



Automatic 2D modelling of inner roof planes boundaries starting from Lidar data

Fayez Tarsha Kurdi, Mohammad Awrangjeb and Nosheen Munir

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

May 27, 2019

AUTOMATIC 2D MODELLING OF INNER ROOF PLANES BOUNDARIES STARTING FROM LIDAR DATA

Commission WG IV/10

KEY WORDS: Modelling, Segmentation, Lidar, Planes boundaries, 3D point cloud

ABSTRACT:

Despite the large quantity of researches and publications achieved during the last three decades about 3D building modelling by using Lidar data, the question of inner roof plane boundaries modelling needs to be more extracted in detail. This paper focuses on detection and 2D modelling of building inner roof plane boundaries. This operation presents an imperative junction between roof planes detection and 3D building model generation. Therefore, it presents key procedure in data driven approaches. For achieving this purpose, roof boundaries are classified in four categories: outer building boundaries, inner roof plane boundaries, roof details (chimneys and windows) boundaries and boundaries related to non-detectable roof details. This paper concentrates on detection and modelling of inner roof plane boundaries and roof details (chimneys and windows) boundaries. Moreover, it details the modelling procedures step by step that is envisaged rarely in the literature. The proposed approach starts by analysing the adjacency relationship between roof planes. Then, the inner roof plane boundaries are detected. Finally, the junction relationships between boundaries are analysed before detecting the roof vertices. Once the 2D roof model is calculated, the visual deformations in addition to modelling accuracy are discussed.

1. INTRODUCTION

Nowadays, the need for automatic calculation of 3D urban models is increasing incessantly. That is why the acquisition of 3D data by airborne laser scanning has gained a considerable position among the other data acquisition techniques. This system provides fast, day or night, 3D point cloud covering the scanned region, without needing matching processes. The automation possibility of the modelling procedures is the major advantage of Lidar data. There are two principal approaches for automatic generation of 3D building models: model-driven approach or parametric approach, which searches the most appropriate model among basic building models contained in a models library. The second approach is data-driven or non-parametric approach. It calculates 3D model for a complex or simple buildings by using series of more or less complex operations (Tarsha Kurdi et al., 2007). Hence, the data-driven approach responses to extensive cases of building typology, whereas the model-driven approach stays limited for the given primitive buildings.

Once Lidar technology started to become popular, it was realised that the modelling of roof planes and their topological relationships are necessary for generating 3D building models. Therefore, significant efforts were made towards this objective. The detection and modelling of 2D roof plane boundaries present a transition step between the detection of roof planes and the calculation of 3D building model. The majority of researches in bibliography considers this step as a less important step than the first two ones. Then, this procedure was not given its correct position that it merits. Therefore, the goal of this paper is to focus on automatic detection and modelling of 2D roof plane boundaries.

At this stage, it is important to know that several terms are used in the literature to refer to line segments drawn inside building boundary in 2D roof models. They are called sometimes

“breaklines” as in (Briese, 2004) and (Shan, Toth, 2008). Other times they are called “linear features” (Sohn, Dowman, 2007) and (Perera et al., 2012), and some authors called them “plane boundaries”, “inner boundaries” or “boundaries” (Xiong et al., 2015), (Zhang et al., 2014) and (Sohn et al., 2008). Breaklines and linear features have almost the same meaning, but roof plane boundaries differ from breaklines because one breakline may represent multiple planes’ boundary lines while the opposite is not always right.

2. RELATED WORK

For modelling inner roof plane boundaries, (Rottensteiner et al., 2005) suggested to take into account the uncertainty of both the two neighbouring planes and the approximate positions of the vertices of the polygon segments. Hence, statistical tests were introduced to minimise its dependence on the thresholds at all stages, including detection and classification of roof planes, boundaries, and step edges. Another approach by (Ghaffarian et al., 2016) utilised the Purposive Fast Independent Component Analysis algorithm (PFICA) for detecting roof plane boundaries, then Douglas-Peucker algorithm (Douglas, Peucker, 1973) is used after enhancing the obtained result by morphological filter.

In the same context, it is suggested to use aerial images in addition to Lidar data. (Schenk, Csatho, 2002) detected roof plane boundaries from panchromatic images based on the texture discontinuity and utilised them to refine the results from Lidar data. (Sohn, Dowman, 2007) used the IKONOS multispectral images for roof plane boundary detection. (Sampath, Shan, 2006) detected roof planes and plane boundaries simultaneously by using exclusively Lidar data. Hence, they applied the linear space theory to separate plane boundaries from planar points. (Gross, Thoennessen, 2006) used the same principle for detecting the linear features from 3D point cloud. However, several authors suggest detecting firstly roof planes (Vosselman, Dijkman, 2001). Then, they considered the set of roof planes as

input for detecting planes boundaries. (Sohn et al., 2008) applied region growing and TIN analysis for extracting roof planes. Then, Hough transform and neighbouring plane intersection lines are applied in order to extract planes boundaries. (Xiao et al., 2015) started also by extracting roof planes. Then, plane boundaries and vertices were calculated by using an undirected graph model of the building roof. By consequent, for obtaining reasonable and usable models with regular building boundary segments, the RANSAC algorithm is applied to find long segments and the corresponding parameters. Furthermore, another constrains were performed by applying parallel merging and orthogonal adjustments.

At this stage, it is important to refer that it is sometimes advised to start by modelling the outer building boundaries. (Awrangjeb, 2016) suggested a new algorithm for outer building boundary extraction. This algorithm has two major steps: outer boundary identification by using Delaunay triangulation and boundary tracking and smoothing before applying the least squares theory.

