AUTOMATIC 3D BUILDING RECONSTRUCTION FROM AIRBORNE LASER SCANNING AND CADASTRAL DATA USING HOUGH TRANSFORM

Jens Overby [a, *], Lars Bodum [a], Erik Kjems [a], Peer M. Ilsøe [a]

[a] Centre for 3DGI, Aalborg University, Niels Jernes Vej 14, 9220 Aalborg, Denmark - (jeo, lbo, kjems, ilsoe)@3dgi.dk

KEY WORDS: Building, Reconstruction, Algorithms, Point Cloud, Geometry, Laser scanning, Virtual Reality, GIS

ABSTRACT:

Urban environments of modern cities are described digitally in large public databases and datasets of e.g. laser scanning and ortho photos. These data sets are not necessarily linked to each other, except trough their geometry attributes (coordinates), which are mutually displaced and have a low degree of details. However, it is possible to create virtual 3D models of buildings, by processing these data. Roof polygons are generated using airborne laser scanning of 1x1 meter grid and ground plans (footprints) extracted from technical feature maps. An effective algorithm is used for fixing the mutual displacement of these datasets. The well known Hough Transform is extended to 3D for extracting planes from the point cloud. Generally speaking, planes are rejected if the clusters of points on these planes, do not span a considerable area. Furthermore, it is assumed that valid planes are close to parallel to one of the ground plan lines. Points corresponding to each valid plane are subtracted from the original point cloud, and the Hough Transform is performed on the remainder. By this approach, the disturbing influence of already evaluated points, is avoided. Small variations of the roof surface might lead to multiple slightly differing planes. Such planes are detected and merged. Intersecting planes are identified, and a polygon mesh of the roof is constructed. Due to the low precision of the laser scanning, a rule-based post-processing of the roof is applied before adding the walls.

1. INTRODUCTION

Virtual 3D city models have become increasingly popular in recent years. Often these models are used for supporting urban planning processes and decision making. Other applications of 3D city models are in the tele communication industry and for analysing propagation of noise and air pollution in downtown urban areas.

The so-called “position attributed information” is information which is attached to a position, but isn't in itself geometrical. It doesn't span any volume, neither has colours or surfaces – it can be considered as zero dimensional. Most information in public databases today is zero-dimensional, and again most of this is attributed to a position. There is an increasing need for linking 3D geometrical and non-geometrical information, which gives whole new possibilities for querying information in public information servers. Queries are done directly by navigation in 3D. However, the roofs of buildings constitute a crucial part of any city model. The whole of all building roofs generates a “roof landscape”, which is by far the most dominating picture when “flying” over a city model.

One of the widely used methods for acquiring 3D geometrical context is by using airborne laser scanners. The grid produced by such scannings must be processed, ideally to reduce information to the lowest possible level of entropy. This process introduces removal of noise, intelligently adding missing information, and removing the huge amount of redundant information present in the surface scan, where in fact only 3 degrees of freedom (d.o.f.) are necessary for defining any unbounded plane in 3D. (A normal vector where the length is the distance from the origin of the coordinate system).

The data used for the building reconstruction presented in this article is a regular grid of 1x1 meter, acquired by an aeroplane mounted LIDAR system. However, the algorithms also work on irregular grids. For building reconstruction this density is low, considering that at least three points are required for defining a plane, and many of these points are inaccurate. Furthermore, the regular grid is produced by interpolation, which has introduced even more noise to the data. All these things considered, small roof objects like chimneys, windows, and cornices become noise instead of valuable data. It is also very difficult to identify vertical gable ends separating different parts of a building roof.

In order to acquire a high density laser scanning, a helicopter based system is necessary. Such systems can acquire point densities of five to ten points per square meter [Baltsavias, 1999]. Algorithms dealing with high density laser scannings are described in e.g. [Haala and Brenner, 1997] and [Vosselman, 1999]. Only very few areas are scanned with less than a 1x1 meter grid today, which is why there is a need for building reconstruction algorithms, which can operate in low density digital elevation models. This paper reports on our progress in this particular field of research.

