



Fairing Algorithm of T-spline Surface Based on the Improved Sigmoid Function

Ling Li, Aizeng Wang, Weixiao Tan, Yazui Liu and Gang Zhao

School of Mechanical Engineering & Automation, Beihang University, Beijing 100191, PR China

Corresponding author: Aizeng Wang, azwang@buaa.edu.cn

Abstract. T-spline surfaces play important role in the integration of CAD and CAE, the quality of the surface affects not only the design of applications but also the analysis in simulations. So fairing is a necessary connection after designing and before analysis inherently. The Sigmoid function is usually adopted in a neural network as an activation function and possess continuous and smooth property, we introduce it into surface fairness taking for the weight of the fairing function, and the parameter of the improved Sigmoid function is determined by considering space angle and fitting previous approach. Several experiments are exhibited to indicate the validity of the proposed method, and it reaches a competitive result while less procedure and more stability compared with the existing algorithm.

Keywords: T-splines; Sigmoid function; Surface fairing

DOI: <https://doi.org/10.14733/cadaps.2023.456-465>

1 INTRODUCTION

Surface fairing is of substantial significance in computer-aided design (CAD), which can improve an aesthetic sense of designing and even reduce the machining difficulty in manufacturing. At present, non-uniform rational B-spline (NURBS) curves and surfaces are the primary geometric modeling approaches. Compared with a traditional NURBS surface, the T-spline surface reduces the number of control points significantly, especially for the refinement algorithm which is an important step in the analysis. What's more, because of the existence of the T-junction [14], T-spline simplifies the modeling representation increasing flexibility in surface modeling. Based on these advantages, we focus on the exploration of T-splines.

Surface smoothing is an essential link between modeling and analysis, and researchers have demonstrated that surfaces with fairness physical/computation domain will lead to more accurate and reliable analysis results [10, 11, 21, 22]. To achieve the integration of model design and analysis, it's necessary to take further exploration on surface fairing [2, 4, 5, 12].

Many scholars have worked on faring the physical domain, the main methods can be classified into three types [15, 19, 25]. The first refine the splines, local domain parameterization is constructed by T-splines, PHT splines, and THB splines. The second is the harmonic mapping method, but this method fails to guarantee the bijection of the mapping. The last method is based on

nonlinear optimization, which generally minimizes some distortion while imposing constraints on the bijection of the parameterization. And the weakness of nonlinear optimization is that it is computationally expensive to solve nonlinear optimization problems.

There exist some fairing approaches in less computational complexity on discrete surface representation, such as polygonal mesh, point cloud, and so on. The thought of piecewise function is used by Sun et al. [16] in 3D mesh denoising by setting high weights to similar neighboring normals and zero weights to quite different neighboring normals to achieve remarkable denoising results. Laplacian coordinate [1, 17, 20, 24] is applied to mesh smoothing because the cotangent form possesses the property of restricting triangle faces to be equilateral as much as possible, hence it is a crucial strategy in mesh denoising to avoid the trilateral degeneration. Bilateral filtering [6, 9, 23] considering the spatial difference and signal difference is extensively generalized in mesh denoising as well as image smoothing.

To take advantage of the nonlinear optimization method and reduce the computational time, we propose a fairing algorithm for T-spline surfaces based on the well-known Sigmoid function [3, 7, 8] in this paper. Sigmoid function has widely applied to neural networks, which are used as activation functions to enable neural networks suitable for the nonlinear problem. Also, the Sigmoid function is continuous everywhere and convenient for derivation, this may provide smoothing weight transition in T-surface fairness without oscillation. What's more, the range of Sigmoid function values distributes on special values, we can compress it to desirable intervals by adjusting parameters to meet our requirement.

Inspired by the above, we generalize the improved Sigmoid function as a weighting coefficient in control points movement expression to the T-spline surfaces fairing problem. The parameter in the improved Sigmoid function is obtained by fitting another piecewise constant function proposed by Wang et al. [18]. This reference developed piecewise constant function had accomplished T-spline smoothing managing local abruptness without iteration and global fairness with several iterations. Our method achieves a better result compared with the referenced algorithm on the same model, but with less programming by omitting judgment of space angle. Gaussian curvature of the T-spline surface is used to measure the fairness of the surface. We also show several examples to verify the effectiveness of the proposed algorithm.

