RECONSTRUCTING BUILDINGS WITH DISCONTINUITIES AND ROOF OVERHANGS FROM OBLIQUE AERIAL IMAGERY

D. Frommholz*, M. Linkiewicz, H. Meissner, D. Dahlke

DLR Institute of Optical Sensor Systems, Berlin, Germany -
(dirk.frommholz, magdalena.linkiewicz, henry.meissner, dennis.dahlke)@dlr.de

Commission II, WG II/4

KEY WORDS: Aerial, Oblique, Building Reconstruction, Discontinuities, Roof Overhangs, Texture Mapping, CityGML

ABSTRACT:

This paper proposes a two-stage method for the reconstruction of city buildings with discontinuities and roof overhangs from oriented nadir and oblique aerial images. To model the structures the input data is transformed into a dense point cloud, segmented and filtered with a modified marching cubes algorithm to reduce the positional noise. Assuming a monolithic building the remaining vertices are initially projected onto a 2D grid and passed to RANSAC-based regression and topology analysis to geometrically determine finite wall, ground and roof planes. If this should fail due to the presence of discontinuities the regression will be repeated on a 3D level by traversing voxels within the regularly subdivided bounding box of the building point set. For each cube a planar piece of the current surface is approximated and expanded. The resulting segments get mutually intersected yielding both topological and geometrical nodes and edges. These entities will be eliminated if their distance-based affiliation to the defining point sets is violated leaving a consistent building hull including its structural breaks. To add the roof overhangs the computed polygonal meshes are projected onto the digital surface model derived from the point cloud. Their shapes are offset equally along the edge normals with subpixel accuracy by detecting the zero-crossings of the second-order directional derivative in the gradient direction of the height bitmap and translated back into world space to become a component of the building. As soon as the reconstructed objects are finished the aerial images are further used to generate a compact texture atlas for visualization purposes. An optimized atlas bitmap is generated that allows perspective-correct multi-source texture mapping without prior rectification involving a partially parallel placement algorithm. Moreover, the texture atlases undergo object-based image analysis (OBIA) to detect window areas which get reintegrated into the building models. To evaluate the performance of the proposed method a proof-of-concept test on sample structures obtained from real-world data of Heligoland/Germany has been conducted. It revealed good reconstruction accuracy in comparison to the cadastral map, a speed-up in texture atlas optimization and visually attractive render results.

1. INTRODUCTION

Oblique aerial cameras nowadays have become a standard tool for remote sensing tasks in urban areas. In contrast to solely nadir-looking sensors these devices feature one or more camera modules in different configurations providing tilted (off-nadir) views of the scene (Remondino and Gehrke, 2015). This allows to record very high-resolution images of the façades of buildings because the incidence angles between the merely orthogonal surfaces and the viewing vectors converge. Therefore oblique aerial cameras are in particular qualified for texturing virtual city models. For example, (Smith et al., 2009) demonstrate the projection of unrectified nadir and oblique images from different sensors onto triangular meshes. While this work already addresses visibility issues in urban quadrangles and due to shadow occlusion handling is not mentioned yet. In (Stilla et al., 2009) infrared aerial images that have been rectified in an offline preprocessing step are used to decorate an existing 3D model. A z-buffer approach in conjunction with backface culling and polygon clipping is used to deal with hidden objects and to remove invisible faces. Similarly, (Frueh et al., 2004) project oblique aerial bitmaps onto triangular meshes of city buildings generated from terrestrial laser scanner data. The source image candidates are selected per face by evaluating the viewing angle, spatial coherence and degree of occlusion which again is detected by a z-buffer algorithm omitting objects not contained in the mesh itself. The resulting texture patches are compactly arranged in a texture atlas, however, multiple images from different views do not get combined yet in order to possibly color hidden object parts.

Aside from texture mapping oblique aerial cameras have also been successfully used for the generation of 3D building models. This typically involves a preprocessing step in which the oriented images are photogrammetrically transformed into a point cloud or digital surface model (DSM). In (Haala et al., 2015) an optimized dense semi-global stereo matcher is described that accepts off-nadir views and processes them into watertight meshes using a triangulator based on restricted quadtrees. However, while the detailed output might be suitable for visualization and will preserve building discontinuities and roof overhangs the resulting number of polygons may become prohibitive for standard 3D data formats used by geographical databases. A high polygon count does not arise in (Panday and Gerke, 2011) where parametric building models are fitted to oblique images. For this purpose roof wire frame edges of a rough mesh of a structure are matched to the off-nadir input bitmaps, and roof overhangs already get detected by plane sweeping. However, since the required initial building hull is derived from an airborne laser scanner instead of the imagery the method described is not self-contained. In contrast (Sun et al., 2016) generate prismatic building reconstructions from a normalized DSM which gets segmented and merged allowing a variety of point sources including radar. Although discontinuities are supported by this method roof overhangs and

*Corresponding author

This contribution has been peer-reviewed.
doi:10.5194/isprs-archives-XLII-1-W1-465-2017

gabled roofs are not. The non-parametric approach mentioned in (Xiao, 2012) utilizes point clouds obtained from oblique aerial imagery to identify wall surface and roof patches with a trained classifier, however, roof overhangs silently get merged with the housetops as indicated by the results.

