3D Point Smoothening Using Modified Local Regression for Reverse Engineering Process

Nur Ilham Aminullah Abdulqawi[1] **and Mohd Salman Abu Mansor**[1,*]

[1]School of Mechanical Engineering, Engineering Campus, Universiti Sains Malaysia, 14300 Nibong Tebal, Pulau Pinang, Malaysia

[*]Corresponding author's e-mail: mesalman@usm.my

Abstract. This research will introduce an existing statistical curve fitting tool i.e. local regression into reverse engineering process. Instead of using the local regression on statistical data, it will be applied on 3D point clouds obtained from 3D scanners. As most 3D scanners capture the smallest surface details such as textured and non-smooth surfaces, it makes the 3D point cloud to surface conversion process problematic. By modifying this existing statistical tool of local regression with the proposed algorithm, it helps to smooth out rough surfaces obtained from 3D point cloud, minimising the error when generating the virtual surface from smoothened point cloud rather than from raw point cloud. The related method of performing the smoothening process on the 3D point will also be included in this paper. At the end, comparison of the smoothened point cloud and the unsmoothened point cloud will be visualised and discussed.

1. Introduction

Reverse engineering process always starts with using 3D scanners to obtain 3D point cloud data, but the 3D scanners also detect every single detail on the object's surface, including irregular surfaces. In actual situation, objects do not have perfectly smooth surfaces, moreover some objects have textured surfaces, and hence the data produced by 3D scanner are usually considered as non smooth data or known as noisy data. To overcome this problem, some of the conventional reverse engineering processes need to be modified for improvement.

Computer programming has contributed to industries over the recent years ranging from basic computer applications to complex data processing tools. Computer programming has also become one of the common computing tool for scientists and engineers to perform various technical and research works, making computer programming more popular in research institutions and universities recently.

This paper will target on computer programming and its application on smoothening 3D points and its effect on the CAD model's surface. A modified local regression as a tool will also be adapted to this 3D point smoothening process. With this modified local regression smoothening introduced in reverse engineering process, it can benefit the designers by reducing errors on surface generation process.

The objective of this research is to minimise error on the surface generation by including point smoothening in the beginning of the process flow.

2. Related Works

There are a few similar research works that have been done on setting up methods on creating 3D CAD model by reverse engineering. Bénière et al. [1] had introduced a template that comes with a standard three steps of process from mesh convert to geometric primitives, geometric primitives to wireframe, then only from wireframe to 3D solid model, however the examples showed had very distinctive geometrical outlook and does not come out raw from a 3D scanner. Another work of Schnabel et al. [2] focused on point cloud conversion to geometrical primitives, and the primitives are sub-classified into basic shapes. This approach is useful for point cloud that resembles regular form objects, free-form objects will have more difficulties in order to achieve the same result, unless the point cloud data are reduced to approximation of basic shapes.

Even for mesh to solid conversion process, it is not always a smooth process because meshes generated from 3D point cloud are not always flawless. That is why a lot of methods had been introduced to repair or overcome those flaws on meshes. Marchandise et al. [3] had introduced mesh repair algorithm for mainly hole filling purposes. Some other works are more specific on mesh process only like Müller and Schwab [4] had targeted mesh refinement for finite element models. Another mesh processing work of Wei et al. [5] proposed a two step approach to smoothen mesh, the method is seen to be successful but required more computation time compared to smoothening points.

To overcome the mesh to solid conversion error, one must target the real root cause which originates from the irregularity of the point cloud data. Khanna and Rajpal [6] had a slightly different approach which directly uses point cloud to convert to curves and lines, at the same time the point cloud are being curve fitted relatively so that the final produced curves are smooth, however the approach does not aim to proceed to produce any surface or solid. There are also works on detecting and removing unwanted point cloud data in order to generate smooth final results, Nurunnabi et al. [7] also [8] Wang and Feng, both used their own methods to detect outlier points. The research methods of either reducing the unwanted points or perfecting point cloud is important because the refined point clouds are more ready to be processed for mesh and proceed to final CAD model.

3. Methodology

Conventionally, converting from 3D point cloud to 3D CAD model starts with mesh generation from 3D point cloud first, followed by smoothening the mesh, and then the subsequent processes. The conventional process flow found in most commercial CAD softwares is to generate the mesh first and then smoothen the mesh later. The methodology proposed here is different where point smoothening is introduced before mesh generation. Figure 1 is the illustration of proposed process flow versus the common CAD software process flow concept which includes mesh smoothening process.

