3D BUILDING RECONSTRUCTION FROM LIDAR BASED ON A CELL DECOMPOSITION APPROACH

Martin Kada[a], Laurence McKinley[b]

[a] Institute for Photogrammetry, University of Stuttgart, Geschwister-Scholl-Str. 24D, 70174 Stuttgart, Germany
martin.kada@ifp.uni-stuttgart.de
[b] Virtual City Systems, Zellescher Weg 3, 01069 Dresden, Germany
lmckinley@virtualcitysystems.de

Commission III, WG III/4

KEY WORDS: LIDAR, Reconstruction, Building, Automation, Algorithms

ABSTRACT:

The reconstruction of 3D city models has matured in recent years from a research topic and niche market to commercial products and services. When constructing models on a large scale, it is inevitable to have reconstruction tools available that offer a high level of automation and reliably produce valid models within the required accuracy. In this paper, we present a 3D building reconstruction approach, which produces LOD2 models from existing ground plans and airborne LIDAR data. As well-formed roof structures are of high priority to us, we developed an approach that constructs models by assembling building blocks from a library of parameterized standard shapes. The basis of our work is a 2D partitioning algorithm that splits a building's footprint into nonintersecting, mostly quadrangular sections. A particular challenge thereby is to generate a partitioning of the footprint that approximates the general shape of the outline with as few pieces as possible. Once at hand, each piece is given a roof shape that best fits the LIDAR points in its area and integrates well with the neighbouring pieces. An implementation of the approach is used now for quite some time in a production environment and many commercial projects have been successfully completed. The second part of this paper reflects the experiences that we have made with this approach working on the 3D reconstruction of the entire cities of East Berlin and Cologne.

1. INTRODUCTION

3D building reconstruction has been a topic for quite some time now. Many research papers have been published; commercial services and software are available. (Brenner, 2005), e.g., gives a good overview of reconstruction methods and points out that "research is still far from the goal of the initially envisioned fully automatic reconstruction systems". This situation has not yet changed much, although a lot of research is still devoted to this topic, as can be seen in the multitude of recent publications (e.g. (Arefi et al., 2008), (Möser et al., 2009), (Sohn et al., 2008)).

The subject of this paper is on the generation of realistic 3D city models in LOD2 as it is defined in the official OGC standard CityGML (see e.g. (Kolbe, 2009)). At this LOD, buildings have distinctive roof structures and flat facades that are textured from terrestrial or oblique aerial images.

As the data basis, we rely on existing ground plans and airborne LIDAR data. A frequent requirement, especially from customers within the mainland Europe, is that the provided building outlines are to be preserved with only little tolerance and that ridge and eaves heights must be very accurate. This is especially important so that the facades and roofs can be properly mapped from oblique aerial images.

The presented reconstruction approach is motivated from our research on the simplification of 3D building models for map-like representations (Kada, 2007). An integral part of this work lies on a new method to decompose a 2D building footprint into a small set of nonintersecting primitives. Although the resulting partitioning only approximates the original outline, it is still accurate enough for reconstruction purposes. The benefit is, however, that the algorithm separates the sections nicely, especially for residential houses with gabled or hipped roofs. This eases the task of determining and assembling a valid roof structure from parameterized, standard shapes.

In the second part of the paper, we give insight into two large-area projects that we have completed using the described 3D reconstruction system: East Berlin and Cologne. Figure 1 shows the reconstructed 3D city model of Berlin with textures mapped from oblique imagery.

