

Article

Interactive Mesh Sculpting with Arbitrary Topologies in Head-Mounted VR Environments

Xiaoqiang Zhu *  and Yifei Yang

School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China; icbyone@shu.edu.cn

* Correspondence: xqzhu@shu.edu.cn

Abstract: Shape modeling is a dynamic area in computer graphics with significant applications in computer-aided design, animation, architecture, and entertainment. Virtual sculpting, a key paradigm in free-form modeling, has traditionally been performed on desktop computers where users manipulate meshes with controllers and view the models on two-dimensional displays. However, the advent of Extended Reality (XR) technology has ushered in immersive interactive experiences, expanding the possibilities for virtual sculpting across various environments. A real-time virtual sculpting system implemented in a Virtual Reality (VR) setting is introduced in this paper, utilizing quasi-uniform meshes as the foundational structure. In our innovative sculpting system, we design an integrated framework encompassing a surface selection algorithm, mesh optimization technique, mesh deformation strategy, and topology fusion methodology, which are all tailored to meet the needs of the sculpting process. The universal, user-friendly sculpting tools designed to support free-form topology are offered in this system, ensuring that the meshes remain watertight, manifold, and free from self-intersections throughout the sculpting process. The models produced are versatile and suitable for use in diverse fields such as gaming, art, and education. Experimental results confirm the system's real-time performance and universality, highlighting its user-centric design.

Keywords: virtual sculpting; Virtual Reality; adaptive topology; animation; video game

MSC: 68U05



Citation: Zhu, X.; Yang, Y. Interactive Mesh Sculpting with Arbitrary Topologies in Head-Mounted VR Environments. *Mathematics* **2024**, *12*, 2428. <https://doi.org/10.3390/math12152428>

Academic Editor: Sheng-Wei Chi

Received: 1 May 2024

Revised: 8 July 2024

Accepted: 2 August 2024

Published: 5 August 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/>).

1. Introduction

Extended Reality (XR) encompasses all immersive technologies, including Augmented Reality (AR), Virtual Reality (VR), and Mixed Reality (MR), each of which enhances the way digital content interacts with the real world. Augmented Reality overlays virtual objects onto the real world, allowing users to experience this blend through devices such as AR glasses, smartphones, or tablets. Virtual Reality offers a 360-degree view of a digital environment, achieved through VR headsets or head-mounted displays, fully immersing users in a simulated world. Mixed Reality merges real-world and virtual elements, enabling them to coexist and interact in real time. These technologies facilitate unprecedented creative interaction between humans and machines, significantly advancing disciplines like science, art, and technology [1–5]. For instance, the integration of the online automobile rental and e-marketplace with augmented reality technology [6], leveraging its unique advantages such as real-time interaction and precise 3D recognition, has provided a more convenient, intuitive, and efficient service model for the automobile rental industry.

In the realm of computer graphics, shape modeling is pivotal, with a rich history of diverse methodologies. These range from 3D shape reconstruction [7–9], where users construct 3D models from RGB or depth images, to statistical shape modeling [10], which generates models semantically similar to pre-existing references. Free-form modeling is another prevalent method, which is split mainly into 3D sketching [11–13] and virtual

sculpting [14–16]. Three-dimensional (3D) sketching allows users to begin with spontaneous doodles in space, which evolve into complex curves or surfaces, forming objects that can be detailed without deformation. Conversely, virtual sculpting typically starts from a basic 3D model that users shape using tools like inflating, deflating, and smoothing, and it is becoming increasingly significant in creating 3D content for industries such as gaming, film, and entertainment.

Traditionally, virtual sculpting was performed using desktop displays or holographic two-dimensional screens [17]. The evolution of Mixed Reality technologies, including devices like the HTC Vive, Oculus Rift, and Hololens, has revolutionized this field, allowing for highly immersive and interactive sculpting experiences. However, most of the existing sculpting tools in VR fall into mainly two categories, sculpting a manifold mesh with simple topologies and modeling a free topological implicit surface embedded in a space-partitioned grid with limited resolutions. Motivated by these advancements, we have developed a real-time virtual sculpting system in a VR environment, which allows arbitrary topologies and resolutions. The main contributions of our system are summarized as follows:

1. The concept of quasi-uniform mesh is extended to virtual reality sculpting systems. Through a series of meticulously designed algorithms, we achieve the capability to perform arbitrary topological deformations on mesh within a VR environment. Concurrently, the robustness of the mesh structure can be ensured throughout the sculpting process.
2. Building on the foundation of achieving arbitrary topology and maintaining mesh robustness, we have integrated a suite of essential sculpting tools—pull, push, flatten, smooth, and paint—into our system. These tools are designed to cater to the diverse and universal sculpting needs of users, enabling a broad range of artistic and technical manipulations within the virtual environment.
3. Considering the diversity of the user base, we have meticulously designed an intuitive operation interface for the sculpting system, ensuring it is accessible even to novices who have never engaged in sculpting before. This design significantly lowers the learning curve, allowing users to quickly become proficient with the system. The interface is especially user-friendly for beginners, facilitating an easy and efficient introduction to digital sculpting.

The rest of this manuscript is organized as follows: we first review the related work in Section 2. The details of the implementation of our sculpting system are described in Section 3, encompassing the algorithms utilized during a sculpting process and the underlying mesh architecture. The details of the system interface and sculpting results are discussed in Section 4, which are followed by the conclusion and future work in Section 5.

2. Related Work

2.1. Modeling Paradigm

The 3D modeling process encompasses a variety of techniques including parametric modeling, Non-Uniform Rational B-Splines (NURBS) modeling, polygon modeling, and sculpting methods like volumetric and polygon sculpting.

Parametric modeling, widely utilized in Computer-Aided Design (CAD) applications such as SOLIDWORKS [18] and Autodesk Inventor [19], generates models based on user-defined parameters like thickness or radius. This method allows for the easy modification of model features by adjusting parameters, significantly reducing repetitive modeling tasks. However, constraints and dependencies need to be meticulously defined by users to maintain consistency in design updates, demanding substantial expertise and thus resulting in a higher learning curve. Parametric modeling is especially beneficial for designs requiring strict adherence to performance metrics and manufacturing standards.

NURBS modeling, which uses mathematical formulas to precisely represent object geometries, is optimal for modeling complex curved surfaces. Despite its accuracy, it becomes less practical for very complex surfaces. Industries requiring high-quality surface

finishes, such as automotive and aerospace, commonly use NURBS modeling software like Rhino 8 [20].

Polygon modeling, employed by prominent 3D software like Maya 2024 [21], 3DS Max 2024 [22], Blender 4.2 [23] and Cinema 4D 2024 [24], involves the direct manipulation of a mesh's vertices [25]. This approach provides an intuitive editing experience, though it approximates true geometrical shapes like spheres or arcs with multiple polygons, potentially compromising model accuracy. Despite this, polygon modeling is highly efficient for rendering and is predominantly used in film, animation, and gaming.

Sculpture-based modeling, including volumetric and polygon sculpting, offers an intuitive model editing experience. Volumetric sculpting, proposed by Galyean et al. [26], segments three-dimensional space into a voxel grid, where each voxel represents a portion of the object. This method uses the Marching Cubes algorithm [27] to create surfaces based on voxel states, allowing for flexible edits independent of other model parts. However, model detail is limited by voxel resolution and high resolutions can be memory intensive, making it less suited for intricate models. Subsequently, various volumetric sculpting methods have been proposed [28–31]. Because each voxel is independent and does not interfere with others, volumetric sculpting is relatively flexible, and the current state of the model does not affect the subsequent editing process. However, the level of detail in the model is limited by the resolution of the voxels, and higher resolutions can lead to issues with excessive memory usage. Therefore, volumetric sculpting is more suitable for creating simple models without sharp features.

Polygon sculpting, as exemplified by software such as Zbrush [32], enables the direct manipulation of vertices and faces to construct detailed models. This method strikes an effective balance between model precision and memory efficiency, rendering it ideal for a wide range of modeling tasks. However, the inherent simplicity in the topologies used during the sculpting process can sometimes restrict its ability to preserve intricate details. This limitation may affect the sculpting of highly detailed or complex geometries, where more advanced topological features might be necessary.

To this end, we have developed a universal sculpting tool tailored for real-time virtual reality environments. Our approach is grounded in the paradigm of polygon sculpting, which not only supports free topology but is also well suited for the dynamic and immersive nature of virtual reality scenarios. This tool is designed to seamlessly integrate into VR systems, providing users with an intuitive and flexible modeling experience that leverages the benefits of polygon sculpting techniques.

2.2. Virtual Sculpting

In the virtual sculpting method proposed in 1991 by Galyean et al. [26], an initial mesh structure was used to represent the starting form for the user's sculpting process, with changes recorded through customized interactive tools within a voxel grid space. The voxel data were then converted into a mesh using the Marching Cubes algorithm. Wong et al. [15] introduced a method for direct shape control using electronic gloves as input devices, creating a control surface based on hand nodes, including joints and fingertips, to guide the sculpting operation. Surfaces were parameterized, and the model was updated by interpolating changes in hand positions. McDonnell et al. [16] presented an interactive sculpting framework called Virtual Clay, which was based on subdivision solids and physical modeling. This digital sculpting technology simulates the workflow of using physical clay where the starting object is a virtual three-dimensional elastic clay. Users can modify the topological structure of the virtual clay using various provided sculpting tools, with the virtual clay responding predictably to user actions, which is enhanced by tactile feedback for a realistic experience. In the realm of commercial applications, Geomagic Freeform [33] emerges as a 3D design software that integrates seamlessly with tactile devices, specifically featuring a force-feedback sculpting pen. This combination enables a seamless transition from manual craftsmanship to digital design through intuitive

tactile interactions, significantly reducing the design learning curve and accelerating the design realization.

De Goes et al. [34] proposed Regularized Kelvinlets, which is a physically-based virtual sculpting method where model deformation is achieved through the regularization of linear elastic fundamental solutions, allowing for the realistic simulation of elastic materials during interactive sculpting. Another study introduced a method for the real-time sculpting of triangular manifold meshes with arbitrary deforming surfaces using quasi-uniform grids, where users can select various gestures from a toolset for operations ranging from smoothing to inflating surface regions, ensuring the output mesh remains closed and non-self-intersecting [25].

In the previous works mentioned, the manipulation of sculpted models typically required custom interaction tools and was visualized on desktop displays, limiting users to a two-dimensional projection of three-dimensional objects. Users often needed to rotate the scene to observe the global model. However, with the advancement of mixed reality technology, free-form shape modeling has increasingly adopted the interactive mechanisms of Virtual Reality (VR) and Augmented Reality (AR). VR and AR displays provide users a complete three-dimensional view and an immersive experience, enabling an intuitive design and review of three-dimensional models from any viewpoint. This significantly enhances the artistic process in sculpture modeling within VR or AR environments.

Jang et al. [35] introduced an AR-based 3D sculpting system that enables users to sculpt objects in real-world scenarios using midair gestures. This system utilizes a head-mounted display (HMD) and camera to track the user's hands and head in three-dimensional space, rendering virtual sculpted objects that seamlessly integrate with the real-world environment. Lu et al. [36] developed a method for the real-time non-photorealistic rendering of virtual sculpted content within augmented reality scenes, allowing for surface deformation through tactile interaction. Their approach includes sophisticated post-processing techniques to enhance visualization effects and ensure consistent rendering across time and space.