Some studies developed solutions for analysing topological relationships between building roof planes. (Ameri, 2000) used the Voronoï diagram of the roof planes. (Jaya et al., 2018), (Xiong et al., 2015) and (Perera et al., 2012) used the roof topology graph based on the closed cycle graphs summarising the topological relationships between the roof segments.

3. DATA

In order to test our approach on different point densities, two types of data are used (Table 1).

	Hermanni	Strasbourg
Acquisition	End of June 2002	Begin September 2004
Sensor	TopoEye	TopScan (Optech ALTM 1225)
Points density	7-9 points / m ²	1.3 points / m ²
Flight height	200 m	1440 m

Table 1. Characteristics of the two datasets used in this paper

The first test site ‘‘Hermanni’’ is a residential area in periphery of Helsinki, where large and spaced storey houses are surrounded by vegetation. This point cloud belongs to the building extraction project of EuroSDR (www.eurosd.org). The second point cloud is of Strasbourg city in France. Several zones with different typology natures are covered. But point density and accuracy are lower in compression to Hermanni point cloud.

The following paragraph presents the calculation of building label image. This image will be the input data for detecting and modelling roof planes boundaries.

4. CALCULATION OF BUILDING LABEL IMAGE

In order to detect automatically the roof planes, the RANSAC algorithm (RANdom SAMple Consensus) was extended to exceed its limitations. Its major limitation is that it searches to detect the best mathematical plane among 3D building point cloud even if this plane does not always represent a roof plane. The proposed extension allows harmonizing the mathematical aspect of the algorithm with the geometry of a roof. At this stage, it is important to note that the building point cloud is the input of the extended RANSAC algorithm. The final result of roof planes detection is a building label image which represents the building roof in 2D. This image is segmented according to the roof planes

(Figure 1e). For obtaining more details about the extended RANSAC algorithm see (Tarsha-Kurdi et al., 2008). Once the building label image is calculated, the automatic modelling of roof planes boundaries can be started.

5. AUTOMATIC DETECTION OF THE INNER ROOF PLANE BOUNDARIES

After detection of roof planes and calculating building label image (Figure 1e), the first step toward 2D roof modelling is the detection of the roof plane boundaries. In the literature (Ameri, 2000) and (Rottensteiner, 2003), it is suggested to use of Voronoï diagram to achieve this task. However, this solution is unsatisfactory because it creates distortions not only on the actual position of the planes boundaries, but also on adjacency relationships between planes. Furthermore, (Vosselman, Dijkman, 1999) suggested to calculate the intersection between the adjacent planes for detecting the plane boundaries. This solution is not advised because the roof plane equations do not describe precisely the original roof planes in the presence of noisy points and indiscernible roof details. This generates sometimes undesirable deformations in the level of 2D building models. Therefore, a new method is proposed for constructing a more reliable model. This method detects the inner roof plane boundaries directly from the building label image. This choice is adopted because the building label image is calculated and corrected precisely (Tarsha Kurdi et al., 2008) and the position of the inner roof plane boundaries in this image is satisfactory.

The input data for detecting roof boundaries is the roof label image, which presents segmented roof (Figure 1e). In order to detect the roof plane boundaries, there are four successive steps: identification of the adjacency relationship between neighbouring planes, detection of roof plane boundaries, analysis of the junction relationships between boundaries and automatic detection of roof vertices.

5.1 Adjacency relationship between roof planes

In order to identify neighbourhood relationship between roof planes, a new square matrix called *neighbourhood* matrix or so-called *plane_adjacent* matrix is calculated. The number of columns or lines in this matrix is equal to the number of detected roof planes. For example, since the roof of the building presented in (Figure 1e) is composed of seven planes; therefore, the matrix *plane_adjacent* is a 7x7 matrix. This matrix is a binary matrix, i.e. It contains only two values (‘0’ and ‘1’). For example: if two planes 1 and 2 are adjacent, then the cell (1, 2) of the *plane_adjacent* matrix is equal to ‘1’, moreover the cell (2, 1) must have the value ‘1’. The symmetry in this matrix represents an undesirable redundancy. In order to avoid that, the following rule can be set: one cell has to be filled by the value ‘1’ if the two planes are adjacent and the line number is lower than the column one; otherwise, the cell is set to ‘0’. Following this rule, the cell (2, 1) of the *plane_adjacent* matrix takes the value ‘0’.

In order to fill the *plane_adjacent* matrix, the following operation is repeated for each one of the roof planes, i.e. Plane 2 in the building (Figure 1e) is considered: the building label image matrix (Figure 1e) is called *image_seg*. Figure 1a visually represents the binary image of Plane 2. The name of this matrix is *plane_2*. Figure 1c represents the negative image of *plane_2*; it is called *negative_p2*. Then, a band of pixels around this plane is added to the matrix *plane_2* defining a new matrix called *plane2_extended* (Figure 1b).

At this stage, a new matrix called *plane_2_adjacent* is defined (Equation 1 and Figure 1d). This matrix allows determining all adjacent planes of Plane 2. Then, it allows filling the part concerning the plane in the *plane_adjacent* matrix as shown in Table 2.

