Roughness as a Shape Measure

Rakesh Kushunapally[1], Anshuman Razdan[2] and Nathan Bridges[3]

[1]PDI/Dreamworks, rkushunapa@pdi.com
[2]Arizona State University, razdan@asu.edu
[3]Jet Propulsion Laboratory, Nathan.bridges@nasa.gov

ABSTRACT

In this paper, we present a measure to quantify the multi-neighborhood level roughness of the surface using mean curvature. The surface is scaled to a unit sphere to enable comparison between different models with scale taken out of the comparison equation. Our roughness measure can be used as a shape indicator of the surface as it gives information regarding the vertex distribution at multiple neighborhood levels. In addition to computing the surface roughness, we use our measure as a unifying method to analyze different smoothing algorithms on different models including the effect of different vertex updating methods. We also use our measure to illustrate the differences in roughness at different neighborhood levels due to the irregular sampling of the surface. We specifically target triangle surface mesh representation for this paper, as it is the most common and other polygonal models can be converted to it by using local triangulation. Results are presented to demonstrate the usefulness of our roughness measure.

Keywords: Roughness, multi-neighborhood level, local roughness, global roughness, mean curvature, mesh smoothing.

1. INTRODUCTION

Recent advances in 3D data acquisition technology have given rise to large and complex triangle mesh models. Many new mesh smoothing and simplification techniques have been developed to handle such complex meshes. With the increase in mesh complexity and number of methods available, it is always difficult to select an appropriate method for a particular mesh as different methods produce different results on the same mesh. There is no unified way of comparing these methods as each method tries to achieve its goal by minimizing various criteria, which are not common to all the methods. Thus, more information about a surface is needed to come up with a better prediction of the effect of these methods on the surface. One such piece of information is the roughness of the surface at different neighborhood levels. Since roughness is opposite to smoothness, surface roughness information is valuable. A k -ring neighborhood level of a vertex x_i is defined as the union of all vertices, which are connected to x_i by at most k edges.

In this paper we present a measure to quantify the multi-neighborhood level roughness of the surface using an intrinsic property of the mesh. Traditional methods in physical sciences and engineering treat roughness as a stochastic measure whereas we compute it over the complete surface at different neighborhood levels. We are specifically targeting triangle surface mesh representations for this paper, as it is the most common and other polygonal models can be converted to it by using local triangulation.

The remainder of this paper is organized as follows. Section 2 reviews related work. In section 3, we introduce the concept of neighborhood level roughness, local roughness and global roughness of the surface. Section 4 presents our measure to compute multi-neighborhood level roughness of the surface. We discuss applications of our roughness measure in section 5. Results are presented and a comparison between different smoothing methods is done in section 6. Finally, conclusions are drawn in section 7.

2. RELATED WORK

The notion of surface roughness is the subject of a large variety of investigations in different fields of engineering and sciences. Physical and chemical properties of surfaces are to a significant degree determined by their surface structure.

The effect of surface roughness is being studied in different fields like industrial engineering, oceanography and atmospheric sciences [19].

Different methods have been used in various fields to measure the roughness of a surface. In case of 1D signals, the most common methods of distortion measurement involve measuring the heights of the sampled points from a horizontal line. The resulting series of values are used in equations like maximum height difference (R_z), mean value of absolute heights (R_a) and root mean square roughness (R_q) to measure the noise in the signal. If the measured heights are represented as $z(x)$, then R_z , R_a and R_q are defined as

$$R_z = max(z(x)) - min(z(x)). \tag{1}$$

$$R_a = \frac{1}{N}\sum_{x=1}^{N} z(x). \tag{2}$$

$$R_q = \sqrt{\frac{1}{N}\sum_{x=1}^{N}(z(x) - \overline{z(x)})^2}. \tag{3}$$

where N is the number of sampled points and $\overline{z(x)}$ is the average height. R_z and R_a are generally used in the context of industrial and engineering applications whereas R_q is used generally in physical and other sciences [19]. The above approaches have been extended to the surface case by using the surface profile information in equations (1), (2) and (3). But these simple equations fail to completely characterize the complexity of surface roughness. Also, these methods fail to measure the multi-neighborhood level roughness of the surface.

Rippa [12] defines roughness of a voronoi triangulation as the L^2 norm squared of the gradient of the piecewise linear surface, integrated over the triangulated region. The roughness of the data vector F , relative to triangulation T is given by

$$R(F,T) = | f_t |. \tag{4}$$

where $|.|_{T,1}$ is the Sobolev semi-norm defined as:

$$| g |_{T,1} = \sum_{i=1}^{n_t} | g |_{T_i,1}. \tag{5}$$

$$| g |_{T_i,1} = \int_{T_i} [\frac{\partial g}{\partial x}^2 + \frac{\partial g}{\partial y}^2]dxdy. \tag{6}$$

where n_t is the number of triangles. Jian-Hua Wu et al in [8] proposed a measure to compute the per-face roughness of the mesh by making statistical considerations about the dihedral angles and valence of the vertices associated with each face of the mesh. Even though these methods successfully compute the local roughness, i.e. the roughness at one ring neighborhood level; they do not compute the multi-neighborhood level roughness of the surface as our measure does.

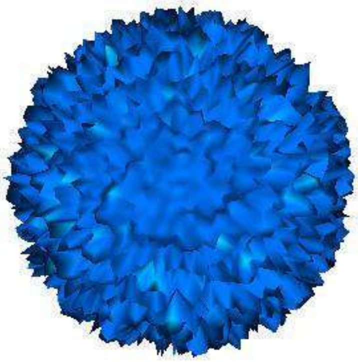
A modified version of [8] is proposed by Corsini et al in [3], where the per-vertex roughness is computed by averaging the roughness of all the faces in the neighborhood of the vertex. In spite of taking multiple neighborhood levels in to consideration, this method can not compute the roughness at higher neighborhood levels as simple averaging does not reflect the roughness of the vertices with respect to larger regions. This can be observed in fig 1(e), where the global roughness of the noisy sphere, computed using the method in [3], is greater than the local roughness, which is quite opposite to reality since globally its shape is still close to a sphere.

Lee et. al propose *mesh saliency* - a perceptually-based measure of regional importance [20]. Saliency is formulated in terms of mean curvature. A Gaussian filter is placed over a vertex and used to compute a weighted average of the mean curvature. This average is computed at multiple "scales" where the scale is varied by changing the cut-off for the Gaussian filter. The saliency at a vertex is computed as the absolute difference between the Gaussian weighted average of the mean curvature at a finer and coarser scale. The motivation for the multi-scale formulation is that all features are not important at all scales. The authors consider scales of up to 1.8 % the length of the body diagonal of the object bounding box. Neighbourhoods larger than this are not considered. Our method evaluates roughness at arbitrarily large neighbourhoods and accounts for features of any size (a sizeable fraction of the bounding box).

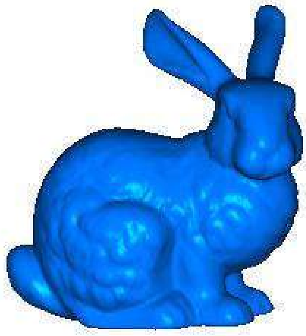
3. LOCAL AND GLOBAL ROUGHNESS

In this section, we introduce the concept of *local roughness* and *global roughness* of a surface. We will then demonstrate the usefulness of these notions.