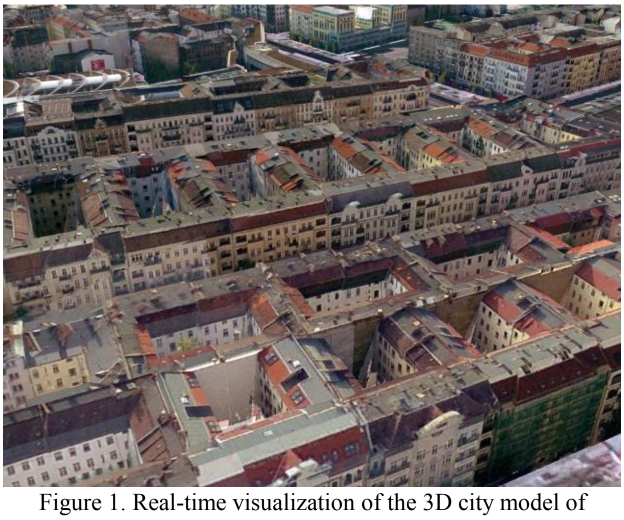

Figure 1. Real-time visualization of the 3D city model of Berlin.

2. RECONSTRUCTION ALGORITHM

In our approach, we assume that the majority of residential houses have either one main section or multiple connected sections, with additional smaller extensions, and that a partition thereof can be properly derived from the outline polygon. Once such a partition is found, a general geometrical description of the roof can be constructed by assigning a parameterized standard shape to each section. However, the difficulty to generate correct facade and roof shapes from a partition increases with the number, shape and arrangement of its elements. We therefore generate a set of non-overlapping, mostly quadrilateral shaped polygons that together approximate the original footprint (cp. Figure 2). Other ground shapes may also occur, but those primitives are then restricted to only bear certain roof shapes.

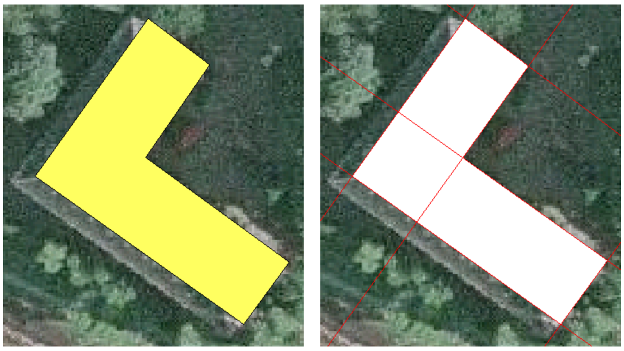

Figure 2. Building footprint and its decomposition into cells.

The roof is then reconstructed by determining a shape for each cell from the LIDAR points with regard to the neighbour cells (cp. Figure 3). After identifying the points inside a cell, the normal vectors from the local regression planes of the points are tested against all possible shapes. Here, only the orientation is used to speed up comparing the many shapes we support. The one that best fits is then chosen and its parameters estimated from the 3D point coordinates. Cells whose neighbour configurations suggest corner-, t- and cross-junctions are examined again and replaced if a junction shape can be fitted according to the neighbour shapes and parameters.

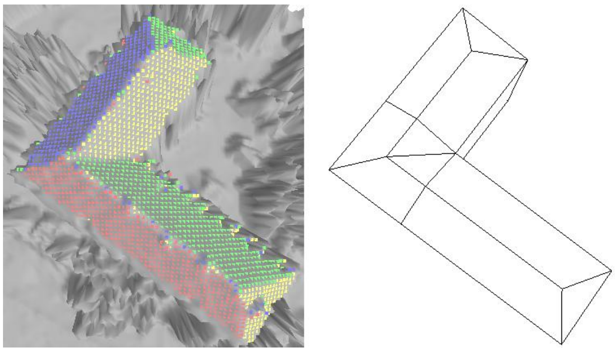

Figure 3. LIDAR points inside the cells coloured according to their local regression plane and the best fitting roof shapes.

After the geometric reconstruction, the building models are textured from oblique aerial images. Any lack of geometric detail that is due to our rather restricting model oriented approach is then hardly noticeable in the result.

2.1 Cell Decomposition

As referred to in (Foley et. al, 1996), a spatial partitioning representation in solid modelling, where solids are decomposed into nonintersecting, typically parameterized primitives, is called cell decomposition.