The construction of this paper is organized as flowing: In section 2, several basic knowledge is introduced on the T-spline surface composed of parameter space and blending functions, meanwhile, the Sigmoid function is also included. We detail the proposed method in section 3 concentrating on the way defining parameter of the Sigmoid function, the movement manner of control points, and the constraint of smoothing. Several experiments are outlined in chapter 4 to show the smoothed results and compare them with the previous method.

2 T-SPLINE SURFACES

T-spline surface is defined on a T-mesh, given the T-mesh and corresponding control points, the T-spline surface is expressed by:

$$\mathbf{S}(u, v) = \sum_{i=1}^n \mathbf{P}_i B_i(u, v) \quad (2.1)$$

where $\mathbf{P}_i = (w_i x_i, w_i y_i, w_i z_i, w_i)$ is homogenous coordinate, when the control point $\tilde{\mathbf{P}}_i = (x_i, y_i, z_i)$ locates in Cartesian coordinate system, the surface is denoted as follows:

$$\tilde{\mathbf{S}}(u, v) = \frac{\sum_{i=1}^n \tilde{\mathbf{P}}_i w_i B_i(u, v)}{\sum_{j=1}^n w_j B_j(u, v)} \quad (2.2)$$

The $B_i(u, v)$ appeared in expression (2.1)-(2.2) are the blending functions of T-splines, they are calculated by:

$$B_i(u, v) = N[\mathbf{u}_i](u)N[\mathbf{v}_i](v) \quad (2.3)$$

where $N[\mathbf{u}_i](u)$ and $N[\mathbf{v}_i](v)$ are B-Spline basis functions defined on knot vectors \mathbf{u}_i and \mathbf{v}_i . Fig. 1 gives a simple T-mesh and its corresponding surface based on bi-cubic T-splines, the definition of knot vectors \mathbf{u}_i and \mathbf{v}_i follow the ray-intersection method [13].

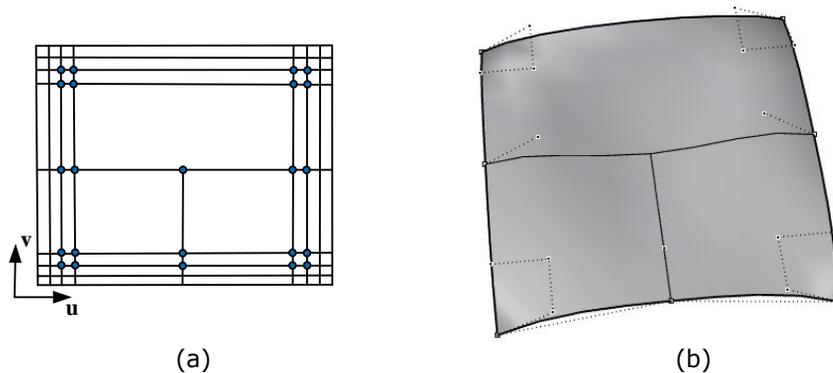


Figure 1: The visualization of a bi-cubic T-Spline. (a) T-mesh, (b) T-spline surface.

3 FAIRING ALGORITHM

3.1 The 1-ring Neighboring Space Angle

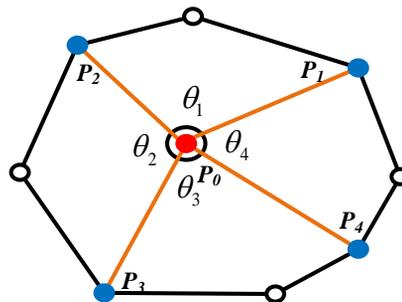


Figure 2: The 1-ring neighboring control point, edge, and corresponding space angle of the control point p_0 in physical space.

