



Jinyoung Choi 🗅, Kangmin Kim 🗅, Seongil Kim 🕩, Minseok Kim 🕩, Taekgwan Nam 🕩 and Youngjin Park *🕩

Department of Computer Engineering, Dong-A University, Busan 49315, Republic of Korea; 1824158@donga.ac.kr (J.C.); sdkup1317@donga.ac.kr (K.K.); dongaksi7@donga.ac.kr (S.K.); 1824160@donga.ac.kr (M.K.); 1824588@donga.ac.kr (T.N.) * Correspondence: yjpark@dau.ac.kr

Abstract: Triangular mesh data structures are principal in computer graphics, serving as the foundation for many 3D models. To effectively utilize these 3D models across diverse industries, it is important to understand the model's overall shape and geometric features thoroughly. In this work, we introduce a novel method for generating camera paths that emphasize the model's local geometric characteristics. This method uses a toroidal patch-based spatial data structure, approximating the mesh's faces within a predetermined tolerance ϵ , encapsulating their geometric intricacies. This facilitates the determination of the camera position and gaze path, ensuring the mesh's key characteristics are captured. During the path construction, we create a bounding cylinder for the mesh, project the mesh's faces and associated toroidal patches onto the cylinder's lateral surface, and sequentially select grids of the cylinder containing the highest number of toroidal patches as we traverse the lateral surface. The centers of the selected grids are used as control points for a periodic B-spline curve, which serves as our foundational path. After initial curve generation, we generated camera position and gaze path from the curve by multiplying factors to ensure a uniform camera amplitude. We applied our method to ten triangular mesh models, demonstrating its effectiveness and adaptability across various mesh configurations.

Keywords: camera path; bounding volume; toroidal patch; periodic B-spline curve

1. Introduction

It is known that 3D models are widely used in computer graphics, such as computeraided design (CAD), 3D printing, animation, virtual reality (VR), augmented reality (AR), and so on. The digital twin concept has made 3D models a crucial component across various industries, from state-of-the-art IT sectors to traditional manufacturing, as they continue to embrace digital twin technology for innovation. Among the various data structures for these 3D models, the triangular mesh, which comprises a set of vertices (V), edges (E), and triangles (F), stands out as a predominant choice in both industry and academic research [1–4].

For the effective utilization of 3D models (including triangular meshes) across various domains, it is essential to identify the overall shape of the model and observe its characteristics. Commercial 3D computer graphics software like Blender 3.5 or 3ds Max 2024 offer tools for generating camera paths to aid in model observation. As illustrated, Figure 1a shows the camera path generation process in Blender, and Figure 1b shows the same process in 3ds Max. Within this software, although users can directly inspect the model's shape, generating the optimal camera path can be labor-intensive and time-consuming. Moreover, the results are frequently limited by the user's knowledge, resulting in inconsistencies. Previous studies have proposed various camera path generation methods to deal with these problems [5–14].

In this paper, we introduce a new method for generating a camera path that utilizes the curvature of triangular meshes. Initially, we approximate the curvature of the vertices and construct toroidal patches [15,16]. Each toroidal patch is generated based on the curvature information of its corresponding vertex and extends its domain to approximate neighboring



Citation: Choi, J.; Kim, K.; Kim, S.; Kim, M.; Nam, T.; Park, Y. Camera Path Generation for Triangular Mesh Using Toroidal Patches. *Appl. Sci.* 2024, *14*, 490. https://doi.org/ 10.3390/app14020490

Academic Editor: Yuanyuan Liu

Received: 16 November 2023 Revised: 18 December 2023 Accepted: 19 December 2023 Published: 5 January 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/).



Figure 1. Camera path generation examples in 3d modeling software: (**a**) process for camera path in Blender 3.5, (**b**) process for camera path in 3ds Max 2024.

To ensure that a camera path captures all the intricate geometric details and follows a seamless and continuous, we enclosed the entire triangular mesh within a cylinder and placed a third-order periodic B-spline curve on the lateral surface of the cylinder. The curve's control points can be selected one by one from the grids that contain the most toroidal patches as they move along the lateral surface. Two paths derive from the B-spline curve: the camera position path, which traces the camera's spatial trajectory, and the camera gaze path, which determines the camera's focal orientation.

