



# Hough-Transform and Extended RANSAC Algorithms for Automatic Detection of 3D Building Roof Planes from Lidar Data

Fayez Tarsha-Kurdi, Tania Landes, Pierre Grussenmeyer

## ► To cite this version:

Fayez Tarsha-Kurdi, Tania Landes, Pierre Grussenmeyer. Hough-Transform and Extended RANSAC Algorithms for Automatic Detection of 3D Building Roof Planes from Lidar Data. ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007, Sep 2007, Espoo, Finland. pp.407-412. halshs-00264843

HAL Id: halshs-00264843

<https://halshs.archives-ouvertes.fr/halshs-00264843>

Submitted on 19 May 2008

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# HOUGH-TRANSFORM AND EXTENDED RANSAC ALGORITHMS FOR AUTOMATIC DETECTION OF 3D BUILDING ROOF PLANES FROM LIDAR DATA

F. Tarsha-Kurdi\*, T. Landes, P. Grussenmeyer

Photogrammetry and Geomatics Group MAP-PAGE UMR 694, Graduate School of Science and Technology (INSA), 24 Boulevard de la Victoire, 67084 STRASBOURG, France.  
(fayez.tarshakurdi|tania.landes|pierre.grussenmeyer@insa-strasbourg.fr)

**Commission III, WG III/4**

**KEY WORDS:** LIDAR, Processing, Detection, Extraction, Building, Segmentation, Modelling

## ABSTRACT:

Airborne laser scanner technique is broadly the most appropriate way to acquire rapidly and with high density 3D data over a city. Once the 3D Lidar data are available, the next task is the automatic data processing, with major aim to construct 3D building models. Among the numerous automatic reconstruction methods, the techniques allowing the detection of 3D building roof planes are of crucial importance. Three main methods arise from the literature: region growing, Hough-transform and Random Sample Consensus (RANSAC) paradigm. Since region growing algorithms are sometimes not very transparent and not homogeneously applied, this paper focuses only on the Hough-transform and the RANSAC algorithm. Their principles, their pseudocode - rarely detailed in the related literature - as well as their complete analyses are presented in this paper. An analytic comparison of both algorithms, in terms of processing time and sensitivity to cloud characteristics, shows that despite the limitation encountered in both methods, RANSAC algorithm is still more efficient than the first one. Under other advantages, its processing time is negligible even when the input data size is very large. On the other hand, Hough-transform is very sensitive to the segmentation parameters values. Therefore, RANSAC algorithm has been chosen and 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. So the proposed extension allows harmonizing the mathematical aspect of the algorithm with the geometry of a roof. At last, it is shown that the extended approach provides very satisfying results, even in the case of very weak point density and for different levels of building complexity. Therefore, once the roof planes are successfully detected, the automatic building modelling can be carried out.

## 1. INTRODUCTION

The quick acquisition of 3D data as well as the automatic data processing are two key-tasks for the majority of surveying fields. Airborne laser scanning systems generate 3D data with high speed, good accuracy and density. Thus, the use of this technique in urban region is more and more frequent.

In order to construct automatically 3D city models, two successive steps have to be considered. The first one is the automatic segmentation of the point cloud into three classes which are terrain, vegetation and buildings. Once the city cloud is segmented, the modelling of buildings can start. Two types of approach called *model-driven* and *data-driven* approaches in the literature are proposed for constructing building models. The model-driven approaches search the most appropriate model among primitive building models contained in a model library (Maas and Vosselman, 1999). They consider that a primitive building can be described by a set of parameters. That implies to calculate the values of the parameters before constructing the 3D model. On the other hand, data-driven approaches try to simulate each part of the building point cloud for obtaining the nearest or the more faithful polyhedral model (Rottensteiner, 2003).

In the context of data-driven approaches which provide more universal models, the automatic detection of planes is a crucial operation. Many methods are proposed in order to carry out this procedure such as region growing, 3D Hough-transform and RANSAC. Only the two last techniques are studied in this paper since region growing algorithm are sometimes not very transparent and not homogeneously applied. Furthermore, the principles and the pseudocodes of 3D Hough-transform and

RANSAC algorithms are detailed and compared. In order to clarify their operating mode and assess them, they are applied on samples of buildings with different forms and complexity levels. At last, the RANSAC algorithm is extended to be able to solve the majority of building cases.

