CGI2024 manuscript No.
(will be inserted by the editor)

FusionDeformer: Text-guided Mesh Deformation using Diffusion Models

Hao Xu · Yiqian Wu · Xiangjun Tang · Jing Zhang · Yang Zhang · Zhebin Zhang · Chen Li · Xiaogang Jin*

Abstract Mesh deformation has a wide range of applications, including character creation, geometry modeling, deforming animation, and morphing. Recently, mesh deformation methods based on CLIP models demonstrated the ability to perform automatic text-guided mesh deformation. However, using 2D guidance to deform a 3D mesh attempts to solve an ill-posed problem and leads to distortion and unsmoothness, which cannot be eliminated by CLIP-based methods because they focus on semantic-aware features and cannot identify these artifacts. To this end, we propose FusionDeformer, a novel automatic text-guided mesh deformation method that leverages diffusion models. The deformation is achieved by Score Distillation Sampling (SDS), which minimizes the KL-divergence between the distribution of rendered deformed mesh and the text-conditioned distribution. To alleviate the intrinsic ill-posed problem, we incorporate two approaches into our framework. The first approach involves combining multiple orthogonal views into a single image, providing robust deformation while avoiding the need for additional memory. The second approach incorporates a new regularization to address the unsmooth artifacts. Our experimental results show that the proposed method can generate high-quality, smoothly deformed meshes that align precisely with the input text description while preserving the topological relationships. Additionally, our method offers a text2morphing approach to animation design, enabling common users to produce special effects animation.

Keywords Diffusion Model · Mesh Deformation · Score Distillation Sampling

Hao Xu
State Key Lab of CAD&CG, Zhejiang University

Yiqian Wu
State Key Lab of CAD&CG, Zhejiang University

Xiangjun Tang
State Key Lab of CAD&CG, Zhejiang University

Jing Zhang
State Key Lab of CAD&CG, Zhejiang University

Yang Zhang
College of Computer Science and Technology, Zhejiang University

Zhebin Zhang
OPPO US Research Center, United States

Chen Li
OPPO US Research Center, United States

Xiaogang Jin (jin@cad.zju.edu.cn)
State Key Lab of CAD&CG, Zhejiang University

1 Introduction

Mesh is the most prevalent representation for 3D models and is universally compatible with the majority of graphic hardware systems to facilitate accelerated rendering. Mesh deformation, which changes the shape of a mesh without modifying its topology or the number of vertices, edges, and faces, is a valuable technique with extensive applications in geometric modeling, content creation [8], character posing [17], and morphing [23].

Traditional mesh deformation methods [41] demand considerable human intervention to yield results that align with human preferences. To counter this, automatic mesh deformation techniques are introduced to reduce the requirement for manual labor. To further effectively steer the results of this automatic process, and leverage the simplicity and intuitiveness of textual instructions, newly developed techniques [21, 29, 9] incorporate text guidance into deformation by employing the Contrastive Language-Image Pre-training (CLIP) model [35]. These methods aim to minimize the disparity between the CLIP image embedding (pre-training to

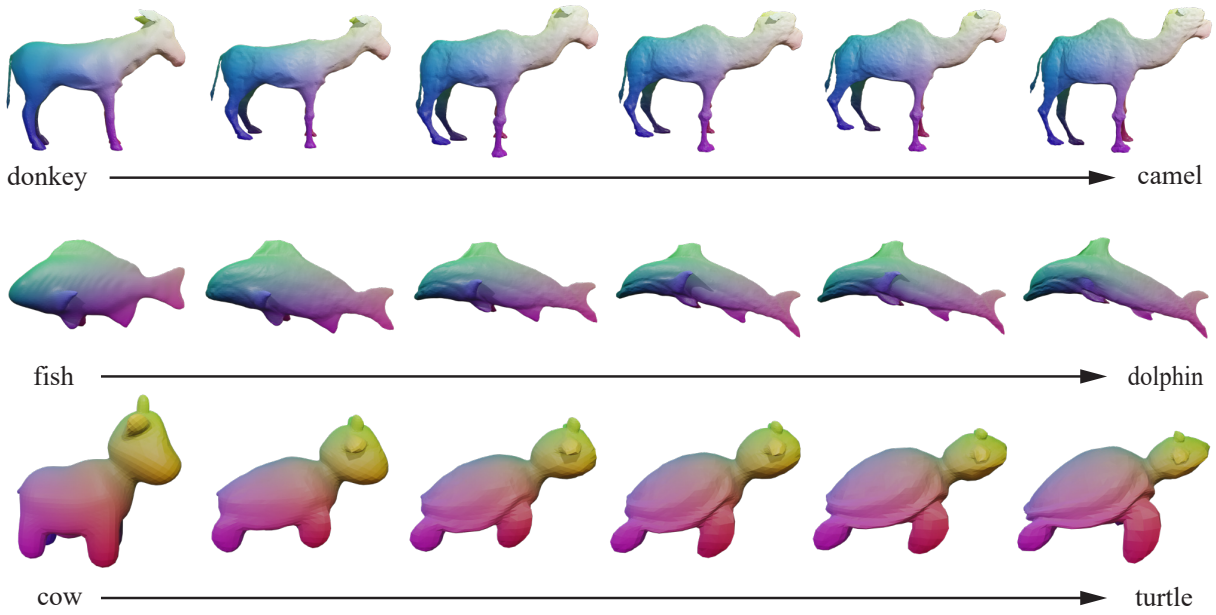

Fig. 1 Our proposed pipeline performs text-guided deformation, initiating with a source mesh, and generating a high-quality target mesh that aligns with a given text prompt. The resulting mesh preserves the topological relationships amongst the vertices, thereby facilitating a seamless transformation from the source mesh and making our approach suited for mesh morphing applications.

the rendered mesh results) and the CLIP text embedding (associated with the input text description), effectively producing results that comply with the text description while also preserving the semantic attributes of the input geometry. For example, vertices associated with a particular semantic attribute will uphold this attribute in the deformed result.

However, the above-mentioned methods attempt to deform a 3D mesh by constraining its rendered results within a specific 2D space, which attempts to solve an ill-posed problem and leads to undesired deformations such as distortions or compromises in geometric quality, such as a lack of surface smoothness. In addition, even though CLIP-based methods incorporate rendering results from various camera poses in a batch to achieve multi-view guidance during optimization, they utilize CLIP-based embeddings which focus on semantic-aware features and overlook the fine-grained details of the images, hence they cannot identify distortions. Therefore, these methods cannot effectively eliminate these artifacts and also demonstrate limited expressiveness, which is validated by our experiments. Unlike the CLIP model, diffusion models [13,36] offer a comprehensive and direct mapping from the text space to the image space. These models exhibit exceptional capabilities in generating two-dimensional images, suggesting a promising solution to the constraints in expressiveness inherent in CLIP-guided deformations.