When the reign of a T-spline surface is flat, the control points distribute on a plane, but the abruptness occurs if this reign contains sharp features. We use the space angle to measure the curvature of surfaces. As visualized in Fig. 2, the 1-ring neighboring control point of p_0 is consisted by a group of blue points that share uniform yellow edges with p_0 , i.e., $\{p_1, p_2, p_3, p_4\}$. The yellow edges

represented as $\{\overline{p_0p_1}, \overline{p_0p_2}, \overline{p_0p_3}, \overline{p_0p_4}\}$ is called the 1-ring neighboring edge. And the 1-ring neighboring space angle θ is defined as following:

$$\theta = \sum_{i=1}^4 \theta_i \quad (3.1)$$

3.2 The improved Sigmoid Function

The Sigmoid function possesses many excellent properties, such as nonlinearity, monotonicity, and differentiability. This makes it widely generalized in neural networks. Considering the continuous and limited output range, we adopt it as the weight of movement function. The common Sigmoid form is expressed in (3.2) and the curve is plotted in Fig. 3a.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (3.2)$$

The previous smoothing method, which utilizes the piecewise function to assign the weight in smoothing equation according to the space angle, has been used by [18] and achieves remarkable smoothing results. This algorithm gives higher weight to similar control points, while lower weight to quite different control points. As shown in Fig. 3b and the blue line in Fig. 3c, the weight of the piecewise function mutates when space angles approach $\frac{\pi}{2}, \frac{3}{4}\pi, \pi$, and $\frac{3}{2}\pi$, that may lead to irregular oscillation as smoothing iterates. To overcome the disadvantage and acquire stable smoothing results, we developed the improved Sigmoid function as the weight. The parameter of the improved Sigmoid function is determined by fitting the piecewise function, we get:

$$w = \frac{1}{1 + 9e^{-\theta}} \quad (0 < \theta \leq 2\pi) \quad (3.3)$$

This expression accomplishes continuous fitting and decreases the programming process by omitting the judgment of space angle. The smooth translation of the weighting function may help to avoid the oscillation when the fairness continues.

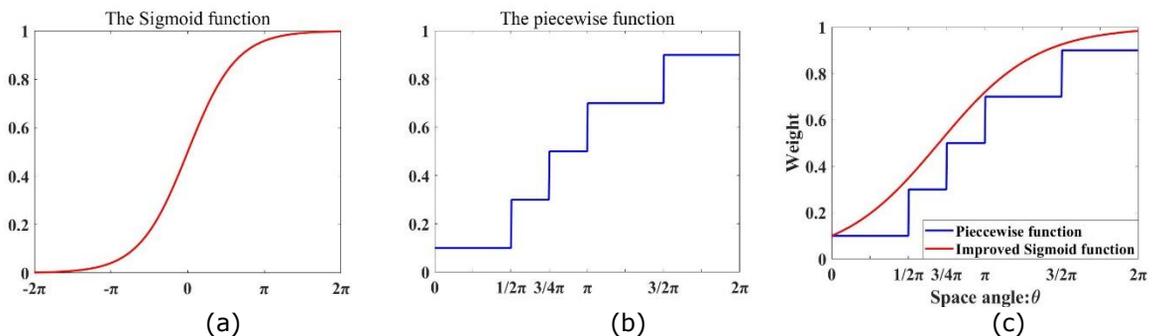


Figure 3: (a) the Sigmoid function; (b) the piecewise function; (c) the fitting result of the Sigmoid function.

3.3 Fairing Method

We adapt the control point movement method to smooth the T-spline surfaces. As demonstrated in expression (3.4), the control points in k iterations are calculated from the weighted average of the control points and their 1-ring neighboring control point in $k-1$ iterations.

$$p_i^k = wp_i^{k-1} + (1-w)\frac{1}{n}\sum_{j=1}^n p_j^{k-1} \tag{3.4}$$

where p_i represents the current control point, $\{p_j\}_{j=1}^n$ is the 1-ring neighboring control point of p_i , w is determined by the improved Sigmoid function introduced in Section 3.2, and k , which is used as a superscript, indicates the number of iterations.