## 2. 3D HOUGH-TRANSFORM

### 2.1 Related works and principle

The 2D Hough-transform technique (Hough, 1962) is normally used in the field of digital image processing in order to detect geometric primitives. Many applications in this field as well as its algorithm are presented by (Davies, 1988; Gonzalez *et al.*, 2004; Nguyen *et al.*, 2005). This technique is used to detect the straight lines like building contour polygons, and curves such as circles and ellipses. With the 3D point cloud, the demand is increased for detecting 3D planes. In this context, the 2D Hough-transform has been extended to 3D (Vosselman and Dijkman, 2001; Oda *et al.*, 2004; Overby *et al.*, 2004). Later, its principle has been extended to the extraction of other 3D geometric forms like cylinders (Rabbani and Van den Heuvel, 2005).

The principle of the 2D Hough-transform is the representation of a points set, defined initially in the Euclidian space, in another space. This transform allows detecting the points composing specific geometric primitives. For example, in (OXY) space, the equation of a line has the form (1).

$$Y = a \cdot X + b \quad (1)$$

where  $(a, b)$  are the line parameters.

This line can be represented by a point with coordinates  $(a, b)$  in the parameter space  $(O' a b)$ . In an opposite way, one point  $(X_i, Y_i)$  belonging to the space  $(OXY)$  is represented by a line in the parameter space  $(O' a b)$  as expressed in Equation 2.

$$b = -X_i \cdot a + Y_i \quad (2)$$

where  $(X_i, Y_i)$  are the parameters of this line.

Supposing that  $M_1, M_2, \dots M_n$  are a set of points in the space  $(OXY)$  and that they belong to the line  $P$  following Equation 1. Each one of these points represents a line in the parameter space. The intersection of these lines in the parameter space is the point  $(a_1, b_1)$  which represents the parameters of the line  $P$  in a 2D-space.

If the line equation has the form  $X = \text{constant}$ , then it can not be presented in the parameter space  $(O' a b)$ , because the Y-axis coefficient is equal to zero. In order to solve this problem, it is suggested to use the normal form of the line (Equation 3).

$$\cos \theta \cdot X + \sin \theta \cdot Y = \rho \quad (3)$$

where  $\theta$  and  $\rho$  are the parameters of the normal passing through the origin (see Fig.1).

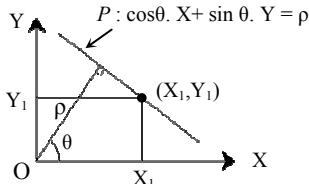


Figure 1. Presentation of one line and its normal in a 2D-space

So,  $\theta$  and  $\rho$  are constant for one line. The parameter space in this case is  $(O' \theta \rho)$ . Hence, one point  $(X_1, Y_1)$  in the 2D-space represents a sinusoid in the parameter space (see Fig.2).

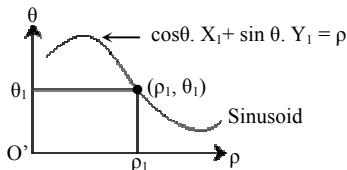


Figure 2. Presentation of a point in the parameter space using the normal form

The same principle can be applied in a 3D case in considering that one plane belonging to the  $(OXYZ)$  space (Equation 4) can be represented by a point  $(a, b, c)$  in the parameter space  $(O'abc)$ .

$$Z = a \cdot X + b \cdot Y + c \quad (4)$$

In the same manner, if the plane equation has the form (5), then it can not be presented in the parameter space because the Z-axis coefficient is equal to zero. In order to solve this problem, (Overby *et al.*, 2004) suggest to use also the normal form of the plane (Equation 6).

$$a \cdot X + b \cdot Y + c = 0 \quad (5)$$

$$\cos \theta \cdot \cos \varphi \cdot X + \sin \theta \cdot \cos \varphi \cdot Y + \sin \varphi \cdot Z = \rho \quad (6)$$

where  $\theta, \varphi$  and  $\rho$  are the parameters of the plane normal passing through the origin (see Fig.3).

So,  $\theta, \varphi$  and  $\rho$  are constant and the parameter space is  $(O' \theta \varphi \rho)$ . In this case, one point  $(X_1, Y_1, Z_1)$  in the 3D-space represents a sinusoidal surface in the parameter space. Since the principles of the 3D Hough-transform are explained, the aim of the next section is to deliver its algorithm.