This data smoothening is seen as the most critical process, as successful point data smoothening can lead to successful subsequence processes such as mesh generation, surface generation and solid generation.

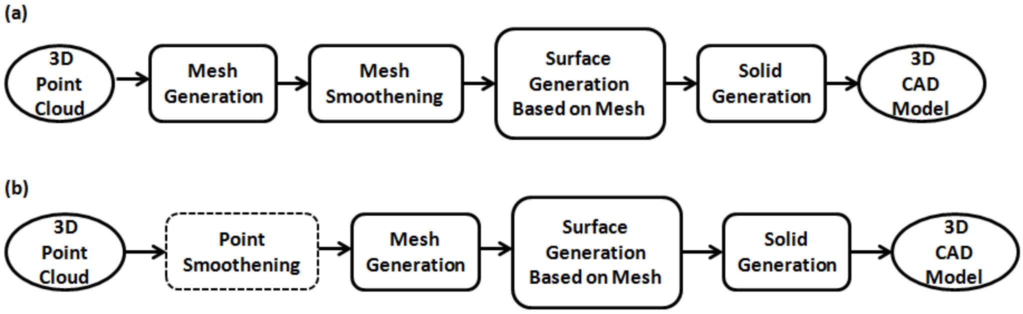

Figure 1. (a) Typical reverse engineering process for 3D point cloud to 3D CAD model conversion versus (b) Proposed reverse process with point smoothening for 3D point cloud to 3D CAD model conversion.

For 3D point smoothening, the local regression method is modified to suit the three dimensional applications specifically for reverse engineering, which is a local regression of using weighted linear least squares and a second degree polynomial model on the points. Furthermore, by doing iterative weighting on a specific range using the tricube function of this weight function can be simplified as

$$w_i = \left(1 - \left(\frac{|z-zi|}{r \times n}\right)^3\right)^3 \tag{1}$$

where, n is the total number points and r is the user defined constant

After specifying the range and input it to constant r, the specific i_{th} data point to be smoothed will have the largest weight influence on the fitting and data points outside the span have zero weight and no influence to the fitting. The smoothening will be based on weighted least square regression of second degree polynomial and the fitted x,y,z value is shown as

$$\hat{z}_i \equiv \sum_{i=1}^{N} w_i \left[a + b(x,y)_i + c(x,y)_i^2 \right. \tag{2}$$

The polynomial constants a, b and c are derived from the least square method based on [9] and applied for second degree polynomial properties. To obtain the constants a, b and c, the least square method equation can be rearranged as matrix for solution. The solution for a, b and c constant is shown as

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} 1 & \sum_{i=1}^{N}(x,y)_i & \sum_{i=1}^{N}(x,y)_i^2 \\ \sum_{i=1}^{N}(x,y)_i & \sum_{i=1}^{N}(x,y)_i^2 & \sum_{i=1}^{N}(x,y)_i^3 \\ \sum_{i=1}^{N}(x,y)_i^2 & \sum_{i=1}^{N}(x,y)_i^3 & \sum_{i=1}^{N}(x,y)_i^4 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{N} z_i \\ \sum_{i=1}^{N} z_i\,(x,y)_i \\ \sum_{i=1}^{N} z_i^2\,(x,y)_i \end{bmatrix} \tag{3}$$

The smoothening process is done based on the i_{th} point's distance to the next i_{th}+1 as a condition to fit this 3D point cloud regenerated from 3D scanner. The condition set for smoothening is that the furthest neighbour point has to be less than 0.05 of the overall sizing of the point cloud. Therefore the smoothened point cloud can be represented by

$$PC_{\text{smoothen}} = \begin{cases} [x, y, \hat{z}\ ] & , \ d \leq 0.05D \\ [x, y, z\ ] & , \ d > 0.05D \end{cases} \tag{4}$$

where, d is the distance between points of the consecutive order
D is overall sizing of the point cloud

4. Implementation and Discussion

The point smoothening effect may not be significant to the average human eye, but when it comes to the processes after mesh generation, smoother points will help prevent errors in subsequence processes, that is why smoother mesh are preferred over the raw point cloud when comes to reverse engineering processes. Figure 2 shows the comparison of the noisy data from an object, named as A, scanned with 3D scanner and the after smoothened point cloud.