Serving as the basis for the building reconstruction process, we first of all generate such a partition for each building footprint. As mentioned above, this is done solely from information found in the building's outline. The big challenge herein is to avoid decomposing the area in too many small cells, for which it becomes increasingly difficult to reconstruct a well-shaped roof, especially if the building outline is very detailed and consists of many short line sections (see Figure 4). So instead of using all the available lines from the outline polygon and infinitely extend them to split the footprint, an adequate subset must be found that results in a set of primitives that together reflects well the characteristic shape of the building. However, the resulting outline will not be identical to the original one, but rather be a generalization thereof. So to best resemble the outline, the set of decomposition lines should approximate well the original points and line segments.

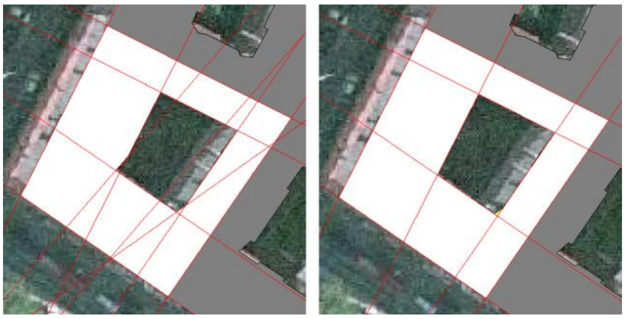

Figure 4. Cell decomposition of a building footprint using all line segments of the outline and only an averaged subset.

Our algorithm for generating cell decompositions from given outlines has been thoroughly described in the context of 3D building generalization (see e.g. (Kada, 2007)). But instead of generating 3D decomposition planes from the facade polygons of a 3D building model, the 2D decomposition lines are now generated from the 2D outline.

In a nutshell, the line segments are grouped into subsets of "parallel" lines that are pair wise a maximum distance away from each other. This is the generalization distance, which means in this context, that the cells resulting from the footprint partitioning will not have sides that are shorter than this length. Line segments are considered parallel if the angle between their directions is below an angle threshold. This allows for a better generalization of connected line segments and therefore helps to keep the number of generated cells low. For each subset of line segments, the associated decomposition line is computed by averaging the line equations of its elements. Short line segments of arbitrary direction, but whose endpoints are both closer to the decomposition line than the parallel line segments, are associated with this subset, but will not contribute to the averaging of this or any other decomposition line.

For example, the green line segments on the left side of Figure 5 are considered parallel under the chosen angle threshold of 15 degrees. The added perpendicular distance of any two endpoints to the red decomposition line, which is the average of the green line segments, is below the generalization distance. While the connecting orange line segment is not parallel to any green line segments, its endpoints also falls under the distance threshold and therefore does not contribute to any decomposition line.

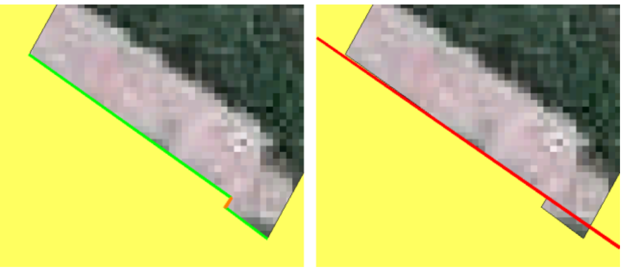

Figure 5. Parallel line segments (green) form decomposition lines (red), rendering short segments in between (orange) unnecessary.

Under the general assumption that ridge and eaves lines should strictly run horizontally, many roof shapes require the ground shape of cells to be trapezoids or rhomboids. Otherwise not all roof faces will be planar and must be split into triangles to form valid solids. Figure 6 shows an extreme example of a cell with a Berliner roof shape where none of the four sides of the ground shape are parallel. The middle face of the roof must be split into two triangles, which is generally not acceptable and should be avoided if possible. Due to the averaging process, the set of resulting decomposition lines are not guaranteed to be parallel. We therefore adjust the decomposition lines slightly so that parallelism and rectangularity are enforced for pairs of decomposition lines with small directional deviations. The same Berliner roof shape of Figure 6 with a trapezoid ground shape results in a valid solid after adjustment.

