

DATA REDUCTION ALGORITHM BASED ON PLANAR SURFACE FITTING

Fahad Alraddady

Computer Engineering Department
College of Computers and Information Technology,
Taif University, Taif, Saudi Arabia.
alraddady@tu.edu.sa

Abstract: Data reduction tools are developed and evaluated using a data analysis framework. Simple and intelligent thinning algorithms are applied to both synthetic and real data and the thinned datasets are ingested into an analysis system. A major problem of data reduction 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 preprocessing 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 computing the surface normal which is fundamental in the most of reverse engineering algorithms. The normal vectors are calculated by fitting the best fit plane to the neighborhood. A point is assigned to normal and the angle between an arbitrary direction and the normal is obtained. The point data is subdivided into cells based on the angles, while the non-uniform cells are obtained. 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.

Keywords: Reverse engineering, median filtering, normal vectors, data reduction.

1. INTRODUCTION

Data reduction allows more data sets to be stored in a given amount of disk or memory space. It also reduces the time required for images to be sent over the Internet or downloaded from Web pages. Furthermore, data reduction is useful because it helps reduce the consumption of expensive resources such as hard disk space or transmission bandwidth [1]. The selection of a data reduction algorithm depends mostly on criteria of achievable compression ratio and the quality of reconstructed images.

Two fundamental components of reduction are redundancy and irrelevancy reduction. Redundancy reduction aims at removing duplication from the signal source (image/video). Irrelevancy reduction omits parts of the signal that will not be noticed by the signal receiver, namely the human visual system. In general, three types of redundancy can be identified:

- 1- Spatial redundancy or correlation between neighboring pixel values.
- 2- Spectral redundancy or correlation between different color planes or spectral bands [1].
- 3- Temporal redundancy or correlation between adjacent frames in a sequence of images (in video applications) [2].

The focused of this paper is limited in phase (1) where the establishing connectivity between adjacent points, reducing redundancy and merging point clouds taken from multiple views. A major problem in this phase is that certain types of scanner produce vast amounts of data, the processing of

which presents a serious problem. Rather than process all of this data at every stage of the reconstruction process, an alternative is to use a strategy in which the data is reduced and used to construct a model. The full set of data is only used to improve this initial model if necessary. The challenge in reducing the data is to maintain sufficient information from which to calculate the object reliably without distorting the surfaces or their boundaries. Being able to reduce such large data sets whilst maintaining the information and accuracy contained in the original data will be advantageous for surface reconstruction and hence for follow on activities, especially in the manufacturing process.

Recently, the researchers have been shown interest by data reduction algorithms to overcome the time consuming in data preprocessing. Aljahdali et al. [3] improved a data reduction method for reducing 3D points for reverse engineering. This technique is based on a discrete Gaussian image of the scanned points which is obtained by estimating surface normals and projecting them into a Gaussian sphere. The discrete Gaussian image is then used to partition the points into cells. In each cell, a reference point and its neighbors are used to determine the cell representative point and all other points are removed. The performance of the proposed method is illustrated using a range of point clouds scanned from typical engineering surfaces.

Lazarus and Splitt [4] proposed thinning algorithms for applying to both synthetic and real data. As a precursor to real-data applications, the algorithms are applied to one- and two-dimensional synthetic datasets. Piotr et al. [5] found way of data reduction can be used with synthetic transmit

aperture method (STA) and it bases on an assumption that a signal obtained from any pair of transducers is the same, no matter which transducer transmits and which receives. According to this postulate, nearly a half of the data can be ignored without image quality decrease.

Surprisingly, little work has been done to combine real data reduction and hold only the pre-processed data for further analysis, especially in 3D data sets. 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.

This paper presents a fast algorithm for data reduction of the given 3D points. The proposed algorithm consists of three steps. In the first step, the neighborhood of the points is estimated, whereby the surface normal is obtained using planar fitting. In the second step, point data is subdivided into cells based on surface normal. The procedure in third step is used to reduce the data in each cell.