The main contributions of this research are summarized as follows:

- We introduce a new method to automatically generate camera paths for observing meshes. This is made possible by utilizing a toroidal patch-based spatial data structure [16], which is constructed using the curvature information that is inherent in triangular meshes.
- We enclose the triangular mesh in a cylinder and divide it into uniform grids. Each grid is assigned a weight based on its relation to neighboring triangles and toroidal patches.
- We observe that the mesh areas with considerable curvature variation have more toroidal patches. This enables quick exploration of the mesh's geometrically distinct areas.
- The camera's movement is determined by a third-order periodic B-spline curve, which is aligned with the cylindrical structure's periphery. This technique ensures that the camera's path is continuous and smooth, and the curve's control points are selected with a focus on the center of the grid that carries the highest curvature variation. Section 3 provides a detailed explanation of this algorithmic process.

2. Related Work

Vázquez et al. [5] proposed a method that uses entropy to automatically determine viewpoints for generating camera paths for 3D models. They proposed general guidelines for automated camera path generation using their method. Habibi et al. [6] proposed a technique for generating camera paths that can move smoothly around 3D point cloud models. Their approach involves estimating motion direction by maximizing photometric entropy, avoiding obstacles, and controlling optical flow to prevent sudden camera jumps. This method was developed to tackle the difficulties associated with observing large-scale models.

Curvature is used to analyze the regional shape of a triangular mesh and select viewpoints for generating camera paths [7–10]. Saleem et al. [7] introduced a method for automatic camera path generation that relied on view saliency. They adjusted the camera's velocity, magnification, and minimization along this path. However, the cost of computing the camera path depends on the resolution of the view sphere and the method used to determine the stable view area. Han et al. [8] developed a method for generating camera paths that are tailored to preview videos featuring mesh sequences. While Saleem et al. [7] focused on examining static models, Han et al. [8] expanded the scope to include the dynamic movements of models in terms of animation. Li et al. [9] improved upon Han et al. [8]'s strategy by incorporating motion information through frame transitions. However, this approach cannot handle mesh sequences that undergo topological changes over time. Zhou et al. [10] suggested an algorithm that incorporates user feedback during the path generation process, building on the insights of Han et al. [8] and Li et al. [9]. When calculating curvature for triangular meshes, outliers with significantly larger or smaller values compared to nearby points may occur because of noise from 3D scanning. In contrast to previous studies, our method can reduce the potential of outlier occurrence by applying smoothing and further mitigate the impact of outliers by incorporating toroidal patches.

To enable the smooth observation of 3D models from online databases while streaming, some researchers proposed a path-planning method that considers network bandwidth [11,12]. Zhao et al. [11,12] developed a streaming framework incorporating techniques like halting the camera, delaying its movement, or reducing the mesh quality when planning the camera path. These guarantee a seamless preview experience by accounting for model transmission speed. While this framework considers network bandwidth for camera path generation, its use is limited to scenarios with predefined viewpoints.

Oskam et al. [13] developed an automated path generation methodology to ensure collision-free camera movement while maintaining the visibility of a specific point in 3D space. In their study, the accuracy and magnitude of sampling the 3D space correlate with the scene's size. When planning routes for large areas, it is important to consider the performance limitations arising from this correlation. Amamra et al. [14] presented a path-planning technique that prioritizes minimizing path curvature. They demonstrated the efficacy of the Pythagorean-Hodograph in mitigating viewer motion sickness by seamlessly linking key-frames from a first-person view. Park et al. [17] utilized the concept of curvature to define and evaluate the sensation of smoothness on surfaces of revolution, where third-order B-spline curves represent the profile curve of these surfaces.

Several noteworthy contributions have emerged for approximating differential geometry on triangular meshes. Taubin [18] proposed a method that utilizes polyhedral approximation to estimate the principal curvature and direction of vertices in a triangular mesh. Meyer et al. [19] proposed an approach to discretely approximate differential geometry data using a triangular mesh's structural and morphological information. Kalogerakis et al. [20] utilized a statistical approach to estimate the curvature of surfaces, resulting in superior accuracy to previous methods. Zhihong et al. [21] exclusively used the triangle's normal vector to estimate curvature, achieving results similar to traditional methods. Goldfeather and Interante [15] developed a more precise and efficient method for approximating the principal direction using a second-order surface-fitting method. Plantl and Váša [22] proposed a method for estimating curvature using the Hermite radial basis function based on the surface-fitting method. Additionally, there has been an increase in research efforts to integrate this differential geometry into various mesh algorithms [23–27].

