A New Error Metric for Geometric Shape Distortion using Depth Values from Orthographic Projections

Maja Krivokuća
The University of Auckland
New Zealand
mkri012@aucklanduni.ac.nz

Burkhard C. Wünsche
The University of Auckland
New Zealand
burkhard@cs.auckland.ac.nz

Waleed Abdulla
The University of Auckland
New Zealand
w.abdulla@auckland.ac.nz

ABSTRACT

Euclidean-based distance metrics are commonly used for measuring geometric shape distortions of 3D models, but have several drawbacks. They have strict requirements regarding model representation and usually necessitate expensive surface sampling or point correspondence matching. Furthermore, many distortion metrics have been designed to capture relatively minor shape changes, rather than large-scale (global) shape distortions which can occur during compression and image-based reconstruction. This paper presents a new geometric distortion metric for offline quality assessment of 3D models. The new metric is largely independent of object representation and does not require any surface sampling or point matching operations. It can capture well both minor and severe shape distortions, on a large and small scale. In the context of measuring the rate-distortion performance of a lossy mesh compression algorithm, the new metric provides a more reliable measure of overall shape distortion than the commonly used Hausdorff distance and a more relevant measure of surface error than the RMSE. Visual distortion maps for the new metric are also created, which indicate that the metric also captures well the *perceived* shape error between two objects.

Keywords

Shape distortion; error metric; mesh compression; depth buffer; z-buffer; orthographic projection.

1. INTRODUCTION

The accurate measurement of geometric shape distortions of 3D models during compression, watermarking, reconstruction and simplification, is still an open problem. The main reason for this is that the term *distortion* lacks a formal definition and, as such, the existing error metrics used for this purpose often produce different, and sometimes conflicting, results. This issue is especially critical in the field of lossy mesh compression, where the basis of evaluation and comparison of different algorithms is the *rate-distortion* curve. Indeed, in a recent investigation [7], it was found that due to the disparities in the performance results produced by different error metrics, a complete evaluation of a lossy mesh compression algorithm can only be achieved by using several different distortion metrics. Furthermore, most existing error metrics currently used for this purpose are purely based on Euclidean distance measures between the two models being compared. Such metrics often have strict constraints on the representations of the models being compared (e.g., the two models must contain the same number of vertices), or else require expensive surface sampling and point correspondence matching techniques. Moreover, it is now generally acknowledged that these purely geometric error metrics do not correlate well with the human visual perception of error. For this reason, the recent trend has been to design perceptual error metrics, which aim to reflect an 'average' observer's perception of distortion on a 3D model [8, 9]. Most of these perceptual metrics have been designed to capture relatively small surface distortions and have not been tested for severe large-scale distortions that alter the global shape of an object.

In this paper, we propose an alternative metric, termed the *depth difference* (DD), for measuring geometric shape distortion on 3D models. This metric is able to capture distortions of different scale and extent, and it is independent of the object representation as long as the object can be rendered in 3D using depth values.

Section 2 of this paper presents a review on existing distortion metrics for lossy mesh compression; Section 3 details the concepts and design procedure behind our proposed error metric; Section 4 covers implementation details; Section 5 describes our experimental procedures and presents the results; Section 6 summarizes the key benefits of the DD metric and addresses some potential limitations; and Section 7 concludes this paper and suggests directions for future research.

2. EXISTING DISTORTION METRICS FOR LOSSY MESH COMPRESSION

Distortion metrics used in lossy mesh compression can be divided into two main categories: *online* error metrics and *offline* error metrics. Online metrics (e.g., [4, 10, 13]) are generally designed to drive a specific compression (or simplification) algorithm, by checking that the quality of the reconstructed (simplified) mesh at certain points during processing satisfies some predetermined quality criteria. Offline error metrics are used to evaluate the quality of a reconstructed 3D model at the end of transmission. Hence, offline error metrics are useful for evaluating and comparing the performance of different compression algorithms. The most commonly applied error metrics for evaluating lossy mesh compression algorithms are the root mean square error (RMSE) (e.g., [2]), the peak signal to noise ratio (PSNR) (e.g., [6]), and the Hausdorff distance (d_H) (e.g., [1]). These metrics require the two models being compared to have the same number of vertices (RMSE, PSNR), or else require surface sampling (d_H) or searching operations to find the closest point correspondences between the two models. Furthermore, the results produced can vary depending on the selected sampling parameters and point correspondences. The requirement for the same number of vertices is impractical in lossy mesh compression, which often