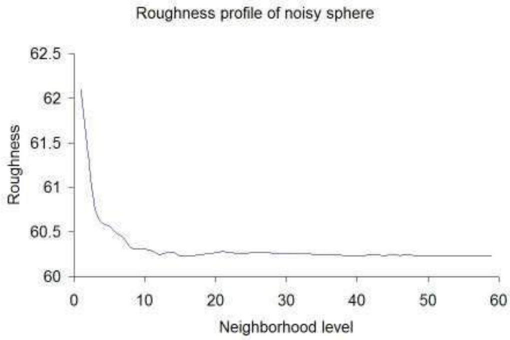
The local roughness of a surface describes the average behavior of all the vertices in their local region, generally a one ring neighborhood, whereas the global roughness of a surface describes the average behavior of all the vertices with respect to a larger region, generally a complete surface. In other words, local roughness gives the average high frequency roughness of a surface whereas global roughness gives the average low frequency roughness. The roughness profile or local to global roughness plot for a given object captures the information about the roughness of an object from the local sense (region around vertices) to the global sense (entire object).



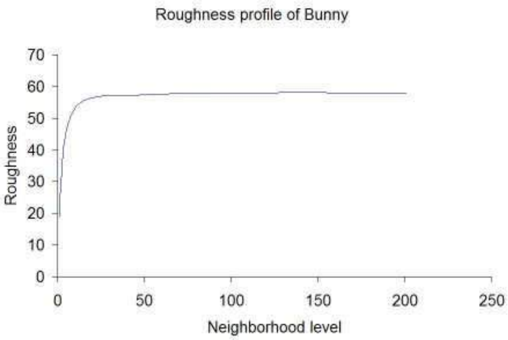
(a)



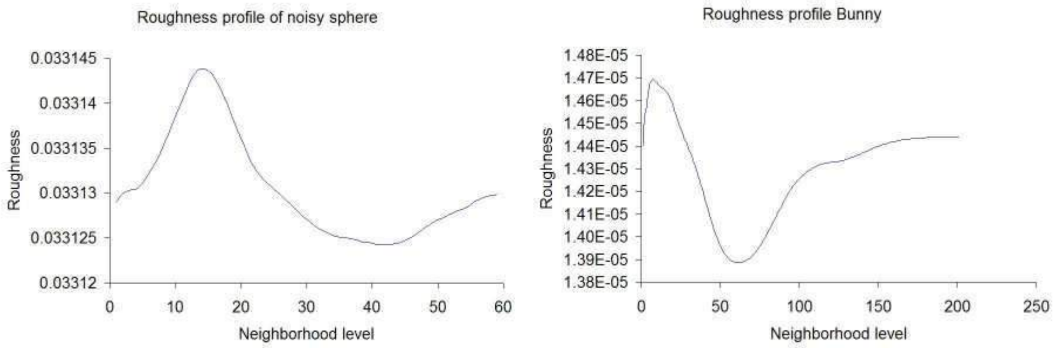
(b)



(c)



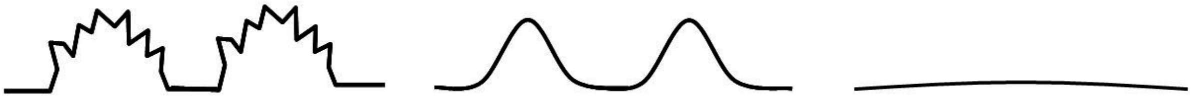
(d)



(e) (f)

Fig. 1: (a) Sphere with 14% noise added in radial direction (b) Bunny and their roughness profiles. (c) & (d) are computed using the roughness measure proposed in this paper, whereas (e) & (f) are computed using the method in [3].

Fig 1(a) & 1(b) show a triangle mesh approximation of a sphere with noise added and its roughness profile respectively. Noise is randomly added by radially perturbing the vertices by 14%. It can be observed that, the local roughness is high compared to the global roughness as the noise is added locally and the sphere still maintains its spherical shape, while for the bunny, shown in fig 1(d), the global roughness is high compared to the local roughness as it contains global features like ears, head, legs etc.



(a) (b) (c)

Fig. 2: (a) A curve with noise distributed at multiple neighborhood levels, (b) after removing local roughness, (c) and after removing roughness at higher neighborhood levels.

In Fig 2, which shows a 1D case, the curve is rough locally as well as globally. Just removing the local roughness doesn't make the curve smooth. A smoothing algorithm should proceed until it removes the roughness at higher neighborhood levels. Similar argument can be made in surface case. Moreover, the smoothing of some models, like the bunny, may require an algorithm to focus more on the local roughness than the global roughness, which in turn means preserving the features while removing the noise, while some other models may require the algorithm to remove the noise at higher neighborhood levels. Different smoothing methods produce different results with the same mesh. Availability of the roughness profile of an object allows the user to compare the results of different smoothing methods on the same mesh and make an informed decision in selecting the number of smoothing iterations.

4. MULTI-NEIGHBORHOOD LEVEL ROUGHNESS

The disadvantage of using the surface height information is that it depends on a base plane and fails to correctly compute the global roughness of a surface. A possible approach to compute the multi-neighborhood level roughness of a surface is to use the mean curvature of the vertices in the *RMS* roughness equation instead of surface height. We choose mean curvature instead of other curvatures as it is the average of all directional curvatures at a point on the surface and averages any directional curvatures spikes. We select the sphere as the base model as it has constant mean curvature surface and our method implies surfaces with constant mean curvature are the smoothest. A *plane* and synthetic surfaces like *Wente's torii* and *Delaunay unduloid*, which do not exist in practical applications, are other examples of constant mean curvature surfaces. Before computing the mean curvature, we resize the model so that the average distance between the centroid of the model and it's vertices is a unit sphere. Since the curvature is scale dependent, resizing enables the roughness measure to compare two different models on the same grounds.

Given a triangle mesh with N vertices, we propose to compute the surface roughness as follows: First scale the model to a unit sphere and compute the mean curvature at each vertex of the triangle mesh. For each vertex x_i, compute the difference $D(x_i)$ between its mean curvature and the average mean curvature of its k-ring neighborhood, where k is the neighborhood level at which the surface roughness is being computed. $D(x_i)$ gives the information regarding the roughness at the vertex x_i. The surface is smooth at vertex x_i if $D(x_i)$ is zero. Finally, we compute the surface roughness by computing the root mean square of $D(x_i)$ over the entire triangle mesh.

Let $K_H(x_i)$ be the mean curvature of the vertex x_i and $K_H^L(x_i)$ be the average mean curvature of its L-ring neighborhood given by

$$K_H^L(x_i) = \frac{1}{|N(x_i)|} \sum_{x_j \in N(x_i)} K_H(x_j) \tag{7}$$

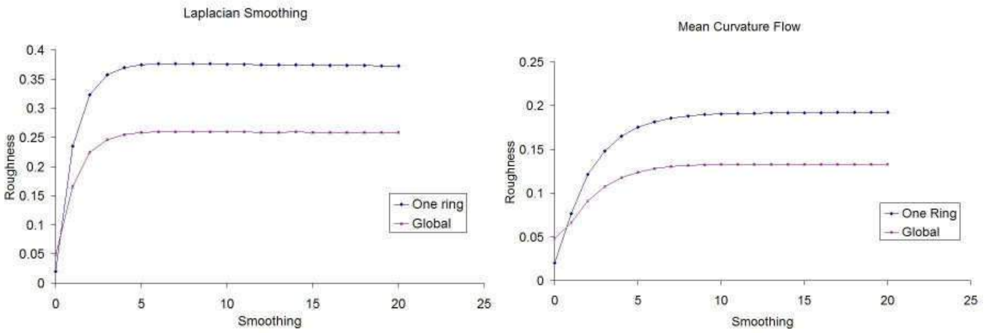
where $N(x_i)$ contains all the vertices in the L-ring neighborhood of the vertex x_i and $|N(x_i)|$ is the number of vertices in $N(x_i)$. Then the equation to compute the surface roughness is given by

$$D(x_i) = K_H(x_i) - K_H^L(x_i), \tag{8}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{x=1}^{N}(D(x_i))^2}. \tag{9}$$

where σ is the roughness of the surface. As L increases, σ represents roughness at higher neighborhood levels. When computing the global roughness of the surface, $N(x_i)$ contain all the vertices except x_i.

