

Mesh Denoising Using Multi-scale Curvature-based Saliency

Somnath Dutta¹, Sumandeep Banerjee¹, Prabir K. Biswas¹, and Partha Bhowmick²

¹ Dept. of Electronics and Electrical Communication Engineering
{somdats,sumandeep.banerjee}@gmail.com, pkb@ece.iitkgp.ernet.in

² Dept. of Computer Science and Engineering
pb@cse.iitkgp.ernet.in
Indian Institute of Technology Kharagpur, India

Abstract. 3D mesh data acquisition is often afflicted by undesirable measurement noise. Such noise has an aversive impact to further processing and also to human perception, and hence plays a pivotal role in mesh processing. We present here a fast saliency-based algorithm that can reduce the noise while preserving the finer details of the original object. In order to capture the object features at multiple scales, our mesh denoising algorithm estimates the mesh saliency from Gaussian weighted curvatures for vertices at fine and coarse scales. The proposed algorithm finds wide application in digitization of archaeological artifacts, such as statues and sculptures, where it is of paramount importance to capture the 3D surface with all its details as accurately as possible. We have tested the algorithm on several datasets, and the results exhibit its speed and efficiency.

1 Introduction

Mesh denoising is an imperative preprocessing technique for improving meshes containing noise that creeps in during the process of data acquisition and subsequent digitization process. It aims at improving the quality of the reconstructed surface by producing a mesh with better perceptual features. So, it involves removal of noise while retaining most of the original features present in the object, and hence should be robust in nature. An important aspect of a denoising algorithm is to adjust vertex positions without any feature disintegration. The time taken to accomplish the entire process is also one of the important considerations, especially if real-time interactive mesh processing is required. The entire process is iterative, where the number of iterations actually depend on the amount of smoothness required with minimum degradation of the actual content. Low-level human visual perception plays an indispensable role in objective evaluation of geometric processing like denoising. Incorporating perceptuality in mesh denoising analogous to image filtering can improve the processing of the meshes. The requirement of quality graphical mesh and their usage in multidisciplinary applications such as digital heritage, explain the

essence of employing the concept of human perception into mesh processing. Mesh saliency, as introduced in [1], gives a measure of regional importance, especially for 3D mesh. It can be integrated into graphics applications, such as mesh denoising, mesh simplification, shape matching, and segmentation. Saliency captures features of an object at multiple scales, since what seems interesting at one scale may not remain the same at other scales. It consequently reveals the difference between the vertex and its surrounding context. Various mesh processing methods can be modified to accommodate saliency into the process so that visually salient features can be preserved into the mesh.

1.1 Existing Work

The classical Laplacian smoothing method [2, 3] is the simplest surface smoothing method for noise removal. However, it over-smooths the mesh and also causes surface shrinkage. Kernel based Laplacian smoothing method proposed in [4] tries to overcome the problem of classical Laplacian approach. There are various anisotropic filtering approaches which vary on the technique used to preserve prominent features. Some algorithms are geometric diffusion based anisotropic method [5–7]. The other class of algorithms comprise both normal update and vertex update for the purpose of denoising [8–10]. Various methods traditionally used for image denoising have been extended to point cloud denoising as well as mesh denoising. Bilateral filter [11] is one such prominent method that has proved to be an effective edge-preserving filter. The earlier work by Fleishman et al. [12] and Jones et al. [13] focused on modifying the vertex position by a suitable weighted function based on spatial difference as well as normal difference. Yagou et al. proposed mesh denoising based on alpha trimming [14] along with mean, median method [9]. Sun et al. [15] proposed an iterative algorithm of filtering noisy normals and then updating each vertex position based on this modified normal following least square criterion. The recent work of Zheng et al. [16] emphasizes on both local iterative scheme and global non-iterative scheme of mesh denoising. Unlike previous methods of processing normals, it considers normals as a surface signal defined over the original mesh. It also presents comparative analysis of such scheme under different constraints like runtime as well as robustness. Denoising can also be accomplished by considering the point cloud without any mesh representation. The central idea behind such methods is to denoise the points and then triangulate those denoised points to obtain a reconstructed denoised surface. [17], [18] are some of the techniques on point cloud denoising. A different approach based on L_0 minimization is undertaken in [19] for mesh denoising. L_0 norm is used to preserve sharp features and smooth the remaining surface.

