

Altering Height Data by Using Natural Logarithm as 3D Modelling Function for Reverse Engineering Application

Nur Ilham Aminullah Abdulqawi¹ and Mohd Salman Abu Mansor¹

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

Email: mesalman@usm.my

Abstract. The raw data extracted from reverse engineering based on vision mostly do not resemble the actual geometrical representation yet. Even though the higher object surface reflected the most visible light towards the camera and yield higher number of value based on Lambertian illumination model, this does not mean the curvature profile are always accurate. After all, there are many mathematical models to shape curvature profiles into the correct representation. However, one of the most appropriate models found is the natural logarithm function. The function itself has alteration properties towards the raw data generated from reverse engineering based on vision.

1. Introduction

Digitization in reverse engineering process can be categorized into 3 main types: contact method [1], scan method [2] and also vision method [3]. The vision method is less common as it uses vision information to generate digitized 3D data, also known as 3D point cloud. The 3D data generated is based on Lambertian illumination reflection as the light source used is visible light. Several techniques has been used for targeting on shading and reconstructing surface from vision information and deemed to be successful [4]. However, most techniques use half tone or greyscale vision information which has limited input data. Furthermore, the accuracy range is still unable to achieve reverse engineering acceptable standard since most algorithms are developed for computer graphics rendering purpose. To suit for reverse engineering purpose, the vision information as input data has to be as compact as possible and that is why RGB information data is used for this research. Furthermore converting the input as height distribution needs a more uniform representation of the RGB information and hence equivalent corresponding color information HSV is necessary for the correction of height data distribution before inverting the function of light structure for the vision data.

2. Related Works

Vision information contain integers as data type [5]. When the integers are inputted into a certain program, they can be manipulated and be transformed into other sets of data. Previous research shows how integers can be harvested from image data for decision making and selective storage use [6]. Besides, integers value can also be transformed into useful outputs such as graphical models [7]. Somehow, integers vision information can be manipulated into 3D point cloud data using numerical conversion method for reverse engineering application[8]. However, the output data were found distorted at the peak of the object surfaces. To overcome this issue, there must be a function that able



correct the distorted height data while retaining the acceptable region of the 3D point cloud data that resemble the actual objects.

3. Methodology

The reverse engineering based on vision used setup as illustrated in figure 1. The camera function is to capture the object surface details as vision information and the captured vision information are transferable to a desktop computer via USB cable.

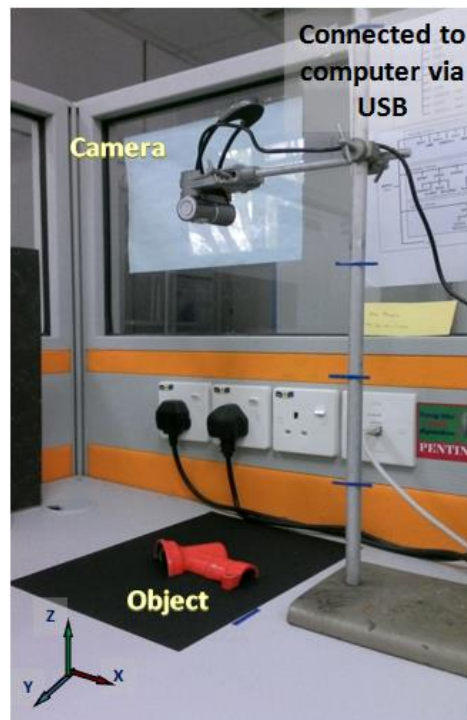


Figure 1. The setup of reverse engineering data extraction using camera to obtain vision information

The captured vision information from object surfaces are readily available as RGB format as default. Meanwhile, the RGB data are readily transferable to the computer memory as computer graphics. In computer graphics, RGB is usually being described as a 3D cube that has three axes of directions that each direction has a maximum value of 255, it is because RGB model is having three main components of color which is the red, green and blue with black is having value of 0 in each component and white is having value of 255 of each component.

Another type of color model that derived from RGB model is the HSV model, the HSV model is having three components namely the hue, saturation and value. HSV model is described as a hex-cone model because the colors are defined as angle in a 360 degree revolution, with the saturation increase by moving away from the centre axis of the hex-cone. The centre axis itself also contain value which the top of hex-cone has value of 1 while the bottom of the hex-cone has value of 0, the value direction is hence it means the increasing of value moves towards lighter colors of hue. The actual graphical description of both RGB model and HSV model is shown in figure 2.

