MJIT 2017 Malaysian Journal of Industrial Technology

RECONSTRUCTION OF SMOOTH SURFACE BY USING CUBIC BEZIER TRIANGULAR PATCH IN GUI

Noorehan Awang[1,2,a] and Rahmita Wirza Rahmat[1,b]

*Corresponding author :
gs43550@student.upm.edu.my

[1] Faculty of Computer Science and Information Technology, Universiti Putra Malaysia,43400 UPM Serdang, Selangor,Malaysia

[2] Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM), Malaysia

[a]gs43550@student.upm.edu.my,
[b]rahmita@fsktm.upm.edu.my

Abstract

A smooth surface reconstruction by using the non-uniform distribution of the data points constitutes a major challenge in the technique involving reconstructing surfaces in the field of reverse engineering. In this study, the Delaunay triangulation process was selected as the method was not involve the deletion of the sample data points and could preserve the original surface topology. The study also focused on developing a smooth surface using two processes, i.e., the derivative estimation and the surface interpolation using the cubic Bezier triangular patch. We estimated the 2nd order partial derivatives by applying the least square minimization technique wherein the process was seen to decrease the errors between the estimated and the actual values. Thereafter, we interpolated the surface with the help of the cubic Bezier triangular patch. In this study, rather than using the results generated by multiple subplots in the MATLAB programming (MATLAB, R2015), the result was represented by using the MATLAB Graphical User Interface (GUI) function for the development of an interface for all the algorithms. Our results show a comparison of the interpolating surface generated by 6 test functions.

Keywords: Delaunay triangulation; Triangular patch; Non-uniform; Partial derivative; Least square minimization;

1.0 INTRODUCTION

Reverse engineering constitutes a method of developing and reproducing a Computer-Aided Design (CAD) model that resembles the actual object. This technique is generally used in cases when the actual design data is unavailable or is lost, or during the inspection and determining the quality of a manufactured product, improving the limitations of any product and also for creating the anatomical objects or relevant structures in the medical imaging field. Two processes are involved in this technique, i.e., the data acquisition and the surface reconstruction of a 3D model [1,2]. The data acquisition consists of a data set of cloud points which are divided into unstructured and structured data. The surface reconstruction is an automated process of generating 3D surfaces from the cloud data of the actual object which has been obtained from the 3D scan [3].

As already mentioned, [4], the data points which have been obtained from the scanned objects or structures are generally the unstructured points and their surface reconstruction can be complex as there is little or no information regarding the connectivity. Also, the surface reconstruction derived from the scanned objects, is faced with a noisy data, misaligned involving multiple scans, non-linear distortions and the lack of features that can lead to an faulty alignment of the scans and also some missing data when the surface scan could not be obtained, thereby resulting in a gap in the subsequent data set.

There is a need to find the best method to represent the smooth surface of scattered data obtained from 3D scanning and could preserve the original surface topology. There are several techniques have been used for the surface reconstruction like the Radial Basis Function (RBF), the Moving Least Square (MLS) and the Delaunay triangulation technique. Out of these, the RBF is a linear combination of the radially symmetrical basis function and it was used for defining the points which are not limited to the standard grid and involves no need for defining the mesh of the patches [5]. The RBF technique is applied in the case of reconstructing incomplete surfaces which consist of holes, but, it also has certain limitations. This methodology can only be used for smooth surfaces, is limited in the hole sizes and the identification of the holes requires a customer interaction [6]. As the RBF is very ill-conditioned for processing a huge data set, it had led to the introduction of the single and the multilevel quasi-interpolation that uses the Compactly Supported RBF (CS-RBF) [7]. However, it was seen that the CS-RBF could not handle the data set with sharp features.

On the other hand, the MLS is seen to be a meshless technique, which allows a local change in the fit results vary with the x value. This method can handle the hole-filling issue. One example was shown in [8], wherein the MLS was applied to many 3D polygonal models that contained holes, non-manifold edges, self-intersections, along with other such defects. Generally, the least square technique does not have to study all of the data points; however, the MLS examines all the points, thus proving that the MLS can provide a much surface fit and a better curve using the interpolating conditions [9].