The mean curvature $K_H(x_i)$ of the vertex x_i can be computed using different methods proposed in [5, 6, 7, 10, 13, 15, 18]. Initially we tested the discrete curvature estimation method proposed in [10] to compute the mean curvature. This method did not give good results for the surfaces with near zero roughness. This can be observed in fig 3, which shows the effect of different smoothing methods on the triangle mesh approximation of a unit sphere which has a near zero roughness. Discrete curvature approximation method [10] is used to compute the mean curvature and pre-updating method is used i.e. vertices are updated immediately after the new position is computed. It can be observed that the roughness of the sphere increases with smoothing, which is quite opposite to reality. Because of the inability of discrete curvature methods to accurately compute the curvature values, we switched to the continuous curvature computation methods, which can produce more accurate results.



(a) (b)

Fig. 3: Effect of (a) Laplacian smoothing and (b) mean curvature flow smoothing algorithms on the unit sphere. Mean curvature is computed using the discrete curvature approximation method. Similar results are observed in case of Modified mean curvature flow smoothing method.

Razdan et al [13] compared different curvature approximation methods and observed that the cubic-order curvature approximation method proposed by Goldfeather et al [5] is stable and performs better than other methods. In Goldfeather et al method, curvature is computed by fitting a quadric approximation to the neighborhood of the vertices. Normal vectors at adjacent vertices are used to create the third degree terms in the least-square solution.

5. APPLICATIONS

Our roughness measure can be used as a shape indicator of the surface in the sense that it gives the information regarding the vertex distribution and their behavior at multiple neighborhood levels. Our roughness measure can be used to compare models with different geometry and topology at different neighborhood levels. Fig 4 shows the roughness profiles of the Venus and Rocker arm. The Rocker arm is smoother than the Venus at all neighborhood levels. This difference is higher in lower neighborhood levels and lower in the higher neighborhood levels. This is evident from the fact that the Rocker arm contains many flat regions locally but contains many global features like crests and valleys, whereas the Venus contains less global features and more local features like eyes, hair etc.

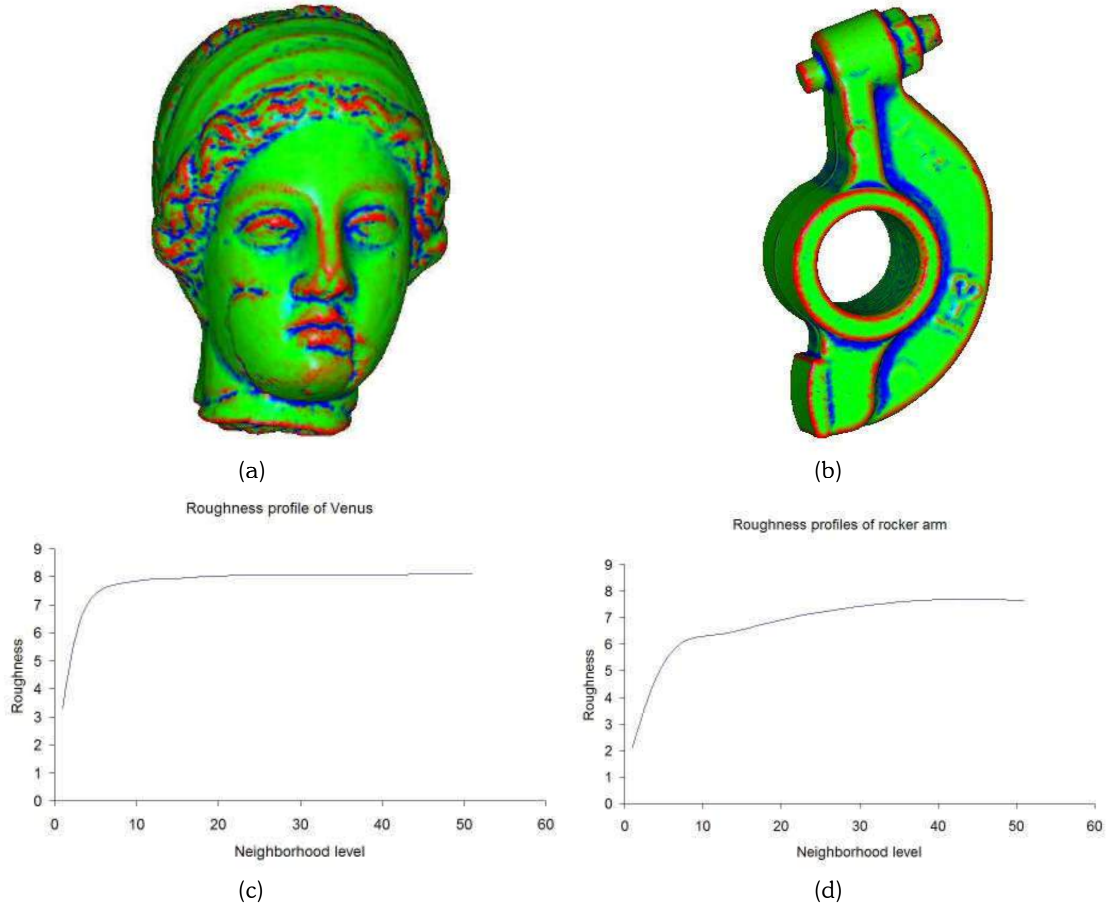


Fig. 4: Mean curvature maps and roughness profiles of (a) & (c) Venus and (b) & (d) Rocker arm.

All the mesh smoothing methods aim at generating a modified mesh according to some criteria, which are different among these methods. However, all these methods aim at producing a visually smooth mesh and this criterion can be used as a unifying ground for comparison. To make this comparison effective, we should define and quantify the smoothness or roughness of the mesh. This is what our roughness measure does. Moreover, our measure enables the comparison at different neighborhood levels. Our measure provides the information whether the noise is distributed locally or at higher neighborhood levels, which helps the user in selecting appropriate smoothing methods. Our measure can also be used to publish the roughness profile of the models, which can assist the user in selecting suitable method for a mesh and come up with a better prediction of number of iterations needed.

Apart from comparing smoothing methods, our roughness measure can also be used to compare different meshes representing same model. Tools like MESH [1], Metro [2] and MeshDev [14] are used to numerically compare triangle meshes. Metro and MESH measure the error in terms of the symmetric mean distance and Hausdorff distance [1] respectively whereas the MeshDev tool measures the error in terms of the geometry deviation. Corsini et al in [3] proposed to measure the error in terms of the roughness of the surface. The error is computed by normalizing the difference in the roughness of two models with respect to the roughness of the original model. The expression used is:

$$R(M,M^{w}) = \frac{\log(R(M) - R(M^{w}))}{R(M)} - \log(k) \quad (10)$$

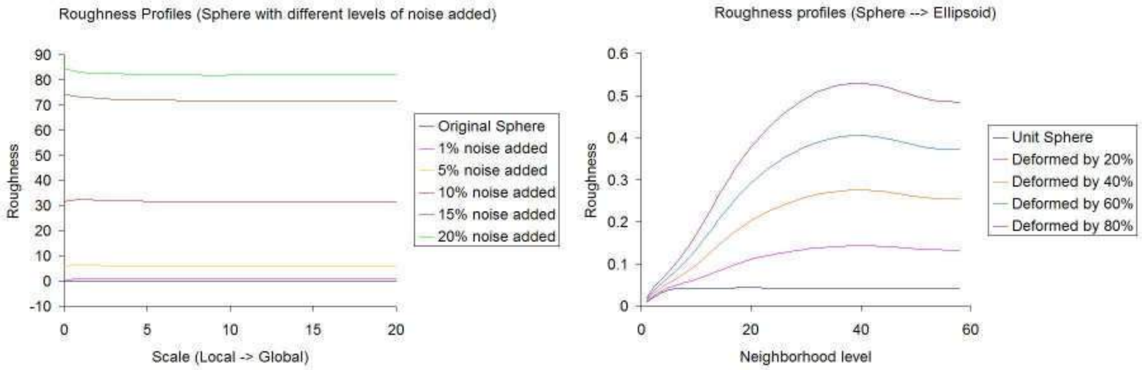
where $R(M)$ is the roughness of the first mesh and $R(M^{w})$ is the roughness of the second mesh, computed as in [3], and k is a constant used to avoid numerical instabilities. Even though, all these measures successfully measure the error between two meshes, they fail to compare the meshes at different neighborhood levels, which provide the user with more information to come up with a suitable mesh simplification and water marking methods for a particular mesh. This multi-neighborhood level comparison can be achieved by using the multi-neighborhood level roughness measure proposed in this paper in equation(10).

