

## 3D model reconstruction from aerial ortho-imagery and LiDAR data

ElSonbaty Loutfia, Hamed Mahmoud, Ali Amr and Salah Mahmoud

Faculty of Engineering Shoubra, Benha University, Cairo, Egypt

Email: loutfia\_karam2001@hotmail.com, prof.mahmoudhamed@yahoo.com, amrhali@feng.bu.edu.eg, engmod2000@yahoo.com

(Received: Jul 13, 2016; in final form: Feb 15, 2017)

**Abstract:** Three dimensional (3D) city model is an interesting research topic in the last decade. This is because achieving rapid, automatic and accurate extraction of a realistic model for the large urban area is still a challenge. Consequently, increasing the efficiency of 3D city modeling is required. The objective of this research is to develop a simple and efficient semi-automatic approach to generate a 3D city model for urban area using the fusion of LiDAR data and ortho-rectified imagery. These data sources provide efficiency for 3D building extraction. This approach uses both LiDAR and imagery data to delineate building outlines, based on fuzzy c-means clustering (FCM) algorithm. The third dimension is obtained automatically from the normalized digital surface model (nDSM) using spatial analyst tool. The 3D model is then generated using the multi-Faceted patch. The accuracy assessment for both height and building outlines is conducted referring to the ground truth and by means of visual inspection and different quantitative statistics. The results showed that the proposed approach can successfully detect different types of buildings from simple rectangle to circular shape and LOD2 (level of detail) is formed by including the roof structures in the model.

**Keywords:** 3D city model, LiDAR, Ortho-rectified aerial imagery, Data fusion, FCM, nDSM

### 1. Introduction

The way of representing earth has changed with the fast growth of technologies. Two-dimensional (2D) maps have turned from the traditional paper-based to digital forms and from planar 2D to a three-dimensional (3D) representation of objects. Among different cartographic products, 3D city models have shown to be valuable for several applications such as urban planning and management, flood simulation, land monitoring, mobile telecommunication, 3D visualization, solar radiation potential assessment, etc. (Tack et al., 2012).

With the fast advancement of spatial and spectral data acquisition systems in recent years, numerous approaches for generating 3D city model from various types of data such as high-resolution aerial images, airborne LiDAR data, terrestrial laser scanning, digital surface derives from stereo and multi-stereo matching and heterogeneous data sources have been presented (Partovi et al., 2013). In this regard, LiDAR and photogrammetry are receiving major attention due to their complementary characteristics and potential.

Nowadays, the algorithms that have been used for automatic extraction and visualization of 3D building archive various level of progress. However, extracting building boundary from LiDAR only is still challenging task where the horizontal spacing of the sample points is scattered. Therefore the need for supplementary data such as digital maps, high-resolution satellite imagery and ortho-imagery is necessary (Park et al., 2011). Different trials are carried out to generate 3D city model.

Ruijin (2004) reconstructed 3D building models from aerial imagery and LiDAR data. They used stereo aerial photographs to improve the geometric accuracy of the

building model. Complex buildings are reconstructed using the polyhedral building model in a data-driven oriented method. The proposed methodology has some limitations caused by the data used. For example, two individual buildings might be detected as one building if they are very close to each other. On the other hand, the algorithm may fail in ordering roof polygons in the correct sequence. Another type of limitation is from the modeling process itself. In this work, it is assumed that all buildings have rectangular footprints. Thus, non-rectangular footprints will be forced to have rectangular shapes.

Hongjian and Shiqiang (2006) presented a 3D building reconstruction approach based on aerial images and LiDAR data. First, an edge detecting algorithm combining Laplacian edge sharpening with the threshold segmentation was developed and employed to detect the edges and lines on the images. Then, a method using bi-direction projection histogram was used to determine the corner points of building and extract the contour of the building by searching and matching gradually. The four corners of the building can be extracted by combining the two directions according to the direction of the histogram. The heights of the building were calculated according to the Laser points within the building boundary. Because of the limitation of using the bi-direction histogram and the method to obtain the height of the roofs, the proposed method seems to be only suitable for buildings with rectangular shapes and flat roofs. It is very hard to apply it for complex building reconstruction.

Langue (2007) developed an object-oriented based method for 3D building extraction by integrating LiDAR data and aerial imagery. The object-oriented building model for 3D building extraction is developed by integrating data collection and construction methods, geometry and topology, semantics and properties as well

as data storage and management of 3D buildings into one model.

Arefi et al. (2008) proposed an automatic approach for reconstructing models in three levels of details. The building outlines are detected by classification of non-ground objects and building outlines are approximated by hierarchical filtering of minimum boundary rectangles (MBR) and RANSAC-based straight line fitting algorithm. Jarvis (2008) outlined the integration of photogrammetric and LiDAR data, within GIS, for the accurate reconstruction of a 3D realistic urban model in a semi-automated procedure.

Kada and McKindle (2009) developed an approach for automatic reconstruction of 3D building models from LiDAR data and existing ground plans by assembling building blocks from a library of parameterized standard shapes. This approach based on an algorithm to decompose the building shape into sets of nonintersecting cells, and for each cell, the roof top is reconstructed by checking the normal direction of digital surface model (DSM). Sirmacek et al. (2012) extracted 3D block models using an object-oriented approach based on data fusion from LiDAR and very high resolution (VHR) optical imageries.

Kwak (2013) developed a framework for fully-automated building model generation by integrating data-driven and model-driven methods as well as exploiting the advantages of images and LiDAR datasets. The major limitation is that it can model only the types of buildings which decompose into rectangles.