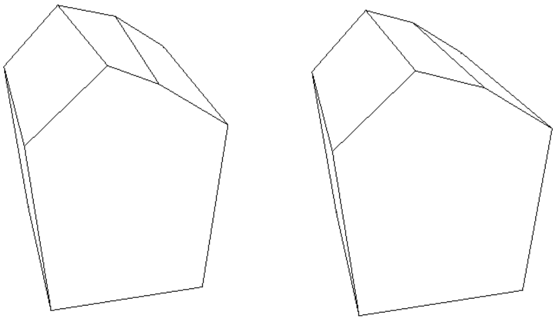

Figure 6. Extreme example of a Berliner roof primitive with a non-parallel before and a trapezoid roof shape after adjustment.

Once the decomposition lines have been generated, a rectangle approximately two times the minimal bounding rectangle is taken and split by these lines, forming nonintersecting cells in the process. Then the cells are compared with the original footprint, and the ones with a low overlap value are discarded. Large cells assure that this classification fails only in few cases.

Figure 7 shows an example cell decomposition of a given footprint. Cells with a low overlap with the original footprint were discarded in the process. The four "horizontal" lines are pair wise parallel, whereas the five "vertical" lines are all parallel, resulting in mostly rhomboid-shaped cells. Although the dotted cells are shaped as trapezoids, most roof shapes fitting between two opposite neighbour cells are valid under these conditions.

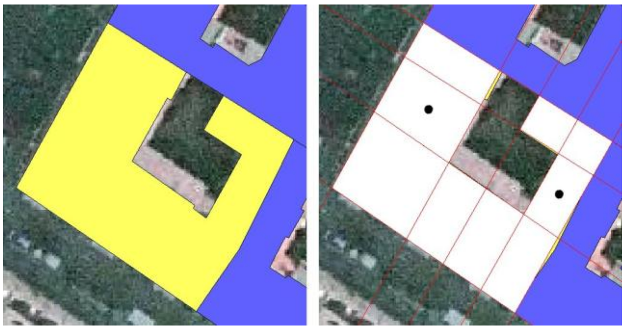

Figure 7. Cell decomposition of a given footprint into rhomboids and trapezoids, the latter marked with dots.

2.2 Roof Shape Determination

Now that a cell decomposition of the footprint is available, the parameterized roof shapes of all cells need to be found. We do this by examining the normal vectors of all points inside the same cell. As point normal vectors are usually not given in surface models, they first have to be generated. If the surface model is structured as a grid, we compute the normal vector of each point from the eight triangles fanned around it and average their normal vectors. However, if the raw data is available in form of an unstructured point cloud, we estimate a point's local plane of regression from its five nearest neighbours and take the resulting surface normal vector.

For the construction of the building's roof, we classify the roof shapes that we use in our approach into three types: basic, connecting and manual shapes. Whereas the shapes of the first two classes can be determined in an automatic process, the last class of roof shapes is only available for manual editing. Among the basic roof shapes are flat, shed, gabled, hipped and Berliner roof (see Figure 8).

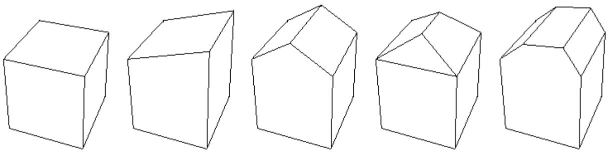

Figure 8. Flat, shed, gabled, hipped and Berliner roof shape.

As not all houses have only one section, there is a need to connect the roofs of the sections with specific junction shapes. Figure 9 shows a small selection of connecting roof shapes.

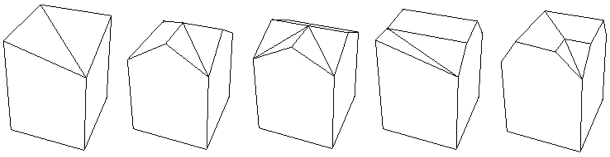

Figure 9. Examples of connecting roof shapes.