This paper discusses an automated approach for the extraction of buildings of western-style architecture with discontinuities and roof overhangs solely from oriented nadir and oblique images captured by a decent aerial camera system. The resulting 3D models made of simple polygons with known surface types get textured with patches directly taken from the original bitmaps to avoid resampling whenever possible and retain the maximum resolution. During texture mapping occluded parts of the meshes are detected by ray-casting a full-size DSM of the scene and colored from different radiometrically adjusted sources if available. Further, the resulting texture atlas is optimized using a partially parallel algorithm for the inherently sequential placement problem. The final models are enabled for perspective-correct visualization by renderers that implement a pre-scaler in form of a shader program running on the graphics processing unit (GPU) and stored as both semantically annotated Alias/Wavefront OBJ files and CityGML objects for GIS applications.

2. DATA PREPROCESSING AND NOISE REDUCTION

As a first step of the proposed workflow the raw aerial images are downloaded from the storage of the oblique camera system and preprocessed. This comprises demosaicking, dark signal and photo response non-uniformity correction and distortion removal. Photogrammetric bundle adjustment refines the initial sensor position and rotation provided by the global navigation satellite system (GNSS) receiver and inertial measurement unit (IMU) of the camera yielding oriented pinhole images with a maximum epipolar line error of about 0.1 to 0.3 pixels. To regain the 3D information of the captured scene the bitmaps subsequently are passed to a dense semi-global matcher (SGM) which outputs perspective digital surface models for each base image (Hirschmüller, 2008). Because the available SGM implementation is optimized for nadir images the orientation of the oblique shots must be altered by the average angular displacement to roughly point into a vertical direction before running the matcher and back-rotated later on to retain the original geometry of the façades.

When stereo matching is finished the DSMs get selectively blurred to reduce the noise without affecting high-frequency details utilizing a multi-threaded bilateral filter with two empirically parametrized Gaussians for the Euclidean pixel distance and radiometric difference. This will merely accommodate matching inaccuracies that for instance are caused by periodic patterns in the scene or sensor noise. However, the bilateral filter will not affect any positional noise that is introduced when the DSMs are triangulated and to which in particular the oblique views are prone because of their special characteristics (Rupnik et al., 2014).

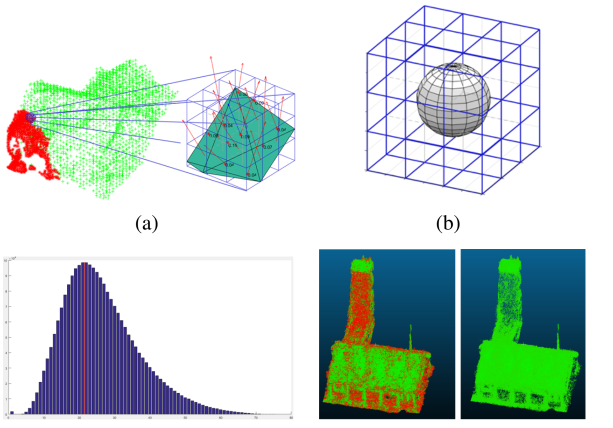
Therefore, after obtaining the 3D point clouds from the DSMs, another voxel-based filter must be executed to further reduce the noise level to be able to also model small building discontinuities later on. To reduce its runtime the filter will be applied to the buildings only which get separated from the terrain by shifting windows of varying size over a unified ortho-projected DSM of the entire scene beforehand as mentioned in (Mayer, 2004). If run together with an orthogonally projected near-infrared channel this process that basically creates a digital terrain model (DTM) will be enabled to eliminate most of the vegetation. As a by-product the generated DTM also provides an estimate for the ground height around the buildings.