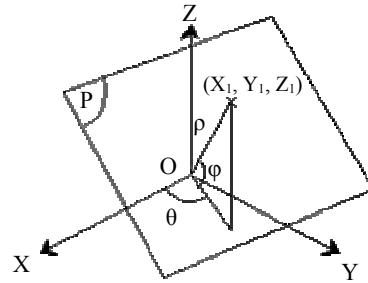


Figure 3. Representation of plane equation elements in the normal form

## 2.2 3D Hough-transform algorithm

The input data are the steps on  $\theta, \varphi$  and  $\rho$  axis (discrete intervals), called  $\theta\_step$ ,  $\varphi\_step$  and  $\rho\_step$  respectively. The 3D point cloud is represented by three coordinate lists  $X$ ,  $Y$  and  $Z$ . Algorithm 1 presents the pseudocode of the 3D Hough-transform.

---

### Algorithm 1: 3D Hough-transform for plane detection

```

1.  $X_{\min} = \min(X); Y_{\min} = \min(Y); Z_{\min} = \min(Z)$ 
2.  $X_{\max} = \max(X); Y_{\max} = \max(Y); Z_{\max} = \max(Z)$ 
3. Calculation of:  $Dis_{\min}; Dis_{\max}$ 
4.  $\theta = \text{from } 0 \text{ to } 360, step = \theta\_step; n_{\theta} = \text{length}(\theta)$ 
5.  $\varphi = \text{from } -90 \text{ to } +90, step = \varphi\_step; n_{\varphi} = \text{length}(\varphi)$ 
6.  $n_{\rho} = 2 * (Dis_{\max} - Dis_{\min}) / \rho\_step$ 
7.  $\rho = \text{from } Dis_{\min} \text{ to } Dis_{\max}; step = \rho\_step$ 
8.  $\theta_{\text{mat}}(n_{\varphi}, n_{\theta}) = [\theta \ \theta \ \theta \dots \theta]^* \pi / 180$ 
9.  $\varphi_{\text{mat}}(n_{\varphi}, n_{\theta}) = [\varphi \ \varphi \ \varphi \dots \varphi]^* \pi / 180$ 
10.  $H(n_{\theta}, n_{\varphi}, n_{\rho}) = 0$ 
11.  $ratio = (n_{\rho} - 1) / (\rho(n_{\rho}) - \rho(1))$ 
12. for  $k = 1$  to  $\text{length}(X)$ 
13.  $\rho_{\text{mat}} = \cos(\varphi_{\text{mat}}) * \cos(\theta_{\text{mat}}) * X(k) + \dots$ 
    $\cos(\varphi_{\text{mat}}) * \sin(\theta_{\text{mat}}) * Y(k) + \sin(\varphi_{\text{mat}}) * Z(k)$ 
14.  $\rho_{\text{index}} = \text{round}(ratio * (\rho_{\text{mat}} - \rho(1) + 1))$ 
15. for  $i = 1$  to  $n_{\varphi}$ 
16. for  $j = 1$  to  $n_{\theta}$ 
17.  $H(j, i, \rho_{\text{index}}(i, j)) = H(j, i, \rho_{\text{index}}(i, j)) + 1$ 
18. next  $j$ ; next  $i$ ; next  $k$ 
```

---

In this algorithm,  $Dis_{\min}$  and  $Dis_{\max}$  are the distances between the origin and the two extremities of the cloud points calculated at lines 1 and 2;  $H$  is a 3D matrix;  $\theta_{\text{mat}}$ ,  $\varphi_{\text{mat}}$  and  $\rho_{\text{mat}}$  are 2D matrices;  $\theta, \varphi$  and  $\rho$  are three lists.

The result of the algorithm is the 3D matrix  $H$  which contains the representation of the original cloud in the parameter space. Each point of  $(OXYZ)$  space gives a sinusoidal surface in the parameter space.

Fig.-4a shows the visualization of one horizontal plane in the 3D matrix  $H$ . Fig.-4b shows the result of the roof planes detection. For improving this result, it is necessary to use parameter values as small as possible. But, in this case the processing time and the needed memory will be much higher. The sample used for this figure is a building whose characteristics are detailed in section 3.3.

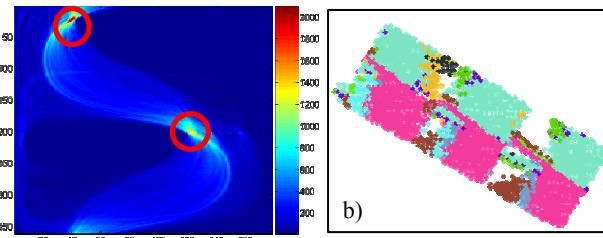


Figure 4. a) Horizontal plane in the 3D matrix  $H$ , (with plane number  $p = 75$ ); b) Roof plane detection result using the 3D Hough-transform.

