Volume 3, Issue 11, November 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Reducing Random and Uniform 3D Points for Acquisition System

E. A .Zanaty, Ashraf Afifi

Information Technology Département,

College of Computers and Information Technology,

Taif University, Taif, Saudi Arabia.

Abstract— This Acquisition system always is used for extracting 3D point clouds of a given object. This system is capable to obtain accurate 3D points that can represent the geometrical structure of the object. A major problem of this system is that certain types of camera or scanner produce vast amounts of data, the processing of which presents serious problems. Rather than process all of this data at every stage of the representation process, an alternative is to use a strategy in which the data is initially reduced, then a pre-processing can be completed without consuming a lot of time. This paper presents an algorithm for managing the amount of point data acquired by laser scanner. The proposed algorithm includes a method based on the surface fitting which is fundamental in the most of reverse engineering algorithms. The surface fitting is calculated by fitting the best fit surface to the neighbourhood. The neighbourhood is obtained by subdividing the point data into cells based on scanned surface structures. Thus, the amount of points can be reduced by sampling the representative points for each cell. Experimental results show that the proposed method has good results and appears to be quite stable even for large scale data reduction. The proposed algorithm is proved by fitting known shapes such as plane using the well-known least squares method. For better comparison, we apply two different projection patterns, we used random squares pattern, and uniform squares pattern. By computing the distance from each 3D point to the plane, the RMS error between the 3D points and the plane fitting is calculated. The experimental results show that the random squares pattern is more accurate and faster than the uniform squares pattern for getting 3D points cloud for scanning objects.

Keywords—Laser scanner, 3D points reduction, surface fitting, RMS error.

I. INTRODUCTION

During recent years, some researchers have shown interest in managing the amount of points acquired by laser scanners [1-3, 5-7, 8, 11, 19]. They are grouped into two categories, the discrete space such as Fourier or Wavelets Transform, and real space image. The first category techniques include the Discrete Fourier Transform (DFT) introduced in [1] and extended in [3], and the Discrete Wavelet Transform (DWT) introduced in [8]. Those techniques interest to reduce the redundant points in one dimension and two dimensions. The idea of Fourier transform technique is to reduce the massive amount of data provided by the input device, or arising as a result of transmission of the image. Such that if we reduce the high frequency component, we should reduce the amount of points in the image more details of that technique was described in [10]. However the discrete Wavelet transform of a signal is calculated and the resultant Wavelet coefficients are passed through a threshold testing. In this case, the coefficients that are smaller than a certain value are removed. Then the resultant coefficients are used to reconstruct the signal [8]. It is not the objective of this chapter to work in this direction, where the techniques that presented in this category do not exist in three dimensions and are practically difficult to be manipulated with very dense 3D data. The simplest approach of the second category is neighborhood averaging, which a point is replaced by the average value of the points contained in some neighborhood when a point is not too far from the neighborhood, otherwise we keep it unchanged. Thus, if we are near to some sort of edge, there will be a large change in points, and points on both sides of it will not be close to the average value [8]. Even better is to use a median filter rather than averaging. A neighborhood around the point under consideration is used, but this time the point value is replaced by the median point in the neighborhood, more discussion can be seen in Martin et. al [11]. Their method used a uniform grid in their EU Copernicus project and used a median filtering approach, which has been widely used in image processing. The procedure starts from building a grid structure, and the input points are assigned to the corresponding grid. From all of the points assigned to a given grid, a median point is selected to represent points belonging to that cell. Their method, however, has drawbacks due to the use of the uniform size grids that can be insensitive in capturing a part shape. However, the weakness of this method is the deleting of the points from every cell without constraint can distort the surface, especially when the cell belongs to the edge. Also, replacing one cell by one point may distort the surface or their boundary. Fujimoto and Kariya [5] also indicated that efforts are needed to reduce the amount of data since a large amount of points causes problem in using them in downstream manufacturing operations. They suggested an improved sequential data reduction method only for 2D digitized points. Their method guarantees an error range of the reduced data to remain within the given angle and distance tolerances. There are data reduction methods for reducing the number of polygons in a polygon model, especially for models with triangular patches. The point data for these models were not

generated by laser scanners but they were created for other purposes such as rendering and analysis [10, 13]. Hamann [7] has also presented a method of data reduction for triangulation files based on an iterative triangle removal principle. As a measure of the reduction of file size, each triangulation is weighted according to the principal curvature estimates at its vertices and interior angles. The weakness of this method is that takes a long time to generate triangulation and estimate its curvature. Zanaty et al.[20] improved a data reduction method for reducing 3D points for reverse engineering. Their method is based on Gaussian image computation [12]. The Gaussian image is partitioned into some parts. In each part, the corresponding points are selected and referred to them by a cell, where a center mass point of the part is chosen as the representative of the cell. This method is very accuracy with ruled and freeform objects, but when the object is planar, this method always gives irregular distribution of points, where in some cases the two adjacent points in an Gaussian image part are not adjacency in a corresponding cell. Surprisingly, little work has been done to combine real data reduction and hold only the pre-processed data for further analysis. Such work is crucial importance since it is extremely difficult to work with dense data in reverse engineering processes. So the present paper introduces new idea to reduce dense data and applicable for all reverse engineering processes. It has advantages of accuracy and computation, where the most computation is the surface normal, which is necessary for all most reverse engineering applications.