Once the buildings have been roughly identified the point set for each structure gets spatially sorted by regularly subdividing its axis-aligned bounding box. The length of the edges a of the resulting cubes can be either specified manually if the noise level is known in advance or guessed automatically from a sparse dry-run of the following loop at random. For distortion removal, for each of the voxels the samples inside a $3 \times 3 \times 3$ environment undergo a fast principal component analysis (PCA) to locally estimate the object surface. In order to decide if the current cube is eligible for noise elimination it must be tested efficiently whether it intersects with the approximated planar patch or not. For this purpose a sphere with the radius of half the voxel diagonal $r = \frac{a}{2}\sqrt{3}$ is constructed around its midpoint. When the estimated surface touches the sphere the Euclidean distances of the patch to the points of the cube are calculated and binned to form a histogram. Otherwise the algorithm continues with the next voxel. Since the 26-neighborhoods of adjacent kernels overlap it is guaranteed by the sampling theorem that the building points are continuously scanned inhibiting structural breaks. Using a sphere to determine whether the locally estimated surface is to be denoisied by a particular voxel or not introduces a small chance that the cube incorrectly accepts the planar patch even though it does not intersect with the element at all. However, since speed is crucial for any volume-based algorithm and the plane normal for the spherical collision test is already available from PCA as the Eigenvector with the smallest Eigenvalue of the underlying covariance matrix the accurate but computationally more expensive intersection test against any of the twelve voxel edges is abandoned in favor of the former.

Having the distance histograms for each voxel noise elimination consists in removing those points from the cube whose distances exceed the value of the class with the maximum sample count. This maximum will be unique if the noise is normally distributed and hence its causes, for instance bundle adjustment issues, do not expose a systematic error. In this case the histogram will follow a discrete version of the χ^2 distribution even though the square root has been taken on the involved differences. Because the square root operation however is a monotonous function the basic profile of the histogram will remain intact. Figure 1 illustrates the voxel-based noise removal.

3. BUILDING RECONSTRUCTION

Building reconstruction initially takes the point clouds from the preprocessing stage and projects them onto a 2D grid. The spacing of the grid must be adjusted empirically according to the remaining noise level, however, due to the filter applied it can be much less than the previously used cube size a allowing more details to be reconstructed. The spatial distribution and z histogram of the projected vertices subsequently is analyzed to identify façade pieces. For the grid cell element set that is considered a part of the façade a regression line is estimated by RANSAC from a consensus set of two projected points (Fischler and Bolles, 1981). The estimator will terminate if either an upper iteration count has been reached or if the percentage of the points of the cell that are within a maximum distance to the regression line running through the consensus set is above a threshold that depends on the grid spacing. When the line segments have been found for individual grid cells the regression will be extended to adjacent cell



(a) (b) (c) (d)

Figure 1. Voxel-based noise removal (a) Surface approximation using PCA (b) Neighborhood and sphere around the kernel for the fast cell intersection test (c) Sample point-to-surface distance histogram (d) Sample cloud showing the removed points in red

groups to form façade fragments of the same orientation. This is achieved by pair-wisely comparing the line directions of the cells within the local 8-neighborhood and the line direction of the center element against a fixed angular threshold. In case of matching directions the respective cells are added to the same cell set. When no more adjacent cells are left to be assigned a common regression line will be estimated through the point set of each cell group. The resulting linear façade fragments will be intersected if their endpoints are locally adjacent within the grid forming a closed two-dimensional building contour. Similarly, the rooftops get modeled by sorting the 3D points into a grid keeping the z coordinates this time. Cells outside the polygonal outline from façade detection get discarded, and regression is performed for each grid element obtaining plane segments. Roof cells are identified by applying a smoothness threshold and get merged into the same labeled cell group if their normals roughly point into the same direction.

To derive the geometry of the housetops from the cell groups their topology needs to be computed. Following region growing to close gaps between the labeled cell clusters caused by roof installations the grown cell sets are tested for connections by shifting a 2×2 window over the axis-aligned bounding box of the projected area. If the window covers two or more different labels this will indicate a topological edge or node between adjacent cell groups respectively to be stored in a table. In a comparable fashion the topology between the roof and the building façades is analyzed. For this purpose the polygonal outline of the structure obtained during wall surface reconstruction gets discretely scanned linearly or circularly around the building corners to obtain arrises.