Significant features should be maintained through smoothing, as in reference, we limit the movement of the control point if the distance between k and $k-1$ iterations is larger than the average length of its 1-ring neighboring edge. Hence, the movement constraint is expressed as following:

$$\|p_i^k - p_i^{k-1}\|_2 \leq \frac{1}{n}\sum_{j=1}^n \|p_j^{k-1} - p_i^{k-1}\|_2 \tag{3.5}$$

where the $\|p_i^k - p_i^{k-1}\|_2$ is called an offset.

3.4 Fairing Error

The fairing error is measured by the average Euclidean distance (AED) of the control points between k iterations and the origin model. Since the error is measured by a distance, the value is normal even greater than 1. The representation is in (3.6), wherein N is the number of control points, p_i^k indicates the control point i in k iterations, and p_i^0 expresses control point i in the origin model.

$$AED = \frac{\sum_{i=1}^N \|p_i^k - p_i^0\|_2}{N} \tag{3.6}$$

3.5 Flowchart of the Algorithm

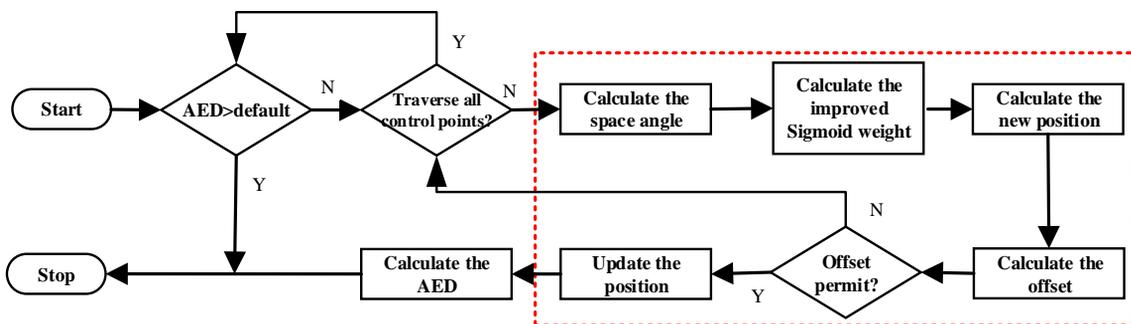


Figure 4: The flowchart of our fairing algorithm.

The proposed T-Spline surface fairing method is accomplished by traversing all the control points on the surface, and the iteration is adaptive to acquire a desirable fairing result. When the presetting AED is less than the default value, this algorithm will iterate automatically.

The whole flowchart of the proposed algorithm is shown in Fig.4, the procedure contains two parts. The left part controls the loop of the program through AED and through the traverse of all

control points as shown above. The right part, which is circled by red dot lines, is the core structure of the fairness method as illustrated in sections 3.1-3.4. It is not necessary to discuss the size of the space angle when calculating the new position of control points, because the weight is assured by an improved Sigmoid function instead of a piecewise function.

4 EXPERIMENTS AND DISCUSSION

In this section, two examples have been shown on complex T-spline models to illustrate the effectiveness of the proposed fairing algorithm. The first experiment shows the result of our method iterating several times by the given AED and the second example is compared with another algorithm [18] in which the weight is determined by a piecewise function. From the results, we can see that fairness is improved by our approach.

Figure 5 shows the smoothing results of the StonesRing model whose detailed information is a list in table 1. Higher curvature changes occur near the torus, as exhibited by zooming in the area of figure 5(b), and these bumps are faired smoothly in the right of figure 5(d). Benefitting from the fairing method, our method maintain significant features of the model and achieves smoothing result.

