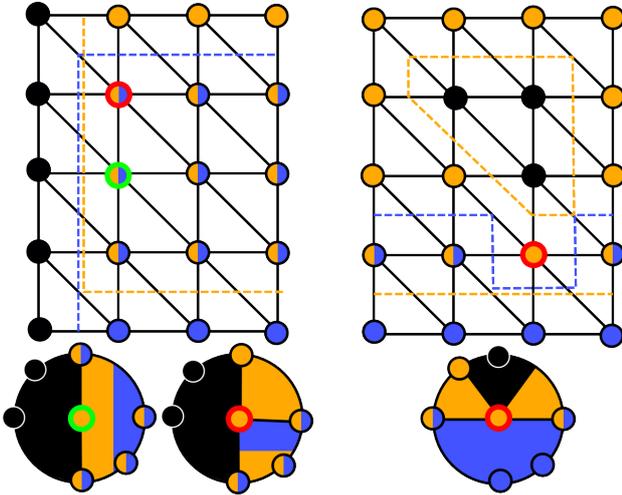


Figure 4: Schematic depiction of critical shape constellations. Left: broad edge, green vertex may collapse along border, red vertex not (implicit corner), Right: red vertex may not collapse because of multiple incident orange borders, lower row: shows the corresponding shape assignment of the highlighted vertices in the one-ring

In both cases the shape topology gives additional hints about the form of shapes and the building structure to prevent unwanted collapses. Therefore, we introduce a more general notion of corners and edges from a topological point of view. Instead of considering planes as regions bounded by edges and corners, we use the dual view in which edges and corners are implicitly defined by the intersection of bounded planar regions. Using this notion, it is quite obvious that only simplifications should be applied which respect the borders of all shapes simultaneously. The border of a shape is always respected by collapses of vertices inside the shape and for border vertices if they collapse to a neighbor vertex of the same border. In this sense, we define topological edges as the intersection area of two shapes, which is similar to the previous definition of edges. In contrast, we define topological corners as those vertices which can not collapse to any neighbor vertex without disrespecting a shape border or which connect multiple borders of the same shape. This means that two criteria have to be checked when collapsing border vertices. Firstly, the collapse operations have to be valid for all involved shapes, other-

wise those shapes become demolished. And secondly, all borders meeting in the vertex which is to collapse, must enter and leave the one-ring in the same two neighbors. Otherwise, borders are diverging or ambiguous in the considered vertex, which classifies it as a topological corner.

### 4.2 Treatment of Points without Shape Information - Tunnel avoiding

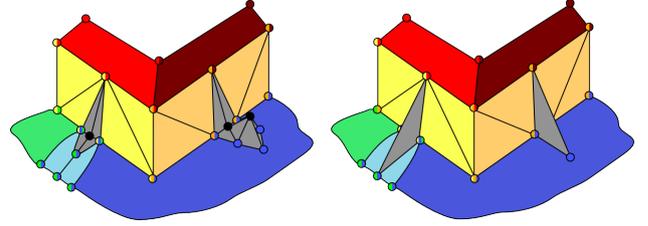


Figure 5: Left: initial situation, pseudo edge connected to the left facade, the lower edge of the right facade is discontinuous due to vegetation. Right: possible result with the original approach, black dots mark vertices without shape assignment

To obtain a strong abstraction, we inevitably introduce holes in the shape map, as fine building details are discarded. The same is true for vegetation or other unwanted objects like cars, which may be either not detected or masked out. Vertices without any shape information have to be treated with great care. Interpreting such vertices as jokers, which can collapse to any other neighbor, even if surpassing the geometric error, might lead to severe artefacts (see figure 6). Especially when dealing with vegetation this can cause a connection between buildings and nearby trees, which is not the desired result. Although we have no direct assignment to a specific shape, we can at least discard certain collapses that conflict with the shape topology in the neighborhood. When collapsing such a vertex, we connect vertices that were previously separated. As these may be associated to a shape, we may also connect shapes, that are not directly connected in the dataset. In the case of vegetation near facades, such connections often cause a direct connection between a ground vertex and a roof vertex. We call such a connection tunnel. The solution is to find a legitimate connection between the corresponding shapes and let the unassigned vertex only collapse to such legitimate connection vertices. Technically, a collapse is valid if the destination vertex has a common shape with each neighbor of the source vertex. In situations as illustrated in figure 5 this effectively avoids artefacts.