The method use in this research known as Delaunay triangulation method is a very popular technique which is used for generating the triangle meshes, wherein the vertices are made up of the sample data points. This process uses the piecewise technique with the triangular patches, which are triangles in the 2D or tetrahedrals in the 3D. A triangulation containing a set of data points is made up of the vertices, edges (that connect the 2 vertices) and the faces (that connect 3 vertices). The triangulation helps in maximising the minimal value of the angles in the triangle and can avoid the skinny triangles [10]. This technique consisted of a property, wherein the circle which circumscribed the 3 vertices of the triangle had no more vertices. This has been explained in Fig. 1, where it is seen that the circle, C_1 does not contain the vertex, v_4, and the circle, C_2,

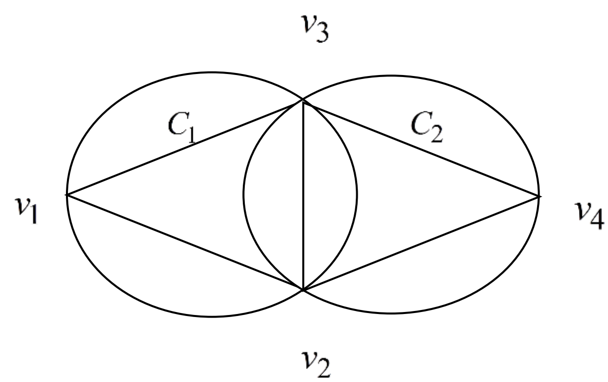

Fig. 1. Circumcircle property of Delaunay triangulation

Several techniques are available for the construction of the Delaunay triangulation for the dispersed data points like the randomised incremental insertion algorithm [11], incremental construction algorithm [12,13], the sweep-line algorithm [14,15], the circle-sweep algorithm [16], flipping algorithm [17], and the divide and conquer algorithm [18]. We have use the built-in function of Delaunay triangulation in MATLAB as it was easily understood and was simple.

For obtaining the smooth surface, the partial derivatives present at the triangle vertices had to be estimated and then the triangular patches generated were interpolated with the help of the Cubic Bezier triangular patch [19]. As the partial derivatives for the triangle vertices for the midpoint of every side are generally not available, they have to be approximated. The derivative vertices can be approximated by the help of the neighbouring data whereas the edge normal derivatives apply the normal derivatives from the 2 vertices that are linked to the triangle edge. The derivative approximation is carried out by using either of the 2 techniques, i.e., the convex combination [20] or the least square minimisation [21] technique. In our study, we have used the least square minimisation technique with the best fit for the quadratic surfaces for estimating the 2nd order partial derivatives for the vertices as it could be easily understood and implemented and used for a better tangential continuity instead of the convex combination technique that is only used for the 1st order derivatives.

In this study motivation of the research is due to the scattered data obtained from 3D scanning Delaunay triangulation method is the best method to represent the surface since it goes through all points ,it does not involve the deletion of the sample data points and could preserve the original surface topology. The objective of this study is to produced a smooth surface by using cubic Bezier triangular patch by using six test data functions and represent in GUI MATLAB.

2.0 EXPERIMENTAL

In this study, we used the technique of estimating the partial derivatives at every control point and the interpolation of the surface using the Cubic Bezier Triangular patch as in Fig. 2. General flow of our research study was explained further in the next subtopic below.

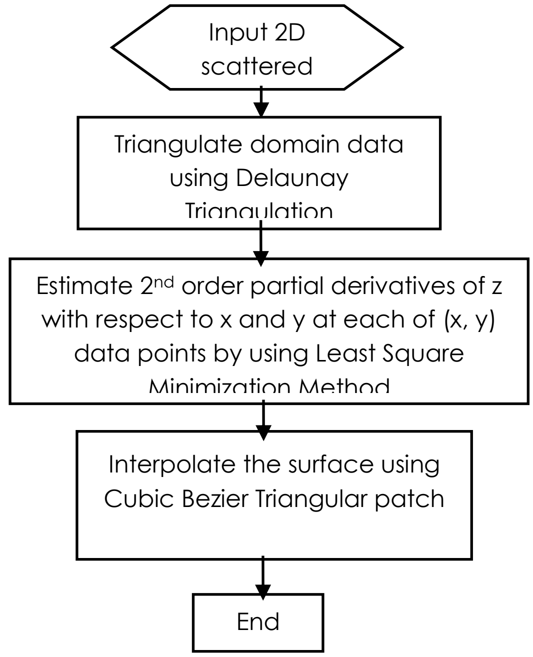

Fig. 2. Flow chart of this research

Triangulate domain data using Delaunay Triangulation

In this research, the built-in Delaunay triangulation in MATLAB 2015 was used to construct the triangular patch. The scattered data point was connected and represent in Fig.4 below.