The concept of saliency has been studied for images to determine salient image location in [20]. Itti et al. [21] computed a saliency map from the information based on center-surround mechanism. Saliency based methods using the concept of [21] have been used in [22] for computing the saliency map of a 3D dynamic scene. The idea described in [23] has been used to find 3D

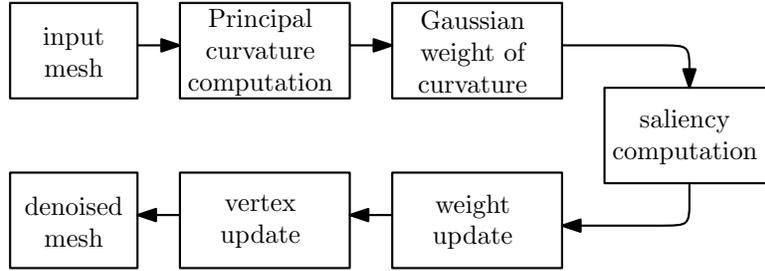


Fig. 1: Proposed denoising process based on saliency.

surface [24] by smoothing noisy data. A user study that compares the previous mesh saliency approach with human eye movements is discussed by Kim et al. [25]. The experimental result discussed in [25] describes the fact that mesh saliency can model human eye movements comparatively better than what can be expected purely by chance. The idea of using only local features or locally prominent salient regions for computing the saliency of a 3D mesh surface is outperformed in [26]. It incorporates the methodology of not only local contrast but even global rarity by defining global saliency on each vertex depending on its contrast with all other vertices.

1.2 Our Work

Our proposed method for mesh denoising is similar to the one followed for denoising in [27] with some modifications. In [27] saliency [1] is combined with contextual discontinuities [10]. In [10], an adaptive smoothing is used to denoise a 2D image, incorporating both inhomogeneity and spatial gradient. Inhomogeneity reveals the incoherence between a pixel and its surrounding pixels. The proposed method also takes mesh saliency into consideration by using curvature as a geometric feature of the object. The saliency value for a vertex is high if it is a salient point and vice versa. The Gaussian weighted average of the principal curvature estimated at each point (vertex) around a neighborhood is considered as a suitable weight function with the weight being amplified depending on certain algorithmic criteria. Curvature of each point v on the surface is computed, and the largest absolute value of the principal curvature is considered for the process, as follows.

$$\kappa(v) = \max(|\kappa_{\max}(v)|, |\kappa_{\min}(v)|) \quad (1)$$

where, $\kappa_{\max}(v)$ and $\kappa_{\min}(v)$ denote peak convex and peak concave curvatures respectively.

2 Proposed Method

Figure 1 presents a block diagram of the proposed method on mesh denoising. Its various stages are briefly explained in this section.

2.1 Curvature Estimation

The method of curvature estimation is based on [28]. It is one of the efficient and widely used techniques for estimating curvature of a 3D dataset. Initially, for all points in the dataset, normals are computed. Then, by principal component analysis (PCA) of these normals, the maximum and the minimum principal curvatures for all data points are obtained.

2.2 Neighborhood Determination and Weight Computation

We consider a distance based threshold as opposed to ring neighborhoods used by [27]. We determine the neighborhood of each vertex (point) of the mesh using *kd*-Tree based ANN algorithm provided by the Point Cloud Library (PCL) [29]. Let the neighborhood $N_2(v, \delta)$ for a vertex v be the set of vertices within a distance δ , measured in L^2 norm. That is, $N_2(v, \delta) = \{x : \|x - v\| < \delta\}$, where x denotes a mesh vertex. The Gaussian-weighted average of the principal curvature is given by

$$G(\kappa(v), \delta) = \frac{\sum_{x \in N_2(v, 2\delta)} \kappa(x) \exp\left(\frac{-\|x - v\|^2}{2\delta^2}\right)}{\sum_{x \in N_2(v, 2\delta)} \exp\left(\frac{-\|x - v\|^2}{2\delta^2}\right)} \quad (2)$$

where, $\kappa(x)$ is the absolute value of principal curvature, and $N_2(v, 2\delta)$ denotes the neighborhood of a vertex v within a distance 2δ . The value δ is the standard deviation of the Gaussian filter. To incorporate the saliency feature, different values of δ are considered to incorporate the idea of multiple resolutions, which in turn, captures the important features of the object at all perceptually meaningful scales.