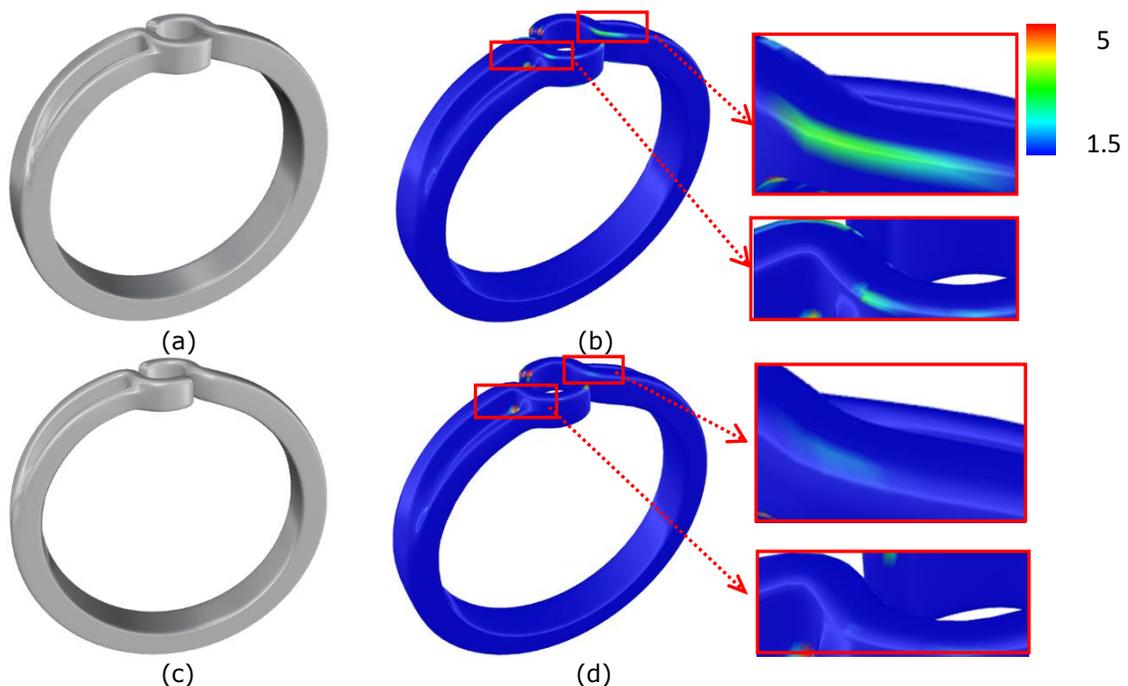


Figure 5: The first row (a-b) displays details of the origin StonesRing model, the second row (c-d) is the reflection of smoothed model. These three columns from left to right describe the rendering, Gaussian curvature, and local enlarged area of Gaussian curvature.

A further demonstration is exhibited in Figure 6, the SunFace model contains substantial details and the component information appears in Table 2. We compare our method with the existing approach developed in reference [18] by setting the same fairing error. The default AED of 0.1 is given in figure 6. (c-d) and (e-f), corresponding to the literature's method and our method respectively. Although our method conducts more times, which is 4 under algorithm [18] and 20 exerting the proposed method, our method achieves a better smooth result in the comparison.

Model name	Model information				iteration	AED	TIME(s)
	face	edge	vertex	Control point			
StonesRing	2208	1106	556	566	15	0.08	15.34

Table 1: The StonesRing model information and performance data.

