First Libyan international Conference on Engineering Sciences & Applications (FLICESA_LA)
13 – 15 March 2023, Tripoli – Libya

3D data compression with: high degree polynomials

Dr Abdusslam Osman
Department of Mathematics
Faculty of Sciences
Sebha University
Sebha Libya
abd.beitalmal@sebhau.edu.ly

Abstract— **The R2 coefficients suggest that there are concerns with accuracy even though the methodology has been established to be a workable way for 3D compression. Additionally, a cursory examination can indicate that such a compression method is suitable for some applications but insufficient for others, such as 3D face recognition. When determining whether the technique is appropriate, factors to take into account include the necessary polynomial degree, which depends on the data's features, as well as the fact that, as this example shows, at very high degrees, the data becomes unstable. As a result, iterative strategies will be taken into account in this study.**

Keywords—3D Data Structures, Boundary Representations, Compression Techniques, Polynomial Interpolation, Data Types, Object Modelling, Range Data, Surface Fitting.

I. INTRODUCTION

In this paper, a novel 3D data compression method based on the parameterization of surface patches is presented. This technique's key characteristic is that it establishes the number of cutting planes on the mesh while defining a set of sampling points at the points where the planes link or meet. These points feature an explicit structure that permits the parametric definition of both the x and y coordinates, and the z-values are interpolated using high degree polynomials. Once each plane has been recovered via the uncompressing approach, reconstruction is then accomplished by evaluating the polynomials from the preserved data, and triangulation is accomplished given the explicit structure and pairing of the planes and data points as detailed in Section

An interpolation through most of the control points using a polynomial function is the intended result. The following is the structure of this Section. The polynomial interpolation technique is introduced in Section II, and the surface patch reconstruction approach is described in detail in Section III. In Section IV, a commentary is provided as a conclusion.

II. POLYNOMIAL INTERPOLATION

A procedure known as interpolation, which is derived from the Latin word "interpolate" and means "to rebuild" or "to patch," can be used to recreate a curve, the surface, or other geometric objects from particular existing data [1] and [3] [2]. As surface patches characterized with polynomials of two variables, portions of curved graph surfaces can be represented. Using this as an illustration, a plane can:

$$z = a_0 + a_1 x + a_2 y \qquad (1)$$

higher-order polynomials can be used to simulate curved surface patches. In general, the first-degree polynomial in x is a straight line if we have two points (x_1, y_1) and (x_2, y_2) on a plane with $x_1 \neq x_2$. Afterwards, given n points in the plane, $(x_k, y_k), k = 1,2,3 \ldots n,$ there is a polynomial in x of degree less than n whose graph passes through or is very close to the points.

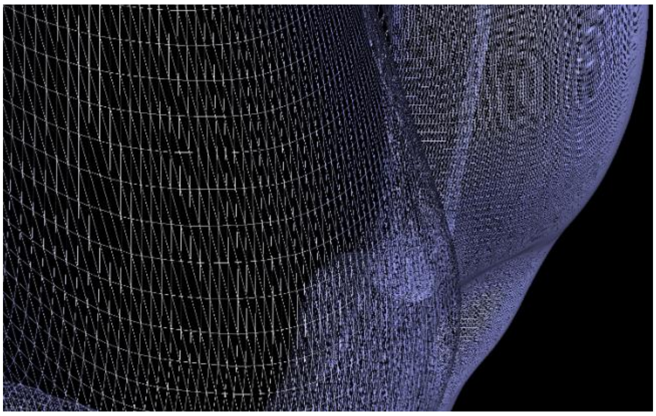


Figure 1:Polygonal mesh detail

The n-th degree polynomial in z has the following form:

$$P(z) = a_0 + a_1 z + a_2 z^2 + a_3 z^3 + \cdots \ldots a_n z^n \qquad (2)$$

where n is the degree of the polynomial, the coefficients are $a_0, a_1, a_2, \ldots . n,$

An effective method of compression for the data from Section 4.5 would be to represent the data in each plane by fitting the highest degree polynomial that best fits the data. In order to recreate the original data inside the given boundaries, only the polynomial coefficients and their borders are required to be retained.

