

A New Approach for 3D Range Image Segmentation using Gradient Method

¹Dina A. Hafiz, ¹Walaa M. Sheta, ²Sahar Bayoumi and ¹Bayumy A.B. Youssef
¹Department of Computer Graphics and Virtual Reality, Informatics Research Institute,
Mubarak City for Scientific Research and Technology,
Borg El Arab, Alexandria, Egypt
²Department of Information Technology,
Institute of Graduate Studies and Research, Alexandria, Egypt

Abstract: Problem statement: Segmentation of 3D range images is widely used in computer vision as an essential pre-processing step before the methods of high-level vision can be applied. Segmentation aims to study and recognize the features of range image such as 3D edges, connected surfaces and smooth regions. **Approach:** This study presents new improvements in segmentation of terrestrial 3D range images based on edge detection technique. The main idea is to apply a gradient edge detector in three different directions of the 3D range images. This 3D gradient detector is a generalization of the classical sobel operator used with 2D images, which is based on the differences of normal vectors or geometric locations in the coordinate directions. The proposed algorithm uses a 3D-grid structure method to handle large amount of unordered sets of points and determine neighborhood points. It segments the 3D range images directly using gradient edge detectors without any further computations like mesh generation. Our algorithm focuses on extracting important linear structures such as doors, stairs and windows from terrestrial 3D range images these structures are common in indoors and outdoors in many environments. **Results:** Experimental results showed that the proposed algorithm provides a new approach of 3D range image segmentation with the characteristics of low computational complexity and less sensitivity to noise. The algorithm is validated using seven artificially generated datasets and two real world datasets. **Conclusion/Recommendations:** Experimental results showed that different segmentation accuracy is achieved by using higher Grid resolution and adaptive threshold.

Key words: Laser scanning, point cloud, edge detection, normal vector estimation, Least Square Fitting (LSF), Principal Component Analysis (PCA), gradient method, range image segmentation, Terrestrial Laser Scanning (TLS), Airborne Laser Scanning (ALS)

INTRODUCTION

Professional laser scanning technologies currently offer accurate and rapid acquirement of range data from real world objects. High speed graphics cards have also become available worldwide at a relatively low cost. These two factors have made available the growth of 3D imaging systems. These systems collect a series of 3D points called "point clouds" which provide the basis for surface reconstruction or modeling. These massive unstructured 3D point clouds are randomly distributed with different local densities, especially in the presence of random noisy points.

Laser scanning systems are currently used in different applications such as civil engineering, culture heritage, industrial monitoring and virtual environments. However, the production of complete

scene from several scans is extremely difficult and needs high processing power. Segmentation, data reduction and data management of acquired point clouds are necessary for the downstream applications to be carried out at a practical computational cost.

In general, segmentation of point cloud means partitioning of the data in the object space in order to create meaningful, coherent and connected subsets. Each connected subset is supposed to include points with similar attributes and represents a surface, edge or parts of the objects.

By comparison to 2D digital imagery, TLS gives explicit 3D information that enables the rapid and accurate capture of the geometry of complex objects such as buildings. Point clouds require a great deal of pre-processing and data management before applying the segmentation algorithms. According to (Boulaassal

Corresponding Author: Walaa M. Sheta, Department of Computer Graphics and Virtual Reality, Informatics Research Institute, Mubarak City for Scientific Research and Technology, Borg El Arab, Alexandria, Egypt

et al., 2009) the 3D point clouds are not immediately compatible with mathematical models, that is to say no planar surfaces and no straight edges are directly provided in the digital model. The capability of carrying out useful analysis is rather limited with unstructured points. A set of point cloud data implies abundant implicit information which can be turned into explicit through some methods of pre-processing and segmentation.

The main contribution of this study is the segmentation of 3D range images using $3 \times 3 \times 3$ gradient edge detector that is applied in three different directions. The 3D range image is represented by a 3D gridded structure. This 3D gradient detector is a generalization of the classical sobel operator used with 2D images. It is based on the differences of normal vectors or geometric locations in the coordinate directions. Our algorithm focuses on extracting important linear structures such as doors, stairs and windows from terrestrial point cloud using $3 \times 3 \times 3$ gradient edge detectors. These structures are common in indoors and outdoors in many environments.

Awwad *et al.* (2010) showed that the extraction of object features from massive unstructured 3D point clouds with different local densities, especially in the presence of random noisy points, is not a trivial task even if that feature is a planar surface. Boulaassal *et al.* (2009) showed the 3D point clouds are dependent on many factors, such as the roughness of different surfaces, the registration step and the TLS instrument resolution. These factors affect the final results, causing the 3D point cloud even for a planar surface to appear not to be flat. In real scans the planer surfaces such as building facades have non-zero thickness or finite thickness.

Segmentation of the linear structures is very essential in many applications like mobile robot navigation and the inspection of industrial parts. The segmented linear structures support the robot's choice of correct paths while navigation (Cang and Hegde, 2009). Another important application that uses segmentation as an essential pre-processing step is 3D building reconstruction and modeling which is a very active topic in the field of computer vision and virtual reality. In recent years, terrestrial laser scanning and Airborne Laser Scanning have been proven as a valuable source for building facade reconstruction. Edges detection for the outer surface of the building's point cloud and their features like windows, outer doors, roof, outer edges of walls and stairs will be a good representation of the model outline.

Many attempts have been made to segment planar surfaces from point clouds and extract their linear features. In many cases these scans were acquired by

TLS instruments (Boulaassal *et al.*, 2009; Boulaassal *et al.*, 2010). Most authors used points acquired by ALS. Many researchers made building facade reconstruction by fusing segmented terrestrial laser points and segmented images, the integration of the 2D outlines with the 3D outlines will lead to a more reliable and automated extraction of 3D features and will give an accurate model representation of the real point cloud data of the buildings.