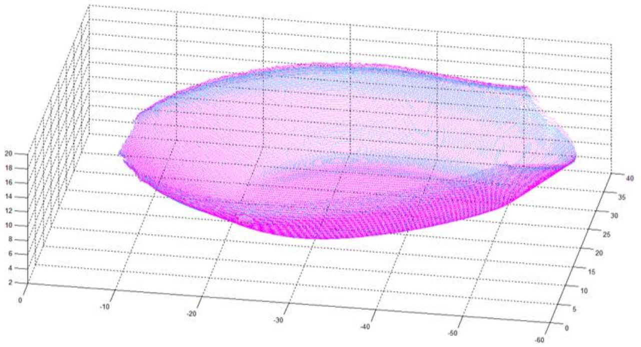

Figure 2. Illustration compare on raw 3D point cloud (magenta) and smoothened 3D point cloud (cyan) in the same scatter plot.

For fair comparison, both mesh sizes are set at 0.41mm distance from each point. Figure 3 shows the mesh comparison of mesh resulted from raw 3D point cloud versus the mesh resulted from smoothen 3D point cloud.

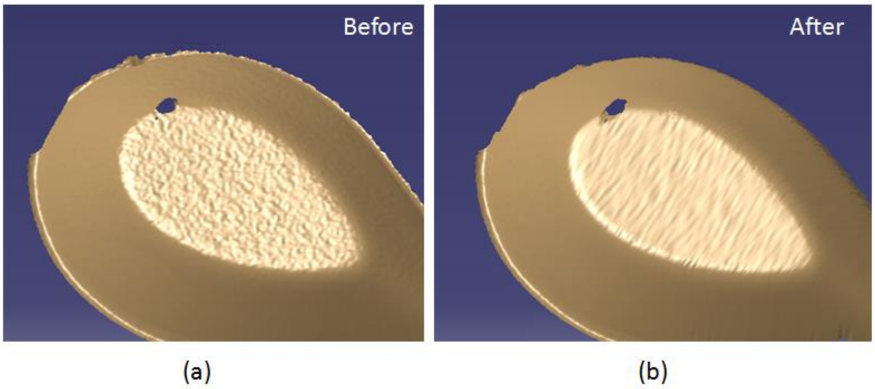

Figure 3. (a) 3D mesh from raw 3D point cloud using 0.41 mm mesh size (b) 3D mesh from smoothened 3D point cloud using 0.41 mm mesh size

The resulted smoothening effect from this proposed method should be discussed further in order to conclude that it has its own advantages. A small portion of point cloud data is extracted as per Figure 4 for further analyses. The points selected are labeled as A to K.

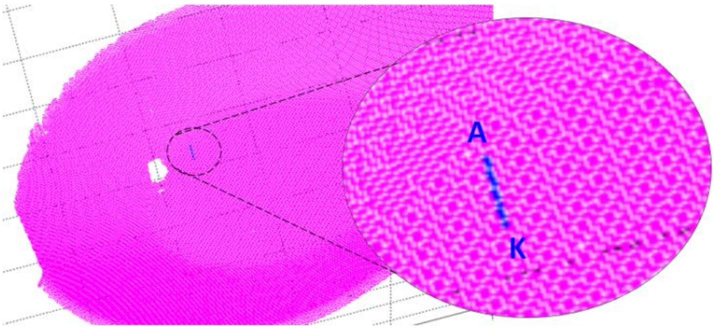

Figure 4. The selected portion of point cloud of A-K

The smoothening effect of point A-K is plotted in Table 1 and is visualised in Figure 5. The table also further computes on the distance of each points with the mean value of the selected portion. This is done to find out the surface roughness of each meshes, so that the smoothening effect can be numerically justified based on the surface roughness value.

Table 1. Extracted point data of A-K and difference of before and after smoothened together with calculated smoothening effect based on surface roughness difference

Point	X	Y	Z_{raw}	Z_{smooth}	Distance from mean (raw)	Distance from mean (smooth)	Squared distance (raw)	Squared distance (smooth)	Surface Roughness	
									R_{RMS} (raw)	R_{RMS} (smooth)
A	17.360	-7.405	11.589	11.674	-0.068	-0.041	0.005	0.002		
B	17.489	-7.426	11.592	11.683	-0.065	-0.032	0.004	0.001		
C	17.618	-7.448	11.597	11.692	-0.060	-0.023	0.004	0.001		
D	17.750	-7.448	11.603	11.700	-0.054	-0.015	0.003	0.000		
E	17.880	-7.460	11.610	11.709	-0.047	-0.006	0.002	0.000		
F	18.009	-7.472	11.626	11.717	-0.031	0.002	0.001	0.000	0.072	0.025
G	18.141	-7.471	11.648	11.724	-0.009	0.009	0.000	0.000		
H	18.271	-7.480	11.679	11.732	0.022	0.017	0.000	0.000		
I	18.401	-7.489	11.717	11.739	0.060	0.024	0.004	0.001		
J	18.532	-7.496	11.762	11.746	0.105	0.031	0.011	0.001		
K	18.663	-7.494	11.808	11.753	0.151	0.038	0.023	0.001		
Smoothening Effect = | R_{RMS} (smooth)- R_{RMS}(raw)| / R_{RMS}(raw) × 100%									65.19 %	