The next step consists in detecting the peaks from the 3D matrix  $H$  as marked by red circles in Fig.-4a. Each peak represents one plane in (OXYZ) space. This operation can be performed by searching voxels having the maximum values in the matrix  $H$ ) and then applying a 3D region growing algorithm.

### 2.3 Analysis

The 3D Hough-transform uses a pure mathematical principle in order to detect the best planes from a 3D point cloud. That means that it looks for point sets which represent statistically the best planes without taking into account their signification in the building point cloud. In this context, the *best plane* does not mean the most probable plane calculated according to the least squares theory. But it means the plane containing the maximum number of points. Therefore, it detects perhaps a set of points which represents several roof planes or which belongs to several planes.

Moreover, the 3D Hough-transform spends a long time for calculating the matrix  $H$  and for detecting the peaks in it. Furthermore, the application of 3D Hough-transform requires the use of four parameters. The first three one are the steps (discrete intervals) on  $\theta$ ,  $\phi$  and  $\rho$  axis. When the used step values are small, the quality of the detected plane is improved, but the processing time and the needed memory are much higher and vice versa. The fourth parameter is a threshold entering in the 3D region growing algorithm. It represents the difference between the voxel value and its neighbours. The determination of the four threshold values are related to the characteristics of the point cloud and of the building roof planes. Thus, it is very difficult to determine them automatically.

## 3. RANSAC ALGORITHM FOR PLANE DETECTION

### 3.1 Related works and principle

In the digital image processing domain, RANDOM SAmples Consensus (RANSAC) algorithm is used to detect mathematical features like straight lines and circles. Its principle is well explained by (Fischler and Bolles, 1981; McGlone McGlone *et al.*, 2004; Nguyen *et al.*, 2005). In the field of automatic buildings modelling based on Lidar data, many authors suggest its use for achieving different tasks. For example, (Ameri and Fritsch, 2000; Brenner, 2000) use RANSAC algorithm for detecting the building roof planes. (Forlani *et al.*, 2004; Forlani *et al.*, 2006) apply RANSAC algorithm in order to correct the building roof segmentation result which are obtained using a partition in 8 classes of the gradient orientation. Moreover, to carry out the 2D segmentation of the building contour polygon pixels in straight lines, the same technique is also applied. (Bretar and Roux, 2005) use the Normal Driven RANSAC

(ND-RANSAC) for extracting 3D planar primitives. For this purpose, they calculate the normal vectors for each point. Then, they select randomly three points but having the same orientation of normal vectors. In our case, RANSAC algorithm is used with the aim of roof planes detection.

The principle of RANSAC algorithm consists to search the best plane among a 3D point cloud. In the same time, it reduces the number of iterations, even if the number of points is very large. For this purpose, it selects randomly three points and it calculates the parameters of the corresponding plane. Then it detects all points of the original cloud belonging to the calculated plane, according to a given threshold. Afterwards, it repeats these procedures  $N$  times; in each one, it compares the obtained result with the last saved one. If the new result is better, then it replaces the saved result by the new one.

### 3.2 RANSAC algorithm

This algorithm needs four input data which are:

- The 3D point cloud (*point\_list*) which is a matrix of three coordinate columns X, Y and Z;
- The tolerance threshold of distance  $t$  between the chosen plane and the other points. Its value is related to the altimetric accuracy of the point cloud;
- The *forseeable\_support* is the maximum probable number of points belonging to the same plane. It is deduced from the point density and the maximum foreseeable roof plane surface.
- The probability  $\alpha$  is a minimum probability of finding at least one good set of observations in  $N$  trials. It lies usually between 0.90 and 0.99.

Algorithm 2 details the pseudocode of RANSAC algorithm.

---

#### Algorithm 2: RANSAC for plane detection

---

```

1. bestSupport = 0; bestPlane(3,1) = [0, 0, 0]
2. bestStd = ∞; i = 0
3. ε = 1 - forseeable_support/length(point_list)
4. N=round (log (1 - α)/log (1 - (1 - ε)^3))
5. while (i <= N)
6. j = pick 3 points randomly among (point_list)
7. pl = pts2plane(j)
8. dis = dist2plan(pl, point_list)
9. s = find(abs(dis)<= t)
10. st = Standard_deviations (s)
11. if (length(s) > bestSupport or (length(s) = ...
    bestSupport and st < bestStd)) then
12. bestSupport = length (s)
13. bestPlan = pl; bestStd = st; endif
14. i = i+1; endwhile