In this paper, a fast algorithm for data reduction of the given 3D points is presented. The proposed algorithm consists of three steps. In the first step, the neighborhood of the points is estimated, whereby the surface normal is obtained. Point data is subdivided into cells based on surface normal the in second step. This gives the proposed method more advantage, where the size of the girds can be varied based on the part of the shape. The procedure in third step is used to reduce the data in each cell.

II. THE ALGORITHM

Assume a point data in which we want to manage it consists of n -points. We fed the point data to our algorithm. The proposed algorithm is consists of three steps. The first step is pre-processing where the data is input, organized, and partially analyzed to prepare the remaining operations. Pre-processing includes data points, sorting, determination of a neighborhood for each point, and computation of an approximate normal vector to the surface at each point. In the second step, an initial partition of point data into cells is obtained. In each cell, we use the normal vectors assigned to the points. An arbitrary direction is selected and referred as a reference direction. The angles between the reference direction and the normal vectors are estimated, where the angles with smallest value and the biggest value are selected. If the value is greater than a prescribe value the cell is partitioned again. The procedure in third step is used to reduce the redundant data in each cell.

The method can be divided into the following steps:

- Computing the surface normal.
- Subdividing 3D points.
- 3D points reduction.

A. Surface normal computation

Nowadays the computing of surface normal is fundamental in the most of reverse engineering algorithms. In Vanco [15-16] three different methods are reviewed for normal vector estimation. The main idea of the first one is based on computing the center mass point of the local neighbourhood of a point p . If the normal vector in centre mass point is estimated, it will be close to the normal vector of p . In the second method, the neighbourhood of p using Delaunay triangulation is triangulated. The estimated normal vector is the average of the normal vectors of the triangles incident with p . In the third method, the set of k -nearest neighbours of a point p computed with respect to the Euclidean distance is the same as the set of the k -nearest neighbors of p with respect to geodesic distance, i.e. on the surface of the initial object. The task of this method is to find the best quadratic or cubic function that fits the neighbouring points of p . Quadratic and cubic surfaces can well approximate the common types of a small point set. Surface of higher degree would approximate a noisy point set "too well", i.e there could occur unwanted oscillations, more discussion can be seen in [15-16]. A very common approach of Faugeras and Herbert [4] to approximate the normal in p is to compute the plane of regression of the data set $n(p)$ and to use the normal of the plane as the approximation. The task of Varady techniques [18] is to find the best fitting plane to the neighbourhood $n(p)$ of a point p in the least squares (LS) sense. After that, the unit normals are assigned to the points, more details of those techniques were described in [18]. It is fundamental in our algorithm that for each data point P_i we estimate a unit surface normal. To estimate normal vector, it is necessary to find the best neighbourhood of point. Then, a method is used to compute the normal vectors. This method that used to find the normal makes use of the tangent plane as an approximation to the surface in the neighborhood of a point. However, the normal vectors are calculated by fitting the best fit plane to the neighbourhood. The unit vector of the best fit plane is assigned to that point. After that, we use the normals assigned to the points.

B. NEIGHBOURHOOD ESTIMATION

neighborhood of a point consists of data point is from the original data set, which are “near” the given point. An ideal definition of neighborhood for the purpose of points reduction would be that a neighborhood includes only those data points that describe the surface at the given point but includes all points necessary for measurement of the variation of the surface around the given point. In Vanco et al.[16] introduced a method capable for extracting k-nearest neighbors. Their algorithm gives for small k which is enough for neighborhood information and is mostly used in the reverse engineering

rather better results. The organizing of the data structures of this method, the search algorithm and an analysis of the complexity of this method has been given on detail in [16].An ideal definition of neighborhood for the purpose of feature point extraction was presented in [14] that a neighborhood includes only those data points that describe the surface at the given point but includes all points necessary for measurement of the variation of the surface around the given point. A neighborhood is chosen for each point that includes all point within a given distance of the point in combination with a limit on the maximum number of points in a neighborhood.