Conference'10, Month 1–2, 2010, City, State, Country.

involves mesh simplification procedures where the simplified model has a different geometry and connectivity to the original model. Moreover, it is now generally acknowledged that these purely geometric error metrics do not correlate well with a human's perceived error, and hence the visual interpretation of results is often ambiguous. For this reason, a recent trend has been to design perceptual error metrics (a survey can be found in [9] and [8]), which aim to reflect an 'average' observer's perception of error on a 3D model. However, most of these perceptual metrics have been designed to capture relatively small surface distortions, and have not been tested for severe large-scale distortions that can alter the overall shape of an object. Indeed, in [7] it was found that even the currently best performing perceptual metric, the *mesh structural distortion measure 2* (MSDM2) [8], tends to under-estimate the perceptual effects of large-scale shape distortions, especially on very smooth, low-detail models. Additionally, most of the existing perceptual metrics still require some form of point correspondence searching on object surfaces. Several image-based perceptual metrics have also been proposed, which do not have this requirement (see [9]); however, generally the perceived degradation of still images is not adequate to evaluate the perceived degradation of the equivalent 3D model [15].

Based on these observations, we propose a new metric that is largely independent of object representation, is computationally inexpensive, is able to capture both minor and severe shape distortions on different scales, and has some correlation with the human perception of shape distortion.

3. THE DEPTH DIFFERENCE (DD) METRIC

The metric we are proposing aims to capture the visible volume difference between two 3D models. We achieve this by comparing 2D *depth images* (DIs) obtained using orthographic projections of the 3D models from many different viewpoints. The method assumes that a rendering technique is available (e.g., OpenGL polygon rendering), which determines for every pixel the 'pseudo-depth' [5] of the corresponding 3D surface point(s). A *difference depth image* (DDI) is then computed for a pair of DIs at each viewpoint, by performing a matrix subtraction between them. The individual difference depth values within each DDI are summed up and the result is divided by the total size of the DDI, to obtain the *average depth difference* (ADD) between the two input models, at one viewpoint. Finally, the average depth difference over many viewpoints is computed as the sum of the ADDs at all the viewpoints, divided by the total number of viewpoints. We call this final measure simply the *depth difference* (DD).

We obtain a dense coverage of the 3D object by generating viewpoints using a *spiral point distribution* [14]. For a spiral point set of size *N*, distributed evenly over the surface of a sphere, the spherical coordinates (θ_k, ϕ_k) of the *k*-th viewpoint are:

$$\theta_k := \cos^{-1}(h_k)\,;\; h_k := -1 + \tfrac{2(k-1)}{(N-1)}, \qquad 1 \le k \le N;$$

$$\phi_k := \left(\phi_{k-1} + \frac{C}{\sqrt{N}} \frac{1}{\sqrt{1-h_k^2}}\right)(mod 2\pi), \qquad 2 \le k \le N-1;$$

$$\phi_1 := \phi_N := 0$$

The constant *C* in the equation above is chosen so that the successive points are approximately the same Euclidean distance apart on the surface of this sphere. We use a value of 3.6 for *C*, which has been shown to work well for $N \le 12{,}000$ [14]. For *N*, we choose a value of 100 viewpoints. This value yielded in our experiments a good coverage of a large range of different objects.

To compute the depth differences between two DIs for different models at the same viewpoint *k*, we require depth buffer values in the range [0,1] (default for most graphics APIs), and we compute for each pixel the absolute difference in depth values. We term the resulting 2D array of difference values the *difference depth image* (DDI). Figure 1 gives an example using the Stanford bunny and a highly compressed version of it.

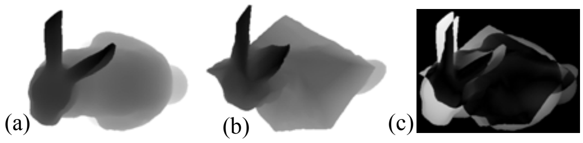

Figure 1: Visualization of depth images: (a) DI of the 3D bunny model from one viewpoint; (b) DI of a distorted version of (a) at the same viewpoint; (c) DDI of (a) and (b).