2. ROOF PLANE DETERMINATION

A very important assumption is that building models can be described by planar faces. Most algorithms for building reconstruction work by detection of planar faces [Hoover et al., 1996]. However, many of these algorithms require the computation of local surface normals. The problem is that these algorithms are extremely sensitive to noise [Vosselman, 1999]. The plane detection can run into local minima, and thereby fail to find an obvious plane seen in the global perspective.

A robust algorithm for this purpose is the Hough Transform [Hough, 1962] extended to 3D [Schindler et al., 2003]. The classical Hough Transform has been extensively used in the

* Corresponding author.

image analysis field for a large range of applications including primitive detection and object recognition.

2.1 Hough Transform

Given a point $p(x_p, y_p)$ and the general equation of a straight line in slope-intercept form, $y_p = ax_p + b$. Infinitely many lines pass through (x_p, y_p), but they all satisfy the equation $y_p = ax_p + b$ for varying values of a and b. If a 2D image contains several points on a straight line, the lines of these points in parameter space will intersect and the position of the intersection yields the parameters of the line in the image.

A problem with describing a line in terms of a slope is that it approaches infinity as the line approaches the vertical. A way around this difficulty is to use the normal representation of a line: $x_p \cos\theta + y_p \sin\theta = s$.

The general Hough Transform is easily extendable to planes in three dimensions. A plane $\Pi \in \Re^3$ is uniquely defined by a triplet (θ, φ, s), where $\theta \in \{0, 2\pi\}$ and $\varphi \in \{-\pi/2, \pi/2\}$ denote the two angles (azimuth and elevation) associated with the spherical representation of the plane's unit length normal vector $n = (n_x, n_y, n_z)$ and $s \geq 0$ denotes the distance from the origin of the coordinate system to plane Π (Figure 1). The three components of n are expressed as follows:

$$n_x = \cos(\varphi)\cos(\theta),\ n_y = \cos(\varphi)\sin(\theta),\ n_z = \sin(\varphi) \qquad (1)$$

The distance s from the origin of the coordinate system to plane Π is expressed as $s = |\, n_x x + n_y y + n_z z \,|$, which yields:

$$s(\theta, \varphi) = \cos(\varphi)\cos(\theta)\, x_p + \cos(\varphi)\sin(\theta)\, y_p + \sin(\varphi)\, z_p \qquad (2)$$

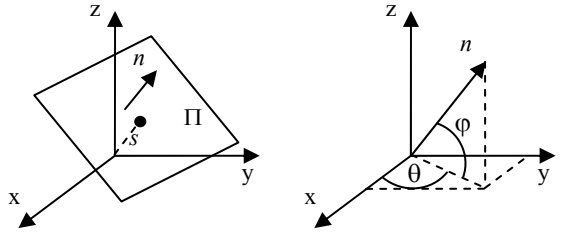


Figure 1. Parameterization of planes in $\Re^3$.

Each sampling point p yields a function $s(\theta, \varphi)$, and the points in parameter space where three or more functions intersect, are planes spanned by tree or more sampling points.

The computational attractiveness of the Hough transform arises from subdividing the parameter space spanned by θ, φ, and s into so-called accumulator cells $(\theta_i, \varphi_j, s_k)$, with $i \in \{0,1,...,N_\theta\text{-}1\}$, $j \in \{0,1,...,N_\varphi\text{-}1\}$, and $k \in \{0,1,...,N_s\text{-}1\}$, where N_θ, N_φ, and N_s are the number of samplings along the axes (θ, φ, s).

If $s(\theta_i, \varphi_j)$ is positive, it is quantified to its closest sampling value s_k, and the accumulator $h(\theta_i, \varphi_j, s_k)$ is incremented with a weight. Negative values are not relevant, as any plane described by a normal vector n and a distance from the origin s, can be described by the negative pair $-n$ and $-s$.

