From Point Clouds to 3D City Models: The Case Study of Villalba (Madrid)

Juan Mancera-Taboada, Pablo Rodriguez-Gonzalvez,
Diego Gonzalez-Aguilera*, Benjamín Arias-Perez
Land and Cartographic Engineering Department
University of Salamanca
Ávila, Spain
juaniyoperote@usal.es, pablorgsf@usal.es,
*daguilera@usal.es, benja@usal.es

David Hernandez-Lopez, Beatriz Felipe-Garcia
Regional Development Institute
University of Castilla-La Mancha
Albacete, Spain
david.hernandez@uclm.es, beatriz.felipe@uclm.es

***Abstract*—This article presents a practical study of the use of LIDAR (Light Detection and Ranging) data processing to transform point clouds into a three-dimensional city model compatible with CAD (Computer-Aided Design) graphic design systems. The article describes the methodology followed while concentrating on an increase in automation and reviewing the algorithms used. This case study demonstrates the importance of the LIDAR technology for 3D city modelling and notes several applications that may arise, especially in the context of urban and regional planning.**

Keywords—LIDAR; classification; reconstruction; 3D modelling; 3D city

I. INTRODUCTION

The airborne LIDAR system is a data capture method that can be used as an alternative or complement to photogrammetry. It also constitutes an effective tool for creating Digital Terrain Models (DTM) and Digital Surface Models (DSM) in urban areas. Although it is a technique that has yet to mature, LIDAR has advantages in certain situations when compared to aerial photogrammetry [1]. Example LIDAR applications include the following: urban areas, strip mines and landfills, snowy areas, dunes, marshes, wetlands, forests and areas with dense vegetation, waterways and water resources, the control and monitoring of coastal erosion and monitoring and managing natural disasters. Additionally, the LIDAR system is better suited to automating the detection of buildings than the current technique of extracting buildings from photogrammetry [2]. The demand for using LIDAR in these fields comes partially from the development of algorithms used to classify LIDAR point clouds. The scenarios that pose the greatest problems are complex urban landscapes, irregular building shapes (e.g., buildings with several floors, patios, stairs or squares) and discontinuities in the field (break-lines). In this sense, the proposed classification algorithms [3] can be sorted using one of the following methods:

A. According to the Data Structure

There are many algorithms that work with the raw point cloud [4-5], whereas other authors [6-9] resample the point cloud based on a mesh with the aim of classifying data in a more optimal and efficient way.

B. According to the Neighbourhood

Point to Point [10-11] is a method in which two points are compared to each other, and the discriminant function is based on the position of both points. If the obtained result is greater than a certain threshold, then one of the points is assumed to belong to a particular class. The greatest drawback to this method is that only one point is classified during each iteration.

Point to Points [4, 12-13] is a method in which the points neighbouring the point of interest are used to solve the discriminant function, and only one point is classified during each iteration.

Points to Points [5-7, 9] is a method in which several points are used to solve the discriminant function, and more than one point is sorted during each iteration.

C. According to the Initial Hypothesis

To use this method, neighbouring points must be adapted to a given parametric surface.

According to the hypothesis of the slope [10-11], where the slope or height difference is measured between two points. If the slope exceeds a certain threshold, then the highest point is sorted into a particular class. In [9], the initial hypothesis is a horizontal plane against which the differences in height are related. In [4-5, 7-9, 13], the discriminant function of the initial hypothesis is a parametric surface, which will act as a reference to establish the height differences among points.

D. According to the Calculation Method

One-step methods [10-11], in which the classification problem is solved linearly without requiring iteration, result in reductions in both computation time and dedicated memory.

Iterative methods [4-7, 9, 13], in which a nonlinear, and therefore iterative, approach is used, yield better results but consume more computational resources.

E. According to the Modification of the Point

Point Removal methods [4-5, 10-11, 13] remove points lying outside of the dominant function of the original cloud (irregular clouds).

Point Replacement methods [6-7, 9] replace the height of a point with a different height determined by interpolation. This type of approach is commonly used in uniform clouds.