Motivated by the success of diffusion models, we propose FusionDeformer, a novel diffusion-based automatic mesh deformation approach, utilizing a pretrained diffusion model to deform a mesh according to the target text description, resulting in a high-quality mesh that matches the text description as well as adheres to the input geometry. Specifically, the mesh deformation is achieved through the optimization on a set of per-face Jacobians, which utilizes Score Distillation Sampling (SDS) to minimize the KL-divergence between the distribution of rendered deformed mesh and the text-conditioned distribution. To address the artifacts caused by the intrinsic ill-posed problem, we incorporated two different approaches into our framework. The first approach involves combining multiple orthogonal views into a single image. As a result, a single-step update concerns information from different perspectives, providing robust deformation at each step while avoiding the need for additional memory. The second approach is intended to address the unsmooth artifacts. This involves incorporating a new regularization based on ARAP, which works to prevent deformation from straying too far from the input mesh while preserving local smoothness. As demonstrated later, this strategy proves to be effective in mitigating the occurrence of intersections and collapses that are induced by large deformations.

Through rigorous experimentation, we have ascertained that our framework is adept at generating smooth, high-quality deformed meshes that resonate closely with textual descriptions. Furthermore, as our method preserves the one-to-one correspondence between the vertices of the input and the resulting meshes, it can facilitate seamless morphing via simple interpolation.

We summarize our contributions as follows:

- We propose FusionDeformer, a novel method for automatic mesh deformation leveraging diffusion prior.
- To achieve robust deformation and avoid additional memory, we develop a multi-view supervision method.
- Our method introduces a new regularization based on ARAP that aims to alleviate the unsmooth artifacts caused by the inherent ill-posed problem.

2 Related Work

2.1 Neural Shape Deformation

Deformation techniques, which are crucial in the realm of computer graphics, have undergone considerable evolution over decades. Many techniques are contingent upon specific models or templates and optimal settings, or intricate pre-processing [41]. Recent neural methods have augmented traditional deformation techniques, including handles, cages, and key points, by incorporating cutting-edge neural networks. Some of them automatically identify 3D key points for shape manipulation [19]. Others utilize neural cages to warp a source shape, aligning it to the broader structure of a target shape but retaining the detailed surface features of the source [47]. Additionally, there are methods that enhance existing linear handle-based subspace models with nonlinear corrections learned from the same subspace [37]. Regularizers such as As-Rigid-As-Possible (ARAP) [14] and the Laplacian [20], are frequently employed in these processes to enhance the overall quality.

There is a strand of research that leverages deep neural networks for the exploration of 3D surfaces. For instance, some approaches employ Variational Autoencoders (VAEs) to delve into the latent space of surfaces [43]. Concurrently, Generative Adversarial Networks (GANs) are leveraged to ensure the derived shapes from specific mappings closely resemble the target shapes [43]. Data-driven strategies also abound for predicting realistic deformations. Some of these methods focus on learning the per-vertex offset derived from diverse mesh datasets [1]. Others empower users to craft geometric deformations anchored by a suite of semantic attributes [51]. There is also interest in methods that assign per-vertex coordinates or offsets based on a foundational model [2,53]. Moreover, using images as references, meshes can be reconstructed by deforming a base mesh [48,46]. While some recent studies propose data-driven approaches to predicting realistic deformations [1,19,47,11,51], their semantic capabilities are limited by a lack of datasets or notations.

Some methods seek to counteract this limitation by incorporating guidance from the powerful visual model CLIP [29,21,9]. To supervise the 3D deformation process at the image-level guidance provided by CLIP, differentiable rendering techniques [24] are utilized to back-propagate gradients from the rendering results to the meshes. With the CLIP model, they achieve text-guided mesh deformation by maximizing the similarity between the text embeddings and the rendered image embeddings through the deformation process. Our objective aligns closely with the CLIP-based methods. However, while their method uses CLIP, our approach employs the diffusion model to drive the deformations of a template shape.

2.2 Text-Guided 3D Synthesis

Generating 3D shapes has long been a challenging task. Numerous research efforts have been devoted to 3D generative modeling, employing various types of 3D representations, such as 3D voxel grids [12,7], point cloud [27,31,49], and implicit representations [6,28]. However, most of these approaches rely on 3D asset datasets, which requires a laborious process of data collection and processing. Thankfully, the usage of differential rendering and large-scale vision-language models like CLIP and diffusion models can potentially eliminate the need for extensive 3D data collection and enable text-guided 3D synthesis.

CLIP [35], which learns a joint embedding space for texts and images, is the foundation of many text-to-3D methods. CLIP-Forge [39] overcomes the lack of a pre-trained counterpart to CLIP for 3D by using renderings of training shapes to bridge the gap between text and 3D data. They first train a voxel encoder and an implicit decoder on available 3D datasets using CLIP image embeddings, then swap image embeddings for text embeddings at inference time. DreamFields [18] leverages Neural Radiance Field (NeRF) [30] to directly optimize views of a 3D shape against a desired text prompt in CLIP's embedding space.

Diffusion models have demonstrated commendable performance in the text-to-image domain [13,36]. To bridge the gap between 2D image generation and 3D content generation, DreamFusion [34] pioneers Score Distillation Sampling (SDS) and optimizes neural implicit representations by distilling knowledge from a

pretrained diffusion models. Its potential in text-to-3D generation has quickly sparked a series of research works. In synergy with DMTets, Magic3D [26] implements a two-stage pipeline. Initially, it generates a sparse 3D hash grid structure using a low-resolution diffusion prior, which is then used as the initial step in the optimization of a textured 3D mesh model using a high-resolution diffusion model. This step-by-step process facilitates the creation of high-quality content. Notably, while earlier methods simultaneously optimized shape and texture, Fantasia3D [5] decouples the generation of geometry and disentangled materials. In addition to general object generation, there is also a considerable amount of task-specific generation work, such as avatars [15,10,4] and human bodies [22,16]. Another line of research focuses on fine-tuning diffusion models to provide more explicit 3D guidance, such as depth [42], orthogonal-view [52,40], and coordinate map [25], leading to outcomes with better view-consistency.