The following steps must be taken in order to apply the polynomial approach to our data;

Valid vertices in the rectangular grid of 3D data are defined by the intersection of the horizontal and vertical planes with the mesh; any missing vertex is flagged as invalid. This intersection is determined from a 2D image by the GMPR scanner[11],[12],[13],[14],[15]. The valid vertices in each plane are used to assess the polynomial. The vector of coefficients of the polynomial can then be used to represent and recover the 3D data.

- First, use the built-in Matlab function $polyfit$ to do a polynomial fit for each plane with a given degree of n to obtain the $n + 1$ set of coefficients that best reflect the data.

$$P1 = polyfit(y, z, n); \qquad (3)$$

where y, z are vertex locations, n is the degree of the polynomial, and $P1$ is a vector with $n + 1$ coefficients.

- Second, for each curve, the indices of the k planes for the first and last valid vertices are recorded along with the coefficients of the polynomial. After that, we obtain a matrix for each model that represents the cutting planes. This matrix is constructed row by row, with each row representing a different cutting plane as follows:

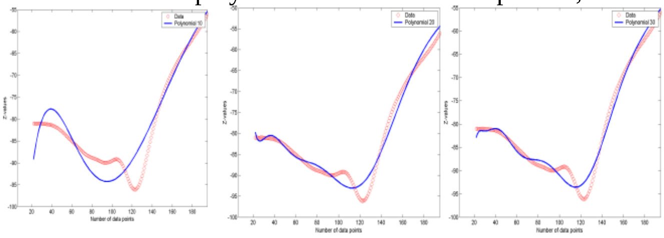
$$P2 = [coefficients - of\ plane - 1 \quad bFirst \quad bLast;$$
$$Coefficients - of\ plane - 2 \quad bFirst \quad bLast;$$
$$Coefficients - of - plane - 3 \quad bFirst \quad bLast;$$
$$\dots\dots\dots\dots\dots\dots..] \qquad (4)$$

- Third, the actual uncompressed mesh is created by rebuilding each set of z-data curve fitting data using the built-in Matlab function $polyval$, which turns each value into a polynomial and generates a corresponding value, as shown below:

$$P = polyval(C, Y) \qquad (5)$$

where C is a vector of coefficients from the current plane (derived from $P2$ above) and Y is a vector of equally spaced (derived from $D2$) $bFirst$ and $bLast$ indices (derived from $P2$ above). In order to replicate the observed data at the known locations, polynomial fitting is carried out in each plane using the z-values as "control points." As a result, data produced by the interpolation will be rather close to the grid of the source. However, it will be fitting a curve (model) to a known data set on the source grid and estimating the values based on the fitted curve in the destination grid. In this scenario, it won't replicate the actual information at the known place.

Additionally, the indices of the k planes for the first and last valid vertices, as well as the polynomial coefficients, are kept for each curve. This is because it is possible for multiple plane intersections to not intersect the mesh, in which case the indices (k_r, k_c) in question must be designated as invalid vertices. The polynomial cannot be extrapolated; it is only



valid between the given vertices.

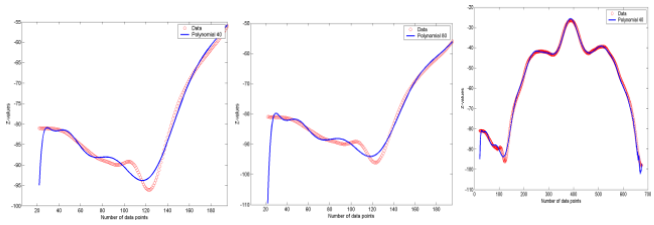


Figure 2: Polynomial interpolation for a cutting plane's initial few vertices; not to scale. Degrees 10, 20, and 30 are in the first row. Degrees 40, 80, and a whole cutting plane with degree 40 are in the second row. Interpolated data is blue, and original data is red.