Here, we use the computation of the neighborhood that presented in Vanco's paper, where this method obtains almost good results and works very fast. We select k to be 10 which sufficient to obtain faithful results. So, for every point its k-nearest neighbors are computed, which describe this small portion of the surface-in ideal case the neighbors are distributed regularly round the point.

C. ESTIMATING NORMAL VECTORS

Here, we develop the computation of the normal vectors of the Varady method [18]. To determine an appropriate coordinate system we start Delaunay Triangulation and compute a first estimation of the normal $\vec{n}_f$ in p. Then $\vec{n}_f$ is used as *z*-axis of our new coordinates system, which is used for the computation of the best fitting plane.

Algorithm 1:

- Estimate the normal $\vec{n}_f$ in p using Delaunary triangulation
- Transform the coordinates of p and all its neighbours $p_i, i=1,...,k$ to the new coordinates system (become z-axis). We denote these points with $p_i^t, i=0,...,k, p_0^t = p^t$
- Approximate p_i^t with a planar function using the least square method. e.g. planar surface: $\sum_{i=0}^{k}(Ax_i + By_i + Cz_i + D) \rightarrow \min; A^2 + B^2 + C^2 + 1$ where $[x_i,y_i,z_i]$ are the coordinates of p_i and *A,B,C,D* are the coefficients to be found
- Compute normal vector $\vec{n}$ as $(\frac{A}{\sqrt{|A|}}, \frac{B}{\sqrt{|B|}}, \frac{C}{\sqrt{|C|}})$
- Transform (rotate) normal $\vec{n}$ to original coordinate system.

D. 3D POINTS PARTITIONING

In this section, we partition the 3D points based on Voronoi region which is fundamental of surface reconstruction and Delaunay triangulation processes. Given two points P_i and P_j in the plane T, the perpendicular to the segment P_iP_j in the middle point divides the plane T into two regions, V_i and V_j. Region V_i contains all and only the points closest to P_i than to P_j ; if we have more points we can easily extend this concept saying that V_i is the region assigned to Pi so that each point belonging to V_i is closest to Pi than to any other point. The subdivision of the space determined by a set of distinct points so that each point has associated with it the region of the space nearer to that point than to any other is called Dirichlet tessellation. This process applied to a closed domain generates a set of convex distinct polygons called Voronoi regions which cover the entire domain. This definition can be extended to higher dimension where, for example in three dimensions, the Voronoi regions are convex polyhedrons. If we connect all the pairs of points sharing a border of a Voronoi region we obtain a triangulation of the convex space containing those points. This triangulation is known as Delaunay triangulation. An example of the relationship between Voronoi regions and Delaunay triangulation in two dimensions is given in Fig. 1. Similarly we can obtain a triangulation for higher dimensions, for example in three dimensions if we connect all pairs of points sharing a common facet in the Voronoi diagram, the result is a set of tetrahedra filling the entire domain.

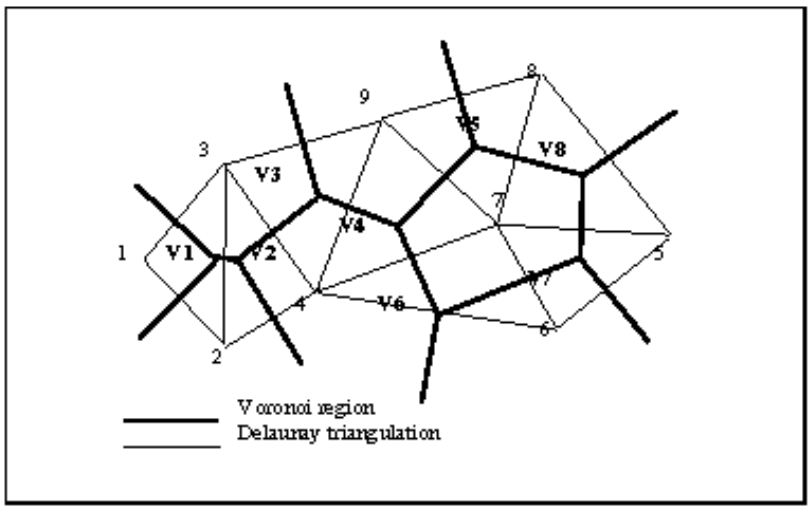

Fig.1: Voronoi regions and associated Delaunay triangulation

E. 3D POINTS REDUCTION

In the following computations, we will always use the normal vectors n_i assigned to the data points. The 3D points are subdivided into cells, where each cell contains some points. In each cell, we use the normal vectors $\vec{n}_i$ assigned to the points $p_i, i = 1,2,...,\beta$, where β is the number of points in cell and has been obtained automatically based on the part of shape. After that an arbitrary direction is selected as a reference direction. Here, we use the unit vector of the x-axis (one can use y-axis or z-axis) of the coordinate system as the reference direction, i.e. we used $\vec{N} = (1,0,0)$. After that, we compute the angles $\theta_i, i = 1,2,...,\beta$ between N and $\vec{n}_i, i = 1,2,...,\beta$. The angles between the reference direction and the normal vectors are estimated, where the angles with smallest value θ_{min} and the biggest value θ_{max} are selected.