Using the spatial relationship between the building surfaces the world coordinates of the roof elements are derived from the information stored in the topological table which itself does not hold any positional information. The housetop is set up by fitting regression planes to the 3D points confined in the respective roof cell groups, and the façade planes get constructed by extruding the respective segments of the building contour perpendicularly to the xy plane. Having these geometric entities the intersection of adjacent pairs of roof planes or a roof plane with a nearby wall surface yields the geographic position and direction of the topological edges, i.e. the roof ridge, arris and the outer edges of the housetop. Since the resulting possibly skew lines will be infinite they need to be limited by the spatial representation of the topological nodes before they can form the skeleton of the building to be modeled. Therefore the node coordinates are computed as the geometric center of the shortest connecting line among all combinations of pairs of lines derived from the topological edges, and the edge lines get adjusted accordingly. To complete the initial reconstruction the missing ground surface is derived from the building contour, and its z value is set to the height estimate from the DTM mentioned before.

However, even though the described algorithm by design already supports certain segmented façade geometries it will fail on buildings with larger discontinuities in particular when the roof is incorporated. In this case inconsistencies between the topological entities and the intersection calculations occur and a restart of the modeling process is triggered for the affected object entirely in 3D which trades speed for correctness. Like for noise reduction the algorithm starts by regularly subdividing the bounding box of the point cloud of the respective building. Since the distortions have been already removed the voxel edges can be set to approximately half the size previously used for the filter which allows to capture equivalent discontinuities. For each cube the local surface as sampled by the contained points is approximated with PCA. Plane segments in adjacent voxels whose normal vectors roughly point into the same direction are used to estimate a common spatially infinite building surface. In order to limit the extent of the surfaces efficiently their spatial relationship would be beneficial. However, neither region growing nor the sampling technique from initial reconstruction is applicable to 3D point sets to gain information on how the potentially adjacent areas are meant to be joined. Therefore, all combinations of triplets of infinite building surfaces get intersected. This results in a considerable set of vertices which can be condensed by dropping those points that are outside the bounding box of the building and those that exceed a user-defined distance to the closest point of the grid cells the three defining planes come from. The remaining ”good” vertices are analyzed for identical positions and consolidated yielding both world coordinates and topological k-nodes. Here k is the number of coinciding surfaces, and having values for k less than the real intersection count at a particular building corner inevitably leads to more or less severe reconstruction errors. To minimize the error rate the input data should comprise of dense point clouds as they can be derived from high-resolution oblique images.

Once the set of vertices is final the topological edges connecting two points remain to be determined to obtain the building hull including any modeled discontinuities. This however is trivial since the necessary information will be an intermediate result when the infinite surface triplets get mutually intersected. Figure 2 illustrates the volumetric reconstruction process.

4. ROOF OVERHANGS

Neither the initial nor the volume-based algorithm preserve the roof overhangs of buildings as in both versions they get cut off by the façade planes. However, since eaves are very common in western architecture and because not modeling them induces errors in particular when the models are to be textured the overhangs are calculated in an additional step. For this purpose the polygonal outline as defined by the wall surfaces of each building is projected onto the orthogonal DSM of the entire urban scene.

The resulting 2D shapes will be offset equally along the edge normal vectors with subpixel accuracy exploiting the zero-crossings of the second-order directional derivative in the gradient direction of the height map (see figure 3). The reconstructed roof polygons of the respective building model then get extended accordingly.

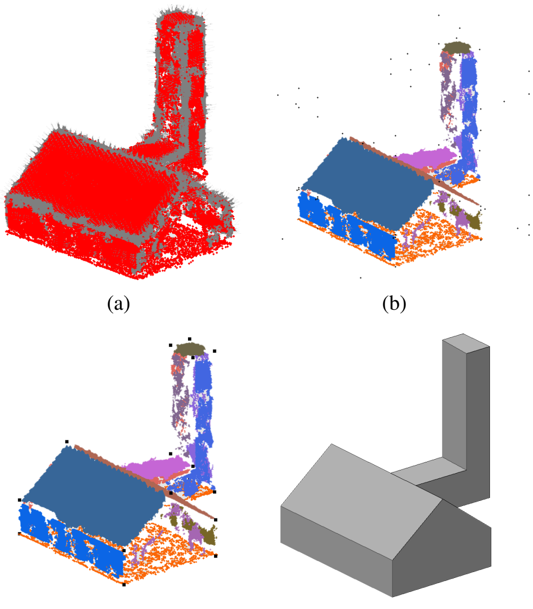

(a) (b)
(c) (d)

Figure 2. Voxel-based building reconstruction (a) Grid cell planes (b) Vertices from intersecting all planes (c) Condensed vertices (d) Polygonal building hull with modeled discontinuities