Figure 2, shows the results of polynomial interpolation by degrees 10, 20, 30, 40, and 80 on the first few vertices specified by the cutting plane data highlighted in red. With significant inaccuracies at the extremities, it is evident that none of the outcomes are satisfactory. The next section will detail the analysis of the full meshes. Remember that the only information present is a sequence of 3D vertex positions, and the goal is to replace the vertex positions with a parametric definition based on high order polynomials. This results in a significant data reduction for high-density data, as is the case with 3D models. Assume a mesh with 100,000 vertices to demonstrate the compactness of this representation using sampled data as stated in [16]. This translates to 300,000 floating point one for each value $(x, y, and\ z)$. Since both (x, y) are defined as regular grids with spacing determined by the constant distance between cutting planes, it is possible to immediately replace these with 4 numbers only: the two constant spacing between cutting planes, as well as the number of rows and columns that make up the regular grid, thereby eliminating 200,000 floating point from representation.

The first and last valid vertex indices for each polynomial, along with a set of 100 polynomial coefficients, are all that are needed to fully reconstruct a mesh that has been split into 100 pieces. For each plane, assuming a polynomial of degree 25, just 28 integers are required: 26 coefficients + 2 vertex indices. This would be a reduction in the previous example from 100,000 to 2,800 floating-point numbers. The (x, y) values are assessed for each combination of (r, c) plane indices through Eqs. 4 and 5, and the polynomials used in Eq. 2 are evaluated for each plane inside their limits (first and final valid vertices). This allows for the reconstruction of the original mesh.

As a result, to completely recreate a mesh that has been sliced into 100 planes, only a list of 100 polynomial coefficients as well as the start and last valid vertex indices for each polynomial are needed. Only 28 numbers 26 coefficients + 2 vertex indices are required for each plane, assuming a polynomial of degree 25. This would result in a decrease of 100,000 to 2,800 floating-point numbers in the previous case. The polynomials used in Eq. 2 are evaluated for each plane within its borders (first and last valid vertices), and the (x, y) values are evaluated through Eqs.4 and 5 for each combination of (r, c) plane indices, in order to recreate the original mesh. For example, to apply the aforementioned technique to a polynomial of degree 3;

$$P(z) = a_0 + a_1 z_1 + a_2 z_2^2 + a_3 z_3^3 \qquad (6)$$

To obtain the four coefficients, a polynomial fit of degree 3 is first carried out for each plane using Eq. 3. Then, save the first and last valid points as well as the four coefficients of each curve. The same approach will also be used for the second plane by preserving the four coefficients with the first and final valid points, and it will be repeated for the remaining planes throughout the model. As a result, we are creating a matrix row by row using Eq. 4, where each row represents a cutting plane with 4 coefficients and 2 vertex indices.

Finally, reconstruction is accomplished by applying Eq. 5 to the polynomials derived from the preserved data. Figure 3 shows how, for a polynomial interpolation of degree 3, triangulation is done by matching the planes and data after each plane has been recovered by the uncompressing approach. This results in an unsatisfactory interpolation because the actual model looks quite shoddy. This is the case because degree 3 polynomial fitting is unable to pass through the majority (if any) of the control points and cannot, therefore, reconstruct the face model in a way that is reasonably similar to the original.

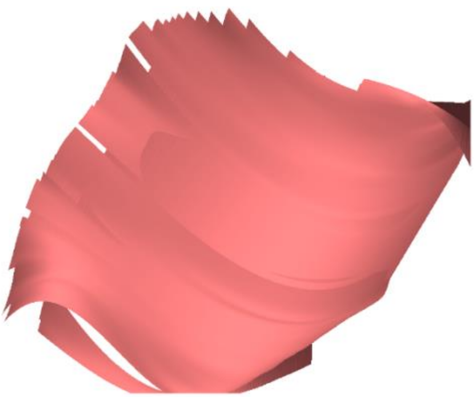


Figure 3:shows a model reconstruction using a degree 3 polynomial interpolation.