Estimate 2nd order partial derivatives of z with respect to x and y at each of (x, y) data points by using Least Square Minimization Method

The least square minimisation technique, which was used earlier [21], has been applied in our study. This technique minimises the errors between the estimated and the actual values and determines the unknown coefficients after solving the Gaussian elimination. The best fit of the quadratic surfaces in the least square minimisation technique has been described in Eq. 1 as follows:

$$f(x,y) = ax^2 + bxy + cy^2 + dx + ey + f \qquad (1)$$

After substituting the values of $P_0, P_1, .., P_k$ and $z_0, z_1, .., z_k$, we can obtain the linear system for $Ax = B$

$$A=\begin{bmatrix} {x_0}^2 & x_0y_0 & {y_0}^2 & x_0 & y_0 & 1 \\ {x_1}^2 & x_1y_1 & {y_1}^2 & x_1 & y_1 & 1 \\ . & . & . & . & . & 1 \\ . & . & . & . & . & 1 \\ . & . & . & . & . & 1 \\ {x_k}^2 & x_ky_k & {y_k}^2 & x_k & y_k & 1 \end{bmatrix} \quad x=\begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} \quad B=\begin{bmatrix} z_0 \\ z_1 \\ . \\ . \\ . \\ z_k \end{bmatrix} \qquad (2)$$

The values for the coefficients of x are obtained by substituting the A and B values from Eq. 2 in the following equation:

$$x=\left(A^T A\right)^{-1} A^T B \qquad (3)$$

Using the value for x obtained in Eq. 3, we can determine the 1st partial derivative, written as:

$$\frac{df}{dx}=2ax+by+d \qquad \frac{df}{dy}=bx+2cy+e \qquad (4)$$

The second order derivatives are:

$$\frac{df}{dxx}=2a \qquad \frac{df}{dyy}=2c \qquad \frac{df}{dxy}=b \qquad (5)$$

Interpolate the surface using Cubic Bezier Triangular patch

Bezier Triangular Patch

The Bernstein polynomials for the degree, n, for the triangle, T, are defined using the barycentric coordinates (u,v,w) as follows:

$$B^n_{i,j,k}(u,v,w)=\frac{n!}{i!\,j!k!}u^i f^j w^k \qquad (6)$$

This is seen to form a base for the bivariate polynomials of the degree.

The parametric equation in the case of the triangular Bernstein Bezier patch, described below, uses Eq. 9 in the form of the Bernstein polynomial.

$$p(u,v,w)=\sum_{\substack{i=j=k=n \\ i,j,k=0}} b_{i,j,k}B^n_{i,j,k}(u,v,w) \qquad (7)$$

Wherein, the coefficients $b_{i,j,k}$ are known as the Bezier control points for $p(u,v,w)$. We have used the cubic-Bezier triangular patch [22, 23], described in Eq. 11 as follows:

$$\begin{aligned} P_3(u,v,w) &= u^3b_{300}+3u^3vb_{210}+3u^2wb_{201}+3uv^2b_{120} \\ &+3uw^2b_{102}+v^3b_{030}+3v^2wb_{021}+3vw^2b_{012} \\ &+w^3b_{003}+6uvwb_{111} \end{aligned} \qquad (8)$$

Where u,v,w are the barycentric coordinates and have a condition of $u+v+w=1$ and $u,v,w\geq 0$

Barycentric Coordinates

A point, P, present in the plane, could be represented as the linear combination of 3 vertices of the triangle, V_1,V_2 and V_3 as shown in Fig. 3 below:

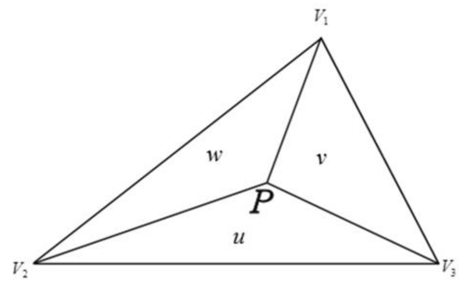

Fig. 3. Barycentric coordinate

$$\begin{aligned} P &= uV_1+vV_2+wV_3 \\ (x,y) &= u(x_1,y_1)+v(x_2,y_2)+w(x_3,y_3) \end{aligned} \qquad (9)$$

$$\begin{aligned} u+v+w &= 1 \\ ux_1+vx_2+wx_3 &= x \\ uy_2+vy_2+wy_3 &= y \end{aligned} \qquad (10)$$