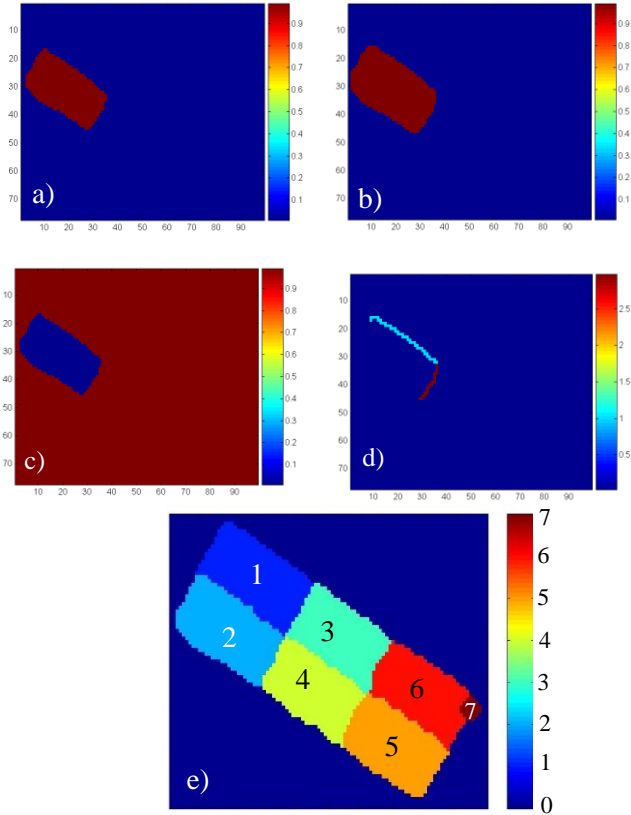


Figure 1. Calculation of the *plane_adjacent* matrix. Visualisation of a) the binary matrix *plane_2*; b) the matrix *plane2_extended*; c) the matrix *negative_p2*; d) the matrix *plane_2_adjacent*; and e) Building label image. The colours in 1d and 1e represent the plane numbers.

$$plane_2_adjacent = plane2_extended * negative_p2 * image_seg \quad (1)$$

where «*» is the element by element multiplication.

N° plane	1	2	3	4	5	6	7
1	0	1	1	1	0	0	0
2	0	0	1	1	0	0	0
3	0	0	0	1	1	1	0
4	0	0	0	0	1	1	0
5	0	0	0	0	0	1	0
6	0	0	0	0	0	0	1
7	0	0	0	0	0	0	0

Table 2. *plane_adjacent* matrix of building shown in Figure 1e

Once the *plane_adjacent* matrix is calculated completely, that means the neighbourhood relationships have been identified. Then, the step of roof plane boundaries detection is started.

5.2 Detection of inner roof planes boundaries

In the building label image, the inner plane boundaries are represented by the pixels located on the border between the adjacent planes. The first step is to detect these pixels. For this purpose, a 3 x 3 moving window is used for testing the vicinity of each pixel. This operation allows localising pixels of inner planes boundaries (Figure 2a).

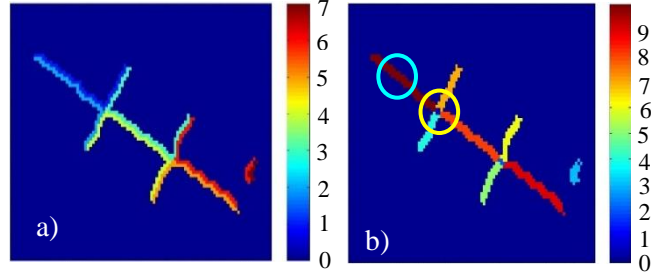


Figure 2. a) Detection of the inner planes boundaries; b) *raster_inner_boundary* matrix; in (Figure 2a) the colours represent the plane numbers but in the (Figure 2b) the colours represent the boundary segment numbers

In Figure 2a, it can be noted that the boundary of two adjacent planes is defined by two pixel rows, one belongs to the first plane and the other belongs to its neighbour. Therefore, the value of each pixel is the number of original containing plane. Then, these couples of pixel rows are merged to give two pixel rows of the same value which is the boundary segment number. Figure 2b represents the visualisation of matrix (*raster_inner_boundary*), in which the pixel numbers represent the boundary segment numbers.

In *raster_inner_boundary* matrix (Figure 2b), two kinds of pixels are met: pixels of boundaries between two adjacent planes (boundary circled in blue in Figure 2b) and pixels located at the intersection of more than two planes (boundary circled in yellow in Figure 2b). The last one seems as the same as a vertex more than a boundary segment. But it is considered as a boundary segment because it appears in the *plane_adjacent* matrix. That is why it is called a punctual boundary. To locate these pixels, it suffices to note that they appear twice in the *plane_adjacent* matrix and each pixel has a minimum of three neighbouring planes, (in Figure 2b there are four neighbouring planes of the boundary circled in yellow).

Once the matrix *raster_inner_boundary* is calculated, the plane boundaries are localised and saved as a new list. It can be observed that one boundary segment is defined by two adjacent pixel rows; each one belongs to a different plane. To limit them as one row, the median row is calculated. Finally, another problem is met frequently during the 2D roof modelling which is the presence of roof details located on the roof plane boundaries (for example, fireplace on the boundary). In this case, despite the boundary between the two planes is defined only one time in *plane_adjacent* matrix, it is decomposed into two portions. For this purpose, a region growing algorithm is applied to detect each portion independently.

After the detection of the inner roof plane boundaries, the automatic detection of the roof vertices is initiated. For carrying out this procedure the junction relationships between the boundaries has to be studied.