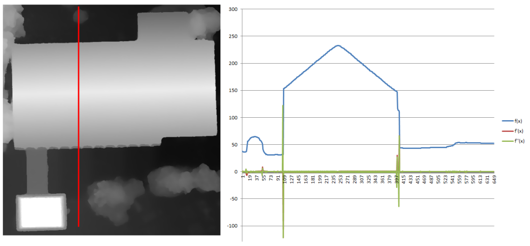


Figure 3. Roof overhang estimation from the ortho-DSM

5. TEXTURE MAPPING

For realistic visualization the polygons of the reconstructed buildings remain to be decorated. In a process called texture mapping the original oriented aerial images are projected onto the model surfaces since they congruently provide the necessary color information. Texture mapping assigns each point of a building polygon a normalized 2D position inside a bitmap pointing to the pixel which effectively provides the spectral intensities at that vertex. Any pixel of the interior zone of the face will then be automatically interpolated by the 3D render engine. Computing the texture map involves finding the set of aerial images that picture a particular polygon best. Occlusions need to be bypassed using multiple source images if available, and radiometric differences on a model polygon must be corrected. Also, those parts of the chosen images that actually contribute to a face have to be compactly arranged and efficiently rendered to a single image called texture atlas. Creating a texture atlas is essential because render engines cannot efficiently handle hundreds or thousands of high-resolution aerial bitmaps at once due to limitations on memory, I/O and hardware mappers on the graphics card.

To find the set of images that optimally colors a particular part of the building model the whole set of oriented photographs is analyzed per polygon. A unified resolution-based measure determines the priority of both the nadir and oblique color sources. This value is computed by initially projecting the vertex of the building polygon with the greatest distance to the current camera into the corresponding image. The four corners of the pixel that gets hit on the virtual sensor are ray-cast, and as a result the intersection points with the polygon plane will span a convex quadrilateral in world space. The quality index is the square root of the area of this quadrilateral, that is, the side length of a coextensive square at the same place, or, the worst-case ground sample distance (GSD) on the surface in terms of photogrammetry.

When the list of source bitmaps has been composed for a reconstructed surface and put in ascending order with respect to the GSD it will be passed to the visibility test to detect occlusions. This works by mapping the building polygon into the camera of the current image in world coordinates and discretely ray-tracing each pixel inside the ortho-DSM of the scene. Collisions with the DSM are kept in a run-length encoded (RLE) binary bitmap per texture patch per source image to save space. Particularly in highly overlapping oblique flight configurations the visibility data gets also involved in removing texture sources that do not provide any additional information. Thus, a graph for each pair of texture patch candidates $(T_i, T_j), i \neq j$ that picture a building surface is constructed from the binary bitmaps. This graph keeps the total count of visible pixels for each T_i in its nodes and stores the non-commutative count of overlapping pixels in its edges. Both values describe the overlap ratio between T_i and T_j, and texture sources violating a user-defined percentage get discarded. Moreover, the graph is analyzed to derive the coefficients for radiometric adjustments between overlapping sources texturing the same building surface. This will produce homogeneous render results when occluded parts of a polygon are padded. To obtain the coefficients of the chosen linear correction model from (Fisher, 1992) the minimum spanning trees and hence disjoint sets of overlapping texture sources are computed among which the radiometry can be adapted (Kruskal, 1956). The resulting factors are added to the graph nodes to be applied when the texture atlas is being finalized.

Once the building faces have been assigned patches of one or more aerial images the actual color information is ready to be written to the texture atlas in two steps. During the placement stage a compact layout is obtained first, however, globally optimizing the arrangement of polygonal pixel sets inside a rectangular bitmap is known to be NP-hard. Therefore, as a local optimization running in polynomial time, new texture patches get placed inside the atlas one by one. To find the next free spot for the current patch P_i the vacant positions inside the destination bitmap are sequentially traversed. Once a free position has been found it will be attempted to paste P_i in there allowing lossless rotations in steps of 90° to utilize the available space as efficiently as possible. If there is no overlap with any existing patch the atlas position and rotation angle for P_i will be stored in a dynamic list and the algorithm will proceed with the next patch. However, in case P_i cannot be pasted anywhere the destination bitmap will get resized horizontally and vertically to the next power of two of pixels as demanded by most graphics adapters.

Since a naive implementation of this brute-force approach is prohibitive because of its computation time the exhaustive search for free spots will be accelerated by a block availability map (BAM). The BAM keeps track of unoccupied scanline segments inside the texture atlas. Also, a coarse 2D grid that effectively sorts the existing texture patches spatially limits the amount of previously arranged pixel sets a P_i has to be tested against for overlaps on average. Moreover, although a particular patch can only be placed after its predecessors without risking a collision and hence an inherent data dependency will prevent the use of multi-threading at first sight, the exhaustive search nevertheless can be at least partially parallelized. This is achieved by running the overlap test, i.e. rotating the current P_i and testing it for a collision with any patch stored by the intersected 2D grid cells, concurrently for each BAM range (see figure 4). The final texture patch placement comprises of the first position inside the current scanline segment, and the other threads bail out as soon as the arrangement with the minimum horizontal coordinate within the range has been found. Thus, the multi-threaded search inside the BAM blocks will pay out primarily when the texture atlas fills up more and more. It will not have a significant effect on the first few placements which however can be expected to run fast anyway.