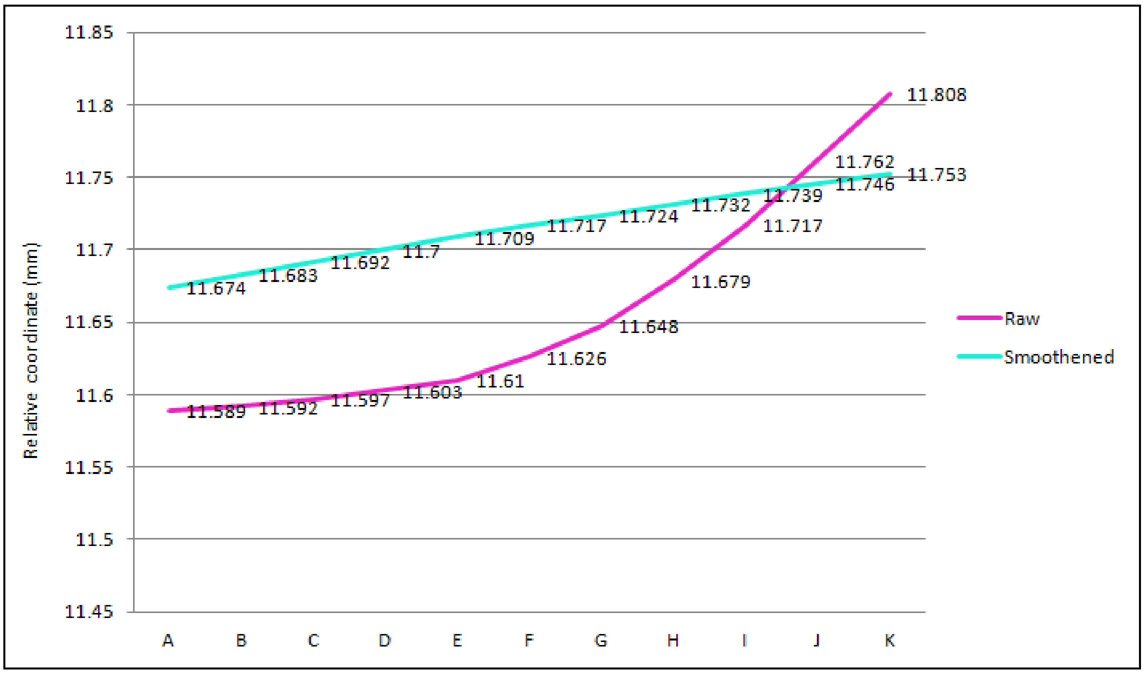

Figure 5. The effect of selected portion of point cloud of A-K before and after smoothened.

The tabulated data and plot showed smoothening effect numerically and graphically. The smoothening effect is recorded at 65.19% by based on comparing the surface roughness by root mean square, $R_{RMS.}$

Another specimen, named as B, is tested for the same smoothening effect. The smoothened effect is showed in Figure 6 as mesh.

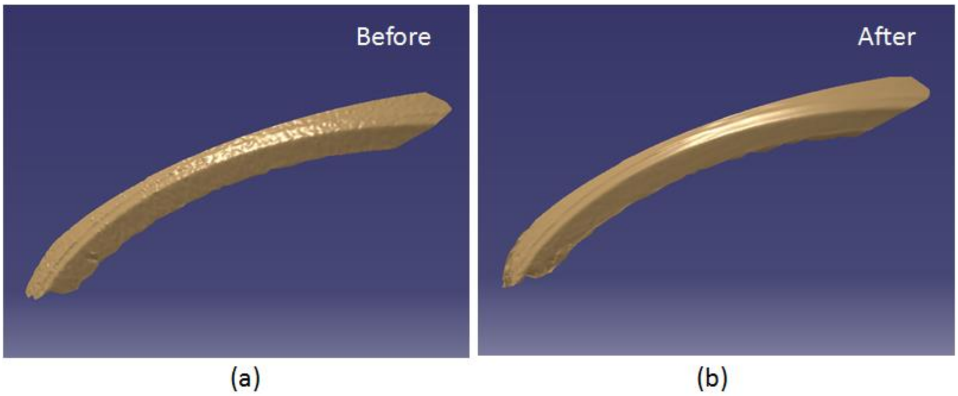

Figure 6. (a) 3D mesh from raw 3D point cloud using 1.3 mm mesh size (b) 3D mesh from smoothened 3D point cloud using 1.3 mm mesh size

The smoothening effect of this second specimen, B is also being quantified to another set of selected points, labelled as L to V as per Figure 7.