6. RESULTS

In this section we will present the results of our surface roughness method and compare five smoothing algorithms.

6.1 Multi-Neighborhood Level Roughness Computation

Fig 5(a) shows the roughness profiles for a sphere with different levels of noise added, computed using our roughness measure. The noise is added in the radial direction of the vertices, i.e. along the normal of the vertices. We observe that, before adding the noise, the roughness of the sphere increases with the neighborhood level as the sphere is approximated by a triangle mesh. As increasing amount of noise is added, we observe an increase in the roughness at all levels. We also observe that the increase in the roughness with the addition of the noise is more at lower neighborhood levels than the higher neighborhood levels. This high increase indicates that the local roughness is sensitive to the noise added locally, hence showing the reliability of our method.



(a) (b)

Fig. 5: Roughness profiles of (a) sphere with different levels of noise added (b) sphere deformed into an ellipsoid.

Fig 5(b) shows the roughness profiles of ellipsoids formed from deforming a sphere by different amounts. As the sphere is deformed more and more, we observe that, the roughness at higher neighborhood levels increases more than the roughness at the lower neighborhood levels. In fact, the change at the lower levels is less. This high increase in the global roughness is due to the fact that the deformation changes the global features of the model.

6.2 Comparison of Smoothing Algorithms

To test the usefulness of our measure, we compared different smoothing methods based on the surface roughness. We selected five polyhedral surface smoothing methods: Laplacian smoothing, Taubin's method [16, 17], bilaplacian smoothing [9], Mean curvature flow [4], and the modified mean curvature flow [11]. For a brief introduction of these methods see [11]. We chose the following four triangular mesh models for the comparison.

• **SphereN:** Unit sphere with 4% of noise added by randomly perturbing the vertices along their normals. This mesh contains approximately 8k triangles. Local roughness is higher than the global roughness because of the local noise added.

• **Ellipsoid:** Formed by deforming a unit sphere along the diameter by 20%. This mesh contains approximately 8k triangles. Since the deformation only changes the global features, global roughness is higher than the local roughness.

• **VenusN:** Venus model with 0.004% noise added by randomly perturbing the vertices in the direction of their normals. This model contains approximately 134K triangles. Since the noise added is very less, global roughness of the model is still higher than the local roughness.

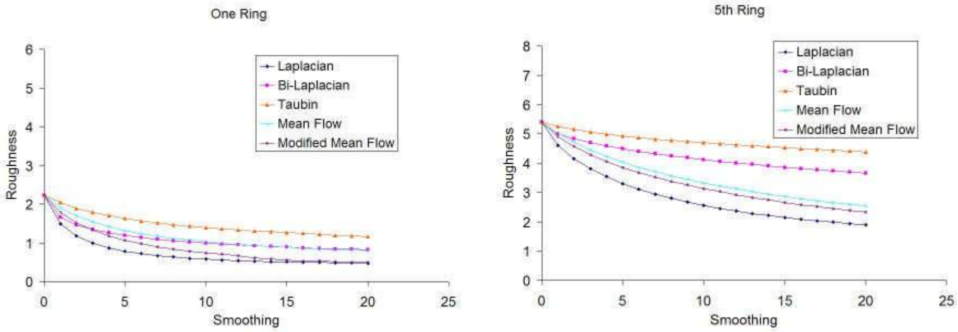
• **Bunny:** Stanford Bunny model with approximately 70K triangles. Because of the global features like creases, ears, tail etc., global roughness of this model is greater than the local roughness.

The smoothing of Bunny and VenusN requires the algorithms to remove the local roughness and preserve the global features, whereas for the ellipsoid, the algorithms should remove the global roughness. In case of sphere, both global and local roughness must be removed.

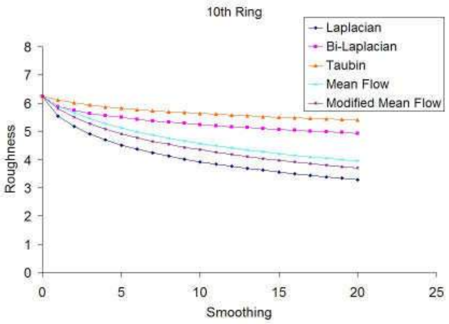
Fig 6 shows the effect of different smoothing methods on Bunny. As shown in the figure, Laplacian smoothing, mean curvature flow and modified mean curvature flow methods perform well at lower neighborhood levels. At higher neighborhood levels, Taubin's method and bilaplacian method perform well by showing less effect on the global roughness i.e. by preserving the global features. The performance of these two methods is competitive with other methods at local levels. In other words, these two methods remove local roughness while preserving the global features of the mesh. Similar results are observed in case of VenusN in fig 7.

In case of noisy sphere (sphereN) in fig 8, we observe that the Laplacian smoothing performs better than other methods by effectively removing the local roughness as well as the global roughness. The performance of the mean curvature flow and the modified mean curvature flow are close to that of the Laplacian method. We also observe that, if smoothed beyond certain point, Laplacian, bilaplacian and Taubin's methods introduce noise to the model. This effect can be clearly observed in case of Ellipsoid in fig 9. Only the mean curvature flow and the improved mean curvature flow perform better in this case. The reason for this behavior can be attributed to the topology of the models. We observe that, the Laplacian and the Taubin's methods creates nodules at some vertices on the ellipsoid whose valance is other than six. This is shown in fig 10, which shows the mean curvature maps for Ellipsoid after smoothing with different smoothing methods. It can be observed that, most of the vertices on Ellipsoid contain six neighbors in their one ring neighborhood and the Laplacin and the Taubin's methods create high curvature regions around only those vertices which have only five neighbors.

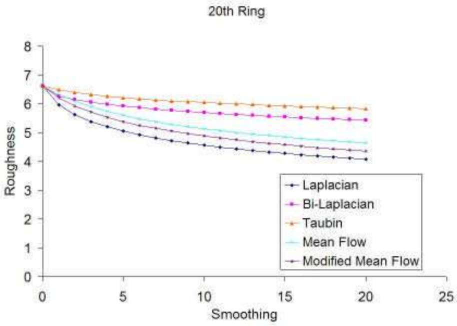
From fig 6-10 we observe that the Taubin's method and the bilaplacian method perform well if the global features of the models have to be preserved, whereas the Laplacian, the mean curvature flow and the modified mean curvature flow methods perform well if the local as well as the global roughness has to be removed. We also observe that Laplacian, bilaplacian and Taubin's methods will introduce noise if used on locally smooth surfaces with non uniform topology. Depending on the end user application, our roughness measure can be used to get an assessment of which algorithm is optimal for a model.