III. Results

In this part, we compare interpolation using a variety of high degree polynomials utilizing the mesh sampling technique mentioned in Section II.

A. Polynomial-based Data Compression

The coefficients and plane indices of the first and last valid points are preserved during polynomial interpolation by using the technique described in Section II. The intersections of all horizontal and vertical planes are shown in Figure 4, and Figure 5, each intersection is indicated by a red point. The model's attributes and the level of accuracy necessary will determine which of the two k_1 and k_2 structures to use for the mesh. Due to the properties of the GMPR scanner, the models employed here typically utilize 8 to 10 times more vertical planes than horizontal ones. In the end, the qualities of the data determine the number of planes; It was discovered that between 50 and 80 horizontal planes across the face were necessary for effective reconstruction. In order to offer a large number of data points for polynomial interpolation, the number of vertical planes was therefore determined at roughly 10 times the horizontal scale, resulting in a grid of 72 by 676 for the specific face model displayed (for different models these dimensions will vary). In Figure??, the vertical planes are less obvious because of their proximity to the horizontal planes, which are more obvious.

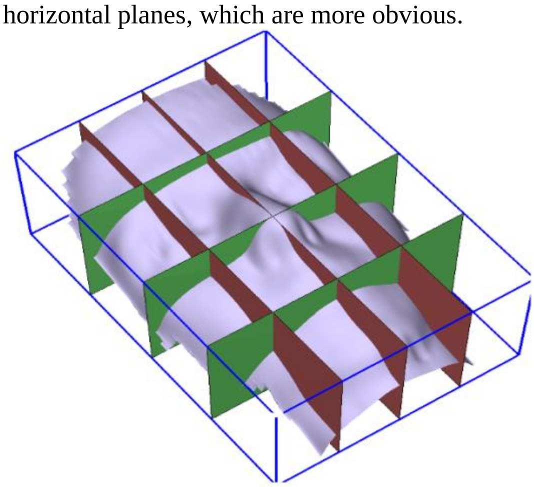


Figure 4: The bounding box and structured cutting planes.

Figure 5: On the mesh, an estimated point is used to indicate the points where each horizontal and vertical plane intersects with the other. 39,743 valid intersection or vertices can be found in the model displayed.

In order to discover a polynomial that nearly intersects the majority of the control points, a high degree polynomial interpolation is advised. The first and last valid points of each list of points, along with each pair of coefficients, the size $k1, k2$ of the sampled 3D data structure, and the separation between planes $D1$ and $D2$, are all preserved in order to later reconstruct the set of points. The file header lists this data in the following order:

$k1$ 72

$k2$ 676

$D1$ 3.3

$D2$ 0.3

$8.0960151e - 031 \ldots 6.7726253e + 002$ 382 482

$\ldots$

$1.8712059e - 032 \ldots 1.0464188e + 007$ 143 437

By evaluating the polynomials using the previously saved data, reconstruction is then accomplished. 72 lines of polynomial coefficients with their beginning and last valid points are listed after the 4 lines of header information in the file structure above. The data are used to determine the polynomial's degree. In the case when each line has, let's say, 23 numbers, the last two are the indices of the first and last valid points, leaving the prior 21 numbers as polynomial coefficients C. The degree of the polynomial is $C - 1$. With a polynomial of degree 20 as a result, the data in this instance was interpolated.

B. 3D Reconstruction

The following gives a high-level overview of the procedure. Apply the cutting planes re-meshing technique to an unstructured mesh to get data that is organized into a regular grid. The only variable that has to be interpolated is the depth value z because the values of (x, y) are known from the grid. Therefore, interpolation is possible for every set of points in the plane. The results of recreating a face model using polynomials of various degrees are displayed below. Results for polynomials of degrees 3, 10, 15, 20, 30, 40, and 80 are shown in Figures 6, 7, and 8.