If the value $|\theta_{\max} - \theta_{\min}|$ is greater than a prescribe value η, the cell is partitioned again. Otherwise the median angle is selected, where it is easy to find the corresponding normal, whereby the corresponding 3D points are chosen as the representative points of the cell.

Algorithm 3:

- j = 1
- Select the arbitrary reference direction
- Select the normal vectors $\vec{n}_i, i = 1,2,.., \beta$, assigned to the points of the cell C_j
- Estimate the angles between a reference direction and the normals.
- If $|\theta_{\max} - \theta_{\min}| \geq \eta$, apply algorithm 2 to subdivide the cell, where η is a prescribed value
- Select the median angle.
- Find the corresponding unit normals of the angle
- Select the corresponding point
- Remove the residual points in C_j
- j = j+1
- Repeat step 3 through 6 until all the cells in the array are processed
- Stop

III Experimental results

Our proposed technique for reduction experimented using datasets of part surfaces for the test scan object and the results are discussed. Before applying the reduction algorithms, the farther noise point from the initial point cloud is removed. These tests have shown, for surface normal computation, which the best neighborhood size for point set with noise sampling is in the range of 10-20. If we used a bigger neighborhood size 20, the normal vectors on smooth surfaces was not estimated better than with k = 10. If the data contains noise or they consists of scan lines with big distances between scan lines and small distance within one scan line, the neighborhood size have to be enlarged to about k = 6.

Our proposed technique for reduction experimented using datasets of part surfaces for the test scan object and the results are discussed. Before applying the reduction algorithms, the farther noise point from the initial point cloud is removed. These tests have shown, for surface normal computation, which the best neighborhood size for point set with noise sampling is in the range of 10-20. If we used a bigger neighborhood size 20, the normal vectors on smooth surfaces was not estimated better than with k = 10. If the data contains noise or they consists of scan lines with big distances between scan lines and small distance within one scan line, the neighborhood size have to be enlarged to about k = 6.

A. RMS errors.

The Root Mean Squared (RMS) difference are used to evaluate similarity between patches. RMS is a measure of dissimilarity i.e. the higher the distance between corresponding pixels in the patches, the higher the output value. RMS is a variant on the Sum of Squares Difference (SSD). For two image patches of similar size, f_1 and f_2, the SSD is given as:

$$SSD = \sum_{x,y \in f} (f_{x,y} - g_{x,y})^2 \quad (4)$$

The values returned by equation (4) tend to be very high, due to being raised to the power of two. To make the values fall closer into line with the cross-correlation values, the RMS was used: :

$$RMS = \left\{ \frac{1}{XY} \sum_{x=1}^{X} \sum_{y=1}^{Y} (f_1 - f_2)^2 \right\}^{\frac{1}{2}} \quad (5)$$

where X and Y are the width and height of the patches [21]. In order to evaluate the accuracy of 3D points, a plane was fitted to the 3D points. By compute the distance from each 3D point to the plane, the RMS error between the 3D points and the plane fitting is calculated.

B. Random squares pattern projection

The random squares pattern is generated to scan the front surface of test object, the size of each square is (60 x 60) pixel, and the size of projection window is (1200 x 1600) pixel. The front surface of test object has been scanned with different

resolution (i.e. 500-600, 2000-3000, 5000-7000 points), after we got the points cloud, by implementing the Delaunay Triangulation to generate polygons out of this points. Furthermore, the RMS error for these points has been calculated. Then we apply our method for reduction for getting points and calculate the RMS error for points after reduction**.** To experiment our proposed system, we generated random squires and project them to the front surface of the test scan object , in the end of scanning, we got 500, 2100 and 5000 points.