Wherein, (u,v,w) are the barycentric coordinates of P for the three vertices, V_1,V_2 and V_3; while $u+v+w=1$ is seen to be equivalent to the linear system, $Ax=b$, wherein:

$$A=\begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} \quad x=\begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad b=\begin{bmatrix} 1 \\ x \\ y \end{bmatrix} \qquad (11)$$

After solving for the coordinates using the Cramer's rule, we obtain:

$$u=\frac{(x_2y_3-y_2x_3)-x(y_3-y_2)+y(x_3-x_2)}{\Delta}$$
$$v=\frac{(x_1y_3-y_1x_3)-x(y_3-y_1)+y(x_3-x_1)}{\Delta}$$
$$w=\frac{(x_1y_2-y_1x_2)-x(y_2-y_1)+y(x_2-x_1)}{\Delta}$$

Δ was the determinant for A, and could be represented as follows:

$$\Delta=x_1y_2+x_2y_3+x_3y_1-x_2y_1-x_3y_2-x_1y_3$$

3.0 RESULTS AND DISCUSSION

A test domain with 36 data points has been used earlier [24]. The points have been triangulated with the help of the Delaunay triangulation technique. All these coordinates of the 36 points are tabulated in Table 1 below:

Table 1.The coordinates of the 36 Data points

No	***Coordinates***		***No***	***Coordinates***	
	X	***y***		***x***	***y***
1	0.00	0.00	**19**	0.80	0.85
2	0.50	0.00	**20**	0.85	0.65
3	1.00	0.00	**21**	1.00	0.50
4	0.15	0.15	**22**	1.00	1.00
5	0.70	0.15	**23**	0.50	1.00
6	0.50	0.20	**24**	0.10	0.85
7	0.25	0.30	**25**	0.00	1.00
8	0.40	0.30	**26**	0.25	0.00
9	0.75	0.40	**27**	0.75	0.00
10	0.85	0.25	**28**	0.25	1.00
11	0.55	0.45	**29**	0.00	0.25
12	0.00	0.50	**30**	0.75	1.00
13	0.20	0.45	**31**	0.00	0.75
14	0.45	0.55	**32**	1.00	0.25
15	0.60	0.65	**33**	1.00	0.75
16	0.25	0.70	**34**	0.19	0.19
17	0.40	0.80	**35**	0.32	0.75
18	0.65	0.75	**36**	0.79	0.46

To illustrate the technique of producing smooth surface, the six well-defined test functions [21], as described in Eq. 12 – Eq. 17 was used. The six different test functions was used since it has been used regularly by other researcher when involving testing for surface reconstruction.

1. Franke's exponential function

$$F_1(x,y) = 0.75e^{-\left(\frac{(9x-2)^2+(9y-2)^2}{4}\right)} + 0.75e^{-\left(\frac{(9x+1)^2}{49}+\frac{9y+1}{10}\right)} + 0.50e^{-\left(\frac{(9x-7)^2+(9y-3)^2}{4}\right)} - 0.20e^{-\left((9x-4)^2+(9y-7)^2\right)} \tag{12}$$

2. Cliff function

$$F_2(x,y) = \frac{\tanh(9y-9x)+1}{9} \tag{13}$$

3. Saddle function

$$F_3(x,y) = \frac{1.25+\cos(4.5y)}{6+6(3x-1)^2} \tag{14}$$

4. Gentle function

$$F_4(x,y) = \exp\left(-\left(\frac{81}{16}\right)\left((x-0.5)^2+(y-0.5)^2\right)\right)/3 \tag{15}$$

5. Steep function

$$F_5(x,y) = \exp\left(-\left(\frac{81}{4}\right)\left((x-0.5)^2+(y-0.5)^2\right)\right)/3 \tag{16}$$