Several researchers utilized toroidal patches to address complex geometric challenges [28–30]. Park et al. [28] developed a hybrid bounding volume hierarchy (BVH) for freeform surfaces, employing a rectangle-swept sphere (RSS) at internal nodes and a second-order toroidal patch at leaf nodes to calculate an intersection curve between two freeform surfaces. In a follow-up study, Park et al. [29] introduced an innovative approach optimized for the efficient computation of self-intersecting curve(s) on a freeform surface. This method utilizes a ternary-structured BVH that incorporates toroidal patches to share rele-

vant data, unlike the conventional binary-structured BVH. Son et al. [30] employed toroidal patches to compute the Hausdorff distance between two freeform surfaces. Kim et al. [16] introduced a new spatial data structure that integrates toroidal patches with a triangular mesh. This paper proposes a novel method for generating an optimal camera path to observe triangular meshes, building on previous approaches. Our method bridges the gap between traditional path generation techniques and advanced geometric analysis by utilizing the toroidal patch-based spatial data structure [16].

3. Camera Path Generation

3.1. Preprocessing

The triangular mesh used for camera path generation can be sourced from 3D modeling software or acquired via 3D scanning. During the 3D scanning process, vertex data may contain noise, which can complicate the calculation of precise curvature. This imprecision can subsequently adversely affect the quality of curvature-based camera path optimization. To mitigate such issues, a mesh smoothing technique is employed in the preprocessing stage [23]. This step is crucial in minimizing the impact of scanning-induced noise and maintaining the integrity of the curvature-based camera path generation process.

After smoothing, we estimate the Gaussian curvature at each vertex using the surfacefitting method [15], which is the basis of constructing a toroidal patch-based spatial data structure [16]. This structure offers a refined mesh approximation and incorporates the computed curvature data. Each toroidal patch begins at a specific vertex of the smoothed mesh, using the curvature information to initiate the process. The patch then expands its domain by comparing distances to adjacent triangles. The expansion must adhere to a predefined tolerance value, the maximum allowable distance between the set of bounding faces and the toroidal patch itself. If a face is not included in any other toroidal patches, we generate a toroidal patch for that face using one of its vertices. However, sometimes this process generates toroidal patches that do not meet the tolerance. To ensure accuracy, we exclude inaccurate patches after generating toroidal patches.

In an area characterized by minimal variations in curvature, a toroidal patch expands to approximate multiple adjacent triangles within a single patch. This expansion is driven by extending the domain of the toroidal patch, covering from the reference vertex to the vertices of its adjacent triangles, all within the predetermined tolerance. Conversely, in an area with pronounced changes in curvature, which implies a visually distinct or robust appearance, the toroidal patch tends to be smaller. Therefore, a region with a high concentration of toroidal patches indicates distinct geometric attributes. We can determine the camera path by using the number of patches as a weighting factor. Without directly employing the toroidal patches, we initially enclose the mesh within a bounding cylinder, subdivide it into uniform segments, and assign the number of toroidal patches to each segment as the weight for the camera path.

The lateral surface of the bounding cylinder is denoted as $C_{h,r}(u, v)$, where $u, v \in [0, 1] \times [0, 2\pi]$. This surface is the foundation for storing the weights for camera path determination. In this formulation, *h* represents the height of the cylinder, while *r* stands for its radius. The generated $C_{h,r}(u, v)$ is uniformly divided by *m* in the *u* direction and *n* in the *v* direction, creating grids to allocate $m \times n$ weights. Figure 2 displays the results of equal partitioning of bounding cylinders for ten triangular mesh models. To allocate the weight with the toroidal patches to the uniform grids of $C_{h,r}(u, v)$, we orthogonally project the centroid of each face (triangle) onto $C_{h,r}(u, v)$. This projection involves computing the direction vector \vec{D} that originates from a point on the axis of $C_{h,r}(u, v)$ and shares the same height as the face's centroid, pointing towards $C_{h,r}(u, v)$ (i.e., \vec{D} passes by the face's centroid and a point on $C_{h,r}(u, v)$). The centroid is then offset by *r* in the direction of \vec{D} , and the resulting projected vertex is mapped to its corresponding grid.