5.3 Analysis of the junction relationships between boundaries

Since the processing till now is in 2D, a vertex means the junction point of two boundaries at least. The importance of analysing the junction relationships between inner boundaries comes from the fact that a vertex is located at the junction of several boundary extremities. But the vertex coordinates (X, Y) will not be exactly the same for all boundaries extremities. That is why it must be able to find out how to join the boundaries extremities. The innovative idea in this contribution is to describe the junction relationships between boundaries through a boundary number in a specific order. For example, if all boundaries surrounding one plane are taken together, then each boundary corresponds to an order of succession. The proposed algorithm begins by highlighting all roof boundaries, both inner boundaries and outer ones. This leads to create a new matrix *all_roof_boundaries* obtained by adding the building outer boundaries matrix to the matrix *raster_inner_boundaries* (Figure 3c).

Before presenting an example, it is important to explain the method of computing of the junction relationship between the boundaries enveloping one plane. Starting from the *all_roof_boundaries* matrix (Figure 3c) and by using the binary matrix of selected plane (called *b_plane*) (Figure 4a); the boundaries enveloping this plane can be detected by multiplying the last two matrixes element by element, as shown in Equation 2.

$$\text{boundary_enveloping_plane} = \text{all_roof_boundaries} * \text{b_plane} \quad (2)$$

where « * » is the multiplication element by element

This (Equation 2) generates a new matrix called *boundary_enveloping_plane* (Figure 4b).

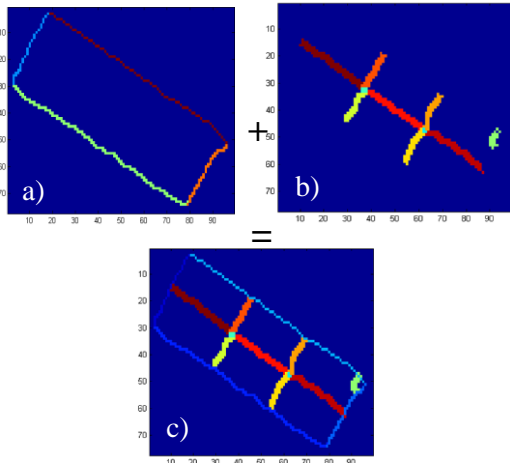


Figure 3. a) Visualisation of the building outer boundaries matrix; b) *raster_inner_boundaries* matrix; c) *all_roof_boundaries* matrix

In the matrix *all_roof_boundaries*, the numeration of the outer building boundaries starts from 1 to *m*; where *m* is the number of outer building boundaries. After that, the numeration of the inner building boundaries starts from *m + 1* to *m + s*; where *s* is the number of inner building boundaries (Figure 4c).

At this stage, the analysis of the junction relationships between boundaries can be started. For this purpose, the junction relationship between boundaries enveloping each plane is calculated. Consequently, *n* junction relations are obtained; where *n* is the number of roof planes. Finally, global analysis of all junction relationships allows determining the boundary which passes through each vertex. The roof vertices are also detected and assigned.

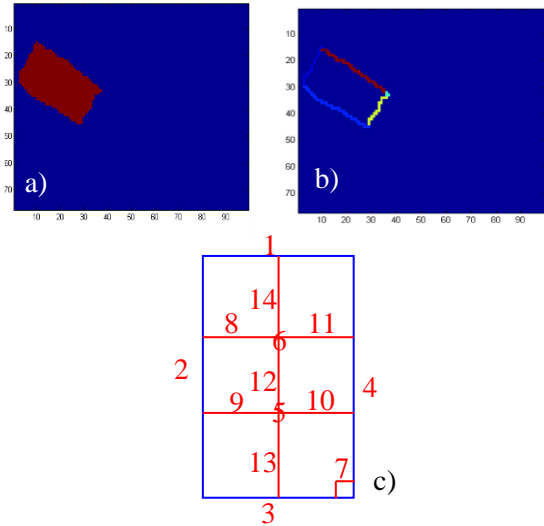


Figure 4. Calculation of *boundary_enveloping_plane* matrix a) *b_plane* matrix of the plane N°2; b) *boundary_enveloping_plane* matrix; c) Numeration of roof boundaries

From the *boundary_enveloping_plane* matrix (Figure 4b) which presents the boundaries enveloping Plane 2 (Figure 4a), the junction relationship between boundaries is deduced as the following: 1, 14, 6, 8, 2. This junction relationship presents the order of the boundaries enveloping Plane 2. In the same way a new junction relationship can be deduced for each roof plane. Moreover, each junction relationship generates one or several triples. For example, the junction relationship of Plane 2 generates the following triples: (1, 14, 6), (14, 6, 8), (6, 8, 2), (8, 2, 1) and (2, 1, 14). Let us take the triple (14, 6, 8), it means that Boundaries 6 and 14 pass through the same vertex; furthermore, the Boundaries 6 and 8 pass also through the same vertex. Once the inner roof boundaries are detected and all roof planes assigned, the detection of roof vertices can be carried out in the following paragraph.

In order to detect the vertex, it is necessary to distinguish between planes having a single adjacent plane nominate “plane of single adjacency” (ex: planes 1, 2 and 3 in Figure 5a) and planes having more than one adjacent planes which called “plane of multiple adjacencies” (ex: planes 1, 2, 3, 4, 5 and 6 in Figure 5b).