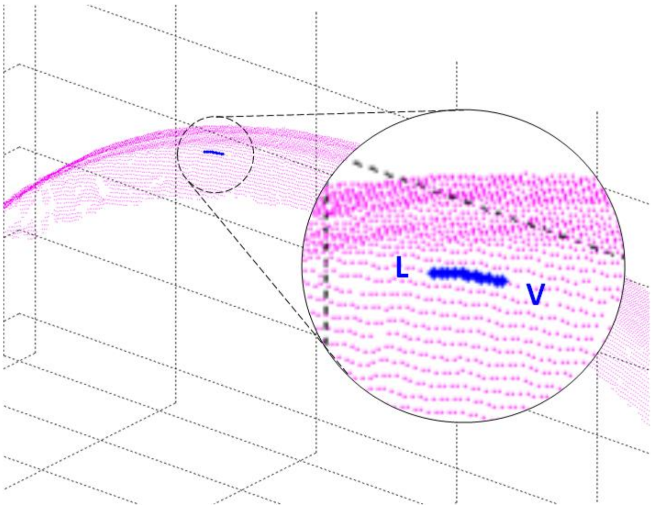

Figure 7. The selected portion of point cloud of L-V

The smoothening effect of point L-V is plotted in Table 2 and is visualised in Figure 8.

Table 2. Extracted point data of L-V and difference between before and after smoothened together with calculated smoothening effect based on surface roughness difference

Point	X	Y	Z_{raw}	Z_{smooth}	Distance from mean (raw)	Distance from mean (smooth)	Squared distance (raw)	Squared distance (smooth)	Surface Roughness	
									R_{RMS} (raw)	R_{RMS} (smooth)
L	27.691	-100.681	10.405	10.407	0.137	0.140	0.019	0.020	0.095	0.089
M	27.826	-100.686	10.382	10.379	0.114	0.112	0.013	0.013		
N	27.962	-100.695	10.346	10.351	0.078	0.084	0.006	0.007		
O	28.097	-100.701	10.322	10.323	0.054	0.056	0.003	0.003		
P	28.231	-100.705	10.305	10.295	0.037	0.028	0.001	0.001		
Q	28.366	-100.709	10.289	10.267	0.021	0.000	0.000	0.000		
R	28.501	-100.715	10.262	10.239	-0.006	-0.028	0.000	0.001		
S	28.638	-100.725	10.222	10.211	-0.047	-0.056	0.002	0.003		
T	28.775	-100.735	10.179	10.183	-0.089	-0.084	0.008	0.007		
U	28.912	-100.746	10.136	10.155	-0.132	-0.112	0.018	0.013		
V	29.049	-100.756	10.096	10.127	-0.172	-0.140	0.030	0.020		
Smoothening Effect = $\lvert R_{RMS}$ (smooth) - R_{RMS}(raw)$\rvert$ / R_{RMS}(raw) × 100%									6.91 %	