```

---

In this pseudocode,  $\epsilon$  is a percentage of observations allowed to be erroneous; the function *pts2plane* calculates the plane parameters from three chosen points. It is advised to use the normal form of the plane instead of the classical form (see Equation 6) in order to consider the general expression of a plane; the function *dist2plan* calculates the signed distances between point set and given plane (the distance takes negative or positive value) as given in Equation 7.

$$\text{dist2plan} = \cos \theta \cos \varphi X + \sin \theta \cos \varphi Y + \sin \varphi Z - \rho \quad (7)$$

where X, Y and Z are the three columns of the matrix *point\_list*;  $\theta$ ,  $\varphi$  and  $\rho$  are the plane parameters (see Equation 6).

It is important to note that the number of trials  $N$  can be considered directly as an input of the algorithm, instead of calculating it by a pure probability law. For this purpose, a table of different urban typologies and point densities can suggest the  $N$  value. Therefore, it replaces the introduction of values for *forseeable\_support* and  $\alpha$ . This operation is one of the modifications proposed for improving the basic RANSAC algorithm.

In order to detect the whole roof planes, the algorithm is applied several successive times. In each iteration, the set of considered points is excluded from the original cloud. This operation is repeated until the number of non-modelled points becomes smaller than a given threshold.

### 3.3 Comparison and quantitative analysis

In order to assess the capacities of the algorithm, two samples of buildings are used. They contain buildings of different forms and complexity levels. Only some results are illustrated in this paper, but they are based on characteristic samples (covering simple as well as complex building types) and consider low and high point densities. The first sample contains 12 buildings and its point density is equal to 7 points/m<sup>2</sup>. The second sample contains 46 buildings, with a point density of 1.3 points/m<sup>2</sup>.

Fig.5 presents the results of roof planes detection using RANSAC algorithm.

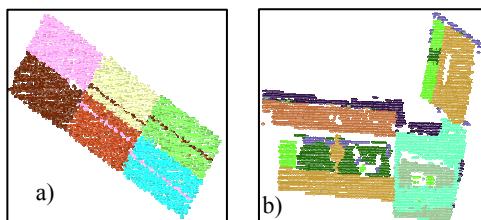


Figure 5. Visualisation of the 2D point clouds resulting from roof planes detection using RANSAC.  
a) Point density: 7 pts/m<sup>2</sup>. b) Point density: 1.3 pts/m<sup>2</sup>.

The colours represent the different building roof planes.

The application of classic RANSAC algorithm on these samples gives successful results in 70% of cases for different building forms and different point cloud densities. It means that it detects correctly the roof planes for 41 buildings. For example, Fig.-5a illustrates very good plane detections, whereas Fig.-5b shows unsatisfying results. In extreme situations, the algorithm can provide unacceptable errors (see Fig.-7b). That can be explained by the use of a pure mathematical principle, without taking into account the particularity of the building Lidar data. The same remark has already been made for the 3D Hough-transform in section 2.3. That is why, it may detect a set of points which represents several roof planes or which belongs to several planes. Therefore, the classic algorithm needs to be adapted in order to detect the best roof planes instead of the best mathematical planes in a 3D point cloud.

(McGlone *et al.*, 2004) note that the RANSAC algorithm aims at significantly reducing the number of necessary trials for large  $N$  values. However, it reduces  $N$  at the expense of having no guarantee for a solution free of gross errors. That means that there is not any guarantee for obtaining the same result after each iteration.

Afterwards, several experiments have been made on the point cloud including the 41 buildings, i.e. the cloud for which RANSAC generated successful results. They demonstrate that the iterative application of RANSAC algorithm gives the same

set of roof planes, but in a different order. Since the plane order is not important here, the RANSAC algorithm can be considered as an algorithm which guarantees a successful result. Furthermore, the processing time, even in the case of a large point cloud, is negligible in comparison with the processing time required by 3D Hough-transform.

It is important to note that the segmentation quality could be actually evaluated only after the stage of 3D modelling. Moreover, the quantitative comparison between the results of 3D Hough-transform and RANSAC algorithms will not be correct. Indeed, the Hough-transform results are related to three aspects: the segmentation quality, the processing time and the needed memory, while the last two aspects are negligible for RANSAC algorithm.