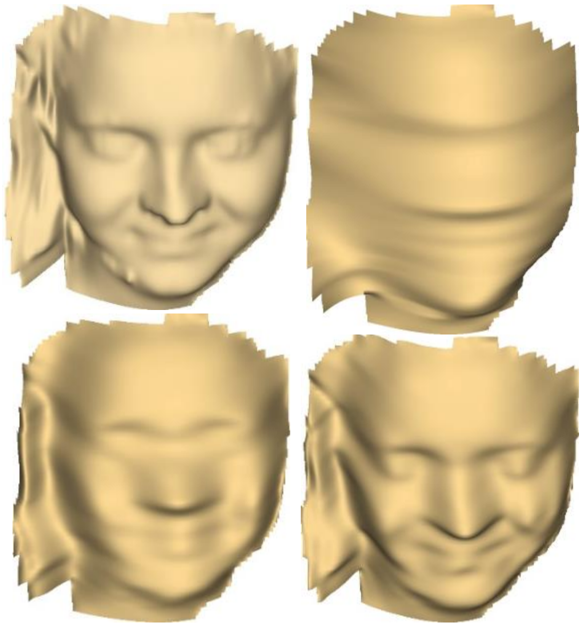


Figure 6: Polynomial interpolation degrees 3 to 15 are shown. The initial face model in the top row to the left has a file size of 4MB; in the top row to the right, a polynomial interpolation of degree 3 has reduced the file size to 8KB. Polynomial degree 10 reduces the file size to 16KB in the bottom row left, and degree 15 reduces it to 25KB in the bottom row right.

TABLE 1: PERCENTAGE COMPRESSION RATES.

Degree	20	30	40	50	80
Rate	99.35	99.07	98.79	98.53	97.66

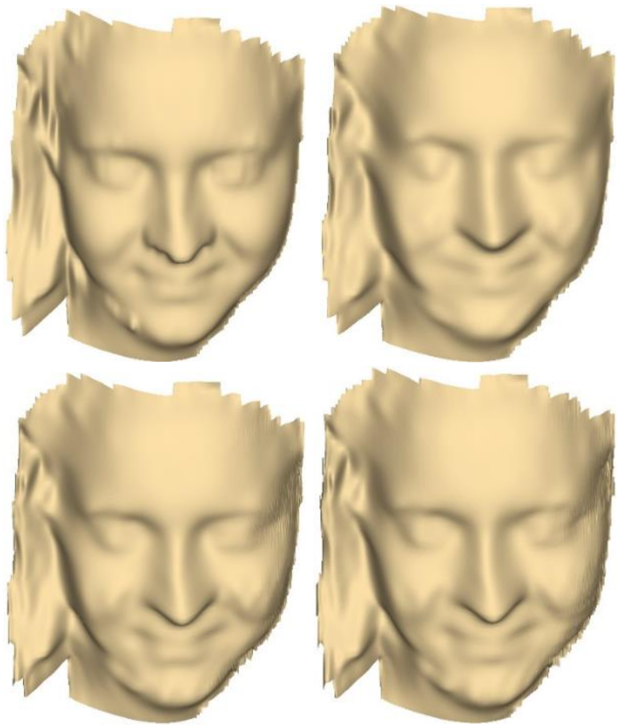


Figure 7: 20-to-40-degree polynomial interpolation. The original face model in the top row left has a file size of 4MB; in the top row right, a polynomial interpolation of degree 20 reduces the file size to 26KB. Polynomial degree 30 reduces the file size to 37.2KB on the bottom row left, while degree 40 reduces the file size to 48.5KB.