6. Sphere function

$$F_6(x,y) = \sqrt{64-81\left((x-0.5)^2+(y-0.5)^2\right)/9-0.5} \tag{17}$$

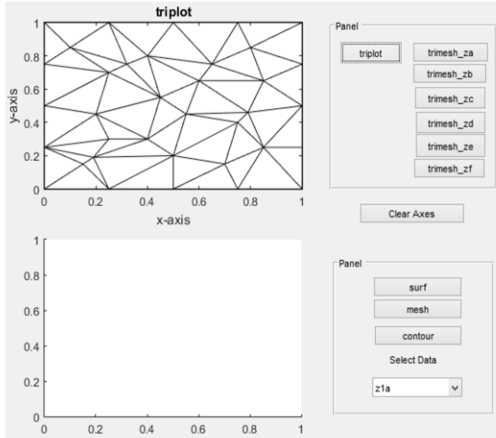

Fig. 4. The Delaunay triangulation for the 36 data points

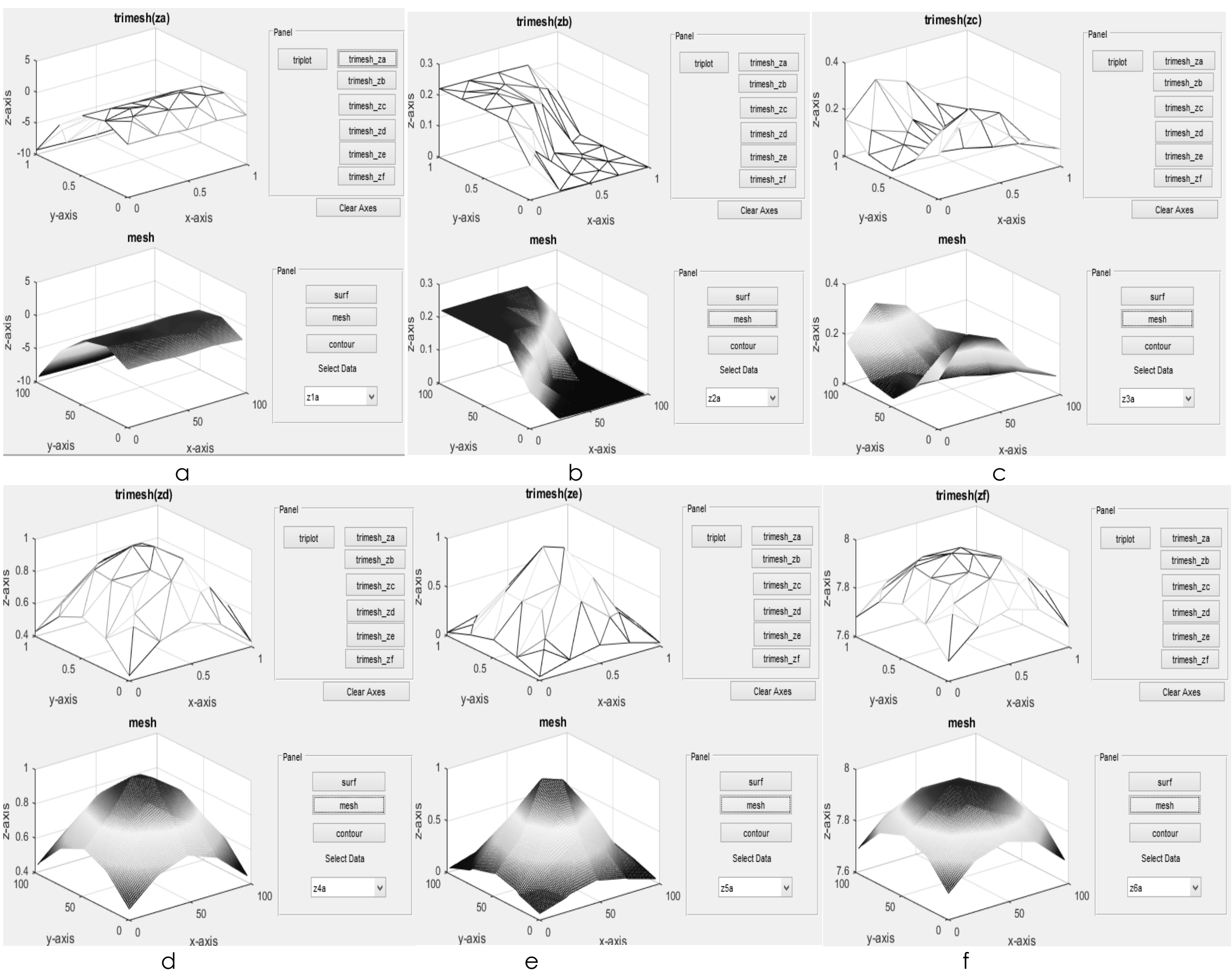

Fig. 5. The triangular representation in 3D and the mesh for 6 test functions in GUI (MATLAB 2015) which are (a) Franke's Exponential, (b) Cliff, (c) Saddle, (d) Gentle,(e) Steep and the (f) Sphere function.