The result is that cells with highest accumulation, give the parameters $(\theta_i, \varphi_j, s_k)$ of the planes spanned by laser scan points in cartesian space.

2.2 Determining true 3d planes

Processing the laser scan through the Hough Transform yields a huge amount of planes. As earlier mentioned, the method works globally on the data and thereby planes can be generated by points which in reality are present on different roof surfaces having completely different surface normals.

The strategy that is used when extracting planes from parameter space is to start with the accumulator cell with most hits. This plane is then evaluated according to its corresponding point-cloud using the *Projected Cluster-area Test* (section 2.2.1). If it passes the test, the points representing it are removed from parameter space, and again the cell that has most hits will be evaluated. If it does not pass, evaluation just moves on to the cell with second most hits, and so on. In our case, extraction ends when the maximum number of hits is less than 5.

The importance of removing the point cloud of an identified plane appears clearly from figure 3. Here the locations in parameter space which are in fact true real world planes are marked with arrows, and the numbers are the order in which these are identified. When removing the point cloud of a plane, the graph in parameter space will change and remaining planes will then be more obvious than their initial appearance.

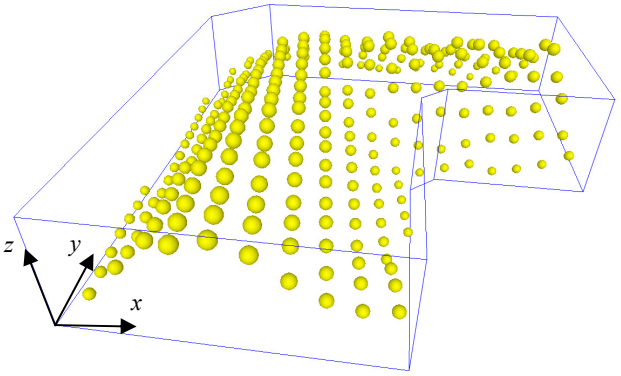


Figure 2. Surface scan in Cartesian Space.

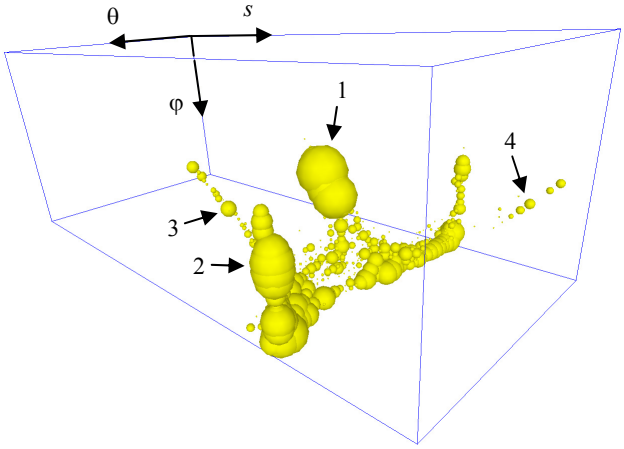


Figure 3. Surface of figure 2 Hough transformed to Parameter Space. The parameter space is sampled along the axes, and the accumulator cells are shown as spheres. For illustrational purposes, only cells with more than 30 hits are shown. The radius of each sphere is proportional to the corresponding number of hits in its accumulator.

Finally, the resulting planes must pass a footprint parallel check filter, as it is assumed that roof planes are parallel to at least one of the lines of the footprint [Vosselman, 2001]. This can filter many erroneous planes caused by e.g. trees, failed correlation of displaced datasets, parked cars, and general noise.

2.2.1 Projected Cluster-Area Test

Each Hough-generated plane extracted from accumulator cells is evaluated using a method based on projected areas of point clusters, which has proven to be very efficient for effective filtering of erroneous planes.