However, both NeRF and DMTets-based methods are unable to perform direct mesh deformation, as they require additional neural implicit representation fitting for initialization. These methods also necessitate further mesh extraction to acquire the final mesh results. This extraction not only leads to the loss of vertex relationship before and after deformation, but it is also a non-trivial process for NeRF due to its density-based representation [45,44,50]. Unlike those methods, our approach deforms the mesh directly, allowing for seamless integration with the existing graphics pipeline.

3 Preliminaries

3.1 Diffusion Model

Diffusion models [13] are generative models that learn to gradually transform a sample from a tractable noise distribution towards a data distribution $x_0 \sim q_0(x)$. The forward pass follows the Markov Chain to gradually add noise to the input image x_0 towards a Gaussian noise $\mathcal{N}(0,1)$. At each step in the forward process, the diffused image x_t is computed by adding noise with variance β_t to the previous image x_{t-1} as:

$$x_t \sim \mathcal{N}(\sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}). \tag{1}$$

Given x_0, we obtain the diffused image x_t according to the independence property of the Markov Chain:

$$\begin{aligned} x_t &\sim \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I}) \\ x_t &= \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon. \end{aligned} \tag{2}$$

Here, $\bar{\alpha}_t$ is the total noise variance at step t, defined as $\bar{\alpha}_t = \prod_{s=0}^{t}\alpha_s$, wherein $\alpha_t = 1-\beta_t$.

At the reverse denoising process, the estimation $\hat{x}_{t-1}$ for the next step is acquired by predicting the mean, $\mu_\theta(x_t,t)$, and the covariance, $\sigma_\theta(x_t,t)$, of x_{t-1} with x_t serving as the input, then $\hat{x}_{t-1}$ is sampled from the normal distribution defined by the predicted parameters:

$$\hat{x}_{t-1} \sim \mathcal{N}(\mu_\theta(x_t,t), \sigma_\theta(x_t,t)). \tag{3}$$

Instead of directly predicting $\mu_\theta(x_t,t)$, [13] advocates for the use of a network $\hat{\epsilon}_\phi(x_t,t)$ to predict the noise ϵ added to x_0. Then $\mu_\theta(x_t,t)$ is computed utilizing Bayes' theorem:

$$\mu_\theta(x_t,t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\hat{\epsilon}_\phi(x_t,t)\right). \tag{4}$$

The covariance $\sigma_\theta(x_t,t)$ can either be maintained as constant, as suggested by [13], or learned through a neural network, as proposed by [32].

In a conditional generation process, such as text-to-image diffusion models, a provided text prompt y also functions as the input for the neural network as $\hat{\epsilon}_\phi(x_t;y,t)$.

To reduce computational expenses, the latent diffusion model [36] extends the application of the diffusion model to the latent space of a pre-trained autoencoder. Specifically, the image x_0 in RGB space is replaced by the latent $z_0 = \mathcal{E}(x_0)$ from the encoder $\mathcal{E}$. For text-guided latent diffusion models like Stable Diffusion [36], the noise prediction is stated as $\hat{\epsilon}_\phi(z_t;y,t)$.

3.2 Score Distillation Sampling

Score Distillation Sampling (SDS) was first introduced in DreamFusion [34]. This technique aids in generating 3D content (represented by Mip-NeRF [3]) that aligns with the provided input text prompt under the supervision of a pretrained diffusion model. The optimization of the Mip-NeRF g with parameters θ starts with adding noise ϵ to the rendered image $\mathbf{x}_0 = g(\theta)$ at a randomly sampled noise level t, getting diffused image $\mathbf{x}_t$. Then SDS is applied to calculate the gradient by computing the difference between the predicted text-conditioned noise $\hat{\epsilon}_\phi(\mathbf{x}_t;y,t)$ and the injected noise ϵ as follows:

$$\nabla_\theta\mathcal{L}_{\text{SDS}}(\phi,\mathbf{x}_0) \triangleq \mathbb{E}_{t,\epsilon}\left[\omega(t)\left(\hat{\epsilon}_\phi(\mathbf{x}_t;y,t)-\epsilon\right)\frac{\partial\mathbf{x}_0}{\partial\theta}\right], \tag{5}$$

where $\omega(t)$ is a weighting function which absorbs the constant $\sqrt{\bar{\alpha}_t}\mathbf{I} = \partial\mathbf{x}_t/\partial\mathbf{x}_0$. By applying SDS loss, θ is optimized to render images that align with the distribution learned by the pretrained diffusion model.

Instead of relying on the inaccessible Imagen model [38] employed by DreamFusion, our approach utilizes the open-source latent diffusion model Stable Diffusion [36] through the Diffusers library [33].

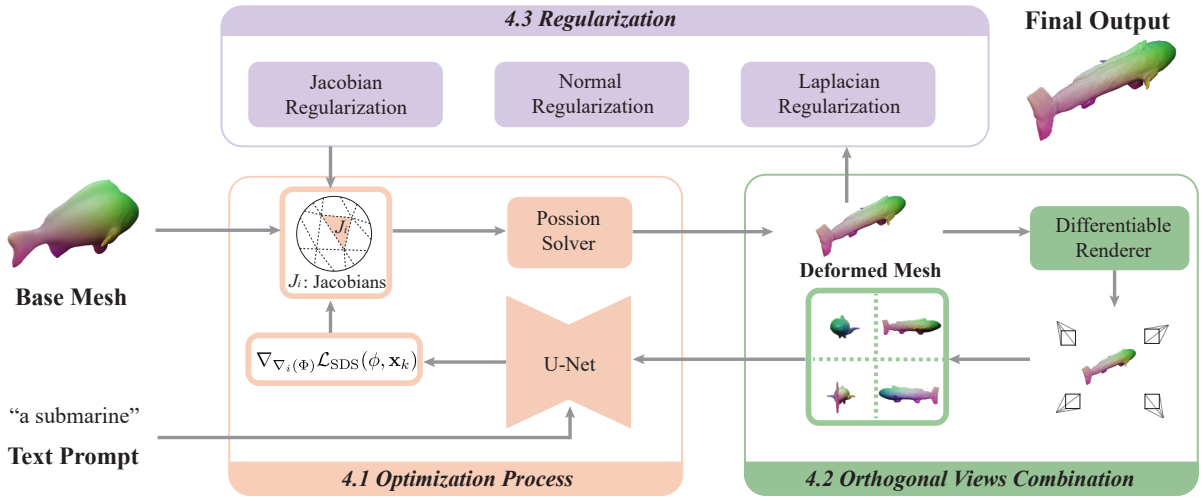

Fig. 2 The **overview** of our pipeline. Given the input base mesh, we initialize the per-face Jacobians as identity matrices (implying no deformation). These Jacobians are subsequently processed by a Poisson solver to compute the deformation map, resulting in an intermediate deformed mesh. This intermediate mesh is then rendered from four orthogonal views, which are concatenated into a 2x2 grid and merged to form a single image. Following this, we employ the Score Distillation Sampling (SDS) on this grid image to compute the gradients necessary for updating the Jacobians. Furthermore, regularization terms are applied to both the Jacobians and the deformed mesh.