Note that black and white in the images above indicate, respectively, the minimum and maximum depth values in the chosen view volume. By visually inspecting the images in Figure 1(a) and Figure 1(b), we see that the DDI has correctly captured the shape differences (around the ears, nose and body) between these two models at this viewpoint.

Following the computation of the DDI at each viewpoint, the *average depth difference* (ADD) between the two input models at viewpoint k (where $(1 \le k \le N)$), is obtained in the following manner:

$$ADD_k = \frac{1}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m}|DDI_k(m,n)|$$

where $m \times n$ is the size of the DDI. The averaging ensures that the depth difference values are in the range [0,1] and are independent of the size of the depth buffer used for rendering.

Once the ADD for each of the *N* chosen viewpoints has been computed, all of the *N* ADDs are summed up and the result is divided by *N*. We call the result the *depth difference* (DD), and it represents the average depth difference over *N* viewpoints, for the two input 3D models:

$$DD = \frac{1}{N}\sum_{i=1}^{N}(ADD_k)_i$$

Similarly to the case with the ADD values, the averaging step is performed to ensure that the depth difference values are in the range [0,1], and to remove the variability factor of different choices for values of *N*. A DD value of 0 indicates that the two models being compared are identical in shape (with respect to the depth buffer resolution), while values closer to 1 indicate greater differences in shape between these models. The computation of the DD relies on the assumption that the two models being compared are aligned according to a common reference point (e.g., their centres of mass), so that the only difference in size between them is caused by the distortions.

The following section discusses additional implementation considerations, which must be taken into account to ensure that the DD metric is computed accurately.

4. IMPLEMENTATION DETAILS

We use MATLAB for displaying the 3D models and computing the DD values, and OpenGL for rendering and obtaining the depth images. Since we only require depth images, results are not affected by shading, lighting, or texturing.

We use *orthographic* projection - rather than *perspective* projection – since in an orthographic projection the viewing volume is a rectangular parallelepiped. This means that the relative size of objects is maintained regardless of the distance to the virtual camera. As a result, two corresponding points (pixels in our case) can be measured. Furthermore, with orthographic projection all the depth values in the z-buffer have the same precision, regardless of how far away the object is from the camera. In perspective projection, greater depth precision is reserved for objects closer to the camera.

In order to obtain comparable depth values, the same viewing volume must be used for the objects to be compared. We choose the viewing volume to be the smallest cube enclosing the axis aligned bounding spheres of both models.

All our experiments use a viewport size of 420x560 pixels. The viewport dimensions determine the size of the depth images (DIs) which we extract from the depth buffer, and hence limit the spatial resolution of any features we can capture.

5. EVALUATION

The suitability of the DD as a shape distortion metric was tested by evaluating its performance in several different distortion scenarios using a number of different 3D models. The test scenarios, choice of test models, and experimental results for each scenario are presented in the following sections.

5.1 Experiment 1: Isolated Distortions

The purpose of this experiment was to test how the DD metric captures the two basic types of shape distortion that can occur on a 3D model's surface: convex distortions and concave distortions. Convex distortions protrude outwards from the surface and are generally manifested as bulges or spikes, while concave distortions bend the surface of a model inwards. The distortions in this experiment were created manually, by changing the positions of certain vertices on the models. The test models were chosen to be simple geometric objects, with little or no surface detail, so that the isolated distortions would be easily discernible on their surfaces. The results from two such test cases are presented below.

5.1.1 Convex Distortions

Figure 2 illustrates a model of an icosahedron, followed by three distorted versions of this model. The only distortion present in the model in Figure 2(b) is the spike protruding from the icosahedron's surface. The distorted model in Figure 2(c) contains this same spike plus an additional, smaller spike, and the model in Figure 2(d) contains both of the spikes present in the previous two distorted models plus an additional third spike. In all the distorted models, the spike from Figure 2(b) is the largest distortion present.

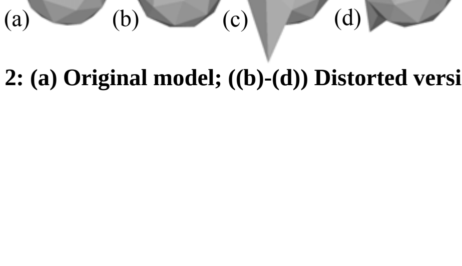

Figure 2: (a) Original model; ((b)-(d)) Distorted versions of (a).