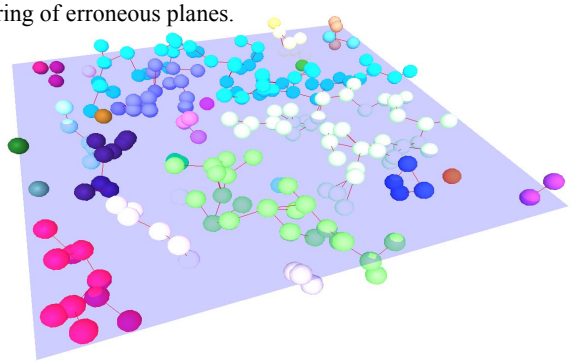


Figure 4. The point cloud of a plane is divided in clusters.

When using accumulator cells for determining planes, the accuracies of the plane parameters are not optimum. Adjustments are made by calculating the Least-Squares plane using LU decomposition [Press et al., 1988].

All points belonging to a plane are grouped in clusters having mutual distances less than an appropriately chosen threshold ρ. Each cluster has a defined area of projection on the belonging Least-Squares mean plane.

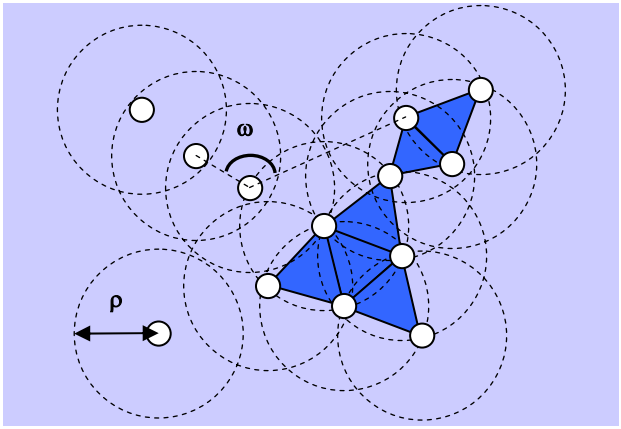


Figure 5. When projected on the Least Squares mean plane, each cluster spans a 2D area. Creating a triangle requires $\omega < \tau$. In this case $\tau = \pi/2$.

The area of projection is the triangulated area of the projected points on the plane. However, it is most likely that a concentrated cluster of points is present on a true real world plane. Consequently, triangles are only created if all angles of a triangle are less than a threshold angle τ and all mutual distances are less than the threshold distance ρ.

A plane is accepted if the summed area of all its clusters is considerable, relatively to the amount of points *n* making up the plane, expressed by:

$$\sum A_{cluster} > \frac{n}{K_{area}} \qquad (3)$$

In formula 3, K_{area} is a tuneable constant.

2.3 Merging planes

Sometimes a real-world true plane generates more than one plane using the previously described methods. In these cases it is advantageous to merge such planes to secure uniqueness.

2.3.1 Finding merge candidates

Given a pair of planes $[\Pi_1, \Pi_2]$. Whether these planes are candidates of merge is evaluated by following figures in combination:

- The dot product of unit length plane normals. If this figure approaches 1, planes are parallel.
- The two distances from Π_1 and Π_2 to the centre of mass of the merged point clouds of Π_1 and Π_2.

If planes $[\Pi_1, \Pi_2]$ and $[\Pi_2, \Pi_3]$ are candidates, these sets are combined to a common set of merge candidates $[\Pi_1, \Pi_2, \Pi_3]$.