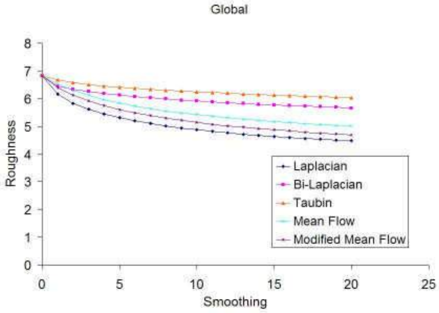
(a) (b)



(c)

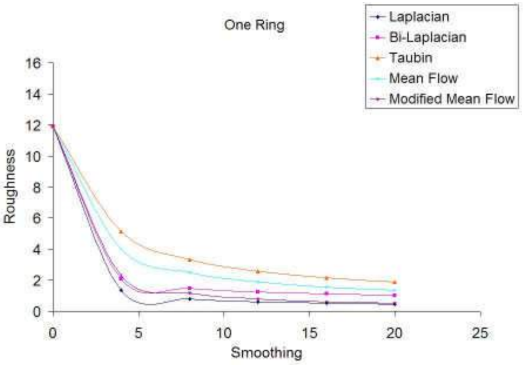


(d)

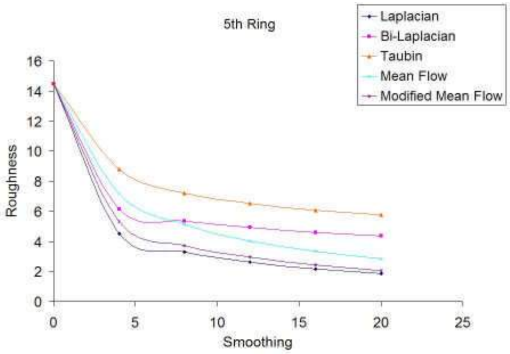


(*e*)

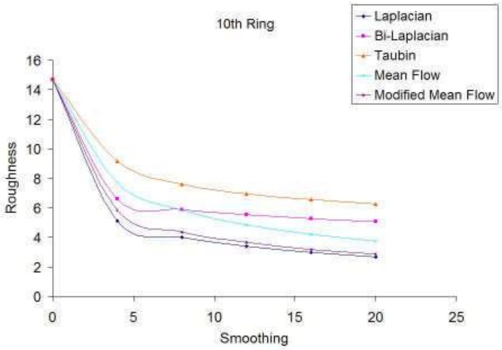
Fig. 6: Comparison of different smoothing algorithms on bunny at different neighborhood levels (a) One ring (b) five ring (c) 10 ring (d) 20 Ring neighborhood and (*e*) Global.



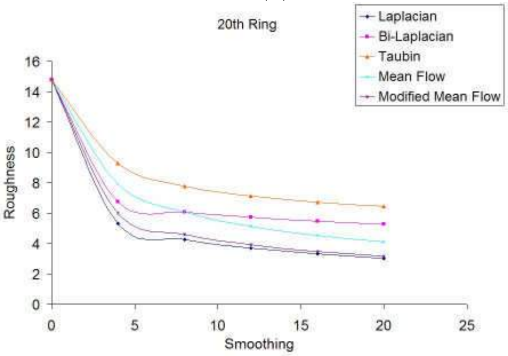
(a)



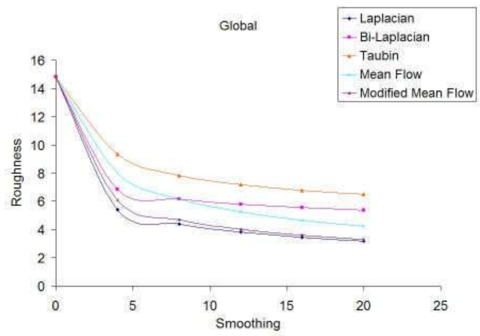
(b)



(c)

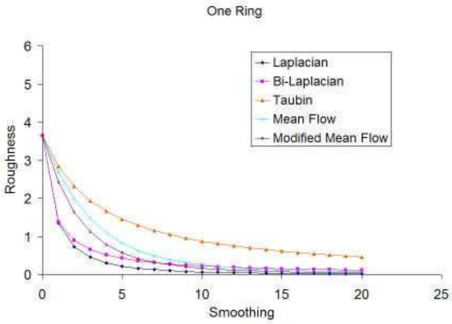


(d)

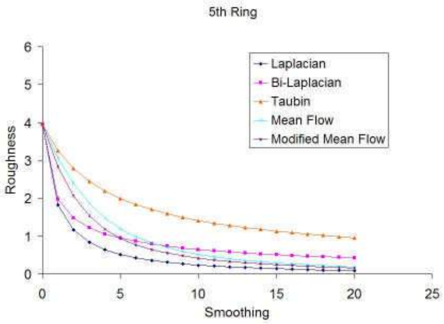


(e)

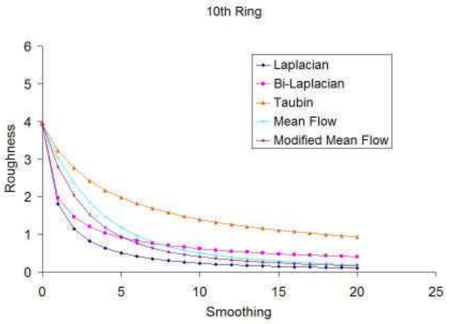
Fig. 7: Comparison of different smoothing algorithms on VenusN at different neighborhood levels (a) One ring (b) five ring (c) 10 ring (d) 20 Ring neighborhood and (e) Global.



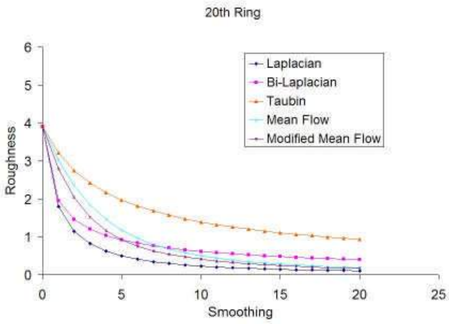
(a)



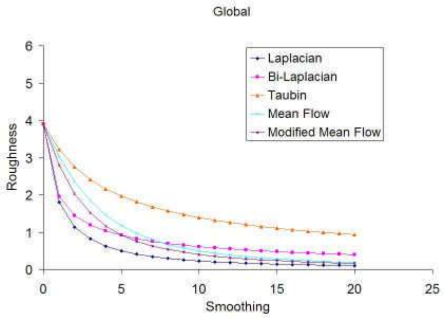
(b)



(c)



(d)



(e)

Fig. 8: Comparison of different smoothing algorithms on sphereN at different neighborhood levels (a) One ring (b) five ring (c) 10 ring (d) 20 Ring neighborhood and (e) Global.