In another advancement, a virtual sculpting system described in the literature [37] employs a physical surface equipped with pressure sensors, simulating the sensation of direct mesh contact during editing for a more immersive experience. Extending the foundational research on virtual clay by McDonnell et al. [16], various studies have adapted this sculpting technique to different contexts. Using a virtual elastic clay model to shape final designs was proposed by Milliez et al. [38], rather than traditional surfaces, enabling the blending of individual models into cohesive structures. Gao et al. [39] introduced a virtual pottery system within a VR environment, featuring an intuitive and sophisticated interface for artistic creation. This system employs a cylindrical clay mesh as the base for sculpting, allowing users to dynamically adjust parameters such as height, thickness, and radius to mold the mesh in real time, achieving the desired form.

Further research continues to explore the capabilities and applications of virtual clay, as noted by Gao et al. [40,41], though the use of virtual clay mediums has been predominantly confined to creating ceramic models and may not be suitable for other sculpting scenarios. Additionally, a notable study by Zeidler et al. [42] focuses on sculpting human bodies within virtual reality environments, demonstrating the breadth of VR sculpting applications. Zhu et al. [43] developed a method for 3D modeling and sculpting in VR environments based on surface deformation, employing a process that iteratively adjusts the influence region of the initial base model, performs weighted deformation, and locally reconstructs topology to gradually sculpt the target shape. However, this method does not support changes in the topological genus, which introduces certain limitations on the shapes that can be modeled.

Inspired by these pioneering works, we have developed a real-time VR sculpting tool that supports free topology, integrating common functions used in general sculpting modeling and meeting the requirements of various sculpting scenarios. Our inter-

face is designed to be simple and user-friendly, making it suitable for both professionals and amateurs.

3. System Implementation

Our system is equipped with two types of brush modes and multiple sculpting modes, which are designed to process sculpting tasks in three-dimensional space based on the user's controller inputs. The workflow for handling a single controller input involves several key steps: selecting surface points, optimizing the mesh, deforming the mesh, and fusing the topology. These steps ensure that each sculpting action is accurately executed, reflecting the user's intent and enhancing the overall precision and quality of the 3D sculpting process.

3.1. Efficient Sculpting Region Selection

During the preliminary stage of mesh sculpting, selecting the surface vertices is paramount because it determines the extent of subsequent deformations. To enhance the efficiency of this step, an octree data structure is employed to organize the vertex data. Furthermore, to cater to the diverse needs of sculpting tasks, two brush modes based on Euclidean distance and geodesic distance, respectively, have been designed, as shown in Figure 1. These brush modes allow for the flexible adjustment of vertex selection strategies according to the specific demands of artistic creation, thereby achieving more refined and individualized sculpting results.

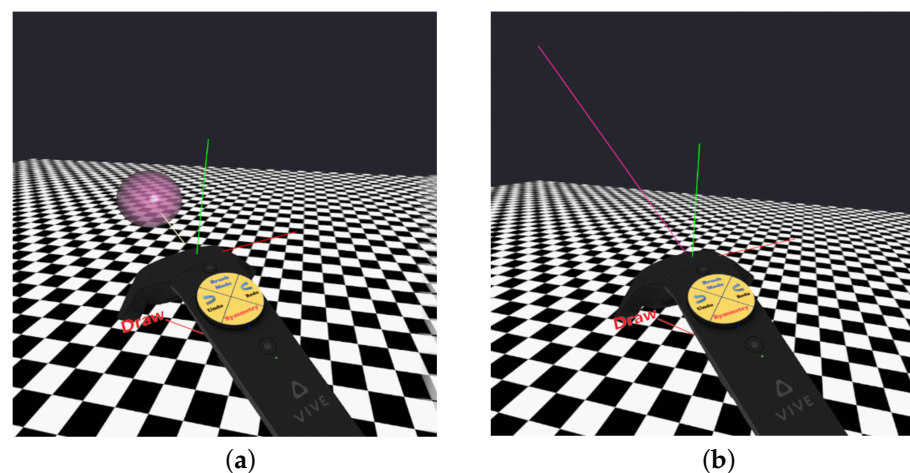


Figure 1. Two brush modes. The pink sphere in (a) is a spherical brush with its points selection strategy based on the Euclidean distance. The pink ray in (b) is a ray-shaped brush with its points selection strategy based on the geodesic distance. These brushes are oriented toward the negative direction of the Z-axis. Additionally, the green ray points toward the positive direction of the Y-axis, and the red ray points toward the positive direction of the X-axis.

3.1.1. Octree-Accelerated Triangle-Vertices Query

The octree structure, first proposed by Dr. Hunter in 1978, extends the quad-tree concept into three-dimensional space. This tree-like data structure offers significant advantages in spatial decomposition, leading to its widespread adoption for describing three-dimensional spaces. The octree structure efficiently partitions geometric entities in three-dimensional space by dividing them into voxels of equal volume; these spatial objects are recursively subdivided into smaller cubes with edge lengths of 2^n using a looped recursive approach that forms a directional graph with a root node. Within this structure, if a subdivided voxel possesses homogeneous attributes, it is designated as a leaf node. If not, the voxel is further subdivided into eight smaller cubes. This process of recursive subdivision continues, as illustrated in Figure 2.

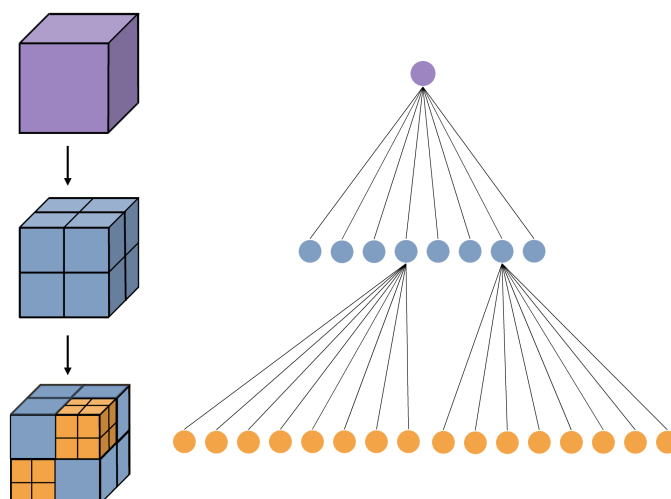


Figure 2. The process of dividing the octree structure and the schematic diagram of its structure.

The octree data structure accelerates spatial queries, which we utilize to expedite the surface points selection process. During the model initialization phase, we construct an octree that synchronously stores triangles. In the query phase, the starting point of sculpting serves as the center, and the sculpting radius defines the query radius. Candidate triangles are searched for within this radius in the octree. These candidate triangles are then utilized in the subsequent vertex selection process. This approach allows our points selection process to avoid global traversal, focusing searches only within the specified local region. This method significantly enhances the efficiency and responsiveness of our modeling tool.

3.1.2. Points Selection for Spherical Brush

In the spherical brush mode, the selection of grid vertices in three-dimensional space is determined based on Euclidean distance. After candidate triangles have been identified in the previous step, the associated vertices are extracted as candidate points. These candidate points are then traversed with the Euclidean distance being calculated between the coordinates of each candidate point and the sculpting starting point. Points with a distance less than the sculpting radius are selected as points pending update. This process effectively completes the selection of points on the grid in the spherical brush mode, ensuring that only vertices within the designated sculpting area are targeted for modification.

3.1.3. Points Selection for Ray-Casting Brush

The spherical brush mode effectively meets the requirements of conventional sculpting and excels in the detailed sculpting of local areas. However, certain scenarios demand a broader, more global perspective for sculpting. In these cases, the original model may be positioned far from the user, encompassing a large sculpting area, making the spherical brush mode less suitable. To accommodate the needs of large-scale sculpting while still focusing on detail and accurately selecting points on the mesh surface, the ray-shaped brush mode has been developed. In this mode, geodesic distance is employed during the point selection process. This approach ensures the precise selection of surface points within a defined geodesic radius, even in complex regions of the mesh, thereby preventing the selection of irrelevant points.

Unlike Euclidean distance, which denotes the direct shortest distance between two points in space, geodesic distance on a three-dimensional mesh refers to the length of the shortest path along the surface between two vertices. As shown in Figure 3, the Euclidean distance between the starting point p_1 and the ending point p_6 is represented by the length d_{16} of the dashed line segment. Conversely, the geodesic distance is the sum of the lengths of the actual path segments along the mesh surface, which is calculated as the minimum

value of $d_{12} + d_{23} + d_{34} + d_{45} + d_{56}$. This measurement is crucial for accurate point selection in complex sculpting scenarios.

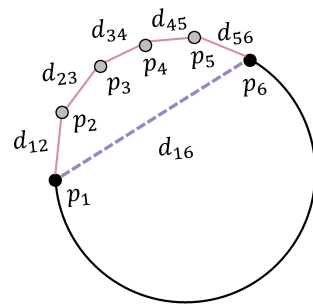


Figure 3. The geodesic distance and Euclidean distance between two points on the surface.

Calculating the geodesic distance involves assessing the accessibility between points. Firstly, the vertices of the candidate triangles identified in the previous steps are designated as candidate points. The triangle information is then utilized to establish neighboring points for each vertex, creating an adjacency graph among the vertices. Subsequently, the candidate points are screened using the Euclidean distance criterion, selecting those with a Euclidean distance less than the sculpting radius. These points form the set of geodesic candidate points. If the coordinates of the starting point coincide with any existing vertices, the overlapping vertex is designated as the starting point; otherwise, the neighboring information of the starting point is incorporated into the adjacency graph. Then, starting from the starting point, Dijkstra's algorithm is employed to calculate the shortest path lengths to each candidate point. These distances represent the geodesic distances from the starting point to the candidate points. Finally, the geodesic candidate points are iterated over, selecting those with a geodesic distance less than the sculpting radius as the points pending update. This completes the process of selecting points under the ray-shaped brush mode.

This structured approach ensures precise and efficient point selection, optimizing the sculpting process in complex scenarios. In various sculpting contexts, both the spherical brush and the ray-shaped brush offer distinct advantages, as demonstrated through our work on an existing dog model. As depicted in Figure 4a, when adding ornamental details to the back area of the dog, both brushes perform comparably on surfaces that are smooth and simply structured, as illustrated in Figure 4b,c.

Typically, we prefer using the spherical brush in standard sculpting contexts, especially for fine detailing at close range, where it provides more precise control. However, for more complex model structures, such as placing a collar around the dog's neck as shown in Figure 4d, which differs from the flat body region depicted in Figure 4a, there is a gap between the ear and the head surface in Figure 4d. In such cases, the points selected under the two brush modes may be different. As depicted in Figure 4e, the spherical brush selects points based on Euclidean distance from the brush's center regardless of whether these points form a continuous surface. This can lead to the selection of points on discontinuous surfaces like the dog's ear and neck. In contrast, the ray-shaped brush, as shown in Figure 4f, selects points based on geodesic distance, which allows for precise selection on the neck without affecting the ear's surface.

Owing to its design, the ray-shaped brush can extend its reach to more distant areas, making it more suitable for long-distance sculpting or when precise point selection is necessary. Therefore, for tasks requiring meticulous point accuracy or in contexts involving extended distances, we recommend using the ray-shaped brush mode.