After mapping all faces to the grid, each grid's weight is determined by averaging the number of toroidal patches associated with faces within a two-ring neighborhood. We employ a two-ring neighborhood because, in certain instances, the smoothing process may not completely remove unintended cusps at vertices. This can lead to unexpectedly high curvature values, negatively influencing the subsequent camera path generation. Utilizing this broader neighborhood offers a more comprehensive context, effectively counterbalancing localized anomalies. Higher weights indicate a larger number of the toroidal patches are contained within the corresponding grid, while the lower weights imply a lower count of the toroidal patches. Since the camera path is generated along points with the highest weights along the lateral side of the cylinder, the path passes by areas including a larger number of toroidal patches. Algorithm 1 represents all this preprocessing step.



Figure 2. $C_{h,r}(u, v)$ about various triangular mesh models. In sequence from left to right and then top to bottom: Bear, Rocker, Horse, Kitten, Bunny, Haris, Venus, Popiersie, Armadillo, and Pieta.

Algorithm 1: Preprocessing.
Input : <i>M</i> —an input triangular mesh; <i>m</i> , <i>n</i> —segment numbers of the <i>M</i> 's
bounding cylinder
Output : $C_{h,r}$ —lateral surface of the cylinder
$M \leftarrow \text{Smoothing}(M)$
$T \leftarrow \text{GenerateToroidalPatches}(M)$
$C_{h,r} \leftarrow \text{GenerateCylinder}(M, m, n)$
forall $f \in M$ do
$C_{h,r}$.ProjectFaceToGrid(f)
end forall
for $i \leftarrow 0$ to $m-1$ do
for $j \leftarrow 0$ to $n-1$ do
$C_{h,r}$.SetWeight(<i>i</i> , <i>j</i> , <i>T</i>)
end for
end for

3.2. Path Generation with Periodic B-Spline Curve

Generating the camera path involves a two-stage process: determining a camera position path and gaze path. Both paths are constructed using a third-order periodic B-spline curve, ensuring C^1 continuity and seamless transitions at both endpoints. Control points for this curve are derived from the pre-processed grid weights of $C_{h,r}(u, v)$. Specifically, for grids with identical v values, the center of the grid with the maximum weight is selected as the control point, resulting in n control points for the curve. Figure 3 provides a visual overview of this control point generation procedure, complete with sample weight values. Figure 3a illustrates the sample weight distribution on $C_{h,r}(u, v)$, Figure 3b shows the grids with the maximum weights for identical v values, colored in orange. Figure 3c shows the selected control points, colored in green.



Figure 3. Example of control point selection process: (a) grids with their weights, (b) selected grids with maximum weight, and (c) selected control points for generating periodic B-spline curve.

We can generate a periodic B-spline curve over the bounding cylinder's lateral surface using selected control points. The B-spline curve B(u) can be expressed as Equation (1) where \mathbf{p}_i express the control points, and N_i^n are the B-spline basis functions with degree n. The periodic B-spline curve can be obtained by constructing the control points and the knot sequence to satisfy the Equations (2) and (3). \triangle_i expresses knot interval between *i*th knot and i + 1th knot and K means the number of knots of the curve.

$$B(u) = \mathbf{p}_0 N_0^n(u) + \dots + \mathbf{p}_{D-1} N_{D-1}^n(u), 0 \le u \le 1$$
(1)

$$\mathbf{p}_0 = \mathbf{p}_{D-(n+1)}, \, \mathbf{p}_1 = \mathbf{p}_{D-n}, \dots, \, \mathbf{p}_n = \mathbf{p}_{D-1}$$
 (2)

$$\Delta_0, \Delta_1, \dots, \Delta_n, \Delta_{n+1}, \dots, \Delta_{K-(n+3)}, \Delta_0, \Delta_1, \dots, \Delta_n \tag{3}$$

Applying the initially generated curve directly to the camera position or gaze path can lead to excessive camera movement, depending on the distribution or configuration of the control points. This may cause visual instability or a shaky camera effect. Moreover, it may accidentally exclude certain areas of the triangle mesh from the camera's field of view. To address these issues, we adjust the radius of the $C_{h,r}(u, v)$ when generating the camera position path. If the ratio of *h* to *r* is greater than 2, we multiply *r* by 6; otherwise, we multiply *r* by 4. Subsequently, we multiply the generated curve by the scaling factor. Specifically, the camera position path is modified by a factor of 0.25, while the camera gaze path is scaled by a factor of 0.5 relative to the original periodic B-spline curve.