### 3.4 3D Hough-transform or RANSAC algorithm?

As evoked in previous sections, RANSAC algorithm provides not only results in a shorter time but also results of higher quality with a large percentage of successful results in comparison with 3D Hough-transform. This assertion is made after several experiments carried out on the same data for both algorithms. For example, the result of 3D Hough-transform mentioned in Fig.-4b is based on the same building as those used with RANSAC and presented in Fig.-5a. Therefore it is chosen in our approach leading to detect automatically building roof planes using Lidar data. Therefore, in the next paragraph, the RANSAC algorithm is extended in order to increase the percentage of successful plane detection from 70% to more than 95%.

## 4. EXTENSION OF RANSAC ALGORITHM

Two directions are proposed for extending the capacities of RANSAC algorithm to a better roof plane detection. The first one is the improvement of the data quality; the second one is the adaptation of RANSAC algorithm to roof detection.

### 4.1 Improvement of data quality

It is well known that the point cloud coordinates contain errors related to position accuracy, artefacts, and multi ways. Moreover, noise and the small details composing building roofs are considered as obstacles. At last, variable point densities may occur for the same building roof. So, irregular distribution of points on a building roof is also a cause of errors in the calculated plane. All these reasons allow thinking about the necessity of improving the quality of the point cloud.

This remark leads to generate a new point cloud. On the one hand, the new cloud should present a homogeneous point density, and on the other hand, the errors of point coordinates and the noise should be eliminated or decreased.

For this purpose, a resampling of the building point cloud is performed firstly. The sampling value defining the generated DSM is deduced from the average point density (Tarsha-kurdi *et al.*, 2007); and then values are assigned to the DSM cells. In the latter operation, the original cloud is superimposed on the DSM grid. Hence, some cells are empty and other cells contain one or more points. In the case of a non empty cell, the corresponding DSM pixel takes the maximum of the Z values occurring among the points. In the case of an empty cell in the building body, the corresponding DSM pixel value takes the mean of the non null neighbouring pixels. On the one hand, this operation allows eliminating a high quantity of points describing the facades. On the other hand, it allows filling the

empty pixels, while respecting the mathematical characteristic of the plane.

Secondly, in order to decrease the errors of point coordinates and the noise, a simple low-pass filter is applied. The last step consists in converting the generated DSM into a 3D cloud. The analysis of the new point cloud shows that the position accuracy of the inner roof plane boundaries has decreased. This has to be related to the low-pass filtering. Hence, the new cloud is used exclusively for detecting the roof planes, but not for the future building modelling operations where the return to the original point cloud is inevitable.

#### 4.2 Adaptation of RANSAC algorithm

The second enhancement consists in adapting RANSAC algorithm, in order to adapt the mathematical aspect of the algorithm with the geometry of a roof. Indeed, RANSAC algorithm searches to detect the best mathematical plane in a building cloud, regardless if the detected plane represents a roof plane or another plane.

The adaptation of RANSAC algorithm consists of improving its pseudocode, and of using additional procedures for improving the quality of the detected planes.

**4.2.1 Improvement of RANSAC algorithm:** In section 3.2, the 11th line is the essential line in the algorithm, because it represents a gate which allows to accept or to refuse the calculated plane. Indeed, the used condition is the number of points belonging to the calculated plane. Then the algorithm accepts the new plane if it contains more points than the last calculated one, else the new plane will be refused.

After the experiments, it was found that the best condition for validating plane detection is to take into account not only the number of points, but also simultaneously the standard deviation. Indeed, the use of standard deviation decreases the negative influence of the distance tolerance threshold  $t$ . As already mentioned, this threshold allows accepting whole points having distances to the plane smaller than  $t$ .

For example, let us take a “bad” plane which does not represent a roof plane, with a large standard deviation and containing a large number of points. In this case, in reason of the condition imposed by the number of points, the RANSAC algorithm will not accept another plane for replacing it. For solving this problem, a new threshold is introduced. This threshold is the number of points of the smallest foreseeable plane surface (PN\_S). It is equal to the product of the smallest foreseeable plane surface by the point density. Then the 11th line in the algorithm becomes:

```
if st < bestStd and length(s) > PN_S then
```

(8)

After this modification, the percentage of successful results reached by the application of the adapted RANSAC algorithm reaches 85%.

**4.2.2 Improvement of the detected planes quality:** As already mentioned, the application of RANSAC algorithm allows the detection of planes which do not necessarily present roof planes. It represents perhaps one roof plane in addition to other noisy points which belong to other roof planes, as the points inside the red circles in Fig.-6a. These noisy points have to be eliminated from the detected plane, and have to be reassigned to the initial cloud.