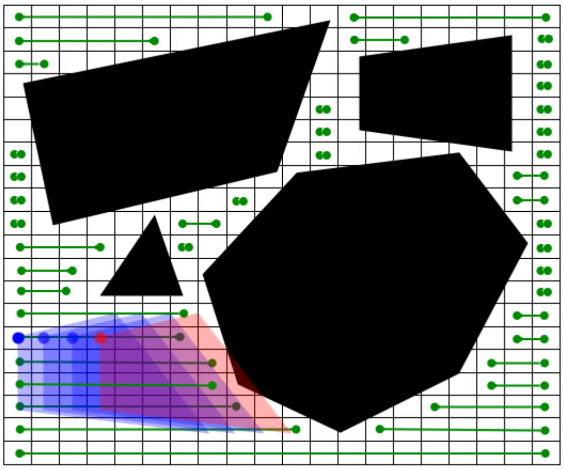


Figure 4. Sample four-threaded texture patch placement on BAM segments indicating free atlas spots (green). The leftmost blue position will be picked, and the red position collides with a previously placed polygon (black).

In the render stage the color information from the respective aerial source images gets actually written to the designated places inside the atlas bitmap. In case of the root image as of the overlap tree contributing most of the pixels to a building polygon no resampling is involved. However, if occlusions must be bridged subordinate images need to be warped to the shape of the root image. Also, the radiometric adjustment coefficients are applied now recursively, and untextured areas inside the atlas get interpolated by inverse distance weighting.

To correctly render the colored building models in a 3D viewer the 2D coordinates of the vertices inside the texture atlas are required to be normalized by the atlas dimensions. Moreover, because the texture patches have not been ortho-rectified and the perspective correction hard-coded in graphics processors applies to the viewer's position only but not the world position of the root image distortions may be introduced during visualization (Low, 2002). The solution to this problem is to pre-scale the texture pixels in the moment they get displayed. This is done by projecting the vertices v_i of each building face orthogonally onto the sensor plane of the respective root camera running through its center of projection C_w, i.e. computing

$$d_i = |n\left((v_i - C_w) \cdot n\right)| \tag{1}$$

where d_i is the distance between the vertex and its projection in world space and n denotes the camera normal. During visualization the value of d_i must be multiplied with the texture coordinates assigned to the respective vertex, and the product undergoes the normal view-dependent perspective-correct interpolation by the raster unit. Before the color samples can be correctly retrieved from the texture atlas and displayed the interpolated product is divided by the now interpolated value of d_i effectively performing online rectification. Since the distances d_i cannot by calculated online by the GPU because it lacks the camera orientation the values are added to the normalized texture coordinates forming texture coordinate triplets.

6. WINDOW CLASSIFICATION

In a last optional step the generated atlas bitmap can be passed to a commercial object-based image analysis (OBIA) tool executing a custom rule set to identify windows on the contained unrectified façade patches as described in (Frommholz et al., 2016). Following segmentation and classification based on the brightness and contrast potential window objects are tested against geometric constraints, conditionally grown, fused and morphologically filtered. The output polygons get vectorized and reintegrated into the previously reconstructed buildings by sparsely ray-tracing their vertices. The final model comprising 3D point coordinates, indexed polygon descriptions and texture information is stored in the open Alias/Wavefront OBJ format for visualization. Also, the building models are written as semantically annotated CityGML objects for GIS applications. The required surface types are implicitly deduced from the plane normals during reconstruction.

7. RESULTS

In a proof-of-concept the proposed method has been evaluated on images of the island of Heligoland/Germany taken by the DLR MACS HALE aerial camera (Brauchle et al., 2015). This device comes with one nadir-looking and two oblique RGB sensors (4844 x 3212 pixels, 12 bits per component) pointing to the left and right at an angle of 36° to the vertical axis. An on-board GNSS receiver and IMU is available to assign each image the approximate camera position and rotation. Preprocessing transformed the 3478 input images into point clouds, DSMs and a DTM at a resolution of 5 cm per pixel.