2.3 Saliency Computation

The mesh saliency that tends to capture the most prominent features at multiple scales is estimated as

$$S_k(v) = |G(\kappa(v), \delta_k) - G(\kappa(v), \delta_{k+1})| \quad (3)$$

where, $S_k(v)$ is the saliency value of a vertex v at a scale δ_k w.r.t. the next scale δ_{k+1} . The difference between the two captures the importance of saliency of a vertex v at two successive scales. The scales used in our method are from

$\{k\varepsilon : 1 \leq k \leq 6\}$, where $\varepsilon = \chi\|\bar{e}\|$ is considered as a multiple of the average edge length $\|\bar{e}\|$ of the mesh

The *coarseness factor* χ multiplied with $\|\bar{e}\|$ to obtain ε is a parameter supplied manually, depending on the object to be denoised, and we have found values in the range 1–4 to be most effective. For a high value of χ , the finer details are lost, whereas a low value of χ can capture the finer details. The saliency of a vertex is finally estimated as the average of its saliency values computed at different scales, in accordance with the following equation.

$$S(v) = \frac{1}{n_s - 1} \sum_{k=1}^{n_s-1} S_k(v) \quad (4)$$

where, n_s is the number of scales used (6 in our experiments).

2.4 Weight Update

The maximum saliency, $S_{\max}(v)$, and the minimum saliency, $S_{\min}(v)$, are computed for all mesh vertices, and then the saliency value of each vertex $S(v)$ is normalized for adaptive smoothing. The normalized saliency value is given by

$$\tilde{S}(v) = \frac{S(v) - S_{\min}(v)}{S_{\max}(v) - S_{\min}(v)}. \quad (5)$$

A high saliency value usually corresponds to a surface feature. In other words, vertices with higher saliency values maintain their sharp features during denoising. Hence, we choose a value β such that all saliency values greater than β are amplified, while the rest do retain the original. We choose β in the saliency interval of 60-80th percentile; in most the cases, β is considered as the 80th percentile of the saliency value, while the amplifying factor λ is selected in the range 4–10. As a result, the normalized saliency obtained in Eqn. 5 is finally modified as per the following equation.

$$\tilde{\tilde{S}}(v) = \begin{cases} \lambda\tilde{S}(v) & \text{if } \tilde{S}(v) \geq \beta \\ \tilde{S}(v) & \text{otherwise} \end{cases} \quad (6)$$

2.5 Vertex Update

To denoise the mesh, we recompute the position of each vertex in the mesh. This step involves causing a vector displacement to each vertex. We begin by considering a neighborhood within a distance δ around the vertex, and determine the centroid of the concerned neighborhood. We then compute the centroid normal C and the point normal P (mapped vector difference between the centroid and the respective point). The mapped normal overcomes the anomaly of using the true point normal as described in paper [17]. The vertex is replaced by the weighted average computed around the neighborhood of that vertex. The weight function is Gaussian in nature with $P_j \cdot C$ as the variable, and the saliency value from Eqn. 6 as the scale.

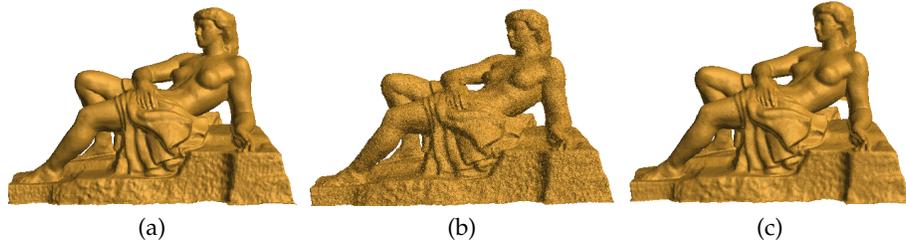


Fig. 2: (a) Meduse original surface (b) Noisy (Gaussian) surface (c) Denoised surface