### 4.3 Topological filtering

Double facade segments, usually visible as spikes (see figure 7), are hard to remove. Often they wrongly belong to a roof shape, which prevents a collapse to ground shapes. Even worse they can also provoke simplification deadlocks in parts nearby, because geometric errors such as flipping of orientations or the degeneration of neighboring triangle may be introduced. To remove these double facades, we propose a topological filter, which deletes shape-IDs if they are not supported by a neighboring triangle. For regular shapes this is always the case, whereas isolated, unbounded parts shrink during the simplification process until only a single line or vertex remains, which will then be removed.

But even if allowed by all constraints, such collapses would cause a huge Hausdorff error and thus are forbidden if only a moderate error is allowed. In this case we could increase the possible geometric error for such vertices, but since we also have similar spikes at vertices without erroneous shape information. Since we

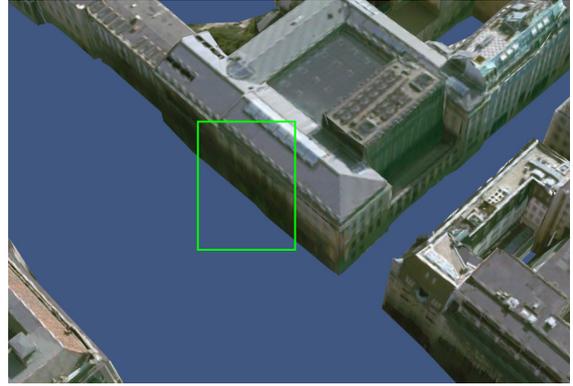
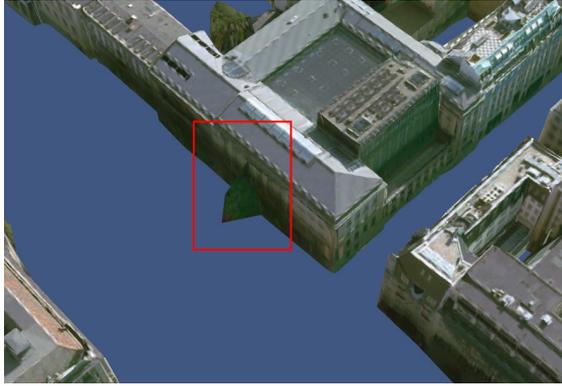


Figure 6: Simplification results. Left: without tunnel constraint. Right: with active tunnel constraint. Less artefacts are caused by vegetation.

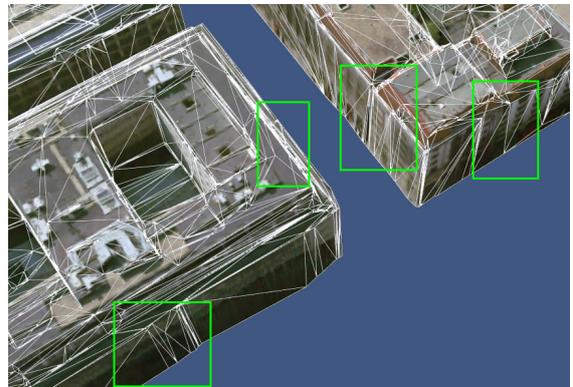
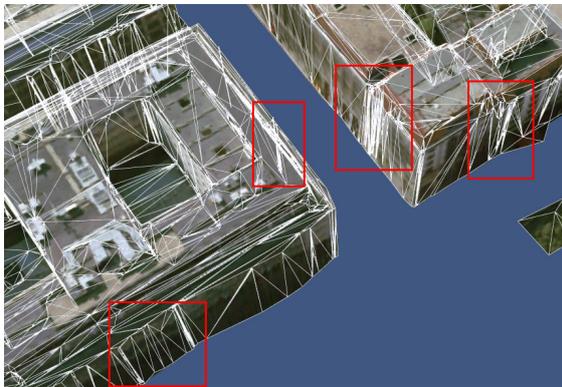


Figure 7: Simplification results. Left: without topological filtering. Right: with topological filtering. Removing spikes drastically decreases the triangles count.