The classification of LIDAR points in urban areas is conducted using automatic building extraction algorithms. Currently, the goal of total and automatic building classification has not been met, mainly due to the complexity of the urban scene [14]. Automatic building extraction can be conducted either using only LIDAR data or using LIDAR data supplemented with orthoimages. Studies have been conducted that combined both strategies, using the LIDAR data to classify buildings, and the images to identify and differentiate between vegetation areas and buildings [2, 15]. This strategy, however, has lacked horizontal accuracy in the detected buildings, especially along their edges [16]. As discussed in [17], it is difficult to extract a straight line accurately using only LIDAR data because the data resolution directly influences the quality of the geometric extraction [18]. To solve this problem, [19] used a methodology to extract buildings from LIDAR point clouds using the Hough´s Transformation [20]. Other authors [14, 21-24] have used integration techniques to incorporate both LIDAR data and images into the building extraction process. This strategy lends greater horizontal accuracy to the building detection. In particular, [21] applied a pixel-based classification strategy using a Normalised Digital Surface Model (NDSM) is used as an additional channel of aerial images, while [23] used aerial images and the point cloud to extract straight lines around the buildings by analysing the angle of the dominant line.

In this paper, a case study is presented that demonstrates the creation of a 3D city model from a LIDAR point cloud, with an emphasis on increased quality and automation. Following this brief introduction to LIDAR point classification algorithms and building extraction, the next section details the proposed LIDAR data processing methodology with special emphasis placed on the employed point classification algorithm and the building extraction. A practical case study focused on the town of Villalba (Madrid) is subsequently discussed, and finally, the most relevant conclusions and future perspectives are presented.

II. FROM POINT CLOUDS TO 3D CITY MODELS

To generate a DTM or DSM, extract buildings or model a 3D city, it is necessary to have a good classification of LIDAR data. This classification can take the form of either a simple discrimination between terrain points and non-terrain points or a more sophisticated discrimination of vegetation, buildings and invalid points. Points collected on flat surfaces are regularly distributed, and the differences in elevation between neighbouring points are smooth, linear and continuous. By contrast, the points obtained in rough terrain, wooded or urban areas, where there are several surface changes, have greater differences in elevation between neighbouring points and significant discontinuities in the data. Additionally, LIDAR data return a specific pattern depending on a surface's physical characteristics and materials. On metallic objects and glass, where the surfaces and reflections are small, the points usually appear in isolated groups. On roads and smooth surfaces, where the points are uniformly spaced and the differences in elevation between neighbouring points are small (less than 0.2 m), only a return echo is generated.

Depending on the type of roof, building surfaces are commonly smooth and regular, and they produce a single return (echo). Additionally, elevation differences between neighbouring point clouds will be considerable. In areas of vegetation, multiple echoes are commonly generated, and the resulting zone is characterised by an irregular distribution of points. There are also elevation differences between the clouds of neighbouring points. In water areas, the mirror-like surface behaviour reflects echoes away from the sensor, and there will be no collected information in these areas. Therefore, according to the number of returns (or echoes), an area that includes an echo can be classified as a ground surface, the roof of a building, or the top cover area of dense vegetation; similarly, if there are multiple echoes, the area can be classified as medium to high vegetation or a building edge. Finally, if there is no return, then the given area has a specular behaviour (e.g., water).

The following table (Table I) lists the possible LIDAR data point classifications.

TABLE I. DIFFERENT LIDAR DATA POINT CLASSIFICATIONS

Classes Of Points In Lidar Data			
Wrong information			
Low (Blunders)			
Ground			
Road			
Vegetation	High	Medium	Low
Building			

A. Point Cloud Classification

After considering all of the approaches mentioned in the introduction, it was decided that the methodology proposed in [12] be used for LIDAR data classification. This methodology is an automatic process requiring the user to only input critical parameters for data point classification. Additionally, this process is iterative and begins with an initial surface generated from a randomly chosen set of points. This set of points is triangulated and constituted as a reference surface (TIN - Triangulated Irregular Network). Subsequently, new points will be added only if they satisfy the established thresholds. The densification parameters of the TIN, the distance to the faces of the TIN and the vertices angles are derived from LIDAR data based on a simple statistical analysis using the minimum, median and maximum values of histograms.

Another important characteristic is the typology of the area; in forest areas, there are different characteristics and morphologies than in urban areas. In forest areas, variations in the terrain will be more continuous, whereas in urban areas, a flat surface with occasional discontinuities will be the norm. During each iteration, a point will be added to the TIN reference surface if it satisfies the distance threshold and angle criteria. These thresholds are adaptive and update after each iteration. This iterative process continues until there are no more points to add (Fig. 1 and Fig. 2).

The outline of the algorithm is as follows:

1) Establish the initial TIN parameters, distance thresholds (to the faces of the TIN) and angles (that are formed with the nodes).