The two plane types can be distinguished directly from the plane junction relationship. Indeed, if a plane shares exclusively its boundaries with another plane (for example, Plane 1 in Figure 5a), then the plane is a “single adjacency” (for example, Planes 2 and 3, Figure 5a). In all other cases, the plane is a “multiple adjacencies”. At present, it must similarly distinguish between two types of inner roof plane boundaries. The first one is a boundary that envelops a single adjacency plane and the second one is that envelops a “multiple adjacencies” plane. Since it cannot be said a boundary of single adjacency, therefore this boundary type is called *boundary of type S* and the other

boundary type is called *boundary of type M*. Likewise, when a vertex joints boundaries of type *S*, then it is called *vertex of type S*, and when a vertex joints boundaries of type *M*, then it is called *vertex of type M*.

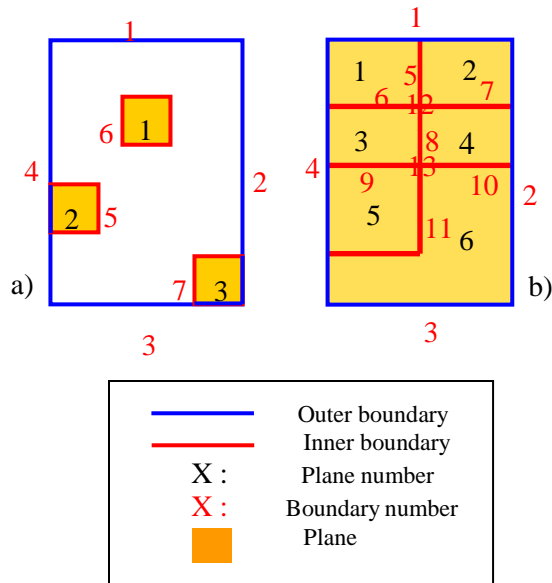


Figure 5. Representation of two roof planes types; a) Representation of single adjacency planes; b) Representation of multiple adjacencies planes

The boundary are detected and distinguished by their types; therefore, the vertices detection can be achieved.

5.4 Automatic detection of the roof vertices

Since, a vertex combines two or more boundaries; the boundary extremities passing through the same vertex must have the same coordinates *X* and *Y*. But it is not the case for the detected boundaries discussed in Section 5.2. So, in order to merge all potential vertices into a single one, it is necessary to detect them firstly. Previously, two vertex types were defined depending on the type of adjacent planes. Both vertex types are the *vertex of type S* and the *vertex of type M*. In the following, methods of automatic detection of each vertex type are exposed.

5.4.1 Automatic detection of vertices of type M: It is shown above the junction relationship of each plane can be presented by a set of triples. Each one is a suite of three boundaries, by consequent, it defines two vertices. For example, the triple (14, 6, 8) presents two vertices: (14, 6) and (6, 8) (Figure 4c).

After detection all roof triples, then it can be noted that one vertex can be presented within several triples. Another example can also be taken, in the Figure 6a, the following vertices: (1, 2), (1, 3), (2, 3) represent the same vertex. Another example is presented in Figure 6b. In this example, the following vertices: (1, 5), (2, 5), (3, 5) and (4, 5) should represent the same vertex.

5.4.2 Automatic detection of vertices of type S: It has been seen that the boundary of type *S*, the junction relationship contains only one inner boundary. In this case, the plane area (in pixel unit) becomes a criterion for decision. Because if the roof detail area is smaller than a given threshold (in pixel unit), then its geometry becomes vague. Moreover, the threshold is taken in pixel unit because it relates to the point density. Indeed, if the plane area is bigger than a given threshold, it is possible to apply Douglas-Peucker algorithm to decompose the plane outline

polygon according to its sides. Otherwise, if the plane area is smaller than a given threshold (in pixel unit), it is necessary to distinguish two kinds of plane of type *S*. The first one is the case of junction relationship consists of one inner boundary only (Plane 1 in Figure 5a). The second kind is the case of junction relationship containing several boundaries but only one among them is inner boundary (Planes 2 and 3 in Figure 5a). This distinction is made because each one of the two plane kinds has a specific processing way.

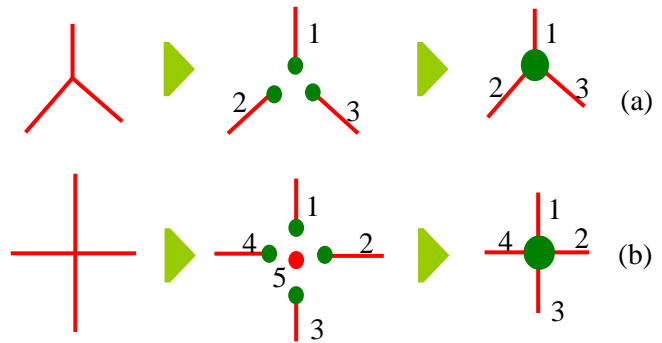


Figure 6. Merging of vertices; a) Three boundary segments pass in the same vertex; b) Five boundary segments pass in the same vertex

- Case of junction relationship consists of one boundary only:

Let us take the example of Plane 1 in Figure 5a. On the one hand, the junction relationship is composed of a single boundary according to boundary definition (a set of pixels describing the same adjacency relationship between two planes), but it has, on the other hand, four vertices (the plane outline polygon corners). In this case, geometric constraints are introduced by assuming the geometric plane form is rectangular (because its geometry in building label image is vague), which is the most frequent case. So the coordinates of the four rectangle corners (vertices) are calculated by using the modelling method based on static moments. All details and equations of this method are given at (Tarsha Kurdi et al., 2007) in the context of parametric modelling approach of building footprint.