The majority of mesh comparison techniques were created to compare a mesh before and after a process, and we are interested in learning how the procedure has changed the mesh. With polynomial compression, a clear trend has been seen. Although the recovered data are worthless for lower polynomial degrees, such as degree 3, the compression rate is relatively great. The reconstruction gets better as the degree goes up, but there is a breaking point where the data becomes unstable at very high degrees. Figure 8, illustrates this by juxtaposing the original face model on the left with the one that was created through reconstruction using an 80-degree polynomial. It is shown that polynomial compression has an ideal point for the type of 3D utilized here, and this point appears to be around degree 30. The method is quite effective in terms of compression rates, as evidenced by the fact that the file size in OBJ format was decreased from 4MB to 26KB for a polynomial interpolation of degree 20. Similar reductions were made for other polynomials as well; a summary of these reductions is shown in Table 2, which also includes compression rates at the ideal point. This represents a decrease of 99.35%.

C. Measuring the fit

There are a variety of tests that may be used, including statistical summaries, to judge how well a polynomial regression fits the data or how well the recovered data points match the original data. Visually evaluating the quality by plotting the original and regression data sets is by far the most insightful method. By visually analyzing the models in Figure 7, it is indicated that a polynomial interpolation of degrees 20 to 40 would be a good fit for the majority of applications and provide a good description of the data.

Examining the residuals and plotting them against the anticipated values is an additional method of evaluating quality. A polynomial of degree 30 is used to interpolate data, as shown in Figure 9. There should be no patterns or trends seen on the plot for a successful match. It is a good indicator of the quality of the fit when the scatter plot appears to be filled with random noise. The residuals should show a straight line on a normal-probability plot if the fit is good. According to the plot shown in Figure 10, the majority of polynomials calculated at each plane do indeed define a straight line, demonstrating a satisfactory fit.

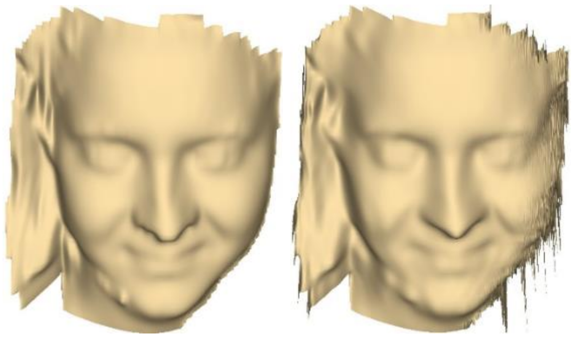


Figure 8:The original face model is on the left; the 80th degree polynomial interpolation is on the right. It is highlighted that the model destabilizes.

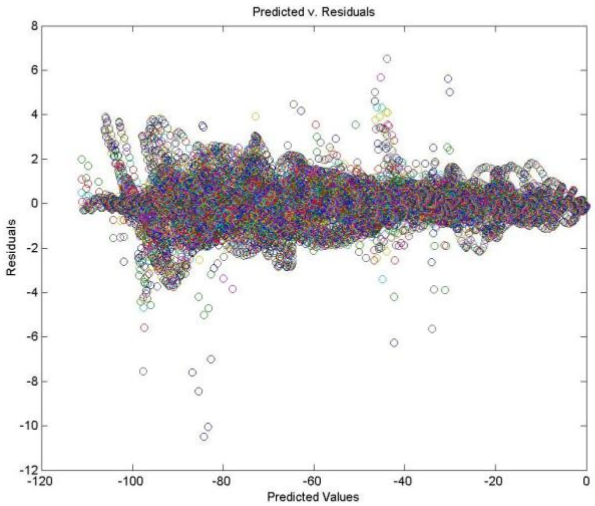


Figure 9: Predicted Values against Residuals are depicted in a scatter plot. It should not exhibit any patterns or tendencies for a good fit. The figure displays what appears to be random noise, which denotes a good match.