2) Select points that define the initial surface from a random sample.

3) TIN iterative densification:

a) Calculate the parameters used for each iteration from the points added to the TIN.

b) Points are added to the TIN if they are within the threshold values set for distance and angle.

4) Repeat until all points have been classified as a terrain or object.

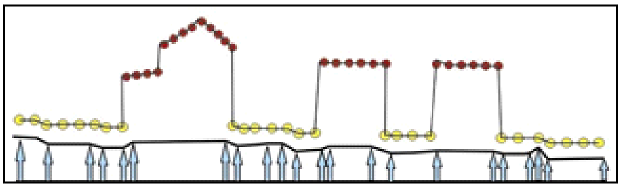

Figure 1. Example of building the adaptive TIN model of Axelsson

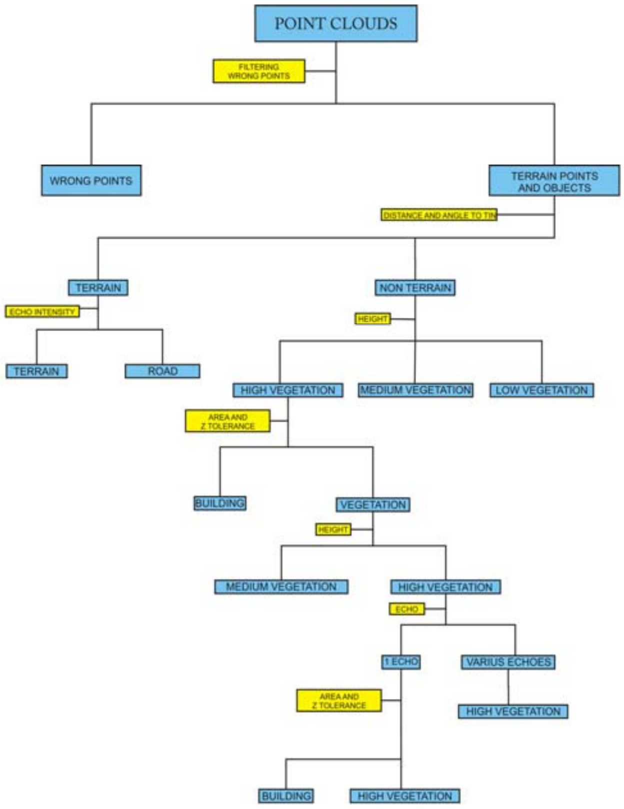

Figure 2. Workflow of the point cloud classification

B. *Automatic Building Reconstruction*

The automatic detection of buildings has advanced in recent years, but the available algorithms are still not fully autonomous because they always require human operator intervention. For this reason, the fully automatic acquisition of vector models for cities is still a challenge. The ultimate goal is to represent all of the entities in a city using a three-dimensional model while retaining the entities' physical characteristics of size and shape. Methods for extracting building models can be classified according to the input data: (i) reconstruction based on original data, without interpolation or generation of a regular grid [25]; (ii) reconstruction based on an interpolation of the original data [26]; (iii) reconstruction based on LIDAR data and image registration [21, 23]; or (iv) the direct extraction of parametric shapes such as planes, cylinders and spheres.

In this study, as mentioned in the introduction, the reconstruction was exclusively based on the use of the point cloud data without using any images as support for the automatic reconstruction [27]. This method was chosen due to a lack of suitable and geo-referenced images of the case study area that could support the automatic reconstruction process. For this reason, the available images were only used for radiometric mapping and manual edge correction.

There are many architectural objects that can be represented as flat shapes, cylinders and spheres, which allows the objects to be described using controllable parameters while also allowing them to be extracted using robust methods that detect groups in a parameter space. This study focuses on the extraction of planes, the most common elements in architectural construction. In the ideal case of a noise-free plane point cloud, all locally orthogonal surfaces should point in the same direction. Given a sufficiently reliable and discreet class, the plane extraction corresponding to the roofs of buildings was conducted using the Hough Transformation [20] extrapolated to a three-dimensional context and consistent in the parameterisation of a set of points defined initially in LIDAR space (*O'XYZ*) to a parameter space (*O'abd*).