The proposed algorithm works directly on the raw laser scanner point clouds without converting it to another form i.e., mesh generation. This algorithm requires a pre-processing that includes generation of virtual 3D-grid structure to handle a large amount of unordered sets of point data and establish relationships between neighbors. The final constructed virtual 3D-grid will contain an $N \times M \times L$ cells which are equally spaced. The virtual 3D-grid structure enables us to segment the 3D range image by convoluting the computed normal vectors with $3 \times 3 \times 3$ gradient edge detectors. This virtual 3D-grid is similar in its shape to the volumetric structure used in the representation of the medical images.

Extensive experiments have been performed on artificial datasets and real world datasets captured by Riegl LMS-Z620 <http://www.riegl.com> of informatics institute building (IRI) in Mubarak City for Science and Technology (MUCSAT), Alexandria, Egypt. The results show that the proposed algorithm works well on various types of artificial datasets ranging from simple to complicated ones. The results show that the algorithm is efficient in terms of accuracy and sensitivity to noise.

Literature review and related work: Since range images convey more 3D information of real world objects than intensity images and because the evolution of 3D laser scanners allows digitalizing a real scene fast and efficiently, 3D range image analysis become one of the most important areas in computer vision and has been applied to many fields. A number of range image segmentation techniques have been proposed in the next discussion.

Segmentation is the process of aggregating points with similar attributes or portioning a point cloud data into smaller coherent connected regions. Edge segmentation refers to extraction of the point cloud connected edges and corners; it attempts to detect the discontinuities in the surface that form close boundaries of different segments. The segmentation techniques for range images could be generalized into two main categories: Region based segmentation, where points are classified into Regions and Edge based segmentation, where the region boundaries are detected.

Hybrid methods also exist which are a mix of the region and edge techniques. Edge based techniques obtain step edges and roof edges respectively by locating the points with depth discontinuities and normal inconsistency then link them to divide the image into different regions. Region based techniques select a set of seed regions and grows these regions by iteratively merging neighborhood points with similar properties.

Edge based techniques play a key role in 3D range image segmentation as region based techniques suffer from a number of problems such as distortion of region boundaries, dependency on the initial seed selection, over-segmentation and is time consuming. While edge based techniques are more suitable to preserve the border locations but their results still need further refinement and the difficulty in obtaining closed segments is its main drawback, which is solved by applying edge thinning and closing operations. However, edge based techniques can be used in combination with region based segmentation techniques to obtain better results.

Edge-based methods attempt to detect the discontinuities in the surface that form close boundaries of different segments. In range images, there are three categories of edges to be extracted: jump edges / step edges, crease edges / roof edges / fold edges and smooth edges. Jump edges are found where range values (in depth or height values) are discontinuous, such as when one object occludes another. Jump edges / Step edges are used to segment point clouds into surface patches from different objects. In post-processing stage, the surface patches are grouped into identifiable objects, such as ground and buildings.

Crease edges/Roof edges occur where two surfaces meet; such edges are characterized by discontinuities in surface normal's at the intersection line which make an angle greater than the given threshold. Crease edges/Roof edges are used to further segment points into adjacent planar regions, which are inside the surface patch defined by the jump edges. Crease edges/Roof edges are usually detected by threshold the difference in surface normal's or the maximum curvature. Crease edges/Roof edges prove difficult to detect as they do not correspond to large range variation and therefore are usually inclined to be suppressed along with noise. The smooth edge is identified by continuity of the surface but discontinuity of the curvature. This particular type of edge is most difficult to detect as it does not correspond to rapid range variation. Most edge detection techniques for range image data do not attempt to extract smooth edges. Other known techniques distinguish only between jump and crease edges.

Most of the existing range image based edge detection algorithms base their detection criterion on depth or curvature changes. However, depth or curvature changes do not have been sensitivity on detecting roof or crease edges. Using normal changes as a detecting criterion, on the contrary, can easily detect the existence of a roof edge even the change across a boundary is slight small. Coleman *et al.* (2010a) used gradient operators for feature extraction and characterisation in range images. Coleman *et al.* (2010b) applied laplacian operators for edge detecting for range data.

Region based techniques are used for the extraction of object's surface but they suffer from selecting the start /seed points and the unstructured behavior of the operator. Also these techniques rely just on the range value. Surface fitting methods are good in extraction the geometrical solid objects such as plane, cylinder and coin. The drawback of the geometric surface fitting methods is that these methods are very restricted to simple structured scenes. Finite element surface fitting methods like mesh generation are independent of the scene and object shapes and generate topological relationships. These methods suffer from their computational complexity and their implementation not as easy as the previous named methods. Mazouzi and Batouche (2007) used randomized region growing and bayesian edge regularization.

Some techniques used gradient operators for edge detection. Several operators have been proposed to detect 3D edges. 3D generalization of the Hueckel operator based on determining a best fitting step function, they proposed a mathematical model to determine the optimal 3D gradient operators using a functional analysis and their theory was a generalization of the 2D sobel operator. The Zucker-Hummel operator used a radial functions that smooth the calculated gradient which have a positive effect in noisy data. The optimal gradient operator described in Zucker-Hummel study is a $3 \times 3 \times 3$ anti-symmetric operator that is applied in 3 different directions.

The 3D surface operators proposed by Zucker and Morgenthaler belong to facet model approaches and made use of polynomial surface fitting. Chowdhury *et al.*, (2006) made an evaluation of 3D gradient filters like the $3 \times 3 \times 3$ Zucker-Hummel operator and $5 \times 5 \times 5$ sobel operator for estimation of the surface orientation in medical images.

Segmentation using 2D-3D sobel operator: For 3D images, the approaches of segmentation are usually extension of the 2D approaches to 3D in which the region based approaches extract homogeneous volumes

instead of areas and the edge based approaches identify surfaces forming volume boundaries. 3D edge operator is generally not a simple extension of 2D operators, 3D sobel operators have been generalized to the 3D space by using the separability property in the frequency domain. The coefficients of the masks can be computed by Taylor's development of the partial derivatives. Monga *et al.* (1991) investigated theoretical and experimental reasons why 3D edge detection is of greater interest than 2D edge detection in Computed Tomography (CT) 3D images. Monga and Deriche pointed out that 3D edge detection compared with 2D edge detection yields to better immunity to the noise, better estimation of the gradient magnitude and accurate computation of the 3D gradient.