Furthermore, inside the detected plane, there are some lost points (inside the blue circle in Fig.6a). These points have to be added to the fitted plane and extracted from the cloud in the same time.

The last two problems can be solved by applying mathematical morphology procedures on the binary Digital Surface Model (DSMb) calculated for the detected plane.

Results obtained by processing the data of Fig.-6a are shown in Fig.-6b. If the detected plane represents a set of points belonging to different roof planes and distributed stochastically, then the plane is rejected. Moreover, an additional condition checking if the new parameters never occurred previously is added automatically to the 11th line of the algorithm. So this plane is avoided in the next trials.

After detecting all planes covering perfectly the roof, the remaining points are normally either noisy points or small roof details (Fig.-6c). For classifying these points, a region growing algorithm is used, deciding if the set of points represents noise or roof details. Hence, the two criteria used are: the smallest foreseeable surface of a roof detail and the segment form. Thus, if the doubtful set of points represents noise, it is added to the nearest plane, else it is considered as a new plane. Fig.-6d presents the final result of roof planes detection obtained with the extended RANSAC algorithm.

#### 4.3 Results and accuracy analysis

The building used for testing the approach in Fig.7 has numerous details on its roof (chimneys, dormers, windows). Moreover, the majority of roof plane surfaces are small regarding the point density. Furthermore, its point density is relatively weak (1.3 points/m<sup>2</sup>).

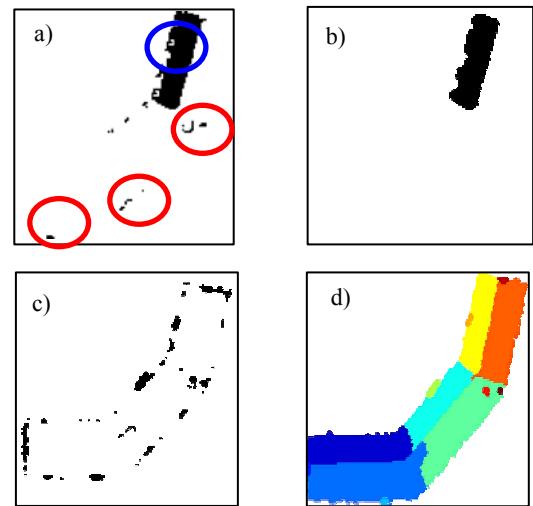


Figure 6. a) DSMB of the detected plane. It corresponds to the yellow plane in Fig.6d; b) DSMB of the improved detected plane; c) DSMB of the remaining points after detecting the roof planes; d) Final result of roof planes detection.

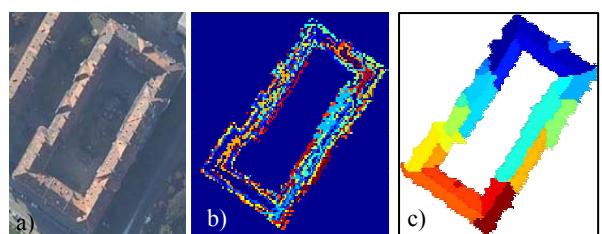


Figure 7. Roof planes detection results. a) Aerial image. b) Using classic RANSAC algorithm. c) After extension of RANSAC algorithm. The colours in b) and c) represent the different building roof planes

All these reasons lead to plane misdetection when the original RANSAC algorithm is applied (see Fig.-7b). On the other hand, after applying the extended RANSAC algorithm, the automatic roof plane detection over the same building is satisfactory. Fig.-7c illustrates clearly the improvements gained by the extension of RANSAC algorithm.

Finally, the same test has been achieved on almost 58 buildings with different forms and different Lidar point densities. The good results confirm the potentiality of the extended RANSAC algorithm. Although the improvements showed promising results, it must be noted that the level of generalisation and consequently the result quality depend obviously on the point cloud characteristics (point density, position accuracy, noise), on the architectural complexity of the building roof and on the dimensions of the building roof planes and their details.

## 5. CONCLUSION

This paper presented and compared two methods for automatic roof planes detection from Lidar data. These methods are 3D Hough-transform and RANSAC algorithm. The principle and the pseudocode of each one were detailed. In order to test the original and the improved algorithms, two sets of point clouds characterized by different densities and containing different building forms were used.

It is stated that both methods are based on pure mathematical principles in order to detect the best planes from 3D point clouds. This characteristic leads sometimes to the production of intolerable errors. The main advantage of RANSAC algorithm is its rapidity and the percentage of successful detected roof planes. These reasons were our main motivations.