C. REDUCTION OF RANDOM POINTS

Table I shows some results that were experimented on the proposed data reduction technique. We have fitted the planes, to the data using the least squares method that is presented at [9] pre and post applying the proposed method. The average of the distances (the error) between the surface and the points are computed after and before fitting, where the reduction degree is the number of passing the data to the proposed algorithm without modifying the thresholds. We have selected $k = 10$ as the best neighborhood size for a point set. The parameter β has been obtained automatically based on the part of shape. It varies from cell to another depending on the size of a cell. For that non-uniform cells in which the size of cells can be varied based on the part shape are created. After testing of many examples, we noted that the selection of the parameter to be $\eta = 0.1$. We note that when this parameter is decreased, the number of output data points is extremely reduced. The 500, 2100 and 5500 points are fed to the proposed algorithm and the result is presented in Fig. 2,3, and 4 respectively. The results show in Table I that the fitted surface is better than that one before. For example, the 500 points is fed to our algorithm. We note that it has 430 points if the reduction degree is 1. However, in every reduction step a surface is fitted to the data point and the RMS error is estimated. We noted that before reduction the RMS error is 0.761, after applying the algorithm to the data, the RMS error became 0.521. The 2100 points has the RMS error 0.896, after reduction became 1820 and its RMS error was 0.630. The 5500 points has RMS error 1.89 before reduction and RMS error 0.932 after reduction and became 5012 points. This shows that the propped method is very faithful to obtain accurate results while keeping quality point data.

TABLE I
The RMS for random 3D points cloud.

Number of Points	Number of Points after Reduction	Reduction degree	RMS	RMS After Reduction
500	430	1	0.761	0.521
2100	1820	1	0.896	0.630
5500	5012	1	1.89	0.932