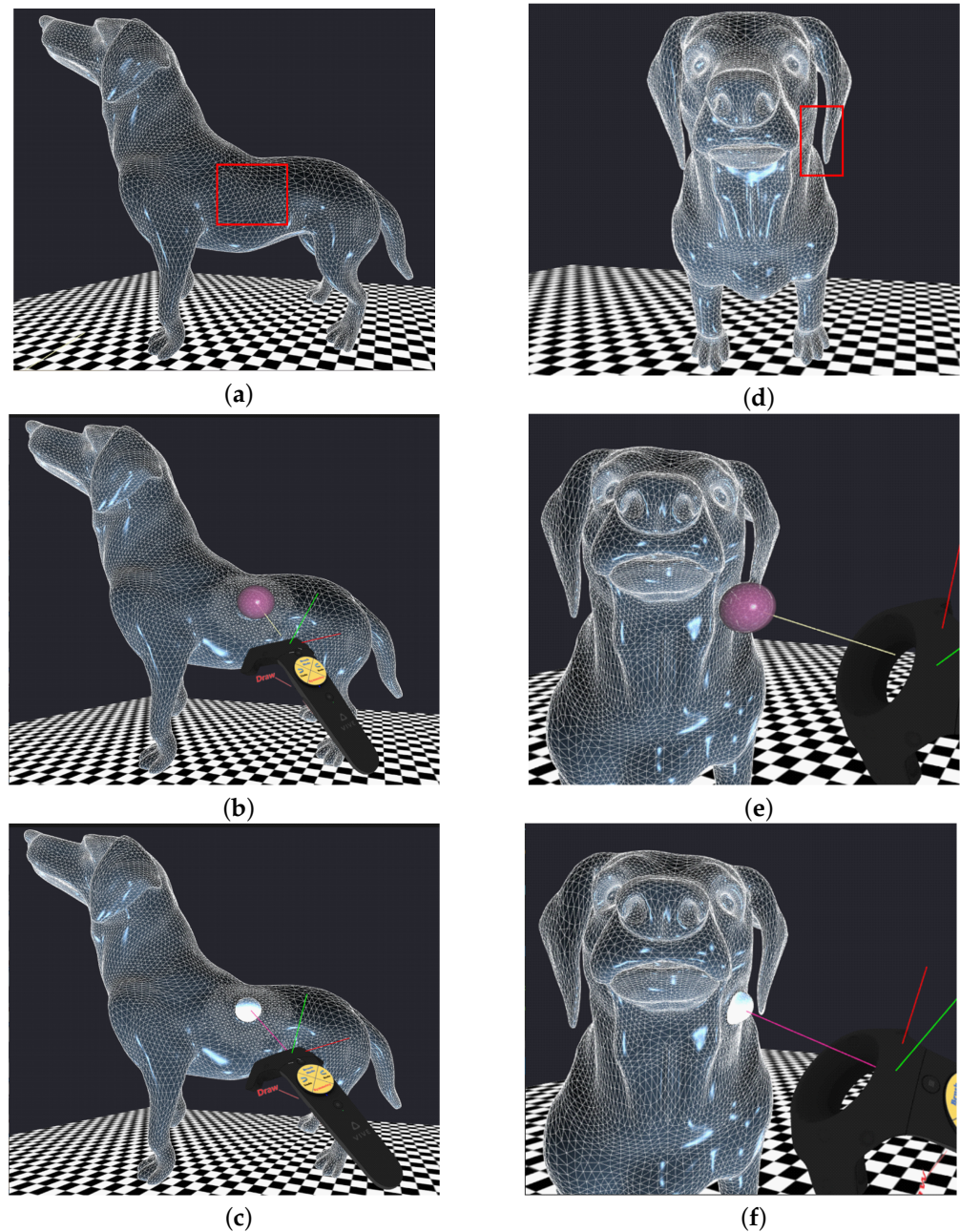


Figure 4. Points selection scenarios for two brush modes. (a,d) is the origin dog model; the area in the red wireframe is the section we intend to sculpt. (b,c), respectively, correspond to the sculpting outcomes of the spherical brush and the ray-shaped brush on simple surfaces. (e,f), on the other hand, represent the sculpting results of the spherical brush and the ray-shaped brush in more complex area.

3.2. Mesh Optimization

To ensure robustness during the deformation of manifold grids, it is essential to perform pre-deformation optimization on the grid to achieve high-quality results once the vertices pending update are identified. The core aspect of maintaining robustness throughout the sculpting process is the suitability of the mesh structure. Uniform meshes consist of nearly identical polygons, whereas non-uniform meshes are composed of polygonal faces of varying sizes. Uniform grid models often exhibit superior characteristics in many respects, including more stable structures and support for more efficient advanced geometric algorithms, compared to non-uniform mesh models. However, creating models made up of uniform meshes presents challenges, especially for surface models with complex topological structures. For our system, a simple, automated, base-level geometric model

that can adapt to various deformations at interactive frame rates is required. Inspired by the work of Stanculescu et al. [25], the quasi-uniform meshes are selected as the underlying model for mesh optimization. Quasi-uniform meshes are not strictly uniform structures, but they satisfy the robustness requirements necessary for mesh deformation in our system.

Before discussing quasi-uniform meshes, we define two terms:

- A d_{detail} tight mesh for a given closed manifold mesh M and a threshold d_{detail} , where M is considered tight if every edge in M is smaller than d_{detail} . This configuration has the advantage of possessing enough vertices to accurately reflect the geometry of the underlying surface with precision better than d_{detail} .
- A manifold mesh conforms to a minimum length d if it is derived by iterating over all the edges of the initial mesh M and collapsing those edges whose lengths are less than d .

With this foundation, a quasi-uniform mesh can be defined as follows: there exist d_{detail} and $d \leq d_{detail}/2$, such that M ensures all edges are shorter than d_{detail} but tends to maintain edge lengths greater than d . This mesh structure favors a uniform distribution of surface vertices. According to the definition of quasi-uniform mesh, to achieve such a mesh, our optimization iteration steps are primarily divided into two processes: edge splitting and edge collapsing.

3.2.1. Edge Split

Given a manifold mesh M , all edges of M are traversed, and those edges whose length exceeds d_{detail} are split. This splitting operation involves adding a vertex at the midpoint of an edge and subsequently splitting the associated triangle into two adjacent triangles. This edge iteration process is performed using a simple queue, where newly formed edges are inserted into the queue. This procedure is repeated until a finite number of iterations results in a d_{detail} tight mesh.

The splitting operation is illustrated in Figure 5. The pink edge in the left figure, being longer than d_{detail} , is selected at its midpoint, splitting the edge into two equal parts. The resultant four green edges, shown in the right figure, each have a length less than that of the original longest edge. This ensures that the mesh gradually conforms to the desired tightness without excessively elongating any individual segment.

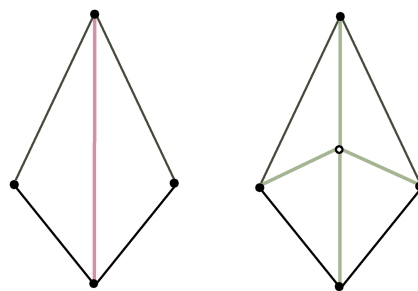


Figure 5. Edge split. As the pink edge splits into two equal segments, the four created green edges are smaller than the largest initial edge.

3.2.2. Edge Collapse

Although the mesh obtained from the above steps has the d_{detail} tight property, this alone does not guarantee the high quality of the triangle mesh. To improve the quality of the triangles, it is necessary to address edges shorter than d regardless of the increased time complexity this entails.

During the collapsing process, the vertices of the edge to be collapsed are moved to the midpoint of the edge. This adjustment ensures that no edge lengths in mesh M exceed the collapse threshold d . As depicted in Figure 6, the green edge in the left figure is shorter than d , prompting a collapsing operation. The resultant mesh structure, shown in the right figure, reflects the changes made to optimize edge length distribution within the mesh.

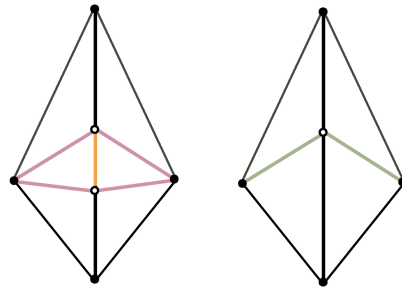


Figure 6. Edge collapse. The orange edge shorter than d will be collapsed, and the four pink edges will become two green edges.

In addition, it is crucial to determine the legality of a collapse operation before its execution. As illustrated in Figure 7, if the collapse operation results in intersecting triangles, such an operation is deemed illegal, and therefore, no collapse will be performed on the current edge. This precaution ensures that the mesh maintains its structural integrity and avoids geometric anomalies.

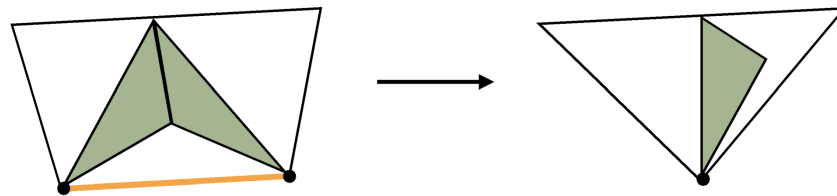


Figure 7. The illegal collapse operation. The collapse of the orange edge results in the original two adjacent green triangles becoming self-intersecting.

While the collapse operation may increase the length of some of some edges, it is a challenge to ensure that all edges post-operation meet the minimum length d . However, it can be ensured that the majority of the mesh edge lengths adhere to this minimum length. Additionally, the collapse operation might compromise the d_{detail} tight property of the mesh. Therefore, it is necessary to first execute the collapse algorithm first and subsequently restore the d_{detail} tight property. Given that d is set to satisfy $d \leq d_{detail}/2$, it can be ensured that once the d_{detail} tight property is re-established after collapse, any edges in the mesh longer than d_{detail} will not be split into shorter edges than d . This approach helps maintain the desired mesh properties while preventing undue fragmentation of the mesh structure.

During mesh optimization, the number of optimized triangles will be adaptively adjusted based on the brush radius. The maximum and minimum edge lengths in the quasi-uniform mesh are positively correlated with the brush radius, enabling the adjustment of sculpting resolution by altering the brush radius. We have empirically established this relationship based on experience:

$$d_{detail}^2 = 0.05 * radius^2$$

$$d^2 = 0.25 * d_{detail}^2$$

Consequently, for the same original mesh, a smaller brush radius results in smaller maximum and minimum edge lengths, leading to more edges being spilt and fewer edges being folded in the mesh. This results in obtaining more triangles and a denser optimized mesh. Conversely, when the brush radius is larger, the maximum and minimum edge lengths are also larger, leading to fewer edges being spilt and more edges being folded in the mesh. This results in obtaining fewer triangles and a sparser optimized mesh.

Compared to voxel-based sculpting systems [28–31], the use of quasi-uniform mesh better accommodates varying sculpting resolution requirements without imposing signif-

icant memory burdens associated with higher resolutions, making it highly suitable for general sculpting scenarios.

3.3. Mesh Deformation

After the initial steps are completed, an optimized quasi-uniform mesh for the region to be updated is obtained. Then, this mesh is deformed using a deformation field whose resolution matches d_{detail} . When handling the quasi-uniform mesh M and the deformation field D , a sampling-guaranteed tightness approach is adopted, considering only the values of the deformation field at the mesh vertices. This approach eliminates the need for interpolating the field in the surrounding space, as the deformation primarily depends on the vertex properties.

To ensure robustness during mesh deformation, it is essential to define the maximum allowable displacement step, d_{move} , during surface evolution. This step size prevents a vertex from moving through a non-incident facet as the mesh evolves. To determine d_{move} , the minimum thickness, $d_{thickness}$, allowed by the volume bounded by the surface is first established. This $d_{thickness}$ denotes the minimum thickness that a quasi-uniform mesh can support, meaning the distance between two non-adjacent vertices cannot be less than $d_{thickness}$. For a quasi-uniform mesh M with a resolution d_{detail} and a thickness not exceeding $d_{thickness}$, the maximum displacement step d_{move} must satisfy the following inequality:

$$4d_{move}^2 \leq d_{thickness}^2 - d_{detail}^2/3$$

Additionally, to prevent edge inversion caused by the partial tangential displacement of vertices (which would result in the normals of neighboring triangles pointing inward), it is crucial to ensure that d_{move} is less than half of the minimum distance d_{min} between two neighboring vertices. Although d_{min} dynamically changes during mesh deformation, its trend closely follows the parameter d . Thus, a constant d_{move} is selected directly based on d rather than d_{min} in the system.