The main goal of this research is to outline a semi-automatic method for reconstructing 3D city model in a third level of details from both LiDAR data and ortho-aerial imagery. The proposed work was accomplished using a combination of the following software sets: 1) Erdas Imagine 9.2 for data preprocessing, and 2) a set of programs generated by the authors in Matlab environment for the rest of the work.

## 2. Study area and data sources

The area is a part of the university of New South Wales Campus, Sydney, Australia. It is largely urban area containing residential buildings, large campus buildings, a network of main roads as well as minor road, trees, and green areas. The multispectral imagery was captured by film camera on June 2005 at 1:6000 scale. The film was scanned in red, green and blue colour bands with 15  $\mu\text{m}$  pixel size (GSD of 0.096m) and radiometric resolution of 16-bit. On the other hand, LiDAR data were acquired over the study area on April 2005 and provided in ASCII format (easting, northing, heights, intensity and returns for first and last pulses). The LiDAR system used was the Optech ALTM 1225. Figures 1 and 2 show the multispectral imagery and the produced image from the LiDAR points respectively. The characteristics of aerial image and LiDAR data are provided in tables 1 and 2 respectively.

**Table 1: Characteristics of image datasets**

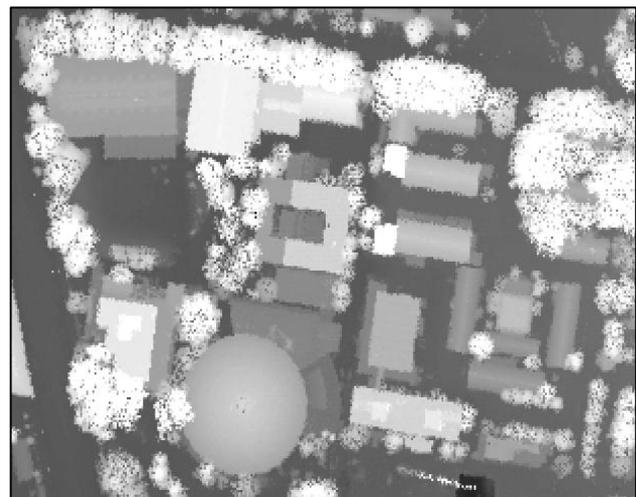
bands	Cell size (m)	Camera	Look Angle	
			along track	across track
RGB	0.096	LMK1000	$\pm 30^\circ$	$\pm 30^\circ$

**Table 2: Characteristics of LiDAR datasets**

Spacing/across track	1.15m
Spacing/along track	1.15m
Vertical accuracy	0.10m
Horizontal accuracy	0.5m
Density	1 Point/m <sup>2</sup>
Wavelength	1.047 $\mu\text{m}$
Altitude	1100m
Swath width	800m



**Figure 1: Ortho-rectified image of the test area**



**Figure 2: Digital Surface Model (DSM) generated from the original LiDAR pointcloud**

## 3. Methodology

This study proposes a semi-automatic method for generating 3D city model by integrating single aerial imagery and LiDAR data. This method is composed of five key steps. The five main procedures are discussed in the following sections. Figure 3 summarizes the workflow for the proposed techniques.

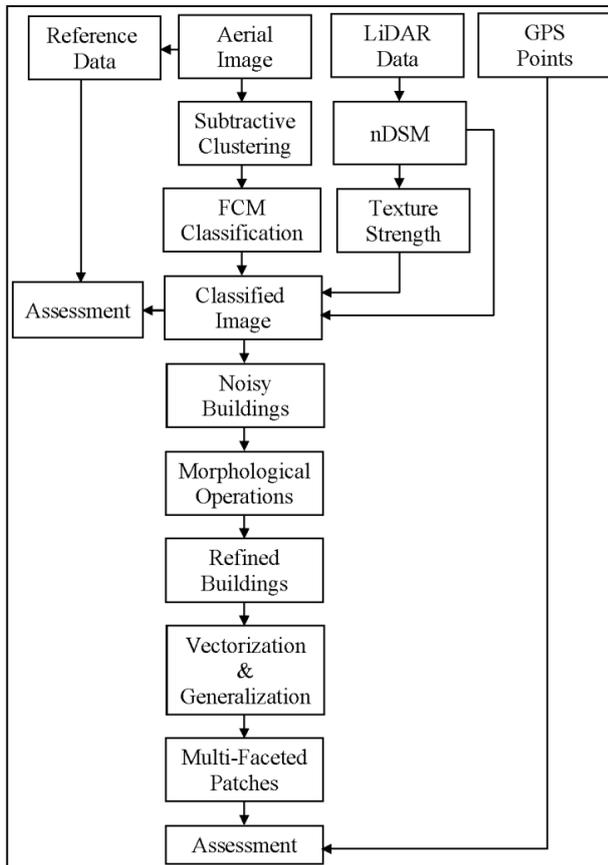


Figure 3: Ortho-rectified image of the test area

3.1. Data preparation

3.1.1 Image and LiDAR data co-registration: Image registration is the method of bringing different datasets into a single coordinate system. After multiple datasets acquired by different sensors having the same coordinate system it still requires some kind of additional pixel-to-pixel matching in order to ensure higher reliability in data fusion techniques. This kind of matching is known as image co-registration (Fikri, 2012).