This paper is organized as follows: In section 2, the proposed algorithm is described including the neighborhood is computed, fitting the plane surface to neighborhood, normal vectors are estimated, partitioning the data into cells are presented and reducing 3D data reduction. To show the efficiency of the proposed method, some examples are presented in section 3. We conclude the paper in section 4.

2. The 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:

- Neighborhood estimation
- Computing the plane surface to each neighbor
- Estimating normal vector
- Reducing 3D points.

2.1 Neighborhood estimation

A 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. An ideal definition of neighborhood for the purpose of feature point extraction was presented in [6,7] 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.

2.2 . Computing Planer Surface to Neighbor

Assume a point data in which we want to manage it consists of *n* -points. We have the neighborhood points of a point $p_i = (x_i, y_i, z_i), i = 1, 2, \dots, k$, where *k* is the number of neighborhood and we want to fit the best plane to those points. Then the equation of the plane is:

$$Ax_i + By_i + Cz_i + D = 0$$

$$A^2 + B^2 + C^2 = 1$$

the plane is determined by solving the following least squares problem minimizing the distances between the points and plane:

$$\sum_{i=1}^k (Ax_i + By_i + Cz_i + D)^2 = 0$$

$$A^2 + B^2 + C^2 = 1$$

$$\sum_{i=1}^n d_i^2 = \sum_{i=1}^n \langle Ax_i + By_i + Cz_i + D \rangle^2 = \sum_{i=1}^n \langle Ax_i + By_i + Cz_i + D \rangle^T \langle Ax_i + By_i + Cz_i + D \rangle$$

$$\sum X^T \begin{bmatrix} y_i & z_i & 1 \end{bmatrix} \begin{bmatrix} y_i & z_i & 1 \end{bmatrix} \bullet X = X^T QX \tag{1}$$

$$\bullet X = \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix}$$

Also the constraint equation:

$$X^T \bullet N \bullet X = 1 \tag{2}$$

Where,

$$N = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

to solve the Eq. 1 with constant in Eq.2, using Lagrange method, we get:

$$Q - \lambda N \vec{X} = 0 \tag{3}$$

$$X^T N X = 1$$

We find λ by computing det. $Q' = Q - \lambda N = 0$

Where

$$Q' = \begin{vmatrix} \sum x_i^2 - \lambda & \sum x_i y_i & \sum x_i z_i & \sum x_i \\ \sum x_i y_i & \sum y_i^2 - \lambda & \sum y_i z_i & \sum y_i \\ \sum x_i z_i & \sum y_i z_i & \sum z_i^2 - \lambda & \sum z_i \\ \sum x_i & \sum y_i & \sum z_i & 1 - \lambda \end{vmatrix}$$

Assume $\lambda_1 \leq \lambda_2 \leq \lambda_3$ be the eigenvectors of Q' and (V_1, V_2, V_3) the corresponding eigenvectors. We substitute the smallest eigenvector λ_1 in Q and then Eq.3 becomes:

$$Q' = Q - \lambda_1 N$$

Or:

$$Q' = \begin{vmatrix} \sum x_i^2 - \lambda_1 & \sum x_i y_i & \sum x_i z_i & \sum x_i \\ \sum x_i y_i & \sum y_i^2 - \lambda_1 & \sum y_i z_i & \sum y_i \\ \sum x_i z_i & \sum y_i z_i & \sum z_i^2 - \lambda_1 & \sum z_i \\ \sum x_i & \sum y_i & \sum z_i & 1 - \lambda_1 \end{vmatrix}$$

We solve the nonlinear system:

$$Q' X = 0$$

$$X = \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} \vec{X}$$

by choosing arbitrary value of D.

2.3 Estimating Normal Vectors

Now, we have the planar surface that approximates p_i 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.