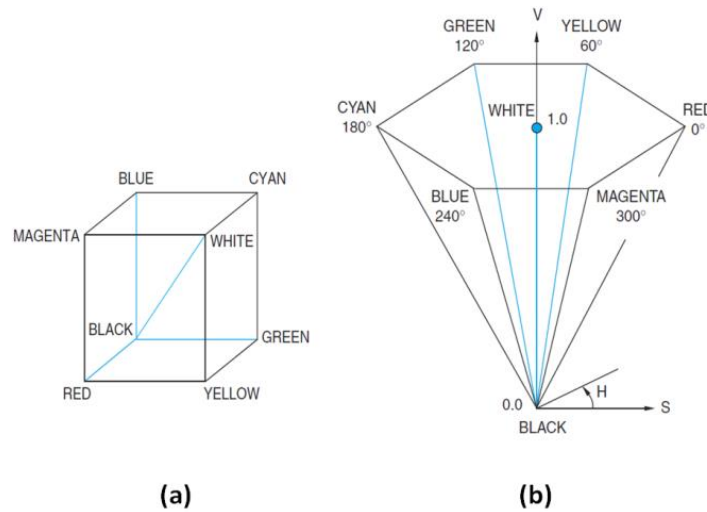


Figure 2. (a) RGB cube color model and (b) HSV hex-cone color model. Source from Elsevier [9]

Since HSV model data is the corresponding representation of the RGB model, while HSV model has the ability to interpret color as intensity distribution, it is used as the intermediate input for correcting the light structure of an object's surface captured as vision information.

The vision information structure of RGB and HSV is same accept of the composition of the components as shown

$$[RGB_{(i,j)}] \supseteq \{ [Red_{(i,j)}] \cup [Green_{(i,j)}] \cup [Blue_{(i,j)}] \} \quad (1)$$

$$[HSV_{(i,j)}] \supseteq \{ [Hue_{(i,j)}] \cup [Saturation_{(i,j)}] \cup [Value_{(i,j)}] \} \quad (2)$$

HSV model representation with subsequence process into height variation can be represented by the arrangement of X and Y and the equation can be represented by

$$[Grid_{X,Y}] = [pixel_{1,1}, \dots, pixel_{m,1}, pixel_{1,2}, \dots, pixel_{m,2}, \dots, pixel_{1,n}, \dots, pixel_{m,n}]^T \quad (3)$$

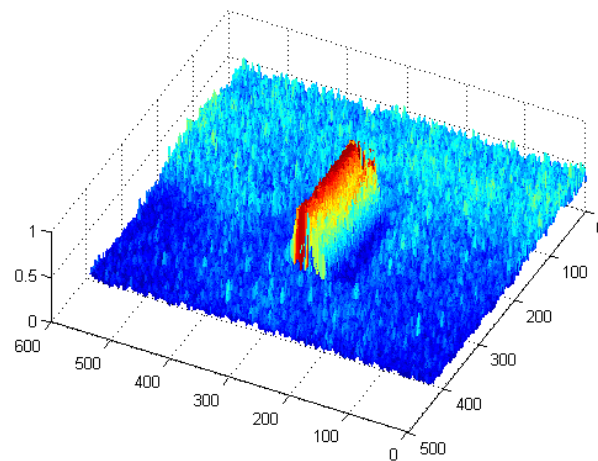
Meanwhile, the generated 3D data with the height variation of each X and Y is represented by

$$3D \text{ Data} = [X, Y, Height_{X,Y}] \quad (4)$$

At this point the 3D data with height data as coordinates generated are actually the raw data. To verify on how the raw data was distorted, a small cylindrical machining workpiece was used as reference. Figure 3 shows data distribution of raw data generated.



(a)



(b)

Figure 3. (a) The actual cylindrical profile used and (b) the raw data generated

By slicing the raw data into one side of the symmetrical profile, it shows that the generated raw data actually has distribution of exponential distribution. Figure 4 shows the detailed of the raw data with visible exponentially distributed pattern.