At each time step, the current deformation field influences the displacement of vertices in the selected region. If a single deformation exceeds the maximum allowed vertex displacement d_{move} , the initial deformation must be subdivided into basic deformation steps that conform to d_{move} to complete the overall deformation. The specific deformation process is related to brush operations, which in our work include Pull, Push, Flatten, and Smooth. The handle menu interface for these operations is shown in Figure 8.

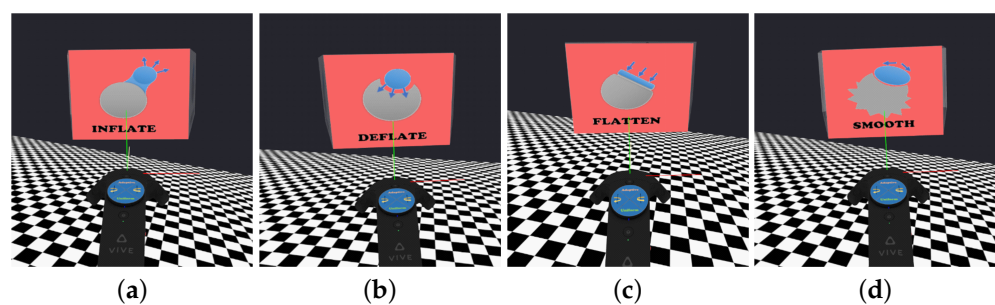


Figure 8. Operation menu for switching between sculpting modes. (a) Pull. (b) Push. (c) Flatten. (d) Smooth.

In this work, the deformation field is contingent on surface features such as normals and Laplace coordinates. Specifically, the deformation fields for the Pull, Push, and Flatten operations depend solely on the normals of each vertex, whereas the Smooth operation relies on the Laplace coordinates.

The deformation field expressions for Pull and Push are defined as

$$D(r_i) = \varepsilon \cdot b(r_i/R) \cdot N_i$$

Here, r_i represents the position of vertex i relative to the center of the spherical tool, N_i denotes the normal at vertex i , R is the radius of the tool, also known as the engraving radius, and ε takes the values $+1$ or -1 , corresponding to the Pull and Push operations, respectively. The function b is a decay function defined over the interval $[0, 1]$, which is expressed as

$$b(x) = (n-1)x^n - nx^{n-1} + 1$$

In this expression, the integer parameter $n > 2$ ensures that the first-order derivatives at $x = 0$ and 1 are zero, facilitating a smooth transition between the deformed and undeformed regions.

As illustrated in Figure 9b, during the Pull operation, each vertex moves in the direction of its normal, i.e., it is deformed along the positive direction. Conversely, the Push operation moves vertices in the opposite direction of their normals.

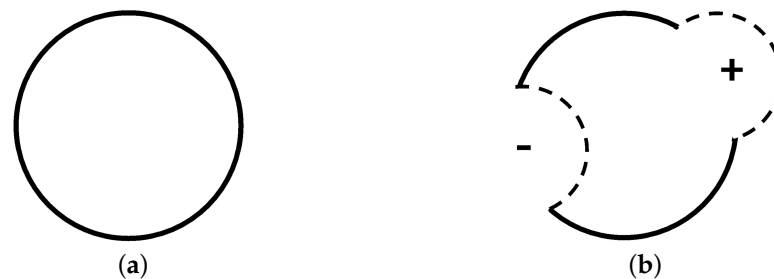


Figure 9. Pull and Push operations on spherical shape. (a) Original shape. (b) Pull (+) and Push (-).

In the Flatten operation, the mean normal N_{avg} and the center point P_{center} of the vertices in the region to be updated are first determined. Subsequently, the vertices to be updated are projected onto the plane defined by P_{center} and N_{avg} . This operation facilitates the sculpting of a platform-shaped mesh with relative ease.

The Smooth operation allows users to perform Laplacian mesh editing, specially through Laplacian smoothing. The principle of this algorithm is to reposition each vertex to the centroid of its neighboring vertices. As depicted in Figure 10, the yellow point represents the original position of a vertex, while the pink point indicates the average position of its neighboring nodes. The position of each vertex is recalculated using the following formula:

$$Laplacian(P) = \frac{1}{n} \sum_{i=0}^{n-1} Adj_i(P)$$

This method smooths the mesh by averaging the local vertex positions, thereby enhancing the geometric continuity of the surface.

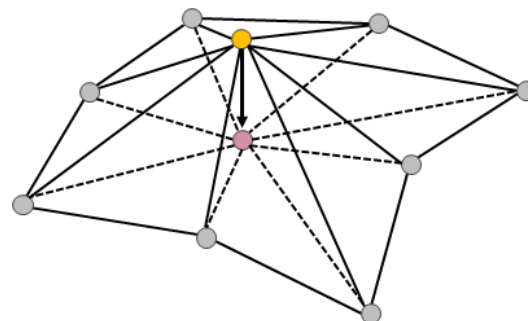


Figure 10. Position update in Laplacian smoothing. The gray points are the adjacent vertices of the yellow point, and the yellow point will be relocated to the position of the pink point.

Under a deformation field that satisfies the previous constraints, the deformation operation of our quasi-uniform mesh M is guaranteed to be disjoint, and the d_{detail} tight property is preserved.

For consistency in experimentation, the same initial model was used with a fixed sculpting radius. Various types of operations were performed on this model once each. The results of these operations are illustrated in Figure 11.

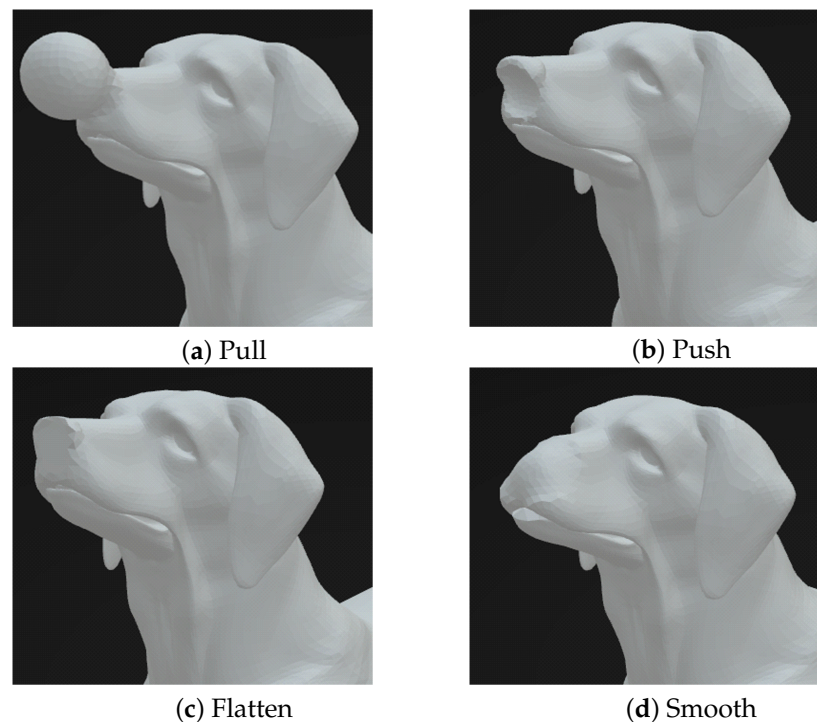


Figure 11. Different operations on the same part of the mesh. (a) shows the result obtained by the Pull operation, and (b) shows the result of the Push operation. While these two operations may appear similar to the union and difference operations in Boolean operations, the deformations are gradual processes that can be halted at any time to obtain intermediate results, unlike Boolean operations which directly yield final outcomes. (c) represents the result of the Flatten operation, and (d) shows the result of the Smooth operation.

3.4. Topology Fusion

If the process of modifying mesh vertices through simple topological operations results in over-thinning of the surface, it can lead to changes in the topological genus, as illustrated in Figure 12. In this work, potential self-intersections on the mesh surface are proactively detected by systematically sampling the evolving mesh and the corresponding pairwise deformation fields over time. This approach enables us to predict the changes in the topological genus and facilitate a seamless transition in the mesh topology without the need for computationally intensive triangle intersection calculations.

The topology fusion process is outlined as follows: before any potential local overlap of two non-adjacent surface parts occurs, a simple collision detection is conducted. This is achieved by centering a sphere with a diameter $d_{thickness}$ around the vertices and measuring the distance between the points in their final positions. If the distance between two non-adjacent vertices is less than $d_{thickness}$, a change in the topological genus occurs. This change is achieved through a vertex-joining operation, which eliminates closely positioned non-adjacent vertices and merges the remaining vertices.

As depicted in Figure 13, the merging of two vertex neighborhoods is facilitated by connecting their 1-rings. The Bresenham line algorithm [44] is utilized to connect the X vertices on the first 1-ring with the Y vertices on the second 1-ring, while potential triangle self-intersections are addressed through local smoothing. Following such a joining process, additional iterations are necessary to refine the newly formed mesh structure, ensuring the restoration of its quasi-uniform mesh properties.

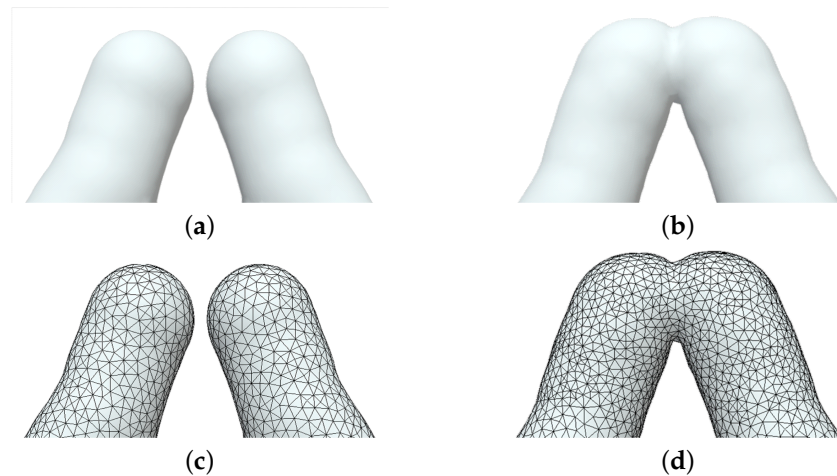


Figure 12. Topology fusion induced during the mesh deformation. (a) is the mesh before topology fusion, and (c) is the corresponding wireframe. (b) is the mesh before topology fusion, and (d) is the corresponding wireframe.

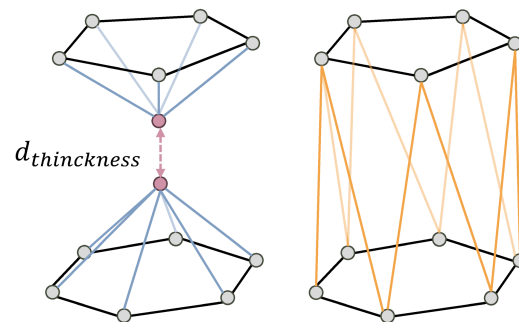


Figure 13. Merge the neighborhoods of two non-adjacent vertices if the distance between them is less than the threshold $d_{thickness}$.

3.5. More Specific Advantages

3.5.1. Free Topology

Our system supports free topology. In the spherical brush mode, the existing mesh can be deformed following the previous steps, or a new mesh model can be directly added in the mesh-free area. If no points on the surface of the mesh are selected in the previous step, a new spherical mesh will be automatically added in the mesh-free area.