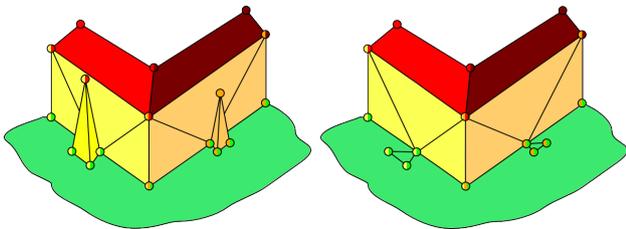


Figure 8: Left: Two spikes at roof-edges, the left is protected by shape constraints and the right one only by its size, Right: spikes are removed using the topological filtering

detected planes with a given accuracy, we assume that the error introduced by inner shape collapses is tightly bounded. As the overall shape is preserved by the detected edges and corners, we can safely increase the possible simplification error for non border vertices until all artifacts are removed.

## 5. OUT-OF-CORE, PARALLEL COMPUTATION

We achieve both out-of-core and parallel computation by cutting the dataset into smaller pieces either using a regular grid or a quadtree hierarchy. Since all steps in the pipeline depend on neighborhood information, e.g. shapes in the proximity of a point, or need consistent transition between pieces, we can not treat them independently. Generally to cope with this issue, cells are processed consecutively line by line from west to east, while data from neighboring cells in the east and south is included in

the computation. Changes that influence unprocessed cells are propagated to ensure consistent transitions. Applied to shape detection, we use a regular grid whose cell size has to be at least as big as the maximal shape size to detect. To guarantee such shapes to be completely inside the considered point-cloud, points from the three neighbors in east and south direction are included in the detection. Shapes that intersect the cell borders are propagated to the corresponding cells while the shape connection points in the intersection region are preserved to allow a correct continuation. In consecutive cells, existing shapes are continued and the algorithm restarts.

The same approach is applicable to the constrained simplification. We first partition the domain of the DSM using a quadtree. In the finest level, each cell contains a quadtree triangulation of that part of the height-field. The simplification works from level to level from fine to coarse. First, the finest level is simplified in the above mentioned order under the constraint, that north and west borders are left untouched whereas south and east borders have to apply the same collapses in both cells. In the next level, the simplified meshes of the previous level are stitched together and likewise simplified. This is continued until the coarsest level is completed. While this greedy approach allows us to process huge datasets, it complicates the computation of cells in parallel. In order to obtain correct results, we have to ensure correct data propagation and that no two threads access the same cells. As we have only forward dependencies in two directions, this can be achieved if each thread processes a different row and only moves east if the previous thread has a column distance of two cells (see figure 9). Once the two cell distance is established it does no longer cause synchronisation overhead if we assume equal pro-

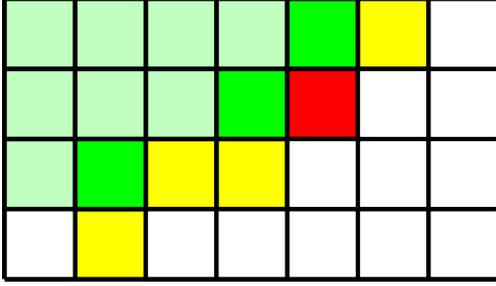


Figure 9: Example of parallel computation: Three cells are computed in parallel (green). The thread associated with the second row is in conflict with the first row as it tries to access the same cell (red). It will wait on the first thread to finish that cell. Needed neighbor cells without a conflict are marked yellow.

cessing time for each cell. Thus, synchronization overhead is negligible if the number of processors is small compared to the edge length of the dataset.

## 6. RESULTS

With the above mentioned techniques, we are able to rapidly reconstruct entire cities from full-featured DSMs. To evaluate our method, we used an infinite geometric error in the subsequent simplification, which leaves only the influence of our topological constraints. The experiment was performed on an intel Core 2 Quad processor with 8GB of RAM. However, the actual memory footprint only depends on the number of vertices in the current active cells as well as the cache size and cell size. For the example the memory footprint was consistently below 200MB. However, these parameters can be adapted if memory cost is an issue.

The test dataset is a  $6.2km^2$  digital surface model with a resolution of  $7cm$  per pixel of downtown Berlin, Germany. Figure 10 gives an overview of the reconstruction result of the complete dataset. The height-field, in combination with additional points for the facades, contained 1.6 billion points (35.86 GB), in which we detected 11,945 facade and 26,802 roof planes. This took 777 minutes and additional 257 minutes for the computation of the shape map. The simplification took 680 minutes, which equals 30,000 vertices per second and is thus approximately 2.15 times faster than the processing of the 4 million points dataset with the original in-core implementation which had less constraints and no parallelization. Consequently, we can state that the proposed out-of-core approach is truly scalable. Counting all steps together, the whole dataset was processed in 28.6 hours, which is approximately 4.6 hours per square kilometer.