Figure 4 shows camera paths generated using our method for the Rocker model. The red curve represents the original third-order periodic B-spline curve derived from the control points; the blue curve shows the camera gaze path, and the green curve shows the camera position path. Algorithm 2 provides the procedure for generating these camera position and gaze paths.



Figure 4. Camera path visualization for the Rocker model. The camera gaze path (shown in blue) and the camera position path (shown in green) are generated based on the initial curve (shown in red).

Algorithm 2: Get Camera Path.
Input : $C_{h,r}$ —lateral surface of the <i>M</i> 's bounding cylinder
Output : P_g —camera gaze path, P_p —camera position path
CtrlPts \leftarrow GetControlPoints($C_{h,r}$)
$Curve \leftarrow MakePeriodicBsplines(CtrlPts)$
$P_g \leftarrow \text{Curve.Stabilization}(0.5)$
$P_{p} \leftarrow \text{Curve.Stabilization}(0.25)$
if $h/r > 2$ then
P_p .AdjustRadius(6 r)
else
P_p .AdjustRadius(4 r)
end if

Figure 5 presents the results of applying the complete process to the triangular mesh model. Figure 5a shows the smoothing result [23], and Figure 5b shows faces with different colors according to their toroidal patches' relationship. Figure 5c displays the mesh face weights, while Figure 5d illustrates mesh portions within each cylinder grid, with colors derived from the weights in Figure 5c. In both figures, faces closer to red indicate higher weights, while those closer to green indicate lower weights. Figure 5e displays the camera gaze path in blue, which is generated using our method.



Figure 5. Camera path generation workflow: (a) smoothing result of the input mesh, (b) colored faces according to included toroidal patch, (c) colored faces according to their weight, (d) colored faces according to corresponding cylinder grid's weight, and (e) camera gaze path generation result.

4. Experimental Result and Discussion

The triangular meshes were stored and represented using the PMP library [31], which employs a half-edge-based data structure. To ensure consistent experimental results, all input meshes were enclosed within an axis-aligned bounding box (AABB) and resized so that the diagonal length of its AABB equals 1. For the toroidal patch generation, the tolerance between the toroidal patch and the corresponding face(s) was set to 10^{-3} . The bounding cylinder $C_{h,r}(u, v)$ was uniformly divided into *m* and *n* segments, with m = 11 and n = 20.

Table 1 displays the count of toroidal patches corresponding to each of the ten selected models. The number of toroidal patches does not exhibit a direct proportionality to the face count of the models, suggesting that toroidal patch distribution is influenced more by geometric features, such as local curvature variations, than by the sheer number of mesh faces. For example, despite having a higher triangle count than the Haris model, the Venus model generates fewer toroidal patches, implying a lower curvature variation in the model. This discrepancy arises from variations in the models' geometric characteristics, particularly their local curvature variation. For instance, while the Venus model has more triangles than the Haris model, it yields fewer toroidal patches, indicating fewer variations in its curvature. The Rocker model exhibits the highest ratio of the toroidal patches to faces at 13%, this is because many toroidal patches are formed by a single triangle of the model. This phenomenon is due to the model's large flat areas compared to other models, preventing the merging of adjacent faces into toroidal patches. Additionally, the model's relatively low face count amplifies the effect of single-face toroidal patches on this ratio, resulting in the highest percentage observed among the models.

Model	# Face	# Vertex	# T _p	$T_p/Face$
Bear	20,188	10,096	1622	8%
Rocker	24,364	12,182	3064	13%
Horse	39,696	19,850	4393	11%
Kitten	55,304	27,652	2487	4%
Bunny	69,630	34,817	7200	10%
Haris	132,632	66,316	14,840	11%
Venus	201,514	100,759	7423	4%
Popiersie	300,000	150,002	24,594	8%
Armadillo	345,944	172,974	19,022	5%
Pieta	509,650	255,129	37,194	7%

Table 1. Numerical information of input models.