To prevent the user from inadvertently adding a new sphere inside the original mesh, which could compromise the mesh's robustness, the positional legitimacy of the mesh needs to be verified before addition. The specific steps are as follows:

1. Ray Construction: Initialize from the center of the new spherical mesh, constructing a ray that extends in the positive z-axis direction.
2. Region Search: Utilize the ray to search within the octree of the original mesh and identify the triangles in the impacted regions as candidate triangles.
3. Triangle Traversal: Traverse through the candidate triangles and count how many are intersected by the ray.
4. Legitimacy Check: If the ray intersects an odd number of triangles, this indicates that the new spherical mesh will be added inside the original mesh; if even, the new sphere will be added outside.

After these steps, the legality of the new mesh's position can be confirmed. When the center point of the brush is outside the original mesh and no vertices are selected within the brush's radius, a spherical mesh will be added at the brush position. This new mesh becomes part of the overall mesh structure, and the octree is updated synchronously to reflect this addition.

Compared to existing virtual clay-based sculpting systems [39–41], our design allows users to freely alter the topology of the model. For instance, in the same scenario of sculpting a vase, existing systems can only add or remove elements from the base model. In contrast, our system enables the addition of new models beyond the original model and the connection of these new models with the original, offering greater freedom in sculpting.

3.5.2. Symmetrical Sculpting

A symmetry sculpting system function has been developed for sculpting tasks that require symmetry. This function considers the center point of the model as the origin and the x - z plane as the symmetry plane. We determine the symmetry point of the current sculpting start point and simultaneously carry out the processes of surface selection, mesh optimization, mesh deformation, and topology fusion in coordination with the original start point. This approach ensures that the sculpting effects are symmetrically mirrored across the designated plane, achieving a consistent and balanced aesthetic in the sculpted model.

3.5.3. Vertex Coloring

During the mesh optimization, the color of each new vertex is set to the average color of its two nearest neighbors. This approach helps maintain reasonably consistent color attributes during the sculpting process and facilitates automatic color filling. However, the need may arise to alter the color of the mesh during the creation of a virtual sculpture. To accommodate this, a coloring function has been designed that allows users to adjust the brush color using a palette. After selecting surface points, users can set the color of vertices within the selected region to match the brush's color, thus achieving manual coloring and enhancing creative flexibility.

3.5.4. Model Rotating Sculpting

During traditional sculpting, continuous surface textures are often created by rotating solids. Inspired by this technique, a model rotation function was developed in our sculpting system. The center point of the model is designated as the origin, and the corresponding Y -axis is used as the rotation axis. The model can be rotated around this axis in both clockwise and counterclockwise directions. Concurrently with the rotation, sculpting operations can still be performed, akin to piping frosting on a cake. This functionality allows for relatively uniform modifications to the model, enhancing both the versatility and the realism of the sculpting process.

4. Results and Discussion

4.1. User Interface

The HTC Vive is utilized as the experimental hardware platform for our system, which includes an HMD and two wireless handheld motion controllers. The HMD has a resolution of 2160×1200 (1080×1200 per eye) and a refresh rate of 90 Hz. Users can observe 3D models through the headset and use the controllers for input. Figure 14 illustrates the operation interface of our sculpting system. While it differs from the interface of traditional 2D sculpting software, our interface is designed to be intuitive and clear, enabling users to effortlessly learn how to sculpt three-dimensional models in a VR environment. This user-friendly design facilitates a smooth transition for artists and designers who may be new to virtual reality sculpting tools.

As illustrated in Figure 14a,c, the touchpad on the left controller is segmented into four functional areas. Pressing on the left or right sides of the touchpad results in clockwise or counterclockwise rotation of the model, respectively. Similarly, pressing on the top or bottom sides toggles between uniform and adaptive topology modes. Additionally, users can scale the model by touching the touchpad of the left controller to meet various visualization requirements.

The left-hand menu button toggles the display of the sculpting menu. When active, pressing the left or right sides of the touchpad allows users to scroll through different

sculpting modes, including Pull, Push, Flatten, Smooth, and Paint. Holding down the trigger on the left controller enables users to reposition the model to a desired location. Moreover, pressing the side button on the left controller facilitates saving the final model to a predefined directory.

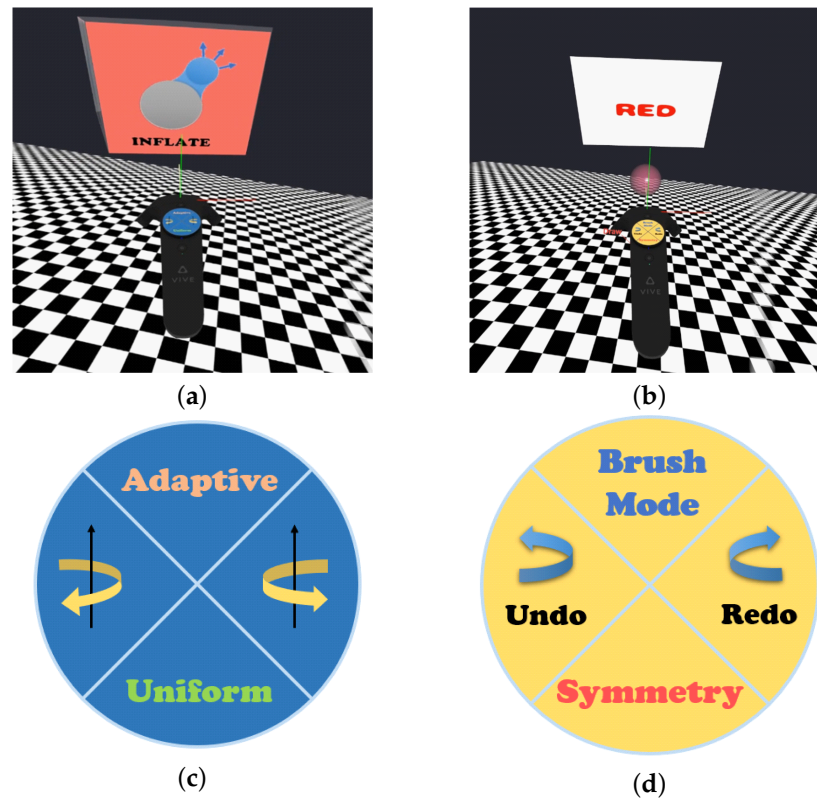


Figure 14. The interface of two controllers. (a) is the left-hand controller operation interface, and (b) is the right-hand controller operation interface. (c,d) denote the left-hand and right-hand touchpad partitions, respectively.

The functions associated with the right-hand controller are depicted in Figure 14b,d. The touchpad on the right-hand controller is divided into four functional zones. Pressing the upper half toggles between the spherical brush mode and ray-shaped brush mode, while pressing the lower half switches between symmetric and asymmetric modes. Pressing on the left side of the touchpad undoes the most recent mesh operation, and pressing on the right side redoes the action. Sliding on the touchpad adjusts the sculpting radius to accommodate different resolutions required for precise sculpting.

The right-hand menu button activates the color adjustment interface. When this interface is active, pressing the left or right sides of the touchpad selects the current color channel for adjustment. Sliding up or down on the touchpad adjusts the value of the selected color channel within the range of 0 to 255. This adjustment is crucial for applying the desired brush color to selected vertices. Holding the trigger on the right-hand controller executes the mesh operation corresponding to the current sculpting mode. In addition, pressing the side button on the right-hand controller clears all current operations and restores the model to its initial state.

4.2. System Usability

The experimental procedure was conducted on a computer equipped with a 2.5 GHz Intel i7-11700F CPU, 16 GB of RAM, and an NVIDIA RTX 3060Ti graphics card. The results of the experiment highlight the performance and visual quality of our work. A superior sculpting system must ensure that the model maintains a robust mesh structure throughout the sculpting process, requiring the resulting manifold mesh to be closed and free from

self-intersections. In our system, a series of algorithms have been designed specifically for the virtual sculpting. Each step in the mesh transformation is executed sequentially, adhering to the robustness requirements of the mesh. The importance of these steps will be demonstrated through the experimental results presented in the following sections.

Topological Auto-fusion: During the sculpting process, if the distance between two non-adjacent mesh points falls below a certain threshold, a change in topology becomes necessary. Topological auto-fusion is a critical step for achieving seamless topology changes and is essential for maintaining mesh quality. In this phase, mesh conditions with and without the implementation of topological auto-fusion mechanisms are compared. As shown in Figure 15a, the absence of topological auto-fusion leads to self-intersections within the triangular meshes. Figure 15b,c display the corresponding mesh wireframes of the model, revealing disordered surface triangles. Figure 15d illustrates the mesh after applying topological auto-fusion, showcasing well-integrated triangles without any self-intersections. Upon examining the mesh wireframe in Figure 15e,f, it becomes evident that the post-fusion mesh maintains relatively uniform characteristics, resulting in a smooth topological transformation.

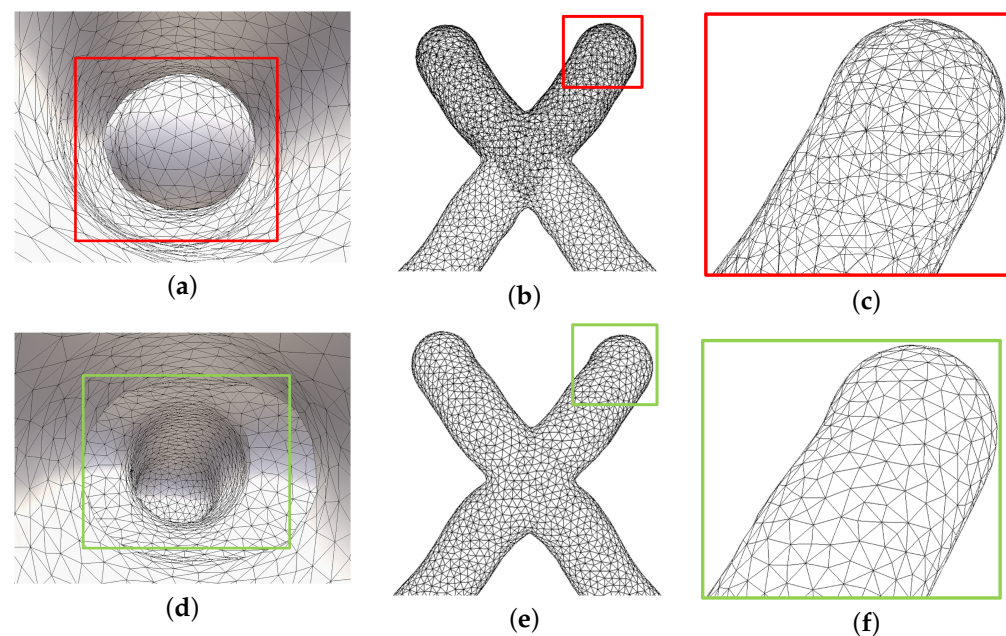


Figure 15. The case of mesh with and without the topological auto-fusion mechanism. (a) is the case where intersections appear as can be seen from the inside of the mesh without the topology auto-fusion mechanism. We can see that the outer mesh appears misaligned in (b), and the zoomed-in case can be seen in (c). Correspondingly, (d) corresponds to the case of the interior of the mesh with the topological auto-fusion mechanism. As can be seen in (e), the outer mesh maintains good properties, and the zoomed-in case can be seen in (f).

Position Reasonability Inspection: When adding new spherical models to a mesh-free region, defining the addition of a new model inside the original model as an illegal operation is crucial. Conversely, adding a model outside the original model is considered a legal operation. Therefore, inspecting the rationality of the position where a model is to be added is an indispensable step. The impact of including or omitting this inspection phase on the mesh was evaluated. In the absence of this inspection phase, the default spherical model could be added both inside and outside the original model, as shown in Figure 16a. Adding a new model inside the original model can lead to surface breakage, compromising the robustness of the original mesh, as depicted in Figure 16b. By implementing the rationality inspection phase, illegal additions inside the original model are prevented, while the addition of new models outside the original model remains unaffected, as demonstrated in

Figure 16c,d. Ensuring only legal additions helps prevent surface breakage and effectively maintains the robustness of the mesh.