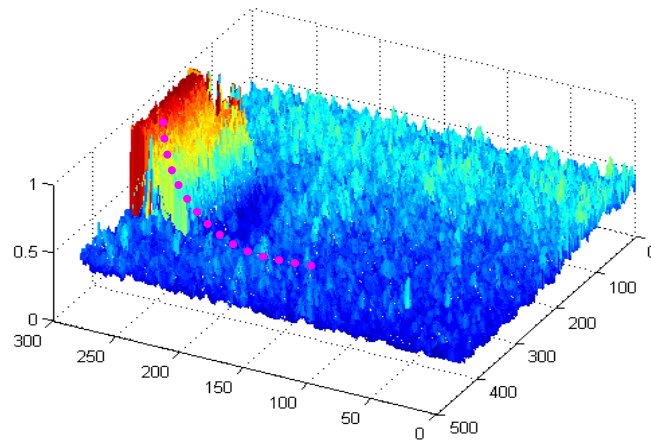


Figure 4. The sliced out data distribution from one side of the symmetrical profile

Meanwhile, by referring to the basic exponential equation of $y=e^x$, the starting y-value of the original exponential function for y is at 1, the equation was altered for interception at 0 to mimic the distributed pattern of HSV conversion model as

$$y = 0.582 (e^x - 1) \quad (5)$$

Since the raw data has more of an exponential heights representation in surface profile, a counteracting function towards this exponentially rising pattern has to be implemented to the 3D data in order to correctly represent the shape of the actual object. By treating the 3D data as 2D profile, height data distribution is visualized as 2D plot using Desmos graph plotting tool in Figure 5.

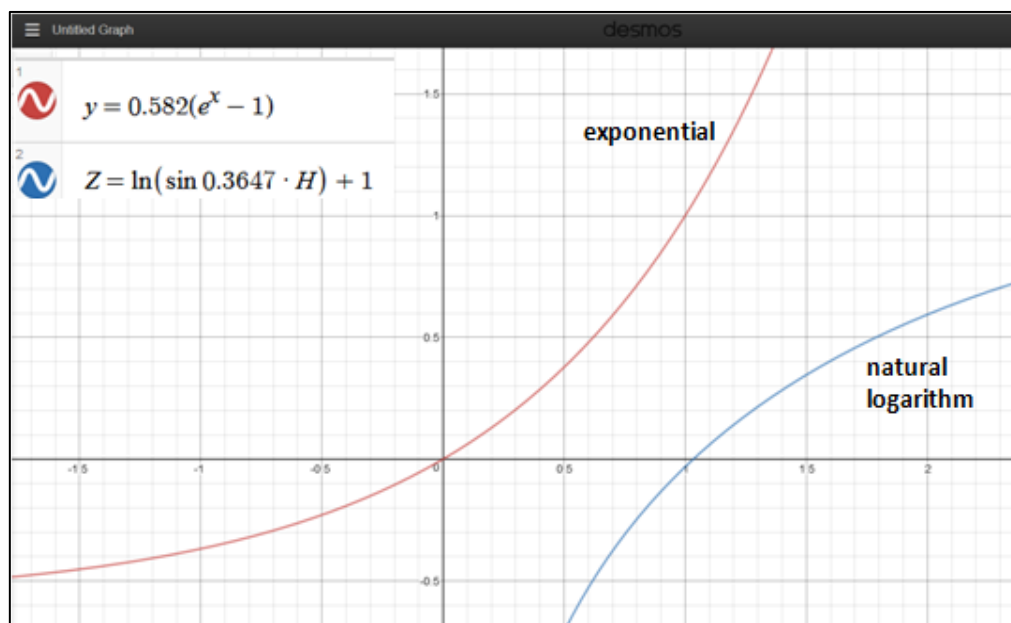


Figure 5. Exponential function of the distributed point cloud and the counteracting function is the natural logarithm function

To inverse the exponentially distributed height, the natural logarithm function used for this inverse effect is derived from the assumed point cloud's 2D profile distribution is