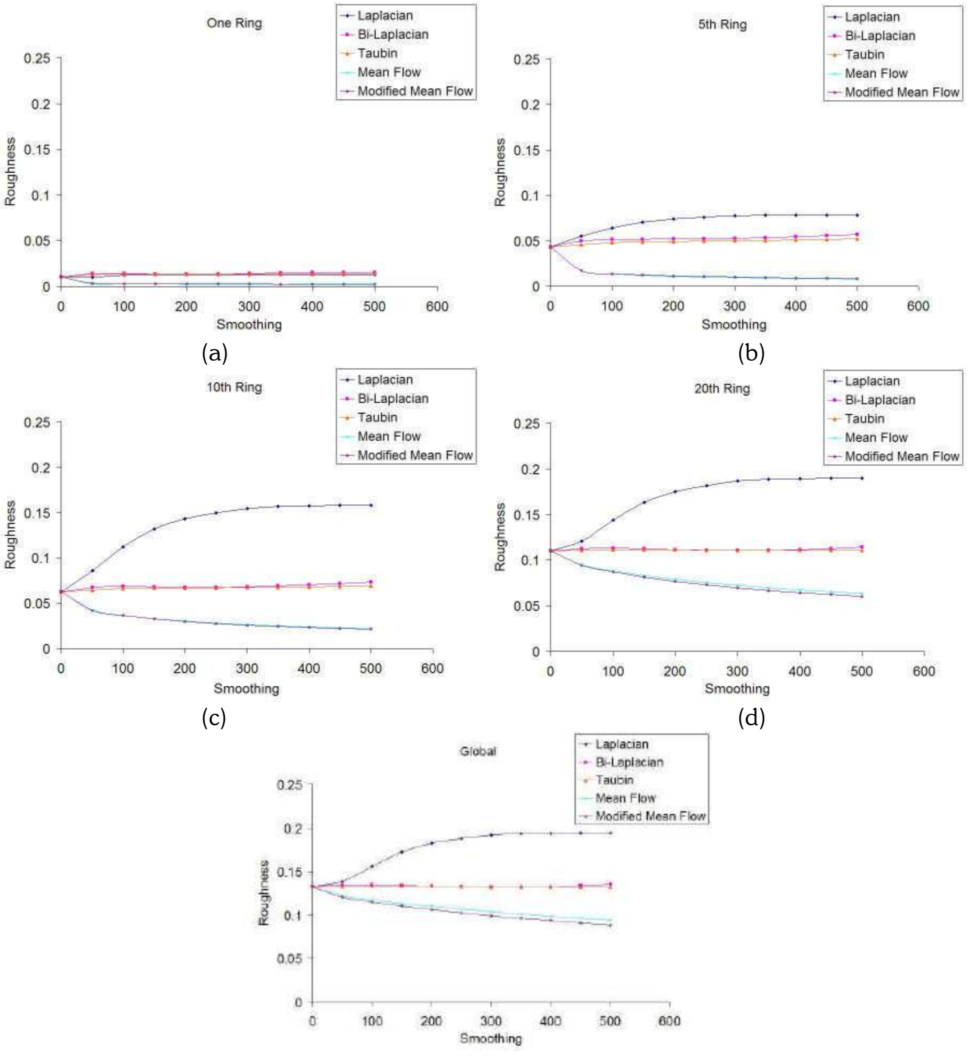
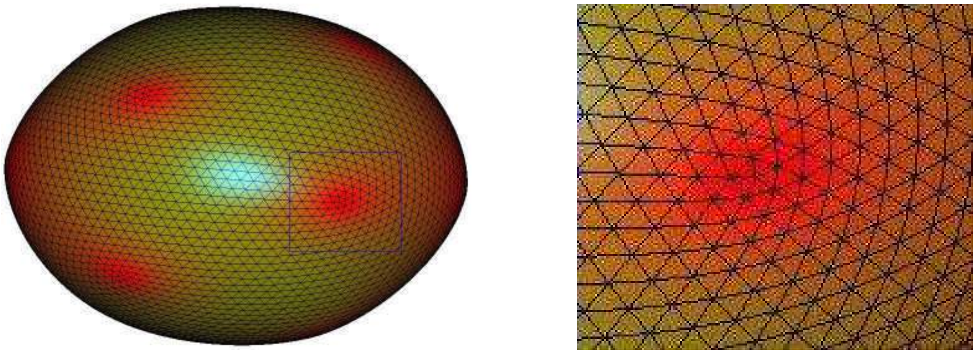
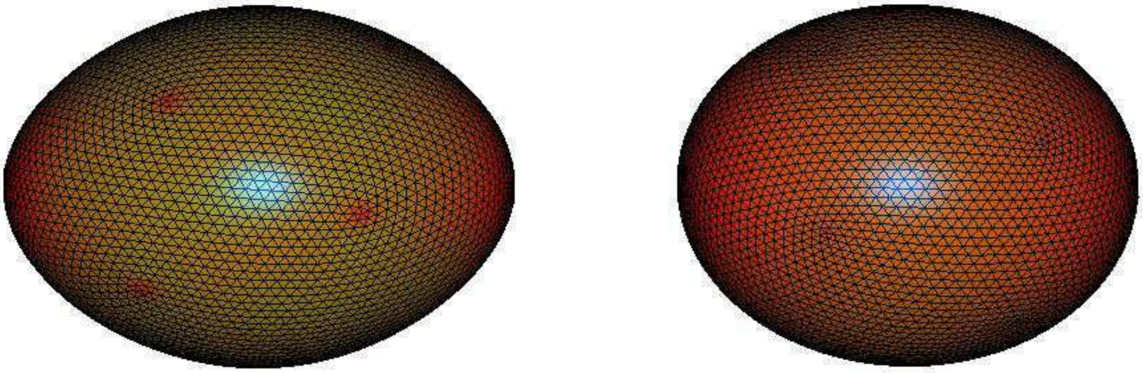


Fig. 9: Comparison of different smoothing algorithms on ellipsoid at different neighborhood levels (a) One ring (b) five ring (c) 10 ring (d) 20 Ring neighborhood and (e) Global.



(a) (b)

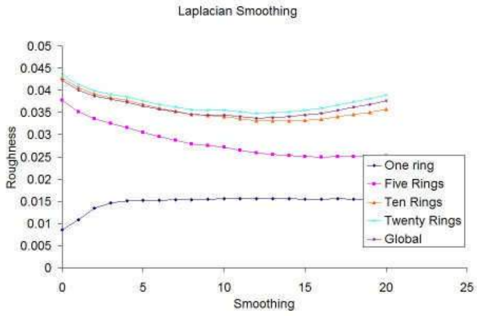


(c) (d)

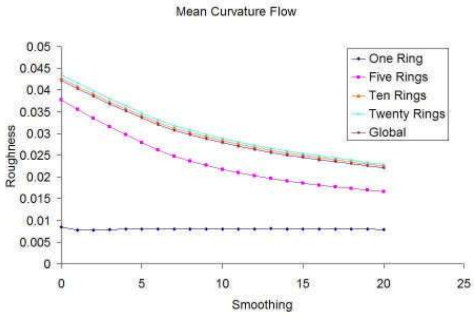
Fig. 10: Mean curvature maps of ellipsoid after smoothing with (a) Laplacian method (c)Taubins method (d) Mean curvature flow method. Close look at the boxed region in (a) is shown in (b). Red color represents regions with high curvature and green color represents regions with zero curvature.

6.3 Pre-updating Vs Post-updating

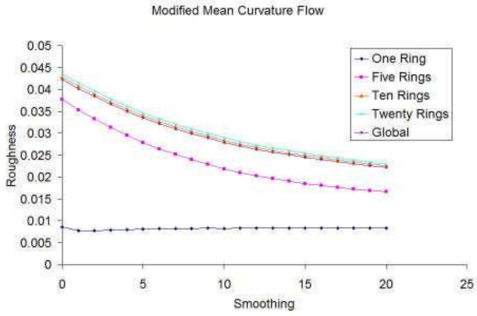
In this paper, we also compare two different mesh updating approaches used by smoothing algorithms - pre-updating and post-updating. Post-updating updates the positions of the vertices after the algorithm completes one iteration while pre-updating updates the vertex positions immediately after the new position is computed. Sometimes the results can be vastly different depending on which method is used. Our measure can be used to quantify the effect of pre-updating method versus post-updating method. For a sphere, as shown in fig 11, post-updating gives better results than the pre-updating method for all smoothing methods. In case of Bunny, Venus and Rocker arm, as shown in fig 12, both pre and post updating performs well at local neighborhood levels. But at higher neighborhood levels post-updating method performs better than the pre-updating method by preserving the global features. From these results we observe that, post-updating preserves the global features, whereas pre-updating method performs better when the global roughness must be removed.