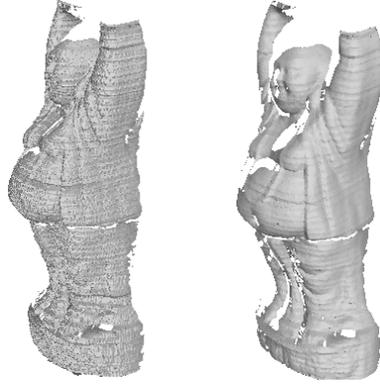


Fig. 3: (a) Buddha laser scanned surface (b) Denoised Buddha surface

3 Results and Discussion

The algorithm is tested on various datasets acquired by our system as well as data available from standard databases [30, 31]. The user-defined parameters λ is used for amplifying the saliency value so that features are preserved during smoothing. In majority of the dataset, only one iteration of denoising is used to ensure that the object does not get over-smoothed and blurry. Figure 2 illustrates the effect of denoising on a surface corrupted by Gaussian noise. The parameters used is $\lambda = 4$. The object shown in Figure 4 is of a column of a temple. The result obtained for this object shows another instance of the effectiveness of the algorithm in denoising an object while preserving its features. The user-defined parameters in this case is $\lambda = 5$. The Buddha dataset obtained using our own scanner without any synthetically added noise also responds effectively to denoising process, as shown in Figure 3. The parameters used for denoising in this case are the same used for the result shown in Figure 2.

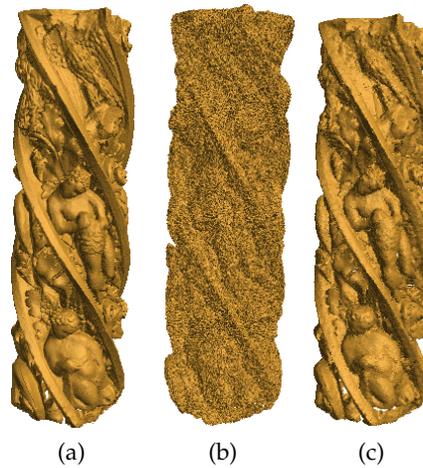


Fig. 4: (a) Column surface (b) Noisy (Gaussian) surface (c) Denoised surface

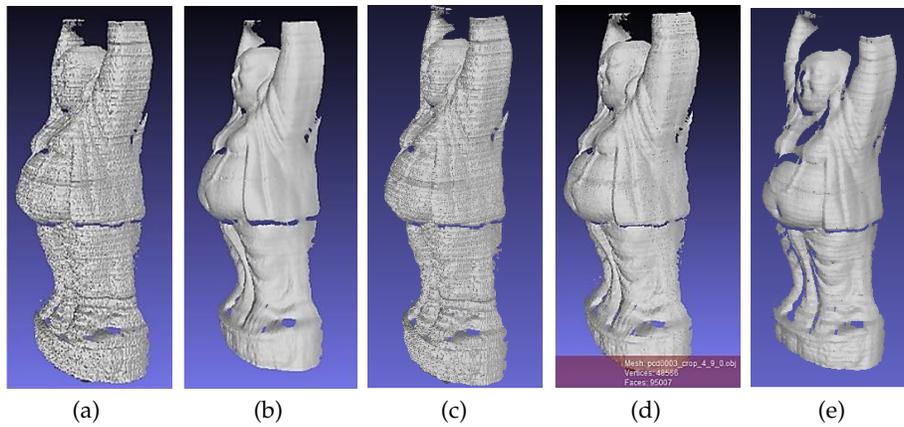


Fig. 5: (a) Noisy surface (b) Laplacian (c) Fleishman bilateral Filter (d) Normal filtering (e) Denoised (Saliency-based) surface

3.1 Comparative Analysis

The quantitative evaluation as well as the objective evaluation based on perceptual metric can be used for comparing the results, albeit the focus is on quantitatively evaluating the outputs of different denoising method. The quantitative evaluation is based on root mean square (RMS) error method. It takes into consideration the correspondence among the vertices of the two objects under comparison, and hence it is limited to the comparison between two meshes sharing the same connectivity. The quantitative error is estimated as