For segmentation of images using sobel operator for 3D volumes, there are two approaches.

First approach: 2D techniques are used on 3D data (3D volumes) which involves first segmenting each 2D image in the sequence of slices and then combining the results in some manner to get the 3D segmentation. The segmentation is applied on contours, or 2D slices, of the data apart from directly utilizing all of the information in the volume data. Such 2D methods give rise to various ambiguity and connectivity problems between the 2D segmented planes. That is actually what happened in medical volumes like Computed Tomography (CT), magnetic resonance imaging (MRI). They are typically stored as a sequence of parallel 2D slices. So for 2D images pixel (x, y) become a 3D image voxel (x, y, z) namely volume pixel. A voxel represents a quantity of 3D data just as a pixel represents a point or cluster of points in 2D data. This approach could successfully detect all the edges in the 2D slice however, it is very difficult to find the correlation between edges, region of interest in the successive slices, as the third dimension is not taken into account while calculations of gradients.

Second approach: The 3D sobel edge detector is operating on three adjacent slices (a subset of the image solid). The results of edge detection are put in the central slice. It works directly with the volume data as a whole, thereby utilizing all of the 3D information making use of the 26 neighbors not only the 8 neighbors (Chi *et al.*, 2008) The utilization of the 26 voxels has several good properties both in the magnitude response and in considering direction responses.

MATERIALS AND METHODS

Pre-processing: The specification of an appropriate data structure and a proper data processing

methodology are both necessary if the intended efficiency in processing enormous amounts of TLS data is to be realized. This stage does not change the format of the scanned point clouds to another form like meshes.

The segmentation technique is applied to a virtually arranged point cloud in a coordinate system suitable to segmentation. Most of the existing segmentation methods are based on 2.5D range image or Triangular Irregular Network (TIN) model. Unlike the 2.5D range image and TIN model; the point clouds do not explicitly represent topology information. As a result, most existing segmentation methods encountered two difficulties. First, converting data from irregular 3D point clouds to other models usually leads to information loss which is particularly a serious drawback for range image based algorithms. Second, the high computation cost of converting a large volume of point data is a considerable problem for any large scale TLS applications. Therefore, it is preferred that segmentation of TLS is performed on point clouds directly.

The pre-processing stage aims to minimize the operations done on the point clouds. Due to the enormous number of points in real world datasets, the searching of particular locations of points cannot be completed in a constant time if the scanned points are not arranged on a proper data structure as they are irregularly distributed geometrically. Also, the geometric position of each point (X, Y and Z) and the estimated surface normal's could be affected by noisy points. These noisy points directly affect the segmentation results. Therefore, massive unstructured and noisy point clouds can lead to bad segmentation (over-segmentation, under-segmentation or no segmentation).

The pre-processing stage is used to reduce an amount of noisy points which have influence on the segmentation results. The virtually 3D-grid structure is used for data storage, fast search of adjacent points, organizes the huge number of point clouds based on properties of the point's distribution and estimation of the normal vectors.

Finding bounding box: Bounding box volume (Bbox) is a very simple approximation of the scanned object. The size of an object can be indicated by the dimensions of the Bbox, namely the smaller parallelepiped that contains the object. As the bounding box became tighter, the processing time decreases and the dimensions of the 3D-grid decrease. Bounding volumes are used to optimize and speed up many graphic algorithms. The dimensions of this box could be found by scanning the 3D surface and finding the points with the minimum/maximum coordinates along each

dimension. Obviously the dimensions of the Bbox depend on the orientation of the scanned surface.

Existing types of bounding boxes are: axes parallel bounding boxes (Bboxes), bounding spheres, bounding slabs, arbitrarily transformed Bboxes. This study uses the PCA method to determine the tighter bounding box. PCA finds a best approximating plane p_0 . After the parameters of plane p_0 are estimated, rotation matrix R and translation vector T are constructed. Consequently that data in scan is transformed into a coordinate system suitable to segmentation.

Determination of the translation and rotation parameters: Any movement of an object in the 3D space is a superposition of a 3D rotation and a 3D translation. Consequently, the movement of a point $p = (x, y, z)$ on the object to $p' = (x', y', z')$ is represented by Eq. 1:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R \begin{bmatrix} x \\ y \\ z \end{bmatrix} + T \quad (1)$$

Where, R, T denote the rotation matrix and translation vector respectively.

3D-Grid structure: TLS data consists of millions of points distributed irregularly in 3D space. These points are stored in the order in which they are scanned, forming a unique trajectory according to the specific type of scanner. These points can easily become irregular when the laser beam emitted by the scanner meets edges, corners of objects or scattered back. Because of this, the method for storing the point data is important to ensure low processing costs, fast access times of points and its surrounding information and less memory space.

Much of TLS data processing relies on the operations of querying points at specific locations along with their neighbors. However, such operations cannot be efficiently executed when TLS point data are stored in common data structures such as the stack or queue. A 3D-grid structure can be a solution as the location of neighbors are fast located in this numerous data, but the large computational overhead in forming a high resolution grid with enormous amounts of TLS data is a drawback, $O(n^3)$ storage if you have $N \times M \times L$ cell. However, this weakness can be overcome if enough resources are provided.

The virtual 3D-grid structure main advantage is that searching of points in particular locations can be completed in a constant time thus minimizing the time complexity. Another advantage is that 3D-grid structure

is used to provide an acceptable level of precision or accuracy in the estimation of geometric properties such as surface normal vectors or points simplifications.