From this data four buildings with discontinuities and roof overhangs have been selected to be processed. Following noise removal which reduced the initial point cloud size per object from roughly 200 megabytes to 30 megabytes reconstruction with a MATLAB prototype implementing the the three-dimensional regression took about 90 seconds for each structure. As mentioned the software is not fully automatic yet but requires some data-dependent input parameter to be estimated in advance.

To assess the accuracy of the volumetric reconstruction the building outlines have been compared with the cadastral map (Automatisierte Liegenschaftskarte, ALK) of Heligoland. For this purpose the models excluding the roof overhangs were ortho-projected, and the corner points got manually assigned to their ALK counterparts (see figure 5). As a result a median positional deviation of 0.29 m and a mean positional deviation of 0.36 m was found. This is approximately half the distance obtained without noise removal which, using the same processing parameters, initially yielded a median deviation of 0.53 m. The roofs including their overhangs were visually compared to an ortho-DSM obtained from the nadir images only since no ground measurements were available and the operating experience on the photogrammetric workflow for the vertical views is likely to be less error-prone than the all-embracing oblique process. Except for point noise no significant discrepancies have been found, and the overhang estimation always was within twice the DSM resolution, i.e. 10 cm.

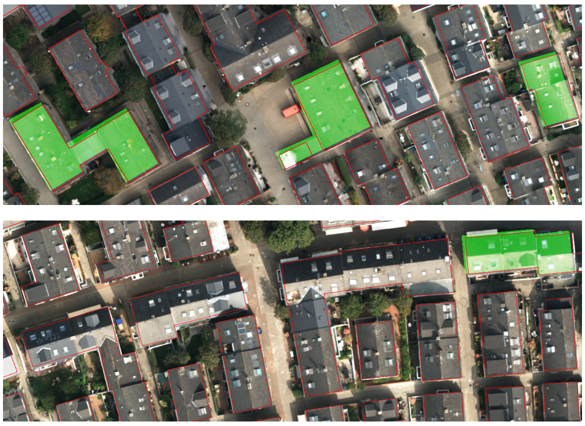


Figure 5. Comparison of the building outlines without roof overhangs (transparent green) and the cadastral map (red)

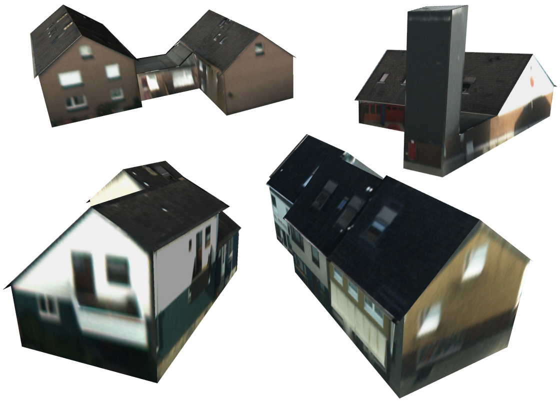


Figure 6. Textured buildings with discontinuities, roof overhangs

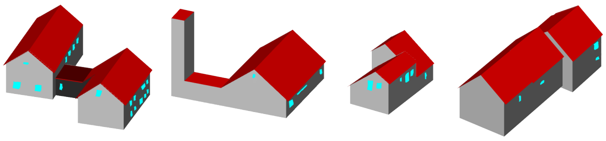


Figure 7. Buildings with windows as CityGML objects

Texture mapping on the four buildings exposed more than 100 image candidates for each polygon due to the highly overlapping flight pattern. In the end a maximum of 15 bitmaps has been kept after the visibility test and redundancy removal. The worst-case GSD on the polygons ranges from 5 to 11 cm depending on the surface slope and building orientation on the ground relative to the camera. The l3tex+ software written in C++ needed 7:42 minutes for the 73 polygons in total on a 2012 server. A great share was absorbed by the I/O-intensive image validation and the visibility test taking over three minutes each. The time required for texture placement alone reduced from four to about two seconds when the partially parallel algorithm was enabled using eight threads. Because these values are too small to allow an assessment of the change in performance due to the limited quantity of structures texture mapping was repeated on a larger data set comprising 75 buildings which were extracted with the initial reconstruction algorithm only. For its 632 faces texture placement took 3:29 minutes on a single CPU core and 51 seconds using eight threads of execution concurrently. Figure 6 depicts the four colored sample buildings. Visualization glitches can be spotted particularly on the façades because the reconstruction does not match reality and due to image orientation inaccuracies.