Then the normal vector \vec{n} can be computed from the relation

$$\left(\frac{A}{\sqrt{A^2}}, \frac{B}{\sqrt{B^2}}, \frac{C}{\sqrt{C^2}} \right)$$

2.4 Point data partitioning

Let the set of points which we want to handle is put in an arbitrary array with x, y and z direction. We used a binary search data structure to subdivide the point data as in Fig.(1). The point data is subdivided into two cells, which, each cell can be subdivided into two cells if it is need. We refer to the partitioning stage as levels, so the set of points is partitioned based on ℓ levels, where $level_0, level_1, \dots, level_\ell$ are the level's

partitioning and $2^0, 2^1, \dots, 2^\ell$ are the corresponding cells number respectively.

Algorithm 1:

- 1- $j=0, i=1$
- 2- Sort the points in A1 corresponding of x-direction in an array
- 3- Partition the data into two cells and put them in the sub-arrays A1 and A2
- 4- Go to algorithm 3 to check the data(i.e if the cell needs to partition again or not)
- 5- Sort the points in A1 corresponding of y-direction
- 6- Partition the points into two cells and put it into the sub-arrays B1 and B2
- 7- Go to algorithm 3 to check the data(i.e if the cell needs to partition again or not)
- 8- Sort the points in A1 corresponding of z-direction in an array
- 9- Partition the points into two cells and put it into the sub-arrays C_i and C_{i+1}
- 10- Goto 6 to select $B_1=B_2, i=i+2$;
- 11- Goto 6 to select $B_1=B_2, i=i+2$;
- 12- $j=j+1$.
- 13- Repeat the statements from 2-6 for each C_i until desired
- j.

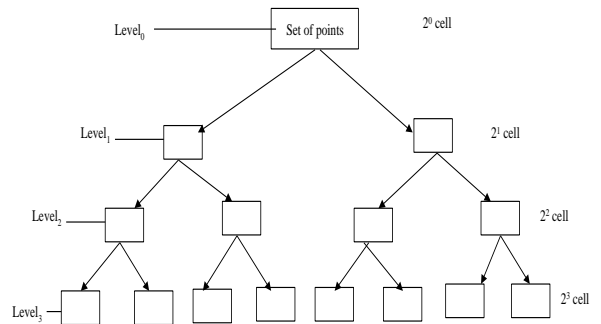


Fig. (1): The point data is subdivided into 2^3 cells.

2.5 3D point's 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 $p_i, i=1,2, \dots, \beta$, 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 $\theta_i, i=1,2, \dots, \beta$ between N and $\vec{n}_i, i=1,2, \dots, \beta$. 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:

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

3. Experimental results

The proposed technique is experimented using different types of part surfaces and the results are discussed. It was tested with real data measured using laser scanner and simulated data. Before applying the reduction algorithms, the farther noise point from the initial point cloud is removed. These tests have shown, for surface normal computation that 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 $k = 6$.

Table (1) shows some results that were experimented on the proposed data reduction technique. We have fitted the planes, spheres, and cylinders to the data using the Least squares method that is presented at [10] 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.

Fig.(2a) shows the point cloud of a planar surface (it is part from mechanical object) and it consists of 1654 points. The points is fed to the proposed algorithm and the result is presented in Fig.(2b). Since the examples in Fig.(3a,4a) show the 1854 and 2676 points has obtain from two different cylinder parts. The proposed algorithm has been tested by those two parts. Fig(3b, 4b) shows the result after applying the algorithm to the point cloud of a cylinder surface. The data in Fig.(4b) has been passed to the algorithm again and the result is presented in Fig.(4c). In the last example, we have tested the proposed method by simulated data to show the applicability of the method. In Fig.(3b) the sphere data is generated by sphere equation and add 50 noise points to the data. The data is passed to the algorithm and the output result has been shown in Fig.(3b).

The results show in table (1) that the fitted surface is better than that one before. For example, the cylinder segment (has 2676 points) is fed to our algorithm. We note, that it has 1900 points if the reduction degree is 1. The part with 1900 points is reduced again and the result is the part with 570 points (the reduction degree in this case is 2). However, in every reduction step a surface is fitted to the data point and the error is estimated. We noted that before reduction the error is 1.0094, after applying the algorithm to the data, the error became 0.00643. The output data is fed again to the algorithm, The data is fitted to the cylinder surface. Then the error became 0.00048. This shows that the propped method is very faithful to obtain accurate results while keeping quality point data.