The virtual 3D-grid is divided into an $N \times M \times L$ uniform cubes whose cube length is l_{cube} which is a user input, see Eq. 2-4. The lengths of the Bbox are L_X, L_Y, L_Z respectively:

$$N = L_X / l_{\text{cube}}, L_X = x_{\text{max}} - x_{\text{min}} \quad (2)$$

$$M = L_Y / l_{\text{cube}}, L_Y = y_{\text{max}} - y_{\text{min}} \quad (3)$$

$$L = L_Z / l_{\text{cube}}, L_Z = z_{\text{max}} - z_{\text{min}} \quad (4)$$

Finally, insert every point into the linear list corresponding cell. Points are clustered into the cells according to a user-defined interval l_{cube} in directions of x, y and z. Each cell refers to a range of object space and contains points within it. The index of the start of each cell and the numbers of points contained within are then stored to allow for fast retrieval of the points. As a result, the neighbor information between points can be easily established. Depending on the spacing of the grid, each cell should only contain a few features of interest and can be easily separated. Additional information from neighboring grid cells can be compared and incorporated to see if common features or properties exist across the cell. In practice, an appropriate grid size can be determined by checking point density; density is the average number of points within a unit area.

Other researchers like (Han *et al.*, 2009) used the octree based 3D-grid structure for segmentation of 3D point clouds from ALS. Belton and Bae (2010) proposed to store the 3D TLS data into a 2D grid using a modified counting search method.

Point cloud simplification: Simplification or data reduction of point cloud data is one of the key pre-processing technologies in reverse engineering. Where a typical laser scanned data set often contains millions of data points and this leads to significant computational challenges in processing the point cloud data for practical applications such as multiple scene registration, inspection of industrial parts.

Simplification is done to reduce the number of the massive data points to facilitate geometric computation while preserving the geometric feature of point cloud (Peng *et al.*, 2009). By simplifying the point set first, any subsequent surface reconstruction becomes significantly faster. Point cloud simplification tends to be computationally more efficient and less memory demanding than mesh simplification since no mesh data structures need to be maintained.

In this study we use a simple simplification method. First, points in point cloud are distributed into uniform cubes or cells as previously discussed. Next, each cell includes one point which is equivalent to the average value of the clustered points in that cell. The cells that have just one or two points will be discarded and labeled as noisy data. Dealing with the average value of each cube makes the segmentation process very fast rather than segmenting the whole point cloud. So after the segmentation process of the simplified point cloud, the results will be referred back to the original points. This simplification method depends mainly on the l_{cube} of the grid and its resolution. Points in each cell are clustered based on properties of the point's distribution which lead to minimizing the deformation of the resulted simplified point cloud while keeping its geometric features.

Normal vector estimation: The gradient method depends on the normal vector changes as their values are much more significant than those of depth or color changes. In literature, researchers found that by detecting significant normal changes both step edges and roof edges can be easily detected. In order to calculate the normal vector for the point cloud, for every cell in the 3D grid, the normal vector from the enclosed points is computed as an index of local surface formation.

There are two popular approaches to find optimal planes which are LSF and Analysis PCA. LSF is an iterative algorithm for finding the best-fit plane with the least-squares constraint of the distances from the scanned points to the plane.

The PCA method calculates the eigenvector of point cloud as the normal for finding out the parameters of the optimal plane. Huang and Tseng (2008) compare the PCA and the LSF methods in terms of algorithm implementation, computation time and robust estimation. Sharma *et al.* (2006) used PCA to reduce linearly redundant components that may present in higher dimensional space.

This study uses the PCA analysis for calculating the normal vectors. PCA uses Eq. 5 and 6 to calculate the eigenvector as the normal of optimal plane, the solution obtaining three eigenvalues is the calculation of three eigenvectors. The eigenvectors represent the three axes of point clouds and the eigenvalues denote the square sum of point's deviations along the corresponding axis. Hence the eigenvector corresponding to the smallest eigenvalue is exactly the required normal of the best-fitted plane:

$$C = \begin{bmatrix} p_{i_1} & \dots & \bar{p} \\ \dots & \dots & \dots \\ p_{i_k} & \dots & \bar{p} \end{bmatrix}^T \begin{bmatrix} p_{i_1} & \dots & \bar{p} \\ \dots & \dots & \dots \\ p_{i_k} & \dots & \bar{p} \end{bmatrix}, i_j \in N_p \quad (5)$$

$$C \cdot V_1 = \lambda_1 \cdot V_1, i \in \{0,1,2\} \quad (6)$$

Where:

- c = Denotes the 3×3 covariance matrix of local neighbor points
- p_i = The i^{th} point in the point cloud
- \bar{p} = The average of all 3D points
- N_p = The points in the target cell
- λ and V = eigenvalue and eigenvector respectively

Assume that $\lambda_0 > \lambda_1 > \lambda_2$ denotes the eigenvalues of covariance matrix associate with the unit eigenvectors of V_0, V_1 and V_2 , respectively. V_2 is chosen as normal vector which is corresponding to the smallest eigenvalue if $\lambda_0 > \lambda_1 > \lambda_2$. By means of such method, a normal vector on a local surface within a cell could be determined. Using the previous method, for each point p_i a normal vector $n_i = [n_{ix}, n_{iy}, n_{iz}]$ is estimated.

The problem with this method is that the density and points distribution is not always consistent and sometimes the cells have only one or two points (insufficient points). As result of that, normal vector will be calculated using the k-nearest neighbor cells of p_i .

Segmentation using 3D gradient: Edges in 3D range images are defined by the points where changes in the local surface properties exceed a given threshold. The local surface properties mostly used are surface normals, gradients, principal curvatures or higher order derivatives. The extraction of the gradient information from 3D objects plays an important role for many applications including object reconstruction and 3D medical imaging. The gradient information is used to identify strong features in the image such as edges or the local orientation of the curves and the surfaces. In image processing the extraction of the 3D gradient information is generally performed by filtering the input 3D data with filters that are typically implemented using 3×3×3 masks or higher ones.

These 3×3×3 filters usually extract the gradient information using the small surrounding neighborhood. As result, the estimated gradient information will be sensitive to image noise. In order to minimize this image noise we applied the previous pre-processing to the input data before applying the 3D filters. The idea behind the 3D edge detector is to measure the changes of geometric locations across each point of the given 3D grid, i.e., the change being measured along the three directions parallel to the coordinate axes.