Given a plane defined by the analytical equation *Ax + By + Cz + D = 0*, the equation can be transformed with reference to the *z* coordinate of the LIDAR point cloud as follows:

$$z = -\frac{A}{C}x - \frac{B}{C}y - \frac{D}{C} \tag{1}$$

Therefore, a *Z* plane in the LIDAR space can be defined as a point *(a, b, d)* in the parameter space:

$$\mathrm{z} = \mathrm{ax} + \mathrm{by} + \mathrm{d}, \tag{2}$$

with *a =- A / C, b =- B / C, d =- D / C*, and where *xyz* are the coordinates of a point belonging to the LIDAR space, *ab* are the coordinates of the point in the parameter space, and *d* is the distance from the point to the *Z* plane.

Although the classification process filters and optimises the building class, the number of points makes it impossible to undertake a raw parameterisation. Therefore, it was decided to use the robust estimator RANSAC (Random Sample Consensus) on a LIDAR point cloud in a three-dimensional space to find the best existing plane. For this purpose, RANSAC selects three random points from the building class and calculates the parameters of the plane that they constitute. It then detects all of the points of the LIDAR cloud belonging to the random plane using a certain threshold, usually the orthogonal distance to the plane. RANSAC repeats this process *N* times, each time comparing the resulting plane with the previously calculated one and keeping the plane that contains the most points. Ideally, the extracted plane will be obtained along with a list of possible out of plane points (outliers).

In particular, the algorithm needs the following three inputs:

1) The point cloud classified into the building class

2) A tolerance threshold according to *t,* which is the distance between the chosen plane and the rest of the points that takes into account the altimetric uncertainty associated with the LIDAR data.

3) The maximum number of probable points belonging to a single plane, which is deduced from the point density and the general characteristics of the object to be extracted.

Additionally, it is important to note that the number of RANSAC random combinations (*N*) can be considered an input parameter, or it can be calculated directly if a minimum probability of finding at least one set of observations is determined using the following equation:

$$P_{\min} = 1 - (1 - (1 - \varepsilon)^{U})^{t_{\min}}, \qquad (3)$$

where the minimum number of combinations is determined based on the number of unknowns, *U* (*U* = 3 in our case), and ε is the expected percentage of gross errors for a specified probability (in this case 95%). The number of random combinations can then be found as:

$$N(P_{\min}, \varepsilon, U) = \frac{\ln(1 - P_{\min})}{\ln(1 - (1 - \varepsilon)^{U})} \qquad (4)$$

The following table (Table II) shows the necessary number of combinations to guarantee the correct solution under a certain probability and depending on the number of unknowns that we want to resolve [28].

TABLE II. NUMBER OF COMBINATIONS (N) REQUIRED IN RANSAC PROCESS FOR A GIVEN PROBABILITY (ε) AND A NUMBER OF UNKNOWNS (U). HIGHLIGHTING SHOWS THE NUMBER OF COMBINATIONS SELECTED IN OUR CASE.

N	**ε =0.1**	**0.2**	**0.4**	**0.6**	**0.8**	**0.9**
U = 1	2	2	4	6	14	29
2	2	3	7	18	74	299
3	3	5	13	46	373	2995
4	3	6	22	116	1871	29956
5	4	8	38	292	9361	299572
6	4	10	63	730	46807	2995731
7	5	13	106	1827	234041	29957322
8	6	17	177	4570	1170207	299573226

For the RANSAC algorithm to be successfully applied in a three-dimensional context (the LIDAR point cloud), the set of those points considered in the plane extraction will be excluded from the original point cloud after each iteration. This process is repeated until the number of points extracted or unmodelled falls below a certain threshold. In this way, every point belongs to only a single plane, and therefore, a point contributes to only the adjustment of the plane to which it belongs.

To obtain a plane extraction as accurately as possible, each RANSAC combination selected as a plane is iteratively adjusted through the use of a least squares calculation using all of the points belonging to the RANSAC selected combination. In this case, the least squares criterion is the orthogonal distance to the extracted plane.

Considering that most of the building roofs will not be constituted by a single plane, it is necessary to determine the intersection lines among planes and, thereby, the entire structure of the building eaves. For this purpose, the equation for the intersection between two planes is used. Given two planes, π_1:$A_1x+B_1y+C_1z=0$ and π_2:$A_2x+B_2y+C_2z=0$, their intersection may be defined by the line *l*. The direction vector of the line *l* is calculated using the cross product of the normal vectors of both planes:

$$\pi_1 x \pi_2 = (A_1, B_1, C_1) x (A_2, B_2, C_2), \qquad (5)$$

where (A_1, B_1, C_1) are the parameters of the normal vector to the plane π_1 and (A_2,B_2,C_2) are the parameters of the normal vector to the plane π_2.