The orthorectified image (already orthorectified by AAMHatch with a RMSE of 0.41m) is registered to the LiDAR intensity image using the projective transformation in ERDAS 9.2 environment. The Root Mean Square (RMS) error from the modeling process was 0.098m. Following the projective transformation, the image was resampled to 30cm x 30cm pixel size to match the resolution of the LiDAR data. The bilinear interpolation was used for resampling, which results in a better quality image and requires less processing time.

3.1.2 Generation of the nDSM: The nDSM represents the absolute heights, of non-ground objects such as buildings and trees, above the ground. First, the DSM was generated

from both the first and the last echoes. The DSM was then filtered to generate a digital terrain model (DTM) as shown in figure 4. In this case, the Tilted Surface Method (Salah, 2010) was used. In order to compensate for the difference in resolution between image and LiDAR data, DSM and DTM grids were interpolated to 30cm interval. Then, nDSM was generated. Finally, a height threshold of 3m was applied to nDSM to eliminating other objects such as cars as shown in Figure 5.



Figure 4: Digital Terrain Model (DTM) generated using the simple tilted plane filtering method

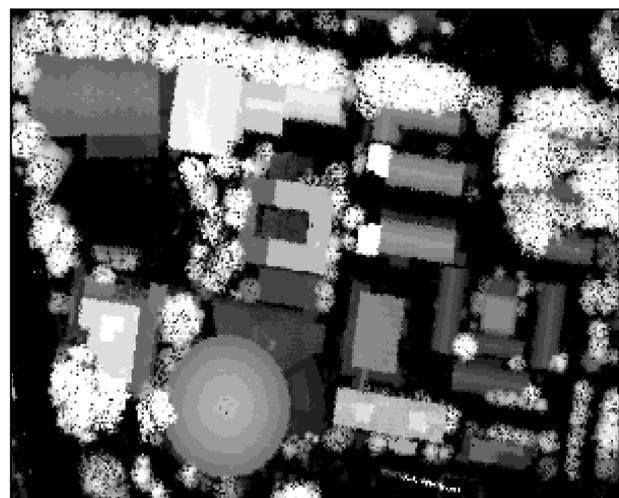


Figure 5: nDSM of the test area

3.1.3 Texture strength: Texture strength is based on a statistical analysis of the gray level gradients  $\nabla g(r, c)$ , which are the first derivative of the gray level function  $g(r, c)$ . The framework of the polymorphic texture strength based on the Förstner operator (Förstner and Gülch, 1987) has been applied. The gray level gradient  $\nabla g(r, c)$  can be computed from Equation 1:

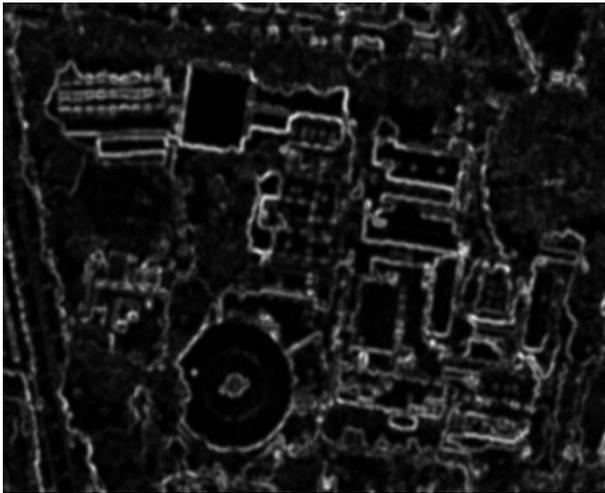
$$\nabla g(r, c) = \begin{bmatrix} \nabla g_r(r, c) \\ \nabla g_c(r, c) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} g(r+1, c) - g(r-1, c) \\ g(r, c+1) - g(r, c-1) \end{bmatrix} \quad (1)$$

From the gray level gradients  $\nabla g(r,c)$  of small windows,  $3 \times 3$  pixels, a measure  $W$  for texture strength is calculated as the average squared norm of the gray level gradients normalized by  $\sigma_n^2$  as shown in Equation 2:

$$W = L * \left\| \frac{1}{\sigma_n^2} \cdot \nabla g(r,c) \right\|^2 = L * \left( \frac{\Delta g_r^2 + \Delta g_c^2}{\sigma_n^2} \right) = L * \left( \frac{\Delta g_r^2 + \Delta g_c^2}{k \cdot \sigma_n^2} \right) \quad (2)$$

with  $L$  being a linear low pass filter, Gaussian filter with  $\sigma = 0.71$ ,  $k$  equals the squared sum of components of the convolution kernel. Thus,  $k = 0.5^2 + 0.5^2 = 0.5$ . The noise variance  $\sigma_n^2$  is equal to the square of the norm of the gray level grad  $\| \nabla g(r,c) \|^2$ .  $W$  is high in image windows containing large gray level differences.

For texture strength calculation, a window of  $3 \times 3$  pixels is placed over the top left  $3 \times 3$  block in the image and then the texture is calculated for all pixel values within that window. The texture value is then written to the central pixel of that window in a new raster layer. Then the window "moves" over one pixel and the process is repeated until all the pixels in the image have served as central pixels - except the ones around the outside. These edge pixels were filled in with the nearest texture calculation. Finally and since most texture calculations are not integers, images were linearly scaled to the full range for 8-bit data (0-255) as shown in figure 6.



**Figure 6: Texture strength of the nDSM**

**3.1.4 Reference data:** In order to evaluate the accuracy of the classifications, reference data were captured by digitizing buildings, trees, roads and ground in the ortho-photo as shown in figure 7. During this process, adjacent buildings that were joined but obviously separated were digitized as individual buildings; otherwise, they were digitized as one polygon. Roofs were first digitized and then shifted so that at least one point of the polygon coincided with the corresponding point on the ground. This is to overcome the horizontal layover problem of tall objects such as buildings.