2.3.2 Merging planes in pairs utilizing Variance

Now having a set of merge candidates $[\Pi_1, \Pi_2,..., \Pi_n]$, the planes are again evaluated as pairs. Each possible combinatorial pairing of planes $[\Pi_j, \Pi_k]$ within the set is temporarily merged and the variance $V_{j,k}$ defined as the statistical variance of distances from the point clouds $[P_{j,1}, P_{j,2},..., P_{j,m}]$ and $[P_{k,1}, P_{k,2},..., P_{k,n}]$ to the merged plane $\Pi(j,k)$ is calculated. The merged plane is calculated using Least Squares Method on the two point clouds. Normal vector and distance from the origin of the coordinate system given by $N_{\Pi(j,k)}$ and $S_{\Pi(j,k)}$, respectively.

$$V_{j,k} = \frac{\sum_{i=1}^{m} (P_{j,i} N_{\Pi(j,k)} S_{\Pi(j,k)})^2 + \sum_{i=1}^{n} (P_{k,i} N_{\Pi(j,k)} S_{\Pi(j,k)})^2}{m+n} \qquad (4)$$

Finally, the pair of planes $[\Pi_j, \Pi_k]$ with the smallest variance $V_{j,k}$ is merged, if this variance is below a threshold value. If planes are merged, the process of merging planes in pairs is repeated.

3. MODEL RECONSTRUCTION

Creating the building model from the set of detected planes, each with a corresponding point cloud, is in fact the most difficult part of the whole process of building reconstruction.

The main problem is the lack of precision in the source laser scanning, resulting in missing planes not identifiable in the parameter space of the Hough Transform. It is also difficult to prevent that erroneous planes eventually will pass the filters. Accordingly, the input to the model reconstruction is potentially erroneous, and the algorithms should be able to handle this.

The strategy used is simply to go on with cutting the detected planes, erroneous or not, and finally do a repair procedure of the generated mesh.

3.1 Cutting the planes

All planes of the total set of detected roof planes and vertical wall planes created from the footprint are cut combinatively with each other, provided that the line of intersection is within a pre-specified minimum distance of the point cloud. This results in a potential huge amount of bounded faces, where all faces lying outside of the footprint are removed.

3.2 Filtering faces

The roof of the building is now represented by bounded faces, of which some are overlapping in the z-direction, typically when gable ends meet. Overlapping faces are removed if these are not represented by any points of the point cloud. If overlaps are still present, these faces are given probability priorities.

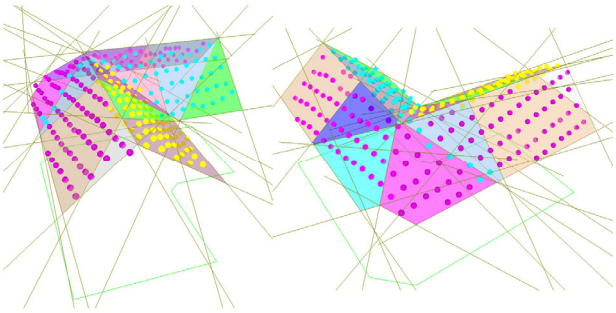


Figure 6. Roof faces generated by cutting planes and filtering overlapping faces. The source point cloud is identical to that of figure 2.

3.3 Removing degenerate parts of mesh

When performing a weld operation on a mesh (connecting faces through edges), edges are only allowed to be connected to two faces – right and left. If an edge is bounding more than two faces, we call it a "hot spot" edge. An example of such a hot spot is shown in figure 7, marked by a dashed line.

By separating the mesh at the hotspot edge, two meshes are defined in figure 7. The face marked by the arrow defines the erroneous mesh which is excluded, and the other faces define the other mesh, which is then the true resulting mesh.

Separate meshes can also be connected through a common vertex, which is then called a "hot spot" vertex.