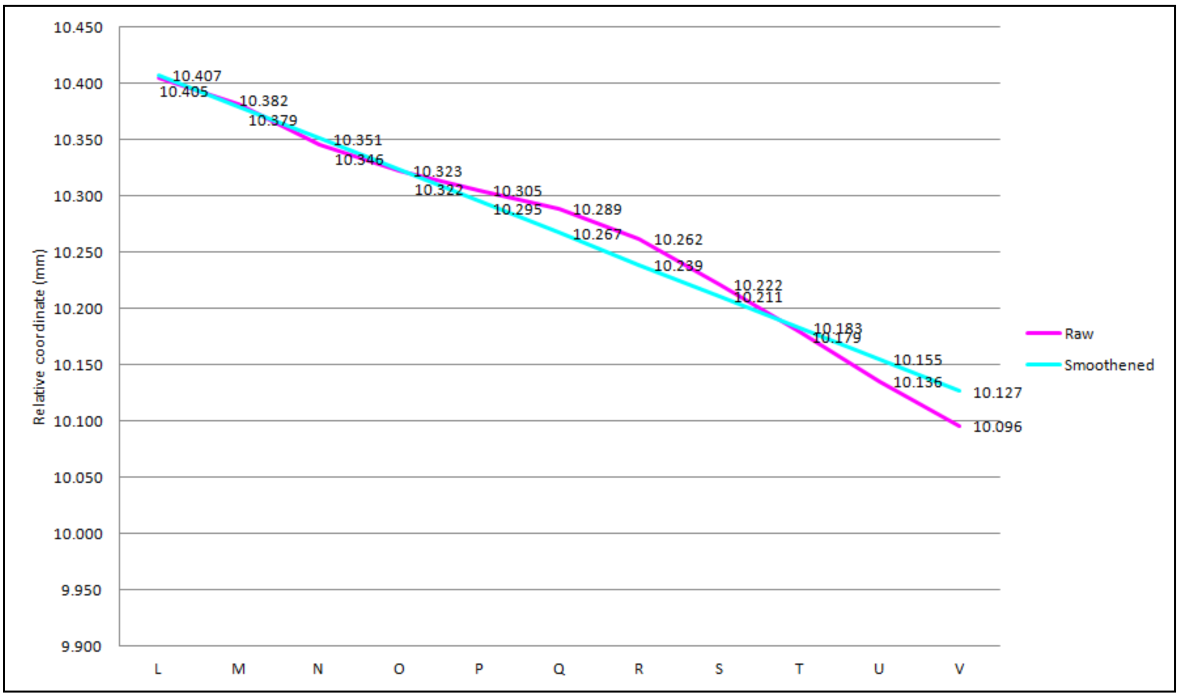

Figure 8. The effect of selected portion of point cloud of L-V before and after smoothening.

The tabulated data and plot showed smoothening effect numerically and graphically. The smoothening effect is recorded at 6.91% by based on comparing the surface roughness by root mean square, $R_{RMS.}$

For the second specimen object B, the smoothening effect of selected area is 10 times less significant compared to first specimen object A, based on the R_{RMS} percentage value. This is due to object A's point cloud has larger number of points compare to object B's. The equation used has a weight factor that is influenced by the number of points. The weight factor is inversely proportional to the number of points, making point cloud with higher number points less influenced by the weight of nearer points during smoothening. Hence this explained the point smoothening effect is significant for object A compare to object B.

Using computer to run the smoothening prior mesh generation can reduce subsequent error compared to using conventional reverse engineering process by CAD software. It is because most software does not have any point smoothening feature but mesh smoothening features instead, then the smoothened mesh has to go through various modification in order to come out as a surface.

Even though some CAD software has direct mesh to surface conversion option, most of the time errors will occur if the geometry of the object is not recognized by the CAD software. Instead to overcome this problem, users have to always create the drafting lines first, then users have to fill the surface according to the line boundaries one by one in order to create the complete surface model. Compared to this proposed methodology, after 3D point is smoothened, the subsequent tasks of creating surface model from point cloud will be easier with less error prompted by the CAD software since the smoothened points has better geometry representation than irregular raw points.

4.1. Benchmark with Mesh Smoothing Toolbox

Even though point cloud smoothening process is different from mesh smoothing process, but both results are comparable due to the point cloud can be easily triangulated into mesh representation. Mesh generation from smoothened point cloud using modified local regression and smoothened mesh from raw mesh using smoothing toolbox of CATIA software are supposed to have similar effect of removing textures found on raw 3D scanned virtual surfaces. Figure 9 is showing the effect of mesh generated from point cloud prior went through with modified local regression point smoothening, and also the mesh smoothened with CATIA's mesh smoothening toolbox with different iterative numbers.