4. Conclusion

A fast technique for data reduction is presented, where some experimental results of that technique are given to show the ability of the developed method. We have demonstrated that when surfaces are fitted to reduced points produced by the proposed technique, the fitting results are improved from those found by using the entire data point, even with quite large factors of data reduction. The proposed method overcomes the limitation of uniform grid [12] method by using the normal vectors assign to the points, which give information about part shape. Thus, all edges are preserved, and all differential properties can be estimated without introducing errors even with the noisy cells. Also, upon calculating the normal vectors of all points, a planar grid is generated. The size of a grid is defined automatically corresponding to the user threshold and grid size depends on the intended data reduction ratio for the given part shape.

SURFACE	Number of points	The error	Reduction degree
CYLINDER	2676	1.009	0
	1900	4	1
	570	0.006	2
	43	0.000	48
Plane	1654	0.967	0
	805	0.000	1
	96		
Cylinder	1854	2.003	1
	753	0.008	
Sphere	553	7	0
	287	2.005	1
		4	
		0.003	4

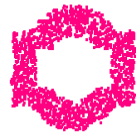


Fig. (2a): A planar data set 1654



Fig. (2b): After reduction, it has



Fig. (3a): A cylindrical set(1854 points) |



Fig. (3b): After reduction, it has 753 points and is fitted to plane and the error is 0.0087.



Fig. (4a): A cylindrical data : (2676 points) has fitted cylinder and the error is 1.009



Fig. (4c): The reduction degree it becomes 570 points and is fitted cylinder and the error is 0.006



Fig. (4b): The reduction degree is 1, it becomes 1900 points and is fitted to cylinder and the error is 0.00643.



Fig. (5a): Shows 543 points (include 50 noise points, simulated data), it has fitted to sphere the error is 2.0054.



Fig. (5b): Shows the result 287 points after running the algorithm, the result has fitted to sphere and the error is 0.0034.

Table (1): Some data sets have been tested by surface fitting.

[1] Vy, Vuong Dao , “Data reduction algorithms for wireless sensor networks in environment monitoring and warning applications”, Computational Intelligence, Modelling and Simulation (CIMSIM), Fourth International Conference on Date of Conference: 25-27 Sept. 2012.

[2] Chun-Lung Lin, Pei-Hsuan Tsai, Hsiao-Chuan Liang, Jia-Shung Wang, “Data reduction algorithm on the monitoring of extreme values in WSNs”, SENSORCOMM 2012 , The Sixth International Conference on Sensor Technologies and Applications, 2012.

[3] Sultan Aljahdali, E.A.Zanaty, Narayan Debnath, “Improving data reduction for 3D shape preserving, Journal of Computational Methods in Sciences and Engineering, Volume 9 Issue 1,2S1, pp. 35-51, April 2009.

[4] Lazarus, Steven M., Michael E. Splitt, Michael

D. Lueken, Rahul Ramachandran, Xiang Li, Sunil Movva, Sara J. Graves, Bradley T. Zavodsky, “Evaluation of data reduction algorithms for real-time analysis”, *Wea. Forecasting*, 25, pp. 837–851, 2010.

[5] Karwat, Piotr, Klimonda, Ziemowit, Sęklewski, Michał, Lewandowski, Marcin, Nowicki, Andrzej, “Data reduction method for synthetic transmit aperture algorithm”, Polish Academy of Sciences, Institute of Fundamental Technological Research, Pawińskiego 5B, 02-106 Warszawa, Poland, 35, Issue 4, pp. 635–642, 2010.

[6] E.A.Zanaty, “An Efficient method for 3D digitized data reduction for reverse engineering”, *IJCIS Journal*, June, pp.75-85, 2006.

[7] E.A.Zanaty and M.R. Girgis, “Collect the fitted surface into complex based on C^0 continuity”, *Journal of universal computer*, Vol.11,No 6., pp. 1102-1114, 2005.