Table 2 provides comprehensive data for the camera path, which includes the radius and height of the bounding cylinder $C_{h,r}(u, v)$. The amplitude ratio A indicates the proportion of the cylinder height occupied by the curve, with a higher value suggesting an expanded vertical field of view. An analysis of the number of extreme points of the camera path and the A offers a perspective on the distribution of the model's geometric characteristics. The maximum (W_{max}) and average (W_{avg}) values associated with the number of toroidal patches allocated to the grids (with a 2-ring neighborhood relationship) also provided. Generally, as the ratio of toroidal patches to faces in Table 1 increases, there is a corresponding increase in W_{max} , W_{avg} values in Table 2. However, this relationship may not always hold true, especially when considering N-Ring scalability. For example, while the ratio of toroidal patches to faces of the Rocker model exceeds that of the Haris model, W_{max} of the Rocker is lower.

Figure 6 illustrates the camera's starting position on the camera path for each input model. This starting point is determined based on the $C_{h,r}(u, v)$ grid with the maximum weight, where the model's curvature undergoes the most significant change. Figure 7 presents the generated camera gaze paths for the input models in 3D, depicted in blue. For clarity comparison, Figure 8 shows the generated camera gaze path (blue), the position path (green), and the initial third-order periodic B-spline curve (red) in the parameter domain of $C_{h,r}(u, v)$. These paths are closely aligned with areas exhibiting significant curvature changes. In instances like Figure 7a,c,d,g, points with significant curvature

changes are concentrated in specific areas compared to other models, leading to a more detailed observation of these areas. As shown in Figure 7e,f,h, the model has a flat area where the evaluated curvature is nearly zero. When this flat area is notably extensive, the camera path is designed to focus on this flat area. The substantial curvature variation between the flat and non-flat area boundary resulted in more toroidal patches allocated to the boundary faces than other areas. However, although Figure 7j exhibits similar characteristics, its path behaves differently. This difference is due to the numerous points with significant changes in curvature, present not only at the bottom but also throughout other areas of the model. In Figure 7b,i, points with significant changes in curvature are distributed throughout the model, guiding the camera path to capture the entire model.

Model	$C_{h,r}($	$C_{h,r}(u,v)$		# Extrama pta	1 47	TA 7
	r	h	Л	# Extreme pts	v v max	<i>w</i> avg
Bear	0.27	0.80	30%	8	24.71	8.08
Rocker	0.45	0.44	29%	10	24.92	11.12
Horse	0.42	0.60	32%	8	18.15	5.76
Kitten	0.27	0.75	34%	8	12.48	5.53
Bunny	0.35	0.61	44%	10	19.20	8.25
Haris	0.36	0.73	39%	4	30.06	9.62
Venus	0.35	0.63	24%	6	9.56	5.26
Popiersie	0.26	0.80	30%	6	17.41	10.22
Armadillo	0.35	0.66	26%	10	11.13	5.52
Pieta	0.29	0.78	33%	10	23.00	7.96

Table 2. Generated cylinder and curve information.



Figure 6. Snapshot of each model that the camera is at the starting point of the generated camera path.



Figure 7. Ten selected triangular mesh models with bounding cylinder and camera gaze path: (a) Bear, (b) Rocker, (c) Horse, (d) Kitten, (e) Bunny, (f) Haris, (g) Venus, (h) Popiersie, (i) Armadillo, and (j) Pieta.



Figure 8. Visualization of our camera paths for the models. The camera gaze path (shown in blue) and the camera position path (shown in green) are generated based on the initial third-order periodic B-spline curve (shown in red).