Thus, two enhancements were suggested in order to increase its capacities. The first one was the improvement of the original data by generating a new point cloud. The second improvement was the adaptation of the algorithm, so that the extended algorithm allows detecting the best roof plane instead of the best mathematical one. At last, the satisfying results obtained for different clouds even with weak point density validate the proposed processing chain. Once the building roof planes are detected automatically, it becomes easier to complete the processing chain and achieve the last steps leading to the complete 3D building model.

## REFERENCES

- Ameri, B. and Fritsch, D., 2000. Automatic 3D building reconstruction using plane-roof structures, Published by the ASPRS, Washington DC.
- Brenner, C., 2000. Towards fully automatic generation of city models. In: *Int. Arch. Photogrammetry and Remote Sensing*, vol. 32, Part 3. Amsterdam, pp. 85–92.
- Bretar, F. and Roux, M., 2005. Hybrid image segmentation using LiDAR 3D planar primitives. *ISPRS Proceedings*. Workshop Laser scanning. Enschede, the Netherlands, September 12-14, 2005.
- Davies, E. R., 1988. Application of the generalized Hough transformation to corner detection. *IEEE proceedings*, Vol 135, Pt. E, No. 1.
- Fischler, M. A. and Bolles, R. C., 1981. Random Sample Consensus: A paradigm for model fitting with application to image analysis and automated cartography. *Communications of the ACM*, 24(6): 381-395.
- Forlani, G., Nardinocchi, C., Scaioni, M. and Zingaretti, P., 2004. Building reconstruction and visualization from Lidar data. *Int. Arch. of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXIV, Part 5/W12.
- Forlani, G., Nardinocchi, C., Scaioni, M. and Zingaretti, P., 2006. Complete classification of raw Lidar data and 3D reconstruction of buildings. *Pattern Anal. Applic.* (2006) 8: 357–374. DOI 10.1007/s10044-005-0018-2.
- Gonzalez, RC., Woods, RE. and Eddins, SL., 2004. *Digital Image processing using MATLAB*. Printed in USA, Pearson Prentice Hall. ISBN 0-13-008519-7. 609p.
- Hough, P.V.C., 1962. Method and Means for Recognizing Complex Patterns. *U.S. Patent* 3.069.654.
- Maas, H.-G. and Vosselman, G., 1999. Two algorithms for extracting building models from raw laser altimetry data. *ISPRS Journal of Photogrammetry & Remote Sensing* Vol. 54, No. 2/3, pp. 153-163
- McGlone, J. C., M. Mikhail, E. and Bethel, J., 2004. *Manual of photogrammetry (5th edition)*. Published by the ASPRS, ISBN 1-57083-071-1, 1151p.
- Nguyen, V., Martinelli, A., Tomatis, N. and Siegwart, R., 2005. A comparison of line extraction algorithms using 2D laser rangefinder for indoor mobile robotics. IEEE/RSJ Proceedings. *Int. conference on intelligent robots and systems, IROS*, Edmonton, Canada.
- Oda, K., Takano, T., Doihara, T. and Shibasaki, R., 2004. Automatic building extraction and 3-D city modeling from lidar data based on Hough transformation. *Int. Arch. of Photogrammetry and Remote Sensing*, Vol. XXXV, part B3.
- Overby, J., Bodum, L., Kjems, E. and Ilsoe, P. M., 2004. Automatic 3D building reconstruction from airborne laser scanning and cadastral data using Hough transform. *Int. Arch. of Photogrammetry and Remote Sensing*, Vol. XXXV, part B3.
- Rabbani, T. and Van den Heuvel, F., 2005. Efficient Hough transform for automatic detection of cylinders in point clouds. *ISPRS Proceedings*. Workshop Laser scanning. Enschede, the Netherlands, September 12-14, 2005.
- Rottensteiner, F., 2003. Automatic generation of high-quality building models from Lidar data. *IEEE Computer Graphics and Applications* 23(6), pp. 42-51.
- Tarsha-Kurdi, F., Landes, T. and Grussenmeyer, P., 2007. Joint combination of point cloud and DSM for 3D building reconstruction using airborne laser scanner data. *Urban Remote Sensing Joint Event URBAN/URS 2007*. 11-13 April Télécom Paris.
- Vosselman, G. and Dijkman, S., 2001. 3D building model reconstruction from point clouds and ground plans. *Int. Arch. of Photogrammetry and Remote Sensing*, XXXIV-3/W4: 37-43.