Comparison of discrete Laplacian-Beltrami operators and their applications in surface metrology

H. Huang, W. Zeng, S. Lou, X. Jiang, P. Scott
EPSRC the Future Metrology Hub, Centre for Precision Technologies, School of Computing and Engineering, University of Huddersfield, Huddersfield, HD1 3DH UK

Abstract

In addition to computer image processing, mesh smoothing can also be used in surface metrology to remove noise from free-form surfaces, which are widely represented by triangular meshes. The diffusion equation, also known as the heat equation, is generalised and applied to smooth meshes. The Laplacian operator in the equation is generalised to the Laplacian-Beltrami operator so the equation can be used not only on scalar functions defined in Euclidean space but also on functions defined on non-Euclidean manifolds. The triangular meshes are piecewise linear approximations of a free-form surface. Therefore, the Laplacian-Beltrami operator needs to be discrete.

There are three kinds of discretisation methods widely used currently: uniform graph Laplacian, distance Laplacian and cotangent formula. Their algorithms will be briefly introduced and their performance will be compared in four aspects: smoothness, shrinkage degree, computation time and edge distortion degree. Tests will be conducted on simulated surfaces and measured surfaces. Then, recommendations will be given regarding the requirements of measurement and characteristics of samples.

1 Introduction

Free-form surfaces can be widely found in nature and in many engineering products. For example, in aerospace, the turbine impellers have free-form surfaces. The surface design is critical to engine performance and energy efficiency. In the optical field, compared with a traditional optical component, free-form optical components have not only the advantages of a larger range of views but also the benefit of lower cost due to simplified structures and flexible design [1, 2]. Free-form surfaces are also commonly used in automobiles, such as car bodies [3]. In medical science, a free-form artificial joint is much closer to the real human joint and thus more adapted to the human body [4]. Overall, free-form surfaces have a wide range of applications and great potential for development.

However, most surface characterisation methods for traditional Euclidean surfaces cannot be used directly on free-form surfaces because of their non-zero curvature. To do so, residual surfaces are evaluated instead when the forms are not the object of study[5]. Residual surfaces are generated by comparing measured mesh surfaces and the reference form surfaces. The curved underlying forms are removed, and residuals can be treated as Euclidean surfaces and use traditional methods for further characterisation. In additive manufacturing, sometimes, the reference form is unknown, and it is time-consuming to find a fitted model. Then, a smoothed version of that surface can be seen as the reference form. Because free-form surfaces are widely represented by triangular mesh, the smoothed free-form surface can be achieved by mesh smoothing. Unlike image processing, surface metrology requires a higher level of accuracy and precision[6].

$$\frac{\partial f(x,t)}{\partial t} = \lambda \Delta f(x,t) \tag{1}$$

Eq.1 is the diffuison equation, where t refers to time, $f(x,t)$ is a continuous function that changes over time, λ is the scalar diffusion coefficient,and Δ represents the laplacian operator. When applied in mesh smoothing, the continuous function $f(x,t)$ is replaced by sample values at the mesh vertices $\begin{bmatrix} f(x_1,t) \\ f(x_2,t) \\ \dots \\ f(x_3,t) \end{bmatrix}$. The function is usually chosen as the vertex postion. Time t is determined by iteration times and sample time. For the sake of concise, in the following part, t will not be written out. The Laplacian operator Δ in the equation also needs to be discretised to apply to meshes.

A discrete Laplacian is defined in Eq.2, where w_{ij} is the weight of vertex x_j regarding to centre vertex x_i and vertex x_j belongs to the its one-ring neighbourhood $\mathcal{N}(x_i)$, and w_i is the sum of weight w_{ij}[7, 8]. There are different functions to compute weight w_{ij} and the discrete strategy will influence the mesh processing afterwards and the surface reconstruction.

$$\Delta f(x_i) = w_i \sum_{v_{j\in\mathcal{N}(x_i)}} w_{ij}\left(f(x_j) - f(x_i)\right) \tag{2}$$

The common types of weight include uniform weight, distance weight and cotangent weight. This paper first introduces these three weight functions. Their algorithms are reviewed and comparisons are made between the performance of Laplacian smoothing using different weights. Based on the evaluation, suggestions are given for different applications.