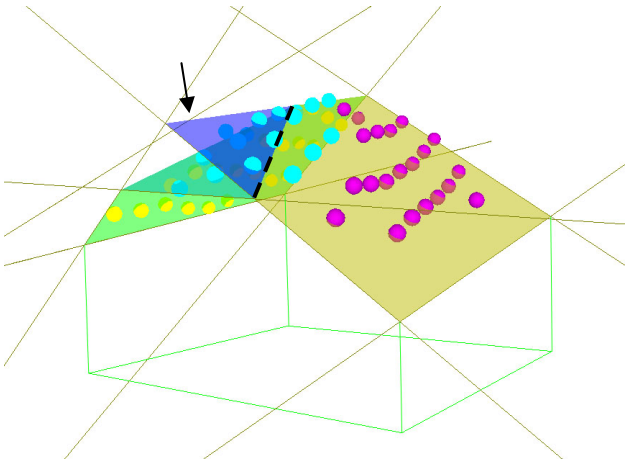


Figure 7. This mesh is erroneous, because the dashed edge is a hot spot edge connected to more than two faces.

3.4 Fill holes in mesh

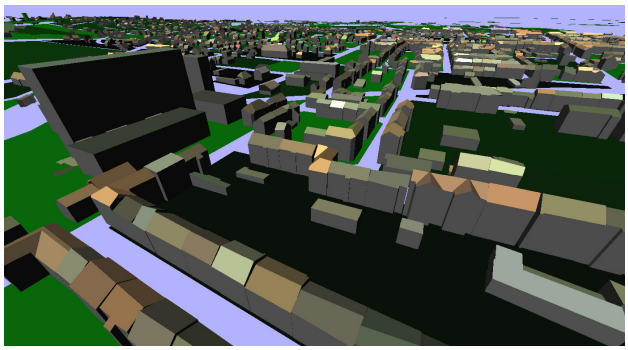
Unfortunately, holes will appear in the mesh when removing degenerate parts. Filling holes is by no means trivial, but luckily it is a known discipline within Computational Geometry [Barequet et al., 1997].

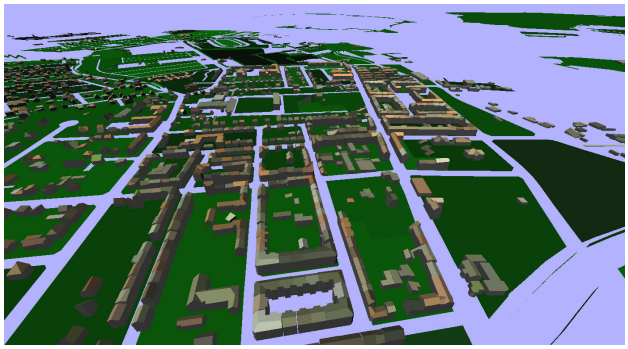
3.5 Finalization

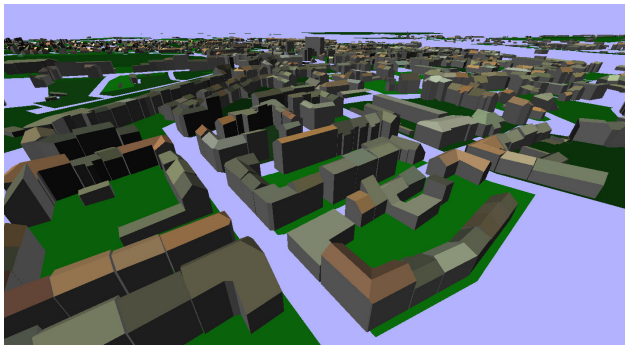
In the final step of the reconstruction, all parallel adjacent faces are merged. Thereby faces might be concave, but the size of the BRep is reduced considerably. Furthermore, a RGB colour of a guaranteed "on the roof" spot is extracted from the ortho photo, and this colour code is added to the roof mesh.

The resulting per building geometry is dumped in VRML format, of which there are standard loaders for various programming languages.