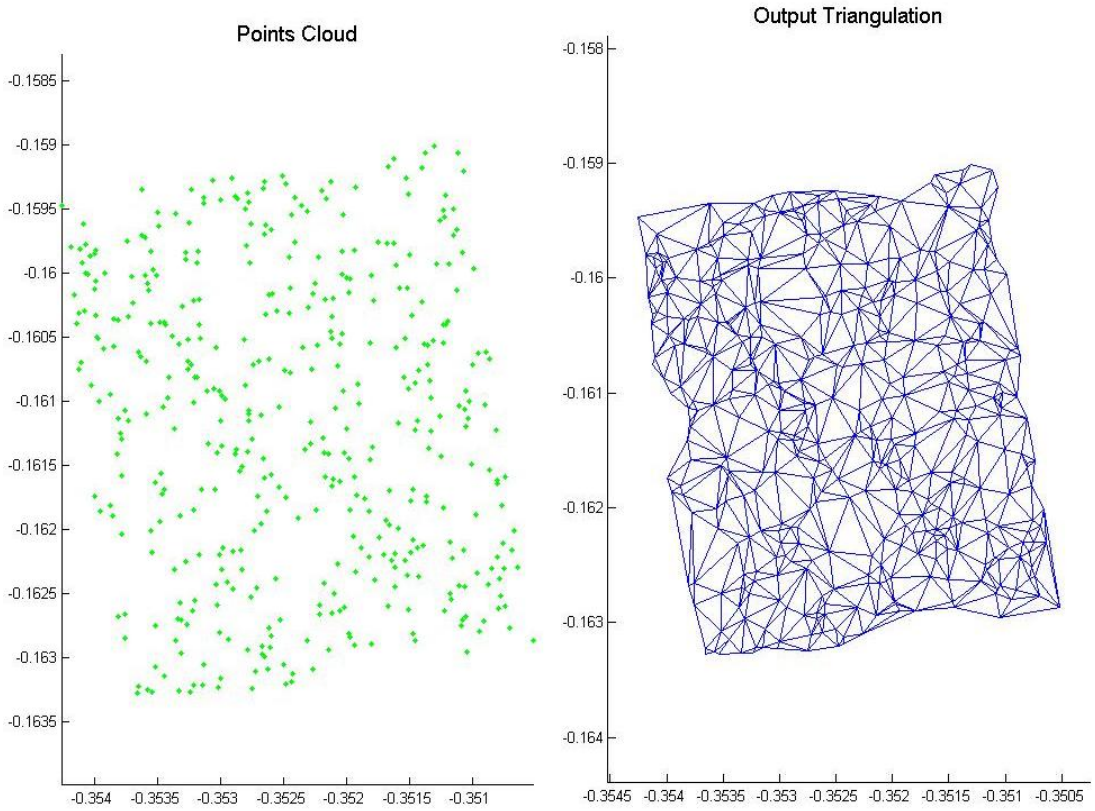

Fig. 2: 500 3D points cloud after reduction, it has 430 points

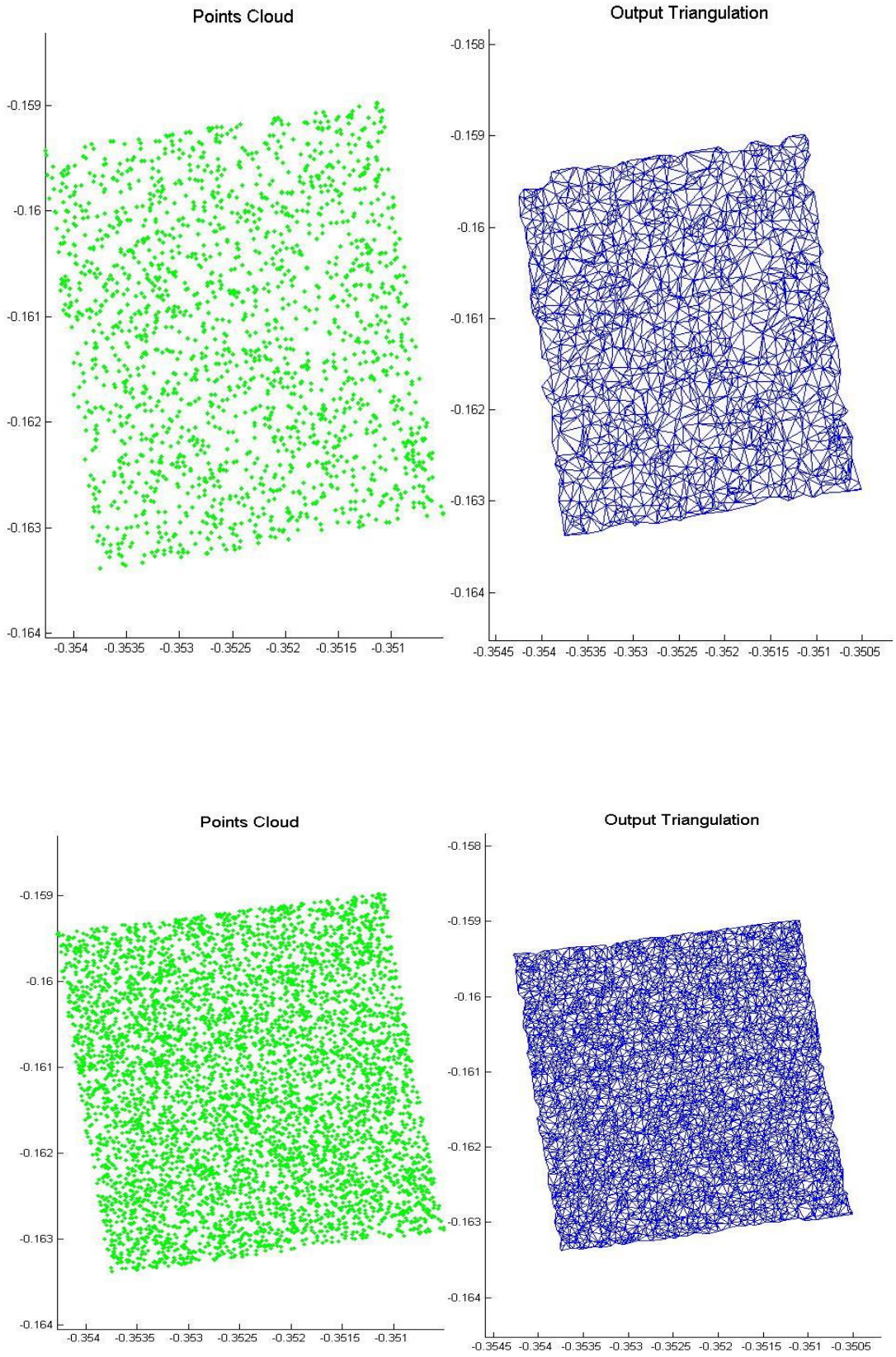

Fig. 4: 5500 3D points cloud after reduction, it has 5012 points.

D. Uniform squares pattern projection The uniform squares pattern is generate to scan the front surface of test object, the size of each square is (60 x 60) pixel, and the size of projection window is (1200 x 1600) pixel. The front surface of test object has been scanned with different resolution (i.e. 500-600, 2000-3000, 5000-7000 points).