**Figure 7: Reference data**

### 3.2. Image classification

**3.2.1 Fuzzy C-Means clustering (FCM):** A cluster can be defined as a group of pixels that are more similar to each other than to members of other clusters. In most of the clustering approaches, the distance measure used is the Euclidean distance. Thus, the distance measure is an important means by which the research can influence the outcome of clustering (Velmurugan and Santhanam, 2011).

Clustering can be divided into two main approaches: hard clustering and the other one is fuzzy clustering (Moertini, 2002). In hard clustering, data is partitioning into a specified number of mutually exclusive subsets (Babuska, 2001). In hard clustering method, the boundary between clusters is fully defined. However in many real cases, the boundaries between clusters cannot be clearly defined, where some patterns may belong to more than one cluster. In such cases, the fuzzy clustering method provides better results (Moertini, 2002). FCM is the most representative fuzzy clustering algorithms since it is suitable for tasks dealing with overlapping clustering.

In FCM, each data point belongs to a cluster to some degree that is specified by a membership grade (Bora and Gupta, 2014). This technique was introduced in 1973 and first reported in 1974 and subsequently improved in 1981 (Suganya and Shanthi, 2012). It provides a method of how to group data points that populate some multidimensional space into a specific number of different clusters. In Fuzzy clustering methods, the objects could belong to several clusters simultaneously with different degrees of membership, between 0 and 1 indicating their partial membership (Babuska, 2001).

This gives the flexibility to express that data points can belong to more than one cluster (Bora and Gupta, 2014). The clustering algorithm is performed with an iterative optimization of minimizing a fuzzy objective function ( $J_m$ ) defined as Equation (3).

$$J_m = \sum_{i=1}^c \sum_{k=1}^n (\mu_{ik})^m d^2(x_k, V_i) \quad (3)$$

where

$n$  = number of pixels

$c$  = number of clusters

$\mu_{ik}$  = membership value of  $i^{\text{th}}$  cluster of  $k^{\text{th}}$  pixel

$m$  = fuzziness for each fuzzy membership.

$x_k$  = vector of  $k^{\text{th}}$  pixel

$V_i$  = center vector of  $i^{\text{th}}$  cluster

$d^2(x_k, V_i)$  = Euclidean distance between  $x_k$  and  $V_i$

The membership ( $\mu_{ik}$ ) can be estimated from the distance between  $k^{\text{th}}$  pixel and center of  $i^{\text{th}}$  cluster as follows:

$$\begin{cases} 0 \leq \mu_{ik} \leq 1 & \text{for all } i, k \\ \sum_{i=1}^c \mu_{ik} = 1 & \text{for all } k \\ 0 < \sum_{k=1}^n \mu_{ik} < n & \text{for all } i \end{cases} \quad (4)$$

The center of cluster ( $V_i$ ) could be calculated by equations (5) and the membership value ( $\mu^{ik}$ ) could be calculated by equations (6) as follow.

$$V_i = \frac{\sum_{k=1}^n (\mu_{ik})^m x_k}{\sum_{k=1}^n (\mu_{ik})^m}, 1 \leq i \leq c \quad (5)$$

$$\mu_{ik} = \left[ \sum_{j=1}^c \left( \frac{d(x_k, V_i)}{d(x_k, V_j)} \right)^{\frac{2}{m-1}} \right]^{-1}, 1 \leq i \leq c, 1 \leq k \leq n \quad (6)$$

$J_m$  can be minimized by iteration through equations (5) and (6). The first step of the iteration is to initialize the following parameters: a fixed  $c$ , a fuzziness parameter ( $m$ ), a threshold of convergence  $\varepsilon$ , and an initial center for each cluster, then computing  $\mu_{ik}$  and  $V_i$  using Equations (5) and (6) respectively. The iteration is stop when the change in  $V_i$  between two iterations is smaller than  $\varepsilon$ . At last, each pixel is classified into a combination of memberships of clusters.

Several parameters must be specified before using the FCM algorithm which include: the number of clusters,  $c$ , the 'fuzziness' exponent,  $m$ , the termination tolerance,  $\varepsilon$ , and the norm-inducing matrix,  $A$ . Moreover, the fuzzy partition matrix,  $U$ , must be initialized (Babuska, 2001).

**3.2.2 Subtractive clustering:** Fuzzy C-Means algorithm requires the analyst to pre-specify the number of cluster centers and their initial locations. The quality of the results depends strongly on the number of cluster centers and their initial locations. Chiu (1994) proposed an effective algorithm, called the subtractive clustering, for estimating the number and initial location of cluster centers. By using this method, the computation is simply proportional to the number of data points and independent of the dimension problem as shown in equation 7 (Moertini, 2002). For a problem of  $c$  clusters and  $m$  data points, the required number of calculations is:

$$N = m^2 + (c-1)m \quad (7)$$

1<sup>st</sup> clustreminderclusters

Consider a group of  $n$  data points  $\{x_1, x_2, \dots, x_n\}$ , where,  $x_i$  is a vector in the feature space. Assume that the feature space is normalized so that all data are bounded by a unit hypercube. As well, consider each data point as a potential cluster center and define a measure of the point to serve as a cluster center. The potential of  $x_i$  denoted as  $P_i$  is given in equation 8.