As the axis of the line *l* is not uniquely defined by the vector obtained in (5), it is necessary to obtain a point p_0 = (x_1, y_1, z_1) that belongs to both planes and thus to the line *l.* This point is achieved by restricting the value of one of the coordinates (e.g., *z* = 0) and solving the resulting system with two equations and two unknowns.

Finally, the volume of the building is generated through an orthogonal projection over the DTM of the different edges of the extracted planes. Unfortunately, this automatic process yielded no definitive results and the model suffered from errors. Therefore, each building was checked, and multiple errors were removed using manual tools.

III. EXPERIMENTAL RESULTS: THE CASE STUDY OF VILLALBA

The case study was conducted on the town of Collado Villalba in the Province of Madrid (Spain). The centre of the town of Collado Villalba contains a consolidated area with a density of 2062.57 inhabitants/km^2. In this space, there are large areas with houses as well as areas with abundant vegetation. The project focused on an area of approximately 2 km^2 that contained gentle slopes, high vegetation and buildings of multiple dimensions and heights. This region yielded a cloud of approximately 4 million raw data points.

The LIDAR data used in this study were taken with the Leica ALS sensor 50_II (Table III). This sensor is an airborne laser scanner with a 95.8 kHz pulse rate (95800 pulses per second), an opening angle of 45° and a capture height of approximately 1000 m from the ground. The data are generated in the LAS1.0 free format, with an average density of 1.9 points per square meter, with an initial vertical accuracy of 12 cm. An average overlap between transversal scan passes of 32.33% was determined.

TABLE III. TECHNICAL CHARACTERISTICS OF THE DATA COLLECTED WITH THE 50_II SN48 ALS SENSOR.

Sensor	**ALS 50_II SN48**
Laser Pulse Rate Used	95800.00 Hz
Scan FOV (half angle)	22.50 (Deg.)
Point Density (average)	1.87 point/m^2
Estimated error Z	0.12 m
Overlap	32.33%
Terrain Elevation AMSL (minimum in survey area)	850 m
Terrain Elevation AMSL (maximum in survey area)	1050 m
Nominal Flying Height Above Minimum Terrain Elevation	1000 m
Nominal Flying Altitude	1850 m
Assumed GPS Error	0.05 m

The initial pre-processing tasks have been excluded from this article as they are outside of the article scope. The pre-processing comprised the identification of overlapped areas between runs, their alignment and the subsequent removal of redundant information for those points belonging to adjacent runs. Additionally, all of the points with a weak signal or systematic failures, points below the terrain class, or noise points such as birds and moving objects were deleted.

In this study, the data point classification was conducted using a bottom-up hierarchical strategy. This classification allowed an initial discretisation of the terrain points and non-terrain points and then focused on discerning non-terrain points into the following entities: vegetation (low, medium and high) and buildings. Specifically, in the case study, we begin with an initial terrain points model composed of the lowest elevation points. An initial TIN is generated based on Delaunay triangulation, and this TIN allows us to establish the reference surface. The triangles in this initial model are mostly below the true ground surface. The routine then begins to analyse the model iteratively from the bottom up, adding new points as it progresses.

Each added point makes the model of the soil surface closer to the true terrain. The parameters that during each iteration whether a point is added to the soil surface during each iteration are the angular parameter, consisting of the maximum angle between a candidate point and the nearest triangle vertices, and the distance parameter, which is the orthogonal distance between a candidate point and the closest triangular plane. These parameters are suitable for a terrain with smooth and continuous break lines. In the case study, these parameters were set as 6 degrees for the threshold angle and 1.4 m for the threshold distance. These parameters can be varied for adapting to other types of land if it is very abrupt. A total of 361,086 points were classified as terrain (Fig. 3), and the remaining non-terrain points were classified as either vegetation or building.

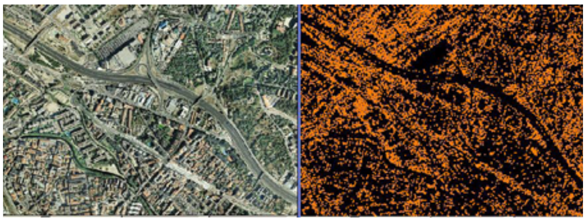

Figure 3. Results of the terrain points filtering process (right) and the validation with the orthophotograpy (left).