- Case of junction relationship containing several boundaries:

Taking the example of the Planes 2 and 3 in Figure 5a, two different situations arise: either the plane is attached to a single outer boundary (Plane 2), or the plane is adjacent to two outer boundaries (Plane 3). In the first case, the plane is defined by three inner boundaries and four vertices. On the other case, while the plane is adjacent to two outer boundaries, it is defined by two inner boundaries and three vertices. Vertex calculation follows a process that studies the angle formed by two successive outer boundaries of the considered plane. This hypothesis can be explained by the fact that it is possible to meet a plane adjacent to two outer boundaries but the angle between the two outer boundaries is close to 180° (Figure 7c).

Despite the plane is attached to two outer boundaries but its model likes the case of plane attached to a single outer boundary. In order to distinguish between the two cases, the farthest pixel, among the plane circumference ones, from the plane gravity centre has to be detected. This point and the plane gravity centre define a circle radius. The centre of this circle is the plane gravity centre. Then, we calculate the intersection of this circle with the outer boundaries that surrounding the plane (points C and D in Figure 7c).

Then the angle β is calculated from the centre of gravity and the two last points C and D (Figure 7). If this angle β is close to 180°

(Figure 7a), then the plane is considered as attached to two outer boundaries, else the plane is taken as attached to one outer boundary (Figure 7b and 7c). Finally, two perpendicular lines are drawn from the intersection points (C and D). If our case corresponds to Figure 7a, then the intersection of the two perpendicular lines is calculated. Otherwise, the intersection points of the two perpendicular lines with the circle are calculated.

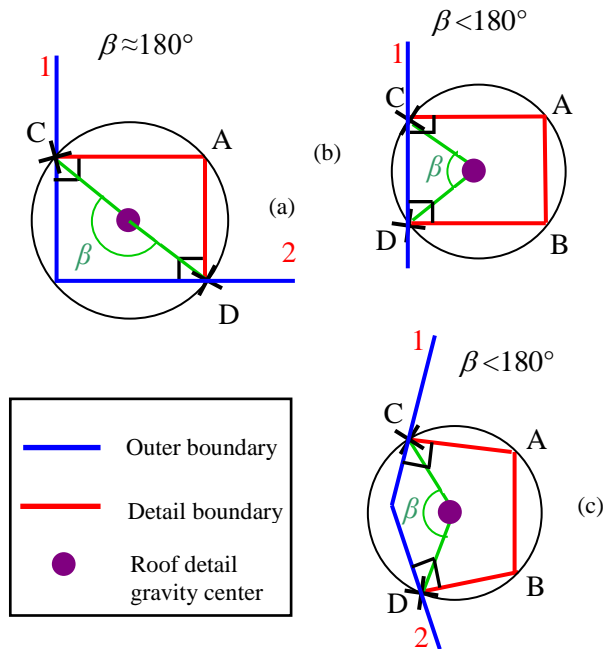


Figure 7. Modelling of roof planes of type *S* attached to building outer boundaries; a) Plane attached to two outer boundaries; b) Plane attached to a single outer boundary; c) Plane attached to two outer boundaries but it is treated as the case of Figure 7b

Once the inner roof plane boundaries modelled in 2D, the technique Douglas-Peucker is applied on each boundary. This procedure allows decomposing the inner boundary segments according to the lines. At this stage, the fundamental elements, which are necessary for calculating the 2D building roof model, are available.

6. RESULTS

Figure 8 presents the final result of automatic 2D building modelling for Hermanni site (point density= 7 points/m²). Moreover, three colours are used and three buildings are zoomed in Figure 9 for illustrating the model deformations. The blue lines represent the building outer boundaries, the red represent the inner roof plane boundaries and the green represent the roof details boundaries.

It has to be noted that some inner boundaries are also drawn in green; this is explained simply by the fact that the green boundaries does not cross another inner boundary. Therefore, the construction algorithm considers this plane as a roof detail (see Buildings 1, 4, 5, 9, 10 and 11 in Figures 8 and 9). But that will not effect on the final 2D model. Finally, the analysis qualitative and the accuracy quantification of this result are carried out in the following.

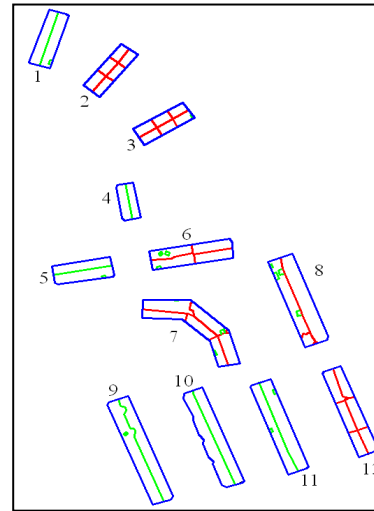


Figure 8. Final result of automatic 2D building modelling; Hermanni site; in blue: the building outlines; in red: the inner roof boundaries; in green: the roof details

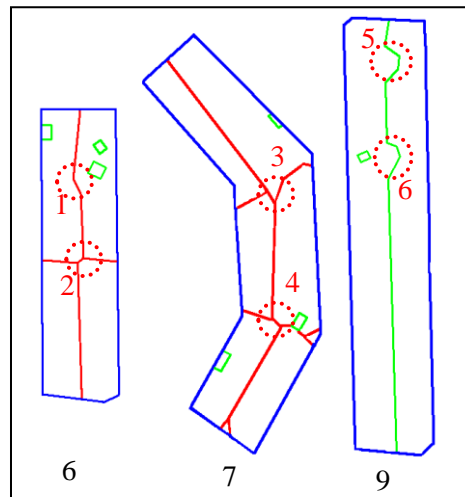


Figure 9. Final result of automatic 2D building modelling; Three selected buildings (Buildings 6, 7 and 9); in blue: building outlines; in red: inner roof boundaries; in green: roof details