$$\sum_{j=1}^n \exp\left(-\frac{\|x_i - x_j\|^2}{(r_a/2)^2}\right) \quad (8)$$

Where  $r_a$  is a positive constant defining a neighbourhood radius  $\| \cdot \|$  denotes the Euclidean distance. A data point that has many neighbouring data points will have a higher potential value and the points outside will have little influence on its potential. The first cluster center  $c_1$  is chosen as the point with the highest potential. The potential of  $c_1$  is referred to as  $PotVal(c_1)$ . The potential of each data point  $x_i$  is then revised as follows:

$$P_i = P_i - PotVal(c_i) \exp\left(-\frac{\|x_i - c_i\|^2}{\left(\frac{r_b}{2}\right)^2}\right) \quad (9)$$

To avoid obtaining closely spaced cluster centers,  $r_b$  is usually set to 1.5  $r_a$ . The data points near the first cluster center will greatly reduce their potential and will unlikely be selected as the next center. From equation 9 the potential of all data points will be reduced, after that the point with the highest potential is selected as the second center. After the  $k^{\text{th}}$  cluster center  $c_k$  is determined, the potential is revised as follows:

$$P_i = P_i - PotVal(c_k) \exp\left(-\frac{\|x_i - c_k\|^2}{\left(\frac{r_b}{2}\right)^2}\right) \quad (10)$$

where

$c_k$  = the location of the  $k^{\text{th}}$  cluster center

$PotVal(c_k)$  = potential value.

The process proceeds until the stopping criterion is reached. From the clustering process, two conclusions can be drawn: 1) a point with relatively high potential has more chance to be selected as center than less potential point; 2) Cluster centers are selected only from the data points even if the actual cluster centers are in the dataset or not. (Chen et al., 2008). The Strengths of the subtractive clustering are: 1) reduces the time complexity; and 2) results are fixed and has no random cluster value. On the other hand, accuracy is less and cautious about choosing the neighbour radius (Leela et al., 2014).

### 3.3. Post-processing

**3.3.1 Morphologic operations:** Morphologic operations have been applied to separate objects in the image from the background. The basic operations of binary morphology are: erosion, dilation, opening, and closing. A dilation operation enlarges a region, while erosion makes it smaller. An opening operation (erosion followed by

dilation) can get rid of small portions of the region that jut out from the boundary into the background region. A closing operation (dilation followed by erosion) can close up internal holes in a region and eliminate bays along the boundary (Shapiro and Stockman, 2011).

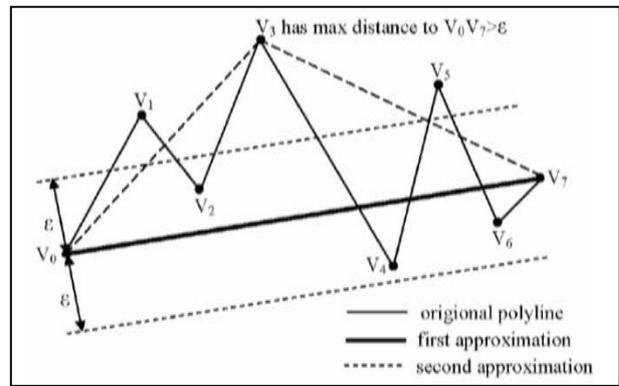
For clarity, the small buildings were merged into larger ones or deleted according to a 1m distance and 30m<sup>2</sup> area thresholds. A certain building was retained if it was larger than 30m<sup>2</sup> and/or adjacent to another building by a distance less than 1m. The area threshold represents the expected minimum building size, while the distance threshold was set to 1m to fill in any holes or gaps produced by the classification process. Building borders were then cleaned by removing regions that were smaller than 5 pixels in size and that were connected to the building border. Cleaning thresholds less than 5 pixels may leave the original buildings uncleaned, while thresholds larger than 5 pixels may remove parts of the original buildings. The results are the detected buildings without holes or any noisy features.

**3.3.2 Vectorization and generalization:** In order to extract building boundaries, the smoothed binary image is converted from raster to vector format. After that, the obtained boundaries need more processing to overcome the problem of irregularities and to adjust the rectangularity of the polygons. One of the most common used generalization algorithms is The Ramer–Douglas–Peucker algorithm (RDP). This algorithm reduces the number of points in a curve that is approximated by a series of points. The first form of the algorithm was suggested in 1972 and then modified by Douglas and Peucker (1973).

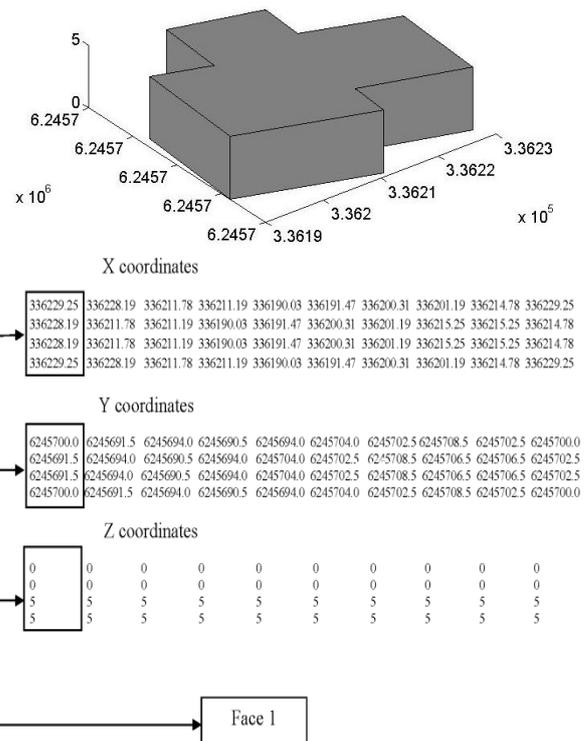
This approach automatically marks the first and last points to be kept, and then it finds the furthest point from the line segment between the first and last points as end points. If the vertex is closer than the tolerance ( $\epsilon$ ) to the line segment then any points not currently marked to be kept can be discarded without the simplified curve being worse than  $\epsilon$ . If the vertex that is furthest away from the line segment is greater than  $\epsilon$  from the approximation then that point must be kept. The algorithm recursively calls itself with the first point and the worst point and then with the worst point and the last point (which includes marking the worst point being marked as kept). When the recursion is completed a new output curve can be generated consisting of all (and only) those points that have been marked as kept (Douglas and Peucker, 1973).