4. RESULTS







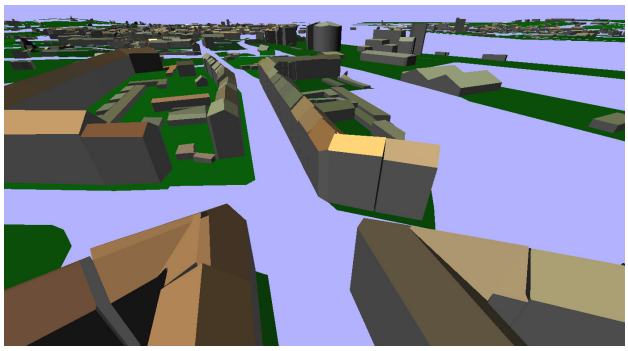


Figure 8. Four screen shots in selected areas in the site of the 24,887 buildings created for the city of Aalborg.

5. CONCLUSIONS

The overall performance of the algorithms is surprisingly good, considering the relatively low resolution of the laser scanning in combination with distortions caused by a very unfortunate interpolation of the raw data. In addition, the mutual displacements of the two datasets (footprints and laser scanning) are sometimes difficult to correlate, and undoubtedly the cause of many problems. In our case, creating a building has a success rate of 84 percent, although the qualities of these are not always satisfactory.

In particular, the following algorithms are believed to be very efficient and general for the purpose:

- Projected cluster area filter used for extraction of planes from Hough space.
- Degenerate mesh filter where error detection is based on mesh welding techniques.

These algorithms are innovative and not described in similar articles like e.g. [Vosselman, 2001].

However, there are a number of obvious problems and drawbacks of the methods described in this article.

5.1 Height jump lines

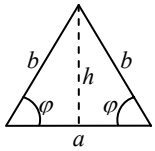
Currently, we have no method for detecting height jump lines, as the sampling resolution is considered too low for this purpose. Instead, by splitting the footprints into smaller units, the influence of height jump lines is reduced, as the footprints are often cut exactly at lines convergent to these. However, as the resolution of laser scannings increase, solutions of height jump lines are needed.

5.2 Aligning ridges and gables

In our methods we rely on using the Hough Transform for detection of planes in $\Re^3$. Planes are among the most simple primitives in $\Re^3$, but not necessarily the "atomic" unconstrained primitives of building roofs.

In the case of roofs, some of the d.o.f. of a plane are often constrained by another adjacent plane on the other side of the ridge, just like the dependencies between the two sides of equal length in an isosceles triangle. A *Ridge Plane Pair* is two planes intersecting at the line of the ridge. Roof planes which have an angle of more than 20 deg from horizontal often have such accompanying planes. The line of intersection (the ridge) is in most cases a strictly horizontal line, and the angle from horizontal is often the same for both planes.

Figure 9. The Isosceles Triangle is constrained in the same manner as the Ridge Plane Pair.



Furthermore, planes are often strictly parallel to a line of the footprint, meaning that the orientation of a roof plane is achievable by rotating it from horizontal around an axis convergent to one of the footprint lines.

Utilizing this knowledge on the dependencies of planes, major achievements in precision is expectable. The Hough Transform is a general method for detecting primitives in parameter space. The more complex primitives, the more parameters to describe these in parameter space, and thereby, more dimensions to sample and traverse. However, being able to set up constraining rules drastically reduces the number of samplings necessary in each dimension of parameter space. Consequently, it is possible to use the Hough Transform for detecting more complex primitives, which should be done in future research.

6. ACKNOWLEDGEMENTS

The laser scanning and ortho photos were provided by COWI A/S, Consultancy within Engineering, Environment and Economics. Geometries of footprints and cadastres from the technical feature map were provided by the Municipality of Aalborg. The authors thank both providers for making these data available.

7. APPENDIX – PRE PROCESSING OF SOURCE DATA

Proper pre processing of the source data is of vital importance for the reconstruction algorithms.

7.1 Fixing geometries of footprints

It is desirable that the footprints as much as possible reflect the actual 2D borders of the individual buildings.

Figure 10. The background upper picture shows a snapshot of raw footprints from the technical map. The lower front picture shows the resulting shapes of these footprints when cut by polygons of cadastres.