Figure 3 illustrates the DIs, DDIs and DD values for each of the distorted icosahedrons in Figure 2, as well as the Hausdorff distance values for comparison. The DIs and DDIs have been chosen at one salient viewpoint and generated in the same way as for Figure 1.

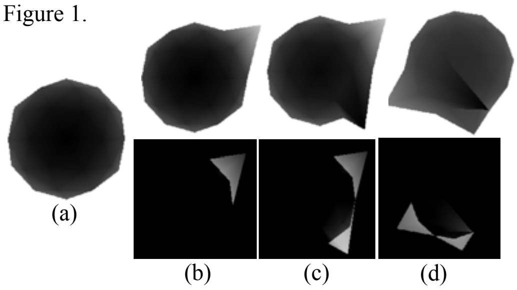

Figure 3: (a) DI of the original model; ((b)-(d)) DIs and associated DDIs (under each depth image) of the distorted versions of (a): (b) DD = 0.00363, d_H = 1.035; (c) DD = 0.00590, d_H = 1.035; (d) DD = 0.00684, d_H = 1.035.

The DDIs in Figure 3 demonstrate that the DD metric is able to correctly capture all the convex distortions present on the icosahedrons' surfaces. Moreover, the increasing DD values corresponding to the increasing number of spikes on the icosahedrons show that the DD changes appropriately to reflect the changing number of errors on a model's surface. This gives a better indication of the overall amount of distortion present on a 3D object than the commonly used Hausdorff distance, which only reflects the maximum local error on a surface and fails to capture other, smaller errors that may be simultaneously present on this surface. This limitation of the Hausdorff distance is portrayed by the d_H values in Figure 3, which remain the same for all the distorted icosahedrons despite the fact that some of them have a larger number of distortions than others.

5.1.2 Concave Distortions

Figure 4 illustrates an example of a concave distortion, which causes one face of the cube model to bend inwards at a certain point.

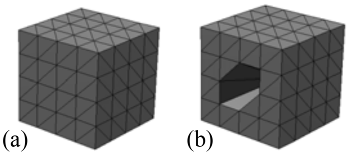

Figure 4: (a) Undistorted cube; (b) Same cube as in (a), but with a concavity in one face.

Figure 5 shows the depth image (DI) for each of these cubes, as well as the corresponding DDI, at the viewpoint where a viewer is looking directly at the face with the concavity so that the other cube faces are temporarily hidden from view. Hence the DI corresponding to the undistorted cube is just a black square, because this face is flat and nearest to the viewer. The DI for the distorted cube, however, indicates the presence of higher depth values around the centre of the cube face, which is where the concavity is located. The DDI in Figure 5 demonstrates that the DD metric correctly captures this difference in shape between the two cubes.

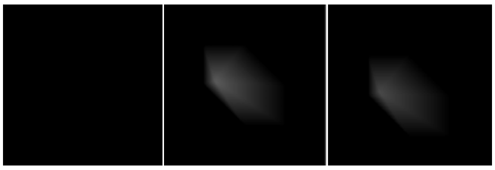

Figure 5: (left) DI for the undistorted cube; (middle) DI for the distorted cube; (right) DDI of these depth images.

5.2 Experiment 2: Combination of Distortions from a Real-Use Case Scenario

The purpose of this experiment was to test how the DD behaves in a real-use case scenario, where a combination of convex and concave distortions may be present in various sizes on a 3D model. In particular, three aspects of the DD were being tested here: whether the DD values can consistently follow a generally decreasing trend when distortion levels are systematically decreased; how this DD trend compares to the trends produced by other commonly used geometric distortion metrics in the same scenario; and how the DD captures severe and minor cases of these 'real' shape distortions. The scenario that was selected for this purpose was the measurement of the *rate-distortion* (R-D) performance of the wavelet-based mesh compression algorithm implemented in [7]. Similarly to [7], here the *rate* is represented as the percentage of wavelet coefficients used at each resolution level for mesh reconstruction. The wavelet algorithm was selected for two reasons: firstly, the investigation of this method in [7] inspired the idea for the DD metric; and secondly, this algorithm can easily be used to systematically create different levels of shape distortion on a 3D model, by discarding different numbers of wavelet coefficients in the mesh reconstruction. The two metrics that have been selected for comparison with the DD are the commonly used RMSE and the d_H. The d_H values in all our experiments were obtained from the Metro tool [3] with the default settings, while the RMSE values were obtained by an own implementation of the RMSE formula used in [7]. In order to facilitate comparisons between the R-D curves produced by each of these metrics, the RMSE, d_H, and DD values have been normalized (scaled) to fit into the [0,1] range, in the same way as was done for the d_H and RMSE values in [7]. The test models used for this experiment were the three key models used in [7]: the Torus, Star, and Bunny (see Figures 7, 8, and 9). These three models represent a range of interesting shapes to test the DD metric on: the Torus has a globally smooth surface, with a hole; the Star has sharp points and a concave surface between each pair of points; and the Bunny contains a variety of different-sized convexities and concavities on its surface. The results of the experiment are presented in the following sections.