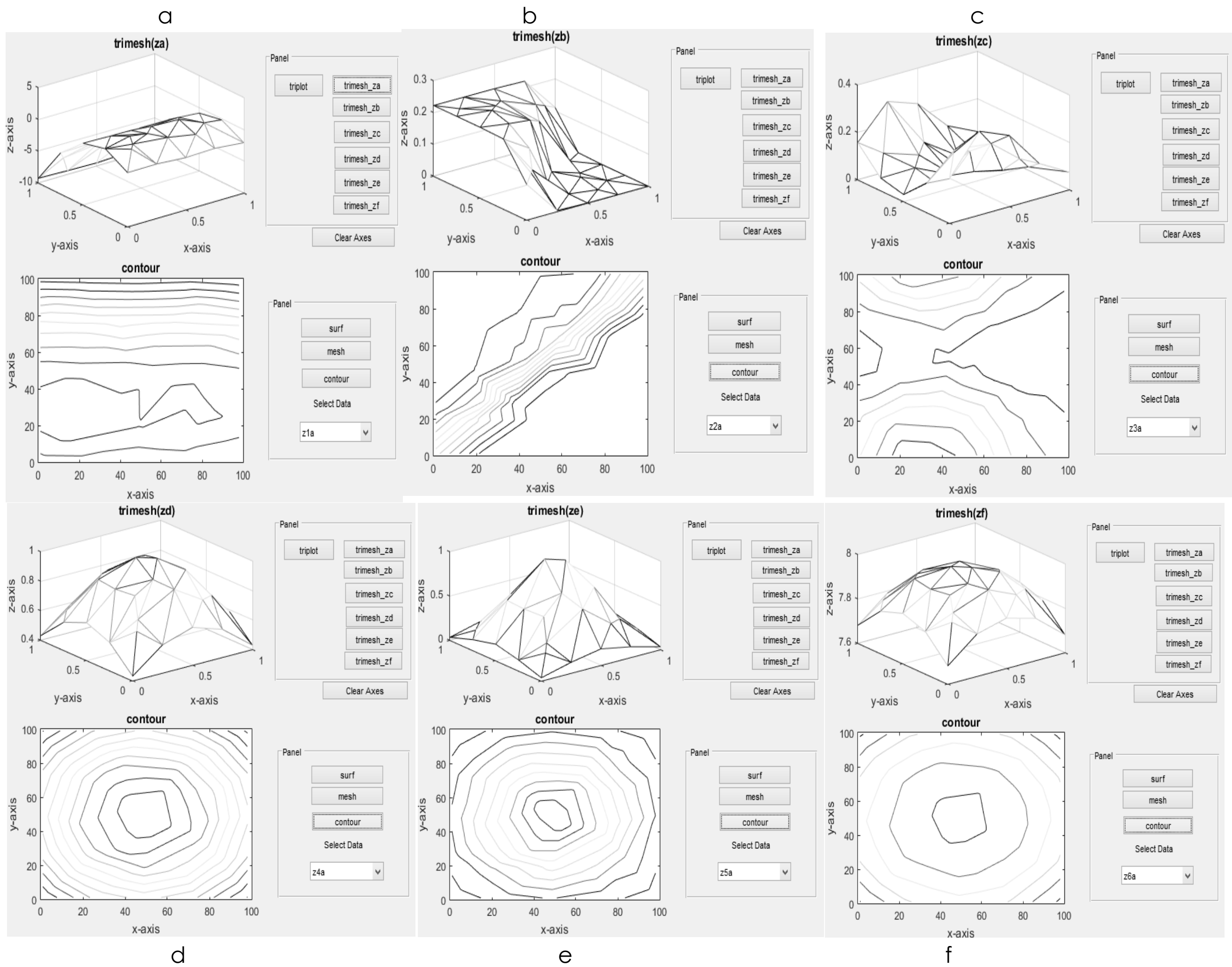

Fig. 6. The triangular representation in 3D and the contour for 6 test functions in GUI (MATLAB 2015) which are (a) Franke's Exponential, (b) Cliff, (c) Saddle, (d) Gentle,(e) Steep and the (f) Sphere function.