4 Methodology

In this section, we first introduce the indirect mesh deformation strategy, which results in smoother and more stable deformations compared to the direct displacement of vertices. This strategy optimizes per-face Jacobians under the guidance of SDS loss (Sec. 4.1). Then, we delve into our innovative orthogonal-views combination strategy (Sec. 4.2), which leads to fewer global distortions in the resulting meshes. Finally, we explore mesh regularization (Sec. 4.3), with the goal of preventing the deformation from straying too far from the input mesh and mitigating the occurrence of intersections and collapses.

4.1 Optimization Process

A mesh $\mathcal{M}$ is composed of a set of vertices $\mathcal{V}$ and faces $\mathcal{F}$. Instead of directly optimizing vertices $\mathcal{V}$ coordinates as CLIP-Mesh [21] does, we employ the Neural Jacobian Field [1] to indirectly deform the mesh as the TextDeformer [9] suggested. It is proven to produce smoother and larger deformations and preserve the connection at the same time. Specifically, the shape is parameterized using a collection of per-triangle Jacobians, which serve to describe deformations. To obtain the deformation map Φ, we resolve the Poisson problem [1] as:

$$\Phi^* = \min_{\Phi} \sum_{f_i \in \mathcal{F}} |f_i| \left\| \nabla_i(\Phi) - J_i \right\|_2^2, \tag{6}$$

where $\nabla_i(\Phi)$ denotes the Jacobian of the deformation map Φ at triangle f_i, $|f_i|$ denotes the area of the triangle. Hence the optimization of mesh can be achieved by optimizing the per-face Jacobian $\nabla_i(\Phi)$.

At the k^{th} step of the mesh optimization process, the intermediate mesh is denoted as $\mathcal{M}_k$. Utilizing a sampled camera poses c, $\mathcal{M}_k$ is rendered against an arbitrary background color, yielding the resulting image $\mathbf{x}_k = R(\mathcal{M}_k, c)$, where R denotes a differentiable renderer [24]. We then feed $\mathbf{x}_k$ into encoder $\mathcal{E}$, getting the latent $\mathbf{z}_0 = \mathcal{E}(\mathbf{x}_k)$, where the subscript 0 in $\mathbf{z}_0$ implying that the latent is devoid of noise. Subsequently, noise is introduced to $\mathbf{z}_0$ to noise level t, producing the noised latent $\mathbf{z}_t$. Then we apply SDS to compute the gradient which is used in the Jacobians updating as:

$$\begin{aligned} &\nabla_{\nabla_i(\Phi)}\mathcal{L}_{\text{SDS}}(\phi, \mathbf{x}_k) \\ &\triangleq \mathbb{E}_{t,\epsilon}\left[\omega(t)\left(\hat{\epsilon}_\phi\left(\mathbf{z}_t; y, t\right) - \epsilon\right)\frac{\partial \mathbf{z}_0}{\partial \mathbf{x}_k}\frac{\partial \mathbf{x}_k}{\partial \nabla_i(\Phi)}\right], \end{aligned} \tag{7}$$

where $\omega(t)$ is a weighting function, which absorbs the constant $\sqrt{\bar{a}_t}\mathbf{I} = \partial \mathbf{z}_t / \partial \mathbf{z}_0$.

4.2 Orthogonal Views Combination

Being a 2D generative model, the Stable Diffusion model falls short in providing the required 3D prior guidance imperative for achieving realistic geometry. This shortcoming becomes particularly evident when only part of the faces of the mesh are visible in the resulting renderings, leading to an under-determined issue when

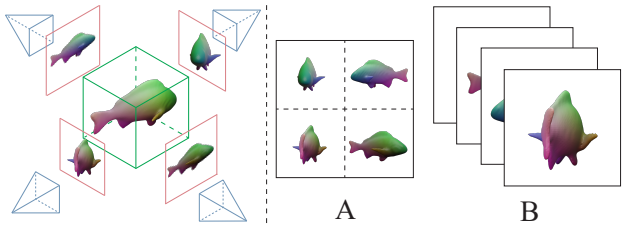

Fig. 3 On the left we show the orthogonal rendering views setup, the cameras are rotating along the y-axis. On the right are two ways to organize our rendering results. A) Concatenate 4 images into a 2×2 grid. B) Feed the images into the Diffusion Model respectively. We take the A scheme because it saves the running time memory with the number of faces seen and offers more coherence.

characterizing the geometry of the invisible area. Consequently, significant distortions ensue within the final meshes (see Sec. 5.3).

Previous CLIP-based methods [21,9] include rendering results from several camera viewpoints within a single batch to facilitate multi-view guidance. However, given the high computational demand of the diffusion model, this strategy could prove to be excessively pricey.

To attain more realistic geometry with less distortion, while circumventing the need for additional memory, we adopt an Orthogonal Views Combination (OVC) strategy, inspired by [45].

Specifically, we first sample an azimuth angle $a \in [-180°, 180°]$, and then render four orthogonal views of the mesh using azimuth angles $a_0 = a, a_1 = a + 90°, a_2 = a + 180°, a_3 = a + 270°$. The elevation angles of the four views are randomly sampled from the range $[0°, 30°]$. Consequently, we obtain $\{\mathbf{x}_k^0, \mathbf{x}_k^1, \mathbf{x}_k^2, \mathbf{x}_k^3\}$, which gets tiled on a 2×2 grid and merged into a single image as:

$$\mathbf{x}_k^{tiled} = \text{Grid}(\mathbf{x}_k^0, \mathbf{x}_k^1, \mathbf{x}_k^2, \mathbf{x}_k^3), \tag{8}$$

where Grid is a function to concatenate four images into a 2×2 grid.