Figure 11 illustrates the high quality of our results even for complex buildings. The left image shows the results without topological constraints. Although all buildings are recognizable using pure geometric simplification, the result is far from satisfying as important edges and corners are demolished, which causes severe visual artifacts especially when textured. In the right image we see that our proposed constraints effectively preserve feature edges with high positional accuracy. Figure 7 depicts the effects of our topological filtering. The white lines visualize the underlying triangle mesh, which is due to artifacts in the height-field very fine in the left image. As the right image shows, the topological filter is able to significantly coarsen the mesh without introducing artifacts. In Figure 6 we see a situation where a tree grows close to a building causing a discontinuity in the detected edge of facade and ground. Additionally, a glancing intersection prevents the tree from being absorbed by the facade. As shown



Figure 10: Simplification results: Whole dataset

in the right image, the tunnel constraint inhibits the collapse of the undetected part of the edge causing the tree to stay separated. Unfortunately, if in such a situation the vegetation is directly connected with a buildings, as we can see at the cathedral in Figure 11, we are not able to remove these vegetation parts, as there is no edge to preserve.

Figure 7 depicts the effects of our topological filtering. The white lines visualize the underlying triangle mesh, which is due to artifacts in the height-field very fine in the left image. As the right image shows, the topological filter is able to significantly coarsen the mesh without introducing artifacts. In Figure 6 we see a situation where a tree grows close to a building causing a discontinuity in the detected edge of facade and ground. Additionally, a glancing intersection prevents the tree to be absorbed by the facade. As shown in the right image, the tunnel constraint inhibits the collapse of the undetected part of the edge causing the tree to stay separated. Unfortunately, if in such a situation the vegetation is directly connected with a buildings, as we can see at the cathedral in Figure 11, we are not able to remove these vegetation parts, as there is not edge to preserve.

## 7. CONCLUSIONS

We proposed a framework for rapid reconstruction of city models from full-featured DSMs. We extended the original approach with efficient out-of-core algorithms to handle huge datasets and in combination with parallel computation on multi-core CPUs are able to process a complete city in a matter of hours.

The refined topological constraints improve the already good results of the previously proposed constrained simplification especially in the presence of severe noise and errors in the height-field or only partially detected features. Nevertheless, since we not only reconstruct roof-structures but also the facades, our approach may still produce artefacts if these facades are partially not represented in the DSM. Although the number of possible constellations in the 1-ring is finite if we consider the vertex degree bounded and only deal with up to 4 different shapes per vertex. It contradicts the simplicity of the chosen approach if we would define rules for very complex situations. After all, we rely on a greedy simplification mechanism, which will be disturbed in its effectiveness if too many interdependencies of collapses appear.

In the future we plan to extend our approach to other primitive shapes, such as spheres or cylinders, to achieve more accurate re-