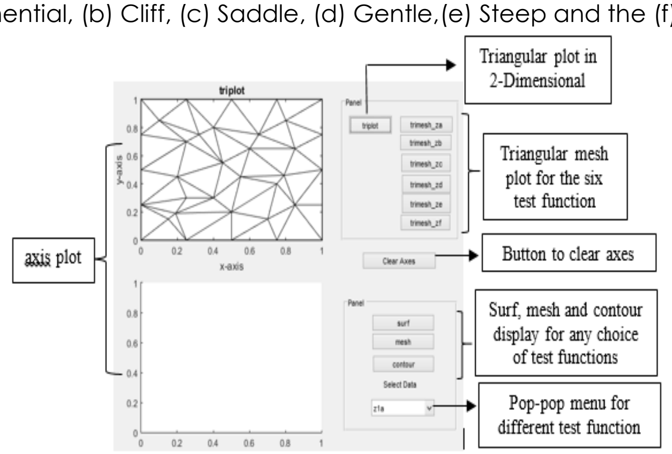

Fig. 7. Interface of the GUI

We used the MATLAB programme (MATLAB, R2015) for coding the surface results. In this study, instead of displaying the results of multiple subplots, we used the Graphical User Interface (GUI) function of MATLAB, for developing the algorithm interface. A GUI refers to a graphical display present in either one or many windows containing the controls called as the components which enable the user to carry out many interactive tasks. The GUI user does not need to create any kind of script or typing of any commands at the command line for carrying out these tasks. The different GUI components are toolbars, menus, push buttons, list boxes, radio buttons, and sliders. The GUIs, which have been created with the help of the MATLAB tools, are able to carry out any computation, can read or write the data files, and communicate with the other GUIs and the data is displayed in the form of tables or plots.

In Fig. 7, we have shown the GUI interface. The 2-axis plot is easier for displaying the results. The 11 push-button was used for displaying the surf, triangular mesh, mesh and the contour, for all 6 test functions. Finally, using the clear axis button enabled the user to erase the earlier results for displaying the subsequent plot properly.

Fig. 4 represents the Delaunay triangulation method for the 36 data points which were generated from Table 1 using the GUI function of the MATLAB R2015. We used an incremental construction algorithm for constructing the triangulation. Thus, it could follow the triangulation properties where the circle circumscribed the 3 vertices in the triangle and did not consist of any additional vertices. Also, the tri-plot push button present in the GUI represented the triangular plot in the case of the Delaunay triangulation method for a 2D surface.

Fig. 5 showed the triangular representation in 3D and the mesh for the 6 test functions. All the test functions were obtained using Eq. 12 - Eq. 17. The triangular representation has been described in the axis 1 with the help of 6 push buttons, namely trimesh_za, trimesh_zb, trimesh_zc, trimesh_zd, trimesh_ze and trimesh_zf. Also, the mesh results have been displayed in the 2nd axis plot using the mesh push buttons. The user can select 6 test functions for the display with the help of the pop-up menu. In Fig. 5, we have represented the mesh results, while the surf represents the results for the surface. The main difference between the surf and the mesh was that the mesh results produced a wireframe surface which coloured the lines which connected the important points, whereas the surf displayed the surface face and the lines in colour.

In Fig. 6, we have shown the triangular representation of the 3D and the contour for the 6 test functions. Similar to Fig. 5, the triangular representation has been described in the axis 1 with the help of 6 push buttons. Also, the contour results have been displayed in the 2nd axis plot using the contour push buttons. The user can select 6 test functions for the display with the help of the pop-up menu. The contour plots are represented by the matrix Z, wherein the Z-axis was interpreted as the height with regards to an *x-y* plane

4.0 CONCLUSION

In this study, we were able to generate a smooth surface that involved the estimation of the derivatives using the least square minimisation method. Furthermore, the Cubic Bezier Triangular patch was used as the surface of the Delaunay triangulation. The significance of this research, the technique to construct smooth surface by using Bezier Triangular patch was explained and can be use by reference by other researcher. In this study, the surface coding was carried out using the MATLAB programme (MATLAB, R2015) and was displayed with the help of the Graphical User Interface (GUI) function of the MATLAB programme as the surf, mesh and contour interface for 6 test functions in the 2D (2-Dimensional) surface. Our results indicated that the GUI function in the MATLAB 2015 programme made it very easy for the user to show their MATLAB results while analysing, interpreting or drawing any conclusions from the results.

Acknowledgement

I would like to give my gratitude and thank you for Universiti Teknologi Mara (UiTM) and Kementerian Pelajaran Malaysia (KPM) scholarship for funding scholarship for the conferences.

References

[1] Goran S., Zoran L. and Ivan M. (2015). Reverse Engineering. *Springer International Publishing Switzerland*, 319-353.