**3.4. Three dimensional model construction**

For the construction of the three dimensional model, Multi-Faceted Patches are used. In this regard, x, y, and z coordinate of the faces of a given building are specified as matrix. MATLAB draws one face per column, producing a single patch with multiple faces as shown in figure 9(The MathWorks, 2015).



**Figure 8: Smoothing a line segment with the Douglas–Peucker algorithm (Douglas and Peucker, 1973).**

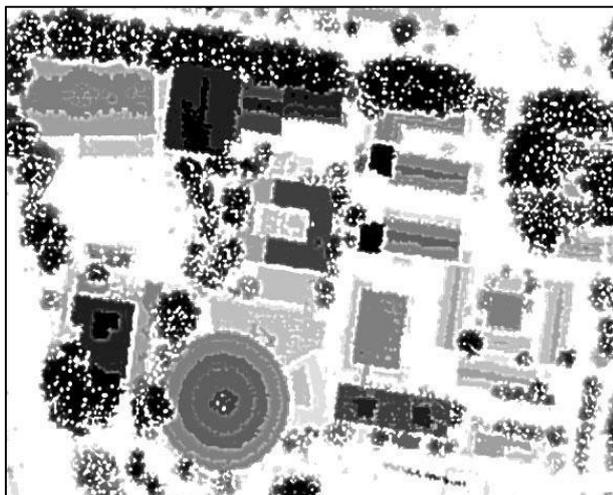


**Figure 9: The concept of multi-faceted patches**

**4. Analysis and results**

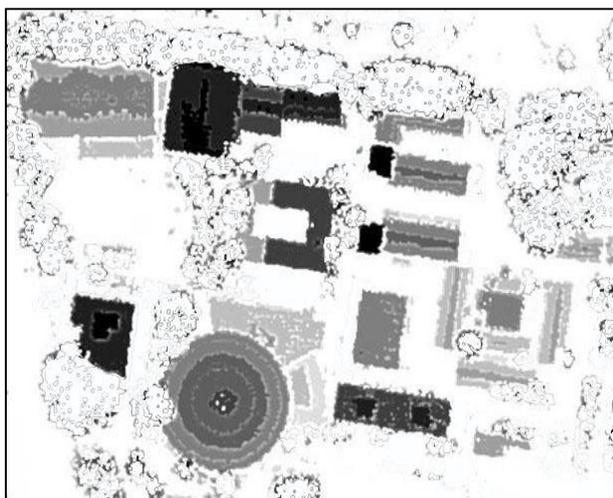
To initialize FCM algorithm, the parameters are set to the following values: the total number of clusters  $c$  is initialized as 13 (as obtained by the subtractive clustering), the maximum number of iteration as 100, the exponent for  $\mu_{ik}$  as 2.0 and a minimum improvement  $\epsilon$  of  $1e^{-6}$ . The clustering process terminates when the maximum number of iterations is reached, or when the objective function improvement between two consecutive iterations is less than the minimum amount of improvement specified. Figure 10 shows the FCM output.

The obtained overall classification accuracy was 87.84%, while the per-class accuracies were 83.51%, 89.06%, 82.83% and 92.33% for buildings, trees, roads and grass respectively.



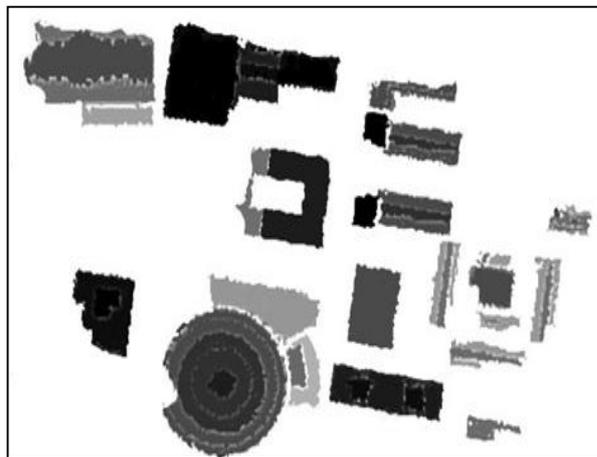
**Figure 10: Classified image using fuzzy c-means clustering**

In order to extract buildings from the classified image, the classified image was compared with the nDSM and texture strength images. Digital values of the classified image are converted to 1 (background) if it corresponds to 0 in the nDSM and/or higher value (over 0.5) in the texture strength image. Otherwise, the pixel value is kept as it is. The result is a building image with noisy features as shown in figure 11.