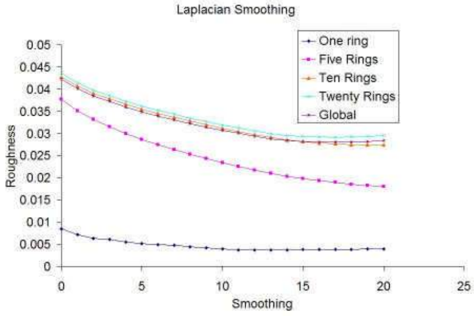
(a)



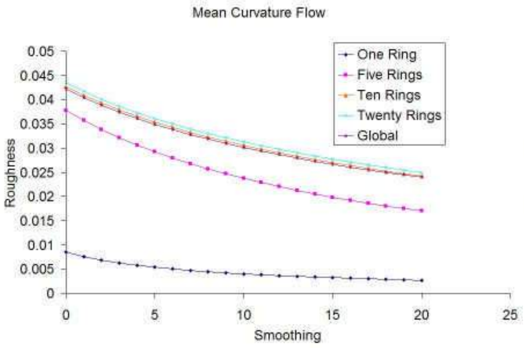
(b)



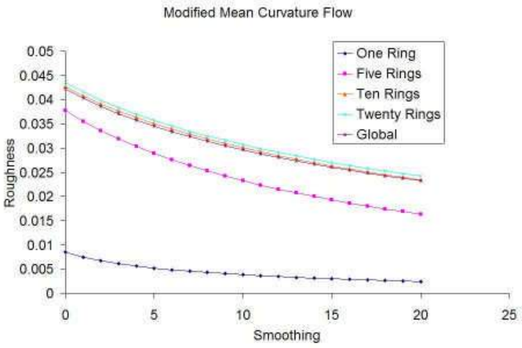
(c)



(d)

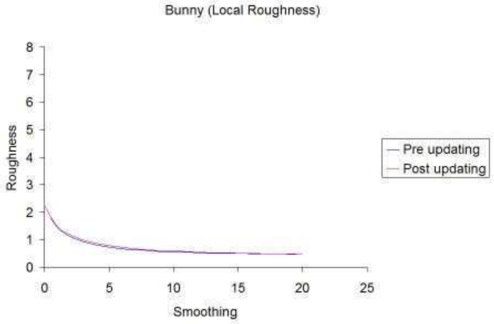


(e)

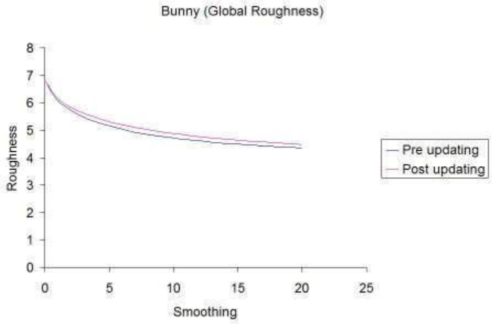


(f)

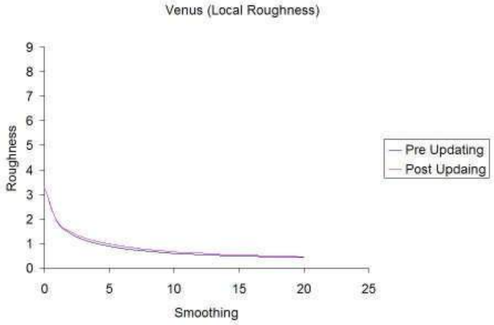
Fig. 11: Laplacian, mean flow and modified mean flow of sphere using pre- and post-updating. (a) (b) & (c) use pre-updating and (d) (e) & (f) use post-updating.



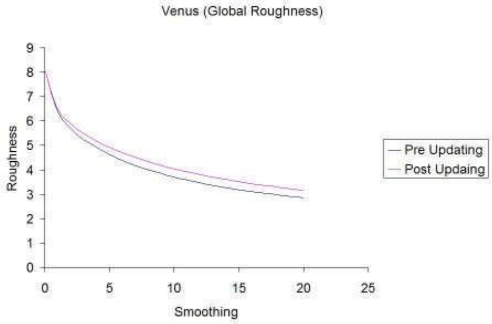
(a)



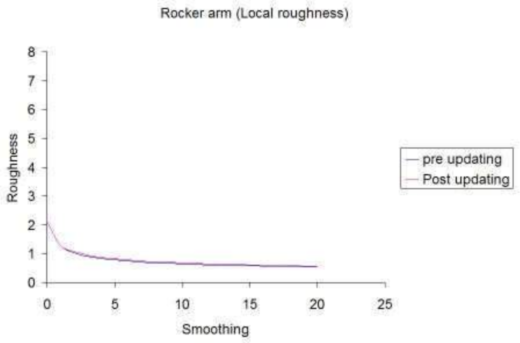
(b)



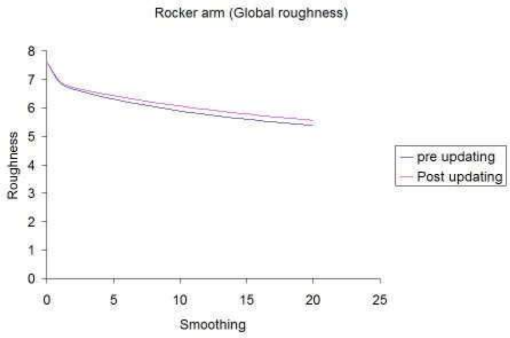
(c)



(d)



(e)



(f)

Fig. 12: Comparison of Laplacian smoothing at local and global levels using pre- and post- updating methods on (a) & (b) Bunny (c) & (d) Venus (e) & (f) Rocker arm. Similar results are observed in the case of mean curvature flow and the modified mean curvature flow.

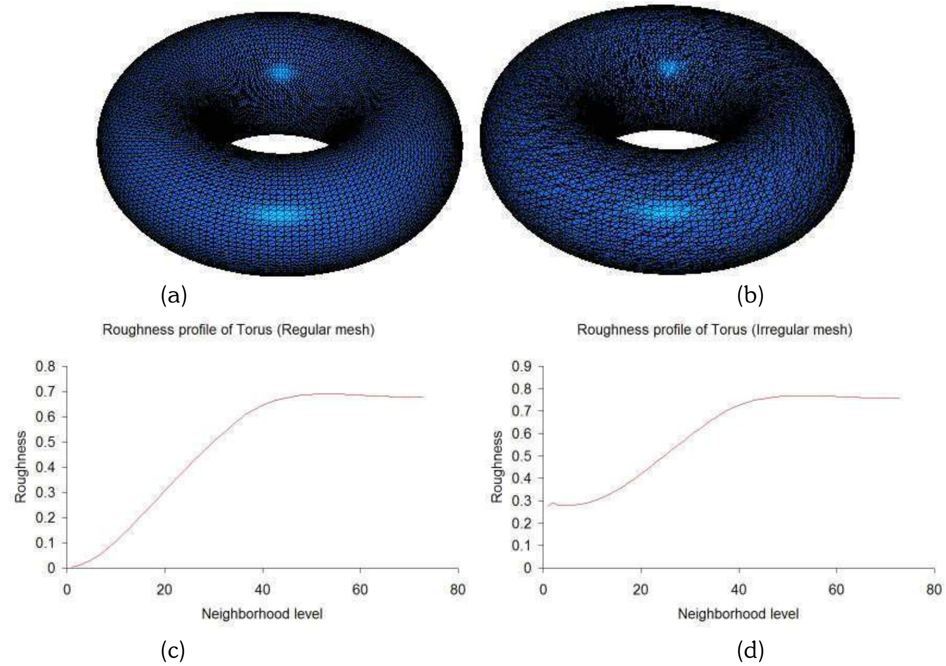


Fig.13: Roughness profiles for torus (a) & (c) regular mesh (b) & (d) irregular mesh.