7. ACCURACY ANALYSIS

In the context of visual deformations of building roof model, according to the point density and the roof detail dimensions, it can be distinguished between two roof detail types: detectable roof details and non-detectable ones. Indeed, the presence of non-detectable details in the cloud generates deformations in 2D building model as already seen in Figure 9 (red circles). Deformations encountered here appear as reflection of presence of the indiscernible details on the building roof. While they do not come from geometric uncertainty. The arising result at this stage confirms that these deformations become eventually very beneficial because they indicate the presence of indiscernible details in the original building.

The method developed by (Tarsha Kurdi, 2008) is applied to analyse the deformations and the accuracy of building models “pixel by pixel”. It is based on the calculation of distance (deviation) between each point and its mean plane. Consequently, a new matrix called “error map” is calculated (Figure 10).

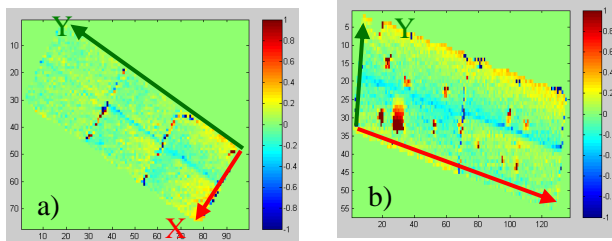


Figure 10. Error map visualisation; a) and b) Buildings 3 and 6

The pixel values in this matrix are equal to the deviation ones. To visualise the error map matrix representatively, it has to replace the variances bigger than 1 m by 1 m, and the variance smaller than -1 m by -1 m. Hence, all pixel values in this matrix are between 1 and -1. The (Figures 10a and 10b) illustrate the error maps of Buildings 3 and 6.

From Figure 10, it can be observed that the building error map expresses the 2D building model accuracy. For example, if pixel value is equal or close to zero, the model in this position is precise and vice versa. Furthermore, the non-detectable roof details appear clearly in building error map. Therefore, the error map is used for estimating the building model accuracy. For this purpose, the deviation of distribution has to be analysed.

For example, from Table 3 it can be noted that more than 90% of building points (Buildings 6 and 3) have an acceptable deviation regarding the accuracy of point positions. The application of the same test for all Hermanni site buildings shows also the same result. This result explains the height quality of obtained building models.

Building Number	Deviation values (m)			Point percentage (%)
	-1 To -0.25	-0.25 To 0.25	0.25 To 1	
6	4	90	6	
3	2	95	3	

Table 3. Analysis of the deviation distribution

The point cloud of building presented in (Figure 11) has a low point density (1.3 point/ m²). From error map and building aerial image (Figures 11a and 11d) it can be noted that there are a lot of no detectable details on building roof. Hence, that can be explained by the weak point density and architectural building complexity. Moreover, these no detectable details can be seen clearly on error map. Therefore, error map can be used to localize the no detectable details and to reduce their negative effects on the 2D model (visual deformations). It can also be used to model these roof details.

Generally, the proposed approach provides good results. Nevertheless, more investigation is necessary to calculate the 2D building model more faithful to reality. And by consequent, calculate accurate 3D building model.

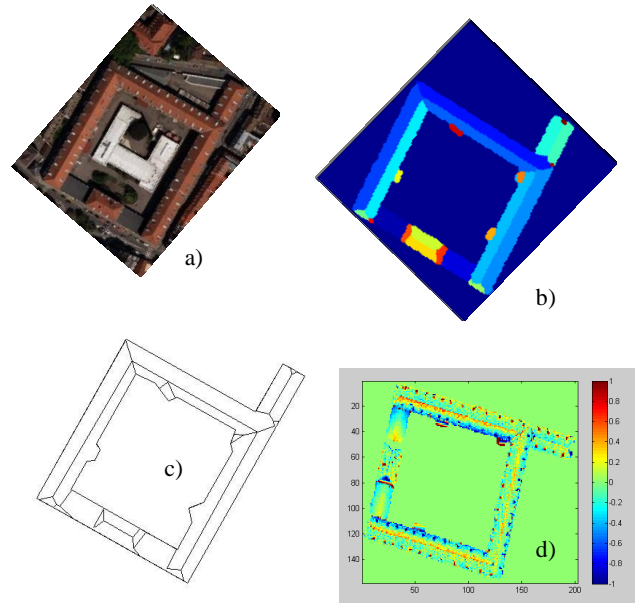


Figure 11. a) Building aerial image; b) Building label image; c) 2D roof model; d) Error map visualisation; this building is in Strasbourg city.

8. CONCLUSION AND PERSPECTIVE

This paper presented a full workflow of automatic 2D inner roof boundaries modelling from Lidar data exclusively. The generalisation level of the constructed model is related directly to five factors: the point density, the cloud accuracy, the cloud homogeneity, the presence of no detectable details and the architectural building complexity. At this stage, the error map can play an important role for judging if the total 2D building model is reliable or not.