Subsequently, we utilize $\mathbf{x}_k^{tiled}$ as a substitute for $\mathbf{x}_k$, and then apply SDS on $\mathbf{x}_k^{tiled}$ to compute gradients as discussed in 4.1. The single-step update featured in our OVC strategy accounts for information from different perspectives, leading to multi-view guidance and robust deformation.

4.3 Regularization

We observe that large deformations within the results induced problems such as intersections and collapses (see 5.3). To prevent the deformation from straying too far from the input mesh, we adopt an As-Rigid-As-Possible (ARAP) regularization strategy [14]. The ARAP regularization comprises two distinct components: Jacobian regularization and mesh smooth regularization, which facilitate a more precise and controlled deformation process, ensuring that the changes remain faithful to the initial structure while allowing for necessary transformations.

Regarding the Jacobian regularization component, we penalize the difference between the updated per-face Jacobians and the identity matrices, which represent the case of no deformation. Inspired by TextDeformer [9], we compute the Jacobian regularization as:

$$L_{Jacobian} = \alpha \sum_{i=1}^{|\mathcal{F}|} ||J_i - I||_2. \tag{9}$$

For the mesh smooth regularization component, we incorporate two geometry regularization terms, the Laplacian term and the normal consistency term, as proposed in NDS [48]:

$$L_{smooth} = \lambda_{Laplacian} L_{Laplacian} + \lambda_{normal} L_{normal}, \tag{10}$$

where $\lambda_{Laplacian}$ and λ_{normal} denote the weights of the loss terms.

The Laplacian loss term is defined as follows:

$$\begin{aligned} L_{Laplacian} &= \frac{1}{n} \sum_{i=1}^{n} ||\delta_i||_2^2 \\ \delta_i &= (\mathcal{L}\mathcal{V})_i \in \mathcal{R}^3, \end{aligned} \tag{11}$$

where δ_i is the differential coordinates of the i^{th} vertex, $\mathcal{V}$ and $\mathcal{L}$ denote the vertices set and the graph Laplacian of $\mathcal{M}$ respectively. This term minimizes the distance of each vertex from the average position of its neighboring vertices.

The normal consistency loss term is defined as follows:

$$L_{normal} = \frac{1}{|\bar{\mathcal{F}}|} \sum_{(i,j) \in \bar{\mathcal{F}}} (1 - \mathbf{n}_i \mathbf{n}_j)^2, \tag{12}$$

where $\bar{\mathcal{F}}$ is the set of the adjacent triangles, $\mathbf{n}_i$ is the normal of the i^{th} triangle, and $\mathbf{n}_j$ is the normal of the j^{th} triangle. It aims to maximize the cosine similarity between the normals of the adjacent triangles.

The total regularization is defined as follows:

$$L_{ARAP} = L_{Jacobian} + L_{smooth}. \tag{13}$$

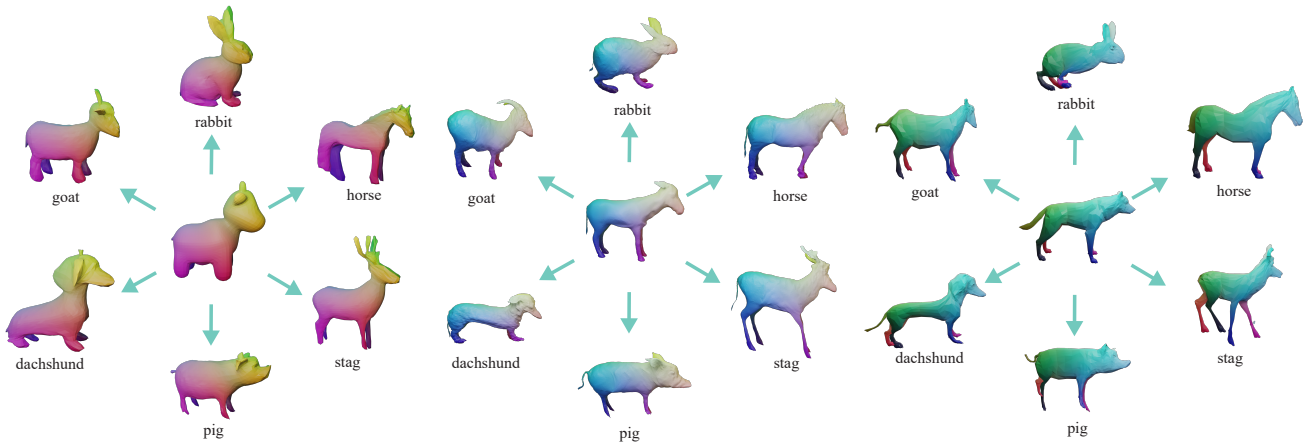

Fig. 4 Deforming a single mesh using distinctive input prompts. For each case, the source meshes are centrally positioned. Our methodology proves its capability to transform the same source mesh into diverse targets using varying prompts. Additionally, even with identical prompts, the outcomes of the deformations are influenced by the characteristics of the input mesh.

5 Experiments

5.1 Implementation Details

Our experiments are conducted on a system equipped with an Intel Core i9-13900k processor, 64GB of RAM, and an RTX 4090 graphics card. Notably, our method requires only a graphics card with 12GB of memory. We use Nvdiffrast [24] as our renderer, which is a deep-learning library that provides high-performance primitive operations for rasterization-based differentiable rendering. The learning rate is set to 0.0025. The version of Stable Diffusion we use is stable-diffusion-2-1-base, which is fine-tuned on the stable-diffusion-2-base with 220k extra steps taken on the same dataset. All meshes employed in our experiments range between 1,000 and 20,000 triangles.

5.2 Main Results

Text-guided deformation. FusionDeformer shows versatility in handling a broad spectrum of target prompts, as illustrated in Fig. 5. For each case, the source mesh is displayed in the top left corner. Our method not only deforms the source mesh in accordance with the input prompts, but also improves the realism of the geometry by incorporating intricate details. Additionally, our method has the ability to reshape faces into specific identities, such as *"Trump"*.

Fig. 4 showcases three instances of deforming a single mesh from distinctive input prompts using our method. For each case, the source meshes are centrally positioned. FusionDeformer demonstrates its generalization to deform the same source mesh into diverse targets using different prompts. Crucially, while the deformation results accurately align with the prompts, they are also affected by the specific features of the input mesh.