5. Conclusions

We introduced a novel automated methodology for generating camera paths optimized for observing triangular mesh models. Using a toroidal patch-based spatial data structure [16], we can approximate local geometric characteristics in the mesh and it can be used as weights for camera path generation. The derived camera position and gaze path are defined by a third-order periodic B-spline curve along the lateral surface of the cylinder that bounds the model. This approach ensures a smooth and uninterrupted path designed for areas with significant changes in curvature. However, we recognize certain limitations of our approach, especially when applied to models predominantly composed of flat faces, which might result in undesired camera path configurations. Future work will aim to improve the handling of these limitations to enhance the robustness and versatility of our method. Additionally, we plan to conduct an integrated user study to compare our enhanced methodology with the results of previous studies [7–10]. **Author Contributions:** Conceptualization, J.C. and Y.P.; methodology, J.C. and Y.P.; software, J.C., K.K. and Y.P.; writing—original draft preparation, J.C., K.K., M.K., T.N., S.K. and Y.P.; writing—review and editing, Y.P.; visualization, J.C.; supervision, and project administration, Y.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2022-00165809, RS-2023-00245613), and in part by the Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00622, Digital Twin Testbed Establishment).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The 3D model data presented in this study are available on request from the corresponding author. The data are not publicly available because the private and public data are mixed, and we resized the model for experiments.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- 1. Zhao, F.; Sun, G. Planar Delaunay Mesh Smoothing Method Based on Angle and a Deep Q-Network. *Appl. Sci.* 2023, *13*, 9157. [CrossRef]
- Park, J.H.; Moon, J.H.; Park, S.; Yoon, S.H. GeoStamp: Detail Transfer Based on Mean Curvature Field. *Mathematics* 2022, 10, 500. [CrossRef]
- 3. Gao, M.; Ruan, N.; Shi, J.; Zhou, W. Deep Neural Network for 3D Shape Classification Based on Mesh Feature. *Sensors* 2022, 22, 7040. [CrossRef] [PubMed]
- 4. Ha, Y.; Park, J.H.; Yoon, S.H. Geodesic Hermite Spline Curve on Triangular Meshes. *Symmetry* 2021, 13, 1936. [CrossRef]
- Vázquez, P.P.; Feixas, M.; Sbert, M.; Heidrich, W. Viewpoint selection using viewpoint entropy. In Proceedings of the VMV, Citeseer, Stuttgart, Germany, 21–23 November 2001; Volume 1, pp. 273–280.
- Habibi, Z.; Caron, G.; Mouaddib, E.M. 3d model automatic exploration: Smooth and intelligent virtual camera control. In Proceedings of the Computer Vision—ACCV 2014 Workshops, Singapore, 1–2 November 2014; Springer: Berlin/Heidelberg, Germany, 2015; pp. 612–626.
- Saleem, W.; Song, W.; Belyaev, A.; Seidel, H.P. On computing best fly. In Proceedings of the 23rd Spring Conference on Computer Graphics, Budmerice, Slovakia, 26–28 April 2007; pp. 115–121.
- Han, S.R.; Yamasaki, T.; Aizawa, K. Automatic preview video generation for mesh sequences. In Proceedings of the 2010 IEEE International Conference on Image Processing, Hong Kong, China, 26–29 September 2010; pp. 2945–2948.
- 9. Li, Y.; Cui, Q.; Dou, F.; Zhang, L.; Zhou, Z. Automatic mesh animation preview. In Proceedings of the 2014 IEEE International Conference on Multimedia and Expo (ICME), Chengdu, China, 14–18 July 2014; pp. 1–6.
- 10. Zhou, Z.; Jiang, N.; Chen, K.; Zhang, J. Automatic Mesh Animation Preview with User Voting-Based Refinement. *IEEE Trans. Multimed.* **2016**, *19*, 327–339. [CrossRef]
- Zhao, S.; Ooi, W.T.; Carlier, A.; Morin, G.; Charvillat, V. 3D mesh preview streaming. In Proceedings of the 4th ACM Multimedia Systems Conference, Oslo, Norway, 26 February–1 March 2013; pp. 178–189.
- Zhao, S.; Ooi, W.T.; Carlier, A.; Morin, G.; Charvillat, V. Bandwidth adaptation for 3D mesh preview streaming. ACM Trans. Multimed. Comput. Commun. Appl. TOMM 2014, 10, 1–20. [CrossRef]
- Oskam, T.; Sumner, R.W.; Thuerey, N.; Gross, M. Visibility transition planning for dynamic camera control. In Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, New Orleans, LA, USA, 1–2 August 2009; pp. 55–65.
- Amamra, A.; Amara, Y.; Benaissa, R.; Merabti, B. Optimal camera path planning for 3D visualisation. In Proceedings of the 2016 SAI Computing Conference (SAI), London, UK, 13–15 July 2016; pp. 388–393.
- 15. Goldfeather, J.; Interrante, V. A novel cubic-order algorithm for approximating principal direction vectors. *ACM Trans. Graph. TOG* **2004**, *23*, 45–63. [CrossRef]
- 16. Kim, K.; Choi, J.; Park, Y. A New Spatial Data Structure for Triangular Mesh with Toroidal Patches. J. King Saud Univ. Comput. Inf. Sci. 2024, 36, 101891. [CrossRef]
- 17. Park, H.; Kim, C.; Park, Y. The Variables of Surface of Revolution and its effects on Human Visual Preference. *J. Korea Comput. Graph. Soc.* **2022**, *28*, 31–40. [CrossRef]
- 18. Taubin, G. Estimating the tensor of curvature of a surface from a polyhedral approximation. In Proceedings of the IEEE International Conference on Computer Vision, Cambridge, MA, USA, 20–23 June 1995; pp. 902–907.
- Meyer, M.; Desbrun, M.; Schröder, P.; Barr, A.H. Discrete differential-geometry operators for triangulated 2-manifolds. In Proceedings of the Visualization and Mathematics III, Berlin, Germany, 22–25 May 2003; Springer: Berlin/Heidelberg, Germany, 2003; pp. 35–57.