2 Discretisation methods

In most circumstances, the continuous Laplace or Laplace-Beltrami can be discretised at a mesh vertex x_i by a linear combination of its one-ring neighbours. The weight can be computed in different ways. Three common types of weight functions will be introduced in the following sections： uniform weight, distance weight and cotangent weight. The flowchart below shows the update process for each vertex using three different discretisation methods.

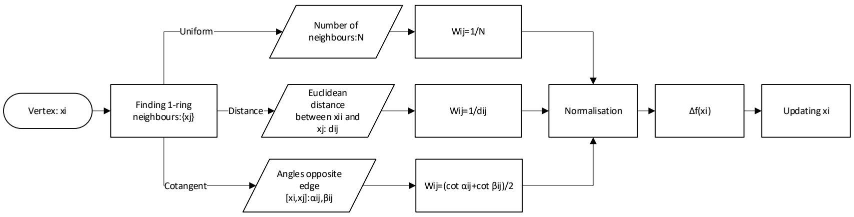


Figure 1 scheme of three discretisation methods for one vertex

2.1 Uniform weight

$$\Delta f(x_i) = w_i \sum_{x_{j \in \mathcal{N}(x_i)}} \frac{1}{N}\left(f(x_j) - f(x_i)\right) \tag{3}$$

In Eq.3, N is the number of one-ring neighbour vertices, x_i represents centre vertex and vertex x_j belongs to the its one-ring neighbourhood $\mathcal{N}(x_i)$. The uniform Laplacian gives equal weight to every one-ring neighbour vertex. The sum of the weight for one vertex is set to 1 for normalisation, so each neighbour's weight is set to $\frac{1}{N}$.

It is simple to compute because it only depends on the number of neighbour vertices, but it is not very appropriate for non-uniform meshes. Because it gives the same weight to all neighbours, which means all neighbours have the same influence on the centre vertex. But actually, some neighbours are closer while some are relatively far.

2.2 Distance weight

$$\Delta f(x_i) = w_i \sum_{x_{j \in \mathcal{N}(x_i)}} \frac{1}{\left\|f(x_j) - f(x_i)\right\|}\left(f(x_j) - f(x_i)\right) \tag{4}$$

In Eq.4, $\left\|f(x_j) - f(x_i)\right\|$ is the distance between the centre vertex x_i and its neighbours x_j. Distance weight using the distance reciprocal between the vertex and its neighbours as the weight.

It overcomes the effect of mesh size or density. It will not cause problems in curve smoothing, but from a three-dimensional perspective, it causes a shift of the centre vertex moving in both tangent plane and normal directions, which causes vertex drifting.

2.3 Cotangent weight

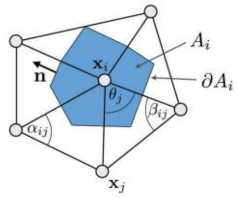


Figure 2: Illustration of parameters in Eq.5 [9]

From discrete differential geometry, $\int_{A_i} \Delta f(u)dA = \int_{\partial A} \nabla f(u)n(u)ds$, where A_i is the local average domain , ∂A is the boundary of A_i and $n(u)$ is the normal vector [10]. Combined with vector operations, it can be simplified to

$$\int_{A_i} \Delta f(u)dA = \frac{1}{2}\sum_{x_{j\in\mathcal{N}(x_i)}} (cot\alpha_{ij} + cot\beta_{ij})\left(f(x_j) - f(x_i)\right) \qquad (5)$$

Thus,

$$\Delta f(x_i) = \frac{1}{2A_i}\sum_{x_{j\in\mathcal{N}(x_i)}}(cot\alpha_{ij} + cot\beta_{ij})\left(f(x_j) - f(x_i)\right) \qquad (6)$$

It considers not only the spatial distance but also geometric information. The problem is that the cotangent value will become negative when the angle is larger than $\pi/2$~~$\frac{\pi}{2}$~~ . Thus, the weight will be negative when $\alpha_{ij} + \beta_{ij} > \pi$ .Usually, the absolute value of the cotangent will be used instead, but it indicates the triangles are flipped.