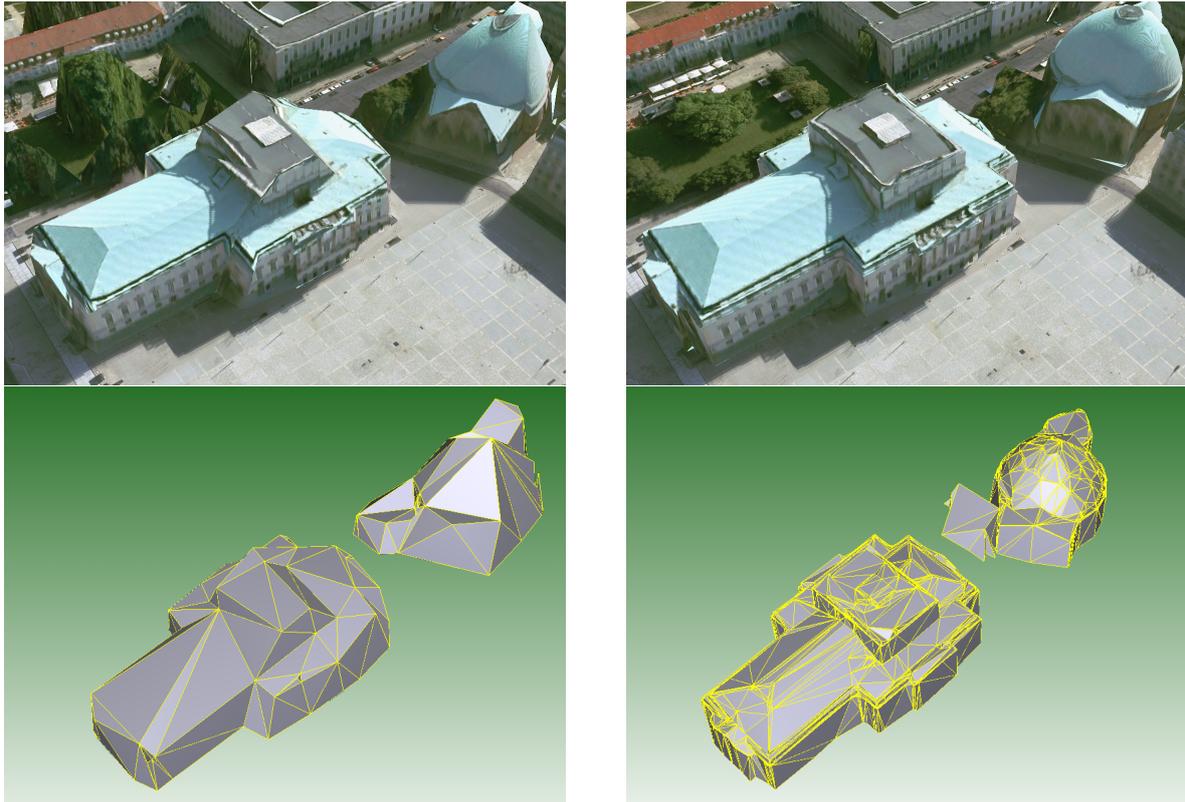


Figure 11: Simplification results. Left: unconstrained. Right: With topological constraints. Important roof-structures are preserved.

sults for buildings with curved features. Another research direction will be the specification of LODs for primitive shapes and thus important features.

#### ACKNOWLEDGEMENTS

We thank the DLR - Institute of Robotics and Mechatronics and RSS - Remote Sensing Solutions GmbH for providing us the high-detailed dataset of the city center of Berlin. This work was partially funded by the German Research Foundation DFG as a part of the bundle project “Abstraction of Geographic Information within the Multi-Scale Acquisition, Administration, Analysis and Visualization”.

#### REFERENCES

- Arefi, H., Engels, J., Hahn, M. and Mayer, H., 2008. Level of detail in 3d building reconstruction from lidar data. ISPRS Congress Beijing 2008, Proceedings of Commission III 37, pp. 485–490.
- Dorninger, P. and Pfeifer, N., 2008. A comprehensive automated 3d approach for building extraction, reconstruction, and regularization from airborne laser scanning point clouds. *Sensors* 8(11), pp. 7323–7343.
- Garland, M. and Heckbert, P. S., 1997. Surface simplification using quadric error metrics. In: *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, pp. 209–216.
- Hirschmüller, H., 2008. Stereo processing by semiglobal matching and mutual information. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 30(2), pp. 328–341.
- Maas, H.-G. and Vosselmann, G., 1999. Two algorithms for extracting building models from raw laser altimetry data. *ISPRS Journal of Photogrammetry and Remote Sensing* 54(2/3), pp. 153–163.
- Rottensteiner, F. and Briese, C., 2002. A new method for building extraction in urban areas from high-resolution lidar data. *IAPRS International Archives of Photogrammetry and Remote Sensing and Spatial Information Sciences* 34(3A), pp. 295–301.
- Schnabel, R., Wahl, R. and Klein, R., 2007. Ransac based out-of-core point-cloud shape detection for city-modeling. *Schriftenreihe des DVW, Terrestrisches Laser-Scanning (TLS 2007)*.
- Tarsha-Kurdi, F., Landes, T., Grussenmeyer, P. and Koehl, M., 2007. Model-driven and data-driven approaches using lidar data: Analysis and comparison. In: U. Stilla, H. Mayer, F. Rottensteiner, C. Heipke and S. Hinz (eds), *PIA07: Photogrammetric Image Analysis*, International Society for Photogrammetry and Remote Sensing, pp. 87–92.
- Wahl, R., Schnabel, R. and Klein, R., 2008. From detailed digital surface models to city models using constrained simplification. *Photogrammetrie, Fernerkundung, Geoinformation* pp. 207–215.
- Weidner, U. and Förstner, W., 1995. Towards automatic building extraction from high resolution digital elevation models. *ISPRS Journal of Photogrammetry and Remote Sensing* 50, pp. 38–49.