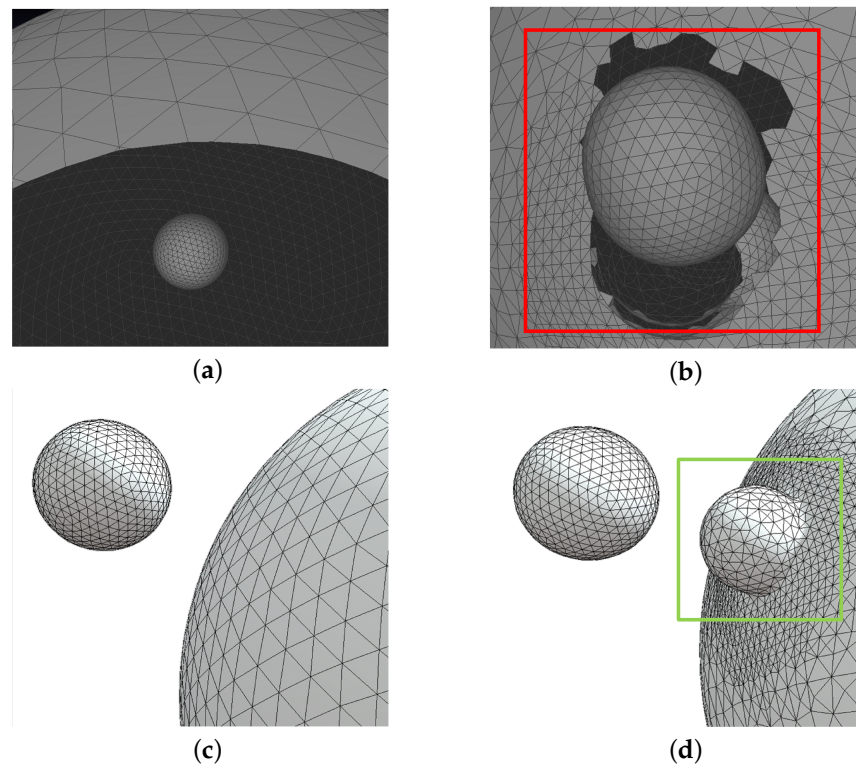


Figure 16. Illegal and legal model adding operations. (a) presents the illegal operation, and the resulting broken mesh is shown in (b). (c) shows the legal operation, and it does not affect the subsequent sculpting as shown in (d).

Real-Time Interactivity: Beyond maintaining the robustness of the mesh during the sculpting process, the real-time responsiveness of interactions within a VR environment is crucial, as it significantly enhances the user's immersive experience. The experiments on a spherical model comprising 2562 vertices were conducted with a constant sculpting radius throughout the experiment. Table 1 presents the time taken to perform a single operation under various sculpting modes. Due to slight variations in the underlying implementations of different brush modes, the duration of operations varies. For instance, the ray-shaped brush mode involves calculating geodesic distances for point selection, which leads to longer execution times compared to the spherical brush mode. Moreover, a complete operation on the mesh includes additional processing steps, resulting in a total operation time that exceeds the sum of the durations of the initial components. The overall time taken for a single operation demonstrates that the execution time for a regular mesh operation is swift in our system, thus ensuring excellent real-time responsiveness to user inputs.

Table 1. Time consumed in each phase of an operation. We take the average of 20 operations as results.

Brush Mode	Sculpting Mode	Points Selection	Mesh Optimization	Mesh Deformation	Total
Spherical Brush	Pull	0.092 ms	0.133 ms	0.0019 ms	3.944 ms
	Push	0.071 ms	0.101 ms	0.0019 ms	3.993 ms
	Flatten	0.068 ms	0.022 ms	0.0028 ms	3.643 ms
	Smooth	0.075 ms	0.028 ms	0.0054 ms	3.782 ms
Ray-Shaped Brush	Pull	0.796 ms	0.085 ms	0.0017 ms	4.737 ms
	Push	0.869 ms	0.086 ms	0.0016 ms	4.692 ms
	Flatten	1.009 ms	0.029 ms	0.0028 ms	4.897 ms
	Smooth	0.897 ms	0.041 ms	0.0062 ms	4.732 ms

User Study: The goal of the user study is to compare the usability of the proposed system with an existing commercial virtual reality modeling system. We used Shapelab [45] as a representative of commercial systems. Both systems run on HTC Vive virtual reality devices and have similar operations.

Fifteen participants were invited to participate in the study, including nine males and six females, aged between 22 and 40 years (mean = 25.07, std = 4.25). Four of the participants had previous experience with VR systems (26.7%). Considering that many subjects were new to virtual reality, each subject received a quick VR training session before the study began to mitigate the influence of “wow factors” on subjective perceptions. Following the training, participants were tasked with completing two tasks. The order of the tasks was counterbalanced among the subjects.

Task 1 The participants were required to freely explore the functions of the proposed system within 10 min and create their own models with creativity.

Task 2 The participants were required to freely explore the functions of the Shapelab within 10 min and create their own models with creativity.

To evaluate the various performance aspects of the two systems, relevant indicators of subjective usability judgments were collected, encompassing six dimensions: system ease of use, functional completeness, modeling robustness, detail implementation, topological freedom, and overall evaluation. A tailored usability questionnaire was designed targeting these six dimensions, consisting of seven-point Likert-scale items (see Appendix A).

The results of the user study are presented in Table 2 and illustrated in Figure 17. From the results, it can be observed that in terms of system ease of use, the proposed system scored slightly higher than Shapelab, indicating that under free exploration conditions, the proposed system is more user-friendly for novices and its functions are more intuitive. However, the integrated functions are relatively basic and general. Compared to Shapelab, the proposed system lacks some detailed sculpting tools for different shapes, resulting in lower scores in terms of functional completeness and detail implementation. Furthermore, the scores of the two systems are similar in terms of modeling robustness and topological freedom, suggesting that the proposed system can maintain good model robustness during the sculpting. Benefiting from the functional design of free topology, users can freely add new models in space, enjoying a high degree of freedom. Finally, in the overall evaluation, our score is lower than that of Shapelab, which is a reasonable outcome. This indicates that while the proposed system can basically meet some basic sculpting requirements, there is still much improvement for further study.

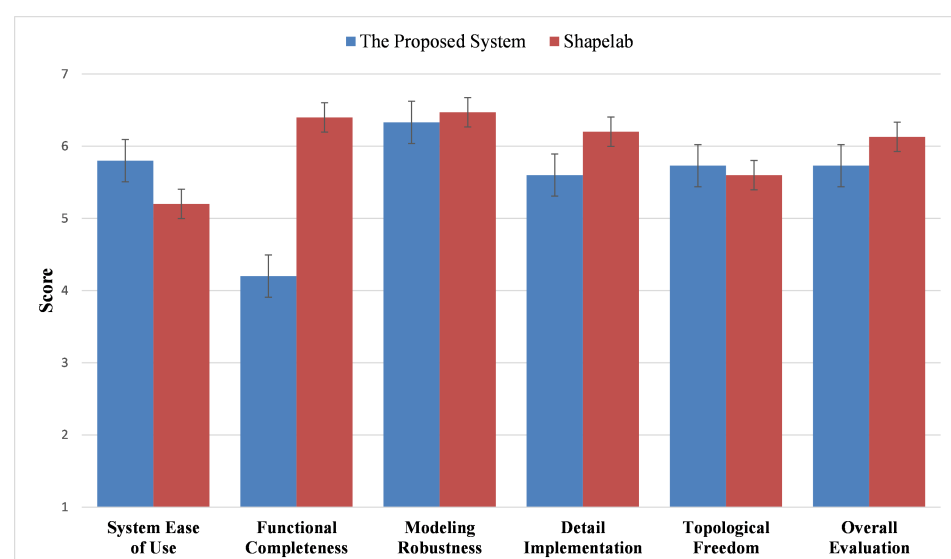


Figure 17. Scores of the two compared systems in six evaluation dimensions, including system ease of use, functional completeness, modeling robustness, detail implementation, topological freedom, and overall evaluation.

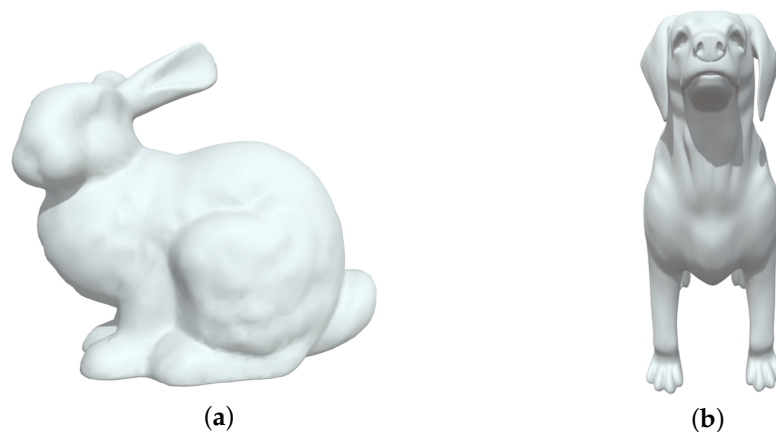
Table 2. Average and standard deviation of user study (upper row: average, bottom row: standard deviation).

System	System Ease of Use	Functional Completeness	Modeling Robustness	Detail Implementation	Topological Freedom	Overall Evaluation
The Proposed	5.80	4.20	6.33	5.60	5.73	5.73
	0.86	0.77	0.49	0.63	0.46	0.59
Shapelab	5.20	6.40	6.47	6.20	5.67	6.13
	1.08	0.63	0.52	0.56	0.72	0.35

4.3. Created Models

Finally, we showcased models with complex topological structures created using the VR sculpting application described in this paper, as illustrated in Figures 18–20. The number of vertices in these models ranges from 10,000 to 100,000, and the time taken to sculpt them varies from 15 min to 1 h. Specifically, the model depicted in Figure 18 was crafted based on an imported model; the model in Figure 19 was created by the user from scratch. Additionally, as illustrated in Figure 20, the user can also freely apply colors to the models, enhancing their visual appeal and realism.

Additionally, Figure 21 presents the complete process of sculpting a beautiful vase. Fundamental operations such as Pull, Push, and Flatten were utilized to construct the main body of the vase, thereby establishing its basic shape. Subsequently, the Pull operation was employed while rotating the model to skillfully create a suspended ring, and through the Push operation, unique ornamental holes were added to this ring. To enhance the visual impact of the vase, the Pull technique was further applied to delicately create decorative patterns connecting the ring to the main body of the vase. Thereafter, elegant handles were designed and sculpted on both sides of the vase. The addition of these details not only elevated the aesthetic appeal but also enhanced the practical value of the vase. Subsequently, undulating decorative motifs were created along the outer wall of the vase by rotating the model and using the Pull operation. Finally, to ensure the overall beauty of the vase, the Pull technique was employed to meticulously sculpt symmetrical textures on its surface. The entire sculpting process was carried out in a symmetric pattern, and extensive use was made of Smooth operations to ensure the smoothness of the model.