Edge detection using 3D sobel edge operator emphasizes edge details using three 3×3×3 masks, one detecting x-gradients, the second one detects the y-gradients and the third one detects the z-gradients. The

magnitude and orientation of an edge is determined using the sum of absolute value of each gradient. 3D sobel masks are convoluted with the corresponding normal vectors of the range image and the cells that have no points take zero value.

A gridded 3D point cloud $f(x, y, z)$ is represented by a matrix of cells, each cell is a small cubic region of $f(x, y, z)$ and has a point $\bar{p}|x$ equal to the average value of all points throughout this region. The gradient at that Cartesian coordinate location $(x, y$ and $z)$ is defined in Eq. 7. And the magnitude of the gradient is given by Eq. 8:

$$\nabla f = [G_x \ G_y \ G_z]^T \tag{7}$$

$$G(n_x, n_y, n_z) = \sqrt{G_x^2 + G_y^2 + G_z^2} \tag{8}$$

We use three 3D gradient operators, or masks, to do the computation, capitalizing the ideas of Zucker and Hummel where all the 26 neighborhoods which are associated with each cell of the grid are involved in computing the gradient vector components. Zucker and Hummel were the first to introduce a 3D gradient operator based on $N \times N \times N$ cell neighborhoods. They provided mathematical foundations to derive three basic functions that define their local operator. Discrete approximations to these basis functions can form three $3 \times 3 \times 3$ masks which are used to compute the components of the gradient vector for the cell located at the center of the mask. One common form of the 3D Sobel operator that approximates the first derivatives G_x , G_y and G_z using the $3 \times 3 \times 3$ filters is shown in Eq. 9-11. Here each filter is expressed as three 2D matrices, which when stacked vertically, form the $3 \times 3 \times 3$ filter. G_y and G_z are basically a rotation in space of G_x :

$$G_x: \begin{bmatrix} -2 & -4 & -2 \\ 0 & 0 & 0 \\ 2 & 4 & 2 \end{bmatrix}, \begin{bmatrix} -4 & -8 & -4 \\ 0 & 0 & 0 \\ 4 & 8 & 4 \end{bmatrix}, \begin{bmatrix} -2 & -4 & -2 \\ 0 & 0 & 0 \\ 2 & 4 & 2 \end{bmatrix} \tag{9}$$

$$G_y: \begin{bmatrix} -2 & 0 & 2 \\ -4 & 0 & 4 \\ -2 & 0 & 2 \end{bmatrix}, \begin{bmatrix} -4 & 0 & 4 \\ -8 & 0 & 8 \\ -4 & 0 & 4 \end{bmatrix}, \begin{bmatrix} -2 & 0 & 2 \\ -4 & 0 & 4 \\ -2 & 0 & 2 \end{bmatrix} \tag{10}$$

$$G_z: \begin{bmatrix} -2 & -4 & -2 \\ -4 & -8 & -4 \\ -2 & -4 & -2 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 2 & 4 & 2 \\ 4 & 8 & 4 \\ 2 & 4 & 2 \end{bmatrix} \tag{11}$$

Selecting threshold: By analyzing of the output gradient magnitude $G(x, z, y)$ we can apply a suitable threshold T in order to detect the edges as shown in Eq. 12:

$$G(x, y, z) = \begin{cases} 1, \nabla f(x, y, z) \leq T \\ 0, \nabla f(x, y, z) > T \end{cases} \tag{12}$$

The point clouds became structured after the pre-processing stage and contain relatively less noise, so as result of doing many experiments the threshold T is set to zero. In other situations, an adaptive threshold could be used. The compromise between having under-segmentation or over-segmentation or well segmented is tackled by setting appropriate thresholds to the three gradients. In order to achieve a satisfactory final segmentation results, one more additional step is done. The point cloud is segmented using different filters with the same threshold. The output is selected based on repetitive edges from these filters; other non repetitive edges are neglected.

Post-processing and edge thinning: There are some problems that face the effectiveness of the segmentation techniques in 2D and in 3D. In the case of 2D images, sobel operator is characterized by its smoothing effect on the output images. As a result of using gradient filters, edges in the output image are several pixels wide and they are a bit broader. The image may be sampled in such a way so that changes in gray level may extend across some number of pixels. Some thinning may be desirable to counter this well known problem.

Edge thinning is a well known technique used to remove the unwanted spurious points on the edge of an image. This technique is employed after the image has been filtered for noise (using median, Gaussian filter). Also it is applied after the edge operator has been applied to detect the edges and after the edges have been smoothed using an appropriate threshold value. This removes all the unwanted pixels and if applied carefully, results in one pixel thick edge elements.

In the case of 3D images, the digitized 3D sobel masks have similar effect on the output 3D edges like in 2D case. But in the case of 3D range images it will differ a little bit, where the points which are detected as edges are doubled instead of being several pixels wide, as the gradient effect of real edges extent to the neighborhoods non-edges where the edges and their neighborhoods could have similar gradient values.

Moreover, 3D image may be sampled in such a way so that the edges is deviated from its real position and could appear in a near position. Furthermore, the influence of the image noise on the output. This noise has unrepresentative value that can significantly affect the gradient value of all the points in its neighborhoods. The Zucker-Hummel used radial functions that smooth the calculated gradient that have a positive effect in noisy data.

As a result of the previous decision, the edges could not be precisely localized and may be corrupted by noise. In general, 3D thinning is a straightforward extension of the 2D image thinning. A number of 3D thinning methods existed for processing the medical images represented by voxels, like histogram equalization and interpolation techniques.

In order to keep the edges extracted after applying the sobel filter precisely localized and closely to that of the actual point clouds, a thinning algorithm is used as a post-processing step after a threshold is applied to the output of the sobel masks.