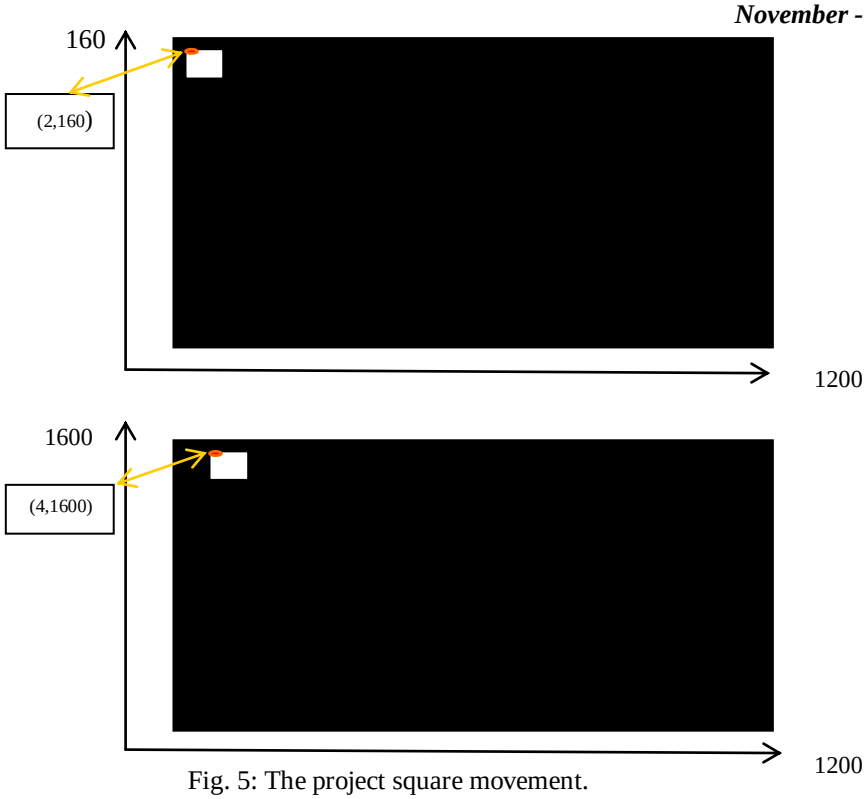

Fig. 5: The project square movement.

Because of the size projection window is (1200 x 1600) pixel, we can only project 533 square with this size, and for scanning high resolution (i.e. 2000-3000, 5000-7000 points) we did the following:

- In the first scanning run, we start project first square from the position (2, 1600) of projection window
- In the Scand scanning run, we move the first square 2 pixel in direction x to starts from (4, 1600) and so on (see Fig. 5)

After we got the points cloud, the Delaunay Triangulation is implemented to generate polygons out of this points and the RMS error for these points has been calculated. Then we apply our method for reduction for getting points and calculate the RMS error for points after reduction, Fig. 2, 3 and 4 show the results for 500 , 2100 and 5500 points respectively .

E. Reduction of uniform points

The proposed technique for reduction experimented using 3D points getting from uniform projection of squares of part surfaces for the test scan object. The TABLE III shows the rustles, the 500 points is reduction to 429 if the reduction degree is 1, with 0.855 RMS error before reduced and 0.567 RMS error after reduced. The 2100 points is reduced to 2062 points and the RMS errors are 0.923 and 0698, before and after reduction respectively. The 5500 points has 1.982 RMS error and after reduction became 5446 points and has 0.997 RMS error. Fig. 4 Plot this result, the yellow line draws the values of RMS errors for uniform 3D points 500, 2100, and 5500 before reduction and in Fig. 5 yellow line draws the values of RMS errors for uniform 3D points 500, 2100, and 5500 after reduction.

TABLE IIII
The RMS for uniform 3D points.

Number of Points	Number of Points after Reduction	Reduction degree	RMS	RMS After Reduction
500	492	1	0.855	0.567
2100	2062	1	0.923	0.698
5500	5446	1	1.982	0.997

F. Comparison between uniform and random projection

The results in Tables I and II show that the RMS errors for 3D point getting from random projection method is less than the RMS errors for 3D points getting from uniform projection, which means, that the destines between the fitting plane and 3D random point is smaller than the destines between fitting plane and 3D uniform point. After applying our reduction algorithm for 3D points, The results in Tables I and II show that the RMS errors for random projection method is less than the RMS errors for the uniform projection one.

IV. Conclusion

This paper has been introduced a fast algorithm for data reduction of the given 3D points cloud. The proposed algorithm consists of three steps. In the first step, the neighborhood of the points is estimated, whereby the surface normal is obtained. 3D Points cloud is subdivided into cells based on surface normal in the second step. This gives the proposed method more advantage, where the size of the girds can be varied based on the part of the shape. The procedure in the third

step is used to reduce the data in each cell. The Delaunay Triangulation technique is then employed to establish connectivity between the unstructured 3D points cloud representing the object surface.

The proposed acquisition algorithm was experimented by scanning different objects with different shapes and for different numbers of scanning points. For better comparison, we apply two different projection patterns, we used random squares pattern, and uniform squares pattern. For evaluating the accuracy of 3D points cloud, a plane was fitted to the 3D points. By computing the distance from each 3D point to the plane, the RMS error between the 3D points and the plane fitting is calculated. Finally the reduction process is governed by three threshold values β and η. The parameter β controls the amount of data partitioning and η controls the density of points within each cell. Thus the user is able to specify the amount of data to be reduced and control the algorithm to suit specific data sets.