In summary, we determine a cell's roof type by comparing the points' normal vectors with the roof faces of all possible shapes and compute the percentage of points that fit the direction of the roof part they are inside. For a gabled roof, e.g., we divide the cell into two equal parts, distribute the points accordingly and count the number of points whose normal vectors are in accordance with the respective side (see Figure 10). Each roof type defines one or more parts, whose size may or may not be dependent on the roof parameters. E.g., the ridge line length of a hipped roof is variable and therefore affects the size of the four roof parts. The longer the ridge line grows, the smaller the two side hips become. This affects how accurately the shape can be determined.

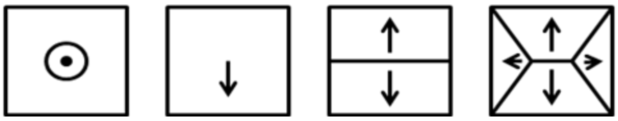

Figure 10. The face normal directions of the four basic roof shapes: flat, shed, gabled and hipped. The flat roof face shows upwards.

2.2.1 Flat, Shed and Gabled Roof: When considering all junction elements, these basic shapes make up over twenty different shapes. The high number comes from the fact, that non-symmetric shapes can be rotated four times, resulting each time in a new shape. Only rotational symmetric shapes result in one shape and axial symmetric shapes in two shapes.

To efficiently determine if the points fit any of these basic roof types, or a connecting shape thereof, each cells footprint is broken into eight sections. For each section, the points are classified as pointing up, north, east, south and west depending on the cells orientation, where the first side of a cell is considered the south side. For a point to be classified as up, the angle between the point's normal direction and the upward vector must be below 30 degrees. For the other four classes, the 2D component of the point's normal vector must point more towards that side than to the other three, which reflects an angle below 45 degrees. Once all the points are classified, the percentage of matching points can be simply added up for all shapes.

Figure 11 shows four types of gabled roofs. For these classes of roof shapes, also the corner elements are used as they are basically free to compute. The basic gabled shape is axial symmetric and therefore only has two variants, the corner- and T-junctions can be rotated four times and therefore result in four variants each and the cross-junction is axial symmetric and therefore has one variant. The number of matching points for the gabled roof can be easily computed by adding the number of points in the green sections that show northwards and the number of points in the red sections that show southwards. The other shapes are computed accordingly, where the points in the blue sections must show westwards and the points in the yellow sections eastwards.

Once the points have been distributed to the eight sections and classified according to their normal direction, the time to do the summation is neglectable. This makes roof shapes whose shape can be reduced to the eight sections very appealing.

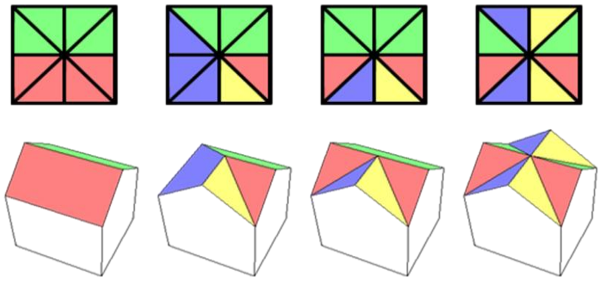

Figure 11. Gabled roof and its corner-, T- and cross-junctions and the direction points inside a particular face must show to.

2.2.2 Hipped Roof: For hipped (and other roof shapes that cannot be as easily divided into the eight sections as the aforementioned shapes), the roof area is divided individually. This is, however, not as efficient as before and some assumptions have to be made for some shapes. E.g. the ridge length of a hipped roof should be variable, but we assume that all four slopes are the same, which enforces a certain ridge length. This way only one variant must be evaluated, but it still reliably differentiates a hipped from e.g. a tent or gabled roof.

2.2.3 Berliner Roof: The Berliner roof is an asymmetric roof shape, which is basically a shed roof disinclined slightly to the back side. By having a steep slant at the front and sometimes also at the back side, the roof appears to be gabled from a pedestrians point of view. This shape is very common for Berlin apartment houses build during the period of promoterism in the 19th century.