**Figure 18.** Cont.

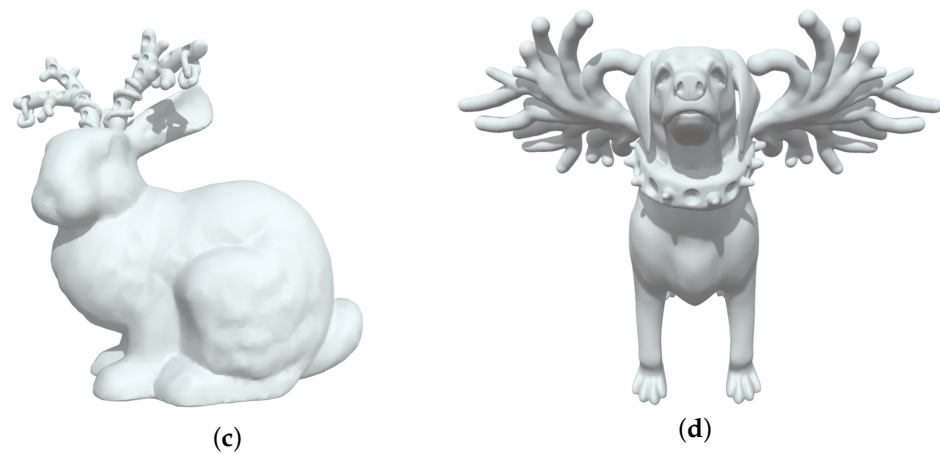


Figure 18. Models created by novice users based on existing models. (a,b) are origin models. (c) is a rabbit with horns based on (a), and (d) is a dog with wings and collar based on (b).

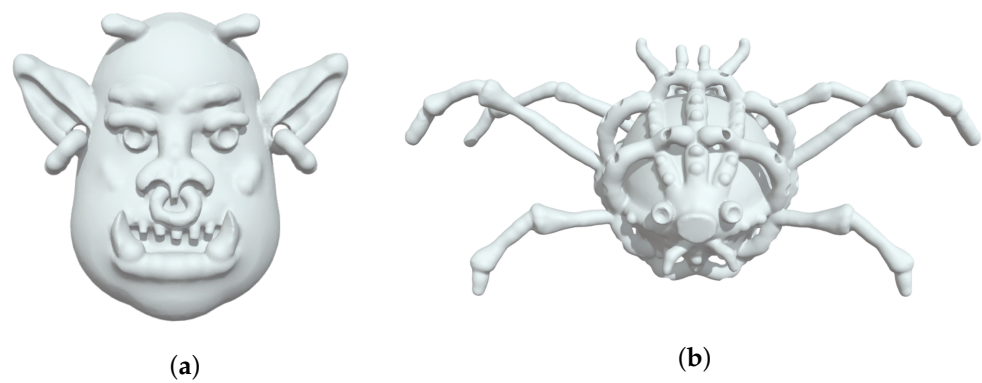


Figure 19. Models created by novice users from scratch. (a) is a monster head, and (b) is a fantasy arthropod.

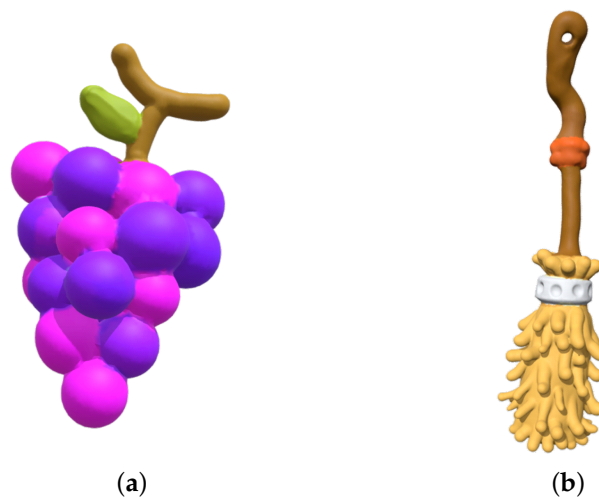


Figure 20. Sculpted colored models. (a) is a bunch of grapes, and (b) is a magic broom.

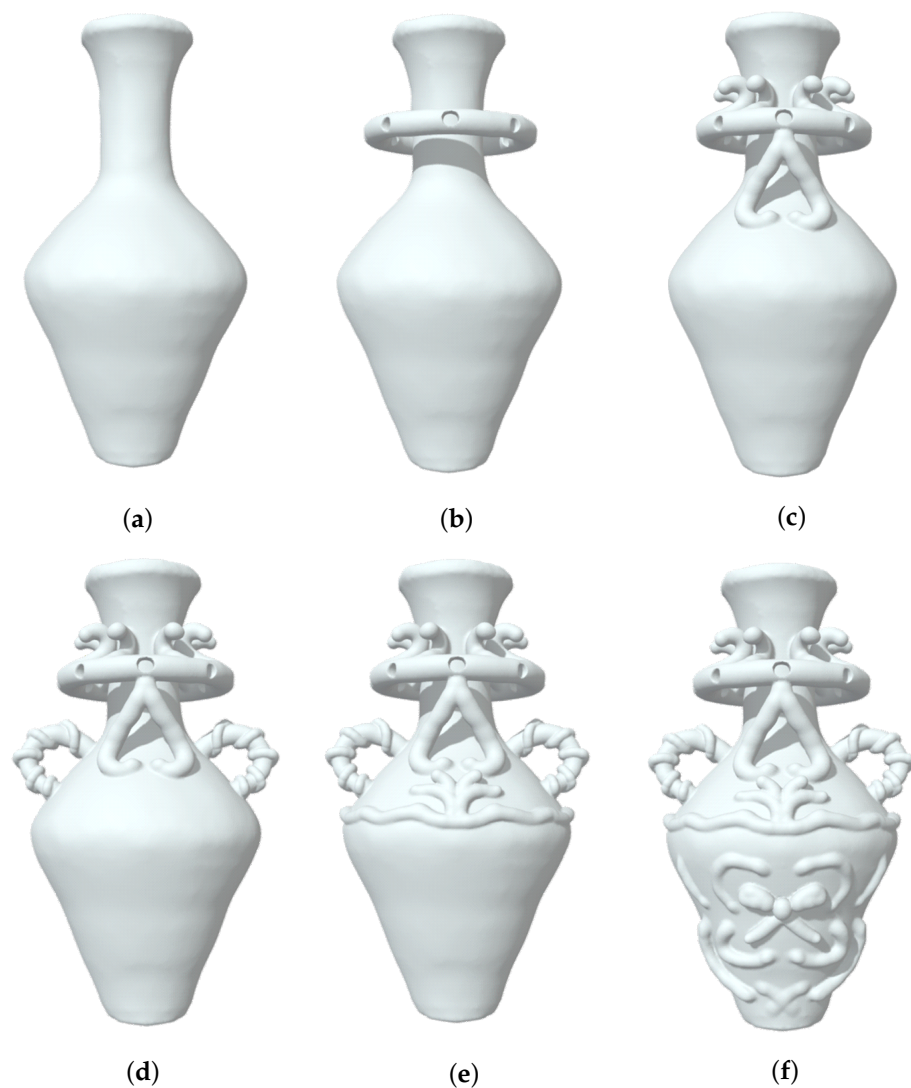


Figure 21. Sculpting sequence of a vase model. (a) is the basic shape of the model. A ring is created with holes as shown in (b), and the connection between the ring and the model's body is established in (c). In (d), the handles are added, and then we add the undulating motifs around the surface in (e). Finally, we sculpt some symmetrical textures for the visual impact as shown in (f).

The model created by this system possesses the attributes of being closed, manifold, and free from self-intersection, making it versatile for applications such as animation design, game character modeling, and 3D printing, depending on specific requirements. This versatility underscores its high level of freedom and entertainment potential. Figure 22 demonstrates a model printed using 3D printing technology. The model possesses high-quality characteristics, eliminating the need for corrections or similar operations. It can be printed directly after adjusting to the appropriate size according to specific requirements.

In the experimental results presented, all models were created by a novice without formal training. These results demonstrate that our sculpting system is exceptionally user-friendly for beginners. Users can quickly familiarize themselves with the functions of different buttons on the controller and freely express their creativity using customized initial models. Additionally, within the immersive digital environment of VR, users can observe a 360-degree view of the model at any time. This feature significantly enhances the user's perception of the position and distance of sculpting, making it more convenient to adjust model details.



(a)



(b)

Figure 22. Three-dimensional (3D) printing model entities. (a) is the entity of vase, and (b) is the entity of fantasy arthropod.

In summary, the strengths of this system lie in its robust sculpting, high-quality mesh generation, and simple operation for model creation, allowing even novices to complete their own designed models, thus offering strong entertainment value. However, the system has limitations in that it relies on spherical-based sculpting models with a single brush shape and lacks a pinching sculpting mode, making it difficult to create sharp features.

5. Conclusions and Future Work

This paper introduces a virtual sculpting system specially designed for the virtual reality environment. The system incorporates an integrated toolkit that includes tools such as Pull, Push, Flatten, Smooth and Paint, allowing users to easily sculpt their desired models. The user-centric sculpting interface is intuitive and friendly, reducing learning costs. This paper applies the quasi-uniform mesh to virtual reality sculpting and implements a series of well-designed algorithms to achieve arbitrary topological deformation in VR settings. These enhancements ensure the continuous robustness of the mesh structure.

Overall, our sculpting system provides a deep sense of immersion, allowing users to directly edit and observe models in three-dimensional space. The models created by sculpting are universal and can be used in various fields such as 3D printing, animation and games. The experimental results further emphasize the ease of use of the system for beginners and verify the efficacy and accessibility.

However, our current work still has some limitations that we plan to address in future developments. Currently, the update of surface vertices is based on a spherical shape, and we intend to support the importation of custom models in the future, using the shapes of these models as the basis for editing the mesh. Additionally, we plan to replace the default spherical model used in mesh-free regions with imported models. The controller currently implemented in our system is the handle that accompanies the HTC Vive virtual reality device. This controller lacks tactile feedback compared to an authentic sculpting process, thus compromising the sense of realism. In future work, we will actively experiment with the integration of virtual reality devices and haptic gloves. This combination will enable digital sculpting through the natural movement of hand joints while providing real-time feedback on the pressure applied to the hands. Using haptic gloves allows users to feel as if they are interacting with a real object, ultimately resulting in more intuitive sculpting.

Furthermore, with recent advancements in artificial intelligence, there have been numerous studies on generative modeling in the field of modeling, such as those by Poole et al. (2022) [46], Jambon et al. (2023) [47], Ding et al. (2023) [48], Lin et al. (2023) [49], He (2024) [50], and Wang (2023) [51]. These works typically generate three-dimensional models based on semantic descriptions or images, although the resulting models may not always fully align with expectations. Users might wish to adjust specific parts of these models, and since our system can handle initial models in any format, we are considering integrating AI modeling technology in the future to enhance the downstream processes of generative modeling. Our goal is to develop a comprehensive system within the VR environment that facilitates the complete flow from model generation to model adjustment.

Author Contributions: Conceptualization, X.Z.; methodology, X.Z. and Y.Y.; software, X.Z. and Y.Y.; validation, X.Z. and Y.Y.; formal analysis, X.Z. and Y.Y.; investigation, Y.Y.; resources, X.Z.; data curation, Y.Y.; writing—original draft preparation, X.Z. and Y.Y.; writing—review and editing, X.Z. and Y.Y.; visualization, Y.Y.; supervision, X.Z.; project administration, X.Z.; funding acquisition, X.Z.; All authors have read and agreed to the published version of the manuscript.

Funding: This research is Supported by the Open Project Program of the State Key Laboratory of CAD & CG (Grant No. A2422), Zhejiang University.

Data Availability Statement: The code presented in this study is available on request from the corresponding author.

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

Abbreviations

The following abbreviations are used in this manuscript:

XR	Extended Reality
AR	Augmented Reality
VR	Virtual Reality
MR	Mixed Reality
3D	three-dimensional
2D	two-dimensional
NURBS	Non-Uniform Rational B-Splines
CAD	computer-aided design
CPU	Central Processing Unit

Appendix A. Usability Questionnaire

1. Rate how easy it was for you to master the use of the system (1 = extremely difficult, 7 = extremely easy).
2. Rate the completeness of the design functions in the system (1 = extremely incomplete, 7 = extremely complete).
3. Rate the frequency of errors occurring during the sculpting process (1 = extremely high frequency, 7 = extremely low frequency).