REFERENCES

[1] S. M. Metev and V. P. Veiko, *Laser Assisted Microtechnology*, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.

[2] Agrawal, R., Faloutsos, C., and Swami, A. "Efficient Similarity Search in Sequence Databases". In Proc. Of the Fourth International Conference of Foundations of data Organization and Algorithms, 1993.

[3] Daniel, C., David, L., and Offir, R., "Progressive compression of arbitrary triangular meshes ", Proceedings, Korea Israel Bi-National conference on Geometrical Modeling and Computer Graphics in the world wide Web Era, Seoul, Korea, pp. 83-96, Sptember 1999.

[4] Faloutsos, C., Ranganathan, M., Manolopoulos, "Fast subsequence Matching in Time-Series Databases". In Proc. International Conference on Management of Data (ACM SIGMOD), Minneapolis, 1994.

[5] Faugeras, O.D. and Herbert, M., "The representation recognition and locating of 3D object", International Journal of Robotics Research 5, pp.27-52, 1986.

[6] Fujimoto, M., and Kariya, K., "An improved method for digitized data reduction for computer integrated manufacturing", Proc. International Conference on Industrial Electronics, Control Instrumentation, and Automatic, pp. 896-901, 1992.

[7] Gongde Guo, Hui Wang, David A. Bell: Data Reduction and Noise Filtering for Predicting Times Series". International Conference of Web-Age Information Management (WAIM 2002), pp.421-429, 2002.

[8] Hamann ,B., "A data reduction scheme for triangulation surface", Comput.-Aided Design (CAD), 11, pp.197-214, 1994.

[9] Kin-pong, C., and Ada, W. F., "Efficient Time Series Matching by Wavelets", Proc. of the 15th International Conference on Data Engineering, 1999.

[10] Lukacs G., Marshall A. D. and Martin, R. R. ,"Faithful least- squares fitting of spheres, cylinders, cones and tori for reliable segmentation", Proc. of the Fifth European Conference on Computer Vision Freiburg, 1998.

[11] Marshall, A.D. and Martin, R.R., "Computer vision, models and inspection," World Scientific Series in Robotics and Automated Systems, 4, 1992.

[12] Martin, R.R., Stround, I.A., and Marshall, A.D., "Data reduction for reverse engineering", RECCAD, Deliverable Document 1 COPERUNICUS project, No 10 1068, Computer and Automation Institute of Hungarian Academy of Science, January 1996.

[13] Randrup T., "Approximation of surfaces by cylinders", Comput.-Aided Design (CAD), 30(10), pp.807-812, 1998.

[14] Shroeder, W.J., Zarge, J.A. and. Lorensen, W. E., "Decimation of Triangle Meshes", Computer Graphics, 26(2), pp. 65-69, July 1992.

[15] Toll, Bill, Cheng, Fuhua (Frank), "Surface Reconstruction from Point Clouds." SSM'98 Sculptured Surface Machining Conference "Machining Impossible Shapes" an International IFIP TC5/WG53 Conference, Auburn Hills, MI, proceedings to be published by Chapman Hall, November 1998.

[16] Vanco,. M., Brunnett, G. and Schreiber, Th., "A direct approach towards automatic surface segmentation of unorganized 3D points", Proc. Spring Conference on Computer Graphics 2000 (SCCG 2000), 2000.

[17] Vanco, M., Brunnett, G. and Schreiber, Th., "A hashing strategy for efficient k -nearest neighbours computation", Proceedings Computer Graphics International, June 7-11, Commore, Alberta, Canada, pp.120-128, 1999.

[18] Varady T. , Martin R.R. and Cox, J. , "Reverse engineering of geometric models-an introduction", Comput.-Aided Design (CAD), 29, pp.255-268, 1997.

[19] Varady,T., Benko, P. and Kos Geza, "Reverse engineering regular objects: simple segmentation and surface fitting procedures", International Journal of Shape Modeling 4(3), pp.127-141, 1998.

[20] Yi, B, K., Jagadish, H., & Faloutsos, Efficient Retreval of Similar Time Sequences Under Time Warping. IEEE International Conference on Data Engineering, P.201-208, 1998.

[21] Zanaty, E.A. ,M. R. Girgis and R. J. Gripps "Data reduction technique for reverse engineering", SCC Statistics and Computer Science and their Application, Cairo, pp.17-22, 2004.

[22] Oliver Fokerd "Improving Stereo Given Initial 3D Registration". Msc. Artifcial Intelligence Project Report, School of Infomatics, University of Edinburgh, 2009.