3 Error Metrics

To get a smoothed version of the mesh surface, the first requirement is smoothness. The smoother the surface, the more surface textures will remain in the residual surface, and the more accurate form evaluation can usually be achieved. But if the smoothed surface is shrinkage, which means the curvature of the reference form is wrong, the residual surface will deviate from the actual data. Therefore, the degree of shrinkage is also an important performance indicator of the results. Last, for edge distortion, normally, it is possible to intercept the middle segment of the surface for analysis. But this is not a good way to handle it when the amount of data is small or the measured object is tiny. Without boundary distortion, less test data will be discarded.

The performances are evaluated from smoothness, shrinkage effect and edge preservation. The edge-preserving performance can be evaluated by comparing visually in three-dimension or by taking a cross-sectional profile comparison. Apart from edge-preserving performance, smoothness and shrinkage comparison is evaluated by different error metrics.

3.1 Metrics for evaluating the smoothness

There are two kinds of error metrics to evaluate smoothness. The first kind is vertex-based, which measures the distance between vertex v_0 and a triangle of the reference mesh M which is closest to vertex v_0. In this report, the simulated original mesh is treated as a reference mesh. The original mesh is then added noise and smoothed by different methods. The vertex distance between the smoothed mesh and the reference mesh reflects how approximate is the estimation (smoothed mesh) to the reference mesh. The root mean square (RMS) of the distance is chosen as the error metric to estimate the imperfection of the fit to the reference data.

The other kind of error metric is normal-based. These metrics measure deviations between the corresponding normals of two meshes M and M_0. Because of this, it is sensitive to mesh degradation at sharp features and highly curved regions[11].

3.2 Metrics for evaluating shrinkage effect

There are two ways to check the shrinkage. One is aligning two meshes and then to compare the differences. In this paper, we will use the software CloudCompare to show the comparison.

For closure surfaces, it can calculate the whole volume before and after mesh smoothing. The shrinkage effect can be qualified by comparing the original mesh volume with the volume of the smoothed mesh.

For a cylinder or sphere-based surface, the least square circle or sphere fitting can be used to fit to the mesh surfaces before and after smoothing. Radius is been calculated and to evaluate the shrinkage effect.

4 Result

Three methods are implemented for every example. A simulated F-lens surface is used to evaluate their performance on non-Euclidean surfaces. A simulated sphere, which radius is unit one, is used to evaluate most on how much they shrink and a simulated block surface is used to test the edge preservation performance. The original simulated surfaces are ideal smooth meshes. They are remeshed in MeshLab to achieve enough vertex points. Then, random noises are added to in the direction of vertex normal. The measurement surface is a golf ball surface, which underlying form is a sphere. The details of these surfaces are shown in Table 1.

The λ in the diffusion equation is set to 0.8. It controls the smooth speed, and it should be set between 0 and 1. The bigger it is, the smoother result will be got with the same iterations numbers. The iteration number is set from 10 to 100, with 10 as the step size. Some examples also are tested for 200 iteration times. Most smooth results will become stable after 100 iterations.

Table 1: Information about tested surfaces

Name	Number of vertices	Number of faces
F-lens surface	2244	4288
Sphere	40962	81920
Block	8771	17550
Golf ball	78721	157438

4.1 F-lens surface

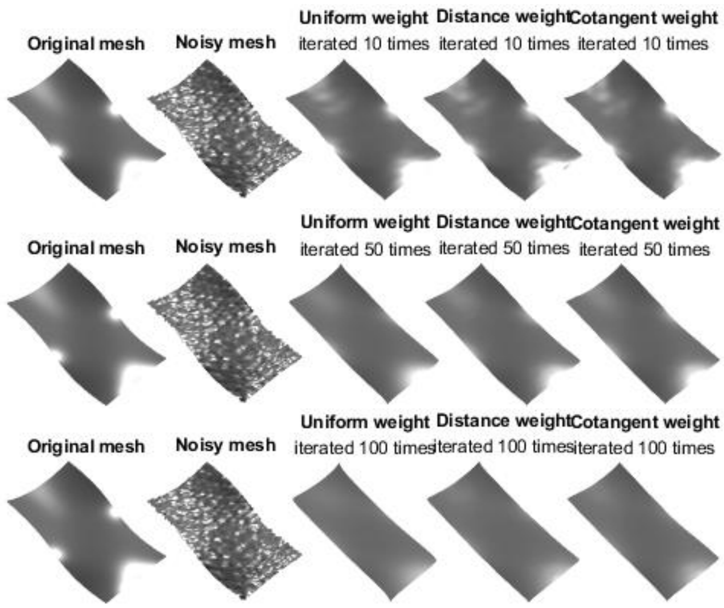
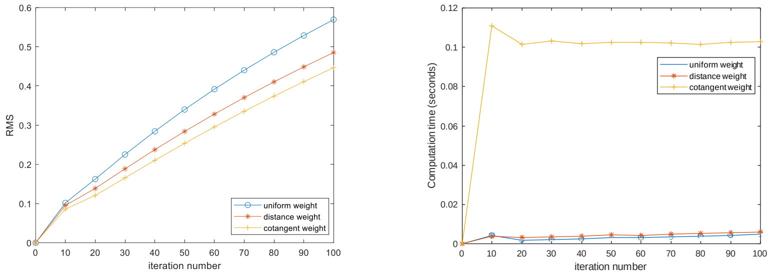