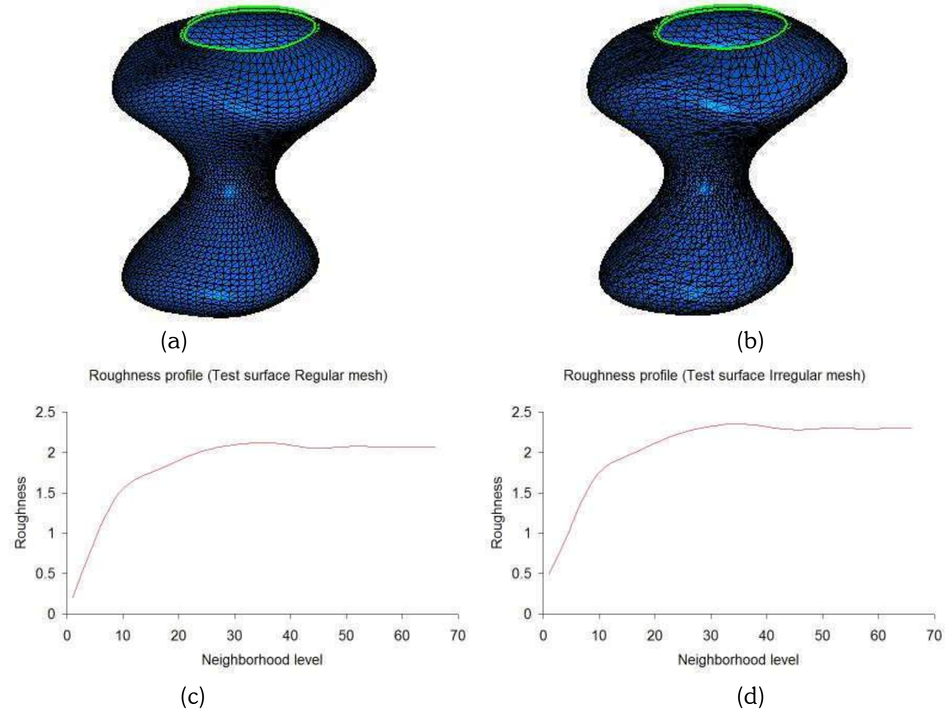


Fig. 14: Roughness profiles for a test surface shown in [5] (a) & (c) regular mesh (b) & (d) irregular mesh.

6.3 Regular Vs Irregular Triangulation

Fig 13 shows the roughness profiles for a torus represented by two different meshes, a regular mesh, formed by regularly sampled vertices on the surface, and an irregular mesh, formed by irregularly sampled vertices on the surface. It can be observed that, the irregularity in the mesh will increase the local roughness of the surface but has very little effect on the roughness at global level. This highlights the challenge of good curvature estimation. Similar results can be observed on Goldfeather's surface [5] in fig 14. From fig 13 & 14, we can say that the global roughness of a surface does not depend on the approximating mesh but mainly depends on the global features of the surface.

7. CONCLUSIONS AND FUTURE WORK

We have presented a measure which successfully measures the multi-neighborhood level roughness. Our measure uses the mean curvature, which is an intrinsic property of the surface to compute the surface roughness. The model is scaled to a unit sphere before computing the mean curvature so that different models can be compared on the same grounds. The benefits of using this measure are demonstrated with the help of relevant data. We also introduced the concept of local roughness, global roughness and roughness profile of a surface. The computations are simple, making our scheme easy to implement.

Our measure can also be used as a shape indicator of the surface. Depending on the application, our measure can help the user in selecting a suitable smoothing method for a particular mesh and to make an intelligent decision in selecting the number of iterations instead of guessing. We have demonstrated the usefulness of our measure in providing a unified method to compare different smoothing algorithms. A comparison between the pre and post-updating methods is done and we observed that the post-updating method generally produces better results than the pre-updating method. The effect of the irregularities in the mesh on the surface roughness is also studied and an observation has been made that the local roughness of the surface depends on the approximating mesh of the surface, while the global roughness does not depend on the approximating mesh.

For future work, we would like to investigate the possibility of using our roughness measure for comparing the performance of the decimation and watermarking methods.

8. ACKNOWLEDGMENTS

This work was supported in part by Jet Propulsion Laboratory (grant 1260617). The authors would like to thank Partnership for Research in Spatial Modeling (PRISM) research center at Arizona State University. Special thanks to John Femiani, Saif Ali and Pushpak Karnick for review and corrections.

9. REFERENCES

[1] Aspert, N.; Santa-Cruz, D.; Ebrahimi, T.: MESH:- Measuring Error between Surfaces using the Hausdorff distance, Proceedings of the IEEE ICME (2002), 2002, vol. I, 705-708.

[2] Cingnoni, P.; Rocchini, C.; Scopigno, R.: Metro: Measuring Error on Simplified Surfaces, Computer Graphics Forum 17, 1998, 167-174.

[3] Corsini, M.; Drelie Gelasca, E.; Ebrahimi, T.: A Multi-Scale Roughness Metric for 3D Watermarking Quality Assessment, Workshop on Image Analysis for Multimedia Interactive Services 2005.

[4] Desbrun, M.; Meyer, M.; Schroder, P.; Barr, A. H.; Implicit Fairing of Irregular Meshes using Diffusion and Curvature Flow, Computer Graphics (Proceedings of SIGGRAPH 99), 317-324.

[5] Goldfeather, J.; Interrante, V.: A novel Cubic-order Algorithm for Approximating Principal Direction Vectors, ACM Transactions on Graphics (TOG) 2004, 23(1), 45-63.

[6] Hamann, B.: Curvature Approximation for Triangulated Surfaces, Computing Suppl., Vol. 8, 1993, 139-153.

[7] Theisel, Holger; Rossl, Christian; Zayer, Rhaleb; Seidel, Hans-Peter: Normal Based Estimation of the Curvature Tensor for Triangular Meshes, 12th Pacific Conference on Computer Graphics and Applications (2004), 288-297.

[8] Wu, Jian-Hua; Hu, Shi-Min; Sun, Jia-Guang; Tai, Chiew-Lan:. An Effective Feature-Preserving Mesh Simplification Scheme Based on Face Constriction, Proceedings of the 9th Pacific Conference on Computer Graphics and Applications, 2001, 12-21.

[9] Kobbelt, L.; Campagna, S.; Vorsatz, J.; Seidel, H.-P.: Interactive Multi-resolution Modeling on Arbitrary Meshes, ACM Computer Graphics (Proceedings of SIGGRAPH '98), 105-114.

[10] Meyer, M.; Desbrun, M.; Schröder, P.; Barr, Alan H.: Discrete Differential-Geometry Operators for Triangulated 2-Manifolds, Proceedings of VisMath (2002), 35-57.

[11] Ohtake, Y.; Belyaev, A.G.; Bogaevski, I.A.: Polyhedral Surface Smoothing with Simultaneous Mesh Regularization, Geometric Modeling and Processing 2000, April 2000, 229-237.
[12] Rippa, S.: Minimal Roughness Property of the Delaunay Triangulation, Computer Aided Geometric Design 7(6), November 1990, 489-497.
[13] Razdan, A.; Bae, M.: Curvature Estimation Scheme for Triangle Meshes Using Biquadratic B´ezier Patches, Accepted for publication in CAD journal.
[14] Roy, M.; Nicolier, F; Foufou, S.; Turchetet, F.; Koschan, A.; Abidi, M.: Assessment of Mesh Simplification Algorithm Quality, In Proceedings of SPIE Electronic Imaging, January 2002, 128-137.
[15] Rusinkiewicz, Szymon: Estimation of Curvatures and their Derivatives on Triangle Meshes, 2nd International Symposium on 3D Data Processing (2004), 486-493.
[16] Taubin, G.: A Signal Processing Approach to Fair Surface Design, Computer Graphics (Proceedings of SIGGRAPH' 95), 351-358.
[17] Taubin, G.: Curve and Surface Smoothing without Shrinkage, Proceedings of the Fifth International Conference on Computer Vision, 1995, 852-857.
[18] Taubin, G.: Estimating the Tensor of Curvature of a Surface from a Polyhedra Approximation, ICCV (1995), 902-907.
[19] Waechter, M.; Kantz H.; Peinke J.: Stochastic Analysis of Surface Roughness, Eoruphys. Lett., 64(5),2003, 579-585.
[20] Lee, Chang Ha; Varshney, Amitabh; Jacobs, David W.: Mesh saliency, ACM Transactions on Graphics, 24(3), 2005, 659-666.