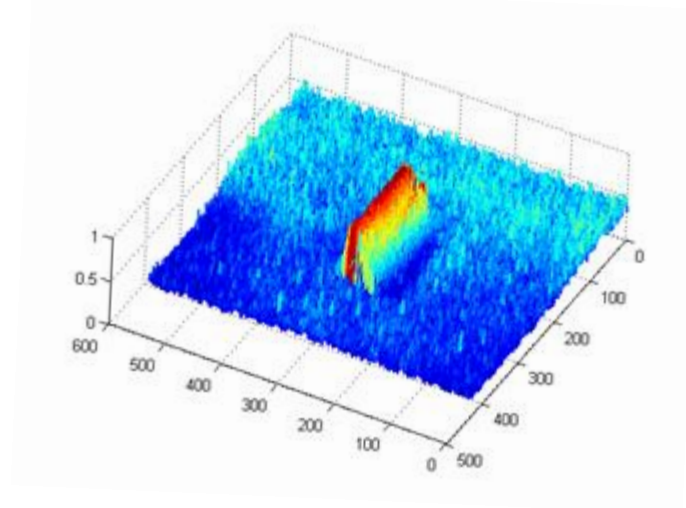
$$(6) \quad 3D \text{ Data}_{\text{Altered}} = [X, Y, \ln(\sin 0.3647 \times \text{Height}_{X,Y}) + 1]$$

At this point the 3D data should resemble the actual object surface profile. As to verify the equation, results and implementation was done and will be explained in the following section.

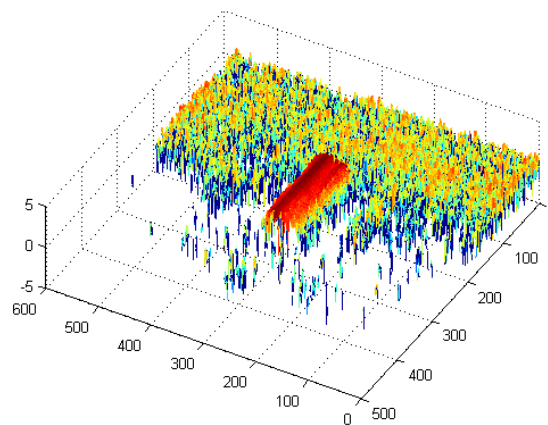
4. Result and Implementation

4.1. Intermediate result from test object

Using the cylindrical machining workpiece as test geometry, applying the mentioned natural logarithm function will enable the height data to be altered into the correct representation. Figure 6 shows the after applying natural logarithm function effect.



(a)



(b)

Figure 6. The results of (a) before and (b) after applying the natural logarithm function

Besides of observing the exponential profile being altered into natural logarithm profile, the background which initially were at 0 value altered into non-value as any logarithm of zero will resulted an indefinable output as $\log(0)$ will be infinity.

4.2. Implementation into actual object

Meanwhile for actual implementation, it was done on a bush cutter's cover and the so that the inversing effect can be visualised. The bush cutter cover is shown as in figure 7 and the object is captured from the top view before further going through additional processing into 3D point cloud.

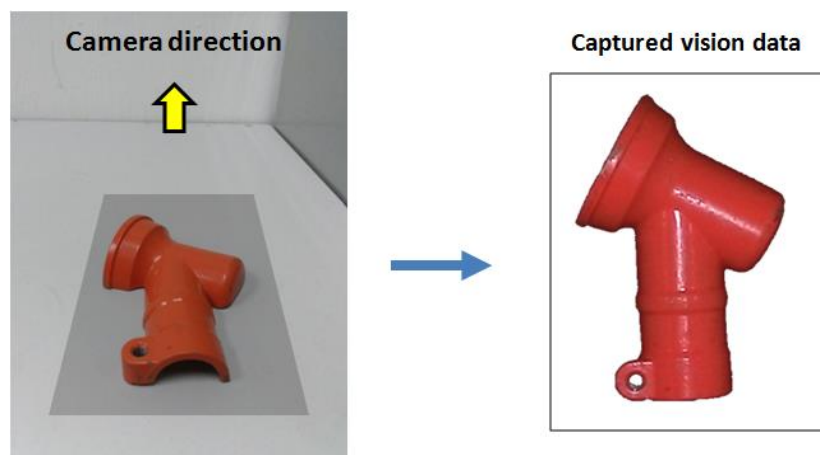


Figure 7. The specimen of bush cutter's cover placed below the camera to captured as vision data for further processing

The vision data that contains color and light intensity information as integers will be going through RGB to HSV conversion process and height extracting process in order to finally produced a set of 3D data or 3D point cloud in X,Y and Z coordinates which they resemble the surface detail of the object. The produce point cloud is readable by most computer aided design software. For this case, the software used is CATIA V5. Figure 8 shows the end result of the raw data point cloud right after processing of vision data.