To identify the front side of a cell with a possible Berliner roof, we seek the side closest to the building's oriented bounding rectangle. If the cell is a corner cell, or if all cells are side by side, then two or more sides of the cell should be within closest distance to the bounding rectangle. Here, the side with the highest number of normal vectors pointing towards to is determined. This is in most cases the back side. Both methods are necessary, as the second one generally fails more often, but is the only one that works for the latter case.

Then, the distances from the front and back side to the two fake ridge lines are determined using a plane sweep approach. At the front ridge line, the 2D components of the points' normal vectors show in opposite directions. As for the back ridge line, we say that all points' normal vectors with an angle below 30 degree compared to the upward vector belong to the shed part of the roof. Using these two criteria, we can accurately determine the two ridge lines that separate the three roof regions. Their height is computed from the plane equations estimated from the points of the two steep slant sections.

2.3 Parameter Estimation

Roof parameters vary from shape to shape. However, all shapes have one eaves height and up to two ridge heights, which are to be estimated from the LIDAR points. Among others, the cell's footprint defines the directions of the eaves and ridge lines. As all face slopes are linearly related, it allows determining them at once by simply estimating one plane equation from the given points. While one face defines a reference system, the points in other faces are translated into it accordingly. From the resulting plane equation, the eaves and ridge heights can be determined from the reference face. The resulting shape parameters best fits all the faces to the input points.

2.4 Roof Junctions:

Cells that have neighbor cells at two consecutive sides or at three or more sides are examined again. These cells are candidates for connecting shapes. Based on the shape types, the parameters and the arrangement of the neighbor cells, compatible connecting shapes are determined. The one that connects the most neighbor cells to a sound roof structure is then chosen and its parameters determined from the parameters of the neighbor cells.

2.5 Manuel Editing

Because not all roof structures can be fully automatically reconstructed, there is a need for manual editing. In our editing tool, the decomposition lines can be copied, added, deleted, translated and rotated. The cells' roof shapes are automatically reconstructed after every manual step, so that the operator can immediately see the results. Once the cell decomposition fits the roof's shape, the cell parameters can be manually adjusted or even copied from other cells. If the decomposition produces too many small cells, then their number can be decreased by a merging operation.

Even though editing the building models using decomposition lines is not so straight-forward, we found that operators got used to it very quickly and can efficiently produce even landmarks with complex geometry. The manual mode also allows for more complex roof shapes like mansard, cupola, barrel and even some detail elements like dormers.

3. PROJECTS

While still in development, we started using the reconstruction software in a real production environment. Several large-area projects have since been successfully completed. The feedback in the early stages of development helped us to recognize and adapt to arising problems. Two of our early projects were the 3D reconstruction of East Berlin and Cologne, two major cities in Germany. The 3D city model of Berlin is also available online for use in Google Earth (Berlin 3D, 2009).

3.1 East Berlin, Germany

The first project with our new software was to perform a 3D building reconstruction from Berlin's LIDAR data. The total area of the project was 498 km² with approximately 244,000 buildings. The project was an extension of the original 3D City model of Berlin, Germany, which is to date still the largest city model transported to the Google Earth platform. Input data included a DTM, airborne LIDAR and building footprints. See Figure 12 and Figure 13 for the resulting model.

Due to the large number of buildings in East Berlin and project time constraints, photogrammetric extraction was immediately deemed as being too time consuming and costly. It was therefore decided to use LIDAR data instead. All LOD 2 building models are geo-referenced geometry, which were later textured using aerial oblique imagery.

The Berliner Roof– a particularly unusual roof type typically found on many buildings in Berlin – presented a challenge as well as numerous inner courtyards presented problems during extraction. Therefore, the reconstruction approach had to be adapted to automatically detect this unique roof structure. As a result, a total of 17 individual roof types have been additionally integrated into the software to enable greater accuracy during reconstruction and to reduce the amount of manual editing needed.