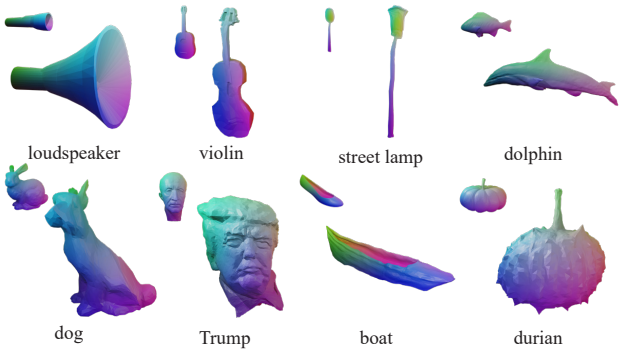

Fig. 5 Text-driven Deformation. Our method is capable of producing a wide range of results, influenced both by the characteristics of the input mesh (displayed in the top left corner) and the specific input text prompts (listed below).

Method	User Preference ↑
Ours	73.04%
TextDeformer [9]	18.91%
CLIP-Mesh [21]	8.05%

Table 1 User Study. The user study's distribution evaluates the quality of deformation results from different methods. It indicates that the results of our method are most preferred by user study participants.

Qualitative Comparisons. In Fig. 6, we conduct a qualitative comparison of our FusionDeformer method against two SOTA text-based mesh deformation techniques, CLIP-Mesh [21] and TextDeformer [9]. We present the comparison using five distinct target text prompts: *"shark"*, *"Albert Einstein"*, *"giraffe"*, *"submarine"*, *"rabbit"*. The results indicate that CLIP-Mesh tends to produce models with sub-optimal surface quality, characterized by floating faces. TextDeformer produces results with improved surface quality but still lacks realistic de-

Fig. 6 Qualitative Comparison. From left to right: the source mesh, CLIP-Mesh [21], TextDeformer [9] and our method. We present the comparison using five distinct target text prompts: *"shark"*, *"Albert Einstein"*, *"giraffe"*, *"submarine"*, *"rabbit"*. The results generated by our method are the best aligned with the target text prompts and possess the highest quality.

tails and fails to align well with the input prompt (evidenced by the fish-like *"submarine"* and the coarse geometry of *"Einstein"*). In contrast, our FusionDeformer yields results with less distortion, better aligning with the input prompt, and improved realistic details (see the gill of *"shark"*). Notably, while CLIP-Mesh and TextDeformer incorporate 25 multi-view results in a single batch to achieve multi-view guidance, our FusionDeformer achieves better results using only 4 views in a single batch, thanks to our carefully designed framework.

Runtime Performance. In regards to runtime performance, FusionDeformer accomplishes a 5,000-step deformation process in merely 15 minutes on a single RTX 4090 GPU. In contrast, TextDeformer [9] requires 30 minutes to complete the same task on the same machine. This shows FusionDeformer's superior efficiency over TextDeformer.

User Study. CLIP-Mesh [21] and TextDeformer [9] leverage the CLIP R-Precision benchmark to evaluate the semantic discrepancies between the input text and generated meshes. However, as both these methods employ CLIP-based text-image similarity for their training objectives, this evaluation approach could result in an unfair comparison. As such, we evaluate whether our method and SOTA methods meet human expectations by conducting a user study. We presented 20 groups of mesh rendering results in a random sequence to 23 participants. Each group consisted of the target text prompt and three mesh rendering results, encompassing those generated by our method, TextDeformer, and CLIP-Mesh, presented in a random order. Participants were asked to select the best result that matched the text prompt while also demonstrating the highest realism. As displayed in 1, our results were most preferred by the participants, indicating a positive reception of our method's effectiveness in corresponding with text prompts and attaining realistic outcomes.

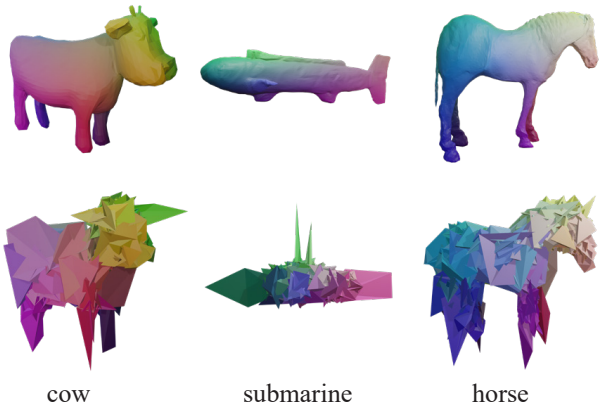

Fig. 7 Jacobians Ablation. The first row displays results produced via Jacobians, and the second row shows results generated by direct vertex displacement. Deformation based on Jacobians exhibits a more global impact and results in better surface quality.

5.3 Ablation Study

In this section, we conducted ablation studies to evaluate the effectiveness of the diverse components in our method.

Jacobians. As mentioned in 4.1, the Jacobians remain integral to our deformation process. Fig. 7 contrasts results achieved through Jacobian-based deformation (the 1st row) with those obtained via direct vertex displacement (the 2nd row). The direct vertex displacement reduces the surface quality in all three examples, underscoring the importance of Jacobians in our deformation method.

Orthogonal views combination. To illustrate that the Orthogonal Views Combination strategy results in fewer global distortions within the resulting meshes, we

Fig. 8 Orthogonal-Views Combination Ablation. The results are rendered from the same camera pose. We compare the results of using the Orthogonal-Views Combination (OVC) technique with three alternative foundational experiments: using four orthogonal views within a batch, four random views within a batch, and one random view within a batch. It can be found that the method with orthogonal views combination can produce results with less distortion. Additionally, it significantly outpaces the two aforementioned methods that employ the utilization of four images in a batch, in terms of speed.

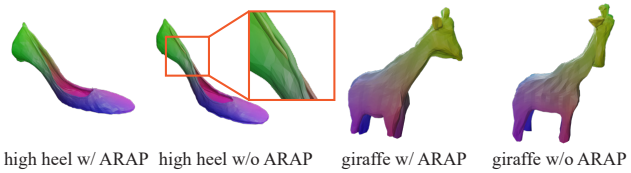

Fig. 9 Regularization Ablation. For each case, the left result uses regularization, while the right one does not. The regularization mitigates the occurrence of intersections and collapse provoked by large deformations. The self-intersections of the high heel are highlighted by the orange box.

conduct three baseline experiments. The first experiment computed SDS by combining the results of four randomly sampled orthogonal-view renderings. The second experiment used four randomly sampled renderings in a batch, whereas the third used one randomly sampled view. As illustrated in Fig. 8, it is evident that the method employing orthogonal view combination yields representations of *"Biden"* with considerably less distortion. Additionally, it outperforms the two methods that use four images in a batch in terms of speed.