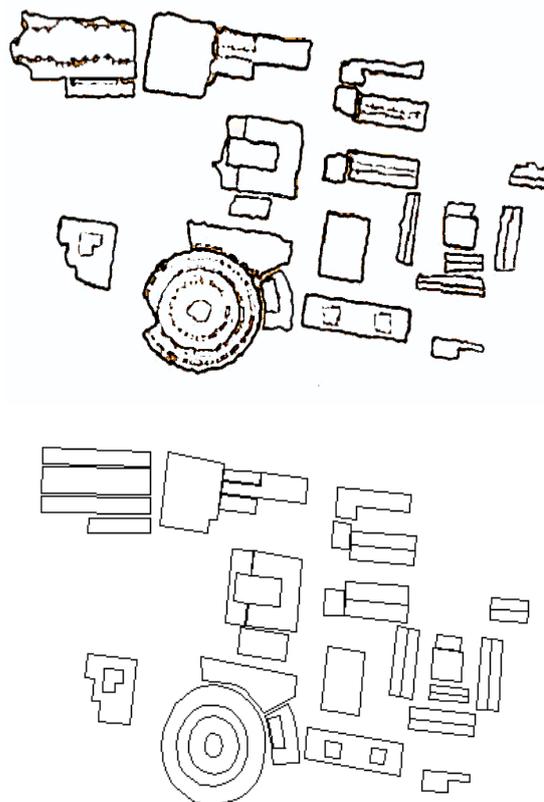
**Figure 11: Classified image using fuzzy c-means clustering**

Morphologic operations were then applied to merge small buildings into larger ones and fill in holes according to the specified 1m distance and 30m<sup>2</sup> area thresholds. Building borders were then cleaned according to the specified 5 pixels threshold. The result was an image that represents the detected buildings with a considerable lower degree of noisy features as shown in Figure 12.



**Figure 12: The final detected buildings**

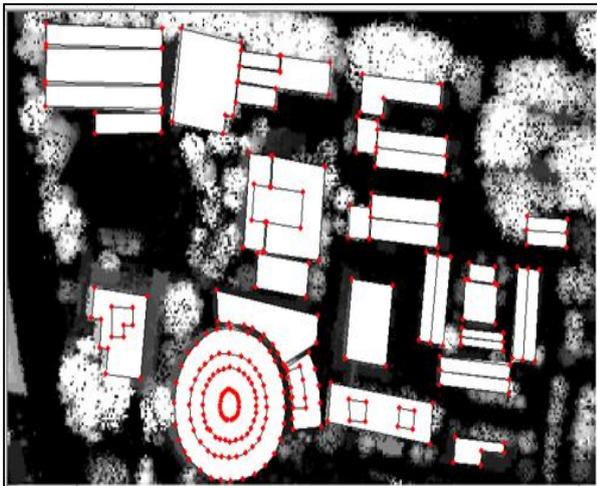
The smoothed image is then converted from raster to vector format to extract building boundaries. The Ramer–Douglas–Peucker algorithm (RDP) was used to overcome the problems of irregularities and adjust the rectangularity of the polygons. The tolerance was initially specified equal to the pixel size of the data. If the output still contains too much detail, then the tolerance can be doubled and so on. Similarly, if the output lines do not have enough detail, the tolerance can be halved. Figure 13 shows the extracted buildings before and after the simplification and adjustment of the rectangularity.



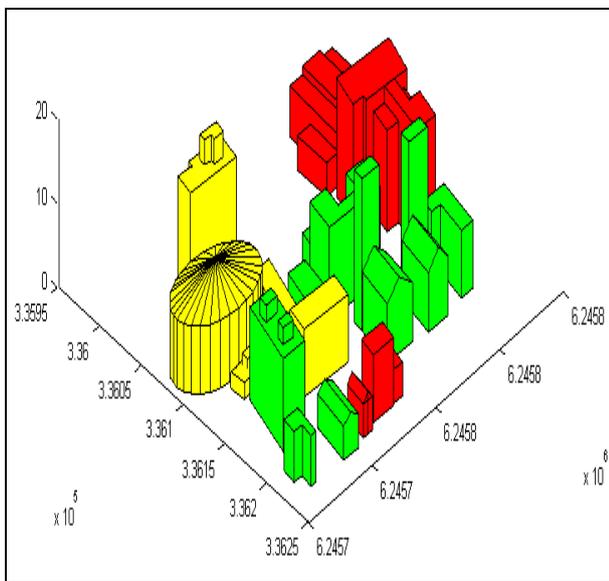
**Figure 13: Buildings map before and after adjusting the rectangularity**

In order to construct 3D model, two main items must exist: building outlines, and height information. In the proposed

methodology and to determine the heights of the buildings, a set of random sample points (constrained by building footprints) are generated at the corners of each building as shown in figure 14. The result is a feature class containing groups of points. Elevation information extracted from elevation surface can be added to each point as an attribute. The MATLAB code is then used to construct the 3D model as shown in figure 15.



**Figure 14: Generation of a set of random sample points (constrained by building footprints)**



**Figure 15: buildings extracted from the study area**

In order to evaluate the planimetric accuracy of the resulted vector map, three GCPs were determined by field surveys. The GCPs were selected to be evenly distributed throughout the study area as shown in Figure 16, and a comparison was carried out between GPS observations and the extracted building data coordinates with a RMSE of 0.51m as shown in Table 3. The vertical accuracy of the constructed 3D building model is within 15- 20 cm which matches with a vertical accuracy of LiDAR.