- 20. Kalogerakis, E.; Simari, P.; Nowrouzezahrai, D.; Singh, K. Robust statistical estimation of curvature on discretized surfaces. In Proceedings of the Symposium on Geometry Processing, Barcelona, Spain, 4–6 July 2007; Volume 13, pp. 110–114.
- 21. Zhihong, M.; Guo, C.; Yanzhao, M.; Lee, K. Curvature estimation for meshes based on vertex normal triangles. *Comput. Aided Des.* **2011**, 43, 1561–1566. [CrossRef]
- Prantl, M.; Váša, L. Estimation of differential quantities using Hermite RBF interpolation. *Vis. Comput.* 2018, 34, 1645–1659. [CrossRef]
- Desbrun, M.; Meyer, M.; Schröder, P.; Barr, A.H. Implicit fairing of irregular meshes using diffusion and curvature flow. In Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '99, Los Angeles, CA, USA, 8–13 August 1999; pp. 317–324.
- 24. Dyn, N.; Hormann, K.; Kim, S.J.; Levin, D. Optimizing 3D triangulations using discrete curvature analysis. *Math. Methods Curve Surf.* 2001, *1*, 135–146.
- Kim, S.J.; Kim, S.K.; Kim, C.H. Discrete differential error metric for surface simplification. In Proceedings of the 10th Pacific Conference on Computer Graphics and Applications, Beijing, China, 9–11 October 2002; pp. 276–283.
- 26. Kim, S.J.; Kim, C.H.; Levin, D. Surface simplification using a discrete curvature norm. *Comput. Graph.* **2002**, *26*, 657–663. [CrossRef]
- 27. Zhao, H.; Xu, G. Triangular surface mesh fairing via Gaussian curvature flow. J. Comput. Appl. Math. 2006, 195, 300–311. [CrossRef]
- 28. Park, Y.; Son, S.H.; Kim, M.S.; Elber, G. Surface–surface-intersection computation using a bounding volume hierarchy with osculating toroidal patches in the leaf nodes. *Comput. Aided Des.* **2020**, *127*, 102866. [CrossRef]
- 29. Park, Y.; Hong, Q.Y.; Kim, M.S.; Elber, G. Self-intersection computation for freeform surfaces based on a regional representation scheme for miter points. *Comput. Aided Geom. Des.* **2021**, *86*, 101979. [CrossRef]
- Son, S.H.; Kim, M.S.; Elber, G. Precise Hausdorff distance computation for freeform surfaces based on computations with osculating toroidal patches. *Comput. Aided Geom. Des.* 2021, *86*, 101967. [CrossRef]
- 31. Botsch, M.; Kobbelt, L.; Pauly, M.; Alliez, P.; Lévy, B. Polygon Mesh Processing; CRC Press: Boca Raton, FL, USA, 2010.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.