Figure 3 : Smoothing results of F-lens surface using different weight functions

Figure 3 shows the smooth results of the F-lense surface using three different weights and iterated 10 to 100 times with 10 step size. Horizontally, the smoothness looks very close for the three different weights, but the uniform weight result has distortions at the boundary, especially when the iteration number becomes bigger. Vertically, the smoothing effect becomes more and more obvious as the number of iterations increases, but they shrink as well.

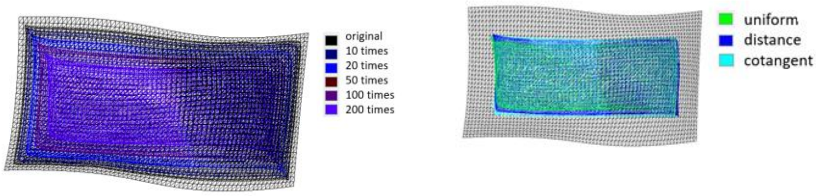
From Figure 4a, we can see cotangent weight results are the best-fitted approximation of the original mesh. Distance weight results are close, and uniform weight results are far. The gap becomes even bigger when the iteration number increases. However, the cotangent weight function needs much more time to compute, almost 20 times more than the other two methods(Figure 4b).

From Figure 5a, we can see the edge distortion and shrinkage effect. The problems become severe as the iteration increases. As can be seen in Figure 5b, all these methods have the shrinkage problem, and cotangent weight has the best performance regarding edge preserving, while uniform weight is the worst.



(a) (b)

Figure 4: (a):Root mean square value of the distance from smoothed free-form surfaces to reference free-form surface;(b): Computation time



(a) (b)

Figure 5: (a):Mesh display of the smoothed free-form meshes usinguniform weight;(b): Mesh display of the smoothed meshes for 200 iterations

4.2 Unit sphere surface

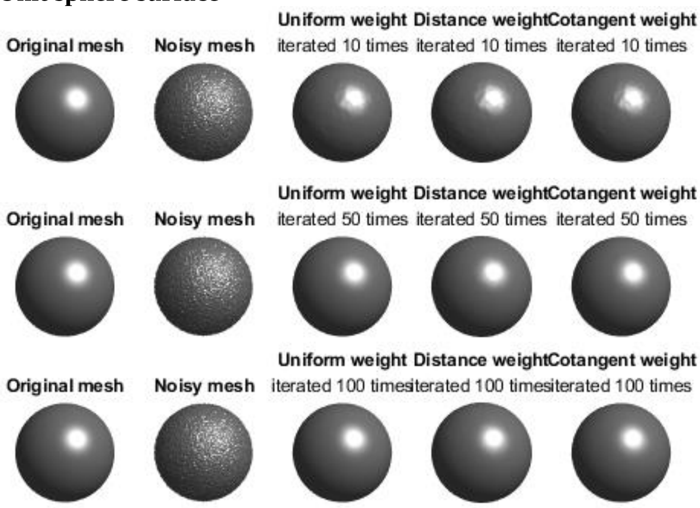
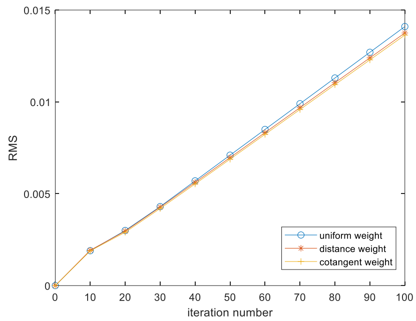