The thinning algorithm is as follow:

- Search for cells with no points that are marked as edges (this step is applied to eliminate the edges which are deviated from their real position and appear in a near position)
 - Calculate the Euclidean distance between these cells center value and the closest points in space to them with certain distance
 - The matched points are marked as edges instead of the last ones
 - If no points are found within this distance the location of edge is neglected and is labeled as false edge.
- Search for cells that include points marked as edges and their neighborhoods which have similar gradient values that are marked also as edges. (This step is applied to eliminate the double edge effect)
 - Calculate the Euclidean distance between these points' value and the closest points in space to them with certain distance
 - The matched points are marked as edges instead of the last ones

The standard deviation is then calculated to find the deviation of the sobel output and the corresponding corrected edges after applying the thinning algorithm. The rms error ϵ represents the difference between the measurement and its true value, n represents the number of cells. The rms error can be defined as in Eq. 13. The standard deviation of the distribution of these 3D measurements about the average value is defined in Eq. 14:

$$\epsilon_i = \sum_{i=0}^n (x_i - \bar{x})^2 + \sum_{i=0}^n (y_i - \bar{y})^2 + \sum_{i=0}^n (z_i - \bar{z})^2 \quad (13)$$

$$\sigma = \pm \sqrt{\frac{\epsilon_i}{n}} \quad (14)$$

RESULTS

This algorithm is implemented using standard c language and tested on variety of artificial data sets and real world data sets. This is necessary for two reasons.

First, as the 3D gradient operators are tested on 2D data set or 2D slices with certain limitations, we need to know if the proposed algorithm is working properly with different types of 3D artificial and real world datasets.

Second, most real world data are noisy and can often suffer from corruptions when the laser beam emitted by the scanner meets objects of sharply differing heights or may suffer from incomplete values due to partial sensor failure, environmental conditions, data storage corruption or non reflective surfaces such as windows are missing. This noise may impact the decisions made based on the data. Moreover, noise can reduce the system performance in terms of segmentation accuracy.

Thus, empirical experiments on artificial and real world datasets will give us intuitions on the ranges of degree of applicability and how is this algorithm able to detect edges and come over some of these difficulties. The artificial datasets have shapes that are represented by portion of primitive surfaces such as planes, cylinders and so on. The algorithm is used to extract the edges of these shapes. In this study we validate our algorithm in both free noise artificial datasets and noisy real world datasets.

Datasets descriptions: The seven artificial datasets have been generated with different shape complexity. Table 1 shows the CAD models of the artificial datasets with their description. We conduct similar experiments on two real world datasets. These sets are scanned using the terrestrial laser scanner RIEGL LMS-Z620 (<http://www.riegl.com>). These scans are taken for different positions of Mubarak City for Science and Technology (MUCSAT), informatics institute, Alexandria, Egypt (<http://www.mucsat.sci.eg>). The data sets were scanned from three scan positions, then the scan positions were registered and merged as one point cloud. The registered data represents the front and the side facades of the MUCSAT informatics institute which has a pyramid shape. Each data set contains position in x , y , z , color and intensity per point. Here we focus on the geometry information instead of the color information as a clue for our segmentation algorithm.

We have chosen the informatics institute's building because of its geometric shape and its important building features like linear structures such as doors, stairs and windows. As well as, you can find rounded, rectangular, triangular windows, rounded, cylindrical edges and straight Edges. The detection and localization

of these linear structures are very important in variety of applications like robot navigation and inspection of industrial parts.

Table 1: Shows a description for the artificial datasets








Artificial dataset			
Object	CAD model	Number of points	Number of edges
1		1100	12
2		900	24
3		69264	24
4		190910	64
5		16003	156
6		88290	48
7		84920	48 Edges 3 Semi rounded shapes

Table 2: Shows a description for the real world datasets



Real world datasets			
Object	CAD model	Description	number of points
1		Informatics institute Front facade	81053
2		Informatics institute Side facade	74752

Table 3: Shows a description for the real world datasets

Real world datasets		
	Object 1	Object 2
Number of rounded windows	8	14
Number of rectangular windows	61	98
Number of triangular windows	33	33
Number of cylindrical parts	1	3
Existing doors	1	1
Existing stairs	1	1

Segmenting the MUCSAT facades is very challenging for many reasons. First, it is not restricted to simple structures or simple edges. Second, you can find a variety of edge lengths that range from small, medium, large. Third, it has many non reflective surfaces such as windows. The non reflective surfaces are a very important problem that faces most of the segmentation algorithms when dealing with uncompleted point clouds. Fourth, it is composed of three registered scans from different scan positions. If the point cloud contains multiple scans, then the segmentation becomes more complex because it may suffer from corruptions, noise and repetitive points. Also, it has a dense number of points. All the previous conditions and difficulties are used to exam the degree of applicability of the proposed algorithm. Here we will exam our proposed algorithm on two real world datasets described in Table 2 and 3.

Experimental results: the proposed algorithm is evaluated using two real world scans as an outdoor datasets. Also it is validated using six artificial datasets. Experimental results for the artificial datasets are illustrated in Table 4 and the experimental results for the real world datasets are illustrated in Table 5. Tables (4 and 5) show the detected edges for each object after the filtering step, Tables (4 and 5) show each object surrounded by its edges. Table 6 shows the number of edges in the artificial models compared to the number of detected edges after the segmentation step.

As previously discussed, the deviation analysis is very important. Table 7 shows the deviation of the detected edges from its real ones. Considering the real world datasets object 1 and object 2, the proposed algorithm segmented its important features like rounded windows, rectangular windows, triangular windows, boundary, door and stairs. Table 8 shows the segmentation results. Table 9 shows the deviation of the detected edge from its real ones. These low deviation results for the registered real world datasets is not correspondence to the accuracy of the scanner, but the results show the deviation of the sobel output and the corresponding corrected edges after applying the thinning algorithm.