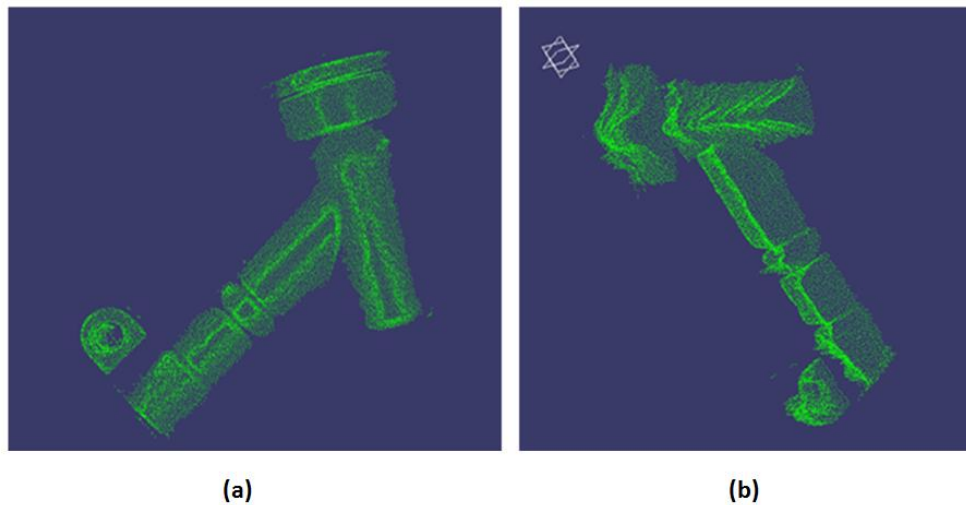


Figure 8. The point cloud generated right after height distribution is process from vision information with (a) first view angle and (b) second view angle

The point cloud at this point shows incorrect distribution of height which has exponential characteristic where the middle area had sharp increase in height data, while the lower height data are at the bottom and hence removed. In order to correct the height data distribution on the point cloud, the inverse function mentioned in above section is applied to enhance on the final output so that it resemble the actual surface profile of the object. Figure 9 shows the after inverse effect by applying the mentioned natural logarithm to counteract on the exponentially distributed data.

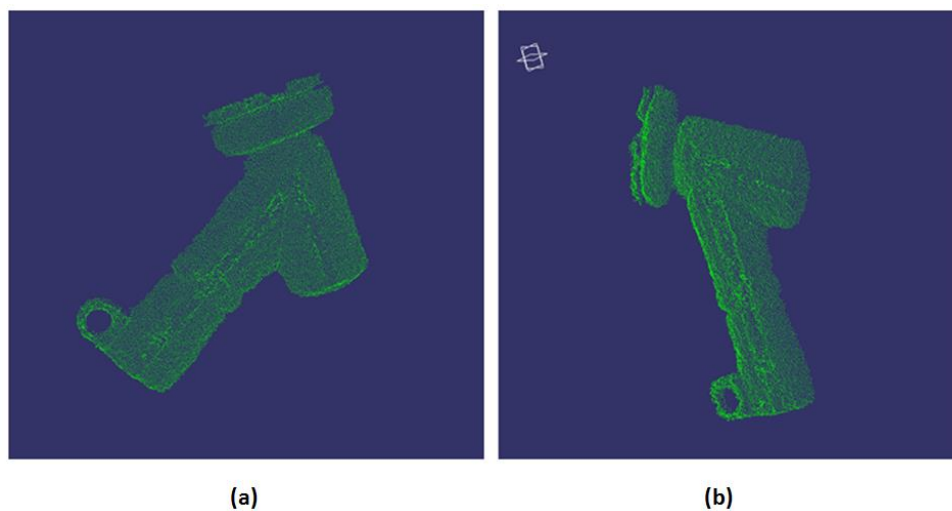


Figure 9. The point cloud representation after inverse function of natural logarithm is implemented with (a) first view angle and (b) second view angle

After seeing the altered height data of the 3D point cloud to have better resemblance distribution, they can be used to further generate mesh and the height distribution can be visualised with 3D plot of 3D mesh.