The footprints from the technical feature map do not always cover the individual buildings. Sometimes a whole building block is grouped in a footprint, which is not desirable. As previously mentioned, there is no detection of height jump lines in the algorithms and consequently, units must contain as few height jump lines as possible.

By cutting the raw footprints with polygons of cadastres, it is possible to reduce the frequency of height jump lines considerably. Furthermore, it is desirable to remove very small footprints, because the reconstruction of these is likely to fail due to the low resolution of the laser scanning. All footprints less than 50 m^2 are removed.

7.2 Fixing displacements of data sets

Unfortunately, the data sets of the technical feature map containing footprints and the 1x1 meter grid laser scanning are not aligned. The displacements are up to 1.5 meters. However, the displacement can be calculated numerically, by setting up a cost function $\Phi(x_{disp},y_{disp})$ expressing a cost of displacing the footprint by x_{disp} and y_{disp} along the x and y axes, respectively.

$$\Phi(x_{disp}, y_{disp}) = \sum_{i=1}^{m}\sum_{j=1}^{n}(z_{\max} - Inside(z_{i,j})_{x_{disp},y_{disp}})^2 \quad (5)$$

where:

- z_{max} is the maximum elevation in the source point cloud.
- $z_{i,j}$ is the elevation at point (x_i, y_j) with $i \in \{1,2,...,m\}$ and $j \in \{1,2,...,n\}$.
- m is the number of samplings along the x axis.
- n is the number of samplings along the y axis.
- $Inside(z_{i,j})_{x\_disp,y\_disp}$ is a function that returns the input value $z_{i,j}$ if the sampling point (x_i, y_j) is inside the footprint displaced by x_{disp} and y_{disp}. Else it returns z_{max}.

The task is then to find the displacement (x_{disp}, y_{disp}) which generates the minimum cost, which is done by displacing the footprint in small steps along the x and y axes in the local area of the building.

In order for the numerical displacement to be successful, it is important that buildings to be displaced are isolated from neighbouring buildings. There should be a certain gap to the closest neighbour (in our case a gap of 2 meters is sufficient). Neighbouring buildings must be grouped before the displacement of the group as a whole can be determined.

8. REFERENCES

Vosselman, G., 1999. Building Reconstruction using Planar Faces in Very High Density Height Data. International Archives of Photogrammetry and Remote Sensing, vol. 32, part 3-2W5, pp. 87-92.

Vosselman, G. and Dijkman, S., 2001. 3D Building Model Reconstruction from Point Clouds and Ground Plans. Proceedings of the ISPRS workshop on Land Surface Mapping en Characterization Using Laser Altimetry, 22nd to 24th october 2001, Annapolis, Maryland, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol XXXIV part 3/W 4 Commission III. ISSN 0246 1840, pp.37- 44

Hoover, A., Jean-Baptiste, G., Jiang, X., Flynn, P.J., Bunke, H., Goldgof, D.B., Bowyer, K., Eggert, D.W., Fitzgibbon, A. and Fisher, R.B., 1996. An experimental comparison of range image segmentation algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (7): 673-689.

Hough, P.V.C., 1962. Method and Means for Recognizing Complex Patterns. U.S. Patent 3.069.654.

Baltsavias, E., 1999. Airborne laser scanning: existing systems and firms and other resources. ISPRS Journal of Photogrammetry and Remote Sensing, 54 (2-3): 164-198.

Haala, N. and C. Brenner, 1997. Generation of 3D city models from airborne laser scanning data. Proceedings EARSEL workshop on LIDAR remote sensing on land and sea, pp. 105-112 Tallinn, Estonia.

Schindler, K. and Bauer, J., 2003. Towards Feature-Based Building Reconstruction From Images. WSCG'2003 Short Papers proceedings, Plzen, Czech Republic.

Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T., 1988. Numerical Recipes in C, The Art of Scientific Computing, Cambridge University Press.

Barequet, G. and Subodh, K., 1997. Repairing CAD Models, John Hopkins University, IEEE Visualization '97.