The coefficient of determination, often known as R^2, which represents the proportion of the variation in the data that is explained by the model, is one of many alternative statistical metrics that can be used to evaluate the accuracy or suitability of a model. This can be approximated by first figuring out the original data set's deviation, which provides a measurement of the spread. The variation that is not accounted for is represented by the sum of the residuals squared, while the total variation to be accounted for (SST) is supplied by the sum of deviation squared (SSE).

$$R^2 = 1 - \frac{SSE}{SST} \qquad (6)$$

Table 2 provides descriptions of certain interpolated models' R^2 values. This is the best interpolation point in the data set, according to the table, which demonstrates a trend of rising R^2 as polynomial degree rises and peaked at roughly degree 30. R^2 declines monotonically for increasing degrees, and this is further supported by a visual examination of the 3D reconstructed models, whose quality declines as polynomials of greater degrees are used in them and cause instability.

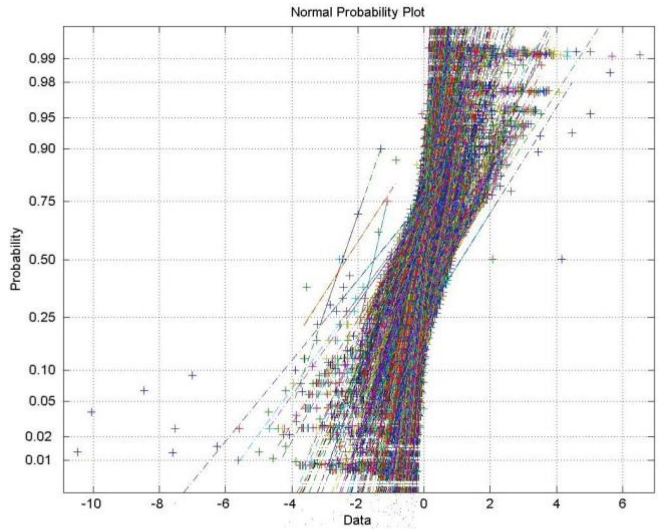


Figure 10: The residuals' normal-probability plot. Each polynomial curve should be described by a straight line in a good fit, which is confirmed by the plot.

.

TABLE 2: **R^2 coefficients of determination for polynomial fits to the given data with degrees ranging from 20 to 80.**

Degree	20	30	40	50	80
R^2	0.9995	0.9996	0.9995	0.9994	0.9909

IV. CONCLUSION

A novel approach for 3D data compression based on various levels of polynomial interpolation has been introduced and tested in this paper. The parameterization of surface patches, which was explored and tested, is the foundation of the novel compression technique. While it is simple to compute the (x, y) values of each vertex on a regular grid, the real z-values are interpolated using a high degree polynomial, and the results reveal compression rates of over 99%. The R^2 coefficients suggest that there are concerns with accuracy even though the methodology has been established to be a workable way for 3D compression. Additionally, visual examination may indicate that such a compression method may be suitable for some applications but unsatisfactory for others, such as 3D face recognition.

The required polynomial degree, which depends on the data's features, and the fact that the data becomes unstable at very high degrees, as seen above, are factors to take into account when determining if the technique is appropriate. As a result, iterative strategies will be taken into account in this study. A method for Fourier-based data compression and PDE-based data uncompressing will be introduced in the next paper.

REFERENCES

[1] Alliez, P. and C. Gotsman (2003). Recent Advances in Compression of 3D Meshes. *Inria Sophia Antipolis Research*

Report 4966,Oct 2003, 26pp.

[2] Bergh, J{\"o}ran and L{\"o}fstr{\"o}m, J{\"o}rgen (2012) . Interpolation spaces: an introduction. Springer Science \& Business Media, Volume 223.

[3] Davis, Philip J (1975). Interpolation and approximation. Courier Corporation.

[4] Shepard, Donald (1968). A two-dimensional interpolation function for irregularly-spaced data. Proceedings of the 1968 23rd ACM national conference. Pages 517—524.

[5] Brink, Willie and Robinson, Alan and Rodrigues, Marcos A. (2008). Indexing Uncoded Stripe Patterns in Structured Light Systemsby Maximum Spanning Trees, *British Machine Vision Conference BMVC 2008*, *Citeseer* volume 2018, 1–10 Sep 2008.