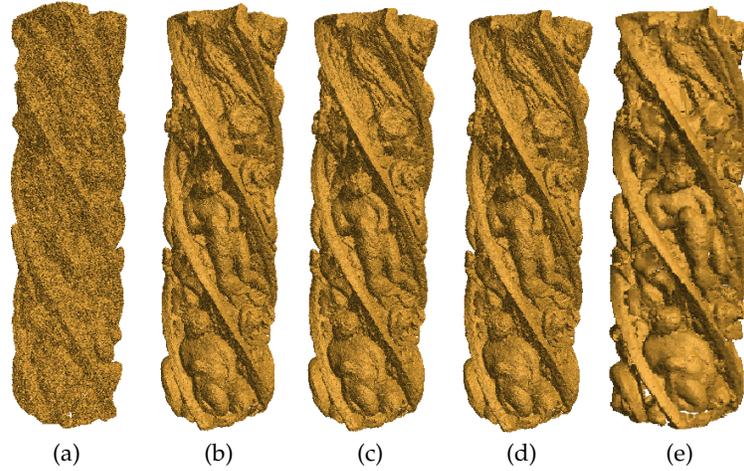


Fig. 6: Various denoised (Column dataset) output (a) Noisy surface (b) Laplacian surface (c) Bilateral filtered surface (d) Normal filtered surface and (e) Saliency surface (proposed method)

$$Error(A, B) = \frac{1}{n} \sum_{i=1}^n \|v_i^A - v_i^B\|^{1/2} \quad (7)$$

where, n is the number of vertices of the mesh, v_i^B is the vertex of denoised mesh B , and v_i^A is its corresponding vertex in the original mesh A .

A comparison of the proposed method with some of the existing mesh-denoising algorithms is presented in Figure 5. The parameters of the existing methods have been tuned to generate the best possible outputs. The Laplacian method, as shown in Figure 5(b), almost smooths the surface, hence degrading its features. Fleishman bilateral filter extends the concept of 2D bilateral filter to 3D mesh denoising. The result shown in Figure 5(c) is able to preserve details to some extent but takes considerable amount of time to denoise the mesh. The Normal filtering technique (Sun et al.) requires a large number of iterations initially for the task of normal update and also for vertex update, hence increasing the runtime for denoising. However, the details of the object remain intact to a large extent. The output shown in Figure 5(d) required ten normal iterations and fifteen vertex iterations to complete the denoising process. The result in Figure 6 shows the comparison on Column data with some existing methods. Table 1 presents a comparison of the proposed method with some of the existing algorithms in terms of quantitative error and also with respect to the execution time of the denoising process.

Table 1: Performance analysis

Algorithms	Data (vertices)	Error (E_o)	Time (in secs)
Laplacian smoothing	Column (480932)	2.28×10^1	18
	Meduse (358904)	6.66×10^{-3}	14
Bilateral filter (Fleishman et al.)	Column (480932)	2.26×10^1	13752
	Meduse (358904)	6.68×10^{-3}	11356
Normal filtering (Sun et al.)	Column (480932)	2.24×10^1	385
	Meduse (358904)	6.51×10^{-3}	215
Saliency based denoising	Column (480932)	2.21×10^1	261
	Meduse (358904)	6.32×10^{-3}	203

4 Conclusion

The proposed technique based on mesh saliency is suitable for denoising due to its simplicity, execution speed, and ability to retain the originality of the object while removing the noise. It comes up with a significantly low error rate on different datasets, although its execution is quite fast compared to some of the existing methods. Its fast execution time merits its readiness to larger datasets with archaeological artifacts of historical importance, such as statues, sculptures, monuments, and temple structures.

References

1. Lee, C.H., Varshney, A., Jacobs, D.W.: Mesh saliency. *ACM Trans. Graph.* (2005) 659–666
2. Field, D.A.: Laplacian smoothing and delaunay triangulations. *Communications in Applied Numerical Methods* (1988) 709–712
3. Vollmer, J., Mencl, R., Muller, H.: Improved laplacian smoothing of noisy surface meshes. In: *Computer Graphics Forum*. (1999) 131–138
4. Badri, H., El Hassouni, M., Aboutajdine, D.: Kernel-based laplacian smoothing method for 3d mesh denoising. In Elmoataz, A., Mammass, D., Lezoray, O., Nouboud, F., Aboutajdine, D., eds.: *Image and Signal Processing. Volume 7340 of Lecture Notes in Computer Science*. Springer Berlin Heidelberg (2012) 77–84
5. Desbrun, M., Meyer, M., Schrder, P., Barr, A.H.: Anisotropic feature-preserving denoising of height fields and bivariate data. In: *Graphics Interface*. (2000) 145–152
6. Tasdizen, T., Whitaker, R., Burchard, P., Osher, S.: Geometric surface smoothing via anisotropic diffusion of normals. In: *Proceedings of the Conference on Visualization '02. VIS '02, Washington, DC, USA, IEEE Computer Society* (2002) 125–132
7. Hildebrandt, K., Polthier, K.: Anisotropic filtering of non-linear surface features. *Computer Graphics Forum* (2004) 391–400
8. G.Taubin: Linear anisotropic mesh filtering. *IBM Research Report RC22213(W0110-051)* (2001)