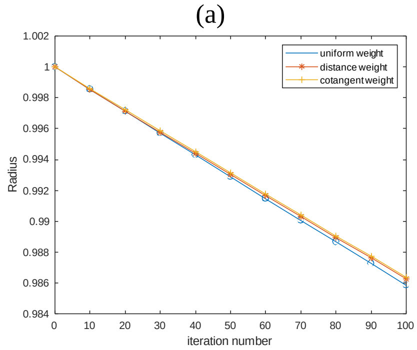
Figure 6 : Smoothed results of a unit sphere

Figure 6 shows the smooth results of the simulated unit sphere surface. They all achieved a smoothed version of the mesh after 50 iterations. From Figure 7a, the RMS values of the distance between the original mesh and smoothed meshes steadily grow. Among these three weight functions, the line shows that cotangent weight is also best for approximating sphere-shape underlying form. But the computation time of the cotangent weight function becomes even higher after 60 iterations, almost 200 times that of the other two methods(Figure 7c).

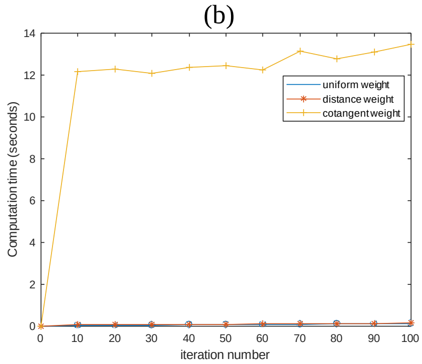
In Figure 7b, the radius of uniform weight results drops when the iteration number increases, which means it shrinkages and becomes more significant as the iteration number increases. The results of the distance weight function and cotangent weight function has similar radius regardless of the iteration number, so as the root mean square values of distance.



(a)



(b)



(c)

Figure 7: (a):Root mean square value of the distance between reference mesh and smoothed meshes;(b) Radius of fitted spheres;(c): Computation time

4.3 Block surface

Figure 8 illustrates the smoothed block surfaces using three weight functions. Three methods can achieve a smooth result after 10 iterations. However, from the second row, we can see that the edges are blurring and tend to shrink to the object centre. This phenomenon is more obvious after 100 iterations. In Figure 9a, the distance weight function line and cotangent weight function line almost overlap, indicating their smoothness performances are very similar. But cotangent weight is better in avoiding the shrinkage effect because the volumes of its results decrease slower than the distance weight in Figure 9b. However, higher accuracy comes at the expense of efficiency. Its computation time is almost 10 times than other two methods.