5.2.1 Measuring Decreasing Distortion Levels

Figure 6 demonstrates that the DD curves follow a decreasing trend as the compression rate increases. Because a higher compression rate (more wavelet coefficients) results in less distortion on the reconstructed model, this indicates that the DD is able to reflect these decreasing distortion levels. A visual example of this is presented for the Star model in Figure 7.

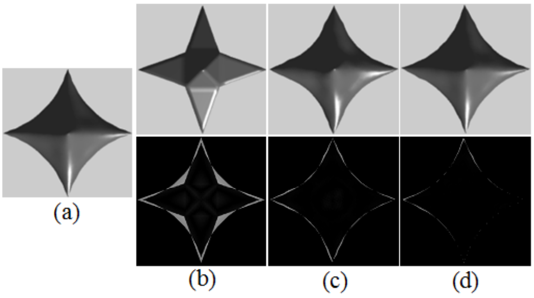

Figure 7: (a) Original Star model; ((b)-(d)) Reconstructed versions of (a) using, respectively, 0%, 50%, and 80% wavelet coefficients (associated DDIs are under each model): (b) DD = 0.883, (c) DD = 0.269, (d) DD = 0.221.

The DDIs in Figure 7 show decreasing amounts of mismatch in shape (grey areas) between the original and reconstructed models for higher compression rates. This is confirmed by the decreasing (scaled) DD values from the model in Figure 7(b) to the model in Figure 7(d).

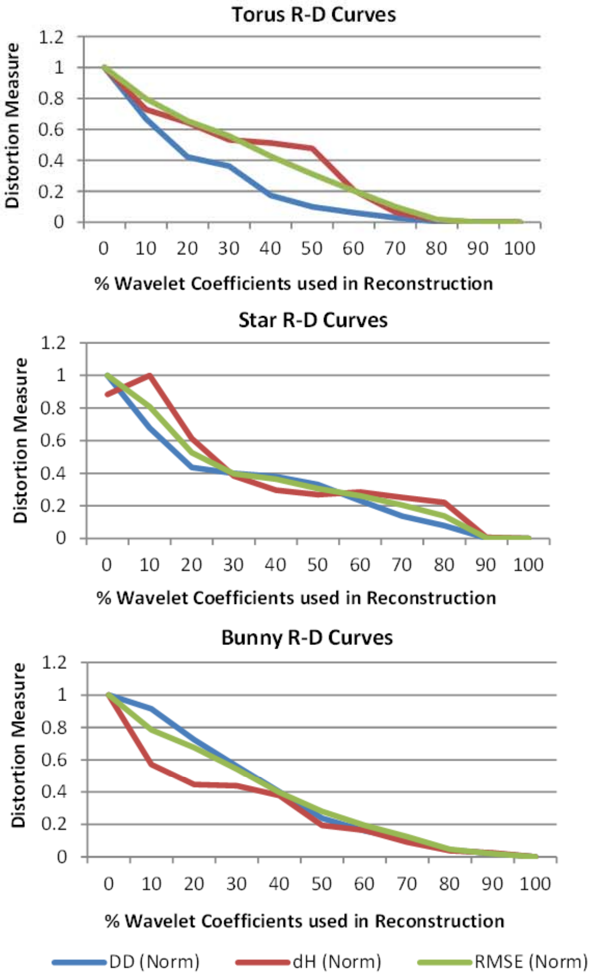

Figure 6: Rate-distortion curves.