Considering that the points of the roads and highways have been classified as area-points due to the similarity between their characteristics and the ground-points, the two categories must be differentiated. This is accomplished using the response intensity, which depends on the material surface [4]. The reflectivity value of each point is represented as gray levels from 0 to 255, where 0 corresponds to no light incident on the sensor, and 255 is the maximum reflectivity. The average intensity of the reflected signal for a road corresponds to a gray level of 55.26, while the average for the ground is 148.13. In this case study, there was a double threshold of intensity values (40-100) that filtered out those points that belong to highways or roads. As a result, 185,430 points were classified as road (Fig. 4).

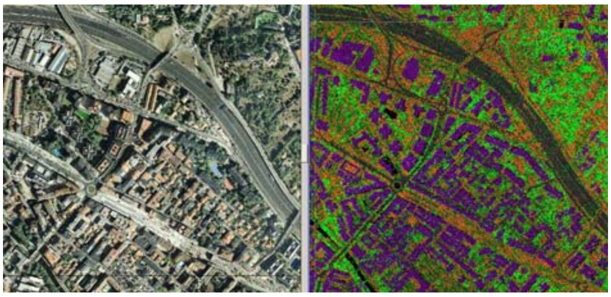

Figure 4. Classification of the road entity in black (right) and the validation with the orthophotography (left).

Subsequently, the non-terrain class was divided into the different types of vegetation entities. For that classification, a height threshold was used to classify a point into the following three possible types of vegetation: low vegetation (from 0.01 to 0.2 m), medium vegetation (from 0.2 to 3 m), and high vegetation (from 3 to 150 meters). In total, 50,543 points were classified as low vegetation, 331,291 points as medium vegetation, and 746,932 points as high vegetation (Fig. 5).

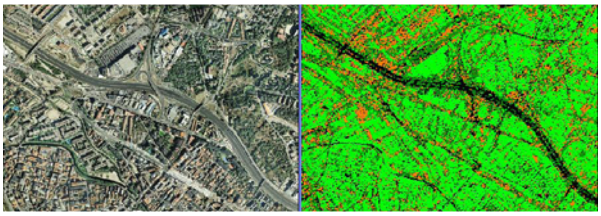

Figure 5. Classification of the vegetation entity represented in three shades of green (right) and validation with the orthophoto (left).

In regards to the building classification, it should be mentioned that the area of study is an urban area in which there are residential buildings between 3 and 5 plants with heights ranging from 8 to 24 meters. For the initial building classification, the following two thresholds were set: the minimum area of the building and the tolerance in height. For the initial classification process of the building class, an initial threshold was set at 40 m^2 for the building area and 0.65 m for the minimum height. Additionally, any points classified as a building had to only have a single return (echo). As a result of this process, 402,055 points were classified as belonging to the building class.

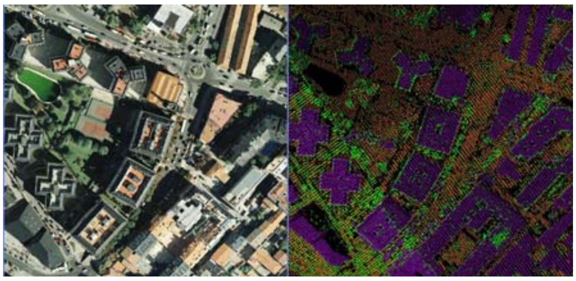

Figure 6. Classification of the buildings in purple (right) and validation with orthophotography (left).

As shown in Fig. 6, the established thresholds incorrectly classified the edges and the roofs of buildings as high vegetation. Therefore, manual intervention by the user was necessary to allow for a more accurate classification of the building class.

Once all of the data points had been classified, the next step was to apply the reconstruction algorithms to the buildings. This task, performed automatically, yielded incomplete results and numerous errors. It was necessary to manually supervise the process via photo-interpretation using the orthophotos as well as via a CAD cleansing task that fixed the closure and connection between the different planes of certain buildings. The following figures (Fig. 7-10) illustrate some of the common bug fixes that were required during the 3D building reconstruction. The data processing shown in this study case has been rather expensive in computational terms, i.e., data point classification needed 4 hours, reconstruction algorithms required 8 hours, and for CAD design 3 days were spent.

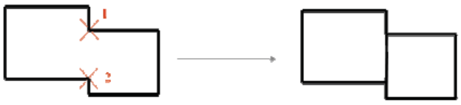

Figure 7. In this case, two buildings are defined instead of one.