Window classification on the only texture atlas using OBIA is automatic and occupied two CPU cores on the eCognition workstation for about two minutes for a bitmap of 2048 x 1024 pixels. No new accuracy analysis has been performed since completeness and correctness numbers for the Heligoland data set have been already presented in the paper cited above. Figure 7 illustrates the semantically annotated CityGML objects of the four buildings.

8. CONCLUSION

In this paper a reconstruction approach for buildings with discontinuities and roof overhangs has been presented that relies on aerial images acquired by a decent oblique camera system only. The dense point clouds on the wall surfaces derived from the tilted views enable voxel-based algorithms to accurately model structural breaks. Also, the input data has been successfully utilized for automatic high-quality texture mapping and subsequent window extraction. In this context a partially parallel placement algorithm has been developed which in practice achieves a considerable speed-up for a problem with inherent data dependencies.

However, even though the proposed method performs well on an exceptional data set scaling the involved algorithms to larger areas still remains to be evaluated. In particular when a high image overlap is not feasible and gaps have to be bridged by the voxel-based algorithm the probability for reconstruction errors will increase. This issue is currently under examination, and a first solution based on point cloud extrapolation has shown promising results.

Moreover, difficulties persist to obtain appropriate three-dimensional ground truth beyond the cadastral maps for a thorough accuracy analysis of the developed algorithms. Undoubtedly, having a standardized multi-purpose artificial test scene suitable for both reconstruction and texture mapping would be very beneficial. Because aspects like resolution flexibility, illumination models and reproducibility must be addressed consistently first obtaining a basically error-free reference will remain an ambitious task.

ACKNOWLEDGMENTS

We would like to especially thank Anna Maria Poznanska for kindly providing us the window classification data during her leave.

REFERENCES

Brauchle, J., Hein, D. and Berger, R., 2015. Detailed and highly accurate 3D models of high mountain areas by the MACS - Himalaya aerial camera platform. *Int. Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 40, pp. 1129–1136.

Fischler, M. A. and Bolles, R. C., 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* 24(6), pp. 381–395.

Fisher, Y., 1992. Fractal image compression. *SIGGRAPH'92 Course Notes*.

Frommholz, D., Linkiewicz, M. and Poznanska, A. M., 2016. Inlining 3D reconstruction, multi-source texture mapping and semantic analysis using oblique aerial imagery. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLI-B3, pp. 605–612.

Frueh, C., Sammon, R. and Zakhor, A., 2004. Automated texture mapping of 3D city models with oblique aerial imagery. *3D Data Processing, Visualization and Transmission Proceedings* pp. 396–403.

Haala, N., Rothermel, M. and Cavegn, S., 2015. Extracting 3D urban models from oblique aerial images. *Joint Urban Remote Sensing Event (JURSE) 2015 Proceedings* 38, pp. 1 – 4.

Hirschmüller, H., 2008. Stereo processing by semi-global matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30(2), pp. 328–341.

Kruskal, J., 1956. On the shortest spanning subtree and the traveling salesman problem. *Proceedings of the American Mathematical Society* 7, pp. 48–50.

Low, K.-L., 2002. Perspective-correct interpolation. *Technical writing, Department of Computer Science, University of North Carolina at Chapel Hill*.

Mayer, S., 2004. Automatisierte Objekterkennung zur Interpretation hochauflösender Bilddaten in der Erdfernerkundung. PhD thesis, Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II.

Panday, U. S. and Gerke, M., 2011. Fitting of parametric building models to oblique aerial images. *Int. Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 38, pp. 233–238.

Remondino, F. and Gehrke, M., 2015. Oblique aerial imagery - A review. *Photogrammetric Week '15* 12, pp. 75–81.

Rupnik, E., Nex, F. and Remondino, F., 2014. Oblique multi-camera systems - orientation and dense matching issues. *Int. Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 40 (3/W1), pp. 107–114.

Smith, M., Hamruni, A. M. and Jamieson, A., 2009. 3-D Urban Modelling using Airborne Oblique and Vertical Imagery. *Archives of the ISPRS 38-1-4-7/W5*.

Stilla, U., Kolecki, J. and Hoegner, L., 2009. Texture mapping of 3D building models with oblique direct geo-referenced airborne IR image sequences. *Archives of the ISPRS 38-1-4-7/W5*.

Sun, Y., Shahzad, M. and Zhu, X., 2016. First prismatic building model reconstruction from TOMOSAR point clouds. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLI-B3, pp. 381–386.

Xiao, J., 2012. Automatic building outlining from multi-view oblique images. *Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* I-3, pp. 323–328.

This contribution has been peer-reviewed.
doi:10.5194/isprs-archives-XLII-1-W1-465-2017