Figure 10 shows the 3D mesh with different color according to the individual heights of each data point.

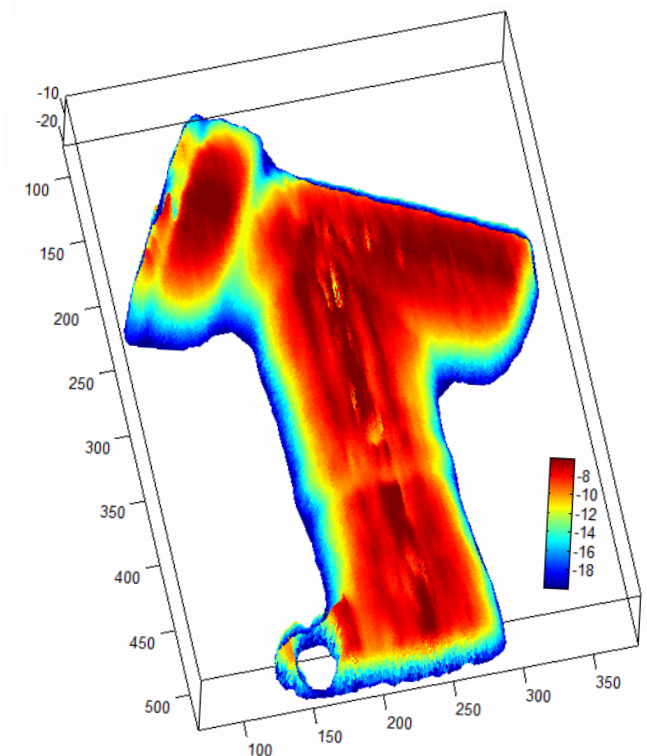


Figure 10. 3D mesh generated with different colors indicating the height data distribution.

5. Conclusion

Reverse engineering based on vision still faces problems in implementation. One of the problem is the generated height data distributions are inaccurate. Remodelling the height data distribution can correct the inaccuracy so that the following processes are errorless. Reverse engineering based on vision usually produced raw point clouds with exponentially distributed heights after applying HSV height extraction process. The proposed natural logarithm function is capable in solving the raw point clouds problem of having exponentially distributed heights. Furthermore, the proposed function improves the overall shape representation of the point clouds, and hence improves the overall accuracy for the outputs of reverse engineering that based on vision.

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] Pedone P, Audrito E and Balsamo A 2014 Compensation of CMM geometrical errors by the GEMIL technique: Experimental results *CIRP Ann. - Manuf. Technol.* **63** 489–92
- [2] Paulic M, Irgolic T, Balic J, Cus F, Cupar A, Brajljh T and Drstvensek I 2014 Reverse Engineering of Parts with Optical Scanning and Additive Manufacturing *Procedia Eng.* **69**

- 795–803
- [3] Peng Q and Loftus M 2001 Using image processing based on neural networks in reverse engineering *Int. J. Mach. Tools Manuf.* **41** 625–40
 - [4] Chow C K and Yuen S Y 2009 Recovering shape by shading and stereo under lambertian shading model *Int. J. Comput. Vis.* **85** 58–100
 - [5] Chernov V, Alander J and Bochko V 2015 Integer-based accurate conversion between RGB and HSV color spaces *Comput. Electr. Eng.* **46** 328–37
 - [6] Verma M, Raman B and Murala S 2015 Local extrema co-occurrence pattern for color and texture image retrieval *Neurocomputing* **165** 255–69
 - [7] Piatkowski N, Lee S and Morik K 2016 Integer undirected graphical models for resource-constrained systems *Neurocomputing* **173** 9–23
 - [8] Abdulqawi N I A and Abu Mansor M S 2016 A Computer Method for Generating 3D Point Cloud from 2D Digital Image *J. Image Graph.* **4** 89–92
 - [9] Jack K 2008 *Digital Video and DSP* (Elsevier)