[6] COLLADA (2012). Digital Asset and FX Exchange Schema, *https://collada.org*, accessed Oct 2012.

[7] Deering, Michael (1995). Geometry Compression, *SIGGRAPH 95 Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques.* 13-20.

[8] Hill Jr, Francis S (2008). *Computer Graphics Using OpenGL*, 2nd edition, *Pearson Education*. Prentice-Hall Inc, 922pp.

[9] Qian, Shen-En and Hollinger, Allan B and Williams, Dan and Manak, Davinder (1998). 3D data compression of hyperspectral imagery using vector quantization with NDVI-based multiple codebooks, *IEEE International Geoscience and Remote Sensing Symposium IGARSS'98*, volume 5, 2680–2684.

[10] Lahanas, Michael and Kemmerer, Thorsten and Milickovic, Natasa and Karouzakis, Kostas and Baltas, Dimos and Zamboglou, Nikolaos (2000). Optimized bounding boxes for three-dimensional treatment planning in brachytherapy, *Medical Physics* 27(10):2333–2342 *Wiley Online Library* 2000.

[11] Matlab Revision R2012b Documentation, *MathWorks Online Documentation*, www.mathworks.co.uk/help/ accessed Oct 2012.

[12] Peng, Jingliang and Kim, Chang-Su and Kuo, C-C Jay (2005). Technologies for 3D mesh compression: A survey. *Journal of visual communication and image representation. Elsevier (2005)* 16/6 688–733.

[13] Robinson, Alan and Alboul, Lyuba and Rodrigues, Marcos (2004). Methods for Indexing Stripes in Uncoded Structured Light Scanning Systems, *Journal of WSCG*, 12(3), 2004, pp 371–378.

[14] Rodrigues, Marcos and Robinson, Alan (2011a). Real-time 3D Face Recognition using Line Projection and Mesh Sampling.In: *EG 3DOR 2011 - Euro graphics 2011 Workshop on 3D Object Retrieval*, Llandudno, UK, 10th April 2011. Euro graphics Association. p9–16.

[15] Rodrigues, Marcos A and Robinson, Alan (2011b). Fast 3D recognition for forensics and counter-terrorism applications. In: AKHGAR, Babak and YATES, Simeon, (eds.) *Intelligence management: knowledge driven frameworks for combating terrorism and organized crime. Advanced information and knowledge processing*, London, Springer-Verlag, 95–109.

[16] Rodrigues, Marcos A and Robinson, Alan and Osman, Abdulsslam (2010a). Efficient 3D Data Compression Through Parameterization of Free-Form Surface Patches, *2*010 International Conference on Signal Processing and Multimedia Applications (SIGMAP), 26-28 July 2010, p130–135.*IEEE*

[17] Rodrigues, Marcos and Robinson, Alan (2010b). Novel methods for real-time 3D facial recognition. In: SARRAFZADEH, Majid and PETRATOS, Panagiotis, (eds.) *Strategic Advantage of Computing Information Systems in Enterprise Management*, Athens, Greece, ATINER, 169–180.

[18] Rodrigues, M.A., A. Robinson, and W. Brink (2008). Fast 3D Reconstruction and Recognition, in *New Aspects of Signal Processing, Computational Geometry and Artificial Vision*, 8th WSEAS ISCGAV, Rhodes, 2008, 15–21

[19] Osman, Abdusslam (2014). 3D modelling using partial differential equations (PDEs), *Sheffield Hallam University (UnitedKingdom)*.

[20] Rodrigues, Marcos and Osman, Abdusslam and Robinson, Alan (2013). Partial differential equations for 3D data compressionand reconstruction. *ADSA Advances in Dynamical Systems and Applications Research India Publications*, Vol 8(2),303–315.

[21] Shikhare, D., S.V. Babji, and S.P. Mudur (2002). Compression techniques for distributed use of 3D data: an emerging mediatype on the internet, *15th International Conference on Computer Communication*, India, pp 676–696.