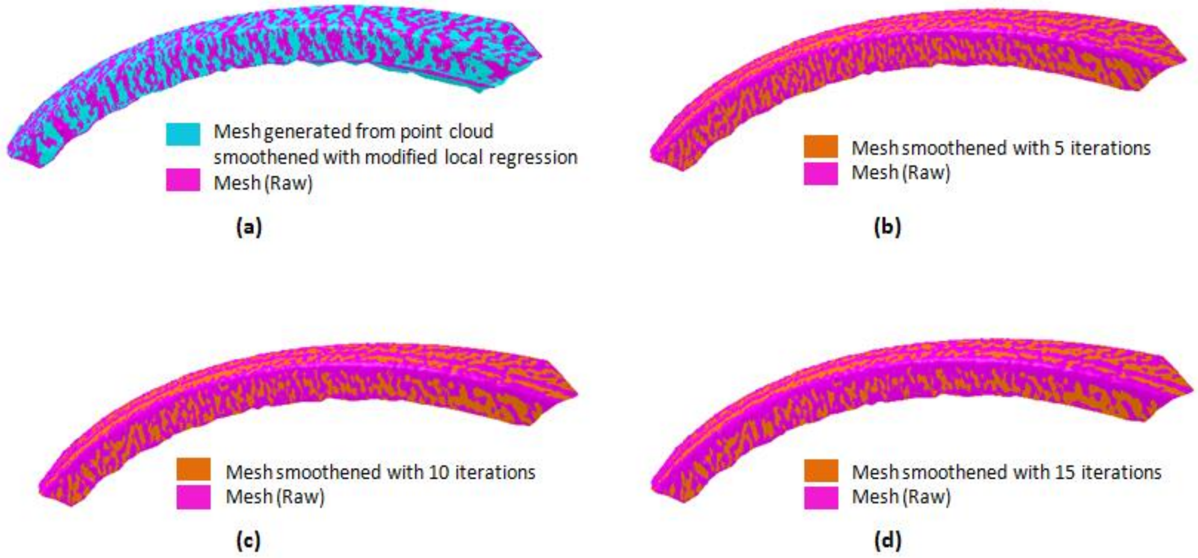
Mesh generated from point cloud smoothened with modified local regression
Mesh (Raw)
(a)
Mesh smoothened with 5 iterations
Mesh (Raw)
(b)
Mesh smoothened with 10 iterations
Mesh (Raw)
(c)
Mesh smoothened with 15 iterations
Mesh (Raw)
(d)

Figure 9. (a) Mesh smoothed with modified local regression point smoothening followed by mesh generation and mesh smoothed with mesh smoothing toolbox from CATIA with (b) 5 iterations, (c) 10 iterations and (d) 15 iterations.

Meanwhile the meshes are then mapped together for deviation analysis in order to verify smoothening performance of the modified local regression technique. Figure 10 is showing the different combination of meshed mapped together for deviation analysis.