In the introduction we talked about the linear features detection, if we considered the theoretical ideal case in its detection, if the surface is planar every neighboring group of points that are clustered based on properties of the point's distribution must lead to the same normal

Table 4: Shows the results for the artificial datasets


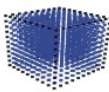

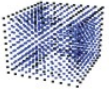














Artificial datasets results		
Object	Detected edges (a)	Object surrounded by edges (b)
1		
2		
3		
4		
5		
6		
7		

Table 5: The segmentation results of the real world datasets

Real world datasets results		
Object	Detected edges (a)	Object surrounded by edges (b)
1		
2		

vector of that original surface. Also the perpendicular distance between that point, which found some neighbors and its plane must be equal to zero. On the other hand, when considering practical cases, the surface of buildings with planer facades is provided as 3D point clouds (X, Y, Z). These points are not immediately compatible with mathematical models, that is to say no planar surfaces and no straight edges are directly provided in the digital model. Moreover the registered planer surfaces appear with a finite thickness. That means that they are in general not flat facades with

Table 6: Shows the results of the artificial datasets

Artificial datasets			
Object	Number of real edges	Number of detected edges	Accuracy %
1	12	12	100%
2	24	24	100%
3	24	24	100%
4	64	64	100%
5	156	156	100%
6	48	48	100%
7	48 and three rounded shapes	48 and three rounded shapes	100%

Table 7: Shows deviation results of the artificial data sets

Artificial datasets							
Deviation in mm	Object						
	1	2	3	4	5	6	7
x direction	0.28	0.30	0.20	0.25	0.220	0.21	0.08
y direction	0.30	0.31	0.34	0.31	0.230	0.32	0.18
z direction	0.27	0.30	0.26	0.25	0.220	0.18	0.15
Average	0.28	0.30	0.27	0.27	0.224	0.23	0.1

Table 8: Shows results of real world datasets

Real world datasets						
	Object					
	Number of real edges		Number of detected edges		Accuracy %	
	1	2	1	2	1	2
Number of rounded windows	8	14	8	14	100.00%	100.000%
Number of rectangular windows	61	98	24	92	39.30%	93.880%
Number of triangular windows	32	32	14	3	43.75%	9.375%
Number of cylindrical parts	1	3	1	3	100.00%	100.000%
Stairs	1	1	1	1	100.00%	100.000%
Doors	1	1	1	1	100.00%	100.000%
Triangular boundary	1	1	1	1	100.00%	100.000%

Table 9: Shows deviation results of the real world data sets

Real world datasets		
Deviation in mm	Object 1	Object 2
x direction	0.03	0.03
y direction	0.07	0.05
z direction	0.04	0.04
Average	0.05	0.04

none zero thickness. Most of the edges in the small triangular windows are missed and the detected are visually squared shaped more than being triangular shaped but these results could be enhanced using smaller L_{cube} or grid with smaller resolution in this particular space.

DISCUSSION

Here we discuss the main differences between the first and second approaches (see the literature review)

and our approach. In the two approaches, the target samples are different where the first and second approaches are using a sobel filter that works on consecutive 2D images or 2D slices stacked together with constant depth (usually medical images), while our approach is using real 3D data which have information about geometric coordinates and color.

In the first approach, the contribution of the 26 points or 26 voxels do not exist as the depth is constant. In the second approach, the depth is taken into account but it is still equally spaced in z direction. In our approach the depth info is taken into account as the point's depth may vary from cell to another or even no points exist in certain depth.

The first approach uses two filters 3×3, one for x direction the second one for y direction, while the second and our approach use 3 filters 3×3×3 one for x direction, the second for y direction and the third for z direction

In the two approaches, the convolution works usually on the intensity data in each pixel or voxel (for medical images) but in our approach it works on the normal vectors or the geometric locations of each cell. Normal vectors and the geometric locations are more accurate as the color or gray level is generally meaningless in calculating of the edges in 3D data. Also the normal vector is calculated using the PCA of the included points within the each cell and their neighbors.

The two approaches work well on images with good contrast between different regions; the edges are detected and can be verified visually. On the down side, as these approaches detect most of the edges. It is very difficult to find a correlation between edges in successive image slices. The two approaches work well on images with good contrast between different regions; the edges are detected and can be verified visually. These two approaches are also susceptible to noise. In most of the cases, they are followed by other segmentation algorithms to find an accurate segmented volume. Additionally, unlike 2D images, we cannot use the color information or the texture to detect surfaces or edges, as they lose the depth information in the projection process from 3D-2D.

Most of 3D segmentation algorithms use the geometric properties as a clue. Also, unlike most of computer graphics models, the input is a noisy point cloud representing a registered scene or real world objects rather than a clean surface model of individual object.

We summarize below the steps used in the computation of the proposed algorithm:

Input: 3D-grid contains n points, represented by $p(p_1, p_2, p_3, p_n)$, Where each cell has $p_i(\overline{p_{ix}}, \overline{p_{iy}}, \overline{p_{iz}})^t$,

Where $\overline{p_{ix}}$ denotes the average point in that cell in x direction.

Output: geometric edge detector $G_f(x,y,z)$

Steps: For each cell:

- Calculate the G_x, G_y, G_z and Compute $G(x,y,z)$
- Threshold $G(x,y,z)$
- Filtering phase

Computational complexity: The sobel algorithm runs in linear time, which is if N is the number of cells then the estimated run time for the proposed edge detector would be $O(N)$. Its performance is only weakly affected by the huge increases in the input model size that can be easily solved by using the appropriate hardware.

The calculation of the G_x, G_y and G_z for each cell is independent from the calculation of the rest of the cells. Besides, the grid size may be huge due to small step size and high resolution grid. As a result of the previous, it is more preferable that the 3D range image is divided into small windows (3×3×3) that could be processed in parallel with appropriate hardware or using a parallel 3D sobel algorithm rather than using the sequential one. Han *et al.* (2009) applied a parallel processing method for airborne laser scanning data using a pc cluster and a virtual grid in order to control the huge amount of point cloud data.