As the software was constantly improved during the duration of the project, the amount of manual editing needed for the reconstructed 3D buildings was reduced from 30 percent in denser areas to 20 percent; manual editing for the outer lying areas also experienced a sharp improvement: from 20 percent to 15 percent.

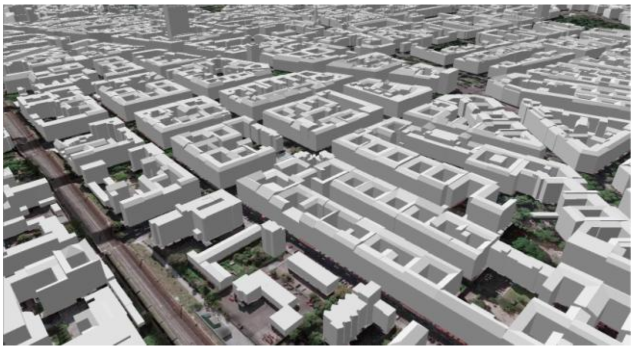

Figure 12. 3D city model of East Berlin.

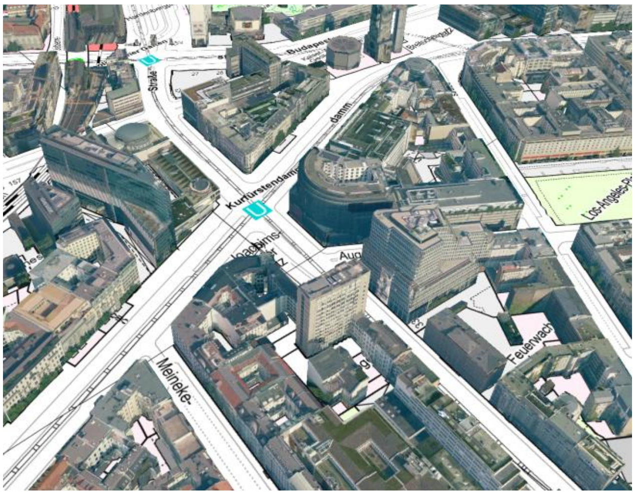

Figure 13. 3D city model of Berlin textured from oblique images showing part of the prominent Kurfürstendamm.

3.2 Cologne, Germany

The existing 3D city model of Cologne is used by several administrative departments as a complement to the existing GIS data inventory held by the Cologne Survey Department. It was created from the basis of building storeys using two-dimensional footprints; therefore the true heights of the buildings were not accurate. In order to produce a more realistic representation of Cologne in 3D to be used for urban planning and emergency response, the survey department decided to use the data from the most recent LIDAR flyover to perform a real 3D building reconstruction. See Figure 14 and Figure 15 for the results.

Cologne's city boundaries encompass approximately 415 km² with 280,000 buildings; therefore it was decided to use airborne

LIDAR instead of photogrammetry. In many areas of the inner city, Cologne has an extreme building density, which complicated a clean separation of building geometry and roof forms, even though the building outlines contained in the ground cadastre map were examined beforehand for their accuracy.

In addition, there were many special building structures such as churches that had to be extracted from the airborne LiDAR data. Pre-processing efforts were further complicated by the fact that Cologne's ground plan data was outdated or incomplete as several new buildings that had been erected and still others had been torn down since the last update made to the ground cadastre map.

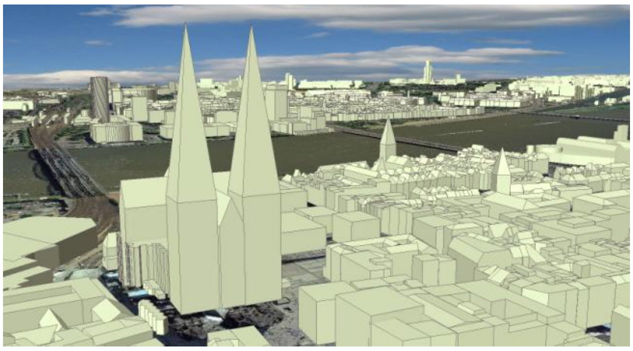

Figure 14. 3D city model of Cologne.