4. Rate how convenient it was for you to implement the model details (1 = extremely inconvenient, 7 = extremely convenient).
5. Rate how freely you can alter the model's topology (1 = extremely restricted, 7 = extremely free).
6. Rate the overall system (1 = extremely poor, 7 = extremely good).

References

1. Keefe, D.; Zeleznik, R.; Laidlaw, D. Drawing on air: Input techniques for controlled 3D line illustration. *IEEE Trans. Vis. Comput. Graph.* **2007**, *13*, 1067–1081. [\[CrossRef\]](#) [\[PubMed\]](#)
2. Maxim, B.; Gorgan, D.; Moldoveanu, A.; Dix, A. ArtworkVR: Novel Interaction Techniques for Virtual Painter. In Proceedings of the RoCHI, Bucharest, Romania, 17–18 October 2019; pp. 30–37.
3. Schkolne, S.; Pruett, M.; Schröder, P. Surface drawing: Creating organic 3D shapes with the hand and tangible tools. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Seattle, WA, USA, 31 March–5 April 2001; pp. 261–268.
4. Medium. Available online: <https://www.oculus.com/medium/> (accessed on 25 April 2024).
5. Tilt Brush. Available online: <https://www.tiltbrush.com/> (accessed on 25 April 2024).
6. Vinod, T.; Kumaresan, N.; Guban, I.; Dhanasekaran, S.; Ramprathap, K.; Chinnasamy, P. Online Automobile Rental and E-Marketplace with Augmented Reality (AR). In Proceedings of the 2022 International Conference on Advancements in Smart, Secure and Intelligent Computing (ASSIC), Bhubaneswar, India, 19–20 November 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–5.
7. Geiger, A.; Ziegler, J.; Stiller, C. Stereoscan: Dense 3d reconstruction in real-time. In Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, 5–9 June 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 963–968.
8. Izadi, S.; Kim, D.; Hilliges, O.; Molyneaux, D.; Newcombe, R.; Kohli, P.; Shotton, J.; Hodges, S.; Freeman, D.; Davison, A.; et al. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, Santa Barbara, CA, USA, 16–19 October 2011; pp. 559–568.
9. Mouragnon, E.; Lhuillier, M.; Dhome, M.; Dekeyser, F.; Sayd, P. Real time localization and 3d reconstruction. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 17–22 June 2006; IEEE: Piscataway, NJ, USA, 2006; Volume 1, pp. 363–370.
10. Davies, R.H.; Cootes, T.F.; Twining, C.J.; Taylor, C.J. An Information Theoretic Approach to Statistical Shape Modelling. In Proceedings of the BMVC, Manchester, UK, 10–13 September 2001; Citeseer: Princeton, NJ, USA, 2001; pp. 1–10.
11. Wang, H.; Markosian, L. Free-form sketch. In Proceedings of the 4th Eurographics Workshop on Sketch-Based Interfaces and Modeling, Riverside, CA, USA, 2–3 August 2007; pp. 53–58.
12. Yu, E.; Arora, R.; Baerentzen, J.A.; Singh, K.; Bousseau, A. Piecewise-smooth surface fitting onto unstructured 3D sketches. *ACM Trans. Graph. (TOG)* **2022**, *41*, 1–16. [\[CrossRef\]](#)
13. Mikaeili, A.; Perel, O.; Safaee, M.; Cohen-Or, D.; Mahdavi-Amiri, A. Sked: Sketch-guided text-based 3d editing. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–6 October 2023; pp. 14607–14619.
14. Lamousin, H.J.; Waggenspack, N. NURBS-based free-form deformations. *IEEE Comput. Graph. Appl.* **1994**, *14*, 59–65. [\[CrossRef\]](#)
15. Wong, J.P.; Lau, R.W.; Ma, L. Virtual 3d sculpting. *J. Vis. Comput. Animat.* **2000**, *11*, 155–166. [\[CrossRef\]](#)
16. McDonnell, K.T.; Qin, H.; Wlodarczyk, R.A. Virtual clay: A real-time sculpting system with haptic toolkits. In Proceedings of the 2001 Symposium on Interactive 3D Graphics, Chapel Hill, NC, USA, 26–29 March 2001; pp. 179–190.
17. Keefe, D.F.; Feliz, D.A.; Moscovich, T.; Laidlaw, D.H.; LaViola, J.J., Jr. CavePainting: A fully immersive 3D artistic medium and interactive experience. In Proceedings of the 2001 Symposium on Interactive 3D Graphics, Chapel Hill, NC, USA, 26–29 March 2001; pp. 85–93.
18. Solidworks. Available online: <https://www.solidworks.com/> (accessed on 25 April 2024).
19. Inventor. Available online: <https://www.autodesk.com/products/inventor/overview> (accessed on 25 April 2024).
20. Rhino. Available online: <https://www.rhino3d.com> (accessed on 25 April 2024).
21. Maya. Available online: <https://www.autodesk.com.tw/products/maya/overview> (accessed on 25 April 2024).
22. 3ds Max. Available online: <https://www.autodesk.com.cn/products/3ds-max/overview> (accessed on 11 June 2024).
23. Blender. Available online: <https://www.blender.org/> (accessed on 11 June 2024).
24. Cinema 4D. Available online: <https://www.maxon.net/cinema-4d> (accessed on 25 April 2024).
25. Stanculescu, L.; Chaine, R.; Cani, M.P. Freestyle: Sculpting meshes with self-adaptive topology. *Comput. Graph.* **2011**, *35*, 614–622. [\[CrossRef\]](#)
26. Galyean, T.A.; Hughes, J.F. Sculpting: An interactive volumetric modeling technique. *ACM SIGGRAPH Comput. Graph.* **1991**, *25*, 267–274. [\[CrossRef\]](#)
27. Lorensen, W.E.; Cline, H.E. Marching cubes: A high resolution 3D surface construction algorithm. In *Seminal Graphics: Pioneering Efforts That Shaped the Field*; Association for Computing Machinery: New York, NY, USA, 1998; pp. 347–353.
28. Ferley, E.; Cani, M.P.; Gascuel, J.D. Practical volumetric sculpting. *Vis. Comput.* **2000**, *16*, 469–480. [\[CrossRef\]](#)
29. Perng, K.L.; Wang, W.T.; Flanagan, M.; Ouhyoung, M. A real-time 3D virtual sculpting tool based on modified marching cubes. In Proceedings of the International Conference on Artificial Reality and Teleexistence, Quito, Ecuador, 27–29 October 2001; pp. 64–72.

30. Ferley, E.; Cani, M.P.; Gascuel, J.D. Resolution adaptive volume sculpting. *Graph. Models* **2001**, *63*, 459–478. [CrossRef]
31. Chen, C.W.; Hu, M.C.; Chu, W.T.; Chen, J.C. A real-time sculpting and terrain generation system for interactive content creation. *IEEE Access* **2021**, *9*, 114914–114928. [CrossRef]
32. ZBrush. Available online: <https://pixologic.com/> (accessed on 25 April 2024).
33. Geomagic Freeform. Available online: <https://www.artec3d.cn/3d-software/geomagic-freeform> (accessed on 11 June 2024).
34. De Goes, F.; James, D.L. Regularized kelinlets: Sculpting brushes based on fundamental solutions of elasticity. *ACM Trans. Graph. (TOG)* **2017**, *36*, 1–11. [CrossRef]
35. Jang, S.A.; Kim, H.i.; Woo, W.; Wakefield, G. Airsculpt: A wearable augmented reality 3d sculpting system. In *Proceedings of the Distributed, Ambient, and Pervasive Interactions: Second International Conference, DAPI 2014, Held as Part of HCI International 2014, Heraklion, Crete, Greece, 22–27 June 2014. Proceedings 2*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 130–141.
36. Lu, P.; Sheng, B.; Luo, S.; Jia, X.; Wu, W. Image-based non-photorealistic rendering for realtime virtual sculpting. *Multimed. Tools Appl.* **2015**, *74*, 9697–9714. [CrossRef]
37. Callens, E.; Danieau, F.; Costes, A.; Guillotel, P. A tangible surface for digital sculpting in virtual environments. In *Proceedings of the Haptics: Science, Technology, and Applications: 11th International Conference, EuroHaptics 2018, Pisa, Italy, 13–16 June 2018, Proceedings, Part II 11*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 157–168.
38. Milliez, A.; Wand, M.; Cani, M.P.; Seidel, H.P. Mutable elastic models for sculpting structured shapes. *Comput. Graph. Forum* **2013**, *32*, 21–30. [CrossRef]
39. Gao, Z.; Li, J.; Wang, H.; Feng, G. Digiclay: An interactive installation for virtual pottery using motion sensing technology. In *Proceedings of the 4th International Conference on Virtual Reality, Qingdao, China, 22–24 October 2018*; pp. 126–132.
40. Gao, Z.; Wang, H.; Feng, G.; Guo, F.; Lv, H.; Li, B. RealPot: An immersive virtual pottery system with handheld haptic devices. *Multimed. Tools Appl.* **2019**, *78*, 26569–26596. [CrossRef]
41. Dashti, S.; Prakash, E.; Navarro-Newball, A.A.; Hussain, F.; Carroll, F. PotteryVR: Virtual reality pottery. *Vis. Comput.* **2022**, *38*, 4035–4055. [CrossRef]
42. Zeidler, D.; McGinity, M. Bodylab: In virtuo sculpting, painting and performing of full-body avatars. *Proc. ACM Comput. Graph. Interact. Tech.* **2023**, *6*, 1–12. [CrossRef]
43. Xiaoqiang, Z.; Tao, Y. Interactive sculpture modeling based on mesh deformation in HMD VR environment. *J. Zhejiang Univ. (Eng. Sci.)* **2018**, *52*, 599–604.
44. Bresenham, J.E. Algorithm for computer control of a digital plotter. In *Seminal Graphics: Pioneering Efforts That Shaped the Field*; Association for Computing Machinery: New York, NY, USA, 1998; pp. 1–6.
45. Shapelab. Available online: <https://shapelabvr.com> (accessed on 2 July 2024).
46. Poole, B.; Jain, A.; Barron, J.T.; Mildenhall, B. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv* **2022**, arXiv:2209.14988.
47. Jambon, C.; Kerbl, B.; Kopanas, G.; Diolatzis, S.; Leimkühler, T.; Drettakis, G. Nerfshop: Interactive editing of neural radiance fields. *Proc. ACM Comput. Graph. Interact. Tech.* **2023**, *6*, 1–21. [CrossRef]
48. Ding, L.; Dong, S.; Huang, Z.; Wang, Z.; Zhang, Y.; Gong, K.; Xu, D.; Xue, T. Text-to-3D Generation with Bidirectional Diffusion using both 2D and 3D priors. *arXiv* **2023**, arXiv:2312.04963.
49. Lin, C.H.; Gao, J.; Tang, L.; Takikawa, T.; Zeng, X.; Huang, X.; Kreis, K.; Fidler, S.; Liu, M.Y.; Lin, T.Y. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023*; pp. 300–309.
50. He, Y.; Bai, Y.; Lin, M.; Sheng, J.; Hu, Y.; Wang, Q.; Wen, Y.H.; Liu, Y.J. Text-Image Conditioned Diffusion for Consistent Text-to-3D Generation. *Comput. Aided Geom. Des.* **2024**, *111*, 102292. [CrossRef]
51. Wang, Z.; Lu, C.; Wang, Y.; Bao, F.; Li, C.; Su, H.; Zhu, J. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 8406–8441.

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.