CONCLUSION

Segmentation is an essential pre-processing step in image understanding. We have proposed a straightforward 3D edge detector for artificial datasets and real world datasets, generalizing the classical sobel operator of 2D images. The presented method is used to extract edges from 3D range images. We have validated our algorithm on seven artificially generated datasets and two real world datasets captured by the terrestrial laser scanner system RIEGL LMS-Z620.

The proposed algorithm is applied in three stages the pre-processing stage is first applied followed by 3D gradient edge based segmentation and finally the post-processing stage which is used for thinning the output edges. The pre-processing stage includes virtual 3D-grid generation, normal vector calculations, point cloud data reduction. The point cloud is fitted in a virtual 3D-grid for the efficient search of the neighbors thus minimizing the time taken for segmentation. This tight 3D-grid structure shrunk the area to be searched and handles huge number of points.

The proposed algorithm depends on the normal vector changes and the geometric locations; x , y and z ; as a clue for segmentation of 3D range images. As normal vector values are much more significant than those of depth or color changes. As the color or gray level information are generally meaningless in calculating of edges in 3D data. The Principal Component Analysis (PCA) is applied here to derive the local normal vectors. Besides that, the PCA is applied for the analysis of rotation and translation of the point clouds. The factors which control the output of being well segmented are the step size used to define the virtual 3D-grid and the threshold that is applied after the 3D sobel filter.

Experiments have been conducted for testing the performance of the proposed algorithm using seven artificial datasets and two real world datasets. Linear structures, such as doors, stairs and windows are detected using the proposed algorithm. These linear structures are common in indoors and outdoors in many environments. Segmentation of these linear features is very essential in many applications like mobile robot navigation and is useful for building modeling and surface reconstruction. The proposed algorithm is efficient in the detection of all the edges of different shapes in both the artificial datasets and the real world datasets. This means that our segmentation algorithm is not restricted to simple structured objects only.

For future work: The 3D range image can be divided into small windows ($3 \times 3 \times 3$) or ($5 \times 5 \times 5$) that could be processed in parallel with appropriate hardware and parallel 3D sobel technique instead of the sequential one.

REFERENCES

- Awwad, T.M., Q. Zhu, Z. Du and Y. Zhang, 2010. An improved segmentation approach for planar surfaces from unstructured 3d point clouds. *Photogrammetric Record*, 25: 5-23. DOI: 10.1111/j.1477-9730.2009.00564.x
- Belton, D. and K.H. Bae, 2010. Automating post-processing of terrestrial laser scanning point clouds for road feature surveys. *Proceedings of ISPRS Commission V Mid-Term Symposium, (ISPRS'10)*, Newcastle upon Tyne, UK., pp: 74-79.
- Boulaassal, H., C. Chevrier and T. Landes, 2010. From laser data to parametric models: Towards an automatic method for building façade modelling. *Proceedings of the 3rd International Conference on Digital Heritage, (ICDH'10)*, Springer-Verlag Berlin, Heidelberg, pp: 42-55.
- Boulaassal, H., T. Landes and P. Grussenmeyer, 2009. Automatic extraction of planar clusters and their contours on building facades recorded by terrestrial laser scanner. *Int. J. Arch. Comput.*, 7: 1-20. DOI: ijac20097101
- Cang, Y. and P.M.G. Hegde, 2009. Robust edge extraction for Swissranger SR-3000 range images. *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, May 12-17, IEEE Xplore, Piscataway, NJ, USA., pp: 2437-2442. DOI: 10.1109/ROBOT.2009.5152559
- Chi, J., X. Wu and C. Zhang, 2008. Medical CT image preprocessing based on edge detection and spline fitting. *Proceedings of the IEEE International Symposium on IT in Medicine and Education*, Dec. 12-14, IEEE Xplore, Xiamen, pp: 709-714. DOI: 10.1109/ITME.2008.4743958
- Chowdhury, A. Tarik and O. Ghita and P.F. Whelan, 2006. Evaluation of 3d gradient filters for estimation of the surface orientation in CTC. *Proceedings of the IMVIP 2006-10th International Machine Vision and Image Processing Conference*, Aug. 30-1 Sep., Dublin, Ireland, pp: 1-8.
- Coleman, S.A., B.W. Scotney and S. Suganthan, 2010b. Edge detecting for range data using laplacian operators. *Trans. Img. Proc.*, 19: 2814-2824. DOI: 10.1109/TIP.2010.2050733
- Coleman, S.A., S. Suganthan and B.W. Scotney, 2010a. Gradient operators for feature extraction and characterisation in range images. *Pattern Recogn. Lett.*, 31: 1028-1040. DOI: 10.1016/j.patrec.2009.12.022
- Han, S.H., J. Heo, H.G. Sohn and K. Yu, 2009. Parallel processing method for airborne laser scanning data using a pc cluster and a virtual grid. *Sensors*, 9: 2555-2573. DOI: 10.3390/s90402555
- Huang, C.M. and Y.H. Tseng, 2008. Plane fitting methods of lidar point cloud. *National Cheng Kung University*.
http://www.topografia.upm.es/pdi/an_dom/Doctorado/lidaryestrobusta.pdf
- Mazouzi, S. and M. Batouche, 2007. Range image segmentation by randomized region growing and bayesian edge regularization. *J. Comput. Sci.*, 3: 310-317. DOI: 10.3844/jcssp.2007.310.317
- Monga, O., R. Deriche and J.M. Rocchisani, 1991. 3D edge detection using recursive filtering: Application to scanner images. *CVGIP: Image Understand.*, 53: 76-87. DOI: 10.1016/1049-9660(91)90006-B

- Peng, X., W. Huang, P. Wen and X. Wu, 2009. Simplification of scattered point cloud based on feature extraction. Proceedings of the 3rd International Conference on Genetic and Evolutionary Computing, Oct. 14-17, Guilin, China pp: 335-338.
- Sharma, A., K.K. Paliwal and G.C. Onwubolu, 2006. Splitting technique initialization in local PCA. *J. Comput. Sci.*, 2: 53-58. DOI: 10.3844/jcssp.2006.53.58