.

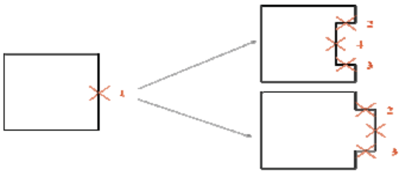

Figure 8. The correction of incoming or outgoing roofs

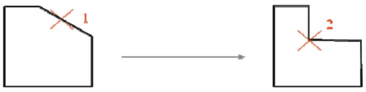

Figure 9. The correction of a building corner definition

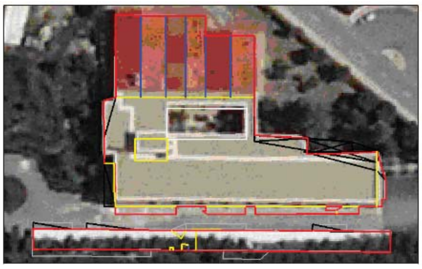

Figure 10. Common errors in the automatic vectorisation of the building edges and their corrections.

Fig. 11 presents the results of the building reconstruction after the photo interpretation and CAD debug phases.

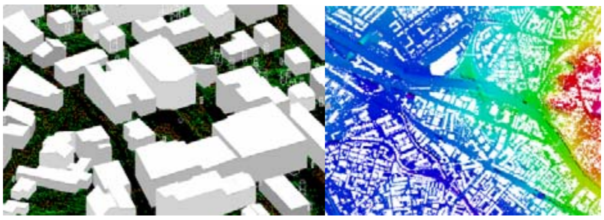

Figure 11. 3D CAD model of the buildings: detailed perspective view (left) and overview on the ground (right)

IV. Concluding Remarks and Future Perspectives

This study presented the generation of a 3D city model from LIDAR data taken over the urban centre of the city of Collado Villalba, Madrid (Spain). To obtain DTM and DSM of sufficient quality, it is crucial to utilise efficient algorithms that automatically classify the raw data points. Working with efficient algorithms is also critical for automatic building reconstruction and model triangulation. A classification methodology that operates according to the geometric features and proprieties of the data points has been presented alongside a robust strategy for extracting the roof planes of buildings. Although significant levels of automation have been attained, it is still necessary to manually correct certain errors relating to building vectorisation. The automatic vectorisation was ultimately successful and able to distinguish close buildings, although the time spent correcting model errors was very high and necessitates the further development of more efficient algorithms. Regardless of its shortcomings, a 3D city model was created that allows a multitude of applications in the field of urbanism and planning. In conclusion, the use of LIDAR data to create three-dimensional models of urban areas is valid provided that a sufficient point density (greater than 2 points/m^2) is utilised and the model is produced under the supervision of an operator capable of manually correcting any errors that arise.

In relation to the future perspectives, there is two relevant issues to be considered: (i) the quality of the building reconstruction process, which could be solved through the evaluation of different TIN interpolation methods, such as the inverse distance weighting (IDW), spline or Kriging algorithms; (ii) the accuracy assessment of final DEM and final building reconstruction by the use of Geomatics techniques (terrestrial laser scanner) as ground truth for evaluation purposes.

REFERENCES

[1] E. P. Baltsavias, "A comparison between photogrammetry and laser scanning," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 54, pp. 83-94, 1999.

[2] T. T. Vu, F. Yamazaki, and M. Matsuoka, "Multi-scale solution for building extraction from LiDAR and image data," International Journal of Applied Earth Observation and Geoinformation, vol. 11, pp. 281-289, 2009.

[3] G. Sithole and G. Vosselman, "Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 59, pp. 85-101, 2004.

[4] P. Axelsson, "Processing of laser scanner data--algorithms and applications," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 54, pp. 138-147, 1999.

[5] N. Pfeifer, P. Stadler, and C. Briese, "Derivation of digital terrain models in the SCOP++ environment," in OEEPE Workshop on Airborne Laserscanning and Interferometric SAR for Digital Elevation Models, Stockholm, Sweden, 2001.

[6] M. Brovelli, M. Cannata, and U. Longoni, "Managing and processing LIDAR data within GRASS," in Open source GIS - GRASS users conference, Trento, Italy, 2002.

[7] M. Elmqvist, "Ground estimation of lasar radar data using active shape models," in OEEPE Workshop on Airborne Laserscanning and Interferometric SAR for Digital Elevation Models, Stockholm, Sweden, 2001.