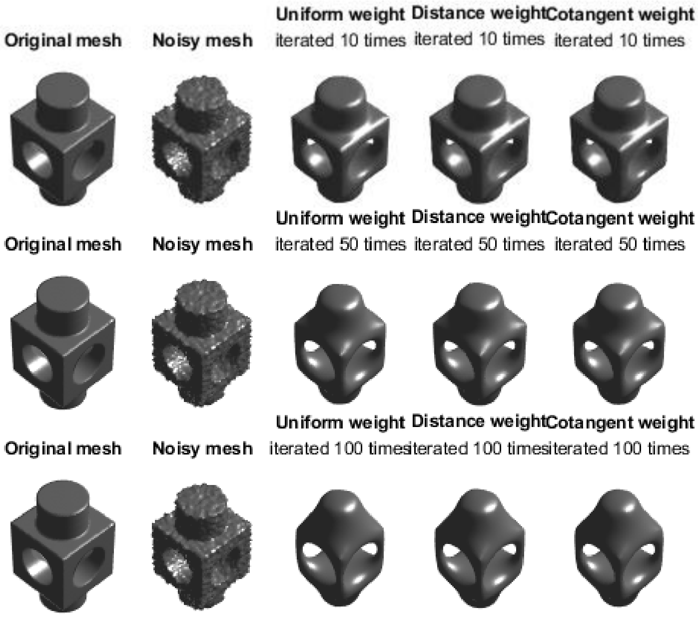


Figure 8 Smoothed results of simulated block

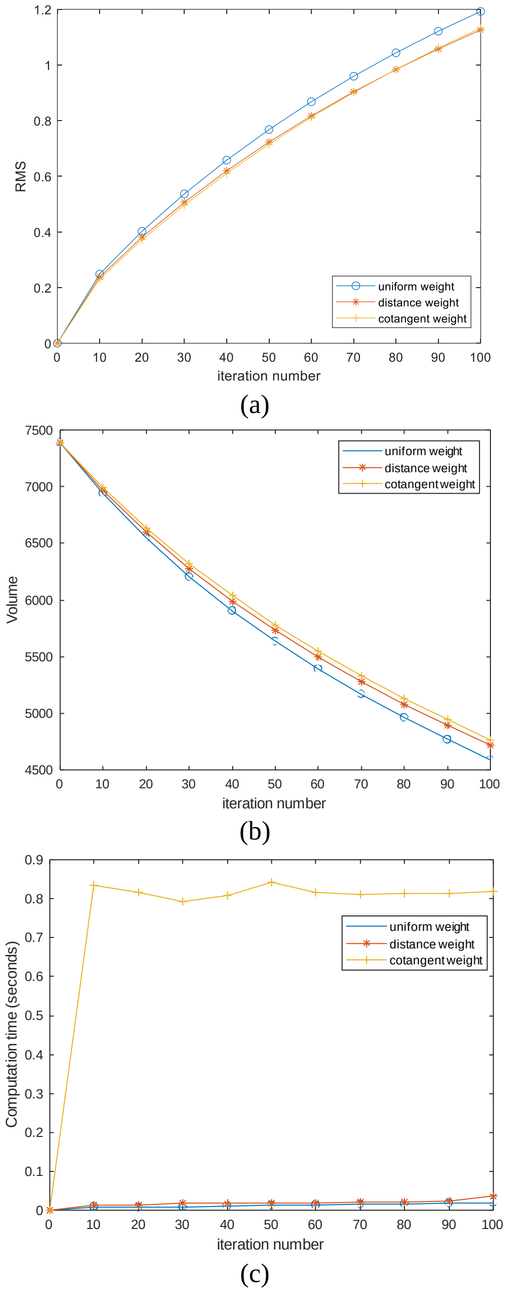


Figure 9: (a):RMS values of distances between smoothed mesh and original mesh;(b): Total volumes of smoothed blocks;(c): Computation time

4.4 Golf ball surface

The golf ball surface is a measured surface so the reference form is unknown in advance. Therefore, we did not calculate the RMS value for it. From Figure 10 we can see that after 100 iterations, the surface of the golf ball has been smoothed to a smooth sphere with three weight functions though the second and third methods still have one or two tinny bumps. We increased the iteration number to 2000 times and these features still exist. The reason needs further investigation.

From Figure 11, the distance weight results have the biggest radius while the uniform weight is the smallest. The distance weight computes as similar speed as uniform weight while the cotangent weight is much slower.

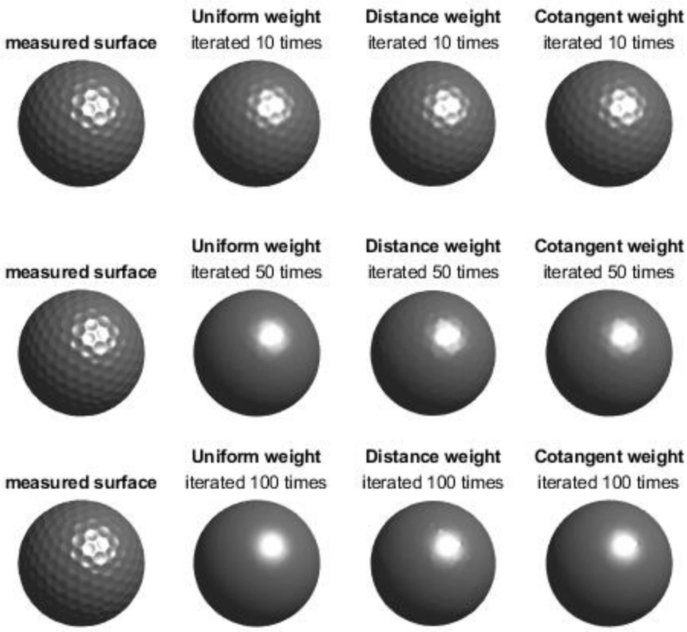
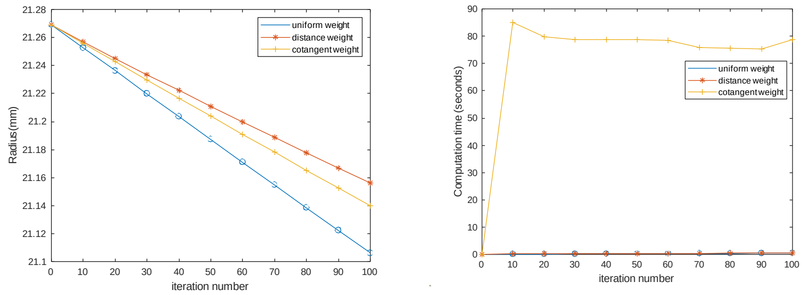