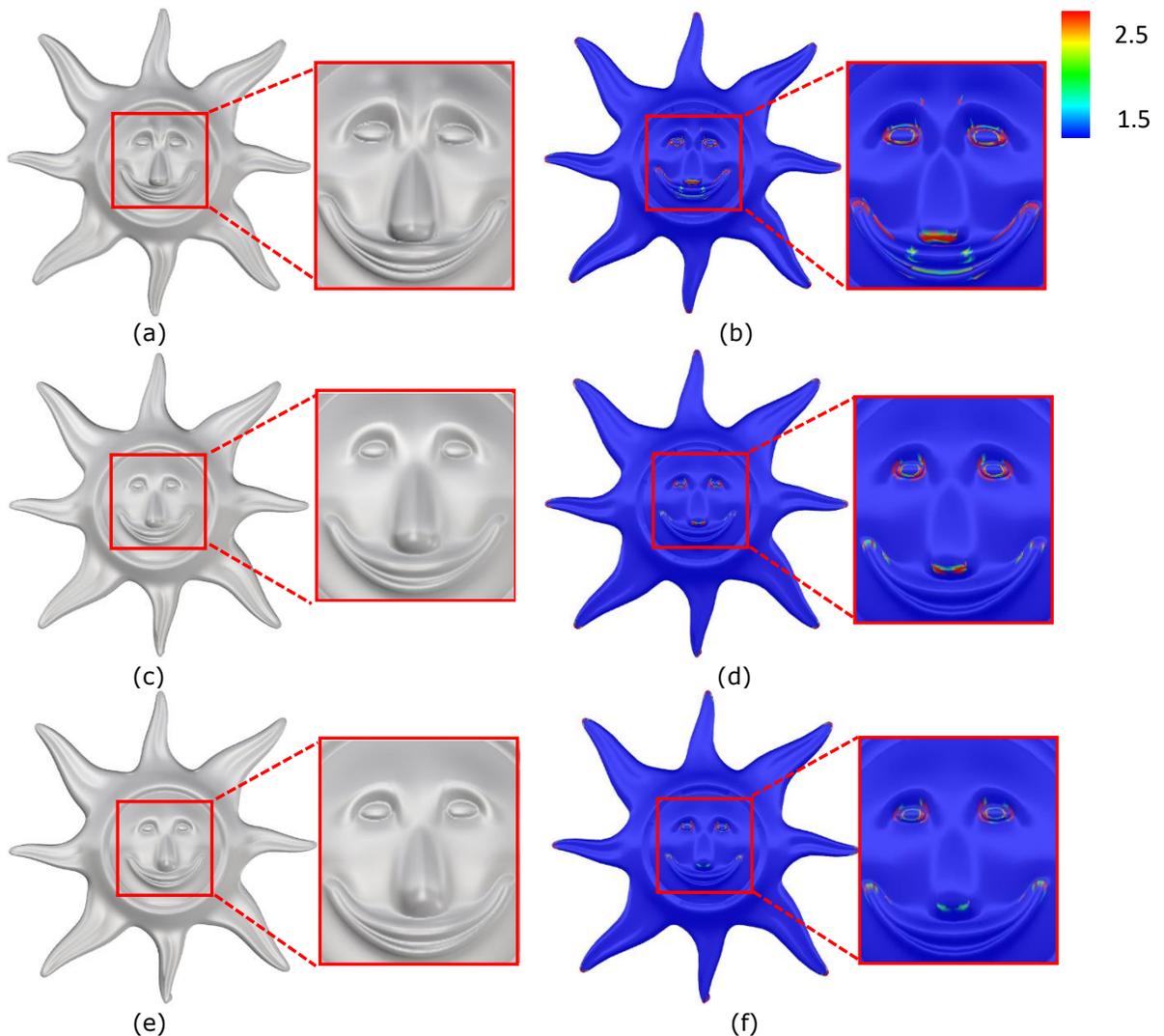


Figure 6: A compared experiment is executed between our method and algorithm [18]. The first row is the rendering and Gaussian curvature of the Sunface model, the second row (c-d) shows the denoised results of the literature [18] by presetting AED 0.1, and the third row (e-f) is our method on the same default errors.

As shown in Figure 6, the fairing effects are obvious in both methods especially the parts at the eyes,

nose, and mouth. The depressions with abrupt high Gaussian curvature are repaired above the two eyeholes, but our method reaches significant fairness on the nostril and at the corners of the mouth. Under the improved Sigmoid weight, our method eliminates the vibration problem caused by piecewise function smoothing and performs robustness even iteration increases.

<i>SunFace</i> <i>model</i>	<i>Model information</i>				<i>iteration</i>	<i>AED</i>	<i>TIME(s)</i>
	<i>face</i>	<i>edge</i>	<i>vertex</i>	<i>Control point</i>			
Algorithm [18]	1024	2052	1060	1091	4	0.10	11.25
<i>Our</i> <i>algorithm</i>					20		49.25

Table 2: The SunFace model information and performance data.