5.2.2 Comparison to RMSE and d_H

The DD trends in Figure 6 seem to have a closer correlation with the RMSE curves than with the d_H curves, especially for the Star and Bunny models. The reason for this similarity with the RMSE is that both the RMSE and DD rely on average distance measures between points on the model surfaces. However, the RMSE curves in Figure 6 generally follow a much more linear trend than the DD curves. This is because the RMSE only takes into account vertex-to-vertex distances, whereas the DD measures differences between pixels, which may include more (or less) than one vertex. This makes the DD a more relevant measure of surface error than the RMSE, especially on low-detail models where the vertices are few and far apart. This also means that, for the DD, the two models being compared do not need to contain the same number of vertices, which is not the case for the RMSE. In fact, with the DD, the two models do not even need to have the same representation. The advantage of the DD over the d_H is illustrated in Figure 3 and confirmed in the Torus plot in Figure 6. In the Torus R-D plot, between a rate of 30% and 40% wavelet coefficients the d_H curve is almost flat, while the DD curve decreases quite rapidly. Figure 8 depicts the DDIs for the Torus models between these compression rates, which indicate an improvement in the shape of the Torus (particularly on the outer boundary) that the d_H fails to capture. Furthermore, unlike the d_H, the DD is a symmetric metric and does not require any surface sampling.

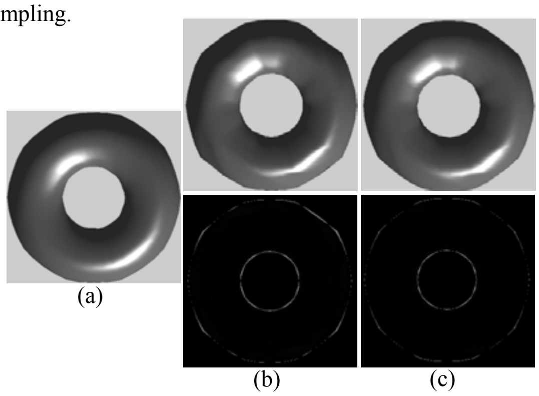

Figure 8: (a) Original Torus model; ((b)-(c)) Reconstructed versions of (a) using, respectively, 30% and 40% wavelet coefficients (corresponding DDIs are under each model).

5.2.3 Capturing Severe Large-Scale Shape Distortions

Figures 7 and 8 illustrate the ability of the DD metric to capture relatively minor large-scale shape distortions. Figures 3 and 5 indicate the DD's ability to capture localized distortions on a smaller scale. However, the DD is also able to capture severe large-scale shape distortions, and an example of this is illustrated in Figure 9. In Figure 9, the distorted version of the original Bunny model corresponds to a reconstruction of this model with 10% wavelet coefficients at each resolution level. The associated DDIs at several important viewpoints illustrate that the DD metric captures well the shape dissimilarities between the original and distorted Bunny models. Indeed, the DDIs illustrate that the DD essentially gives a measure of the average volume difference between two models, which is an intuitive way to think about shape differences.

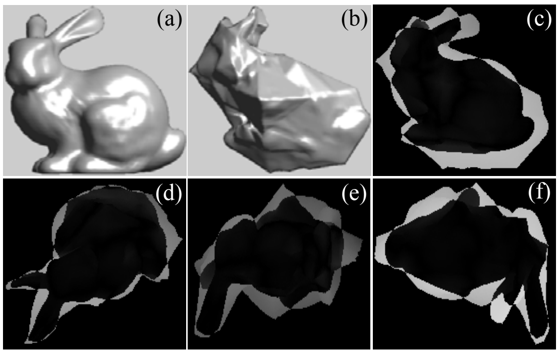

Figure 9: (a) Original Bunny model; (b) Distorted version of (a); ((c)-(f)) DDIs between (a) and (b) at several salient viewpoints.

5.3 Comparison with the "Surface Roving" Metric

An alternative way of using depth values to measure shape dissimilarity was proposed in [12]. The key differences between that metric and our metric are: (1) The metric in [12] is designed for measuring minor shape distortions caused by geometry compression in a guided multi-resolution modelling scenario (indeed, the results in [12] are demonstrated only for minor shape distortions); our metric is designed to capture a wider range of distortion levels in a more general setting. (2) The method by which the multiple viewpoints (and hence the 2D projections) are generated in [12] is different to our approach, as we do not use a surface roving camera. (3) The results of the metric in [12] are presented for different representations of only one model (the Bunny), whereas we present experimental results of our metric for several models with different shapes and surface details. (4) We also present a comparison of our metric with two other commonly used geometric error metrics in terms of how they capture different levels of distortion on different models, which is not present in [12]. (5) Additionally, we generate visual distortion maps associated with our metric, which is not the case in [12].