After a careful study of the digital ground map it was determined that first several adjustments had to be made, for example removing underground buildings and structures such as parking garages and identify torn down buildings. This required examining the discrepancies between the DTM, DSM and building outlines to create the final 3D city model.

Finally, many larger buildings appeared in several different attribute tables containing sometimes conflicting information, therefore presented a challenge for both the client as well as the operators because these buildings still needed to be reconstructed without altering their original building footprints.

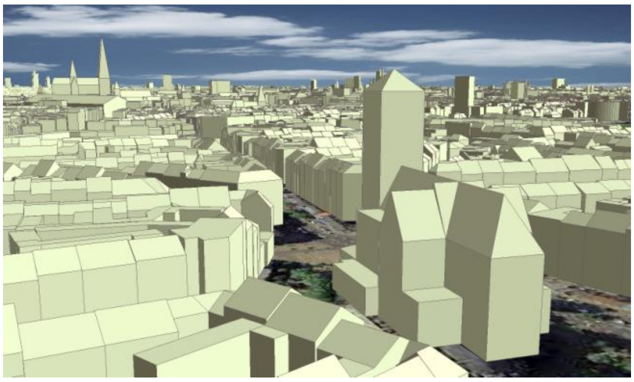

Figure 15. 3D city model of Cologne.

We have completed a wide-area 3D city model in LOD2 for the whole of Cologne by the end of September 2008. This new model will be integrated into the existing model, thereby replacing the GIS data with a much more accurate representation of the building heights. Because the inner city buildings will receive realistic façade textures, highly accurate building heights and roof structures as well as building details were a key project requirement.

The entire model will be used a decision making tool for urban planning and serves as a visualisation tool and complement to Cologne's Master Plan. The amount of overall manual post editing required with the software has been reduced since working on the East Berlin model to 15 percent.

4. CONCLUSION AND FUTURE WORK

We have presented an approach for the automatic reconstruction of 3D building models from LIDAR data and existing ground plans. It is based on an algorithm to decompose given footprints into sets of nonintersecting cells, for which roof shapes are then determined from the normal directions of the LIDAR points. The validity of this approach has been proven effective, as can be judged by the 3D city models of East Berlin and Cologne.

The next step is to increase the amount of detail by loosening some of the restrictions of our shapes and by making them more flexible. This is already possible in manual editing. However, to increase both the richness in detail and the automation, we plan to integrate a segmentation of the roof points to selectively decompose the footprints without generating more cells.

5. REFERENCES

Arefi, H., Engels, J., Hahn, M. and Mayer, H., 2008. Levels of Detail in 3D Building Reconstruction from LIDAR Data. In: *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXVII-B3b*.

Berlin 3D, 2009. 3D-Stadtmodell Berlin. http://www.3d-stadtmodell-berlin.de (accessed 6 April 2009)

Brenner. C., 2005. Building Reconstruction from Images and Laser Scanning. In: *International Journal of Applied Earth Observation and Geoinformation (Theme Issue 'Data Quality in Earth Observation Techniques), Vol. 6 (3-4)*, p. 187-198.

Kada, M. 2007. Scale-Dependent Simplification of 3D Building Models Based on Cell Decomposition and Primitive Instancing. In: Spatial Information Theory: 8th International Conference, COSIT 2007, pp. 222-237.

Kolbe, T.H., 2009. Representing and Exchanging 3D City Models with CityGML. In: *Lee, Zlatanova (Eds.): 3D Geo-Information Sciences*, Springer-Verlag Berlin Heidelberg, pp. 15-32.

Möser, S., Wahl, R. and Klein, R., 2009. Out-Of-Core Topologically Constrained Simplification for City Modeling from Digital Surface Models. In: *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXVIII-5/W1*.

Sohn, G., Huang, X. and Tao, V., 2008. Using a Binary Space Partitioning Tree for Reconstructing Polyhedral Building Models from Airborne Lidar Data. In: *Photogrammetric Engineering & Remote Sensing, Vol. 74 No. 11*, pp. 1425-1438.