5 CONCLUSION AND FUTURE WORK

We developed a T-spline fairness method based on the weighted movement of control points, the weight is determined by an improved Sigmoid function. The proposed Sigmoid weight is calculated with the 1-ring space angle without a judging process that reduces the programming procedure, and this continuous Sigmoid function enhances the stability of fairness work without oscillation as iteration increases.

Although several references have demonstrated that the fairness of models affects the simulation results in IGA, the comparison has not been done on complex geometry, the relative exploring is inspired in the future. Another aspect of smoothing is essential at the interface of two jointing surfaces. Both the parameter space and physical space are needed to be smoothed.

ACKNOWLEDGEMENTS

REFERENCES

- [1] Casaca, W.; Gois, J. P.; Batagelo, H. C.; Taubin, G.; Nonato, L. G.: Laplacian coordinates: Theory and methods for seeded image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(8), 2020, 2665-2681. <https://doi.org/10.1109/TPAMI.2020.2974475>.
- [2] Chan, C.L.; Anitescu, C.; Rabczuk, T.: Volumetric parametrization from a level set boundary representation with pht-splines, *Computer-Aided Design*, 82, 2017, 29-41. <https://doi.org/10.1016/j.cad.2016.08.008>.
- [3] Crnjanski, J.; Krstić, M.; Totović, A.; Pleros, N.; Gvozdić, D.: Adaptive sigmoid-like and PReLU activation functions for all-optical perceptron, *Optics Letters*, 46(9), 2021, 2003-2006. <https://doi.org/10.1364/OL.422930>.
- [4] Escobar, J.; Cascón, J.; Rodríguez, E.; Montenegro, R.: A new approach to solid modeling with trivariate t-splines based on mesh optimization, *Computer Methods in Applied Mechanics and Engineering*, 200(45), 2011, 3210-22. <https://doi.org/10.1016/j.cma.2011.07.004>.
- [5] Falini, A.; Speh, J.; Jüttler, B.: Planar domain parameterization with thb-splines, *Computer Aided Geometric Design*, 35, 2015, 95-108. <https://doi.org/10.1016/j.cagd.2015.03.014>.
- [6] Guo, M.; Song, Z.; Han, C.; Zhong, S.; Lv, R.; Liu, Z.: Mesh denoising via adaptive consistent neighborhood, *Sensors*, 21(2), 2021, 412. <https://doi.org/10.3390/s21020412>.
- [7] Han, J.; Moraga, C.: The influence of the sigmoid function parameters on the speed of backpropagation learning//International workshop on artificial neural networks. Springer, Berlin, Heidelberg, 1995, 195-201. <https://doi.org/10.1007/3-540-59497-3175>.