Figure 10: Smoothed results of a golf ball surface



(a) (b)

Figure 11: (a):Radius of the fitted sphere for different weight function smoothing results;(b): Computation time

5 Conclusion and Future Work

We compared the smoothed performance using three different weight functions to discretise the Laplacian-Belatrami operator. The result shows that they all can achieve decent smoothness after 100 iterations, and they both cause edge blurring if the iteration number is more than 50. The uniform weight function computes fast and only needs the number of one-ring neighbours. However, its shrinkage effect is significant. The cotangent weight function has the best performance in most aspects, but it needs much more time to compute. The distance weight function is a good compromise solution between them because its smoothness degree is close to cotangent weight, but its computation time is much shorter, similar to uniform weight. Therefore, the distance weight function is recommended to discretise the Laplacian-Beltrami operator in mesh smoothing. If higher accuracy is needed, use cotangent weight instead and while time is very limited, uniform weight can also be used but the accuracy is not guaranteed.

The future work includes comparing using other weight functions. The test should also be tested on more measurement surfaces. The question that if it is possible to find a weight function that does not blur edges also needs further investigation.

Acknowledgement

The authors gratefully acknowledge the UK's Engineering and Physical Sciences Research Council (EPSRC) funding via the Future Advanced Metrology Hub (EP/P006930/1), EPSRC Fellowship in Manufacturing (EP/R024162/1), and new investigator award (EP/S000453/1) for supporting this work.

References

[1] Jiang, X.J. and D.J. Whitehouse, *Technological shifts in surface metrology*. CIRP Annals, 2012. **61**(2): p. 815-836.

[2] Whitehouse, D.J., *Surface geometry, miniaturisation and metrology*. Philos Trans A Math Phys Eng Sci, 2012. **370**(1973): p. 4042-65.

[3] Bewilogua, K., et al., *Surface technology for automotive engineering.* CIRP Annals, 2009. **58**(2): p. 608-627.

[4] Lawonn, K., et al., *Adaptive and robust curve smoothing on surface meshes.* COMPUTERS & GRAPHICS-UK, 2014. **40**: p. 22-35.

[5] Jiang, X.J. and P.J. Scott, *Chapter 6 - Free-form surface filtering using the diffusion equation*, in *Advanced Metrology*, X.J. Jiang and P.J. Scott, Editors. 2020, Academic Press. p. 129-142.

[6] Hussein S Abdul-Rahman, S.L., Wenhan Zeng, Xiangqian Jiang, Paul J. Scott, *Free-form texture representation and characterisation based on triangular mesh projection techniques.*

[7] Sorkine, O., *Laplacian Mesh Processing*, in *EUROGRAPHICS 2005*. 2005.

[8] Taubin, G. *A signal processing approach to fair surface design*. in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. 1995.

[9] Botsch, M., et al., *Polygon mesh processing*. 2010: CRC press.

[10] Meyer, M., et al., *Visualisation and Mathematics III: Discrete differential geometry operator for triangulated 2-manifolds*. 2002.

[11] Belyaev, A. and Y. Ohtake, *A Comparison of Mesh Smoothing Methods.* 2003.