Regularization. To demonstrate the effectiveness of the regularization terms, we draw a comparison between results with and without the implementation of regularization, as illustrated in Fig. 9. For example, in the transformation from *"shoe"* to *"high heel"*, the lack of regularization results in self-intersections (highlighted by the orange box). In the case of transformation from *"cow"* to *"giraffe"*, the structure of the head collapses. In sum, the regularization terms are critical in more effectively mitigating the occurrence of intersections and collapse provoked by large deformations.

5.4 Application

Morphing. Our deformation method maintains the one-to-one correspondence between mesh vertices, a feature that allows for morphing without additional specifications. This property facilitates the use of 3D processing software such as MAYA to effortlessly generate morphing targets. Consequently, our approach holds significant potential for applications in computer animation, particularly in character transformations, blend shapes and so on.

6 Conclusion and Future Work

We introduce FusionDeformer, a novel framework to automatically deform mesh according to text prompts via diffusion model. To compute the gradients of the mesh optimization, we utilize per-face Jacobians update and Score Distillation Sampling (SDS). We further introduce a multi-view supervision method to achieve robust deformation and avoid additional memory usage, as well as a new regularization to alleviate the unsmooth artifacts. Our innovative method delivers results that well align with the input prompts, exhibits realistic details, and retains meaningful correspondences between the original and deformed shapes.

Our method has limitations. It currently focuses solely on geometry, leading to the meshes trying to replicate target-specific features such as hair or speckles. In the future, texture integration could be used to separate geometry and appearance to address this problem. Furthermore, per-face Jacobians focus on global deformation and are unable to accomplish local editing. Future advancements may incorporate more refined control mechanisms for text-guided editing, thereby enhancing the precision and versatility of our method.

7 Acknowledgements

This work was supported by Key R&D Program of Zhejiang (No. 2023C01047).

References

1. Aigerman, N., Gupta, K., Kim, V.G., Chaudhuri, S., Saito, J., Groueix, T.: Neural jacobian fields: learning intrinsic mappings of arbitrary meshes. ACM Trans. Graph. **41**(4), 109:1–109:17 (2022)
2. Bailey, S.W., Omens, D., Dilorenzo, P., O'Brien, J.F.: Fast and deep facial deformations. ACM Trans. Graph. **39**(4) (2020)
3. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021, pp. 5835–5844 (2021)

4. Cao, Y., Cao, Y.P., Han, K., Shan, Y., Wong, K.Y.K.: Guide3D: Create 3D Avatars from Text and Image Guidance. arXiv preprint arXiv:2308.09705 (2023)
5. Chen, R., Chen, Y., Jiao, N., Jia, K.: Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 22,246–22,256 (2023)
6. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, pp. 5939–5948 (2019)
7. Gadelha, M., Maji, S., Wang, R.: 3d shape induction from 2d views of multiple objects. In: 2017 International Conference on 3D Vision, 3DV 2017, Qingdao, China, October 10-12, 2017, pp. 402–411 (2017)
8. Gal, R., Sorkine, O., Mitra, N.J., Cohen-Or, D.: iwires: an analyze-and-edit approach to shape manipulation. ACM Trans. Graph. **28**(3), 33 (2009)
9. Gao, W., Aigerman, N., Groueix, T., Kim, V., Hanocka, R.: Textdeformer: Geometry manipulation using text guidance. In: ACM SIGGRAPH 2023 Conference Proceedings, SIGGRAPH 2023, Los Angeles, CA, USA, August 6-10, 2023, pp. 82:1–82:11 (2023)
10. Han, X., Cao, Y., Han, K., Zhu, X., Deng, J., Song, Y.Z., Xiang, T., Wong, K.Y.K.: Headsculpt: Crafting 3d head avatars with text. Advances in Neural Information Processing Systems **36** (2024)
11. Hanocka, R., Fish, N., Wang, Z., Giryes, R., Fleishman, S., Cohen-Or, D.: Alignet: Partial-shape agnostic alignment via unsupervised learning. ACM Trans. Graph. **38**(1), 1:1–1:14 (2019). DOI 10.1145/3267347. URL https://doi.org/10.1145/3267347
12. Henzler, P., Mitra, N.J., Ritschel, T.: Escaping plato's cave: 3d shape from adversarial rendering. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019, pp. 9983–9992 (2019)
13. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, pp. 6840–6851 (2020)
14. Huang, Q., Huang, X., Sun, B., Zhang, Z., Jiang, J., Bajaj, C.: Arapreg: An as-rigid-as possible regularization loss for learning deformable shape generators. In: 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021, pp. 5795–5805 (2021)
15. Huang, S., Yang, Z., Li, L., Yang, Y., Jia, J.: Avatarfusion: Zero-shot generation of clothing-decoupled 3d avatars using 2d diffusion. In: Proceedings of the 31st ACM International Conference on Multimedia, MM 2023, Ottawa, ON, Canada, 29 October 2023- 3 November 2023, pp. 5734–5745 (2023)
16. Huang, Y., Yi, H., Xiu, Y., Liao, T., Tang, J., Cai, D., Thies, J.: TeCH: Text-guided Reconstruction of Lifelike Clothed Humans. In: International Conference on 3D Vision (3DV) (2024)
17. Jacobson, A.: Algorithms and interfaces for real-time deformation of 2d and 3d shapes. Ph.D. thesis, ETH Zurich (2013)
18. Jain, A., Mildenhall, B., Barron, J.T., Abbeel, P., Poole, B.: Zero-shot text-guided object generation with dream fields. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022, pp. 857–866 (2022)
19. Jakab, T., Tucker, R., Makadia, A., Wu, J., Snavely, N., Kanazawa, A.: Keypointdeformer: Unsupervised 3d keypoint discovery for shape control. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021, pp. 12,783–12,792 (2021)
20. Kanazawa, A., Tulsiani, S., Efros, A.A., Malik, J.: Learning category-specific mesh reconstruction from image collections. In: Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XV, *Lecture Notes in Computer Science*, vol. 11219, pp. 386–402 (2018)
21. Khalid, N.M., Xie, T., Belilovsky, E., Popa, T.: Clip-mesh: Generating textured meshes from text using pretrained image-text models. In: SIGGRAPH Asia 2022 Conference Papers, SA 2022, Daegu, Republic of Korea, December 6-9, 2022, pp. 25:1–25:8 (2022)
22. Kim, B., Kwon, P., Lee, K., Lee, M., Han, S., Kim, D., Joo, H.: Chupa: Carving 3d clothed humans from skinned shape priors using 2d diffusion probabilistic models. In: IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023, pp. 15,919–15,930 (2023)
23. Kraevoy, V., Sheffer, A.: Cross-parameterization and compatible remeshing of 3d models. ACM Trans. Graph. **23**(3), 861–869 (2004)
24. Laine, S., Hellsten, J., Karras, T., Seol, Y., Lehtinen, J., Aila, T.: Modular primitives for high-performance differentiable rendering. ACM Trans. Graph. **39**(6), 194:1–194:14 (2020)
25. Li, W., Chen, R., Chen, X., Tan, P.: Sweetdreamer: Aligning geometric priors in 2d diffusion for consistent text-to-3d. arXiv preprint arXiv:2310.02596 (2023)
26. Lin, C., Gao, J., Tang, L., Takikawa, T., Zeng, X., Huang, X., Kreis, K., Fidler, S., Liu, M., Lin, T.: Magic3d: High-resolution text-to-3d content creation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023, pp. 300–309 (2023)
27. Luo, S., Hu, W.: Diffusion probabilistic models for 3d point cloud generation. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021, pp. 2837–2845 (2021)
28. Mescheder, L.M., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, pp. 4460–4470 (2019)
29. Michel, O., Bar-On, R., Liu, R., Benaim, S., Hanocka, R.: Text2mesh: Text-driven neural stylization for meshes. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022, pp. 13,482–13,492 (2022)
30. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I, *Lecture Notes in Computer Science*, vol. 12346, pp. 405–421 (2020)
31. Mo, K., Guerrero, P., Yi, L., Su, H., Wonka, P., Mitra, N.J., Guibas, L.J.: Structurenet: hierarchical graph networks for 3d shape generation. ACM Trans. Graph. **38**(6), 242:1–242:19 (2019)
32. Nichol, A.Q., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: Proceedings of the 38th International Conference on Machine Learning, ICML 2021,