However, several adjustments were achieved to improve the quality of the 2D building model. In a future research, this model can be refined by introducing certain geometric constraints, such as parallelism or orthogonality. These improvements will make the building model closer to reality. The visual deformations were obviously harmful, but they also represent a source of valuable information. Therefore farther investigations will allow locating the no detectable roof details. Then, it will be possible as the first step to eliminate their negative influence from the calculated building model. In the second step the no detectable details can be modelled.

9. REFERENCES

- Ameri, B., 2000. Automatic recognition and 3D reconstruction of buildings from digital imagery, PhD thesis, German Geodetic Commission (DGK)-C 526, Institute for Photogrammetry, Stuttgart University.
- Awrangjeb, M., 2016: Using point cloud data to identify, trace and regularize the outlines of buildings. *International Journal Of Remote Sensing*, VOL. 37, NO. 3, 551–579, doi:10.1080/01431161.2015.1131868.
- Briese, C., 2004. Three-Dimensional Modelling of Breaklines from Airborne Laser Scanner Data. *Int. Arch. of the Photogrammetry and Remote Sensing*, XXth ISPRS Congress: Geo-imagery bridging continents. Istanbul, Turkey.

- Douglas, D.H., Peucker, T.K., 1973: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer*, 10 (2), 112–122. 1973.
- Ghaffarian, S., Ghaffarian, S., Elmerabet, Y., Samir, Z., Ruichek, Y., 2016. Automatic building roof segmentation based on PFICA algorithm and morphological filtering from Lidar point clouds. In *37th Asian Conference on Remote Sensing (ACRS 2016)*, promoting spatial data infrastructure for sustainable economic development, 17-21 October 2016, Colombo, Sri Lanka.
- Gross, H., Thoennessen, U., 2006. Extraction of lines from laser point cloud. *Int. Arch. of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Proceedings of the ISPRS Commission III Symposium on Photogrammetric and Computer Vision, Volume XXXVI, Bonn, Germany, 2006, pp. 86–91.
- Jaya, S.N., Akshay, J., Beena, K., 2018: Contour Extraction in Buildings in Airborne Lidar Point Clouds Using Multiscale Local Geometric Descriptors and Visual Analytics. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, PP. 1-16. 10.1109/JSTARS.2018.2833801.
- Perera, S.N., Nalani, H.A., Maas, H.G., 2012. An automated method for 3D roof outline generation and regularization in airborne laser scanner data. *Int. Arch. of the Photogrammetry, Remote Sensing, ISPRS Ann. Photogrammetry. Remote Sens. Spatial Inform. Sci.* I-3, 281–286.
- Rottensteiner, F., Trinder, J., Clode, S., Kubik, K., 2005. Automated delineation of roof planes from Lidar data. *Int. Arch. of the Photogrammetry, Remote Sensing, ISPRS WG III/3, III/4, V/3 Workshop "Laser scanning 2005"*, Enschede, Netherlands, September 12-14, 2005. pp. 221-226.
- Rottensteiner, F., 2003: Automatic generation of high-quality building models from Lidar data. *IEEE CG&A*, 23(6), pp. 42-51.
- Sampath, A., Shan, J., 2006. Clustering based planar roof extraction from Lidar data. *American Society for Photogrammetry and Remote Sensing, Annual Conference*, Reno, Nevada, May 1–6.
- Shan, J., Toth, C.K., 2008: *Topographic Laser Ranging and Scanning Principles and Processing*. By Taylor & Francis Group, LLC. ISBN-13: 978-1-4200-5142-1 (Hardcover), 593 P.
- Schenk, T., Csatho, B., 2002. Fusion of Lidar data and aerial imagery for a more complete surface description. *Int. Arch. of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Proceedings of the ISPRS Commission III Symposium on Photogrammetric and Computer Vision, Volume XXXIV, Graz, Austria, 2002, pp. 310–317.
- Sohn, G., Dowman, I., 2007: Data fusion of high-resolution satellite imagery and Lidar data for automatic building extraction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 62(1), May, 43–63.
- Sohn, G., Huang, X., Tao, V., 2008: Using a binary space partitioning tree for reconstructing polyhedral building models from airborne Lidar data. *Photogrammetric Engineering and Remote Sensing*, 1436 November 2008.
- Tarsha Kurdi, F., Landes, T., Grussenmeyer, P., Koehl, M., 2007. Model-driven and data-driven approaches using Lidar data: analysis and comparison. *Int. Arch. of the Photogrammetry, Remote Sensing, PIA07, Photogrammetry Image Analysis*. 19 – 21 September 2007, Munich, Germany.
- Tarsha Kurdi, F., 2008. 3D building extraction and modelling from Lidar data. PhD grade at Photogrammetry and Geomatics Group MAP-PAGE UMR 694, Graduate School of Science and Technology of Strasbourg (INSA of Strasbourg) (ULP University).
- Tarsha Kurdi, F., Landes, T., Grussenmeyer, P., 2008: Extended RANSAC algorithm for automatic detection of building roof planes from Lidar data. *The Photogrammetric Journal of Finland*, vol. 21, n°1, 2008, pp. 97-109.
- Vosselman, G., Dijkman, S., 2001. 3D building model reconstruction from point clouds and ground plans *Int. Arch. of the Photogrammetry, Remote Sensing*, XXXIV-3/W4, pp. 37-43.
- Xiao, Y., Wang, C., Li, J., Zhang, W., Xi, X., Wang, C., Dong, P., 2015: Building segmentation and modelling from airborne Lidar data, *International Journal of Digital Earth*, 8:9, 694-709, DOI: 10.1080/17538947.2014.914252.