[2] Shi, M., Zhang, Y. F., Loh, H. T., Bradley, C., & Wong, Y. S. (2006). Triangular mesh generation employing a boundary expansion technique. *The International Journal of Advanced Manufacturing Technology*, *30*(1-2), 54-60.

[3] Bolitho, M. G. (2010). *The Reconstruction of Large Three-dimensional Meshes*. Johns Hopkins University.

[4] Kazhdan, M., Bolitho, M., & Hoppe, H. (2006, June). Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing* (Vol. 7).

[5] Saaban, A.,Ahmad, N.,Hassan, M. H., Mansor, K. H., Mohamad, M. S. A., Alipiah, F. M. & Khalid K. (2012). Scattered data interpolation using combination method of triangular patches. *Monographs of Applied Mathematics.*

[6] Branch, J., Prieto, F., & Boulanger, P. (2006, January). A hole-filling algorithm for triangular meshes using local radial basis function. In*Proceedings of the 15th International Meshing Roundtable* (pp. 411-431). Springer Berlin Heidelberg.

[7] Liu, S., & Wang, C. C. (2012). Quasi-interpolation for surface reconstruction from scattered data with radial basis function. *Computer Aided Geometric Design*, *29*(7), 435-447.

[8] Shen, C. (2006). Building interpolating and approximating implicit surfaces using moving least squares (Vol. 68, No. 03).

[9] Ni, H., Li, Z., & Song, H. (2010, October). Moving least square curve and surface fitting with interpolation conditions. In *Computer Application and System Modeling (ICCASM), 2010 International Conference on* (Vol. 13, pp. V13-300). IEEE.

[10] Grise, G., & Meyer-Hermann, M. (2011). Surface reconstruction using Delaunay triangulation for applications in life sciences. *Computer Physics.*

[11] Guibas, L. J., Knuth, D. E., & Sharir, M. (1992). Randomized incremental construction of Delaunay and Voronoi diagrams. *Algorithmica*, *7*(1-6), 381-413.

[12] Fang, T. P., & Piegl, L. A. (1992). Algorithm for Delaunay triangulation and convex-hull computation using a sparse matrix. *Computer-Aided Design*, *24*(8), 425-436.

[13] Freedman, D. (2007). An incremental algorithm for reconstruction of surfaces of arbitrary codimension. *Computational Geometry*, *36*(2), 106-116.

[14] Fortune, S. (1987). A sweepline algorithm for Voronoi diagrams.*Algorithmica*, *2*(1-4), 153-174.

[15] Žalik, B. (2005). An efficient sweep-line Delaunay triangulation algorithm. *Computer-Aided Design*, *37*(10), 1027-1038

[16] Biniaz, A., & Dastghaibyfard, G. (2012). A faster circle-sweep Delaunay triangulation algorithm. *Advances in Engineering Software*, *43*(1), 1–13.

[17] Sibson, R. (1978). Locally equiangular triangulations. *The computer journal*, *21*(3), 243-245.

[18] Cignoni, P., Montani, C., Perego, R., & Scopigno, R. (1993, August). Parallel 3d delaunay triangulation. In *Computer Graphics Forum* (Vol. 12, No. 3, pp. 129-142). Blackwell Science Ltd.

[19] Chang, L. H. T., & Wong, Y. P. (2002). Partial derivative estimation using convex combination methods. In *Proceedings of the 7th Asian Technology Conference in Mathematics 2002 (WC Yang, SC Chu, T. de Alwis and FM Bhatti, eds.)* (pp. 477-486).

[20] Goodman M. P., Said, H. B., & Chang, L. H. T. (1995). Local derivative estimation for scattered data interpolation. *Applied Mathematics and Computation*, *68*(1), pp. 41-50.

[21] Renka, R. J., & Cline, A. K. (1984). A triangle-based C1 interpolationf method. *Rocky Mountain J. Math*, *14*(1).

[22] Kong, V. P., Ong, B. H., & Saw, K. H. (2004). Range restricted interpolation using cubic Bézier triangles.

[23] Saaban, A., Piah, A. R. M., & Majid, A. A. (2006, July). Positivity-Preserving Scattered Data Interpolating Surface using C^ 1 Piecewise Cubic Triangular Patches. In *Computer Graphics, Imaging and Visualisation, 2006 International Conference on* (pp. 490-495). IEEE.

[24] Whelan, T. (1986). A representation of a C2 interpolant over triangles.*Computer Aided Geometric Design*, *3*(1), 53-66.