6. RESULTS AND DISCUSSION

The following section summarizes key benefits that the DD can offer as a shape distortion metric. Potential limitations of the metric are also identified and addressed.

6.1 Benefits of the DD Metric

The DD is an image-based metric, so it is largely independent of object representation - if the object can be rendered in 3D with depth values, it can be measured by the DD. Because the DD relies on directly comparing the depth values of two overlapping pixels, it does not require any surface sampling or point correspondence searching on the 3D models being compared. This is an advantage over most existing shape distortion metrics (such as the RMSE and d_H), which do have this requirement and can therefore be quite computationally expensive to compute for large and complex models. In fact, our method does not require a surface mesh and can also be used in combination with a point cloud renderer or ray tracer for CSG objects. Furthermore, the DD is better able to capture the overall difference in shape between two models than the commonly used Hausdorff distance. Unlike the d_H, the DD is also a symmetric metric, so that only a one-way calculation is needed between the two models being compared. Additionally, this new metric is able to capture distortions of various sizes as long as the available depth resolution allows these distortions to be detected (see Section 6.2). The visual distortion maps (DDIs) of the DD at different viewpoints can also give an indication of the dependency of viewpoint on the perceived shape error. Although the DD is technically a geometric error metric, inspection of the associated DDIs in our experiments suggests that the DD is also able to capture well the *perceived* shape differences between different objects. This indicates that the DD has promise as a perceptual metric, but in order to confirm this a more thorough investigation is required, where the DD is compared to the subjective distortion scores from a large group of human subjects judging a variety of different models. This will be the subject of a future investigation. Another key benefit of the DD is its generality as an offline metric: although designed with the case of lossy mesh compression in mind, the DD can actually be used in any application where the shape preservation of a 3D model is important but there is no strict requirement for exact geometry reconstruction.

6.2 Potential Limitations of the DD Metric

There are three aspects of the DD metric that may be seen as limitations. Firstly, the computation of the DD assumes a pixel graphics display device. However, this is the most common type of display device used nowadays, so this requirement of the DD is not a serious limitation. Secondly, the 3D models to be compared must be rendered at many different viewpoints. However, modern entry-level graphics cards are able to achieve rendering speeds from 120-210 million triangles per second, with an impressive 1.3-1.8 billion triangles per second for the ultra-high-end range [11]. These performance capabilities, combined with the fact that the DD metric requires only basic rendering (see Section 4), means that the rendering at many viewpoints can be achieved almost in real-time for most 3D models. Because the DD is designed as an offline quality metric, the rendering speed may not be an issue in the first place; however, for very large models, a fast computation of the DD may be achieved by providing input models with efficient representations, which are able to be rendered quickly. Finally, the size of distortion that can be detected by the DD depends on the size (depth) of the available z-buffer. However, most modern graphics cards use a 24-bit z-buffer, which allows 2^{24} different depth values to be represented. By further setting the viewing volume of the 3D object in such a way as to take the greatest advantage of the available depth resolution (discussed in Section 4), this can allow the DD to detect a wide range of distortion sizes.

7. CONCLUSION

We have presented a new image-based error metric for measuring geometric shape distortion on 3D models. This new metric, termed the *depth difference* (DD), works by comparing the z-buffer depth values of orthographic projections of a pair of 3D models rendered at many viewpoints, and computing the average depth difference between them. The new metric has been found to work well for capturing minor as well as severe shape distortions, both on a large and small scale. The metric has also been tested in the real-use case of measuring the rate-distortion performance of a wavelet-based mesh compression algorithm. In this scenario, it has been found to provide a more reliable measure of overall shape distortion than the commonly used Hausdorff distance and a more relevant measure of surface error than the RMSE. The main advantages that the DD metric enjoys over the existing geometric error metrics, which are commonly applied to measuring shape distortion, are that it is largely independent of 3D object representation and it requires no surface sampling or point correspondence matching. The DD is intended to be used for offline quality evaluation of 3D models, in applications where the shape preservation of the model is important but there is no strict requirement on exact geometry reconstruction. Future work will involve testing the DD metric against human subjects in a subjective distortion experiment, as well as against existing perceptual metrics, to determine whether the DD may be used as an objective perceptual error metric.