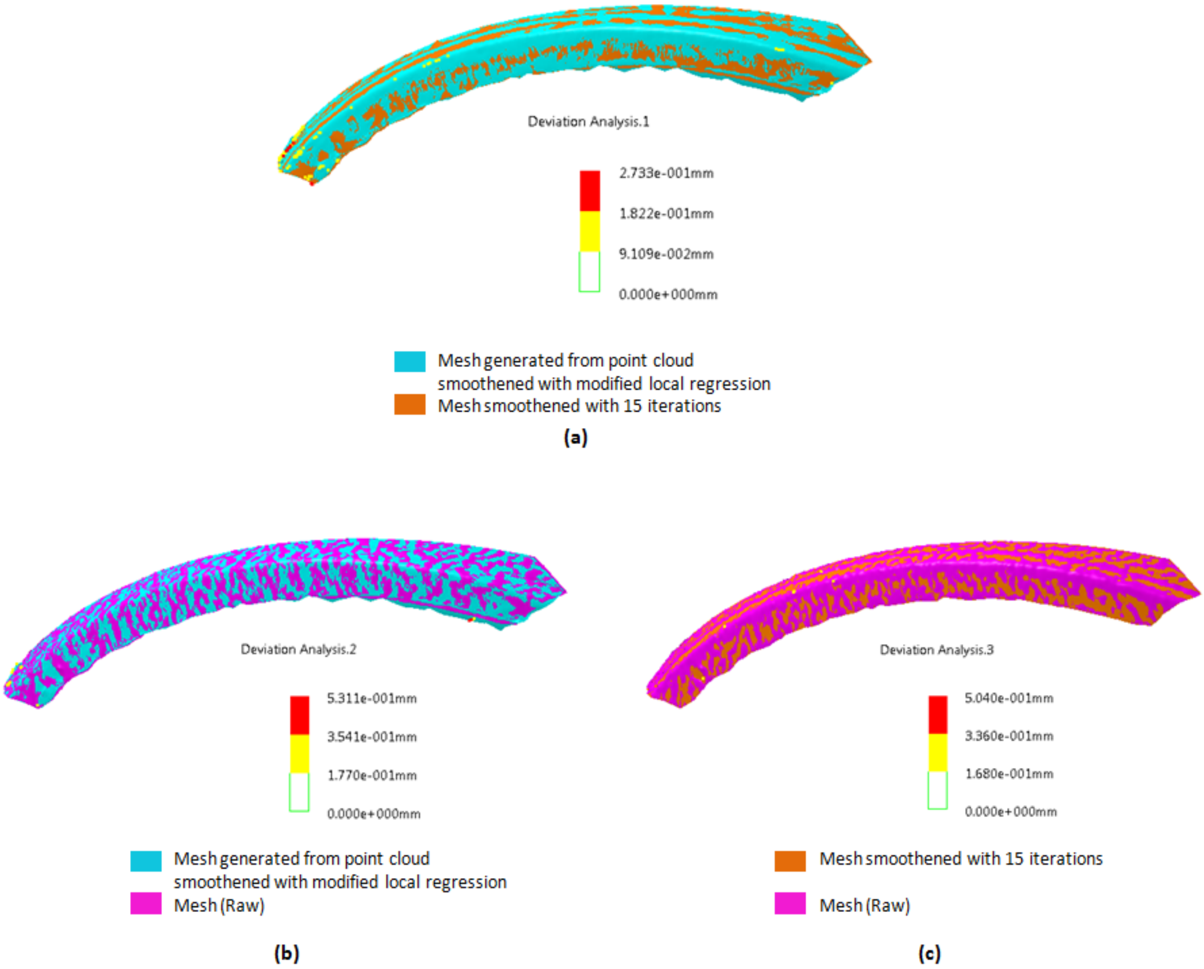

Figure 10. The deviation analysis of (a) deviation analysis 1, (b) deviation analysis 2 and (c) deviation analysis 3

The analysis based on Figure 10 shows that mesh generated from point smoothening using modified local regression does not deviate much from the mesh smoothened with CATIA’s mesh smoothing toolbox. By referring to the figure’s result of comparison for both mesh generated by point smoothening using modified local regression versus the mesh smoothened with 15 iterations CATIA’s mesh smoothing toolbox, majority of the deviation are at level of 9.109×10^{-2} mm.

Meanwhile the figure also shows that both methods of point cloud smoothening and mesh smoothening are comparable when mapped with the raw mesh. The deviation level is almost similar to each other except on the deviation locations.

5. Conclusion

This proposed modified local regression as a tool to smoothen 3D point clouds can lead to smoothened mesh. The smoothening effect of 3D points based on the tested specimens is recorded at 65.19% and 6.91% for both tested objects. This point smoothening technique will help reduce at least some certain percentage amount of error in subsequence process namely mesh generation. This means that smoothening of 3D points will reduce the error in creating the virtual surfaces in reverse engineering process, hence easing the users by reducing manual tasks of using 3D CAD software in reverse engineering.

Besides, this 3D point smoothening also introduces another process sequence for reverse engineering, where the point data are smoothened at the beginning rather than the mesh later.

Acknowledgement

This research is supported by Universiti Sains Malaysia under the Research University Grant (No: 814247). The first author also would like to thank the tuition fee sponsorship of MyMaster by the Ministry of Higher Education Malaysia.

References

[1] Bénière R, Subsol G, Gesquière G, Le Breton F and Puech W 2013 A comprehensive process of reverse engineering from 3D meshes to CAD models *Computer-Aided Design* **45** 1382–93

[2] Schnabel R, Wahl R and Klein R 2007 Efficient RANSAC for Point-Cloud Shape Detection *Computer Graphics Forum* **26** 214–26

[3] Marchandise E, Piret C and Remacle J-F 2012 CAD and mesh repair with Radial Basis Functions *Journal of Computational Physics* **231** 2376–87

[4] Müller F L and Schwab C 2015 Finite Elements with mesh refinement for wave equations in polygons *Journal of Computational and Applied Mathematics* **283** 163–81

[5] Wei M, Shen W, Qin J, Wu J, Wong T-T and Heng P-A 2013 Feature-preserving optimization for noisy mesh using joint bilateral filter and constrained Laplacian smoothing *Optics and Lasers in Engineering* **51** 1223–34

[6] Khanna K and Rajpal N 2015 Reconstruction of curves from point clouds using fuzzy logic and ant colony optimization *Neurocomputing* **161** 72–80

[7] Nurunnabi A, West G and Belton D 2015 Outlier detection and robust normal-curvature estimation in mobile laser scanning 3D point cloud data *Pattern Recognition* **48** 1404–19

[8] Wang Y and Feng H-Y 2015 Outlier detection for scanned point clouds using majority voting *Computer-Aided Design* **62** 31–43

[9] Weisstein E Least Squares Fitting From MathWorld- A Wolfram Web Resource *http://mathworld.wolfram.com/LeastSquaresFitting.html* n.p.n.d. Web. 16 Jan 2016.