- [8] Langer, S.: Approximating smooth functions by deep neural networks with sigmoid activation function, *Journal of Multivariate Analysis*, 2021, 182, 104696. <https://doi.org/10.1016/j.jmva.2020.104696>.
- [9] Liu, S.; Rho, S.; Wang, R.; Jiang, F.: Feature-preserving mesh denoising based on guided normal filtering. *Multimedia Tools and Applications*, 77(17), 2018, 23009-23021. <https://doi.org/10.1007/s11042-018-5735-9>.
- [10] Martin, T.; Cohen, E.; Kirby, R.: Volumetric parameterization and trivariate b-spline fitting using harmonic functions. *Computer Aided Geometric Design*, 26(6), 2009, 648-64. <https://doi.org/10.1016/j.cagd.2008.09.008>.
- [11] Nguyen, T.; Jüttler, B.: Parameterization of contractible domains using sequences of harmonic maps, In: *International conference on curves and surfaces*, 2010, 501-14. <https://doi.org/10.1007/978-3-642-27413-832>.
- [12] Nian, X.; Chen, F.: Planar domain parameterization for isogeometric analysis based on teichmüller mapping, *Computer Methods in Applied Mechanics and Engineering*, 311, 2016, 41-55. <https://doi.org/10.1016/j.cma.2016.07.035>.
- [13] Sederberg, T. W.; Cardon, D. L.; Finnigan, G. T.; North, NS.; Zheng, J.; Lyche, T.: T-spline simplification and local refinement, *ACM Trans Graph* 2004, 23(3):276-83. <https://doi.org/10.1145/1015706.1015715>.
- [14] Sederberg, T. W.; Zheng, J.; Bakenov, A., Nasri, A.: T-splines and T-NURCCs. *ACM transactions on graphics (TOG)*, 22(3), 2003, 477-484. <https://doi.org/10.1145/882262.882295>.
- [15] Shang, C.; Fu, J.; Feng, J.; Lin, Z.; Li, B.: Effective re-parameterization and GA based knot structure optimization for high quality T-spline surface fitting, *Computer Methods in Applied Mechanics and Engineering*, 351, 2019, 836-859. <https://doi.org/10.1016/j.cma.2019.03.033>.
- [16] Sun, X.; Rosin, P. L.; Martin, R.; Langbein, F.: Fast and effective feature-preserving mesh denoising, *IEEE transactions on visualization and computer graphics*, 13(5), 2007, 925-938. <https://doi.org/10.1109/TVCG.2007.1065>.
- [17] Vollmer, J.; Mencl, R.; Mueller, H.: Improved laplacian smoothing of noisy surface meshes, In *Computer graphics forum*, Oxford, UK and Boston, USA: Blackwell Publishers Ltd, 18(3), 1999, 131-138. <https://doi.org/10.1111/1467-8659.00334>.
- [18] Wang, A.; Li, L.; Chang, H.; Zhao, G.; Wang, W.; Yang J.: T-spline surface smoothing based on 1-ring neighborhood space angle, *Journal of Computational Design and Engineering*, 2022, in preparation.
- [19] Wang, J.; Lu, Y.; Ye, L.; Chen, R.; Leach, R.: Efficient analysis-suitable T-spline fitting for freeform surface reconstruction and intelligent sampling, *Precision Engineering*, 66, 2020, 417-428. <https://doi.org/10.1016/j.precisioneng.2020.08.008>.
- [20] Wu, W.; Wang, Z. Y.; Li, Z.; Liu, W.; Fuxin, L.: Pointpwc-net: Cost volume on point clouds for (self-) supervised scene flow estimation, In *European conference on computer vision*, Springer, Cham, 2020, 88-107. <https://doi.org/10.1007/978-3-030-58558-76>.
- [21] Xu, G.; Mourrain, B.; Duvigneau, R.; Galligo, A.: Constructing analysis suitable parameterization of computational domain from cad boundary by variational harmonic method. *Journal of Computational Physics*, 252, 2013, 275-89. <https://doi.org/10.1016/j.jcp.2013.06.029>.
- [22] Xu, G.; Mourrain, B.; Duvigneau, R.; Galligo, A.: Optimal analysis-aware parameterization of computational domain in 3d isogeometric analysis, *Computer-Aided Design*, 45(4), 2013, 812-21. <https://doi.org/10.1016/j.cad.2011.05.007>.
- [23] Zhang, Q.; Shen, X.; Xu, L.; Jia, J.: Rolling guidance filter, In *European conference on computer vision*, Springer, Cham, 2014, 815-830. <https://doi.org/10.1007/978-3-319-10578-953>.
- [24] Zhao, Y.; Liu, X. G.; Peng, Q.; Bao, H. J.: Rigidity constraints for large mesh deformation, *Journal of Computer Science and Technology*, 24(1), 2009, 47-55. <https://doi.org/10.1007/s11390-009-9213-8>.
- [25] Zheng, J.; Wang, Y.; Seah H. S.: Adaptive T-Spline Surface Fitting to Z-Map Models, *Proc. of the 3rd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia*, 2005, 405-411. <https://doi.org/10.1145/1101389.1101468>.