[8] M. Elmqvist, E. Jungert, F. Lantz, A. Persson, and U. Söderman, "Terrain modelling and analysis using laser scanner data," in ISPRS Workshop - Land Surface Mapping and Characterization using laser altimetry, Annapolis, Maryland, 2001, pp. 219-226.

[9] R. Wack and A. Wimmer, "Digital Terrain Models from Airborne Laserscanner Data-a Grid Based Approach," in Photogrammetric Computer Vision (PCV02), Graz, Austria, 2002, pp. 293-296.

[10] G. Sithole, "Filtering of laser altimetry data using a slope adaptative filter," in ISPRS Workshop - Land Surface Mapping and Characterization using laser altimetry, Annapolis, Maryland, 2001, pp. 203-210.

[11] M. Roggero, "Airborne laser scanning: clustering in raw data," in ISPRS Workshop - Land Surface Mapping and Characterization using laser altimetry, Annapolis, Maryland, 2001, pp. 227-232.

[12] P. Axelsson, "DEM generation from laser scanner data using adaptive TIN models," in XIXth ISPRS Congress, Amsterdam, The Netherlands, 2000, pp. 110-117.

[13] G. Sohn and I. Dowman, "Terrain surface reconstruction by the use of tetrahedron model with the MDL Criterion," in Photogrammetric Computer Vision (PCV02), Graz, Austria, 2002, pp. 336-344.

[14] D. H. Lee, K. M. Lee, and S. U. Lee, "Fusion of lidar and imagery for reliable building extraction," Photogrammetric Engineering and Remote Sensing, vol. 74, pp. 215-225, 2008.

[15] F. Rottensteiner, J. Trinder, S. Clode, and K. Kubik, "Using the Dempster-Shafer method for the fusion of LIDAR data and multi-spectral images for building detection," Information Fusion, vol. 6, pp. 283-300, 2005.

[16] Y. Li and H. Wu, "Adaptive building edge detection by combining lidar data and aerial images," in XXIst ISPRS Congress, Beijing, China, 2008, pp. 197-202.

[17] L. Cheng, J. Gong, X. Chen, and P. Han, "Building boundary extraction from high resolution imagery and lidar data," in XXIst ISPRS Congress, Beijing, China, 2008, pp. 693–698.

[18] A. Sampath and J. Shan, "Building boundary tracing and regularization from airborne lidar point clouds," Photogrammetric Engineering & Remote Sensing vol. 73, pp. 805-812, 2007.

[19] G. Vosselman, "Building reconstruction using planar faces in very high density height data," in ISPRS Workshop - Automatic Objects from Digital Imagery, Munich, Germany, 1999, pp. 87-94.

[20] P. V. C. Hough, "Method and means for recognizing complex patterns," U.S. Patent 3.069.654, 1962.

[21] N. Haala and C. Brenner, "Extraction of buildings and trees in urban environments," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 54, pp. 130-137, 1999.

[22] L. C. Chen, T. A. Teo, Y. C. Shao, Y. C. Lai, and J. Y. Rau, "Fusion of LIDAR data and optical imagery for building modeling," in XXth ISPRS Congress, Istanbul, Turkey., 2004, pp. 732-737.

[23] G. Sohn and I. Dowman, "Data fusion of high-resolution satellite imagery and LiDAR data for automatic building extraction," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 62, pp. 43-63, 2007.

[24] N. Demir, D. Poli, and E. Baltsavias, "Extraction of buildings using images & LIDAR data and a combination of various methods," in Object Extraction for 3D City Models, Road Databases and Traffic Monitoring - Concepts, Algorithms and Evaluation (CMRT09), Paris, France, 2009, pp. 71-76.

[25] H.-G. Maas and G. Vosselman, "Two algorithms for extracting building models from raw laser altimetry data," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 54, pp. 153-163, 1999.

[26] F. Rottensteiner and C. Briese, "A New Method for Building Extraction in Urban Areas from High-Resolution LIDAR Data," in Photogrammetric Computer Vision (PCV02), Graz, Austria, 2002, pp. 295-301.

[27] G. Vosselman and S. Dijkman, "3D building Model Reconstruction from Point Clouds and Ground Plans," in ISPRS Workshop - Land Surface Mapping and Characterization using laser altimetry, Annapolis, Maryland, 2001, pp. 37-44.

[28] R. Hartley and A. Zisserman, Multiple view geometry in computer vision. New York: Cambridge University Press 2003.