8. ACKNOWLEDGMENTS

The authors would like to thank AIM@SHAPE and Gabriel Peyré for providing the 3D models used in this paper.

9. REFERENCES

[1] Ahn, J.-H., Chang, E.-Y. and Ho, Y.-S. 2001. Shape-Preserving Progressive Coding of 3-D Models. In *IEEE Image Proc.* (Thessaloniki, Greece, Oct. 7-10, 2001). ICIP 2001. IEEE, vol. 3, 118-121.

[2] Ahn, J.-K., Lee, D.-Y., Ahn, M., Kim, J.D.K., Kim, C.Y. and Kim, C.-S. 2010. Progressive Compression of 3D Triangular Meshes using Topology-Based Karhunen-Loève Transform. In *IEEE Image Proc.* (Hong Kong, China, Sept. 26-29, 2010). ICIP 2010. IEEE, 3417-3420.

[3] Cignoni, P., Rocchini, C. and Scopigno, R. 1998. Metro: Measuring Error on Simplified Surfaces. *Comput. Graph. Forum*. 17, 2 (Jun. 1998), 167-174.

[4] Garland, M. and Heckbert, P.S. 1997. Surface Simplification Using Quadric Error Metrics. In *Comp. Graph.* (Los Angeles, CA, Aug. 3-8, 1997). SIGGRAPH 97. ACM, 209-216.

[5] Hill, F.S., Jr. and Kelley, S.M., Jr. 2007. Rendering Faces for Visual Realism. In *Computer Graphics: Using OpenGL*. Pearson Prentice Hall, Upper Saddle River, NJ, 426-428.

[6] Khodakovsky, A., Schröder, P. and Sweldens, W. 2000. Progressive Geometry Compression. In *Comp. Graph.* (New Orleans, LA, Jul. 23-28, 2000). SIGGRAPH 2000. ACM, 271-278.

[7] Krivokuća, M., Wuensche, B., Abdulla, W. and Lavoué, G. 2012. Investigating the Rate-Distortion Performance of a Wavelet-Based Mesh Compression Algorithm by Perceptual and Geometric Distortion Metrics. In *The 20th Int.Conf.on Computer Graphics, Visualization and Computer Vision. Communications Proc. (Part II)* (Plzen, Czech Republic, Jun. 25-28, 2012). WSCG 2012. Vaclav Skala - Union Agency, 259-268.

[8] Lavoué, G. 2011. A Multiscale Metric for 3D Mesh Visual Quality Assessment. *Comput. Graph. Forum*. 30, 5 (Aug. 2011), 1427-1437.

[9] Lavoué, G. and Corsini, M. 2010. A Comparison of Perceptually-Based Metrics for Objective Evaluation of Geometry Processing. *IEEE T. Multimedia*. 12, 7 (Nov. 2010), 636-649.

[10] Lindstrom, P. and Turk, G. 2000. Image-Driven Simplification. *ACM Trans. Graph*. 19, 3 (Jul. 2000), 204-241.

[11] NVIDIA Corporation. 2012. Quadro Product Comparison. Available at http://www.nvidia.com/content/PDF/product-comparison/Quadro-Product-Comparison.pdf.

[12] Park, I.K., Lee, K.M. and Lee, S.U. 2002. Efficient Measurement of Shape Dissimilarity Between 3D Models Using Z-Buffer and Surface Roving Method. *EURASIP J. Appl. Signal Process.* 10, 1 (Jan. 2002), 1127-1134.

[13] Payan, F. and Antonini, M. 2006. Mean Square Error Approximation for Wavelet-Based Semiregular Mesh Compression. *IEEE T. Vis. Comput. Gr.* 12, 4 (Jul./Aug. 2006), 649-657.

[14] Rakhmanov, E.A., Saff, E.B. and Zhou, Y.M. 1994. Minimal Discrete Energy on the Sphere. *Math. Res. Lett.* 1, 6 (1994), 647-662.

[15] Rogowitz, B.E. and Rushmeier, H.E. Are Image Quality Metrics Adequate to Evaluate the Quality of Geometric Objects? In *P. Soc. Photo-Opt. Ins.* (San Jose, CA, Jan. 22-25, 2001). SPIE, vol. 4299, 340-348.

.