18-24 July 2021, Virtual Event, *Proceedings of Machine Learning Research*, vol. 139, pp. 8162–8171 (2021)
33. von Platen, P., Patil, S., Lozhkov, A., Cuenca, P., Lambert, N., Rasul, K., Davaadorj, M., Wolf, T.: Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers (2022)
34. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. In: The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023 (2023)
35. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event, *Proceedings of Machine Learning Research*, vol. 139, pp. 8748–8763 (2021)
36. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022, pp. 10,674–10,685 (2022)
37. Romero, C., Casas, D., Pérez, J., Otaduy, M.A.: Learning contact corrections for handle-based subspace dynamics. ACM Trans. Graph. **40**(4), 131:1–131:12 (2021)
38. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, S.K.S., Lopes, R.G., Ayan, B.K., Salimans, T., Ho, J., Fleet, D.J., Norouzi, M.: Photorealistic text-to-image diffusion models with deep language understanding. In: Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022, pp. 36,479–36,494 (2022)
39. Sanghi, A., Chu, H., Lambourne, J.G., Wang, Y., Cheng, C., Fumero, M., Malekshan, K.R.: Clip-forge: Towards zero-shot text-to-shape generation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022, pp. 18,582–18,592 (2022)
40. Shi, Y., Wang, P., Ye, J., Long, M., Li, K., Yang, X.: Mvdream: Multi-view diffusion for 3d generation. arXiv preprint arXiv:2308.16512 (2023)
41. Sorkine, O., Botsch, M.: Interactive shape modeling and deformation. In: Eurographics (Tutorials), pp. 11–37 (2009)
42. Stan, G.B.M., Wofk, D., Fox, S., Redden, A., Saxton, W., Yu, J., Aflalo, E., Tseng, S.Y., Nonato, F., Muller, M., et al.: Ldm3d: Latent diffusion model for 3d. arXiv preprint arXiv:2305.10853 (2023)
43. Tan, Q., Gao, L., Lai, Y., Xia, S.: Variational autoencoders for deforming 3d mesh models. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, pp. 5841–5850 (2018)
44. Tang, J., Zhou, H., Chen, X., Hu, T., Ding, E., Wang, J., Zeng, G.: Delicate textured mesh recovery from nerf via adaptive surface refinement. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 17,739–17,749 (2023)
45. Tsalicoglou, C., Manhardt, F., Tonioni, A., Niemeyer, M., Tombari, F.: Textmesh: Generation of realistic 3d meshes from text prompts. arXiv preprint arXiv:2304.12439 (2023)
46. Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, Y.: Pixel2mesh: Generating 3d mesh models from single RGB images. In: Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XI, *Lecture Notes in Computer Science*, vol. 11215, pp. 55–71 (2018)
47. Wang, Y., Aigerman, N., Kim, V.G., Chaudhuri, S., Sorkine-Hornung, O.: Neural cages for detail-preserving 3d deformations. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020, pp. 72–80 (2020)
48. Worchel, M., Diaz, R., Hu, W., Schreer, O., Feldmann, I., Eisert, P.: Multi-view mesh reconstruction with neural deferred shading. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022, pp. 6177–6187 (2022)
49. Yang, G., Huang, X., Hao, Z., Liu, M., Belongie, S.J., Hariharan, B.: Pointflow: 3d point cloud generation with continuous normalizing flows. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019, pp. 4540–4549 (2019)
50. Yariv, L., Gu, J., Kasten, Y., Lipman, Y.: Volume rendering of neural implicit surfaces. In: Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, pp. 4805–4815 (2021)
51. Yümer, M.E., Chaudhuri, S., Hodgins, J.K., Kara, L.B.: Semantic shape editing using deformation handles. ACM Trans. Graph. **34**(4), 86:1–86:12 (2015)
52. Zhao, M., Zhao, C., Liang, X., Li, L., Zhao, Z., Hu, Z., Fan, C., Yu, X.: Efficientdreamer: High-fidelity and robust 3d creation via orthogonal-view diffusion prior. arXiv preprint arXiv:2308.13223 (2023)
53. Zheng, M., Zhou, Y., Ceylan, D., Barbic, J.: A deep emulator for secondary motion of 3d characters. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021, pp. 5932–5940 (2021)