9. Yagou, H., Ohtake, Y., Belyaev, A.: Mesh smoothing via mean and median filtering applied to face normals. In: Geometric Modeling and Processing, 2002. Proceedings. (2002) 124–131
10. Chen, K.: Adaptive smoothing via contextual and local discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2005) 1552–1567
11. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Sixth International Conference on Computer vision. (1998) 839–846
12. Fleishman, S., Drori, I., Cohen-or, D.: Bilateral mesh denoising. *ACM Transactions on Graphics* (2003) 950–953
13. Jones, T., Durand, F., Desbrun, M.: Non-iterative, feature preserving mesh smoothing. *ACM Transactions on Graphics* (2003) 943–949
14. Yagou, H., Ohtake, Y., Belyaev, A.: Mesh denoising via iterative alpha-trimming and nonlinear diffusion of normals with automatic thresholding. In: Computer Graphics International, 2003. Proceedings. (2003) 28–33
15. Sun, X., Rosin, P., Martin, R., Langbein, F.: Fast and effective feature-preserving mesh denoising. *Visualization and Computer Graphics, IEEE Transactions on* (2007) 925–938
16. Zheng, Y., Fu, H., Au, O.C., Tai, C.L.: Bilateral normal filtering for mesh denoising. *Visualization and Computer Graphics, IEEE Transactions on* (2011)
17. Miropolsky, A., Fischer, A.: Reconstruction with 3d geometric bilateral filter. In: Proceedings of the Ninth ACM Symposium on Solid Modeling and Applications. SM '04, Aire-la-Ville, Switzerland, Switzerland, Eurographics Association (2004) 225–229
18. Deschaud, J.E., Goulette, F.: Point cloud non local denoising using local surface descriptor similarity. *ISPRS Technical Commission III Symposium, PCV 2010 - Photogrammetric Computer Vision and Image Analysis XXXVIII - Part 3A*, (2010)
19. He, L., Schaefer, S.: Mesh denoising via l0 minimization. *ACM Trans. Graph.* (2013) 64:1–64:8
20. Koch, C., Ullman, S.: Shifts in selective visual attention: Towards the underlying neural circuitry. In Vaina, L., ed.: *Matters of Intelligence*. Synthese Library. Springer Netherlands (1987) 115–141
21. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* (1998) 1254–1259
22. Yee, H., Pattanaik, S., Greenberg, D.P.: Spatiotemporal sensitivity and visual attention for efficient rendering of dynamic environments. *ACM Trans. Graph.* (2001) 39–65
23. Guy, G., Medioni, G.: Inferring global perpetual contours from local features. *International Journal of Computer Vision* (1996) 113–133
24. Guy, G., Medioni, G.: Inference of surfaces, 3d curves, and junctions from sparse, noisy, 3d data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* (1997) 1265–1277
25. Kim, Y., Varshney, A., Jacobs, D.W., Guimbretière, F.: Mesh saliency and human eye fixations. *ACM Trans. Appl. Percept.* (2010) 12:1–12:13
26. Wu, J., Shen, X., Zhu, W., Liu, L.: Mesh saliency with global rarity. *Graph. Models* (2013) 255–264
27. Mao, Z.h., Ma, L.z., Zhao, M.x., Li, Z.: Feature-preserving mesh denoising based on contextual discontinuities. *Journal of Zhejiang University SCIENCE A* (2006) 1603–1608
28. Taubin, G.: Estimating the tensor of curvature of a surface from a polyhedral approximation. In: *Computer Vision, 1995. Proceedings., Fifth International Conference on.* (1995) 902–907

29. Rusu, R., Cousins, S.: 3d is here: Point cloud library (pcl). In: Robotics and Automation (ICRA), 2011 IEEE International Conference on. (2011) 1–4
30. Online: Epfl, computer graphics and geometry laboratory. (<http://lgg.epfl.ch/statues.php?p=dataset/>)
31. Online: Aim@shape repository. (<http://shapes.aim-at-shape.net/>)