**Figure 16: Distribution of the GPS control points**

**Table 3: The accuracy estimate of the building vectorization process**

Point	$\Delta_E$ (GPS – map)	$\Delta_N$ (GPS – map)	$\sqrt{\Delta E^2 + \Delta N^2}$
1	0.47	-0.46	0.6576
2	0.14	-0.03	0.1431
3	0.32	0.74	0.8062
Mean	0.31	0.41	
RMS	0.51		

**5. Conclusion and future work**

This paper discusses the fusion of LiDAR and aerial orthorectified imagery data for the construction of 3D city models. In the proposed approach data classification was carried out with an overall accuracy of almost 83.51%. The horizontal accuracy of building outlines reached 0.51 m, while vertical accuracy ranged between 15- 20 cm. As a future step and in order to maximize the benefits of the proposed method, the authors aim at increasing the degree of automation and level of details. Due to the limitation of LiDAR data on hand, current work has been done with only one data set. In the future it is planned to process more and larger test areas in order to confirm the results found so far.

**Acknowledgements**

The Authors wish to acknowledge AAMHatch for the provision of the UNSW dataset.

**References**

Arefi, H., J. Engels, M. Hahn and H. Mayer (2008). Levels of detail in 3D building reconstruction from LIDAR data. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. 37, 485–490

Babuska, R. (2001). Fuzzy and neural control - DISC course lecture notes. Faculty of Information Technology and Systems, Delft University of Technology, Delft, the Netherlands.

- Bora, D. and A. Gupta (2014). A comparative study between fuzzy clustering algorithm and hard clustering algorithm. *International Journal of Computer Trends and Technology (IJCTT)* – Vol 10, No. 2, 108 – 113.
- Chen, J., Z. Qin, Z. and J. Jia (2008). A weighted mean subtractive clustering algorithm. *Information Technology Journal* 7(2), 356-360.
- Chiu, S. (1994). Fuzzy model identification based on cluster estimation. Rockwell Science Center Thousand Oaks, California, 91360.
- Douglas, D. and T. Peucker (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.
- Fikri, A. (2012). A methodology for processing raw LiDAR data to support urban flood modeling framework. Ph.D. Dissertation, Institute for Water Education, Delft University of Technology.
- Förstner, W. and E. Gülch (1987). A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proceedings of the ISPRS Intercommission Workshop on Fast Processing of Photogrammetric Data*, 2–4 June 1987, Interlaken, Switzerland, 281–305.
- Hongjian, Y. and Z. Shiqiang (2006). 3D building reconstruction from aerial CCD image and sparse laser sample data. *Optics and Lasers in Engineering* 44(6): 555-566
- Jarvis, A. (2008). Integration of photogrammetric and LiDAR data for accurate reconstruction and visualization of urban environments. MSc. Dissertation, Department of Geomatics Engineering, University of Calgary.
- Kada, M. and L. McKinley (2009). 3D building reconstruction from LiDAR based on a cell decomposition approach. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38, 47–52.
- Kwak, E. (2013) Automatic 3D building model generation by integrating LiDAR and aerial images using a hybrid approach. Ph.D. Dissertation, Department of Geometrics Engineering, University of Calgary.
- Langue, W. (2007) Object-oriented model based 3D building extraction using airborne laser scanning points and aerial imagery. M.Sc. Dissertation, Institute of Geo-Information Science and Earth Observation, ITC.
- Leela, V., K. Sakthi priya and R. Manikandan (2014). Comparative study of clustering techniques in Iris data sets. *World Applied Science Journal*, 29, 24-29.
- Moertini, V.S. (2002). Introduction to five data clustering algorithms. *Integral*, Vol 7, No 2, 87 – 96.
- Park, H., M. Salah and S. Lim (2011). Accuracy of 3D models derived from aerial laser scanning and aerial ortho-imagery. *Surveying Review*, 43, 320, 109-122.
- Partovi, T., H. Arafai, T. Kraub and P. Reinartz (2013). Automatic model selection of 3D reconstruction of buildings from satellite imagery. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*. Volume XL-1/W2. Tehran, Iran.
- Ruijin, Ma. (2004). Building model reconstruction from LiDAR data and aerial photographs. PhD thesis, The Ohio State University.
- Salah, M. (2010). Towards automatic feature extraction from high resolution digital imagery and LiDAR data for GIS applications. Ph.D. Dissertation, Department of Surveying Engineering, University of Benha.
- Shapiro, L. and G. Stockman (2011). *Computer vision*. Prentice Hall, New Jersey.
- Sirmacek, B., H. Taubenboeck and P. Reinartz (2012). A novel 3D city modelling approach for satellite stereo data using 3D active shape models on DSMS. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Volume XXXIX-B3, 2012-XXIII ISPRS Congress, Melbourne, Australia.
- Suganya, R. and R. Shanthi (2012). Fuzzy C- means algorithm- A review. *International Journal of Scientific and Research Publications*, 2(11), November 2012 Edition, ISSN 2250-3153.
- Tack, F., G. Buyuksalih and R. Goossens (2012). 3D building reconstruction based on given ground plan information and surface models extracted from spaceborne imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, vol (67), 52-64.
- The MathWorks (2015). 3-D visualization. The MathWorks, Inc. [http://www.mathworks.com/help/pdf\\_doc/matlab/visualize.pdf](http://www.mathworks.com/help/pdf_doc/matlab/visualize.pdf)
- Velmurugan, T. and T. Santhanam (2011). A comparative analysis between k-medoids and fuzzy C-means clustering algorithms for statistically distributed data points. *Journal of Theoretical and Applied Information Technology*, Vol 27 No 1, 19 – 30.